LatticeECP2/M Family Handbook

HB1003 Version 05.3, February 2012

LatticeECP2/M Family Handbook Table of Contents

February 2012

Section I. LatticeECP2/M Family Data Sheet

Introduction

Architecture

Section I. LatticeECP2/M Family Data Sheet

Version 03.9, January 2011

LatticeECP2/M Family Data Sheet Introduction

January 2012 Data Sheet DS1006

Features

- **High Logic Density for System Integration**
  - 6K to 95K LUTs
  - 90 to 583 I/Os
- **Embedded SERDES (LatticeECP2M Only)**
  - Data Rates 250 Mbps to 3.125 Gbps
  - Up to 16 channels per device
    PCI Express, Ethernet (1GbE, SGMII), OBSAI, CPRI and Serial RapidIO.
- **sysDSP™ Block**
  - 3 to 42 blocks for high performance multiply and accumulate
  - Each block supports
    - One 36x36, four 18x18 or eight 9x9 multipliers
- **Flexible Memory Resources**
  - 55Kbits to 5308Kbits sysMEM™ Embedded Block RAM (EBR)
    - 18Kbit block
    - Single, pseudo dual and true dual port
    - Byte Enable Mode support
  - 12K to 202Kbits distributed RAM
    - Single port and pseudo dual port
- **sysCLOCK Analog PLLs and DLLs**
  - Two GPLLs and up to six SPLLs per device
    - Clock multiply, divide, phase & delay adjust
    - Dynamic PLL adjustment
  - Two general purpose DLLs per device
- **Pre-Engineered Source Synchronous I/O**
  - DDR registers in I/O cells
  - Dedicated gearing logic
  - Source synchronous standards support
    - SPI4.2, SFI4 (DDR Mode), XGMII
    - High Speed ADC/DAC devices
  - Dedicated DDR and DDR2 memory support
    - DDR1: 400 (200MHz) / DDR2: 533 (266MHz)
  - Dedicated DQS support
- **Programmable sysI/O™ Buffer Supports Wide Range Of Interfaces**
  - LVTTL and LVCMOS 33/25/18/15/12
  - SSTL 3/2/18 I, II
  - HSTL15 I and HSTL18 I, II
  - PCI and Differential HSTL, SSTL
  - LVDS, RSDS, Bus-LVDS, MLVDS, LVPECL
- **Flexible Device Configuration**
  - 1149.1 Boundary Scan compliant
  - Dedicated bank for configuration I/Os
  - SPI boot flash interface
  - Dual boot images supported
  - TransFR™ I/O for simple field updates
  - Soft Error Detect macro embedded
- **Optional Bitstream Encryption (LatticeECP2/M "S" Versions Only)**
- **System Level Support**
  - ispTRACY™ internal logic analyzer capability
  - On-chip oscillator for initialization & general use
  - 1.2V power supply

Table 1-1. LatticeECP2 (Including "S-Series") Family Selection

Device	ECP2-6	ECP2-12	ECP2-20	ECP2-35	ECP2-50	ECP2-70
LUTs (K)	6	12	21	32	48	68
Distributed RAM (Kbits)	12	24	42	64	96	136
EBR SRAM (Kbits)	55	221	276	332	387	1032
EBR SRAM Blocks	3	12	15	18	21	60
sysDSP Blocks	3	6	7	8	18	22
18x18 Multipliers	12	24	28	32	72	88
GPLL + SPLL + DLL	2+0+2	2+0+2	2+0+2	2+0+2	2+2+2	2+4+2
Maximum Available I/O	190	297	402	450	500	583
Packages and I/O Combinations						
144-pin TQFP (20 x 20 mm)	90	93				
208-pin PQFP (28 x 28 mm)		131	131			
256-ball fpBGA (17 x 17 mm)	190	193	193			
484-ball fpBGA (23 x 23 mm)		297	331	331	339	
672-ball fpBGA (27 x 27 mm)			402	450	500	500
900-ball fpBGA (31 x 31 mm)						583

Table 1-2. LatticeECP2M (Including "S-Series") Family Selection

Device	ECP2M20	ECP2M35	ECP2M50	ECP2M70	ECP2M100
LUTs (K)	19	34	48	67	95
sysMEM Blocks (18kb)	66	114	225	246	288
Embedded Memory (Kbits)	1217	2101	4147	4534	5308
Distributed Memory (Kbits)	41	71	101	145	202
sysDSP Blocks	6	8	22	24	42
18x18 Multipliers	24	32	88	96	168
GPLL+SPLL+DLL	2+6+2	2+6+2	2+6+2	2+6+2	2+6+2
Maximum Available I/O	304	410	410	436	520
Packages and SERDES / I/O Combinations					
256-ball fpBGA (17 x 17 mm)	4 / 140	4 / 140			
484-ball fpBGA (23 x 23 mm)	4 / 304	4 / 303	4 / 270		
672-ball fpBGA (27 x 27 mm)		4 / 410	8 / 372		
900-ball fpBGA (31 x 31 mm)			8 / 410	16 / 416	16 / 416
1152-ball fpBGA (35 x 35 mm)				16 / 436	16 / 520

Introduction

The LatticeECP2/M family of FPGA devices is optimized to deliver high performance features such as advanced DSP blocks, high speed SERDES (LatticeECP2M family only) and high speed source synchronous interfaces in an economical FPGA fabric. This combination was achieved through advances in device architecture and the use of 90nm technology.

The LatticeECP2/M FPGA fabric is optimized with high performance and low cost in mind. The LatticeECP2/M devices include LUT-based logic, distributed and embedded memory, Phase Locked Loops (PLLs), Delay Locked Loops (DLLs), pre-engineered source synchronous I/O support, enhanced sysDSP blocks and advanced configuration support, including encryption ("S" versions only) and dual boot capabilities.

The LatticeECP2M device family features high speed SERDES with PCS. These high jitter tolerance and low transmission jitter SERDES with PCS blocks can be configured to support an array of popular data protocols including PCI Express, Ethernet (1GbE and SGMII), OBSAI and CPRI. Transmit Pre-emphasis and Receive Equalization settings make SERDES suitable for chip to chip and small form factor backplane applications.

Lattice Diamond® design software allows large complex designs to be efficiently implemented using the LatticeECP2/M FPGA family. Synthesis library support for LatticeECP2/M is available for popular logic synthesis tools. The Diamond software uses the synthesis tool output along with the constraints from its floor planning tools to place and route the design in the LatticeECP2/M device. The Diamond design tool extracts the timing from the routing and back-annotates it into the design for timing verification.

Lattice provides many pre-engineered IP (Intellectual Property) modules for the LatticeECP2/M family. By using these IP cores as standardized blocks, designers are free to concentrate on the unique aspects of their design, increasing their productivity.

LatticeECP2/M Family Data Sheet Architecture

January 2012 Data Sheet DS1006

Architecture Overview

Each LatticeECP2/M device contains an array of logic blocks surrounded by Programmable I/O Cells (PIC). Interspersed between the rows of logic blocks are rows of sysMEM™ Embedded Block RAM (EBR) and rows of sysDSP™ Digital Signal Processing blocks, as shown in Figure 2-1. In addition, the LatticeECP2M family contains SERDES Quads in one or more of the corners. Figure 2-2 shows the block diagram of ECP2M20 with one quad.

There are two kinds of logic blocks, the Programmable Functional Unit (PFU) and Programmable Functional Unit without RAM (PFF). The PFU contains the building blocks for logic, arithmetic, RAM and ROM functions. The PFF block contains building blocks for logic, arithmetic and ROM functions. Both PFU and PFF blocks are optimized for flexibility, allowing complex designs to be implemented quickly and efficiently. Logic Blocks are arranged in a two-dimensional array. Only one type of block is used per row.

The LatticeECP2/M devices contain one or more rows of sysMEM EBR blocks. sysMEM EBRs are large dedicated 18K fast memory blocks. Each sysMEM block can be configured in a variety of depths and widths of RAM or ROM. In addition, LatticeECP2/M devices contain up to two rows of DSP Blocks. Each DSP block has multipliers and adder/accumulators, which are the building blocks for complex signal processing capabilities.

The LatticeECP2M devices feature up to 16 embedded 3.125Gbps SERDES (Serializer / Deserializer) channels. Each SERDES channel contains independent 8b/10b encoding / decoding, polarity adjust and elastic buffer logic. Each group of four SERDES channels along with its Physical Coding Sub-layer (PCS) block, creates a quad. The functionality of the SERDES/PCS Quads can be controlled by memory cells set during device configuration or by registers that are addressable during device operation. The registers in every quad can be programmed by a soft IP interface, referred to as the SERDES Client Interface (SCI). These quads (up to four) are located at the corners of the devices.

Each PIC block encompasses two PIOs (PIO pairs) with their respective sysI/O buffers. The sysI/O buffers of the LatticeECP2/M devices are arranged in eight banks, allowing the implementation of a wide variety of I/O standards. In addition, a separate I/O bank is provided for the programming interfaces. PIO pairs on the left and right edges of the device can be configured as LVDS transmit/receive pairs. The PIC logic also includes pre-engineered support to aid in the implementation of high speed source synchronous standards such as SPI4.2, along with memory interfaces including DDR2.

Other blocks provided include PLLs, DLLs and configuration functions. The LatticeECP2/M architecture provides two General PLLs (GPLL) and up to six Standard PLLs (SPLL) per device. In addition, each LatticeECP2/M family member provides two DLLs per device. The GPLLs and DLLs blocks are located in pairs at the end of the bottom-most EBR row; the DLL block is located towards the edge of the device. The SPLL blocks are located at the end of the other EBR/DSP rows.

The configuration block that supports features such as configuration bit-stream decryption, transparent updates and dual boot support is located toward the center of this EBR row. The Ball Grid Array (BGA) package devices in the LatticeECP2/M family supports a sysCONFIG™ port located in the corner between banks four and five, which allows for serial or parallel device configuration.

In addition, every device in the family has a JTAG port. This family also provides an on-chip oscillator. The LatticeECP2/M devices use 1.2V as their core voltage.

Figure 2-1. Simplified Block Diagram, ECP2-6 Device (Top Level)

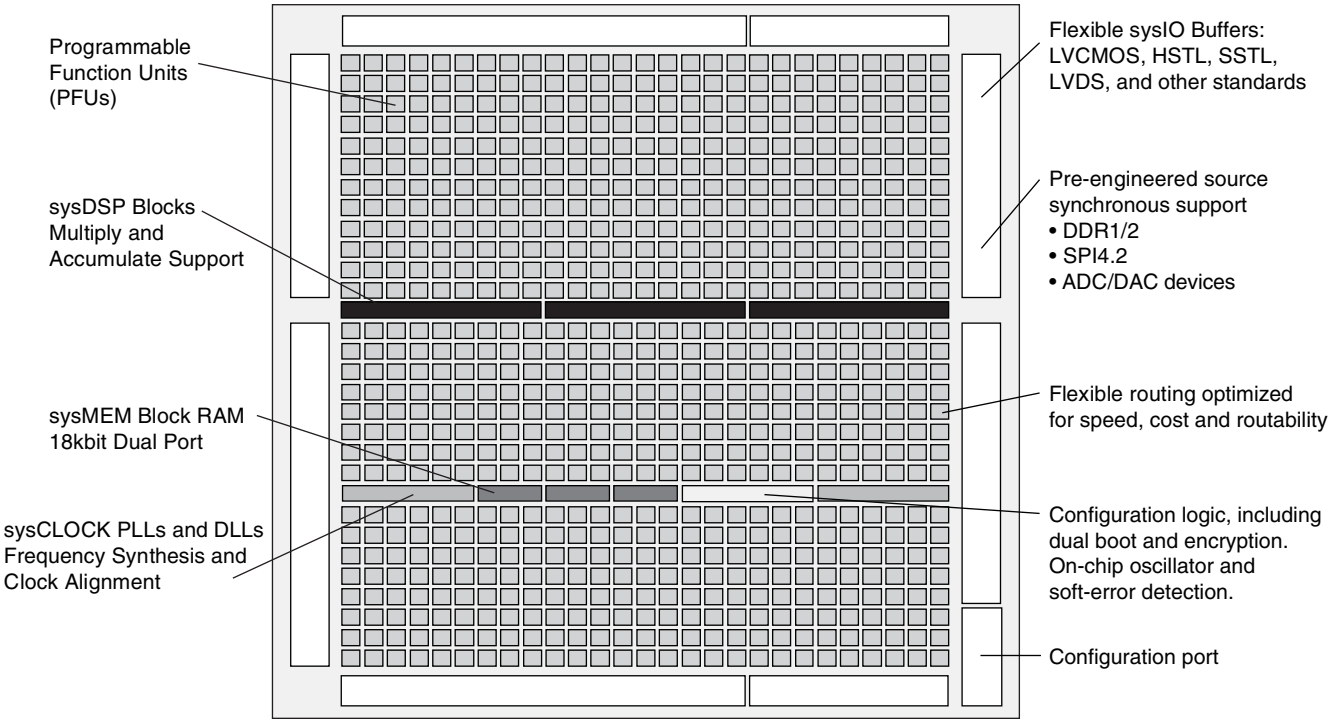
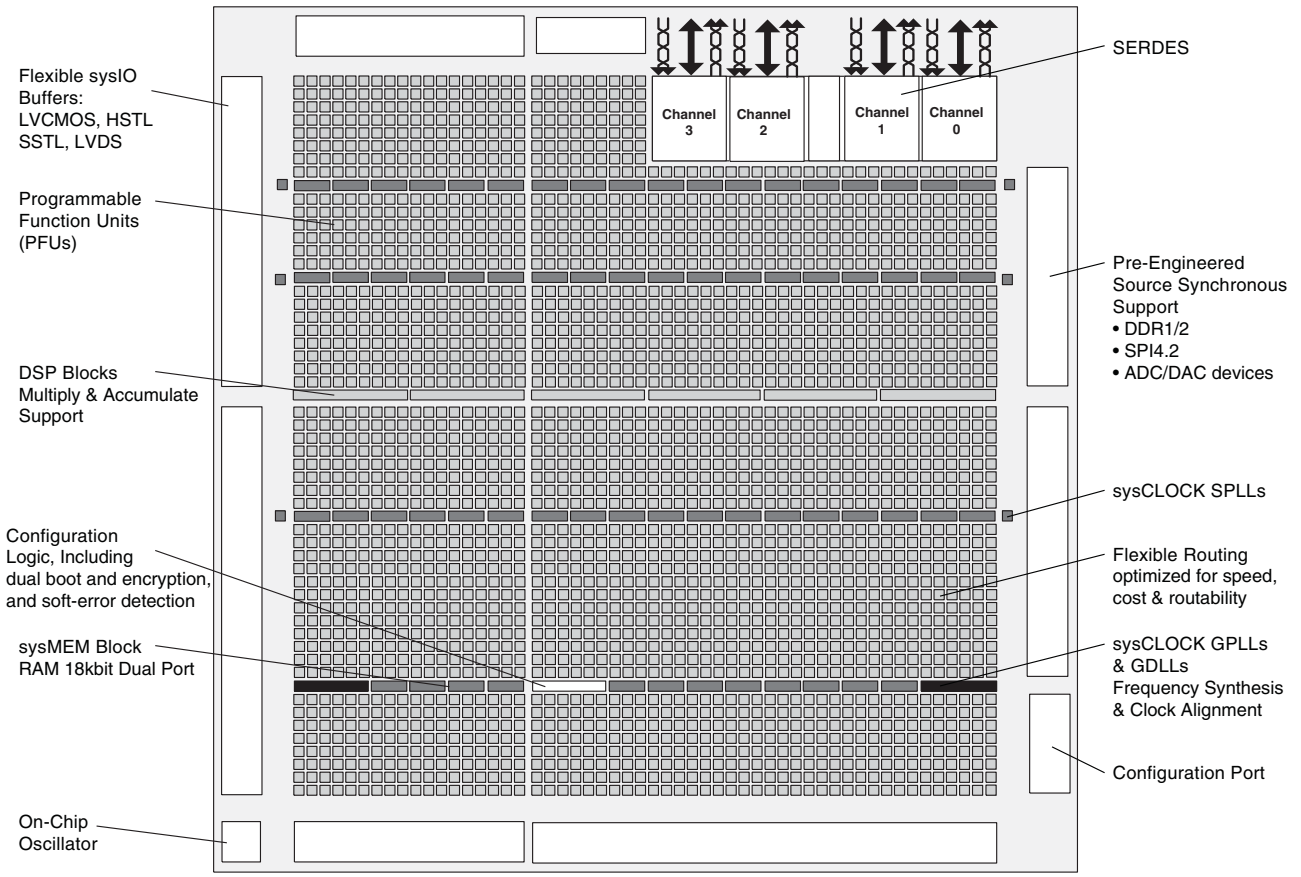


Figure 2-2. Simplified Block Diagram, ECP2M20 Device (Top Level)

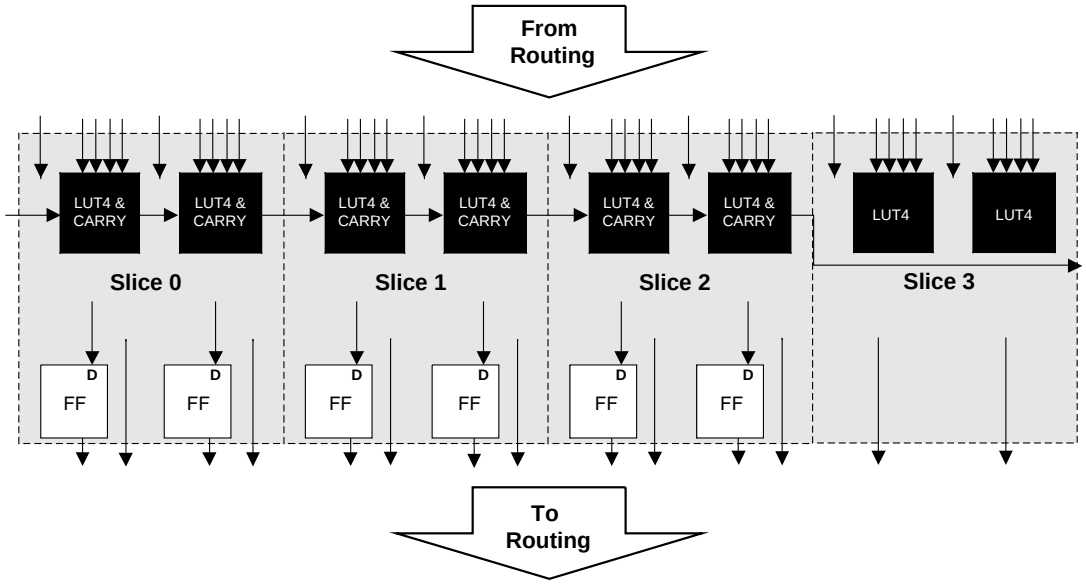


PFU Blocks

The core of the LatticeECP2/M device consists of PFU blocks, which are provided in two forms, the PFU and PFF. The PFUs can be programmed to perform Logic, Arithmetic, Distributed RAM and Distributed ROM functions. PFF blocks can be programmed to perform Logic, Arithmetic and ROM functions. Except where necessary, the remainder of this data sheet will use the term PFU to refer to both PFU and PFF blocks.

Each PFU block consists of four interconnected slices, numbered 0-3 as shown in Figure 2-3. All the interconnections to and from PFU blocks are from routing. There are 50 inputs and 23 outputs associated with each PFU block.

Figure 2-3. PFU Diagram



Slice

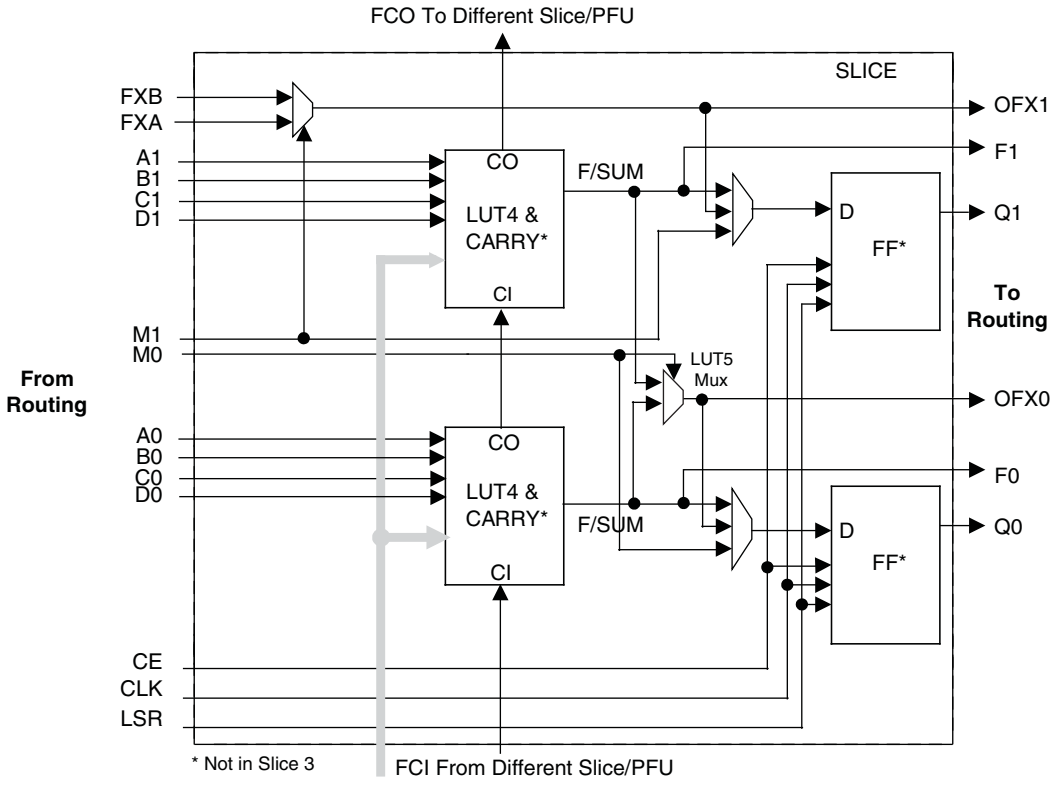
Slice 0 through Slice 2 contain two LUT4s feeding two registers, whereas Slice 3 contains two LUT4s only. For PFUs, Slice 0 and Slice 2 can also be configured as distributed memory, a capability not available in the PFF. Table 2-1 shows the capability of the slices in both PFF and PFU blocks along with the operation modes they enable. In addition, each PFU contains some logic that allows the LUTs to be combined to perform functions such as LUT5, LUT6, LUT7 and LUT8. There is control logic to perform set/reset functions (programmable as synchronous/asynchronous), clock select, chip-select and wider RAM/ROM functions. Figure 2-4 shows an overview of the internal logic of the slice. The registers in the slice can be configured for positive/negative and edge triggered or level sensitive clocks.

Table 2-1. Resources and Modes Available per Slice

Slice	PFU BLock		PFF Block	
	Resources	Modes	Resources	Modes
Slice 0	2 LUT4s and 2 Registers	Logic, Ripple, RAM, ROM	2 LUT4s and 2 Registers	Logic, Ripple, ROM
Slice 1	2 LUT4s and 2 Registers	Logic, Ripple, ROM	2 LUT4s and 2 Registers	Logic, Ripple, ROM
Slice 2	2 LUT4s and 2 Registers	Logic, Ripple, RAM, ROM	2 LUT4s and 2 Registers	Logic, Ripple, ROM
Slice 3	2 LUT4s	Logic, ROM	2 LUT4s	Logic, ROM

Slices 0, 1 and 2 have 14 input signals: 13 signals from routing and one from the carry-chain (from the adjacent slice or PFU). There are seven outputs: six to routing and one to carry-chain (to the adjacent PFU). Slice 3 has 13 input signals from routing and four signals to routing. Table 2-2 lists the signals associated with Slice 0 to Slice 2.

Figure 2-4. Slice Diagram



For Slices 0 and 2, memory control signals are generated from Slice 1 as follows:

WCK is CLK
WRE is from LSR
DI[3:2] for Slice 2 and DI[1:0] for Slice 0 data
WAD [A:D] is a 4bit address from slice 1 LUT input

Table 2-2. Slice Signal Descriptions

Function	Type	Signal Names	Description
Input	Data signal	A0, B0, C0, D0	Inputs to LUT4
Input	Data signal	A1, B1, C1, D1	Inputs to LUT4
Input	Multi-purpose	M0	Multipurpose Input
Input	Multi-purpose	M1	Multipurpose Input
Input	Control signal	CE	Clock Enable
Input	Control signal	LSR	Local Set/Reset
Input	Control signal	CLK	System Clock
Input	Inter-PFU signal	FC	Fast Carry-in[1]
Input	Inter-slice signal	FXA	Intermediate signal to generate LUT6 and LUT7
Input	Inter-slice signal	FXB	Intermediate signal to generate LUT6 and LUT7
Output	Data signals	F0, F1	LUT4 output register bypass signals
Output	Data signals	Q0, Q1	Register outputs
Output	Data signals	OFX0	Output of a LUT5 MUX
Output	Data signals	OFX1	Output of a LUT6, LUT7, LUT8[2] MUX depending on the slice
Output	Inter-PFU signal	FCO	Slice 2 of each PFU is the fast carry chain output[1]

1. See Figure 2-4 for connection details.
2. Requires two PFUs.

Modes of Operation

Each slice has up to four potential modes of operation: Logic, Ripple, RAM and ROM.

Logic Mode
In this mode, the LUTs in each slice are configured as 4-input combinatorial lookup tables. A LUT4 can have 16 possible input combinations. Any four input logic functions can be generated by programming this lookup table. Since there are two LUT4s per slice, a LUT5 can be constructed within one slice. Larger look-up tables such as LUT6, LUT7 and LUT8 can be constructed by concatenating other slices. Note LUT8 requires more than four slices.

Ripple Mode
Ripple mode supports the efficient implementation of small arithmetic functions. In ripple mode, the following functions can be implemented by each slice:

- Addition 2-bit
- Subtraction 2-bit
- Add/Subtract 2-bit using dynamic control
- Up counter 2-bit
- Down counter 2-bit
- Up/Down counter with Async clear
- Up/Down counter with preload (sync)
- Ripple mode multiplier building block
- Multiplier support
- Comparator functions of A and B inputs
  - A greater-than-or-equal-to B
  - A not-equal-to B
  - A less-than-or-equal-to B

Ripple Mode includes an optional configuration that performs arithmetic using fast carry chain methods. In this configuration (also referred to as CCU2 mode) two additional signals, Carry Generate and Carry Propagate, are generated on a per slice basis to allow fast arithmetic functions to be constructed by concatenating Slices.

RAM Mode
In this mode, a 16x4-bit distributed single port RAM (SPR) can be constructed using each LUT block in Slice 0 and Slice 2 as a 16x1-bit memory. Slice 1 is used to provide memory address and control signals. A 16x2-bit pseudo dual port RAM (PDPR) memory is created by using one Slice as the read-write port and the other companion slice as the read-only port.

The Lattice design tools support the creation of a variety of different size memories. Where appropriate, the software will construct these using distributed memory primitives that represent the capabilities of the PFU. Table 2-3 shows the number of slices required to implement different distributed RAM primitives. For more information about using RAM in LatticeECP2/M devices, please see the list of additional technical documentation at the end of this data sheet.

Table 2-3. Number of Slices Required to Implement Distributed RAM

	SPR 16X4	PDPR 16X4
Number of slices	3	3

Note: SPR = Single Port RAM, PDPR = Pseudo Dual Port RAM

ROM Mode

ROM mode uses the LUT logic; hence, Slices 0 through 3 can be used in ROM mode. Preloading is accomplished through the programming interface during PFU configuration.

Routing

There are many resources provided in the LatticeECP2/M devices to route signals individually or as buses with related control signals. The routing resources consist of switching circuitry, buffers and metal interconnect (routing) segments.

The inter-PFU connections are made with x1 (spans two PFU), x2 (spans three PFU) and x6 (spans seven PFU). The x1 and x2 connections provide fast and efficient connections in horizontal and vertical directions. The x2 and x6 resources are buffered, allowing the routing of both short and long connections between PFUs.

The LatticeECP2/M family has an enhanced routing architecture that produces a compact design. The Diamond design software takes the output of the synthesis tool and places and routes the design. Generally, the place and route tool is completely automatic, although an interactive routing editor is available to optimize the design.

sysCLOCK Phase Locked Loops (GPLL/SPLL)

The sysCLOCK PLLs provide the ability to synthesize clock frequencies. All the devices in the LatticeECP2/M family support two General Purpose PLLs (GPLLs) which are full-featured PLLs. In addition, some of the larger devices have two to six Standard PLLs (SPLLs) that have a subset of GPLL functionality.

General Purpose PLL (GPLL)

The architecture of the GPLL is shown in Figure 2-5. A description of the GPLL functionality follows.

CLKI is the reference frequency (generated either from the pin or from routing) for the PLL. CLKI feeds into the Input Clock Divider block. The CLKFB is the feedback signal (generated from CLKOP or from a user clock PIN/logic). This signal feeds into the Feedback Divider. The Feedback Divider is used to multiply the reference frequency.

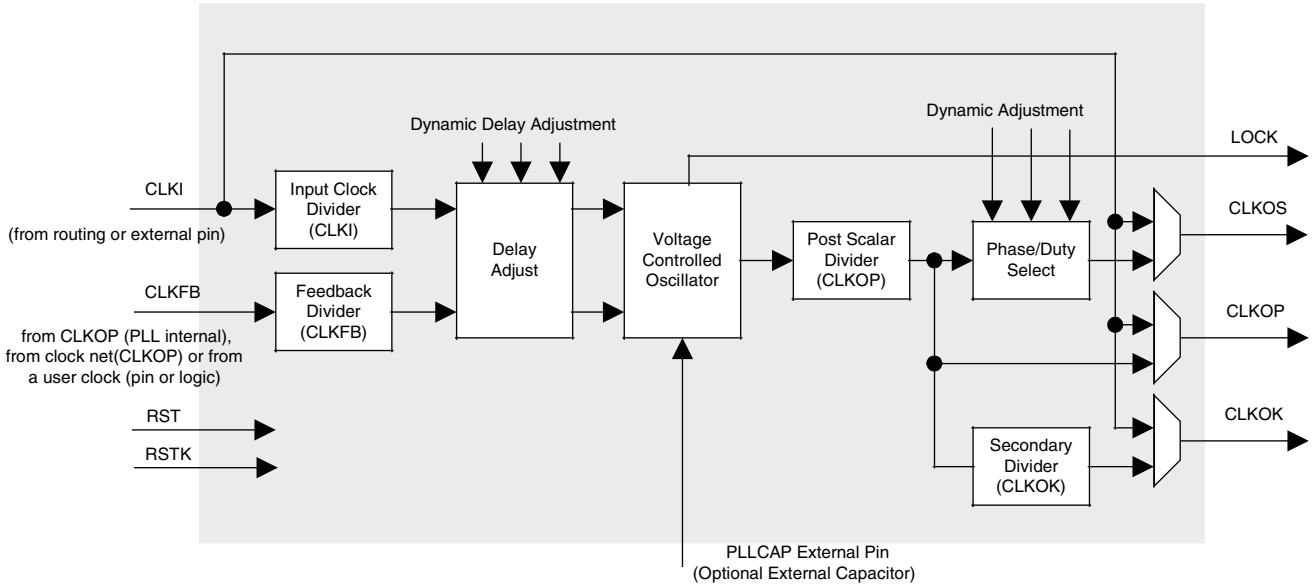
The Delay Adjust Block adjusts either the delays of the reference or feedback signals. The Delay Adjust Block can either be programmed during configuration or can be adjusted dynamically. The setup, hold or clock-to-out times of the device can be improved by programming a delay in the feedback or input path of the PLL, which will advance or delay the output clock with reference to the input clock.

Following the Delay Adjust Block, both the input path and feedback signals enter the Voltage Controlled Oscillator (VCO) block. In this block the difference between the input path and feedback signals is used to control the frequency and phase of the oscillator. A LOCK signal is generated by the VCO to indicate that the VCO has locked onto the input clock signal. In dynamic mode, the PLL may lose lock after a dynamic delay adjustment and not relock until the t_{LOCK} parameter has been satisfied. LatticeECP2/M devices have two dedicated pins on the left and right edges of the device for connecting optional external capacitors to the VCO. This allows the PLLs to operate at a lower frequency. This is a shared resource that can only be used by one PLL (GPLL or SPLL) per side.

The output of the VCO then enters the post-scalar divider. The post-scalar divider allows the VCO to operate at higher frequencies than the clock output (CLKOP), thereby increasing the frequency range. A secondary divider takes the CLKOP signal and uses it to derive lower frequency outputs (CLKOK). The Phase/Duty Select block adjusts the phase and duty cycle of the CLKOP signal and generates the CLKOS signal. The phase/duty cycle setting can be pre-programmed or dynamically adjusted.

The primary output from the post scalar divider CLKOP along with the outputs from the secondary divider (CLKOK) and Phase/Duty select (CLKOS) are fed to the clock distribution network.

Figure 2-5. General Purpose PLL (GPLL) Diagram



Standard PLL (SPLL)

Some of the larger devices have two to six Standard PLLs (SPLLs). SPLLs have the same features as GPLLs but without delay adjustment capability. SPLLs also provide different parametric specifications. For more information, please see the list of additional technical documentation at the end of this data sheet.

Table 2-4 provides a description of the signals in the GPLL and SPLL blocks.

Table 2-4. GPLL and SPLL Blocks Signal Descriptions

Signal	I/O	Description
CLKI	I	Clock input from external pin or routing
CLKFB	I	PLL feedback input from CLKOP (PLL internal), from clock net (CLKOP) or from a user clock (PIN or logic)
RST	I	“1” to reset PLL counters, VCO, charge pumps and M-dividers
RSTK	I	“1” to reset K-divider
CLKOS	O	PLL output clock to clock tree (phase shifted/duty cycle changed)
CLKOP	O	PLL output clock to clock tree (no phase shift)
CLKOK	O	PLL output to clock tree through secondary clock divider
LOCK	O	“1” indicates PLL LOCK to CLKI
DDAMODE[1]	I	Dynamic Delay Enable. “1”: Pin control (dynamic), “0”: Fuse Control (static)
DDAIZR[1]	I	Dynamic Delay Zero. “1”: delay = 0, “0”: delay = on
DDAILAG[1]	I	Dynamic Delay Lag/Lead. “1”: Lead, “0”: Lag
DDAIDEL[2:0][1]	I	Dynamic Delay Input
DPA MODES	I	DPA (Dynamic Phase Adjust/Duty Cycle Select) mode
DPHASE [3:0]	I	DPA Phase Adjust inputs
DDDUTY [3:0]	—	DPA Duty Cycle Select inputs

1. These signals are not available in SPLL.

Delay Locked Loops (DLL)

In addition to PLLs, the LatticeECP2/M family of devices has two DLLs per device.

CLKI is the input frequency (generated either from the pin or routing) for the DLL. CLKI feeds into the output muxes block to bypass the DLL, directly to the DELAY CHAIN block and (directly or through divider circuit) to the reference input of the Phase Frequency Detector (PFD) input mux. The reference signal for the PFD can also be generated from the Delay Chain and CLKFB signals. The feedback input to the PFD is generated from the CLKFB pin, CLKI or from tapped signal from the Delay chain.

The PFD produces a binary number proportional to the phase and frequency difference between the reference and feedback signals. This binary output of the PFD is fed into a Arithmetic Logic Unit (ALU). Based on these inputs, the ALU determines the correct digital control codes to send to the delay chain in order to better match the reference and feedback signals. This digital code from the ALU is also transmitted via the Digital Control bus (DCNTL) bus to its associated DLLDELA delay block. The ALUHOLD input allows the user to suspend the ALU output at its current value. The UDDCNTL signal allows the user to latch the current value on the DCNTL bus.

The DLL has two independent clock outputs, CLKOP and CLKOS. These outputs can individually select one of the outputs from the tapped delay line. The CLKOS has optional fine phase shift and divider blocks to allow this output to be further modified, if required. The fine phase shift block allows the CLKOS output to phase shifted a further 45, 22.5 or 11.25 degrees relative to its normal position. Both the CLKOS and CLKOP outputs are available with optional duty cycle correction. Divide by two and divide by four frequencies are available at CLKOS. The LOCK output signal is asserted when the DLL is locked. Figure 2-6 shows the DLL block diagram and Table 2-5 provides a description of the DLL inputs and outputs.

The user can configure the DLL for many common functions such as time reference delay mode and clock injection removal mode. Lattice provides primitives in its design tools for these functions. For more information about the DLL, please see the list of additional technical documentation at the end of this data sheet.

Figure 2-6. Delay Locked Loop Diagram (DLL)

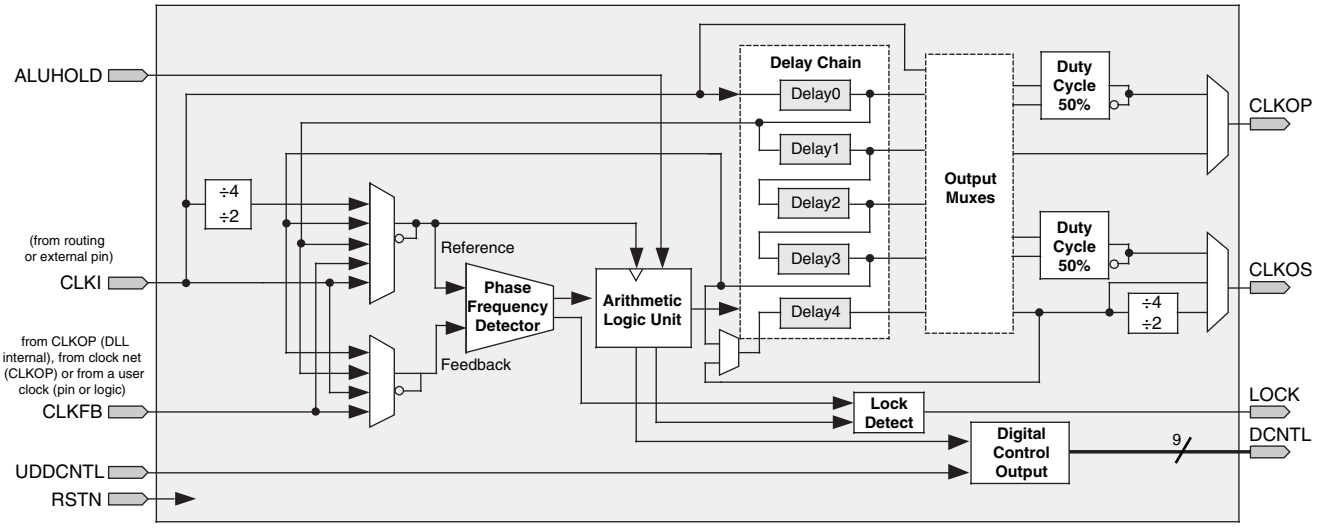


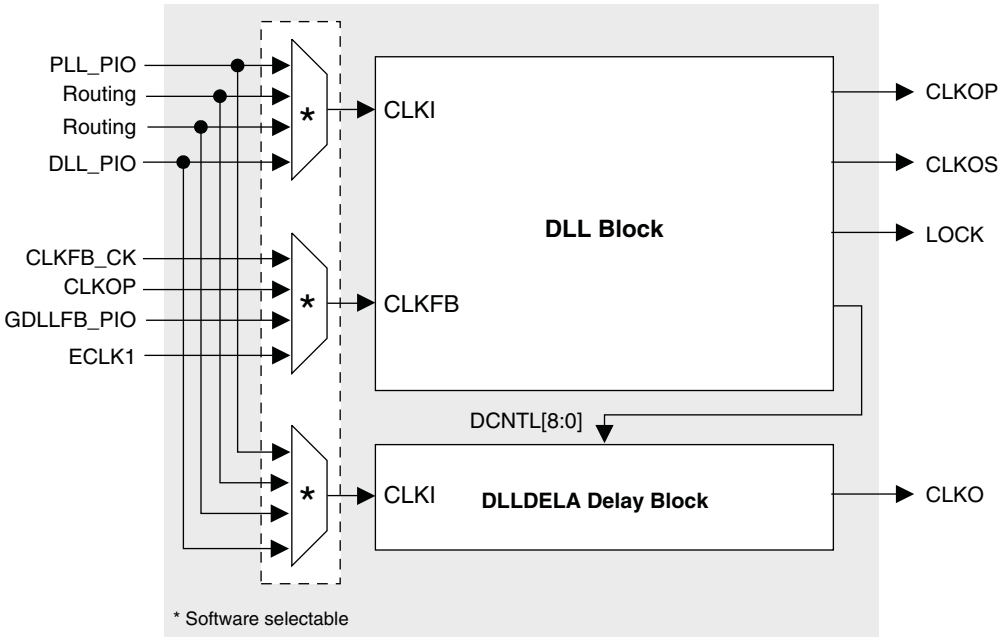
Table 2-5. DLL Signals

Signal	I/O	Description
CLKI	I	Clock input from external pin or routing
CLKFB	I	DLL feed input from DLL output, clock net, routing or external pin
RSTN	I	Active low synchronous reset
ALUHOLD	I	Active high freezes the ALU
UDDCNTL	I	Synchronous enable signal (hold high for two cycles) from routing
DCNTL[8:0]	O	Encoded digital control signals for PIC INDEL and slave delay calibration
CLKOP	O	The primary clock output
CLKOS	O	The secondary clock output with fine phase shift and/or division by 2 or by 4
LOCK	O	Active high phase lock indicator

DLLDELA Delay Block

Closely associated with each DLL is a DLLDELA block. This is a delay block consisting of a delay line with taps and a selection scheme that selects one of the taps. The DCNTL[8:0] bus controls the delay of the CLKO signal. Typically this is the delay setting that the DLL uses to achieve phase alignment. This results in the delay providing a calibrated 90° phase shift that is useful in centering a clock in the middle of a data cycle for source synchronous data. The CLKO signal feeds the edge clock network. Figure 2-7 shows the connections between the DLL block and the DLLDELA delay block. For more information, please see the list of additional technical documentation at the end of this data sheet.

Figure 2-7. DLLDELA Delay Block



PLL/DLL Cascading

LatticeECP2/M devices have been designed to allow certain combinations of PLL (GPLL and SPLL) and DLL cascading. The allowable combinations are:

- PLL to PLL supported
- PLL to DLL supported

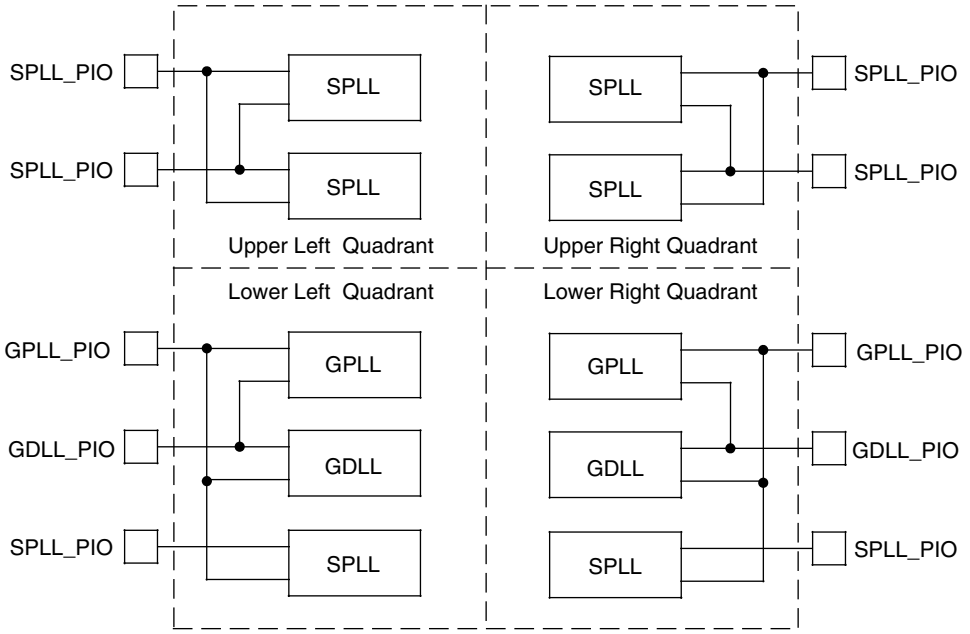
The DLLs in the LatticeECP2/M are used to shift the clock in relation to the data for source synchronous inputs. PLLs are used for frequency synthesis and clock generation for source synchronous interfaces. Cascading PLL and DLL blocks allows applications to utilize the unique benefits of both DLLs and PLLs.

For further information about the DLL, please see the list of additional technical documentation at the end of this data sheet.

GPLL/SPLL/GDLL PIO Input Pin Connections (LatticeECP2M Family Only)

All LatticeECP2M devices contain two GDLLs, two GPLLs and six SPLLs, arranged in quadrants as shown in Figure 2-8. In the LatticeECP2M devices GPLLs, SPLLs and GDLLs share their input pins. Figure 2-8 shows the sharing of SPLLs input pin connections in the upper two quadrants and the sharing of GDLL, GPLL and SPLL input pin connections in the lower two quadrants.

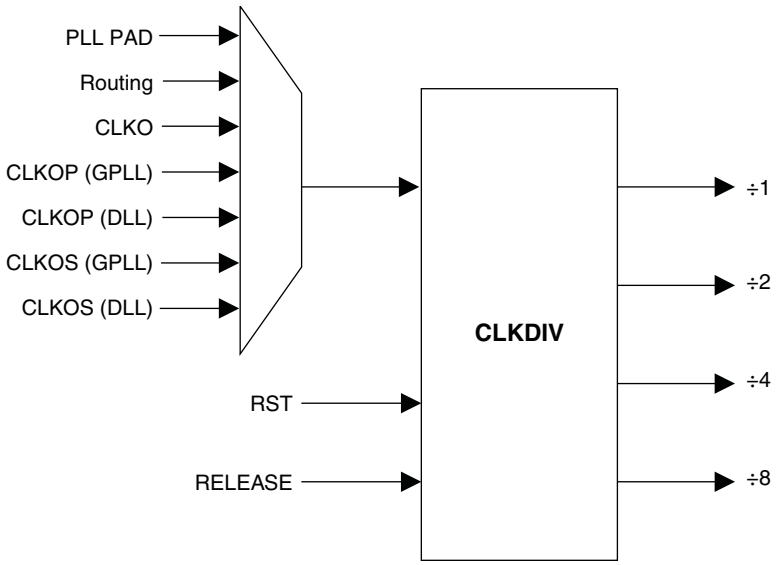
Figure 2-8. Sharing of PIO Pins by GPLL, SPLL and GDLL in LatticeECP2M Devices



Clock Dividers

LatticeECP2/M devices have two clock dividers, one on the left side and one on the right side of the device. These are intended to generate a slower-speed system clock from a high-speed edge clock. The block operates in a ÷2, ÷4 or ÷8 mode and maintains a known phase relationship between the divided down clock and the high-speed clock based on the release of its reset signal. The clock dividers can be fed from selected PLL/DLL outputs, DLL-DELA delay blocks, routing or from an external clock input. The clock divider outputs serve as primary clock sources and feed into the clock distribution network. The Reset (RST) control signal resets input and synchronously forces all outputs to low. The RELEASE signal releases outputs synchronously to the input clock. For further information about clock dividers, please see the list of additional technical documentation at the end of this data sheet. Figure 2-9 shows the clock divider connections.

Figure 2-9. Clock Divider Connections



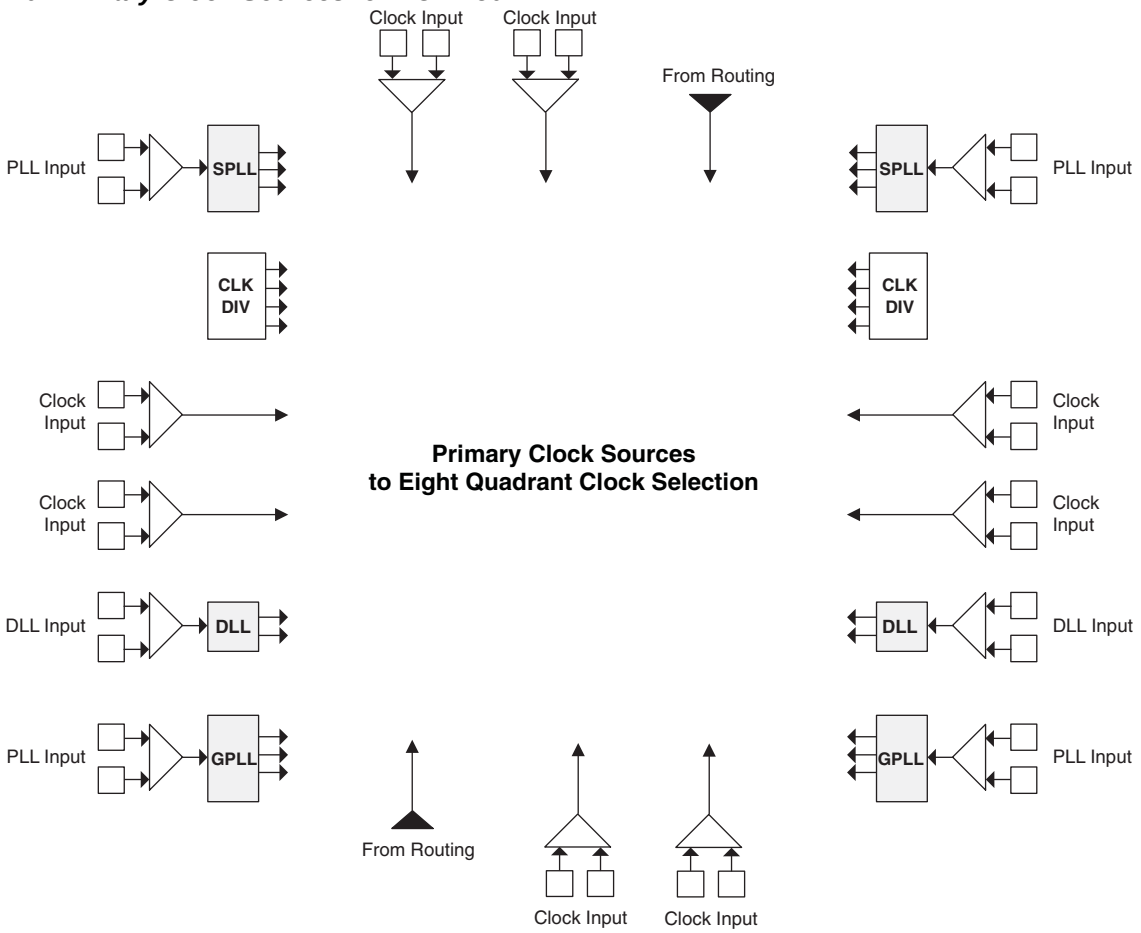
Clock Distribution Network

LatticeECP2/M devices have eight quadrant-based primary clocks and eight flexible region-based secondary clocks/control signals. Two high performance edge clocks are available on each edge of the device to support high speed interfaces. These clock inputs are selected from external I/Os, the sysCLOCK PLLs, DLLs or routing. These clock inputs are fed throughout the chip via a clock distribution system.

Primary Clock Sources

LatticeECP2/M devices derive clocks from five primary sources: PLL (GPLL and SPLL) outputs, DLL outputs, CLKDIV outputs, dedicated clock inputs and routing. LatticeECP2/M devices have two to eight sysCLOCK PLLs and two DLLs, located on the left and right sides of the device. There are eight dedicated clock inputs, two on each side of the device, with the exception of the LatticeECP2M 256-fpBGA package devices which have six dedicated clock inputs on the device. Figure 2-10 shows the primary clock sources.

Figure 2-10. Primary Clock Sources for ECP2-50

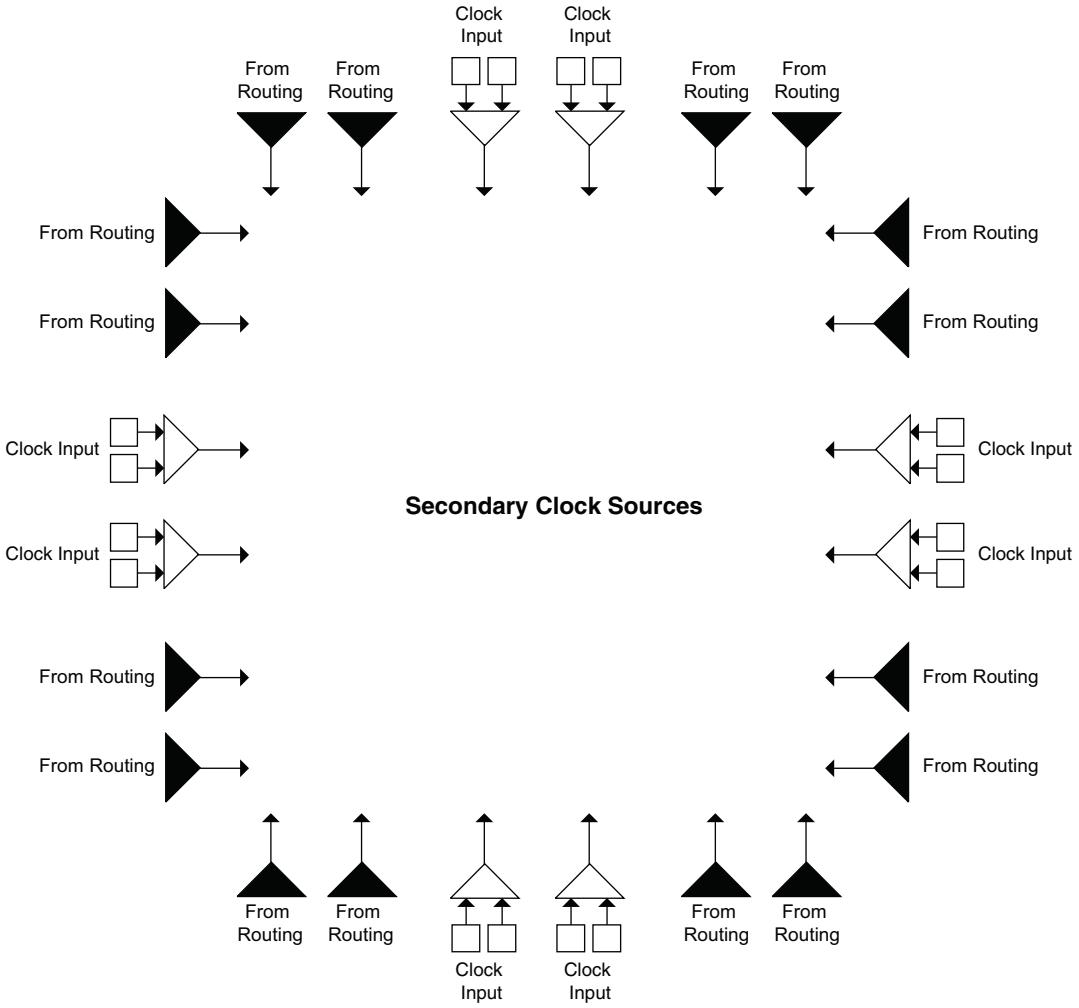


Note: This diagram shows sources for the ECP2-50 device. Smaller LatticeECP2 devices have fewer SPLLs. All LatticeECP2M device have six SPLLs.

Secondary Clock/Control Sources

LatticeECP2/M devices derive secondary clocks (SC0 through SC7) from eight dedicated clock input pads and the rest from routing. Figure 2-11 shows the secondary clock sources.

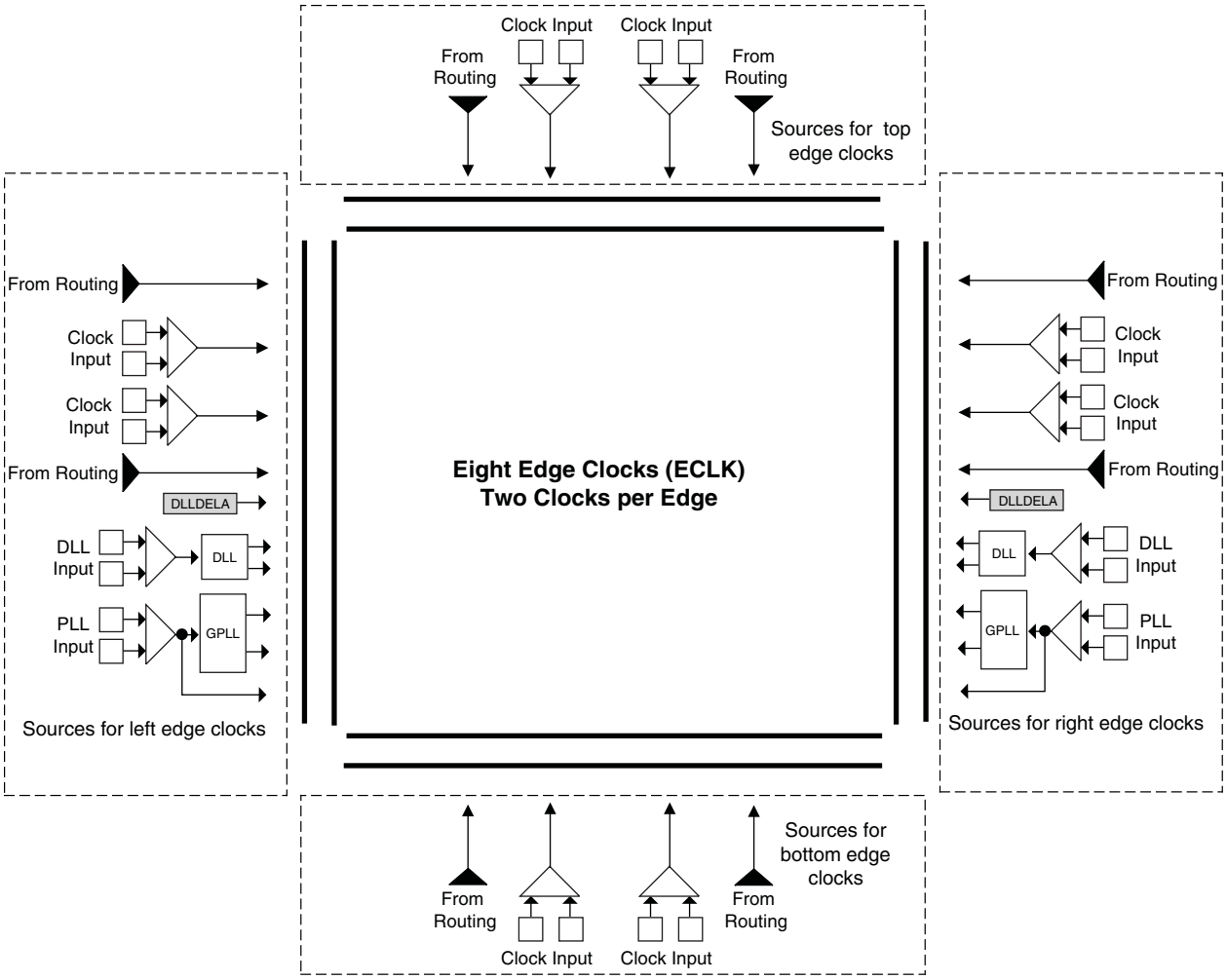
Figure 2-11. Secondary Clock Sources



Edge Clock Sources

Edge clock resources can be driven from a variety of sources at the same edge. Edge clock resources can be driven from adjacent edge clock PIOs, primary clock PIOs, PLLs/DLLs and clock dividers as shown in Figure 2-12.

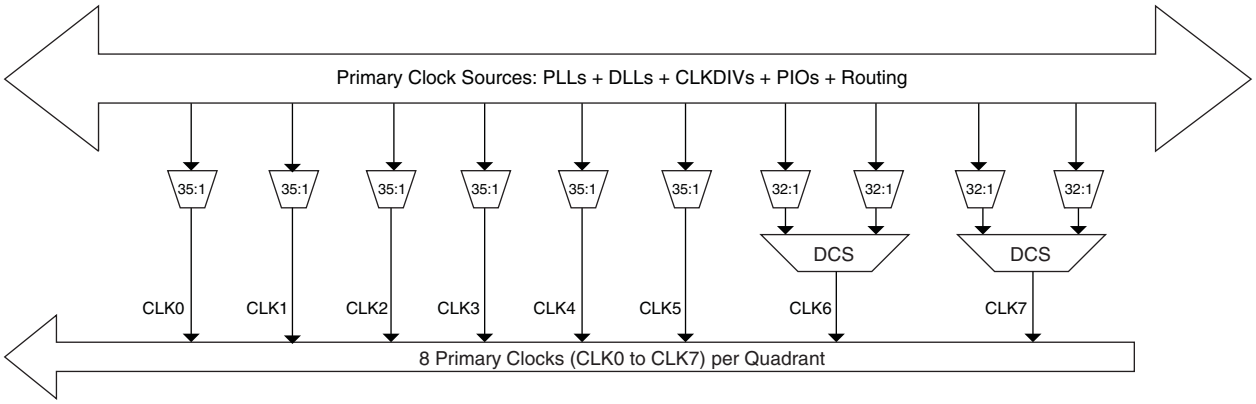
Figure 2-12. Edge Clock Sources



Primary Clock Routing

The clock routing structure in LatticeECP2/M devices consists of a network of eight primary clock lines (CLK0 through CLK7) per quadrant. The primary clocks of each quadrant are generated from muxes located in the center of the device. All the clock sources are connected to these muxes. Figure 2-13 shows the clock routing for one quadrant. Each quadrant mux is identical. If desired, any clock can be routed globally

Figure 2-13. Per Quadrant Primary Clock Selection

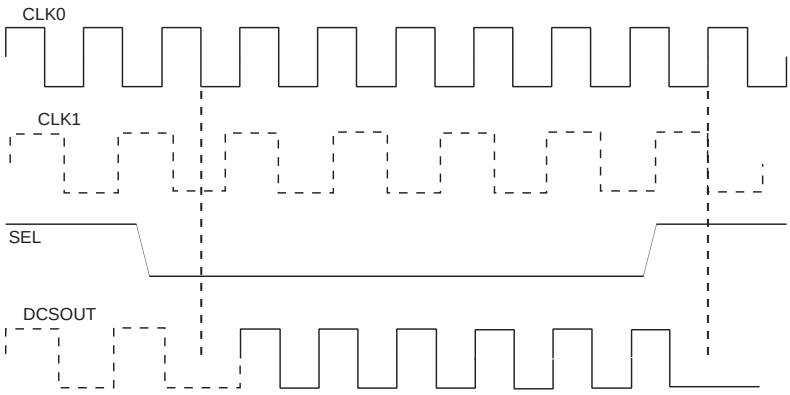


Dynamic Clock Select (DCS)

The DCS is a smart multiplexer function available in the primary clock routing. It switches between two independent input clock sources without any glitches or runt pulses. This is achieved regardless of when the select signal is toggled. There are two DCS blocks per quadrant; in total, there are eight DCS blocks per device. The inputs to the DCS block come from the center muxes. The output of the DCS is connected to primary clocks CLK6 and CLK7 (see Figure 2-13).

Figure 2-14 shows the timing waveforms of the default DCS operating mode. The DCS block can be programmed to other modes. For more information about the DCS, please see the list of additional technical documentation at the end of this data sheet.

Figure 2-14. DCS Waveforms



Secondary Clock/Control Routing

Secondary clocks in the LatticeECP2 devices are region-based resources. The benefit of region-based resources is the relatively low injection delay and skew within the region, as compared to primary clocks. EBR/DSP rows and a special vertical routing channel bound the secondary clock regions. This special vertical routing channel aligns with either the left edge of the center DSP block in the DSP row or the center of the DSP row. Figure 2-15 shows

this special vertical routing channel and the eight secondary clock regions for the ECP2-50. LatticeECP2 devices have four secondary clocks (SC0 to SC3) which are distrubed to every region.

The secondary clock muxes are located in the center of the device. Figure 2-16 shows the mux structure of the secondary clock routing. Secondary clocks SC0 to SC3 are used for clock and control and SC4 to SC7 are used for high fan-out signals.

Figure 2-15. Secondary Clock Regions ECP2-50

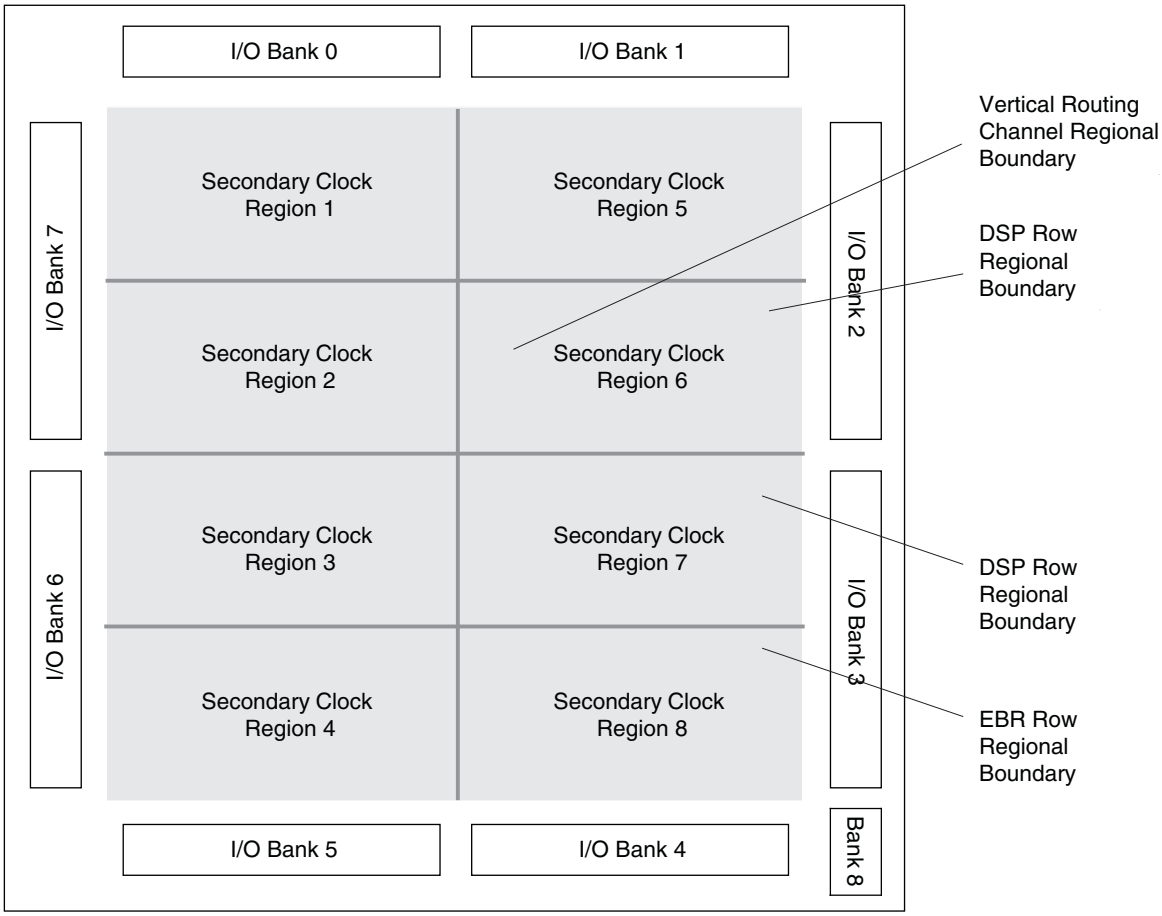
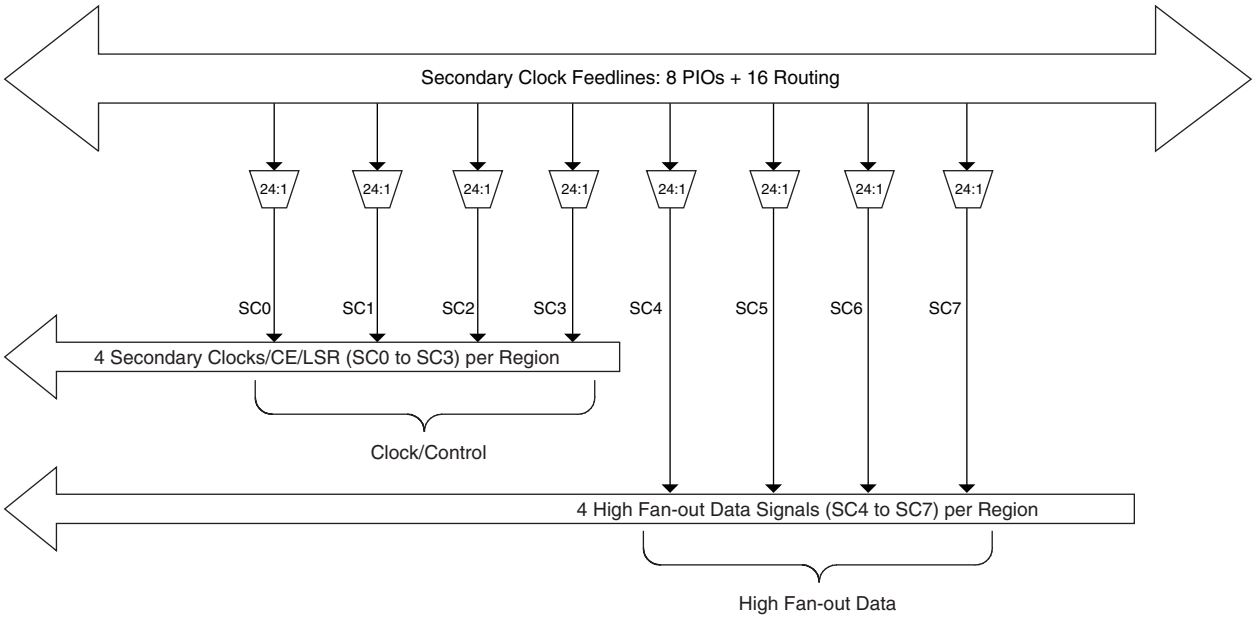


Figure 2-16. Secondary Clock Selection



Slice Clock Selection

Figure 2-17 shows the clock selections and Figure 2-18 shows the control selections for Slice0 through Slice2. All the primary clocks and the four secondary clocks are routed to this clock selection mux. Other signals can be used as a clock input to the slices via routing. Slice controls are generated from the secondary clocks or other signals connected via routing.

If none of the signals are selected for both clock and control then the default value of the mux output is 1. Slice 3 does not have any registers; therefore it does not have the clock or control muxes.

Figure 2-17. Slice0 through Slice2 Clock Selection

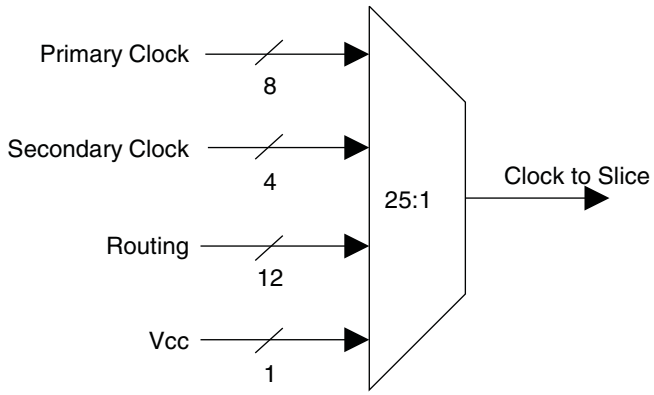
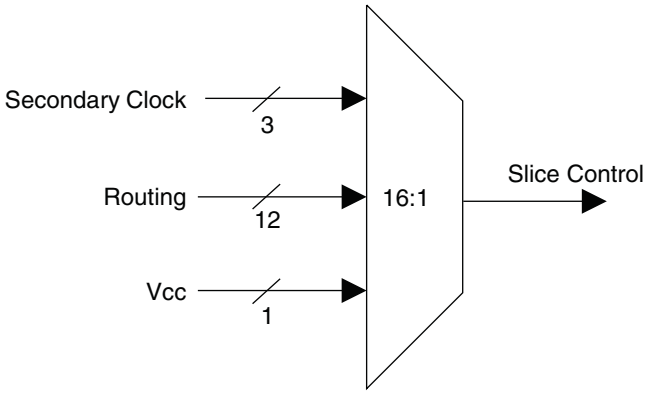


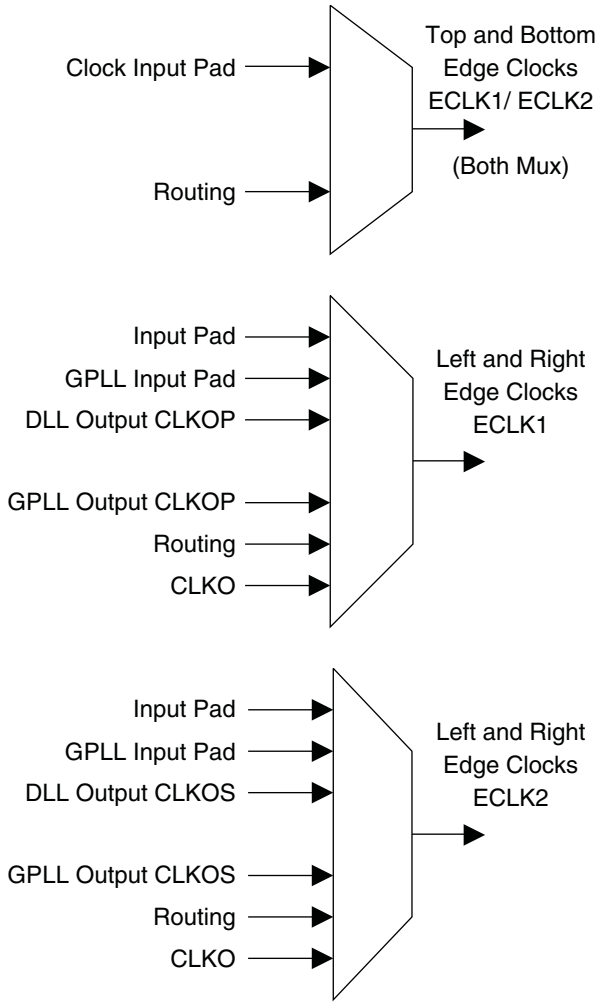
Figure 2-18. Slice0 through Slice2 Control Selection



Edge Clock Routing

LatticeECP2/M devices have a number of high-speed edge clocks that are intended for use with the PIOs in the implementation of high-speed interfaces. There are eight edge clocks per device: two edge clocks per edge. Different PLL and DLL outputs are routed to the two muxes on the left and right sides of the device. In addition, the CLKO signal (generated from the DLLDELA block) is routed to all the edge clock muxes on the left and right sides of the device. Figure 2-19 shows the selection muxes for these clocks.

Figure 2-19. Edge Clock Mux Connections



sysMEM Memory

LatticeECP2/M devices contains a number of sysMEM Embedded Block RAM (EBR). The EBR consists of an 18-Kbit RAM with dedicated input and output registers.

sysMEM Memory Block

The sysMEM block can implement single port, dual port or pseudo dual port memories. Each block can be used in a variety of depths and widths as shown in Table 2-6. FIFOs can be implemented in sysMEM EBR blocks by implementing support logic with PFUs. The EBR block facilitates parity checking by supporting an optional parity bit for each data byte. EBR blocks provide byte-enable support for configurations with18-bit and 36-bit data widths.

Table 2-6. sysMEM Block Configurations

Memory Mode	Configurations
Single Port	16,384 x 1 8,192 x 2 4,096 x 4 2,048 x 9 1,024 x 18 512 x 36
True Dual Port	16,384 x 1 8,192 x 2 4,096 x 4 2,048 x 9 1,024 x 18
Pseudo Dual Port	16,384 x 1 8,192 x 2 4,096 x 4 2,048 x 9 1,024 x 18 512 x 36

Bus Size Matching

All of the multi-port memory modes support different widths on each of the ports. The RAM bits are mapped LSB word 0 to MSB word 0, LSB word 1 to MSB word 1, and so on. Although the word size and number of words for each port varies, this mapping scheme applies to each port.

RAM Initialization and ROM Operation

If desired, the contents of the RAM can be pre-loaded during device configuration. By preloading the RAM block during the chip configuration cycle and disabling the write controls, the sysMEM block can also be utilized as a ROM.

Memory Cascading

Larger and deeper blocks of RAM can be created using EBR sysMEM Blocks. Typically, the Lattice design tools cascade memory transparently, based on specific design inputs.

Single, Dual and Pseudo-Dual Port Modes

In all the sysMEM RAM modes the input data and address for the ports are registered at the input of the memory array. The output data of the memory is optionally registered at the output.

EBR memory supports two forms of write behavior for single port or dual port operation:

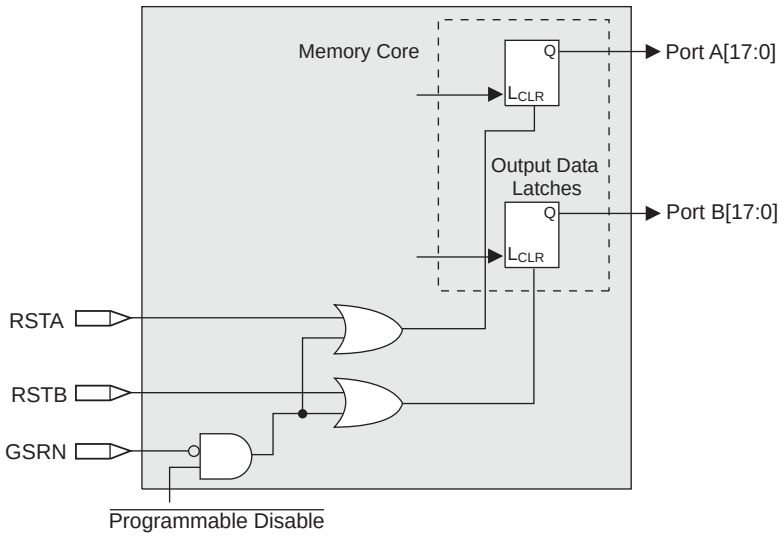
1. Normal – Data on the output appears only during a read cycle. During a write cycle, the data (at the current address) does not appear on the output. This mode is supported for all data widths.

2. Write Through – A copy of the input data appears at the output of the same port during a write cycle. This mode is supported for all data widths.

Memory Core Reset

The memory array in the EBR utilizes latches at the A and B output ports. These latches can be reset asynchronously or synchronously. RSTA and RSTB are local signals, which reset the output latches associated with Port A and Port B, respectively. The Global Reset (GSRN) signal resets both ports. The output data latches and associated resets for both ports are as shown in Figure 2-20.

Figure 2-20. Memory Core Reset

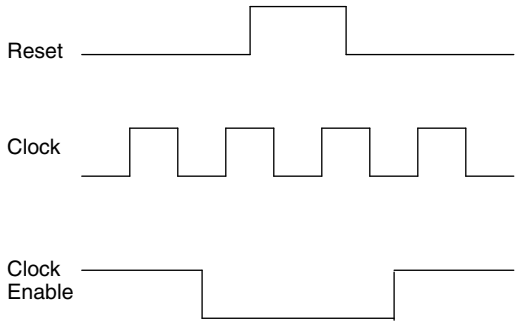


For further information about the sysMEM EBR block, please see the the list of additional technical documentation at the end of this data sheet.

EBR Asynchronous Reset

EBR asynchronous reset or GSR (if used) can only be applied if all clock enables are low for a clock cycle before the reset is applied and released a clock cycle after the reset is released, as shown in Figure 2-21. The GSR input to the EBR is always asynchronous.

Figure 2-21. EBR Asynchronous Reset (Including GSR) Timing Diagram



If all clock enables remain enabled, the EBR asynchronous reset or GSR may only be applied and released after the EBR read and write clock inputs are in a steady state condition for a minimum of $1/f_{MAX}$ (EBR clock). The reset release must adhere to the EBR synchronous reset setup time before the next active read or write clock edge.

If an EBR is pre-loaded during configuration, the GSR input must be disabled or the release of the GSR during device Wake Up must occur before the release of the device I/Os becomes active.

These instructions apply to all EBR RAM and ROM implementations.

Note that there are no reset restrictions if the EBR synchronous reset is used and the EBR GSR input is disabled.

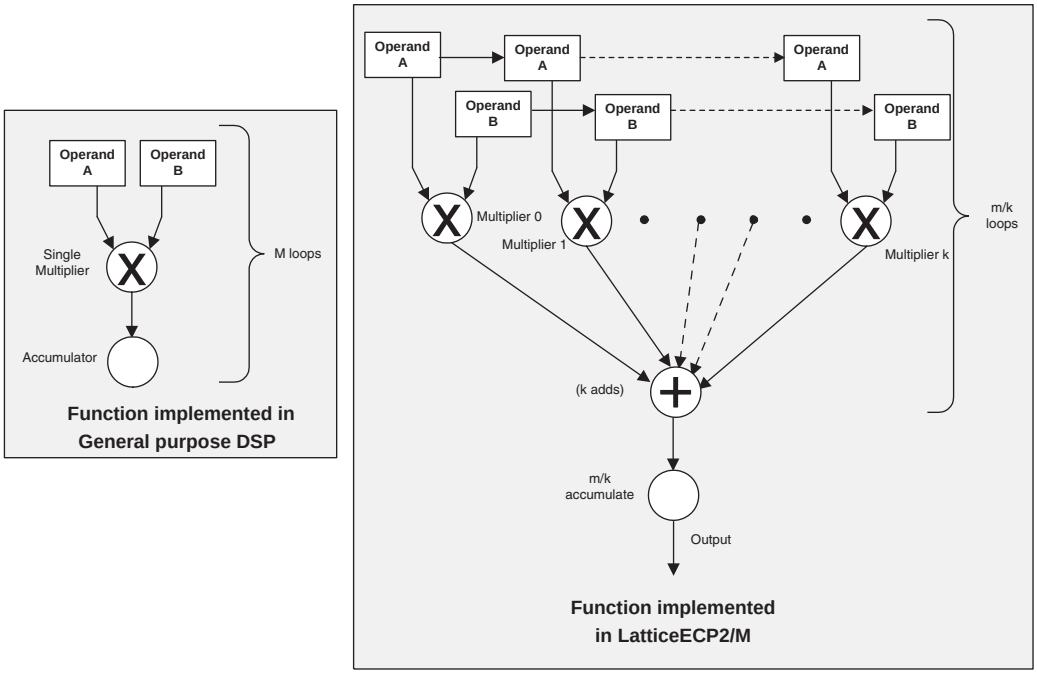
sysDSP™ Block

The LatticeECP2/M family provides a sysDSP block, making it ideally suited for low cost, high performance Digital Signal Processing (DSP) applications. Typical functions used in these applications are Finite Impulse Response (FIR) filters, Fast Fourier Transforms (FFT) functions, Correlators, Reed-Solomon/Turbo/Convolution encoders and decoders. These complex signal processing functions use similar building blocks such as multiply-adders and multiply-accumulators.

sysDSP Block Approach Compared to General DSP

Conventional general-purpose DSP chips typically contain one to four (Multiply and Accumulate) MAC units with fixed data-width multipliers; this leads to limited parallelism and limited throughput. Their throughput is increased by higher clock speeds. The LatticeECP2/M, on the other hand, has many DSP blocks that support different data-widths. This allows the designer to use highly parallel implementations of DSP functions. The designer can optimize the DSP performance vs. area by choosing an appropriate level of parallelism. Figure 2-22 compares the fully serial and the mixed parallel and serial implementations.

Figure 2-22. Comparison of General DSP and LatticeECP2/M Approaches



sysDSP Block Capabilities

The sysDSP block in the LatticeECP2/M family supports four functional elements in three 9, 18 and 36 data path widths. The user selects a function element for a DSP block and then selects the width and type (signed/unsigned) of its operands. The operands in the LatticeECP2/M family sysDSP Blocks can be either signed or unsigned but not mixed within a function element. Similarly, the operand widths cannot be mixed within a block. In the LatticeECP2/M family the DSP elements can be concatenated.

The resources in each sysDSP block can be configured to support the following elements:

- MULT (Multiply)
- MAC (Multiply, Accumulate)
- MULTADDSUB (Multiply, Addition/Subtraction)
- MULTADDSUBSUM (Multiply, Addition/Subtraction, Accumulate)

The number of elements available on each block depends in the width selected from the three available options x9, x18, and x36. A number of these elements are concatenated for highly parallel implementations of DSP functions. Table 2-7 shows the capabilities of the block.

Table 2-7. Maximum Number of Elements in a Block

Width of Multiply	x9	x18	x36
MULT	8	4	1
MAC	2	2	—
MULTADDSUB	4	2	—
MULTADDSUBSUM	2	1	—

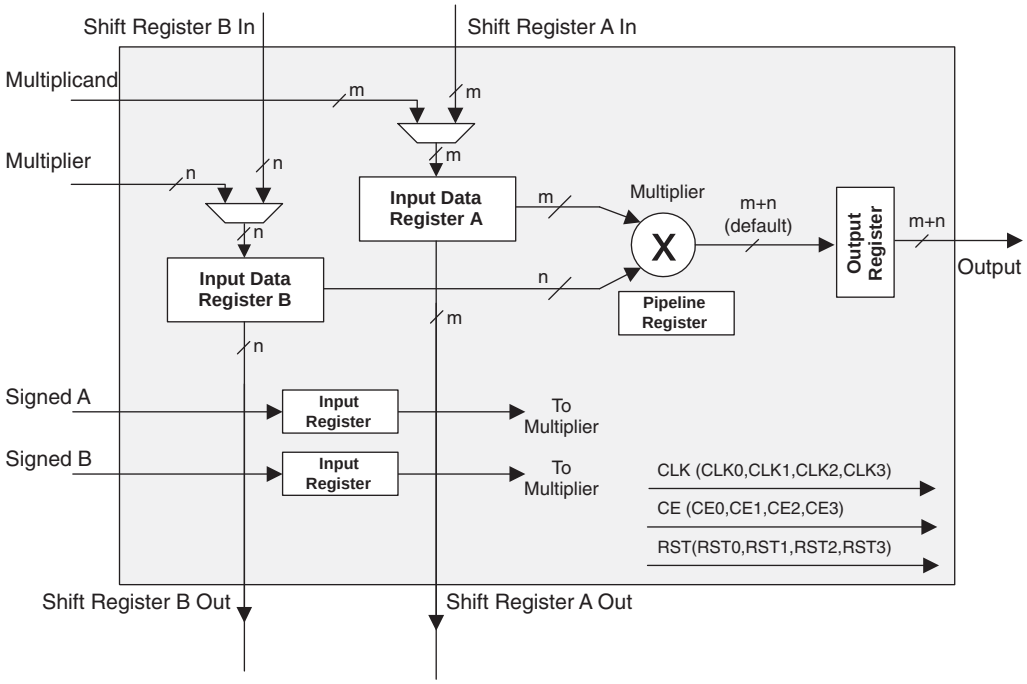
Some options are available in four elements. The input register in all the elements can be directly loaded or can be loaded as a shift register from previous operand registers. By selecting “dynamic operation” the following operations are possible:

- In the ‘Signed/Unsigned’ options the operands can be switched between signed and unsigned on every cycle.
- In the ‘Add/Sub’ option the Accumulator can be switched between addition and subtraction on every cycle.
- The loading of operands can switch between parallel and serial operations.

MULT sysDSP Element

This multiplier element implements a multiply with no addition or accumulator nodes. The two operands, A and B, are multiplied and the result is available at the output. The user can enable the input/output and pipeline registers. Figure 2-23 shows the MULT sysDSP element.

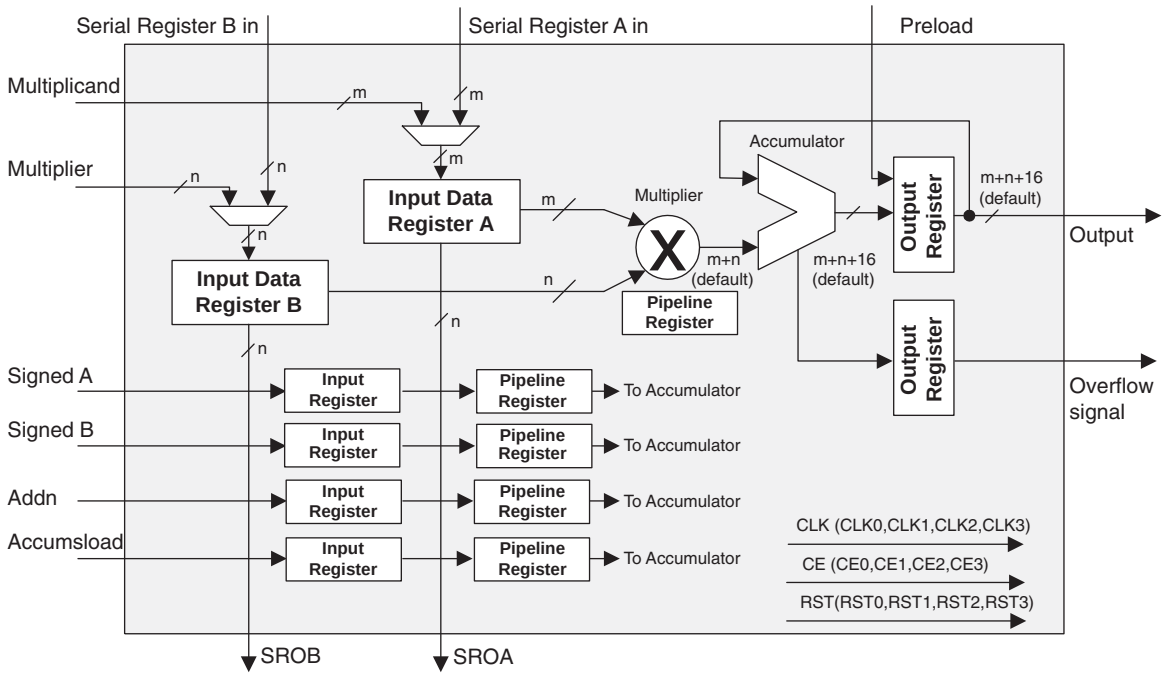
Figure 2-23. MULT sysDSP Element



MAC sysDSP Element

In this case, the two operands, A and B, are multiplied and the result is added with the previous accumulated value. This accumulated value is available at the output. The user can enable the input and pipeline registers, but the output register is always enabled. The output register is used to store the accumulated value. The Accumulators in the DSP blocks in the LatticeECP2/M family can be initialized dynamically. A registered overflow signal is also available. The overflow conditions are provided later in this document. Figure 2-24 shows the MAC sysDSP element.

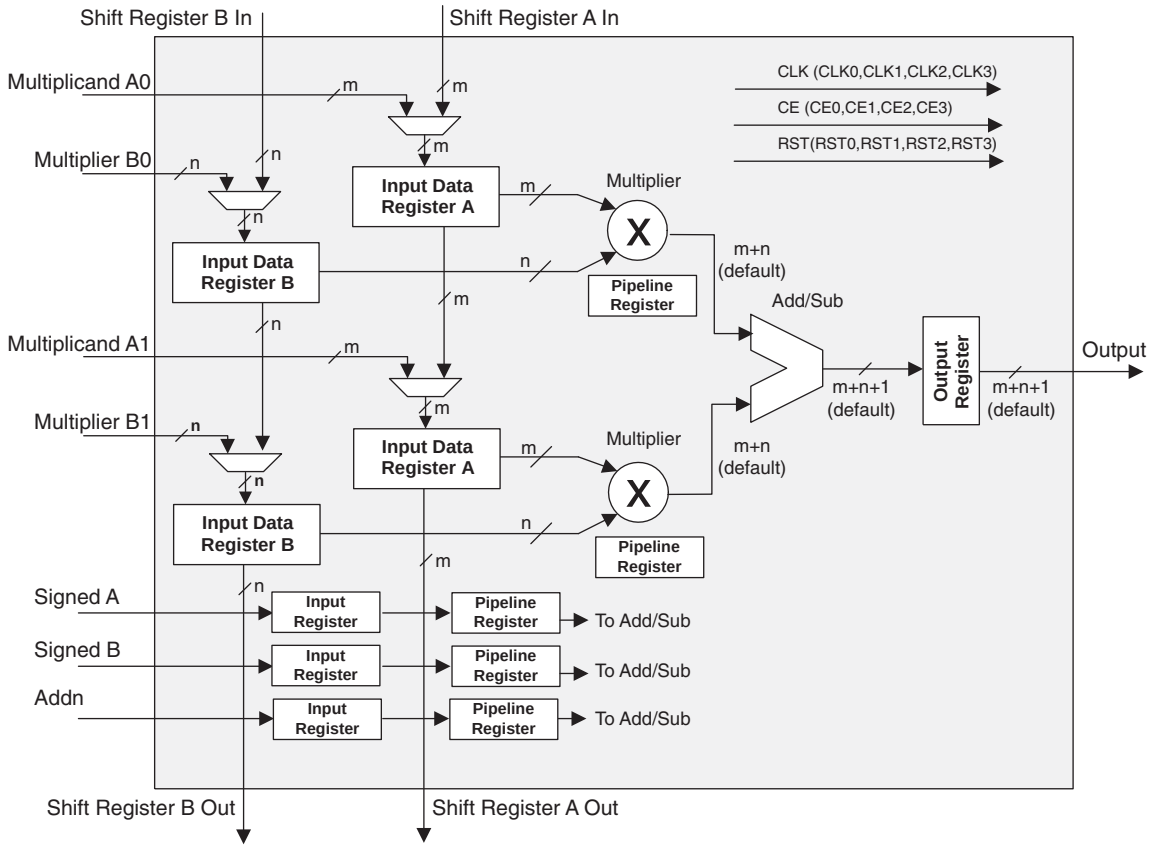
Figure 2-24. MAC sysDSP



MULTADDSUB sysDSP Element

In this case, the operands A0 and B0 are multiplied and the result is added/subtracted with the result of the multiplier operation of operands A1 and A2. The user can enable the input, output and pipeline registers. Figure 2-25 shows the MULTADDSUB sysDSP element.

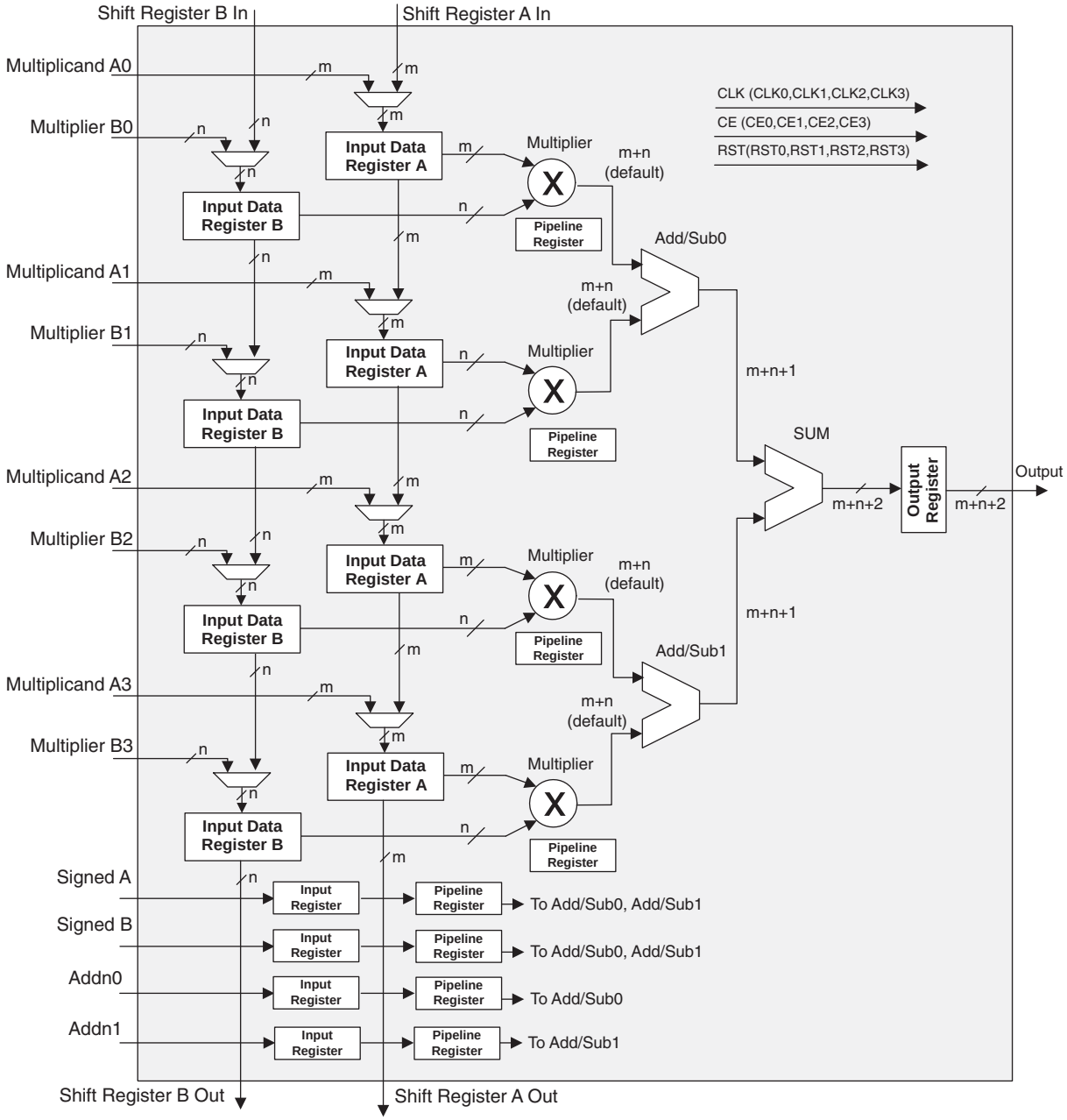
Figure 2-25. MULTADDSUB



MULTADDSUBSUM sysDSP Element

In this case, the operands A0 and B0 are multiplied and the result is added/subtracted with the result of the multiplier operation of operands A1 and B1. Additionally the operands A2 and B2 are multiplied and the result is added/subtracted with the result of the multiplier operation of operands A3 and B3. The result of both addition/subtraction are added in a summation block. The user can enable the input, output and pipeline registers. Figure 2-26 shows the MULTADDSUBSUM sysDSP element.

Figure 2-26. MULTADDSUBSUM



Clock, Clock Enable and Reset Resources

Global Clock, Clock Enable and Reset signals from routing are available to every DSP block. Four Clock, Reset and Clock Enable signals are selected for the sysDSP block. From four clock sources (CLK0, CLK1, CLK2, CLK3)

one clock is selected for each input register, pipeline register and output register. Similarly Clock enable (CE) and Reset (RST) are selected from their four respective sources (CE0, CE1, CE2, CE3 and RST0, RST1, RST2, RST3) at each input register, pipeline register and output register.

Signed and Unsigned with Different Widths

The DSP block supports different widths of signed and unsigned multipliers besides x9, x18 and x36 widths. For unsigned operands, unused upper data bits should be filled to create a valid x9, x18 or x36 operand. For signed two's complement operands, sign extension of the most significant bit should be performed until x9, x18 or x36 width is reached. Table 2-8 provides an example of this.

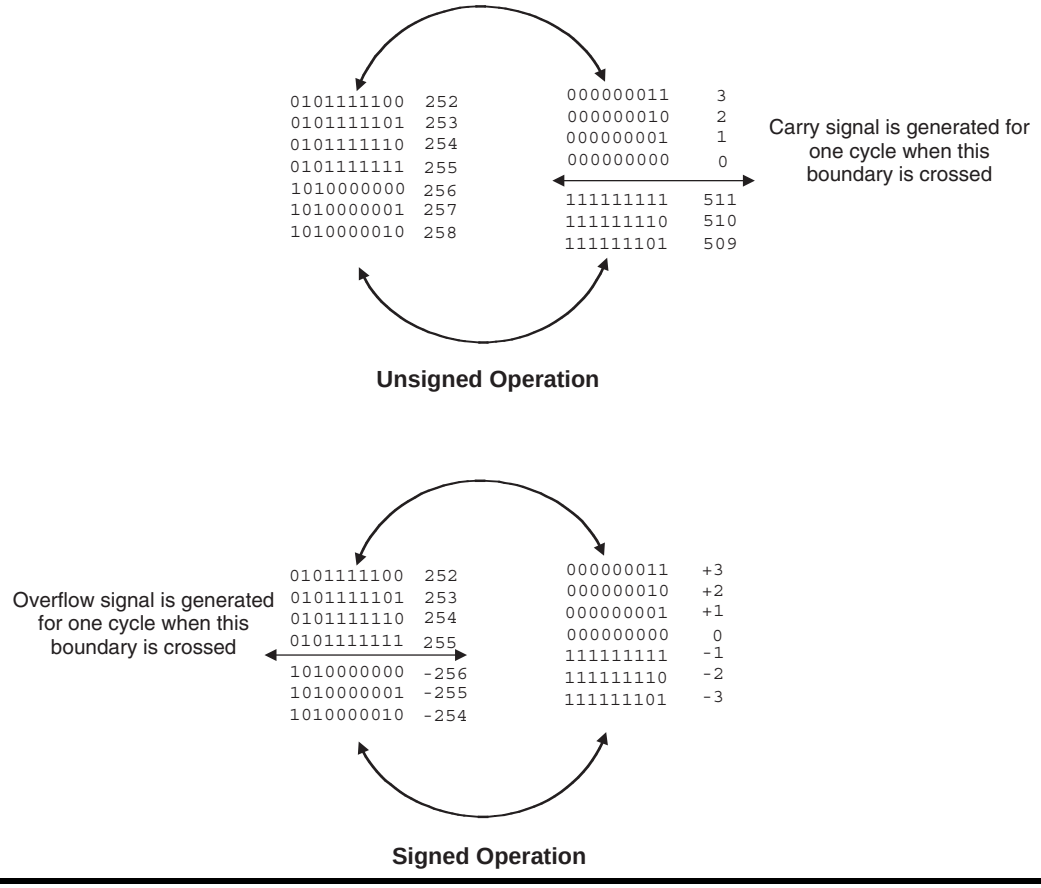
Table 2-8. Sign Extension Example

Number	Unsigned	Unsigned 9-bit	Unsigned 18-bit	Signed	Two's Complement Signed 9 Bits	Two's Complement Signed 18 Bits
+5	0101	000000101	000000000000000101	0101	000000101	000000000000000101
-6	N/A	N/A	N/A	1010	111111010	111111111111111010

OVERFLOW Flag from MAC

The sysDSP block provides an overflow output to indicate that the accumulator has overflowed. When two unsigned numbers are added and the result is a smaller number than the accumulator, "roll-over" is said to have occurred and an overflow signal is indicated. When two positive numbers are added with a negative sum and when two negative numbers are added with a positive sum, then the accumulator "roll-over" is said to have occurred and an overflow signal is indicated. Note that when overflow occurs the overflow flag is present for only one cycle. By counting these overflow pulses in FPGA logic, larger accumulators can be constructed. The conditions overflow signals for signed and unsigned operands are listed in Figure 2-27.

Figure 2-27. Accumulator Overflow/Underflow



IPexpress™

The user can access the sysDSP block via the IPexpress tool, which provides the option to configure each DSP module (or group of modules) or by direct HDL instantiation. In addition, Lattice has partnered with The MathWorks® to support instantiation in the Simulink® tool, a graphical simulation environment. Simulink works with Diamond to dramatically shorten the DSP design cycle in Lattice FPGAs.

Optimized DSP Functions

Lattice provides a library of optimized DSP IP functions. Some of the IP cores planned for the LatticeECP2/M DSP include the Bit Correlator, Fast Fourier Transform, Finite Impulse Response (FIR) Filter, Reed-Solomon Encoder/Decoder, Turbo Encoder/Decoder and Convolutional Encoder/Decoder. Please contact Lattice to obtain the latest list of available DSP IP cores.

Resources Available in the LatticeECP2/M Family

Table 2-9 shows the maximum number of multipliers for each member of the LatticeECP2/M family. Table 2-10 shows the maximum available EBR RAM Blocks in each LatticeECP2/M device. EBR blocks, together with Distributed RAM can be used to store variables locally for fast DSP operations.

Table 2-9. Maximum Number of DSP Blocks in the LatticeECP2/M Family

Device	DSP Block	9x9 Multiplier	18x18 Multiplier	36x36 Multiplier
ECP2-6	3	24	12	3
ECP2-12	6	48	24	6
ECP2-20	7	56	28	7
ECP2-35	8	64	32	8
ECP2-50	18	144	72	18
ECP2-70	22	176	88	22
ECP2M20	6	48	24	6
ECP2M35	8	64	32	8
ECP2M50	22	176	88	22
ECP2M70	24	192	96	24
ECP2M100	42	336	168	42

Table 2-10. Embedded SRAM in the LatticeECP2/M Family

Device	EBR SRAM Block	Total EBR SRAM (Kbits)
ECP2-6	3	55
ECP2-12	12	221
ECP2-20	15	277
ECP2-35	18	332
ECP2-50	21	387
ECP2-70	60	1106
ECP2M20	66	1217
ECP2M35	114	2101
ECP2M50	225	4147
ECP2M70	246	4534
ECP2M100	288	5308

LatticeECP2/M DSP Performance

Table 2-11 lists the maximum performance in millions of MAC operations per second (MMAC) for each member of the LatticeECP2/M family.

Table 2-11. DSP Performance

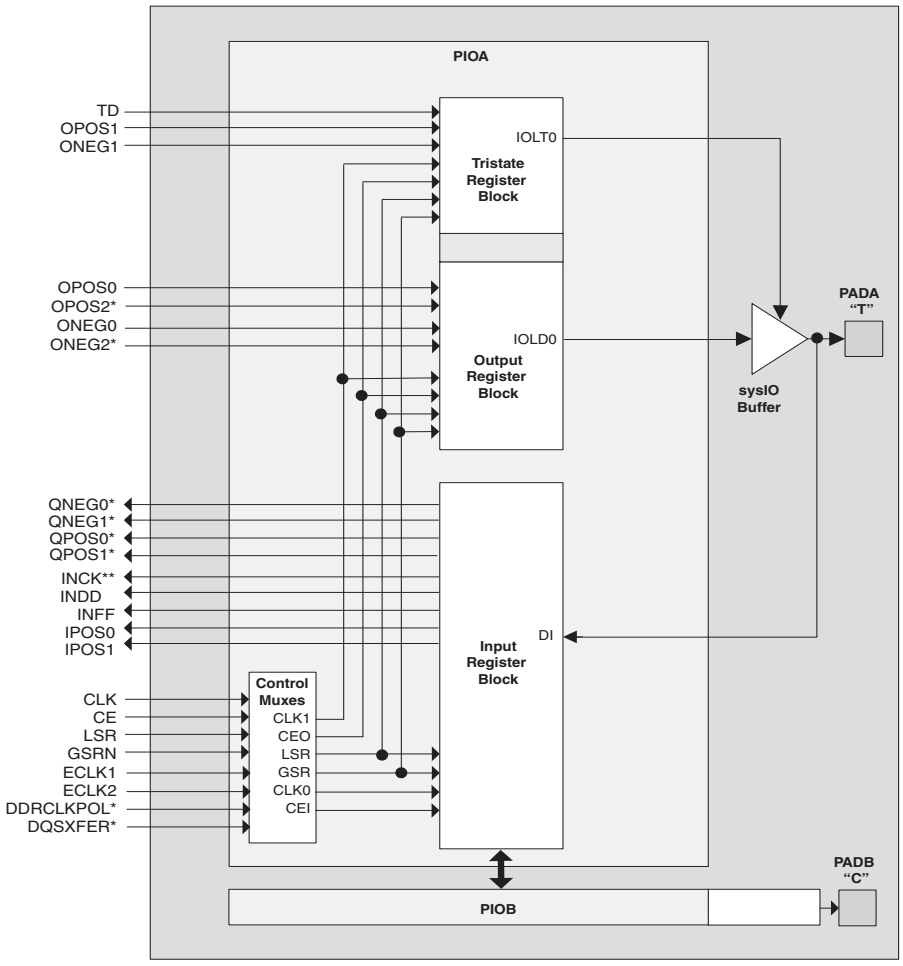
Device	DSP Block	DSP Performance GMAC
ECP2-6	3	3.9
ECP2-12	6	7.8
ECP2-20	7	9.1
ECP2-35	8	10.4
ECP2-50	18	23.4
ECP2-70	22	28.6
ECP2M20	6	7.8
ECP2M35	8	10.4
ECP2M50	22	28.6
ECP2M70	24	31.2
ECP2M100	42	54.6

For further information about the sysDSP block, please see the list of additional technical information at the end of this data sheet.

Programmable I/O Cells (PIC)

Each PIC contains two PIOs connected to their respective sysI/O buffers as shown in Figure 2-28. The PIO Block supplies the output data (DO) and the tri-state control signal (TO) to the sysI/O buffer and receives input from the buffer. Table 2-12 provides the PIO signal list.

Figure 2-28. PIC Diagram



*Signals are available on left/right/bottom edges only.
** Selected blocks.

Two adjacent PIOs can be joined to provide a differential I/O pair (labeled as “T” and “C”) as shown in Figure 2-28. The PAD Labels “T” and “C” distinguish the two PIOs. Approximately 50% of the PIO pairs on the left and right edges of the device can be configured as true LVDS outputs. All I/O pairs can operate as inputs.

Table 2-12. PIO Signals List

Name	Type	Description
CE0, CE1	Control from the core	Clock enables for input and output block flip-flops
CLK0, CLK1	Control from the core	System clocks for input and output blocks
ECLK1, ECLK2	Control from the core	Fast edge clocks
LSR	Control from the core	Local Set/Reset
GSRN	Control from routing	Global Set/Reset (active low)
INCK[2]	Input to the core	Input to Primary Clock Network or PLL reference inputs
DQS	Input to PIO	DQS signal from logic (routing) to PIO
INDD	Input to the core	Unregistered data input to core
INFF	Input to the core	Registered input on positive edge of the clock (CLK0)
IPOS0, IPOS1	Input to the core	Double data rate registered inputs to the core
QPOS0[1], QPOS1[1]	Input to the core	Gearbox pipelined inputs to the core
QNEG0[1], QNEG1[1]	Input to the core	Gearbox pipelined inputs to the core
OPOS0, ONEG0, OPOS2, ONEG2	Output data from the core	Output signals from the core for SDR and DDR operation
OPOS1 ONEG1	Tristate control from the core	Signals to Tristate Register block for DDR operation
DEL[3:0]	Control from the core	Dynamic input delay control bits
TD	Tristate control from the core	Tristate signal from the core used in SDR operation
DDRCLKPOL	Control from clock polarity bus	Controls the polarity of the clock (CLK0) that feed the DDR input block
DQSXFER	Control from core	Controls signal to the Output block

1. Signals available on left/right/bottom only.
2. Selected I/O.

PIO

The PIO contains four blocks: an input register block, output register block, tristate register block and a control logic block. These blocks contain registers for operating in a variety of modes along with the necessary clock and selection logic.

Input Register Block

The input register blocks for PIOs in left, right and bottom edges contain delay elements and registers that can be used to condition high-speed interface signals, such as DDR memory interfaces and source synchronous interfaces, before they are passed to the device core. Figure 2-29 shows the diagram of the input register block for left, right and bottom edges. The input register block for the top edge contains one memory element to register the input signal as shown in Figure 2-30. The following description applies to the input register block for PIOs in the left, right and bottom edges of the device.

Input signals are fed from the sysI/O buffer to the input register block (as signal DI). If desired, the input signal can bypass the register and delay elements and be used directly as a combinatorial signal (INDD), a clock (INCK) and, in selected blocks, the input to the DQS delay block. If an input delay is desired, designers can select either a fixed delay or a dynamic delay DEL[3:0]. The delay, if selected, reduces input register hold time requirements when using a global clock.

The input block allows three modes of operation. In the single data rate (SDR) the data is registered, by one of the registers in the single data rate sync register block, with the system clock. In DDR Mode, two registers are used to sample the data on the positive and negative edges of the DQS signal, creating two data streams, D0 and D1. These two data streams are synchronized with the system clock before entering the core. Further discussion on this topic is in the DDR Memory section of this data sheet.

By combining input blocks of the complementary PIOs and sharing some registers from output blocks, a gearbox function can be implemented, which takes a double data rate signal applied to PIOA and converts it as four data streams, IPOS0A, IPOS1A, IPOS0B and IPOS1B. Figure 2-29 shows the diagram using this gearbox function. For more information about this topic, please see information regarding additional documentation at the end of this data sheet.

The signal DDRCLKPOL controls the polarity of the clock used in the synchronization registers. It ensures adequate timing when data is transferred from the DQS to the system clock domain. For further information about this topic, see the DDR Memory section of this data sheet.

Figure 2-29. Input Register Block for Left, Right and Bottom Edges

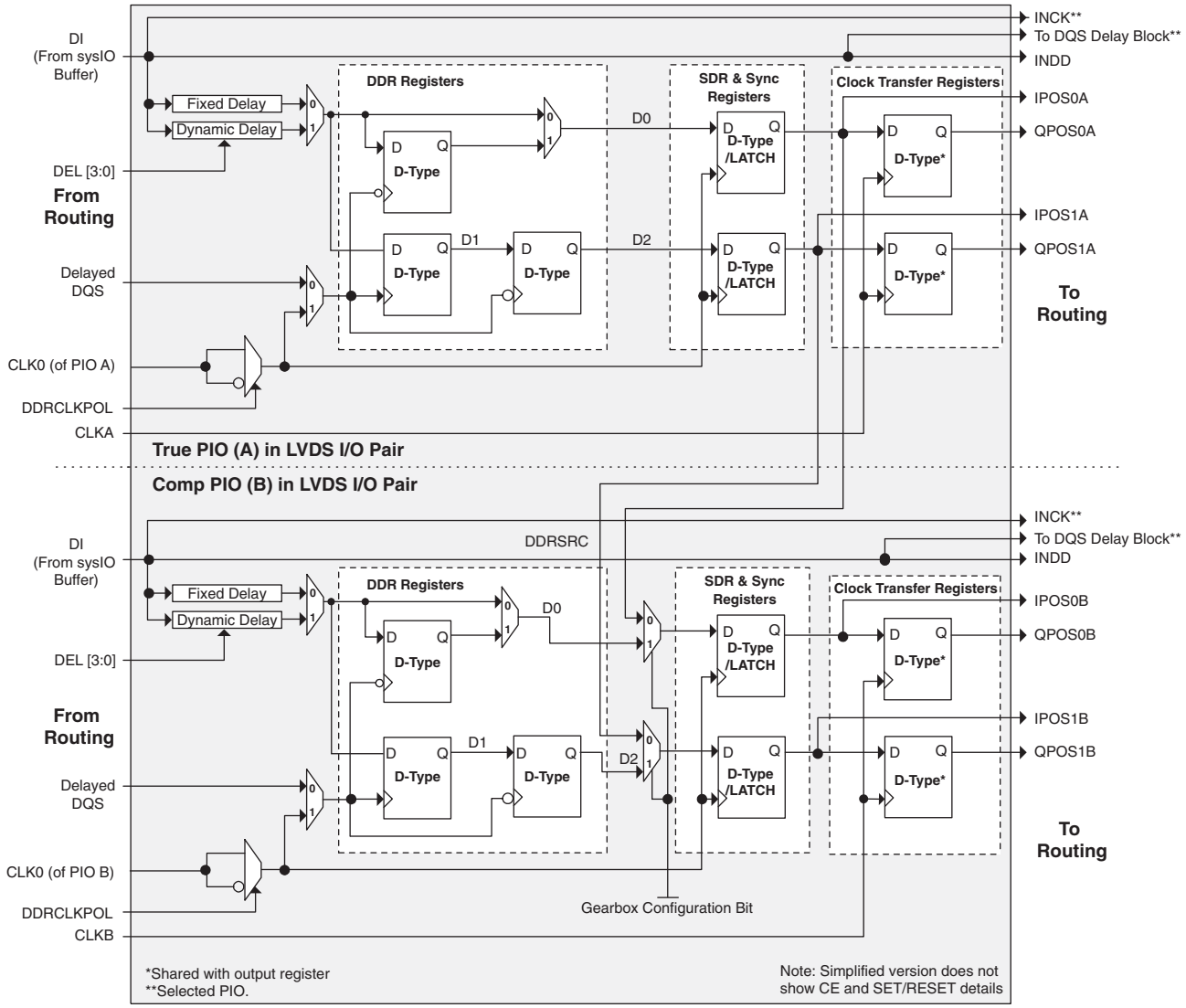
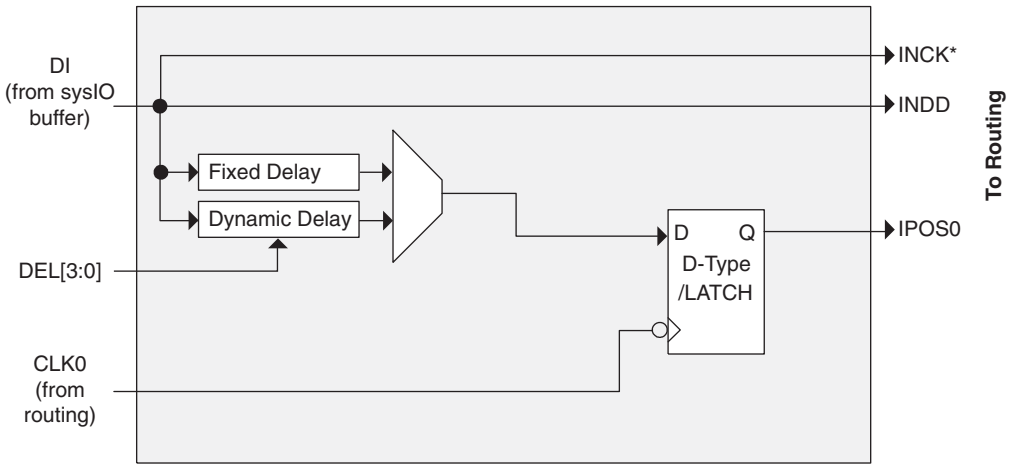


Figure 2-30. Input Register Block Top Edge



Note: Simplified version does not show CE and SET/RESET details.
*On selected blocks.

Output Register Block

The output register block provides the ability to register signals from the core of the device before they are passed to the sysI/O buffers. The blocks on the PIOs on the left, right and bottom contain a register for SDR operation that is combined with an additional latch for DDR operation. Figure 2-31 shows the diagram of the Output Register Block for PIOs on the left, right and the bottom edges. Figure 2-32 shows the diagram of the Output Register Block for PIOs on the top edge of the device.

In SDR mode, ONEG0 feeds one of the flip-flops that then feeds the output. The flip-flop can be configured as a D-type or latch. In DDR mode, ONEG0 and OPOS0 are fed into registers on the positive edge of the clock. Then at the next clock cycle this registered OPOS0 is latched. A multiplexer running off the same clock selects the correct register for feeding to the output (D0).

By combining the output blocks of the complementary PIOs and sharing some registers from input blocks, a gearbox function can be implemented, that takes four data streams: ONEG0A, ONEG1A, ONEG1B and ONEG1B. Figure 2-32 shows the diagram using this gearbox function. For more information about this topic, please see information regarding additional documentation at the end of this data sheet.

Figure 2-31. Output and Tristate Block for Left, Right and Bottom Edges

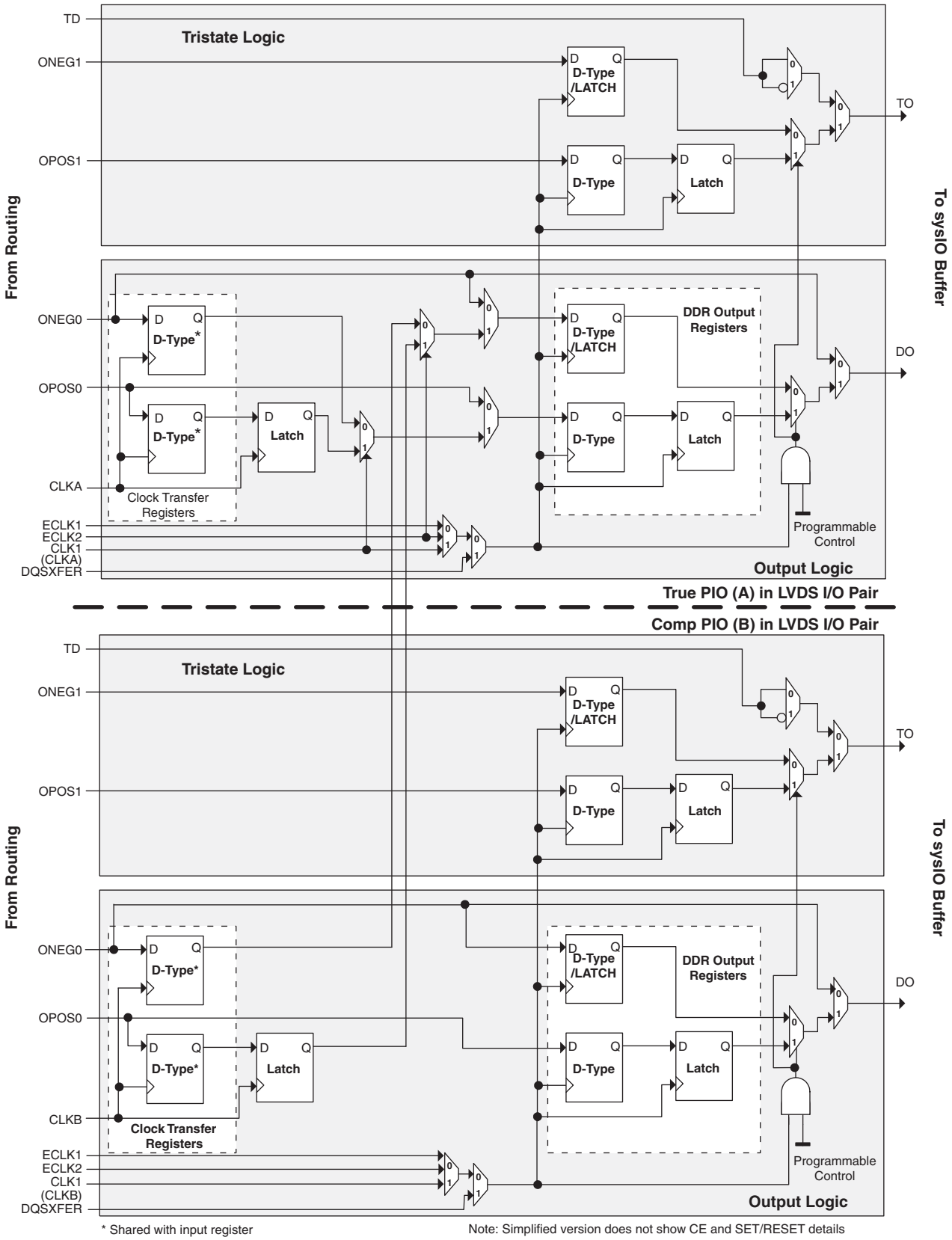
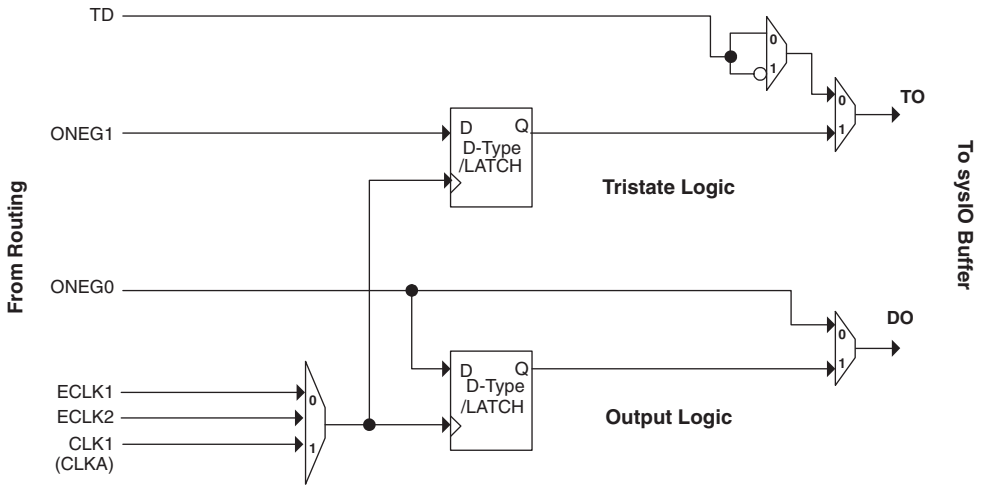


Figure 2-32. Output and Tristate Block, Top Edge



Note: Simplified version does not show CE and SET/RESET details.

Tristate Register Block

The tristate register block provides the ability to register tri-state control signals from the core of the device before they are passed to the sysI/O buffers. The block contains a register for SDR operation and an additional latch for DDR operation. Figure 2-31 shows the diagram of the Tristate Register Block with the Output Block for the left, right and bottom edges and Figure 2-32 shows the diagram of the Tristate Register Block with the Output Block for the top edge.

In SDR mode, ONEG1 feeds one of the flip-flops that then feeds the output. The flip-flop can be configured a D-type or latch. In DDR mode, ONEG1 and OPOS1 are fed into registers on the positive edge of the clock. Then in the next clock the registered OPOS1 is latched. A multiplexer running off the same clock cycle selects the correct register for feeding to the output (D0).

Control Logic Block

The control logic block allows the selection and modification of control signals for use in the PIO block. A clock is selected from one of the clock signals provided from the general purpose routing, one of the edge clocks (ECLK1/ ECLK2) and a DQS signal provided from the programmable DQS pin and provided to the input register block. The clock can optionally be inverted.

DDR Memory Support

Certain PICs have additional circuitry to allow the implementation of high speed source synchronous and DDR memory interfaces. The support varies by the edge of the device as detailed below.

Left and Right Edges

PICs on these edges have registered elements that support DDR memory interfaces. One of every 16 PIOs contains a delay element to facilitate the generation of DQS signals. The DQS signal feeds the DQS bus that spans the set of 16 PIOs. Figure 2-33 shows the assignment of DQS pins in each set of 16 PIOs.

Bottom Edge

PICs on the bottom edge have registered elements that support DDR memory interfaces. One of every 18 PIOs contains a delay element to facilitate the generation of DQS signals. The DQS signal feeds the DQS bus that spans the set of 18 PIOs. Figure 2-34 shows the assignment of DQS pins in each set of 18 PIOs.

Top Edge

The PICs on the top edge are different from PIOs on the left, right and bottom edges. PIOs on this edge do not have DDR registers or DQS signals.

The exact DQS pins are shown in a dual function in the Logic Signal Connections table in this data sheet. Additional detail is provided in the Signal Descriptions table. The DQS signal from the bus is used to strobe the DDR data from the memory into input register blocks. Interfaces on the left and right edges are designed for DDR memories that support 16 bits of data, whereas interfaces on the bottom are designed for memories that support 18 bits of data.

Figure 2-33. DQS Input Routing for the Left and Right Edges of the Device

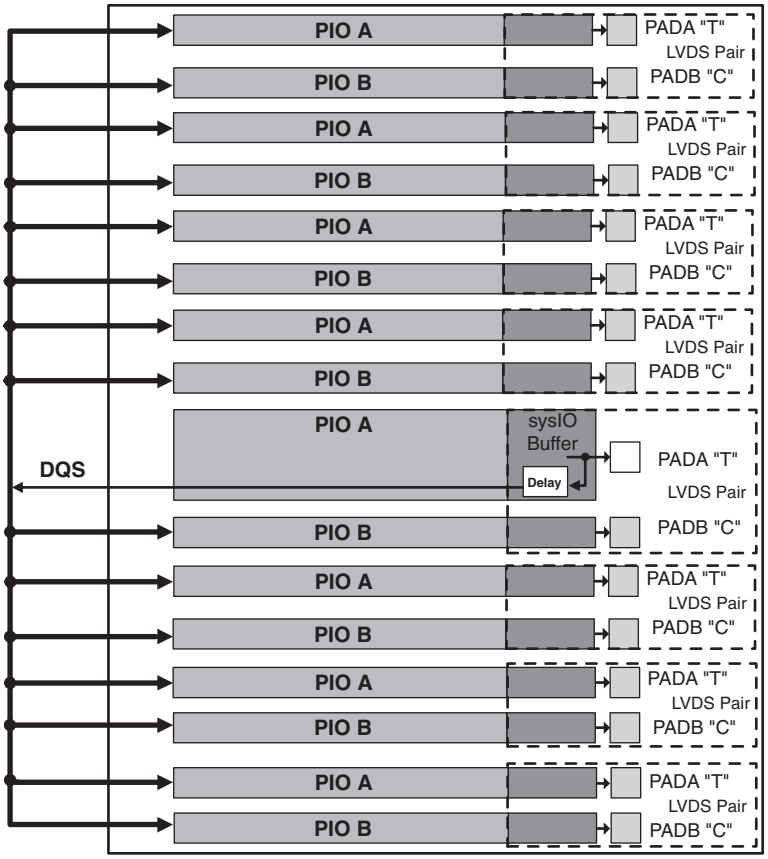
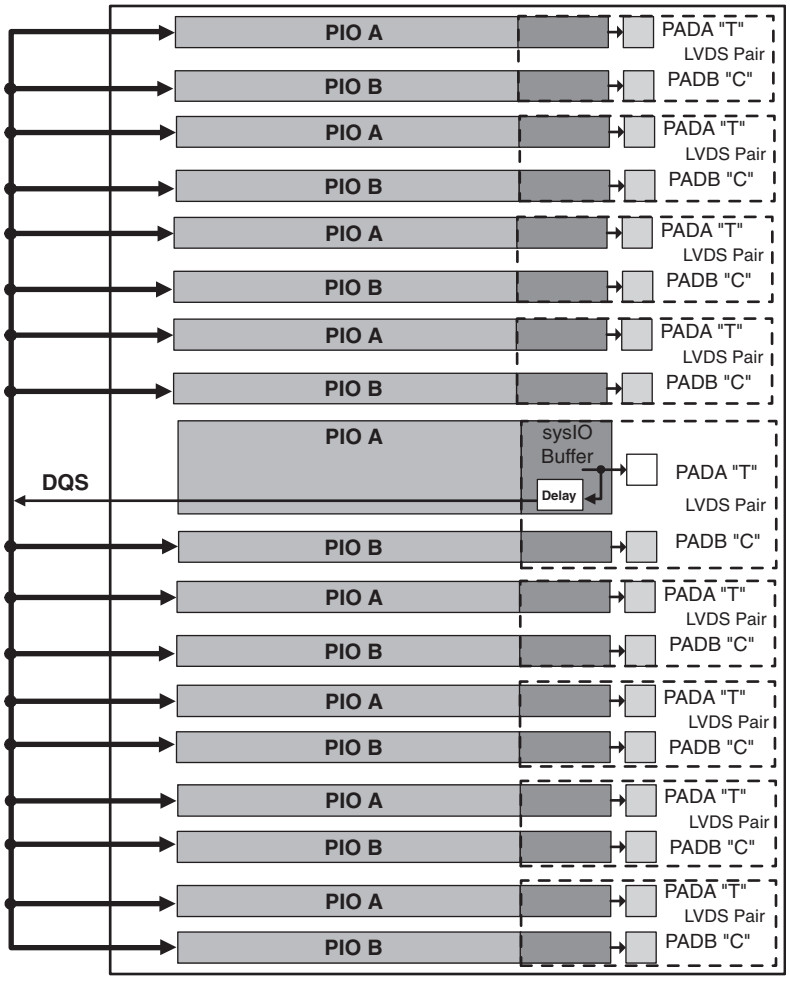


Figure 2-34. DQS Input Routing for the Bottom Edge of the Device



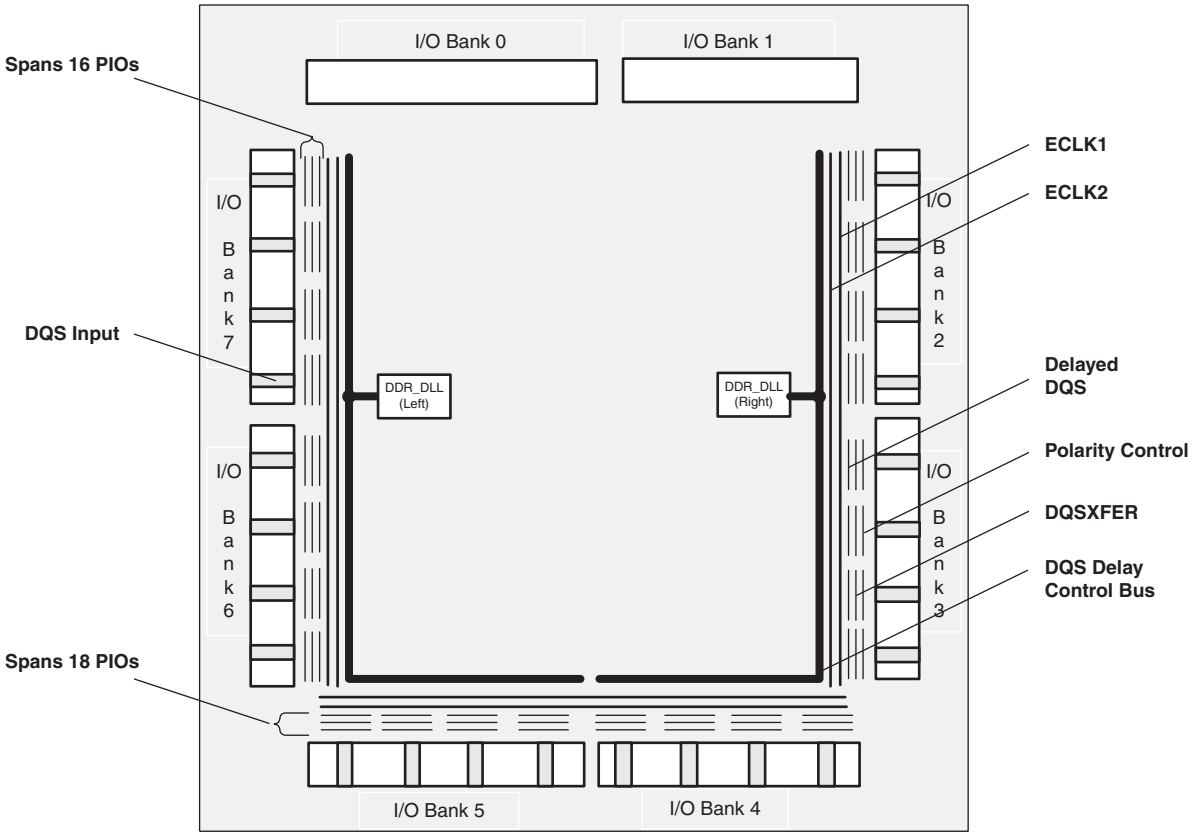
DLL Calibrated DQS Delay Block

Source synchronous interfaces generally require the input clock to be adjusted in order to correctly capture data at the input register. For most interfaces a PLL is used for this adjustment. However, in DDR memories the clock (referred to as DQS) is not free-running so this approach cannot be used. The DQS Delay block provides the required clock alignment for DDR memory interfaces.

The DQS signal (selected PIOs only, as shown in Figure 2-35) feeds from the PAD through a DQS delay element to a dedicated DQS routing resource. The DQS signal also feeds polarity control logic, which controls the polarity of the clock to the sync registers in the input register blocks. Figure 2-35 and Figure 2-36 show how the DQS transition signals are routed to the PIOs.

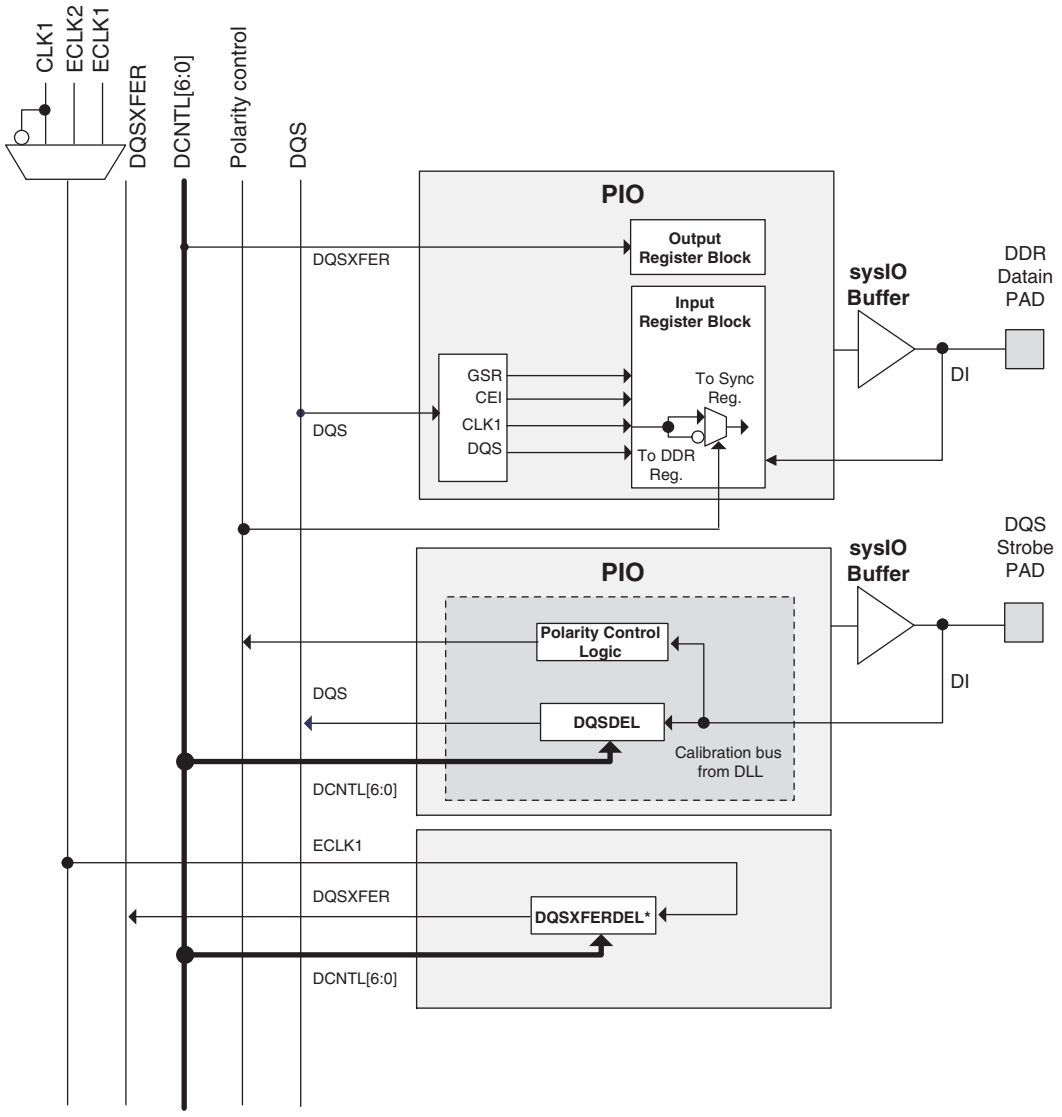
The temperature, voltage and process variations of the DQS delay block are compensated by a set of calibration (6-bit bus) signals from two dedicated DLLs (DDR_DLL) on opposite sides of the device. Each DLL compensates DQS delays in its half of the device as shown in Figure 2-35. The DLL loop is compensated for temperature, voltage and process variations by the system clock and feedback loop.

Figure 2-35. Edge Clock, DLL Calibration and DQS Local Bus Distribution



Note: Bank 8 is not shown.

Figure 2-36. DQS Local Bus



*DQSXFERDEL shifts ECLK1 by 90% and is not associated with a particular PIO.

Polarity Control Logic

In a typical DDR Memory interface design, the phase relationship between the incoming delayed DQS strobe and the internal system clock (during the READ cycle) is unknown.

The LatticeECP2/M family contains dedicated circuits to transfer data between these domains. To prevent set-up and hold violations, at the domain transfer between DQS (delayed) and the system clock, a clock polarity selector is used. This changes the edge on which the data is registered in the synchronizing registers in the input register block. This requires evaluation at the start of each READ cycle for the correct clock polarity.

Prior to the READ operation in DDR memories, DQS is in tristate (pulled by termination). The DDR memory device drives DQS low at the start of the preamble state. A dedicated circuit detects the first DQS rising edge after the preamble state. This signal is used to control the polarity of the clock to the synchronizing registers.

DQSXFER

LatticeECP2/M devices provide a DQSXFER signal to the output buffer to assist it in data transfer to DDR memories that require DQS strobe be shifted 90°. This shifted DQS strobe is generated by the DQSDEL block. The DQSXFER signal runs the span of the data bus.

sysI/O Buffer

Each I/O is associated with a flexible buffer referred to as a sysI/O buffer. These buffers are arranged around the periphery of the device in groups referred to as banks. The sysI/O buffers allow users to implement the wide variety of standards that are found in today's systems including LVCMOS, SSTL, HSTL, LVDS and LVPECL.

sysI/O Buffer Banks

LatticeECP2/M devices have nine sysI/O buffer banks: eight banks for user I/Os arranged two per side. The ninth sysI/O buffer bank (Bank 8) is located adjacent to Bank 3 and has dedicated/shared I/Os for configuration. When a shared pin is not used for configuration it is available as a user I/O. Each bank is capable of supporting multiple I/O standards. Each sysI/O bank has its own I/O supply voltage (V_{CCIO}). In addition, each bank, except Bank 8, has voltage references, V_{REF1} and V_{REF2}, which allow it to be completely independent from the others. Bank 8 shares two voltage references, V_{REF1} and V_{REF2}, with Bank 3. Figure 2-37 shows the nine banks and their associated supplies.

In LatticeECP2/M devices, single-ended output buffers and ratioed input buffers (LVTTL, LVCMOS and PCI) are powered using V_{CCIO}. LVTTL, LVCMOS33, LVCMOS25 and LVCMOS12 can also be set as fixed threshold inputs independent of V_{CCIO}.

Each bank can support up to two separate V_{REF} voltages, V_{REF1} and V_{REF2}, that set the threshold for the referenced input buffers. Some dedicated I/O pins in a bank can be configured to be a reference voltage supply pin. Each I/O is individually configurable based on the bank's supply and reference voltages.

Figure 2-37. LatticeECP2 Banks

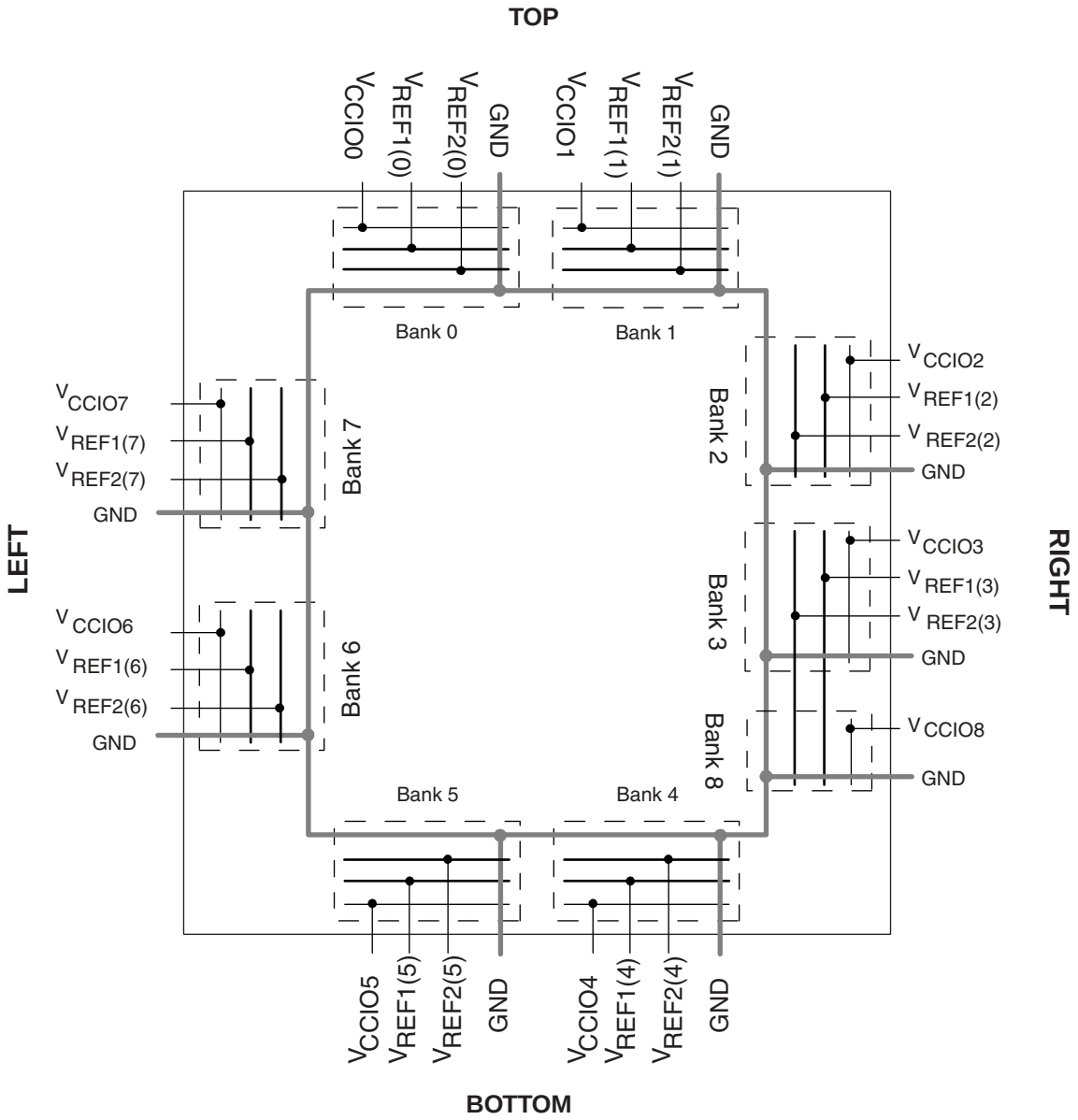
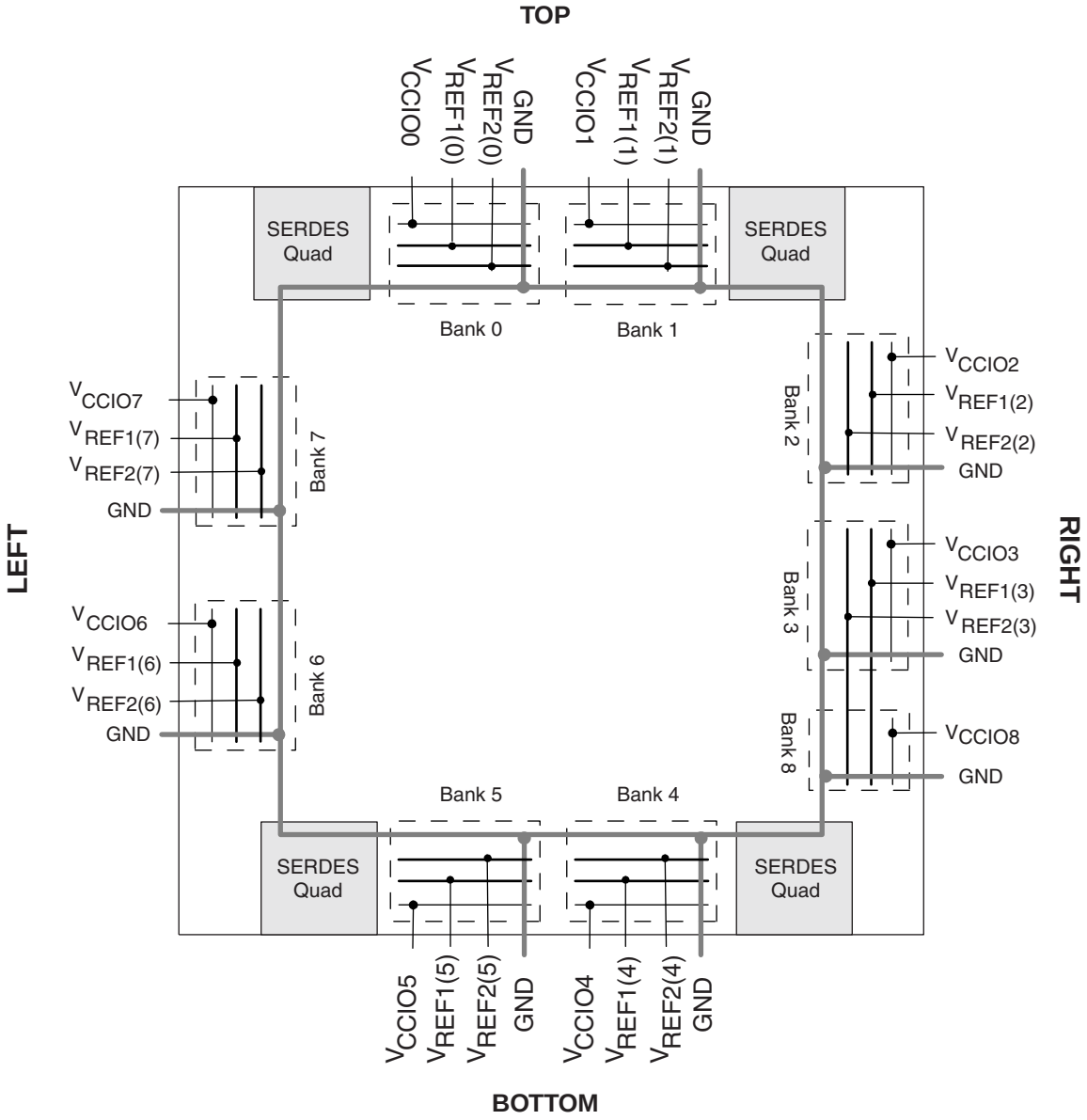


Figure 2-38. LatticeECP2M Banks



LatticeECP2/M devices contain two types of sysI/O buffer pairs.

1. **Top (Bank 0 and Bank 1) sysI/O Buffer Pairs (Single-Ended Outputs Only)**
   The sysI/O buffer pairs in the top banks of the device consist of two single-ended output drivers and two sets of single-ended input buffers (both ratioed and referenced). One of the referenced input buffers can also be configured as a differential input.

   The two pads in the pair are described as “true” and “comp”, where the true pad is associated with the positive side of the differential input buffer and the comp (complementary) pad is associated with the negative side of the differential input buffer.

2. **Bottom (Bank 4 and Bank 5) sysI/O Buffer Pairs (Single-Ended Outputs Only)**
   The sysI/O buffer pairs in the bottom banks of the device consist of two single-ended output drivers and two

sets of single-ended input buffers (both ratioed and referenced). One of the referenced input buffers can also be configured as a differential input.

The two pads in the pair are described as “true” and “comp”, where the true pad is associated with the positive side of the differential input buffer and the comp (complementary) pad is associated with the negative side of the differential input buffer.

3. **Left and Right (Banks 2, 3, 6 and 7) sysI/O Buffer Pairs (50% Differential and 100% Single-Ended Outputs)**
   The sysI/O buffer pairs in the left and right banks of the device consist of two single-ended output drivers, two sets of single-ended input buffers (both ratioed and referenced) and one differential output driver. One of the referenced input buffers can also be configured as a differential input. In these banks the two pads in the pair are described as “true” and “comp”, where the true pad is associated with the positive side of the differential I/O, and the comp (complementary) pad is associated with the negative side of the differential I/O.

   LVDS differential output drivers are available on 50% of the buffer pairs on the left and right banks.

4. **Bank 8 sysI/O Buffer Pairs (Single-Ended Outputs, Only on Shared Pins When Not Used by Configuration)**
   The sysI/O buffers in Bank 8 consist of single-ended output drivers and single-ended input buffers (both ratioed and referenced). The referenced input buffer can also be configured as a differential input.

   The two pads in the pair are described as “true” and “comp”, where the true pad is associated with the positive side of the differential input buffer and the comp (complementary) pad is associated with the negative side of the differential input buffer.

In LatticeECP2 devices, only the I/Os on the bottom banks have programmable PCI clamps. In LatticeECP2M devices, the I/Os on the left and bottom banks have programmable PCI clamps.

Typical sysI/O I/O Behavior During Power-up

The internal power-on-reset (POR) signal is deactivated when V_{CC}, V_{CCIO8} and V_{CCAUX} have reached satisfactory levels. After the POR signal is deactivated, the FPGA core logic becomes active. It is the user’s responsibility to ensure that all other V_{CCIO} banks are active with valid input logic levels to properly control the output logic states of all the I/O banks that are critical to the application. For more information about controlling the output logic state with valid input logic levels during power-up in LatticeECP2/M devices, see the list of additional technical documentation at the end of this data sheet.

The V_{CC} and V_{CCAUX} supply the power to the FPGA core fabric, whereas the V_{CCIO} supplies power to the I/O buffers. In order to simplify system design while providing consistent and predictable I/O behavior, it is recommended that the I/O buffers be powered-up prior to the FPGA core fabric. V_{CCIO} supplies should be powered-up before or together with the V_{CC} and V_{CCAUX} supplies.

Prior to and throughout programming of the FPGA, the I/O of the device have a weak-pullup resistor to V_{CCIO} on the input buffer and the output buffer is tri-stated. A pullup to V_{CCIO} is present on the input until the user programs the input differently in the FPGA design. See the DC Electrical Characteristics table of this data sheet. The pullup value will be between 20-30K ohms based on the V_{CCIO} voltage supplied on the board. This pullup will also remain active if the design does not use a particular I/O.

Supported sysI/O Standards

The LatticeECP2/M sysI/O buffer supports both single-ended and differential standards. Single-ended standards can be further subdivided into LVCMOS, LVTTL and other standards. The buffers support the LVTTL, LVCMOS 1.2V, 1.5V, 1.8V, 2.5V and 3.3V standards. In the LVCMOS and LVTTL modes, the buffer has individual configuration options for drive strength, bus maintenance (weak pull-up, weak pull-down, or a bus-keeper latch) and open drain. Other single-ended standards supported include SSTL and HSTL. Differential standards supported include LVDS, MLVDS, BLVDS, LVPECL, RSDS, differential SSTL and differential HSTL. Tables 2-13 and 2-14 show the I/

O standards (together with their supply and reference voltages) supported by LatticeECP2/M devices. For further information about utilizing the sysI/O buffer to support a variety of standards please see the the list of additional technical information at the end of this data sheet.

Table 2-13. Supported Input Standards

Input Standard	V_{REF} (Nom.)	V_{CCIO}[1] (Nom.)
Single Ended Interfaces		
LVTTL	—	—
LVCMOS33	—	—
LVCMOS25	—	—
LVCMOS18	—	1.8
LVCMOS15	—	1.5
LVCMOS12	—	—
PCI 33	—	3.3
HSTL18 Class I, II	0.9	—
HSTL15 Class I	0.75	—
SSTL3 Class I, II	1.5	—
SSTL2 Class I, II	1.25	—
SSTL18 Class I, II	0.9	—
Differential Interfaces		
Differential SSTL18 Class I, II	—	—
Differential SSTL2 Class I, II	—	—
Differential SSTL3 Class I, II	—	—
Differential HSTL15 Class I	—	—
Differential HSTL18 Class I, II	—	—
LVDS, MLVDS, LVPECL, BLVDS, RSDS	—	—

1 When not specified, V_{CCIO} can be set anywhere in the valid operating range (page 3-1).

Table 2-14. Supported Output Standards

Output Standard	Drive	V_{CCIO} (Nom.)
Single-ended Interfaces		
LVTTL	4mA, 8mA, 12mA, 16mA, 20mA	3.3
LVCMOS33	4mA, 8mA, 12mA 16mA, 20mA	3.3
LVCMOS25	4mA, 8mA, 12mA, 16mA, 20mA	2.5
LVCMOS18	4mA, 8mA, 12mA, 16mA	1.8
LVCMOS15	4mA, 8mA	1.5
LVCMOS12	2mA, 6mA	1.2
LVCMOS33, Open Drain	4mA, 8mA, 12mA 16mA, 20mA	—
LVCMOS25, Open Drain	4mA, 8mA, 12mA 16mA, 20mA	—
LVCMOS18, Open Drain	4mA, 8mA, 12mA 16mA	—
LVCMOS15, Open Drain	4mA, 8mA	—
LVCMOS12, Open Drain	2mA, 6mA	—
PCI33	N/A	3.3
HSTL18 Class I, II	N/A	1.8
HSTL15 Class I	N/A	1.5
SSTL3 Class I, II	N/A	3.3
SSTL2 Class I, II	N/A	2.5
SSTL18 Class I, II	N/A	1.8
Differential Interfaces		
Differential SSTL3, Class I, II	N/A	3.3
Differential SSTL2, Class I, II	N/A	2.5
Differential SSTL18, Class I, II	N/A	1.8
Differential HSTL18, Class I, II	N/A	1.8
Differential HSTL15, Class I	N/A	1.5
LVDS	N/A	2.5
MLVDS[1]	N/A	2.5
BLVDS[1]	N/A	2.5
LVPECL[1]	N/A	3.3
RSDS[1]	N/A	2.5
LVCMOS33D[1]	4mA, 8mA, 12mA, 16mA, 20mA	3.3

1. Emulated with external resistors. For more detail, please see information regarding additional technical documentation at the end of this data sheet.

Hot Socketing

LatticeECP2/M devices have been carefully designed to ensure predictable behavior during power-up and power-down. During power-up and power-down sequences, the I/Os remain in tri-state until the power supply voltage is high enough to ensure reliable operation. In addition, leakage into I/O pins is controlled within specified limits. This allows for easy integration with the rest of the system. These capabilities make the LatticeECP2/M ideal for many multiple power supply and hot-swap applications.

SERDES and PCS (Physical Coding Sublayer)

LatticeECP2M devices feature up to 16 channels of embedded SERDES arranged in quads at the corners of the devices. Figure 2-39 shows the position of the quad blocks in relation to the PFU array for LatticeECP2M70 and LatticeECP2M100 devices. Table 2-15 shows the location of Quads for all the devices.

Each quad contains four dedicated SERDES (Ch0 to Ch3) for high-speed, full-duplex serial data transfer. Each quad also has a PCS block that interfaces to the SERDES channels and contains digital logic to support an array of popular data protocols. PCS also contains logic to the interface to FPGA core.

Figure 2-39. SERDES Quads (LatticeECP2M70/LatticeECP2M100)

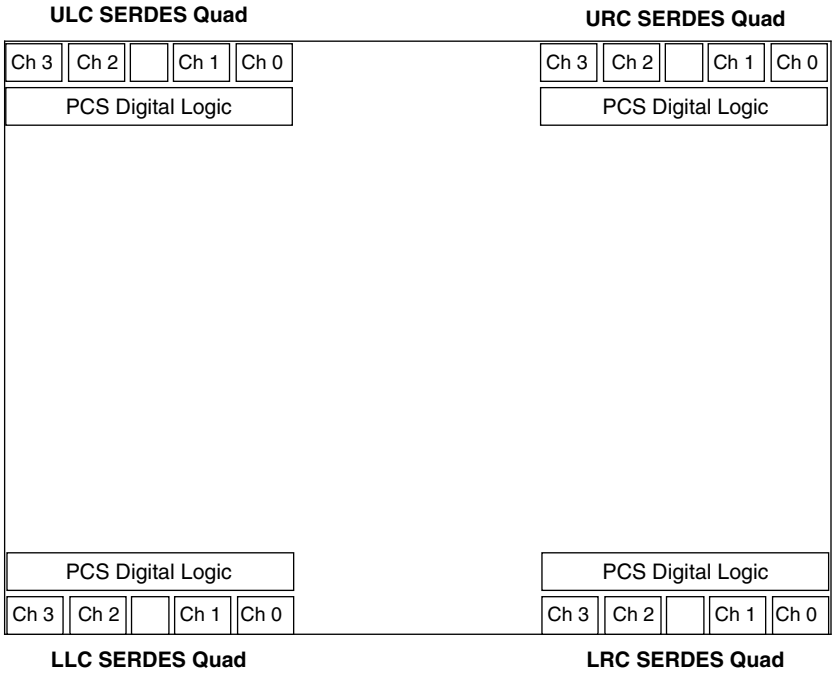


Table 2-15. Available SERDES Quads per LatticeECP2M Devices

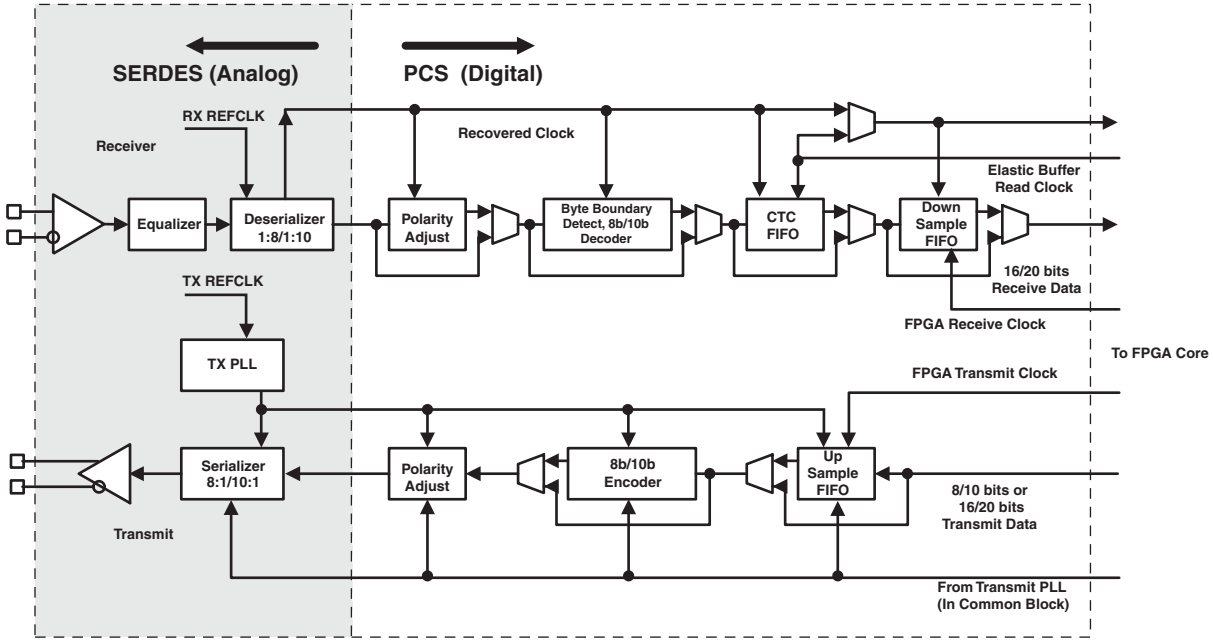
Device	URC Quad	ULC Quad	LRC Quad	LLC Quad
ECP2M20	Available	—	—	—
ECP2M35	Available	—	—	—
ECP2M50	Available	—	Available	—
ECP2M70	Available	Available	Available	Available
ECP2M100	Available	Available	Available	Available

SERDES Block

A differential receiver receives the serial encoded data stream, equalizes the signal, extracts the buried clock and de-serializes the data-stream before passing the 8- or 10-bit data to the PCS logic. The transmit channel receives the parallel (8- or 10-bit) encoded data, serializes the data and transmits the serial bit stream through the differential buffers. There is a single transmit clock per quad. Figure 2-40 shows a single channel SERDES and its interface to the PCS logic. Each SERDES receiver channel provides a recovered clock to the PCS block and to the FPGA core logic.

Each Transmit and Receive channel has its independent power supplies. The Output and Input buffers of each channel also have their own independent power supplies. In addition, there are separate power supplies for PLL, terminating resistor per quad.

Figure 2-40. Simplified Channel Block Diagram for SERDES and PCS



PCS

As shown in Figure 2-40, the PCS receives the parallel digital data from the deserializer receivers and adjusts the polarity, detects, byte boundary, decodes (8b/10b) and provides Clock Tolerance Compensation (CTC) FIFO for changing the clock domain from receiver clock to the FPGA Clock.

For the transmit channel, the PCS block receives the parallel data from the FPGA core, encodes it with 8b/10b, adjusts the polarity and passes the 8/10 bit data to the transmit SERDES channel.

The PCS also provides bypass modes that allow a direct 8-bit or 10-bit interface from the SERDES to the FPGA logic. The PCS interface to FPGA can also be programmed to run at 1/2 speed for a 16-bit or 20-bit interface to the FPGA logic.

SCI (SERDES Client Interface) Bus

The SERDES Client Interface (SCI) is a soft IP interface that allow the SERDES/PCS Quad block to be controlled by registers as opposed to the configuration memory cells. It is a simple register configuration interface.

The Diamond design tools support all modes of the PCS. Most modes are dedicated to applications associated with a specific industry standard data protocol. Other more general purpose modes allow users to define their own operation. With Diamond, the user can define the mode for each quad in a design.

Popular standards such as 10Gb Ethernet and x4 PCI-Express and 4x Serial RapidIO can be implemented using IP (provided by Lattice), a single quad (Four SERDES channels and PCS) and some additional logic from the core.

For further information about SERDES, please see the list of additional technical documentation at the end of this data sheet.

IEEE 1149.1-Compliant Boundary Scan Testability

All LatticeECP2/M devices have boundary scan cells that are accessed through an IEEE 1149.1 compliant Test Access Port (TAP). This allows functional testing of the circuit board, on which the device is mounted, through a serial scan path that can access all critical logic nodes. Internal registers are linked internally, allowing test data to be shifted in and loaded directly onto test nodes, or test data to be captured and shifted out for verification. The test access port consists of dedicated I/Os: TDI, TDO, TCK and TMS. The test access port has its own supply voltage V_{CCJ} and can operate with LVCMOS3.3, 2.5, 1.8, 1.5 and 1.2 standards.

For more details on boundary scan test, please see information regarding additional technical documentation at the end of this data sheet.

Device Configuration

All LatticeECP2/M devices contain two ports that can be used for device configuration. The Test Access Port (TAP), which supports bit-wide configuration, and the sysCONFIG port, support both byte-wide and serial configuration, including the standard SPI Flash interface. The TAP supports both the IEEE Standard 1149.1 Boundary Scan specification and the IEEE Standard 1532 In- System Configuration specification. The sysCONFIG port is a 20-pin interface with six I/Os used as dedicated pins with the remainder used as dual-use pins. See TN1108, LatticeECP2/M sysCONFIG Usage Guide for more information about using the dual-use pins as general purpose I/Os.

On power-up, the FPGA SRAM is ready to be configured using the selected sysCONFIG port. Once a configuration port is selected, it will remain active throughout that configuration cycle. The IEEE 1149.1 port can be activated any time after power-up by sending the appropriate command through the TAP port.

Enhanced Configuration Option

LatticeECP2/M devices have enhanced configuration features such as: decryption support, TransFR™ I/O and dual boot image support.

1. **Decryption Support**
   LatticeECP2/M devices provide on-chip, One Time Programmable (OTP) non-volatile key storage to support decryption of a 128-bit AES encrypted bitstream, securing designs and deterring design piracy.

2. **TransFR (Transparent Field Reconfiguration)**
   TransFR I/O (TFR) is a unique Lattice technology that allows users to update their logic in the field without interrupting system operation using a single ispVM® command. TransFR I/O allows I/O states to be frozen during device configuration. This allows the device to be field updated with a minimum of system disruption and downtime. See TN1087, Minimizing System Interruption During Configuration Using TransFR Technology, for details.

3. **Dual Boot Image Support**
   Dual boot images are supported for applications requiring reliable remote updates of configuration data for the system FPGA. After the system is running with a basic configuration, a new boot image can be downloaded remotely and stored in a separate location in the configuration storage device. Any time after the update the LatticeECP2/M can be re-booted from this new configuration file. If there is a problem, such as corrupt data during download or incorrect version number with this new boot image, the LatticeECP2/M device can revert back to the original backup configuration and try again. This all can be done without power cycling the system.

For more information about device configuration, please see the list of additional technical documentation at the end of this data sheet.

Soft Error Detect (SED) Support

LatticeECP2/M devices have dedicated logic to perform CRC checks. During configuration, the configuration data bitstream can be checked with the CRC logic block. In addition, the LatticeECP2 device can also be programmed

for checking soft errors (SED) in SRAM. SED can be run on a programmed device when the user logic is not active. If a soft error occurs, during user mode (normal operation) the device can be programmed to either reload from a known good boot image or generate an error signal.

For further information about Soft Error Detect (SED) support, please see the list of additional technical documentation at the end of this data sheet.

External Resistor

LatticeECP2/M devices require a single external, 10K ohm ±1% value between the XRES pin and ground. Device configuration will not be completed if this resistor is missing. There is no boundary scan register on the external resistor pad.

On-Chip Oscillator

Every LatticeECP2/M device has an internal CMOS oscillator which is used to derive a Master Clock for configuration. The oscillator and the Master Clock run continuously and are available to user logic after configuration is completed. The software default value of the Master Clock is 2.5MHz. Table 2-16 lists all the available Master Configuration Clock frequencies for normal non-encrypted mode and encrypted mode. When a different Master Clock is selected during the design process, the following sequence takes place:

1. Device powers up with a Master Clock frequency of 3.1MHz.
2. During configuration, users select a different master clock frequency.
3. The Master Clock frequency changes to the selected frequency once the clock configuration bits are received.
4. If the user does not select a master clock frequency, then the configuration bitstream defaults to the Master Clock frequency of 2.5MHz.

This internal CMOS oscillator is available to the user by routing it as an input clock to the clock tree. For further information about the use of this oscillator for configuration or user mode, please see the list of additional technical documentation at the end of this data sheet.

Table 2-16. Selectable Master Clock (CCLK) Frequencies During Configuration

Non-Encrypted Mode CCLK (MHz)			Encrypted Mode CCLK (MHz)
2.5[1]	13.0	45.0	2.5[1]
4.3	15.0	55.0	5.4
5.4	20.0	60.0	10.0
6.9	26.0	—	34.0
8.1	30.0	—	41.0
9.2	34.0	—	45.0
10.0	41.0	130.0	—

1. Software default frequency.

Density Shifting

The LatticeECP2/M family is designed to ensure that different density devices in the same family and in the same package have the same pinout. Furthermore, the architecture ensures a high success rate when performing design migration from lower density devices to higher density devices. In many cases, it is also possible to shift a lower utilization design targeted for a high-density device to a lower density device. However, the exact details of the final resource utilization will impact the likelihood of success in each case. Design migration between LatticeECP2 and LatticeECP2M families is not possible. For specific requirements relating to sysCONFIG pins of the ECP2M50, M70 and M100, see the Logic Signal Connections tables.

LatticeECP2/M Family Data Sheet DC and Switching Characteristics

January 2012 Data Sheet DS1006

Absolute Maximum Ratings[1, 2, 3]

Supply Voltage V_{CC} . . . . . -0.5 to 1.32V

Supply Voltage V_{CCAUX} . . . . . -0.5 to 3.75V

Supply Voltage V_{CCJ} . . . . . -0.5 to 3.75V

Output Supply Voltage V_{CCIO} . . . . . -0.5 to 3.75V

Input or I/O Tristate Voltage Applied[4]. . . . . -0.5 to 3.75V

Storage Temperature (Ambient) . . . . . -65 to 150°C

Junction Temperature (Tj) . . . . . +125°C

1. Stress above those listed under the "Absolute Maximum Ratings" may cause permanent damage to the device. Functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.
2. Compliance with the Lattice Thermal Management document is required.
3. All voltages referenced to GND.
4. Overshoot and undershoot of -2V to (V_{IHMAX} + 2) volts is permitted for a duration of <20ns.

Recommended Operating Conditions[7]

Symbol	Parameter	Min.	Max.	Units
V_{CC}[1, 4, 5]	Core Supply Voltage	1.14	1.26	V
V_{CCAUX}[1, 3, 4, 5]	Auxiliary Supply Voltage	3.135	3.465	V
V_{CCPLL}	PLL Supply Voltage	1.14	1.26	V
V_{CCIO}[1, 2, 4]	I/O Driver Supply Voltage	1.14	3.465	V
V_{CCJ}[1]	Supply Voltage for IEEE 1149.1 Test Access Port	1.14	3.465	V
t_{JCOM}	Junction Temperature, Commercial Operation	0	85	°C
t_{JIND}	Junction Temperature, Industrial Operation	-40	100	°C
SERDES External Power Supply (For LatticeECP2M Family Only)				
V_{CCIB}	Input Buffer Power Supply (1.2V)	1.14	1.26	V
	Input Buffer Power Supply (1.5V)	1.425	1.575	V
V_{CCOB}	Output Buffer Power Supply (1.2V)	1.14	1.26	V
	Output Buffer Power Supply (1.5V)	1.425	1.575	V
$V_{CCAUX33}$	Termination Resistor Switching Power Supply	3.135	3.465	V
V_{CCRX}[6]	Receive Power Supply	1.14	1.26	V
V_{CCTX}[6]	Transmit Power Supply	1.14	1.26	V

Symbol	Parameter	Min.	Max.	Units
V_{CCP}[6]	PLL and Reference Clock Buffer Power	1.14	1.26	V

1. If V_{CCIO} or V_{CCJ} is set to 1.2V, they must be connected to the same power supply as V_{CC}. If V_{CCIO} or V_{CCJ} is set to 3.3V, they must be connected to the same power supply as V_{CCAUX}. V_{CCPLL} must be connected to the same power supply as V_{CC} through careful filtering and decoupling.
2. See recommended voltages by I/O standard in subsequent table.
3. V_{CCAUX} ramp rate must not exceed 30mV/µs during power-up when transitioning between 0V and 3.3V.
4. For proper power-up configuration, users must ensure that the configuration control signals such as the CFGx, INITN, PROGRAMN and DONE pins are driven to the proper logic levels when the device powers up. The device power-up is triggered by the last of V_{CC}, V_{CCAUX} or V_{CCIO8} supplies that reaches its minimum valid levels. Alternatively, if the configuration control signals are pulled up by V_{CCIO8}, the V_{CCIO8} (configuration I/O bank) voltage must be powered up prior to or at the same time as the last of VCC or VCCAUX reaches its minimum levels.
5. For power-up, V_{CC} must reach its valid minimum value before powering up V_{CCAUX} (LatticeECP2/M “S” version devices only).
6. V_{CCRX},V_{CCTX} and V_{CCP} must be tied together in each quad and all quads need to be powered up.
7. For more power supply design recommendations, refer to TN1114 Electrical Recommendations for Lattice SERDES.

Hot Socketing Specifications[1, 2, 3, 4]

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
I_{DK}	Input or I/O leakage current	$0 \le V_{IN} \le V_{IH}$ (MAX.)	—	—	+/-1000	µA
I_{HDIN}[5]	SERDES average input current when device is powered down and inputs are driven		—	—	4	mA

1. V_{CC}, V_{CCAUX} and V_{CCIO} should rise/fall monotonically. V_{CC} and V_{CCPLL} must be connected to the same power supply (applies to ECP2-6, ECP2-12 and ECP2-20 only).
2. $0 \le V_{CC} \le V_{CC}$ (MAX), $0 \le V_{CCIO} \le V_{CCIO}$ (MAX) or $0 \le V_{CCAUX} \le V_{CCAUX}$ (MAX).
3. I_{DK} is additive to I_{PU}, I_{PW} or I_{BH}.
4. LVCMOS and LVTTL only.
5. Assumes that the device is powered down with all supplies grounded, both P and N inputs driven by a CML driver with maximum allowed V_{CCIB} of 1.575V, 8b10b data and internal AC coupling.

ESD Performance

Please refer to LatticeECP2/M Product Family Qualification Summary for complete qualification data, including ESD performance.

DC Electrical Characteristics

Over Recommended Operating Conditions

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
I_{IL}, I_{IH}[1, 2]	Input or I/O Low Leakage	$0 \le V_{IN} \le (V_{CCIO} - 0.2V)$	—	—	10	µA
I_{IH}[1, 3]	Input or I/O High Leakage	$(V_{CCIO} - 0.2V) < V_{IN} \le 3.6V$	—	—	150	µA
I_{PU}	I/O Active Pull-up Current	$0 \le V_{IN} \le 0.7\ V_{CCIO}$	-30	—	-210	µA
I_{PD}	I/O Active Pull-down Current	V_{IL} (MAX) $\le V_{IN} \le V_{IH}$ (MAX)	30	—	210	µA
I_{BHLS}	Bus Hold Low Sustaining Current	$V_{IN} = V_{IL}$ (MAX)	30	—	—	µA
I_{BHHS}	Bus Hold High Sustaining Current	$V_{IN} = 0.7\ V_{CCIO}$	-30	—	—	µA
I_{BHLO}	Bus Hold Low Overdrive Current	$0 \le V_{IN} \le V_{CCIO}$	—	—	210	µA
I_{BHHO}	Bus Hold High Overdrive Current	$0 \le V_{IN} \le V_{CCIO}$	—	—	-210	µA
V_{BHT}	Bus Hold Trip Points	$0 \le V_{IN} \le V_{IH}$ (MAX)	V_{IL} (MAX)	—	V_{IH} (MIN)	V
C1[4]	I/O Capacitance	V_{CCIO} = 3.3V, 2.5V, 1.8V, 1.5V, 1.2V, V_{CC} = 1.2V, V_{IO} = 0 to V_{IH} (MAX)	—	5	8	pf
C2[4]	Dedicated Input Capacitance	V_{CCIO} = 3.3V, 2.5V, 1.8V, 1.5V, 1.2V, V_{CC} = 1.2V, V_{IO} = 0 to V_{IH} (MAX)	—	5	6	pf

1. Input or I/O leakage current is measured with the pin configured as an input or as an I/O with the output driver tri-stated. It is not measured with the output driver active. Bus maintenance circuits are disabled.
2. When used as V_{REF}, maximum leakage = 25uA
3. Applicable to general purpose I/Os in top and bottom banks.
4. T_A 25°C, f = 1.0MHz.

LatticeECP2 Supply Current (Standby)[1, 2, 3, 4]

Over Recommended Operating Conditions

Symbol	Parameter	Device	Typ.[5]	Units
I_{CC}	Core Power Supply Current	ECP2-6	10	mA
		ECP2-12	20	mA
		ECP2-20	30	mA
		ECP2-35	50	mA
		ECP2-50	70	mA
		ECP2-70	100	mA
I_{CCAUX}	Auxiliary Power Supply Current	ECP2-6	24	mA
		ECP2-12	24	mA
		ECP2-20	24	mA
		ECP2-35	24	mA
		ECP2-50	24	mA
		ECP2-70	24	mA
I_{CCGPLL}	GPLL Power Supply Current (per GPLL)	ECP2-35, -50, -70 Only	0.5	mA
I_{CCSPLL}	GPLL Power Supply Current (per SPLL)	ECP2-35, -50, -70 Only	0.5	mA
I_{CCIO}	Bank Power Supply Current (Per Bank)	ECP2-6	2	mA
		ECP2-12	2	mA
		ECP2-20	2	mA
		ECP2-35	2	mA
		ECP2-50	2	mA
		ECP2-70	2	mA
I_{CCJ}	VCCJ Power Supply Current	All Devices	3	mA

1. For further information about supply current, please see the list of additional technical documentation at the end of this data sheet.
2. Assumes all outputs are tristated, all inputs are configured as LVCMOS and held at the V_{CCIO} or GND.
3. Frequency 0MHz.
4. Pattern represents a “blank” configuration data file.
5. T_J = 25°C, power supplies at normal voltage.

LatticeECP2M Supply Current (Standby)[1, 2, 3, 4]

Over Recommended Operating Conditions

Symbol	Parameter	Device	Typ.[5]	Units
I_{CC}	Core Power Supply Current	ECP2M20	25	mA
		ECP2M35	50	mA
		ECP2M50	85	mA
		ECP2M70	100	mA
		ECP2M100	100	mA
I_{CCAUX}	Auxiliary Power Supply Current	ECP2M20	24	mA
		ECP2M35	24	mA
		ECP2M50	24	mA
		ECP2M70	24	mA
		ECP2M100	24	mA
I_{CCGPLL}	GPLL Power Supply Current (per GPLL)	All Devices	0.5	mA
I_{CCSPLL}	GPLL Power Supply Current (per SPLL)	All Devices	0.5	mA
I_{CCIO}	Bank Power Supply Current (Per Bank)	ECP2M20	2	mA
		ECP2M35	2	mA
		ECP2M50	2	mA
		ECP2M70	2	mA
		ECP2M100	2	mA
I_{CCJ}	V_{CCJ} Power Supply Current	All Devices	3	mA

1. For further information about supply current, please see the list of additional technical documentation at the end of this data sheet.
2. Assumes all outputs are tristated, all inputs are configured as LVCMOS and held at the V_{CCIO} or GND.
3. Frequency 0MHz.
4. Pattern represents a “blank” configuration data file.
5. T_J = 25°C, power supplies at normal voltage.

LatticeECP2 Initialization Supply Current[1, 2, 3, 4]

Over Recommended Operating Conditions

Symbol	Parameter	Device	Typ.[5, 6, 7]	Units
I_{CC}	Core Power Supply Current	ECP2-6	34	mA
		ECP2-12	54	mA
		ECP2-20	82	mA
		ECP2-35	135	mA
		ECP2-50	187	mA
		ECP2-70	267	mA
I_{CCAUX}	Auxiliary Power Supply Current	ECP2-6	30	mA
		ECP2-12	30	mA
		ECP2-20	30	mA
		ECP2-35	30	mA
		ECP2-50	30	mA
		ECP2-70	30	mA
I_{CCGPLL}	GPLL Power Supply Current (per GPLL)	ECP2-35, -50, -70 Only	0.5	mA
I_{CCSPLL}	SPLL Power Supply Current (per SPLL)	ECP2-35, -50, -70 Only	0.5	mA
I_{CCIO}	Bank Power Supply Current (per Bank)	All Devices	3	mA
I_{CCJ}	VCCJ Power Supply Current	All Devices	4	mA

1. Until DONE signal is active.
2. For further information about supply current, please see the list of additional technical documentation at the end of this data sheet.
3. Assumes all outputs are tristated, all inputs are configured as LVCMOS and held at the V_{CCIO} or GND.
4. Frequency 0MHz.
5. T_J = 25°C, power supplies at nominal voltage.
6. A specific configuration pattern is used that scales with the size of the device; consists of 75% PFU utilization, 50% EBR, and 25% I/O configuration.
7. Values shown in this column are the typical average DC current during configuration. Use the Power Calculator tool to find the peak startup current.

LatticeECP2M Initialization Supply Current[1, 2, 3, 4]

Over Recommended Operating Conditions

Symbol	Parameter	Device	Typ.[5, 6, 7]	Units
I_{CC}	Core Power Supply Current	ECP2M20	41	mA
		ECP2M35	107	mA
		ECP2M50	169	mA
		ECP2M70	254	mA
		ECP2M100	378	mA
I_{CCAUX}	Auxiliary Power Supply Current	ECP2M20	30	mA
		ECP2M35	30	mA
		ECP2M50	30	mA
		ECP2M70	30	mA
		ECP2M100	30	mA
I_{CCGPLL}	GPLL Power Supply Current (per GPLL)	All Devices	0.5	mA
I_{CCSPLL}	SPLL Power Supply Current (per SPLL)	All Devices	0.5	mA
I_{CCIO}	Bank Power Supply Current (per Bank)	All Devices	3	mA
I_{CCJ}	VCCJ Power Supply Current	All Devices	4	mA

1. Until DONE signal is active.
2. For further information about supply current, please see the list of additional technical documentation at the end of this data sheet.
3. Assumes all outputs are tristated, all inputs are configured as LVCMOS and held at the V_{CCIO} or GND.
4. Frequency 0MHz.
5. T_J = 25ºC, power supplies at nominal voltage.
6. A specific configuration pattern is used that scales with the size of the device; consists of 75% PFU utilization, 50% EBR, and 25% I/O configuration.
7. Values shown in this column are the typical average DC current during configuration. Use the Power Calculator tool to find the peak startup current.

SERDES Power Supply Requirements (LatticeECP2M Family Only)[1]

Over Recommended Operating Conditions

Symbol	Description	Typ.[2]	Units
Standby (Power Down)			
$I_{CCTX-SB}$	V_{CCTX} current (per channel)	10	µA
$I_{CCRX-SB}$	V_{CCRX} current (per channel)	75	µA
$I_{CCIB-SB}$	Input buffer current (per channel)	0	µA
$I_{CCOB-SB}$	Output buffer current (per channel)	0	µA
I_{CCP-SB}	SERDES PLL current (per quad)	30	µA
$I_{CCAX33-SB}$	SERDES termination current (per quad)	10	µA
Operating (Data Rate = 3.125 Gbps)			
$I_{CCTX-OP}$	V_{CCTX} current (per channel)	19	mA
$I_{CCRX-OP}$	V_{CCRX} current (per channel)	34	mA
$I_{CCIB-OP}$	Input buffer current (per channel)	4	mA
$I_{CCOB-OP}$	Output buffer current (per channel)	13	mA
I_{CCP-OP}	SERDES PLL current (per quad)	26	mA
$I_{CCAX33-OP}$	SERDES termination current (per quad)	0.01	mA

1. Equalization enabled, pre-emphasis disabled.
2. T_J = 25°C, power supplies at nominal voltage.

SERDES Power (LatticeECP2M Family Only)

Table 3-1 presents the SERDES power for one channel.

Table 3-1. SERDES Power[1]

Symbol	Description	Typ.[2]	Units
$P_{S-1CH-31}$	SERDES power (one channel @ 3.125 Gbps)	90	mW
$P_{S-1CH-25}$	SERDES power (one channel @ 2.5 Gbps)	87	mW
$P_{S-1CH-12}$	SERDES power (one channel @ 1.25 Gbps)	86	mW
$P_{S-1CH-02}$	SERDES power (one channel @ 250 Mbps)	76	mW

1. One quarter of the total quad power (includes contribution from common circuits, all channels in the quad operating, pre-emphasis disabled, equalization enabled).
2. Typical values measured at 25ºC and 1.2V.

sysI/O Recommended Operating Conditions

Standard	V_{CCIO}			V_{REF} (V)		
	Min.	Typ.	Max.	Min.	Typ.	Max.
LVCMOS 3.3[2]	3.135	3.3	3.465	—	—	—
LVCMOS 2.5[2]	2.375	2.5	2.625	—	—	—
LVCMOS 1.8	1.71	1.8	1.89	—	—	—
LVCMOS 1.5	1.425	1.5	1.575	—	—	—
LVCMOS 1.2[2]	1.14	1.2	1.26	—	—	—
LVTTL[2]	3.135	3.3	3.465	—	—	—
PCI	3.135	3.3	3.465	—	—	—
SSTL18[2] Class I, II	1.71	1.8	1.89	0.833	0.9	0.969
SSTL2[2] Class I, II	2.375	2.5	2.625	1.15	1.25	1.35
SSTL3[2] Class I, II	3.135	3.3	3.465	1.3	1.5	1.7
HSTL[2] 15 Class I	1.425	1.5	1.575	0.68	0.75	0.9
HSTL[2] 18 Class I, II	1.71	1.8	1.89	0.816	0.9	1.08
LVDS[2]	2.375	2.5	2.625	—	—	—
MLVDS25[1]	2.375	2.5	2.625	—	—	—
LVPECL33[1, 2]	3.135	3.3	3.465	—	—	—
BLVDS25[1, 2]	2.375	2.5	2.625	—	—	—
RSDS[1, 2]	2.375	2.5	2.625	—	—	—
SSTL18D_I[2], II[2]	1.71	1.8	1.89	—	—	—
SSTL25D_ I[2], II[2]	2.375	2.5	2.625	—	—	—
SSTL33D_ I[2], II[2]	3.135	3.3	3.465	—	—	—
HSTL15D_ I[2]	1.425	1.5	1.575	—	—	—
HSTL18D_ I[2], II[2]	1.71	1.8	1.89	—	—	—

1. Inputs on chip. Outputs are implemented with the addition of external resistors.
2. Input on this standard does not depend on the value of V_{CCIO}.

sysI/O Single-Ended DC Electrical Characteristics

Input/Output Standard	V_{IL}		V_{IH}		V_{OL} Max. (V)	V_{OH} Min. (V)	I_{OL}[1] (mA)	I_{OH}[1] (mA)
	Min. (V)	Max. (V)	Min. (V)	Max. (V)				
LVCMOS 3.3	-0.3	0.8	2.0	3.6	0.4	V_{CCIO} - 0.4	20, 16, 12, 8, 4	-20, -16, -12, -8, -4
					0.2	V_{CCIO} - 0.2	0.1	-0.1
LVTTL	-0.3	0.8	2.0	3.6	0.4	V_{CCIO} - 0.4	20, 16, 12, 8, 4	-20, -16, -12, -8, -4
					0.2	V_{CCIO} - 0.2	0.1	-0.1
LVCMOS 2.5	-0.3	0.7	1.7	3.6	0.4	V_{CCIO} - 0.4	20, 16, 12, 8, 4	-20, -16, -12, -8, -4
					0.2	V_{CCIO} - 0.2	0.1	-0.1
LVCMOS 1.8	-0.3	0.35 V_{CCIO}	0.65 V_{CCIO}	3.6	0.4	V_{CCIO} - 0.4	16, 12, 8, 4	-16, -12, -8, -4
					0.2	V_{CCIO} - 0.2	0.1	-0.1
LVCMOS 1.5	-0.3	0.35 V_{CCIO}	0.65 V_{CCIO}	3.6	0.4	V_{CCIO} - 0.4	8, 4	-8, -4
					0.2	V_{CCIO} - 0.2	0.1	-0.1
LVCMOS 1.2	-0.3	0.35 V_{CC}	0.65 V_{CC}	3.6	0.4	V_{CCIO} - 0.4	6, 2	-6, -2
					0.2	V_{CCIO} - 0.2	0.1	-0.1
PCI	-0.3	0.3 V_{CCIO}	0.5 V_{CCIO}	3.6	0.1 V_{CCIO}	0.9 V_{CCIO}	1.5	-0.5
SSTL3 Class I	-0.3	V_{REF} - 0.2	V_{REF} + 0.2	3.6	0.7	V_{CCIO} - 1.1	8	-8
SSTL3 Class II	-0.3	V_{REF} - 0.2	V_{REF} + 0.2	3.6	0.5	V_{CCIO} - 0.9	16	-16
SSTL2 Class I	-0.3	V_{REF} - 0.18	V_{REF} + 0.18	3.6	0.54	V_{CCIO} - 0.62	7.6	-7.6
							12	-12
SSTL2 Class II	-0.3	V_{REF} - 0.18	V_{REF} + 0.18	3.6	0.35	V_{CCIO} - 0.43	15.2	-15.2
							20	-20
SSTL18 Class I	-0.3	V_{REF} - 0.125	V_{REF} + 0.125	3.6	0.4	V_{CCIO} - 0.4	6.7	-6.7
SSTL18 Class II	-0.3	V_{REF} - 0.125	V_{REF} + 0.125	3.6	0.28	V_{CCIO} - 0.28	8	-8
							11	-11
HSTL Class I	-0.3	V_{REF} - 0.1	V_{REF} + 0.1	3.6	0.4	V_{CCIO} - 0.4	4	-4
							8	-8
HSTL18 Class I	-0.3	V_{REF} - 0.1	V_{REF} + 0.1	3.6	0.4	V_{CCIO} - 0.4	8	-8
							12	-12
HSTL18 Class II	-0.3	V_{REF} - 0.1	V_{REF} + 0.1	3.6	0.4	V_{CCIO} - 0.4	16	-16

1. The average DC current drawn by I/Os between GND connections, or between the last GND in an I/O bank and the end of an I/O bank, as shown in the logic signal connections table shall not exceed n * 8mA, where n is the number of I/Os between bank GND connections or between the last GND in a bank and the end of a bank.

sysI/O Differential Electrical Characteristics

LVDS

Over Recommended Operating Conditions

Parameter	Description	Test Conditions	Min.	Typ.	Max.	Units
V_{INP}, V_{INM}	Input Voltage		0	—	2.4	V
V_{CM}	Input Common Mode Voltage	Half the Sum of the Two Inputs	0.05	—	2.35	V
V_{THD}	Differential Input Threshold	Difference Between the Two Inputs	+/-100	—	—	mV
I_{IN}	Input Current	Power On or Power Off	—	—	+/-10	µA
V_{OH}	Output High Voltage for V_{OP} or V_{OM}	R_T = 100 Ohm	—	1.38	1.60	V
V_{OL}	Output Low Voltage for V_{OP} or V_{OM}	R_T = 100 Ohm	0.9V	1.03	—	V
V_{OD}	Output Voltage Differential	(V_{OP} - V_{OM}), R_T = 100 Ohm	250	350	450	mV
ýV$_{OD}$	Change in V_{OD} Between High and Low		—	—	50	mV
V_{OS}	Output Voltage Offset	(V_{OP} + V_{OM})/2, R_T = 100 Ohm	1.125	1.20	1.375	V
ýV$_{OS}$	Change in V_{OS} Between H and L		—	—	50	mV
I_{SA}	Output Short Circuit Current	V_{OD} = 0V Driver Outputs Shorted to Ground	—	—	24	mA
I_{SAB}	Output Short Circuit Current	V_{OD} = 0V Driver Outputs Shorted to Each Other	—	—	12	mA

Differential HSTL and SSTL

Differential HSTL and SSTL outputs are implemented as a pair of complementary single-ended outputs. All allowable single-ended output classes (class I and class II) are supported in this mode.

For further information about LVPECL, RSDS, MLVDS, BLVDS and other differential interfaces please see the list of additional technical information at the end of this data sheet.

LVDS25E

The top and bottom sides of LatticeECP2/M devices support LVDS outputs via emulated complementary LVCMOS outputs in conjunction with a parallel resistor across the driver outputs. The scheme shown in Figure 3-1 is one possible solution for point-to-point signals.

Figure 3-1. LVDS25E Output Termination Example

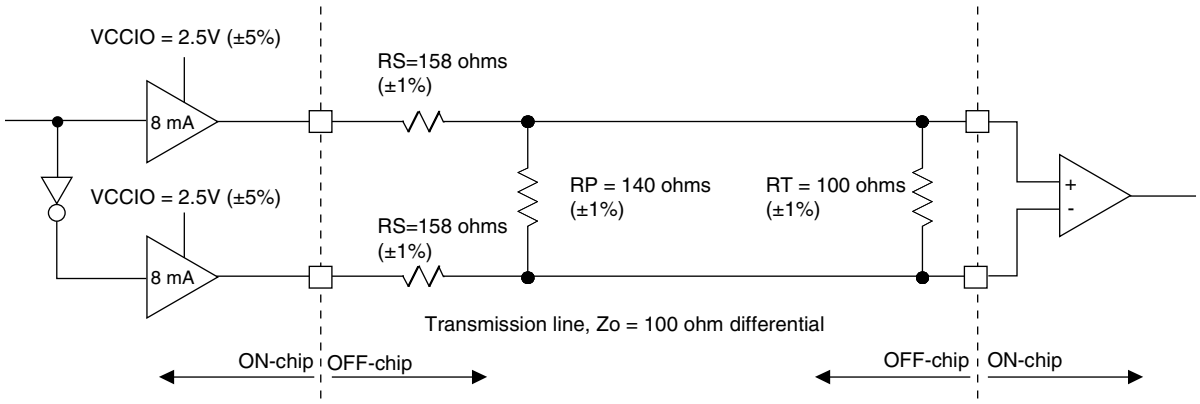


Table 3-2. LVDS25E DC Conditions

Parameter	Description	Typical	Units
V_{CCIO}	Output Driver Supply (+/-5%)	2.50	V
Z_{OUT}	Driver Impedance	20	Ω
R_S	Driver Series Resistor (+/-1%)	158	Ω
R_P	Driver Parallel Resistor (+/-1%)	140	Ω
R_T	Receiver Termination (+/-1%)	100	Ω
V_{OH}	Output High Voltage	1.43	V
V_{OL}	Output Low Voltage	1.07	V
V_{OD}	Output Differential Voltage	0.35	V
V_{CM}	Output Common Mode Voltage	1.25	V
Z_{BACK}	Back Impedance	100.5	Ω
I_{DC}	DC Output Current	6.03	mA

LVCMOS33D

All I/O banks support emulated differential I/O using the LVCMOS33D I/O type. This option, along with the external resistor network, provides the system designer the flexibility to place differential outputs on an I/O bank with 3.3V VCCIO. The default drive current for LVCMOS33D output is 12mA with the option to change the device strength to 4mA, 8mA, 16mA or 20mA. Follow the LVCMOS33 specifications for the DC characteristics of the LVCMOS33D.

BLVDS

The LatticeECP2/M devices support the BLVDS standard. This standard is emulated using complementary LVCMOS outputs in conjunction with a parallel external resistor across the driver outputs. BLVDS is intended for use when multi-drop and bi-directional multi-point differential signaling is required. The scheme shown in Figure 3-2 is one possible solution for bi-directional multi-point differential signals.

Figure 3-2. BLVDS Multi-point Output Example

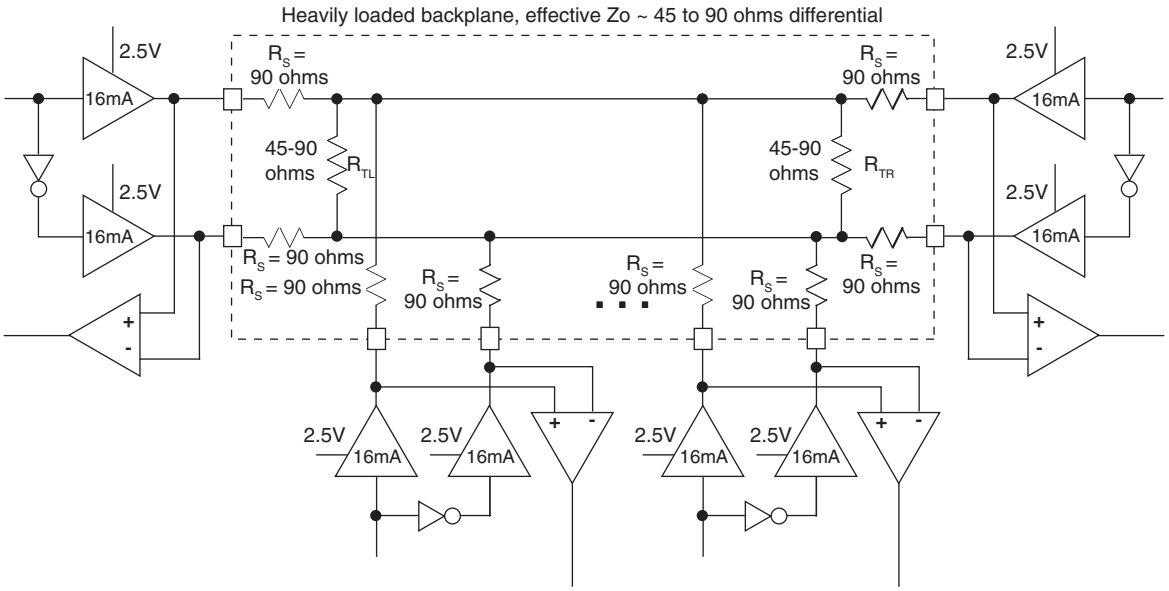


Table 3-3. BLVDS DC Conditions[1]

Over Recommended Operating Conditions

Parameter	Description	Typical		Units
		Zo = 45Ω	Zo = 90Ω	
V_{CCIO}	Output Driver Supply (+/- 5%)	2.50	2.50	V
Z_{OUT}	Driver Impedance	10.00	10.00	Ω
R_S	Driver Series Resistor (+/- 1%)	90.00	90.00	Ω
R_{TL}	Driver Parallel Resistor (+/- 1%)	45.00	90.00	Ω
R_{TR}	Receiver Termination (+/- 1%)	45.00	90.00	Ω
V_{OH}	Output High Voltage	1.38	1.48	V
V_{OL}	Output Low Voltage	1.12	1.02	V
V_{OD}	Output Differential Voltage	0.25	0.46	V
V_{CM}	Output Common Mode Voltage	1.25	1.25	V
I_{DC}	DC Output Current	11.24	10.20	mA

1. For input buffer, see LVDS table.

LVPECL

The LatticeECP2/M devices support the differential LVPECL standard. This standard is emulated using complementary LVCMOS outputs in conjunction with a parallel resistor across the driver outputs. The LVPECL input standard is supported by the LVDS differential input buffer. The scheme shown in Figure 3-3 is one possible solution for point-to-point signals.

Figure 3-3. Differential LVPECL

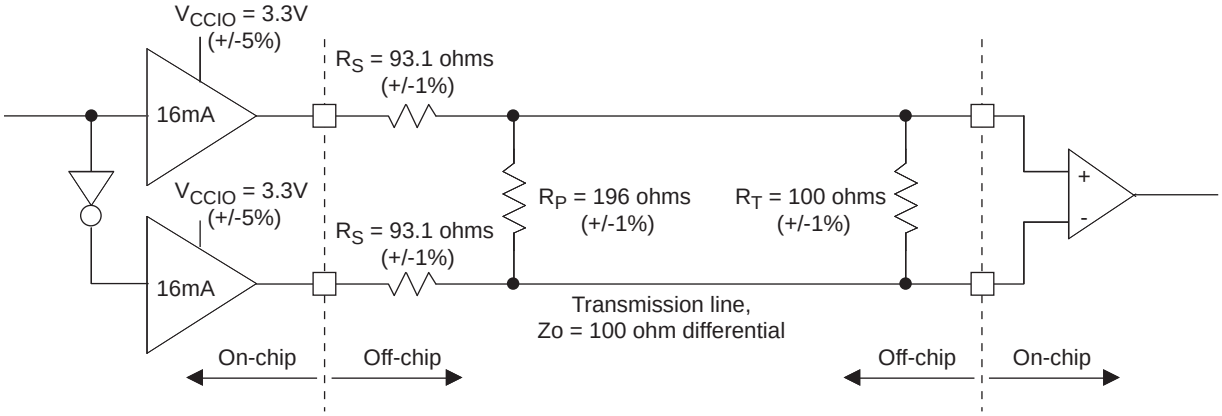


Table 3-4. LVPECL DC Conditions[1]

Over Recommended Operating Conditions

Parameter	Description	Typical	Units
V_{CCIO}	Output Driver Supply (+/-5%)	3.30	V
Z_{OUT}	Driver Impedance	10	Ω
R_S	Driver Series Resistor (+/-1%)	93	Ω
R_P	Driver Parallel Resistor (+/-1%)	196	Ω
R_T	Receiver Termination (+/-1%)	100	Ω
V_{OH}	Output High Voltage	2.05	V
V_{OL}	Output Low Voltage	1.25	V
V_{OD}	Output Differential Voltage	0.80	V
V_{CM}	Output Common Mode Voltage	1.65	V
Z_{BACK}	Back Impedance	100.5	Ω
I_{DC}	DC Output Current	12.11	mA

1. For input buffer, see LVDS table.

RSDS

The LatticeECP2/M devices support differential RSDS standard. This standard is emulated using complementary LVCMOS outputs in conjunction with a parallel resistor across the driver outputs. The RSDS input standard is supported by the LVDS differential input buffer. The scheme shown in Figure 3-4 is one possible solution for RSDS standard implementation. Resistor values in Figure 3-4 are industry standard values for 1% resistors.

Figure 3-4. RSDS (Reduced Swing Differential Signaling)

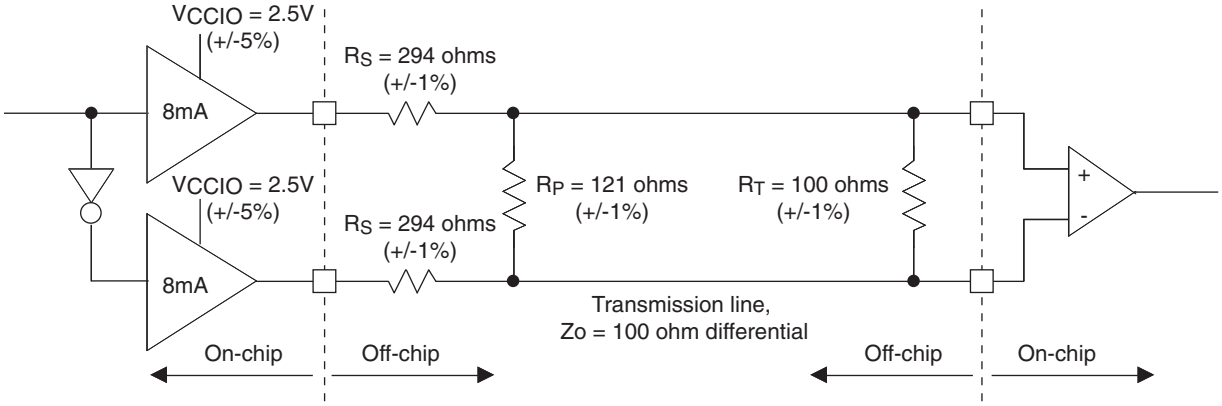


Table 3-5. RSDS DC Conditions[1]

Over Recommended Operating Conditions

Parameter	Description	Typical	Units
V_{CCIO}	Output Driver Supply (+/-5%)	2.50	V
Z_{OUT}	Driver Impedance	20	Ω
R_S	Driver Series Resistor (+/-1%)	294	Ω
R_P	Driver Parallel Resistor (+/-1%)	121	Ω
R_T	Receiver Termination (+/-1%)	100	Ω
V_{OH}	Output High Voltage	1.35	V
V_{OL}	Output Low Voltage	1.15	V
V_{OD}	Output Differential Voltage	0.20	V
V_{CM}	Output Common Mode Voltage	1.25	V
Z_{BACK}	Back Impedance	101.5	Ω
I_{DC}	DC Output Current	3.66	mA

1. For input buffer, see LVDS table.

MLVDS

The LatticeECP2/M devices support the differential MLVDS standard. This standard is emulated using complementary LVCMOS outputs in conjunction with a parallel resistor across the driver outputs. The MLVDS input standard is supported by the LVDS differential input buffer. The scheme shown in Figure 3-5 is one possible solution for MLVDS standard implementation. Resistor values in Figure 3-5 are industry standard values for 1% resistors.

Figure 3-5. MLVDS (Multipoint Low Voltage Differential Signaling)

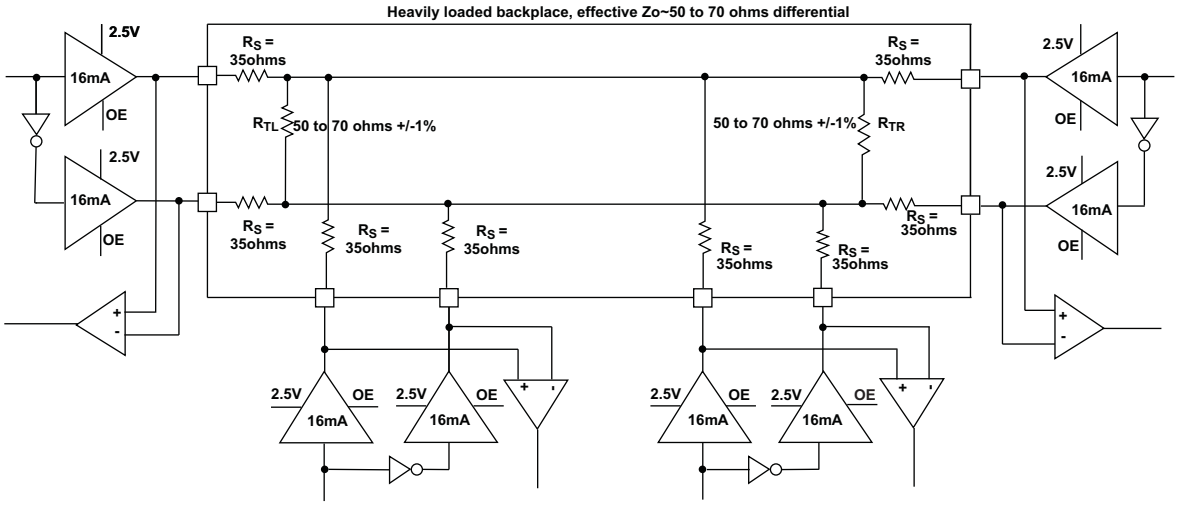


Table 3-6. MLVDS DC Conditions[1]

Parameter	Description	Typical		Units
		Zo=50Ω	Zo=70Ω	
V_{CCIO}	Output Driver Supply (+/-5%)	2.50	2.50	V
Z_{OUT}	Driver Impedance	10.00	10.00	Ω
R_S	Driver Series Resistor (+/-1%)	35.00	35.00	Ω
R_{TL}	Driver Parallel Resistor (+/-1%)	50.00	70.00	Ω
R_{TR}	Receiver Termination (+/-1%)	50.00	70.00	Ω
V_{OH}	Output High Voltage	1.52	1.60	V
V_{OL}	Output Low Voltage	0.98	0.90	V
V_{OD}	Output Differential Voltage	0.54	0.70	V
V_{CM}	Output Common Mode Voltage	1.25	1.25	V
I_{DC}	DC Output Current	21.74	20.00	mA

1. For input buffer, see LVDS table.

For further information about LVPECL, RSDS, MLVDS, BLVDS and other differential interfaces please see the list of additional technical information at the end of this data sheet.

Typical Building Block Function Performance[1]

Pin-to-Pin Performance (LVCMOS25 12mA Drive)

Function	-7 Timing	Units
Basic Functions		
16-bit Decoder	3.8	ns
32-bit Decoder	4.5	ns
64-bit Decoder	5.0	ns
4:1 MUX	3.2	ns
8:1 MUX	3.4	ns
16:1 MUX	3.5	ns
32:1 MUX	4.0	ns

1. These timing numbers were generated using the ispLEVER 8.0 design tool. Exact performance may vary with device and tool version. The tool uses internal parameters that have been characterized but are not tested on every device.

Register-to-Register Performance

Function	-7 Timing	Units
Basic Functions		
16-bit Decoder	599	MHz
32-bit Decoder	542	MHz
64-bit Decoder	417	MHz
4:1 MUX	847	MHz
8:1 MUX	803	MHz
16:1 MUX	660	MHz
32:1 MUX	577	MHz
8-bit Adder	591	MHz
16-bit Adder	500	MHz
64-bit Adder	306	MHz
16-bit Counter	488	MHz
32-bit Counter	378	MHz
64-bit Counter	260	MHz
64-bit Accumulator	253	MHz
Embedded Memory Functions		
512x36 Single Port RAM, EBR Output Registers	370	MHz
1024x18 True-Dual Port RAM (Write Through or Normal, EBR Output Registers)	370	MHz
1024x18 True-Dual Port RAM (Write Through or Normal, PLC Output Registers)	280	MHz
Distributed Memory Functions		
16x4 Pseudo-Dual Port RAM (One PFU)	819	MHz
32x4 Pseudo-Dual Port RAM	521	MHz
64x8 Pseudo-Dual Port RAM	435	MHz
DSP Functions		
18x18 Multiplier (All Registers)	420	MHz
9x9 Multiplier (All Registers)	420	MHz

Register-to-Register Performance (Continued)

Function	-7 Timing	Units
36x36 Multiplier (All Registers)	372	MHz
18x18 Multiplier/Accumulate (Input and Output Registers)	295	MHz
18x18 Multiplier-Add/Sub-Sum (All Registers)	420	MHz
DSP IP Functions		
16-Tap Fully-Parallel FIR Filter	304	MHz
1024-pt, Radix 4, Decimation in Frequency FFT	227	MHz
8x8 Matrix Multiplier	223	MHz

Derating Timing Tables

Logic timing provided in the following sections of this data sheet and the Diamond design tool are worst case numbers in the operating range. Actual delays at nominal temperature and voltage for best case process, can be much better than the values given in the tables. The Diamond design tool can provide logic timing numbers at a particular temperature and voltage.

LatticeECP2/M External Switching Characteristics[9]

Over Recommended Operating Conditions

Parameter	Description	Device	-7		-6		-5		Units
			Min.	Max.	Min.	Max.	Min.	Max.	
General I/O Pin Parameters (using Primary Clock without PLL)[1]									
t_{CO}	Clock to Output - PIO Output Register	LFE2-6	—	3.50	—	3.90	—	4.20	ns
		LFE2-12	—	3.50	—	3.90	—	4.20	ns
		LFE2-20	—	3.50	—	3.90	—	4.20	ns
		LFE2-35	—	3.50	—	3.90	—	4.20	ns
		LFE2-50	—	3.50	—	3.90	—	4.20	ns
		LFE2-70	—	3.70	—	4.10	—	4.40	ns
		LFE2M20	—	3.90	—	4.30	—	4.70	ns
		LFE2M35	—	3.90	—	4.30	—	4.70	ns
		LFE2M50	—	4.50	—	5.00	—	5.40	ns
		LFE2M70	—	4.50	—	5.00	—	5.40	ns
		LFE2M100	—	4.50	—	5.00	—	5.40	ns
t_{SU}	Clock to Data Setup - PIO Input Register	LFE2-6	0.00	—	0.00	—	0.00	—	ns
		LFE2-12	0.00	—	0.00	—	0.00	—	ns
		LFE2-20	0.00	—	0.00	—	0.00	—	ns
		LFE2-35	0.00	—	0.00	—	0.00	—	ns
		LFE2-50	0.00	—	0.00	—	0.00	—	ns
		LFE2-70	0.00	—	0.00	—	0.00	—	ns
		LFE2M20	0.00	—	0.00	—	0.00	—	ns
		LFE2M35	0.00	—	0.00	—	0.00	—	ns
		LFE2M50	0.00	—	0.00	—	0.00	—	ns
		LFE2M70	0.00	—	0.00	—	0.00	—	ns
		LFE2M100	0.00	—	0.00	—	0.00	—	ns
t_{H}	Clock to Data Hold - PIO Input Register	LFE2-6	1.40	—	1.70	—	1.90	—	ns
		LFE2-12	1.40	—	1.70	—	1.90	—	ns
		LFE2-20	1.40	—	1.70	—	1.90	—	ns
		LFE2-35	1.40	—	1.70	—	1.90	—	ns
		LFE2-50	1.40	—	1.70	—	1.90	—	ns
		LFE2-70	1.40	—	1.70	—	1.90	—	ns
		LFE2M20	1.40	—	1.70	—	1.90	—	ns
		LFE2M35	1.40	—	1.70	—	1.90	—	ns
		LFE2M50	1.80	—	2.10	—	2.30	—	ns
		LFE2M70	1.80	—	2.10	—	2.30	—	ns
		LFE2M100	1.80	—	2.10	—	2.30	—	ns

LatticeECP2/M External Switching Characteristics[9] (Continued)

Over Recommended Operating Conditions

Parameter	Description	Device	-7		-6		-5		Units
			Min.	Max.	Min.	Max.	Min.	Max.	
t_{SU_DEL}	Clock to Data Setup - PIO Input Register with Data Input Delay	LFE2-6	1.40	—	1.70	—	1.90	—	ns
		LFE2-12	1.40	—	1.70	—	1.90	—	ns
		LFE2-20	1.40	—	1.70	—	1.90	—	ns
		LFE2-35	1.40	—	1.70	—	1.90	—	ns
		LFE2-50	1.40	—	1.70	—	1.90	—	ns
		LFE2-70	1.40	—	1.70	—	1.90	—	ns
		LFE2M20	1.40	—	1.70	—	1.90	—	ns
		LFE2M35	1.40	—	1.70	—	1.90	—	ns
		LFE2M50	1.40	—	1.70	—	1.90	—	ns
		LFE2M70	1.40	—	1.70	—	1.90	—	ns
		LFE2M100	1.40	—	1.70	—	1.90	—	ns
t_{H_DEL}	Clock to Data Hold - PIO Input Register with Input Data Delay	LFE2-6	0.00	—	0.00	—	0.00	—	ns
		LFE2-12	0.00	—	0.00	—	0.00	—	ns
		LFE2-20	0.00	—	0.00	—	0.00	—	ns
		LFE2-35	0.00	—	0.00	—	0.00	—	ns
		LFE2-50	0.00	—	0.00	—	0.00	—	ns
		LFE2-70	0.00	—	0.00	—	0.00	—	ns
		LFE2M20	0.00	—	0.00	—	0.00	—	ns
		LFE2M35	0.00	—	0.00	—	0.00	—	ns
		LFE2M50	0.00	—	0.00	—	0.00	—	ns
		LFE2M70	0.00	—	0.00	—	0.00	—	ns
		LFE2M100	0.00	—	0.00	—	0.00	—	ns
f_{MAX_IO}	Clock Frequency of I/O Register and PFU Register	ECP2/M	—	420	—	357	—	311	MHz
General I/O Pin Parameters (using Edge Clock without PLL)[1]									
t_{COE}	Clock to Output - PIO Output Register	LFE2-6	—	2.60	—	2.90	—	3.20	ns
		LFE2-12	—	2.60	—	2.90	—	3.20	ns
		LFE2-20	—	2.60	—	2.90	—	3.20	ns
		LFE2-35	—	2.60	—	2.90	—	3.20	ns
		LFE2-50	—	2.60	—	2.90	—	3.20	ns
		LFE2-70	—	2.60	—	2.90	—	3.20	ns
		LFE2M20	—	2.60	—	2.90	—	3.20	ns
		LFE2M35	—	2.60	—	2.90	—	3.20	ns
		LFE2M50	—	3.10	—	3.40	—	3.70	ns
		LFE2M70	—	3.10	—	3.40	—	3.70	ns
		LFE2M100	—	3.10	—	3.40	—	3.70	ns

LatticeECP2/M External Switching Characteristics[9] (Continued)

Over Recommended Operating Conditions

Parameter	Description	Device	-7		-6		-5		Units
			Min.	Max.	Min.	Max.	Min.	Max.	
t_{SUE}	Clock to Data Setup - PIO Input Register	LFE2-6	0.00	—	0.00	—	0.00	—	ns
		LFE2-12	0.00	—	0.00	—	0.00	—	ns
		LFE2-20	0.00	—	0.00	—	0.00	—	ns
		LFE2-35	0.00	—	0.00	—	0.00	—	ns
		LFE2-50	0.00	—	0.00	—	0.00	—	ns
		LFE2-70	0.00	—	0.00	—	0.00	—	ns
		LFE2M20	0.00	—	0.00	—	0.00	—	ns
		LFE2M35	0.00	—	0.00	—	0.00	—	ns
		LFE2M50	0.00	—	0.00	—	0.00	—	ns
		LFE2M70	0.00	—	0.00	—	0.00	—	ns
		LFE2M100	0.00	—	0.00	—	0.00	—	ns
t_{HE}	Clock to Data Hold - PIO Input Register	LFE2-6	0.90	—	1.10	—	1.30	—	ns
		LFE2-12	0.90	—	1.10	—	1.30	—	ns
		LFE2-20	0.90	—	1.10	—	1.30	—	ns
		LFE2-35	0.90	—	1.10	—	1.30	—	ns
		LFE2-50	0.90	—	1.10	—	1.30	—	ns
		LFE2-70	0.90	—	1.10	—	1.30	—	ns
		LFE2M20	0.90	—	1.10	—	1.30	—	ns
		LFE2M35	0.90	—	1.10	—	1.30	—	ns
		LFE2M50	1.20	—	1.40	—	1.60	—	ns
		LFE2M70	1.20	—	1.40	—	1.60	—	ns
		LFE2M100	1.20	—	1.40	—	1.60	—	ns
t_{SU_DELE}	Clock to Data Setup - PIO Input Register with Data Input Delay	LFE2-6	1.00	—	1.30	—	1.60	—	ns
		LFE2-12	1.00	—	1.30	—	1.60	—	ns
		LFE2-20	1.00	—	1.30	—	1.60	—	ns
		LFE2-35	1.00	—	1.30	—	1.60	—	ns
		LFE2-50	1.00	—	1.30	—	1.60	—	ns
		LFE2-70	1.00	—	1.30	—	1.60	—	ns
		LFE2M20	1.20	—	1.60	—	1.90	—	ns
		LFE2M35	1.20	—	1.60	—	1.90	—	ns
		LFE2M50	1.20	—	1.60	—	1.90	—	ns
		LFE2M70	1.20	—	1.60	—	1.90	—	ns
		LFE2M100	1.20	—	1.60	—	1.90	—	ns

LatticeECP2/M External Switching Characteristics[9] (Continued)

Over Recommended Operating Conditions

Parameter	Description	Device	-7		-6		-5		Units
			Min.	Max.	Min.	Max.	Min.	Max.	
t_{H_DELE}	Clock to Data Hold - PIO Input Register with Input Data Delay	LFE2-6	0.00	—	0.00	—	0.00	—	ns
		LFE2-12	0.00	—	0.00	—	0.00	—	ns
		LFE2-20	0.00	—	0.00	—	0.00	—	ns
		LFE2-35	0.00	—	0.00	—	0.00	—	ns
		LFE2-50	0.00	—	0.00	—	0.00	—	ns
		LFE2-70	0.00	—	0.00	—	0.00	—	ns
		LFE2M20	0.00	—	0.00	—	0.00	—	ns
		LFE2M35	0.00	—	0.00	—	0.00	—	ns
		LFE2M50	0.00	—	0.00	—	0.00	—	ns
		LFE2M70	0.00	—	0.00	—	0.00	—	ns
		LFE2M100	0.00	—	0.00	—	0.00	—	ns
f_{MAX_IOE}	Clock Frequency of I/O and PFU Register	ECP2/M	—	420	—	357	—	311	MHz
General I/O Pin Parameters (using Primary Clock with PLL)[1]									
t_{COPLL}[10]	Clock to Output - PIO Output Register	LFE2-6	—	2.30	—	2.60	—	2.80	ns
		LFE2-12	—	2.30	—	2.60	—	2.80	ns
		LFE2-20	—	2.30	—	2.60	—	2.80	ns
		LFE2-35	—	2.30	—	2.60	—	2.80	ns
		LFE2-50	—	2.30	—	2.60	—	2.80	ns
		LFE2-70	—	2.30	—	2.60	—	2.80	ns
		LFE2M20	—	2.30	—	2.60	—	2.80	ns
		LFE2M35	—	2.30	—	2.60	—	2.80	ns
		LFE2M50	—	2.60	—	2.90	—	3.10	ns
		LFE2M70	—	2.60	—	2.90	—	3.10	ns
		LFE2M100	—	2.70	—	3.00	—	3.20	ns
t_{SUPLL}	Clock to Data Setup - PIO Input Register	LFE2-6	0.70	—	0.80	—	0.90	—	ns
		LFE2-12	0.70	—	0.80	—	0.90	—	ns
		LFE2-20	0.70	—	0.80	—	0.90	—	ns
		LFE2-35	0.70	—	0.80	—	0.90	—	ns
		LFE2-50	0.70	—	0.80	—	0.90	—	ns
		LFE2-70	0.70	—	0.80	—	0.90	—	ns
		LFE2M20	0.70	—	0.80	—	0.90	—	ns
		LFE2M35	0.70	—	0.80	—	0.90	—	ns
		LFE2M50	0.70	—	0.80	—	0.90	—	ns
		LFE2M70	0.70	—	0.80	—	0.90	—	ns
		LFE2M100	0.80	—	0.90	—	1.00	—	ns

LatticeECP2/M External Switching Characteristics[9] (Continued)

Over Recommended Operating Conditions

Parameter	Description	Device	-7		-6		-5		Units
			Min.	Max.	Min.	Max.	Min.	Max.	
t_{HPLL}	Clock to Data Hold - PIO Input Register	LFE2-6	1.00	—	1.20	—	1.40	—	ns
		LFE2-12	1.00	—	1.20	—	1.40	—	ns
		LFE2-20	1.00	—	1.20	—	1.40	—	ns
		LFE2-35	1.00	—	1.20	—	1.40	—	ns
		LFE2-50	1.00	—	1.20	—	1.40	—	ns
		LFE2-70	1.00	—	1.20	—	1.40	—	ns
		LFE2M20	1.00	—	1.20	—	1.40	—	ns
		LFE2M35	1.00	—	1.20	—	1.40	—	ns
		LFE2M50	1.00	—	1.20	—	1.40	—	ns
		LFE2M70	1.00	—	1.20	—	1.40	—	ns
		LFE2M100	1.00	—	1.20	—	1.40	—	ns
t_{SU_DELPLL}	Clock to Data Setup - PIO Input Register with Data Input Delay	LFE2-6	1.80	—	2.00	—	2.20	—	ns
		LFE2-12	1.80	—	2.00	—	2.20	—	ns
		LFE2-20	1.80	—	2.00	—	2.20	—	ns
		LFE2-35	1.80	—	2.00	—	2.20	—	ns
		LFE2-50	1.80	—	2.00	—	2.20	—	ns
		LFE2-70	1.80	—	2.00	—	2.20	—	ns
		LFE2M20	1.80	—	2.00	—	2.20	—	ns
		LFE2M35	1.80	—	2.00	—	2.20	—	ns
		LFE2M50	1.90	—	2.10	—	2.30	—	ns
		LFE2M70	1.90	—	2.10	—	2.30	—	ns
		LFE2M100	2.00	—	2.20	—	2.40	—	ns
t_{H_DELPLL}	Clock to Data Hold - PIO Input Register with Input Data Delay	LFE2-6	0.00	—	0.00	—	0.00	—	ns
		LFE2-12	0.00	—	0.00	—	0.00	—	ns
		LFE2-20	0.00	—	0.00	—	0.00	—	ns
		LFE2-35	0.00	—	0.00	—	0.00	—	ns
		LFE2-50	0.00	—	0.00	—	0.00	—	ns
		LFE2-70	0.00	—	0.00	—	0.00	—	ns
		LFE2M20	0.00	—	0.00	—	0.00	—	ns
		LFE2M35	0.00	—	0.00	—	0.00	—	ns
		LFE2M50	0.00	—	0.00	—	0.00	—	ns
		LFE2M70	0.00	—	0.00	—	0.00	—	ns
		LFE2M100	0.00	—	0.00	—	0.00	—	ns
DDR I/O Pin Parameters[2]									
t_{DVADQ}	Data Valid After DQS (DDR Read)	ECP2/M	—	0.225	—	0.225	—	0.225	UI
t_{DVEDQ}	Data Hold After DQS (DDR Read)	ECP2/M	0.640	—	0.640	—	0.640	—	UI
t_{DQVBS}	Data Valid Before DQS (DDR Write)	ECP2/M	0.250	—	0.250	—	0.250	—	UI
t_{DQVAS}	Data Valid After DQS (DDR Write)	ECP2/M	0.250	—	0.250	—	0.250	—	UI
f_{MAX_DDR}	DDR Clock Frequency[6]	ECP2/M	95	200	95	166	95	133	MHz
DDR2 I/O Pin Parameters[3]									
t_{DVADQ}	Data Valid After DQS (DDR Read)	ECP2/M	—	0.225	—	0.225	—	0.225	UI
t_{DVEDQ}	Data Hold After DQS (DDR Read)	ECP2/M	0.640	—	0.640	—	0.640	—	UI

LatticeECP2/M External Switching Characteristics[9] (Continued)

Over Recommended Operating Conditions

Parameter	Description	Device	-7		-6		-5		Units
			Min.	Max.	Min.	Max.	Min.	Max.	
t_{DQVBS}	Data Valid Before DQS (DDR Write)	ECP2/M	0.250	—	0.250	—	0.250	—	UI
t_{DQVAS}	Data Valid After DQS (DDR Write)	ECP2/M	0.250	—	0.250	—	0.250	—	UI
f_{MAX_DDR2}	DDR Clock Frequency	ECP2/M	133	266	133	200	133	166	MHz
SPI4.2 I/O Pin Parameters Static Alignment[4, 8, 11]									
	Maximum Data Rate	ECP2-20	—	750	—	622	—	622	Mbps
		ECP2-35	—	750	—	622	—	622	Mbps
		ECP2-50	—	750	—	622	—	622	Mbps
		ECP2-70	—	750	—	622	—	622	Mbps
		ECP2M20	—	622	—	622	—	622	Mbps
		ECP2M35	—	622	—	622	—	622	Mbps
		ECP2M50	—	622	—	622	—	622	Mbps
		ECP2M70	—	622	—	622	—	622	Mbps
		ECP2M100	—	622	—	622	—	622	Mbps
$t_{DVACLKSPI}$	Data Valid After CLK (Receive)	ECP2-20	—	0.25	—	0.25	—	0.25	UI
		ECP2-35	—	0.25	—	0.25	—	0.25	UI
		ECP2-50	—	0.25	—	0.25	—	0.25	UI
		ECP2-70	—	0.25	—	0.25	—	0.25	UI
		ECP2M20	—	0.21	—	0.21	—	0.21	UI
		ECP2M35	—	0.21	—	0.21	—	0.21	UI
		ECP2M50	—	0.21	—	0.21	—	0.21	UI
		ECP2M70	—	0.21	—	0.21	—	0.21	UI
		ECP2M100	—	0.21	—	0.21	—	0.21	UI
$t_{DVECLKSPI}$	Data Hold After CLK (Receive)	ECP2-20	0.75	—	0.75	—	0.75	—	UI
		ECP2-35	0.75	—	0.75	—	0.75	—	UI
		ECP2-50	0.75	—	0.75	—	0.75	—	UI
		ECP2-70	0.75	—	0.75	—	0.75	—	UI
		ECP2M20	0.79	—	0.79	—	0.79	—	UI
		ECP2M35	0.79	—	0.79	—	0.79	—	UI
		ECP2M50	0.79	—	0.79	—	0.79	—	UI
		ECP2M70	0.79	—	0.79	—	0.79	—	UI
		ECP2M100	0.79	—	0.79	—	0.79	—	UI
t_{DIASPI}	Data Invalid After Clock (Transmit)	ECP2-20	—	280	—	280	—	280	ps
		ECP2-35	—	280	—	280	—	280	ps
		ECP2-50	—	280	—	280	—	280	ps
		ECP2-70	—	280	—	280	—	280	ps
		ECP2M20	—	230	—	230	—	230	ps
		ECP2M35	—	230	—	230	—	230	ps
		ECP2M50	—	230	—	230	—	230	ps
		ECP2M70	—	230	—	230	—	230	ps
		ECP2M100	—	230	—	230	—	230	ps

LatticeECP2/M External Switching Characteristics[9] (Continued)

Over Recommended Operating Conditions

Parameter	Description	Device	-7		-6		-5		Units
			Min.	Max.	Min.	Max.	Min.	Max.	
t_{DIBSPI}	Data Invalid Before Clock (Transmit)	ECP2-20	—	280	—	280	—	280	ps
		ECP2-35	—	280	—	280	—	280	ps
		ECP2-50	—	280	—	280	—	280	ps
		ECP2-70	—	280	—	280	—	280	ps
		ECP2M20	—	230	—	230	—	230	ps
		ECP2M35	—	230	—	230	—	230	ps
		ECP2M50	—	230	—	230	—	230	ps
		ECP2M70	—	230	—	230	—	230	ps
		ECP2M100	—	230	—	230	—	230	ps
XGMII I/O Pin Parameters (312 Mbps)[5]									
$t_{SUXGMII}$	Data Setup Before Read Clock	ECP2/M	480	—	480	—	480	—	ps
t_{HXGMII}	Data Hold After Read Clock	ECP2/M	480	—	480	—	480	—	ps
$t_{DVBCKXGMII}$	Data Valid Before Clock	ECP2/M	960	—	960	—	960	—	ps
$t_{DVACKXGMII}$	Data Valid After Clock	ECP2/M	960	—	960	—	960	—	ps
Primary									
f_{MAX_PRI}[7]	Frequency for Primary Clock Tree	ECP2/M	—	420	—	357	—	311	MHz
t_{W_PRI}	Clock Pulse Width for Primary Clock	ECP2/M	0.95	—	1.19	—	2.00	—	ns
t_{SKEW_PRI}	Primary Clock Skew Within a Bank	ECP2/M	—	300	—	360	—	420	ps
Edge Clock									
f_{MAX_EDGE}[7]	Frequency for Edge Clock	ECP2/M	—	420	—	357	—	311	MHz
t_{W_EDGE}	Clock Pulse Width for Edge Clock	ECP2/M	0.95	—	1.19	—	2.00	—	ns
t_{SKEW_EDGE}	Edge Clock Skew Within an Edge of the Device	ECP2/M	—	300	—	360	—	420	ps

1. General timing numbers based on LVCMOS 2.5, 12mA, 0pf load.
2. DDR timing numbers based on SSTL25 for BGA packages only.
3. DDR2 timing numbers based on SSTL18 for BGA packages only.
4. SPI4.2 and SFI4 timing numbers based on LVDS25 for BGA packages only.
5. XGMII timing numbers based on HSTL class I. A corresponding left/right dedicated clock buffer is used when using the SPI4.2 interface to the left or right edge of the device. For SPI4.2 mode, the software tool will help in selecting the appropriate clock buffer.
6. IP will be used to support DDR and DDR2 memory data rates down to 95MHz. This approach uses a free-running clock and PFU register to sample the data instead of the hardwired DDR memory interface.
7. Using the LVDS I/O standard.
8. ECP2-6 and ECP2-12 do not support SPI4.2
9. The AC numbers do not apply to PCLK6 and PCLK7.
10. Applies to CLKOP only.
11. Please refer to TN1159, LatticeECP2/M Pin Assignment Recommendations for best performance.

Figure 3-6. SPI4.2 Parameters

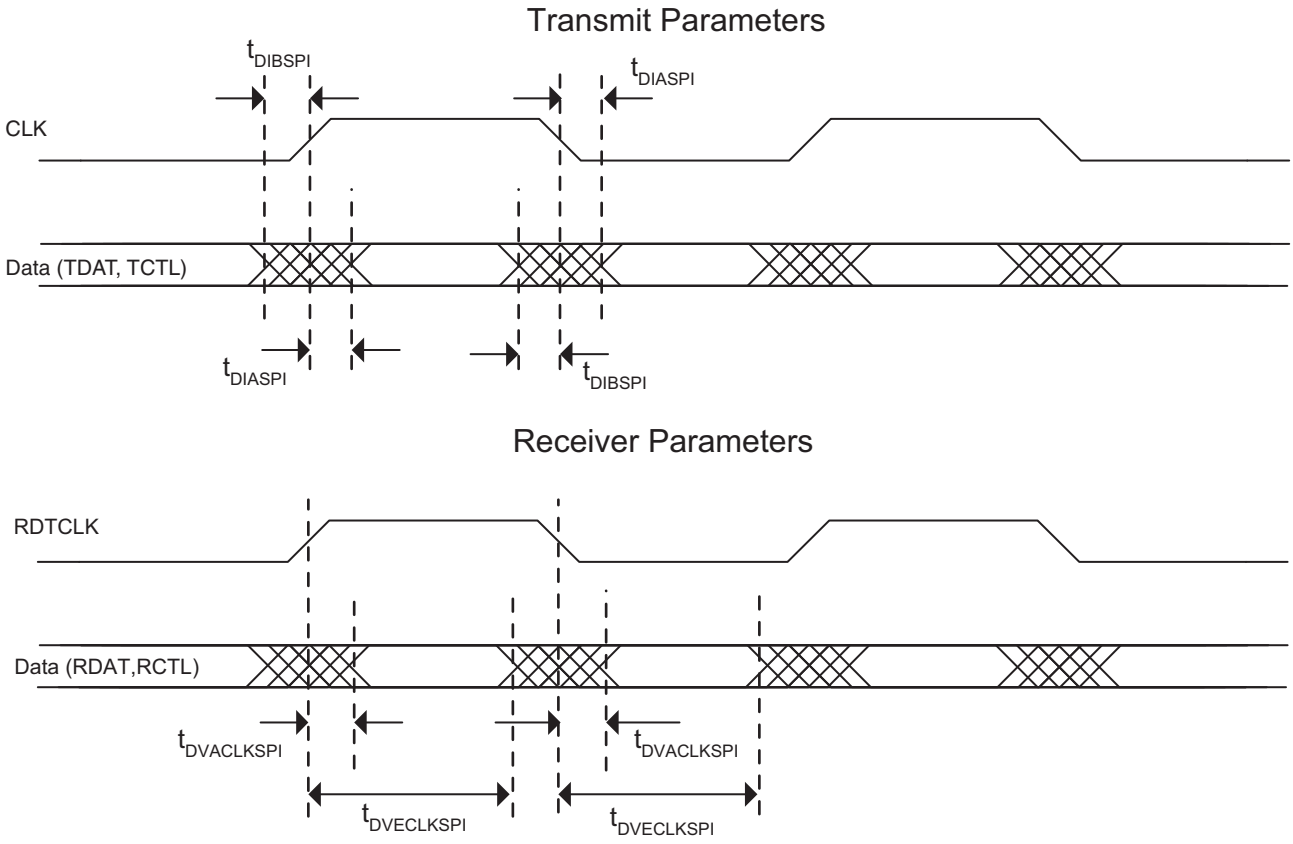


Figure 3-7. DDR and DDR2 Parameters

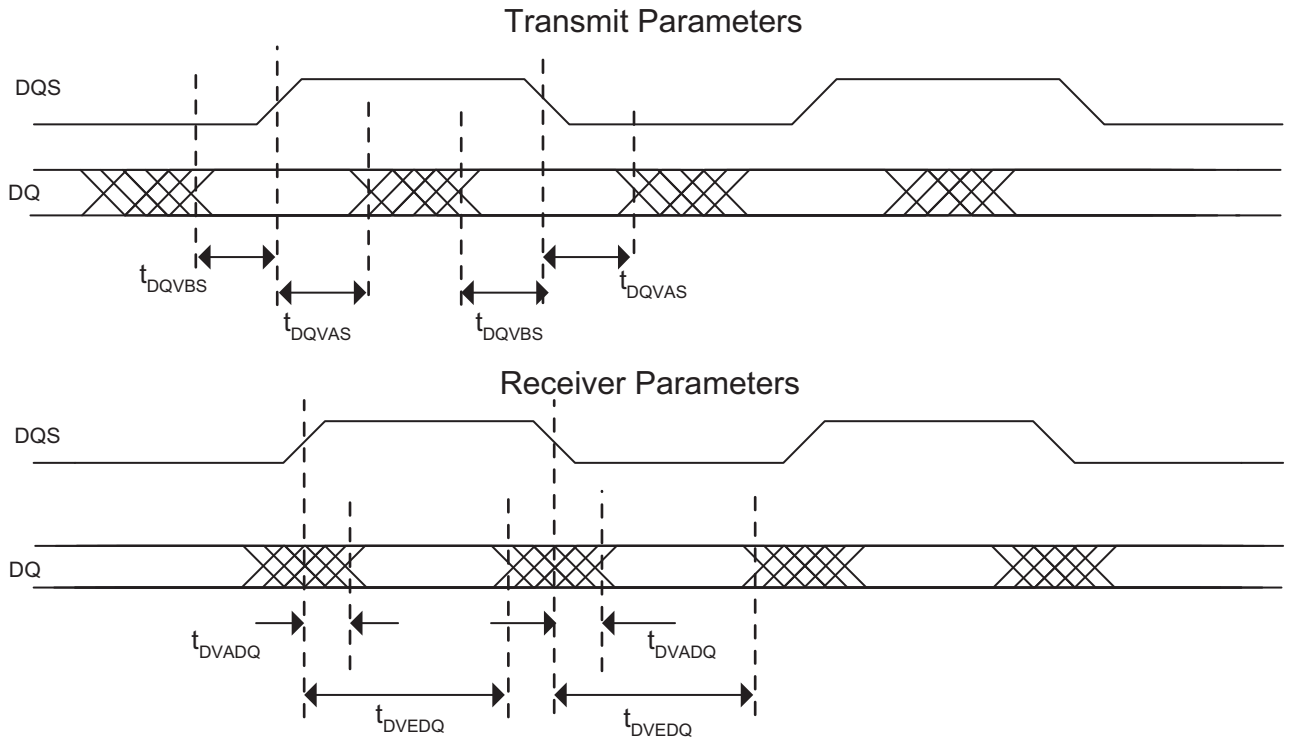
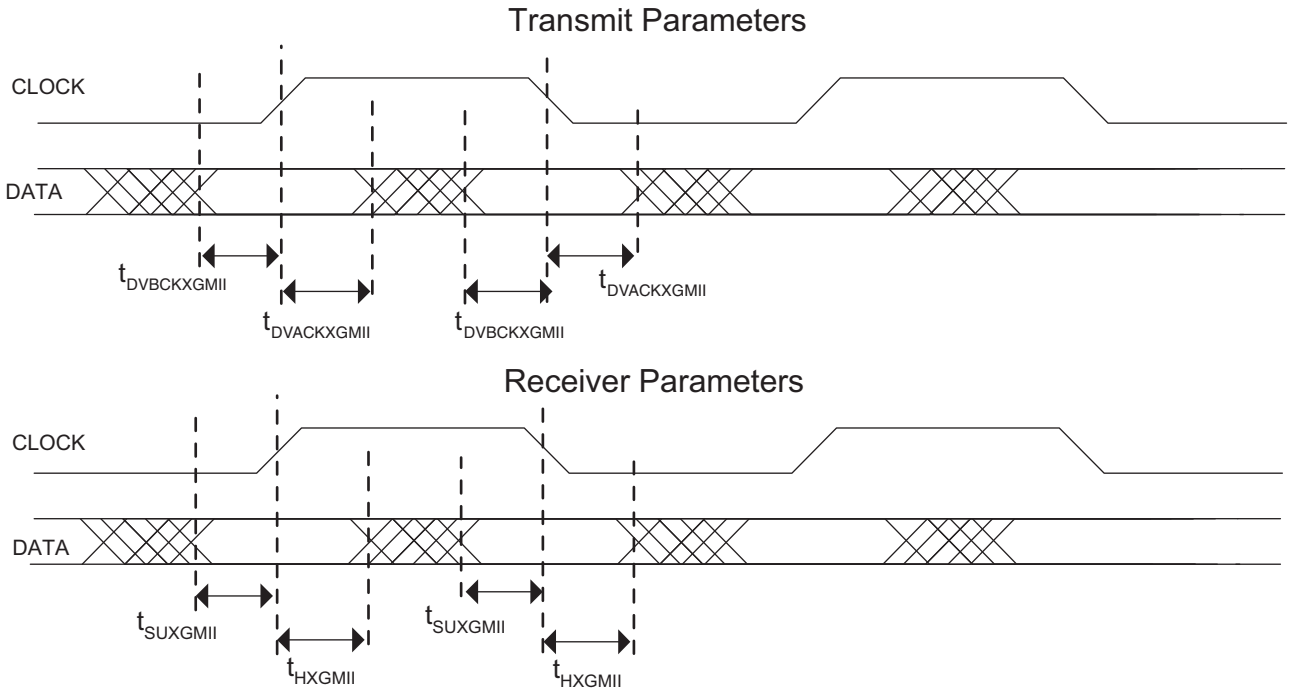


Figure 3-8. XGMII Parameters



LatticeECP2/M Internal Switching Characteristics[1]

Over Recommended Operating Conditions

Parameter	Description	-7		-6		-5		Units
		Min.	Max.	Min.	Max.	Min.	Max.	
PFU/PFF Logic Mode Timing								
t_{LUT4_PFU}	LUT4 delay (A to D inputs to F output)	—	0.180	—	0.198	—	0.216	ns
t_{LUT6_PFU}	LUT6 delay (A to D inputs to OFX output)	—	0.304	—	0.331	—	0.358	ns
t_{LSR_PFU}	Set/Reset to output of PFU (Asynchronous)	—	0.600	—	0.655	—	0.711	ns
t_{SUM_PFU}	Clock to Mux (M0,M1) Input Setup Time	0.128	—	0.129	—	0.129	—	ns
t_{HM_PFU}	Clock to Mux (M0,M1) Input Hold Time	-0.051	—	-0.049	—	-0.046	—	ns
t_{SUD_PFU}	Clock to D input setup time	0.061	—	0.071	—	0.081	—	ns
t_{HD_PFU}	Clock to D input hold time	0.002	—	0.003	—	0.003	—	ns
t_{CK2Q_PFU}	Clock to Q delay, (D-type Register Configuration)	—	0.285	—	0.309	—	0.333	ns
PFU Dual Port Memory Mode Timing								
t_{CORAM_PFU}	Clock to Output (F Port)	—	0.902	—	1.083	—	1.263	ns
t_{SUDATA_PFU}	Data Setup Time	-0.172	—	-0.205	—	-0.238	—	ns
t_{HDATA_PFU}	Data Hold Time	0.199	—	0.235	—	0.271	—	ns
t_{SUADDR_PFU}	Address Setup Time	-0.245	—	-0.284	—	-0.323	—	ns
t_{HADDR_PFU}	Address Hold Time	0.246	—	0.285	—	0.324	—	ns
t_{SUWREN_PFU}	Write/Read Enable Setup Time	-0.122	—	-0.145	—	-0.168	—	ns
t_{HWREN_PFU}	Write/Read Enable Hold Time	0.132	—	0.156	—	0.180	—	ns
PIC Timing								
PIO Input/Output Buffer Timing								
t_{IN_PIO}	Input Buffer Delay (LVCMOS25)	—	0.613	—	0.681	—	0.749	ns
t_{OUT_PIO}	Output Buffer Delay (LVCMOS25)	—	1.115	—	1.115	—	1.343	ns
IOLOGIC Input/Output Timing								
t_{SUI_PIO}	Input Register Setup Time (Data Before Clock)	0.596	—	0.645	—	0.694	—	ns
t_{HI_PIO}	Input Register Hold Time (Data after Clock)	-0.570	—	-0.614	—	-0.658	—	ns
t_{COO_PIO}	Output Register Clock to Output Delay	—	0.61	—	0.66	—	0.72	ns
t_{SUCE_PIO}	Input Register Clock Enable Setup Time	0.032	—	0.037	—	0.041	—	ns
t_{HCE_PIO}	Input Register Clock Enable Hold Time	-0.022	—	-0.025	—	-0.028	—	ns
t_{SULSR_PIO}	Set/Reset Setup Time	0.184	—	0.201	—	0.217	—	ns
t_{HLSR_PIO}	Set/Reset Hold Time	-0.080	—	-0.086	—	-0.093	—	ns
EBR Timing								
t_{CO_EBR}	Clock (Read) to output from Address or Data	—	2.51	—	2.75	—	2.99	ns
t_{COO_EBR}	Clock (Write) to output from EBR output Register	—	0.33	—	0.36	—	0.39	ns
t_{SUDATA_EBR}	Setup Data to EBR Memory	-0.157	—	-0.181	—	-0.205	—	ns
t_{HDATA_EBR}	Hold Data to EBR Memory	0.173	—	0.195	—	0.217	—	ns
t_{SUADDR_EBR}	Setup Address to EBR Memory	-0.115	—	-0.130	—	-0.145	—	ns
t_{HADDR_EBR}	Hold Address to EBR Memory	0.138	—	0.155	—	0.172	—	ns
t_{SUWREN_EBR}	Setup Write/Read Enable to PFU Memory	-0.128	—	-0.149	—	-0.170	—	ns

LatticeECP2/M Internal Switching Characteristics[1] (Continued)

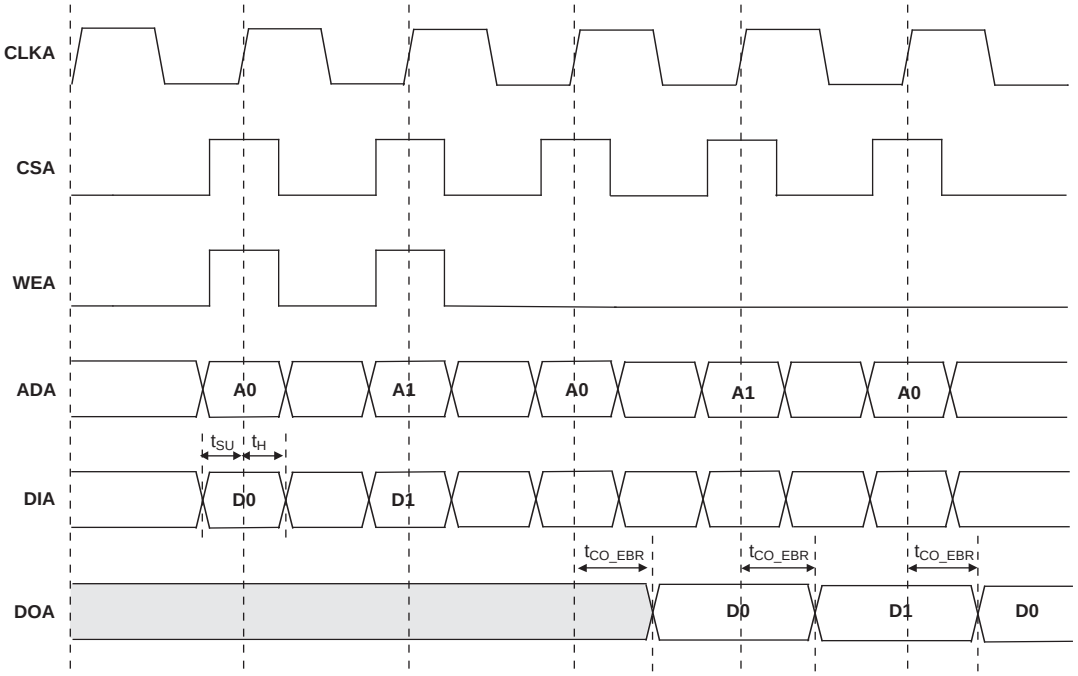
Over Recommended Operating Conditions

Parameter	Description	-7		-6		-5		Units
		Min.	Max.	Min.	Max.	Min.	Max.	
t_{HWREN_EBR}	Hold Write/Read Enable to PFU Memory	0.139	—	0.156	—	0.173	—	ns
t_{SUCE_EBR}	Clock Enable Setup Time to EBR Output Register	0.123	—	0.134	—	0.145	—	ns
t_{HCE_EBR}	Clock Enable Hold Time to EBR Output Register	-0.081	—	-0.090	—	-0.100	—	ns
t_{RSTO_EBR}	Reset To Output Delay Time from EBR Output Register	—	1.03	—	1.15	—	1.26	ns
t_{SUBE_EBR}	Byte Enable Set-Up Time to EBR Output Register	-0.115	—	-0.130	—	-0.145	—	ns
t_{HBE_EBR}	Byte Enable Hold Time to EBR Output Register	0.138	—	0.155	—	0.172	—	ns
GPLL Parameters								
t_{RSTREC_GPLL}	Reset Recovery to Rising Clock	1.00	—	1.00	—	1.00	—	ns
SPLL Parameters								
t_{RSTREC_SPLL}	Reset Recovery to Rising Clock	1.00	—	1.00	—	1.00	—	ns
DSP Block Timing[2,3]								
t_{SUI_DSP}	Input Register Setup Time	0.12	—	0.13	—	0.14	—	ns
t_{HI_DSP}	Input Register Hold Time	0.02	—	-0.01	—	-0.03	—	ns
t_{SUP_DSP}	Pipeline Register Setup Time	2.18	—	2.42	—	2.66	—	ns
t_{tHP_DSP}	Pipeline Register Hold Time	-0.68	—	-0.77	—	-0.86	—	ns
t_{SUO_DSP}	Output Register Setup Time	4.26	—	4.71	—	5.16	—	ns
t_{HO_DSP}	Output Register Hold Time	-1.25	—	-1.40	—	-1.54	—	ns
t_{COI_DSP}	Input Register Clock to Output Time	—	3.92	—	4.30	—	4.68	ns
t_{COP_DSP}	Pipeline Register Clock to Output Time	—	1.87	—	1.98	—	2.08	ns
t_{COO_DSP}	Output Register Clock to Output Time	—	0.50	—	0.52	—	0.55	ns
$t_{SUADDSUB}$	AddSub Input Register Setup Time	-0.24	—	-0.26	—	-0.28	—	ns
$t_{HADDSUB}$	AddSub Input Register Hold Time	0.27	—	0.29	—	0.32	—	ns

1. Internal parameters are characterized but not tested on every device.
2. These parameters apply to LatticeECP devices only.
3. DSP Block is configured in Multiply Add/Sub 18x18 Mode.

Timing Diagrams

Figure 3-9. Read/Write Mode (Normal)



Note: Input data and address are registered at the positive edge of the clock and output data appears after the positive edge of the clock.

Figure 3-10. Read/Write Mode with Input and Output Registers

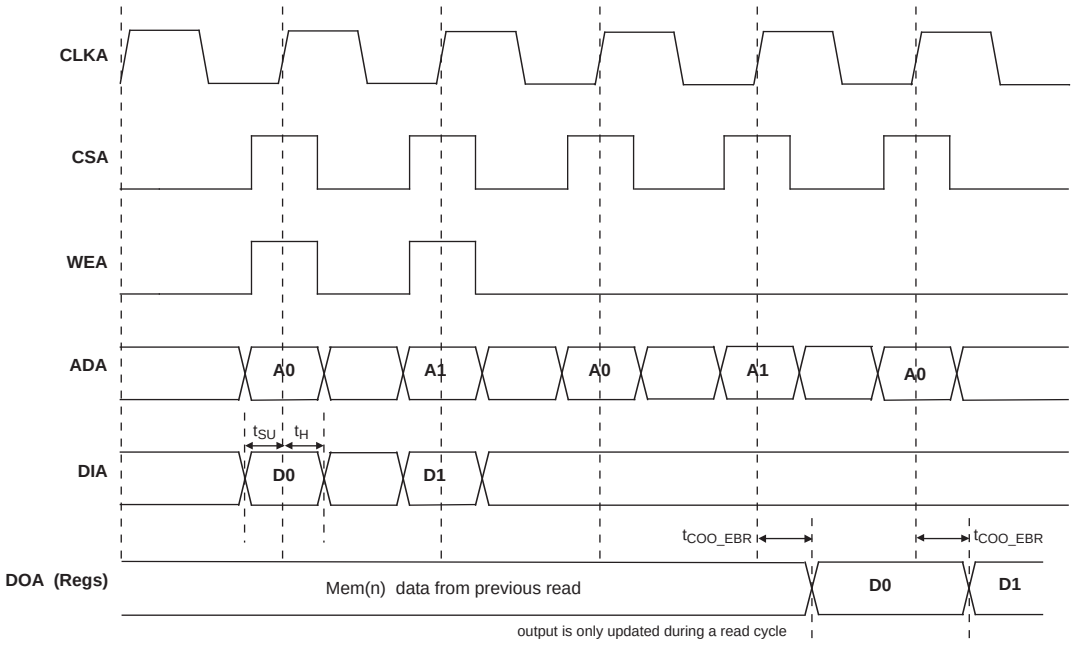
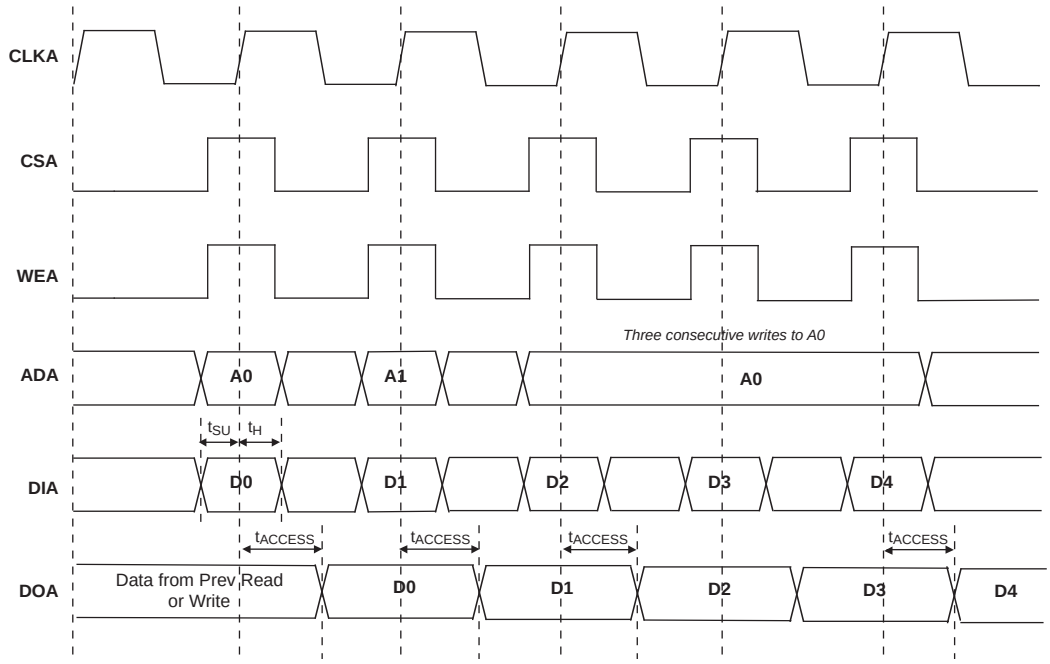


Figure 3-11. Write Through (SP Read/Write on Port A, Input Registers Only)



Note: Input data and address are registered at the positive edge of the clock and output data appears after the positive edge of the clock.

LatticeECP2/M Family Timing Adders[1, 2, 3]

Over Recommended Operating Conditions

Buffer Type	Description	-7	-6	-5	Units
Input Adjusters					
LVDS25	LVDS	-0.04	-0.02	0.00	ns
BLVDS25	BLVDS	-0.04	-0.09	-0.15	ns
MLVDS	LVDS	-0.15	-0.15	-0.15	ns
RSDS	RSDS	-0.15	-0.15	-0.15	ns
LVPECL33	LVPECL	0.16	0.15	0.13	ns
HSTL18_I	HSTL_18 class I	0.01	-0.01	-0.04	ns
HSTL18_II	HSTL_18 class II	0.01	-0.01	-0.04	ns
HSTL18D_I	Differential HSTL 18 class I	0.01	-0.01	-0.04	ns
HSTL18D_II	Differential HSTL 18 class II	0.01	-0.01	-0.04	ns
HSTL15_I	HSTL_15 class I	0.01	-0.01	-0.04	ns
HSTL15D_I	Differential HSTL 15 class I	0.01	-0.01	-0.04	ns
SSTL33_I	SSTL_3 class I	-0.03	-0.07	-0.10	ns
SSTL33_II	SSTL_3 class II	-0.03	-0.07	-0.10	ns
SSTL33D_I	Differential SSTL_3 class I	-0.03	-0.07	-0.10	ns
SSTL33D_II	Differential SSTL_3 class II	-0.03	-0.07	-0.10	ns
SSTL25_I	SSTL_2 class I	-0.04	-0.07	-0.10	ns
SSTL25_II	SSTL_2 class II	-0.04	-0.07	-0.10	ns
SSTL25D_I	Differential SSTL_2 class I	-0.04	-0.07	-0.10	ns
SSTL25D_II	Differential SSTL_2 class II	-0.04	-0.07	-0.10	ns
SSTL18_I	SSTL_18 class I	-0.01	-0.04	-0.07	ns
SSTL18_II	SSTL_18 class II	-0.01	-0.04	-0.07	ns
SSTL18D_I	Differential SSTL_18 class I	-0.01	-0.04	-0.07	ns
SSTL18D_II	Differential SSTL_18 class II	-0.01	-0.04	-0.07	ns
LVTTL33	LVTTL	-0.16	-0.16	-0.16	ns
LVCMOS33	LVCMOS 3.3	-0.08	-0.12	-0.16	ns
LVCMOS25	LVCMOS 2.5	0.00	0.00	0.00	ns
LVCMOS18	LVCMOS 1.8	-0.16	-0.17	-0.17	ns
LVCMOS15	LVCMOS 1.5	-0.14	-0.14	-0.14	ns
LVCMOS12	LVCMOS 1.2	-0.04	-0.01	0.01	ns
PCI33	PCI	-0.08	-0.12	-0.16	ns
Output Adjusters					
LVDS25E	LVDS 2.5 E[4]	0.25	0.19	0.13	ns
LVDS25	LVDS 2.5	0.10	0.13	0.17	ns
BLVDS25	BLVDS 2.5	0.00	-0.01	-0.03	ns
MLVDS	MLVDS 2.5[4]	0.00	-0.01	-0.03	ns
RSDS	RSDS 2.5[4]	0.25	0.19	0.13	ns
LVPECL33	LVPECL 3.3[4]	-0.02	-0.04	-0.06	ns
HSTL18_I	HSTL_18 class I 8mA drive	-0.19	-0.22	-0.25	ns
HSTL18_II	HSTL_18 class II	-0.30	-0.34	-0.37	ns
HSTL18D_I	Differential HSTL 18 class I 8mA drive	-0.19	-0.22	-0.25	ns
HSTL18D_II	Differential HSTL 18 class II	-0.30	-0.34	-0.37	ns

LatticeECP2/M Family Timing Adders[1, 2, 3] (Continued)

Over Recommended Operating Conditions

Buffer Type	Description	-7	-6	-5	Units
HSTL15_I	HSTL_15 class I 4mA drive	-0.22	-0.25	-0.27	ns
HSTL15D_I	Differential HSTL 15 class I 4mA drive	-0.22	-0.25	-0.27	ns
SSTL33_I	SSTL_3 class I	-0.12	-0.15	-0.18	ns
SSTL33_II	SSTL_3 class II	-0.20	-0.23	-0.27	ns
SSTL33D_I	Differential SSTL_3 class I	-0.12	-0.15	-0.18	ns
SSTL33D_II	Differential SSTL_3 class II	-0.20	-0.23	-0.27	ns
SSTL25_I	SSTL_2 class I 8mA drive	-0.16	-0.19	-0.22	ns
SSTL25_II	SSTL_2 class II 16mA drive	-0.19	-0.22	-0.25	ns
SSTL25D_I	Differential SSTL_2 class I 8mA drive	-0.16	-0.19	-0.22	ns
SSTL25D_II	Differential SSTL_2 class II 16mA drive	-0.19	-0.22	-0.25	ns
SSTL18_I	SSTL_1.8 class I	-0.14	-0.17	-0.20	ns
SSTL18_II	SSTL_1.8 class II 8mA drive	-0.20	-0.23	-0.25	ns
SSTL18D_I	Differential SSTL_1.8 class I	-0.14	-0.17	-0.20	ns
SSTL18D_II	Differential SSTL_1.8 class II 8mA drive	-0.20	-0.23	-0.25	ns
LVTTL33_4mA	LVTTL 4mA drive	0.52	0.60	0.68	ns
LVTTL33_8mA	LVTTL 8mA drive	0.06	0.08	0.09	ns
LVTTL33_12mA	LVTTL 12mA drive	0.04	0.04	0.05	ns
LVTTL33_16mA	LVTTL 16mA drive	0.03	0.02	0.02	ns
LVTTL33_20mA	LVTTL 20mA drive	-0.09	-0.09	-0.10	ns
LVCMOS33_4mA	LVCMOS 3.3 4mA drive, fast slew rate	0.52	0.60	0.68	ns
LVCMOS33_8mA	LVCMOS 3.3 8mA drive, fast slew rate	0.06	0.08	0.09	ns
LVCMOS33_12mA	LVCMOS 3.3 12mA drive, fast slew rate	0.04	0.04	0.05	ns
LVCMOS33_16mA	LVCMOS 3.3 16mA drive, fast slew rate	0.03	0.02	0.02	ns
LVCMOS33_20mA	LVCMOS 3.3 20mA drive, fast slew rate	-0.09	-0.09	-0.10	ns
LVCMOS25_4mA	LVCMOS 2.5 4mA drive, fast slew rate	0.41	0.47	0.53	ns
LVCMOS25_8mA	LVCMOS 2.5 8mA drive, fast slew rate	0.01	0.01	0.00	ns
LVCMOS25_12mA	LVCMOS 2.5 12mA drive, fast slew rate	0.00	0.00	0.00	ns
LVCMOS25_16mA	LVCMOS 2.5 16mA drive, fast slew rate	0.04	0.04	0.04	ns
LVCMOS25_20mA	LVCMOS 2.5 20mA drive, fast slew rate	-0.09	-0.10	-0.11	ns
LVCMOS18_4mA	LVCMOS 1.8 4mA drive, fast slew rate	0.37	0.40	0.43	ns
LVCMOS18_8mA	LVCMOS 1.8 8mA drive, fast slew rate	0.10	0.12	0.13	ns
LVCMOS18_12mA	LVCMOS 1.8 12mA drive, fast slew rate	-0.02	-0.02	-0.02	ns
LVCMOS18_16mA	LVCMOS 1.8 16mA drive, fast slew rate	-0.02	-0.03	-0.03	ns
LVCMOS15_4mA	LVCMOS 1.5 4mA drive, fast slew rate	0.29	0.31	0.32	ns
LVCMOS15_8mA	LVCMOS 1.5 8mA drive, fast slew rate	0.05	0.05	0.06	ns
LVCMOS12_2mA	LVCMOS 1.2 2mA drive, fast slew rate	0.58	0.69	0.79	ns
LVCMOS12_6mA	LVCMOS 1.2 6mA drive, fast slew rate	0.13	0.19	0.26	ns
LVCMOS33_4mA	LVCMOS 3.3 4mA drive, slow slew rate	2.17	2.44	2.71	ns
LVCMOS33_8mA	LVCMOS 3.3 8mA drive, slow slew rate	2.50	2.67	2.83	ns
LVCMOS33_12mA	LVCMOS 3.3 12mA drive, slow slew rate	1.72	1.88	2.05	ns
LVCMOS33_16mA	LVCMOS 3.3 16mA drive, slow slew rate	1.64	1.63	1.62	ns
LVCMOS33_20mA	LVCMOS 3.3 20mA drive, slow slew rate	1.33	1.36	1.39	ns

LatticeECP2/M Family Timing Adders[1, 2, 3] (Continued)

Over Recommended Operating Conditions

Buffer Type	Description	-7	-6	-5	Units
LVCMOS25_4mA	LVCMOS 2.5 4mA drive, slow slew rate	2.18	2.26	2.33	ns
LVCMOS25_8mA	LVCMOS 2.5 8mA drive, slow slew rate	2.19	2.35	2.51	ns
LVCMOS25_12mA	LVCMOS 2.5 12mA drive, slow slew rate	1.50	1.66	1.82	ns
LVCMOS25_16mA	LVCMOS 2.5 16mA drive, slow slew rate	1.60	1.59	1.58	ns
LVCMOS25_20mA	LVCMOS 2.5 20mA drive, slow slew rate	1.43	1.39	1.34	ns
LVCMOS18_4mA	LVCMOS 1.8 4mA drive, slow slew rate	2.22	2.27	2.32	ns
LVCMOS18_8mA	LVCMOS 1.8 8mA drive, slow slew rate	1.93	2.08	2.23	ns
LVCMOS18_12mA	LVCMOS 1.8 12mA drive, slow slew rate	1.43	1.51	1.58	ns
LVCMOS18_16mA	LVCMOS 1.8 16mA drive, slow slew rate	1.47	1.46	1.45	ns
LVCMOS15_4mA	LVCMOS 1.5 4mA drive, slow slew rate	2.32	2.38	2.43	ns
LVCMOS15_8mA	LVCMOS 1.5 8mA drive, slow slew rate	1.84	1.98	2.12	ns
LVCMOS12_2mA	LVCMOS 1.2 2mA drive, slow slew rate	2.52	2.63	2.74	ns
LVCMOS12_6mA	LVCMOS 1.2 6mA drive, slow slew rate	1.69	1.83	1.96	ns
PCI33	PCI33	0.04	0.04	0.04	ns

1. Timing Adders are characterized but not tested on every device.
2. LVCMOS timing measured with the load specified in Switching Test Condition table.
3. All other standards tested according to the appropriate specifications.
4. These timing adders are measured with the recommended resistor values.

Timing v.A 0.11

sysCLOCK GPLL Timing

Over Recommended Operating Conditions

Parameter	Description	Conditions	Min.	Typ.	Max.	Units
f_{IN}	Input Clock Frequency (CLKI, CLKFB)	Without external capacitor	20	—	420	MHz
		With external capacitor[5, 6]	2	—	420	MHz
f_{OUT}	Output Clock Frequency (CLKOP, CLKOS)	Without external capacitor	20	—	420	MHz
		With external capacitor[5]	5	—	50	MHz
f_{OUT2}	K-Divider Output Frequency (CLKOK)	Without external capacitor	0.156	—	210	MHz
		With external capacitor[5]	0.039	—	25	MHz
f_{VCO}	PLL VCO Frequency		640	—	1280	MHz
f_{PFD}	Phase Detector Input Frequency	Without external capacitor	20	—	420	MHz
		With external capacitor[5, 6]	2	—	50	MHz
AC Characteristics						
t_{DT}	Output Clock Duty Cycle	Default duty cycle selected[3]	45	50	55	%
t_{PH}[4]	Output Phase Accuracy		—	—	±0.05	UI
t_{OPJIT}[1]	Output Clock Period Jitter	$f_{OUT} \geq 100$ MHz	—	—	±125	ps
		$50 \leq f_{OUT} < 100$ MHz	—		0.025	UIPP
		$f_{OUT} < 50$ MHz	—	—	0.04	UIPP
t_{SK}	Input Clock to Output Clock Skew	N/M = integer	—	—	±250	ps
t_W	Output Clock Pulse Width	At 90% or 10%	1	—	—	ns
t_{LOCK}[2]	PLL Lock-in Time	Without external capacitor	—	—	150	µs
		With external capacitor[5]	—	—	500	µs
t_{PA}	Programmable Delay Unit		85	130	360	ps
t_{IPJIT}	Input Clock Period Jitter		—	—	±200	ps
t_{FBKDLY}	External Feedback Delay		—	—	10	ns
t_{HI}	Input Clock High Time	90% to 90%	0.5	—	—	ns
t_{LO}	Input Clock Low Time	10% to 10%	0.5	—	—	ns
t_{RST}	RST Pulse Width (RESETM/RESETK)		15	—	—	ns
	Reset Signal Pulse Width (CNTRST)	Without external capacitor	500	—	—	ns
		With external capacitor[5]	20	—	—	µs

1. Jitter sample is taken over 10,000 samples of the primary PLL output with clean reference clock and no additional I/O pins toggling.
2. Output clock is valid after t_{LOCK} for PLL reset and dynamic delay adjustment.
3. Using LVDS output buffers.
4. Relative to CLKOP.
5. Value of external capacitor: 5.6 nF ±20%, NPO dielectric, ceramic chip capacitor, 1206 or smaller package, connected to PLLCAP pin.
6. f_{OUT} (max) = f_{IN} * 10 for f_{IN} < 5MHz.

sysCLOCK SPLL Timing

Over Recommended Operating Conditions

Parameter	Description	Conditions	Min.	Typ.	Max.	Units
f_{IN}	Input Clock Frequency (CLKI, CLKFB)	Without external capacitor	33	—	420	MHz
		With external capacitor[5, 6]	2	—	420	MHz
f_{OUT}	Output Clock Frequency (CLKOP, CLKOS)	Without external capacitor	33	—	420	MHz
		With external capacitor[5]	5	—	50	MHz
f_{OUT2}	K-Divider Output Frequency (CLKOK)	Without external capacitor	0.258	—	210	MHz
		With external capacitor[5]	0.039	—	25	MHz
f_{VCO}	PLL VCO Frequency		640	—	1280	MHz
f_{PFD}	Phase Detector Input Frequency	Without external capacitor	33	—	420	MHz
		With external capacitor[6]	2	—	50	MHz
AC Characteristics						
t_{DT}	Output Clock Duty Cycle	Default Duty Cycle Selected[3]	45	50	55	%
t_{PH}[4]	Output Phase Accuracy		—	—	±0.05	UI
t_{OPJIT}[1]	Output Clock Period Jitter	$f_{OUT} \geq 100$ MHz	—	—	±125	ps
		$50 \leq f_{OUT} < 100$ MHz	—	—	0.025	UIPP
		$f_{OUT} < 50$ MHz	—	—	0.04	UIPP
t_{SK}	Input Clock to Output Clock Skew	Divider Ratio = Integer	—	—	±250	ps
t_W	Output Clock Pulse Width	At 90% or 10%	1	—	—	ns
t_{LOCK}[2]	PLL Lock-in Time	Without external capacitor	—	—	150	µs
		With external capacitor[5]	—	—	500	µs
t_{IPJIT}	Input Clock Period Jitter		—	—	±200	ps
t_{FBKDLY}	External Feedback Delay		—	—	10	ns
t_{HI}	Input Clock High Time	90% to 90%	0.5	—	—	ns
t_{LO}	Input Clock Low Time	10% to 10%	0.5	—	—	ns
t_{RST}	RST Pulse Width (RESETM/RESETK)		15	—	—	ns
	Reset Signal Pulse Width (CNTRST)	Without external capacitor	500	—	—	ns
		With external capacitor[5]	20	—	—	µs

1. Jitter sample is taken over 10,000 samples of the primary PLL output with clean reference clock and no additional I/O pins toggling.
2. Output clock is valid after t_{LOCK} for PLL reset and dynamic delay adjustment.
3. Using LVDS output buffers.
4. Phase accuracy of CLKOS compared to CLKOP.
5. Value of external capacitor: 5.6 nF ±20%, NPO dielectric, ceramic chip capacitor, 1206 or smaller package, connected to PLLCAP pin.
6. f_{OUT} (max) = f_{IN} * 10 for f_{IN} < 5MHz.

DLL Timing

Over Recommended Operating Conditions

Parameter	Description	Min.	Typ.	Max.	Units
f_{REF}	Input reference clock frequency (on-chip or off-chip)	100	—	500	MHz
f_{FB}	Feedback clock frequency (on-chip or off-chip)	100	—	500	MHz
f_{CLKOP}[1]	Output clock frequency, CLKOP	100	—	500	MHz
f_{CLKOS}[2]	Output clock frequency, CLKOS	25	—	500	MHz
t_{PJIT}	Output clock period jitter (clean input)		—	250	ps p-p
t_{CYJIT}	Output clock cycle to cycle jitter (clean input)			250	ps p-p
t_{DUTY}	Output clock duty cycle (at 50% levels, 50% duty cycle input clock, 50% duty cycle circuit turned off, time reference delay mode)	35		65	%
$t_{DUTYTRD}$	Output clock duty cycle (at 50% levels, arbitrary duty cycle input clock, 50% duty cycle circuit enabled, time reference delay mode)	40		60	%
$t_{DUTYCIR}$	Output clock duty cycle (at 50% levels, arbitrary duty cycle input clock, 50% duty cycle circuit enabled, clock injection removal mode)	40		60	%
t_{SKEW}[3]	Output clock to clock skew between two outputs with the same phase setting	—	—	100	ps
t_{PWH}	Input clock minimum pulse width high (at 80% level)	750	—	—	ps
t_{PWL}	Input clock minimum pulse width low (at 20% level)	750	—	—	ps
t_{INSTB}	Input clock period jitter	—	—	+/-250	ps
t_{LOCK}	DLL lock time	18,500	—	—	cycles
t_{RSWD}	Digital reset minimum pulse width (at 80% level)	3	—	—	ns
t_{PA}	Delay step size	16.5	42	59.4	ps
t_{RANGE1}	Max. delay setting for single delay block (144 taps)	2.376	6	8.553	ns
t_{RANGE4}	Max. delay setting for four chained delay blocks	9.504	24	34.214	ns

1. CLKOP runs at the same frequency as the input clock.
2. CLKOS minimum frequency is obtained with divide by 4.
3. This is intended to be a “path-matching” design guideline and is not a measurable specification.

SERDES High-Speed Data Transmitter (LatticeECP2M Family Only)[1, 2]

Table 3-7. Serial Output Timing and Levels

Symbol	Description	Frequency	Min.	Typ.	Max.	Units
$V_{TX-DIFF-P-P-1}$	Differential swing (1V setting)[1, 2]	0.25 to 3.125 Gbps	0.79	0.99	1.19	V, p-p
$V_{TX-DIFF-P-P-1.25}$	Differential swing (1.25V setting)[1, 2]	0.25 to 3.125 Gbps	1.00	1.25	1.50	V, p-p
$V_{TX-DIFF-P-P-1.3}$	Differential swing (1.3V setting)[1, 2]	0.25 to 3.125 Gbps	1.04	1.30	1.56	V, p-p
$V_{TX-DIFF-P-P-1.35}$	Differential swing (1.35V setting)[1, 2]	0.25 to 3.125 Gbps	1.08	1.35	1.62	V, p-p
V_{OCM}	Output common mode voltage	—	V_{CCOB} - 0.75	V_{CCOB} - 0.60	V_{CCOB} - 0.45	V
T_{TX-R}	Rise time (20% to 80%)	—	—	70	—	ps
T_{TX-F}	Fall time (80% to 20%)	—	—	70	—	ps
$Z_{TX-OI-SE}$	Output impedance 50/75/HiZ K Ohms (single-ended)	—	—	50/70 HiZ	—	Ohms
R_{TX-RL}	Return loss (with package)	—	—	9	—	dB

1. All measurements are with 50 ohm impedance.
2. See TN1124, LatticeECP2M SERDES/PCS Usage Guide for actual binary settings.

Table 3-8. Channel Output Jitter - x10 Mode

Description	Frequency	Min.	Typ.	Max.	Units
Deterministic	3.125 Gbps	—	0.08	0.12	UI, p-p
Random	3.125 Gbps	—	0.22	0.38	UI, p-p
Total	3.125 Gbps	—	0.33	0.43	UI, p-p
Deterministic	2.5 Gbps	—	0.08	0.17	UI, p-p
Random	2.5 Gbps	—	0.20	0.25	UI, p-p
Total	2.5 Gbps	—	0.25	0.35	UI, p-p
Deterministic	1.25 Gbps	—	0.03	0.10	UI, p-p
Random	1.25 Gbps	—	0.14	0.19	UI, p-p
Total	1.25 Gbps	—	0.17	0.24	UI, p-p
Deterministic	250 Mbps	—	0.04	0.17	UI, p-p
Random	250 Mbps	—	0.12	0.13	UI, p-p
Total	250 Mbps	—	0.15	0.29	UI, p-p

Note: Values are measured with PRBS 2^7-1, all channels operating, FPGA Logic active, I/Os around SERDES pins quiet, reference clock at x10 mode.

Table 3-9. Channel Output Jitter - x20 Mode

Description	Frequency	Min.	Typ.	Max.	Units
Deterministic	3.125 Gbps	—	0.08	0.12	UI, p-p
Random	3.125 Gbps	—	0.27	0.51	UI, p-p
Total	3.125 Gbps	—	0.35	0.59	UI, p-p
Deterministic	2.5 Gbps	—	0.09	0.19	UI, p-p
Random	2.5 Gbps	—	0.23	0.34	UI, p-p
Total	2.5 Gbps	—	0.29	0.45	UI, p-p
Deterministic	1.25 Gbps	—	0.05	0.11	UI, p-p
Random	1.25 Gbps	—	0.16	0.22	UI, p-p
Total	1.25 Gbps	—	0.20	0.28	UI, p-p

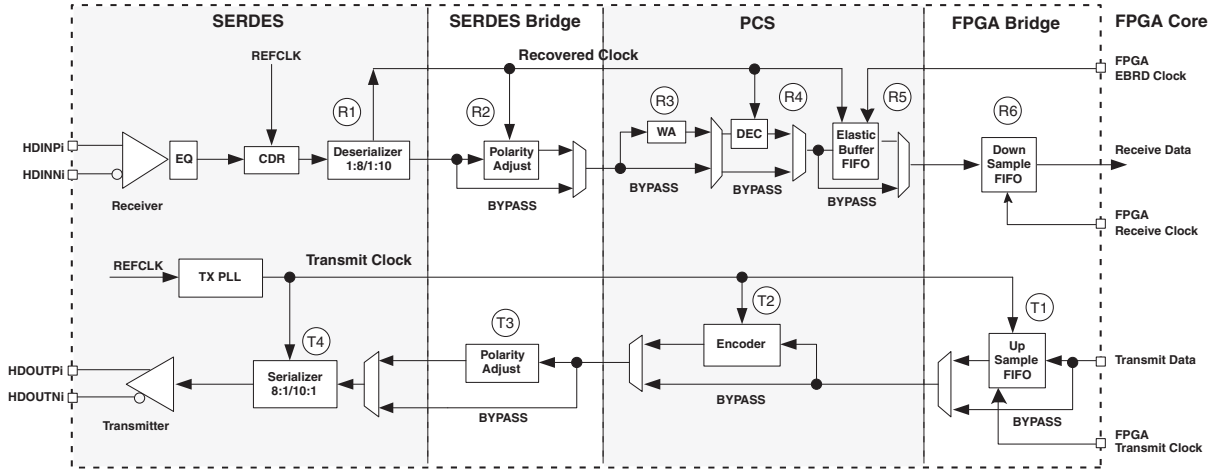
Note: Values are measured with PRBS 2^7-1, all channels operating, FPGA Logic active, I/Os around SERDES pins quiet, reference clock at x20 mode.

Table 3-10. SERDES/PCS Latency Breakdown (Parallel Clock Cycle)

Item	Description	Min.	Average	Max.	Fixed	Bypass	Units
Transmit Data Latency							
T1	FPGA Bridge Transmit[2]	1	3	5		1	word clk
T2	8b10b Encoder	—	—	—	2	1	word clk
T3	SERDES Bridge Transmit	—	—	—	2	1	word clk
T4[3]	Serializer: 8-bit mode	—	—	—	15 + Δ1	—	UI + ps
	Serializer: 10-bit mode	—	—	—	18 + Δ1	—	UI + ps
Receive Data Latency							
R1[3]	Deserializer: 8-bit mode	—	—	—	10 + Δ2	—	UI + ps
	Deserializer: 10-bit mode	—	—	—	12 + Δ2	—	UI + ps
R2	SERDES Bridge Receive	—	—	—	2	1	word clk
R3	Word Alignment	3.1	—	4	—	0	word clk
R4	8b10b Decoder	—	—	—	1	1	word clk
R5	Clock Tolerance Compensation	7	15	23		1	word clk
R6	FPGA Bridge Receive[2]	1	3	5		1	word clk

1. PCS internal parallel clock. This clock rate is the same as rxfullclk.
2. FPGA Bridge latency varies by the upsample/downsample FIFO read/write. The numbers given are for the 8b10b interface. The depth of the downsample/upsample FIFO is 4. The earliest read can be done after the write clock cycle (one clock) in downsample FIFO. The latest read will be done after the FIFO is full (4 + 1 = 5). For the 16b20b interface, the numbers are doubled: min. = 2, max. = 10. This latency depends on the internal FIFO flag operation.
3. Δ1 = -245ps, Δ2 = 700ps

Figure 3-12. Transmitter and Receiver Block Diagram



SERDES High Speed Data Receiver (LatticeECP2M Family Only)

Table 3-11. Serial Input Data Specifications

Symbol	Description	Min.	Typ.	Max.	Units
$RX\text{-}CID_S$	Stream of nontransitions[1] (CID = Consecutive Identical Digits) @ 10^{-12} BER		7 @ 3.125 Gbps 20 @ 1.25 Gbps		Bits
$V_{RX\text{-}DIFF\text{-}S}$	Differential input sensitivity	100	—	—	mV, p-p
$V_{RX\text{-}IN}$	Input levels	0	—	V_{CCRX} + 0.8	V
$V_{RX\text{-}CM\text{-}DC}$	Input common mode range (DC coupled)	0.5	—	1.2	V
$V_{RX\text{-}CM\text{-}AC}$	Input common mode range (AC coupled)[3]	0	—	1.5	V
$T_{RX\text{-}RELOCK}$	CDR re-lock time[2]	—	—	3000	Bits
$Z_{RX\text{-}TERM}$	Input termination 50/75 Ohm/High Z	—	50		Ohms
$RL_{RX\text{-}RL}$	Return loss (without package)	—	9	—	dB

1. This is the number of bits allowed without a transition on the incoming data stream when using DC coupling.
2. This is the typical number of bit times to re-lock to a new phase of frequency within +/- 300 ppm, assuming 8b10b encoded data and the CDR is in lock state. When CDR is in un-lock state, or reset is applied, the total re-lock settling time will be approximately 4ms including analog settle time, calibration time, and acquisition time.
3. AC coupling is used to interface to LVPECL and LVDS.

Input Data Jitter Tolerance

A receiver's ability to tolerate incoming signal jitter is very dependent on jitter type. High speed serial interface standards have recognized the dependency on jitter type and have recently modified specifications to indicate tolerance levels for different jitter types as they relate to specific protocols (e.g. FC, etc.). Sinusoidal jitter is considered to be a worst case jitter type.

Table 3-12. Receiver Total Jitter Tolerance Specification[1]

Description	Frequency	Condition	Min.	Typ.	Max.	Units
Deterministic	3.125 Gbps	600 mV differential eye	—	—	0.54	UI, p-p
Random		600 mV differential eye	—	—	0.26	UI, p-p
Total		600 mV differential eye	—	—	0.80	UI, p-p
Deterministic	2.5 Gbps	600 mV differential eye	—	—	0.61	UI, p-p
Random		600 mV differential eye	—	—	0.22	UI, p-p
Total		600 mV differential eye	—	—	0.81	UI, p-p
Deterministic	1.25 Gbps	600 mV differential eye	—	—	0.53	UI, p-p
Random		600 mV differential eye	—	—	0.22	UI, p-p
Total		600 mV differential eye	—	—	0.80	UI, p-p
Deterministic	250 Mbps[2]	600 mV differential eye	—	—	0.42	UI, p-p
Random		600 mV differential eye	—	—	0.10	UI, p-p
Total		600 mV differential eye	—	—	0.60	UI, p-p

1. Values are measured with PRBS 2^7-1, all channels operating, FPGA Logic active, I/Os around SERDES pins quiet, voltages are nominal, room temperature.
2. Jitter specification is limited by measurement equipment capability.

Table 3-13. Periodic Receiver Jitter Tolerance Specification[1]

Description	Frequency	Condition	Min.	Typ.	Max.	Units
Periodic	3.125 Gbps	600 mV differential eye	—	—	0.20	UI, p-p
	2.5 Gbps	600 mV differential eye	—	—	0.22	UI, p-p
	1.25 Gbps	600 mV differential eye	—	—	0.20	UI, p-p
	250 Mbps[2]	600 mV differential eye	—	—	0.08	UI, p-p

1. Values are measured with PRBS 2^7-1, all channels operating.
2. Jitter specification is limited by measurement equipment capability.

SERDES External Reference Clock (LatticeECP2M Family Only)

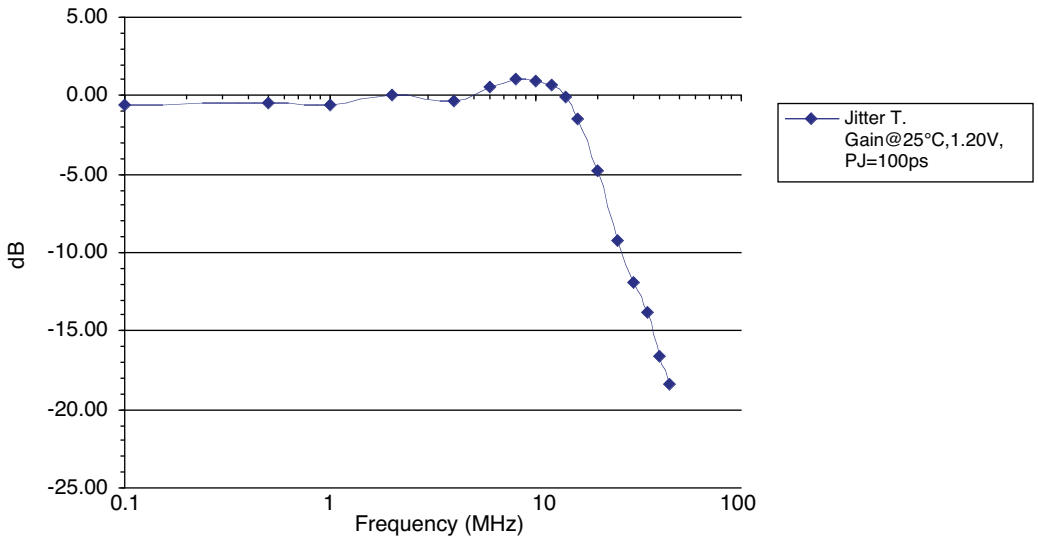
The external reference clock selection and its interface are a critical part of system applications for this product. Table 3-14 specifies reference clock requirements, over the full range of operating conditions.

Table 3-14. External Reference Clock Specification (refclkp/refclkn)

Symbol	Description	Min.	Typ.	Max.	Units
F_{REF}	Frequency range	25	—	320	MHz
$F_{REF-PPM}$	Frequency tolerance	-300	—	300	ppm
$V_{REF-IN-SE}$	Input swing, single-ended clock[1]	100	—	1200	mV, p-p
V_{REF-IN}	Input levels	0	—	V_{CCP} + 0.8	V
$V_{REF-CM-DC}$	Input common mode range (DC coupled)	0.5	—	1.2	V
$V_{REF-CM-AC}$	Input common mode range (AC coupled)[2]	0	—	1.5	V
D_{REF}	Duty cycle[3]	40	—	60	%
T_{REF-R}	Rise time (20% to 80%)		500	1000	ps
T_{REF-F}	Fall time (80% to 20%)		500	1000	ps
$Z_{REF-IN-TERM}$	Input termination		50/2K		Ohms
$C_{REF-IN-CAP}$	Input capacitance[4]	—	—	1.5	pF

1. The signal swing for a single-ended input clock must be as large as the p-p differential swing of a differential input clock to get the same gain at the input receiver. Lower swings for the clock may be possible, but will tend to increase jitter.
2. When AC coupled, the input common mode range is determined by:
(Min input level) + (Peak-to-peak input swing)/2 ≤ (Input common mode voltage) ≤ (Max input level) - (Peak-to-peak input swing)/2
3. Measured at 50% amplitude.
4. Input capacitance of 1.5pF is total capacitance, including both device and package.

Figure 3-13. Jitter Transfer



Note: This graph is for a nominal device.

SERDES Power-Down/Power-Up Specification

Table 3-15. Power-Down and Power-Up Specification

Symbol	Description	Max.	Units
t_{PWRDN}	Power-down time after all power down register bits set to '0'	10	µs
t_{PWRUP}	Power-up time after all power down register bits set to '1'	100	µs

PCI Express Electrical and Timing Characteristics

AC and DC Characteristics

Table 3-16. Transmit[1, 2]

Symbol	Description	Test Conditions	Min	Typ	Max	Units
UI	Unit interval		399.88	400	400.12	ps
$V_{TX-DIFF\_P-P}$	Differential peak-to-peak output voltage		0.8	1.0	1.2	V
$V_{TX-DE-RATIO}$	De-emphasis differential output voltage ratio		0	-3.5	-7.96	dB
$V_{TX-CM-AC\_P}$	RMS AC peak common-mode output voltage		—	20	—	mV
$V_{TX-CM-DC-LINE-DELTA}$	Maximum Common mode voltage delta between n and p channels		—	—	25	mV
$V_{TX-DC-CM}$	Tx DC common mode voltage		0	—	V_{CCOB} + 5%	V
$I_{TX-SHORT}$	Output short circuit current	V_{TX-D+}=0.0V V_{TX-D-}=0.0V	—	—	90	mA
$Z_{TX-DIFF-DC}$	Differential output impedance		80	100	120	Ohms
$T_{TX-RISE}$	Tx output rise time	20 to 80%	0.125	—	—	UI
$T_{TX-FALL}$	Tx output fall time	20 to 80%	0.125	—	—	UI
$L_{TX-SKEW}$	Lane-to-lane static output skew for all lanes in port/link		—	—	1.3	ns
T_{TX-EYE}	Transmitter eye width		0.75	—	—	UI
$T_{TX-EYE-MEDIAN-TO-MAX-JITTER}$[3]			—	—	0.125	UI
C_{TX}	AC coupling capacitor		75	—	200	nF

1. Values are measured at 2.5 Gbps.
2. Compliant to PCI Express v1.1.
3. Measured at 60ps with plug-in board and jitter due to socket removed.

Table 3-17. Receive

Symbol	Description	Test Conditions	Min.	Typ.	Max.	Units
UI	Unit Interval		399.88	400	400.12	ps
$V_{RX-DIFF\_P-P}$	Differential peak-to-peak input voltage		0.175	—	—	V
$V_{RX-IDLE-DET-DIFF\_P-P}$	Idle detect threshold voltage		65	—	175	mV
$Z_{RX-DIFF-DC}$	DC differential input impedance		80	100	120	Ohms
Z_{RX-DC}	DC input impedance		40	50	60	Ohms
$Z_{RX-HIGH-IMP-DC}$[1]	Power-down DC input impedance		200K	—	—	Ohms
T_{RX-EYE}	Receiver eye width		0.4	—	—	UI
$T_{RX-EYE-MEDIAN-TO-MAX-JITTER}$			—	—	0.3	UI

Notes:
1. Measured with external AC-coupling on the receiver
2. Values are measured at 2.5 Gbps

Table 3-18. Reference Clock

Symbol	Description	Test Conditions	Min.	Typ.	Max.	Units
F_{REFCLK}	Reference clock frequency		—	100	—	MHz
V_{CM}	Input common mode voltage		—	0.65	—	V
T_R/T_F	Clock input rise/fall time		—	—	1.0	ns
V_{SW}	Differential input voltage swing		0.6	—	1.6	V
DC_{REFCLK}	Input clock duty cycle		40	50	60	%
PPM	Reference clock tolerance		-300	—	+300	ppm

LatticeECP2/M sysCONFIG Port Timing Specifications

Over Recommended Operating Conditions

Parameter	Description	Min.	Max.	Units
sysCONFIG Byte Data Flow				
t_{SUCBDI}	Byte D[0:7] Setup Time to CCLK	7	—	ns
t_{HCBDI}	Byte D[0:7] Hold Time to CCLK	1	—	ns
t_{CODO}	CCLK to DOUT in Flowthrough Mode	—	12	ns
t_{SUCS}	CSN[0:1] Setup Time to CCLK	7	—	ns
t_{HCS}	CSN[0:1] Hold Time to CCLK	1	—	ns
t_{SUWD}	Write Signal Setup Time to CCLK	7	—	ns
t_{HWD}	Write Signal Hold Time to CCLK	1	—	ns
t_{DCB}	CCLK to BUSY Delay Time	—	12	ns
t_{CORD}	CCLK to Out for Read Data	—	12	ns
sysCONFIG Byte Slave Clocking				
t_{BSCH}	Byte Slave CCLK Minimum High Pulse	6	—	ns
t_{BSCL}	Byte Slave CCLK Minimum Low Pulse	9	—	ns
t_{BSCYC}	Byte Slave CCLK Cycle Time	15	—	ns
sysCONFIG Serial (Bit) Data Flow				
t_{SUSCDI}	DI Setup Time to CCLK Slave Mode	7	—	ns
t_{HSCDI}	DI Hold Time to CCLK Slave Mode	1	—	ns
t_{CODO}	CCLK to DOUT in Flowthrough Mode	—	12	ns
sysCONFIG Serial Slave Clocking				
t_{SSCH}	Serial Slave CCLK Minimum High Pulse	6	—	ns
t_{SSCL}	Serial Slave CCLK Minimum Low Pulse	6	—	ns
sysCONFIG POR, Initialization and Wake-up				
t_{ICFG}	Minimum Vcc to INITN High	—	28	ms
t_{VMC}	Time from t_{ICFG} to Valid Master CCLK	—	2	us
t_{PRGMRJ}	PROGRAMN Pin Pulse Rejection	—	8	ns
t_{PRGM}	PROGRAMN Low Time to Start Configuration	25	—	ns
t_{DINIT}	PROGRAMN High to INITN High Delay[1]	—	1.5	ms
$t_{DPPINIT}$	Delay Time from PROGRAMN Low to INITN Low	—	37	ns
$t_{DPPDONE}$	Delay Time from PROGRAMN Low to DONE Low	—	37	ns
t_{IODISS}	User I/O Disable from PROGRAMN Low	—	35	ns
t_{IOENSS}	User I/O Enabled Time from CCLK Edge During Wake-up Sequence	—	25	ns
t_{MWC}	Additional Wake Master Clock Signals after DONE Pin High	120	—	cycles
sysCONFIG SPI Port[2]				
t_{CFGX}	INITN High to CCLK Low	—	1	µs
t_{CSSPI}	INITN High to CSSPIN Low	—	2	us
t_{CSCCLK}	CCLK Low before CSSPIN Low	0	—	ns
t_{SOCDO}	CCLK Low to Output Valid	—	15	ns
t_{SOE}	CSSPIN[0:1] Active Setup Time	300	—	ns
t_{CSPID}	CSSPIN[0:1] Low to First CCLK Edge Setup Time	300+3cyc	600+6cyc	ns
f_{MAXSPI}	Max. CCLK Frequency - SPI Flash Read Opcode (0x03) (SPIFASTN = 1)	—	20	MHz
	Max. CCLK Frequency - SPI Flash Fast Read Opcode (0x0B) (SPIFASTN = 0)	—	50	MHz

LatticeECP2/M sysCONFIG Port Timing Specifications (Continued)

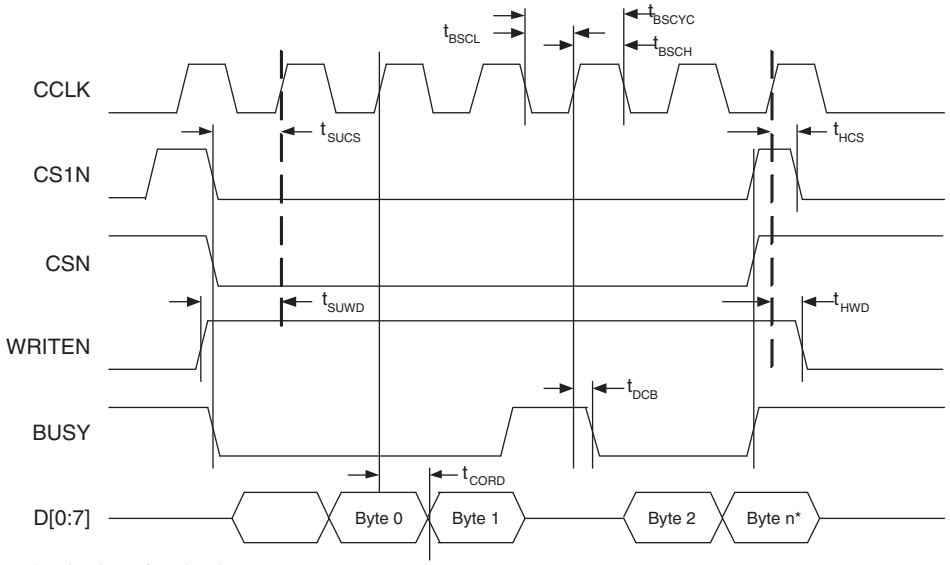
Over Recommended Operating Conditions

Parameter	Description	Min.	Max.	Units
t_{SUSPI}	SOSPI Data Setup Time Before CCLK	7	—	ns
t_{HSPI}	SOSPI Data Hold Time After CCLK	2	—	ns

1. Re-toggling the PROGRAMN pin is not permitted until the INITN pin is high. Avoid consecutive toggling of the PROGRAMN.
2. For SED (Soft Error Detect), the SEDCLKIN operating frequency must be at least 20MHz. SEDCLKIN is derived from Master Clock Frequency that has a +/-30% variation..

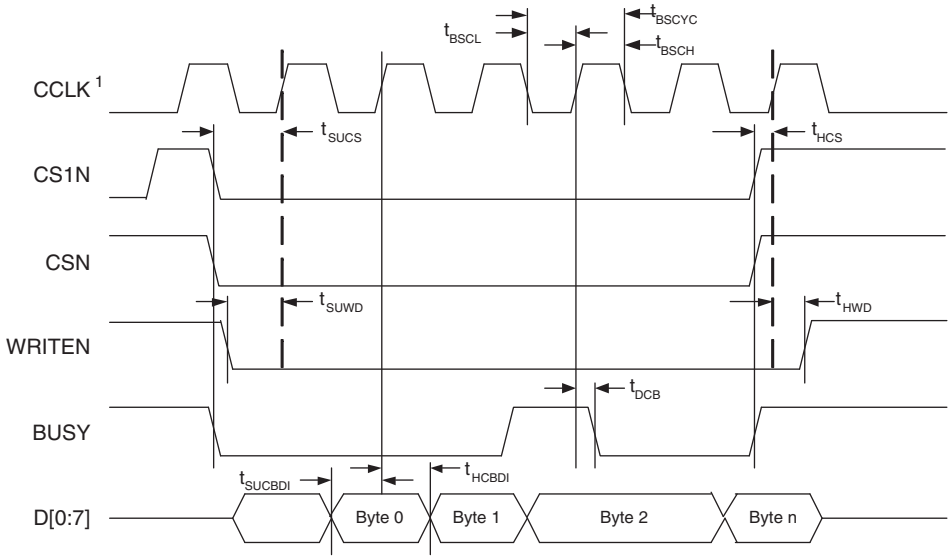
Parameter	Min.	Max.	Units
Master Clock Frequency	Selected value - 30%	Selected value + 30%	MHz
Duty Cycle	40	60	%

Figure 3-14. sysCONFIG Parallel Port Read Cycle



*n = last byte of read cycle.

Figure 3-15. sysCONFIG Parallel Port Write Cycle



1. In Master Parallel Mode the FPGA provides CCLK. In Slave Parallel Mode the external device provides CCLK.

Figure 3-16. sysCONFIG Slave Serial Port Timing

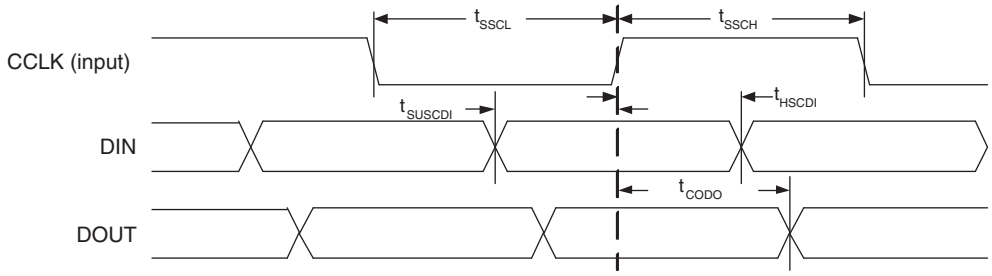
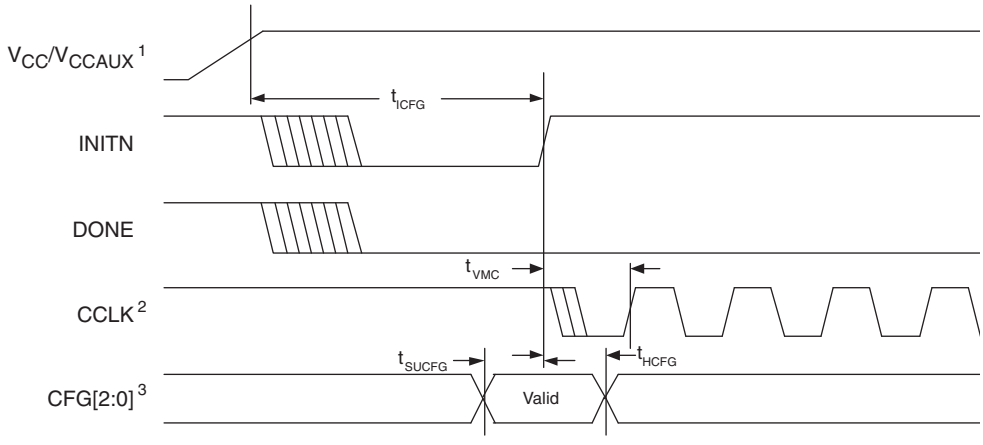
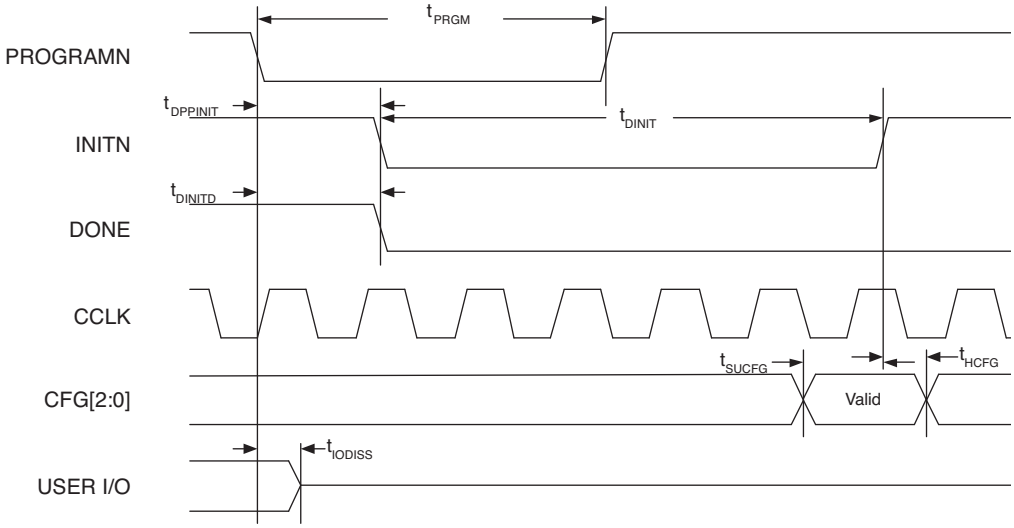


Figure 3-17. Power-On-Reset (POR) Timing



1. Time taken from V_{CC} or V_{CCAUX}, whichever is the last to reach its V_{MIN}.
2. Device is in a Master Mode.
3. The CFG pins are normally static (hard wired).

Figure 3-18. Configuration from PROGRAMN Timing



1. The CFG pins are normally static (hard wired)

Figure 3-19. Wake-Up Timing

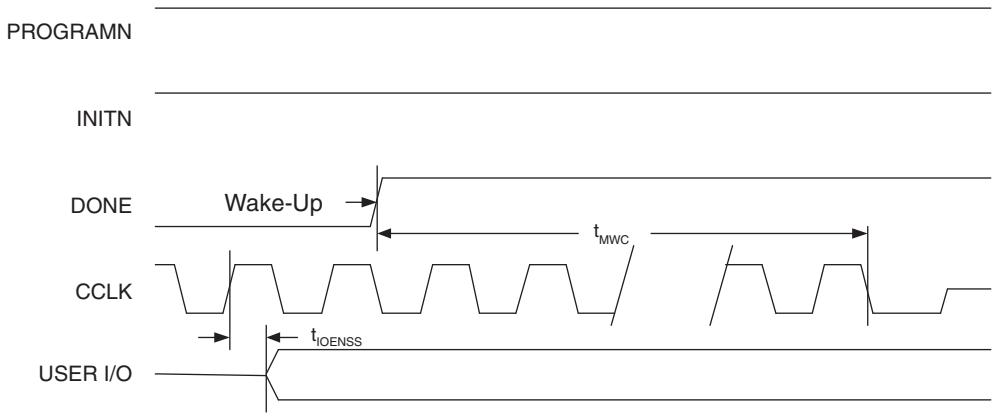
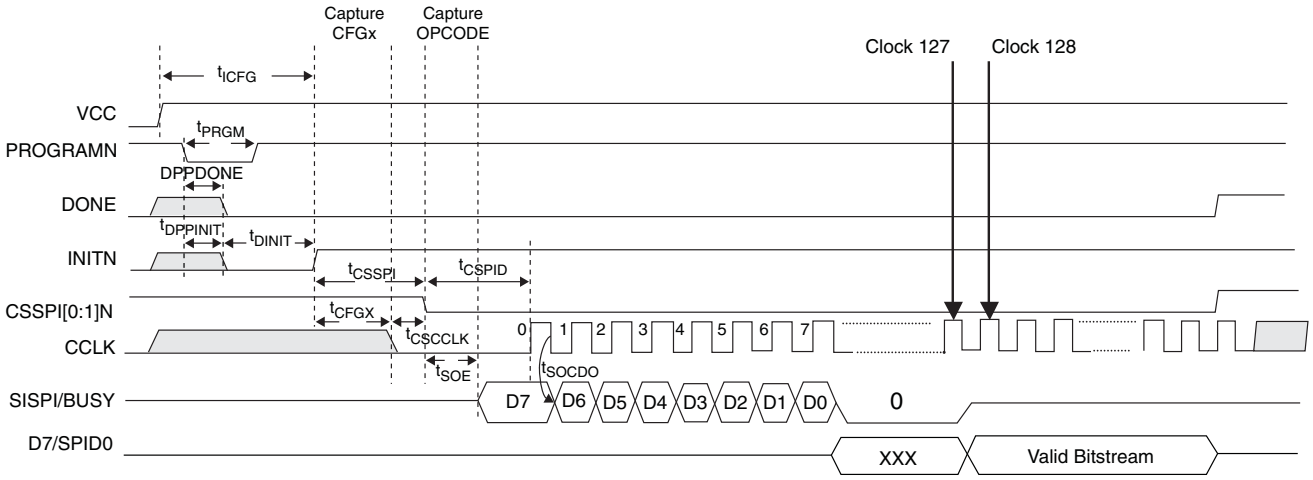


Figure 3-20. SPI/SPIm Configuration Waveforms



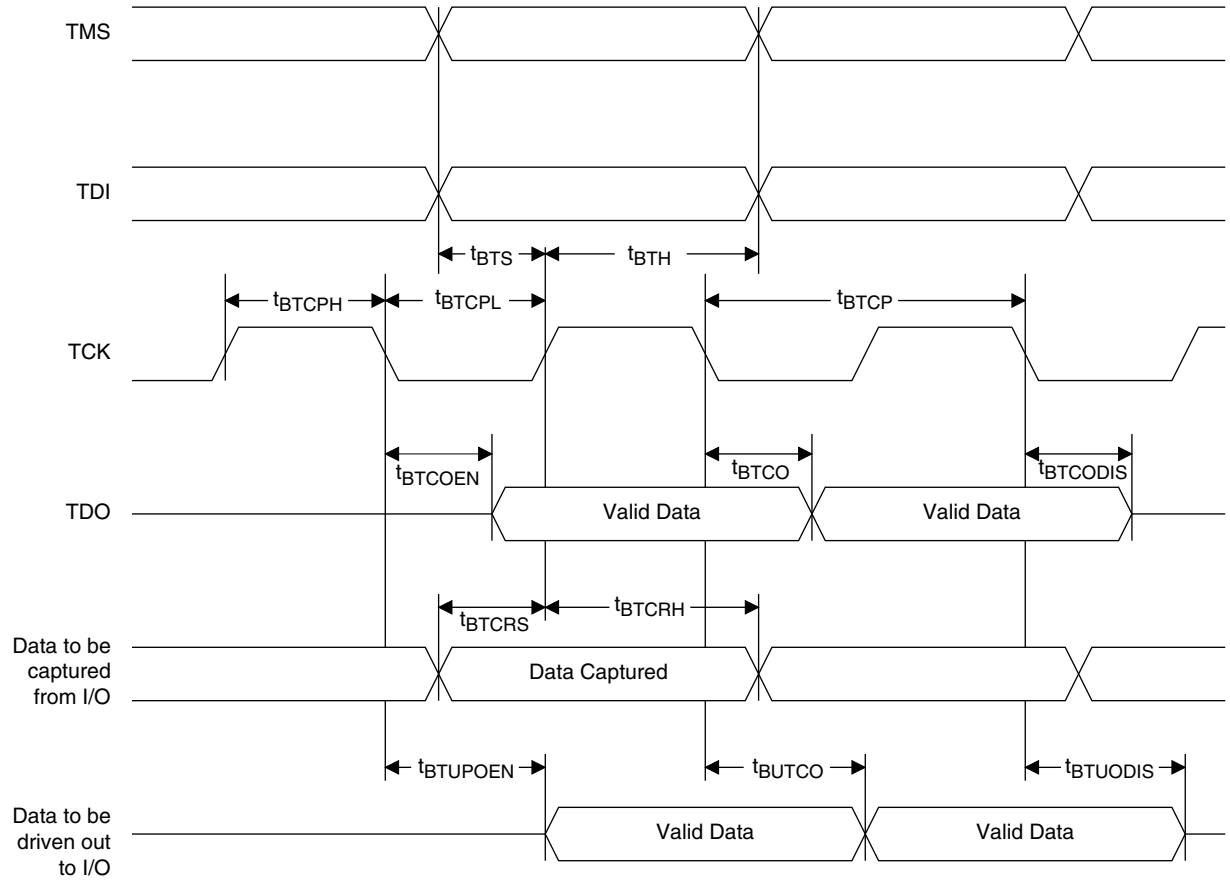
JTAG Port Timing Specifications

Over Recommended Operating Conditions

Symbol	Parameter	Min	Max	Units
f_{MAX}	TCK clock frequency	—	25	MHz
t_{BTCP}	TCK [BSCAN] clock pulse width	40	—	ns
t_{BTCPH}	TCK [BSCAN] clock pulse width high	20	—	ns
t_{BTCPL}	TCK [BSCAN] clock pulse width low	20	—	ns
t_{BTS}	TCK [BSCAN] setup time	8	—	ns
t_{BTH}	TCK [BSCAN] hold time	10	—	ns
t_{BTRF}	TCK [BSCAN] rise/fall time	50	—	mV/ns
t_{BTCO}	TAP controller falling edge of clock to valid output	—	10	ns
$t_{BTCODIS}$	TAP controller falling edge of clock to valid disable	—	10	ns
t_{BTCOEN}	TAP controller falling edge of clock to valid enable	—	10	ns
t_{BTCRS}	BSCAN test capture register setup time	8	—	ns
t_{BTCRH}	BSCAN test capture register hold time	25	—	ns
t_{BUTCO}	BSCAN test update register, falling edge of clock to valid output	—	25	ns
$t_{BTUODIS}$	BSCAN test update register, falling edge of clock to valid disable	—	25	ns
$t_{BTUPOEN}$	BSCAN test update register, falling edge of clock to valid enable	—	25	ns

Timing v.A 0.11

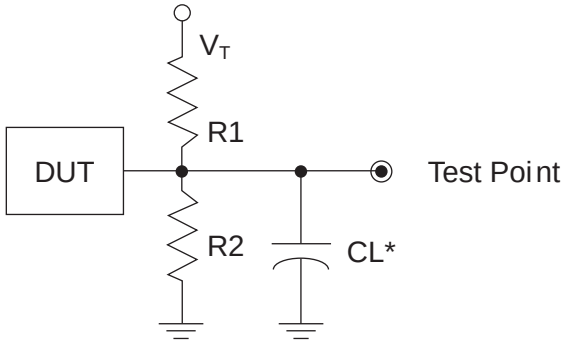
Figure 3-21. JTAG Port Timing Waveforms



Switching Test Conditions

Figure 3-22 shows the output test load that is used for AC testing. The specific values for resistance, capacitance, voltage, and other test conditions are shown in Table 3-19.

Figure 3-22. Output Test Load, LVTTL and LVCMOS Standards



*CL Includes Test Fixture and Probe Capacitance

Table 3-19. Test Fixture Required Components, Non-Terminated Interfaces

Test Condition	R_1	R_2	C_L	Timing Ref.	V_T
LVTTL and other LVCMOS settings (L -> H, H -> L)	∞	∞	0pF	LVCMOS 3.3 = 1.5V	—
				LVCMOS 2.5 = $V_{CCIO}/2$	—
				LVCMOS 1.8 = $V_{CCIO}/2$	—
				LVCMOS 1.5 = $V_{CCIO}/2$	—
				LVCMOS 1.2 = $V_{CCIO}/2$	—
LVCMOS 2.5 I/O (Z -> H)	∞	1MΩ		$V_{CCIO}/2$	—
LVCMOS 2.5 I/O (Z -> L)	1MΩ	∞		$V_{CCIO}/2$	V_{CCIO}
LVCMOS 2.5 I/O (H -> Z)	∞	100		V_{OH} - 0.10	—
LVCMOS 2.5 I/O (L -> Z)	100	∞		V_{OL} + 0.10	V_{CCIO}

Note: Output test conditions for all other interfaces are determined by the respective standards.

Signal Descriptions

Signal Name	I/O	Description
General Purpose		
P[Edge] [Row/Column Number*]_[A/B]	I/O	[Edge] indicates the edge of the device on which the pad is located. Valid edge designations are L (Left), B (Bottom), R (Right), T (Top). [Row/Column Number] indicates the PFU row or the column of the device on which the PIC exists. When Edge is T (Top) or B (Bottom), only need to specify Row Number. When Edge is L (Left) or R (Right), only need to specify Column Number. [A/B] indicates the PIO within the PIC to which the pad is connected. Some of these user-programmable pins are shared with special function pins. These pins, when not used as special purpose pins, can be programmed as I/Os for user logic. During configuration the user-programmable I/Os are tri-stated with an internal pull-up resistor enabled. If any pin is not used (or not bonded to a package pin), it is also tri-stated with an internal pull-up resistor enabled after configuration. See "Typical sysI/O I/O Behavior During Power-up" for more information about I/O behavior during power-up.
GSRN	I	Global RESET signal (active low). Any I/O pin can be GSRN.
NC	—	No connect.
GND	—	Ground. Dedicated pins.
V_{CC}	—	Power supply pins for core logic. Dedicated pins.
V_{CCAUX}	—	Auxiliary power supply pin. This dedicated pin powers all the differential and referenced input buffers.
V_{CCIOx}	—	Dedicated power supply pins for I/O bank x.
V_{CCPLL}	—	PLL supply pins. Should be tied to V_{CC} even when the corresponding PLL is unused.
V_{REF1_x}, V_{REF2_x}	—	Reference supply pins for I/O bank x. Pre-determined pins in each bank are assigned as V_{REF} inputs. When not used, they may be used as I/O pins.
XRES[4]	—	10K ohm +/-1% resistor must be connected between this pad and ground.
PLLCAP[4]	—	External capacitor connection for PLL.
PLL, DLL and Clock Functions (Used as user programmable I/O pins when not in use for PLL or clock pins)		
[LOC][num]_V_{CCPLL}	—	Power supply pin for PLL: LUM, LLM, RUM, RLM, num = row from center.
[LOC][num]_GPLL[T, C]_IN_A	I	General Purpose PLL (GPLL) input pads: LUM, LLM, RUM, RLM, num = row from center, T = true and C = complement, index A,B,C...at each side.
[LOC][num]_GPLL[T, C]_FB_A	I	Optional feedback GPLL input pads: LUM, LLM, RUM, RLM, num = row from center, T = true and C = complement, index A,B,C...at each side.
[LOC][num]_SPLL[T, C]_IN_A[5]	I	Secondary PLL (SPLL) input pads: LUM, LLM, RUM, RLM, num = row from center, T = true and C = complement, index A,B,C...at each side.
[LOC][num]_SPLL[T, C]_FB_A[5]	I	Optional feedback (SPLL) input pads: LUM, LLM, RUM, RLM, num = row from center, T = true and C = complement, index A,B,C...at each side.
[LOC][num]_DLL[T, C]_IN_A	I	DLL input pads: LUM, LLM, RUM, RLM, num = row from center, T = true and C = complement, index A,B,C...at each side.
[LOC][num]_DLL[T, C]_FB_A	I	Optional feedback (DLL) input pads: LUM, LLM, RUM, RLM, num = row from center, T = true and C = complement, index A,B,C...at each side.
PCLK[T, C]_[n:0]_[3:0]	I	Primary Clock pads, T = true and C = complement, n per side, indexed by bank and 0,1,2,3 within bank.

Signal Descriptions (Cont.)

Signal Name	I/O	Description
[LOC]DQS[num]	I/O	DQ input/output pads: T (top), R (right), B (bottom), L (left), DQS, num = ball function number.
[LOC]DQ[num]	I/O	DQ input/output pads: T (top), R (right), B (bottom), L (left), DQ, associated DQS number.
Test and Programming (Dedicated Pins)		
TMS	I	Test Mode Select input, used to control the 1149.1 state machine. Pull-up is enabled during configuration.
TCK	I	Test Clock input pin, used to clock the 1149.1 state machine. No pull-up enabled.
TDI	I	Test Data in pin. Used to load data into device using 1149.1 state machine. After power-up, this TAP port can be activated for configuration by sending appropriate command. (Note: once a configuration port is selected it is locked. Another configuration port cannot be selected until the power-up sequence). Pull-up is enabled during configuration.
TDO	O	Output pin. Test Data Out pin used to shift data out of a device using 1149.1.
VCCJ	—	Power supply pin for JTAG Test Access Port.
Configuration Pads (Used During sysCONFIG)		
CFG[2:0]	I	Mode pins used to specify configuration mode values latched on rising edge of INITN. During configuration, a pull-up is enabled. These are dedicated pins.
INITN	I/O	Open Drain pin. Indicates the FPGA is ready to be configured. During configuration, a pull-up is enabled. It is a dedicated pin.
PROGRAMN	I	Initiates configuration sequence when asserted low. This pin always has an active pull-up. This is a dedicated pin.
DONE	I/O	Open Drain pin. Indicates that the configuration sequence is complete, and the startup sequence is in progress. This is a dedicated pin.
CCLK	I/O	Configuration Clock for configuring an FPGA in sysCONFIG mode.
BUSY/SISPI	I/O	Read control command in SPI or SPIm mode.
CSN	I	sysCONFIG chip select (active low). During configuration, a pull-up is enabled.
CS1N	I	sysCONFIG chip select (active low). During configuration, a pull-up is enabled.
WRITEN	I	Write Data on Parallel port (active low).
D[0]/SPIFASTN	I/O	sysCONFIG Port Data I/O for Parallel mode. sysCONFIG Port Data I/O for SPI or SPIm. When using the SPI or SPIm mode, this pin should either be tied high or low, must not be left floating.
D[1:6]	I/O	sysCONFIG Port Data I/O for Parallel
D[7]/SPID0	I/O	sysCONFIG Port Data I/O for Parallel, SPI, SPIm
DOUT/CSON	O	Output for serial configuration data (rising edge of CCLK) when using sysCONFIG port.
DI/CSSPI0N	I/O	Input for serial configuration data (clocked with CCLK) when using sysCONFIG port. During configuration, a pull-up is enabled. Output when used in SPI/SPIm modes.
Dedicated SERDES Signals[1, 2, 3]		
[LOC]_SQ_VCCAUX33	—	Termination resistor switching power (3.3V). This pin must be tied to 3.3V even if the quad is unused.
[LOC]_SQ_REFCLKN	I	Negative Reference Clock Input
[LOC]_SQ_REFCLKP	I	Positive Reference Clock Input
[LOC]_SQ_VCCP	—	PLL and Reference clock buffer power (1.2V). This pin must be tied to 1.2V even if the quad is unused.

Signal Descriptions (Cont.)

Signal Name	I/O	Description
[LOC]_SQ_VCCIBm	—	Input buffer power supply, channel m (1.2V/1.5V). This pin should be left floating if the channel is unused.
[LOC]_SQ_VCCOBm	—	Output buffer power supply, channel m (1.2V/1.5V). This pin should be left floating if the channel is unused.
[LOC]_SQ_HDOUTNm	O	High-speed output, negative channel m
[LOC]_SQ_HDOUTPm	O	High-speed output, positive channel m
[LOC]_SQ_HDINNm	I	High-speed input, negative channel m
[LOC]_SQ_HDINPm	I	High-speed input, positive channel m
[LOC]_SQ_VCCTXm[4]	—	Transmitter power supply, channel m (1.2V). This pin must be tied to 1.2V even if the channel is unused.
[LOC]_SQ_VCCRXm[4]	—	Receiver power supply, channel m (1.2V). This pin must be tied to 1.2V even if the channel is unused.

1. These signals are relevant for LatticeECP2M family.
2. m defines the associated channel in the Quad.
3. These signals are defined in Quads [LOC] indicates the corner SERDES Quad is located: ULC (upper left), URC (upper right), LLC (lower left), LRC (lower right).
4. When placing switching I/Os around these critical pins that are designed to supply the device with the proper reference or supply voltage, care must be given. For more information, refer to TN1159, LatticeECP2/M Pin Assignment Recommendations.
5. There may be SPLLs that do not have dedicated I/Os.

PICs and DDR Data (DQ) Pins Associated with the DDR Strobe (DQS) Pin

PICs Associated with DQS Strobe	PIO Within PIC	DDR Strobe (DQS) and Data (DQ) Pins
For Left and Right Edges of the Device		
P[Edge] [n-4]	A	DQ
	B	DQ
P[Edge] [n-3]	A	DQ
	B	DQ
P[Edge] [n-2]	A	DQ
	B	DQ
P[Edge] [n-1]	A	DQ
	B	DQ
P[Edge] [n]	A	[Edge]DQSn
	B	DQ
P[Edge] [n+1]	A	DQ
	B	DQ
P[Edge] [n+2]	A	DQ
	B	DQ
P[Edge] [n+3]	A	DQ
	B	DQ
For Bottom Edge of the Device		
P[Edge] [n-4]	A	DQ
	B	DQ
P[Edge] [n-3]	A	DQ
	B	DQ
P[Edge] [n-2]	A	DQ
	B	DQ
P[Edge] [n-1]	A	DQ
	B	DQ
P[Edge] [n]	A	[Edge]DQSn
	B	DQ
P[Edge] [n+1]	A	DQ
	B	DQ
P[Edge] [n+2]	A	DQ
	B	DQ
P[Edge] [n+3]	A	DQ
	B	DQ
P[Edge] [n+4]	A	DQ
	B	DQ

Notes:
1. “n” is a row PIC number.
2. The DDR interface is designed for memories that support one DQS strobe up to 15 bits of data for the left and right edges and up to 17 bits of data for the bottom edge. In some packages, all the potential DDR data (DQ) pins may not be available. PIC numbering definitions are provided in the “Signal Names” column of the Signal Descriptions table.

LatticeECP2 Pin Information Summary, LFE2-6 and LFE2-12

Pin Type		LFE2-6		LFE2-12			
		144 TQFP	256 fpBGA	144 TQFP	208 PQFP	256 fpBGA	484 fpBGA
Single Ended User I/O		90	190	93	131	193	297
Differential Pair User I/O		43	95	45	62	96	148
Configuration	TAP Pins	5	5	5	5	5	5
	Muxed Pins	14	14	14	14	14	14
	Dedicated Pins (Non TAP)	7	7	7	7	7	7
Non Configuration	Muxed Pins	34	54	33	40	54	57
	Dedicated Pins	3	3	3	3	3	3
VCC		10	7	10	14	7	16
VCCAUX		4	4	4	8	4	16
VCCPLL		0	0	0	0	0	0
VCCIO	Bank0	1	2	1	2	2	4
	Bank1	1	2	1	2	2	4
	Bank2	1	2	1	2	2	4
	Bank3	1	2	1	2	2	4
	Bank4	1	2	1	2	2	4
	Bank5	1	2	1	2	2	4
	Bank6	1	2	1	2	2	4
	Bank7	1	2	1	2	2	4
	Bank8	1	1	1	2	1	2
GND, GND0 to GND7		12	20	12	22	20	60
NC		4	3	1	0	0	44
Single Ended/ Differential I/O Pairs per Bank (including emulated with resistors)	Bank0	8/4	18/6	8/4	18/9	18/9	50/25
	Bank1	17/8	34/17	18/9	18/9	34/17	46/23
	Bank2	4/2	20/10	4/2	11/5	20/10	24/12
	Bank3	8/4	12/6	8/4	11/5	12/6	16/8
	Bank4	18/9	32/16	18/9	19/9	32/16	46/23
	Bank5	8/4	14/7	10/5	18/9	17/8	46/23
	Bank6	9/4	26/13	9/4	18/8	26/13	32/16
	Bank7	12/6	20/10	12/6	12/6	20/10	23/11
	Bank8	6/2	14/7	6/2	6/2	14/7	14/7
True LVDS I/O Pairs per Bank	Bank0 (Top Edge)	0	0	0	0	0	0
	Bank1 (Top Edge)	0	0	0	0	0	0
	Bank2 (Right Edge)	1	5	1	4	5	6
	Bank3 (Right Edge)	3	3	3	3	3	4
	Bank4 (Bottom Edge)	0	0	0	0	0	0
	Bank5 (Bottom Edge)	0	0	0	0	0	0
	Bank6 (Left Edge)	2	7	2	6	7	8
	Bank7 (Left Edge)	5	5	5	5	5	5
	Bank8 (Right Edge)	0	0	0	0	0	0

LatticeECP2 Pin Information Summary, LFE2-6 and LFE2-12 (Cont.)

Pin Type		LFE2-6		LFE2-12			
		144 TQFP	256 fpBGA	144 TQFP	208 PQFP	256 fpBGA	484 fpBGA
Available DDR-Interfaces per I/O Bank[1]	Bank0	0	0	0	0	0	0
	Bank1	0	0	0	0	0	0
	Bank2	0	1	0	0	1	1
	Bank3	0	0	0	0	0	0
	Bank4	0	2	0	0	2	3
	Bank5	0	1	0	0	1	3
	Bank6	0	1	0	0	1	1
	Bank7	0	1	0	0	1	1
	Bank8	0	0	0	0	0	0
PCI Capable I/Os per Bank	Bank0	0	0	0	0	0	0
	Bank1	0	0	0	0	0	0
	Bank2	0	0	0	0	0	0
	Bank3	0	0	0	0	0	0
	Bank4	18	32	18	19	32	46
	Bank5	8	14	10	18	17	46
	Bank6	0	0	0	0	0	0
	Bank7	0	0	0	0	0	0
	Bank8	0	0	0	0	0	0

1. Minimum requirement to implement a fully functional 8-bit wide DDR bus. Available DDR interface consists of at least 12 I/Os (1 DQS + 1 DQSB + 8 DQs + 1 DM + Bank VREF1).

LatticeECP2 Pin Information Summary, LFE2-20 and LFE2-35

Pin Type		LFE2-20				LFE2-35	
		208 PQFP	256 fpBGA	484 fpBGA	672 fpBGA	484 fpBGA	672 fpBGA
Single Ended User I/O		131	193	331	402	331	450
Differential Pair User I/O		62	96	165	200	165	224
Configuration	TAP Pins	5	5	5	5	5	5
	Muxed Pins	14	14	14	14	14	14
	Dedicated Pins (Non TAP)	7	7	7	7	7	7
Non Configuration	Muxed Pins	42	54	60	64	60	68
	Dedicated Pins	3	3	3	3	3	3
VCC		14	7	18	24	16	22
VCCAUX		8	4	16	16	16	16
VCCPLL		0	0	0	0	2	2
VCCIO	Bank0	2	2	4	5	4	5
	Bank1	2	2	4	5	4	5
	Bank2	2	2	4	5	4	5
	Bank3	2	2	4	5	4	5
	Bank4	2	2	4	5	4	5
	Bank5	2	2	4	5	4	5
	Bank6	2	2	4	5	4	5
	Bank7	2	2	4	5	4	5
	Bank8	2	1	2	2	2	2
GND, GND0 to GND7		22	20	60	72	60	72
NC		0	1	8	101	8	102
Single Ended/ Differential I/O Pairs per Bank (including emulated with resistors)	Bank0	18/9	18/9	50/25	67/33	50/25	67/33
	Bank1	18/9	34/17	46/23	52/26	46/23	52/26
	Bank2	11/5	20/10	34/17	36/18	34/17	48/24
	Bank3	11/5	12/6	22/11	32/16	22/11	42/21
	Bank4	19/9	32/16	46/23	50/25	46/23	54/27
	Bank5	18/9	17/8	46/23	68/34	46/23	68/34
	Bank6	18/8	26/13	40/20	48/24	40/20	58/29
	Bank7	12/6	20/10	33/16	35/17	33/16	47/23
	Bank8	6/2	14/7	14/7	14/7	14/7	14/7
True LVDS I/O Pairs per Bank	Bank0 (Top Edge)	0	0	0	0	0	0
	Bank1 (Top Edge)	0	0	0	0	0	0
	Bank2 (Right Edge)	4	5	9	9	9	12
	Bank3 (Right Edge)	3	3	5	8	5	9
	Bank4 (Bottom Edge)	0	0	0	0	0	0
	Bank5 (Bottom Edge)	0	0	0	0	0	0
	Bank6 (Left Edge)	6	7	10	12	10	13
	Bank7 (Left Edge)	5	5	8	8	8	11
	Bank8 (Right Edge)	0	0	0	0	0	0

LatticeECP2 Pin Information Summary, LFE2-20 and LFE2-35 (Cont.)

Pin Type		LFE2-20				LFE2-35	
		208 PQFP	256 fpBGA	484 fpBGA	672 fpBGA	484 fpBGA	672 fpBGA
Available DDR-Interfaces per I/O Bank[1]	Bank0	0	0	0	0	0	0
	Bank1	0	0	0	0	0	0
	Bank2	0	1	2	2	2	3
	Bank3	0	0	0	2	0	2
	Bank4	0	2	3	3	3	3
	Bank5	0	1	3	4	3	4
	Bank6	0	1	2	3	1	3
	Bank7	0	1	2	2	2	3
	Bank8	0	0	0	0	0	0
PCI Capable I/Os per Bank	Bank0	0	0	0	0	0	0
	Bank1	0	0	0	0	0	0
	Bank2	0	0	0	0	0	0
	Bank3	0	0	0	0	0	0
	Bank4	19	32	46	50	46	54
	Bank5	18	17	46	68	46	68
	Bank6	0	0	0	0	0	0
	Bank7	0	0	0	0	0	0
	Bank8	0	0	0	0	0	0

1. Minimum requirement to implement a fully functional 8-bit wide DDR bus. Available DDR interface consists of at least 12 I/Os (1 DQS + 1 DQSB + 8 DQs + 1 DM + Bank VREF1).

LatticeECP2 Pin Information Summary, LFE2-50 and LFE2-70

Pin Type		LFE2-50		LFE2-70	
		484 fpBGA	672 fpBGA	672 fpBGA	900 fpBGA
Single Ended User I/O		339	500	500	583
Differential Pair User I/O		169	249	249	290
Configuration	TAP Pins	5	5	5	5
	Muxed Pins	14	14	14	14
	Dedicated Pins (Non TAP)	7	7	7	7
Non Configuration	Muxed Pins	68	79	79	89
	Dedicated Pins	3	3	3	3
VCC		16	20	20	26
VCCAUX		16	16	16	17
VCCPLL		4	4	2	4
VCCIO	Bank0	4	5	5	6
	Bank1	4	5	5	6
	Bank2	4	5	5	6
	Bank3	4	5	5	6
	Bank4	4	5	5	6
	Bank5	4	5	5	6
	Bank6	4	5	5	6
	Bank7	4	5	5	6
	Bank8	2	2	2	2
GND, GND0 to GND7		60	72	72	104
NC		0	3	5	101
Single Ended/ Differential I/O Pairs per Bank (including emulated with resistors)	Bank0	50/25	67/33	67/33	84/42
	Bank1	46/23	66/33	66/33	76/38
	Bank2	38/19	56/28	56/28	74/37
	Bank3	22/11	48/24	48/24	48/24
	Bank4	46/23	62/31	62/31	72/35
	Bank5	46/23	68/34	68/34	80/40
	Bank6	40/20	64/32	64/32	64/32
	Bank7	37/18	55/27	55/27	71/35
	Bank8	14/7	14/7	14/7	14/7
True LVDS I/O Pairs per Bank	Bank0 (Top Edge)	0	0	0	0
	Bank1 (Top Edge)	0	0	0	0
	Bank2 (Right Edge)	9	13	13	18
	Bank3 (Right Edge)	5	12	12	12
	Bank4 (Bottom Edge)	0	0	0	0
	Bank5 (Bottom Edge)	0	0	0	0
	Bank6 (Left Edge)	10	16	16	16
	Bank7 (Left Edge)	8	12	12	16
	Bank8 (Right Edge)	0	0	0	0

LatticeECP2 Pin Information Summary, LFE2-50 and LFE2-70 (Cont.)

Pin Type		LFE2-50		LFE2-70	
		484 fpBGA	672 fpBGA	672 fpBGA	900 fpBGA
Available DDR-Interfaces per I/O Bank[1]	Bank0	0	0	0	0
	Bank1	0	0	0	0
	Bank2	2	3	3	4
	Bank3	0	3	3	3
	Bank4	3	4	4	4
	Bank5	3	4	4	5
	Bank6	1	4	4	4
	Bank7	2	3	3	4
	Bank8	0	0	0	0
PCI Capable I/Os per Bank	Bank0	0	0	0	0
	Bank1	0	0	0	0
	Bank2	0	0	0	0
	Bank3	0	0	0	0
	Bank4	46	62	62	72
	Bank5	46	68	68	80
	Bank6	0	0	0	0
	Bank7	0	0	0	0
	Bank8	0	0	0	0

1. Minimum requirement to implement a fully functional 8-bit wide DDR bus. Available DDR interface consists of at least 12 I/Os (1 DQS + 1 DQSB + 8 DQs + 1 DM + Bank VREF1).

LatticeECP2M Pin Information Summary, LFE2M20 and LFE2M35

Pin Type		LFE2M20		LFE2M35		
		256 fpBGA	484 fpBGA	256 fpBGA	484 fpBGA	672 fpBGA
Single Ended User I/O		140	304	140	303	410
Differential Pair User I/O		70	152	70	151	199
Configuration	TAP Pins	5	5	5	5	5
	Muxed Pins	14	14	14	14	14
	Dedicated Pins (Non TAP)	7	7	7	7	7
Non Configuration	Muxed Pins	64	84	60	84	89
	Dedicated Pins	3	3	3	3	3
VCC		6	16	6	16	29
VCCAUX		4	8	4	8	17
VCCPLL		1	4	1	4	8
VCCIO	Bank0	1	4	1	4	5
	Bank1	1	3	1	3	4
	Bank2	2	4	2	4	5
	Bank3	2	4	2	4	5
	Bank4	2	4	2	4	4
	Bank5	2	4	2	4	5
	Bank6	2	4	2	4	5
	Bank7	2	4	2	4	5
	Bank8	1	2	1	2	2
GND, GND0 to GND7		22	57	22	57	80
NC		17	11	17	12	37
Single Ended/ Differential I/O Pairs per Bank (including emulated with resistors)	Bank0	0/0	36/18	0/0	36/18	63/31
	Bank1	0/0	18/9	0/0	18/9	18/9
	Bank2	14/7	30/15	14/7	30/15	50/25
	Bank3	16/8	36/18	16/8	36/18	43/21
	Bank4	32/16	62/31	32/16	62/31	50/21
	Bank5	20/10	28/14	20/10	28/14	60/30
	Bank6	16/8	40/20	16/8	39/19	52/25
	Bank7	28/14	40/20	28/14	40/20	60/30
	Bank8	14/7	14/7	14/7	14/7	14/7
True LVDS I/O Pairs per Bank	Bank0 (Top Edge)	0	0	0	0	0
	Bank1 (Top Edge)	0	0	0	0	0
	Bank2 (Right Edge)	3	7	3	7	12
	Bank3 (Right Edge)	4	9	4	9	11
	Bank4 (Bottom Edge)	0	0	0	0	0
	Bank5 (Bottom Edge)	0	0	0	0	0
	Bank6 (Left Edge)	4	10	4	10	14
	Bank7 (Left Edge)	7	10	7	10	15
	Bank8 (Right Edge)	0	0	0	0	0

LatticeECP2M Pin Information Summary, LFE2M20 and LFE2M35 (Cont.)

Pin Type		LFE2M20		LFE2M35		
		256 fpBGA	484 fpBGA	256 fpBGA	484 fpBGA	672 fpBGA
Available DDR-Interfaces per I/O Bank[1]	Bank0	0	0	0	0	0
	Bank1	0	0	0	0	0
	Bank2	0	1	0	1	3
	Bank3	0	1	0	1	2
	Bank4	2	4	2	4	3
	Bank5	1	2	1	2	3
	Bank6	0	3	0	1	2
	Bank7	1	2	1	2	3
	Bank8	0	0	0	0	0
PCI Capable I/Os per Bank	Bank0	0	0	0	0	0
	Bank1	0	0	0	0	0
	Bank2	0	0	0	0	0
	Bank3	0	0	0	0	0
	Bank4	32	62	32	62	50
	Bank5	20	28	20	28	60
	Bank6	16	40	16	39	52
	Bank7	28	40	28	40	60
	Bank8	0	0	0	0	0

1. Minimum requirement to implement a fully functional 8-bit wide DDR bus. Available DDR interface consists of at least 12 I/Os (1 DQS + 1 DQSB + 8 DQs + 1 DM + Bank VREF1).

LatticeECP2M Pin Information Summary, LFE2M50, LFE2M70 and LFE2M100

Pin Type		LFE2M50			LFE2M70		LFE2M100	
		484 fpBGA	672 fpBGA	900 fpBGA	900 fpBGA	1152 fpBGA	900 fpBGA	1152 fpBGA
Single Ended User I/O		270	372	410	416	436	416	520
Differential Pair User I/O		135	185	205	208	218	207	260
Configuration	TAP Pins	5	5	5	5	5	5	5
	Muxed Pins	14	14	14	14	14	14	14
	Dedicated Pins (Non TAP)	7	7	7	7	7	7	7
Non Configuration	Muxed Pins	69	72	72	75	76	74	78
	Dedicated Pins	3	3	3	3	3	3	3
VCC		16	20	62	44	44	44	44
VCCAUX		8	26	18	16	12	16	12
VCCPLL		4	8	4	4	4	4	4
VCCIO	Bank0	4	5	6	6	7	6	7
	Bank1	3	4	6	6	7	6	7
	Bank2	4	5	9	9	9	9	9
	Bank3	4	5	9	9	9	9	9
	Bank4	4	4	6	6	7	6	7
	Bank5	4	5	6	6	7	6	7
	Bank6	4	5	9	9	9	9	9
	Bank7	4	5	9	9	9	9	9
	Bank8	2	2	2	2	2	2	2
GND, GND0 to GND7		57	80	122	122	134	122	134
NC		31	35	121	63	283	63	199
Single Ended/ Differential I/O Pairs per Bank (including emulated with resistors)	Bank0	36/18	63/31	56/28	34/17	46/23	34/17	54/27
	Bank1	18/9	18/9	36/18	42/21	34/17	42/21	44/22
	Bank2	30/15	50/25	54/27	70/35	72/36	70/35	80/40
	Bank3	36/18	43/21	44/22	60/30	64/32	60/30	80/40
	Bank4	42/21	24/12	38/19	38/19	40/20	38/19	44/22
	Bank5	28/14	60/30	58/29	40/20	40/20	40/20	46/23
	Bank6	40/20	54/27	60/30	62/31	66/33	62/31	82/41
	Bank7	40/20	60/30	64/32	70/35	74/37	70/35	90/45
	Bank8	0/0	0/0	0/0	0/0	0/0	0/0	0/0
True LVDS I/O Pairs per Bank	Bank0 (Top Edge)	0	0	0	0	0	0	0
	Bank1 (Top Edge)	0	0	0	0	0	0	0
	Bank2 (Right Edge)	7	12	13	17	18	17	20
	Bank3 (Right Edge)	9	11	11	15	16	15	20
	Bank4 (Bottom Edge)	0	0	0	0	0	0	0
	Bank5 (Bottom Edge)	0	0	0	0	0	0	0
	Bank6 (Left Edge)	10	14	15	15	16	15	20
	Bank7 (Left Edge)	10	15	17	17	18	17	22
	Bank8 (Right Edge)	0	0	0	0	0	0	0

LatticeECP2M Pin Information Summary, LFE2M50, LFE2M70 and LFE2M100 (Cont.)

Pin Type		LFE2M50			LFE2M70		LFE2M100	
		484 fpBGA	672 fpBGA	900 fpBGA	900 fpBGA	1152 fpBGA	900 fpBGA	1152 fpBGA
Available DDR-Interfaces per I/O Bank[1]	Bank0	0	0	0	0	0	0	0
	Bank1	0	0	0	0	0	0	0
	Bank2	2	2	2	4	4	4	4
	Bank3	2	1	1	3	4	3	5
	Bank4	3	1	3	3	3	3	3
	Bank5	2	3	3	2	3	2	3
	Bank6	1	2	2	3	4	3	5
	Bank7	3	3	3	4	4	4	5
	Bank8	0	0	0	0	0	0	0
PCI Capable I/Os per Bank	Bank0	0	0	0	0	0	0	0
	Bank1	0	0	0	0	0	0	0
	Bank2	0	0	0	0	72	0	80
	Bank3	0	0	0	0	64	0	80
	Bank4	50	24	48	48	40	48	44
	Bank5	60	60	50	40	40	40	46
	Bank6	52	54	60	62	66	62	82
	Bank7	60	60	68	70	74	70	90
	Bank8	0	0	0	0	0	0	0

1. Minimum requirement to implement a fully functional 8-bit wide DDR bus. Available DDR interface consists of at least 12 I/Os (1 DQS + 1 DQSB + 8 DQs + 1 DM + Bank VREF1).

Available Device Resources by Package, LatticeECP2

Resource	Device	256 fpBGA	484 fpBGA	672 fpBGA	900 fpBGA
PLL/DLL	ECP2-6	4	—	—	—
	ECP2-12	4	4	—	—
	ECP2-20	4	4	4	—
	ECP2-35	—	4	4	—
	ECP2-50	—	6	6	—
	ECP2-70	—	—	8	8

Available Device Resources by Package, LatticeECP2M

Resource	Device	256 fpBGA	484 fpBGA	672 fpBGA	900 fpBGA	1152 fpBGA
PLL/DLL	ECP2M20	10	10	—	—	—
	ECP2M35	10	10	10	—	—
	ECP2M50	—	10	10	10	—
	ECP2M70	—	—	—	10	10
	ECP2M100	—	—	—	10	10

LatticeECP2 Power Supply and NC

Signals	144 TQFP[3]	208 PQFP[3]	256 fpBGA[4]	484 fpBGA[4]
VCC	16, 22, 29, 48, 54, 83, 94, 102, 128, 135	12, 19, 28, 40, 74, 80, 97, 116, 129, 140, 146, 171, 188, 198	**LFE2-6:** G7, G9, G10, H7, J10, K10, K8 **LFE2-12/LFE2-20:** G7, G9, G10, H7, J10, K10, K8	**LFE2-12/LFE2-20:** N6, N18, J10, J11, J12, J13, K14, K9, L14, L9, M14, M9, N14, N9, P10, P11, P12, P13 **LFE2-35/LFE2-50:** J10, J11, J12, J13, K14, K9, L14, L9, M14, M9, N14, N9, P10, P11, P12, P13
VCCIO0	139	195, 206	C5, E7	G10, G9, H8, H9
VCCIO1	117	162, 170	C12, E10	G11, G12, G13, G14
VCCIO2	106	143, 148	E14, G12	H14, H15, J15, K16
VCCIO3	89	123, 135	K12, M14	L16, M16, N16, P16
VCCIO4	64	93, 100	M10, P12	R14, T12, T13, T14
VCCIO5	42	55, 63	M7, P5	R9, T10, T11, T9
VCCIO6	31	38, 44	K5, M3	N7, P7, P8, R8
VCCIO7	9	10, 14	E3, G5	J8, K7, L7, M7
VCCIO8	85	113, 118	T15	P15, R15
VCCJ	35	51	K7	T8
VCCAUX	6, 39, 90, 142	7, 30, 70, 86, 125, 151, 174, 190	G8, H10, J7, K9	G5, K5, R5, V7, V11, V8, V13, V15, M17, P17, E17, G18, D11, F13, C5, E6
VCCPLL	None	None	None	**LFE2-12/LFE2-20:** None **LFE2-35:** N6, N18 **LFE2-50:** N6, N18, K6, J16
GND[1]	11, 21, 30, 47, 51, 61, 81, 95, 105, 120, 133, 138	5, 13, 17, 25, 32, 42, 60, 68, 77, 81, 89, 102, 115, 122, 139, 145, 159, 169, 175, 184, 192, 201	A1, A16, B12, B5, C8, E15, E2, H14, H8, H9, J3, J8, J9, M15, M2, P9, R12, R5, T1, T16	A22, AA19, AA4, AB1, AB22, B19, B4, C14, C9, D2, D21, F17, F6, H10, H11, H12, H13, J14, J20, J3, J9, K10, K11, K12, K13, K15, K8, L10, L11, L12, L13, L15, L8, M10, M11, M12, M13, M15, M8, N10, N11, N12, N13, N15, N8, P14, P20, P3, P9, R10, R11, R12, R13, U17, U6, W2, W21, Y14, Y9, A1
NC[2]	**LFE2-6:** 45, 46, 124, 127 **LFE2-12:** 127	None	**LFE2-6:** K6, R3, P4 **LFE2-12/LFE2-20:** None	**LFE2-12:** E3, F3, F1, H4, F2, H5, G1, G3, G2, G4, K6, N1, M2, N2, M1, N3, N5, N4, P5, N19, M19, J22, L22, H22, K22, J16, D22, F21, E21, E22, H19, G20, G19, F20, C21, C22, H6, J6, H3, H2, H17, H16, H20, H18 **LFE2-20/LFE2-35:** K6, J16, H6, J6, H3, H2, H17, H16, H20, H18 **LFE2-50:** None

1. All grounds must be electrically connected at the board level. For fpBGA packages, the total number of GND balls is less than the actual number of GND logic connections from the die to the common package GND plane.
2. NC pins should not be connected to any active signals, VCC or GND.
3. Pin orientation follows the conventional order from the pin 1 marking of the top side view and counter-clockwise.
4. Pin orientation A1 starts from the upper left corner of the top side view with alphabetical order ascending vertically and numerical order ascending horizontally.

LatticeECP2 Power Supply and NC (Cont.)

Signals	672 fpBGA[3]	900 fpBGA[3]
VCC	**LFE2-20:** R8, P18, M8, L20, L12, L13, L14, L15, M11, M12, M15, M16, N11, N16, P11, P16, R11, R12, R15, R16, T12, T13, T14, T15 **LFE2-35/LFE2-50:** L12, L13, L14, L15, M11, M12, M15, M16, N11, N16, P11, P16, R11, R12, R15, R16, T12, T13, T14, T15 **LFE2-70:** L12, L13, L14, L15, M11, M12, M15, M16, N11, N16, P11, P16, R11, R12, R15, R16, T12, T13, T14, T15	AA11, AA20, K11, K21, K22, L11, L12, L13, L18, L19, L20, M11, M20, N11, N20, V11, V20, W11, W20, Y10, Y11, Y12, Y13, Y18, Y19, Y20
VCCIO0	D11, D6, G9, J12, K12	J13, J14, K12, K13, K14, K15
VCCIO1	D16, D21, G18, J15, K15	J17, J18, J20, K17, K18, K20
VCCIO2	F23, J20, L23, M17, M18	L21, M21, M22, N21, N22, R21
VCCIO3	AA23, R17, R18, T23, V20	U21, U22, V21, V22, W21, Y22
VCCIO4	AC16, AC21, U15, V15, Y18	AA16, AA17, AA18, AA19, AB17, AB18
VCCIO5	AC11, AC6, U12, V12, Y9	AA12, AA13, AA14, AB12, AB13, AB14
VCCIO6	AA4, R10, R9, T4, V7	U10, U9, V10, W10, W9, Y9
VCCIO7	F4, J7, L4, M10, M9	L10, L9, M10, N10, P10, R10
VCCIO8	AE25, V18	AA21, Y21
VCCJ	AB5	AD3
VCCAUX	J10, J11, J16, J17, K18, L18, T18, U18, V16, V17, V10, V11, T9, U9, K9, L9	AA15, AB11, AB19, AB20, J11, J12, J19, K19, L22, M9, N9, P21, P9, T10, T21, V9, W22
VCCPLL	**LFE2-20:** None **LFE2-35/LFE2-70:** R8, P18 **LFE2-50:** R8, P18, M8, L20	P22, P8, T22, Y7
GND[1]	A2, A25, AA18, AA24, AA3, AA9, AD11, AD16, AD21, AD6, AE1, AE26, AF2, AF25, B1, B26, C11, C16, C21, C6, F18, F24, F3, F9, J13, J14, J21, J6, K10, K11, K13, K14, K16, K17, L10, L11, L16, L17, L24, L3, M13, M14, N10, N12, N13, N14, N15, N17, P10, P12, P13, P14, P15, P17, R13, R14, T10, T11, T16, T17, T24, T3, U10, U11, U13, U14, U16, U17, V13, V14, V21, V6	A1, A30, AC28, AC3, AH13, AH18, AH23, AH28, AH3, AH8, AK1, AK30, C13, C18, C23, C28, C3, C8, H28, H3, L14, L15, L16, L17, M12, M13, M14, M15, M16, M17, M18, M19, N12, N13, N14, N15, N16, N17, N18, N19, N28, N3, P11, P12, P13, P14, P15, P16, P17, P18, P19, P20, R11, R12, R13, R14, R15, R16, R17, R18, R19, R20, T11, T12, T13, T14, T15, T16, T17, T18, T19, T20, U11, U12, U13, U14, U15, U16, U17, U18, U19, U20, V12, V13, V14, V15, V16, V17, V18, V19, V28, V3, W12, W13, W14, W15, W16, W17, W18, W19, Y14, Y15, Y16, Y17
NC[2]	**LFE2-20:** E4, E3, E2, E1, H6, H5, F2, F1, H8, J9, G4, G3, K3, K2, K1, L2, L1, M2, M1, N2, T1, T2, P8, P6, P5, P4, U1, V1, P3, R3, R4, U2, V2, W2, T6, R5, AA19, W17, Y19, Y17, AF20, AE20, AA20, W18, AD20, AE21, AF21, AF22, R22, T21, P26, P25, R24, R23, P20, R19, P21, P19, P23, P22, N22, R21, N26, N25, J26, J25, J23, K23, H26, H25, H24, H23, F22, E24, D25, C25, D24, B25, H21, G22, B24, C24, D23, C23, E19, C19, B21, B20, D19, B19, G17, E18, G19, F17, A20, A19, E17, D18, M3, N6, P24 **LFE2-35:** K3, K2, K1, L2, L1, M2, M1, N2, M8, P3, R3, R4, U2, V2, W2, AF20, AE20, AA20, W18, AD20, AE21, AF21, AF22, P26, P25, R24, R23, P20, R19, L20, J26, J25, J23, K23, H26, H25, H24, H23, E19, C19, B21, B20, D19, B19, G17, E18, G19, F17, A20, A19, E17, D18, M3, N6, P24 **LFE2-50:** N6, P24, M3 **LFE2-70:** M8, L20, M3, P24, N6	A2, A3, A4, A5, AB28, AC4, AD23, AE1, AE2, AE29, AE3, AE30, AE4, AE5, AE6, AF1, AF2, AF23, AF26, AF27, AF28, AF29, AF3, AF30, AF4, AF5, AG1, AG13, AG16, AG18, AG2, AG26, AG27, AG28, AG29, AG3, AG30, AG4, AG8, AH1, AH16, AH2, AH26, AH27, AH29, AH30, AH4, AJ1, AJ2, AJ27, AJ28, AJ29, AJ3, AJ30, AK2, AK27, AK28, AK29, AK3, B1, B2, B3, B30, B4, B5, C1, C2, C29, C30, C4, D13, D18, D23, D28, D29, D3, D30, D4, E25, E26, E27, E28, E29, E3, E30, E4, E5, E6, F25, F5, F6, G6, G7, K10, K9, N27, N4, R1, R2, V27, V4

1. All grounds must be electrically connected at the board level. For fpBGA packages, the total number of GND balls is less than the actual number of GND logic connections from the die to the common package GND plane.
2. NC pins should not be connected to any active signals, VCC or GND.
3. Pin orientation A1 starts from the upper left corner of the top side view with alphabetical order ascending vertically and numerical order ascending horizontally.

LatticeECP2M Power Supply and NC

Signal	256 fpBGA	484 fpBGA
V_{CC}	G7, G9, H7, J10, K10, K8	J10, J11, J12, J13, K14, K9, L14, L9, M14, M9, N14, N9, P10, P11, P12, P13
V_{CCIO0}	E7	B5, B9, E7, H9
V_{CCIO1}	E10	D13, E16, H14
V_{CCIO2}	E14, G12	E21, G18, J15, K19
V_{CCIO3}	K12, M14	N19, P15, T18, V21
V_{CCIO4}	M10, P12	AA18, R14, V16, W13
V_{CCIO5}	M7, P5	AA5, R9, V7, W10
V_{CCIO6}	K5, M3	N4, P8, T5, V2
V_{CCIO7}	E3, G5	E2, G5, J8, K4
V_{CCIO8}	T15	AA22, U19
V_{CCJ}	K7	W4
V_{CCAUX}	G8, H10, J7, K9	H11, H12, L15, L8, M15, M8, R11, R12
V_{CCPLL}	G10	R8, H15, H8, R15
SERDES Power[3]	C15, B15, C12, A12, C11, C10, C14, C13, B9, C9, C5, C4, C8, C7, A6, C6, B3, C3	C22, B22, C19, A19, C18, C17, C21, C20, B16, C16, C12, C11, C15, C14, A13, C13, B10, C10
GND[1]	A1, A15, A16, A3, A9, B12, B6, E15, E2, H14, H8, H9, J3, J8, J9, M15, M2, P9, R12, R5, T1, T16	A1, A10, A16, A22, AA19, AA4, AB1, AB22, B13, B19, B4, D16, D2, D21, D7, G19, G4, H10, H13, J14, J9, K10, K11, K12, K13, K15, K20, K3, K8, L10, L11, L12, L13, M10, M11, M12, M13, N10, N11, N12, N13, N15, N20, N3, N8, P14, P9, R10, R13, T19, T4, W16, W2, W21, W7, Y10, Y13
NC[2]	D10, D11, D12, D13, D14, D4, D5, D6, D7, E11, E6, E8, E9, F10, F7, F8, F9	**LFE2M20:** D14, D15, E14, E15, F13, F14, F15, G12, G13, G14, G15 **LFE2M35:** D14, D15, E14, E15, F13, F14, F15, G12, G13, G14, G15, U6 **LFE2M50:** Y15, W15, AB20, AB21, AA20, AB19, AB18, Y22, Y21, Y17, Y18, Y16, W17, Y19, Y20, W19, W18, V17, V18, D15, G14, G15, D14, E15, E14, F15, F14, F13, G12, G13

1. All grounds must be electrically connected at the board level. For fpBGA packages, the total number of GND balls is less than the actual number of GND logic connections from the die to the common package GND plane.
2. NC pins should not be connected to any active signals, VCC or GND.
3. For package migration across device densities, the designer must comprehend the package pin requirements for the SERDES blocks. Specifically, the SERDES power pins of the largest density device must be accounted to accommodate migration to other smaller devices using the same package. Please refer to TN1160, LatticeECP2/M Density Migration for more details.

LatticeECP2M Power Supply and NC (Cont.)

Signal	672 fpBGA	900 fpBGA
V_{CC}	**LFE2M35:** AD13, AD14, AD16, AD17, AD19, AD21, AD22, AD24, AD25, L12, L13, L14, L15, M11, M12, M15, M16, N11, N16, P11, P16, R11, R12, R15, R16, T12, T13, T14, T15 **LFE2M50:** L12, L13, L14, L15, M11, M12, M15, M16, N11, N16, P11, P16, R11, R12, R15, R16, T12, T13, T14, T15	**LFE2M50:** AH1, AH4, AH5, AH2, AH7, AH12, AH9, AH10, AH13, C13, C10, C9, C12, C7, C2, C5, C4, C1, L12, L13, L18, L19, M11, M12, M13, M14, M15, M16, M17, M18, M19, M20, N11, N12, N19, N20, P12, P19, R12, R19, T12, T19, U12, U19, V11, V12, V19, V20, W11, W12, W13, W14, W15, W16, W17, W18, W19, W20, Y12, Y13, Y18, Y19 **LFE2M70/LFE2M100:** L12, L13, L18, L19, M11, M12, M13, M14, M15, M16, M17, M18, M19, M20, N11, N12, N19, N20, P12, P19, R12, R19, T12, T19, U12, U19, V11, V12, V19, V20, W11, W12, W13, W14, W15, W16, W17, W18, W19, W20, Y12, Y13, Y18, Y19
V_{CCIO0}	B12, B7, F11, J13, K12	D14, E6, E9, F12, K12, K13
V_{CCIO1}	D18, F16, J14, K15	D17, E22, E25, F19, K18, K19
V_{CCIO2}	G25, L21, M17, M25, N18	F28, J25, K28, M21, M24, N21, N28, P21, R25
V_{CCIO3}	P18, R17, R25, T21, Y25	AA28, AB25, AE28, T25, U21, V21, V28, W21, W24
V_{CCIO4}	AA16, AC18, U15, V14	AA18, AA19, AE19, AF22, AG17, AG25
V_{CCIO5}	AA11, AE12, AE7, U12, V13	AA12, AA13, AE12, AF9, AG14, AG6
V_{CCIO6}	P9, R10, R2, T6, Y2	AA3, AB6, AE3, T6, U10, V10, V3, W10, W7
V_{CCIO7}	G2, L6, M10, M2, N9	F3, J6, K3, M10, M7, N10, N3, P10, R6
V_{CCIO8}	AC24, U17	AA25, AD28
V_{CCJ}	AA7	AG1
V_{CCAUX}	**LFE2M35:** AE19, J11, J12, J15, J16, L18, L9, M18, M9, R18, R9, T18, T9, V11, V12, V15, V16 **LFE2M50:** J11, J12, J15, J16, L18, L9, M18, M9, R18, R9, T18, T9, V11, V12, V15, V16	**LFE2M50:** AJ7, B7, AA10, AA11, AA20, AA21, K10, K11, K20, K21, L10, L11, L20, L21, Y10, Y11, Y20, Y21 **LFE2M70/LFE2M100:** AA10, AA11, AA20, AA21, K10, K11, K20, K21, L10, L11, L20, L21, Y10, Y11, Y20, Y21
V_{CCPLL}	H7, K6, P7, R8, V18, P20, J17, G19	N13, N18, V13, V18
SERDES Power[3]	**LFE2M35:** C25, B25, C22, A22, C21, C20, C24, C23, B19, C19, C15, C14, C18, C17, A16, C16, B13, C13 **LFE2M50:** AD13, AE13, AD16, AF16, AD17, AD18, AD14, AD15, AD19, AE19, AD23, AD24, AD20, AD21, AF22, AD22, AE25, AD25, C25, B25, C22, A22, C21, C20, C24, C23, B19, C19, C15, C14, C18, C17, A16, C16, B13, C13	**LFE2M50:** AH18, AJ18, AH21, AK21, AH22, AH23, AH19, AH20, AH24, AJ24, AH28, AH29, AH25, AH26, AK27, AH27, AJ30, AH30, C30, B30, C27, A27, C26, C25, C29, C28, B24, C24, C20, C19, C23, C22, A21, C21, B18, C18 **LFE2M70/LFE2M100:** C13, B13, C10, A10, C9, C8, C12, C11, B7, C7, C3, C2, C6, C5, A4, C4, B1, C1, C30, B30, C27, A27, C26, C25, C29, C28, B24, C24, C20, C19, C23, C22, A21, C21, B18, C18, AH18, AJ18, AH21, AK21, AH22, AH23, AH19, AH20, AH24, AJ24, AH28, AH29, AH25, AH26, AK27, AH27, AJ30, AH30, AH1, AJ1, AH4, AK4, AH5, AH6, AH2, AH3, AH7, AJ7, AH11, AH12, AH8, AH9, AK10, AH10, AJ13, AH13

LatticeECP2M Power Supply and NC (Cont.)

Signal	672 fpBGA	900 fpBGA
GND[1]	A13, A19, A2, A25, AA2, AA25, AB18, AB22, AB5, AB9, AE1, AE11, AE16, AE22, AE26, AE6, AF13, AF19, AF2, AF25, B1, B11, B16, B22, B26, B6, E18, E22, E5, E9, F2, F25, G11, G16, J22, J5, K11, K13, K14, K16, L10, L11, L16, L17, L2, L20, L25, L7, M13, M14, N10, N12, N13, N14, N15, N17, P10, P12, P13, P14, P15, P17, R13, R14, T10, T11, T16, T17, T2, T20, T25, T7, U11, U13, U14, U16, V22, V5, Y11, Y16	**LFE2M50:** A1, A13, A18, A24, A30, A7, AA14, AA15, AA16, AA17, AA24, AA27, AA4, AB24, AB7, AD12, AD19, AD27, AE22, AE27, AE4, AE9, AF14, AF17, AF25, AF6, AJ10, AJ21, AJ27, AJ4, AK1, AK13, AK18, AK24, AK30, AK7, B10, B21, B27, B4, D25, D6, E14, E17, F22, F27, F4, F9, G12, G19, J24, J7, K14, K15, K16, K17, K27, K4, L14, L15, L16, L17, M23, M8, N14, N15, N16, N17, N27, N4, P11, P13, P14, P15, P16, P17, P18, P20, R10, R11, R13, R14, R15, R16, R17, R18, R20, R21, R24, R7, T10, T11, T13, T14, T15, T16, T17, T18, T20, T21, T24, T7, U11, U13, U14, U15, U16, U17, U18, U20, V14, V15, V16, V17, V27, V4, W23, W8, Y14, Y15, Y16, Y17 **LFE2M70/LFE2M100:** A1, A13, A18, A24, A30, A7, AA14, AA15, AA16, AA17, AA24, AA27, AA4, AB24, AB7, AD12, AD19, AD27, AE22, AE27, AE4, AE9, AF14, AF17, AF25, AF6, AJ10, AJ21, AJ27, AJ4, AK1, AK13, AK18, AK24, AK30, AK7, B10, B21, B27, B4, D25, D6, E14, E17, F22, F27, F4, F9, G12, G19, J24, J7, K14, K15, K16, K17, K27, K4, L14, L15, L16, L17, M23, M8, N14, N15, N16, N17, N27, N4, P11, P13, P14, P15, P16, P17, P18, P20, R10, R11, R13, R14, R15, R16, R17, R18, R20, R21, R24, R7, T10, T11, T13, T14, T15, T16, T17, T18, T20, T21, T24, T7, U11, U13, U14, U15, U16, U17, U18, U20, V14, V15, V16, V17, V27, V4, W23, W8, Y14, Y15, Y16, Y17
NC[2]	**LFE2M35:** AB3, AB4, AC1, AC2, AD15, AD18, AD20, AD23, AE13, AE25, AF16, AF22, B4, B5, C26, D20, D21, D22, D23, D24, D25, D26, E20, E21, E25, E26, F20, G20, K10, K17, R4, U10, U23, V10, W7, N7, V7 **LFE2M50:** AB3, AB4, AC1, AC2, B4, B5, C26, D20, D21, D22, D23, D24, D25, D26, E20, E21, E25, E26, F20, G20, K10, K17, R4, U10, U23, V10, W7, AB21, AC20, AC21, AC22, AC23, AC25, AD26, W20	**LFE2M50:** G5, G4, K7, K8, E1, F2, F1, G3, G2, G1, L9, L7, K6, K5, L8, L6, AA1, AA2, Y3, AB1, Y9, Y8, Y7, AA7, AB2, AB3, AA5, AA6, AB4, AB5, AA8, AA9, AJ1, AK4, AH6, AH3, AH11, AH8, AK10, AJ13, AB26, AB27, Y24, Y25, AA29, Y28, Y30, Y29, W22, V22, Y27, Y26, W30, W29, W25, W26, L24, L23, D30, D29, K24, K25, J27, K26, J26, H26, H27, G26, H23, H24, D28, E28, J18, J19, H17, J17, F18, F17, B13, A10, C8, C11, C3, C6, A4, B1, AA26, AB11, AB12, AB13, AB14, AB15, AB16, AB17, AB19, AB20, AB21, AC11, AC21, AC22, AD21, AD22, AE23, AF20, AF23, AG23, AG26, F20, F23, G10, G20, G21, H19, H20, H21, H22, J20, J21, R9, U22, W9 **LFE2M70/LFE2M100:** AA26, AB10, AB11, AB12, AB13, AB14, AB15, AB16, AB17, AB19, AB20, AB21, AB9, AC10, AC11, AC21, AC22, AC8, AC9, AD21, AD22, AD4, AD5, AD6, AD7, AD8, AE23, AE5, AE6, AE7, AF20, AF23, AF5, AG23, AG26, D10, E10, E11, F10, F20, F23, F8, G10, G20, G21, G7, G8, G9, H19, H20, H21, H22, H6, H8, H9, J10, J20, J21, J9, K9, R9, U22, W9

1. All grounds must be electrically connected at the board level. For fpBGA packages, the total number of GND balls is less than the actual number of GND logic connections from the die to the common package GND plane.
2. NC pins should not be connected to any active signals, VCC or GND.
3. For package migration across device densities, the designer must comprehend the package pin requirements for the SERDES blocks. Specifically, the SERDES power pins of the largest density device must be accounted to accommodate migration to other smaller devices using the same package. Please refer to TN1160, LatticeECP2/M Density Migration for more details.

LatticeECP2M Power Supply and NC (Cont.)

Signal	1152 fpBGA
V_{CC}	AA13, AA14, AA15, AA16, AA17, AA18, AA19, AA20, AA21, AA22, AB14, AB15, AB20, AB21, N14, N15, N20, N21, P13, P14, P15, P16, P17, P18, P19, P20, P21, P22, R13, R14, R21, R22, T14, T21, U14, U21, V14, V21, W14, W21, Y13, Y14, Y21, Y22
V_{CCIO0}	C12, C16, E14, H12, H16, M14, M15
V_{CCIO1}	C19, C23, E21, H19, H23, M20, M21
V_{CCIO2}	G32, K28, K32, N27, N32, P23, R23, T27, T32
V_{CCIO3}	AA23, AB27, AB32, AE28, AE32, AH32, W27, W32, Y23
V_{CCIO4}	AC20, AC21, AG19, AG23, AK21, AM19, AM23
V_{CCIO5}	AC14, AC15, AG12, AG16, AK14, AM12, AM16
V_{CCIO6}	AA12, AB3, AB8, AE3, AE7, AH3, W3, W8, Y12
V_{CCIO7}	G3, K3, K7, N3, N8, P12, R12, T3, T8
V_{CCIO8}	AD28, AG32
V_{CCJ}	AK3
V_{CCAUX}	AB12, AB13, AB22, AB23, AC13, AC22, M13, M22, N12, N13, N22, N23
V_{CCPLL}	R15, R20, Y15, Y20
SERDES Power[3]	D7, B9, B8, D9, B7, E7, B6, D8, E6, D6, D4, B5, D3, B4, C1, B3, B1, B2, B33, B34, B32, C34, B31, D32, B30, D31, E29, D29, D27, B29, E28, B28, D26, B27, B26, D28, AL28, AN26, AN27, AL26, AN28, AK28, AN29, AL27, AL29, AK29, AL31, AN30, AL32, AN31, AM34, AN32, AN34, AN33, AN2, AN1, AN3, AM1, AN4, AL3, AN5, AL4, AL6, AK6, AL8, AN6, AK7, AN7, AL9, AN8, AN9, AL7
GND[1]	A1, A10, A13, A22, A25, A34, AB16, AB17, AB18, AB19, AB26, AB31, AB4, AB9, AC16, AC17, AC18, AC19, AD27, AE27, AE31, AE4, AE8, AF12, AF16, AF19, AF23, AG31, AH31, AH4, AJ14, AJ21, AK27, AK8, AL10, AL16, AL19, AL2, AL25, AL33, AP1, AP10, AP13, AP22, AP25, AP34, D10, D16, D19, D2, D25, D33, E27, E8, F14, F21, G31, G4, J12, J16, J19, J23, K27, K31, K4, K8, M16, M17, M18, M19, N16, N17, N18, N19, N26, N31, N4, N9, R16, R17, R18, R19, T12, T13, T15, T16, T17, T18, T19, T20, T22, T23, T26, T31, T4, T9, U12, U13, U15, U16, U17, U18, U19, U20, U22, U23, V12, V13, V15, V16, V17, V18, V19, V20, V22, V23, W12, W13, W15, W16, W17, W18, W19, W20, W22, W23, W26, W31, W4, W9, Y16, Y17, Y18, Y19
NC[2]	**LFE2M70:** H2, H1, G5, G6, M9, M10, H3, H4, P3, P4, P9, M7, P1, P2, N7, P7, AC7, AC5, AC6, AD5, AD4, AD3, AD10, AD8, AD2, AD1, AD9, AC11, AD6, AD7, AE1, AE2, AJ12, AH12, AL13, AK13, AE14, AG13, AH22, AH21, AG22, AG21, AF33, AF34, AC27, AC28, AD29, AD30, AE33, AE34, AD32, AD31, AB25, AC25, AB28, AA26, AD33, AD34, P30, P29, P31, P32, R25, T24, N34, N33, F24, G23, J22, G22, H21, K21, L19, L20, L18, K19, J14, L15, H14, K14, F12, D11, F11, E11, A11, A12, A23, A24, AA11, AB11, AC26, AC30, AD11, AD12, AD13, AD14, AD15, AD19, AD21, AD22, AD23, AE10, AE11, AE12, AE13, AE19, AE21, AE22, AE23, AF11, AF21, AF22, AF24, AF8, AF9, AG10, AG11, AG24, AG25, AG26, AG3, AG7, AG8, AG9, AH10, AH11, AH13, AH24, AH25, AH26, AH27, AH5, AH6, AH7, AH8, AH9, AJ10, AJ11, AJ13, AJ24, AJ25, AJ26, AJ27, AJ3, AJ4, AJ5, AJ6, AJ7, AJ8, AJ9, AK10, AK11, AK12, AK24, AK25, AK26, AK4, AK9, AL11, AL12, AL34, AM10, AM11, AM13, AM25, AN10, AN11, AN12, AN13, AN24, AN25, AP11, AP12, AP24, B10, B11, B12, B13, B22, B23, B24, B25, C10, C11, C13, C22, C24, C25, D1, D15, D24, D34, E10, E24, E25, E26, E3, E31, E32, E33, E34, E4, E9, F10, F25, F26, F27, F28, F29, F30, F31, F32, F33, F34, F5, F6, F7, F8, F9, G10, G11, G24, G25, G26, G27, G28, G29, G30, G33, G34, G7, G8, G9, H10, H11, H24, H25, H26, H27, H28, H29, H8, H9, J10, J11, J24, J25, J26, J9, K10, K11, K12, K13, K23, K24, K25, K26, L11, L12, L13, L14, L21, L22, L23, L24, L25, L26, M11, M24, M25, M6, M8, N10, N11, P10, P25, P26, R9, T11, U11, W11, Y10, Y11 **LFE2M100:** A11, A12, A23, A24, AA11, AB11, AC26, AC30, AD11, AD12, AD13, AD14, AD15, AD19, AD21, AD22, AD23, AE10, AE11, AE12, AE13, AE19, AE21, AE22, AE23, AF11, AF21, AF22, AF24, AF8, AF9, AG10, AG11, AG24, AG25, AG26, AG3, AG7, AG8, AG9, AH10, AH11, AH13, AH24, AH25, AH26, AH27, AH5, AH6, AH7, AH8, AH9, AJ10, AJ11, AJ13, AJ24, AJ25, AJ26, AJ27, AJ3, AJ4, AJ5, AJ6, AJ7, AJ8, AJ9, AK10, AK11, AK12, AK24, AK25, AK26, AK4, AK9, AL11, AL12, AL34, AM10, AM11, AM13, AM25, AN10, AN11, AN12, AN13, AN24, AN25, AP11, AP12, AP24, B10, B11, B12, B13, B22, B23, B24, B25, C10, C11, C13, C22, C24, C25, D1, D15, D24, D34, E10, E24, E25, E26, E3, E31, E32, E33, E34, E4, E9, F10, F25, F26, F27, F28, F29, F30, F31, F32, F33, F34, F5, F6, F7, F8, F9, G10, G11,G24, G25, G26, G27, G28, G29, G30, G33, G34, G7, G8, G9, H10, H11, H24, H25, H26, H27, H28, H29, H8, H9, J10, J11, J24, J25, J26, J9, K10, K11, K12, K13, K23, K24, K25, K26, L11, L12, L13, L14, L21, L22, L23, L24, L25, L26, M11, M24, M25, M6, M8, N10, N11, P10, P25, P26, R9, T11, U11, W11, Y10, Y11

1. All grounds must be electrically connected at the board level. For fpBGA packages, the total number of GND balls is less than the actual number of GND logic connections from the die to the common package GND plane.
2. NC pins should not be connected to any active signals, VCC or GND.
3. For package migration across device densities, the designer must comprehend the package pin requirements for the SERDES blocks. Specifically, the SERDES power pins of the largest density device must be accounted to accommodate migration to other smaller devices using the same package. Please refer to TN1160, LatticeECP2/M Density Migration for more details.

LFE2-6E/SE and LFE2-12E/SE Logic Signal Connections: 144 TQFP

LFE2-6E/SE					LFE2-12E/12SE			
Pin Number	Pin/Pad Function	Bank	Dual Function	Differential	Pin/Pad Function	Bank	Dual Function	Differential
1	PL2A	7	VREF2_7	T (LVDS)*	PL2A	7	VREF2_7	T (LVDS)*
2	PL2B	7	VREF1_7	C (LVDS)*	PL2B	7	VREF1_7	C (LVDS)*
3	PL4A	7		T (LVDS)*	PL4A	7		T (LVDS)*
4	PL4B	7		C (LVDS)*	PL4B	7		C (LVDS)*
5	PL6A	7	LDQ10	T (LVDS)*	PL6A	7	LDQ10	T (LVDS)*
6	VCCAUX	-			VCCAUX	-		
7	PL6B	7	LDQ10	C (LVDS)*	PL6B	7	LDQ10	C (LVDS)*
8	PL8A	7	LDQ10	T (LVDS)*	PL8A	7	LDQ10	T (LVDS)*
9	VCCIO7	7			VCCIO7	7		
10	PL8B	7	LDQ10	C (LVDS)*	PL8B	7	LDQ10	C (LVDS)*
11	GND	-			GND	-		
12	PL12A	7	LDQ10	T (LVDS)*	PL12A	7	LDQ10	T (LVDS)*
13	PL12B	7	LDQ10	C (LVDS)*	PL12B	7	LDQ10	C (LVDS)*
14	PL13A	7	PCLKT7_0/LDQ10	T	PL13A	7	PCLKT7_0/LDQ10	T
15	PL13B	7	PCLKC7_0/LDQ10	C	PL13B	7	PCLKC7_0/LDQ10	C
16	VCC	-			VCC	-		
17	PL15A	6	PCLKT6_0	T (LVDS)*	PL15A	6	PCLKT6_0	T (LVDS)*
18	PL15B	6	PCLKC6_0	C (LVDS)*	PL15B	6	PCLKC6_0	C (LVDS)*
19	PL16A	6	VREF2_6	T	PL16A	6	VREF2_6	T
20	PL16B	6	VREF1_6	C	PL16B	6	VREF1_6	C
21	GND	-			GND	-		
22	VCC	-			VCC	-		
23	PL18A	6	LLM0_GDLLT_FB_A	T	PL18A	6	LLM0_GDLLT_FB_A	T
24	PL18B	6	LLM0_GDLLC_FB_A	C	PL18B	6	LLM0_GDLLC_FB_A	C
25	LLM0_PLLCAP	6			LLM0_PLLCAP	6		
26	PL20A	6	LLM0_GPLLT_IN_A**	T (LVDS)*	PL20A	6	LLM0_GPLLT_IN_A**	T (LVDS)*
27	PL20B	6	LLM0_GPLLC_IN_A**	C (LVDS)*	PL20B	6	LLM0_GPLLC_IN_A**	C (LVDS)*
28	PL22A	6			PL22A	6		
29	VCC	-			VCC	-		
30	GND	-			GND	-		
31	VCCIO6	6			VCCIO6	6		
32	TCK	-			TCK	-		
33	TDI	-			TDI	-		
34	TDO	-			TDO	-		
35	VCCJ	-			VCCJ	-		
36	TMS	-			TMS	-		
37	PB2A	5	VREF2_5/BDQ6	T	PB2A	5	VREF2_5/BDQ6	T
38	PB2B	5	VREF1_5/BDQ6	C	PB2B	5	VREF1_5/BDQ6	C
39	VCCAUX	-			VCCAUX	-		
40	PB4A	5	BDQ6	T	PB6A	5	BDQS6	T
41	PB4B	5	BDQ6	C	PB6B	5	BDQ6	C
42	VCCIO5	5			VCCIO5	5		
43	PB6A	5	BDQS6	T	PB12A	5	BDQ15	T
44	PB6B	5	BDQ6	C	PB12B	5	BDQ15	C
45	NC	5			PB16A	5	BDQ15	T

LFE2-6E/SE and LFE2-12E/SE Logic Signal Connections: 144 TQFP (Cont.)

LFE2-6E/SE					LFE2-12E/12SE			
Pin Number	Pin/Pad Function	Bank	Dual Function	Differential	Pin/Pad Function	Bank	Dual Function	Differential
46	NC	5			PB16B	5	BDQ15	C
47	GND	-			GND	-		
48	VCC				VCC	-		
49	PB8A	5	PCLKT5_0/BDQ6	T	PB26A	5	PCLKT5_0/BDQ24	T
50	PB8B	5	PCLKC5_0/BDQ6	C	PB26B	5	PCLKC5_0/BDQ24	C
51	GND	-			GND	-		
52	PB13A	4	PCLKT4_0/BDQ15	T	PB31A	4	PCLKT4_0/BDQ33	T
53	PB13B	4	PCLKC4_0/BDQ15	C	PB31B	4	PCLKC4_0/BDQ33	C
54	VCC	-			VCC	-		
55	PB14A	4	BDQ15	T	PB34A	4	BDQ33	T
56	PB14B	4	BDQ15	C	PB34B	4	BDQ33	C
57	PB16A	4	BDQ15	T	PB40A	4	BDQ42	T
58	PB16B	4	BDQ15	C	PB40B	4	BDQ42	C
59	PB18A	4	BDQ15	T	PB44A	4	BDQ42	T
60	PB18B	4	BDQ15	C	PB44B	4	BDQ42	C
61	GND	-			GND	-		
62	PB20A	4	BDQ24	T	PB48A	4	BDQ51	T
63	PB20B	4	BDQ24	C	PB48B	4	BDQ51	C
64	VCCIO4	4			VCCIO4	4		
65	PB22A	4	BDQ24	T	PB50A	4	BDQ51	T
66	PB22B	4	BDQ24	C	PB50B	4	BDQ51	C
67	PB24A	4	BDQS24	T	PB52A	4	BDQ51	T
68	PB24B	4	BDQ24	C	PB52B	4	BDQ51	C
69	PB26A	4	BDQ24	T	PB54A	4	BDQ51	T
70	PB26B	4	BDQ24	C	PB54B	4	BDQ51	C
71	PB28A	4	VREF2_4/BDQ24	T	PB55A	4	VREF2_4/BDQ51	T
72	PB28B	4	VREF1_4/BDQ24	C	PB55B	4	VREF1_4/BDQ51	C
73	CFG1	8			CFG1	8		
74	CFG2	8			CFG2	8		
75	PROGRAMN	8			PROGRAMN	8		
76	INITN	8			INITN	8		
77	CFG0	8			CFG0	8		
78	CCLK	8			CCLK	8		
79	DONE	8			DONE	8		
80	PR29A	8	D0/SPIFASTN		PR29A	8	D0/SPIFASTN	
81	GND	-			GND	-		
82	PR26A	8	D6		PR26A	8	D6	
83	VCC	-			VCC	-		
84	PR25B	8	D7/SPID0	C	PR25B	8	D7/SPID0	C
85	VCCIO8	8			VCCIO8	8		
86	PR25A	8	DI/CSSPI0N	T	PR25A	8	DI/CSSPI0N	T
87	PR24B	8	DOUT/CSON	C	PR24B	8	DOUT/CSON	C
88	PR24A	8	BUSY/SISPI	T	PR24A	8	BUSY/SISPI	T
89	VCCIO3	3			VCCIO3	3		
90	VCCAUX	-			VCCAUX	-		

LFE2-6E/SE and LFE2-12E/SE Logic Signal Connections: 144 TQFP (Cont.)

LFE2-6E/SE					LFE2-12E/12SE			
Pin Number	Pin/Pad Function	Bank	Dual Function	Differential	Pin/Pad Function	Bank	Dual Function	Differential
91	PR20B	3	RLM0_GPLLC_IN_A**	C (LVDS)*	PR20B	3	RLM0_GPLLC_IN_A**	C (LVDS)*
92	PR20A	3	RLM0_GPLLT_IN_A**	T (LVDS)*	PR20A	3	RLM0_GPLLT_IN_A**	T (LVDS)*
93	RLM0_PLLCAP	3			RLM0_PLLCAP	3		
94	VCC	-			VCC	-		
95	GND	-			GND	-		
96	PR17B	3	RLM0_GDLLC_IN_A**	C (LVDS)*	PR17B	3	RLM0_GDLLC_IN_A**	C (LVDS)*
97	PR17A	3	RLM0_GDLLT_IN_A**	T (LVDS)*	PR17A	3	RLM0_GDLLT_IN_A**	T (LVDS)*
98	PR16B	3	VREF2_3	C	PR16B	3	VREF2_3	C
99	PR16A	3	VREF1_3	T	PR16A	3	VREF1_3	T
100	PR15B	3	PCLKC3_0	C (LVDS)*	PR15B	3	PCLKC3_0	C (LVDS)*
101	PR15A	3	PCLKT3_0	T (LVDS)*	PR15A	3	PCLKT3_0	T (LVDS)*
102	VCC	-			VCC	-		
103	PR13B	2	PCLKC2_0/RDQ10	C	PR13B	2	PCLKC2_0/RDQ10	C
104	PR13A	2	PCLKT2_0/RDQ10	T	PR13A	2	PCLKT2_0/RDQ10	T
105	GND	-			GND	-		
106	VCCIO2	2			VCCIO2	2		
107	PR2B	2	VREF2_2	C (LVDS)*	PR2B	2	VREF2_2	C (LVDS)*
108	PR2A	2	VREF1_2	T (LVDS)*	PR2A	2	VREF1_2	T (LVDS)*
109	PT28B	1	VREF2_1	C	PT55B	1	VREF2_1	C
110	PT28A	1	VREF1_1	T	PT55A	1	VREF1_1	T
111	PT26B	1		C	PT54B	1		C
112	PT26A	1		T	PT54A	1		T
113	PT24B	1		C	PT52B	1		C
114	PT24A	1		T	PT52A	1		T
115	PT22B	1		C	PT50B	1		C
116	PT22A	1		T	PT50A	1		T
117	VCCIO1	1			VCCIO1	1		
118	PT20B	1		C	PT48B	1		C
119	PT20A	1		T	PT48A	1		T
120	GND	-			GND	-		
121	PT18B	1		C	PT44B	1		C
122	PT18A	1		T	PT44A	1		T
123	PT16A	1			PT40B	1		C
124	NC	1			PT40A	1		T
125	PT14B	1		C	PT34B	1		C
126	PT14A	1		T	PT34A	1		T
127	NC	1			NC	1		
128	VCC	-			VCC	-		
129	PT12B	1	PCLKC1_0	C	PT30B	1	PCLKC1_0	C
130	PT12A	1	PCLKT1_0	T	PT30A	1	PCLKT1_0	T
131	PT10B	0	PCLKC0_0	C	PT28B	0	PCLKC0_0	C
132	XRES	0			XRES	0		
133	GND	-			GND	-		
134	PT10A	0	PCLKT0_0	T	PT28A	0	PCLKT0_0	T
135	VCC	-			VCC	-		

LFE2-6E/SE and LFE2-12E/SE Logic Signal Connections: 144 TQFP (Cont.)

LFE2-6E/SE					LFE2-12E/12SE			
Pin Number	Pin/Pad Function	Bank	Dual Function	Differential	Pin/Pad Function	Bank	Dual Function	Differential
136	PT6B	0		C	PT16B	0		C
137	PT6A	0		T	PT16A	0		T
138	GND	-			GND	-		
139	VCCIO0	0			VCCIO0	0		
140	PT4B	0		C	PT6B	0		C
141	PT4A	0		T	PT6A	0		T
142	VCCAUX	-			VCCAUX	-		
143	PT2B	0	VREF2_0	C	PT2B	0	VREF2_0	C
144	PT2A	0	VREF1_0	T	PT2A	0	VREF1_0	T

* Supports true LVDS. Other differential signals must be emulated with external resistors.
** These dedicated input pins can be used for GPLLs or GDLLs within the respective quadrant.

Note: VCCIO and GND pads are used to determine the average DC current drawn by I/Os between GND/VCCIO connections, or between the last GND/VCCIO in an I/O bank and the end of an I/O bank. The substrate pads listed in the Pin Table do not necessarily have a one-to-one connection with a package ball or pin.

LFE2-12E/SE and LFE2-20E/SE Logic Signal Connections: 208 PQFP

LFE2-12E/SE					LFE2-20E/SE			
Pin Number	Pin/Pad Function	Bank	Dual Function	Differential	Pin/Pad Function	Bank	Dual Function	Differential
1	PL2A	7	VREF2_7	T (LVDS)*	PL2A	7	VREF2_7	T (LVDS)*
2	PL2B	7	VREF1_7	C (LVDS)*	PL2B	7	VREF1_7	C (LVDS)*
3	PL4A	7		T (LVDS)*	PL6A	7	LDQ8	T (LVDS)*
4	PL4B	7		C (LVDS)*	PL6B	7	LDQ8	C (LVDS)*
5	GND	-			GND	-		
6	PL6A	7	LDQ10	T (LVDS)*	PL12A	7	LDQ16	T (LVDS)*
7	VCCAUX	-			VCCAUX	-		
8	PL6B	7	LDQ10	C (LVDS)*	PL12B	7	LDQ16	C (LVDS)*
9	PL8A	7	LDQ10	T (LVDS)*	PL14A	7	LDQ16	T (LVDS)*
10	VCCIO7	7			VCCIO7	7		
11	PL8B	7	LDQ10	C (LVDS)*	PL14B	7	LDQ16	C (LVDS)*
12	VCC	-			VCC	-		
13	GND	-			GND	-		
14	VCCIO7	7			VCCIO7	7		
15	PL12A	7	LDQ10	T (LVDS)*	PL18A	7	LDQ16	T (LVDS)*
16	PL12B	7	LDQ10	C (LVDS)*	PL18B	7	LDQ16	C (LVDS)*
17	GND	-			GND	-		
18	PL13A	7	PCLKT7_0/LDQ10	T	PL19A	7	PCLKT7_0/LDQ16	T
19	VCC	-			VCC	-		
20	PL13B	7	PCLKC7_0/LDQ10	C	PL19B	7	PCLKC7_0/LDQ16	C
21	PL15A	6	PCLKT6_0	T (LVDS)*	PL21A	6	PCLKT6_0/LDQ25	T (LVDS)*
22	PL15B	6	PCLKC6_0	C (LVDS)*	PL21B	6	PCLKC6_0/LDQ25	C (LVDS)*
23	PL16A	6	VREF2_6	T	PL22A	6	VREF2_6/LDQ25	T
24	PL16B	6	VREF1_6	C	PL22B	6	VREF1_6/LDQ25	C
25	GND	-			GND	-		
26	PL17A	6	LLM0_GDLLT_IN_A**	T (LVDS)*	PL27A	6	LLM0_GDLLT_IN_A**/LDQ25	T (LVDS)*
27	PL17B	6	LLM0_GDLLC_IN_A**	C (LVDS)*	PL27B	6	LLM0_GDLLC_IN_A**/LDQ25	C (LVDS)*
28	VCC	-			VCC	-		
29	LLM0_PLLCAP	6			LLM0_PLLCAP	6		
30	VCCAUX	-			VCCAUX	-		
31	PL20A	6	LLM0_GPLLT_IN_A**	T (LVDS)*	PL30A	6	LLM0_GPLLT_IN_A**/LDQ34	T (LVDS)*
32	GND	-			GND	-		
33	PL21A	6	LLM0_GPLLT_FB_A	T	PL31A	6	LLM0_GPLLT_FB_A/LDQ34	T
34	PL20B	6	LLM0_GPLLC_IN_A**	C (LVDS)*	PL30B	6	LLM0_GPLLC_IN_A**/LDQ34	C (LVDS)*
35	PL21B	6	LLM0_GPLLC_FB_A	C	PL31B	6	LLM0_GPLLC_FB_A/LDQ34	C
36	PL23A	6			PL33A	6	LDQ34	
37	PL24A	6	LDQ28	T (LVDS)*	PL38A	6	LDQ42	T (LVDS)*
38	VCCIO6	6			VCCIO6	6		
39	PL24B	6	LDQ28	C (LVDS)*	PL38B	6	LDQ42	C (LVDS)*
40	VCC	-			VCC	-		
41	PL26A	6	LDQ28	T (LVDS)*	PL40A	6	LDQ42	T (LVDS)*
42	GND	-			GND	-		
43	PL26B	6	LDQ28	C (LVDS)*	PL40B	6	LDQ42	C (LVDS)*
44	VCCIO6	6			VCCIO6	6		
45	PL28A	6	LDQS28	T (LVDS)*	PL42A	6	LDQS42	T (LVDS)*

LFE2-12E/SE and LFE2-20E/SE Logic Signal Connections: 208 PQFP (Cont.)

LFE2-12E/SE					LFE2-20E/SE			
Pin Number	Pin/Pad Function	Bank	Dual Function	Differential	Pin/Pad Function	Bank	Dual Function	Differential
46	PL28B	6	LDQ28	C (LVDS)*	PL42B	6	LDQ42	C (LVDS)*
47	PL30A	6	LDQ28		PL44A	6	LDQ42	
48	TCK	-			TCK	-		
49	TDI	-			TDI	-		
50	TDO	-			TDO	-		
51	VCCJ	-			VCCJ	-		
52	TMS	-			TMS	-		
53	PB2A	5	VREF2_5/BDQ6	T	PB2A	5	VREF2_5/BDQ6	T
54	PB2B	5	VREF1_5/BDQ6	C	PB2B	5	VREF1_5/BDQ6	C
55	VCCIO5	5			VCCIO5	5		
56	PB6A	5	BDQS6	T	PB6A	5	BDQS6	T
57	PB6B	5	BDQ6	C	PB6B	5	BDQ6	C
58	PB8A	5	BDQ6	T	PB8A	5	BDQ6	T
59	PB8B	5	BDQ6	C	PB8B	5	BDQ6	C
60	GND	-			GND	-		
61	PB12A	5	BDQ15	T	PB12A	5	BDQ15	T
62	PB12B	5	BDQ15	C	PB12B	5	BDQ15	C
63	VCCIO5	5			VCCIO5	5		
64	PB16A	5	BDQ15	T	PB16A	5	BDQ15	T
65	PB16B	5	BDQ15	C	PB16B	5	BDQ15	C
66	PB18A	5	BDQ15	T	PB18A	5	BDQ15	T
67	PB18B	5	BDQ15	C	PB18B	5	BDQ15	C
68	GND	-			GND	-		
69	PB20A	5	BDQ24	T	PB30A	5	BDQ33	T
70	VCCAUX	-			VCCAUX	-		
71	PB20B	5	BDQ24	C	PB30B	5	BDQ33	C
72	PB22A	5	BDQ24	T	PB32A	5	BDQ33	T
73	PB22B	5	BDQ24	C	PB32B	5	BDQ33	C
74	VCC	-			VCC	-		
75	PB26A	5	PCLKT5_0/BDQ24	T	PB35A	5	PCLKT5_0/BDQ33	T
76	PB26B	5	PCLKC5_0/BDQ24	C	PB35B	5	PCLKC5_0/BDQ33	C
77	GND	-			GND	-		
78	PB31A	4	PCLKT4_0/BDQ33	T	PB40A	4	PCLKT4_0/BDQ42	T
79	PB31B	4	PCLKC4_0/BDQ33	C	PB40B	4	PCLKC4_0/BDQ42	C
80	VCC	-			VCC	-		
81	GND	-			GND	-		
82	PB34A	4	BDQ33	T	PB42A	4	BDQS42	T
83	PB34B	4	BDQ33	C	PB42B	4	BDQ42	C
84	PB36A	4	BDQ33	T	PB44A	4	BDQ42	T
85	PB36B	4	BDQ33	C	PB44B	4	BDQ42	C
86	VCCAUX	-			VCCAUX	-		
87	PB40A	4	BDQ42	T	PB50A	4	BDQ51	T
88	PB40B	4	BDQ42	C	PB50B	4	BDQ51	C
89	GND	-			GND	-		
90	PB42A	4	BDQS42	T	PB52A	4	BDQ51	T
91	PB42B	4	BDQ42	C	PB52B	4	BDQ51	C

LFE2-12E/SE and LFE2-20E/SE Logic Signal Connections: 208 PQFP (Cont.)

LFE2-12E/SE					LFE2-20E/SE			
Pin Number	Pin/Pad Function	Bank	Dual Function	Differential	Pin/Pad Function	Bank	Dual Function	Differential
92	PB44A	4	BDQ42	T	PB54A	4	BDQ51	T
93	VCCIO4	4			VCCIO4	4		
94	PB44B	4	BDQ42	C	PB54B	4	BDQ51	C
95	PB48A	4	BDQ51	T	PB58A	4	BDQ60	T
96	PB48B	4	BDQ51	C	PB58B	4	BDQ60	C
97	VCC	-			VCC	-		
98	PB52A	4	BDQ51	T	PB60A	4	BDQS60	T
99	PB52B	4	BDQ51	C	PB60B	4	BDQ60	C
100	VCCIO4	4			VCCIO4	4		
101	PB54A	4	BDQ51		PB63A	4	BDQ60	
102	GND	-			GND	-		
103	PB55A	4	VREF2_4/BDQ51	T	PB64A	4	VREF2_4/BDQ60	T
104	PB55B	4	VREF1_4/BDQ51	C	PB64B	4	VREF1_4/BDQ60	C
105	CFG1	8			CFG1	8		
106	PROGRAMN	8			PROGRAMN	8		
107	CFG2	8			CFG2	8		
108	INITN	8			INITN	8		
109	CFG0	8			CFG0	8		
110	CCLK	8			CCLK	8		
111	DONE	8			DONE	8		
112	PR29A	8	D0/SPIFASTN		PR43A	8	D0/SPIFASTN	
113	VCCIO8	8			VCCIO8	8		
114	PR26A	8	D6		PR40A	8	D6	
115	GND	-			GND	-		
116	VCC	-			VCC	-		
117	PR25B	8	D7/SPID0	C	PR39B	8	D7/SPID0	C
118	VCCIO8	8			VCCIO8	8		
119	PR25A	8	DI/CSSPI0N	T	PR39A	8	DI/CSSPI0N	T
120	PR24B	8	DOUT/CSON	C	PR38B	8	DOUT/CSON	C
121	PR24A	8	BUSY/SISPI	T	PR38A	8	BUSY/SISPI	T
122	GND	-			GND	-		
123	VCCIO3	3			VCCIO3	3		
124	PR21A	3	RLM0_GPLLT_FB_A		PR31A	3	RLM0_GPLLT_FB_A/RDQ34	
125	VCCAUX	-			VCCAUX	-		
126	PR20B	3	RLM0_GPLLC_IN_A**	C (LVDS)*	PR30B	3	RLM0_GPLLC_IN_A**/RDQ34	C (LVDS)*
127	PR20A	3	RLM0_GPLLT_IN_A**	T (LVDS)*	PR30A	3	RLM0_GPLLT_IN_A**/RDQ34	T (LVDS)*
128	RLM0_PLLCAP	3			RLM0_PLLCAP	3		
129	VCC	-			VCC	-		
130	PR18B	3	RLM0_GDLLC_FB_A	C	PR28B	3	RLM0_GDLLC_FB_A/RDQ25	C
131	PR18A	3	RLM0_GDLLT_FB_A	T	PR28A	3	RLM0_GDLLT_FB_A/RDQ25	T
132	PR17B	3	RLM0_GDLLC_IN_A**	C (LVDS)*	PR27B	3	RLM0_GDLLC_IN_A**/RDQ25	C (LVDS)*
133	PR17A	3	RLM0_GDLLT_IN_A**	T (LVDS)*	PR27A	3	RLM0_GDLLT_IN_A**/RDQ25	T (LVDS)*
134	PR16B	3	VREF2_3	C	PR22B	3	VREF2_3/RDQ25	C
135	VCCIO3	3			VCCIO3	3		
136	PR16A	3	VREF1_3	T	PR22A	3	VREF1_3/RDQ25	T
137	PR15B	3	PCLKC3_0	C (LVDS)*	PR21B	3	PCLKC3_0/RDQ25	C (LVDS)*

LFE2-12E/SE and LFE2-20E/SE Logic Signal Connections: 208 PQFP (Cont.)

LFE2-12E/SE					LFE2-20E/SE			
Pin Number	Pin/Pad Function	Bank	Dual Function	Differential	Pin/Pad Function	Bank	Dual Function	Differential
138	PR15A	3	PCLKT3_0	T (LVDS)*	PR21A	3	PCLKT3_0/RDQ25	T (LVDS)*
139	GND	-			GND	-		
140	VCC	-			VCC	-		
141	PR13B	2	PCLKC2_0/RDQ10	C	PR19B	2	PCLKC2_0/RDQ16	C
142	PR13A	2	PCLKT2_0/RDQ10	T	PR19A	2	PCLKT2_0/RDQ16	T
143	VCCIO2	2			VCCIO2	2		
144	PR12A	2	RDQ10		PR16A	2	RDQS16	
145	GND	-			GND	-		
146	VCC	-			VCC	-		
147	PR8B	2	RDQ10	C (LVDS)*	PR14B	2	RDQ16	C (LVDS)*
148	VCCIO2	2			VCCIO2	2		
149	PR8A	2	RDQ10	T (LVDS)*	PR14A	2	RDQ16	T (LVDS)*
150	PR6B	2	RDQ10	C (LVDS)*	PR12B	2	RDQ16	C (LVDS)*
151	VCCAUX	-			VCCAUX	-		
152	PR6A	2	RDQ10	T (LVDS)*	PR12A	2	RDQ16	T (LVDS)*
153	PR4B	2		C (LVDS)*	PR6B	2	RDQ8	C (LVDS)*
154	PR4A	2		T (LVDS)*	PR6A	2	RDQ8	T (LVDS)*
155	PR2B	2	VREF2_2	C (LVDS)*	PR2B	2	VREF2_2	C (LVDS)*
156	PR2A	2	VREF1_2	T (LVDS)*	PR2A	2	VREF1_2	T (LVDS)*
157	PT55B	1	VREF2_1	C	PT64B	1	VREF2_1	C
158	PT55A	1	VREF1_1	T	PT64A	1	VREF1_1	T
159	GND	-			GND	-		
160	PT54B	1		C	PT62B	1		C
161	PT54A	1		T	PT62A	1		T
162	VCCIO1	1			VCCIO1	1		
163	PT52B	1		C	PT60B	1		C
164	PT52A	1		T	PT60A	1		T
165	PT50B	1		C	PT58B	1		C
166	PT50A	1		T	PT58A	1		T
167	PT48B	1		C	PT56B	1		C
168	PT48A	1		T	PT56A	1		T
169	GND	-			GND	-		
170	VCCIO1	1			VCCIO1	1		
171	VCC	-			VCC	-		
172	PT40B	1		C	PT50B	1		C
173	PT40A	1		T	PT50A	1		T
174	VCCAUX	-			VCCAUX	-		
175	GND	-			GND	-		
176	PT36B	1		C	PT44B	1		C
177	PT36A	1		T	PT44A	1		T
178	PT34B	1		C	PT42B	1		C
179	PT34A	1		T	PT42A	1		T
180	PT30B	1	PCLKC1_0	C	PT39B	1	PCLKC1_0	C
181	PT30A	1	PCLKT1_0	T	PT39A	1	PCLKT1_0	T
182	XRES	1			XRES	1		
183	PT28B	0	PCLKC0_0	C	PT37B	0	PCLKC0_0	C

LFE2-12E/SE and LFE2-20E/SE Logic Signal Connections: 208 PQFP (Cont.)

LFE2-12E/SE					LFE2-20E/SE			
Pin Number	Pin/Pad Function	Bank	Dual Function	Differential	Pin/Pad Function	Bank	Dual Function	Differential
184	GND	-			GND	-		
185	PT28A	0	PCLKT0_0	T	PT37A	0	PCLKT0_0	T
186	PT26B	0		C	PT36B	0		C
187	PT26A	0		T	PT36A	0		T
188	VCC	-			VCC	-		
189	PT20B	0		C	PT30B	0		C
190	VCCAUX	-			VCCAUX	-		
191	PT20A	0		T	PT30A	0		T
192	GND	-			GND	-		
193	PT18B	0		C	PT26B	0		C
194	PT18A	0		T	PT26A	0		T
195	VCCIO0	0			VCCIO0	0		
196	PT16B	0		C	PT20B	0		C
197	PT16A	0		T	PT20A	0		T
198	VCC	-			VCC	-		
199	PT12B	0		C	PT12B	0		C
200	PT12A	0		T	PT12A	0		T
201	GND	-			GND	-		
202	PT8B	0		C	PT8B	0		C
203	PT8A	0		T	PT8A	0		T
204	PT6B	0		C	PT6B	0		C
205	PT6A	0		T	PT6A	0		T
206	VCCIO0	0			VCCIO0	0		
207	PT2B	0	VREF2_0	C	PT2B	0	VREF2_0	C
208	PT2A	0	VREF1_0	T	PT2A	0	VREF1_0	T

* Supports true LVDS. Other differential signals must be emulated with external resistors.

** These dedicated input pins can be used for GPLLs or GDLLs within the respective quadrant.

Note: VCCIO and GND pads are used to determine the average DC current drawn by I/Os between GND/VCCIO connections, or between the last GND/VCCIO in an I/O bank and the end of an I/O bank. The substrate pads listed in the Pin Table do not necessarily have a one to one connection with a package ball or pin.

LFE2-6E/SE and LFE2-12E/SE Logic Signal Connections: 256 fpBGA

LFE2-6E/SE					LFE2-12E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
C3	PL2A	7	VREF2_7	T (LVDS)*	PL2A	7	VREF2_7	T (LVDS)*
C2	PL2B	7	VREF1_7	C (LVDS)*	PL2B	7	VREF1_7	C (LVDS)*
VCCIO	VCCIO7	7			VCCIO7	7		
-	-	-			-	-		
D3	PL5A	7		T	PL5A	7		T
D4	PL4A	7		T (LVDS)*	PL4A	7		T (LVDS)*
D2	PL5B	7		C	PL5B	7		C
GND	GNDIO7	-			GNDIO7	-		
E4	PL4B	7		C (LVDS)*	PL4B	7		C (LVDS)*
B1	PL7A	7	LDQ10	T	PL7A	7	LDQ10	T
C1	PL7B	7	LDQ10	C	PL7B	7	LDQ10	C
F5	PL9A	7	LDQ10	T	PL9A	7	LDQ10	T
VCCIO	VCCIO7	7			VCCIO7	7		
F4	PL8A	7	LDQ10	T (LVDS)*	PL8A	7	LDQ10	T (LVDS)*
G6	PL9B	7	LDQ10	C	PL9B	7	LDQ10	C
G4	PL8B	7	LDQ10	C (LVDS)*	PL8B	7	LDQ10	C (LVDS)*
D1	PL10A	7	LDQS10	T (LVDS)*	PL10A	7	LDQS10	T (LVDS)*
GND	GNDIO7	-			GNDIO7	-		
E1	PL10B	7	LDQ10	C (LVDS)*	PL10B	7	LDQ10	C (LVDS)*
F3	PL11A	7	LDQ10	T	PL11A	7	LDQ10	T
G3	PL11B	7	LDQ10	C	PL11B	7	LDQ10	C
VCCIO	VCCIO7	7			VCCIO7	7		
F2	PL12A	7	LDQ10	T (LVDS)*	PL12A	7	LDQ10	T (LVDS)*
F1	PL12B	7	LDQ10	C (LVDS)*	PL12B	7	LDQ10	C (LVDS)*
GND	GNDIO7	-			GNDIO7	-		
G2	PL13A	7	PCLKT7_0/LDQ10	T	PL13A	7	PCLKT7_0/LDQ10	T
G1	PL13B	7	PCLKC7_0/LDQ10	C	PL13B	7	PCLKC7_0/LDQ10	C
H6	PL15A	6	PCLKT6_0	T (LVDS)*	PL15A	6	PCLKT6_0	T (LVDS)*
VCCIO	VCCIO6	6			VCCIO6	6		
H5	PL15B	6	PCLKC6_0	C (LVDS)*	PL15B	6	PCLKC6_0	C (LVDS)*
H4	PL16A	6	VREF2_6	T	PL16A	6	VREF2_6	T
GND	GNDIO6	-			GNDIO6	-		
H3	PL16B	6	VREF1_6	C	PL16B	6	VREF1_6	C
H2	PL17A	6	LLM0_GDLLT_IN_A**	T (LVDS)*	PL17A	6	LLM0_GDLLT_IN_A**	T (LVDS)*
H1	PL17B	6	LLM0_GDLLC_IN_A**	C (LVDS)*	PL17B	6	LLM0_GDLLC_IN_A**	C (LVDS)*
G10	VCC	-			VCC	-		
J4	PL18A	6	LLM0_GDLLT_FB_A	T	PL18A	6	LLM0_GDLLT_FB_A	T
J5	PL18B	6	LLM0_GDLLC_FB_A	C	PL18B	6	LLM0_GDLLC_FB_A	C
J6	LLM0_PLLCAP	6			LLM0_PLLCAP	6		
K4	PL20A	6	LLM0_GPLLT_IN_A**	T (LVDS)*	PL20A	6	LLM0_GPLLT_IN_A**	T (LVDS)*
GND	GNDIO6	-			GNDIO6	-		
J1	PL21A	6	LLM0_GPLLT_FB_A	T	PL21A	6	LLM0_GPLLT_FB_A	T
K3	PL20B	6	LLM0_GPLLC_IN_A**	C (LVDS)*	PL20B	6	LLM0_GPLLC_IN_A**	C (LVDS)*
VCCIO	VCCIO6	6			VCCIO6	6		
J2	PL21B	6	LLM0_GPLLC_FB_A	C	PL21B	6	LLM0_GPLLC_FB_A	C

LFE2-6E/SE and LFE2-12E/SE Logic Signal Connections: 256 fpBGA (Cont.)

LFE2-6E/SE					LFE2-12E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
GND	GNDIO6	-			GNDIO6	-		
L2	PL24A	6	LDQ28	T (LVDS)*	PL24A	6	LDQ28	T (LVDS)*
K2	PL25A	6	LDQ28	T	PL25A	6	LDQ28	T
L3	PL24B	6	LDQ28	C (LVDS)*	PL24B	6	LDQ28	C (LVDS)*
K1	PL25B	6	LDQ28	C	PL25B	6	LDQ28	C
VCCIO	VCCIO6	6			VCCIO6	6		
L4	PL26A	6	LDQ28	T (LVDS)*	PL26A	6	LDQ28	T (LVDS)*
L1	PL27A	6	LDQ28	T	PL27A	6	LDQ28	T
L5	PL26B	6	LDQ28	C (LVDS)*	PL26B	6	LDQ28	C (LVDS)*
M1	PL27B	6	LDQ28	C	PL27B	6	LDQ28	C
GND	GNDIO6	-			GNDIO6	-		
N1	PL29A	6	LDQ28	T	PL29A	6	LDQ28	T
N2	PL28A	6	LDQS28	T (LVDS)*	PL28A	6	LDQS28	T (LVDS)*
P1	PL29B	6	LDQ28	C	PL29B	6	LDQ28	C
VCCIO	VCCIO6	6			VCCIO6	6		
P2	PL28B	6	LDQ28	C (LVDS)*	PL28B	6	LDQ28	C (LVDS)*
R1	PL30A	6	LDQ28	T (LVDS)*	PL30A	6	LDQ28	T (LVDS)*
GND	GNDIO6	-			GNDIO6	-		
R2	PL30B	6	LDQ28	C (LVDS)*	PL30B	6	LDQ28	C (LVDS)*
N4	TDI	-			TDI	-		
M4	TCK	-			TCK	-		
P3	TDO	-			TDO	-		
N3	TMS	-			TMS	-		
K7	VCCJ	-			VCCJ	-		
M5	PB2A	5	VREF2_5/BDQ6	T	PB2A	5	VREF2_5/BDQ6	T
K6	NC	-			PB3A	5	BDQ6	
M6	PB2B	5	VREF1_5/BDQ6	C	PB2B	5	VREF1_5/BDQ6	C
R3	NC	-			PB5A	5	BDQ6	T
P4	NC	-			PB5B	5	BDQ6	C
-	-	-			VCCIO	5		
-	-	-			GNDIO5	5		
N5	PB3A	5	BDQ6	T	PB21A	5	BDQ24	T
N6	PB3B	5	BDQ6	C	PB21B	5	BDQ24	C
T2	PB4A	5	BDQ6	T	PB22A	5	BDQ24	T
P6	PB5A	5	BDQ6	T	PB23A	5	BDQ24	T
VCCIO	VCCIO5	5			VCCIO5	5		
T3	PB4B	5	BDQ6	C	PB22B	5	BDQ24	C
R6	PB5B	5	BDQ6	C	PB23B	5	BDQ24	C
GND	GNDIO5	-			GNDIO5	-		
R4	PB6A	5	BDQS6	T	PB24A	5	BDQS24	T
L6	PB7A	5	BDQ6	T	PB25A	5	BDQ24	T
T4	PB6B	5	BDQ6	C	PB24B	5	BDQ24	C
L7	PB7B	5	BDQ6	C	PB25B	5	BDQ24	C
N7	PB8A	5	PCLKT5_0/BDQ6	T	PB26A	5	PCLKT5_0/BDQ24	T
VCCIO	VCCIO5	5			VCCIO5	5		

LFE2-6E/SE and LFE2-12E/SE Logic Signal Connections: 256 fpBGA (Cont.)

LFE2-6E/SE					LFE2-12E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
M8	PB8B	5	PCLKC5_0/BDQ6	C	PB26B	5	PCLKC5_0/BDQ24	C
GND	GNDIO5	-			GNDIO5	-		
P7	PB13A	4	PCLKT4_0/BDQ15	T	PB31A	4	PCLKT4_0/BDQ33	T
R8	PB13B	4	PCLKC4_0/BDQ15	C	PB31B	4	PCLKC4_0/BDQ33	C
VCCIO	VCCIO4	4			VCCIO4	4		
T5	PB14A	4	BDQ15	T	PB32A	4	BDQ33	T
T6	PB14B	4	BDQ15	C	PB32B	4	BDQ33	C
T8	PB15A	4	BDQS15	T	PB33A	4	BDQS33	T
GND	GNDIO4	-			GNDIO4	-		
R7	PB16A	4	BDQ15	T	PB34A	4	BDQ33	T
T9	PB15B	4	BDQ15	C	PB33B	4	BDQ33	C
T7	PB16B	4	BDQ15	C	PB34B	4	BDQ33	C
L8	PB17A	4	BDQ15	T	PB35A	4	BDQ33	T
VCCIO	VCCIO4	4			VCCIO4	4		
P8	PB18A	4	BDQ15	T	PB36A	4	BDQ33	T
L9	PB17B	4	BDQ15	C	PB35B	4	BDQ33	C
N8	PB18B	4	BDQ15	C	PB36B	4	BDQ33	C
R9	PB19A	4	BDQ15	T	PB37A	4	BDQ33	T
GND	GNDIO4	-			GNDIO4	-		
R10	PB19B	4	BDQ15	C	PB37B	4	BDQ33	C
-	-	-			VCCIO	4		
-	-	-			GNDIO4	4		
N9	PB20A	4	BDQ24	T	PB47A	4	BDQ51	T
T10	PB21A	4	BDQ24	T	PB48A	4	BDQ51	T
M9	PB20B	4	BDQ24	C	PB47B	4	BDQ51	C
R11	PB21B	4	BDQ24	C	PB48B	4	BDQ51	C
P10	PB22A	4	BDQ24	T	PB49A	4	BDQ51	T
N11	PB23A	4	BDQ24	T	PB50A	4	BDQ51	T
VCCIO	VCCIO4	4			VCCIO4	4		
N10	PB22B	4	BDQ24	C	PB49B	4	BDQ51	C
P11	PB23B	4	BDQ24	C	PB50B	4	BDQ51	C
T11	PB24A	4	BDQS24	T	PB51A	4	BDQS51	T
GND	GNDIO4	-			GNDIO4	-		
M11	PB25A	4	BDQ24	T	PB52A	4	BDQ51	T
T12	PB24B	4	BDQ24	C	PB51B	4	BDQ51	C
L11	PB25B	4	BDQ24	C	PB52B	4	BDQ51	C
T13	PB26A	4	BDQ24	T	PB53A	4	BDQ51	T
R13	PB27A	4	BDQ24	T	PB54A	4	BDQ51	T
VCCIO	VCCIO4	4			VCCIO4	4		
T14	PB26B	4	BDQ24	C	PB53B	4	BDQ51	C
P13	PB27B	4	BDQ24	C	PB54B	4	BDQ51	C
GND	GNDIO4	-			GNDIO4	-		
N12	PB28A	4	VREF2_4/BDQ24	T	PB55A	4	VREF2_4/BDQ51	T
M12	PB28B	4	VREF1_4/BDQ24	C	PB55B	4	VREF1_4/BDQ51	C
R15	CFG2	8			CFG2	8		

LFE2-6E/SE and LFE2-12E/SE Logic Signal Connections: 256 fpBGA (Cont.)

LFE2-6E/SE					LFE2-12E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
N14	CFG1	8			CFG1	8		
N13	PROGRAMN	8			PROGRAMN	8		
N15	CFG0	8			CFG0	8		
P15	PR30B	8	WRITEN	C	PR30B	8	WRITEN	C
L12	INITN	8			INITN	8		
N16	PR29B	8	CSN	C	PR29B	8	CSN	C
GND	GNDIO8	-			GNDIO8	-		
R14	CCLK	8			CCLK	8		
P14	PR30A	8	CS1N	T	PR30A	8	CS1N	T
M13	DONE	8			DONE	8		
R16	PR28B	8	D1	C	PR28B	8	D1	C
VCCIO	VCCIO8	8			VCCIO8	8		
M16	PR29A	8	D0/SPIFASTN	T	PR29A	8	D0/SPIFASTN	T
P16	PR28A	8	D2	T	PR28A	8	D2	T
L15	PR27B	8	D3	C	PR27B	8	D3	C
GND	GNDIO8	-			GNDIO8	-		
L14	PR26A	8	D6	T	PR26A	8	D6	T
L16	PR27A	8	D4	T	PR27A	8	D4	T
L10	PR25B	8	D7/SPID0	C	PR25B	8	D7/SPID0	C
L13	PR26B	8	D5	C	PR26B	8	D5	C
VCCIO	VCCIO8	8			VCCIO8	8		
K11	PR25A	8	DI/CSSPI0N	T	PR25A	8	DI/CSSPI0N	T
K14	PR24B	8	DOUT/CSON	C	PR24B	8	DOUT/CSON	C
K13	PR24A	8	BUSY/SISPI	T	PR24A	8	BUSY/SISPI	T
GND	GNDIO8	-			GNDIO8	-		
K15	PR21B	3	RLM0_GPLLC_FB_A	C	PR21B	3	RLM0_GPLLC_FB_A	C
VCCIO	VCCIO3	3			VCCIO3	3		
K16	PR21A	3	RLM0_GPLLT_FB_A	T	PR21A	3	RLM0_GPLLT_FB_A	T
GND	GNDIO3	-			GNDIO3	-		
J16	PR20B	3	RLM0_GPLLC_IN_A**	C (LVDS)*	PR20B	3	RLM0_GPLLC_IN_A**	C (LVDS)*
J15	PR20A	3	RLM0_GPLLT_IN_A**	T (LVDS)*	PR20A	3	RLM0_GPLLT_IN_A**	T (LVDS)*
J14	RLM0_PLLCAP	3			RLM0_PLLCAP	3		
J13	PR18B	3	RLM0_GDLLC_FB_A	C	PR18B	3	RLM0_GDLLC_FB_A	C
J12	PR18A	3	RLM0_GDLLT_FB_A	T	PR18A	3	RLM0_GDLLT_FB_A	T
H12	PR17B	3	RLM0_GDLLC_IN_A**	C (LVDS)*	PR17B	3	RLM0_GDLLC_IN_A**	C (LVDS)*
GND	GNDIO3	-			GNDIO3	-		
H13	PR17A	3	RLM0_GDLLT_IN_A**	T (LVDS)*	PR17A	3	RLM0_GDLLT_IN_A**	T (LVDS)*
H15	PR16B	3	VREF2_3	C	PR16B	3	VREF2_3	C
VCCIO	VCCIO3	3			VCCIO3	3		
H16	PR16A	3	VREF1_3	T	PR16A	3	VREF1_3	T
H11	PR15B	3	PCLKC3_0	C (LVDS)*	PR15B	3	PCLKC3_0	C (LVDS)*
J11	PR15A	3	PCLKT3_0	T (LVDS)*	PR15A	3	PCLKT3_0	T (LVDS)*
G16	PR13B	2	PCLKC2_0/RDQ10	C	PR13B	2	PCLKC2_0/RDQ10	C
GND	GNDIO2	-			GNDIO2	-		
G15	PR13A	2	PCLKT2_0/RDQ10	T	PR13A	2	PCLKT2_0/RDQ10	T

LFE2-6E/SE and LFE2-12E/SE Logic Signal Connections: 256 fpBGA (Cont.)

LFE2-6E/SE					LFE2-12E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
F15	PR11B	2	RDQ10	C	PR11B	2	RDQ10	C
G11	PR12B	2	RDQ10	C (LVDS)*	PR12B	2	RDQ10	C (LVDS)*
F14	PR11A	2	RDQ10	T	PR11A	2	RDQ10	T
VCCIO	VCCIO2	2			VCCIO2	2		
F12	PR12A	2	RDQ10	T (LVDS)*	PR12A	2	RDQ10	T (LVDS)*
G14	PR10B	2	RDQ10	C (LVDS)*	PR10B	2	RDQ10	C (LVDS)*
G13	PR10A	2	RDQS10	T (LVDS)*	PR10A	2	RDQS10	T (LVDS)*
GND	GNDIO2	-			GNDIO2	-		
F16	PR8B	2	RDQ10	C (LVDS)*	PR8B	2	RDQ10	C (LVDS)*
F9	PR9B	2	RDQ10	C	PR9B	2	RDQ10	C
E16	PR8A	2	RDQ10	T (LVDS)*	PR8A	2	RDQ10	T (LVDS)*
F10	PR9A	2	RDQ10	T	PR9A	2	RDQ10	T
VCCIO	VCCIO2	2			VCCIO2	2		
D16	PR7B	2	RDQ10	C	PR7B	2	RDQ10	C
D15	PR7A	2	RDQ10	T	PR7A	2	RDQ10	T
C15	PR4B	2		C (LVDS)*	PR4B	2		C (LVDS)*
C16	PR5B	2		C	PR5B	2		C
GND	GNDIO2	-			GNDIO2	-		
D14	PR4A	2		T (LVDS)*	PR4A	2		T (LVDS)*
B16	PR5A	2		T	PR5A	2		T
F13	PR2B	2	VREF2_2	C (LVDS)*	PR2B	2	VREF2_2	C (LVDS)*
VCCIO	VCCIO2	2			VCCIO2	2		
E13	PR2A	2	VREF1_2	T (LVDS)*	PR2A	2	VREF1_2	T (LVDS)*
F11	PT28B	1	VREF2_1	C	PT55B	1	VREF2_1	C
E11	PT28A	1	VREF1_1	T	PT55A	1	VREF1_1	T
GND	GNDIO1	-			GNDIO1	-		
A15	PT27B	1		C	PT54B	1		C
E12	PT26B	1		C	PT53B	1		C
B15	PT27A	1		T	PT54A	1		T
VCCIO	VCCIO1	1			VCCIO1	1		
D12	PT26A	1		T	PT53A	1		T
B14	PT25B	1		C	PT52B	1		C
C14	PT24B	1		C	PT51B	1		C
A14	PT25A	1		T	PT52A	1		T
D13	PT24A	1		T	PT51A	1		T
C13	PT23B	1		C	PT50B	1		C
GND	GNDIO1	-			GNDIO1	-		
A13	PT22B	1		C	PT49B	1		C
B13	PT23A	1		T	PT50A	1		T
VCCIO	VCCIO1	1			VCCIO1	1		
A12	PT22A	1		T	PT49A	1		T
B11	PT21B	1		C	PT48B	1		C
D11	PT20B	1		C	PT47B	1		C
A11	PT21A	1		T	PT48A	1		T
C11	PT20A	1		T	PT47A	1		T

LFE2-6E/SE and LFE2-12E/SE Logic Signal Connections: 256 fpBGA (Cont.)

LFE2-6E/SE					LFE2-12E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
-	-	-			GNDIO1	1		
-	-	-			VCCIO	1		
D10	PT19B	1		C	PT37B	1		C
C10	PT19A	1		T	PT37A	1		T
GND	GNDIO1	-			GNDIO1	-		
B10	PT18B	1		C	PT36B	1		C
A9	PT17B	1		C	PT35B	1		C
A10	PT18A	1		T	PT36A	1		T
B9	PT17A	1		T	PT35A	1		T
VCCIO	VCCIO1	1			VCCIO1	1		
A8	PT16B	1		C	PT34B	1		C
D9	PT15B	1		C	PT33B	1		C
B8	PT16A	1		T	PT34A	1		T
C9	PT15A	1		T	PT33A	1		T
GND	GNDIO1	-			GNDIO1	-		
B7	PT14B	1		C	PT32B	1		C
E9	PT13B	1		C	PT31B	1		C
A7	PT14A	1		T	PT32A	1		T
D8	PT13A	1		T	PT31A	1		T
VCCIO	VCCIO1	1			VCCIO1	1		
A6	PT12B	1	PCLKC1_0	C	PT30B	1	PCLKC1_0	C
B6	PT12A	1	PCLKT1_0	T	PT30A	1	PCLKT1_0	T
E6	XRES	-			XRES	1		
F8	PT10B	0	PCLKC0_0	C	PT28B	0	PCLKC0_0	C
GND	GNDIO0	-			GNDIO0	-		
E8	PT10A	0	PCLKT0_0	T	PT28A	0	PCLKT0_0	T
A5	PT9B	0		C	PT27B	0		C
A3	PT8B	0		C	PT26B	0		C
A4	PT9A	0		T	PT27A	0		T
VCCIO	VCCIO0	0			VCCIO0	0		
B3	PT8A	0		T	PT26A	0		T
A2	PT7B	0		C	PT25B	0		C
C7	PT6B	0		C	PT24B	0		C
B2	PT7A	0		T	PT25A	0		T
D7	PT6A	0		T	PT24A	0		T
D6	PT5B	0		C	PT23B	0		C
GND	GNDIO0	-			GNDIO0	-		
F7	PT4B	0		C	PT22B	0		C
C6	PT5A	0		T	PT23A	0		T
VCCIO	VCCIO0	0			VCCIO0	0		
F6	PT4A	0		T	PT22A	0		T
C4	PT3B	0		C	PT21B	0		C
B4	PT3A	0		T	PT21A	0		T
-	-	-			GNDIO0	0		
-	-	-			VCCIO	0		

LFE2-6E/SE and LFE2-12E/SE Logic Signal Connections: 256 fpBGA (Cont.)

LFE2-6E/SE					LFE2-12E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
D5	PT2B	0	VREF2_0	C	PT2B	0	VREF2_0	C
E5	PT2A	0	VREF1_0	T	PT2A	0	VREF1_0	T
G7	VCC	-			VCC	-		
G9	VCC	-			VCC	-		
H7	VCC	-			VCC	-		
J10	VCC	-			VCC	-		
K10	VCC	-			VCC	-		
K8	VCC	-			VCC	-		
G8	VCCAUX	-			VCCAUX	-		
H10	VCCAUX	-			VCCAUX	-		
J7	VCCAUX	-			VCCAUX	-		
K9	VCCAUX	-			VCCAUX	-		
C5	VCCIO0	0			VCCIO0	0		
E7	VCCIO0	0			VCCIO0	0		
C12	VCCIO1	1			VCCIO1	1		
E10	VCCIO1	1			VCCIO1	1		
E14	VCCIO2	2			VCCIO2	2		
G12	VCCIO2	2			VCCIO2	2		
K12	VCCIO3	3			VCCIO3	3		
M14	VCCIO3	3			VCCIO3	3		
M10	VCCIO4	4			VCCIO4	4		
P12	VCCIO4	4			VCCIO4	4		
M7	VCCIO5	5			VCCIO5	5		
P5	VCCIO5	5			VCCIO5	5		
K5	VCCIO6	6			VCCIO6	6		
M3	VCCIO6	6			VCCIO6	6		
E3	VCCIO7	7			VCCIO7	7		
G5	VCCIO7	7			VCCIO7	7		
T15	VCCIO8	8			VCCIO8	8		
A1	GND	-			GND	-		
A16	GND	-			GND	-		
B12	GND	-			GND	-		
B5	GND	-			GND	-		
C8	GND	-			GND	-		
E15	GND	-			GND	-		
E2	GND	-			GND	-		
H14	GND	-			GND	-		
H8	GND	-			GND	-		
H9	GND	-			GND	-		
J3	GND	-			GND	-		
J8	GND	-			GND	-		
J9	GND	-			GND	-		
M15	GND	-			GND	-		
M2	GND	-			GND	-		
P9	GND	-			GND	-		

LFE2-6E/SE and LFE2-12E/SE Logic Signal Connections: 256 fpBGA (Cont.)

LFE2-6E/SE					LFE2-12E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
R12	GND	-			GND	-		
R5	GND	-			GND	-		
T1	GND	-			GND	-		
T16	GND	-			GND	-		

* Supports true LVDS. Other differential signals must be emulated with external resistors.
** These dedicated input pins can be used for GPLLs or GDLLs within the respective quadrant.

Note: VCCIO and GND pads are used to determine the average DC current drawn by I/Os between GND/VCCIO connections, or between the last GND/VCCIO in an I/O bank and the end of an I/O bank. The substrate pads listed in the Pin Table do not necessarily have a one to one connection with a package ball or pin.

LFE2-20E/SE Logic Signal Connections: 256 fpBGA

LFE2-20E/SE					
Ball Number	**Ball Number**	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
C3	C3	PL2A	7	VREF2_7	T (LVDS)*
C2	C2	PL2B	7	VREF1_7	C (LVDS)*
VCCIO	VCCIO	VCCIO7	7		
-	GND	GNDIO7	7		
D3	D3	PL7A	7	LDQ8	T
D4	D4	PL6A	7	LDQ8	T (LVDS)*
D2	D2	PL7B	7	LDQ8	C
GND	GND	GNDIO7	-		
E4	E4	PL6B	7	LDQ8	C (LVDS)*
B1	B1	PL13A	7	LDQ16	T
C1	C1	PL13B	7	LDQ16	C
F5	F5	PL15A	7	LDQ16	T
VCCIO	VCC	VCCIO	7		
F4	F4	PL14A	7	LDQ16	T (LVDS)*
G6	G6	PL15B	7	LDQ16	C
G4	G4	PL14B	7	LDQ16	C (LVDS)*
D1	D1	PL16A	7	LDQS16	T (LVDS)*
GND	GND	GNDIO7	-		
E1	E1	PL16B	7	LDQ16	C (LVDS)*
F3	F3	PL17A	7	LDQ16	T
G3	G3	PL17B	7	LDQ16	C
VCCIO	VCCIO	VCCIO7	7		
F2	F2	PL18A	7	LDQ16	T (LVDS)*
F1	F1	PL18B	7	LDQ16	C (LVDS)*
GND	GND	GNDIO7	-		
G2	G2	PL19A	7	PCLKT7_0/LDQ16	T
G1	G1	PL19B	7	PCLKC7_0/LDQ16	C
H6	H6	PL21A	6	PCLKT6_0/LDQ25	T (LVDS)*
VCCIO	VCCIO	VCCIO6	6		
H5	H5	PL21B	6	PCLKC6_0/LDQ25	C (LVDS)*
H4	H4	PL22A	6	VREF2_6/LDQ25	T
GND	GND	GNDIO6	-		
H3	H3	PL22B	6	VREF1_6/LDQ25	C
H2	H2	PL27A	6	LLM0_GDLLT_IN_A**/LDQ25	T (LVDS)*
H1	H1	PL27B	6	LLM0_GDLLC_IN_A**/LDQ25	C (LVDS)*
G10	G10	VCC	-		
J4	J4	PL28A	6	LLM0_GDLLT_FB_A/LDQ25	T
J5	J5	PL28B	6	LLM0_GDLLC_FB_A/LDQ25	C
J6	J6	LLM0_PLLCAP	6		
K4	K4	PL30A	6	LLM0_GPLLT_IN_A**/LDQ34	T (LVDS)*
GND	GND	GNDIO6	-		

LFE2-20E/SE Logic Signal Connections: 256 fpBGA (Cont.)

LFE2-20E/SE					
Ball Number	Ball Number	Ball/Pad Function	Bank	Dual Function	Differential
J1	J1	PL31A	6	LLM0_GPLLT_FB_A/LDQ34	T
K3	K3	PL30B	6	LLM0_GPLLC_IN_A**/LDQ34	C (LVDS)*
VCCIO	VCCIO	VCCIO6	6		
J2	J2	PL31B	6	LLM0_GPLLC_FB_A/LDQ34	C
GND	GND	GNDIO6	-		
L2	L2	PL38A	6	LDQ42	T (LVDS)*
K2	K2	PL39A	6	LDQ42	T
L3	L3	PL38B	6	LDQ42	C (LVDS)*
K1	K1	PL39B	6	LDQ42	C
VCCIO	VCCIO	VCCIO6	6		
L4	L4	PL40A	6	LDQ42	T (LVDS)*
L1	L1	PL41A	6	LDQ42	T
L5	L5	PL40B	6	LDQ42	C (LVDS)*
M1	M1	PL41B	6	LDQ42	C
GND	GND	GNDIO6	-		
N1	N1	PL43A	6	LDQ42	T
N2	N2	PL42A	6	LDQS42	T (LVDS)*
P1	P1	PL43B	6	LDQ42	C
VCCIO	VCCIO	VCCIO6	6		
P2	P2	PL42B	6	LDQ42	C (LVDS)*
R1	R1	PL44A	6	LDQ42	T (LVDS)*
GND	GND	GNDIO6	-		
R2	R2	PL44B	6	LDQ42	C (LVDS)*
N4	N4	TDI	-		
M4	M4	TCK	-		
P3	P3	TDO	-		
N3	N3	TMS	-		
K7	K7	VCCJ	-		
M5	M5	PB2A	5	VREF2_5/BDQ6	T
K6	K6	PB3A	5	BDQ6	
M6	M6	PB2B	5	VREF1_5/BDQ6	C
R3	R3	PB5A	5	BDQ6	T
P4	P4	PB5B	5	BDQ6	C
-	VCC	VCCIO	5		
-	GND	GNDIO5	5		
N5	N5	PB30A	5	BDQ33	T
N6	N6	PB30B	5	BDQ33	C
T2	T2	PB31A	5	BDQ33	T
P6	P6	PB32A	5	BDQ33	T
VCCIO	VCCIO	VCCIO5	5		
T3	T3	PB31B	5	BDQ33	C
R6	R6	PB32B	5	BDQ33	C

LFE2-20E/SE Logic Signal Connections: 256 fpBGA (Cont.)

LFE2-20E/SE					
Ball Number	Ball Number	Ball/Pad Function	Bank	Dual Function	Differential
GND	GND	GNDIO5	-		
R4	R4	PB33A	5	BDQS33	T
L6	L6	PB34A	5	BDQ33	T
T4	T4	PB33B	5	BDQ33	C
L7	L7	PB34B	5	BDQ33	C
N7	N7	PB35A	5	PCLKT5_0/BDQ33	T
VCCIO	VCCIO	VCCIO5	5		
M8	M8	PB35B	5	PCLKC5_0/BDQ33	C
GND	GND	GNDIO5	-		
P7	P7	PB40A	4	PCLKT4_0/BDQ42	T
R8	R8	PB40B	4	PCLKC4_0/BDQ42	C
VCCIO	VCCIO	VCCIO4	4		
T5	T5	PB41A	4	BDQ42	T
T6	T6	PB41B	4	BDQ42	C
T8	T8	PB42A	4	BDQS42	T
GND	GND	GNDIO4	-		
R7	R7	PB43A	4	BDQ42	T
T9	T9	PB42B	4	BDQ42	C
T7	T7	PB43B	4	BDQ42	C
L8	L8	PB44A	4	BDQ42	T
VCCIO	VCCIO	VCCIO4	4		
P8	P8	PB45A	4	BDQ42	T
L9	L9	PB44B	4	BDQ42	C
N8	N8	PB45B	4	BDQ42	C
R9	R9	PB46A	4	BDQ42	T
GND	GND	GNDIO4	-		
R10	R10	PB46B	4	BDQ42	C
-	VCC	VCCIO	4		
-	GND	GNDIO4	4		
N9	N9	PB56A	4	BDQ60	T
T10	T10	PB57A	4	BDQ60	T
M9	M9	PB56B	4	BDQ60	C
R11	R11	PB57B	4	BDQ60	C
P10	P10	PB58A	4	BDQ60	T
N11	N11	PB59A	4	BDQ60	T
VCCIO	VCCIO	VCCIO4	4		
N10	N10	PB58B	4	BDQ60	C
P11	P11	PB59B	4	BDQ60	C
T11	T11	PB60A	4	BDQS60	T
GND	GND	GNDIO4	-		
M11	M11	PB61A	4	BDQ60	T
T12	T12	PB60B	4	BDQ60	C

LFE2-20E/SE Logic Signal Connections: 256 fpBGA (Cont.)

LFE2-20E/SE					
Ball Number	Ball Number	Ball/Pad Function	Bank	Dual Function	Differential
L11	L11	PB61B	4	BDQ60	C
T13	T13	PB62A	4	BDQ60	T
R13	R13	PB63A	4	BDQ60	T
VCCIO	VCCIO	VCCIO4	4		
T14	T14	PB62B	4	BDQ60	C
P13	P13	PB63B	4	BDQ60	C
GND	GND	GNDIO4	-		
N12	N12	PB64A	4	VREF2_4/BDQ60	T
M12	M12	PB64B	4	VREF1_4/BDQ60	C
R15	R15	CFG2	8		
N14	N14	CFG1	8		
N13	N13	PROGRAMN	8		
N15	N15	CFG0	8		
P15	P15	PR44B	8	WRITEN	C
L12	L12	INITN	8		
N16	N16	PR43B	8	CSN	C
GND	GND	GNDIO8	-		
R14	R14	CCLK	8		
P14	P14	PR44A	8	CS1N	T
M13	M13	DONE	8		
R16	R16	PR42B	8	D1	C
VCCIO	VCCIO	VCCIO8	8		
M16	M16	PR43A	8	D0/SPIFASTN	T
P16	P16	PR42A	8	D2	T
L15	L15	PR41B	8	D3	C
GND	GND	GNDIO8	-		
L14	L14	PR40A	8	D6	T
L16	L16	PR41A	8	D4	T
L10	L10	PR39B	8	D7/SPID0	C
L13	L13	PR40B	8	D5	C
VCCIO	VCCIO	VCCIO8	8		
K11	K11	PR39A	8	DI/CSSPI0N	T
K14	K14	PR38B	8	DOUT/CSON	C
K13	K13	PR38A	8	BUSY/SISPI	T
GND	GND	GNDIO8	-		
K15	K15	PR31B	3	RLM0_GPLLC_FB_A/RDQ34	C
VCCIO	VCCIO	VCCIO3	3		
K16	K16	PR31A	3	RLM0_GPLLT_FB_A/RDQ34	T
GND	GND	GNDIO3	-		
J16	J16	PR30B	3	RLM0_GPLLC_IN_A**/RDQ34	C (LVDS)*
J15	J15	PR30A	3	RLM0_GPLLT_IN_A**/RDQ34	T (LVDS)*
J14	J14	RLM0_PLLCAP	3		

LFE2-20E/SE Logic Signal Connections: 256 fpBGA (Cont.)

LFE2-20E/SE					
Ball Number	Ball Number	Ball/Pad Function	Bank	Dual Function	Differential
J13	J13	PR28B	3	RLM0_GDLLC_FB_A/RDQ25	C
J12	J12	PR28A	3	RLM0_GDLLT_FB_A/RDQ25	T
H12	H12	PR27B	3	RLM0_GDLLC_IN_A**/RDQ25	C (LVDS)*
GND	GND	GNDIO3	-		
H13	H13	PR27A	3	RLM0_GDLLT_IN_A**/RDQ25	T (LVDS)*
H15	H15	PR22B	3	VREF2_3/RDQ25	C
VCCIO	VCCIO	VCCIO3	3		
H16	H16	PR22A	3	VREF1_3/RDQ25	T
H11	H11	PR21B	3	PCLKC3_0/RDQ25	C (LVDS)*
J11	J11	PR21A	3	PCLKT3_0/RDQ25	T (LVDS)*
G16	G16	PR19B	2	PCLKC2_0/RDQ16	C
GND	GND	GNDIO2	-		
G15	G15	PR19A	2	PCLKT2_0/RDQ16	T
F15	F15	PR17B	2	RDQ16	C
G11	G11	PR18B	2	RDQ16	C (LVDS)*
F14	F14	PR17A	2	RDQ16	T
VCCIO	VCCIO	VCCIO2	2		
F12	F12	PR18A	2	RDQ16	T (LVDS)*
G14	G14	PR16B	2	RDQ16	C (LVDS)*
G13	G13	PR16A	2	RDQS16	T (LVDS)*
GND	GND	GNDIO2	-		
F16	F16	PR14B	2	RDQ16	C (LVDS)*
F9	F9	PR15B	2	RDQ16	C
E16	E16	PR14A	2	RDQ16	T (LVDS)*
F10	F10	PR15A	2	RDQ16	T
VCCIO	VCCIO	VCCIO2	2		
D16	D16	PR13B	2	RDQ16	C
D15	D15	PR13A	2	RDQ16	T
C15	C15	PR6B	2	RDQ8	C (LVDS)*
C16	C16	PR7B	2	RDQ8	C
GND	GND	GNDIO2	-		
D14	D14	PR6A	2	RDQ8	T (LVDS)*
B16	B16	PR7A	2	RDQ8	T
F13	F13	PR2B	2	VREF2_2	C (LVDS)*
VCCIO	VCCIO	VCCIO2	2		
E13	E13	PR2A	2	VREF1_2	T (LVDS)*
F11	F11	PT64B	1	VREF2_1	C
E11	E11	PT64A	1	VREF1_1	T
GND	GND	GNDIO1	-		
A15	A15	PT63B	1		C
E12	E12	PT62B	1		C
B15	B15	PT63A	1		T

LFE2-20E/SE Logic Signal Connections: 256 fpBGA (Cont.)

LFE2-20E/SE					
Ball Number	Ball Number	Ball/Pad Function	Bank	Dual Function	Differential
VCCIO	VCCIO	VCCIO1	1		
D12	D12	PT62A	1		T
B14	B14	PT61B	1		C
C14	C14	PT60B	1		C
A14	A14	PT61A	1		T
D13	D13	PT60A	1		T
C13	C13	PT59B	1		C
GND	GND	GNDIO1	-		
A13	A13	PT58B	1		C
B13	B13	PT59A	1		T
VCCIO	VCCIO	VCCIO1	1		
A12	A12	PT58A	1		T
B11	B11	PT57B	1		C
D11	D11	PT56B	1		C
A11	A11	PT57A	1		T
C11	C11	PT56A	1		T
-	GND	GNDIO1	1		
-	VCC	VCCIO	1		
D10	D10	PT46B	1		C
C10	C10	PT46A	1		T
GND	GND	GNDIO1	-		
B10	B10	PT45B	1		C
A9	A9	PT44B	1		C
A10	A10	PT45A	1		T
B9	B9	PT44A	1		T
VCCIO	VCCIO	VCCIO1	1		
A8	A8	PT43B	1		C
D9	D9	PT42B	1		C
B8	B8	PT43A	1		T
C9	C9	PT42A	1		T
GND	GND	GNDIO1	-		
B7	B7	PT41B	1		C
E9	E9	PT40B	1		C
A7	A7	PT41A	1		T
D8	D8	PT40A	1		T
VCCIO	VCCIO	VCCIO1	1		
A6	A6	PT39B	1	PCLKC1_0	C
B6	B6	PT39A	1	PCLKT1_0	T
E6	E6	XRES	1		
F8	F8	PT37B	0	PCLKC0_0	C
GND	GND	GNDIO0	-		
E8	E8	PT37A	0	PCLKT0_0	T

LFE2-20E/SE Logic Signal Connections: 256 fpBGA (Cont.)

LFE2-20E/SE					
Ball Number	Ball Number	Ball/Pad Function	Bank	Dual Function	Differential
A5	A5	PT36B	0		C
A3	A3	PT35B	0		C
A4	A4	PT36A	0		T
VCCIO	VCCIO	VCCIO0	0		
B3	B3	PT35A	0		T
A2	A2	PT34B	0		C
C7	C7	PT33B	0		C
B2	B2	PT34A	0		T
D7	D7	PT33A	0		T
D6	D6	PT32B	0		C
GND	GND	GNDIO0	-		
F7	F7	PT31B	0		C
C6	C6	PT32A	0		T
VCCIO	VCCIO	VCCIO0	0		
F6	F6	PT31A	0		T
C4	C4	PT30B	0		C
B4	B4	PT30A	0		T
-	GND	GNDIO0	0		
-	VCC	VCCIO	0		
D5	D5	PT2B	0	VREF2_0	C
E5	E5	PT2A	0	VREF1_0	T
G7	G7	VCC	-		
G9	G9	VCC	-		
H7	H7	VCC	-		
J10	J10	VCC	-		
K10	K10	VCC	-		
K8	K8	VCC	-		
G8	G8	VCCAUX	-		
H10	H10	VCCAUX	-		
J7	J7	VCCAUX	-		
K9	K9	VCCAUX	-		
C5	C5	VCCIO0	0		
E7	E7	VCCIO0	0		
C12	C12	VCCIO1	1		
E10	E10	VCCIO1	1		
E14	E14	VCCIO2	2		
G12	G12	VCCIO2	2		
K12	K12	VCCIO3	3		
M14	M14	VCCIO3	3		
M10	M10	VCCIO4	4		
P12	P12	VCCIO4	4		
M7	M7	VCCIO5	5		

LFE2-20E/SE Logic Signal Connections: 256 fpBGA (Cont.)

LFE2-20E/SE					
Ball Number	**Ball Number**	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
P5	P5	VCCIO5	5		
K5	K5	VCCIO6	6		
M3	M3	VCCIO6	6		
E3	E3	VCCIO7	7		
G5	G5	VCCIO7	7		
T15	T15	VCCIO8	8		
A1	A1	GND	-		
A16	A16	GND	-		
B12	B12	GND	-		
B5	B5	GND	-		
C8	C8	GND	-		
E15	E15	GND	-		
E2	E2	GND	-		
H14	H14	GND	-		
H8	H8	GND	-		
H9	H9	GND	-		
J3	J3	GND	-		
J8	J8	GND	-		
J9	J9	GND	-		
M15	M15	GND	-		
M2	M2	GND	-		
P9	P9	GND	-		
R12	R12	GND	-		
R5	R5	GND	-		
T1	T1	GND	-		
T16	T16	GND	-		

* Supports true LVDS. Other differential signals must be emulated with external resistors.
** These dedicated input pins can be used for GPLLs or GDLLs within the respective quadrant.

Note: VCCIO and GND pads are used to determine the average DC current drawn by I/Os between GND/VCCIO connections, or between the last GND/VCCIO in an I/O bank and the end of an I/O bank. The substrate pads listed in the Pin Table do not necessarily have a one to one connection with a package ball or pin.

LFE2-12E/SE and LFE2-20E/SE Logic Signal Connections: 484 fpBGA

LFE2-12E/12SE					LFE2-20E/20SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
E4	PL2A	7	VREF2_7	T (LVDS)*	PL2A	7	VREF2_7	T (LVDS)*
E5	PL2B	7	VREF1_7	C (LVDS)*	PL2B	7	VREF1_7	C (LVDS)*
-	-	-			GNDIO7	-		
E3	NC	-			PL4A	7	LDQ8	T (LVDS)*
F4	PL3A	7		T	PL5A	7	LDQ8	T
F3	NC	-			PL4B	7	LDQ8	C (LVDS)*
F5	PL3B	7		C	PL5B	7	LDQ8	C
VCCIO	VCCIO7	7			VCCIO7	7		
E2	PL4A	7		T (LVDS)*	PL6A	7	LDQ8	T (LVDS)*
G6	PL5A	7		T	PL7A	7	LDQ8	T
E1	PL4B	7		C (LVDS)*	PL6B	7	LDQ8	C (LVDS)*
G7	PL5B	7		C	PL7B	7	LDQ8	C
GNDIO	GNDIO7	-			GNDIO7	-		
F1	NC	-			PL9A	7	LDQ8	T
H4	NC	-			PL8A	7	LDQS8	T (LVDS)*
F2	NC	-			PL9B	7	LDQ8	C
-	-	-			VCCIO7	7		
H5	NC	-			PL8B	7	LDQ8	C (LVDS)*
G1	NC	-			PL11A	7	LDQ8	T
G3	NC	-			PL10A	7	LDQ8	T (LVDS)*
G2	NC	-			PL11B	7	LDQ8	C
-	-	-			GNDIO	-		
G4	NC	-			PL10B	7	LDQ8	C (LVDS)*
J4	PL7A	7	LDQ10	T	PL13A	7	LDQ16	T
H1	PL6A	7	LDQ10		PL12A	7	LDQ16	T (LVDS)*
J5	PL7B	7	LDQ10	C	PL13B	7	LDQ16	C
L6	PL9A	7	LDQ10	T	PL15A	7	LDQ16	T
VCCIO	VCCIO7	7			VCCIO7	7		
J2	PL8A	7	LDQ10	T (LVDS)*	PL14A	7	LDQ16	T (LVDS)*
L5	PL9B	7	LDQ10	C	PL15B	7	LDQ16	C
J1	PL8B	7	LDQ10	C (LVDS)*	PL14B	7	LDQ16	C (LVDS)*
K3	PL10A	7	LDQS10	T (LVDS)*	PL16A	7	LDQS16	T (LVDS)*
GNDIO	GNDIO7	-			GNDIO	-		
K4	PL10B	7	LDQ10	C (LVDS)*	PL16B	7	LDQ16	C (LVDS)*
K2	PL11A	7	LDQ10	T	PL17A	7	LDQ16	T
VCCIO	VCCIO7	7			VCCIO7	7		
K1	PL11B	7	LDQ10	C	PL17B	7	LDQ16	C
L4	PL12A	7	LDQ10	T (LVDS)*	PL18A	7	LDQ16	T (LVDS)*
GNDIO	GNDIO7	-			GNDIO	-		
L3	PL12B	7	LDQ10	C (LVDS)*	PL18B	7	LDQ16	C (LVDS)*
L2	PL13A	7	PCLKT7_0/LDQ10	T	PL19A	7	PCLKT7_0/LDQ16	T
L1	PL13B	7	PCLKC7_0/LDQ10	C	PL19B	7	PCLKC7_0/LDQ16	C
M5	PL15A	6	PCLKT6_0	T (LVDS)*	PL21A	6	PCLKT6_0/LDQ25	T (LVDS)*
VCCIO	VCCIO6	6			-	-		
M6	PL15B	6	PCLKC6_0	C (LVDS)*	PL21B	6	PCLKC6_0/LDQ25	C (LVDS)*

LFE2-12E/SE and LFE2-20E/SE Logic Signal Connections: 484 fpBGA

LFE2-12E/12SE					LFE2-20E/20SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
M3	PL16A	6	VREF2_6	T	PL22A	6	VREF2_6/LDQ25	T
GNDIO	GNDIO6	-			-	-		
M4	PL16B	6	VREF1_6	C	PL22B	6	VREF1_6/LDQ25	C
-	-	-			VCCIO6	6		
N1	NC	-			PL24A	6	LDQ25	T
M2	NC	-			PL23A	6	LDQ25	T (LVDS)*
N2	NC	-			PL24B	6	LDQ25	C
M1	NC	-			PL23B	6	LDQ25	C (LVDS)*
-	-	-			GNDIO	-		
N3	NC	-			PL25A	6	LDQS25	T (LVDS)*
N5	NC	-			PL26A	6	LDQ25	T
N4	NC	-			PL25B	6	LDQ25	C (LVDS)*
-	-	-			VCCIO6	6		
P5	NC	-			PL26B	6	LDQ25	C
P1	PL17A	6	LLM0_GDLLT_IN_A**	T (LVDS)*	PL27A	6	LLM0_GDLLT_IN_A**/LDQ25	T (LVDS)*
P2	PL17B	6	LLM0_GDLLC_IN_A**	C (LVDS)*	PL27B	6	LLM0_GDLLC_IN_A**/LDQ25	C (LVDS)*
P4	PL18A	6	LLM0_GDLLT_FB_A	T	PL28A	6	LLM0_GDLLT_FB_A/LDQ25	T
-	-	-			GNDIO	-		
R4	PL18B	6	LLM0_GDLLC_FB_A	C	PL28B	6	LLM0_GDLLC_FB_A/LDQ25	C
P6	LLM0_PLLCAP	6			LLM0_PLLCAP	6		
R1	PL20A	6	LLM0_GPLLT_IN_A**	T (LVDS)*	PL30A	6	LLM0_GPLLT_IN_A**/LDQ34	T (LVDS)*
GNDIO	GNDIO6	-			-	-		
R3	PL21A	6	LLM0_GPLLT_FB_A	T	PL31A	6	LLM0_GPLLT_FB_A/LDQ34	T
R2	PL20B	6	LLM0_GPLLC_IN_A**	C (LVDS)*	PL30B	6	LLM0_GPLLC_IN_A/LDQ34	C (LVDS)*
T4	PL21B	6	LLM0_GPLLC_FB_A	C	PL31B	6	LLM0_GPLLC_FB_A/LDQ34	C
T5	PL23A	6		T	PL33A	6	LDQ34	T
VCCIO	VCCIO6	6			VCCIO6	6		
T1	PL22A	6		T (LVDS)*	PL32A	6	LDQ34	T (LVDS)*
T3	PL23B	6		C	PL33B	6	LDQ34	C
T2	PL22B	6		C (LVDS)*	PL32B	6	LDQ34	C (LVDS)*
GNDIO	GNDIO6	-			GNDIO6	-		
-	-	-			VCCIO6	6		
V1	PL25A	6	LDQ28	T	PL39A	6	LDQ42	T
-	-	-			GNDIO	-		
V2	PL25B	6	LDQ28	C	PL39B	6	LDQ42	C
U1	PL24A	6	LDQ28	T (LVDS)*	PL38A	6	LDQ42	T (LVDS)*
U3	PL27A	6	LDQ28	T	PL41A	6	LDQ42	T
VCCIO	VCCIO6	6			VCCIO6	6		
U2	PL24B	6	LDQ28	C (LVDS)*	PL38B	6	LDQ42	C (LVDS)*
U4	PL27B	6	LDQ28	C	PL41B	6	LDQ42	C
R6	PL26A	6	LDQ28	T (LVDS)*	PL40A	6	LDQ42	T (LVDS)*
R7	PL29A	6	LDQ28	T	PL43A	6	LDQ42	T
GNDIO	GNDIO6	-			GNDIO	-		
T7	PL29B	6	LDQ28	C	PL43B	6	LDQ42	C
T6	PL26B	6	LDQ28	C (LVDS)*	PL40B	6	LDQ42	C (LVDS)*

LFE2-12E/SE and LFE2-20E/SE Logic Signal Connections: 484 fpBGA

LFE2-12E/12SE					LFE2-20E/20SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
AA2	PL31A	6	LDQ28	T	PL45A	6	LDQ42	T
VCCIO	VCCIO6	6			VCCIO6	6		
Y1	PL28A	6	LDQS28	T (LVDS)*	PL42A	6	LDQS42	T (LVDS)*
AA1	PL31B	6	LDQ28	C	PL45B	6	LDQ42	C
W1	PL28B	6	LDQ28	C (LVDS)*	PL42B	6	LDQ42	C (LVDS)*
V3	PL30B	6	LDQ28	C (LVDS)*	PL44B	6	LDQ42	C (LVDS)*
GNDIO	GNDIO6	-			GNDIO	-		
V4	PL30A	6	LDQ28	T (LVDS)*	PL44A	6	LDQ42	T (LVDS)*
U5	TDI	-			TDI	-		
U7	TCK	-			TCK	-		
V6	TDO	-			TDO	-		
V5	TMS	-			TMS	-		
T8	VCCJ	-			VCCJ	-		
W4	PB3A	5	BDQ6	T	PB3A	5	BDQ6	T
Y3	PB2A	5	VREF2_5/BDQ6	T	PB2A	5	VREF2_5/BDQ6	T
W3	PB3B	5	BDQ6	C	PB3B	5	BDQ6	C
Y2	PB2B	5	VREF1_5/BDQ6	C	PB2B	5	VREF1_5/BDQ6	C
AB3	PB5A	5	BDQ6	T	PB5A	5	BDQ6	T
VCCIO	VCCIO5	5			VCCIO5	5		
W5	PB4A	5	BDQ6	T	PB4A	5	BDQ6	T
AB2	PB5B	5	BDQ6	C	PB5B	5	BDQ6	C
W6	PB4B	5	BDQ6	C	PB4B	5	BDQ6	C
AB5	PB7A	5	BDQ6	T	PB7A	5	BDQ6	T
GNDIO	GNDIO5	-			GNDIO	-		
Y4	PB6A	5	BDQS6	T	PB6A	5	BDQS6	T
AB4	PB7B	5	BDQ6	C	PB7B	5	BDQ6	C
AA3	PB6B	5	BDQ6	C	PB6B	5	BDQ6	C
AB6	PB9A	5	BDQ6	T	PB9A	5	BDQ6	T
VCCIO	VCCIO5	5			VCCIO5	5		
AA5	PB8A	5	BDQ6	T	PB8A	5	BDQ6	T
AA6	PB9B	5	BDQ6	C	PB9B	5	BDQ6	C
Y5	PB8B	5	BDQ6	C	PB8B	5	BDQ6	C
GNDIO	GNDIO5	-			GNDIO	-		
-	-	-			VCCIO5	5		
Y6	PB12A	5	BDQ15	T	PB21A	5	BDQ24	T
W7	PB11A	5	BDQ15	T	PB20A	5	BDQ24	T
Y7	PB12B	5	BDQ15	C	PB21B	5	BDQ24	C
W8	PB11B	5	BDQ15	C	PB20B	5	BDQ24	C
U8	PB14A	5	BDQ15	T	PB23A	5	BDQ24	T
VCCIO	VCCIO5	5			VCCIO5	5		
AA7	PB13A	5	BDQ15	T	PB22A	5	BDQ24	T
U9	PB14B	5	BDQ15	C	PB23B	5	BDQ24	C
AB7	PB13B	5	BDQ15	C	PB22B	5	BDQ24	C
Y8	PB16A	5	BDQ15	T	PB25A	5	BDQ24	T
GNDIO	GNDIO5	-			GNDIO	-		

LFE2-12E/SE and LFE2-20E/SE Logic Signal Connections: 484 fpBGA

LFE2-12E/12SE					LFE2-20E/20SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
W9	PB15A	5	BDQS15	T	PB24A	5	BDQS24	T
AA8	PB16B	5	BDQ15	C	PB25B	5	BDQ24	C
V9	PB15B	5	BDQ15	C	PB24B	5	BDQ24	C
AB8	PB18A	5	BDQ15	T	PB27A	5	BDQ24	T
VCCIO	VCCIO5	5			VCCIO5	5		
W10	PB17A	5	BDQ15	T	PB26A	5	BDQ24	T
AA9	PB18B	5	BDQ15	C	PB27B	5	BDQ24	C
V10	PB17B	5	BDQ15	C	PB26B	5	BDQ24	C
GNDIO	GNDIO5	-			GNDIO	-		
Y10	PB21A	5	BDQ24	T	PB30A	5	BDQ33	T
AB9	PB20A	5	BDQ24	T	PB29A	5	BDQ33	T
AA10	PB21B	5	BDQ24	C	PB30B	5	BDQ33	C
AB10	PB20B	5	BDQ24	C	PB29B	5	BDQ33	C
AB11	PB23A	5	BDQ24	T	PB32A	5	BDQ33	T
U10	PB22A	5	BDQ24	T	PB31A	5	BDQ33	T
VCCIO	VCCIO5	5			VCCIO5	5		
AA11	PB23B	5	BDQ24	C	PB32B	5	BDQ33	C
U11	PB22B	5	BDQ24	C	PB31B	5	BDQ33	C
GNDIO	GNDIO5	-			GNDIO5	-		
AB12	PB25A	5	BDQ24	T	PB34A	5	BDQ33	T
Y11	PB24A	5	BDQS24	T	PB33A	5	BDQS33	T
AA12	PB25B	5	BDQ24	C	PB34B	5	BDQ33	C
W11	PB24B	5	BDQ24	C	PB33B	5	BDQ33	C
AB13	PB26A	5	PCLKT5_0/BDQ24	T	PB35A	5	PCLKT5_0/BDQ33	T
VCCIO	VCCIO5	5			VCCIO5	5		
AB14	PB26B	5	PCLKC5_0/BDQ24	C	PB35B	5	PCLKC5_0/BDQ33	C
GNDIO	GNDIO5	-			GNDIO5	-		
Y12	PB32A	4	BDQ33	T	PB41A	4	BDQ42	T
W12	PB32B	4	BDQ33	C	PB41B	4	BDQ42	C
VCCIO	VCCIO4	4			VCCIO4	4		
U12	PB31A	4	PCLKT4_0/BDQ33	T	PB40A	4	PCLKT4_0/BDQ42	T
V12	PB31B	4	PCLKC4_0/BDQ33	C	PB40B	4	PCLKC4_0/BDQ42	C
U13	PB34A	4	BDQ33	T	PB43A	4	BDQ42	T
GNDIO	GNDIO4	-			GNDIO4	-		
AA13	PB33A	4	BDQS33	T	PB42A	4	BDQS42	T
U14	PB34B	4	BDQ33	C	PB43B	4	BDQ42	C
Y13	PB33B	4	BDQ33	C	PB42B	4	BDQ42	C
AB16	PB36A	4	BDQ33	T	PB45A	4	BDQ42	T
VCCIO	VCCIO4	4			VCCIO4	4		
AB15	PB35A	4	BDQ33	T	PB44A	4	BDQ42	T
AB17	PB36B	4	BDQ33	C	PB45B	4	BDQ42	C
AA14	PB35B	4	BDQ33	C	PB44B	4	BDQ42	C
W13	PB37A	4	BDQ33	T	PB46A	4	BDQ42	T
GNDIO	GNDIO4	-			GNDIO4	-		
W14	PB37B	4	BDQ33	C	PB46B	4	BDQ42	C

LFE2-12E/SE and LFE2-20E/SE Logic Signal Connections: 484 fpBGA

LFE2-12E/12SE					LFE2-20E/20SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
AB18	PB39A	4	BDQ42	T	PB48A	4	BDQ51	T
AB19	PB39B	4	BDQ42	C	PB48B	4	BDQ51	C
Y15	PB41A	4	BDQ42	T	PB50A	4	BDQ51	T
V14	PB40A	4	BDQ42	T	PB49A	4	BDQ51	T
VCCIO	VCCIO4	4			VCCIO4	4		
AA15	PB41B	4	BDQ42	C	PB50B	4	BDQ51	C
W15	PB40B	4	BDQ42	C	PB49B	4	BDQ51	C
GNDIO	GNDIO4	-			GNDIO	-		
AB20	PB43A	4	BDQ42	T	PB52A	4	BDQ51	T
AA16	PB42A	4	BDQS42	T	PB51A	4	BDQS51	T
AB21	PB43B	4	BDQ42	C	PB52B	4	BDQ51	C
AA17	PB42B	4	BDQ42	C	PB51B	4	BDQ51	C
Y16	PB45A	4	BDQ42	T	PB54A	4	BDQ51	T
U15	PB44A	4	BDQ42	T	PB53A	4	BDQ51	T
VCCIO	VCCIO4	4			VCCIO4	4		
W16	PB45B	4	BDQ42	C	PB54B	4	BDQ51	C
U16	PB44B	4	BDQ42	C	PB53B	4	BDQ51	C
AA18	PB46A	4	BDQ42	T	PB55A	4	BDQ51	T
AA20	PB46B	4	BDQ42	C	PB55B	4	BDQ51	C
GNDIO	GNDIO4	-			GNDIO	-		
V16	PB49A	4	BDQ51	T	PB58A	4	BDQ60	T
V17	PB49B	4	BDQ51	C	PB58B	4	BDQ60	C
AA21	PB48A	4	BDQ51	T	PB57A	4	BDQ60	T
VCCIO	VCCIO4	4			VCCIO4	4		
Y19	PB51A	4	BDQS51	T	PB60A	4	BDQS60	T
AA22	PB48B	4	BDQ51	C	PB57B	4	BDQ60	C
Y20	PB51B	4	BDQ51	C	PB60B	4	BDQ60	C
Y18	PB50A	4	BDQ51	T	PB59A	4	BDQ60	T
GNDIO	GNDIO4	-			GNDIO4	-		
Y21	PB53A	4	BDQ51	T	PB62A	4	BDQ60	T
Y17	PB50B	4	BDQ51	C	PB59B	4	BDQ60	C
Y22	PB53B	4	BDQ51	C	PB62B	4	BDQ60	C
W17	PB52A	4	BDQ51	T	PB61A	4	BDQ60	T
VCCIO	VCCIO4	4			VCCIO4	4		
U18	PB54A	4	BDQ51	T	PB63A	4	BDQ60	T
W18	PB52B	4	BDQ51	C	PB61B	4	BDQ60	C
V18	PB54B	4	BDQ51	C	PB63B	4	BDQ60	C
GNDIO	GNDIO4	-			GNDIO4	-		
T15	PB55A	4	VREF2_4/BDQ51	T	PB64A	4	VREF2_4/BDQ60	T
T16	PB55B	4	VREF1_4/BDQ51	C	PB64B	4	VREF1_4/BDQ60	C
W19	CFG2	8			CFG2	8		
V19	CFG1	8			CFG1	8		
V20	PROGRAMN	8			PROGRAMN	8		
W20	CFG0	8			CFG0	8		
U22	PR28B	8	D1	C	PR42B	8	D1	C

LFE2-12E/SE and LFE2-20E/SE Logic Signal Connections: 484 fpBGA

LFE2-12E/12SE					LFE2-20E/20SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
V22	INITN	8			INITN	8		
R16	PR30B	8	WRITEN	C	PR44B	8	WRITEN	C
GNDIO	GNDIO8	-			GNDIO8	-		
W22	CCLK	8			CCLK	8		
R17	PR30A	8	CS1N	T	PR44A	8	CS1N	T
V21	DONE	8			DONE	8		
VCCIO	VCCIO8	8			VCCIO8	8		
U19	PR29B	8	CSN	C	PR43B	8	CSN	C
T17	PR26B	8	D5	C	PR40B	8	D5	C
U20	PR29A	8	D0/SPIFASTN	T	PR43A	8	D0/SPIFASTN	T
U21	PR28A	8	D2	T	PR42A	8	D2	T
GNDIO	GNDIO8	-			GNDIO8	-		
T18	PR26A	8	D6	T	PR40A	8	D6	T
T20	PR27B	8	D3	C	PR41B	8	D3	C
T21	PR25B	8	D7/SPID0	C	PR39B	8	D7/SPID0	C
T19	PR27A	8	D4	T	PR41A	8	D4	T
VCCIO	VCCIO8	8			VCCIO8	8		
T22	PR25A	8	DI/CSSPI0N	T	PR39A	8	DI/CSSPI0N	T
R18	PR24B	8	DOUT/CSON	C	PR38B	8	DOUT/CSON	C
R19	PR24A	8	BUSY/SISPI	T	PR38A	8	BUSY/SISPI	T
-	-	-			VCCIO3	3		
GNDIO	GNDIO3	-			GNDIO3	-		
P18	PR22B	3		C (LVDS)*	PR32B	3	RDQ34	C (LVDS)*
R22	PR23B	3		C	PR33B	3	RDQ34	C
P19	PR22A	3		T (LVDS)*	PR32A	3	RDQ34	T (LVDS)*
R21	PR23A	3		T	PR33A	3	RDQ34	T
VCCIO	VCCIO3	3			VCCIO3	3		
R20	PR21B	3	RLM0_GPLLC_FB_A	C	PR31B	3	RLM0_GPLLC_FB_A/RDQ34	C
P22	PR21A	3	RLM0_GPLLT_FB_A	T	PR31A	3	RLM0_GPLLT_FB_A/RDQ34	T
P21	PR20B	3	RLM0_GPLLC_IN_A**	C (LVDS)*	PR30B	3	RLM0_GPLLC_IN_A**/RDQ34	C (LVDS)*
N21	PR20A	3	RLM0_GPLLT_IN_A**	T (LVDS)*	PR30A	3	RLM0_GPLLT_IN_A**/RDQ34	T (LVDS)*
N17	RLM0_PLLCAP	3			RLM0_PLLCAP	3		
N22	PR18B	3	RLM0_GDLLC_FB_A	C	PR28B	3	RLM0_GDLLC_FB_A/RDQ25	C
M22	PR17B	3	RLM0_GDLLC_IN_A**	C (LVDS)*	PR27B	3	RLM0_GDLLC_IN_A**/RDQ25	C (LVDS)*
GNDIO	GNDIO3	-			GNDIO3	-		
N20	PR18A	3	RLM0_GDLLT_FB_A	T	PR28A	3	RLM0_GDLLT_FB_A/RDQ25	T
M21	PR17A	3	RLM0_GDLLT_IN_A**	T (LVDS)*	PR27A	3	RLM0_GDLLT_IN_A**/RDQ25	T (LVDS)*
N19	NC	-			PR26B	3	RDQ25	C
-	-	-			VCCIO3	3		
M19	NC	-			PR26A	3	RDQ25	T
J22	NC	-			PR23B	3	RDQ25	C (LVDS)*
-	-	-			GNDIO	-		
L22	NC	-			PR24B	3	RDQ25	C
H22	NC	-			PR23A	3	RDQ25	T (LVDS)*
K22	NC	-			PR24A	3	RDQ25	T

LFE2-12E/SE and LFE2-20E/SE Logic Signal Connections: 484 fpBGA

LFE2-12E/12SE					LFE2-20E/20SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
M20	PR16B	3	VREF2_3	C	PR22B	3	VREF2_3/RDQ25	C
VCCIO	VCCIO3	3			VCCIO3	3		
L21	PR16A	3	VREF1_3	T	PR22A	3	VREF1_3/RDQ25	T
K21	PR15B	3	PCLKC3_0	C (LVDS)*	PR21B	3	PCLKC3_0/RDQ25	C (LVDS)*
J21	PR15A	3	PCLKT3_0	T (LVDS)*	PR21A	3	PCLKT3_0/RDQ25	T (LVDS)*
M18	PR13B	2	PCLKC2_0/RDQ10	C	PR19B	2	PCLKC2_0/RDQ16	C
GNDIO	GNDIO2	-			GNDIO2	-		
L17	PR13A	2	PCLKT2_0/RDQ10	T	PR19A	2	PCLKT2_0/RDQ16	T
L19	PR12B	2	RDQ10	C (LVDS)*	PR18B	2	RDQ16	C (LVDS)*
K18	PR10B	2	RDQ10	C (LVDS)*	PR16B	2	RDQ16	C (LVDS)*
L20	PR12A	2	RDQ10	T (LVDS)*	PR18A	2	RDQ16	T (LVDS)*
VCCIO	VCCIO2	2			VCCIO2	2		
K19	PR10A	2	RDQS10	T (LVDS)*	PR16A	2	RDQS16	T (LVDS)*
L18	PR11B	2	RDQ10	C	PR17B	2	RDQ16	C
K17	PR11A	2	RDQ10	T	PR17A	2	RDQ16	T
GNDIO	GNDIO2	-			GNDIO2	-		
J17	PR8B	2	RDQ10	C (LVDS)*	PR14B	2	RDQ16	C (LVDS)*
G22	PR9B	2	RDQ10	C	PR15B	2	RDQ16	C
J18	PR8A	2	RDQ10	T (LVDS)*	PR14A	2	RDQ16	T (LVDS)*
F22	PR9A	2	RDQ10	T	PR15A	2	RDQ16	T
VCCIO	VCCIO2	2			VCCIO2	2		
H21	PR6B	2	RDQ10	C (LVDS)*	PR12B	2	RDQ16	C (LVDS)*
K20	PR7B	2	RDQ10	C	PR13B	2	RDQ16	C
G21	PR6A	2	RDQ10	T (LVDS)*	PR12A	2	RDQ16	T (LVDS)*
J19	PR7A	2	RDQ10	T	PR13A	2	RDQ16	T
D22	NC	-			PR10B	2	RDQ8	C (LVDS)*
F21	NC	-			PR11B	2	RDQ8	C
-	-	-			GNDIO	-		
E21	NC	-			PR10A	2	RDQ8	T (LVDS)*
E22	NC	-			PR11A	2	RDQ8	T
H19	NC	-			PR8B	2	RDQ8	C (LVDS)*
G20	NC	-			PR9B	2	RDQ8	C
-	-	-			VCCIO2	2		
G19	NC	-			PR8A	2	RDQS8	T (LVDS)*
F20	NC	-			PR9A	2	RDQ8	T
G17	PR5B	2		C	PR7B	2	RDQ8	C
GNDIO	GNDIO2	-			GNDIO2	-		
E20	PR4B	2		C (LVDS)*	PR6B	2	RDQ8	C (LVDS)*
F19	PR5A	2		T	PR7A	2	RDQ8	T
D20	PR4A	2		T (LVDS)*	PR6A	2	RDQ8	T (LVDS)*
F18	PR3B	2		C	PR5B	2	RDQ8	C
VCCIO	VCCIO2	2			VCCIO2	2		
C21	NC	-			PR4B	2	RDQ8	C (LVDS)*
F16	PR3A	2		T	PR5A	2	RDQ8	T
C22	NC	-			PR4A	2	RDQ8	T (LVDS)*

LFE2-12E/SE and LFE2-20E/SE Logic Signal Connections: 484 fpBGA

LFE2-12E/12SE					LFE2-20E/20SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
-	-	-			GNDIO	-		
D19	PR2B	2	VREF2_2	C (LVDS)*	PR2B	2	VREF2_2	C (LVDS)*
E19	PR2A	2	VREF1_2	T (LVDS)*	PR2A	2	VREF1_2	T (LVDS)*
B21	PT55B	1	VREF2_1	C	PT64B	1	VREF2_1	C
B22	PT55A	1	VREF1_1	T	PT64A	1	VREF1_1	T
GNDIO	GNDIO1	-			GNDIO1	-		
D18	PT53B	1		C	PT62B	1		C
C20	PT54B	1		C	PT63B	1		C
E18	PT53A	1		T	PT62A	1		T
C19	PT54A	1		T	PT63A	1		T
VCCIO	VCCIO1	1			VCCIO1	1		
D17	PT51B	1		C	PT60B	1		C
B20	PT52B	1		C	PT61B	1		C
C18	PT51A	1		T	PT60A	1		T
A19	PT52A	1		T	PT61A	1		T
GNDIO	GNDIO1	-			GNDIO1	-		
A18	PT49B	1		C	PT58B	1		C
A21	PT50B	1		C	PT59B	1		C
B18	PT49A	1		T	PT58A	1		T
A20	PT50A	1		T	PT59A	1		T
VCCIO	VCCIO1	1			VCCIO1	1		
D16	PT47B	1		C	PT56B	1		C
G16	PT48B	1		C	PT57B	1		C
E16	PT47A	1		T	PT56A	1		T
G15	PT48A	1		T	PT57A	1		T
C17	PT46B	1		C	PT55B	1		C
GNDIO	GNDIO1	-			GNDIO1	-		
C16	PT46A	1		T	PT55A	1		T
A17	PT44B	1		C	PT53B	1		C
B17	PT45B	1		C	PT54B	1		C
A16	PT44A	1		T	PT53A	1		T
VCCIO	VCCIO1	1			VCCIO1	1		
B16	PT45A	1		T	PT54A	1		T
E15	PT42B	1		C	PT51B	1		C
C15	PT43B	1		C	PT52B	1		C
F15	PT42A	1		T	PT51A	1		T
D15	PT43A	1		T	PT52A	1		T
B15	PT40B	1		C	PT49B	1		C
GNDIO	GNDIO1	-			GNDIO1	-		
A15	PT40A	1		T	PT49A	1		T
VCCIO	VCCIO1	1			VCCIO1	1		
A14	PT39A	1		T	PT48A	1		T
B14	PT39B	1		C	PT48B	1		C
D14	PT37B	1		C	PT46B	1		C
E14	PT36B	1		C	PT45B	1		C

LFE2-12E/SE and LFE2-20E/SE Logic Signal Connections: 484 fpBGA

LFE2-12E/12SE					LFE2-20E/20SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
GNDIO	GNDIO1	-			GNDIO1	-		
C13	PT37A	1		T	PT46A	1		T
F14	PT36A	1		T	PT45A	1		T
A13	PT35B	1		C	PT44B	1		C
E13	PT34B	1		C	PT43B	1		C
VCCIO	VCCIO1	1			VCCIO1	1		
B13	PT35A	1		T	PT44A	1		T
D13	PT34A	1		T	PT43A	1		T
E12	PT33B	1		C	PT42B	1		C
GNDIO	GNDIO1	-			GNDIO1	-		
D12	PT33A	1		T	PT42A	1		T
A12	PT31B	1		C	PT40B	1		C
B12	PT30B	1	PCLKC1_0	C	PT39B	1	PCLKC1_0	C
VCCIO	VCCIO1	1			VCCIO1	1		
A11	PT31A	1		T	PT40A	1		T
C12	PT30A	1	PCLKT1_0	T	PT39A	1	PCLKT1_0	T
F12	XRES	1			XRES	1		
B10	PT28B	0	PCLKC0_0	C	PT37B	0	PCLKC0_0	C
GNDIO	GNDIO0	-			GNDIO0	-		
B11	PT28A	0	PCLKT0_0	T	PT37A	0	PCLKT0_0	T
C11	PT26B	0		C	PT35B	0		C
A10	PT27B	0		C	PT36B	0		C
C10	PT26A	0		T	PT35A	0		T
VCCIO	VCCIO0	0			VCCIO0	0		
A9	PT27A	0		T	PT36A	0		T
A8	PT24B	0		C	PT33B	0		C
E11	PT25B	0		C	PT34B	0		C
A7	PT24A	0		T	PT33A	0		T
F11	PT25A	0		T	PT34A	0		T
GNDIO	GNDIO0	-			GNDIO0	-		
B8	PT23B	0		C	PT32B	0		C
VCCIO	VCCIO0	0			VCCIO0	0		
B9	PT23A	0		T	PT32A	0		T
C8	PT20B	0		C	PT29B	0		C
B7	PT21B	0		C	PT30B	0		C
D8	PT20A	0		T	PT29A	0		T
A6	PT21A	0		T	PT30A	0		T
GNDIO	GNDIO0	-			GNDIO0	-		
C7	PT17B	0		C	PT26B	0		C
D10	PT18B	0		C	PT27B	0		C
C6	PT17A	0		T	PT26A	0		T
E10	PT18A	0		T	PT27A	0		T
VCCIO	VCCIO0	0			VCCIO0	0		
F10	PT15B	0		C	PT24B	0		C
B6	PT16B	0		C	PT25B	0		C

LFE2-12E/SE and LFE2-20E/SE Logic Signal Connections: 484 fpBGA

LFE2-12E/12SE					LFE2-20E/20SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
D9	PT15A	0		T	PT24A	0		T
B5	PT16A	0		T	PT25A	0		T
GNDIO	GNDIO0	-			GNDIO0	-		
A5	PT13B	0		C	PT22B	0		C
F9	PT14B	0		C	PT23B	0		C
A4	PT13A	0		T	PT22A	0		T
E9	PT14A	0		T	PT23A	0		T
VCCIO	VCCIO0	0			VCCIO0	0		
G8	PT11B	0		C	PT20B	0		C
A3	PT12B	0		C	PT21B	0		C
E8	PT11A	0		T	PT20A	0		T
A2	PT12A	0		T	PT21A	0		T
GNDIO	GNDIO0	-			GNDIO0	-		
-	-	-			VCCIO0	0		
C3	PT10B	0		C	PT10B	0		C
B3	PT10A	0		T	PT10A	0		T
-	-	-			GNDIO0	-		
E7	PT8B	0		C	PT8B	0		C
F8	PT9B	0		C	PT9B	0		C
F7	PT8A	0		T	PT8A	0		T
D7	PT9A	0		T	PT9A	0		T
VCCIO	VCCIO0	0			VCCIO0	0		
D4	PT6B	0		C	PT6B	0		C
D5	PT7B	0		C	PT7B	0		C
C4	PT6A	0		T	PT6A	0		T
D6	PT7A	0		T	PT7A	0		T
GNDIO	GNDIO0	-			GNDIO	-		
J7	PT4B	0		C	PT4B	0		C
B2	PT5B	0		C	PT5B	0		C
H7	PT4A	0		T	PT4A	0		T
B1	PT5A	0		T	PT5A	0		T
VCCIO	VCCIO0	0			VCCIO0	0		
D1	PT2B	0	VREF2_0	C	PT2B	0	VREF2_0	C
D3	PT3B	0		C	PT3B	0		C
C1	PT2A	0	VREF1_0	T	PT2A	0	VREF1_0	T
C2	PT3A	0		T	PT3A	0		T
J10	VCC	-			VCC	-		
J11	VCC	-			VCC	-		
J12	VCC	-			VCC	-		
J13	VCC	-			VCC	-		
K14	VCC	-			VCC	-		
K9	VCC	-			VCC	-		
L14	VCC	-			VCC	-		
L9	VCC	-			VCC	-		
M14	VCC	-			VCC	-		

LFE2-12E/SE and LFE2-20E/SE Logic Signal Connections: 484 fpBGA

LFE2-12E/12SE					LFE2-20E/20SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
M9	VCC	-			VCC	-		
N14	VCC	-			VCC	-		
N9	VCC	-			VCC	-		
P10	VCC	-			VCC	-		
P11	VCC	-			VCC	-		
P12	VCC	-			VCC	-		
P13	VCC	-			VCC	-		
G10	VCCIO0	0			VCCIO0	0		
G9	VCCIO0	0			VCCIO0	0		
H9	VCCIO0	0			VCCIO0	0		
H8	VCCIO0	0			VCCIO0	0		
G11	VCCIO1	1			VCCIO1	1		
G12	VCCIO1	1			VCCIO1	1		
G13	VCCIO1	1			VCCIO1	1		
G14	VCCIO1	1			VCCIO1	1		
H14	VCCIO2	2			VCCIO2	2		
H15	VCCIO2	2			VCCIO2	2		
J15	VCCIO2	2			VCCIO2	2		
K16	VCCIO2	2			VCCIO2	2		
L16	VCCIO3	3			VCCIO3	3		
M16	VCCIO3	3			VCCIO3	3		
N16	VCCIO3	3			VCCIO3	3		
P16	VCCIO3	3			VCCIO3	3		
R14	VCCIO4	4			VCCIO4	4		
T12	VCCIO4	4			VCCIO4	4		
T13	VCCIO4	4			VCCIO4	4		
T14	VCCIO4	4			VCCIO4	4		
R9	VCCIO5	5			VCCIO5	5		
T10	VCCIO5	5			VCCIO5	5		
T11	VCCIO5	5			VCCIO5	5		
T9	VCCIO5	5			VCCIO5	5		
N7	VCCIO6	6			VCCIO6	6		
P7	VCCIO6	6			VCCIO6	6		
P8	VCCIO6	6			VCCIO6	6		
R8	VCCIO6	6			VCCIO6	6		
J8	VCCIO7	7			VCCIO7	7		
K7	VCCIO7	7			VCCIO7	7		
L7	VCCIO7	7			VCCIO7	7		
M7	VCCIO7	7			VCCIO7	7		
P15	VCCIO8	8			VCCIO8	8		
R15	VCCIO8	8			VCCIO8	8		
C5	VCCAUX	-			VCCAUX	-		
D11	VCCAUX	-			VCCAUX	-		
E17	VCCAUX	-			VCCAUX	-		
E6	VCCAUX	-			VCCAUX	-		

LFE2-12E/SE and LFE2-20E/SE Logic Signal Connections: 484 fpBGA

LFE2-12E/12SE					LFE2-20E/20SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
F13	VCCAUX	-			VCCAUX	-		
G18	VCCAUX	-			VCCAUX	-		
G5	VCCAUX	-			VCCAUX	-		
K5	VCCAUX	-			VCCAUX	-		
M17	VCCAUX	-			VCCAUX	-		
P17	VCCAUX	-			VCCAUX	-		
R5	VCCAUX	-			VCCAUX	-		
V11	VCCAUX	-			VCCAUX	-		
V13	VCCAUX	-			VCCAUX	-		
V15	VCCAUX	-			VCCAUX	-		
V7	VCCAUX	-			VCCAUX	-		
V8	VCCAUX	-			VCCAUX	-		
A1	GND	-			GND	-		
A22	GND	-			GND	-		
AA19	GND	-			GND	-		
AA4	GND	-			GND	-		
AB1	GND	-			GND	-		
AB22	GND	-			GND	-		
B19	GND	-			GND	-		
B4	GND	-			GND	-		
C14	GND	-			GND	-		
C9	GND	-			GND	-		
D2	GND	-			GND	-		
D21	GND	-			GND	-		
F17	GND	-			GND	-		
F6	GND	-			GND	-		
H10	GND	-			GND	-		
H11	GND	-			GND	-		
H12	GND	-			GND	-		
H13	GND	-			GND	-		
J14	GND	-			GND	-		
J20	GND	-			GND	-		
J3	GND	-			GND	-		
J9	GND	-			GND	-		
K10	GND	-			GND	-		
K11	GND	-			GND	-		
K12	GND	-			GND	-		
K13	GND	-			GND	-		
K15	GND	-			GND	-		
K8	GND	-			GND	-		
L10	GND	-			GND	-		
L11	GND	-			GND	-		
L12	GND	-			GND	-		
L13	GND	-			GND	-		
L15	GND	-			GND	-		

LFE2-12E/SE and LFE2-20E/SE Logic Signal Connections: 484 fpBGA

LFE2-12E/12SE					LFE2-20E/20SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
L8	GND	-			GND	-		
M10	GND	-			GND	-		
M11	GND	-			GND	-		
M12	GND	-			GND	-		
M13	GND	-			GND	-		
M15	GND	-			GND	-		
M8	GND	-			GND	-		
N10	GND	-			GND	-		
N11	GND	-			GND	-		
N12	GND	-			GND	-		
N13	GND	-			GND	-		
N15	GND	-			GND	-		
N8	GND	-			GND	-		
P14	GND	-			GND	-		
P20	GND	-			GND	-		
P3	GND	-			GND	-		
P9	GND	-			GND	-		
R10	GND	-			GND	-		
R11	GND	-			GND	-		
R12	GND	-			GND	-		
R13	GND	-			GND	-		
U17	GND	-			GND	-		
U6	GND	-			GND	-		
W2	GND	-			GND	-		
W21	GND	-			GND	-		
Y14	GND	-			GND	-		
Y9	GND	-			GND	-		
H6	NC	-			NC	-		
J6	NC	-			NC	-		
H3	NC	-			NC	-		
H2	NC	-			NC	-		
H17	NC	-			NC	-		
H16	NC	-			NC	-		
H20	NC	-			NC	-		
H18	NC	-			NC	-		
K6	NC	-			NC	-		
J16	NC	-			NC	-		
N18	VCC	-			VCC	-		
N6	VCC	-			VCC	-		

* Supports true LVDS. Other differential signals must be emulated with external resistors.
** These dedicated input pins can be used for GPLLs or GDLLs within the respective quadrant.

Note: VCCIO and GND pads are used to determine the average DC current drawn by I/Os between GND/VCCIO connections, or between the last GND/VCCIO in an I/O bank and the end of an I/O bank. The substrate pads listed in the Pin Table do not necessarily have a one to one connection with a package ball or pin.

LFE2-35E/SE and LFE2-50E/SE Logic Signal Connections: 484 fpBGA

LFE2-35E/SE					LFE2-50E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
E4	PL2A	7	VREF2_7/LDQ6	T (LVDS)*	PL2A	7	VREF2_7	T (LVDS)*
E5	PL2B	7	VREF1_7/LDQ6	C (LVDS)*	PL2B	7	VREF1_7	C (LVDS)*
VCCIO	VCCIO7	-			GNDIO7	-		
GNDIO	GNDIO7	-			VCCIO	7		
E3	PL10A	7	LDQ14	T (LVDS)*	PL12A	7	LDQ16	T (LVDS)*
F3	PL10B	7	LDQ14	C (LVDS)*	PL12B	7	LDQ16	C (LVDS)*
F4	PL11A	7	LDQ14	T	PL13A	7	LDQ16	T
F5	PL11B	7	LDQ14	C	PL13B	7	LDQ16	C
E2	PL12A	7	LDQ14	T (LVDS)*	PL14A	7	LDQ16	T (LVDS)*
VCCIO	VCCIO7	7			VCCIO	7		
E1	PL12B	7	LDQ14	C (LVDS)*	PL14B	7	LDQ16	C (LVDS)*
G6	PL13A	7	LDQ14	T	PL15A	7	LDQ16	T
G7	PL13B	7	LDQ14	C	PL15B	7	LDQ16	C
H4	PL14A	7	LDQS14	T (LVDS)*	PL16A	7	LDQS16	T (LVDS)*
GNDIO	GNDIO7	-			GNDIO7	-		
H5	PL14B	7	LDQ14	C (LVDS)*	PL16B	7	LDQ16	C (LVDS)*
F1	PL15A	7	LDQ14	T	PL17A	7	LDQ16	T
F2	PL15B	7	LDQ14	C	PL17B	7	LDQ16	C
VCCIO	VCCIO7	7			VCCIO	7		
G3	PL16A	7	LDQ14	T (LVDS)*	PL18A	7	LDQ16	T (LVDS)*
G4	PL16B	7	LDQ14	C (LVDS)*	PL18B	7	LDQ16	C (LVDS)*
G1	PL17A	7	LDQ14	T	PL19A	7	LDQ16	T
G2	PL17B	7	LDQ14	C	PL19B	7	LDQ16	C
GNDIO	GNDIO7	-			GNDIO7	-		
-	-	-			VCCIO	7		
H6	NC	-			PL25A	7	LUM0_SPLLT_IN_A/LDQ24	T
-	-	-			VCCIO	7		
J6	NC	-			PL25B	7	LUM0_SPLLC_IN_A/LDQ24	C
H3	NC	-			PL26A	7	LUM0_SPLLT_FB_A/LDQ24	T
H2	NC	-			PL26B	7	LUM0_SPLLC_FB_A/LDQ24	C
-	-	-			GNDIO7	-		
-	-	-			VCCIO	7		
H1	PL18A	7	LDQ22		PL37A	7	LDQ41	
J4	PL19A	7	LDQ22	T	PL38A	7	LDQ41	T
J5	PL19B	7	LDQ22	C	PL38B	7	LDQ41	C
VCCIO	VCCIO7	7			VCCIO	7		
J2	PL20A	7	LDQ22	T (LVDS)*	PL39A	7	LDQ41	T (LVDS)*
J1	PL20B	7	LDQ22	C (LVDS)*	PL39B	7	LDQ41	C (LVDS)*
L6	PL21A	7	LDQ22	T	PL40A	7	LDQ41	T
L5	PL21B	7	LDQ22	C	PL40B	7	LDQ41	C
GNDIO	GNDIO7	-			GNDIO7	-		
K3	PL22A	7	LDQS22	T (LVDS)*	PL41A	7	LDQS41	T (LVDS)*
K4	PL22B	7	LDQ22	C (LVDS)*	PL41B	7	LDQ41	C (LVDS)*
K2	PL23A	7	LDQ22	T	PL42A	7	LDQ41	T
VCCIO	VCCIO7	7			VCCIO	7		
K1	PL23B	7	LDQ22	C	PL42B	7	LDQ41	C
L4	PL24A	7	LDQ22	T (LVDS)*	PL43A	7	LDQ41	T (LVDS)*

LFE2-35E/SE and LFE2-50E/SE Logic Signal Connections: 484 fpBGA

LFE2-35E/SE					LFE2-50E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
L3	PL24B	7	LDQ22	C (LVDS)*	PL43B	7	LDQ41	C (LVDS)*
L2	PL25A	7	PCLKT7_0/LDQ22	T	PL44A	7	PCLKT7_0/LDQ41	T
GNDIO	GNDIO7	-			GNDIO7	-		
L1	PL25B	7	PCLKC7_0/LDQ22	C	PL44B	7	PCLKC7_0/LDQ41	C
M5	PL27A	6	PCLKT6_0/LDQ31	T (LVDS)*	PL46A	6	PCLKT6_0/LDQ50	T (LVDS)*
M6	PL27B	6	PCLKC6_0/LDQ31	C (LVDS)*	PL46B	6	PCLKC6_0/LDQ50	C (LVDS)*
M3	PL28A	6	VREF2_6/LDQ31	T	PL47A	6	VREF2_6/LDQ50	T
M4	PL28B	6	VREF1_6/LDQ31	C	PL47B	6	VREF1_6/LDQ50	C
M2	PL29A	6	LDQ31	T (LVDS)*	PL48A	6	LDQ50	T (LVDS)*
VCCIO	VCCIO6	6			VCCIO	6		
M1	PL29B	6	LDQ31	C (LVDS)*	PL48B	6	LDQ50	C (LVDS)*
N1	PL30A	6	LDQ31	T	PL49A	6	LDQ50	T
N2	PL30B	6	LDQ31	C	PL49B	6	LDQ50	C
GNDIO	GNDIO6	-			GNDIO6	-		
VCCIO	VCCIO6	6			VCCIO	6		
N3	PL39A	6	LDQS39***	T (LVDS)*	PL58A	6	LDQS58***	T (LVDS)*
N4	PL39B	6	LDQ39	C (LVDS)*	PL58B	6	LDQ58	C (LVDS)*
N5	PL40A	6	LDQ39	T	PL59A	6	LDQ58	T
VCCIO	VCCIO6	6			VCCIO	6		
P5	PL40B	6	LDQ39	C	PL59B	6	LDQ58	C
P1	PL41A	6	LLM0_GDLLT_IN_A**/LDQ39	T (LVDS)*	PL60A	6	LLM0_GDLLT_IN_A**/LDQ58	T (LVDS)*
P2	PL41B	6	LLM0_GDLLC_IN_A**/LDQ39	C (LVDS)*	PL60B	6	LLM0_GDLLC_IN_A**/LDQ58	C (LVDS)*
P4	PL42A	6	LLM0_GDLLT_FB_A/LDQ39	T	PL61A	6	LLM0_GDLLT_FB_A/LDQ58	T
GNDIO	GNDIO6	-			GNDIO6	-		
R4	PL42B	6	LLM0_GDLLC_FB_A/LDQ39	C	PL61B	6	LLM0_GDLLC_FB_D/LDQ58	C
P6	LLM0_PLLCAP	6			LLM0_PLLCAP	6		
R1	PL44A	6	LLM0_GPLLT_IN_A**/LDQ48	T (LVDS)*	PL63A	6	LLM0_GPLLT_IN_A**/LDQ67	T (LVDS)*
R2	PL44B	6	LLM0_GPLLC_IN_A**/LDQ48	C (LVDS)*	PL63B	6	LLM0_GPLLC_IN_A**/LDQ67	C (LVDS)*
R3	PL45A	6	LLM0_GPLLT_FB_A/LDQ48	T	PL64A	6	LLM0_GPLLT_FB_A/LDQ67	T
T4	PL45B	6	LLM0_GPLLC_FB_A/LDQ48	C	PL64B	6	LLM0_GPLLC_FB_A/LDQ67	C
T1	PL46A	6	LDQ48	T (LVDS)*	PL65A	6	LDQ67	T (LVDS)*
VCCIO	VCCIO6	6			VCCIO	6		
T2	PL46B	6	LDQ48	C (LVDS)*	PL65B	6	LDQ67	C (LVDS)*
T5	PL47A	6	LDQ48	T	PL66A	6	LDQ67	T
T3	PL47B	6	LDQ48	C	PL66B	6	LDQ67	C
GNDIO	GNDIO6	-			VCCIO	6		
VCCIO	VCCIO6	-			GNDIO6	-		
U1	PL52A	6	LDQ56	T (LVDS)*	PL71A	6	LDQ75	T (LVDS)*
U2	PL52B	6	LDQ56	C (LVDS)*	PL71B	6	LDQ75	C (LVDS)*
V1	PL53A	6	LDQ56	T	PL72A	6	LDQ75	T
V2	PL53B	6	LDQ56	C	PL72B	6	LDQ75	C
VCCIO	VCCIO6	6			VCCIO	6		
R6	PL54A	6	LDQ56	T (LVDS)*	PL73A	6	LDQ75	T (LVDS)*
T6	PL54B	6	LDQ56	C (LVDS)*	PL73B	6	LDQ75	C (LVDS)*
U3	PL55A	6	LDQ56	T	PL74A	6	LDQ75	T
U4	PL55B	6	LDQ56	C	PL74B	6	LDQ75	C
GNDIO	GNDIO6	-			GNDIO6	-		

LFE2-35E/SE and LFE2-50E/SE Logic Signal Connections: 484 fpBGA

LFE2-35E/SE					LFE2-50E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
Y1	PL56A	6	LDQS56	T (LVDS)*	PL75A	6	LDQS75	T (LVDS)*
W1	PL56B	6	LDQ56	C (LVDS)*	PL75B	6	LDQ75	C (LVDS)*
R7	PL57A	6	LDQ56	T	PL76A	6	LDQ75	T
VCCIO	VCCIO6	6			VCCIO	6		
T7	PL57B	6	LDQ56	C	PL76B	6	LDQ75	C
V4	PL58A	6	LDQ56	T (LVDS)*	PL77A	6	LDQ75	T (LVDS)*
V3	PL58B	6	LDQ56	C (LVDS)*	PL77B	6	LDQ75	C (LVDS)*
AA2	PL59A	6	LDQ56	T	PL78A	6	LDQ75	T
GNDIO	GNDIO6	-			GNDIO6	-		
AA1	PL59B	6	LDQ56	C	PL78B	6	LDQ75	C
U7	TCK	-			TCK	-		
U5	TDI	-			TDI	-		
V5	TMS	-			TMS	-		
V6	TDO	-			TDO	-		
T8	VCCJ	-			VCCJ	-		
Y3	PB2A	5	VREF2_5/BDQ6	T	PB2A	5	VREF2_5/BDQ6	T
Y2	PB2B	5	VREF1_5/BDQ6	C	PB2B	5	VREF1_5/BDQ6	C
W4	PB3A	5	BDQ6	T	PB3A	5	BDQ6	T
W3	PB3B	5	BDQ6	C	PB3B	5	BDQ6	C
W5	PB4A	5	BDQ6	T	PB4A	5	BDQ6	T
W6	PB4B	5	BDQ6	C	PB4B	5	BDQ6	C
VCCIO	VCCIO5	5			VCCIO	5		
AB3	PB5A	5	BDQ6	T	PB5A	5	BDQ6	T
AB2	PB5B	5	BDQ6	C	PB5B	5	BDQ6	C
GNDIO	GNDIO5	-			GNDIO5	-		
Y4	PB6A	5	BDQS6	T	PB6A	5	BDQS6	T
AA3	PB6B	5	BDQ6	C	PB6B	5	BDQ6	C
AB5	PB7A	5	BDQ6	T	PB7A	5	BDQ6	T
AB4	PB7B	5	BDQ6	C	PB7B	5	BDQ6	C
AA5	PB8A	5	BDQ6	T	PB8A	5	BDQ6	T
Y5	PB8B	5	BDQ6	C	PB8B	5	BDQ6	C
VCCIO	VCCIO5	5			VCCIO	5		
AB6	PB9A	5	BDQ6	T	PB9A	5	BDQ6	T
AA6	PB9B	5	BDQ6	C	PB9B	5	BDQ6	C
GNDIO	GNDIO5	-			GNDIO5	-		
VCCIO	VCCIO5	5			VCCIO	5		
W7	PB20A	5	BDQ24	T	PB29A	5	BDQ33	T
W8	PB20B	5	BDQ24	C	PB29B	5	BDQ33	C
Y6	PB21A	5	BDQ24	T	PB30A	5	BDQ33	T
Y7	PB21B	5	BDQ24	C	PB30B	5	BDQ33	C
AA7	PB22A	5	BDQ24	T	PB31A	5	BDQ33	T
VCCIO	VCCIO5	5			VCCIO	5		
AB7	PB22B	5	BDQ24	C	PB31B	5	BDQ33	C
U8	PB23A	5	BDQ24	T	PB32A	5	BDQ33	T
U9	PB23B	5	BDQ24	C	PB32B	5	BDQ33	C
W9	PB24A	5	BDQS24	T	PB33A	5	BDQS33	T
GNDIO	GNDIO5	-			GNDIO5	-		

LFE2-35E/SE and LFE2-50E/SE Logic Signal Connections: 484 fpBGA

LFE2-35E/SE					LFE2-50E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
V9	PB24B	5	BDQ24	C	PB33B	5	BDQ33	C
Y8	PB25A	5	BDQ24	T	PB34A	5	BDQ33	T
AA8	PB25B	5	BDQ24	C	PB34B	5	BDQ33	C
W10	PB26A	5	BDQ24	T	PB35A	5	BDQ33	T
VCCIO	VCCIO5	5			VCCIO	5		
V10	PB26B	5	BDQ24	C	PB35B	5	BDQ33	C
AB8	PB27A	5	BDQ24	T	PB36A	5	BDQ33	T
AA9	PB27B	5	BDQ24	C	PB36B	5	BDQ33	C
GNDIO	GNDIO5	-			GNDIO5	-		
AB9	PB29A	5	BDQ33	T	PB38A	5	BDQ42	T
AB10	PB29B	5	BDQ33	C	PB38B	5	BDQ42	C
Y10	PB30A	5	BDQ33	T	PB39A	5	BDQ42	T
AA10	PB30B	5	BDQ33	C	PB39B	5	BDQ42	C
U10	PB31A	5	BDQ33	T	PB40A	5	BDQ42	T
U11	PB31B	5	BDQ33	C	PB40B	5	BDQ42	C
VCCIO	VCCIO5	5			VCCIO	5		
AB11	PB32A	5	BDQ33	T	PB41A	5	BDQ42	T
AA11	PB32B	5	BDQ33	C	PB41B	5	BDQ42	C
GNDIO	GNDIO5	-			GNDIO5	-		
Y11	PB33A	5	BDQS33	T	PB42A	5	BDQS42	T
W11	PB33B	5	BDQ33	C	PB42B	5	BDQ42	C
AB12	PB34A	5	BDQ33	T	PB43A	5	BDQ42	T
AA12	PB34B	5	BDQ33	C	PB43B	5	BDQ42	C
AB13	PB35A	5	PCLKT5_0/BDQ33	T	PB44A	5	PCLKT5_0/BDQ42	T
AB14	PB35B	5	PCLKC5_0/BDQ33	C	PB44B	5	PCLKC5_0/BDQ42	C
VCCIO	VCCIO5	5			VCCIO	5		
GNDIO	GNDIO5	-			GNDIO5	-		
U12	PB40A	4	PCLKT4_0/BDQ42	T	PB49A	4	PCLKT4_0/BDQ51	T
VCCIO	VCCIO4	4			VCCIO	4		
V12	PB40B	4	PCLKC4_0/BDQ42	C	PB49B	4	PCLKC4_0/BDQ51	C
Y12	PB41A	4	BDQ42	T	PB50A	4	BDQ51	T
W12	PB41B	4	BDQ42	C	PB50B	4	BDQ51	C
AA13	PB42A	4	BDQS42	T	PB51A	4	BDQS51	T
GNDIO	GNDIO4	-			GNDIO4	-		
Y13	PB42B	4	BDQ42	C	PB51B	4	BDQ51	C
U13	PB43A	4	BDQ42	T	PB52A	4	BDQ51	T
U14	PB43B	4	BDQ42	C	PB52B	4	BDQ51	C
AB15	PB44A	4	BDQ42	T	PB53A	4	BDQ51	T
VCCIO	VCCIO4	4			VCCIO	4		
AA14	PB44B	4	BDQ42	C	PB53B	4	BDQ51	C
AB16	PB45A	4	BDQ42	T	PB54A	4	BDQ51	T
AB17	PB45B	4	BDQ42	C	PB54B	4	BDQ51	C
W13	PB46A	4	BDQ42	T	PB55A	4	BDQ51	T
GNDIO	GNDIO4	-			GNDIO4	-		
W14	PB46B	4	BDQ42	C	PB55B	4	BDQ51	C
AB18	PB48A	4	BDQ51	T	PB57A	4	BDQ60	T
AB19	PB48B	4	BDQ51	C	PB57B	4	BDQ60	C

LFE2-35E/SE and LFE2-50E/SE Logic Signal Connections: 484 fpBGA

LFE2-35E/SE					LFE2-50E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
V14	PB49A	4	BDQ51	T	PB58A	4	BDQ60	T
W15	PB49B	4	BDQ51	C	PB58B	4	BDQ60	C
VCCIO	VCCIO4	4			VCCIO	4		
Y15	PB50A	4	BDQ51	T	PB59A	4	BDQ60	T
AA15	PB50B	4	BDQ51	C	PB59B	4	BDQ60	C
GNDIO	GNDIO4	-			GNDIO4	-		
AA16	PB51A	4	BDQS51	T	PB60A	4	BDQS60	T
AA17	PB51B	4	BDQ51	C	PB60B	4	BDQ60	C
AB20	PB52A	4	BDQ51	T	PB61A	4	BDQ60	T
AB21	PB52B	4	BDQ51	C	PB61B	4	BDQ60	C
U15	PB53A	4	BDQ51	T	PB62A	4	BDQ60	T
U16	PB53B	4	BDQ51	C	PB62B	4	BDQ60	C
VCCIO	VCCIO4	4			VCCIO	4		
Y16	PB54A	4	BDQ51	T	PB63A	4	BDQ60	T
W16	PB54B	4	BDQ51	C	PB63B	4	BDQ60	C
AA18	PB55A	4	BDQ51	T	PB64A	4	BDQ60	T
AA20	PB55B	4	BDQ51	C	PB64B	4	BDQ60	C
GNDIO	GNDIO4	-			GNDIO4	-		
VCCIO	VCCIO4	4			VCCIO	4		
AA21	PB66A	4	BDQ69	T	PB75A	4	BDQ78	T
AA22	PB66B	4	BDQ69	C	PB75B	4	BDQ78	C
V16	PB67A	4	BDQ69	T	PB76A	4	BDQ78	T
V17	PB67B	4	BDQ69	C	PB76B	4	BDQ78	C
VCCIO	VCCIO4	4			VCCIO	4		
Y18	PB68A	4	BDQ69	T	PB77A	4	BDQ78	T
Y17	PB68B	4	BDQ69	C	PB77B	4	BDQ78	C
GNDIO	GNDIO4	-			GNDIO4	-		
Y19	PB69A	4	BDQS69	T	PB78A	4	BDQS78	T
Y20	PB69B	4	BDQ69	C	PB78B	4	BDQ78	C
W17	PB70A	4	BDQ69	T	PB79A	4	BDQ78	T
W18	PB70B	4	BDQ69	C	PB79B	4	BDQ78	C
Y21	PB71A	4	BDQ69	T	PB80A	4	BDQ78	T
Y22	PB71B	4	BDQ69	C	PB80B	4	BDQ78	C
VCCIO	VCCIO4	4			VCCIO	4		
U18	PB72A	4	BDQ69	T	PB81A	4	BDQ78	T
V18	PB72B	4	BDQ69	C	PB81B	4	BDQ78	C
T15	PB73A	4	VREF2_4/BDQ69	T	PB82A	4	VREF2_4/BDQ78	T
T16	PB73B	4	VREF1_4/BDQ69	C	PB82B	4	VREF1_4/BDQ78	C
GNDIO	GNDIO4	-			GNDIO4	-		
W19	CFG2	8			CFG2	8		
V19	CFG1	8			CFG1	8		
W20	CFG0	8			CFG0	8		
V20	PROGRAMN	8			PROGRAMN	8		
W22	CCLK	8			CCLK	8		
V22	INITN	8			INITN	8		
V21	DONE	8			DONE	8		
GNDIO	GNDIO8	-			GNDIO8	-		

LFE2-35E/SE and LFE2-50E/SE Logic Signal Connections: 484 fpBGA

LFE2-35E/SE					LFE2-50E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
R16	PR58B	8	WRITEN	C	PR77B	8	WRITEN	C
R17	PR58A	8	CS1N	T	PR77A	8	CS1N	T
U19	PR57B	8	CSN	C	PR76B	8	CSN	C
U20	PR57A	8	D0/SPIFASTN	T	PR76A	8	D0/SPIFASTN	T
VCCIO	VCCIO8	8			VCCIO	8		
U22	PR56B	8	D1	C	PR75B	8	D1	C
U21	PR56A	8	D2	T	PR75A	8	D2	T
T20	PR55B	8	D3	C	PR74B	8	D3	C
GNDIO	GNDIO8	-			GNDIO8	-		
T19	PR55A	8	D4	T	PR74A	8	D4	T
T17	PR54B	8	D5	C	PR73B	8	D5	C
T18	PR54A	8	D6	T	PR73A	8	D6	T
T21	PR53B	8	D7/SPID0	C	PR72B	8	D7/SPID0	C
VCCIO	VCCIO8	8			VCCIO	8		
T22	PR53A	8	DI/CSSPI0N	T	PR72A	8	DI/CSSPI0N	T
R18	PR52B	8	DOUT/CSON	C	PR71B	8	DOUT/CSON	C
R19	PR52A	8	BUSY/SISPI	T	PR71A	8	BUSY/SISPI	T
GNDIO	GNDIO3	-			GNDIO3	-		
VCCIO	VCCIO3	3			VCCIO	3		
R22	PR47B	3	RDQ48	C	PR66B	3	RDQ67	C
R21	PR47A	3	RDQ48	T	PR66A	3	RDQ67	T
P18	PR46B	3	RDQ48	C (LVDS)*	PR65B	3	RDQ67	C (LVDS)*
P19	PR46A	3	RDQ48	T (LVDS)*	PR65A	3	RDQ67	T (LVDS)*
VCCIO	VCCIO3	3			VCCIO	3		
R20	PR45B	3	RLM0_GPLLC_FB_A/RDQ48	C	PR64B	3	RLM0_GPLLC_FB_A/RDQ67	C
P22	PR45A	3	RLM0_GPLLT_FB_A/RDQ48	T	PR64A	3	RLM0_GPLLT_FB_A/RDQ67	T
P21	PR44B	3	RLM0_GPLLC_IN_A**/RDQ48	C (LVDS)*	PR63B	3	RLM0_GPLLC_IN_A**/RDQ67	C (LVDS)*
N21	PR44A	3	RLM0_GPLLT_IN_A**/RDQ48	T (LVDS)*	PR63A	3	RLM0_GPLLT_IN_A**/RDQ67	T (LVDS)*
N17	RLM0_PLLCAP	3			RLM0_PLLCAP	3		
N22	PR42B	3	RLM0_GDLLC_FB_A/RDQ39	C	PR61B	3	RLM0_GDLLC_FB_A/RDQ58	C
N20	PR42A	3	RLM0_GDLLT_FB_A/RDQ39	T	PR61A	3	RLM0_GDLLT_FB_A/RDQ58	T
GNDIO	GNDIO3	-			GNDIO3	-		
M22	PR41B	3	RLM0_GDLLC_IN_A**/RDQ39	C (LVDS)*	PR60B	3	RLM0_GDLLC_IN_A**/RDQ58	C (LVDS)*
M21	PR41A	3	RLM0_GDLLT_IN_A**/RDQ39	T (LVDS)*	PR60A	3	RLM0_GDLLT_IN_A**/RDQ58	T (LVDS)*
N19	PR40B	3	RDQ39	C	PR59B	3	RDQ58	C
M19	PR40A	3	RDQ39	T	PR59A	3	RDQ58	T
VCCIO	VCCIO3	3			VCCIO	3		
GNDIO	GNDIO3	-			GNDIO3	-		
L22	PR30B	3	RDQ31	C	PR49B	3	RDQ50	C
K22	PR30A	3	RDQ31	T	PR49A	3	RDQ50	T
J22	PR29B	3	RDQ31	C (LVDS)*	PR48B	3	RDQ50	C (LVDS)*
H22	PR29A	3	RDQ31	T (LVDS)*	PR48A	3	RDQ50	T (LVDS)*
VCCIO	VCCIO3	3			VCCIO	3		
M20	PR28B	3	VREF2_3/RDQ31	C	PR47B	3	VREF2_3/RDQ50	C
L21	PR28A	3	VREF1_3/RDQ31	T	PR47A	3	VREF1_3/RDQ50	T
K21	PR27B	3	PCLKC3_0/RDQ31	C (LVDS)*	PR46B	3	PCLKC3_0/RDQ50	C (LVDS)*
J21	PR27A	3	PCLKT3_0/RDQ31	T (LVDS)*	PR46A	3	PCLKT3_0/RDQ50	T (LVDS)*

LFE2-35E/SE and LFE2-50E/SE Logic Signal Connections: 484 fpBGA

LFE2-35E/SE					LFE2-50E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
M18	PR25B	2	PCLKC2_0/RDQ22	C	PR44B	2	PCLKC2_0/RDQ41	C
L17	PR25A	2	PCLKT2_0/RDQ22	T	PR44A	2	PCLKT2_0/RDQ41	T
GNDIO	GNDIO2	-			GNDIO2	-		
L19	PR24B	2	RDQ22	C (LVDS)*	PR43B	2	RDQ41	C (LVDS)*
L20	PR24A	2	RDQ22	T (LVDS)*	PR43A	2	RDQ41	T (LVDS)*
L18	PR23B	2	RDQ22	C	PR42B	2	RDQ41	C
K17	PR23A	2	RDQ22	T	PR42A	2	RDQ41	T
VCCIO	VCCIO2	2			VCCIO	2		
K18	PR22B	2	RDQ22	C (LVDS)*	PR41B	2	RDQ41	C (LVDS)*
K19	PR22A	2	RDQS22	T (LVDS)*	PR41A	2	RDQS41	T (LVDS)*
G22	PR21B	2	RDQ22	C	PR40B	2	RDQ41	C
GNDIO	GNDIO2	-			GNDIO2	-		
F22	PR21A	2	RDQ22	T	PR40A	2	RDQ41	T
J17	PR20B	2	RDQ22	C (LVDS)*	PR39B	2	RDQ41	C (LVDS)*
J18	PR20A	2	RDQ22	T (LVDS)*	PR39A	2	RDQ41	T (LVDS)*
K20	PR19B	2	RDQ22	C	PR38B	2	RDQ41	C
VCCIO	VCCIO2	2			VCCIO	2		
J19	PR19A	2	RDQ22	T	PR38A	2	RDQ41	T
H21	PR18B	2	RDQ22	C (LVDS)*	PR37B	2	RDQ41	C (LVDS)*
G21	PR18A	2	RDQ22	T (LVDS)*	PR37A	2	RDQ41	T (LVDS)*
-	-	-			GNDIO2	-		
-	-	-			VCCIO	2		
H17	NC	-			PR26B	2	RUM0_SPLLC_FB_A/RDQ24	C
H16	NC	-			PR26A	2	RUM0_SPLLT_FB_A/RDQ24	T
H20	NC	-			PR25B	2	RUM0_SPLLC_IN_A/RDQ24	C
H18	NC	-			PR25A	2	RUM0_SPLLT_IN_A/RDQ24	T
-	-	-			GNDIO2	-		
-	-	-			VCCIO	2		
F21	PR17B	2	RDQ14	C	PR19B	2	RDQ16	C
GNDIO	GNDIO2	-			GNDIO2	-		
E22	PR17A	2	RDQ14	T	PR19A	2	RDQ16	T
D22	PR16B	2	RDQ14	C (LVDS)*	PR18B	2	RDQ16	C (LVDS)*
E21	PR16A	2	RDQ14	T (LVDS)*	PR18A	2	RDQ16	T (LVDS)*
G20	PR15B	2	RDQ14	C	PR17B	2	RDQ16	C
VCCIO	VCCIO2	2			VCCIO	2		
F20	PR15A	2	RDQ14	T	PR17A	2	RDQ16	T
H19	PR14B	2	RDQ14	C (LVDS)*	PR16B	2	RDQ16	C (LVDS)*
G19	PR14A	2	RDQS14	T (LVDS)*	PR16A	2	RDQS16	T (LVDS)*
GNDIO	GNDIO2	-			GNDIO2	-		
G17	PR13B	2	RDQ14	C	PR15B	2	RDQ16	C
F19	PR13A	2	RDQ14	T	PR15A	2	RDQ16	T
E20	PR12B	2	RDQ14	C (LVDS)*	PR14B	2	RDQ16	C (LVDS)*
D20	PR12A	2	RDQ14	T (LVDS)*	PR14A	2	RDQ16	T (LVDS)*
VCCIO	VCCIO2	2			VCCIO	2		
F18	PR11B	2	RDQ14	C	PR13B	2	RDQ16	C
F16	PR11A	2	RDQ14	T	PR13A	2	RDQ16	T
C21	PR10B	2	RDQ14	C (LVDS)*	PR12B	2	RDQ16	C (LVDS)*

LFE2-35E/SE and LFE2-50E/SE Logic Signal Connections: 484 fpBGA

LFE2-35E/SE					LFE2-50E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
C22	PR10A	2	RDQ14	T (LVDS)*	PR12A	2	RDQ16	T (LVDS)*
VCCIO	VCCIO2	2			VCCIO	2		
GNDIO	GNDIO2	-			GNDIO2	-		
D19	PR2B	2	VREF2_2/RDQ6	C (LVDS)*	PR2B	2	VREF2_2	C (LVDS)*
E19	PR2A	2	VREF1_2/RDQ6	T (LVDS)*	PR2A	2	VREF1_2	T (LVDS)*
B21	PT73B	1	VREF2_1	C	PT82B	1	VREF2_1	C
GNDIO	GNDIO1	-			GNDIO1	-		
B22	PT73A	1	VREF1_1	T	PT82A	1	VREF1_1	T
C20	PT72B	1		C	PT81B	1		C
C19	PT72A	1		T	PT81A	1		T
D18	PT71B	1		C	PT80B	1		C
VCCIO	VCCIO1	1			VCCIO	1		
E18	PT71A	1		T	PT80A	1		T
B20	PT70B	1		C	PT79B	1		C
A19	PT70A	1		T	PT79A	1		T
D17	PT69B	1		C	PT78B	1		C
C18	PT69A	1		T	PT78A	1		T
A21	PT68B	1		C	PT77B	1		C
GNDIO	GNDIO1	-			GNDIO1	-		
A20	PT68A	1		T	PT77A	1		T
A18	PT67B	1		C	PT76B	1		C
VCCIO	VCCIO1	1			VCCIO	1		
B18	PT67A	1		T	PT76A	1		T
G16	PT66B	1		C	PT75B	1		C
G15	PT66A	1		T	PT75A	1		T
D16	PT65B	1		C	PT74B	1		C
E16	PT65A	1		T	PT74A	1		T
GNDIO	GNDIO1	-			GNDIO1	-		
VCCIO	VCCIO1	1			VCCIO	1		
C17	PT55B	1		C	PT64B	1		C
GNDIO	GNDIO1	-			GNDIO1	-		
C16	PT55A	1		T	PT64A	1		T
B17	PT54B	1		C	PT63B	1		C
B16	PT54A	1		T	PT63A	1		T
A17	PT53B	1		C	PT62B	1		C
VCCIO	VCCIO1	1			VCCIO	1		
A16	PT53A	1		T	PT62A	1		T
C15	PT52B	1		C	PT61B	1		C
D15	PT52A	1		T	PT61A	1		T
E15	PT51B	1		C	PT60B	1		C
F15	PT51A	1		T	PT60A	1		T
GNDIO	GNDIO1	-			GNDIO1	-		
B15	PT49B	1		C	PT58B	1		C
VCCIO	VCCIO1	1			VCCIO	1		
A15	PT49A	1		T	PT58A	1		T
B14	PT48B	1		C	PT57B	1		C
A14	PT48A	1		T	PT57A	1		T

LFE2-35E/SE and LFE2-50E/SE Logic Signal Connections: 484 fpBGA

LFE2-35E/SE					LFE2-50E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
D14	PT46B	1		C	PT55B	1		C
C13	PT46A	1		T	PT55A	1		T
GNDIO	GNDIO1	-			GNDIO1	-		
E14	PT45B	1		C	PT54B	1		C
F14	PT45A	1		T	PT54A	1		T
A13	PT44B	1		C	PT53B	1		C
B13	PT44A	1		T	PT53A	1		T
VCCIO	VCCIO1	1			VCCIO	1		
E13	PT43B	1		C	PT52B	1		C
D13	PT43A	1		T	PT52A	1		T
E12	PT42B	1		C	PT51B	1		C
D12	PT42A	1		T	PT51A	1		T
GNDIO	GNDIO1	-			GNDIO1	-		
A12	PT40B	1		C	PT49B	1		C
A11	PT40A	1		T	PT49A	1		T
VCCIO	VCCIO1	1			VCCIO	1		
B12	PT39B	1	PCLKC1_0	C	PT48B	1	PCLKC1_0	C
C12	PT39A	1	PCLKT1_0	T	PT48A	1	PCLKT1_0	T
F12	XRES	1			XRES	1		
B10	PT37B	0	PCLKC0_0	C	PT46B	0	PCLKC0_0	C
GNDIO	GNDIO0	-			GNDIO0	0		
B11	PT37A	0	PCLKT0_0	T	PT46A	0	PCLKT0_0	T
A10	PT36B	0		C	PT45B	0		C
A9	PT36A	0		T	PT45A	0		T
C11	PT35B	0		C	PT44B	0		C
VCCIO	VCCIO0	0			VCCIO	0		
C10	PT35A	0		T	PT44A	0		T
E11	PT34B	0		C	PT43B	0		C
F11	PT34A	0		T	PT43A	0		T
A8	PT33B	0		C	PT42B	0		C
A7	PT33A	0		T	PT42A	0		T
B8	PT32B	0		C	PT41B	0		C
GNDIO	GNDIO0	-			GNDIO0	0		
B9	PT32A	0		T	PT41A	0		T
VCCIO	VCCIO0	0			VCCIO	0		
B7	PT30B	0		C	PT39B	0		C
A6	PT30A	0		T	PT39A	0		T
C8	PT29B	0		C	PT38B	0		C
D8	PT29A	0		T	PT38A	0		T
GNDIO	GNDIO0	-			GNDIO0	0		
D10	PT27B	0		C	PT36B	0		C
E10	PT27A	0		T	PT36A	0		T
C7	PT26B	0		C	PT35B	0		C
C6	PT26A	0		T	PT35A	0		T
VCCIO	VCCIO0	0			VCCIO	0		
B6	PT25B	0		C	PT34B	0		C
B5	PT25A	0		T	PT34A	0		T

LFE2-35E/SE and LFE2-50E/SE Logic Signal Connections: 484 fpBGA

LFE2-35E/SE					LFE2-50E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
F10	PT24B	0		C	PT33B	0		C
D9	PT24A	0		T	PT33A	0		T
GNDIO	GNDIO0	-			GNDIO0	0		
F9	PT23B	0		C	PT32B	0		C
E9	PT23A	0		T	PT32A	0		T
A5	PT22B	0		C	PT31B	0		C
A4	PT22A	0		T	PT31A	0		T
VCCIO	VCCIO0	0			VCCIO	0		
A3	PT21B	0		C	PT30B	0		C
A2	PT21A	0		T	PT30A	0		T
G8	PT20B	0		C	PT29B	0		C
E8	PT20A	0		T	PT29A	0		T
GNDIO	GNDIO0	-			GNDIO0	0		
VCCIO	VCCIO0	0			VCCIO	0		
C3	PT10B	0		C	PT10B	0		C
B3	PT10A	0		T	PT10A	0		T
GNDIO	GNDIO0	-			GNDIO0	0		
F8	PT9B	0		C	PT9B	0		C
D7	PT9A	0		T	PT9A	0		T
E7	PT8B	0		C	PT8B	0		C
VCCIO	VCCIO0	0			VCCIO	0		
F7	PT8A	0		T	PT8A	0		T
D5	PT7B	0		C	PT7B	0		C
D6	PT7A	0		T	PT7A	0		T
D4	PT6B	0		C	PT6B	0		C
C4	PT6A	0		T	PT6A	0		T
GNDIO	GNDIO0	-			GNDIO0	0		
B2	PT5B	0		C	PT5B	0		C
B1	PT5A	0		T	PT5A	0		T
J7	PT4B	0		C	PT4B	0		C
VCCIO	VCCIO0	0			VCCIO	0		
H7	PT4A	0		T	PT4A	0		T
D3	PT3B	0		C	PT3B	0		C
C2	PT3A	0		T	PT3A	0		T
D1	PT2B	0	VREF2_0	C	PT2B	0	VREF2_0	C
C1	PT2A	0	VREF1_0	T	PT2A	0	VREF1_0	T
J10	VCC	-			VCC	-		
J11	VCC	-			VCC	-		
J12	VCC	-			VCC	-		
J13	VCC	-			VCC	-		
K14	VCC	-			VCC	-		
K9	VCC	-			VCC	-		
L14	VCC	-			VCC	-		
L9	VCC	-			VCC	-		
M14	VCC	-			VCC	-		
M9	VCC	-			VCC	-		
N14	VCC	-			VCC	-		

LFE2-35E/SE and LFE2-50E/SE Logic Signal Connections: 484 fpBGA

LFE2-35E/SE					LFE2-50E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
N9	VCC	-			VCC	-		
P10	VCC	-			VCC	-		
P11	VCC	-			VCC	-		
P12	VCC	-			VCC	-		
P13	VCC	-			VCC	-		
G5	VCCAUX	-			VCCAUX	0		
K5	VCCAUX	-			VCCAUX	0		
R5	VCCAUX	-			VCCAUX	1		
V7	VCCAUX	-			VCCAUX	1		
V11	VCCAUX	-			VCCAUX	2		
V8	VCCAUX	-			VCCAUX	2		
V13	VCCAUX	-			VCCAUX	3		
V15	VCCAUX	-			VCCAUX	3		
M17	VCCAUX	-			VCCAUX	4		
P17	VCCAUX	-			VCCAUX	4		
E17	VCCAUX	-			VCCAUX	5		
G18	VCCAUX	-			VCCAUX	5		
D11	VCCAUX	-			VCCAUX	6		
F13	VCCAUX	-			VCCAUX	6		
C5	VCCAUX	-			VCCAUX	7		
E6	VCCAUX	-			VCCAUX	7		
G10	VCCIO0	0			VCCIO0	0		
G9	VCCIO0	0			VCCIO0	0		
H8	VCCIO0	0			VCCIO0	0		
H9	VCCIO0	0			VCCIO0	0		
G11	VCCIO1	1			VCCIO1	1		
G12	VCCIO1	1			VCCIO1	1		
G13	VCCIO1	1			VCCIO1	1		
G14	VCCIO1	1			VCCIO1	1		
H14	VCCIO2	2			VCCIO2	2		
H15	VCCIO2	2			VCCIO2	2		
J15	VCCIO2	2			VCCIO2	2		
K16	VCCIO2	2			VCCIO2	2		
L16	VCCIO3	3			VCCIO3	3		
M16	VCCIO3	3			VCCIO3	3		
N16	VCCIO3	3			VCCIO3	3		
P16	VCCIO3	3			VCCIO3	3		
R14	VCCIO4	4			VCCIO4	4		
T12	VCCIO4	4			VCCIO4	4		
T13	VCCIO4	4			VCCIO4	4		
T14	VCCIO4	4			VCCIO4	4		
R9	VCCIO5	5			VCCIO5	5		
T10	VCCIO5	5			VCCIO5	5		
T11	VCCIO5	5			VCCIO5	5		
T9	VCCIO5	5			VCCIO5	5		
N7	VCCIO6	6			VCCIO6	6		
P7	VCCIO6	6			VCCIO6	6		

LFE2-35E/SE and LFE2-50E/SE Logic Signal Connections: 484 fpBGA

LFE2-35E/SE					LFE2-50E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
P8	VCCIO6	6			VCCIO6	6		
R8	VCCIO6	6			VCCIO6	6		
J8	VCCIO7	7			VCCIO7	7		
K7	VCCIO7	7			VCCIO7	7		
L7	VCCIO7	7			VCCIO7	7		
M7	VCCIO7	7			VCCIO7	7		
P15	VCCIO8	8			VCCIO8	8		
R15	VCCIO8	8			VCCIO8	8		
A22	GND	-			GND	-		
AA19	GND	-			GND	-		
AA4	GND	-			GND	-		
AB1	GND	-			GND	-		
AB22	GND	-			GND	-		
B19	GND	-			GND	-		
B4	GND	-			GND	-		
C14	GND	-			GND	-		
C9	GND	-			GND	-		
D2	GND	-			GND	-		
D21	GND	-			GND	-		
F17	GND	-			GND	-		
F6	GND	-			GND	-		
H10	GND	-			GND	-		
H11	GND	-			GND	-		
H12	GND	-			GND	-		
H13	GND	-			GND	-		
J14	GND	-			GND	-		
J20	GND	-			GND	-		
J3	GND	-			GND	-		
J9	GND	-			GND	-		
K10	GND	-			GND	-		
K11	GND	-			GND	-		
K12	GND	-			GND	-		
K13	GND	-			GND	-		
K15	GND	-			GND	-		
K8	GND	-			GND	-		
L10	GND	-			GND	-		
L11	GND	-			GND	-		
L12	GND	-			GND	-		
L13	GND	-			GND	-		
L15	GND	-			GND	-		
L8	GND	-			GND	-		
M10	GND	-			GND	-		
M11	GND	-			GND	-		
M12	GND	-			GND	-		
M13	GND	-			GND	-		
M15	GND	-			GND	-		
M8	GND	-			GND	-		

LFE2-35E/SE and LFE2-50E/SE Logic Signal Connections: 484 fpBGA

LFE2-35E/SE					LFE2-50E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
N10	GND	-			GND	-		
N11	GND	-			GND	-		
N12	GND	-			GND	-		
N13	GND	-			GND	-		
N15	GND	-			GND	-		
N8	GND	-			GND	-		
P14	GND	-			GND	-		
P20	GND	-			GND	-		
P3	GND	-			GND	-		
P9	GND	-			GND	-		
R10	GND	-			GND	-		
R11	GND	-			GND	-		
R12	GND	-			GND	-		
R13	GND	-			GND	-		
U17	GND	-			GND	-		
U6	GND	-			GND	-		
W2	GND	-			GND	-		
W21	GND	-			GND	-		
Y14	GND	-			GND	-		
Y9	GND	-			GND	-		
A1	GND	-			GND	-		
N18	VCCPLL	-			VCCPLL	-		
K6	NC	-			VCCPLL	-		
N6	VCCPLL	-			VCCPLL	-		
J16	NC	-			VCCPLL	-		

* Supports true LVDS. Other differential signals must be emulated with external resistors.
** These dedicated input pins can be used for GPLLs or GDLLs within the respective quadrant.
***Due to packaging bond out option, this DQS does not have all the necessary DQ pins bonded out for a full 8-bit data width.

Note: VCCIO and GND pads are used to determine the average DC current drawn by I/Os between GND/VCCIO connections, or between the last GND/VCCIO in an I/O bank and the end of an I/O bank. The substrate pads listed in the Pin Table do not necessarily have a one to one connection with a package ball or pin.

LFE2-20E/SE and LFE2-35E/SE Logic Signal Connections: 672 fpBGA

LFE2-20E/20SE					LFE2-35E/35SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
D2	PL2A	7	VREF2_7	T (LVDS)*	PL2A	7	VREF2_7/LDQ6	T (LVDS)*
D1	PL2B	7	VREF1_7	C (LVDS)*	PL2B	7	VREF1_7/LDQ6	C (LVDS)*
GND	GNDIO7	-			GNDIO7	-		
F6	PL3A	7		T	PL3A	7	LDQ6	T
F5	PL3B	7		C	PL3B	7	LDQ6	C
VCCIO	VCCIO7	7			VCCIO7	7		
E4	NC	-			PL4A	7	LDQ6	T (LVDS)*
E3	NC	-			PL4B	7	LDQ6	C (LVDS)*
E2	NC	-			PL5A	7	LDQ6	T
E1	NC	-			PL5B	7	LDQ6	C
GND	GNDIO7	-			GNDIO7	-		
H6	NC	-			PL6A	7	LDQS6	T (LVDS)*
H5	NC	-			PL6B	7	LDQ6	C (LVDS)*
F2	NC	-			PL7A	7	LDQ6	T
VCCIO	VCCIO7	7			VCCIO7	7		
F1	NC	-			PL7B	7	LDQ6	C
H8	NC	-			PL8A	7	LDQ6	T (LVDS)*
J9	NC	-			PL8B	7	LDQ6	C (LVDS)*
G4	NC	-			PL9A	7	LDQ6	T
GND	GNDIO7	-			GNDIO7	-		
G3	NC	-			PL9B	7	LDQ6	C
H7	PL4A	7	LDQ8	T (LVDS)*	PL10A	7	LDQ14	T (LVDS)*
J8	PL4B	7	LDQ8	C (LVDS)*	PL10B	7	LDQ14	C (LVDS)*
G2	PL5A	7	LDQ8	T	PL11A	7	LDQ14	T
G1	PL5B	7	LDQ8	C	PL11B	7	LDQ14	C
H3	PL6A	7	LDQ8	T (LVDS)*	PL12A	7	LDQ14	T (LVDS)*
VCCIO	VCCIO7	7			VCCIO7	7		
H4	PL6B	7	LDQ8	C (LVDS)*	PL12B	7	LDQ14	C (LVDS)*
J5	PL7A	7	LDQ8	T	PL13A	7	LDQ14	T
J4	PL7B	7	LDQ8	C	PL13B	7	LDQ14	C
J3	PL8A	7	LDQS8	T (LVDS)*	PL14A	7	LDQS14	T (LVDS)*
GND	GNDIO7	-			GNDIO7	-		
K4	PL8B	7	LDQ8	C (LVDS)*	PL14B	7	LDQ14	C (LVDS)*
H1	PL9A	7	LDQ8	T	PL15A	7	LDQ14	T
H2	PL9B	7	LDQ8	C	PL15B	7	LDQ14	C
VCCIO	VCCIO7	7			VCCIO7	7		
K6	PL10A	7	LDQ8	T (LVDS)*	PL16A	7	LDQ14	T (LVDS)*
K7	PL10B	7	LDQ8	C (LVDS)*	PL16B	7	LDQ14	C (LVDS)*
J1	PL11A	7	LDQ8	T	PL17A	7	LDQ14	T
J2	PL11B	7	LDQ8	C	PL17B	7	LDQ14	C
GND	GNDIO7	-			GNDIO7	-		
VCCIO	VCCIO7	7			VCCIO7	7		
K3	NC	-			NC	-		
K2	NC	-			NC	-		
GND	GNDIO7	-			GNDIO7	-		
K1	NC	-			NC	-		
L2	NC	-			NC	-		

LFE2-20E/SE and LFE2-35E/SE Logic Signal Connections: 672 fpBGA

LFE2-20E/20SE					LFE2-35E/35SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
L1	NC	-			NC	-		
VCCIO	VCCIO7	7			VCCIO7	7		
M2	NC	-			NC	-		
M1	NC	-			NC	-		
N2	NC	-			NC	-		
GND	GNDIO7	-			GNDIO7	-		
M8	VCC	-			NC	-		
VCCIO	VCCIO7	7			VCCIO7	7		
GND	GNDIO7	-			GNDIO7	-		
N1	PL12A	7	LDQ16		PL18A	7	LDQ22	
L8	PL13A	7	LDQ16	T	PL19A	7	LDQ22	T
K8	PL13B	7	LDQ16	C	PL19B	7	LDQ22	C
VCCIO	VCCIO7	7			VCCIO7	7		
L6	PL14A	7	LDQ16	T (LVDS)*	PL20A	7	LDQ22	T (LVDS)*
K5	PL14B	7	LDQ16	C (LVDS)*	PL20B	7	LDQ22	C (LVDS)*
L7	PL15A	7	LDQ16	T	PL21A	7	LDQ22	T
L5	PL15B	7	LDQ16	C	PL21B	7	LDQ22	C
GND	GNDIO7	-			GNDIO7	-		
P1	PL16A	7	LDQS16	T (LVDS)*	PL22A	7	LDQS22	T (LVDS)*
P2	PL16B	7	LDQ16	C (LVDS)*	PL22B	7	LDQ22	C (LVDS)*
M6	PL17A	7	LDQ16	T	PL23A	7	LDQ22	T
VCCIO	VCCIO7	7			VCCIO7	7		
N8	PL17B	7	LDQ16	C	PL23B	7	LDQ22	C
R1	PL18A	7	LDQ16	T (LVDS)*	PL24A	7	LDQ22	T (LVDS)*
R2	PL18B	7	LDQ16	C (LVDS)*	PL24B	7	LDQ22	C (LVDS)*
M7	PL19A	7	PCLKT7_0/LDQ16	T	PL25A	7	PCLKT7_0/LDQ22	T
GND	GNDIO7	-			GNDIO7	-		
N9	PL19B	7	PCLKC7_0/LDQ16	C	PL25B	7	PCLKC7_0/LDQ22	C
M4	PL21A	6	PCLKT6_0/LDQ25	T (LVDS)*	PL27A	6	PCLKT6_0/LDQ31	T (LVDS)*
M5	PL21B	6	PCLKC6_0/LDQ25	C (LVDS)*	PL27B	6	PCLKC6_0/LDQ31	C (LVDS)*
N7	PL22A	6	VREF2_6/LDQ25	T	PL28A	6	VREF2_6/LDQ31	T
P9	PL22B	6	VREF1_6/LDQ25	C	PL28B	6	VREF1_6/LDQ31	C
N3	PL23A	6	LDQ25	T (LVDS)*	PL29A	6	LDQ31	T (LVDS)*
VCCIO	VCCIO6	6			VCCIO6	6		
N4	PL23B	6	LDQ25	C (LVDS)*	PL29B	6	LDQ31	C (LVDS)*
N5	PL24A	6	LDQ25	T	PL30A	6	LDQ31	T
P7	PL24B	6	LDQ25	C	PL30B	6	LDQ31	C
T1	NC	-			PL31A	6	LDQS31	T (LVDS)*
GND	GNDIO6	-			GNDIO6	-		
T2	NC	-			PL31B	6	LDQ31	C (LVDS)*
P8	NC	-			PL32A	6	LDQ31	T
P6	NC	-			PL32B	6	LDQ31	C
VCCIO	VCCIO6	6			VCCIO6	6		
P5	NC	-			PL33A	6	LDQ31	T (LVDS)*
P4	NC	-			PL33B	6	LDQ31	C (LVDS)*
U1	NC	-			PL34A	6	LDQ31	T
V1	NC	-			PL34B	6	LDQ31	C

LFE2-20E/SE and LFE2-35E/SE Logic Signal Connections: 672 fpBGA

LFE2-20E/20SE					LFE2-35E/35SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
GND	GNDIO6	-			GNDIO6	-		
P3	NC	-			NC	-		
R3	NC	-			NC	-		
R4	NC	-			NC	-		
U2	NC	-			NC	-		
VCCIO	VCCIO6	6			VCCIO6	6		
V2	NC	-			NC	-		
W2	NC	-			NC	-		
T6	NC	-			PL38A	6	LDQ39	T
R5	NC	-			PL38B	6	LDQ39	C
GND	GNDIO6	-			GNDIO6	-		
R6	PL25A	6	LDQS25***	T (LVDS)*	PL39A	6	LDQS39***	T (LVDS)*
R7	PL25B	6	LDQ25	C (LVDS)*	PL39B	6	LDQ39	C (LVDS)*
W1	PL26A	6	LDQ25	T	PL40A	6	LDQ39	T
VCCIO	VCCIO6	6			VCCIO6	6		
Y2	PL26B	6	LDQ25	C	PL40B	6	LDQ39	C
Y1	PL27A	6	LLM0_GDLLT_IN_A**/LDQ25	T (LVDS)*	PL41A	6	LLM0_GDLLT_IN_A**/LDQ39	T (LVDS)*
AA2	PL27B	6	LLM0_GDLLC_IN_A**/LDQ25	C (LVDS)*	PL41B	6	LLM0_GDLLC_IN_A**/LDQ39	C (LVDS)*
T5	PL28A	6	LLM0_GDLLT_FB_A/LDQ25	T	PL42A	6	LLM0_GDLLT_FB_A/LDQ39	T
GND	GNDIO6	-			GNDIO6	-		
T7	PL28B	6	LLM0_GDLLC_FB_A/LDQ25	C	PL42B	6	LLM0_GDLLC_FB_A/LDQ39	C
R8	VCC	6			VCCPLL	6		
T8	LLM0_PLLCAP	6			LLM0_PLLCAP	6		
U3	PL30A	6	LLM0_GPLLT_IN_A**/LDQ34	T (LVDS)*	PL44A	6	LLM0_GPLLT_IN_A**/LDQ48	T (LVDS)*
U4	PL30B	6	LLM0_GPLLC_IN_A**/LDQ34	C (LVDS)*	PL44B	6	LLM0_GPLLC_IN_A**/LDQ48	C (LVDS)*
V3	PL31A	6	LLM0_GPLLT_FB_A/LDQ34	T	PL45A	6	LLM0_GPLLT_FB_A/LDQ48	T
U5	PL31B	6	LLM0_GPLLC_FB_A/LDQ34	C	PL45B	6	LLM0_GPLLC_FB_A/LDQ48	C
V4	PL32A	6	LDQ34	T (LVDS)*	PL46A	6	LDQ48	T (LVDS)*
VCCIO	VCCIO6	6			VCCIO6	6		
V5	PL32B	6	LDQ34	C (LVDS)*	PL46B	6	LDQ48	C (LVDS)*
Y3	PL33A	6	LDQ34	T	PL47A	6	LDQ48	T
Y4	PL33B	6	LDQ34	C	PL47B	6	LDQ48	C
W3	PL34A	6	LDQS34	T (LVDS)*	PL48A	6	LDQS48	T (LVDS)*
GND	GNDIO6	-			GNDIO6	-		
W4	PL34B	6	LDQ34	C (LVDS)*	PL48B	6	LDQ48	C (LVDS)*
AA1	PL35A	6	LDQ34	T	PL49A	6	LDQ48	T
AB1	PL35B	6	LDQ34	C	PL49B	6	LDQ48	C
VCCIO	VCCIO6	6			VCCIO6	6		
U8	PL36A	6	LDQ34	T (LVDS)*	PL50A	6	LDQ48	T (LVDS)*
U7	PL36B	6	LDQ34	C (LVDS)*	PL50B	6	LDQ48	C (LVDS)*
V8	PL37A	6	LDQ34	T	PL51A	6	LDQ48	T
U6	PL37B	6	LDQ34	C	PL51B	6	LDQ48	C
GND	GNDIO6	-			GNDIO6	-		
W6	PL38A	6	LDQ42	T (LVDS)*	PL52A	6	LDQ56	T (LVDS)*
W5	PL38B	6	LDQ42	C (LVDS)*	PL52B	6	LDQ56	C (LVDS)*
AC1	PL39A	6	LDQ42	T	PL53A	6	LDQ56	T
AD1	PL39B	6	LDQ42	C	PL53B	6	LDQ56	C

LFE2-20E/SE and LFE2-35E/SE Logic Signal Connections: 672 fpBGA

LFE2-20E/20SE					LFE2-35E/35SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
VCCIO	VCCIO6	6			VCCIO6	6		
Y6	PL40A	6	LDQ42	T (LVDS)*	PL54A	6	LDQ56	T (LVDS)*
Y5	PL40B	6	LDQ42	C (LVDS)*	PL54B	6	LDQ56	C (LVDS)*
AE2	PL41A	6	LDQ42	T	PL55A	6	LDQ56	T
AD2	PL41B	6	LDQ42	C	PL55B	6	LDQ56	C
GND	GNDIO6	-			GNDIO6	-		
AB3	PL42A	6	LDQS42	T (LVDS)*	PL56A	6	LDQS56	T (LVDS)*
AB2	PL42B	6	LDQ42	C (LVDS)*	PL56B	6	LDQ56	C (LVDS)*
W7	PL43A	6	LDQ42	T	PL57A	6	LDQ56	T
VCCIO	VCCIO6	6			VCCIO6	6		
W8	PL43B	6	LDQ42	C	PL57B	6	LDQ56	C
Y7	PL44A	6	LDQ42	T (LVDS)*	PL58A	6	LDQ56	T (LVDS)*
Y8	PL44B	6	LDQ42	C (LVDS)*	PL58B	6	LDQ56	C (LVDS)*
AC2	PL45A	6	LDQ42	T	PL59A	6	LDQ56	T
GND	GNDIO6	-			GNDIO6	-		
AD3	PL45B	6	LDQ42	C	PL59B	6	LDQ56	C
AC3	TCK	-			TCK	-		
AA8	TDI	-			TDI	-		
AB4	TMS	-			TMS	-		
AA5	TDO	-			TDO	-		
AB5	VCCJ	-			VCCJ	-		
AE3	PB2A	5	VREF2_5/BDQ6	T	PB2A	5	VREF2_5/BDQ6	T
AF3	PB2B	5	VREF1_5/BDQ6	C	PB2B	5	VREF1_5/BDQ6	C
AC4	PB3A	5	BDQ6	T	PB3A	5	BDQ6	T
AD4	PB3B	5	BDQ6	C	PB3B	5	BDQ6	C
AE4	PB4A	5	BDQ6	T	PB4A	5	BDQ6	T
AF4	PB4B	5	BDQ6	C	PB4B	5	BDQ6	C
VCCIO	VCCIO5	5			VCCIO5	5		
V9	PB5A	5	BDQ6	T	PB5A	5	BDQ6	T
W9	PB5B	5	BDQ6	C	PB5B	5	BDQ6	C
GND	GNDIO5	-			GNDIO5	-		
AA6	PB6A	5	BDQS6	T	PB6A	5	BDQS6	T
AB6	PB6B	5	BDQ6	C	PB6B	5	BDQ6	C
AC5	PB7A	5	BDQ6	T	PB7A	5	BDQ6	T
AD5	PB7B	5	BDQ6	C	PB7B	5	BDQ6	C
AA7	PB8A	5	BDQ6	T	PB8A	5	BDQ6	T
AB7	PB8B	5	BDQ6	C	PB8B	5	BDQ6	C
VCCIO	VCCIO5	5			VCCIO5	5		
AE5	PB9A	5	BDQ6	T	PB9A	5	BDQ6	T
AF5	PB9B	5	BDQ6	C	PB9B	5	BDQ6	C
AC7	PB10A	5	BDQ6	T	PB10A	5	BDQ6	T
AD7	PB10B	5	BDQ6	C	PB10B	5	BDQ6	C
VCCIO	VCCIO5	5			VCCIO5	5		
GND	GNDIO5	-			GNDIO5	-		
W10	PB11A	5	BDQ15	T	PB11A	5	BDQ15	T
Y10	PB11B	5	BDQ15	C	PB11B	5	BDQ15	C
W11	PB12A	5	BDQ15	T	PB12A	5	BDQ15	T

LFE2-20E/SE and LFE2-35E/SE Logic Signal Connections: 672 fpBGA

LFE2-20E/20SE					LFE2-35E/35SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
AA10	PB12B	5	BDQ15	C	PB12B	5	BDQ15	C
AC8	PB13A	5	BDQ15	T	PB13A	5	BDQ15	T
AD8	PB13B	5	BDQ15	C	PB13B	5	BDQ15	C
VCCIO	VCCIO5	5			VCCIO5	5		
AB8	PB14A	5	BDQ15	T	PB14A	5	BDQ15	T
AB10	PB14B	5	BDQ15	C	PB14B	5	BDQ15	C
GND	GNDIO5	-			GNDIO5	-		
AE6	PB15A	5	BDQS15	T	PB15A	5	BDQS15	T
AF6	PB15B	5	BDQ15	C	PB15B	5	BDQ15	C
AA11	PB16A	5	BDQ15	T	PB16A	5	BDQ15	T
AC9	PB16B	5	BDQ15	C	PB16B	5	BDQ15	C
AB9	PB17A	5	BDQ15	T	PB17A	5	BDQ15	T
AD9	PB17B	5	BDQ15	C	PB17B	5	BDQ15	C
VCCIO	VCCIO5	5			VCCIO5	5		
Y11	PB18A	5	BDQ15	T	PB18A	5	BDQ15	T
AB11	PB18B	5	BDQ15	C	PB18B	5	BDQ15	C
AE7	PB19A	5	BDQ15	T	PB19A	5	BDQ15	T
AF7	PB19B	5	BDQ15	C	PB19B	5	BDQ15	C
GND	GNDIO5	-			GNDIO5	-		
AC10	PB20A	5	BDQ24	T	PB20A	5	BDQ24	T
AD10	PB20B	5	BDQ24	C	PB20B	5	BDQ24	C
AA12	PB21A	5	BDQ24	T	PB21A	5	BDQ24	T
W12	PB21B	5	BDQ24	C	PB21B	5	BDQ24	C
AB12	PB22A	5	BDQ24	T	PB22A	5	BDQ24	T
VCCIO	VCCIO5	5			VCCIO5	5		
Y12	PB22B	5	BDQ24	C	PB22B	5	BDQ24	C
AD12	PB23A	5	BDQ24	T	PB23A	5	BDQ24	T
AC12	PB23B	5	BDQ24	C	PB23B	5	BDQ24	C
AC13	PB24A	5	BDQS24	T	PB24A	5	BDQS24	T
GND	GNDIO5	-			GNDIO5	-		
AA13	PB24B	5	BDQ24	C	PB24B	5	BDQ24	C
AD13	PB25A	5	BDQ24	T	PB25A	5	BDQ24	T
AC14	PB25B	5	BDQ24	C	PB25B	5	BDQ24	C
AE8	PB26A	5	BDQ24	T	PB26A	5	BDQ24	T
VCCIO	VCCIO5	5			VCCIO5	5		
AF8	PB26B	5	BDQ24	C	PB26B	5	BDQ24	C
AB15	PB27A	5	BDQ24	T	PB27A	5	BDQ24	T
Y13	PB27B	5	BDQ24	C	PB27B	5	BDQ24	C
AE9	PB28A	5	BDQ24	T	PB28A	5	BDQ24	T
GND	GNDIO5	-			GNDIO5	-		
AF9	PB28B	5	BDQ24	C	PB28B	5	BDQ24	C
W13	PB29A	5	BDQ33	T	PB29A	5	BDQ33	T
AA14	PB29B	5	BDQ33	C	PB29B	5	BDQ33	C
AE10	PB30A	5	BDQ33	T	PB30A	5	BDQ33	T
AF10	PB30B	5	BDQ33	C	PB30B	5	BDQ33	C
W14	PB31A	5	BDQ33	T	PB31A	5	BDQ33	T
AB13	PB31B	5	BDQ33	C	PB31B	5	BDQ33	C

LFE2-20E/SE and LFE2-35E/SE Logic Signal Connections: 672 fpBGA

LFE2-20E/20SE					LFE2-35E/35SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
VCCIO	VCCIO5	5			VCCIO5	5		
Y14	PB32A	5	BDQ33	T	PB32A	5	BDQ33	T
AB14	PB32B	5	BDQ33	C	PB32B	5	BDQ33	C
GND	GNDIO5	-			GNDIO5	-		
AE11	PB33A	5	BDQS33	T	PB33A	5	BDQS33	T
AF11	PB33B	5	BDQ33	C	PB33B	5	BDQ33	C
AD14	PB34A	5	BDQ33	T	PB34A	5	BDQ33	T
AA15	PB34B	5	BDQ33	C	PB34B	5	BDQ33	C
AE12	PB35A	5	PCLKT5_0/BDQ33	T	PB35A	5	PCLKT5_0/BDQ33	T
AF12	PB35B	5	PCLKC5_0/BDQ33	C	PB35B	5	PCLKC5_0/BDQ33	C
VCCIO	VCCIO5	5			VCCIO5	5		
GND	GNDIO5	-			GNDIO5	-		
AD15	PB40A	4	PCLKT4_0/BDQ42	T	PB40A	4	PCLKT4_0/BDQ42	T
VCCIO	VCCIO4	4			VCCIO4	4		
AC15	PB40B	4	PCLKC4_0/BDQ42	C	PB40B	4	PCLKC4_0/BDQ42	C
AE13	PB41A	4	BDQ42	T	PB41A	4	BDQ42	T
AF13	PB41B	4	BDQ42	C	PB41B	4	BDQ42	C
AB17	PB42A	4	BDQS42	T	PB42A	4	BDQS42	T
GND	GNDIO4	-			GNDIO4	-		
Y15	PB42B	4	BDQ42	C	PB42B	4	BDQ42	C
AE14	PB43A	4	BDQ42	T	PB43A	4	BDQ42	T
AF14	PB43B	4	BDQ42	C	PB43B	4	BDQ42	C
AA16	PB44A	4	BDQ42	T	PB44A	4	BDQ42	T
VCCIO	VCCIO4	4			VCCIO4	4		
W15	PB44B	4	BDQ42	C	PB44B	4	BDQ42	C
AC17	PB45A	4	BDQ42	T	PB45A	4	BDQ42	T
AB16	PB45B	4	BDQ42	C	PB45B	4	BDQ42	C
AE15	PB46A	4	BDQ42	T	PB46A	4	BDQ42	T
GND	GNDIO4	-			GNDIO4	-		
AF15	PB46B	4	BDQ42	C	PB46B	4	BDQ42	C
AE16	PB47A	4	BDQ51	T	PB47A	4	BDQ51	T
AF16	PB47B	4	BDQ51	C	PB47B	4	BDQ51	C
Y16	PB48A	4	BDQ51	T	PB48A	4	BDQ51	T
AB18	PB48B	4	BDQ51	C	PB48B	4	BDQ51	C
AD17	PB49A	4	BDQ51	T	PB49A	4	BDQ51	T
AD18	PB49B	4	BDQ51	C	PB49B	4	BDQ51	C
VCCIO	VCCIO4	4			VCCIO4	4		
AC18	PB50A	4	BDQ51	T	PB50A	4	BDQ51	T
AD19	PB50B	4	BDQ51	C	PB50B	4	BDQ51	C
GND	GNDIO4	-			GNDIO4	-		
AC19	PB51A	4	BDQS51	T	PB51A	4	BDQS51	T
AE17	PB51B	4	BDQ51	C	PB51B	4	BDQ51	C
AB19	PB52A	4	BDQ51	T	PB52A	4	BDQ51	T
AE19	PB52B	4	BDQ51	C	PB52B	4	BDQ51	C
AF17	PB53A	4	BDQ51	T	PB53A	4	BDQ51	T
AE18	PB53B	4	BDQ51	C	PB53B	4	BDQ51	C
VCCIO	VCCIO4	4			VCCIO4	4		

LFE2-20E/SE and LFE2-35E/SE Logic Signal Connections: 672 fpBGA

LFE2-20E/20SE					LFE2-35E/35SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
W16	PB54A	4	BDQ51	T	PB54A	4	BDQ51	T
AA17	PB54B	4	BDQ51	C	PB54B	4	BDQ51	C
AF18	PB55A	4	BDQ51	T	PB55A	4	BDQ51	T
AF19	PB55B	4	BDQ51	C	PB55B	4	BDQ51	C
GND	GNDIO4	-			GNDIO4	-		
AA19	NC	-			PB56A	4	BDQ60	T
W17	NC	-			PB56B	4	BDQ60	C
Y19	NC	-			PB57A	4	BDQ60	T
Y17	NC	-			PB57B	4	BDQ60	C
AF20	NC	-			NC	-		
VCCIO	VCCIO4	4			VCCIO4	4		
AE20	NC	-			NC	-		
AA20	NC	-			NC	-		
W18	NC	-			NC	-		
AD20	NC	-			NC	-		
GND	GNDIO4	-			GNDIO4	-		
AE21	NC	-			NC	-		
AF21	NC	-			NC	-		
AF22	NC	-			NC	-		
VCCIO	VCCIO4	4			VCCIO4	4		
GND	GNDIO4	-			GNDIO4	-		
AE22	PB56A	4	BDQ60	T	PB65A	4	BDQ69	T
AD22	PB56B	4	BDQ60	C	PB65B	4	BDQ69	C
AF23	PB57A	4	BDQ60	T	PB66A	4	BDQ69	T
AE23	PB57B	4	BDQ60	C	PB66B	4	BDQ69	C
AD23	PB58A	4	BDQ60	T	PB67A	4	BDQ69	T
AC23	PB58B	4	BDQ60	C	PB67B	4	BDQ69	C
VCCIO	VCCIO4	4			VCCIO4	4		
AB20	PB59A	4	BDQ60	T	PB68A	4	BDQ69	T
AC20	PB59B	4	BDQ60	C	PB68B	4	BDQ69	C
GND	GNDIO4	-			GNDIO4	-		
AB21	PB60A	4	BDQS60	T	PB69A	4	BDQS69	T
AC22	PB60B	4	BDQ60	C	PB69B	4	BDQ69	C
W19	PB61A	4	BDQ60	T	PB70A	4	BDQ69	T
AA21	PB61B	4	BDQ60	C	PB70B	4	BDQ69	C
AF24	PB62A	4	BDQ60	T	PB71A	4	BDQ69	T
AE24	PB62B	4	BDQ60	C	PB71B	4	BDQ69	C
VCCIO	VCCIO4	4			VCCIO4	4		
Y20	PB63A	4	BDQ60	T	PB72A	4	BDQ69	T
AB22	PB63B	4	BDQ60	C	PB72B	4	BDQ69	C
Y21	PB64A	4	VREF2_4/BDQ60	T	PB73A	4	VREF2_4/BDQ69	T
AB23	PB64B	4	VREF1_4/BDQ60	C	PB73B	4	VREF1_4/BDQ69	C
GND	GNDIO4	-			GNDIO4	-		
AD24	CFG2	8			CFG2	8		
W20	CFG1	8			CFG1	8		
AC24	CFG0	8			CFG0	8		
V19	PROGRAMN	8			PROGRAMN	8		

LFE2-20E/SE and LFE2-35E/SE Logic Signal Connections: 672 fpBGA

LFE2-20E/20SE					LFE2-35E/35SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
AA22	CCLK	8			CCLK	8		
AB24	INITN	8			INITN	8		
AD25	DONE	8			DONE	8		
GND	GNDIO8	-			GNDIO8	-		
W21	PR44B	8	WRITEN	C	PR58B	8	WRITEN	C
Y22	PR44A	8	CS1N	T	PR58A	8	CS1N	T
AC25	PR43B	8	CSN	C	PR57B	8	CSN	C
AB25	PR43A	8	D0/SPIFASTN	T	PR57A	8	D0/SPIFASTN	T
VCCIO	VCCIO8	8			VCCIO8	8		
AD26	PR42B	8	D1	C	PR56B	8	D1	C
AC26	PR42A	8	D2	T	PR56A	8	D2	T
Y23	PR41B	8	D3	C	PR55B	8	D3	C
GND	GNDIO8	-			GNDIO8	-		
W22	PR41A	8	D4	T	PR55A	8	D4	T
AA25	PR40B	8	D5	C	PR54B	8	D5	C
AB26	PR40A	8	D6	T	PR54A	8	D6	T
W23	PR39B	8	D7/SPID0	C	PR53B	8	D7/SPID0	C
VCCIO	VCCIO8	8			VCCIO8	8		
V22	PR39A	8	DI/CSSPI0N	T	PR53A	8	DI/CSSPI0N	T
Y24	PR38B	8	DOUT/CSON	C	PR52B	8	DOUT/CSON	C
Y25	PR38A	8	BUSY/SISPI	T	PR52A	8	BUSY/SISPI	T
W24	PR37B	3	RDQ34	C	PR51B	3	RDQ48	C
GND	GNDIO3	-			GNDIO3	-		
V23	PR37A	3	RDQ34	T	PR51A	3	RDQ48	T
AA26	PR36B	3	RDQ34	C (LVDS)*	PR50B	3	RDQ48	C (LVDS)*
Y26	PR36A	3	RDQ34	T (LVDS)*	PR50A	3	RDQ48	T (LVDS)*
U21	PR35B	3	RDQ34	C	PR49B	3	RDQ48	C
VCCIO	VCCIO3	3			VCCIO3	3		
U19	PR35A	3	RDQ34	T	PR49A	3	RDQ48	T
W25	PR34B	3	RDQ34	C (LVDS)*	PR48B	3	RDQ48	C (LVDS)*
W26	PR34A	3	RDQS34	T (LVDS)*	PR48A	3	RDQS48	T (LVDS)*
GND	GNDIO3	-			GNDIO3	-		
V24	PR33B	3	RDQ34	C	PR47B	3	RDQ48	C
V25	PR33A	3	RDQ34	T	PR47A	3	RDQ48	T
V26	PR32B	3	RDQ34	C (LVDS)*	PR46B	3	RDQ48	C (LVDS)*
U26	PR32A	3	RDQ34	T (LVDS)*	PR46A	3	RDQ48	T (LVDS)*
VCCIO	VCCIO3	3			VCCIO3	3		
U22	PR31B	3	RLM0_GPLLC_FB_A/RDQ34	C	PR45B	3	RLM0_GPLLC_FB_A/RDQ48	C
U23	PR31A	3	RLM0_GPLLT_FB_A/RDQ34	T	PR45A	3	RLM0_GPLLT_FB_A/RDQ48	T
U24	PR30B	3	RLM0_GPLLC_IN_A**/RDQ34	C (LVDS)*	PR44B	3	RLM0_GPLLC_IN_A**/RDQ48	C (LVDS)*
U25	PR30A	3	RLM0_GPLLT_IN_A**/RDQ34	T (LVDS)*	PR44A	3	RLM0_GPLLT_IN_A**/RDQ48	T (LVDS)*
R20	RLM0_PLLCAP	3			RLM0_PLLCAP	3		
P18	VCC	3			VCCPLL	3		
T19	PR28B	3	RLM0_GDLLC_FB_A/RDQ25	C	PR42B	3	RLM0_GDLLC_FB_A/RDQ39	C
U20	PR28A	3	RLM0_GDLLT_FB_A/RDQ25	T	PR42A	3	RLM0_GDLLT_FB_A/RDQ39	T
GND	GNDIO3	-			GNDIO3	-		
T25	PR27B	3	RLM0_GDLLC_IN_A**/RDQ25	C (LVDS)*	PR41B	3	RLM0_GDLLC_IN_A**/RDQ39	C (LVDS)*

LFE2-20E/SE and LFE2-35E/SE Logic Signal Connections: 672 fpBGA

LFE2-20E/20SE					LFE2-35E/35SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
T26	PR27A	3	RLM0_GDLLT_IN_A**/RDQ25	T (LVDS)*	PR41A	3	RLM0_GDLLT_IN_A**/RDQ39	T (LVDS)*
T20	PR26B	3	RDQ25	C	PR40B	3	RDQ39	C
T22	PR26A	3	RDQ25	T	PR40A	3	RDQ39	T
VCCIO	VCCIO3	3			VCCIO3	3		
R26	PR25B	3	RDQ25	C (LVDS)*	PR39B	3	RDQ39	C (LVDS)*
R25	PR25A	3	RDQS25***	T (LVDS)*	PR39A	3	RDQS39***	T (LVDS)*
R22	NC	-			PR38B	3	RDQ39	C
GND	GNDIO3	-			GNDIO3	-		
T21	NC	-			PR38A	3	RDQ39	T
P26	NC	-			NC	-		
P25	NC	-			NC	-		
R24	NC	-			NC	-		
VCCIO	VCCIO3	3			VCCIO3	3		
R23	NC	-			NC	-		
P20	NC	-			NC	-		
R19	NC	-			NC	-		
P21	NC	-			PR34B	3	RDQ31	C
GND	GNDIO3	-			GNDIO3	-		
P19	NC	-			PR34A	3	RDQ31	T
P23	NC	-			PR33B	3	RDQ31	C (LVDS)*
P22	NC	-			PR33A	3	RDQ31	T (LVDS)*
N22	NC	-			PR32B	3	RDQ31	C
VCCIO	VCCIO3	3			VCCIO3	3		
R21	NC	-			PR32A	3	RDQ31	T
N26	NC	-			PR31B	3	RDQ31	C (LVDS)*
N25	NC	-			PR31A	3	RDQS31	T (LVDS)*
GND	GNDIO3	-			GNDIO3	-		
N19	PR24B	3	RDQ25	C	PR30B	3	RDQ31	C
N20	PR24A	3	RDQ25	T	PR30A	3	RDQ31	T
M26	PR23B	3	RDQ25	C (LVDS)*	PR29B	3	RDQ31	C (LVDS)*
M25	PR23A	3	RDQ25	T (LVDS)*	PR29A	3	RDQ31	T (LVDS)*
VCCIO	VCCIO3	3			VCCIO3	3		
N18	PR22B	3	VREF2_3/RDQ25	C	PR28B	3	VREF2_3/RDQ31	C
N21	PR22A	3	VREF1_3/RDQ25	T	PR28A	3	VREF1_3/RDQ31	T
L26	PR21B	3	PCLKC3_0/RDQ25	C (LVDS)*	PR27B	3	PCLKC3_0/RDQ31	C (LVDS)*
L25	PR21A	3	PCLKT3_0/RDQ25	T (LVDS)*	PR27A	3	PCLKT3_0/RDQ31	T (LVDS)*
N24	PR19B	2	PCLKC2_0/RDQ16	C	PR25B	2	PCLKC2_0/RDQ22	C
M23	PR19A	2	PCLKT2_0/RDQ16	T	PR25A	2	PCLKT2_0/RDQ22	T
GND	GNDIO2	-			GNDIO2	-		
L21	PR18B	2	RDQ16	C (LVDS)*	PR24B	2	RDQ22	C (LVDS)*
K22	PR18A	2	RDQ16	T (LVDS)*	PR24A	2	RDQ22	T (LVDS)*
M24	PR17B	2	RDQ16	C	PR23B	2	RDQ22	C
N23	PR17A	2	RDQ16	T	PR23A	2	RDQ22	T
VCCIO	VCCIO2	2			VCCIO2	2		
K26	PR16B	2	RDQ16	C (LVDS)*	PR22B	2	RDQ22	C (LVDS)*
K25	PR16A	2	RDQS16	T (LVDS)*	PR22A	2	RDQS22	T (LVDS)*
M20	PR15B	2	RDQ16	C	PR21B	2	RDQ22	C

LFE2-20E/SE and LFE2-35E/SE Logic Signal Connections: 672 fpBGA

LFE2-20E/20SE					LFE2-35E/35SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
GND	GNDIO2	-			GNDIO2	-		
M19	PR15A	2	RDQ16	T	PR21A	2	RDQ22	T
L22	PR14B	2	RDQ16	C (LVDS)*	PR20B	2	RDQ22	C (LVDS)*
M22	PR14A	2	RDQ16	T (LVDS)*	PR20A	2	RDQ22	T (LVDS)*
K21	PR13B	2	RDQ16	C	PR19B	2	RDQ22	C
VCCIO	VCCIO2	2			VCCIO2	2		
M21	PR13A	2	RDQ16	T	PR19A	2	RDQ22	T
K24	PR12B	2	RDQ16	C (LVDS)*	PR18B	2	RDQ22	C (LVDS)*
J24	PR12A	2	RDQ16	T (LVDS)*	PR18A	2	RDQ22	T (LVDS)*
GND	GNDIO2	-			GNDIO2	-		
VCCIO	VCCIO2	2			VCCIO2	2		
L20	VCC	-			NC	-		
GND	GNDIO2	-			GNDIO2	-		
J26	NC	-			NC	-		
J25	NC	-			NC	-		
J23	NC	-			NC	-		
K23	NC	-			NC	-		
VCCIO	VCCIO2	2			VCCIO2	2		
H26	NC	-			NC	-		
H25	NC	-			NC	-		
H24	NC	-			NC	-		
GND	GNDIO2	-			GNDIO2	-		
H23	NC	-			NC	-		
VCCIO	VCCIO2	2			VCCIO2	2		
G26	PR11B	2	RDQ8	C	PR17B	2	RDQ14	C
GND	GNDIO2	-			GNDIO2	-		
G25	PR11A	2	RDQ8	T	PR17A	2	RDQ14	T
F26	PR10B	2	RDQ8	C (LVDS)*	PR16B	2	RDQ14	C (LVDS)*
F25	PR10A	2	RDQ8	T (LVDS)*	PR16A	2	RDQ14	T (LVDS)*
K20	PR9B	2	RDQ8	C	PR15B	2	RDQ14	C
VCCIO	VCCIO2	2			VCCIO2	2		
L19	PR9A	2	RDQ8	T	PR15A	2	RDQ14	T
E26	PR8B	2	RDQ8	C (LVDS)*	PR14B	2	RDQ14	C (LVDS)*
E25	PR8A	2	RDQS8	T (LVDS)*	PR14A	2	RDQS14	T (LVDS)*
GND	GNDIO2	-			GNDIO2	-		
J22	PR7B	2	RDQ8	C	PR13B	2	RDQ14	C
H22	PR7A	2	RDQ8	T	PR13A	2	RDQ14	T
G24	PR6B	2	RDQ8	C (LVDS)*	PR12B	2	RDQ14	C (LVDS)*
G23	PR6A	2	RDQ8	T (LVDS)*	PR12A	2	RDQ14	T (LVDS)*
VCCIO	VCCIO2	2			VCCIO2	2		
K19	PR5B	2	RDQ8	C	PR11B	2	RDQ14	C
J19	PR5A	2	RDQ8	T	PR11A	2	RDQ14	T
D26	PR4B	2	RDQ8	C (LVDS)*	PR10B	2	RDQ14	C (LVDS)*
C26	PR4A	2	RDQ8	T (LVDS)*	PR10A	2	RDQ14	T (LVDS)*
F22	NC	-			PR9B	2	RDQ6	C
E24	NC	-			PR9A	2	RDQ6	T
GND	GNDIO2	-			GNDIO2	-		

LFE2-20E/SE and LFE2-35E/SE Logic Signal Connections: 672 fpBGA

LFE2-20E/20SE					LFE2-35E/35SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
D25	NC	-			PR8B	2	RDQ6	C (LVDS)*
C25	NC	-			PR8A	2	RDQ6	T (LVDS)*
D24	NC	-			PR7B	2	RDQ6	C
B25	NC	-			PR7A	2	RDQ6	T
VCCIO	VCCIO2	2			VCCIO2	2		
H21	NC	-			PR6B	2	RDQ6	C (LVDS)*
G22	NC	-			PR6A	2	RDQS6	T (LVDS)*
B24	NC	-			PR5B	2	RDQ6	C
GND	GNDIO2	-			GNDIO2	-		
C24	NC	-			PR5A	2	RDQ6	T
D23	NC	-			PR4B	2	RDQ6	C (LVDS)*
C23	NC	-			PR4A	2	RDQ6	T (LVDS)*
G21	PR3B	2		C	PR3B	2	RDQ6	C
VCCIO	VCCIO2	2			VCCIO2	2		
H20	PR3A	2		T	PR3A	2	RDQ6	T
GND	GNDIO2	-			GNDIO2	-		
E22	PR2B	2	VREF2_2	C (LVDS)*	PR2B	2	VREF2_2/RDQ6	C (LVDS)*
F21	PR2A	2	VREF1_2	T (LVDS)*	PR2A	2	VREF1_2/RDQ6	T (LVDS)*
E23	PT64B	1	VREF2_1	C	PT73B	1	VREF2_1	C
GND	GNDIO1	-			GNDIO1	-		
D22	PT64A	1	VREF1_1	T	PT73A	1	VREF1_1	T
G20	PT63B	1		C	PT72B	1		C
J18	PT63A	1		T	PT72A	1		T
F20	PT62B	1		C	PT71B	1		C
VCCIO	VCCIO1	1			VCCIO1	1		
H19	PT62A	1		T	PT71A	1		T
A24	PT61B	1		C	PT70B	1		C
A23	PT61A	1		T	PT70A	1		T
E21	PT60B	1		C	PT69B	1		C
F19	PT60A	1		T	PT69A	1		T
C22	PT59B	1		C	PT68B	1		C
GND	GNDIO1	-			GNDIO1	-		
E20	PT59A	1		T	PT68A	1		T
B22	PT58B	1		C	PT67B	1		C
VCCIO	VCCIO1	1			VCCIO1	1		
B23	PT58A	1		T	PT67A	1		T
C20	PT57B	1		C	PT66B	1		C
D20	PT57A	1		T	PT66A	1		T
A22	PT56B	1		C	PT65B	1		C
A21	PT56A	1		T	PT65A	1		T
GND	GNDIO1	-			GNDIO1	-		
E19	NC	-			NC	-		
C19	NC	-			NC	-		
VCCIO	VCCIO1	1			VCCIO1	1		
B21	NC	-			NC	-		
B20	NC	-			NC	-		
D19	NC	-			NC	-		

LFE2-20E/SE and LFE2-35E/SE Logic Signal Connections: 672 fpBGA

LFE2-20E/20SE					LFE2-35E/35SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
B19	NC	-			NC	-		
GND	GNDIO1	-			GNDIO1	-		
G17	NC	-			NC	-		
E18	NC	-			NC	-		
G19	NC	-			NC	-		
F17	NC	-			NC	-		
VCCIO	VCCIO1	1			VCCIO1	1		
A20	NC	-			NC	-		
A19	NC	-			NC	-		
E17	NC	-			NC	-		
D18	NC	-			NC	-		
B18	PT55B	1		C	PT55B	1		C
GND	GNDIO1	-			GNDIO1	-		
A18	PT55A	1		T	PT55A	1		T
E16	PT54B	1		C	PT54B	1		C
G16	PT54A	1		T	PT54A	1		T
F16	PT53B	1		C	PT53B	1		C
VCCIO	VCCIO1	1			VCCIO1	1		
H18	PT53A	1		T	PT53A	1		T
A17	PT52B	1		C	PT52B	1		C
B17	PT52A	1		T	PT52A	1		T
C18	PT51B	1		C	PT51B	1		C
B16	PT51A	1		T	PT51A	1		T
C17	PT50B	1		C	PT50B	1		C
GND	GNDIO1	-			GNDIO1	-		
D17	PT50A	1		T	PT50A	1		T
E15	PT49B	1		C	PT49B	1		C
VCCIO	VCCIO1	1			VCCIO1	1		
G15	PT49A	1		T	PT49A	1		T
A16	PT48B	1		C	PT48B	1		C
B15	PT48A	1		T	PT48A	1		T
D15	PT47B	1		C	PT47B	1		C
F15	PT47A	1		T	PT47A	1		T
A14	PT46B	1		C	PT46B	1		C
B14	PT46A	1		T	PT46A	1		T
GND	GNDIO1	-			GNDIO1	-		
C15	PT45B	1		C	PT45B	1		C
A15	PT45A	1		T	PT45A	1		T
A13	PT44B	1		C	PT44B	1		C
B13	PT44A	1		T	PT44A	1		T
VCCIO	VCCIO1	1			VCCIO1	1		
H17	PT43B	1		C	PT43B	1		C
H15	PT43A	1		T	PT43A	1		T
D13	PT42B	1		C	PT42B	1		C
C14	PT42A	1		T	PT42A	1		T
GND	GNDIO1	-			GNDIO1	-		
G14	PT41B	1		C	PT41B	1		C

LFE2-20E/SE and LFE2-35E/SE Logic Signal Connections: 672 fpBGA

LFE2-20E/20SE					LFE2-35E/35SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
E14	PT41A	1		T	PT41A	1		T
A12	PT40B	1		C	PT40B	1		C
B12	PT40A	1		T	PT40A	1		T
VCCIO	VCCIO1	1			VCCIO1	1		
F14	PT39B	1	PCLKC1_0	C	PT39B	1	PCLKC1_0	C
D14	PT39A	1	PCLKT1_0	T	PT39A	1	PCLKT1_0	T
H16	XRES	1			XRES	1		
H14	PT37B	0	PCLKC0_0	C	PT37B	0	PCLKC0_0	C
GND	GNDIO0	-			GNDIO0	-		
H13	PT37A	0	PCLKT0_0	T	PT37A	0	PCLKT0_0	T
A11	PT36B	0		C	PT36B	0		C
B11	PT36A	0		T	PT36A	0		T
C13	PT35B	0		C	PT35B	0		C
VCCIO	VCCIO0	0			VCCIO0	0		
E13	PT35A	0		T	PT35A	0		T
D12	PT34B	0		C	PT34B	0		C
F13	PT34A	0		T	PT34A	0		T
A10	PT33B	0		C	PT33B	0		C
B10	PT33A	0		T	PT33A	0		T
C12	PT32B	0		C	PT32B	0		C
GND	GNDIO0	-			GNDIO0	-		
C10	PT32A	0		T	PT32A	0		T
G13	PT31B	0		C	PT31B	0		C
VCCIO	VCCIO0	0			VCCIO0	0		
H12	PT31A	0		T	PT31A	0		T
A9	PT30B	0		C	PT30B	0		C
B9	PT30A	0		T	PT30A	0		T
E12	PT29B	0		C	PT29B	0		C
G12	PT29A	0		T	PT29A	0		T
A8	PT28B	0		C	PT28B	0		C
B8	PT28A	0		T	PT28A	0		T
GND	GNDIO0	-			GNDIO0	-		
E11	PT27B	0		C	PT27B	0		C
C9	PT27A	0		T	PT27A	0		T
A7	PT26B	0		C	PT26B	0		C
B7	PT26A	0		T	PT26A	0		T
VCCIO	VCCIO0	0			VCCIO0	0		
F12	PT25B	0		C	PT25B	0		C
D10	PT25A	0		T	PT25A	0		T
H11	PT24B	0		C	PT24B	0		C
G11	PT24A	0		T	PT24A	0		T
GND	GNDIO0	-			GNDIO0	-		
A6	PT23B	0		C	PT23B	0		C
B6	PT23A	0		T	PT23A	0		T
D8	PT22B	0		C	PT22B	0		C
C8	PT22A	0		T	PT22A	0		T
VCCIO	VCCIO0	0			VCCIO0	0		

LFE2-20E/SE and LFE2-35E/SE Logic Signal Connections: 672 fpBGA

LFE2-20E/20SE					LFE2-35E/35SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
F11	PT21B	0		C	PT21B	0		C
E10	PT21A	0		T	PT21A	0		T
E9	PT20B	0		C	PT20B	0		C
D9	PT20A	0		T	PT20A	0		T
G10	PT19B	0		C	PT19B	0		C
GND	GNDIO0	-			GNDIO0	-		
H10	PT19A	0		T	PT19A	0		T
A5	PT18B	0		C	PT18B	0		C
B5	PT18A	0		T	PT18A	0		T
C7	PT17B	0		C	PT17B	0		C
VCCIO	VCCIO0	0			VCCIO0	0		
D7	PT17A	0		T	PT17A	0		T
E8	PT16B	0		C	PT16B	0		C
F10	PT16A	0		T	PT16A	0		T
F8	PT15B	0		C	PT15B	0		C
H9	PT15A	0		T	PT15A	0		T
C5	PT14B	0		C	PT14B	0		C
GND	GNDIO0	-			GNDIO0	-		
D5	PT14A	0		T	PT14A	0		T
B4	PT13B	0			PT13B	0		
VCCIO	VCCIO0	0			VCCIO0	0		
GND	GNDIO0	-			GNDIO0	-		
VCCIO	VCCIO0	0			VCCIO0	0		
GND	GNDIO0	-			GNDIO0	-		
VCCIO	VCCIO0	0			VCCIO0	0		
C4	PT10B	0		C	PT10B	0		C
GND	GNDIO0	-			GNDIO0	-		
C3	PT10A	0		T	PT10A	0		T
A4	PT9B	0		C	PT9B	0		C
A3	PT9A	0		T	PT9A	0		T
B3	PT8B	0		C	PT8B	0		C
VCCIO	VCCIO0	0			VCCIO0	0		
B2	PT8A	0		T	PT8A	0		T
D4	PT7B	0		C	PT7B	0		C
D3	PT7A	0		T	PT7A	0		T
C2	PT6B	0		C	PT6B	0		C
C1	PT6A	0		T	PT6A	0		T
G8	PT5B	0		C	PT5B	0		C
GND	GNDIO0	-			GNDIO0	-		
G7	PT5A	0		T	PT5A	0		T
E7	PT4B	0		C	PT4B	0		C
VCCIO	VCCIO0	0			VCCIO0	0		
F7	PT4A	0		T	PT4A	0		T
E6	PT3B	0		C	PT3B	0		C
E5	PT3A	0		T	PT3A	0		T
G6	PT2B	0	VREF2_0	C	PT2B	0	VREF2_0	C
G5	PT2A	0	VREF1_0	T	PT2A	0	VREF1_0	T

LFE2-20E/SE and LFE2-35E/SE Logic Signal Connections: 672 fpBGA

LFE2-20E/20SE					LFE2-35E/35SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
L12	VCC	-			VCC	-		
L13	VCC	-			VCC	-		
L14	VCC	-			VCC	-		
L15	VCC	-			VCC	-		
M11	VCC	-			VCC	-		
M12	VCC	-			VCC	-		
M15	VCC	-			VCC	-		
M16	VCC	-			VCC	-		
N11	VCC	-			VCC	-		
N16	VCC	-			VCC	-		
P11	VCC	-			VCC	-		
P16	VCC	-			VCC	-		
R11	VCC	-			VCC	-		
R12	VCC	-			VCC	-		
R15	VCC	-			VCC	-		
R16	VCC	-			VCC	-		
T12	VCC	-			VCC	-		
T13	VCC	-			VCC	-		
T14	VCC	-			VCC	-		
T15	VCC	-			VCC	-		
D11	VCCIO0	0			VCCIO0	0		
D6	VCCIO0	0			VCCIO0	0		
G9	VCCIO0	0			VCCIO0	0		
K12	VCCIO0	0			VCCIO0	0		
J12	VCCIO0	0			VCCIO0	0		
D16	VCCIO1	1			VCCIO1	1		
D21	VCCIO1	1			VCCIO1	1		
G18	VCCIO1	1			VCCIO1	1		
J15	VCCIO1	1			VCCIO1	1		
K15	VCCIO1	1			VCCIO1	1		
F23	VCCIO2	2			VCCIO2	2		
J20	VCCIO2	2			VCCIO2	2		
L23	VCCIO2	2			VCCIO2	2		
M17	VCCIO2	2			VCCIO2	2		
M18	VCCIO2	2			VCCIO2	2		
AA23	VCCIO3	3			VCCIO3	3		
R17	VCCIO3	3			VCCIO3	3		
R18	VCCIO3	3			VCCIO3	3		
T23	VCCIO3	3			VCCIO3	3		
V20	VCCIO3	3			VCCIO3	3		
AC16	VCCIO4	4			VCCIO4	4		
AC21	VCCIO4	4			VCCIO4	4		
U15	VCCIO4	4			VCCIO4	4		
V15	VCCIO4	4			VCCIO4	4		
Y18	VCCIO4	4			VCCIO4	4		
AC11	VCCIO5	5			VCCIO5	5		
AC6	VCCIO5	5			VCCIO5	5		

LFE2-20E/SE and LFE2-35E/SE Logic Signal Connections: 672 fpBGA

LFE2-20E/20SE					LFE2-35E/35SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
U12	VCCIO5	5			VCCIO5	5		
V12	VCCIO5	5			VCCIO5	5		
Y9	VCCIO5	5			VCCIO5	5		
AA4	VCCIO6	6			VCCIO6	6		
R10	VCCIO6	6			VCCIO6	6		
R9	VCCIO6	6			VCCIO6	6		
T4	VCCIO6	6			VCCIO6	6		
V7	VCCIO6	6			VCCIO6	6		
F4	VCCIO7	7			VCCIO7	7		
J7	VCCIO7	7			VCCIO7	7		
L4	VCCIO7	7			VCCIO7	7		
M10	VCCIO7	7			VCCIO7	7		
M9	VCCIO7	7			VCCIO7	7		
AE25	VCCIO8	8			VCCIO8	8		
V18	VCCIO8	8			VCCIO8	8		
J10	VCCAUX	-			VCCAUX	-		
J11	VCCAUX	-			VCCAUX	-		
J16	VCCAUX	-			VCCAUX	-		
J17	VCCAUX	-			VCCAUX	-		
K18	VCCAUX	-			VCCAUX	-		
K9	VCCAUX	-			VCCAUX	-		
L18	VCCAUX	-			VCCAUX	-		
L9	VCCAUX	-			VCCAUX	-		
T18	VCCAUX	-			VCCAUX	-		
T9	VCCAUX	-			VCCAUX	-		
U18	VCCAUX	-			VCCAUX	-		
U9	VCCAUX	-			VCCAUX	-		
V10	VCCAUX	-			VCCAUX	-		
V11	VCCAUX	-			VCCAUX	-		
V16	VCCAUX	-			VCCAUX	-		
V17	VCCAUX	-			VCCAUX	-		
A2	GND	-			GND	-		
A25	GND	-			GND	-		
AA18	GND	-			GND	-		
AA24	GND	-			GND	-		
AA3	GND	-			GND	-		
AA9	GND	-			GND	-		
AD11	GND	-			GND	-		
AD16	GND	-			GND	-		
AD21	GND	-			GND	-		
AD6	GND	-			GND	-		
AE1	GND	-			GND	-		
AE26	GND	-			GND	-		
AF2	GND	-			GND	-		
AF25	GND	-			GND	-		
B1	GND	-			GND	-		
B26	GND	-			GND	-		

LFE2-20E/SE and LFE2-35E/SE Logic Signal Connections: 672 fpBGA

LFE2-20E/20SE					LFE2-35E/35SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
C11	GND	-			GND	-		
C16	GND	-			GND	-		
C21	GND	-			GND	-		
C6	GND	-			GND	-		
F18	GND	-			GND	-		
F24	GND	-			GND	-		
F3	GND	-			GND	-		
F9	GND	-			GND	-		
J13	GND	-			GND	-		
J14	GND	-			GND	-		
J21	GND	-			GND	-		
J6	GND	-			GND	-		
K10	GND	-			GND	-		
K11	GND	-			GND	-		
K13	GND	-			GND	-		
K14	GND	-			GND	-		
K16	GND	-			GND	-		
K17	GND	-			GND	-		
L10	GND	-			GND	-		
L11	GND	-			GND	-		
L16	GND	-			GND	-		
L17	GND	-			GND	-		
L24	GND	-			GND	-		
L3	GND	-			GND	-		
M13	GND	-			GND	-		
M14	GND	-			GND	-		
N10	GND	-			GND	-		
N12	GND	-			GND	-		
N13	GND	-			GND	-		
N14	GND	-			GND	-		
N15	GND	-			GND	-		
N17	GND	-			GND	-		
P10	GND	-			GND	-		
P12	GND	-			GND	-		
P13	GND	-			GND	-		
P14	GND	-			GND	-		
P15	GND	-			GND	-		
P17	GND	-			GND	-		
R13	GND	-			GND	-		
R14	GND	-			GND	-		
T10	GND	-			GND	-		
T11	GND	-			GND	-		
T16	GND	-			GND	-		
T17	GND	-			GND	-		
T24	GND	-			GND	-		
T3	GND	-			GND	-		
U10	GND	-			GND	-		

LFE2-20E/SE and LFE2-35E/SE Logic Signal Connections: 672 fpBGA

LFE2-20E/20SE					LFE2-35E/35SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
U11	GND	-			GND	-		
U13	GND	-			GND	-		
U14	GND	-			GND	-		
U16	GND	-			GND	-		
U17	GND	-			GND	-		
V13	GND	-			GND	-		
V14	GND	-			GND	-		
V21	GND	-			GND	-		
V6	GND	-			GND	-		
M3	NC	-			NC	-		
N6	NC	-			NC	-		
P24	NC	-			NC	-		

* Supports true LVDS. Other differential signals must be emulated with external resistors.
** These dedicated input pins can be used for GPLLs or GDLLs within the respective quadrant.
***Due to packaging bond out option, this DQS does not have all the necessary DQ pins bonded out for a full 8-bit data width.

Note: VCCIO and GND pads are used to determine the average DC current drawn by I/Os between GND/VCCIO connections, or between the last GND/VCCIO in an I/O bank and the end of an I/O bank. The substrate pads listed in the Pin Table do not necessarily have a one to one connection with a package ball or pin.

LFE2-50E/SE and LFE2-70E/SE Logic Signal Connections: 672 fpBGA

LFE2-50E/SE					LFE2-70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
D2	PL2A	7	VREF2_7	T (LVDS)*	PL2A	7	VREF2_7	T (LVDS)*
D1	PL2B	7	VREF1_7	C (LVDS)*	PL2B	7	VREF1_7	C (LVDS)*
GND	GNDIO7	-			GNDIO7	-		
F6	PL5A	7	LDQ8	T	PL18A	7	LDQ21	T
F5	PL5B	7	LDQ8	C	PL18B	7	LDQ21	C
VCCIO	VCCIO7	7			VCCIO7	7		
E4	PL6A	7	LDQ8	T (LVDS)*	PL19A	7	LDQ21	T (LVDS)*
E3	PL6B	7	LDQ8	C (LVDS)*	PL19B	7	LDQ21	C (LVDS)*
E2	PL7A	7	LDQ8	T	PL20A	7	LDQ21	T
E1	PL7B	7	LDQ8	C	PL20B	7	LDQ21	C
GND	GNDIO7	-			GNDIO7	-		
H6	PL8A	7	LDQS8	T (LVDS)*	PL21A	7	LDQS21	T (LVDS)*
H5	PL8B	7	LDQ8	C (LVDS)*	PL21B	7	LDQ21	C (LVDS)*
F2	PL9A	7	LDQ8	T	PL22A	7	LDQ21	T
VCCIO	VCCIO7	7			VCCIO7	7		
F1	PL9B	7	LDQ8	C	PL22B	7	LDQ21	C
H8	PL10A	7	LDQ8	T (LVDS)*	PL23A	7	LDQ21	T (LVDS)*
J9	PL10B	7	LDQ8	C (LVDS)*	PL23B	7	LDQ21	C (LVDS)*
G4	PL11A	7	LDQ8	T	PL24A	7	LDQ21	T
GND	GNDIO7	-			GNDIO7	-		
G3	PL11B	7	LDQ8	C	PL24B	7	LDQ21	C
H7	PL12A	7	LDQ16	T (LVDS)*	PL25A	7	LDQ29	T (LVDS)*
J8	PL12B	7	LDQ16	C (LVDS)*	PL25B	7	LDQ29	C (LVDS)*
G2	PL13A	7	LDQ16	T	PL26A	7	LDQ29	T
G1	PL13B	7	LDQ16	C	PL26B	7	LDQ29	C
H3	PL14A	7	LDQ16	T (LVDS)*	PL27A	7	LDQ29	T (LVDS)*
VCCIO	VCCIO7	7			VCCIO7	7		
H4	PL14B	7	LDQ16	C (LVDS)*	PL27B	7	LDQ29	C (LVDS)*
J5	PL15A	7	LDQ16	T	PL28A	7	LDQ29	T
J4	PL15B	7	LDQ16	C	PL28B	7	LDQ29	C
J3	PL16A	7	LDQS16	T (LVDS)*	PL29A	7	LDQS29	T (LVDS)*
GND	GNDIO7	-			GNDIO7	-		
K4	PL16B	7	LDQ16	C (LVDS)*	PL29B	7	LDQ29	C (LVDS)*
H1	PL17A	7	LDQ16	T	PL30A	7	LDQ29	T
H2	PL17B	7	LDQ16	C	PL30B	7	LDQ29	C
VCCIO	VCCIO7	7			VCCIO7	7		
K6	PL18A	7	LDQ16	T (LVDS)*	PL31A	7	LDQ29	T (LVDS)*
K7	PL18B	7	LDQ16	C (LVDS)*	PL31B	7	LDQ29	C (LVDS)*
J1	PL19A	7	LDQ16	T	PL32A	7	LDQ29	T
J2	PL19B	7	LDQ16	C	PL32B	7	LDQ29	C
GND	GNDIO7	-			GNDIO7	-		
VCCIO	VCCIO7	7			VCCIO7	7		
K3	PL23A	7	LDQ24	T	PL36A	7	LDQ37	T
K2	PL23B	7	LDQ24	C	PL36B	7	LDQ37	C
GND	GNDIO7	-			GNDIO7	-		
K1	PL24A	7	LDQS24***	T (LVDS)*	PL37A	7	LDQS37***	T (LVDS)*
L2	PL24B	7	LDQ24	C (LVDS)*	PL37B	7	LDQ37	C (LVDS)*

LFE2-50E/SE and LFE2-70E/SE Logic Signal Connections: 672 fpBGA

LFE2-50E/SE					LFE2-70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
L1	PL25A	7	LUM0_SPLLT_IN_A/LDQ24	T	PL38A	7	LUM0_SPLLT_IN_A/LDQ37	T
VCCIO	VCCIO7	7			VCCIO7	7		
M2	PL25B	7	LUM0_SPLLC_IN_A/LDQ24	C	PL38B	7	LUM0_SPLLC_IN_A/LDQ37	C
M1	PL26A	7	LUM0_SPLLT_FB_A/LDQ24	T	PL39A	7	LUM0_SPLLT_FB_A/LDQ37	T
N2	PL26B	7	LUM0_SPLLC_FB_A/LDQ24	C	PL39B	7	LUM0_SPLLC_FB_A/LDQ37	C
GND	GNDIO7	-			GNDIO7	-		
M8	VCCPLL	7			NC	-		
VCCIO	VCCIO7	7			VCCIO7	7		
GND	GNDIO7	-			GNDIO7	-		
N1	PL37A	7	LDQ41		PL50A	7	LDQ54	
L8	PL38A	7	LDQ41	T	PL51A	7	LDQ54	T
K8	PL38B	7	LDQ41	C	PL51B	7	LDQ54	C
VCCIO	VCCIO7	7			VCCIO7	7		
L6	PL39A	7	LDQ41	T (LVDS)*	PL52A	7	LDQ54	T (LVDS)*
K5	PL39B	7	LDQ41	C (LVDS)*	PL52B	7	LDQ54	C (LVDS)*
L7	PL40A	7	LDQ41	T	PL53A	7	LDQ54	T
L5	PL40B	7	LDQ41	C	PL53B	7	LDQ54	C
GND	GNDIO7	-			GNDIO7	-		
P1	PL41A	7	LDQS41	T (LVDS)*	PL54A	7	LDQS54	T (LVDS)*
P2	PL41B	7	LDQ41	C (LVDS)*	PL54B	7	LDQ54	C (LVDS)*
M6	PL42A	7	LDQ41	T	PL55A	7	LDQ54	T
VCCIO	VCCIO7	7			VCCIO7	7		
N8	PL42B	7	LDQ41	C	PL55B	7	LDQ54	C
R1	PL43A	7	LDQ41	T (LVDS)*	PL56A	7	LDQ54	T (LVDS)*
R2	PL43B	7	LDQ41	C (LVDS)*	PL56B	7	LDQ54	C (LVDS)*
M7	PL44A	7	PCLKT7_0/LDQ41	T	PL57A	7	PCLKT7_0/LDQ54	T
GND	GNDIO7	-			GNDIO7	-		
N9	PL44B	7	PCLKC7_0/LDQ41	C	PL57B	7	PCLKC7_0/LDQ54	C
M4	PL46A	6	PCLKT6_0/LDQ50	T (LVDS)*	PL59A	6	PCLKT6_0/LDQ63	T (LVDS)*
M5	PL46B	6	PCLKC6_0/LDQ50	C (LVDS)*	PL59B	6	PCLKC6_0/LDQ63	C (LVDS)*
N7	PL47A	6	VREF2_6/LDQ50	T	PL60A	6	VREF2_6/LDQ63	T
P9	PL47B	6	VREF1_6/LDQ50	C	PL60B	6	VREF1_6/LDQ63	C
N3	PL48A	6	LDQ50	T (LVDS)*	PL61A	6	LDQ63	T (LVDS)*
VCCIO	VCCIO6	6			VCCIO6	6		
N4	PL48B	6	LDQ50	C (LVDS)*	PL61B	6	LDQ63	C (LVDS)*
N5	PL49A	6	LDQ50	T	PL62A	6	LDQ63	T
P7	PL49B	6	LDQ50	C	PL62B	6	LDQ63	C
T1	PL50A	6	LDQS50	T (LVDS)*	PL63A	6	LDQS63	T (LVDS)*
GND	GNDIO6	-			GNDIO6	-		
T2	PL50B	6	LDQ50	C (LVDS)*	PL63B	6	LDQ63	C (LVDS)*
P8	PL51A	6	LDQ50	T	PL64A	6	LDQ63	T
P6	PL51B	6	LDQ50	C	PL64B	6	LDQ63	C
VCCIO	VCCIO6	6			VCCIO6	6		
P5	PL52A	6	LDQ50	T (LVDS)*	PL65A	6	LDQ63	T (LVDS)*
P4	PL52B	6	LDQ50	C (LVDS)*	PL65B	6	LDQ63	C (LVDS)*
U1	PL53A	6	LDQ50	T	PL66A	6	LDQ63	T
V1	PL53B	6	LDQ50	C	PL66B	6	LDQ63	C

LFE2-50E/SE and LFE2-70E/SE Logic Signal Connections: 672 fpBGA

LFE2-50E/SE					LFE2-70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
GND	GNDIO6	-			GNDIO6	-		
P3	PL54A	6	LDQ58	T (LVDS)*	PL67A	6	LDQ71	T (LVDS)*
R3	PL54B	6	LDQ58	C (LVDS)*	PL67B	6	LDQ71	C (LVDS)*
R4	PL55A	6	LDQ58	T	PL68A	6	LDQ71	T
U2	PL55B	6	LDQ58	C	PL68B	6	LDQ71	C
VCCIO	VCCIO6	6			VCCIO6	6		
V2	PL56A	6	LDQ58	T (LVDS)*	PL69A	6	LDQ71	T (LVDS)*
W2	PL56B	6	LDQ58	C (LVDS)*	PL69B	6	LDQ71	C (LVDS)*
T6	PL57A	6	LDQ58	T	PL70A	6	LDQ71	T
R5	PL57B	6	LDQ58	C	PL70B	6	LDQ71	C
GND	GNDIO6	-			GNDIO6	-		
R6	PL58A	6	LDQS58	T (LVDS)*	PL71A	6	LDQS71	T (LVDS)*
R7	PL58B	6	LDQ58	C (LVDS)*	PL71B	6	LDQ71	C (LVDS)*
W1	PL59A	6	LDQ58	T	PL72A	6	LDQ71	T
VCCIO	VCCIO6	6			VCCIO6	6		
Y2	PL59B	6	LDQ58	C	PL72B	6	LDQ71	C
Y1	PL60A	6	LLM0_GDLLT_IN_A**/LDQ58	T (LVDS)*	PL73A	6	LLM0_GDLLT_IN_A**/LDQ71	T (LVDS)*
AA2	PL60B	6	LLM0_GDLLC_IN_A**/LDQ58	C (LVDS)*	PL73B	6	LLM0_GDLLC_IN_A**/LDQ71	C (LVDS)*
T5	PL61A	6	LLM0_GDLLT_FB_A/LDQ58	T	PL74A	6	LLM0_GDLLT_FB_A/LDQ71	T
GND	GNDIO6	-			GNDIO6	-		
T7	PL61B	6	LLM0_GDLLC_FB_D/LDQ58	C	PL74B	6	LLM0_GDLLC_FB_D/LDQ71	C
R8	VCCPLL	6			VCCPLL	-		
T8	LLM0_PLLCAP	6			LLM0_PLLCAP	6		
U3	PL63A	6	LLM0_GPLLT_IN_A**/LDQ67	T (LVDS)*	PL76A	6	LLM0_GPLLT_IN_A**/LDQ80	T (LVDS)*
U4	PL63B	6	LLM0_GPLLC_IN_A**/LDQ67	C (LVDS)*	PL76B	6	LLM0_GPLLC_IN_A**/LDQ80	C (LVDS)*
V3	PL64A	6	LLM0_GPLLT_FB_A/LDQ67	T	PL77A	6	LLM0_GPLLT_FB_A/LDQ80	T
U5	PL64B	6	LLM0_GPLLC_FB_A/LDQ67	C	PL77B	6	LLM0_GPLLC_FB_A/LDQ80	C
V4	PL65A	6	LDQ67	T (LVDS)*	PL78A	6	LDQ80	T (LVDS)*
VCCIO	VCCIO6	6			VCCIO6	6		
V5	PL65B	6	LDQ67	C (LVDS)*	PL78B	6	LDQ80	C (LVDS)*
Y3	PL66A	6	LDQ67	T	PL79A	6	LDQ80	T
Y4	PL66B	6	LDQ67	C	PL79B	6	LDQ80	C
W3	PL67A	6	LDQS67	T (LVDS)*	PL80A	6	LDQS80	T (LVDS)*
GND	GNDIO6	-			GNDIO6	-		
W4	PL67B	6	LDQ67	C (LVDS)*	PL80B	6	LDQ80	C (LVDS)*
AA1	PL68A	6	LDQ67	T	PL81A	6	LDQ80	T
AB1	PL68B	6	LDQ67	C	PL81B	6	LDQ80	C
VCCIO	VCCIO6	6			VCCIO6	6		
U8	PL69A	6	LDQ67	T (LVDS)*	PL82A	6	LDQ80	T (LVDS)*
U7	PL69B	6	LDQ67	C (LVDS)*	PL82B	6	LDQ80	C (LVDS)*
V8	PL70A	6	LDQ67	T	PL83A	6	LDQ80	T
U6	PL70B	6	LDQ67	C	PL83B	6	LDQ80	C
GND	GNDIO6	-			GNDIO6	-		
W6	PL71A	6	LDQ75	T (LVDS)*	PL84A	6	LDQ88	T (LVDS)*
W5	PL71B	6	LDQ75	C (LVDS)*	PL84B	6	LDQ88	C (LVDS)*
AC1	PL72A	6	LDQ75	T	PL85A	6	LDQ88	T
AD1	PL72B	6	LDQ75	C	PL85B	6	LDQ88	C

LFE2-50E/SE and LFE2-70E/SE Logic Signal Connections: 672 fpBGA

LFE2-50E/SE					LFE2-70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
VCCIO	VCCIO6	6			VCCIO6	6		
Y6	PL73A	6	LDQ75	T (LVDS)*	PL86A	6	LDQ88	T (LVDS)*
Y5	PL73B	6	LDQ75	C (LVDS)*	PL86B	6	LDQ88	C (LVDS)*
AE2	PL74A	6	LDQ75	T	PL87A	6	LDQ88	T
AD2	PL74B	6	LDQ75	C	PL87B	6	LDQ88	C
GND	GNDIO6	-			GNDIO6	-		
AB3	PL75A	6	LDQS75	T (LVDS)*	PL88A	6	LDQS88	T (LVDS)*
AB2	PL75B	6	LDQ75	C (LVDS)*	PL88B	6	LDQ88	C (LVDS)*
W7	PL76A	6	LDQ75	T	PL89A	6	LDQ88	T
VCCIO	VCCIO6	6			VCCIO6	6		
W8	PL76B	6	LDQ75	C	PL89B	6	LDQ88	C
Y7	PL77A	6	LDQ75	T (LVDS)*	PL90A	6	LDQ88	T (LVDS)*
Y8	PL77B	6	LDQ75	C (LVDS)*	PL90B	6	LDQ88	C (LVDS)*
AC2	PL78A	6	LDQ75	T	PL91A	6	LDQ88	T
GND	GNDIO6	-			GNDIO6	-		
AD3	PL78B	6	LDQ75	C	PL91B	6	LDQ88	C
AC3	TCK	-			TCK	-		
AA8	TDI	-			TDI	-		
AB4	TMS	-			TMS	-		
AA5	TDO	-			TDO	-		
AB5	VCCJ	-			VCCJ	-		
AE3	PB2A	5	VREF2_5/BDQ6	T	PB2A	5	VREF2_5/BDQ6	T
AF3	PB2B	5	VREF1_5/BDQ6	C	PB2B	5	VREF1_5/BDQ6	C
AC4	PB3A	5	BDQ6	T	PB3A	5	BDQ6	T
AD4	PB3B	5	BDQ6	C	PB3B	5	BDQ6	C
AE4	PB4A	5	BDQ6	T	PB4A	5	BDQ6	T
AF4	PB4B	5	BDQ6	C	PB4B	5	BDQ6	C
VCCIO	VCCIO5	5			VCCIO5	5		
V9	PB5A	5	BDQ6	T	PB5A	5	BDQ6	T
W9	PB5B	5	BDQ6	C	PB5B	5	BDQ6	C
GND	GNDIO5	-			GNDIO5	-		
AA6	PB6A	5	BDQS6	T	PB6A	5	BDQS6	T
AB6	PB6B	5	BDQ6	C	PB6B	5	BDQ6	C
AC5	PB7A	5	BDQ6	T	PB7A	5	BDQ6	T
AD5	PB7B	5	BDQ6	C	PB7B	5	BDQ6	C
AA7	PB8A	5	BDQ6	T	PB8A	5	BDQ6	T
AB7	PB8B	5	BDQ6	C	PB8B	5	BDQ6	C
VCCIO	VCCIO5	5			VCCIO5	5		
AE5	PB9A	5	BDQ6	T	PB9A	5	BDQ6	T
AF5	PB9B	5	BDQ6	C	PB9B	5	BDQ6	C
AC7	PB10A	5	BDQ6	T	PB10A	5	BDQ6	T
AD7	PB10B	5	BDQ6	C	PB10B	5	BDQ6	C
VCCIO	VCCIO5	5			VCCIO5	5		
GND	GNDIO5	-			GNDIO5	-		
W10	PB20A	5	BDQ24	T	PB29A	5	BDQ33	T
Y10	PB20B	5	BDQ24	C	PB29B	5	BDQ33	C
W11	PB21A	5	BDQ24	T	PB30A	5	BDQ33	T

LFE2-50E/SE and LFE2-70E/SE Logic Signal Connections: 672 fpBGA

LFE2-50E/SE					LFE2-70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
AA10	PB21B	5	BDQ24	C	PB30B	5	BDQ33	C
AC8	PB22A	5	BDQ24	T	PB31A	5	BDQ33	T
AD8	PB22B	5	BDQ24	C	PB31B	5	BDQ33	C
VCCIO	VCCIO5	5			VCCIO5	5		
AB8	PB23A	5	BDQ24	T	PB32A	5	BDQ33	T
AB10	PB23B	5	BDQ24	C	PB32B	5	BDQ33	C
GND	GNDIO5	-			GNDIO5	-		
AE6	PB24A	5	BDQS24	T	PB33A	5	BDQS33	T
AF6	PB24B	5	BDQ24	C	PB33B	5	BDQ33	C
AA11	PB25A	5	BDQ24	T	PB34A	5	BDQ33	T
AC9	PB25B	5	BDQ24	C	PB34B	5	BDQ33	C
AB9	PB26A	5	BDQ24	T	PB35A	5	BDQ33	T
AD9	PB26B	5	BDQ24	C	PB35B	5	BDQ33	C
VCCIO	VCCIO5	5			VCCIO5	5		
Y11	PB27A	5	BDQ24	T	PB36A	5	BDQ33	T
AB11	PB27B	5	BDQ24	C	PB36B	5	BDQ33	C
AE7	PB28A	5	BDQ24	T	PB37A	5	BDQ33	T
AF7	PB28B	5	BDQ24	C	PB37B	5	BDQ33	C
GND	GNDIO5	-			GNDIO5	-		
AC10	PB29A	5	BDQ33	T	PB38A	5	BDQ42	T
AD10	PB29B	5	BDQ33	C	PB38B	5	BDQ42	C
AA12	PB30A	5	BDQ33	T	PB39A	5	BDQ42	T
W12	PB30B	5	BDQ33	C	PB39B	5	BDQ42	C
AB12	PB31A	5	BDQ33	T	PB40A	5	BDQ42	T
VCCIO	VCCIO5	5			VCCIO5	5		
Y12	PB31B	5	BDQ33	C	PB40B	5	BDQ42	C
AD12	PB32A	5	BDQ33	T	PB41A	5	BDQ42	T
AC12	PB32B	5	BDQ33	C	PB41B	5	BDQ42	C
AC13	PB33A	5	BDQS33	T	PB42A	5	BDQS42	T
GND	GNDIO5	-			GNDIO5	-		
AA13	PB33B	5	BDQ33	C	PB42B	5	BDQ42	C
AD13	PB34A	5	BDQ33	T	PB43A	5	BDQ42	T
AC14	PB34B	5	BDQ33	C	PB43B	5	BDQ42	C
AE8	PB35A	5	BDQ33	T	PB44A	5	BDQ42	T
VCCIO	VCCIO5	5			VCCIO5	5		
AF8	PB35B	5	BDQ33	C	PB44B	5	BDQ42	C
AB15	PB36A	5	BDQ33	T	PB45A	5	BDQ42	T
Y13	PB36B	5	BDQ33	C	PB45B	5	BDQ42	C
AE9	PB37A	5	BDQ33	T	PB46A	5	BDQ42	T
GND	GNDIO5	-			GNDIO5	-		
AF9	PB37B	5	BDQ33	C	PB46B	5	BDQ42	C
W13	PB38A	5	BDQ42	T	PB47A	5	BDQ51	T
AA14	PB38B	5	BDQ42	C	PB47B	5	BDQ51	C
AE10	PB39A	5	BDQ42	T	PB48A	5	BDQ51	T
AF10	PB39B	5	BDQ42	C	PB48B	5	BDQ51	C
W14	PB40A	5	BDQ42	T	PB49A	5	BDQ51	T
AB13	PB40B	5	BDQ42	C	PB49B	5	BDQ51	C

LFE2-50E/SE and LFE2-70E/SE Logic Signal Connections: 672 fpBGA

LFE2-50E/SE					LFE2-70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
VCCIO	VCCIO5	5			VCCIO5	5		
Y14	PB41A	5	BDQ42	T	PB50A	5	BDQ51	T
AB14	PB41B	5	BDQ42	C	PB50B	5	BDQ51	C
GND	GNDIO5	-			GNDIO5	-		
AE11	PB42A	5	BDQS42	T	PB51A	5	BDQS51	T
AF11	PB42B	5	BDQ42	C	PB51B	5	BDQ51	C
AD14	PB43A	5	BDQ42	T	PB52A	5	BDQ51	T
AA15	PB43B	5	BDQ42	C	PB52B	5	BDQ51	C
AE12	PB44A	5	PCLKT5_0/BDQ42	T	PB53A	5	PCLKT5_0/BDQ51	T
AF12	PB44B	5	PCLKC5_0/BDQ42	C	PB53B	5	PCLKC5_0/BDQ51	C
VCCIO	VCCIO5	5			VCCIO5	5		
GND	GNDIO5	-			GNDIO5	-		
AD15	PB49A	4	PCLKT4_0/BDQ51	T	PB58A	4	PCLKT4_0/BDQ60	T
VCCIO	VCCIO4	4			VCCIO4	4		
AC15	PB49B	4	PCLKC4_0/BDQ51	C	PB58B	4	PCLKC4_0/BDQ60	C
AE13	PB50A	4	BDQ51	T	PB59A	4	BDQ60	T
AF13	PB50B	4	BDQ51	C	PB59B	4	BDQ60	C
AB17	PB51A	4	BDQS51	T	PB60A	4	BDQS60	T
GND	GNDIO4	-			GNDIO4	-		
Y15	PB51B	4	BDQ51	C	PB60B	4	BDQ60	C
AE14	PB52A	4	BDQ51	T	PB61A	4	BDQ60	T
AF14	PB52B	4	BDQ51	C	PB61B	4	BDQ60	C
AA16	PB53A	4	BDQ51	T	PB62A	4	BDQ60	T
VCCIO	VCCIO4	4			VCCIO4	4		
W15	PB53B	4	BDQ51	C	PB62B	4	BDQ60	C
AC17	PB54A	4	BDQ51	T	PB63A	4	BDQ60	T
AB16	PB54B	4	BDQ51	C	PB63B	4	BDQ60	C
AE15	PB55A	4	BDQ51	T	PB64A	4	BDQ60	T
GND	GNDIO4	-			GNDIO4	-		
AF15	PB55B	4	BDQ51	C	PB64B	4	BDQ60	C
AE16	PB56A	4	BDQ60	T	PB65A	4	BDQ69	T
AF16	PB56B	4	BDQ60	C	PB65B	4	BDQ69	C
Y16	PB57A	4	BDQ60	T	PB66A	4	BDQ69	T
AB18	PB57B	4	BDQ60	C	PB66B	4	BDQ69	C
AD17	PB58A	4	BDQ60	T	PB67A	4	BDQ69	T
AD18	PB58B	4	BDQ60	C	PB67B	4	BDQ69	C
VCCIO	VCCIO4	4			VCCIO4	4		
AC18	PB59A	4	BDQ60	T	PB68A	4	BDQ69	T
AD19	PB59B	4	BDQ60	C	PB68B	4	BDQ69	C
GND	GNDIO4	-			GNDIO4	-		
AC19	PB60A	4	BDQS60	T	PB69A	4	BDQS69	T
AE17	PB60B	4	BDQ60	C	PB69B	4	BDQ69	C
AB19	PB61A	4	BDQ60	T	PB70A	4	BDQ69	T
AE19	PB61B	4	BDQ60	C	PB70B	4	BDQ69	C
AF17	PB62A	4	BDQ60	T	PB71A	4	BDQ69	T
AE18	PB62B	4	BDQ60	C	PB71B	4	BDQ69	C
VCCIO	VCCIO4	4			VCCIO4	4		

LFE2-50E/SE and LFE2-70E/SE Logic Signal Connections: 672 fpBGA

LFE2-50E/SE					LFE2-70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
W16	PB63A	4	BDQ60	T	PB72A	4	BDQ69	T
AA17	PB63B	4	BDQ60	C	PB72B	4	BDQ69	C
AF18	PB64A	4	BDQ60	T	PB73A	4	BDQ69	T
AF19	PB64B	4	BDQ60	C	PB73B	4	BDQ69	C
GND	GNDIO4	-			GNDIO4	-		
AA19	PB65A	4	BDQ69	T	PB74A	4	BDQ78	T
W17	PB65B	4	BDQ69	C	PB74B	4	BDQ78	C
Y19	PB66A	4	BDQ69	T	PB75A	4	BDQ78	T
Y17	PB66B	4	BDQ69	C	PB75B	4	BDQ78	C
AF20	PB67A	4	BDQ69	T	PB76A	4	BDQ78	T
VCCIO	VCCIO4	4			VCCIO4	4		
AE20	PB67B	4	BDQ69	C	PB76B	4	BDQ78	C
AA20	PB68A	4	BDQ69	T	PB77A	4	BDQ78	T
W18	PB68B	4	BDQ69	C	PB77B	4	BDQ78	C
AD20	PB69A	4	BDQS69	T	PB78A	4	BDQS78	T
GND	GNDIO4	-			GNDIO4	-		
AE21	PB69B	4	BDQ69	C	PB78B	4	BDQ78	C
AF21	PB70A	4	BDQ69	T	PB79A	4	BDQ78	T
AF22	PB70B	4	BDQ69	C	PB79B	4	BDQ78	C
VCCIO	VCCIO4	4			VCCIO4	4		
GND	GNDIO4	-			GNDIO4	-		
AE22	PB74A	4	BDQ78	T	PB92A	4	BDQ96	T
AD22	PB74B	4	BDQ78	C	PB92B	4	BDQ96	C
AF23	PB75A	4	BDQ78	T	PB93A	4	BDQ96	T
AE23	PB75B	4	BDQ78	C	PB93B	4	BDQ96	C
AD23	PB76A	4	BDQ78	T	PB94A	4	BDQ96	T
AC23	PB76B	4	BDQ78	C	PB94B	4	BDQ96	C
VCCIO	VCCIO4	4			VCCIO4	4		
AB20	PB77A	4	BDQ78	T	PB95A	4	BDQ96	T
AC20	PB77B	4	BDQ78	C	PB95B	4	BDQ96	C
GND	GNDIO4	-			GNDIO4	-		
AB21	PB78A	4	BDQS78	T	PB96A	4	BDQS96	T
AC22	PB78B	4	BDQ78	C	PB96B	4	BDQ96	C
W19	PB79A	4	BDQ78	T	PB97A	4	BDQ96	T
AA21	PB79B	4	BDQ78	C	PB97B	4	BDQ96	C
AF24	PB80A	4	BDQ78	T	PB98A	4	BDQ96	T
AE24	PB80B	4	BDQ78	C	PB98B	4	BDQ96	C
VCCIO	VCCIO4	4			VCCIO4	4		
Y20	PB81A	4	BDQ78	T	PB99A	4	BDQ96	T
AB22	PB81B	4	BDQ78	C	PB99B	4	BDQ96	C
Y21	PB82A	4	VREF2_4/BDQ78	T	PB100A	4	VREF2_4/BDQ96	T
AB23	PB82B	4	VREF1_4/BDQ78	C	PB100B	4	VREF1_4/BDQ96	C
GND	GNDIO4	-			GNDIO4	-		
AD24	CFG2	8			CFG2	8		
W20	CFG1	8			CFG1	8		
AC24	CFG0	8			CFG0	8		
V19	PROGRAMN	8			PROGRAMN	8		

LFE2-50E/SE and LFE2-70E/SE Logic Signal Connections: 672 fpBGA

LFE2-50E/SE					LFE2-70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
AA22	CCLK	8			CCLK	8		
AB24	INITN	8			INITN	8		
AD25	DONE	8			DONE	8		
GND	GNDIO8	-			GNDIO8	-		
W21	PR77B	8	WRITEN	C	PR90B	8	WRITEN	C
Y22	PR77A	8	CS1N	T	PR90A	8	CS1N	T
AC25	PR76B	8	CSN	C	PR89B	8	CSN	C
AB25	PR76A	8	D0/SPIFASTN	T	PR89A	8	D0/SPIFASTN	T
VCCIO	VCCIO8	8			VCCIO8	8		
AD26	PR75B	8	D1	C	PR88B	8	D1	C
AC26	PR75A	8	D2	T	PR88A	8	D2	T
Y23	PR74B	8	D3	C	PR87B	8	D3	C
GND	GNDIO8	-			GNDIO8	-		
W22	PR74A	8	D4	T	PR87A	8	D4	T
AA25	PR73B	8	D5	C	PR86B	8	D5	C
AB26	PR73A	8	D6	T	PR86A	8	D6	T
W23	PR72B	8	D7/SPID0	C	PR85B	8	D7/SPID0	C
VCCIO	VCCIO8	8			VCCIO8	8		
V22	PR72A	8	DI/CSSPI0N	T	PR85A	8	DI/CSSPI0N	T
Y24	PR71B	8	DOUT/CSON	C	PR84B	8	DOUT/CSON	C
Y25	PR71A	8	BUSY/SISPI	T	PR84A	8	BUSY/SISPI	T
W24	PR70B	3	RDQ67	C	PR83B	3	RDQ80	C
GND	GNDIO3	-			GNDIO3	-		
V23	PR70A	3	RDQ67	T	PR83A	3	RDQ80	T
AA26	PR69B	3	RDQ67	C (LVDS)*	PR82B	3	RDQ80	C (LVDS)*
Y26	PR69A	3	RDQ67	T (LVDS)*	PR82A	3	RDQ80	T (LVDS)*
U21	PR68B	3	RDQ67	C	PR81B	3	RDQ80	C
VCCIO	VCCIO3	3			VCCIO3	3		
U19	PR68A	3	RDQ67	T	PR81A	3	RDQ80	T
W25	PR67B	3	RDQ67	C (LVDS)*	PR80B	3	RDQ80	C (LVDS)*
W26	PR67A	3	RDQS67	T (LVDS)*	PR80A	3	RDQS80	T (LVDS)*
GND	GNDIO3	-			GNDIO3	-		
V24	PR66B	3	RDQ67	C	PR79B	3	RDQ80	C
V25	PR66A	3	RDQ67	T	PR79A	3	RDQ80	T
V26	PR65B	3	RDQ67	C (LVDS)*	PR78B	3	RDQ80	C (LVDS)*
U26	PR65A	3	RDQ67	T (LVDS)*	PR78A	3	RDQ80	T (LVDS)*
VCCIO	VCCIO3	3			VCCIO3	3		
U22	PR64B	3	RLM0_GPLLC_FB_A/RDQ67	C	PR77B	3	RLM0_GPLLC_FB_A/RDQ80	C
U23	PR64A	3	RLM0_GPLLT_FB_A/RDQ67	T	PR77A	3	RLM0_GPLLT_FB_A/RDQ80	T
U24	PR63B	3	RLM0_GPLLC_IN_A**/RDQ67	C (LVDS)*	PR76B	3	RLM0_GPLLC_IN_A**/RDQ80	C (LVDS)*
U25	PR63A	3	RLM0_GPLLT_IN_A**/RDQ67	T (LVDS)*	PR76A	3	RLM0_GPLLT_IN_A**/RDQ80	T (LVDS)*
R20	RLM0_PLLCAP	3			RLM0_PLLCAP	3		
P18	VCCPLL	3			VCCPLL	-		
T19	PR61B	3	RLM0_GDLLC_FB_A/RDQ58	C	PR74B	3	RLM0_GDLLC_FB_A/RDQ71	C
U20	PR61A	3	RLM0_GDLLT_FB_A/RDQ58	T	PR74A	3	RLM0_GDLLT_FB_A/RDQ71	T
GND	GNDIO3	-			GNDIO3	-		
T25	PR60B	3	RLM0_GDLLC_IN_A**/RDQ58	C (LVDS)*	PR73B	3	RLM0_GDLLC_IN_A**/RDQ71	C (LVDS)*

LFE2-50E/SE and LFE2-70E/SE Logic Signal Connections: 672 fpBGA

LFE2-50E/SE					LFE2-70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
T26	PR60A	3	RLM0_GDLLT_IN_A**/RDQ58	T (LVDS)*	PR73A	3	RLM0_GDLLT_IN_A**/RDQ71	T (LVDS)*
T20	PR59B	3	RDQ58	C	PR72B	3	RDQ71	C
T22	PR59A	3	RDQ58	T	PR72A	3	RDQ71	T
VCCIO	VCCIO3	3			VCCIO3	3		
R26	PR58B	3	RDQ58	C (LVDS)*	PR71B	3	RDQ71	C (LVDS)*
R25	PR58A	3	RDQS58	T (LVDS)*	PR71A	3	RDQS71	T (LVDS)*
R22	PR57B	3	RDQ58	C	PR70B	3	RDQ71	C
GND	GNDIO3	-			GNDIO3	-		
T21	PR57A	3	RDQ58	T	PR70A	3	RDQ71	T
P26	PR56B	3	RDQ58	C (LVDS)*	PR69B	3	RDQ71	C (LVDS)*
P25	PR56A	3	RDQ58	T (LVDS)*	PR69A	3	RDQ71	T (LVDS)*
R24	PR55B	3	RDQ58	C	PR68B	3	RDQ71	C
VCCIO	VCCIO3	3			VCCIO3	3		
R23	PR55A	3	RDQ58	T	PR68A	3	RDQ71	T
P20	PR54B	3	RDQ58	C (LVDS)*	PR67B	3	RDQ71	C (LVDS)*
R19	PR54A	3	RDQ58	T (LVDS)*	PR67A	3	RDQ71	T (LVDS)*
P21	PR53B	3	RDQ50	C	PR66B	3	RDQ63	C
GND	GNDIO3	-			GNDIO3	-		
P19	PR53A	3	RDQ50	T	PR66A	3	RDQ63	T
P23	PR52B	3	RDQ50	C (LVDS)*	PR65B	3	RDQ63	C (LVDS)*
P22	PR52A	3	RDQ50	T (LVDS)*	PR65A	3	RDQ63	T (LVDS)*
N22	PR51B	3	RDQ50	C	PR64B	3	RDQ63	C
VCCIO	VCCIO3	3			VCCIO3	3		
R21	PR51A	3	RDQ50	T	PR64A	3	RDQ63	T
N26	PR50B	3	RDQ50	C (LVDS)*	PR63B	3	RDQ63	C (LVDS)*
N25	PR50A	3	RDQS50	T (LVDS)*	PR63A	3	RDQS63	T (LVDS)*
GND	GNDIO3	-			GNDIO3	-		
N19	PR49B	3	RDQ50	C	PR62B	3	RDQ63	C
N20	PR49A	3	RDQ50	T	PR62A	3	RDQ63	T
M26	PR48B	3	RDQ50	C (LVDS)*	PR61B	3	RDQ63	C (LVDS)*
M25	PR48A	3	RDQ50	T (LVDS)*	PR61A	3	RDQ63	T (LVDS)*
VCCIO	VCCIO3	3			VCCIO3	3		
N18	PR47B	3	VREF2_3/RDQ50	C	PR60B	3	VREF2_3/RDQ63	C
N21	PR47A	3	VREF1_3/RDQ50	T	PR60A	3	VREF1_3/RDQ63	T
L26	PR46B	3	PCLKC3_0/RDQ50	C (LVDS)*	PR59B	3	PCLKC3_0/RDQ63	C (LVDS)*
L25	PR46A	3	PCLKT3_0/RDQ50	T (LVDS)*	PR59A	3	PCLKT3_0/RDQ63	T (LVDS)*
N24	PR44B	2	PCLKC2_0/RDQ41	C	PR57B	2	PCLKC2_0/RDQ54	C
M23	PR44A	2	PCLKT2_0/RDQ41	T	PR57A	2	PCLKT2_0/RDQ54	T
GND	GNDIO2	-			GNDIO2	-		
L21	PR43B	2	RDQ41	C (LVDS)*	PR56B	2	RDQ54	C (LVDS)*
K22	PR43A	2	RDQ41	T (LVDS)*	PR56A	2	RDQ54	T (LVDS)*
M24	PR42B	2	RDQ41	C	PR55B	2	RDQ54	C
N23	PR42A	2	RDQ41	T	PR55A	2	RDQ54	T
VCCIO	VCCIO2	2			VCCIO2	2		
K26	PR41B	2	RDQ41	C (LVDS)*	PR54B	2	RDQ54	C (LVDS)*
K25	PR41A	2	RDQS41	T (LVDS)*	PR54A	2	RDQS54	T (LVDS)*
M20	PR40B	2	RDQ41	C	PR53B	2	RDQ54	C

LFE2-50E/SE and LFE2-70E/SE Logic Signal Connections: 672 fpBGA

LFE2-50E/SE					LFE2-70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
GND	GNDIO2	-			GNDIO2	-		
M19	PR40A	2	RDQ41	T	PR53A	2	RDQ54	T
L22	PR39B	2	RDQ41	C (LVDS)*	PR52B	2	RDQ54	C (LVDS)*
M22	PR39A	2	RDQ41	T (LVDS)*	PR52A	2	RDQ54	T (LVDS)*
K21	PR38B	2	RDQ41	C	PR51B	2	RDQ54	C
VCCIO	VCCIO2	2			VCCIO2	2		
M21	PR38A	2	RDQ41	T	PR51A	2	RDQ54	T
K24	PR37B	2	RDQ41	C (LVDS)*	PR50B	2	RDQ54	C (LVDS)*
J24	PR37A	2	RDQ41	T (LVDS)*	PR50A	2	RDQ54	T (LVDS)*
GND	GNDIO2	-			GNDIO2	-		
VCCIO	VCCIO2	2			VCCIO2	2		
L20	VCCPLL	2			NC	-		
GND	GNDIO2	-			GNDIO2	-		
J26	PR26B	2	RUM0_SPLLC_FB_A/RDQ24	C	PR39B	2	RUM0_SPLLC_FB_A/RDQ37	C
J25	PR26A	2	RUM0_SPLLT_FB_A/RDQ24	T	PR39A	2	RUM0_SPLLT_FB_A/RDQ37	T
J23	PR25B	2	RUM0_SPLLC_IN_A/RDQ24	C	PR38B	2	RUM0_SPLLC_IN_A/RDQ37	C
K23	PR25A	2	RUM0_SPLLT_IN_A/RDQ24	T	PR38A	2	RUM0_SPLLT_IN_A/RDQ37	T
VCCIO	VCCIO2	2			VCCIO2	2		
H26	PR24B	2	RDQ24	C (LVDS)*	PR37B	2	RDQ37	C (LVDS)*
H25	PR24A	2	RDQS24***	T (LVDS)*	PR37A	2	RDQS37***	T (LVDS)*
H24	PR23B	2	RDQ24	C	PR36B	2	RDQ37	C
GND	GNDIO2	-			GNDIO2	-		
H23	PR23A	2	RDQ24	T	PR36A	2	RDQ37	T
VCCIO	VCCIO2	2			VCCIO2	2		
G26	PR19B	2	RDQ16	C	PR32B	2	RDQ29	C
GND	GNDIO2	-			GNDIO2	-		
G25	PR19A	2	RDQ16	T	PR32A	2	RDQ29	T
F26	PR18B	2	RDQ16	C (LVDS)*	PR31B	2	RDQ29	C (LVDS)*
F25	PR18A	2	RDQ16	T (LVDS)*	PR31A	2	RDQ29	T (LVDS)*
K20	PR17B	2	RDQ16	C	PR30B	2	RDQ29	C
VCCIO	VCCIO2	2			VCCIO2	2		
L19	PR17A	2	RDQ16	T	PR30A	2	RDQ29	T
E26	PR16B	2	RDQ16	C (LVDS)*	PR29B	2	RDQ29	C (LVDS)*
E25	PR16A	2	RDQS16	T (LVDS)*	PR29A	2	RDQS29	T (LVDS)*
GND	GNDIO2	-			GNDIO2	-		
J22	PR15B	2	RDQ16	C	PR28B	2	RDQ29	C
H22	PR15A	2	RDQ16	T	PR28A	2	RDQ29	T
G24	PR14B	2	RDQ16	C (LVDS)*	PR27B	2	RDQ29	C (LVDS)*
G23	PR14A	2	RDQ16	T (LVDS)*	PR27A	2	RDQ29	T (LVDS)*
VCCIO	VCCIO2	2			VCCIO2	2		
K19	PR13B	2	RDQ16	C	PR26B	2	RDQ29	C
J19	PR13A	2	RDQ16	T	PR26A	2	RDQ29	T
D26	PR12B	2	RDQ16	C (LVDS)*	PR25B	2	RDQ29	C (LVDS)*
C26	PR12A	2	RDQ16	T (LVDS)*	PR25A	2	RDQ29	T (LVDS)*
F22	PR11B	2	RDQ8	C	PR24B	2	RDQ21	C
E24	PR11A	2	RDQ8	T	PR24A	2	RDQ21	T
GND	GNDIO2	-			GNDIO2	-		

LFE2-50E/SE and LFE2-70E/SE Logic Signal Connections: 672 fpBGA

LFE2-50E/SE					LFE2-70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
D25	PR10B	2	RDQ8	C (LVDS)*	PR23B	2	RDQ21	C (LVDS)*
C25	PR10A	2	RDQ8	T (LVDS)*	PR23A	2	RDQ21	T (LVDS)*
D24	PR9B	2	RDQ8	C	PR22B	2	RDQ21	C
B25	PR9A	2	RDQ8	T	PR22A	2	RDQ21	T
VCCIO	VCCIO2	2			VCCIO2	2		
H21	PR8B	2	RDQ8	C (LVDS)*	PR21B	2	RDQ21	C (LVDS)*
G22	PR8A	2	RDQS8	T (LVDS)*	PR21A	2	RDQS21	T (LVDS)*
B24	PR7B	2	RDQ8	C	PR20B	2	RDQ21	C
GND	GNDIO2	-			GNDIO2	-		
C24	PR7A	2	RDQ8	T	PR20A	2	RDQ21	T
D23	PR6B	2	RDQ8	C (LVDS)*	PR19B	2	RDQ21	C (LVDS)*
C23	PR6A	2	RDQ8	T (LVDS)*	PR19A	2	RDQ21	T (LVDS)*
G21	PR5B	2	RDQ8	C	PR18B	2	RDQ21	C
VCCIO	VCCIO2	2			VCCIO2	2		
H20	PR5A	2	RDQ8	T	PR18A	2	RDQ21	T
GND	GNDIO2	-			GNDIO2	-		
E22	PR2B	2	VREF2_2	C (LVDS)*	PR2B	2	VREF2_2	C (LVDS)*
F21	PR2A	2	VREF1_2	T (LVDS)*	PR2A	2	VREF1_2	T (LVDS)*
E23	PT82B	1	VREF2_1	C	PT100B	1	VREF2_1	C
GND	GNDIO1	-			GNDIO1	-		
D22	PT82A	1	VREF1_1	T	PT100A	1	VREF1_1	T
G20	PT81B	1		C	PT99B	1		C
J18	PT81A	1		T	PT99A	1		T
F20	PT80B	1		C	PT98B	1		C
VCCIO	VCCIO1	1			VCCIO1	1		
H19	PT80A	1		T	PT98A	1		T
A24	PT79B	1		C	PT97B	1		C
A23	PT79A	1		T	PT97A	1		T
E21	PT78B	1		C	PT96B	1		C
F19	PT78A	1		T	PT96A	1		T
C22	PT77B	1		C	PT95B	1		C
GND	GNDIO1	-			GNDIO1	-		
E20	PT77A	1		T	PT95A	1		T
B22	PT76B	1		C	PT94B	1		C
VCCIO	VCCIO1	1			VCCIO1	1		
B23	PT76A	1		T	PT94A	1		T
C20	PT75B	1		C	PT93B	1		C
D20	PT75A	1		T	PT93A	1		T
A22	PT74B	1		C	PT92B	1		C
A21	PT74A	1		T	PT92A	1		T
GND	GNDIO1	-			GNDIO1	-		
E19	PT71B	1		C	PT85B	1		C
C19	PT71A	1		T	PT85A	1		T
VCCIO	VCCIO1	1			VCCIO1	1		
B21	PT70B	1		C	PT79B	1		C
B20	PT70A	1		T	PT79A	1		T
D19	PT69B	1		C	PT78B	1		C

LFE2-50E/SE and LFE2-70E/SE Logic Signal Connections: 672 fpBGA

LFE2-50E/SE					LFE2-70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
B19	PT69A	1		T	PT78A	1		T
GND	GNDIO1	-			GNDIO1	-		
G17	PT68B	1		C	PT77B	1		C
E18	PT68A	1		T	PT77A	1		T
G19	PT67B	1		C	PT76B	1		C
F17	PT67A	1		T	PT76A	1		T
VCCIO	VCCIO1	1			VCCIO1	1		
A20	PT66B	1		C	PT75B	1		C
A19	PT66A	1		T	PT75A	1		T
E17	PT65B	1		C	PT74B	1		C
D18	PT65A	1		T	PT74A	1		T
B18	PT64B	1		C	PT73B	1		C
GND	GNDIO1	-			GNDIO1	-		
A18	PT64A	1		T	PT73A	1		T
E16	PT63B	1		C	PT72B	1		C
G16	PT63A	1		T	PT72A	1		T
F16	PT62B	1		C	PT71B	1		C
VCCIO	VCCIO1	1			VCCIO1	1		
H18	PT62A	1		T	PT71A	1		T
A17	PT61B	1		C	PT70B	1		C
B17	PT61A	1		T	PT70A	1		T
C18	PT60B	1		C	PT69B	1		C
B16	PT60A	1		T	PT69A	1		T
C17	PT59B	1		C	PT68B	1		C
GND	GNDIO1	-			GNDIO1	-		
D17	PT59A	1		T	PT68A	1		T
E15	PT58B	1		C	PT67B	1		C
VCCIO	VCCIO1	1			VCCIO1	1		
G15	PT58A	1		T	PT67A	1		T
A16	PT57B	1		C	PT66B	1		C
B15	PT57A	1		T	PT66A	1		T
D15	PT56B	1		C	PT65B	1		C
F15	PT56A	1		T	PT65A	1		T
A14	PT55B	1		C	PT64B	1		C
B14	PT55A	1		T	PT64A	1		T
GND	GNDIO1	-			GNDIO1	-		
C15	PT54B	1		C	PT63B	1		C
A15	PT54A	1		T	PT63A	1		T
A13	PT53B	1		C	PT62B	1		C
B13	PT53A	1		T	PT62A	1		T
VCCIO	VCCIO1	1			VCCIO1	1		
H17	PT52B	1		C	PT61B	1		C
H15	PT52A	1		T	PT61A	1		T
D13	PT51B	1		C	PT60B	1		C
C14	PT51A	1		T	PT60A	1		T
GND	GNDIO1	-			GNDIO1	-		
G14	PT50B	1		C	PT59B	1		C

LFE2-50E/SE and LFE2-70E/SE Logic Signal Connections: 672 fpBGA

LFE2-50E/SE					LFE2-70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
E14	PT50A	1		T	PT59A	1		T
A12	PT49B	1		C	PT58B	1		C
B12	PT49A	1		T	PT58A	1		T
VCCIO	VCCIO1	1			VCCIO1	1		
F14	PT48B	1	PCLKC1_0	C	PT57B	1	PCLKC1_0	C
D14	PT48A	1	PCLKT1_0	T	PT57A	1	PCLKT1_0	T
H16	XRES	1			XRES	1		
H14	PT46B	0	PCLKC0_0	C	PT55B	0	PCLKC0_0	C
GND	GNDIO0	-			GNDIO0	-		
H13	PT46A	0	PCLKT0_0	T	PT55A	0	PCLKT0_0	T
A11	PT45B	0		C	PT54B	0		C
B11	PT45A	0		T	PT54A	0		T
C13	PT44B	0		C	PT53B	0		C
VCCIO	VCCIO0	0			VCCIO0	0		
E13	PT44A	0		T	PT53A	0		T
D12	PT43B	0		C	PT52B	0		C
F13	PT43A	0		T	PT52A	0		T
A10	PT42B	0		C	PT51B	0		C
B10	PT42A	0		T	PT51A	0		T
C12	PT41B	0		C	PT50B	0		C
GND	GNDIO0	-			GNDIO0	-		
C10	PT41A	0		T	PT50A	0		T
G13	PT40B	0		C	PT49B	0		C
VCCIO	VCCIO0	0			VCCIO0	0		
H12	PT40A	0		T	PT49A	0		T
A9	PT39B	0		C	PT48B	0		C
B9	PT39A	0		T	PT48A	0		T
E12	PT38B	0		C	PT47B	0		C
G12	PT38A	0		T	PT47A	0		T
A8	PT37B	0		C	PT46B	0		C
B8	PT37A	0		T	PT46A	0		T
GND	GNDIO0	-			GNDIO0	-		
E11	PT36B	0		C	PT45B	0		C
C9	PT36A	0		T	PT45A	0		T
A7	PT35B	0		C	PT44B	0		C
B7	PT35A	0		T	PT44A	0		T
VCCIO	VCCIO0	0			VCCIO0	0		
F12	PT34B	0		C	PT43B	0		C
D10	PT34A	0		T	PT43A	0		T
H11	PT33B	0		C	PT42B	0		C
G11	PT33A	0		T	PT42A	0		T
GND	GNDIO0	-			GNDIO0	-		
A6	PT32B	0		C	PT41B	0		C
B6	PT32A	0		T	PT41A	0		T
D8	PT31B	0		C	PT40B	0		C
C8	PT31A	0		T	PT40A	0		T
VCCIO	VCCIO0	0			VCCIO0	0		

LFE2-50E/SE and LFE2-70E/SE Logic Signal Connections: 672 fpBGA

LFE2-50E/SE					LFE2-70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
F11	PT30B	0		C	PT39B	0		C
E10	PT30A	0		T	PT39A	0		T
E9	PT29B	0		C	PT38B	0		C
D9	PT29A	0		T	PT38A	0		T
G10	PT28B	0		C	PT37B	0		C
GND	GNDIO0	-			GNDIO0	-		
H10	PT28A	0		T	PT37A	0		T
A5	PT27B	0		C	PT36B	0		C
B5	PT27A	0		T	PT36A	0		T
C7	PT26B	0		C	PT35B	0		C
VCCIO	VCCIO0	0			VCCIO0	0		
D7	PT26A	0		T	PT35A	0		T
E8	PT25B	0		C	PT34B	0		C
F10	PT25A	0		T	PT34A	0		T
F8	PT24B	0		C	PT33B	0		C
H9	PT24A	0		T	PT33A	0		T
C5	PT23B	0		C	PT32B	0		C
GND	GNDIO0	-			GNDIO0	-		
D5	PT23A	0		T	PT32A	0		T
B4	PT22B	0			PT31B	0		
VCCIO	VCCIO0	0			VCCIO0	0		
GND	GNDIO0	-			GNDIO0	-		
VCCIO	VCCIO0	0			VCCIO0	0		
GND	GNDIO0	-			GNDIO0	-		
VCCIO	VCCIO0	0			VCCIO0	0		
C4	PT10B	0		C	PT10B	0		C
GND	GNDIO0	-			GNDIO0	-		
C3	PT10A	0		T	PT10A	0		T
A4	PT9B	0		C	PT9B	0		C
A3	PT9A	0		T	PT9A	0		T
B3	PT8B	0		C	PT8B	0		C
VCCIO	VCCIO0	0			VCCIO0	0		
B2	PT8A	0		T	PT8A	0		T
D4	PT7B	0		C	PT7B	0		C
D3	PT7A	0		T	PT7A	0		T
C2	PT6B	0		C	PT6B	0		C
C1	PT6A	0		T	PT6A	0		T
G8	PT5B	0		C	PT5B	0		C
GND	GNDIO0	-			GNDIO0	-		
G7	PT5A	0		T	PT5A	0		T
E7	PT4B	0		C	PT4B	0		C
VCCIO	VCCIO0	0			VCCIO0	0		
F7	PT4A	0		T	PT4A	0		T
E6	PT3B	0		C	PT3B	0		C
E5	PT3A	0		T	PT3A	0		T
G6	PT2B	0	VREF2_0	C	PT2B	0	VREF2_0	C
G5	PT2A	0	VREF1_0	T	PT2A	0	VREF1_0	T

LFE2-50E/SE and LFE2-70E/SE Logic Signal Connections: 672 fpBGA

LFE2-50E/SE					LFE2-70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
L12	VCC	-			VCC	-		
L13	VCC	-			VCC	-		
L14	VCC	-			VCC	-		
L15	VCC	-			VCC	-		
M11	VCC	-			VCC	-		
M12	VCC	-			VCC	-		
M15	VCC	-			VCC	-		
M16	VCC	-			VCC	-		
N11	VCC	-			VCC	-		
N16	VCC	-			VCC	-		
P11	VCC	-			VCC	-		
P16	VCC	-			VCC	-		
R11	VCC	-			VCC	-		
R12	VCC	-			VCC	-		
R15	VCC	-			VCC	-		
R16	VCC	-			VCC	-		
T12	VCC	-			VCC	-		
T13	VCC	-			VCC	-		
T14	VCC	-			VCC	-		
T15	VCC	-			VCC	-		
D11	VCCIO0	0			VCCIO0	0		
D6	VCCIO0	0			VCCIO0	0		
G9	VCCIO0	0			VCCIO0	0		
K12	VCCIO0	0			VCCIO0	0		
J12	VCCIO0	0			VCCIO0	0		
D16	VCCIO1	1			VCCIO1	1		
D21	VCCIO1	1			VCCIO1	1		
G18	VCCIO1	1			VCCIO1	1		
J15	VCCIO1	1			VCCIO1	1		
K15	VCCIO1	1			VCCIO1	1		
F23	VCCIO2	2			VCCIO2	2		
J20	VCCIO2	2			VCCIO2	2		
L23	VCCIO2	2			VCCIO2	2		
M17	VCCIO2	2			VCCIO2	2		
M18	VCCIO2	2			VCCIO2	2		
AA23	VCCIO3	3			VCCIO3	3		
R17	VCCIO3	3			VCCIO3	3		
R18	VCCIO3	3			VCCIO3	3		
T23	VCCIO3	3			VCCIO3	3		
V20	VCCIO3	3			VCCIO3	3		
AC16	VCCIO4	4			VCCIO4	4		
AC21	VCCIO4	4			VCCIO4	4		
U15	VCCIO4	4			VCCIO4	4		
V15	VCCIO4	4			VCCIO4	4		
Y18	VCCIO4	4			VCCIO4	4		
AC11	VCCIO5	5			VCCIO5	5		
AC6	VCCIO5	5			VCCIO5	5		

LFE2-50E/SE and LFE2-70E/SE Logic Signal Connections: 672 fpBGA

LFE2-50E/SE					LFE2-70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
U12	VCCIO5	5			VCCIO5	5		
V12	VCCIO5	5			VCCIO5	5		
Y9	VCCIO5	5			VCCIO5	5		
AA4	VCCIO6	6			VCCIO6	6		
R10	VCCIO6	6			VCCIO6	6		
R9	VCCIO6	6			VCCIO6	6		
T4	VCCIO6	6			VCCIO6	6		
V7	VCCIO6	6			VCCIO6	6		
F4	VCCIO7	7			VCCIO7	7		
J7	VCCIO7	7			VCCIO7	7		
L4	VCCIO7	7			VCCIO7	7		
M10	VCCIO7	7			VCCIO7	7		
M9	VCCIO7	7			VCCIO7	7		
AE25	VCCIO8	8			VCCIO8	8		
V18	VCCIO8	8			VCCIO8	8		
J10	VCCAUX	-			VCCAUX	-		
J11	VCCAUX	-			VCCAUX	-		
J16	VCCAUX	-			VCCAUX	-		
J17	VCCAUX	-			VCCAUX	-		
K18	VCCAUX	-			VCCAUX	-		
K9	VCCAUX	-			VCCAUX	-		
L18	VCCAUX	-			VCCAUX	-		
L9	VCCAUX	-			VCCAUX	-		
T18	VCCAUX	-			VCCAUX	-		
T9	VCCAUX	-			VCCAUX	-		
U18	VCCAUX	-			VCCAUX	-		
U9	VCCAUX	-			VCCAUX	-		
V10	VCCAUX	-			VCCAUX	-		
V11	VCCAUX	-			VCCAUX	-		
V16	VCCAUX	-			VCCAUX	-		
V17	VCCAUX	-			VCCAUX	-		
A2	GND	-			GND	-		
A25	GND	-			GND	-		
AA18	GND	-			GND	-		
AA24	GND	-			GND	-		
AA3	GND	-			GND	-		
AA9	GND	-			GND	-		
AD11	GND	-			GND	-		
AD16	GND	-			GND	-		
AD21	GND	-			GND	-		
AD6	GND	-			GND	-		
AE1	GND	-			GND	-		
AE26	GND	-			GND	-		
AF2	GND	-			GND	-		
AF25	GND	-			GND	-		
B1	GND	-			GND	-		
B26	GND	-			GND	-		

LFE2-50E/SE and LFE2-70E/SE Logic Signal Connections: 672 fpBGA

LFE2-50E/SE					LFE2-70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
C11	GND	-			GND	-		
C16	GND	-			GND	-		
C21	GND	-			GND	-		
C6	GND	-			GND	-		
F18	GND	-			GND	-		
F24	GND	-			GND	-		
F3	GND	-			GND	-		
F9	GND	-			GND	-		
J13	GND	-			GND	-		
J14	GND	-			GND	-		
J21	GND	-			GND	-		
J6	GND	-			GND	-		
K10	GND	-			GND	-		
K11	GND	-			GND	-		
K13	GND	-			GND	-		
K14	GND	-			GND	-		
K16	GND	-			GND	-		
K17	GND	-			GND	-		
L10	GND	-			GND	-		
L11	GND	-			GND	-		
L16	GND	-			GND	-		
L17	GND	-			GND	-		
L24	GND	-			GND	-		
L3	GND	-			GND	-		
M13	GND	-			GND	-		
M14	GND	-			GND	-		
N10	GND	-			GND	-		
N12	GND	-			GND	-		
N13	GND	-			GND	-		
N14	GND	-			GND	-		
N15	GND	-			GND	-		
N17	GND	-			GND	-		
P10	GND	-			GND	-		
P12	GND	-			GND	-		
P13	GND	-			GND	-		
P14	GND	-			GND	-		
P15	GND	-			GND	-		
P17	GND	-			GND	-		
R13	GND	-			GND	-		
R14	GND	-			GND	-		
T10	GND	-			GND	-		
T11	GND	-			GND	-		
T16	GND	-			GND	-		
T17	GND	-			GND	-		
T24	GND	-			GND	-		
T3	GND	-			GND	-		
U10	GND	-			GND	-		

LFE2-50E/SE and LFE2-70E/SE Logic Signal Connections: 672 fpBGA

LFE2-50E/SE					LFE2-70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
U11	GND	-			GND	-		
U13	GND	-			GND	-		
U14	GND	-			GND	-		
U16	GND	-			GND	-		
U17	GND	-			GND	-		
V13	GND	-			GND	-		
V14	GND	-			GND	-		
V21	GND	-			GND	-		
V6	GND	-			GND	-		
M3	NC	-			NC	-		
N6	NC	-			NC	-		
P24	NC	-			NC	-		

* Supports true LVDS. Other differential signals must be emulated with external resistors.
** These dedicated input pins can be used for GPLLs or GDLLs within the respective quadrant.
***Due to packaging bond out option, this DQS does not have all the necessary DQ pins bonded out for a full 8-bit data width.

Note: VCCIO and GND pads are used to determine the average DC current drawn by I/Os between GND/VCCIO connections, or between the last GND/VCCIO in an I/O bank and the end of an I/O bank. The substrate pads listed in the Pin Table do not necessarily have a one to one connection with a package ball or pin.

LFE2-70E/SE Logic Signal Connections: 900 fpBGA

LFE2-70E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
VCCIO	VCCIO7	7		
F4	PL2A	7	VREF2_7	T (LVDS)*
F3	PL2B	7	VREF1_7	C (LVDS)*
H4	PL3A	7		T
G5	PL3B	7		C
GND	GNDIO7	-		
D2	PL4A	7		T (LVDS)*
D1	PL4B	7		C (LVDS)*
E2	PL5A	7		T
VCCIO	VCCIO7	7		
E1	PL5B	7		C
GND	GNDIO7	-		
VCCIO	VCCIO7	7		
F1	PL14A	7	LUM1_SPLLT_IN_A/LDQ12	T (LVDS)*
F2	PL14B	7	LUM1_SPLLC_IN_A/LDQ12	C (LVDS)*
G1	PL15A	7	LUM1_SPLLT_FB_A/LDQ12	T
G2	PL15B	7	LUM1_SPLLC_FB_A/LDQ12	C
GND	GNDIO7	-		
H8	PL18A	7	LDQ21	T
H6	PL18B	7	LDQ21	C
VCCIO	VCCIO7	7		
G4	PL19A	7	LDQ21	T (LVDS)*
G3	PL19B	7	LDQ21	C (LVDS)*
H7	PL20A	7	LDQ21	T
H5	PL20B	7	LDQ21	C
GND	GNDIO7	-		
H2	PL21A	7	LDQS21	T (LVDS)*
H1	PL21B	7	LDQ21	C (LVDS)*
J6	PL22A	7	LDQ21	T
VCCIO	VCCIO7	7		
J8	PL22B	7	LDQ21	C
J2	PL23A	7	LDQ21	T (LVDS)*
J1	PL23B	7	LDQ21	C (LVDS)*
J5	PL24A	7	LDQ21	T
GND	GNDIO7	-		
J7	PL24B	7	LDQ21	C
J4	PL25A	7	LDQ29	T (LVDS)*
J3	PL25B	7	LDQ29	C (LVDS)*
K6	PL26A	7	LDQ29	T
K8	PL26B	7	LDQ29	C
VCCIO	VCCIO7	7		
K2	PL27A	7	LDQ29	T (LVDS)*

LFE2-70E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2-70E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
K1	PL27B	7	LDQ29	C (LVDS)*
K5	PL28A	7	LDQ29	T
K7	PL28B	7	LDQ29	C
GND	GNDIO7	-		
K4	PL29A	7	LDQS29	T (LVDS)*
K3	PL29B	7	LDQ29	C (LVDS)*
L8	PL30A	7	LDQ29	T
VCCIO	VCCIO7	7		
L6	PL30B	7	LDQ29	C
L2	PL31A	7	LDQ29	T (LVDS)*
L1	PL31B	7	LDQ29	C (LVDS)*
L7	PL32A	7	LDQ29	T
GND	GNDIO7	-		
L5	PL32B	7	LDQ29	C
L4	PL33A	7	LDQ37	T (LVDS)*
L3	PL33B	7	LDQ37	C (LVDS)*
M8	PL34A	7	LDQ37	T
M6	PL34B	7	LDQ37	C
VCCIO	VCCIO7	7		
M2	PL35A	7	LDQ37	T (LVDS)*
M1	PL35B	7	LDQ37	C (LVDS)*
M7	PL36A	7	LDQ37	T
M5	PL36B	7	LDQ37	C
GND	GNDIO7	-		
M4	PL37A	7	LDQS37	T (LVDS)*
M3	PL37B	7	LDQ37	C (LVDS)*
N6	PL38A	7	LUM0_SPLLT_IN_A/LDQ37	T
VCCIO	VCCIO7	7		
N8	PL38B	7	LUM0_SPLLC_IN_A/LDQ37	C
N5	PL39A	7	LUM0_SPLLT_FB_A/LDQ37	T
N7	PL39B	7	LUM0_SPLLC_FB_A/LDQ37	C
GND	GNDIO7	-		
VCCIO	VCCIO7	7		
T9	PL50A	7	LDQ54	
R9	PL51A	7	LDQ54	T
P7	PL51B	7	LDQ54	C
VCCIO	VCCIO7	7		
N2	PL52A	7	LDQ54	T (LVDS)*
N1	PL52B	7	LDQ54	C (LVDS)*
P6	PL53A	7	LDQ54	T
P5	PL53B	7	LDQ54	C
GND	GNDIO7	-		
P4	PL54A	7	LDQS54	T (LVDS)*

LFE2-70E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2-70E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
P3	PL54B	7	LDQ54	C (LVDS)*
R6	PL55A	7	LDQ54	T
VCCIO	VCCIO7	7		
R8	PL55B	7	LDQ54	C
P2	PL56A	7	LDQ54	T (LVDS)*
P1	PL56B	7	LDQ54	C (LVDS)*
R5	PL57A	7	PCLKT7_0/LDQ54	T
GND	GNDIO7	-		
R7	PL57B	7	PCLKC7_0/LDQ54	C
R4	PL59A	6	PCLKT6_0/LDQ63	T (LVDS)*
R3	PL59B	6	PCLKC6_0/LDQ63	C (LVDS)*
T5	PL60A	6	VREF2_6/LDQ63	T
T7	PL60B	6	VREF1_6/LDQ63	C
T3	PL61A	6	LDQ63	T (LVDS)*
VCCIO	VCCIO6	6		
T4	PL61B	6	LDQ63	C (LVDS)*
T6	PL62A	6	LDQ63	T
T8	PL62B	6	LDQ63	C
T2	PL63A	6	LDQS63	T (LVDS)*
GND	GNDIO6	-		
T1	PL63B	6	LDQ63	C (LVDS)*
U7	PL64A	6	LDQ63	T
U5	PL64B	6	LDQ63	C
VCCIO	VCCIO6	6		
U4	PL65A	6	LDQ63	T (LVDS)*
U3	PL65B	6	LDQ63	C (LVDS)*
U8	PL66A	6	LDQ63	T
U6	PL66B	6	LDQ63	C
GND	GNDIO6	-		
U2	PL67A	6	LDQ71	T (LVDS)*
U1	PL67B	6	LDQ71	C (LVDS)*
V7	PL68A	6	LDQ71	T
V5	PL68B	6	LDQ71	C
VCCIO	VCCIO6	6		
V2	PL69A	6	LDQ71	T (LVDS)*
V1	PL69B	6	LDQ71	C (LVDS)*
V8	PL70A	6	LDQ71	T
V6	PL70B	6	LDQ71	C
GND	GNDIO6	-		
W1	PL71A	6	LDQS71	T (LVDS)*
W2	PL71B	6	LDQ71	C (LVDS)*
W5	PL72A	6	LDQ71	T
VCCIO	VCCIO6	6		

LFE2-70E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2-70E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
W7	PL72B	6	LDQ71	C
W4	PL73A	6	LLM0_GDLLT_IN_A**/LDQ71	T (LVDS)*
W3	PL73B	6	LLM0_GDLLC_IN_A**/LDQ71	C (LVDS)*
W6	PL74A	6	LLM0_GDLLT_FB_A/LDQ71	T
GND	GNDIO6	-		
W8	PL74B	6	LLM0_GDLLC_FB_D/LDQ71	C
Y8	LLM0_PLLCAP	6		
Y1	PL76A	6	LLM0_GPLLT_IN_A**/LDQ80	T (LVDS)*
Y2	PL76B	6	LLM0_GPLLC_IN_A**/LDQ80	C (LVDS)*
Y5	PL77A	6	LLM0_GPLLT_FB_A/LDQ80	T
Y6	PL77B	6	LLM0_GPLLC_FB_A/LDQ80	C
Y4	PL78A	6	LDQ80	T (LVDS)*
VCCIO	VCCIO6	6		
Y3	PL78B	6	LDQ80	C (LVDS)*
AA6	PL79A	6	LDQ80	T
AA8	PL79B	6	LDQ80	C
AA2	PL80A	6	LDQS80	T (LVDS)*
GND	GNDIO6	-		
AA1	PL80B	6	LDQ80	C (LVDS)*
AA7	PL81A	6	LDQ80	T
AA5	PL81B	6	LDQ80	C
VCCIO	VCCIO6	6		
AA4	PL82A	6	LDQ80	T (LVDS)*
AA3	PL82B	6	LDQ80	C (LVDS)*
AB7	PL83A	6	LDQ80	T
AB5	PL83B	6	LDQ80	C
GND	GNDIO6	-		
AB2	PL84A	6	LDQ88	T (LVDS)*
AB1	PL84B	6	LDQ88	C (LVDS)*
AB8	PL85A	6	LDQ88	T
AB6	PL85B	6	LDQ88	C
VCCIO	VCCIO6	6		
AB4	PL86A	6	LDQ88	T (LVDS)*
AB3	PL86B	6	LDQ88	C (LVDS)*
AC7	PL87A	6	LDQ88	T
AC5	PL87B	6	LDQ88	C
GND	GNDIO6	-		
AC2	PL88A	6	LDQS88	T (LVDS)*
AC1	PL88B	6	LDQ88	C (LVDS)*
AC6	PL89A	6	LDQ88	T
VCCIO	VCCIO6	6		
AD6	PL89B	6	LDQ88	C
AD1	PL90A	6	LDQ88	T (LVDS)*

LFE2-70E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2-70E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
AD2	PL90B	6	LDQ88	C (LVDS)*
AD7	PL91A	6	LDQ88	T
GND	GNDIO6	-		
AB9	PL91B	6	LDQ88	C
AD5	TCK	-		
AE7	TDI	-		
AD4	TMS	-		
AA9	TDO	-		
AD3	VCCJ	-		
AC8	PB2A	5	VREF2_5/BDQ6	T
AE8	PB2B	5	VREF1_5/BDQ6	C
AD8	PB3A	5	BDQ6	T
AF8	PB3B	5	BDQ6	C
AG7	PB4A	5	BDQ6	T
VCCIO	VCCIO5	5		
AH7	PB4B	5	BDQ6	C
AC9	PB5A	5	BDQ6	T
AE9	PB5B	5	BDQ6	C
AD9	PB6A	5	BDQS6	T
GND	GNDIO5	-		
AF9	PB6B	5	BDQ6	C
AB10	PB7A	5	BDQ6	T
AA10	PB7B	5	BDQ6	C
AJ7	PB8A	5	BDQ6	T
VCCIO	VCCIO5	5		
AK7	PB8B	5	BDQ6	C
AC10	PB9A	5	BDQ6	T
AE10	PB9B	5	BDQ6	C
AJ8	PB10A	5	BDQ6	T
GND	GNDIO5	-		
AK8	PB10B	5	BDQ6	C
AF6	PB11A	5	BDQ15	T
AF7	PB11B	5	BDQ15	C
AG5	PB12A	5	BDQ15	T
AH5	PB12B	5	BDQ15	C
AG6	PB13A	5	BDQ15	T
AH6	PB13B	5	BDQ15	C
VCCIO	VCCIO5	5		
AJ4	PB14A	5	BDQ15	T
AK4	PB14B	5	BDQ15	C
GND	GNDIO5	-		
AJ5	PB15A	5	BDQS15	T
AK5	PB15B	5	BDQ15	C

LFE2-70E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2-70E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
AJ6	PB16A	5	BDQ15	T
AK6	PB16B	5	BDQ15	C
VCCIO	VCCIO5	5		
GND	GNDIO5	-		
AD10	PB29A	5	BDQ33	T
AF10	PB29B	5	BDQ33	C
AC11	PB30A	5	BDQ33	T
AD11	PB30B	5	BDQ33	C
AG9	PB31A	5	BDQ33	T
AH9	PB31B	5	BDQ33	C
VCCIO	VCCIO5	99		
AE11	PB32A	5	BDQ33	T
AG10	PB32B	5	BDQ33	C
GND	GNDIO5	-		
AJ9	PB33A	5	BDQS33	T
AK9	PB33B	5	BDQ33	C
AF11	PB34A	5	BDQ33	T
AH10	PB34B	5	BDQ33	C
AC12	PB35A	5	BDQ33	T
AE12	PB35B	5	BDQ33	C
VCCIO	VCCIO5	5		
AD12	PB36A	5	BDQ33	T
AF12	PB36B	5	BDQ33	C
AJ10	PB37A	5	BDQ33	T
AK10	PB37B	5	BDQ33	C
GND	GNDIO5	-		
AG11	PB38A	5	BDQ42	T
AH11	PB38B	5	BDQ42	C
AE13	PB39A	5	BDQ42	T
AC13	PB39B	5	BDQ42	C
AF13	PB40A	5	BDQ42	T
VCCIO	VCCIO5	5		
AD13	PB40B	5	BDQ42	C
AJ11	PB41A	5	BDQ42	T
AK11	PB41B	5	BDQ42	C
AD14	PB42A	5	BDQS42	T
GND	GNDIO5	-		
AC14	PB42B	5	BDQ42	C
AG12	PB43A	5	BDQ42	T
AE14	PB43B	5	BDQ42	C
AJ12	PB44A	5	BDQ42	T
VCCIO	VCCIO5	5		
AK12	PB44B	5	BDQ42	C

LFE2-70E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2-70E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
AH12	PB45A	5	BDQ42	T
AF14	PB45B	5	BDQ42	C
AJ13	PB46A	5	BDQ42	T
GND	GNDIO5	-		
AK13	PB46B	5	BDQ42	C
AB15	PB47A	5	BDQ51	T
AD15	PB47B	5	BDQ51	C
AE15	PB48A	5	BDQ51	T
AF15	PB48B	5	BDQ51	C
AG15	PB49A	5	BDQ51	T
AG14	PB49B	5	BDQ51	C
VCCIO	VCCIO5	5		
AH15	PB50A	5	BDQ51	T
AH14	PB50B	5	BDQ51	C
GND	GNDIO5	-		
AJ14	PB51A	5	BDQS51	T
AK14	PB51B	5	BDQ51	C
AD16	PB52A	5	BDQ51	T
AF16	PB52B	5	BDQ51	C
AJ15	PB53A	5	PCLKT5_0/BDQ51	T
AK15	PB53B	5	PCLKC5_0/BDQ51	C
VCCIO	VCCIO5	5		
GND	GNDIO5	-		
AE16	PB58A	4	PCLKT4_0/BDQ60	T
VCCIO	VCCIO4	4		
AC15	PB58B	4	PCLKC4_0/BDQ60	C
AJ16	PB59A	4	BDQ60	T
AK16	PB59B	4	BDQ60	C
AC16	PB60A	4	BDQS60	T
GND	GNDIO4	-		
AB16	PB60B	4	BDQ60	C
AH17	PB61A	4	BDQ60	T
AG17	PB61B	4	BDQ60	C
AF17	PB62A	4	BDQ60	T
VCCIO	VCCIO4	4		
AD17	PB62B	4	BDQ60	C
AE17	PB63A	4	BDQ60	T
AC17	PB63B	4	BDQ60	C
AJ17	PB64A	4	BDQ60	T
GND	GNDIO4	-		
AK17	PB64B	4	BDQ60	C
AK18	PB65A	4	BDQ69	T
AJ18	PB65B	4	BDQ69	C

LFE2-70E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2-70E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
AD18	PB66A	4	BDQ69	T
AF18	PB66B	4	BDQ69	C
AC18	PB67A	4	BDQ69	T
AE18	PB67B	4	BDQ69	C
VCCIO	VCCIO4	4		
AG19	PB68A	4	BDQ69	T
AH19	PB68B	4	BDQ69	C
GND	GNDIO4	-		
AE19	PB69A	4	BDQS69	T
AF19	PB69B	4	BDQ69	C
AC19	PB70A	4	BDQ69	T
AD19	PB70B	4	BDQ69	C
AJ19	PB71A	4	BDQ69	T
AK19	PB71B	4	BDQ69	C
VCCIO	VCCIO4	4		
AF20	PB72A	4	BDQ69	T
AH20	PB72B	4	BDQ69	C
AE20	PB73A	4	BDQ69	T
AG20	PB73B	4	BDQ69	C
GND	GNDIO4	-		
AD20	PB74A	4	BDQ78	T
AC20	PB74B	4	BDQ78	C
AH21	PB75A	4	BDQ78	T
AF21	PB75B	4	BDQ78	C
AJ20	PB76A	4	BDQ78	T
VCCIO	VCCIO4	4		
AK20	PB76B	4	BDQ78	C
AG21	PB77A	4	BDQ78	T
AE21	PB77B	4	BDQ78	C
AD21	PB78A	4	BDQS78	T
GND	GNDIO4	-		
AC21	PB78B	4	BDQ78	C
AD22	PB79A	4	BDQ78	T
AB21	PB79B	4	BDQ78	C
AJ21	PB80A	4	BDQ78	T
VCCIO	VCCIO4	4		
AK21	PB80B	4	BDQ78	C
GND	GNDIO4	-		
VCCIO	VCCIO4	4		
AJ25	PB87A	4	BDQS87***	T
AK24	PB87B	4	BDQ87	C
AJ24	PB88A	4	BDQ87	T
AK25	PB88B	4	BDQ87	C

LFE2-70E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2-70E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
AH24	PB89A	4	BDQ87	T
AH25	PB89B	4	BDQ87	C
VCCIO	VCCIO4	4		
AJ26	PB90A	4	BDQ87	T
AK26	PB90B	4	BDQ87	C
AF25	PB91A	4	BDQ87	T
AG25	PB91B	4	BDQ87	C
GND	GNDIO4	-		
AK22	PB92A	4	BDQ96	T
AJ22	PB92B	4	BDQ96	C
AE22	PB93A	4	BDQ96	T
AF22	PB93B	4	BDQ96	C
AG22	PB94A	4	BDQ96	T
VCCIO	VCCIO4	4		
AH22	PB94B	4	BDQ96	C
AG24	PB95A	4	BDQ96	T
AG23	PB95B	4	BDQ96	C
AE23	PB96A	4	BDQS96	
GND	GNDIO4	-		
AC22	PB97A	4	BDQ96	
AJ23	PB98A	4	BDQ96	T
VCCIO	VCCIO4	4		
AK23	PB98B	4	BDQ96	C
AD24	PB99A	4	BDQ96	T
AF24	PB99B	4	BDQ96	C
AC23	PB100A	4	VREF2_4/BDQ96	T
GND	GNDIO4	-		
AE24	PB100B	4	VREF1_4/BDQ96	C
AE25	CFG2	8		
AB22	CFG1	8		
AE26	CFG0	8		
AA22	PROGRAMN	8		
AD25	CCLK	8		
AD26	INITN	8		
AC24	DONE	8		
GND	GNDIO4	-		
AC25	PR90B	8	WRITEN	C
AE27	PR90A	8	CS1N	T
AC26	PR89B	8	CSN	C
AE28	PR89A	8	D0/SPIFASTN	T
VCCIO	VCCIO8	8		
AD27	PR88B	8	D1	C
AD28	PR88A	8	D2	T

LFE2-70E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2-70E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
AB24	PR87B	8	D3	C
GND	GNDIO4	-		
AB23	PR87A	8	D4	T
AB25	PR86B	8	D5	C
AB26	PR86A	8	D6	T
AC27	PR85B	8	D7/SPID0	C
VCCIO	VCCIO8	8		
AB27	PR85A	8	DI/CSSPI0N	T
AD29	PR84B	8	DOUT/CSON	C
AD30	PR84A	8	BUSY/SISPI	T
AA25	PR83B	3	RDQ80	C
GND	GNDIO3	-		
AA23	PR83A	3	RDQ80	T
AC29	PR82B	3	RDQ80	C (LVDS)*
AC30	PR82A	3	RDQ80	T (LVDS)*
AA26	PR81B	3	RDQ80	C
VCCIO	VCCIO3	3		
AA24	PR81A	3	RDQ80	T
AB29	PR80B	3	RDQ80	C (LVDS)*
AB30	PR80A	3	RDQS80	T (LVDS)*
GND	GNDIO3	-		
Y23	PR79B	3	RDQ80	C
Y25	PR79A	3	RDQ80	T
AA27	PR78B	3	RDQ80	C (LVDS)*
AA28	PR78A	3	RDQ80	T (LVDS)*
VCCIO	VCCIO3	3		
Y24	PR77B	3	RLM0_GPLLC_FB_A/RDQ80	C
Y26	PR77A	3	RLM0_GPLLT_FB_A/RDQ80	T
AA29	PR76B	3	RLM0_GPLLC_IN_A**/RDQ80	C (LVDS)*
AA30	PR76A	3	RLM0_GPLLT_IN_A**/RDQ80	T (LVDS)*
R22	RLM0_PLLCAP	3		
W23	PR74B	3	RLM0_GDLLC_FB_A/RDQ71	C
W25	PR74A	3	RLM0_GDLLT_FB_A/RDQ71	T
GND	GNDIO3	-		
Y27	PR73B	3	RLM0_GDLLC_IN_A**/RDQ71	C (LVDS)*
Y28	PR73A	3	RLM0_GDLLT_IN_A**/RDQ71	T (LVDS)*
W24	PR72B	3	RDQ71	C
W26	PR72A	3	RDQ71	T
VCCIO	VCCIO3	3		
Y29	PR71B	3	RDQ71	C (LVDS)*
Y30	PR71A	3	RDQS71	T (LVDS)*
V25	PR70B	3	RDQ71	C
GND	GNDIO3	-		

LFE2-70E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2-70E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
V23	PR70A	3	RDQ71	T
W27	PR69B	3	RDQ71	C (LVDS)*
W28	PR69A	3	RDQ71	T (LVDS)*
V26	PR68B	3	RDQ71	C
VCCIO	VCCIO3	3		
V24	PR68A	3	RDQ71	T
W29	PR67B	3	RDQ71	C (LVDS)*
W30	PR67A	3	RDQ71	T (LVDS)*
U25	PR66B	3	RDQ63	C
GND	GNDIO3	-		
U23	PR66A	3	RDQ63	T
V29	PR65B	3	RDQ63	C (LVDS)*
V30	PR65A	3	RDQ63	T (LVDS)*
U26	PR64B	3	RDQ63	C
VCCIO	VCCIO3	3		
U24	PR64A	3	RDQ63	T
U27	PR63B	3	RDQ63	C (LVDS)*
U28	PR63A	3	RDQS63	T (LVDS)*
GND	GNDIO3	-		
T23	PR62B	3	RDQ63	C
T25	PR62A	3	RDQ63	T
U29	PR61B	3	RDQ63	C (LVDS)*
U30	PR61A	3	RDQ63	T (LVDS)*
VCCIO	VCCIO3	3		
T24	PR60B	3	VREF2_3/RDQ63	C
T26	PR60A	3	VREF1_3/RDQ63	T
T27	PR59B	3	PCLKC3_0/RDQ63	C (LVDS)*
T28	PR59A	3	PCLKT3_0/RDQ63	T (LVDS)*
R24	PR57B	2	PCLKC2_0/RDQ54	C
R26	PR57A	2	PCLKT2_0/RDQ54	T
GND	GNDIO2	-		
T29	PR56B	2	RDQ54	C (LVDS)*
T30	PR56A	2	RDQ54	T (LVDS)*
R23	PR55B	2	RDQ54	C
R25	PR55A	2	RDQ54	T
VCCIO	VCCIO2	2		
R27	PR54B	2	RDQ54	C (LVDS)*
R28	PR54A	2	RDQS54	T (LVDS)*
P26	PR53B	2	RDQ54	C
GND	GNDIO2	-		
P24	PR53A	2	RDQ54	T
R29	PR52B	2	RDQ54	C (LVDS)*
R30	PR52A	2	RDQ54	T (LVDS)*

LFE2-70E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2-70E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
P25	PR51B	2	RDQ54	C
VCCIO	VCCIO2	2		
P23	PR51A	2	RDQ54	T
P27	PR50B	2	RDQ54	C (LVDS)*
P28	PR50A	2	RDQ54	T (LVDS)*
GND	GNDIO2	-		
VCCIO	VCCIO2	2		
N24	PR39B	2	RUM0_SPLLC_FB_A/RDQ37	C
N26	PR39A	2	RUM0_SPLLT_FB_A/RDQ37	T
N23	PR38B	2	RUM0_SPLLC_IN_A/RDQ37	C
N25	PR38A	2	RUM0_SPLLT_IN_A/RDQ37	T
VCCIO	VCCIO2	2		
P29	PR37B	2	RDQ37	C (LVDS)*
P30	PR37A	2	RDQS37	T (LVDS)*
M26	PR36B	2	RDQ37	C
GND	GNDIO2	-		
M24	PR36A	2	RDQ37	T
N29	PR35B	2	RDQ37	C (LVDS)*
N30	PR35A	2	RDQ37	T (LVDS)*
M25	PR34B	2	RDQ37	C
VCCIO	VCCIO2	2		
M23	PR34A	2	RDQ37	T
M27	PR33B	2	RDQ37	C (LVDS)*
M28	PR33A	2	RDQ37	T (LVDS)*
L26	PR32B	2	RDQ29	C
GND	GNDIO2	-		
L24	PR32A	2	RDQ29	T
M29	PR31B	2	RDQ29	C (LVDS)*
M30	PR31A	2	RDQ29	T (LVDS)*
L25	PR30B	2	RDQ29	C
VCCIO	VCCIO2	2		
L23	PR30A	2	RDQ29	T
L27	PR29B	2	RDQ29	C (LVDS)*
L28	PR29A	2	RDQS29	T (LVDS)*
GND	GNDIO2	-		
K24	PR28B	2	RDQ29	C
K26	PR28A	2	RDQ29	T
L29	PR27B	2	RDQ29	C (LVDS)*
L30	PR27A	2	RDQ29	T (LVDS)*
VCCIO	VCCIO2	2		
K23	PR26B	2	RDQ29	C
K25	PR26A	2	RDQ29	T
K27	PR25B	2	RDQ29	C (LVDS)*

LFE2-70E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2-70E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
K28	PR25A	2	RDQ29	T (LVDS)*
J24	PR24B	2	RDQ21	C
J26	PR24A	2	RDQ21	T
GND	GNDIO2	-		
K29	PR23B	2	RDQ21	C (LVDS)*
K30	PR23A	2	RDQ21	T (LVDS)*
J23	PR22B	2	RDQ21	C
J25	PR22A	2	RDQ21	T
VCCIO	VCCIO2	99		
J27	PR21B	2	RDQ21	C (LVDS)*
J28	PR21A	2	RDQS21	T (LVDS)*
H26	PR20B	2	RDQ21	C
GND	GNDIO2	-		
H24	PR20A	2	RDQ21	T
J29	PR19B	2	RDQ21	C (LVDS)*
J30	PR19A	2	RDQ21	T (LVDS)*
H25	PR18B	2	RDQ21	C
VCCIO	VCCIO2	2		
H23	PR18A	2	RDQ21	T
G27	PR15B	2	RUM1_SPLLC_FB_A/RDQ12	C
GND	GNDIO2	-		
H27	PR15A	2	RUM1_SPLLT_FB_A/RDQ12	T
G29	PR14B	2	RUM1_SPLLC_IN_A/RDQ12	C (LVDS)*
G28	PR14A	2	RUM1_SPLLT_IN_A/RDQ12	T (LVDS)*
VCCIO	VCCIO2	2		
GND	GNDIO2	-		
G26	PR6B	2		C (LVDS)*
G25	PR6A	2		T (LVDS)*
G30	PR5B	2		C
F30	PR5A	2		T
VCCIO	VCCIO2	2		
F26	PR4B	2		C (LVDS)*
F27	PR4A	2		T (LVDS)*
F29	PR3B	2		C
GND	GNDIO2	-		
F28	PR3A	2		T
H29	PR2B	2	VREF2_2	C (LVDS)*
H30	PR2A	2	VREF1_2	T (LVDS)*
VCCIO	VCCIO2	2		
B26	PT100B	1	VREF2_1	C
A26	PT100A	1	VREF1_1	T
GND	GNDIO1	-		
C25	PT99B	1		C

LFE2-70E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2-70E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
D25	PT99A	1		T
J22	PT98B	1		C
J21	PT98A	1		T
VCCIO	VCCIO1	1		
B25	PT97B	1		C
A25	PT97A	1		T
E24	PT96B	1		C
F24	PT96A	1		T
GND	GNDIO1	-		
F23	PT95B	1		C
H22	PT95A	1		T
D24	PT94B	1		C
C24	PT94A	1		T
VCCIO	VCCIO1	1		
E23	PT93B	1		C
G23	PT93A	1		T
B24	PT92B	1		C
A24	PT92A	1		T
C27	PT91B	1		C
GND	GNDIO1	-		
D27	PT91A	1		T
C26	PT90B	1		C
D26	PT90A	1		T
A27	PT89B	1		C
VCCIO	VCCIO1	1		
B27	PT89A	1		T
A28	PT88B	1		C
B28	PT88A	1		T
A29	PT87B	1		C
B29	PT87A	1		T
GND	GNDIO1	-		
VCCIO	VCCIO1	1		
H21	PT80B	1		C
F22	PT80A	1		T
VCCIO	VCCIO1	1		
B23	PT79B	1		C
A23	PT79A	1		T
G24	PT78B	1		C
E22	PT78A	1		T
GND	GNDIO1	-		
D22	PT77B	1		C
C22	PT77A	1		T
G22	PT76B	1		C

LFE2-70E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2-70E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
E21	PT76A	1		T
VCCIO	VCCIO1	1		
B22	PT75B	1		C
A22	PT75A	1		T
H20	PT74B	1		C
F21	PT74A	1		T
F20	PT73B	1		C
GND	GNDIO1	-		
H19	PT73A	1		T
D21	PT72B	1		C
C21	PT72A	1		T
E20	PT71B	1		C
VCCIO	VCCIO1	1		
G21	PT71A	1		T
B21	PT70B	1		C
A21	PT70A	1		T
F19	PT69B	1		C
G20	PT69A	1		T
E19	PT68B	1		C
GND	GNDIO1	-		
G19	PT68A	1		T
D20	PT67B	1		C
VCCIO	VCCIO1	1		
C20	PT67A	1		T
B20	PT66B	1		C
A20	PT66A	1		T
F18	PT65B	1		C
H18	PT65A	1		T
D19	PT64B	1		C
C19	PT64A	1		T
GND	GNDIO1	-		
G18	PT63B	1		C
E18	PT63A	1		T
H17	PT62B	1		C
F17	PT62A	1		T
VCCIO	VCCIO1	1		
G17	PT61B	1		C
E17	PT61A	1		T
B19	PT60B	1		C
A19	PT60A	1		T
GND	GNDIO1	-		
D17	PT59B	1		C
B18	PT59A	1		T

LFE2-70E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2-70E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
C17	PT58B	1		C
A18	PT58A	1		T
VCCIO	VCCIO1	1		
H16	PT57B	1	PCLKC1_0	C
F16	PT57A	1	PCLKT1_0	T
K16	XRES	1		
E16	PT55B	0	PCLKC0_0	C
GND	GNDIO0	-		
G16	PT55A	0	PCLKT0_0	T
B17	PT54B	0		C
A17	PT54A	0		T
J15	PT53B	0		C
VCCIO	VCCIO0	0		
J16	PT53A	0		T
C16	PT52B	0		C
D16	PT52A	0		T
F15	PT51B	0		C
H15	PT51A	0		T
E15	PT50B	0		C
GND	GNDIO0	-		
G15	PT50A	0		T
C15	PT49B	0		C
VCCIO	VCCIO0	0		
D15	PT49A	0		T
B16	PT48B	0		C
A16	PT48A	0		T
E14	PT47B	0		C
G14	PT47A	0		T
B15	PT46B	0		C
A15	PT46A	0		T
GND	GNDIO0	-		
H14	PT45B	0		C
F14	PT45A	0		T
D14	PT44B	0		C
C14	PT44A	0		T
VCCIO	VCCIO0	0		
G13	PT43B	0		C
E13	PT43A	0		T
B14	PT42B	0		C
A14	PT42A	0		T
GND	GNDIO0	-		
H13	PT41B	0		C
F13	PT41A	0		T

LFE2-70E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2-70E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
G12	PT40B	0		C
E12	PT40A	0		T
VCCIO	VCCIO0	0		
B13	PT39B	0		C
A13	PT39A	0		T
H12	PT38B	0		C
F12	PT38A	0		T
C12	PT37B	0		C
GND	GNDIO0	-		
D12	PT37A	0		T
B12	PT36B	0		C
A12	PT36A	0		T
E11	PT35B	0		C
VCCIO	VCCIO0	0		
G11	PT35A	0		T
F11	PT34B	0		C
H11	PT34A	0		T
C11	PT33B	0		C
D11	PT33A	0		T
B11	PT32B	0		C
GND	GNDIO0	-		
A11	PT32A	0		T
E10	PT31B	0		C
VCCIO	VCCIO0	0		
G10	PT31A	0		T
F10	PT30B	0		C
H10	PT30A	0		T
D10	PT29B	0		C
C10	PT29A	0		T
GND	GNDIO0	-		
VCCIO	VCCIO0	0		
A7	PT16B	0		C
B7	PT16A	0		T
A6	PT15B	0		C
B6	PT15A	0		T
C7	PT14B	0		C
GND	GNDIO0	-		
D7	PT14A	0		T
D8	PT13B	0		C
VCCIO	VCCIO0	0		
E7	PT13A	0		T
C6	PT12B	0		C
D6	PT12A	0		T

LFE2-70E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2-70E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
C5	PT11B	0		C
D5	PT11A	0		T
E9	PT10B	0		C
G9	PT10A	0		T
GND	GNDIO0	-		
B10	PT9B	0		C
A10	PT9A	0		T
D9	PT8B	0		C
C9	PT8A	0		T
VCCIO	VCCIO0	0		
F9	PT7B	0		C
H9	PT7A	0		T
B9	PT6B	0		C
A9	PT6A	0		T
GND	GNDIO0	-		
E8	PT5B	0		C
G8	PT5A	0		T
A8	PT4B	0		C
B8	PT4A	0		T
VCCIO	VCCIO0	0		
F8	PT3B	0		C
F7	PT3A	0		T
J10	PT2B	0	VREF2_0	C
J9	PT2A	0	VREF1_0	T
AA11	VCC	-		
AA20	VCC	-		
K11	VCC	-		
K21	VCC	-		
K22	VCC	-		
L11	VCC	-		
L12	VCC	-		
L13	VCC	-		
L18	VCC	-		
L19	VCC	-		
L20	VCC	-		
M11	VCC	-		
M20	VCC	-		
N11	VCC	-		
N20	VCC	-		
V11	VCC	-		
V20	VCC	-		
W11	VCC	-		
W20	VCC	-		

LFE2-70E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2-70E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
Y10	VCC	-		
Y11	VCC	-		
Y12	VCC	-		
Y13	VCC	-		
Y18	VCC	-		
Y19	VCC	-		
Y20	VCC	-		
J13	VCCIO0	0		
J14	VCCIO0	0		
K12	VCCIO0	0		
K13	VCCIO0	0		
K14	VCCIO0	0		
K15	VCCIO0	0		
J17	VCCIO1	1		
J18	VCCIO1	1		
J20	VCCIO1	1		
K17	VCCIO1	1		
K18	VCCIO1	1		
K20	VCCIO1	1		
L21	VCCIO2	2		
M21	VCCIO2	2		
M22	VCCIO2	2		
N21	VCCIO2	2		
N22	VCCIO2	2		
R21	VCCIO2	2		
U21	VCCIO3	3		
U22	VCCIO3	3		
V21	VCCIO3	3		
V22	VCCIO3	3		
W21	VCCIO3	3		
Y22	VCCIO3	3		
AA16	VCCIO4	4		
AA17	VCCIO4	4		
AA18	VCCIO4	4		
AA19	VCCIO4	4		
AB17	VCCIO4	4		
AB18	VCCIO4	4		
AA12	VCCIO5	5		
AA13	VCCIO5	5		
AA14	VCCIO5	5		
AB12	VCCIO5	5		
AB13	VCCIO5	5		
AB14	VCCIO5	5		

LFE2-70E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2-70E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
U10	VCCIO6	6		
U9	VCCIO6	6		
V10	VCCIO6	6		
W10	VCCIO6	6		
W9	VCCIO6	6		
Y9	VCCIO6	6		
L10	VCCIO7	7		
L9	VCCIO7	7		
M10	VCCIO7	7		
N10	VCCIO7	7		
P10	VCCIO7	7		
R10	VCCIO7	7		
AA21	VCCIO8	8		
Y21	VCCIO8	8		
AA15	VCCAUX	-		
AB11	VCCAUX	-		
AB19	VCCAUX	-		
AB20	VCCAUX	-		
J11	VCCAUX	-		
J12	VCCAUX	-		
J19	VCCAUX	-		
K19	VCCAUX	-		
L22	VCCAUX	-		
M9	VCCAUX	-		
N9	VCCAUX	-		
P21	VCCAUX	-		
P9	VCCAUX	-		
T10	VCCAUX	-		
T21	VCCAUX	-		
V9	VCCAUX	-		
W22	VCCAUX	-		
A1	GND	-		
A30	GND	-		
AC28	GND	-		
AC3	GND	-		
AH13	GND	-		
AH18	GND	-		
AH23	GND	-		
AH28	GND	-		
AH3	GND	-		
AH8	GND	-		
AK1	GND	-		
AK30	GND	-		

LFE2-70E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2-70E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
C13	GND	-		
C18	GND	-		
C23	GND	-		
C28	GND	-		
C3	GND	-		
C8	GND	-		
H28	GND	-		
H3	GND	-		
L14	GND	-		
L15	GND	-		
L16	GND	-		
L17	GND	-		
M12	GND	-		
M13	GND	-		
M14	GND	-		
M15	GND	-		
M16	GND	-		
M17	GND	-		
M18	GND	-		
M19	GND	-		
N12	GND	-		
N13	GND	-		
N14	GND	-		
N15	GND	-		
N16	GND	-		
N17	GND	-		
N18	GND	-		
N19	GND	-		
N28	GND	-		
N3	GND	-		
P11	GND	-		
P12	GND	-		
P13	GND	-		
P14	GND	-		
P15	GND	-		
P16	GND	-		
P17	GND	-		
P18	GND	-		
P19	GND	-		
P20	GND	-		
R11	GND	-		
R12	GND	-		
R13	GND	-		

LFE2-70E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2-70E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
R14	GND	-		
R15	GND	-		
R16	GND	-		
R17	GND	-		
R18	GND	-		
R19	GND	-		
R20	GND	-		
T11	GND	-		
T12	GND	-		
T13	GND	-		
T14	GND	-		
T15	GND	-		
T16	GND	-		
T17	GND	-		
T18	GND	-		
T19	GND	-		
T20	GND	-		
U11	GND	-		
U12	GND	-		
U13	GND	-		
U14	GND	-		
U15	GND	-		
U16	GND	-		
U17	GND	-		
U18	GND	-		
U19	GND	-		
U20	GND	-		
V12	GND	-		
V13	GND	-		
V14	GND	-		
V15	GND	-		
V16	GND	-		
V17	GND	-		
V18	GND	-		
V19	GND	-		
V28	GND	-		
V3	GND	-		
W12	GND	-		
W13	GND	-		
W14	GND	-		
W15	GND	-		
W16	GND	-		
W17	GND	-		

LFE2-70E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2-70E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
W18	GND	-		
W19	GND	-		
Y14	GND	-		
Y15	GND	-		
Y16	GND	-		
Y17	GND	-		
A2	NC	-		
A3	NC	-		
A4	NC	-		
A5	NC	-		
AB28	NC	-		
AC4	NC	-		
AD23	NC	-		
AE1	NC	-		
AE2	NC	-		
AE29	NC	-		
AE3	NC	-		
AE30	NC	-		
AE4	NC	-		
AE5	NC	-		
AE6	NC	-		
AF1	NC	-		
AF2	NC	-		
AF23	NC	-		
AF26	NC	-		
AF27	NC	-		
AF28	NC	-		
AF29	NC	-		
AF3	NC	-		
AF30	NC	-		
AF4	NC	-		
AF5	NC	-		
AG1	NC	-		
AG13	NC	-		
AG16	NC	-		
AG18	NC	-		
AG2	NC	-		
AG26	NC	-		
AG27	NC	-		
AG28	NC	-		
AG29	NC	-		
AG3	NC	-		
AG30	NC	-		

LFE2-70E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2-70E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
AG4	NC	-		
AG8	NC	-		
AH1	NC	-		
AH16	NC	-		
AH2	NC	-		
AH26	NC	-		
AH27	NC	-		
AH29	NC	-		
AH30	NC	-		
AH4	NC	-		
AJ1	NC	-		
AJ2	NC	-		
AJ27	NC	-		
AJ28	NC	-		
AJ29	NC	-		
AJ3	NC	-		
AJ30	NC	-		
AK2	NC	-		
AK27	NC	-		
AK28	NC	-		
AK29	NC	-		
AK3	NC	-		
B1	NC	-		
B2	NC	-		
B3	NC	-		
B30	NC	-		
B4	NC	-		
B5	NC	-		
C1	NC	-		
C2	NC	-		
C29	NC	-		
C30	NC	-		
C4	NC	-		
D13	NC	-		
D18	NC	-		
D23	NC	-		
D28	NC	-		
D29	NC	-		
D3	NC	-		
D30	NC	-		
D4	NC	-		
E25	NC	-		
E26	NC	-		

LFE2-70E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2-70E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
E27	NC	-		
E28	NC	-		
E29	NC	-		
E3	NC	-		
E30	NC	-		
E4	NC	-		
E5	NC	-		
E6	NC	-		
F25	NC	-		
F5	NC	-		
F6	NC	-		
G6	NC	-		
G7	NC	-		
K10	NC	-		
K9	NC	-		
N27	NC	-		
N4	NC	-		
R1	NC	-		
R2	NC	-		
V27	NC	-		
V4	NC	-		
P22	VCCPLL	-		
P8	VCCPLL	-		
T22	VCCPLL	-		
Y7	VCCPLL	-		

* Supports true LVDS. Other differential signals must be emulated with external resistors.
** These dedicated input pins can be used for GPLLs or GDLLs within the respective quadrant.
***Due to packaging bond out option, this DQS does not have all the necessary DQ pins bonded out for a full 8-bit data width.

Note: VCCIO and GND pads are used to determine the average DC current drawn by I/Os between GND/VCCIO connections, or between the last GND/VCCIO in an I/O bank and the end of an I/O bank. The substrate pads listed in the Pin Table do not necessarily have a one to one connection with a package ball or pin.

LFE2M-20E/SE and LFE2M-35E/SE Logic Signal Connections: 256 fpBGA

LFE2M20E/SE					LFE2M35E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
A2	PL2A	7	LDQ6	T (LVDS)*	PL2A	7	LDQ6	T (LVDS)*
B2	PL2B	7	LDQ6	C (LVDS)*	PL2B	7	LDQ6	C(LVDS)*
D3	PL3A	7	LDQ6	T	PL3A	7	LDQ6	T
C2	PL3B	7	LDQ6	C	PL3B	7	LDQ6	C
E4	PL4A	7	LDQ6	T (LVDS)*	PL4A	7	LDQ6	T (LVDS)*
VCCIO	VCCIO7	7			VCCIO7	7		
E5	PL4B	7	LDQ6	C (LVDS)*	PL4B	7	LDQ6	C(LVDS)*
B1	PL5A	7	LDQ6	T	PL5A	7	LDQ6	T
C1	PL5B	7	LDQ6	C	PL5B	7	LDQ6	C
D2	PL6A	7	LDQS6	T (LVDS)*	PL6A	7	LDQS6	T (LVDS)*
GNDIO	GNDIO7	-			GNDIO7	-		
D1	PL6B	7	LDQ6	C (LVDS)*	PL6B	7	LDQ6	C(LVDS)*
E1	PL7A	7	LDQ6	T	PL7A	7	LDQ6	T
F1	PL7B	7	LDQ6	C	PL7B	7	LDQ6	C
VCCIO	VCCIO7	7			VCCIO7	7		
F3	PL8A	7	LDQ6	T (LVDS)*	PL8A	7	LDQ6	T (LVDS)*
F2	PL8B	7	LDQ6	C (LVDS)*	PL8B	7	LDQ6	C(LVDS)*
F6	PL9A	7	VREF2_7/LDQ6	T	PL9A	7	VREF2_7/LDQ6	T
F5	PL9B	7	VREF1_7/LDQ6	C	PL9B	7	VREF1_7/LDQ6	C
GNDIO	GNDIO7	-			GNDIO7	-		
G4	PL11A	7	LUM0_SPLLT_IN_A	T (LVDS)*	PL11A	7	LUM0_SPLLT_IN_A/LDQ15	T (LVDS)*
G3	PL11B	7	LUM0_SPLLC_IN_A	C (LVDS)*	PL11B	7	LUM0_SPLLC_IN_A/LDQ15	C(LVDS)*
G1	PL12A	7	LUM0_SPLLT_FB_A	T	PL12A	7	LUM0_SPLLT_FB_A/LDQ15	T
G2	PL12B	7	LUM0_SPLLC_FB_A	C	PL12B	7	LUM0_SPLLC_FB_A/LDQ15	C
H1	PL13A	7		T (LVDS)*	PL13A	7	LDQ15	T (LVDS)*
VCCIO	VCCIO7	7			VCCIO7	7		
J1	PL13B	7		C (LVDS)*	PL13B	7	LDQ15	C(LVDS)*
H2	PL14A	7		T	PL14A	7	LDQ15	T
H3	PL14B	7		C	PL14B	7	LDQ15	C
GNDIO	GNDIO7	-			GNDIO7	-		
VCCIO	VCCIO7	7			VCCIO7	7		
G6	PL24A	7	LDQ22	T (LVDS)*	PL34A	7	LDQ32	T (LVDS)*
H6	PL24B	7	LDQ22	C (LVDS)*	PL34B	7	LDQ32	C(LVDS)*
J2	PL25A	7	PCLKT7_0/LDQ22	T	PL35A	7	PCLKT7_0/LDQ32	T
GNDIO	GNDIO7	-			GNDIO7	-		
K1	PL25B	7	PCLKC7_0/LDQ22	C	PL35B	7	PCLKC7_0/LDQ32	C
H4	PL27A	6	PCLKT6_0	T (LVDS)*	PL37A	6	PCLKT6_0	T (LVDS)*
H5	PL27B	6	PCLKC6_0	C (LVDS)*	PL37B	6	PCLKC6_0	C(LVDS)*
J4	PL28A	6	VREF2_6	T	PL38A	6	VREF2_6	T
K4	PL28B	6	VREF1_6	C	PL38B	6	VREF1_6	C
VCCIO	VCCIO6	6			VCCIO6	6		
J6	PL31A	6	LLM1_SPLLT_IN_A	T (LVDS)*	PL41A	6	LLM2_SPLLT_IN_A	T (LVDS)*
GNDIO	GNDIO6	-			GNDIO6	-		
J5	PL31B	6	LLM1_SPLLC_IN_A	C (LVDS)*	PL41B	6	LLM2_SPLLC_IN_A	C(LVDS)*
K3	PL32A	6	LLM1_SPLLT_FB_A	T	PL42A	6	LLM2_SPLLT_FB_A	T
K2	PL32B	6	LLM1_SPLLC_FB_A	C	PL42B	6	LLM2_SPLLC_FB_A	C
VCCIO	VCCIO6	6			VCCIO6	6		
GNDIO	GNDIO6	-			GNDIO6	-		
L1	PL42A	6	LLM0_GPLLT_IN_A	T (LVDS)*	PL57A	6	LLM0_GPLLT_IN_A**/LDQS57***	T (LVDS)*

LFE2M-20E/SE and LFE2M-35E/SE Logic Signal Connections: 256 fpBGA

LFE2M20E/SE					LFE2M35E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
GNDIO	GNDIO6	-			GNDIO6	-		
L2	PL42B	6	LLM0_GPLLC_IN_A	C (LVDS)*	PL57B	6	LLM0_GPLLC_IN_A**/LDQ57	C(LVDS)*
L3	PL43A	6	LLM0_GPLLT_FB_A	T	PL58A	6	LLM0_GPLLT_FB_A/LDQ57	T
L4	PL43B	6	LLM0_GPLLC_FB_A	C	PL58B	6	LLM0_GPLLC_FB_A/LDQ57	C
VCCIO	VCCIO6	6			VCCIO6	6		
M1	PL44A	6	LLM0_GDLLT_IN_A	T (LVDS)*	PL59A	6	LLM0_GDLLT_IN_A**/LDQ57	T (LVDS)*
N1	PL44B	6	LLM0_GDLLC_IN_A	C (LVDS)*	PL59B	6	LLM0_GDLLC_IN_A**/LDQ57	C(LVDS)*
N2	PL45A	6	LLM0_GDLLT_FB_A	T	PL60A	6	LLM0_GDLLT_FB_A/LDQ57	T
N3	PL45B	6	LLM0_GDLLC_FB_A	C	PL60B	6	LLM0_GDLLC_FB_A/LDQ57	C
GNDIO	GNDIO6	-			GNDIO6	-		
M4	LLM0_PLLCAP	6			LLM0_PLLCAP	6		
VCCIO	VCCIO6	6			VCCIO6	6		
GNDIO	GNDIO6	-			GNDIO6	-		
K6	TCK	-			TCK	-		
L5	TDI	-			TDI	-		
N4	TMS	-			TMS	-		
N6	TDO	-			TDO	-		
K7	VCCJ	-			VCCJ	-		
M5	PB2A	5	BDQ6	T	PB2A	5	BDQ6	T
N5	PB2B	5	BDQ6	C	PB2B	5	BDQ6	C
L6	PB3A	5	BDQ6	T	PB3A	5	BDQ6	T
M6	PB3B	5	BDQ6	C	PB3B	5	BDQ6	C
P3	PB4A	5	BDQ6	T	PB4A	5	BDQ6	T
VCCIO	VCCIO5	5			VCCIO5	5		
P4	PB4B	5	BDQ6	C	PB4B	5	BDQ6	C
P2	PB5A	5	BDQ6	T	PB5A	5	BDQ6	T
P1	PB5B	5	BDQ6	C	PB5B	5	BDQ6	C
R1	PB6A	5	BDQS6	T	PB6A	5	BDQS6	T
GNDIO	GNDIO5	-			GNDIO5	-		
R2	PB6B	5	BDQ6	C	PB6B	5	BDQ6	C
R3	PB7A	5	BDQ6	T	PB7A	5	BDQ6	T
T2	PB7B	5	BDQ6	C	PB7B	5	BDQ6	C
R4	PB8A	5	BDQ6	T	PB8A	5	BDQ6	T
VCCIO	VCCIO5	5			VCCIO5	5		
T3	PB8B	5	BDQ6	C	PB8B	5	BDQ6	C
T4	PB10A	5	BDQ6	T	PB10A	5	BDQ6	T
GNDIO	GNDIO5	-			GNDIO5	-		
T5	PB10B	5	BDQ6	C	PB10B	5	BDQ6	C
VCCIO	VCCIO5	5			VCCIO5	5		
GNDIO	GNDIO5	-			GNDIO5	-		
T6	PB16A	5	VREF2_5/BDQ15	T	PB34A	5	VREF2_5/BDQ33	T
R6	PB16B	5	VREF1_5/BDQ15	C	PB34B	5	VREF1_5/BDQ33	C
P6	PB17A	5	PCLKT5_0/BDQ15	T	PB35A	5	PCLKT5_0/BDQ33	T
P7	PB17B	5	PCLKC5_0/BDQ15	C	PB35B	5	PCLKC5_0/BDQ33	C
VCCIO	VCCIO5	5			VCCIO5	5		
GNDIO	GNDIO5	-			GNDIO5	-		
T7	PB22A	4	PCLKT4_0/BDQ24	T	PB40A	4	PCLKT4_0/BDQ42	T
VCCIO	VCCIO4	4			VCCIO4	4		
T8	PB22B	4	PCLKC4_0/BDQ24	C	PB40B	4	PCLKC4_0/BDQ42	C

LFE2M-20E/SE and LFE2M-35E/SE Logic Signal Connections: 256 fpBGA

LFE2M20E/SE					LFE2M35E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
L7	PB23A	4	VREF2_4/BDQ24	T	PB41A	4	VREF2_4/BDQ42	T
L8	PB23B	4	VREF1_4/BDQ24	C	PB41B	4	VREF1_4/BDQ42	C
GNDIO	GNDIO4	-			GNDIO4	-		
VCCIO	VCCIO4	4			VCCIO4	4		
GNDIO	GNDIO4	-			GNDIO4	-		
P8	PB29A	4	BDQ33	T	PB47A	4	BDQ51	T
N8	PB29B	4	BDQ33	C	PB47B	4	BDQ51	C
R7	PB30A	4	BDQ33	T	PB48A	4	BDQ51	T
R8	PB30B	4	BDQ33	C	PB48B	4	BDQ51	C
N7	PB31A	4	BDQ33	T	PB49A	4	BDQ51	T
M8	PB31B	4	BDQ33	C	PB49B	4	BDQ51	C
VCCIO	VCCIO4	4			VCCIO4	4		
R9	PB32A	4	BDQ33	T	PB50A	4	BDQ51	T
T9	PB32B	4	BDQ33	C	PB50B	4	BDQ51	C
GNDIO	GNDIO4	-			GNDIO4	-		
T10	PB33A	4	BDQS33	T	PB51A	4	BDQS51	T
R10	PB33B	4	BDQ33	C	PB51B	4	BDQ51	C
N9	PB34A	4	BDQ33	T	PB52A	4	BDQ51	T
P10	PB34B	4	BDQ33	C	PB52B	4	BDQ51	C
VCCIO	VCCIO4	4			VCCIO4	4		
GNDIO	GNDIO4	-			GNDIO4	-		
L9	PB47A	4	BDQ51	T	PB65A	4	BDQ69	T
M9	PB47B	4	BDQ51	C	PB65B	4	BDQ69	C
T11	PB49A	4	BDQ51	T	PB67A	4	BDQ69	T
R11	PB49B	4	BDQ51	C	PB67B	4	BDQ69	C
VCCIO	VCCIO4	4			VCCIO4	4		
T12	PB50A	4	BDQ51	T	PB68A	4	BDQ69	T
T13	PB50B	4	BDQ51	C	PB68B	4	BDQ69	C
GNDIO	GNDIO4	-			GNDIO4	-		
P11	PB51A	4	BDQS51	T	PB69A	4	BDQS69	T
N10	PB51B	4	BDQ51	C	PB69B	4	BDQ69	C
T14	PB52A	4	BDQ51	T	PB70A	4	BDQ69	T
R13	PB52B	4	BDQ51	C	PB70B	4	BDQ69	C
R15	PB53A	4	BDQ51	T	PB71A	4	BDQ69	T
R16	PB53B	4	BDQ51	C	PB71B	4	BDQ69	C
VCCIO	VCCIO4	4			VCCIO4	4		
R14	PB54A	4	BDQ51	T	PB72A	4	BDQ69	T
P15	PB54B	4	BDQ51	C	PB72B	4	BDQ69	C
P16	PB55A	4	BDQ51	T	PB73A	4	BDQ69	T
P14	PB55B	4	BDQ51	C	PB73B	4	BDQ69	C
GNDIO	GNDIO4	-			GNDIO4	-		
L11	CFG2	8			CFG2	8		
L10	CFG1	8			CFG1	8		
P13	CFG0	8			CFG0	8		
N12	PROGRAMN	8			PROGRAMN	8		
N11	CCLK	8			CCLK	8		
M11	INITN	8			INITN	8		
N13	DONE	8			DONE	8		
GNDIO	GNDIO8	-			GNDIO8	-		

LFE2M-20E/SE and LFE2M-35E/SE Logic Signal Connections: 256 fpBGA

LFE2M20E/SE					LFE2M35E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
M12	PR53B	8	WRITEN	C	PR68B	8	WRITEN	C
M13	PR53A	8	CS1N	T	PR68A	8	CS1N	T
N14	PR52B	8	CSN	C	PR67B	8	CSN	C
N15	PR52A	8	D0/SPIFASTN	T	PR67A	8	D0/SPIFASTN	T
VCCIO	VCCIO8	8			VCCIO8	8		
N16	PR51B	8	D1	C	PR66B	8	D1	C
M16	PR51A	8	D2	T	PR66A	8	D2	T
L12	PR50B	8	D3	C	PR65B	8	D3	C
GNDIO	GNDIO8	-			GNDIO8	-		
L13	PR50A	8	D4	T	PR65A	8	D4	T
L16	PR49B	8	D5	C	PR64B	8	D5	C
K16	PR49A	8	D6	T	PR64A	8	D6	T
L14	PR48B	8	D7/SPID0***	C	PR63B	8	D7/SPID0***	C
VCCIO	VCCIO8	8			VCCIO8	8		
L15	PR48A	8	DI/CSSPI0N	T	PR63A	8	DI/CSSPI0N	T
K13	PR47B	8	DOUT/CSON/CSSPI1N	C	PR62B	8	DOUT/CSON/CSSPI1N	C
K14	PR47A	8	BUSY/SISPI	T	PR62A	8	BUSY/SISPI	T
K11	RLM0_PLLCAP	3			RLM0_PLLCAP	3		
K15	PR45B	3	RLM0_GDLLC_FB_A	C	PR60B	3	RLM0_GDLLC_FB_A/RDQ57	C
GNDIO	GNDIO3	-			GNDIO3	-		
J16	PR45A	3	RLM0_GDLLT_FB_A	T	PR60A	3	RLM0_GDLLT_FB_A/RDQ57	T
H16	PR44B	3	RLM0_GDLLC_IN_A	C (LVDS)*	PR59B	3	RLM0_GDLLC_IN_A**/RDQ57	C(LVDS)*
J15	PR44A	3	RLM0_GDLLT_IN_A	T (LVDS)*	PR59A	3	RLM0_GDLLT_IN_A**/RDQ57	T (LVDS)*
J14	PR43B	3	RLM0_GPLLC_IN_A	C	PR58B	3	RLM0_GPLLC_IN_A**/RDQ57	C
VCCIO	VCCIO3	3			VCCIO3	3		
J13	PR43A	3	RLM0_GPLLT_IN_A	T	PR58A	3	RLM0_GPLLT_IN_A**/RDQ57	T
H13	PR42B	3	RLM0_GPLLC_FB_A	C (LVDS)*	PR57B	3	RLM0_GPLLC_FB_A/RDQ57	C(LVDS)*
H12	PR42A	3	RLM0_GPLLT_FB_A	T (LVDS)*	PR57A	3	RLM0_GPLLT_FB_A/RDQS57***	T (LVDS)*
GNDIO	GNDIO3	-			GNDIO3	-		
VCCIO	VCCIO3	3			VCCIO3	3		
G16	PR32B	3	RLM1_SPLLC_FB_A	C	PR42B	3	RLM2_SPLLC_FB_A	C
VCCIO	VCCIO3	3			VCCIO3	3		
H15	PR32A	3	RLM1_SPLLT_FB_A	T	PR42A	3	RLM2_SPLLT_FB_A	T
E16	PR31B	3	RLM1_SPLLC_IN_A	C (LVDS)*	PR41B	3	RLM2_SPLLC_IN_A	C(LVDS)*
F15	PR31A	3	RLM1_SPLLT_IN_A	T (LVDS)*	PR41A	3	RLM2_SPLLT_IN_A	T (LVDS)*
GNDIO	GNDIO3	-			GNDIO3	-		
VCCIO	VCCIO3	3			VCCIO3	3		
F16	PR28B	3	VREF2_3	C	PR38B	3	VREF2_3	C
G15	PR28A	3	VREF1_3	T	PR38A	3	VREF1_3	T
J11	PR27B	3	PCLKC3_0	C (LVDS)*	PR37B	3	PCLKC3_0	C(LVDS)*
J12	PR27A	3	PCLKT3_0	T (LVDS)*	PR37A	3	PCLKT3_0	T (LVDS)*
G14	PR25B	2	PCLKC2_0/RDQ22	C	PR35B	2	PCLKC2_0/RDQ32	C
G13	PR25A	2	PCLKT2_0/RDQ22	T	PR35A	2	PCLKT2_0/RDQ32	T
GNDIO	GNDIO2	-			GNDIO2	-		
F14	PR24B	2	RDQ22	C (LVDS)*	PR34B	2	RDQ32	C(LVDS)*
F13	PR24A	2	RDQ22	T (LVDS)*	PR34A	2	RDQ32	T (LVDS)*
VCCIO	VCCIO2	2			VCCIO2	2		
GNDIO	GNDIO2	-			GNDIO2	-		
H11	PR14B	2		C	PR14B	2	RDQ15	C

LFE2M-20E/SE and LFE2M-35E/SE Logic Signal Connections: 256 fpBGA

LFE2M20E/SE					LFE2M35E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
G11	PR14A	2		T	PR14A	2	RDQ15	T
E13	PR13B	2		C (LVDS)*	PR13B	2	RDQ15	C(LVDS)*
F12	PR13A	2		T (LVDS)*	PR13A	2	RDQ15	T (LVDS)*
VCCIO	VCCIO2	2			VCCIO2	2		
F11	PR12B	2	RUM0_SPLLC_FB_A	C	PR12B	2	RUM0_SPLLC_FB_A/RDQ15	C
E12	PR12A	2	RUM0_SPLLT_FB_A	T	PR12A	2	RUM0_SPLLT_FB_A/RDQ15	T
D16	PR11B	2	RUM0_SPLLC_IN_A	C (LVDS)*	PR11B	2	RUM0_SPLLC_IN_A/RDQ15	C(LVDS)*
D15	PR11A	2	RUM0_SPLLT_IN_A	T (LVDS)*	PR11A	2	RUM0_SPLLT_IN_A/RDQ15	T (LVDS)*
C16	PR9B	2	VREF2_2	C	PR9B	2	VREF2_2	C
GNDIO	GNDIO2	-			GNDIO2	-		
B16	PR9A	2	VREF1_2	T	PR9A	2	VREF1_2	T
VCCIO	VCCIO2	2			VCCIO2	2		
F4	XRES	-			XRES	-		
C15	URC_SQ_VCCRX0	12			URC_SQ_VCCRX0	12		
A14	URC_SQ_HDINP0	12		T	URC_SQ_HDINP0	12		T
B15	URC_SQ_VCCIB0	12			URC_SQ_VCCIB0	12		
B14	URC_SQ_HDINN0	12		C	URC_SQ_HDINN0	12		C
C12	URC_SQ_VCCTX0	12			URC_SQ_VCCTX0	12		
A11	URC_SQ_HDOUTP0	12		T	URC_SQ_HDOUTP0	12		T
A12	URC_SQ_VCCOB0	12			URC_SQ_VCCOB0	12		
B11	URC_SQ_HDOUTN0	12		C	URC_SQ_HDOUTN0	12		C
C11	URC_SQ_VCCTX1	12			URC_SQ_VCCTX1	12		
B10	URC_SQ_HDOUTN1	12		C	URC_SQ_HDOUTN1	12		C
C10	URC_SQ_VCCOB1	12			URC_SQ_VCCOB1	12		
A10	URC_SQ_HDOUTP1	12		T	URC_SQ_HDOUTP1	12		T
C14	URC_SQ_VCCRX1	12			URC_SQ_VCCRX1	12		
B13	URC_SQ_HDINN1	12		C	URC_SQ_HDINN1	12		C
C13	URC_SQ_VCCIB1	12			URC_SQ_VCCIB1	12		
A13	URC_SQ_HDINP1	12		T	URC_SQ_HDINP1	12		T
B9	URC_SQ_VCCAUX33	12			URC_SQ_VCCAUX33	12		
D8	URC_SQ_REFCLKN	12		C	URC_SQ_REFCLKN	12		C
D9	URC_SQ_REFCLKP	12		T	URC_SQ_REFCLKP	12		T
C9	URC_SQ_VCCP	12			URC_SQ_VCCP	12		
A5	URC_SQ_HDINP2	12		T	URC_SQ_HDINP2	12		T
C5	URC_SQ_VCCIB2	12			URC_SQ_VCCIB2	12		
B5	URC_SQ_HDINN2	12		C	URC_SQ_HDINN2	12		C
C4	URC_SQ_VCCRX2	12			URC_SQ_VCCRX2	12		
A8	URC_SQ_HDOUTP2	12		T	URC_SQ_HDOUTP2	12		T
C8	URC_SQ_VCCOB2	12			URC_SQ_VCCOB2	12		
B8	URC_SQ_HDOUTN2	12		C	URC_SQ_HDOUTN2	12		C
C7	URC_SQ_VCCTX2	12			URC_SQ_VCCTX2	12		
B7	URC_SQ_HDOUTN3	12		C	URC_SQ_HDOUTN3	12		C
A6	URC_SQ_VCCOB3	12			URC_SQ_VCCOB3	12		
A7	URC_SQ_HDOUTP3	12		T	URC_SQ_HDOUTP3	12		T
C6	URC_SQ_VCCTX3	12			URC_SQ_VCCTX3	12		
B4	URC_SQ_HDINN3	12		C	URC_SQ_HDINN3	12		C
B3	URC_SQ_VCCIB3	12			URC_SQ_VCCIB3	12		
A4	URC_SQ_HDINP3	12		T	URC_SQ_HDINP3	12		T
C3	URC_SQ_VCCRX3	12			URC_SQ_VCCRX3	12		

LFE2M-20E/SE and LFE2M-35E/SE Logic Signal Connections: 256 fpBGA

LFE2M20E/SE					LFE2M35E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
GNDIO	GNDIO1	-			GNDIO1	-		
VCCIO	VCCIO1	1			VCCIO1	1		
GNDIO	GNDIO0	-			GNDIO0	-		
VCCIO	VCCIO0	0			VCCIO0	0		
G10	VCCPLL	-			VCCPLL	-		
G7	VCC	-			VCC	-		
G9	VCC	-			VCC	-		
H7	VCC	-			VCC	-		
J10	VCC	-			VCC	-		
K10	VCC	-			VCC	-		
K8	VCC	-			VCC	-		
E7	VCCIO0	0			VCCIO0	0		
VCCIO	VCCIO0	0			VCCIO0	0		
E10	VCCIO1	1			VCCIO1	1		
VCCIO	VCCIO1	1			VCCIO1	1		
E14	VCCIO2	2			VCCIO2	2		
G12	VCCIO2	2			VCCIO2	2		
VCCIO	VCCIO2	2			VCCIO2	2		
K12	VCCIO3	3			VCCIO3	3		
M14	VCCIO3	3			VCCIO3	3		
VCCIO	VCCIO3	3			VCCIO3	3		
M10	VCCIO4	4			VCCIO4	4		
P12	VCCIO4	4			VCCIO4	4		
VCCIO	VCCIO4	4			VCCIO4	4		
M7	VCCIO5	5			VCCIO5	5		
P5	VCCIO5	5			VCCIO5	5		
VCCIO	VCCIO5	5			VCCIO5	5		
K5	VCCIO6	6			VCCIO6	6		
M3	VCCIO6	6			VCCIO6	6		
VCCIO	VCCIO6	6			VCCIO6	6		
E3	VCCIO7	7			VCCIO7	7		
G5	VCCIO7	7			VCCIO7	7		
VCCIO	VCCIO7	7			VCCIO7	7		
T15	VCCIO8	8			VCCIO8	8		
VCCIO	VCCIO8	8			VCCIO8	8		
G8	VCCAUX	-			VCCAUX	-		
H10	VCCAUX	-			VCCAUX	-		
J7	VCCAUX	-			VCCAUX	-		
K9	VCCAUX	-			VCCAUX	-		
A1	GND	-			GND	-		
A15	GND	-			GND	-		
A16	GND	-			GND	-		
A3	GND	-			GND	-		
A9	GND	-			GND	-		
B12	GND	-			GND	-		
B6	GND	-			GND	-		
E15	GND	-			GND	-		
E2	GND	-			GND	-		
H14	GND	-			GND	-		

LFE2M-20E/SE and LFE2M-35E/SE Logic Signal Connections: 256 fpBGA

LFE2M20E/SE					LFE2M35E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
H8	GND	-			GND	-		
H9	GND	-			GND	-		
J3	GND	-			GND	-		
J8	GND	-			GND	-		
J9	GND	-			GND	-		
M15	GND	-			GND	-		
M2	GND	-			GND	-		
P9	GND	-			GND	-		
R12	GND	-			GND	-		
R5	GND	-			GND	-		
T1	GND	-			GND	-		
T16	GND	-			GND	-		
D10	NC	-			NC	-		
D11	NC	-			NC	-		
D12	NC	-			NC	-		
D13	NC	-			NC	-		
D14	NC	-			NC	-		
D4	NC	-			NC	-		
D5	NC	-			NC	-		
D6	NC	-			NC	-		
D7	NC	-			NC	-		
E11	NC	-			NC	-		
E6	NC	-			NC	-		
E8	NC	-			NC	-		
E9	NC	-			NC	-		
F10	NC	-			NC	-		
F7	NC	-			NC	-		
F8	NC	-			NC	-		
F9	NC	-			NC	-		

* Supports true LVDS. Other differential signals must be emulated with external resistors.
** These dedicated input pins can be used for GPLLs or GDLLs within the respective quadrant.
***Due to packaging bond out option, this DQS does not have all the necessary DQ pins bonded out for a full 8-bit data width.

Note: VCCIO and GND pads are used to determine the average DC current drawn by I/Os between GND/VCCIO connections, or between the last GND/VCCIO in an I/O bank and the end of an I/O bank. The substrate pads listed in the Pin Table do not necessarily have a one to one connection with a package ball or pin.

LFE2M20E/SE and LFE2M35E/SE Logic Signal Connections: 484 fpBGA

LFE2M20E/SE					LFE2M35E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
D1	PL2A	7	LDQ6	T (LVDS)*	PL2A	7	LDQ6	T (LVDS)*
E1	PL2B	7	LDQ6	C (LVDS)*	PL2B	7	LDQ6	C (LVDS)*
F1	PL3A	7	LDQ6	T	PL3A	7	LDQ6	T
F2	PL3B	7	LDQ6	C	PL3B	7	LDQ6	C
F5	PL4A	7	LDQ6	T (LVDS)*	PL4A	7	LDQ6	T (LVDS)*
VCCIO	VCCIO7	7			VCCIO7	7		
G6	PL4B	7	LDQ6	C (LVDS)*	PL4B	7	LDQ6	C (LVDS)*
F4	PL5A	7	LDQ6	T	PL5A	7	LDQ6	T
F3	PL5B	7	LDQ6	C	PL5B	7	LDQ6	C
G1	PL6A	7	LDQS6	T (LVDS)*	PL6A	7	LDQS6	T (LVDS)*
GNDIO	GNDIO7	-			GNDIO7	-		
G2	PL6B	7	LDQ6	C (LVDS)*	PL6B	7	LDQ6	C (LVDS)*
H1	PL7A	7	LDQ6	T	PL7A	7	LDQ6	T
H2	PL7B	7	LDQ6	C	PL7B	7	LDQ6	C
VCCIO	VCCIO7	7			VCCIO7	7		
H7	PL8A	7	LDQ6	T (LVDS)*	PL8A	7	LDQ6	T (LVDS)*
H6	PL8B	7	LDQ6	C (LVDS)*	PL8B	7	LDQ6	C (LVDS)*
G3	PL9A	7	VREF2_7/LDQ6	T	PL9A	7	VREF2_7/LDQ6	T
H3	PL9B	7	VREF1_7/LDQ6	C	PL9B	7	VREF1_7/LDQ6	C
GNDIO	GNDIO7	-			GNDIO7	-		
H5	PL11A	7	LUM0_SPLLT_IN_A	T (LVDS)*	PL11A	7	LUM0_SPLLT_IN_A/LDQ15	T (LVDS)*
H4	PL11B	7	LUM0_SPLLC_IN_A	C (LVDS)*	PL11B	7	LUM0_SPLLC_IN_A/LDQ15	C (LVDS)*
J1	PL12A	7	LUM0_SPLLT_FB_A	T	PL12A	7	LUM0_SPLLT_FB_A/LDQ15	T
J2	PL12B	7	LUM0_SPLLC_FB_A	C	PL12B	7	LUM0_SPLLC_FB_A/LDQ15	C
J3	PL13A	7		T (LVDS)*	PL13A	7	LDQ15	T (LVDS)*
VCCIO	VCCIO7	7			VCCIO7	7		
J4	PL13B	7		C (LVDS)*	PL13B	7	LDQ15	C (LVDS)*
J7	PL14A	7		T	PL14A	7	LDQ15	T
J6	PL14B	7		C	PL14B	7	LDQ15	C
GNDIO	GNDIO7	-			GNDIO7	-		
VCCIO	VCCIO7	7			VCCIO7	7		
K1	PL18A	7	LUM1_SPLLT_IN_A/LDQ22	T (LVDS)*	PL28A	7	LUM1_SPLLT_IN_A/LDQ32	T (LVDS)*
K2	PL18B	7	LUM1_SPLLC_IN_A/LDQ22	C (LVDS)*	PL28B	7	LUM1_SPLLC_IN_A/LDQ32	C (LVDS)*
J5	PL19A	7	LUM1_SPLLT_FB_A/LDQ22	T	PL29A	7	LUM1_SPLLT_FB_A/LDQ32	T
K5	PL19B	7	LUM1_SPLLC_FB_A/LDQ22	C	PL29B	7	LUM1_SPLLC_FB_A/LDQ32	C
VCCIO	VCCIO7	7			VCCIO7	7		
K7	PL20A	7	LDQ22	T (LVDS)*	PL30A	7	LDQ32	T (LVDS)*
K6	PL20B	7	LDQ22	C (LVDS)*	PL30B	7	LDQ32	C (LVDS)*
L6	PL21A	7	LDQ22	T	PL31A	7	LDQ32	T
L7	PL21B	7	LDQ22	C	PL31B	7	LDQ32	C
GNDIO	GNDIO7	-			GNDIO7	-		
L1	PL22A	7	LDQS22	T (LVDS)*	PL32A	7	LDQS32	T (LVDS)*
L2	PL22B	7	LDQ22	C (LVDS)*	PL32B	7	LDQ32	C (LVDS)*
M7	PL23A	7	LDQ22	T	PL33A	7	LDQ32	T
VCCIO	VCCIO7	7			VCCIO7	7		
L5	PL23B	7	LDQ22	C	PL33B	7	LDQ32	C
L3	PL24A	7	LDQ22	T (LVDS)*	PL34A	7	LDQ32	T (LVDS)*
L4	PL24B	7	LDQ22	C (LVDS)*	PL34B	7	LDQ32	C (LVDS)*
M1	PL25A	7	PCLKT7_0/LDQ22	T	PL35A	7	PCLKT7_0/LDQ32	T

LFE2M20E/SE and LFE2M35E/SE Logic Signal Connections: 484 fpBGA

LFE2M20E/SE					LFE2M35E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
GNDIO	GNDIO7	-			GNDIO7	-		
M2	PL25B	7	PCLKC7_0/LDQ22	C	PL35B	7	PCLKC7_0/LDQ32	C
M6	PL27A	6	PCLKT6_0	T (LVDS)*	PL37A	6	PCLKT6_0	T (LVDS)*
M5	PL27B	6	PCLKC6_0	C (LVDS)*	PL37B	6	PCLKC6_0	C (LVDS)*
M3	PL28A	6	VREF2_6	T	PL38A	6	VREF2_6	T
M4	PL28B	6	VREF1_6	C	PL38B	6	VREF1_6	C
VCCIO	VCCIO6	6			VCCIO6	6		
N7	PL31A	6	LLM1_SPLLT_IN_A	T (LVDS)*	PL41A	6	LLM2_SPLLT_IN_A	T (LVDS)*
GNDIO	GNDIO6	-			GNDIO6	-		
N6	PL31B	6	LLM1_SPLLC_IN_A	C (LVDS)*	PL41B	6	LLM2_SPLLC_IN_A	C (LVDS)*
N1	PL32A	6	LLM1_SPLLT_FB_A	T	PL42A	6	LLM2_SPLLT_FB_A	T
N2	PL32B	6	LLM1_SPLLC_FB_A	C	PL42B	6	LLM2_SPLLC_FB_A	C
VCCIO	VCCIO6	6			VCCIO6	6		
GNDIO	GNDIO6	-			GNDIO6	-		
P6	PL38A	6	LDQS38****	T (LVDS)*	PL48A	6	LDQS48****	T (LVDS)*
N5	PL38B	6	LDQ38	C (LVDS)*	PL48B	6	LDQ48	C (LVDS)*
P1	PL39A	6	LDQ38	T	PL49A	6	LDQ48	T
VCCIO	VCCIO6	6			VCCIO6	6		
P2	PL39B	6	LDQ38	C	PL49B	6	LDQ48	C
P3	PL40A	6	LDQ38	T (LVDS)*	PL50A	6	LDQ48	T (LVDS)*
P4	PL40B	6	LDQ38	C (LVDS)*	PL50B	6	LDQ48	C (LVDS)*
P5	PL41A	6	LDQ38	T	PL51A	6	LDQ48	T
GNDIO	GNDIO6	-			GNDIO6	-		
P7	PL41B	6	LDQ38	C	PL51B	6	LDQ48	C
R1	PL42A	6	LLM0_GPLLT_IN_A**	T (LVDS)*	PL57A	6	LLM0_GPLLT_IN_A**/LDQS57****	T (LVDS)*
GNDIO	GNDIO6	-			GNDIO6	-		
R2	PL42B	6	LLM0_GPLLC_IN_A**	C (LVDS)*	PL57B	6	LLM0_GPLLC_IN_A**/LDQ57	C (LVDS)*
R3	PL43A	6	LLM0_GPLLT_FB_A	T	PL58A	6	LLM0_GPLLT_FB_A/LDQ57	T
R4	PL43B	6	LLM0_GPLLC_FB_A	C	PL58B	6	LLM0_GPLLC_FB_A/LDQ57	C
VCCIO	VCCIO6	6			VCCIO6	6		
R6	PL44A	6	LLM0_GDLLT_IN_A**	T (LVDS)*	PL59A	6	LLM0_GDLLT_IN_A**/LDQ57	T (LVDS)*
R5	PL44B	6	LLM0_GDLLC_IN_A**	C (LVDS)*	PL59B	6	LLM0_GDLLC_IN_A**/LDQ57	C (LVDS)*
T1	PL45A	6	LLM0_GDLLT_FB_A	T	PL60A	6	LLM0_GDLLT_FB_A/LDQ57	T
T2	PL45B	6	LLM0_GDLLC_FB_A	C	PL60B	6	LLM0_GDLLC_FB_A/LDQ57	C
GNDIO	GNDIO6	-			GNDIO6	-		
R7	LLM0_PLLCAP	6			LLM0_PLLCAP	6		
T6	PL47A	6	LDQ51	T (LVDS)*	PL62A	6	LDQ66	T (LVDS)*
T7	PL47B	6	LDQ51	C (LVDS)*	PL62B	6	LDQ66	C (LVDS)*
U1	PL48A	6	LDQ51	T	PL63A	6	LDQ66	T
U2	PL48B	6	LDQ51	C	PL63B	6	LDQ66	C
VCCIO	VCCIO6	6			VCCIO6	6		
T3	PL49A	6	LDQ51	T (LVDS)*	PL64A	6	LDQ66	T (LVDS)*
U3	PL49B	6	LDQ51	C (LVDS)*	PL64B	6	LDQ66	C (LVDS)*
U6	PL50A	6	LDQ51	T	NC	-		
U5	PL50B	6	LDQ51	C	PL65B	6	LDQ66	C
GNDIO	GNDIO6	-			GNDIO6	-		
V5	PL51A	6	LDQS51	T (LVDS)*	PL66A	6	LDQS66	T (LVDS)*
U4	PL51B	6	LDQ51	C (LVDS)*	PL66B	6	LDQ66	C (LVDS)*
V1	PL52A	6	LDQ51	T	PL67A	6	LDQ66	T

LFE2M20E/SE and LFE2M35E/SE Logic Signal Connections: 484 fpBGA

LFE2M20E/SE					LFE2M35E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
VCCIO	VCCIO6	6			VCCIO6	6		
V3	PL52B	6	LDQ51	C	PL67B	6	LDQ66	C
W1	PL53A	6	LDQ51	T (LVDS)*	PL68A	6	LDQ66	T (LVDS)*
Y1	PL53B	6	LDQ51	C (LVDS)*	PL68B	6	LDQ66	C (LVDS)*
AA1	PL54A	6	LDQ51	T	PL69A	6	LDQ66	T
GNDIO	GNDIO6	-			GNDIO6	-		
AA2	PL54B	6	LDQ51	C	PL69B	6	LDQ66	C
V4	TCK	-			TCK	-		
Y2	TDI	-			TDI	-		
Y3	TMS	-			TMS	-		
W3	TDO	-			TDO	-		
W4	VCCJ	-			VCCJ	-		
W5	PB2A	5	BDQ6	T	PB2A	5	BDQ6	T
Y4	PB2B	5	BDQ6	C	PB2B	5	BDQ6	C
W6	PB3A	5	BDQ6	T	PB3A	5	BDQ6	T
V6	PB3B	5	BDQ6	C	PB3B	5	BDQ6	C
AA3	PB4A	5	BDQ6	T	PB4A	5	BDQ6	T
VCCIO	VCCIO5	5			VCCIO5	5		
AB2	PB4B	5	BDQ6	C	PB4B	5	BDQ6	C
T8	PB5A	5	BDQ6	T	PB5A	5	BDQ6	T
U7	PB5B	5	BDQ6	C	PB5B	5	BDQ6	C
U8	PB6A	5	BDQS6	T	PB6A	5	BDQS6	T
GNDIO	GNDIO5	-			GNDIO5	-		
T9	PB6B	5	BDQ6	C	PB6B	5	BDQ6	C
V8	PB7A	5	BDQ6	T	PB7A	5	BDQ6	T
W8	PB7B	5	BDQ6	C	PB7B	5	BDQ6	C
Y6	PB8A	5	BDQ6	T	PB8A	5	BDQ6	T
VCCIO	VCCIO5	5			VCCIO5	5		
Y5	PB8B	5	BDQ6	C	PB8B	5	BDQ6	C
AB3	PB9A	5	BDQ6	T	PB9A	5	BDQ6	T
AB4	PB9B	5	BDQ6	C	PB9B	5	BDQ6	C
AB5	PB10A	5	BDQ6	T	PB10A	5	BDQ6	T
GNDIO	GNDIO5	-			GNDIO5	-		
AA6	PB10B	5	BDQ6	C	PB10B	5	BDQ6	C
V9	PB13A	5	BDQ15	T	PB31A	5	BDQ33	T
U9	PB13B	5	BDQ15	C	PB31B	5	BDQ33	C
VCCIO	VCCIO5	5			VCCIO5	5		
-	-	-			GNDIO5	-		
U10	PB14A	5	BDQ15	T	PB32A	5	BDQ33	T
T10	PB14B	5	BDQ15	C	PB32B	5	BDQ33	C
GNDIO	GNDIO5	-			GNDIO5	-		
W9	PB15A	5	BDQS15****	T	PB33A	5	BDQS33****	T
Y8	PB15B	5	BDQ15	C	PB33B	5	BDQ33	C
AA7	PB16A	5	VREF2_5/BDQ15	T	PB34A	5	VREF2_5/BDQ33	T
Y7	PB16B	5	VREF1_5/BDQ15	C	PB34B	5	VREF1_5/BDQ33	C
AB6	PB17A	5	PCLKT5_0/BDQ15	T	PB35A	5	PCLKT5_0/BDQ33	T
AB7	PB17B	5	PCLKC5_0/BDQ15	C	PB35B	5	PCLKC5_0/BDQ33	C
VCCIO	VCCIO5	5			VCCIO5	5		
GNDIO	GNDIO5	-			GNDIO5	-		

LFE2M20E/SE and LFE2M35E/SE Logic Signal Connections: 484 fpBGA

LFE2M20E/SE					LFE2M35E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
AA8	PB22A	4	PCLKT4_0/BDQ24	T	PB40A	4	PCLKT4_0/BDQ42	T
VCCIO	VCCIO4	4			VCCIO4	4		
AB8	PB22B	4	PCLKC4_0/BDQ24	C	PB40B	4	PCLKC4_0/BDQ42	C
AA9	PB23A	4	VREF2_4/BDQ24	T	PB41A	4	VREF2_4/BDQ42	T
Y9	PB23B	4	VREF1_4/BDQ24	C	PB41B	4	VREF1_4/BDQ42	C
AB9	PB24A	4	BDQS24****	T	PB42A	4	BDQS42****	T
GNDIO	GNDIO4	-			GNDIO4	-		
AB10	PB24B	4	BDQ24	C	PB42B	4	BDQ42	C
AA10	PB25A	4	BDQ24	T	PB43A	4	BDQ42	T
Y11	PB25B	4	BDQ24	C	PB43B	4	BDQ42	C
VCCIO	VCCIO4	4			VCCIO4	4		
GNDIO	GNDIO4	-			GNDIO4	-		
V10	PB29A	4	BDQ33	T	PB47A	4	BDQ51	T
U11	PB29B	4	BDQ33	C	PB47B	4	BDQ51	C
V11	PB30A	4	BDQ33	T	PB48A	4	BDQ51	T
W11	PB30B	4	BDQ33	C	PB48B	4	BDQ51	C
AA11	PB31A	4	BDQ33	T	PB49A	4	BDQ51	T
AB11	PB31B	4	BDQ33	C	PB49B	4	BDQ51	C
VCCIO	VCCIO4	4			VCCIO4	4		
T11	PB32A	4	BDQ33	T	PB50A	4	BDQ51	T
U12	PB32B	4	BDQ33	C	PB50B	4	BDQ51	C
GNDIO	GNDIO4	-			GNDIO4	-		
AA12	PB33A	4	BDQS33	T	PB51A	4	BDQS51	T
Y12	PB33B	4	BDQ33	C	PB51B	4	BDQ51	C
V12	PB34A	4	BDQ33	T	PB52A	4	BDQ51	T
W12	PB34B	4	BDQ33	C	PB52B	4	BDQ51	C
AB12	PB35A	4	BDQ33	T	PB53A	4	BDQ51	T
AA13	PB35B	4	BDQ33	C	PB53B	4	BDQ51	C
VCCIO	VCCIO4	4			VCCIO4	4		
T12	PB36A	4	BDQ33	T	PB54A	4	BDQ51	T
U13	PB36B	4	BDQ33	C	PB54B	4	BDQ51	C
V13	PB37A	4	BDQ33	T	PB55A	4	BDQ51	T
T13	PB37B	4	BDQ33	C	PB55B	4	BDQ51	C
GNDIO	GNDIO4	-			GNDIO4	-		
AB13	PB38A	4	BDQ42	T	PB56A	4	BDQ60	T
AB14	PB38B	4	BDQ42	C	PB56B	4	BDQ60	C
U14	PB39A	4	BDQ42	T	PB57A	4	BDQ60	T
T14	PB39B	4	BDQ42	C	PB57B	4	BDQ60	C
AA14	PB40A	4	BDQ42	T	PB58A	4	BDQ60	T
VCCIO	VCCIO4	4			VCCIO4	4		
Y14	PB40B	4	BDQ42	C	PB58B	4	BDQ60	C
W14	PB41A	4	BDQ42	T	PB59A	4	BDQ60	T
V14	PB41B	4	BDQ42	C	PB59B	4	BDQ60	C
AB15	PB42A	4	BDQS42	T	PB60A	4	BDQS60	T
GNDIO	GNDIO4	-			GNDIO4	-		
AA15	PB42B	4	BDQ42	C	PB60B	4	BDQ60	C
V15	PB43A	4	BDQ42	T	PB61A	4	BDQ60	T
U15	PB43B	4	BDQ42	C	PB61B	4	BDQ60	C
AB16	PB44A	4	BDQ42	T	PB62A	4	BDQ60	T

LFE2M20E/SE and LFE2M35E/SE Logic Signal Connections: 484 fpBGA

LFE2M20E/SE					LFE2M35E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
VCCIO	VCCIO4	4			VCCIO4	4		
AA16	PB44B	4	BDQ42	C	PB62B	4	BDQ60	C
AB17	PB45A	4	BDQ42	T	PB63A	4	BDQ60	T
AA17	PB45B	4	BDQ42	C	PB63B	4	BDQ60	C
Y15	PB46A	4	BDQ42	T	PB64A	4	BDQ60	T
GNDIO	GNDIO4	-			GNDIO4	-		
W15	PB46B	4	BDQ42	C	PB64B	4	BDQ60	C
AB20	PB47A	4	BDQ51	T	PB65A	4	BDQ69	T
AB21	PB47B	4	BDQ51	C	PB65B	4	BDQ69	C
AA21	PB48A	4	BDQ51	T	PB66A	4	BDQ69	T
AA20	PB48B	4	BDQ51	C	PB66B	4	BDQ69	C
AB19	PB49A	4	BDQ51	T	PB67A	4	BDQ69	T
AB18	PB49B	4	BDQ51	C	PB67B	4	BDQ69	C
VCCIO	VCCIO4	4			VCCIO4	4		
Y22	PB50A	4	BDQ51	T	PB68A	4	BDQ69	T
Y21	PB50B	4	BDQ51	C	PB68B	4	BDQ69	C
GNDIO	GNDIO4	-			GNDIO4	-		
Y17	PB51A	4	BDQS51	T	PB69A	4	BDQS69	T
Y18	PB51B	4	BDQ51	C	PB69B	4	BDQ69	C
Y16	PB52A	4	BDQ51	T	PB70A	4	BDQ69	T
W17	PB52B	4	BDQ51	C	PB70B	4	BDQ69	C
Y19	PB53A	4	BDQ51	T	PB71A	4	BDQ69	T
Y20	PB53B	4	BDQ51	C	PB71B	4	BDQ69	C
VCCIO	VCCIO4	4			VCCIO4	4		
W19	PB54A	4	BDQ51	T	PB72A	4	BDQ69	T
W18	PB54B	4	BDQ51	C	PB72B	4	BDQ69	C
V17	PB55A	4	BDQ51	T	PB73A	4	BDQ69	T
V18	PB55B	4	BDQ51	C	PB73B	4	BDQ69	C
GNDIO	GNDIO4	-			GNDIO4	-		
W20	CFG2	8			CFG2	8		
V20	CFG1	8			CFG1	8		
V19	CFG0	8			CFG0	8		
V22	PROGRAMN	8			PROGRAMN	8		
W22	CCLK	8			CCLK	8		
U18	INITN	8			INITN	8		
U22	DONE	8			DONE	8		
GNDIO	GNDIO8	-			GNDIO8	-		
U20	PR53B	8	WRITEN***	C	PR68B	8	WRITEN***	C
U21	PR53A	8	CS1N***	T	PR68A	8	CS1N***	T
U17	PR52B	8	CSN***	C	PR67B	8	CSN***	C
U16	PR52A	8	D0/SPIFASTN***	T	PR67A	8	D0/SPIFASTN***	T
VCCIO	VCCIO8	8			VCCIO8	8		
T16	PR51B	8	D1***	C	PR66B	8	D1***	C
T17	PR51A	8	D2***	T	PR66A	8	D2***	T
T22	PR50B	8	D3***	C	PR65B	8	D3***	C
GNDIO	GNDIO8	-			GNDIO8	-		
R22	PR50A	8	D4***	T	PR65A	8	D4***	T
T15	PR49B	8	D5***	C	PR64B	8	D5***	C
R17	PR49A	8	D6***	T	PR64A	8	D6***	T

LFE2M20E/SE and LFE2M35E/SE Logic Signal Connections: 484 fpBGA

LFE2M20E/SE					LFE2M35E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
T20	PR48B	8	D7/SPID0***	C	PR63B	8	D7/SPID0***	C
VCCIO	VCCIO8	8			VCCIO8	8		
T21	PR48A	8	DI/CSSPI0N***	T	PR63A	8	DI/CSSPI0N***	T
R21	PR47B	8	DOUT/CSON/CSSPI1N***	C	PR62B	8	DOUT/CSON/CSSPI1N***	C
R20	PR47A	8	BUSY/SISPI***	T	PR62A	8	BUSY/SISPI***	T
R16	RLM0_PLLCAP	3			RLM0_PLLCAP	3		
R18	PR45B	3	RLM0_GDLLC_FB_A	C	PR60B	3	RLM0_GDLLC_FB_A/RDQ57	C
GNDIO	GNDIO3	-			GNDIO3	-		
R19	PR45A	3	RLM0_GDLLT_FB_A	T	PR60A	3	RLM0_GDLLT_FB_A/RDQ57	T
P22	PR44B	3	RLM0_GDLLC_IN_A**	C (LVDS)*	PR59B	3	RLM0_GDLLC_IN_A**/RDQ57	C (LVDS)*
P21	PR44A	3	RLM0_GDLLT_IN_A**	T (LVDS)*	PR59A	3	RLM0_GDLLT_IN_A**/RDQ57	T (LVDS)*
P16	PR43B	3	RLM0_GPLLC_IN_A**	C	PR58B	3	RLM0_GPLLC_IN_A**/RDQ57	C
VCCIO	VCCIO3	3			VCCIO3	3		
P17	PR43A	3	RLM0_GPLLT_IN_A**	T	PR58A	3	RLM0_GPLLT_IN_A**/RDQ57	T
P20	PR42B	3	RLM0_GPLLC_FB_A	C (LVDS)*	PR57B	3	RLM0_GPLLC_FB_A/RDQ57	C (LVDS)*
P19	PR42A	3	RLM0_GPLLT_FB_A	T (LVDS)*	PR57A	3	RLM0_GPLLT_FB_A/RDQS57****	T (LVDS)*
GNDIO	GNDIO3	-			GNDIO3	-		
-	-	-			VCCIO3	3		
P18	PR41B	3	RDQ38	C	PR51B	3	RDQ48	C
N16	PR41A	3	RDQ38	T	PR51A	3	RDQ48	T
GNDIO	GNDIO3	-			GNDIO3	-		
N22	PR40B	3	RDQ38	C (LVDS)*	PR50B	3	RDQ48	C (LVDS)*
N21	PR40A	3	RDQ38	T (LVDS)*	PR50A	3	RDQ48	T (LVDS)*
N17	PR39B	3	RDQ38	C	PR49B	3	RDQ48	C
N18	PR39A	3	RDQ38	T	PR49A	3	RDQ48	T
VCCIO	VCCIO3	3			VCCIO3	3		
M22	PR38B	3	RDQ38	C (LVDS)*	PR48B	3	RDQ48	C (LVDS)*
M21	PR38A	3	RDQS38	T (LVDS)*	PR48A	3	RDQS48	T (LVDS)*
M16	PR37B	3	RDQ38	C	PR47B	3	RDQ48	C
GNDIO	GNDIO3	-			GNDIO3	-		
M17	PR37A	3	RDQ38	T	PR47A	3	RDQ48	T
M20	PR36B	3	RDQ38	C (LVDS)*	PR46B	3	RDQ48	C (LVDS)*
M19	PR36A	3	RDQ38	T (LVDS)*	PR46A	3	RDQ48	T (LVDS)*
M18	PR35B	3	RDQ38	C	PR45B	3	RDQ48	C
VCCIO	VCCIO3	3			VCCIO3	3		
L16	PR35A	3	RDQ38	T	PR45A	3	RDQ48	T
L22	PR34B	3	RDQ38	C (LVDS)*	PR44B	3	RDQ48	C (LVDS)*
L21	PR34A	3	RDQ38	T (LVDS)*	PR44A	3	RDQ48	T (LVDS)*
K22	PR32B	3	RLM1_SPLLC_FB_A	C	PR42B	3	RLM2_SPLLC_FB_A	C
VCCIO	VCCIO3	3			VCCIO3	3		
K21	PR32A	3	RLM1_SPLLT_FB_A	T	PR42A	3	RLM2_SPLLT_FB_A	T
L17	PR31B	3	RLM1_SPLLC_IN_A	C (LVDS)*	PR41B	3	RLM2_SPLLC_IN_A	C (LVDS)*
L18	PR31A	3	RLM1_SPLLT_IN_A	T (LVDS)*	PR41A	3	RLM2_SPLLT_IN_A	T (LVDS)*
GNDIO	GNDIO3	-			GNDIO3	-		
L20	PR30B	3		C	PR40B	3		C
L19	PR30A	3		T	PR40A	3		T
K16	PR29B	3		C (LVDS)*	PR39B	3		C (LVDS)*
K17	PR29A	3		T (LVDS)*	PR39A	3		T (LVDS)*
VCCIO	VCCIO3	3			VCCIO3	3		

LFE2M20E/SE and LFE2M35E/SE Logic Signal Connections: 484 fpBGA

LFE2M20E/SE					LFE2M35E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
J16	PR28B	3	VREF2_3	C	PR38B	3	VREF2_3	C
K18	PR28A	3	VREF1_3	T	PR38A	3	VREF1_3	T
J22	PR27B	3	PCLKC3_0	C (LVDS)*	PR37B	3	PCLKC3_0	C (LVDS)*
J21	PR27A	3	PCLKT3_0	T (LVDS)*	PR37A	3	PCLKT3_0	T (LVDS)*
H22	PR25B	2	PCLKC2_0/RDQ22	C	PR35B	2	PCLKC2_0/RDQ32	C
H21	PR25A	2	PCLKT2_0/RDQ22	T	PR35A	2	PCLKT2_0/RDQ32	T
GNDIO	GNDIO2	-			GNDIO2	-		
J17	PR24B	2	RDQ22	C (LVDS)*	PR34B	2	RDQ32	C (LVDS)*
J18	PR24A	2	RDQ22	T (LVDS)*	PR34A	2	RDQ32	T (LVDS)*
J20	PR23B	2	RDQ22	C	PR33B	2	RDQ32	C
J19	PR23A	2	RDQ22	T	PR33A	2	RDQ32	T
VCCIO	VCCIO2	2			VCCIO2	2		
H16	PR22B	2	RDQ22	C (LVDS)*	PR32B	2	RDQ32	C (LVDS)*
H17	PR22A	2	RDQS22	T (LVDS)*	PR32A	2	RDQS32	T (LVDS)*
G22	PR21B	2	RDQ22	C	PR31B	2	RDQ32	C
GNDIO	GNDIO2	-			GNDIO2	-		
G21	PR21A	2	RDQ22	T	PR31A	2	RDQ32	T
H20	PR20B	2	RDQ22	C (LVDS)*	PR30B	2	RDQ32	C (LVDS)*
H19	PR20A	2	RDQ22	T (LVDS)*	PR30A	2	RDQ32	T (LVDS)*
G16	PR19B	2	RUM1_SPLLC_FB_A/RDQ22	C	PR29B	2	RUM1_SPLLC_FB_A/RDQ32	C
VCCIO	VCCIO2	2			VCCIO2	2		
H18	PR19A	2	RUM1_SPLLT_FB_A/RDQ22	T	PR29A	2	RUM1_SPLLT_FB_A/RDQ32	T
F22	PR18B	2	RUM1_SPLLC_IN_A/RDQ22	C (LVDS)*	PR28B	2	RUM1_SPLLC_IN_A/RDQ32	C (LVDS)*
F21	PR18A	2	RUM1_SPLLT_IN_A/RDQ22	T (LVDS)*	PR28A	2	RUM1_SPLLT_IN_A/RDQ32	T (LVDS)*
GNDIO	GNDIO2	-			-	-		
G20	PR16B	2		C	PR26B	2	RDQ23	C
VCCIO	VCCIO2	2			-	-		
F20	PR16A	2		T	PR26A	2	RDQ23	T
-	-	-			GNDIO2	-		
G17	PR15B	2		C (LVDS)*	PR25B	2	RDQ23	C (LVDS)*
F17	PR15A	2		T (LVDS)*	PR25A	2	RDQ23	T (LVDS)*
-	-	-			VCCIO2	2		
GNDIO	GNDIO2	-			GNDIO2	-		
E22	PR14B	2		C	PR14B	2	RDQ15	C
D22	PR14A	2		T	PR14A	2	RDQ15	T
E20	PR13B	2		C (LVDS)*	PR13B	2	RDQ15	C (LVDS)*
D20	PR13A	2		T (LVDS)*	PR13A	2	RDQ15	T (LVDS)*
VCCIO	VCCIO2	2			VCCIO2	2		
D19	PR12B	2	RUM0_SPLLC_FB_A	C	PR12B	2	RUM0_SPLLC_FB_A/RDQ15	C
E19	PR12A	2	RUM0_SPLLT_FB_A	T	PR12A	2	RUM0_SPLLT_FB_A/RDQ15	T
F18	PR11B	2	RUM0_SPLLC_IN_A	C (LVDS)*	PR11B	2	RUM0_SPLLC_IN_A/RDQ15	C (LVDS)*
F19	PR11A	2	RUM0_SPLLT_IN_A	T (LVDS)*	PR11A	2	RUM0_SPLLT_IN_A/RDQ15	T (LVDS)*
E18	PR9B	2	VREF2_2	C	PR9B	2	VREF2_2	C
GNDIO	GNDIO2	-			GNDIO2	-		
D18	PR9A	2	VREF1_2	T	PR9A	2	VREF1_2	T
VCCIO	VCCIO2	2			-	-		
F16	XRES	-			XRES	-		
C22	URC_SQ_VCCRX0	12			URC_SQ_VCCRX0	12		
A21	URC_SQ_HDINP0	12		T	URC_SQ_HDINP0	12		T

LFE2M20E/SE and LFE2M35E/SE Logic Signal Connections: 484 fpBGA

LFE2M20E/SE					LFE2M35E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
B22	URC_SQ_VCCIB0	12			URC_SQ_VCCIB0	12		
B21	URC_SQ_HDINN0	12		C	URC_SQ_HDINN0	12		C
C19	URC_SQ_VCCTX0	12			URC_SQ_VCCTX0	12		
A18	URC_SQ_HDOUTP0	12		T	URC_SQ_HDOUTP0	12		T
A19	URC_SQ_VCCOB0	12			URC_SQ_VCCOB0	12		
B18	URC_SQ_HDOUTN0	12		C	URC_SQ_HDOUTN0	12		C
C18	URC_SQ_VCCTX1	12			URC_SQ_VCCTX1	12		
B17	URC_SQ_HDOUTN1	12		C	URC_SQ_HDOUTN1	12		C
C17	URC_SQ_VCCOB1	12			URC_SQ_VCCOB1	12		
A17	URC_SQ_HDOUTP1	12		T	URC_SQ_HDOUTP1	12		T
C21	URC_SQ_VCCRX1	12			URC_SQ_VCCRX1	12		
B20	URC_SQ_HDINN1	12		C	URC_SQ_HDINN1	12		C
C20	URC_SQ_VCCIB1	12			URC_SQ_VCCIB1	12		
A20	URC_SQ_HDINP1	12		T	URC_SQ_HDINP1	12		T
B16	URC_SQ_VCCAUX33	12			URC_SQ_VCCAUX33	12		
E17	URC_SQ_REFCLKN	12		C	URC_SQ_REFCLKN	12		C
D17	URC_SQ_REFCLKP	12		T	URC_SQ_REFCLKP	12		T
C16	URC_SQ_VCCP	12			URC_SQ_VCCP	12		
A12	URC_SQ_HDINP2	12		T	URC_SQ_HDINP2	12		T
C12	URC_SQ_VCCIB2	12			URC_SQ_VCCIB2	12		
B12	URC_SQ_HDINN2	12		C	URC_SQ_HDINN2	12		C
C11	URC_SQ_VCCRX2	12			URC_SQ_VCCRX2	12		
A15	URC_SQ_HDOUTP2	12		T	URC_SQ_HDOUTP2	12		T
C15	URC_SQ_VCCOB2	12			URC_SQ_VCCOB2	12		
B15	URC_SQ_HDOUTN2	12		C	URC_SQ_HDOUTN2	12		C
C14	URC_SQ_VCCTX2	12			URC_SQ_VCCTX2	12		
B14	URC_SQ_HDOUTN3	12		C	URC_SQ_HDOUTN3	12		C
A13	URC_SQ_VCCOB3	12			URC_SQ_VCCOB3	12		
A14	URC_SQ_HDOUTP3	12		T	URC_SQ_HDOUTP3	12		T
C13	URC_SQ_VCCTX3	12			URC_SQ_VCCTX3	12		
B11	URC_SQ_HDINN3	12		C	URC_SQ_HDINN3	12		C
B10	URC_SQ_VCCIB3	12			URC_SQ_VCCIB3	12		
A11	URC_SQ_HDINP3	12		T	URC_SQ_HDINP3	12		T
C10	URC_SQ_VCCRX3	12			URC_SQ_VCCRX3	12		
E13	PT28B	1		C	PT46B	1		C
D12	PT28A	1		T	PT46A	1		T
GNDIO	GNDIO1	-			GNDIO1	-		
A9	PT27B	1		C	PT45B	1		C
A8	PT27A	1		T	PT45A	1		T
A7	PT26B	1		C	PT44B	1		C
A6	PT26A	1		T	PT44A	1		T
VCCIO	VCCIO1	1			VCCIO1	1		
E12	PT25B	1		C	PT43B	1		C

LFE2M20E/SE and LFE2M35E/SE Logic Signal Connections: 484 fpBGA

LFE2M20E/SE					LFE2M35E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
F12	PT25A	1		T	PT43A	1		T
A5	PT24B	1		C	PT42B	1		C
A4	PT24A	1		T	PT42A	1		T
GNDIO	GNDIO1	-			GNDIO1	-		
B7	PT23B	1		C	PT41B	1		C
B8	PT23A	1		T	PT41A	1		T
G11	PT22B	1		C	PT40B	1		C
E11	PT22A	1		T	PT40A	1		T
VCCIO	VCCIO1	1			VCCIO1	1		
D11	PT21B	1	VREF2_1	C	PT39B	1	VREF2_1	C
D10	PT21A	1	VREF1_1	T	PT39A	1	VREF1_1	T
F11	PT20A	1	PCLKT1_0	T	PT38A	1	PCLKT1_0	T
G10	PT20B	1	PCLKC1_0	C	PT38B	1	PCLKC1_0	C
G9	PT19B	0	PCLKC0_0	C	PT37B	0	PCLKC0_0	C
GNDIO	GNDIO0	-			GNDIO0	-		
F9	PT19A	0	PCLKT0_0	T	PT37A	0	PCLKT0_0	T
C9	PT18B	0	VREF2_0	C	PT36B	0	VREF2_0	C
D9	PT18A	0	VREF1_0	T	PT36A	0	VREF1_0	T
A2	PT17B	0		C	PT35B	0		C
VCCIO	VCCIO0	0			VCCIO0	0		
A3	PT17A	0		T	PT35A	0		T
B3	PT16B	0		C	PT34B	0		C
C4	PT16A	0		T	PT34A	0		T
E10	PT15B	0		C	PT33B	0		C
F10	PT15A	0		T	PT33A	0		T
C7	PT14B	0		C	PT32B	0		C
GNDIO	GNDIO0	-			GNDIO0	-		
B6	PT14A	0		T	PT32A	0		T
C6	PT13B	0		C	PT31B	0		C
VCCIO	VCCIO0	0			VCCIO0	0		
C5	PT13A	0		T	PT31A	0		T
C8	PT12B	0		C	PT30B	0		C
D8	PT12A	0		T	PT30A	0		T
E8	PT11B	0		C	PT29B	0		C
E9	PT11A	0		T	PT29A	0		T
-	-	-			GNDIO0	-		
-	-	-			VCCIO0	0		
F8	PT10B	0		C	PT10B	0		C
G8	PT10A	0		T	PT10A	0		T
GNDIO	GNDIO0	-			GNDIO0	-		
F7	PT9B	0		C	PT9B	0		C
G7	PT9A	0		T	PT9A	0		T
C3	PT8B	0		C	PT8B	0		C
D4	PT8A	0		T	PT8A	0		T
VCCIO	VCCIO0	0			VCCIO0	0		
F6	PT7B	0		C	PT7B	0		C
E6	PT7A	0		T	PT7A	0		T
E5	PT6B	0		C	PT6B	0		C
D6	PT6A	0		T	PT6A	0		T

LFE2M20E/SE and LFE2M35E/SE Logic Signal Connections: 484 fpBGA

LFE2M20E/SE					LFE2M35E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
GNDIO	GNDIO0	-			GNDIO0	-		
D3	PT5B	0		C	PT5B	0		C
E3	PT5A	0		T	PT5A	0		T
D5	PT4B	0		C	PT4B	0		C
E4	PT4A	0		T	PT4A	0		T
VCCIO	VCCIO0	0			VCCIO0	0		
C2	PT3B	0		C	PT3B	0		C
B2	PT3A	0		T	PT3A	0		T
B1	PT2B	0		C	PT2B	0		C
C1	PT2A	0		T	PT2A	0		T
R8	VCCPLL	-			VCCPLL	-		
H15	VCCPLL	-			VCCPLL	-		
H8	VCCPLL	-			VCCPLL	-		
R15	VCCPLL	-			VCCPLL	-		
J10	VCC	-			VCC	-		
J11	VCC	-			VCC	-		
J12	VCC	-			VCC	-		
J13	VCC	-			VCC	-		
K14	VCC	-			VCC	-		
K9	VCC	-			VCC	-		
L14	VCC	-			VCC	-		
L9	VCC	-			VCC	-		
M14	VCC	-			VCC	-		
M9	VCC	-			VCC	-		
N14	VCC	-			VCC	-		
N9	VCC	-			VCC	-		
P10	VCC	-			VCC	-		
P11	VCC	-			VCC	-		
P12	VCC	-			VCC	-		
P13	VCC	-			VCC	-		
B5	VCCIO0	0			VCCIO0	0		
B9	VCCIO0	0			VCCIO0	0		
E7	VCCIO0	0			VCCIO0	0		
H9	VCCIO0	0			VCCIO0	0		
D13	VCCIO1	1			VCCIO1	1		
E16	VCCIO1	1			VCCIO1	1		
H14	VCCIO1	1			VCCIO1	1		
E21	VCCIO2	2			VCCIO2	2		
G18	VCCIO2	2			VCCIO2	2		
J15	VCCIO2	2			VCCIO2	2		
K19	VCCIO2	2			VCCIO2	2		
N19	VCCIO3	3			VCCIO3	3		
P15	VCCIO3	3			VCCIO3	3		
T18	VCCIO3	3			VCCIO3	3		
V21	VCCIO3	3			VCCIO3	3		
AA18	VCCIO4	4			VCCIO4	4		
R14	VCCIO4	4			VCCIO4	4		
V16	VCCIO4	4			VCCIO4	4		
W13	VCCIO4	4			VCCIO4	4		

LFE2M20E/SE and LFE2M35E/SE Logic Signal Connections: 484 fpBGA

LFE2M20E/SE					LFE2M35E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
AA5	VCCIO5	5			VCCIO5	5		
R9	VCCIO5	5			VCCIO5	5		
V7	VCCIO5	5			VCCIO5	5		
W10	VCCIO5	5			VCCIO5	5		
N4	VCCIO6	6			VCCIO6	6		
P8	VCCIO6	6			VCCIO6	6		
T5	VCCIO6	6			VCCIO6	6		
V2	VCCIO6	6			VCCIO6	6		
E2	VCCIO7	7			VCCIO7	7		
G5	VCCIO7	7			VCCIO7	7		
J8	VCCIO7	7			VCCIO7	7		
K4	VCCIO7	7			VCCIO7	7		
AA22	VCCIO8	8			VCCIO8	8		
U19	VCCIO8	8			VCCIO8	8		
H11	VCCAUX	-			VCCAUX	-		
H12	VCCAUX	-			VCCAUX	-		
L15	VCCAUX	-			VCCAUX	-		
L8	VCCAUX	-			VCCAUX	-		
M15	VCCAUX	-			VCCAUX	-		
M8	VCCAUX	-			VCCAUX	-		
R11	VCCAUX	-			VCCAUX	-		
R12	VCCAUX	-			VCCAUX	-		
A1	GND	-			GND	-		
A10	GND	-			GND	-		
A16	GND	-			GND	-		
A22	GND	-			GND	-		
AA19	GND	-			GND	-		
AA4	GND	-			GND	-		
AB1	GND	-			GND	-		
AB22	GND	-			GND	-		
B13	GND	-			GND	-		
B19	GND	-			GND	-		
B4	GND	-			GND	-		
D16	GND	-			GND	-		
D2	GND	-			GND	-		
D21	GND	-			GND	-		
D7	GND	-			GND	-		
G19	GND	-			GND	-		
G4	GND	-			GND	-		
H10	GND	-			GND	-		
H13	GND	-			GND	-		
J14	GND	-			GND	-		
J9	GND	-			GND	-		
K10	GND	-			GND	-		
K11	GND	-			GND	-		
K12	GND	-			GND	-		
K13	GND	-			GND	-		
K15	GND	-			GND	-		
K20	GND	-			GND	-		

LFE2M20E/SE and LFE2M35E/SE Logic Signal Connections: 484 fpBGA

LFE2M20E/SE					LFE2M35E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
K3	GND	-			GND	-		
K8	GND	-			GND	-		
L10	GND	-			GND	-		
L11	GND	-			GND	-		
L12	GND	-			GND	-		
L13	GND	-			GND	-		
M10	GND	-			GND	-		
M11	GND	-			GND	-		
M12	GND	-			GND	-		
M13	GND	-			GND	-		
N10	GND	-			GND	-		
N11	GND	-			GND	-		
N12	GND	-			GND	-		
N13	GND	-			GND	-		
N15	GND	-			GND	-		
N20	GND	-			GND	-		
N3	GND	-			GND	-		
N8	GND	-			GND	-		
P14	GND	-			GND	-		
P9	GND	-			GND	-		
R10	GND	-			GND	-		
R13	GND	-			GND	-		
T19	GND	-			GND	-		
T4	GND	-			GND	-		
W16	GND	-			GND	-		
W2	GND	-			GND	-		
W21	GND	-			GND	-		
W7	GND	-			GND	-		
Y10	GND	-			GND	-		
Y13	GND	-			GND	-		
D15	NC	-			NC	-		
G14	NC	-			NC	-		
G15	NC	-			NC	-		
D14	NC	-			NC	-		
E15	NC	-			NC	-		
E14	NC	-			NC	-		
F15	NC	-			NC	-		
F14	NC	-			NC	-		
F13	NC	-			NC	-		
G12	NC	-			NC	-		
G13	NC	-			NC	-		

* Supports true LVDS. Other differential signals must be emulated with external resistors.
** These dedicated input pins can be used for GPLLs or GDLLs within the respective quadrant.
***For density migration, board design must take into account that these sysCONFIG pins are dual function for the lower density devices (ECP2M20 and ECP2M35). They can be either sysCONFIG pins or general purpose I/Os. These pins are dedicated sysCONFIG pins for the higher density devices (ECP2M50, ECP2M70 and ECP2M100).
****Due to packaging bond out option, this DQS does not have all the necessary DQ pins bonded out for a full 8-bit data width.

Note: VCCIO and GND pads are used to determine the average DC current drawn by I/Os between GND/VCCIO connections, or between the last GND/VCCIO in an I/O bank and the end of an I/O bank. The substrate pads listed in the Pin Table do not necessarily have a one to one connection with a package ball or pin.

LFE2M50E/SE Logic Signal Connections: 484 fpBGA

LFE2M50E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
D1	PL2A	7	LDQ6	T (LVDS)*
E1	PL2B	7	LDQ6	C (LVDS)*
F1	PL3A	7	LDQ6	T
F2	PL3B	7	LDQ6	C
F5	PL4A	7	LDQ6	T (LVDS)*
VCCIO	VCCIO7	7		
G6	PL4B	7	LDQ6	C (LVDS)*
F4	PL5A	7	LDQ6	T
F3	PL5B	7	LDQ6	C
G1	PL6A	7	LDQS6	T (LVDS)*
GNDIO	GNDIO7	-		
G2	PL6B	7	LDQ6	C (LVDS)*
H1	PL7A	7	LDQ6	T
H2	PL7B	7	LDQ6	C
VCCIO	VCCIO7	7		
H7	PL8A	7	LDQ6	T (LVDS)*
H6	PL8B	7	LDQ6	C (LVDS)*
G3	PL9A	7	VREF2_7/LDQ6	T
H3	PL9B	7	VREF1_7/LDQ6	C
GNDIO	GNDIO7	-		
VCCIO	VCCIO7	7		
H5	PL11A	7	LUM0_SPLLT_IN_A	T (LVDS)*
H4	PL11B	7	LUM0_SPLLC_IN_A	C (LVDS)*
J1	PL12A	7	LUM0_SPLLT_FB_A	T
J2	PL12B	7	LUM0_SPLLC_FB_A	C
GNDIO	GNDIO7	-		
J3	PL13A	7		T (LVDS)*
J4	PL13B	7		C (LVDS)*
J7	PL14A	7		T
VCCIO	VCCIO7	7		
J6	PL14B	7		C
GNDIO	GNDIO7	-		
VCCIO	VCCIO7	7		
K1	PL32A	7	LUM3_SPLLT_IN_A/LDQ36	T (LVDS)*
K2	PL32B	7	LUM3_SPLLC_IN_A/LDQ36	C (LVDS)*
J5	PL33A	7	LUM3_SPLLT_FB_A/LDQ36	T
K5	PL33B	7	LUM3_SPLLC_FB_A/LDQ36	C
VCCIO	VCCIO7	7		
K7	PL34A	7	LDQ36	T (LVDS)*
K6	PL34B	7	LDQ36	C (LVDS)*
L6	PL35A	7	LDQ36	T
L7	PL35B	7	LDQ36	C

LFE2M50E/SE Logic Signal Connections: 484 fpBGA (Cont.)

LFE2M50E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
GNDIO	GNDIO7	-		
L1	PL36A	7	LDQS36	T (LVDS)*
L2	PL36B	7	LDQ36	C (LVDS)*
M7	PL37A	7	LDQ36	T
VCCIO	VCCIO7	7		
L5	PL37B	7	LDQ36	C
L3	PL38A	7	LDQ36	T (LVDS)*
L4	PL38B	7	LDQ36	C (LVDS)*
M1	PL39A	7	PCLKT7_0/LDQ36	T
GNDIO	GNDIO7	-		
M2	PL39B	7	PCLKC7_0/LDQ36	C
M6	PL41A	6	PCLKT6_0	T (LVDS)*
M5	PL41B	6	PCLKC6_0	C (LVDS)*
M3	PL42A	6	VREF2_6	T
M4	PL42B	6	VREF1_6	C
VCCIO	VCCIO6	6		
N7	PL45A	6	LLM3_SPLLT_IN_A	T (LVDS)*
GNDIO	GNDIO6	-		
N6	PL45B	6	LLM3_SPLLC_IN_A	C (LVDS)*
N1	PL46A	6	LLM3_SPLLT_FB_A	T
N2	PL46B	6	LLM3_SPLLC_FB_A	C
VCCIO	VCCIO6	6		
GNDIO	GNDIO6	-		
P6	PL52A	6	LDQS52****	T (LVDS)*
N5	PL52B	6	LDQ52	C (LVDS)*
P1	PL53A	6	LDQ52	T
VCCIO	VCCIO6	6		
P2	PL53B	6	LDQ52	C
P3	PL54A	6	LDQ52	T (LVDS)*
P4	PL54B	6	LDQ52	C (LVDS)*
P5	PL55A	6	LDQ52	T
GNDIO	GNDIO6	-		
P7	PL55B	6	LDQ52	C
VCCIO	VCCIO6	6		
GNDIO	GNDIO6	-		
R1	PL62A	6	LLM0_GPLLT_IN_A**	T (LVDS)*
GNDIO	GNDIO6	-		
R2	PL62B	6	LLM0_GPLLC_IN_A**	C (LVDS)*
R3	PL63A	6	LLM0_GPLLT_FB_A	T
R4	PL63B	6	LLM0_GPLLC_FB_A	C
VCCIO	VCCIO6	6		
R6	PL64A	6	LLM0_GDLLT_IN_A**	T (LVDS)*
R5	PL64B	6	LLM0_GDLLC_IN_A**	C (LVDS)*

LFE2M50E/SE Logic Signal Connections: 484 fpBGA (Cont.)

LFE2M50E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
T1	PL65A	6	LLM0_GDLLT_FB_A	T
T2	PL65B	6	LLM0_GDLLC_FB_A	C
GNDIO	GNDIO6	-		
R7	LLM0_PLLCAP	6		
T6	PL67A	6	LDQ71	T (LVDS)*
T7	PL67B	6	LDQ71	C (LVDS)*
U1	PL68A	6	LDQ71	T
U2	PL68B	6	LDQ71	C
VCCIO	VCCIO6	6		
T3	PL69A	6	LDQ71	T (LVDS)*
U3	PL69B	6	LDQ71	C (LVDS)*
U6	PL70A	6	LDQ71	T
U5	PL70B	6	LDQ71	C
GNDIO	GNDIO6	-		
V5	PL71A	6	LDQS71	T (LVDS)*
U4	PL71B	6	LDQ71	C (LVDS)*
V1	PL72A	6	LDQ71	T
VCCIO	VCCIO6	6		
V3	PL72B	6	LDQ71	C
W1	PL73A	6	LDQ71	T (LVDS)*
Y1	PL73B	6	LDQ71	C (LVDS)*
AA1	PL74A	6	LDQ71	T
GNDIO	GNDIO6	-		
AA2	PL74B	6	LDQ71	C
V4	TCK	-		
Y2	TDI	-		
Y3	TMS	-		
W3	TDO	-		
W4	VCCJ	-		
W5	PB2A	5	BDQ6	T
Y4	PB2B	5	BDQ6	C
W6	PB3A	5	BDQ6	T
V6	PB3B	5	BDQ6	C
AA3	PB4A	5	BDQ6	T
AB2	PB4B	5	BDQ6	C
VCCIO	VCCIO5	5		
T8	PB5A	5	BDQ6	T
U7	PB5B	5	BDQ6	C
GNDIO	GNDIO5	-		
U8	PB6A	5	BDQS6	T
T9	PB6B	5	BDQ6	C
V8	PB7A	5	BDQ6	T
W8	PB7B	5	BDQ6	C

LFE2M50E/SE Logic Signal Connections: 484 fpBGA (Cont.)

LFE2M50E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
Y6	PB8A	5	BDQ6	T
Y5	PB8B	5	BDQ6	C
VCCIO	VCCIO5	5		
AB3	PB9A	5	BDQ6	T
AB4	PB9B	5	BDQ6	C
AB5	PB10A	5	BDQ6	T
AA6	PB10B	5	BDQ6	C
GNDIO	GNDIO5	-		
VCCIO	VCCIO5	5		
V9	PB40A	5	BDQ42	T
U9	PB40B	5	BDQ42	C
VCCIO	VCCIO5	5		
U10	PB41A	5	BDQ42	T
T10	PB41B	5	BDQ42	C
GNDIO	GNDIO5	-		
W9	PB42A	5	BDQS42****	T
Y8	PB42B	5	BDQ42	C
AA7	PB43A	5	VREF2_5/BDQ42	T
Y7	PB43B	5	VREF1_5/BDQ42	C
AB6	PB44A	5	PCLKT5_0/BDQ42	T
AB7	PB44B	5	PCLKC5_0/BDQ42	C
VCCIO	VCCIO5	5		
GNDIO	GNDIO5	-		
AA8	PB49A	4	PCLKT4_0/BDQ51	T
VCCIO	VCCIO4	4		
AB8	PB49B	4	PCLKC4_0/BDQ51	C
AA9	PB50A	4	VREF2_4/BDQ51	T
Y9	PB50B	4	VREF1_4/BDQ51	C
AB9	PB51A	4	BDQS51****	T
GNDIO	GNDIO4	-		
AB10	PB51B	4	BDQ51	C
AA10	PB52A	4	BDQ51	T
Y11	PB52B	4	BDQ51	C
VCCIO	VCCIO4	4		
GNDIO	GNDIO4	-		
V10	PB56A	4	BDQ60	T
U11	PB56B	4	BDQ60	C
V11	PB57A	4	BDQ60	T
W11	PB57B	4	BDQ60	C
AA11	PB58A	4	BDQ60	T
AB11	PB58B	4	BDQ60	C
VCCIO	VCCIO4	4		
T11	PB59A	4	BDQ60	T

LFE2M50E/SE Logic Signal Connections: 484 fpBGA (Cont.)

LFE2M50E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
U12	PB59B	4	BDQ60	C
GNDIO	GNDIO4	-		
AA12	PB60A	4	BDQS60	T
Y12	PB60B	4	BDQ60	C
V12	PB61A	4	BDQ60	T
W12	PB61B	4	BDQ60	C
AB12	PB62A	4	BDQ60	T
AA13	PB62B	4	BDQ60	C
VCCIO	VCCIO4	4		
T12	PB63A	4	BDQ60	T
U13	PB63B	4	BDQ60	C
V13	PB64A	4	BDQ60	T
T13	PB64B	4	BDQ60	C
GNDIO	GNDIO4	-		
AB13	PB65A	4	BDQ69	T
AB14	PB65B	4	BDQ69	C
U14	PB66A	4	BDQ69	T
T14	PB66B	4	BDQ69	C
AA14	PB67A	4	BDQ69	T
VCCIO	VCCIO4	4		
Y14	PB67B	4	BDQ69	C
W14	PB68A	4	BDQ69	T
V14	PB68B	4	BDQ69	C
AB15	PB69A	4	BDQS69	T
GNDIO	GNDIO4	-		
AA15	PB69B	4	BDQ69	C
V15	PB70A	4	BDQ69	T
U15	PB70B	4	BDQ69	C
AB16	PB71A	4	BDQ69	T
VCCIO	VCCIO4	4		
AA16	PB71B	4	BDQ69	C
AB17	PB72A	4	BDQ69	T
AA17	PB72B	4	BDQ69	C
GNDIO	GNDIO4	-		
W20	CFG2	8		
V20	CFG1	8		
V19	CFG0	8		
V22	PROGRAMN	8		
W22	CCLK	8		
U18	INITN	8		
U22	DONE	8		
GNDIO	GNDIO8	-		
U20	WRITEN***	8		

LFE2M50E/SE Logic Signal Connections: 484 fpBGA (Cont.)

LFE2M50E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
U21	CS1N***	8		
U17	CSN***	8		
U16	D0/SPIFASTN***	8		
VCCIO	VCCIO8	8		
T16	D1***	8		
T17	D2***	8		
T22	D3***	8		
GNDIO	GNDIO8	-		
R22	D4***	8		
T15	D5***	8		
R17	D6***	8		
T20	D7/SPID0***	8		
VCCIO	VCCIO8	8		
T21	DI/CSSPI0N***	8		
R21	DOUT/CSON/CSSPI1N***	8		
R20	BUSY/SISPI***	8		
R16	RLM0_PLLCAP	3		
R18	PR65B	3	RLM0_GDLLC_FB_A	C
GNDIO	GNDIO3	-		
R19	PR65A	3	RLM0_GDLLT_FB_A	T
P22	PR64B	3	RLM0_GDLLC_IN_A**	C (LVDS)*
P21	PR64A	3	RLM0_GDLLT_IN_A**	T (LVDS)*
P16	PR63B	3	RLM0_GPLLC_IN_A**	C
VCCIO	VCCIO3	3		
P17	PR63A	3	RLM0_GPLLT_IN_A**	T
P20	PR62B	3	RLM0_GPLLC_FB_A	C (LVDS)*
P19	PR62A	3	RLM0_GPLLT_FB_A	T (LVDS)*
GNDIO	GNDIO3	-		
VCCIO	VCCIO3	3		
P18	PR55B	3	RDQ52	C
N16	PR55A	3	RDQ52	T
GNDIO	GNDIO3	-		
N22	PR54B	3	RDQ52	C (LVDS)*
N21	PR54A	3	RDQ52	T (LVDS)*
N17	PR53B	3	RDQ52	C
N18	PR53A	3	RDQ52	T
VCCIO	VCCIO3	3		
M22	PR52B	3	RDQ52	C (LVDS)*
M21	PR52A	3	RDQS52	T (LVDS)*
M16	PR51B	3	RDQ52	C
GNDIO	GNDIO3	-		
M17	PR51A	3	RDQ52	T
M20	PR50B	3	RDQ52	C (LVDS)*

LFE2M50E/SE Logic Signal Connections: 484 fpBGA (Cont.)

LFE2M50E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
M19	PR50A	3	RDQ52	T (LVDS)*
M18	PR49B	3	RDQ52	C
VCCIO	VCCIO3	3		
L16	PR49A	3	RDQ52	T
L22	PR48B	3	RDQ52	C (LVDS)*
L21	PR48A	3	RDQ52	T (LVDS)*
GNDIO	GNDIO3	-		
K22	PR46B	3	RLM3_SPLLC_FB_A	C
VCCIO	VCCIO3	3		
K21	PR46A	3	RLM3_SPLLT_FB_A	T
L17	PR45B	3	RLM3_SPLLC_IN_A	C (LVDS)*
L18	PR45A	3	RLM3_SPLLT_IN_A	T (LVDS)*
GNDIO	GNDIO3	-		
L20	PR44B	3		C
L19	PR44A	3		T
K16	PR43B	3		C (LVDS)*
K17	PR43A	3		T (LVDS)*
VCCIO	VCCIO3	3		
J16	PR42B	3	VREF2_3	C
K18	PR42A	3	VREF1_3	T
J22	PR41B	3	PCLKC3_0	C (LVDS)*
J21	PR41A	3	PCLKT3_0	T (LVDS)*
H22	PR39B	2	PCLKC2_0/RDQ36	C
H21	PR39A	2	PCLKT2_0/RDQ36	T
GNDIO	GNDIO2	-		
J17	PR38B	2	RDQ36	C (LVDS)*
J18	PR38A	2	RDQ36	T (LVDS)*
J20	PR37B	2	RDQ36	C
J19	PR37A	2	RDQ36	T
VCCIO	VCCIO2	2		
H16	PR36B	2	RDQ36	C (LVDS)*
H17	PR36A	2	RDQS36	T (LVDS)*
G22	PR35B	2	RDQ36	C
GNDIO	GNDIO2	-		
G21	PR35A	2	RDQ36	T
H20	PR34B	2	RDQ36	C (LVDS)*
H19	PR34A	2	RDQ36	T (LVDS)*
G16	PR33B	2	RUM3_SPLLC_FB_A/RDQ36	C
VCCIO	VCCIO2	2		
H18	PR33A	2	RUM3_SPLLT_FB_A/RDQ36	T
F22	PR32B	2	RUM3_SPLLC_IN_A/RDQ36	C (LVDS)*
F21	PR32A	2	RUM3_SPLLT_IN_A/RDQ36	T (LVDS)*
G20	PR30B	2	RDQ27	C

LFE2M50E/SE Logic Signal Connections: 484 fpBGA (Cont.)

LFE2M50E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
F20	PR30A	2	RDQ27	T
GNDIO	GNDIO2	-		
G17	PR29B	2	RDQ27	C (LVDS)*
F17	PR29A	2	RDQ27	T (LVDS)*
VCCIO	VCCIO2	2		
GNDIO	GNDIO2	-		
E22	PR14B	2		C
D22	PR14A	2		T
VCCIO	VCCIO2	-		
E20	PR13B	2		C (LVDS)*
D20	PR13A	2		T (LVDS)*
D19	PR12B	2	RUM0_SPLLC_FB_A	C
GNDIO	GNDIO2	-		
E19	PR12A	2	RUM0_SPLLT_FB_A	T
F18	PR11B	2	RUM0_SPLLC_IN_A	C (LVDS)*
F19	PR11A	2	RUM0_SPLLT_IN_A	T (LVDS)*
VCCIO	VCCIO2	-		
E18	PR9B	2	VREF2_2	C
GNDIO	GNDIO2	-		
D18	PR9A	2	VREF1_2	T
VCCIO	VCCIO2	2		
F16	XRES	-		
C22	URC_SQ_VCCRX0	12		
A21	URC_SQ_HDINP0	12		T
B22	URC_SQ_VCCIB0	12		
B21	URC_SQ_HDINN0	12		C
C19	URC_SQ_VCCTX0	12		
A18	URC_SQ_HDOUTP0	12		T
A19	URC_SQ_VCCOB0	12		
B18	URC_SQ_HDOUTN0	12		C
C18	URC_SQ_VCCTX1	12		
B17	URC_SQ_HDOUTN1	12		C
C17	URC_SQ_VCCOB1	12		
A17	URC_SQ_HDOUTP1	12		T
C21	URC_SQ_VCCRX1	12		
B20	URC_SQ_HDINN1	12		C
C20	URC_SQ_VCCIB1	12		
A20	URC_SQ_HDINP1	12		T
B16	URC_SQ_VCCAUX33	12		
E17	URC_SQ_REFCLKN	12		C
D17	URC_SQ_REFCLKP	12		T
C16	URC_SQ_VCCP	12		
A12	URC_SQ_HDINP2	12		T

LFE2M50E/SE Logic Signal Connections: 484 fpBGA (Cont.)

LFE2M50E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
C12	URC_SQ_VCCIB2	12		
B12	URC_SQ_HDINN2	12		C
C11	URC_SQ_VCCRX2	12		
A15	URC_SQ_HDOUTP2	12		T
C15	URC_SQ_VCCOB2	12		
B15	URC_SQ_HDOUTN2	12		C
C14	URC_SQ_VCCTX2	12		
B14	URC_SQ_HDOUTN3	12		C
A13	URC_SQ_VCCOB3	12		
A14	URC_SQ_HDOUTP3	12		T
C13	URC_SQ_VCCTX3	12		
B11	URC_SQ_HDINN3	12		C
B10	URC_SQ_VCCIB3	12		
A11	URC_SQ_HDINP3	12		T
C10	URC_SQ_VCCRX3	12		
GNDIO	GNDIO1	-		
VCCIO	VCCIO1	1		
E13	PT55B	1		C
D12	PT55A	1		T
GNDIO	GNDIO1	-		
A9	PT54B	1		C
A8	PT54A	1		T
A7	PT53B	1		C
A6	PT53A	1		T
VCCIO	VCCIO1	1		
E12	PT52B	1		C
F12	PT52A	1		T
A5	PT51B	1		C
A4	PT51A	1		T
GNDIO	GNDIO1	-		
B7	PT50B	1		C
B8	PT50A	1		T
G11	PT49B	1		C
E11	PT49A	1		T
VCCIO	VCCIO1	1		
D11	PT48B	1	VREF2_1	C
D10	PT48A	1	VREF1_1	T
G10	PT47B	1	PCLKC1_0	C
F11	PT47A	1	PCLKT1_0	T
G9	PT46B	0	PCLKC0_0	C
GNDIO	GNDIO0	-		
F9	PT46A	0	PCLKT0_0	T
C9	PT45B	0	VREF2_0	C

LFE2M50E/SE Logic Signal Connections: 484 fpBGA (Cont.)

LFE2M50E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
D9	PT45A	0	VREF1_0	T
A2	PT44B	0		C
VCCIO	VCCIO0	0		
A3	PT44A	0		T
B3	PT43B	0		C
C4	PT43A	0		T
E10	PT42B	0		C
F10	PT42A	0		T
C7	PT41B	0		C
GNDIO	GNDIO0	-		
B6	PT41A	0		T
C6	PT40B	0		C
VCCIO	VCCIO0	0		
C5	PT40A	0		T
C8	PT39B	0		C
D8	PT39A	0		T
E8	PT38B	0		C
E9	PT38A	0		T
GNDIO	GNDIO0	-		
VCCIO	VCCIO0	0		
F8	PT10B	0		C
GNDIO	GNDIO0	-		
G8	PT10A	0		T
F7	PT9B	0		C
G7	PT9A	0		T
C3	PT8B	0		C
VCCIO	VCCIO0	0		
D4	PT8A	0		T
F6	PT7B	0		C
E6	PT7A	0		T
E5	PT6B	0		C
D6	PT6A	0		T
D3	PT5B	0		C
GNDIO	GNDIO0	-		
E3	PT5A	0		T
D5	PT4B	0		C
VCCIO	VCCIO0	0		
E4	PT4A	0		T
C2	PT3B	0		C
B2	PT3A	0		T
B1	PT2B	0		C
C1	PT2A	0		T
J10	VCC	-		

LFE2M50E/SE Logic Signal Connections: 484 fpBGA (Cont.)

LFE2M50E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
J11	VCC	-		
J12	VCC	-		
J13	VCC	-		
K14	VCC	-		
K9	VCC	-		
L14	VCC	-		
L9	VCC	-		
M14	VCC	-		
M9	VCC	-		
N14	VCC	-		
N9	VCC	-		
P10	VCC	-		
P11	VCC	-		
P12	VCC	-		
P13	VCC	-		
B5	VCCIO0	0		
B9	VCCIO0	0		
E7	VCCIO0	0		
H9	VCCIO0	0		
D13	VCCIO1	1		
E16	VCCIO1	1		
H14	VCCIO1	1		
E21	VCCIO2	2		
G18	VCCIO2	2		
J15	VCCIO2	2		
K19	VCCIO2	2		
N19	VCCIO3	3		
P15	VCCIO3	3		
T18	VCCIO3	3		
V21	VCCIO3	3		
AA18	VCCIO4	4		
R14	VCCIO4	4		
V16	VCCIO4	4		
W13	VCCIO4	4		
AA5	VCCIO5	5		
R9	VCCIO5	5		
V7	VCCIO5	5		
W10	VCCIO5	5		
N4	VCCIO6	6		
P8	VCCIO6	6		
T5	VCCIO6	6		
V2	VCCIO6	6		
E2	VCCIO7	7		

LFE2M50E/SE Logic Signal Connections: 484 fpBGA (Cont.)

LFE2M50E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
G5	VCCIO7	7		
J8	VCCIO7	7		
K4	VCCIO7	7		
AA22	VCCIO8	8		
U19	VCCIO8	8		
H11	VCCAUX	-		
H12	VCCAUX	-		
L15	VCCAUX	-		
L8	VCCAUX	-		
M15	VCCAUX	-		
M8	VCCAUX	-		
R11	VCCAUX	-		
R12	VCCAUX	-		
A1	GND	-		
A10	GND	-		
A16	GND	-		
A22	GND	-		
AA19	GND	-		
AA4	GND	-		
AB1	GND	-		
AB22	GND	-		
B13	GND	-		
B19	GND	-		
B4	GND	-		
D16	GND	-		
D2	GND	-		
D21	GND	-		
D7	GND	-		
G19	GND	-		
G4	GND	-		
H10	GND	-		
H13	GND	-		
J14	GND	-		
J9	GND	-		
K10	GND	-		
K11	GND	-		
K12	GND	-		
K13	GND	-		
K15	GND	-		
K20	GND	-		
K3	GND	-		
K8	GND	-		
L10	GND	-		

LFE2M50E/SE Logic Signal Connections: 484 fpBGA (Cont.)

LFE2M50E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
L11	GND	-		
L12	GND	-		
L13	GND	-		
M10	GND	-		
M11	GND	-		
M12	GND	-		
M13	GND	-		
N10	GND	-		
N11	GND	-		
N12	GND	-		
N13	GND	-		
N15	GND	-		
N20	GND	-		
N3	GND	-		
N8	GND	-		
P14	GND	-		
P9	GND	-		
R10	GND	-		
R13	GND	-		
T19	GND	-		
T4	GND	-		
W16	GND	-		
W2	GND	-		
W21	GND	-		
W7	GND	-		
Y10	GND	-		
Y13	GND	-		
Y15	NC	-		
W15	NC	-		
AB20	NC	-		
AB21	NC	-		
AA21	NC	-		
AA20	NC	-		
AB19	NC	-		
AB18	NC	-		
Y22	NC	-		
Y21	NC	-		
Y17	NC	-		
Y18	NC	-		
Y16	NC	-		
W17	NC	-		
Y19	NC	-		
Y20	NC	-		

LFE2M50E/SE Logic Signal Connections: 484 fpBGA (Cont.)

LFE2M50E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
W19	NC	-		
W18	NC	-		
V17	NC	-		
V18	NC	-		
D15	NC	-		
G14	NC	-		
G15	NC	-		
D14	NC	-		
E15	NC	-		
E14	NC	-		
F15	NC	-		
F14	NC	-		
F13	NC	-		
G12	NC	-		
G13	NC	-		
H8	VCCPLL	-		
H15	VCCPLL	-		
R8	VCCPLL	-		
R15	VCCPLL	-		

* Supports true LVDS. Other differential signals must be emulated with external resistors.
** These dedicated input pins can be used for GPLLs or GDLLs within the respective quadrant.
***For density migration, board design must take into account that these sysCONFIG pins are dual function for the lower density devices (ECP2M20 and ECP2M35). They can be either sysCONFIG pins or general purpose I/Os. These pins are dedicated pins for the higher density devices (ECP2M50, ECP2M70 and ECP2M100).
****Due to packaging bond out option, this DQS does not have all the necessary DQ pins bonded out for a full 8-bit data width.

Note: VCCIO and GND pads are used to determine the average DC current drawn by I/Os between GND/VCCIO connections, or between the last GND/VCCIO in an I/O bank and the end of an I/O bank. The substrate pads listed in the Pin Table do not necessarily have a one to one connection with a package ball or pin.

LFE2M35E/SE and LFE2M50E/SE Logic Signal Connections: 672 fpBGA

LFE2M35E/SE					LFE2M50E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
C2	PL2A	7	LDQ6	T (LVDS)*	PL2A	7	LDQ6	T*
C1	PL2B	7	LDQ6	C (LVDS)*	PL2B	7	LDQ6	C*
F6	PL3A	7	LDQ6	T	PL3A	7	LDQ6	T
H9	PL3B	7	LDQ6	C	PL3B	7	LDQ6	C
D3	PL4A	7	LDQ6	T (LVDS)*	PL4A	7	LDQ6	T*
VCCIO	VCCIO7	7			VCCIO7	7		
D2	PL4B	7	LDQ6	C (LVDS)*	PL4B	7	LDQ6	C*
F5	PL5A	7	LDQ6	T	PL5A	7	LDQ6	T
H8	PL5B	7	LDQ6	C	PL5B	7	LDQ6	C
E3	PL6A	7	LDQS6	T (LVDS)*	PL6A	7	LDQS6	T*
GNDIO	GNDIO7	-			GNDIO7	-		
E2	PL6B	7	LDQ6	C (LVDS)*	PL6B	7	LDQ6	C*
J9	PL7A	7	LDQ6	T	PL7A	7	LDQ6	T
E4	PL7B	7	LDQ6	C	PL7B	7	LDQ6	C
VCCIO	VCCIO7	7			VCCIO7	7		
E1	PL8A	7	LDQ6	T (LVDS)*	PL8A	7	LDQ6	T*
D1	PL8B	7	LDQ6	C (LVDS)*	PL8B	7	LDQ6	C*
J8	PL9A	7	VREF2_7/LDQ6	T	PL9A	7	VREF2_7/LDQ6	T
F4	PL9B	7	VREF1_7/LDQ6	C	PL9B	7	VREF1_7/LDQ6	C
GNDIO	GNDIO7	-			GNDIO7	-		
-	-	-			VCCIO7	7		
F3	PL11A	7	LUM0_SPLLT_IN_A/LDQ15	T (LVDS)*	PL11A	7	LUM0_SPLLT_IN_A	T*
F1	PL11B	7	LUM0_SPLLC_IN_A/LDQ15	C (LVDS)*	PL11B	7	LUM0_SPLLC_IN_A	C*
G6	PL12A	7	LUM0_SPLLT_FB_A/LDQ15	T	PL12A	7	LUM0_SPLLT_FB_A	T
K9	PL12B	7	LUM0_SPLLC_FB_A/LDQ15	C	PL12B	7	LUM0_SPLLC_FB_A	C
-	-	-			GNDIO7	-		
G5	PL13A	7	LDQ15	T (LVDS)*	PL13A	7		T*
VCCIO	VCCIO7	7			-	-		
G4	PL13B	7	LDQ15	C (LVDS)*	PL13B	7		C*
H5	PL14A	7	LDQ15	T	PL14A	7		T
-	-	-			VCCIO7	7		
H6	PL14B	7	LDQ15	C	PL14B	7		C
GNDIO	GNDIO7	-			GNDIO7	-		
J7	PL16A	7	LDQ15	T	PL19A	7		T
H4	PL16B	7	LDQ15	C	PL19B	7		C
H3	PL17A	7	LDQ15	T (LVDS)*	PL20A	7		T*
VCCIO	VCCIO7	7			VCCIO7	7		
G3	PL17B	7	LDQ15	C (LVDS)*	PL20B	7		C*
GNDIO	GNDIO7	-			GNDIO7	-		
G1	PL19A	7	LDQ23	T (LVDS)*	PL23A	7	LDQ27	T*
H1	PL19B	7	LDQ23	C (LVDS)*	PL23B	7	LDQ27	C*
J3	PL20A	7	LDQ23	T	PL24A	7	LDQ27	T
J4	PL20B	7	LDQ23	C	PL24B	7	LDQ27	C
VCCIO	VCCIO7	7			VCCIO7	7		
H2	PL21A	7	LDQ23	T (LVDS)*	PL25A	7	LDQ27	T*
J2	PL21B	7	LDQ23	C (LVDS)*	PL25B	7	LDQ27	C*
K7	PL22A	7	LDQ23	T	PL26A	7	LDQ27	T
J6	PL22B	7	LDQ23	C	PL26B	7	LDQ27	C
GNDIO	GNDIO7	-			GNDIO7	-		

LFE2M35E/SE and LFE2M50E/SE Logic Signal Connections: 672 fpBGA

LFE2M35E/SE					LFE2M50E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
K5	PL23A	7	LDQS23	T (LVDS)*	PL27A	7	LDQS27	T*
L5	PL23B	7	LDQ23	C (LVDS)*	PL27B	7	LDQ27	C*
K4	PL24A	7	LDQ23	T	PL28A	7	LDQ27	T
VCCIO	VCCIO7	7			VCCIO7	7		
L4	PL24B	7	LDQ23	C	PL28B	7	LDQ27	C
K3	PL25A	7	LDQ23	T (LVDS)*	PL29A	7	LDQ27	T*
L3	PL25B	7	LDQ23	C (LVDS)*	PL29B	7	LDQ27	C*
J1	PL26A	7	LDQ23	T	PL30A	7	LDQ27	T
GNDIO	GNDIO7	-			GNDIO7	-		
K2	PL26B	7	LDQ23	C	PL30B	7	LDQ27	C
K1	PL28A	7	LUM1_SPLLT_IN_A/LDQ32	T (LVDS)*	PL32A	7	LUM3_SPLLT_IN_A/LDQ36	T*
L1	PL28B	7	LUM1_SPLLC_IN_A/LDQ32	C (LVDS)*	PL32B	7	LUM3_SPLLC_IN_A/LDQ36	C*
K8	PL29A	7	LUM1_SPLLT_FB_A/LDQ32	T	PL33A	7	LUM3_SPLLT_FB_A/LDQ36	T
M5	PL29B	7	LUM1_SPLLC_FB_A/LDQ32	C	PL33B	7	LUM3_SPLLC_FB_A/LDQ36	C
VCCIO	VCCIO7	7			VCCIO7	7		
M4	PL30A	7	LDQ32	T (LVDS)*	PL34A	7	LDQ36	T*
M3	PL30B	7	LDQ32	C (LVDS)*	PL34B	7	LDQ36	C*
L8	PL31A	7	LDQ32	T	PL35A	7	LDQ36	T
M6	PL31B	7	LDQ32	C	PL35B	7	LDQ36	C
GNDIO	GNDIO7	-			GNDIO7	-		
M1	PL32A	7	LDQS32	T (LVDS)*	PL36A	7	LDQS36	T*
N1	PL32B	7	LDQ32	C (LVDS)*	PL36B	7	LDQ36	C*
N3	PL33A	7	LDQ32	T	PL37A	7	LDQ36	T
VCCIO	VCCIO7	7			VCCIO7	7		
N2	PL33B	7	LDQ32	C	PL37B	7	LDQ36	C
N5	PL34A	7	LDQ32	T (LVDS)*	PL38A	7	LDQ36	T*
N4	PL34B	7	LDQ32	C (LVDS)*	PL38B	7	LDQ36	C*
M7	PL35A	7	PCLKT7_0/LDQ32	T	PL39A	7	PCLKT7_0/LDQ36	T
GNDIO	GNDIO7	-			GNDIO7	-		
M8	PL35B	7	PCLKC7_0/LDQ32	C	PL39B	7	PCLKC7_0/LDQ36	C
P3	PL37A	6	PCLKT6_0	T (LVDS)*	PL41A	6	PCLKT6_0	T*
P2	PL37B	6	PCLKC6_0	C (LVDS)*	PL41B	6	PCLKC6_0	C*
P5	PL38A	6	VREF2_6	T	PL42A	6	VREF2_6	T
N6	PL38B	6	VREF1_6	C	PL42B	6	VREF1_6	C
P4	PL39A	6		T (LVDS)*	PL43A	6		T*
VCCIO	VCCIO6	6			VCCIO6	6		
R3	PL39B	6		C (LVDS)*	PL43B	6		C*
P6	PL40A	6		T	PL44A	6		T
N7	NC	-			PL44B	6		C
P1	PL41A	6	LLM2_SPLLT_IN_A	T (LVDS)*	PL45A	6	LLM3_SPLLT_IN_A	T*
GNDIO	GNDIO6	-			GNDIO6	-		
R1	PL41B	6	LLM2_SPLLC_IN_A	C (LVDS)*	PL45B	6	LLM3_SPLLC_IN_A	C*
N8	PL42A	6	LLM2_SPLLT_FB_A	T	PL46A	6	LLM3_SPLLT_FB_A	T
R5	PL42B	6	LLM2_SPLLC_FB_A	C	PL46B	6	LLM3_SPLLC_FB_A	C
VCCIO	VCCIO6	6			VCCIO6	6		
T3	PL44A	6	LDQ48	T (LVDS)*	PL48A	6	LDQ52	T*
T4	PL44B	6	LDQ48	C (LVDS)*	PL48B	6	LDQ52	C*
P8	PL45A	6	LDQ48	T	PL49A	6	LDQ52	T
R6	PL45B	6	LDQ48	C	PL49B	6	LDQ52	C

LFE2M35E/SE and LFE2M50E/SE Logic Signal Connections: 672 fpBGA

LFE2M35E/SE					LFE2M50E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
VCCIO	VCCIO6	6			VCCIO6	6		
T1	PL46A	6	LDQ48	T (LVDS)*	PL50A	6	LDQ52	T*
U1	PL46B	6	LDQ48	C (LVDS)*	PL50B	6	LDQ52	C*
R7	PL47A	6	LDQ48	T	PL51A	6	LDQ52	T
T5	PL47B	6	LDQ48	C	PL51B	6	LDQ52	C
GNDIO	GNDIO6	-			GNDIO6	-		
U3	PL48A	6	LDQS48	T (LVDS)*	PL52A	6	LDQS52	T*
U4	PL48B	6	LDQ48	C (LVDS)*	PL52B	6	LDQ52	C*
U5	PL49A	6	LDQ48	T	PL53A	6	LDQ52	T
VCCIO	VCCIO6	6			VCCIO6	6		
U6	PL49B	6	LDQ48	C	PL53B	6	LDQ52	C
U2	PL50A	6	LDQ48	T (LVDS)*	PL54A	6	LDQ52	T*
V1	PL50B	6	LDQ48	C (LVDS)*	PL54B	6	LDQ52	C*
W2	PL51A	6	LDQ48	T	PL55A	6	LDQ52	T
GNDIO	GNDIO6	-			GNDIO6	-		
V2	PL51B	6	LDQ48	C	PL55B	6	LDQ52	C
V4	PL55A	6	LDQ57	T (LVDS)*	PL59A	6		T*
VCCIO	VCCIO6	6			VCCIO6	6		
V3	PL55B	6	LDQ57	C (LVDS)*	PL59B	6		C*
-	-	-			GNDIO6	-		
W4	PL57A	6	LLM0_GPLLT_IN_A**/LDQS57****	T (LVDS)*	PL62A	6	LLM0_GPLLT_IN_A	T*
GNDIO	GNDIO6	-			GNDIO6	-		
W3	PL57B	6	LLM0_GPLLC_IN_A**/LDQ57	C (LVDS)*	PL62B	6	LLM0_GPLLC_IN_A	C*
W1	PL58A	6	LLM0_GPLLT_FB_A/LDQ57	T	PL63A	6	LLM0_GPLLT_FB_A	T
Y1	PL58B	6	LLM0_GPLLC_FB_A/LDQ57	C	PL63B	6	LLM0_GPLLC_FB_A	C
VCCIO	VCCIO6	6			VCCIO6	6		
AA1	PL59A	6	LLM0_GDLLT_IN_A**/LDQ57	T (LVDS)*	PL64A	6	LLM0_GDLLT_IN_A	T*
AB1	PL59B	6	LLM0_GDLLC_IN_A**/LDQ57	C (LVDS)*	PL64B	6	LLM0_GDLLC_IN_A	C*
U7	PL60A	6	LLM0_GDLLT_FB_A/LDQ57	T	PL65A	6	LLM0_GDLLT_FB_A	T
V6	PL60B	6	LLM0_GDLLC_FB_A/LDQ57	C	PL65B	6	LLM0_GDLLC_FB_A	C
GNDIO	GNDIO6	-			GNDIO6	-		
T8	LLM0_PLLCAP	6			LLM0_PLLCAP	6		
W5	PL62A	6	LDQ66	T (LVDS)*	PL67A	6	LDQ71	T*
Y4	PL62B	6	LDQ66	C (LVDS)*	PL67B	6	LDQ71	C*
U8	PL63A	6	LDQ66	T	PL68A	6	LDQ71	T
W6	PL63B	6	LDQ66	C	PL68B	6	LDQ71	C
VCCIO	VCCIO6	6			VCCIO6	6		
Y3	PL64A	6	LDQ66	T (LVDS)*	PL69A	6	LDQ71	T*
AA3	PL64B	6	LDQ66	C (LVDS)*	PL69B	6	LDQ71	C*
V7	NC	-			PL70A	6	LDQ71	T
Y5	PL65B	6	LDQ66	C	PL70B	6	LDQ71	C
GNDIO	GNDIO6	-			GNDIO6	-		
AB2	PL66A	6	LDQS66	T (LVDS)*	PL71A	6	LDQS71	T*
AA4	PL66B	6	LDQ66	C (LVDS)*	PL71B	6	LDQ71	C*
Y6	PL67A	6	LDQ66	T	PL72A	6	LDQ71	T
VCCIO	VCCIO6	6			VCCIO6	6		
U9	PL67B	6	LDQ66	C	PL72B	6	LDQ71	C
AA5	PL68A	6	LDQ66	T (LVDS)*	PL73A	6	LDQ71	T*
AA6	PL68B	6	LDQ66	C (LVDS)*	PL73B	6	LDQ71	C*

LFE2M35E/SE and LFE2M50E/SE Logic Signal Connections: 672 fpBGA

LFE2M35E/SE					LFE2M50E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
Y7	PL69A	6	LDQ66	T	PL74A	6	LDQ71	T
GNDIO	GNDIO6	-			GNDIO6	-		
V9	PL69B	6	LDQ66	C	PL74B	6	LDQ71	C
AC3	TCK	-			TCK	-		
W8	TDI	-			TDI	-		
AC4	TMS	-			TMS	-		
V8	TDO	-			TDO	-		
AA7	VCCJ	-			VCCJ	-		
AB6	PB2A	5	BDQ6	T	PB2A	5	BDQ6	T
Y8	PB2B	5	BDQ6	C	PB2B	5	BDQ6	C
AD1	PB3A	5	BDQ6	T	PB3A	5	BDQ6	T
AD2	PB3B	5	BDQ6	C	PB3B	5	BDQ6	C
AC5	PB4A	5	BDQ6	T	PB4A	5	BDQ6	T
AA8	PB4B	5	BDQ6	C	PB4B	5	BDQ6	C
VCCIO	VCCIO5	5			VCCIO5	5		
AC6	PB5A	5	BDQ6	T	PB5A	5	BDQ6	T
W9	PB5B	5	BDQ6	C	PB5B	5	BDQ6	C
AB7	PB6A	5	BDQS6	T	PB6A	5	BDQS6	T
GNDIO	GNDIO5	-			GNDIO5	-		
Y9	PB6B	5	BDQ6	C	PB6B	5	BDQ6	C
AD3	PB7A	5	BDQ6	T	PB7A	5	BDQ6	T
AD4	PB7B	5	BDQ6	C	PB7B	5	BDQ6	C
AA9	PB8A	5	BDQ6	T	PB8A	5	BDQ6	T
W10	PB8B	5	BDQ6	C	PB8B	5	BDQ6	C
VCCIO	VCCIO5	5			VCCIO5	5		
AC7	PB9A	5	BDQ6	T	PB9A	5	BDQ6	T
Y10	PB9B	5	BDQ6	C	PB9B	5	BDQ6	C
AE2	PB10A	5	BDQ6	T	PB10A	5	BDQ6	T
AD5	PB10B	5	BDQ6	C	PB10B	5	BDQ6	C
GNDIO	GNDIO5	-			GNDIO5	-		
AE4	PB11A	5	BDQ15	T	PB11A	5	BDQ15	T
AE3	PB11B	5	BDQ15	C	PB11B	5	BDQ15	C
W11	PB12A	5	BDQ15	T	PB12A	5	BDQ15	T
AB8	PB12B	5	BDQ15	C	PB12B	5	BDQ15	C
AE5	PB13A	5	BDQ15	T	PB13A	5	BDQ15	T
AD6	PB13B	5	BDQ15	C	PB13B	5	BDQ15	C
VCCIO	VCCIO5	5			VCCIO5	5		
AA10	PB14A	5	BDQ15	T	PB14A	5	BDQ15	T
AC8	PB14B	5	BDQ15	C	PB14B	5	BDQ15	C
W12	PB15A	5	BDQS15	T	PB15A	5	BDQS15	T
GNDIO	GNDIO5	-			GNDIO5	-		
AC9	PB15B	5	BDQ15	C	PB15B	5	BDQ15	C
W13	PB16A	5	BDQ15	T	PB16A	5	BDQ15	T
AB10	PB16B	5	BDQ15	C	PB16B	5	BDQ15	C
AF3	PB17A	5	BDQ15	T	PB17A	5	BDQ15	T
AF4	PB17B	5	BDQ15	C	PB17B	5	BDQ15	C
VCCIO	VCCIO5	5			VCCIO5	5		
AF5	PB18A	5	BDQ15	T	PB18A	5	BDQ15	T
AF6	PB18B	5	BDQ15	C	PB18B	5	BDQ15	C

LFE2M35E/SE and LFE2M50E/SE Logic Signal Connections: 672 fpBGA

LFE2M35E/SE					LFE2M50E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
Y12	PB19A	5	BDQ15	T	PB19A	5	BDQ15	T
GNDIO	GNDIO5	-			GNDIO5	-		
AB11	PB19B	5	BDQ15	C	PB19B	5	BDQ15	C
-	-	-			VCCIO5	5		
-	-	-			GNDIO5	-		
AD7	PB20A	5	BDQ24	T	PB29A	5	BDQ33	T
AF7	PB20B	5	BDQ24	C	PB29B	5	BDQ33	C
AD8	PB21A	5	BDQ24	T	PB30A	5	BDQ33	T
AA12	PB21B	5	BDQ24	C	PB30B	5	BDQ33	C
AE8	PB22A	5	BDQ24	T	PB31A	5	BDQ33	T
VCCIO	VCCIO5	5			VCCIO5	5		
AF8	PB22B	5	BDQ24	C	PB31B	5	BDQ33	C
AD9	PB23A	5	BDQ24	T	PB32A	5	BDQ33	T
AC10	PB23B	5	BDQ24	C	PB32B	5	BDQ33	C
AC11	PB24A	5	BDQS24	T	PB33A	5	BDQS33	T
GNDIO	GNDIO5	-			GNDIO5	-		
AB12	PB24B	5	BDQ24	C	PB33B	5	BDQ33	C
AD10	PB25A	5	BDQ24	T	PB34A	5	BDQ33	T
Y13	PB25B	5	BDQ24	C	PB34B	5	BDQ33	C
AF9	PB26A	5	BDQ24	T	PB35A	5	BDQ33	T
VCCIO	VCCIO5	5			VCCIO5	5		
AE9	PB26B	5	BDQ24	C	PB35B	5	BDQ33	C
AF10	PB27A	5	BDQ24	T	PB36A	5	BDQ33	T
AE10	PB27B	5	BDQ24	C	PB36B	5	BDQ33	C
AD11	PB28A	5	BDQ24	T	PB37A	5	BDQ33	T
GNDIO	GNDIO5	-			GNDIO5	-		
AF11	PB28B	5	BDQ24	C	PB37B	5	BDQ33	C
VCCIO	VCCIO5	5			VCCIO5	5		
GNDIO	GNDIO5	-			GNDIO5	-		
AA13	PB33A	5	BDQS33****	T	PB42A	5	BDQS42****	T
AB13	PB33B	5	BDQ33	C	PB42B	5	BDQ42	C
W14	PB34A	5	VREF2_5/BDQ33	T	PB43A	5	VREF2_5/BDQ42	T
AC12	PB34B	5	VREF1_5/BDQ33	C	PB43B	5	VREF1_5/BDQ42	C
AF12	PB35A	5	PCLKT5_0/BDQ33	T	PB44A	5	PCLKT5_0/BDQ42	T
AD12	PB35B	5	PCLKC5_0/BDQ33	C	PB44B	5	PCLKC5_0/BDQ42	C
VCCIO	VCCIO5	5			VCCIO5	5		
GNDIO	GNDIO5	-			GNDIO5	-		
AC13	PB40A	4	PCLKT4_0/BDQ42	T	PB49A	4	PCLKT4_0/BDQ51	T
VCCIO	VCCIO4	4			VCCIO4	4		
Y14	PB40B	4	PCLKC4_0/BDQ42	C	PB49B	4	PCLKC4_0/BDQ51	C
AB20	PB57A	4	BDQ60	T	PB50A	4	VREF2_4/BDQ51	T
AC14	PB41B	4	VREF1_4/BDQ42	C	PB50B	4	VREF1_4/BDQ51	C
AB14	PB42A	4	BDQS42****	T	PB51A	4	BDQS51****	T
GNDIO	GNDIO4	-			GNDIO4	-		
AA14	PB42B	4	BDQ42	C	PB51B	4	BDQ51	C
VCCIO	VCCIO4	4			VCCIO4	4		
GNDIO	GNDIO4	-			GNDIO4	-		
W17	PB65A	4	BDQ69	T	PB56A	4	BDQ60	T
AA19	PB65B	4	BDQ69	C	PB56B	4	BDQ60	C

LFE2M35E/SE and LFE2M50E/SE Logic Signal Connections: 672 fpBGA

LFE2M35E/SE					LFE2M50E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
AC15	PB48A	4	BDQ51	T	PB57A	4	BDQ60	T
Y18	PB68B	4	BDQ69	C	PB57B	4	BDQ60	C
AB15	PB49A	4	BDQ51	T	PB58A	4	BDQ60	T
AC16	PB49B	4	BDQ51	C	PB58B	4	BDQ60	C
VCCIO	VCCIO4	4			VCCIO4	4		
AA17	PB60A	4	BDQS60****	T	PB59A	4	BDQ60	T
AB16	PB50B	4	BDQ51	C	PB59B	4	BDQ60	C
GNDIO	GNDIO4	-			GNDIO4	-		
AA15	PB51A	4	BDQS51****	T	PB60A	4	BDQS60	T
W16	PB59B	4	BDQ60	C	PB60B	4	BDQ60	C
Y15	PB52A	4	BDQ51	T	PB61A	4	BDQ60	T
AC17	PB52B	4	BDQ51	C	PB61B	4	BDQ60	C
AA18	PB61A	4	BDQ60	T	PB62A	4	BDQ60	T
Y17	PB61B	4	BDQ60	C	PB62B	4	BDQ60	C
-	-	-			VCCIO4	4		
GNDIO	GNDIO4	-			-	-		
W15	PB54A	4	BDQ51	T	PB63A	4	BDQ60	T
AB17	PB54B	4	BDQ51	C	PB63B	4	BDQ60	C
GNDIO	GNDIO4	-			GNDIO4	-		
VCCIO	VCCIO4	4			VCCIO4	4		
V17	PB73A	4	BDQ69	T	PB72A	4	BDQ69	T
AA20	PB73B	4	BDQ69	C	PB72B	4	BDQ69	C
GNDIO	GNDIO4	-			GNDIO4	-		
AD13	VCC	-			LRC_SQ_VCCRX3	13		
AF14	PB47A	4	BDQ51	T	LRC_SQ_HDINP3	13		T
AE13	NC	-			LRC_SQ_VCCIB3	13		
AE14	PB41A	4	VREF2_4/BDQ42	T	LRC_SQ_HDINN3	13		C
AD16	VCC	-			LRC_SQ_VCCTX3	13		
AF17	PB51B	4	BDQ51	C	LRC_SQ_HDOUTP3	13		T
AF16	NC	-			LRC_SQ_VCCOB3	13		
AE17	PB50A	4	BDQ51	T	LRC_SQ_HDOUTN3	13		C
AD17	VCC	-			LRC_SQ_VCCTX2	13		
AE18	PB53B	4	BDQ51	C	LRC_SQ_HDOUTN2	13		C
AD18	NC	-			LRC_SQ_VCCOB2	13		
AF18	PB53A	4	BDQ51	T	LRC_SQ_HDOUTP2	13		T
AD14	VCC	-			LRC_SQ_VCCRX2	13		
AE15	PB48B	4	BDQ51	C	LRC_SQ_HDINN2	13		C
AD15	NC	-			LRC_SQ_VCCIB2	13		
AF15	PB47B	4	BDQ51	C	LRC_SQ_HDINP2	13		T
AD19	VCC	-			LRC_SQ_VCCP	13		
AC19	PB57B	4	BDQ60	C	LRC_SQ_REFCLKP	13		T
AB19	PB59A	4	BDQ60	T	LRC_SQ_REFCLKN	13		C
AE19	VCCAUX	-			LRC_SQ_VCCAUX33	13		
AF23	PB64A	4	BDQ60	T	LRC_SQ_HDINP1	13		T
AD23	NC	-			LRC_SQ_VCCIB1	13		
AE23	PB66B	4	BDQ69	C	LRC_SQ_HDINN1	13		C
AD24	VCC	-			LRC_SQ_VCCRX1	13		
AF20	PB55A	4	BDQ51	T	LRC_SQ_HDOUTP1	13		T
AD20	NC	-			LRC_SQ_VCCOB1	13		

LFE2M35E/SE and LFE2M50E/SE Logic Signal Connections: 672 fpBGA

LFE2M35E/SE					LFE2M50E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
AE20	PB55B	4	BDQ51	C	LRC_SQ_HDOUTN1	13		C
AD21	VCC	-			LRC_SQ_VCCTX1	13		
AE21	PB63B	4	BDQ60	C	LRC_SQ_HDOUTN0	13		C
AF22	NC	-			LRC_SQ_VCCOB0	13		
AF21	PB62A	4	BDQ60	T	LRC_SQ_HDOUTP0	13		T
AD22	VCC	-			LRC_SQ_VCCTX0	13		
AE24	PB67B	4	BDQ69	C	LRC_SQ_HDINN0	13		C
AE25	NC	-			LRC_SQ_VCCIB0	13		
AF24	PB67A	4	BDQ69	T	LRC_SQ_HDINP0	13		T
AD25	VCC	-			LRC_SQ_VCCRX0	13		
AA21	CFG2	8			CFG2	8		
AA22	CFG1	8			CFG1	8		
AB23	CFG0	8			CFG0	8		
AC26	PROGRAMN	8			PROGRAMN	8		
AB24	CCLK	8			CCLK	8		
AA23	INITN	8			INITN	8		
AB25	DONE	8			DONE	8		
GNDIO	GNDIO8	-			GNDIO8	-		
Y19	PR68B	8	WRITEN***	C	WRITEN***	8		
Y21	PR68A	8	CS1N***	T	CS1N***	8		
AB26	PR67B	8	CSN***	C	CSN***	8		
Y22	PR67A	8	D0/SPIFASTN***	T	D0/SPIFASTN***	8		
VCCIO	VCCIO8	8				8		
W19	PR66B	8	D1***	C	D1***	8		
Y20	PR66A	8	D2***	T	D2***	8		
W22	PR65B	8	D3***	C	D3***	8		
GNDIO	GNDIO8	-				-		
W18	PR65A	8	D4***	T	D4***	8		
Y23	PR64B	8	D5***	C	D5***	8		
AA24	PR64A	8	D6***	T	D6***	8		
W21	PR63B	8	D7/SPID0***	C	D7/SPID0***	8		
VCCIO	VCCIO8	8			VCCIO8	8		
V20	PR63A	8	DI/CSSPI0N***	T	DI/CSSPI0N***	8		
W23	PR62B	8	DOUT/CSON/CSSPI1N***	C	DOUT/CSON/ CSSPI1N***	8		
Y24	PR62A	8	BUSY/SISPI***	T	BUSY/SISPI***	8		
V19	RLM0_PLLCAP	3			RLM0_PLLCAP	3		
V21	PR60B	3	RLM0_GDLLC_FB_A	C	PR65B	3	RLM0_GDLLC_FB_A	C
GNDIO	GNDIO3	-			GNDIO3	-		
U19	PR60A	3	RLM0_GDLLT_FB_A/RDQ57	T	PR65A	3	RLM0_GDLLT_FB_A	T
AA26	PR59B	3	RLM0_GDLLC_IN_A**/RDQ57	C (LVDS)*	PR64B	3	RLM0_GDLLC_IN_A	C*
Y26	PR59A	3	RLM0_GDLLT_IN_A**/RDQ57	T (LVDS)*	PR64A	3	RLM0_GDLLT_IN_A	T*
V23	PR58B	3	RLM0_GPLLC_IN_A**/RDQ57	C	PR63B	3	RLM0_GPLLC_IN_A	C
VCCIO	VCCIO3	3			VCCIO3	3		
U20	PR58A	3	RLM0_GPLLT_IN_A**/RDQ57	T	PR63A	3	RLM0_GPLLT_IN_A	T
W24	PR57B	3	RLM0_GPLLC_FB_A/RDQ57	C (LVDS)*	PR62B	3	RLM0_GPLLC_FB_A	C*
V24	PR57A	3	RLM0_GPLLT_FB_A/RDQS57	T (LVDS)*	PR62A	3	RLM0_GPLLT_FB_A	T*
GNDIO	GNDIO3	-			GNDIO3	-		
U21	PR56A	3	RDQ57	T	PR60A	3		T
W25	PR55B	3	RDQ57	C (LVDS)*	PR59B	3		C*

LFE2M35E/SE and LFE2M50E/SE Logic Signal Connections: 672 fpBGA

LFE2M35E/SE					LFE2M50E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
W26	PR55A	3	RDQ57	T (LVDS)*	PR59A	3		T*
VCCIO	VCCIO3	3			VCCIO3	3		
U18	PR54B	3	RDQ57	C	PR58B	3		C
U22	PR54A	3	RDQ57	T	PR58A	3		T
V25	PR53B	3	RDQ57	C (LVDS)*	PR57B	3		C*
V26	PR53A	3	RDQ57	T (LVDS)*	PR57A	3		T*
U24	PR51B	3	RDQ48	C	PR55B	3	RDQ52	C
T24	PR51A	3	RDQ48	T	PR55A	3	RDQ52	T
GNDIO	GNDIO3	-			GNDIO3	-		
T22	PR50B	3	RDQ48	C (LVDS)*	PR54B	3	RDQ52	C*
T23	PR50A	3	RDQ48	T (LVDS)*	PR54A	3	RDQ52	T*
U25	PR49B	3	RDQ48	C	PR53B	3	RDQ52	C
U26	PR49A	3	RDQ48	T	PR53A	3	RDQ52	T
VCCIO	VCCIO3	3			VCCIO3	3		
T19	PR48B	3	RDQ48	C (LVDS)*	PR52B	3	RDQ52	C*
R19	PR48A	3	RDQS48	T (LVDS)*	PR52A	3	RDQS52	T*
R21	PR47B	3	RDQ48	C	PR51B	3	RDQ52	C
GNDIO	GNDIO3	-			GNDIO3	-		
R20	PR47A	3	RDQ48	T	PR51A	3	RDQ52	T
T26	PR46B	3	RDQ48	C (LVDS)*	PR50B	3	RDQ52	C*
R26	PR46A	3	RDQ48	T (LVDS)*	PR50A	3	RDQ52	T*
P21	PR45B	3	RDQ48	C	PR49B	3	RDQ52	C
VCCIO	VCCIO3	3			VCCIO3	3		
P19	PR45A	3	RDQ48	T	PR49A	3	RDQ52	T
R23	PR44B	3	RDQ48	C (LVDS)*	PR48B	3	RDQ52	C*
R24	PR44A	3	RDQ48	T (LVDS)*	PR48A	3	RDQ52	T*
-	-	-			GNDIO3	-		
R22	PR42B	3	RLM2_SPLLC_FB_A	C	PR46B	3	RLM3_SPLLC_FB_A	C
VCCIO	VCCIO3	3			VCCIO3	3		
N19	PR42A	3	RLM2_SPLLT_FB_A	T	PR46A	3	RLM3_SPLLT_FB_A	T
P23	PR41B	3	RLM2_SPLLC_IN_A	C (LVDS)*	PR45B	3	RLM3_SPLLC_IN_A	C*
P24	PR41A	3	RLM2_SPLLT_IN_A	T (LVDS)*	PR45A	3	RLM3_SPLLT_IN_A	T*
GNDIO	GNDIO3	-			GNDIO3	-		
N21	PR40B	3		C	PR44B	3		C
P22	PR40A	3		T	PR44A	3		T
N20	PR39B	3		C (LVDS)*	PR43B	3		C*
N22	PR39A	3		T (LVDS)*	PR43A	3		T*
VCCIO	VCCIO3	3			VCCIO3	3		
P25	PR38B	3	VREF2_3	C	PR42B	3	VREF2_3	C
P26	PR38A	3	VREF1_3	T	PR42A	3	VREF1_3	T
M21	PR37B	3	PCLKC3_0	C (LVDS)*	PR41B	3	PCLKC3_0	C*
N23	PR37A	3	PCLKT3_0	T (LVDS)*	PR41A	3	PCLKT3_0	T*
N24	PR35B	2	PCLKC2_0/RDQ32	C	PR39B	2	PCLKC2_0/RDQ36	C
N25	PR35A	2	PCLKT2_0/RDQ32	T	PR39A	2	PCLKT2_0/RDQ36	T
GNDIO	GNDIO2	-			GNDIO2	-		
M22	PR34B	2	RDQ32	C (LVDS)*	PR38B	2	RDQ36	C*
M24	PR34A	2	RDQ32	T (LVDS)*	PR38A	2	RDQ36	T*
M23	PR33B	2	RDQ32	C	PR37B	2	RDQ36	C
N26	PR33A	2	RDQ32	T	PR37A	2	RDQ36	T

LFE2M35E/SE and LFE2M50E/SE Logic Signal Connections: 672 fpBGA

LFE2M35E/SE					LFE2M50E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
VCCIO	VCCIO2	2			VCCIO2	2		
L22	PR32B	2	RDQ32	C (LVDS)*	PR36B	2	RDQ36	C*
L24	PR32A	2	RDQS32	T (LVDS)*	PR36A	2	RDQS36	T*
L23	PR31B	2	RDQ32	C	PR35B	2	RDQ36	C
GNDIO	GNDIO2	-			GNDIO2	-		
M20	PR31A	2	RDQ32	T	PR35A	2	RDQ36	T
M26	PR30B	2	RDQ32	C (LVDS)*	PR34B	2	RDQ36	C*
L26	PR30A	2	RDQ32	T (LVDS)*	PR34A	2	RDQ36	T*
K22	PR29B	2	RUM1_SPLLC_FB_A/RDQ32	C	PR33B	2	RUM3_SPLLC_FB_A/RDQ36	C
VCCIO	VCCIO2	2			VCCIO2	2		
M19	PR29A	2	RUM1_SPLLT_FB_A/RDQ32	T	PR33A	2	RUM3_SPLLT_FB_A/RDQ36	T
K25	PR28B	2	RUM1_SPLLC_IN_A/RDQ32	C (LVDS)*	PR32B	2	RUM3_SPLLC_IN_A/RDQ36	C*
K26	PR28A	2	RUM1_SPLLT_IN_A/RDQ32	T (LVDS)*	PR32A	2	RUM3_SPLLT_IN_A/RDQ36	T*
K24	PR26B	2	RDQ23	C	PR30B	2	RDQ27	C
K23	PR26A	2	RDQ23	T	PR30A	2	RDQ27	T
GNDIO	GNDIO2	-			GNDIO2	-		
L19	PR25B	2	RDQ23	C (LVDS)*	PR29B	2	RDQ27	C*
K21	PR25A	2	RDQ23	T (LVDS)*	PR29A	2	RDQ27	T*
J23	PR24B	2	RDQ23	C	PR28B	2	RDQ27	C
J24	PR24A	2	RDQ23	T	PR28A	2	RDQ27	T
VCCIO	VCCIO2	2			VCCIO2	2		
K20	PR23B	2	RDQ23	C (LVDS)*	PR27B	2	RDQ27	C*
J21	PR23A	2	RDQS23	T (LVDS)*	PR27A	2	RDQS27	T*
H21	PR22B	2	RDQ23	C	PR26B	2	RDQ27	C
GNDIO	GNDIO2	-			GNDIO2	-		
K18	PR22A	2	RDQ23	T	PR26A	2	RDQ27	T
H22	PR21B	2	RDQ23	C (LVDS)*	PR25B	2	RDQ27	C*
J20	PR21A	2	RDQ23	T (LVDS)*	PR25A	2	RDQ27	T*
J25	PR20B	2	RDQ23	C	PR24B	2	RDQ27	C
VCCIO	VCCIO2	2			VCCIO2	2		
J26	PR20A	2	RDQ23	T	PR24A	2	RDQ27	T
G21	PR19B	2	RDQ23	C (LVDS)*	PR23B	2	RDQ27	C*
J19	PR19A	2	RDQ23	T (LVDS)*	PR23A	2	RDQ27	T*
GNDIO	GNDIO2	-			GNDIO2	-		
H23	PR18B	2	RDQ15	C	PR21B	2		C
H24	PR18A	2	RDQ15	T	PR21A	2		T
H25	PR17B	2	RDQ15	C (LVDS)*	PR20B	2		C*
H26	PR17A	2	RDQ15	T (LVDS)*	PR20A	2		T*
VCCIO	VCCIO2	2			VCCIO2	2		
G22	PR16B	2	RDQ15	C	PR19B	2		C
K19	PR16A	2	RDQ15	T	PR19A	2		T
G24	PR15B	2	RDQ15	C (LVDS)*	PR18B	2		C*
G23	PR15A	2	RDQS15	T (LVDS)*	PR18A	2		T*
GNDIO	GNDIO2	-			GNDIO2	-		
J18	PR14B	2	RDQ15	C	PR14B	2		C
F22	PR14A	2	RDQ15	T	PR14A	2		T
-	-	-			VCCIO2	2		
F23	PR13B	2	RDQ15	C (LVDS)*	PR13B	2		C*
F24	PR13A	2	RDQ15	T (LVDS)*	PR13A	2		T*

LFE2M35E/SE and LFE2M50E/SE Logic Signal Connections: 672 fpBGA

LFE2M35E/SE					LFE2M50E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
VCCIO	VCCIO2	2			-	-		
H20	PR12B	2	RUM0_SPLLC_FB_A/RDQ15	C	PR12B	2	RUM0_SPLLC_FB_A	C
-	-	-			GNDIO2	-		
F21	PR12A	2	RUM0_SPLLT_FB_A/RDQ15	T	PR12A	2	RUM0_SPLLT_FB_A	T
G26	PR11B	2	RUM0_SPLLC_IN_A/RDQ15	C (LVDS)*	PR11B	2	RUM0_SPLLC_IN_A	C*
F26	PR11A	2	RUM0_SPLLT_IN_A/RDQ15	T (LVDS)*	PR11A	2	RUM0_SPLLT_IN_A	T*
-	-	-			VCCIO2	2		
E24	PR9B	2	VREF2_2	C	PR9B	2	VREF2_2	C
GNDIO	GNDIO2	-			GNDIO2	-		
E23	PR9A	2	VREF1_2	T	PR9A	2	VREF1_2	T
VCCIO	VCCIO4	4			VCCIO2	2		
H19	XRES	-			XRES	-		
C25	URC_SQ_VCCRX0	12			URC_SQ_VCCRX0	12		
A24	URC_SQ_HDINP0	12		T	URC_SQ_HDINP0	12		T
B25	URC_SQ_VCCIB0	12			URC_SQ_VCCIB0	12		
B24	URC_SQ_HDINN0	12		C	URC_SQ_HDINN0	12		C
C22	URC_SQ_VCCTX0	12			URC_SQ_VCCTX0	12		
A21	URC_SQ_HDOUTP0	12		T	URC_SQ_HDOUTP0	12		T
A22	URC_SQ_VCCOB0	12			URC_SQ_VCCOB0	12		
B21	URC_SQ_HDOUTN0	12		C	URC_SQ_HDOUTN0	12		C
C21	URC_SQ_VCCTX1	12			URC_SQ_VCCTX1	12		
B20	URC_SQ_HDOUTN1	12		C	URC_SQ_HDOUTN1	12		C
C20	URC_SQ_VCCOB1	12			URC_SQ_VCCOB1	12		
A20	URC_SQ_HDOUTP1	12		T	URC_SQ_HDOUTP1	12		T
C24	URC_SQ_VCCRX1	12			URC_SQ_VCCRX1	12		
B23	URC_SQ_HDINN1	12		C	URC_SQ_HDINN1	12		C
C23	URC_SQ_VCCIB1	12			URC_SQ_VCCIB1	12		
A23	URC_SQ_HDINP1	12		T	URC_SQ_HDINP1	12		T
B19	URC_SQ_VCCAUX33	12			URC_SQ_VCCAUX33	12		
E19	URC_SQ_REFCLKN	12		C	URC_SQ_REFCLKN	12		C
D19	URC_SQ_REFCLKP	12		T	URC_SQ_REFCLKP	12		T
C19	URC_SQ_VCCP	12			URC_SQ_VCCP	12		
A15	URC_SQ_HDINP2	12		T	URC_SQ_HDINP2	12		T
C15	URC_SQ_VCCIB2	12			URC_SQ_VCCIB2	12		
B15	URC_SQ_HDINN2	12		C	URC_SQ_HDINN2	12		C
C14	URC_SQ_VCCRX2	12			URC_SQ_VCCRX2	12		
A18	URC_SQ_HDOUTP2	12		T	URC_SQ_HDOUTP2	12		T
C18	URC_SQ_VCCOB2	12			URC_SQ_VCCOB2	12		
B18	URC_SQ_HDOUTN2	12		C	URC_SQ_HDOUTN2	12		C
C17	URC_SQ_VCCTX2	12			URC_SQ_VCCTX2	12		

LFE2M35E/SE and LFE2M50E/SE Logic Signal Connections: 672 fpBGA

LFE2M35E/SE					LFE2M50E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
B17	URC_SQ_HDOUTN3	12		C	URC_SQ_HDOUTN3	12		C
A16	URC_SQ_VCCOB3	12			URC_SQ_VCCOB3	12		
A17	URC_SQ_HDOUTP3	12		T	URC_SQ_HDOUTP3	12		T
C16	URC_SQ_VCCTX3	12			URC_SQ_VCCTX3	12		
B14	URC_SQ_HDINN3	12		C	URC_SQ_HDINN3	12		C
B13	URC_SQ_VCCIB3	12			URC_SQ_VCCIB3	12		
A14	URC_SQ_HDINP3	12		T	URC_SQ_HDINP3	12		T
C13	URC_SQ_VCCRX3	12			URC_SQ_VCCRX3	12		
-	-	-			GNDIO1	-		
-	-	-			VCCIO1	1		
E17	PT46B	1		C	PT55B	1		C
D17	PT46A	1		T	PT55A	1		T
GNDIO	GNDIO1	-			GNDIO1	-		
F17	PT45B	1		C	PT54B	1		C
D16	PT45A	1		T	PT54A	1		T
F19	PT44B	1		C	PT53B	1		C
F18	PT44A	1		T	PT53A	1		T
VCCIO	VCCIO1	1			VCCIO1	1		
E16	PT43B	1		C	PT52B	1		C
D15	PT43A	1		T	PT52A	1		T
G18	PT42B	1		C	PT51B	1		C
E15	PT42A	1		T	PT51A	1		T
GNDIO	GNDIO1	-			GNDIO1	-		
G17	PT41B	1		C	PT50B	1		C
E14	PT41A	1		T	PT50A	1		T
D14	PT40B	1		C	PT49B	1		C
D13	PT40A	1		T	PT49A	1		T
VCCIO	VCCIO1	1			VCCIO1	1		
F15	PT39B	1	VREF2_1	C	PT48B	1	VREF2_1	C
E12	PT39A	1	VREF1_1	T	PT48A	1	VREF1_1	T
H17	PT38B	1	PCLKC1_0	C	PT47B	1	PCLKC1_0	C
E13	PT38A	1	PCLKT1_0	T	PT47A	1	PCLKT1_0	T
C12	PT37B	0	PCLKC0_0	C	PT46B	0	PCLKC0_0	C
GNDIO	GNDIO0	-			GNDIO0	-		
G15	PT37A	0	PCLKT0_0	T	PT46A	0	PCLKT0_0	T
C11	PT36B	0	VREF2_0	C	PT45B	0	VREF2_0	C
F14	PT36A	0	VREF1_0	T	PT45A	0	VREF1_0	T
A12	PT35B	0		C	PT44B	0		C
VCCIO	VCCIO0	0			VCCIO0	0		
A11	PT35A	0		T	PT44A	0		T
D12	PT34B	0		C	PT43B	0		C
H16	PT34A	0		T	PT43A	0		T
H18	PT33B	0		C	PT42B	0		C
H15	PT33A	0		T	PT42A	0		T
A10	PT32B	0		C	PT41B	0		C
GNDIO	GNDIO0	-			GNDIO0	-		
B10	PT32A	0		T	PT41A	0		T

LFE2M35E/SE and LFE2M50E/SE Logic Signal Connections: 672 fpBGA

LFE2M35E/SE					LFE2M50E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
D11	PT31B	0		C	PT40B	0		C
VCCIO	VCCIO0	0			VCCIO0	0		
G14	PT31A	0		T	PT40A	0		T
E11	PT30B	0		C	PT39B	0		C
F13	PT30A	0		T	PT39A	0		T
D10	PT29B	0		C	PT38B	0		C
H14	PT29A	0		T	PT38A	0		T
GNDIO	GNDIO0	-			GNDIO0	-		
VCCIO	VCCIO0	0			VCCIO0	0		
A9	PT24B	0		C	PT24B	0		C
C10	PT23B	0		C	PT23B	0		C
GNDIO	GNDIO0	-			GNDIO0	-		
E8	PT23A	0		T	PT23A	0		T
B9	PT22B	0		C	PT22B	0		C
A8	PT22A	0		T	PT22A	0		T
VCCIO	VCCIO0	0			VCCIO0	0		
F12	PT21B	0		C	PT21B	0		C
E10	PT21A	0		T	PT21A	0		T
G13	PT20B	0		C	PT20B	0		C
C9	PT20A	0		T	PT20A	0		T
B8	PT19B	0		C	PT19B	0		C
GNDIO	GNDIO0	-			GNDIO0	-		
A7	PT19A	0		T	PT19A	0		T
D9	PT18B	0		C	PT18B	0		C
H13	PT18A	0		T	PT18A	0		T
D6	PT17B	0		C	PT17B	0		C
C7	PT17A	0		T	PT17A	0		T
VCCIO	VCCIO0	0			VCCIO0	0		
C8	PT16B	0		C	PT16B	0		C
G12	PT16A	0		T	PT16A	0		T
D8	PT15B	0		C	PT15B	0		C
H12	PT15A	0		T	PT15A	0		T
GNDIO	GNDIO0	-			GNDIO0	-		
A6	PT14B	0		C	PT14B	0		C
A5	PT14A	0		T	PT14A	0		T
A4	PT13B	0		C	PT13B	0		C
A3	PT13A	0		T	PT13A	0		T
VCCIO	VCCIO0	0			VCCIO0	0		
C6	PT12B	0		C	PT12B	0		C
F10	PT12A	0		T	PT12A	0		T
D7	PT11B	0		C	PT11B	0		C
H11	PT11A	0		T	PT11A	0		T
D5	PT10B	0		C	PT10B	0		C
GNDIO	GNDIO0	-			GNDIO0	-		
E6	PT10A	0		T	PT10A	0		T
G10	PT9B	0		C	PT9B	0		C
F9	PT9A	0		T	PT9A	0		T
H10	PT8B	0		C	PT8B	0		C
VCCIO	VCCIO0	0			VCCIO0	0		

LFE2M35E/SE and LFE2M50E/SE Logic Signal Connections: 672 fpBGA

LFE2M35E/SE					LFE2M50E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
E7	PT8A	0		T	PT8A	0		T
B3	PT7B	0		C	PT7B	0		C
C5	PT7A	0		T	PT7A	0		T
B2	PT6B	0		C	PT6B	0		C
C4	PT6A	0		T	PT6A	0		T
G9	PT5B	0		C	PT5B	0		C
GNDIO	GNDIO0	-			GNDIO0	-		
F7	PT5A	0		T	PT5A	0		T
C3	PT4B	0		C	PT4B	0		C
VCCIO	VCCIO0	0			VCCIO0	0		
D4	PT4A	0		T	PT4A	0		T
J10	PT3B	0		C	PT3B	0		C
F8	PT3A	0		T	PT3A	0		T
G8	PT2B	0		C	PT2B	0		C
G7	PT2A	0		T	PT2A	0		T
L12	VCC	-			VCC	-		
L13	VCC	-			VCC	-		
L14	VCC	-			VCC	-		
L15	VCC	-			VCC	-		
M11	VCC	-			VCC	-		
M12	VCC	-			VCC	-		
M15	VCC	-			VCC	-		
M16	VCC	-			VCC	-		
N11	VCC	-			VCC	-		
N16	VCC	-			VCC	-		
P11	VCC	-			VCC	-		
P16	VCC	-			VCC	-		
R11	VCC	-			VCC	-		
R12	VCC	-			VCC	-		
R15	VCC	-			VCC	-		
R16	VCC	-			VCC	-		
T12	VCC	-			VCC	-		
T13	VCC	-			VCC	-		
T14	VCC	-			VCC	-		
T15	VCC	-			VCC	-		
B12	VCCIO0	0			VCCIO0	0		
B7	VCCIO0	0			VCCIO0	0		
F11	VCCIO0	0			VCCIO0	0		
J13	VCCIO0	0			VCCIO0	0		
K12	VCCIO0	0			VCCIO0	1		
D18	VCCIO1	1			VCCIO1	1		
F16	VCCIO1	1			VCCIO1	1		
J14	VCCIO1	1			VCCIO1	1		
K15	VCCIO1	1			VCCIO1	1		
G25	VCCIO2	2			VCCIO2	2		
L21	VCCIO2	2			VCCIO2	2		
M17	VCCIO2	2			VCCIO2	2		
M25	VCCIO2	2			VCCIO2	2		
N18	VCCIO2	2			VCCIO2	2		

LFE2M35E/SE and LFE2M50E/SE Logic Signal Connections: 672 fpBGA

LFE2M35E/SE					LFE2M50E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
P18	VCCIO3	3			VCCIO3	3		
R17	VCCIO3	3			VCCIO3	3		
R25	VCCIO3	3			VCCIO3	3		
T21	VCCIO3	3			VCCIO3	3		
Y25	VCCIO3	3			VCCIO3	3		
AA16	VCCIO4	4			VCCIO4	4		
AC18	VCCIO4	4			VCCIO4	4		
U15	VCCIO4	4			VCCIO4	4		
V14	VCCIO4	4			VCCIO4	4		
AA11	VCCIO5	5			VCCIO5	5		
V13	VCCIO5	5			VCCIO5	5		
AE12	VCCIO5	5			VCCIO5	5		
AE7	VCCIO5	5			VCCIO5	5		
U12	VCCIO5	5			VCCIO5	5		
P9	VCCIO6	6			VCCIO6	6		
R10	VCCIO6	6			VCCIO6	6		
R2	VCCIO6	6			VCCIO6	6		
T6	VCCIO6	6			VCCIO6	6		
Y2	VCCIO6	6			VCCIO6	6		
G2	VCCIO7	7			VCCIO7	7		
L6	VCCIO7	7			VCCIO7	7		
M10	VCCIO7	7			VCCIO7	7		
M2	VCCIO7	7			VCCIO7	7		
N9	VCCIO7	7			VCCIO7	7		
AC24	VCCIO8	8			VCCIO8	8		
U17	VCCIO8	8			VCCIO8	8		
J11	VCCAUX	-			VCCAUX	-		
J12	VCCAUX	-			VCCAUX	-		
J15	VCCAUX	-			VCCAUX	-		
J16	VCCAUX	-			VCCAUX	-		
L18	VCCAUX	-			VCCAUX	-		
L9	VCCAUX	-			VCCAUX	-		
M18	VCCAUX	-			VCCAUX	-		
M9	VCCAUX	-			VCCAUX	-		
R18	VCCAUX	-			VCCAUX	-		
R9	VCCAUX	-			VCCAUX	-		
T18	VCCAUX	-			VCCAUX	-		
T9	VCCAUX	-			VCCAUX	-		
V11	VCCAUX	-			VCCAUX	-		
V12	VCCAUX	-			VCCAUX	-		
V15	VCCAUX	-			VCCAUX	-		
V16	VCCAUX	-			VCCAUX	-		
A13	GND	-			GND	-		
A19	GND	-			GND	-		
A2	GND	-			GND	-		
A25	GND	-			GND	-		
AA2	GND	-			GND	-		
AA25	GND	-			GND	-		
AB18	GND	-			GND	-		

LFE2M35E/SE and LFE2M50E/SE Logic Signal Connections: 672 fpBGA

LFE2M35E/SE					LFE2M50E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
AB22	GND	-			GND	-		
AB5	GND	-			GND	-		
AB9	GND	-			GND	-		
AE1	GND	-			GND	-		
AE11	GND	-			GND	-		
AE16	GND	-			GND	-		
AE22	GND	-			GND	-		
AE26	GND	-			GND	-		
AE6	GND	-			GND	-		
AF13	GND	-			GND	-		
AF19	GND	-			GND	-		
AF2	GND	-			GND	-		
AF25	GND	-			GND	-		
B1	GND	-			GND	-		
B11	GND	-			GND	-		
B16	GND	-			GND	-		
B22	GND	-			GND	-		
B26	GND	-			GND	-		
B6	GND	-			GND	-		
E18	GND	-			GND	-		
E22	GND	-			GND	-		
E5	GND	-			GND	-		
E9	GND	-			GND	-		
F2	GND	-			GND	-		
F25	GND	-			GND	-		
G11	GND	-			GND	-		
G16	GND	-			GND	-		
J22	GND	-			GND	-		
J5	GND	-			GND	-		
K11	GND	-			GND	-		
K13	GND	-			GND	-		
K14	GND	-			GND	-		
K16	GND	-			GND	-		
L10	GND	-			GND	-		
L11	GND	-			GND	-		
L16	GND	-			GND	-		
L17	GND	-			GND	-		
L2	GND	-			GND	-		
L20	GND	-			GND	-		
L25	GND	-			GND	-		
L7	GND	-			GND	-		
M13	GND	-			GND	-		
M14	GND	-			GND	-		
N10	GND	-			GND	-		
N12	GND	-			GND	-		
N13	GND	-			GND	-		
N14	GND	-			GND	-		
N15	GND	-			GND	-		
N17	GND	-			GND	-		

LFE2M35E/SE and LFE2M50E/SE Logic Signal Connections: 672 fpBGA

LFE2M35E/SE					LFE2M50E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
P10	GND	-			GND	-		
P12	GND	-			GND	-		
P13	GND	-			GND	-		
P14	GND	-			GND	-		
P15	GND	-			GND	-		
P17	GND	-			GND	-		
R13	GND	-			GND	-		
R14	GND	-			GND	-		
T10	GND	-			GND	-		
T11	GND	-			GND	-		
T16	GND	-			GND	-		
T17	GND	-			GND	-		
T2	GND	-			GND	-		
T20	GND	-			GND	-		
T25	GND	-			GND	-		
T7	GND	-			GND	-		
U11	GND	-			GND	-		
U13	GND	-			GND	-		
U14	GND	-			GND	-		
U16	GND	-			GND	-		
V22	GND	-			GND	-		
V5	GND	-			GND	-		
Y11	GND	-			GND	-		
Y16	GND	-			GND	-		
AB3	NC	-			NC	-		
AB4	NC	-			NC	-		
AC1	NC	-			NC	-		
AC2	NC	-			NC	-		
B4	NC	-			NC	-		
B5	NC	-			NC	-		
C26	NC	-			NC	-		
D20	NC	-			NC	-		
D21	NC	-			NC	-		
D22	NC	-			NC	-		
D23	NC	-			NC	-		
D24	NC	-			NC	-		
D25	NC	-			NC	-		
D26	NC	-			NC	-		
E20	NC	-			NC	-		
E21	NC	-			NC	-		
E25	NC	-			NC	-		
E26	NC	-			NC	-		
F20	NC	-			NC	-		
G20	NC	-			NC	-		
K10	NC	-			NC	-		
K17	NC	-			NC	-		
R4	NC	-			NC	-		
U10	NC	-			NC	-		
U23	NC	-			NC	-		

LFE2M35E/SE and LFE2M50E/SE Logic Signal Connections: 672 fpBGA

LFE2M35E/SE					LFE2M50E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
V10	NC	-			NC	-		
W7	NC	-			NC	-		
AB21	PB69B	4	BDQ69	C	NC	-		
AC20	PB58A	4	BDQ60	T	NC	-		
AC21	PB63A	4	BDQ60	T	NC	-		
AC22	PB69A	4	BDQS69****	T	NC	-		
AC23	PB71A	4	BDQ69	T	NC	-		
AC25	PB71B	4	BDQ69	C	NC	-		
AD26	PB70B	4	BDQ69	C	NC	-		
W20	PB72B	4	BDQ69	C	NC	-		
H7	L_VCCPLL	-			L_VCCPLL	-		
K6	L_VCCPLL	-			L_VCCPLL	-		
P7	L_VCCPLL	-			L_VCCPLL	-		
R8	L_VCCPLL	-			L_VCCPLL	-		
V18	R_VCCPLL	-			R_VCCPLL	-		
P20	R_VCCPLL	-			R_VCCPLL	-		
J17	R_VCCPLL	-			R_VCCPLL	-		
G19	R_VCCPLL	-			R_VCCPLL	-		

* Supports true LVDS. Other differential signals must be emulated with external resistors.

** These dedicated input pins can be used for GPLLs or GDLLs within the respective quadrant.

*** For density migration, board design must take into account that these sysCONFIG pins are dual function for the lower density devices (ECP2M20 and ECP2M35). They can be either sysCONFIG pins or general purpose I/Os. These pins are dedicated pins for the higher density devices (ECP2M50, ECP2M70 and ECP2M100).

****Due to packaging bond out option, this DQS does not have all the necessary DQ pins bonded out for a full 8-bit data width.

Note: VCCIO and GND pads are used to determine the average DC current drawn by I/Os between GND/VCCIO connections, or between the last GND/VCCIO in an I/O bank and the end of an I/O bank. The substrate pads listed in the Pin Table do not necessarily have a one to one connection with a package ball or pin.

LFE2M50E/SE and LFE2M70E/SE Logic Signal Connections: 900 fpBGA

LFE2M50E/SE					LFE2M70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
D2	PL9A	7	VREF2_7/LDQ6	T	PL9A	7	VREF2_7	T
D3	PL9B	7	VREF1_7/LDQ6	C	PL9B	7	VREF1_7	C
GNDIO	GNDIO7	-			GNDIO7	-		
J8	PL11A	7	LUM0_SPLLT_IN_A	T (LVDS)*	PL11A	7	LUM0_SPLLT_IN_A/LDQ15	T (LVDS)*
H7	PL11B	7	LUM0_SPLLC_IN_A	C (LVDS)*	PL11B	7	LUM0_SPLLC_IN_A/LDQ15	C (LVDS)*
E3	PL12A	7	LUM0_SPLLT_FB_A	T	PL12A	7	LUM0_SPLLT_FB_A/LDQ15	T
E4	PL12B	7	LUM0_SPLLC_FB_A	C	PL12B	7	LUM0_SPLLC_FB_A/LDQ15	C
GNDIO	GNDIO7	-			-	-		
G6	PL13A	7		T (LVDS)*	PL13A	7	LDQ15	T (LVDS)*
F5	PL13B	7		C (LVDS)*	PL13B	7	LDQ15	C (LVDS)*
E2	PL14A	7		T	PL14A	7	LDQ15	T
D1	PL14B	7		C	PL14B	7	LDQ15	C
-	-	-			GNDIO7	-		
G5	NC	-			PL15A	7	LDQS15	T (LVDS)*
G4	NC	-			PL15B	7	LDQ15	C (LVDS)*
K7	NC	-			PL16A	7	LDQ15	T
K8	NC	-			PL16B	7	LDQ15	C
E1	NC	-			PL17A	7	LDQ15	T (LVDS)*
F2	NC	-			PL17B	7	LDQ15	C (LVDS)*
F1	NC	-			PL18A	7	LDQ15	T
-	-	-			GNDIO7	-		
G3	NC	-			PL18B	7	LDQ15	C
H5	PL15A	7		T (LVDS)*	PL21A	7		T (LVDS)*
H4	PL15B	7		C (LVDS)*	PL21B	7		C (LVDS)*
J5	PL16A	7		T	PL22A	7		T
J4	PL16B	7		C	PL22B	7		C
GNDIO	GNDIO7	-			GNDIO7	-		
G2	NC	-			PL24A	7	LDQ28	T (LVDS)*
G1	NC	-			PL24B	7	LDQ28	C (LVDS)*
L9	NC	-			PL25A	7	LDQ28	T
L7	NC	-			PL25B	7	LDQ28	C
K6	NC	-			PL26A	7	LDQ28	T (LVDS)*
K5	NC	-			PL26B	7	LDQ28	C (LVDS)*
L8	NC	-			PL27A	7	LDQ28	T
L6	NC	-			PL27B	7	LDQ28	C
-	-	-			GNDIO7	-		
H3	PL18A	7		T (LVDS)*	PL28A	7	LDQS28	T (LVDS)*
H2	PL18B	7		C (LVDS)*	PL28B	7	LDQ28	C (LVDS)*
N8	PL19A	7		T	PL29A	7	LDQ28	T
M9	PL19B	7		C	PL29B	7	LDQ28	C
J3	PL20A	7		T (LVDS)*	PL30A	7	LDQ28	T (LVDS)*
VCCIO	VCCIO7	7			-	-		
J2	PL20B	7		C (LVDS)*	PL30B	7	LDQ28	C (LVDS)*
H1	PL21A	7		T	PL31A	7	LDQ28	T
GNDIO	GNDIO7	-			GNDIO7	-		
J1	PL21B	7		C	PL31B	7	LDQ28	C
-	-	-			-	-		
-	-	-			-	-		
L5	PL23A	7	LDQ27	T (LVDS)*	PL33A	7	LDQ37	T (LVDS)*

LFE2M50E/SE and LFE2M70E/SE Logic Signal Connections: 900 fpBGA

LFE2M50E/SE					LFE2M70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
L4	PL23B	7	LDQ27	C (LVDS)*	PL33B	7	LDQ37	C (LVDS)*
N9	PL24A	7	LDQ27	T	PL34A	7	LDQ37	T
N7	PL24B	7	LDQ27	C	PL34B	7	LDQ37	C
K2	PL25A	7	LDQ27	T (LVDS)*	PL35A	7	LDQ37	T (LVDS)*
K1	PL25B	7	LDQ27	C (LVDS)*	PL35B	7	LDQ37	C (LVDS)*
P9	PL26A	7	LDQ27	T	PL36A	7	LDQ37	T
P7	PL26B	7	LDQ27	C	PL36B	7	LDQ37	C
GNDIO	GNDIO7	-			GNDIO7	-		
M6	PL27A	7	LDQS27	T (LVDS)*	PL37A	7	LDQS37	T (LVDS)*
M5	PL27B	7	LDQ27	C (LVDS)*	PL37B	7	LDQ37	C (LVDS)*
N5	PL28A	7	LDQ27	T	PL38A	7	LDQ37	T
N6	PL28B	7	LDQ27	C	PL38B	7	LDQ37	C
M4	PL29A	7	LDQ27	T (LVDS)*	PL39A	7	LDQ37	T (LVDS)*
M3	PL29B	7	LDQ27	C (LVDS)*	PL39B	7	LDQ37	C (LVDS)*
P6	PL30A	7	LDQ27	T	PL40A	7	LDQ37	T
GNDIO	GNDIO7	-			GNDIO7	-		
P8	PL30B	7	LDQ27	C	PL40B	7	LDQ37	C
L3	PL32A	7	LUM3_SPLLT_IN_A/LDQ36	T (LVDS)*	PL42A	7	LUM3_SPLLT_IN_A/LDQ46	T (LVDS)*
L2	PL32B	7	LUM3_SPLLC_IN_A/LDQ36	C (LVDS)*	PL42B	7	LUM3_SPLLC_IN_A/LDQ46	C (LVDS)*
P5	PL33A	7	LUM3_SPLLT_FB_A/LDQ36	T	PL43A	7	LUM3_SPLLT_FB_A/LDQ46	T
P4	PL33B	7	LUM3_SPLLC_FB_A/LDQ36	C	PL43B	7	LUM3_SPLLC_FB_A/LDQ46	C
L1	PL34A	7	LDQ36	T (LVDS)*	PL44A	7	LDQ46	T (LVDS)*
M2	PL34B	7	LDQ36	C (LVDS)*	PL44B	7	LDQ46	C (LVDS)*
R5	PL35A	7	LDQ36	T	PL45A	7	LDQ46	T
R4	PL35B	7	LDQ36	C	PL45B	7	LDQ46	C
GNDIO	GNDIO7	-			GNDIO7	-		
M1	PL36A	7	LDQS36	T (LVDS)*	PL46A	7	LDQS46	T (LVDS)*
N2	PL36B	7	LDQ36	C (LVDS)*	PL46B	7	LDQ46	C (LVDS)*
R8	PL37A	7	LDQ36	T	PL47A	7	LDQ46	T
T9	PL37B	7	LDQ36	C	PL47B	7	LDQ46	C
P3	PL38A	7	LDQ36	T (LVDS)*	PL48A	7	LDQ46	T (LVDS)*
P2	PL38B	7	LDQ36	C (LVDS)*	PL48B	7	LDQ46	C (LVDS)*
N1	PL39A	7	PCLKT7_0/LDQ36	T	PL49A	7	PCLKT7_0/LDQ46	T
GNDIO	GNDIO7	-			GNDIO7	-		
P1	PL39B	7	PCLKC7_0/LDQ36	C	PL49B	7	PCLKC7_0/LDQ46	C
T5	PL41A	6	PCLKT6_0	T (LVDS)*	PL51A	6	PCLKT6_0/LDQ55	T (LVDS)*
T4	PL41B	6	PCLKC6_0	C (LVDS)*	PL51B	6	PCLKC6_0/LDQ55	C (LVDS)*
U7	PL42A	6	VREF2_6	T	PL52A	6	VREF2_6/LDQ55	T
T8	PL42B	6	VREF1_6	C	PL52B	6	VREF1_6/LDQ55	C
R3	PL43A	6		T (LVDS)*	PL53A	6	LDQ55	T (LVDS)*
VCCIO	VCCIO6	6			VCCIO6	6		
R2	PL43B	6		C (LVDS)*	PL53B	6	LDQ55	C (LVDS)*
R1	PL44A	6		T	PL54A	6	LDQ55	T
T1	PL44B	6		C	PL54B	6	LDQ55	C
GNDIO	GNDIO6	-			GNDIO6	-		
-	-	-			VCCIO6	6		
T3	PL45A	6	LLM3_SPLLT_IN_A	T (LVDS)*	PL57A	6	LLM3_SPLLT_IN_A/LDQ55	T (LVDS)*
T2	PL45B	6	LLM3_SPLLC_IN_A	C (LVDS)*	PL57B	6	LLM3_SPLLC_IN_A/LDQ55	C (LVDS)*
U9	PL46A	6	LLM3_SPLLT_FB_A	T	PL58A	6	LLM3_SPLLT_FB_A/LDQ55	T

LFE2M50E/SE and LFE2M70E/SE Logic Signal Connections: 900 fpBGA

LFE2M50E/SE					LFE2M70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
U8	PL46B	6	LLM3_SPLLC_FB_A	C	PL58B	6	LLM3_SPLLC_FB_A/LDQ55	C
VCCIO	VCCIO6	6			GNDIO6	-		
U5	PL48A	6	LDQ52	T (LVDS)*	PL60A	6	LDQ64	T (LVDS)*
U4	PL48B	6	LDQ52	C (LVDS)*	PL60B	6	LDQ64	C (LVDS)*
V9	PL49A	6	LDQ52	T	PL61A	6	LDQ64	T
V7	PL49B	6	LDQ52	C	PL61B	6	LDQ64	C
VCCIO	VCCIO6	6			VCCIO6	6		
U3	PL50A	6	LDQ52	T (LVDS)*	PL62A	6	LDQ64	T (LVDS)*
U2	PL50B	6	LDQ52	C (LVDS)*	PL62B	6	LDQ64	C (LVDS)*
V8	PL51A	6	LDQ52	T	PL63A	6	LDQ64	T
U6	PL51B	6	LDQ52	C	PL63B	6	LDQ64	C
GNDIO	GNDIO6	-			GNDIO6	-		
U1	PL52A	6	LDQS52	T (LVDS)*	PL64A	6	LDQS64	T (LVDS)*
V2	PL52B	6	LDQ52	C (LVDS)*	PL64B	6	LDQ64	C (LVDS)*
V5	PL53A	6	LDQ52	T	PL65A	6	LDQ64	T
VCCIO	VCCIO6	6			VCCIO6	6		
V6	PL53B	6	LDQ52	C	PL65B	6	LDQ64	C
V1	PL54A	6	LDQ52	T (LVDS)*	PL66A	6	LDQ64	T (LVDS)*
W1	PL54B	6	LDQ52	C (LVDS)*	PL66B	6	LDQ64	C (LVDS)*
W5	PL55A	6	LDQ52	T	PL67A	6	LDQ64	T
GNDIO	GNDIO6	-			GNDIO6	-		
W6	PL55B	6	LDQ52	C	PL67B	6	LDQ64	C
W3	PL57A	6		T (LVDS)*	PL69A	6	LDQ73	T (LVDS)*
W4	PL57B	6		C (LVDS)*	PL69B	6	LDQ73	C (LVDS)*
W2	PL58A	6		T	PL70A	6	LDQ73	T
Y4	PL58B	6		C	PL70B	6	LDQ73	C
Y1	PL59A	6		T (LVDS)*	PL71A	6	LDQ73	T (LVDS)*
VCCIO	VCCIO6	6			VCCIO6	6		
Y2	PL59B	6		C (LVDS)*	PL71B	6	LDQ73	C (LVDS)*
Y5	PL60A	6		T	PL72A	6	LDQ73	T
Y6	PL60B	6		C	PL72B	6	LDQ73	C
AA1	NC	-			PL73A	6	LDQS73	T (LVDS)*
GNDIO	GNDIO6	-			GNDIO6	-		
AA2	NC	-			PL73B	6	LDQ73	C (LVDS)*
Y3	NC	-			PL74A	6	LDQ73	T
AB1	NC	-			PL74B	6	LDQ73	C
-	-	-			VCCIO6	6		
Y9	NC	-			PL75A	6	LDQ73	T (LVDS)*
Y8	NC	-			PL75B	6	LDQ73	C (LVDS)*
Y7	NC	-			PL76A	6	LDQ73	T
AA7	NC	-			PL76B	6	LDQ73	C
-	-	-			GNDIO6	-		
-	-	-			-	-		
AB2	NC	-			PL78A	6	LDQ82	T (LVDS)*
AB3	NC	-			PL78B	6	LDQ82	C (LVDS)*
AA5	NC	-			PL79A	6	LDQ82	T
AA6	NC	-			PL79B	6	LDQ82	C
AB4	NC	-			PL80A	6	LDQ82	T (LVDS)*
-	-	-			VCCIO6	6		

LFE2M50E/SE and LFE2M70E/SE Logic Signal Connections: 900 fpBGA

LFE2M50E/SE					LFE2M70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
AB5	NC	-			PL80B	6	LDQ82	C (LVDS)*
AA8	NC	-			PL81A	6	LDQ82	T
AA9	NC	-			PL81B	6	LDQ82	C
AC1	PL62A	6	LLM0_GPLLT_IN_A**	T (LVDS)*	PL82A	6	LLM0_GPLLT_IN_A**/LDQS82	T (LVDS)*
GNDIO	GNDIO6	-			GNDIO6	-		
AC2	PL62B	6	LLM0_GPLLC_IN_A**	C (LVDS)*	PL82B	6	LLM0_GPLLC_IN_A**/LDQ82	C (LVDS)*
AC4	PL63A	6	LLM0_GPLLT_FB_A	T	PL83A	6	LLM0_GPLLT_FB_A/LDQ82	T
AC3	PL63B	6	LLM0_GPLLC_FB_A	C	PL83B	6	LLM0_GPLLC_FB_A/LDQ82	C
VCCIO	VCCIO6	6			VCCIO6	6		
AC7	PL64A	6	LLM0_GDLLT_IN_A**	T (LVDS)*	PL84A	6	LLM0_GDLLT_IN_A**/LDQ82	T (LVDS)*
AC6	PL64B	6	LLM0_GDLLC_IN_A**	C (LVDS)*	PL84B	6	LLM0_GDLLC_IN_A**/LDQ82	C (LVDS)*
AC5	PL65A	6	LLM0_GDLLT_FB_A	T	PL85A	6	LLM0_GDLLT_FB_A/LDQ82	T
AD3	PL65B	6	LLM0_GDLLC_FB_A	C	PL85B	6	LLM0_GDLLC_FB_A/LDQ82	C
GNDIO	GNDIO6	-			GNDIO6	-		
AB8	LLM0_PLLCAP	6			LLM0_PLLCAP	6		
AD2	PL67A	6	LDQ71	T (LVDS)*	PL87A	6		T
AD1	PL67B	6	LDQ71	C (LVDS)*	PL87B	6		C
AE2	TCK	-			TCK	-		
AE1	TDI	-			TDI	-		
AF2	TMS	-			TMS	-		
AF1	TDO	-			TDO	-		
AG1	VCCJ	-			VCCJ	-		
AH1	VCC	-			LLC_SQ_VCCRX3	14		
AK2	PB11A	5	BDQ15	T	LLC_SQ_HDINP3	14		T
AJ1	NC	-			LLC_SQ_VCCIB3	14		
AJ2	PB11B	5	BDQ15	C	LLC_SQ_HDINN3	14		C
AH4	VCC	-			LLC_SQ_VCCTX3	14		
AK5	PB13A	5	BDQ15	T	LLC_SQ_HDOUTP3	14		T
AK4	NC	-			LLC_SQ_VCCOB3	14		
AJ5	PB13B	5	BDQ15	C	LLC_SQ_HDOUTN3	14		C
AH5	VCC	-			LLC_SQ_VCCTX2	14		
AJ6	PB14B	5	BDQ15	C	LLC_SQ_HDOUTN2	14		C
AH6	NC	-			LLC_SQ_VCCOB2	14		
AK6	PB14A	5	BDQ15	T	LLC_SQ_HDOUTP2	14		T
AH2	VCC	-			LLC_SQ_VCCRX2	14		
AJ3	PB12B	5	BDQ15	C	LLC_SQ_HDINN2	14		C
AH3	NC	-			LLC_SQ_VCCIB2	14		
AK3	PB12A	5	BDQ15	T	LLC_SQ_HDINP2	14		T
AH7	VCC	-			LLC_SQ_VCCP	14		
AG7	PB15A	5	BDQS15	T	LLC_SQ_REFCLKP	14		T
AF7	PB15B	5	BDQ15	C	LLC_SQ_REFCLKN	14		C
AJ7	VCCAUX	-			LLC_SQ_VCCAUX33	14		
AK11	PB18A	5	BDQ15	T	LLC_SQ_HDINP1	14		T
AH11	NC	-			LLC_SQ_VCCIB1	14		
AJ11	PB18B	5	BDQ15	C	LLC_SQ_HDINN1	14		C
AH12	VCC	-			LLC_SQ_VCCRX1	14		
AK8	PB16A	5	BDQ15	T	LLC_SQ_HDOUTP1	14		T
AH8	NC	-			LLC_SQ_VCCOB1	14		
AJ8	PB16B	5	BDQ15	C	LLC_SQ_HDOUTN1	14		C

LFE2M50E/SE and LFE2M70E/SE Logic Signal Connections: 900 fpBGA

LFE2M50E/SE					LFE2M70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
AH9	VCC	-			LLC_SQ_VCCTX1	14		
AJ9	PB17B	5	BDQ15	C	LLC_SQ_HDOUTN0	14		C
AK10	NC	-			LLC_SQ_VCCOB0	14		
AK9	PB17A	5	BDQ15	T	LLC_SQ_HDOUTP0	14		T
AH10	VCC	-			LLC_SQ_VCCTX0	14		
AJ12	PB19B	5	BDQ15	C	LLC_SQ_HDINN0	14		C
AJ13	NC	-			LLC_SQ_VCCIB0	14		
AK12	PB19A	5	BDQ15	T	LLC_SQ_HDINP0	14		T
AH13	VCC	-			LLC_SQ_VCCRX0	14		
AF10	PB3A	5	BDQ6	T	PB30A	5	BDQ33	T
AE8	PB3B	5	BDQ6	C	PB30B	5	BDQ33	C
AE11	PB4A	5	BDQ6	T	PB31A	5	BDQ33	T
VCCIO	VCCIO5	5			VCCIO5	5		
AD9	PB4B	5	BDQ6	C	PB31B	5	BDQ33	C
AE10	PB5A	5	BDQ6	T	PB32A	5	BDQ33	T
AD10	PB5B	5	BDQ6	C	PB32B	5	BDQ33	C
AE13	PB6A	5	BDQS6	T	PB33A	5	BDQS33	T
GNDIO	GNDIO5	-			GNDIO5	-		
AC12	PB6B	5	BDQ6	C	PB33B	5	BDQ33	C
AG2	PB7A	5	BDQ6	T	PB34A	5	BDQ33	T
AG3	PB7B	5	BDQ6	C	PB34B	5	BDQ33	C
AD13	PB8A	5	BDQ6	T	PB35A	5	BDQ33	T
VCCIO	VCCIO5	5			VCCIO5	5		
AC13	PB8B	5	BDQ6	C	PB35B	5	BDQ33	C
AE14	PB9A	5	BDQ6	T	PB36A	5	BDQ33	T
AC14	PB9B	5	BDQ6	C	PB36B	5	BDQ33	C
AF3	PB10A	5	BDQ6	T	PB37A	5	BDQ33	T
GNDIO	GNDIO5	-			GNDIO5	-		
AF4	PB10B	5	BDQ6	C	PB37B	5	BDQ33	C
VCCIO	VCCIO5	5			-	-		
AG4	PB20A	5	BDQ24	T	PB38A	5	BDQ42	T
AG5	PB20B	5	BDQ24	C	PB38B	5	BDQ42	C
GNDIO	GNDIO5	-			-	-		
VCCIO	VCCIO5	5			-	-		
AD11	PB24A	5	BDQS24****	T	PB39A	5	BDQ42	T
AF13	PB24B	5	BDQ24	C	PB39B	5	BDQ42	C
AF12	PB25A	5	BDQ24	T	PB40A	5	BDQ42	T
-	-	-			VCCIO5	5		
AD14	PB25B	5	BDQ24	C	PB40B	5	BDQ42	C
AG8	PB26A	5	BDQ24	T	PB41A	5	BDQ42	T
AF8	PB26B	5	BDQ24	C	PB41B	5	BDQ42	C
AE15	PB27A	5	BDQ24	T	PB42A	5	BDQS42****	T
-	-	-			GNDIO5	-		
VCCIO	VCCIO5	5			-	-		
AC15	PB27B	5	BDQ24	C	PB42B	5	BDQ42	C
VCCIO	VCCIO5	5			VCCIO5	5		
GNDIO	GNDIO5	-			GNDIO5	-		
AD15	PB38A	5	BDQ42	T	PB47A	5	BDQ51	T
AF15	PB38B	5	BDQ42	C	PB47B	5	BDQ51	C

LFE2M50E/SE and LFE2M70E/SE Logic Signal Connections: 900 fpBGA

LFE2M50E/SE					LFE2M70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
AG10	PB39A	5	BDQ42	T	PB48A	5	BDQ51	T
AG9	PB39B	5	BDQ42	C	PB48B	5	BDQ51	C
AH14	PB40A	5	BDQ42	T	PB49A	5	BDQ51	T
AG12	PB40B	5	BDQ42	C	PB49B	5	BDQ51	C
VCCIO	VCCIO5	5			VCCIO5	5		
AG15	PB41A	5	BDQ42	T	PB50A	5	BDQ51	T
AG13	PB41B	5	BDQ42	C	PB50B	5	BDQ51	C
GNDIO	GNDIO5	-			GNDIO5	-		
AF16	PB42A	5	BDQS42	T	PB51A	5	BDQS51	T
AH15	PB42B	5	BDQ42	C	PB51B	5	BDQ51	C
AC16	PB43A	5	VREF2_5/BDQ42	T	PB52A	5	VREF2_5/BDQ51	T
AE16	PB43B	5	VREF1_5/BDQ42	C	PB52B	5	VREF1_5/BDQ51	C
AG11	PB44A	5	PCLKT5_0/BDQ42	T	PB53A	5	PCLKT5_0/BDQ51	T
AF11	PB44B	5	PCLKC5_0/BDQ42	C	PB53B	5	PCLKC5_0/BDQ51	C
VCCIO	VCCIO5	5			VCCIO5	5		
GNDIO	GNDIO5	-			GNDIO5	-		
AJ14	PB49A	4	PCLKT4_0/BDQ51	T	PB58A	4	PCLKT4_0/BDQ60	T
VCCIO	VCCIO4	4			VCCIO4	4		
AK14	PB49B	4	PCLKC4_0/BDQ51	C	PB58B	4	PCLKC4_0/BDQ60	C
AK15	PB50A	4	VREF2_4/BDQ51	T	PB59A	4	VREF2_4/BDQ60	T
AK16	PB50B	4	VREF1_4/BDQ51	C	PB59B	4	VREF1_4/BDQ60	C
AF18	PB51A	4	BDQS51	T	PB60A	4	BDQS60	T
GNDIO	GNDIO4	-			GNDIO4	-		
AD16	PB51B	4	BDQ51	C	PB60B	4	BDQ60	C
AJ15	PB52A	4	BDQ51	T	PB61A	4	BDQ60	T
AG16	PB52B	4	BDQ51	C	PB61B	4	BDQ60	C
AE17	PB53A	4	BDQ51	T	PB62A	4	BDQ60	T
VCCIO	VCCIO4	4			VCCIO4	4		
AC17	PB53B	4	BDQ51	C	PB62B	4	BDQ60	C
AH16	PB54A	4	BDQ51	T	PB63A	4	BDQ60	T
AK17	PB54B	4	BDQ51	C	PB63B	4	BDQ60	C
AG20	PB55A	4	BDQ51	T	PB64A	4	BDQ60	T
GNDIO	GNDIO4	-			GNDIO4	-		
AG21	PB55B	4	BDQ51	C	PB64B	4	BDQ60	C
AG18	PB56A	4	BDQ60	T	PB65A	4	BDQ69	T
AJ16	PB56B	4	BDQ60	C	PB65B	4	BDQ69	C
AF21	PB57A	4	BDQ60	T	PB66A	4	BDQ69	T
AG22	PB57B	4	BDQ60	C	PB66B	4	BDQ69	C
AD17	PB58A	4	BDQ60	T	PB67A	4	BDQ69	T
AF19	PB58B	4	BDQ60	C	PB67B	4	BDQ69	C
VCCIO	VCCIO4	4			VCCIO4	4		
GNDIO	GNDIO4	-			GNDIO4	-		
AH17	PB62A	4	BDQ60	T	PB71A	4	BDQ69	T
AJ17	PB62B	4	BDQ60	C	PB71B	4	BDQ69	C
VCCIO	VCCIO4	4			VCCIO4	4		
AF26	PB64A	4	BDQ60	T	PB73A	4	BDQ69	T
AE25	PB64B	4	BDQ60	C	PB73B	4	BDQ69	C
GNDIO	GNDIO4	-			GNDIO4	-		
AD24	PB65A	4	BDQ69	T	PB74A	4	BDQ78	T

LFE2M50E/SE and LFE2M70E/SE Logic Signal Connections: 900 fpBGA

LFE2M50E/SE					LFE2M70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
AE24	PB65B	4	BDQ69	C	PB74B	4	BDQ78	C
AD18	PB66A	4	BDQ69	T	PB75A	4	BDQ78	T
AC18	PB66B	4	BDQ69	C	PB75B	4	BDQ78	C
AE18	PB67A	4	BDQ69	T	PB76A	4	BDQ78	T
AG19	PB67B	4	BDQ69	C	PB76B	4	BDQ78	C
VCCIO	VCCIO4	4			VCCIO4	4		
GNDIO	GNDIO4	-			GNDIO4	-		
AC19	PB69A	4	BDQS69	T	PB78A	4	BDQS78	T
AD20	PB69B	4	BDQ69	C	PB78B	4	BDQ78	C
AB18	PB70A	4	BDQ69	T	PB79A	4	BDQ78	T
AC20	PB70B	4	BDQ69	C	PB79B	4	BDQ78	C
AE20	PB71A	4	BDQ69	T	PB80A	4	BDQ78	T
AE21	PB71B	4	BDQ69	C	PB80B	4	BDQ78	C
VCCIO	VCCIO4	4			VCCIO4	4		
AC23	PB72A	4	BDQ69	T	PB81A	4	BDQ78	T
AD23	PB72B	4	BDQ69	C	PB81B	4	BDQ78	C
GNDIO	GNDIO4	-			GNDIO4	-		
AH18	LRC_SQ_VCCRX3	13			LRC_SQ_VCCRX3	13		
AK19	LRC_SQ_HDINP3	13		T	LRC_SQ_HDINP3	13		T
AJ18	LRC_SQ_VCCIB3	13			LRC_SQ_VCCIB3	13		
AJ19	LRC_SQ_HDINN3	13		C	LRC_SQ_HDINN3	13		C
AH21	LRC_SQ_VCCTX3	13			LRC_SQ_VCCTX3	13		
AK22	LRC_SQ_HDOUTP3	13		T	LRC_SQ_HDOUTP3	13		T
AK21	LRC_SQ_VCCOB3	13			LRC_SQ_VCCOB3	13		
AJ22	LRC_SQ_HDOUTN3	13		C	LRC_SQ_HDOUTN3	13		C
AH22	LRC_SQ_VCCTX2	13			LRC_SQ_VCCTX2	13		
AJ23	LRC_SQ_HDOUTN2	13		C	LRC_SQ_HDOUTN2	13		C
AH23	LRC_SQ_VCCOB2	13			LRC_SQ_VCCOB2	13		
AK23	LRC_SQ_HDOUTP2	13		T	LRC_SQ_HDOUTP2	13		T
AH19	LRC_SQ_VCCRX2	13			LRC_SQ_VCCRX2	13		
AJ20	LRC_SQ_HDINN2	13		C	LRC_SQ_HDINN2	13		C
AH20	LRC_SQ_VCCIB2	13			LRC_SQ_VCCIB2	13		
AK20	LRC_SQ_HDINP2	13		T	LRC_SQ_HDINP2	13		T
AH24	LRC_SQ_VCCP	13			LRC_SQ_VCCP	13		
AG24	LRC_SQ_REFCLKP	13		T	LRC_SQ_REFCLKP	13		T
AF24	LRC_SQ_REFCLKN	13		C	LRC_SQ_REFCLKN	13		C
AJ24	LRC_SQ_VCCAUX33	13			LRC_SQ_VCCAUX33	13		
AK28	LRC_SQ_HDINP1	13		T	LRC_SQ_HDINP1	13		T
AH28	LRC_SQ_VCCIB1	13			LRC_SQ_VCCIB1	13		
AJ28	LRC_SQ_HDINN1	13		C	LRC_SQ_HDINN1	13		C
AH29	LRC_SQ_VCCRX1	13			LRC_SQ_VCCRX1	13		
AK25	LRC_SQ_HDOUTP1	13		T	LRC_SQ_HDOUTP1	13		T
AH25	LRC_SQ_VCCOB1	13			LRC_SQ_VCCOB1	13		
AJ25	LRC_SQ_HDOUTN1	13		C	LRC_SQ_HDOUTN1	13		C
AH26	LRC_SQ_VCCTX1	13			LRC_SQ_VCCTX1	13		
AJ26	LRC_SQ_HDOUTN0	13		C	LRC_SQ_HDOUTN0	13		C
AK27	LRC_SQ_VCCOB0	13			LRC_SQ_VCCOB0	13		
AK26	LRC_SQ_HDOUTP0	13		T	LRC_SQ_HDOUTP0	13		T
AH27	LRC_SQ_VCCTX0	13			LRC_SQ_VCCTX0	13		

LFE2M50E/SE and LFE2M70E/SE Logic Signal Connections: 900 fpBGA

LFE2M50E/SE					LFE2M70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
AJ29	LRC_SQ_HDINN0	13		C	LRC_SQ_HDINN0	13		C
AJ30	LRC_SQ_VCCIB0	13			LRC_SQ_VCCIB0	13		
AK29	LRC_SQ_HDINP0	13		T	LRC_SQ_HDINP0	13		T
AH30	LRC_SQ_VCCRX0	13			LRC_SQ_VCCRX0	13		
AG27	CFG2	8			CFG2	8		
AD25	CFG1	8			CFG1	8		
AG28	CFG0	8			CFG0	8		
AG30	PROGRAMN	8			PROGRAMN	8		
AG29	CCLK	8			CCLK	8		
AC24	INITN	8			INITN	8		
AF27	DONE	8			DONE	8		
GNDIO	GNDIO8	-			GNDIO8	-		
AF28	WRITEN***	8			WRITEN***	8		
AE26	CS1N***	8			CS1N***	8		
AB23	CSN***	8			CSN***	8		
AF29	D0/SPIFASTN***	8			D0/SPIFASTN***	8		
VCCIO	VCCIO8	8			VCCIO8	8		
AF30	D1***	8			D1***	8		
AD26	D2***	8			D2***	8		
AE29	D3***	8			D3***	8		
GNDIO	GNDIO8	-			GNDIO8	-		
AE30	D4***	8			D4***	8		
AD29	D5***	8			D5***	8		
AC25	D6***	8			D6***	8		
AD30	D7/SPID0***	8			D7/SPID0***	8		
VCCIO	VCCIO8	8			VCCIO8	8		
AA22	DI/CSSPI0N***	8			DI/CSSPI0N***	8		
AC26	DOUT/CSON/ CSSPI1N***	8			DOUT/CSON/ CSSPI1N***	8		
AA23	BUSY/SISPI***	8			BUSY/SISPI***	8		
AB22	RLM0_PLLCAP	3			RLM0_PLLCAP	3		
AC27	PR65B	3	RLM0_GDLLC_FB_A	C	PR85B	3	RLM0_GDLLC_FB_A/RDQ82	C
GNDIO	GNDIO3	-			GNDIO3	-		
AC28	PR65A	3	RLM0_GDLLT_FB_A	T	PR85A	3	RLM0_GDLLT_FB_A/RDQ82	T
AC29	PR64B	3	RLM0_GDLLC_IN_A**	C (LVDS)*	PR84B	3	RLM0_GDLLC_IN_A**/RDQ82	C (LVDS)*
AC30	PR64A	3	RLM0_GDLLT_IN_A**	T (LVDS)*	PR84A	3	RLM0_GDLLT_IN_A**/RDQ82	T (LVDS)*
AB30	PR63B	3	RLM0_GPLLC_IN_A**	C	PR83B	3	RLM0_GPLLC_IN_A**/RDQ82	C
VCCIO	VCCIO3	3			VCCIO3	3		
AA30	PR63A	3	RLM0_GPLLT_IN_A**	T	PR83A	3	RLM0_GPLLT_IN_A**/RDQ82	T
AB29	PR62B	3	RLM0_GPLLC_FB_A	C (LVDS)*	PR82B	3	RLM0_GPLLC_FB_A/RDQ82	C (LVDS)*
AB28	PR62A	3	RLM0_GPLLT_FB_A	T (LVDS)*	PR82A	3	RLM0_GPLLT_FB_A/RDQS82	T (LVDS)*
GNDIO	GNDIO3	-			GNDIO3	-		
Y22	PR60B	3		C	PR81B	3	RDQ82	C
Y23	PR60A	3		T	PR81A	3	RDQ82	T
AB26	NC	-			PR80B	3	RDQ82	C (LVDS)*
AB27	NC	-			PR80A	3	RDQ82	T (LVDS)*
-	-	-			VCCIO3	3		
Y24	NC	-			PR79B	3	RDQ82	C
Y25	NC	-			PR79A	3	RDQ82	T
AA29	NC	-			PR78B	3	RDQ82	C (LVDS)*

LFE2M50E/SE and LFE2M70E/SE Logic Signal Connections: 900 fpBGA

LFE2M50E/SE					LFE2M70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
Y28	NC	-			PR78A	3	RDQ82	T (LVDS)*
Y30	NC	-			PR76B	3	RDQ73	C
Y29	NC	-			PR76A	3	RDQ73	T
-	-	-			GNDIO3	-		
-	-	-			-	-		
W22	NC	-			PR75B	3	RDQ73	C (LVDS)*
V22	NC	-			PR75A	3	RDQ73	T (LVDS)*
Y27	NC	-			PR74B	3	RDQ73	C
-	-	-			VCCIO3	3		
Y26	NC	-			PR74A	3	RDQ73	T
W30	NC	-			PR73B	3	RDQ73	C (LVDS)*
W29	NC	-			PR73A	3	RDQS73	T (LVDS)*
-	-	-			GNDIO3	-		
W25	NC	-			PR72B	3	RDQ73	C
W26	NC	-			PR72A	3	RDQ73	T
U29	PR59B	3		C (LVDS)*	PR71B	3	RDQ73	C (LVDS)*
V29	PR59A	3		T (LVDS)*	PR71A	3	RDQ73	T (LVDS)*
VCCIO	VCCIO3	3			VCCIO3	3		
V30	PR58B	3		C	PR70B	3	RDQ73	C
U30	PR58A	3		T	PR70A	3	RDQ73	T
W27	PR57B	3		C (LVDS)*	PR69B	3	RDQ73	C (LVDS)*
W28	PR57A	3		T (LVDS)*	PR69A	3	RDQ73	T (LVDS)*
V24	PR55B	3	RDQ52	C	PR67B	3	RDQ64	C
V25	PR55A	3	RDQ52	T	PR67A	3	RDQ64	T
GNDIO	GNDIO3	-			GNDIO3	-		
U28	PR54B	3	RDQ52	C (LVDS)*	PR66B	3	RDQ64	C (LVDS)*
U27	PR54A	3	RDQ52	T (LVDS)*	PR66A	3	RDQ64	T (LVDS)*
U23	PR53B	3	RDQ52	C	PR65B	3	RDQ64	C
V23	PR53A	3	RDQ52	T	PR65A	3	RDQ64	T
VCCIO	VCCIO3	3			VCCIO3	3		
V26	PR52B	3	RDQ52	C (LVDS)*	PR64B	3	RDQ64	C (LVDS)*
U26	PR52A	3	RDQS52	T (LVDS)*	PR64A	3	RDQS64	T (LVDS)*
U25	PR51B	3	RDQ52	C	PR63B	3	RDQ64	C
GNDIO	GNDIO3	-			GNDIO3	-		
U24	PR51A	3	RDQ52	T	PR63A	3	RDQ64	T
T30	PR50B	3	RDQ52	C (LVDS)*	PR62B	3	RDQ64	C (LVDS)*
R30	PR50A	3	RDQ52	T (LVDS)*	PR62A	3	RDQ64	T (LVDS)*
T23	PR49B	3	RDQ52	C	PR61B	3	RDQ64	C
VCCIO	VCCIO3	3			VCCIO3	3		
T22	PR49A	3	RDQ52	T	PR61A	3	RDQ64	T
T29	PR48B	3	RDQ52	C (LVDS)*	PR60B	3	RDQ64	C (LVDS)*
T28	PR48A	3	RDQ52	T (LVDS)*	PR60A	3	RDQ64	T (LVDS)*
R23	PR46B	3	RLM3_SPLLC_FB_A	C	PR58B	3	RLM3_SPLLC_FB_A/RDQ55	C
GNDIO	GNDIO3	-			GNDIO3	-		
VCCIO	VCCIO3	3			-	-		
R22	PR46A	3	RLM3_SPLLT_FB_A	T	PR58A	3	RLM3_SPLLT_FB_A/RDQ55	T
P30	PR45B	3	RLM3_SPLLC_IN_A	C (LVDS)*	PR57B	3	RLM3_SPLLC_IN_A/RDQ55	C (LVDS)*
R29	PR45A	3	RLM3_SPLLT_IN_A	T (LVDS)*	PR57A	3	RLM3_SPLLT_IN_A/RDQ55	T (LVDS)*
T27	PR44B	3		C	PR56B	3	RDQ55	C

LFE2M50E/SE and LFE2M70E/SE Logic Signal Connections: 900 fpBGA

LFE2M50E/SE					LFE2M70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
-	-	-			VCCIO3	3		
T26	PR44A	3		T	PR56A	3	RDQ55	T
GNDIO	GNDIO3	-			GNDIO3	-		
N30	PR43B	3		C (LVDS)*	PR53B	3	RDQ55	C (LVDS)*
N29	PR43A	3		T (LVDS)*	PR53A	3	RDQ55	T (LVDS)*
VCCIO	VCCIO3	3			VCCIO3	3		
R27	PR42B	3	VREF2_3	C	PR52B	3	VREF2_3/RDQ55	C
R28	PR42A	3	VREF1_3	T	PR52A	3	VREF1_3/RDQ55	T
P29	PR41B	3	PCLKC3_0	C (LVDS)*	PR51B	3	PCLKC3_0/RDQ55	C (LVDS)*
P28	PR41A	3	PCLKT3_0	T (LVDS)*	PR51A	3	PCLKT3_0/RDQ55	T (LVDS)*
M30	PR39B	2	PCLKC2_0/RDQ36	C	PR49B	2	PCLKC2_0/RDQ46	C
M29	PR39A	2	PCLKT2_0/RDQ36	T	PR49A	2	PCLKT2_0/RDQ46	T
GNDIO	GNDIO2	-			GNDIO2	-		
P23	PR38B	2	RDQ36	C (LVDS)*	PR48B	2	RDQ46	C (LVDS)*
P24	PR38A	2	RDQ36	T (LVDS)*	PR48A	2	RDQ46	T (LVDS)*
R26	PR37B	2	RDQ36	C	PR47B	2	RDQ46	C
P27	PR37A	2	RDQ36	T	PR47A	2	RDQ46	T
VCCIO	VCCIO2	2			VCCIO2	2		
P25	PR36B	2	RDQ36	C (LVDS)*	PR46B	2	RDQ46	C (LVDS)*
P26	PR36A	2	RDQS36	T (LVDS)*	PR46A	2	RDQS46	T (LVDS)*
K30	PR35B	2	RDQ36	C	PR45B	2	RDQ46	C
GNDIO	GNDIO2	-			GNDIO2	-		
K29	PR35A	2	RDQ36	T	PR45A	2	RDQ46	T
N22	PR34B	2	RDQ36	C (LVDS)*	PR44B	2	RDQ46	C (LVDS)*
P22	PR34A	2	RDQ36	T (LVDS)*	PR44A	2	RDQ46	T (LVDS)*
J30	PR33B	2	RUM3_SPLLC_FB_A/RDQ36	C	PR43B	2	RUM3_SPLLC_FB_A/RDQ46	C
VCCIO	VCCIO2	2			VCCIO2	2		
J29	PR33A	2	RUM3_SPLLT_FB_A/RDQ36	T	PR43A	2	RUM3_SPLLT_FB_A/RDQ46	T
N24	PR32B	2	RUM3_SPLLC_IN_A/RDQ36	C (LVDS)*	PR42B	2	RUM3_SPLLC_IN_A/RDQ46	C (LVDS)*
N23	PR32A	2	RUM3_SPLLT_IN_A/RDQ36	T (LVDS)*	PR42A	2	RUM3_SPLLT_IN_A/RDQ46	T (LVDS)*
N25	PR30B	2	RDQ27	C	PR40B	2	RDQ37	C
N26	PR30A	2	RDQ27	T	PR40A	2	RDQ37	T
GNDIO	GNDIO2	-			GNDIO2	-		
M27	PR29B	2	RDQ27	C (LVDS)*	PR39B	2	RDQ37	C (LVDS)*
M28	PR29A	2	RDQ27	T (LVDS)*	PR39A	2	RDQ37	T (LVDS)*
H30	PR28B	2	RDQ27	C	PR38B	2	RDQ37	C
G30	PR28A	2	RDQ27	T	PR38A	2	RDQ37	T
VCCIO	VCCIO2	2			VCCIO2	2		
M25	PR27B	2	RDQ27	C (LVDS)*	PR37B	2	RDQ37	C (LVDS)*
M26	PR27A	2	RDQS27	T (LVDS)*	PR37A	2	RDQS37	T (LVDS)*
L30	PR26B	2	RDQ27	C	PR36B	2	RDQ37	C
GNDIO	GNDIO2	-			GNDIO2	-		
L29	PR26A	2	RDQ27	T	PR36A	2	RDQ37	T
L28	PR25B	2	RDQ27	C (LVDS)*	PR35B	2	RDQ37	C (LVDS)*
L27	PR25A	2	RDQ27	T (LVDS)*	PR35A	2	RDQ37	T (LVDS)*
H29	PR24B	2	RDQ27	C	PR34B	2	RDQ37	C
VCCIO	VCCIO2	2			VCCIO2	2		
G29	PR24A	2	RDQ27	T	PR34A	2	RDQ37	T
L22	PR23B	2	RDQ27	C (LVDS)*	PR33B	2	RDQ37	C (LVDS)*

LFE2M50E/SE and LFE2M70E/SE Logic Signal Connections: 900 fpBGA

LFE2M50E/SE					LFE2M70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
M22	PR23A	2	RDQ27	T (LVDS)*	PR33A	2	RDQ37	T (LVDS)*
F30	PR21B	2		C	PR31B	2	RDQ28	C
GNDIO	GNDIO2	-			GNDIO2	-		
F29	PR21A	2		T	PR31A	2	RDQ28	T
-	-	-			-	-		
-	-	-			-	-		
E30	PR20B	2		C (LVDS)*	PR30B	2	RDQ28	C (LVDS)*
E29	PR20A	2		T (LVDS)*	PR30A	2	RDQ28	T (LVDS)*
VCCIO	VCCIO2	2			-	-		
L25	PR19B	2		C	PR29B	2	RDQ28	C
L26	PR19A	2		T	PR29A	2	RDQ28	T
-	-	-			VCCIO2	2		
H28	PR18B	2		C (LVDS)*	PR28B	2	RDQ28	C (LVDS)*
J28	PR18A	2		T (LVDS)*	PR28A	2	RDQS28	T (LVDS)*
G28	PR16B	2		C	PR27B	2	RDQ28	C
GNDIO	GNDIO2	-			GNDIO2	-		
G27	PR16A	2		T	PR27A	2	RDQ28	T
L24	NC	-			PR26B	2	RDQ28	C (LVDS)*
L23	NC	-			PR26A	2	RDQ28	T (LVDS)*
D30	NC	-			PR25B	2	RDQ28	C
-	-	-			VCCIO2	2		
D29	NC	-			PR25A	2	RDQ28	T
K24	NC	-			PR24B	2	RDQ28	C (LVDS)*
K25	NC	-			PR24A	2	RDQ28	T (LVDS)*
J27	NC	-			PR22B	2		C
-	-	-			GNDIO2	-		
K26	NC	-			PR22A	2		T
K23	PR15B	2		C (LVDS)*	PR21B	2		C (LVDS)*
K22	PR15A	2		T (LVDS)*	PR21A	2		T (LVDS)*
J22	PR14B	2		C	PR20B	2		C
VCCIO	VCCIO2	-			VCCIO2	2		
J23	PR14A	2		T	PR20A	2		T
-	-	-			GNDIO2	-		
-	-	-			-	-		
J26	NC	-			PR17B	2	RDQ15	C (LVDS)*
H26	NC	-			PR17A	2	RDQ15	T (LVDS)*
H27	NC	-			PR16B	2	RDQ15	C
G26	NC	-			PR16A	2	RDQ15	T
-	-	-			VCCIO2	2		
H23	NC	-			PR15B	2	RDQ15	C (LVDS)*
H24	NC	-			PR15A	2	RDQS15	T (LVDS)*
D28	NC	-			PR14B	2	RDQ15	C
-	-	-			GNDIO2	-		
E28	NC	-			PR14A	2	RDQ15	T
G24	PR13B	2		C (LVDS)*	PR13B	2	RDQ15	C (LVDS)*
H25	PR13A	2		T (LVDS)*	PR13A	2	RDQ15	T (LVDS)*
D27	PR12B	2	RUM0_SPLLC_FB_A	C	PR12B	2	RUM0_SPLLC_FB_A/RDQ15	C
GNDIO	GNDIO2	-			VCCIO2	2		
E27	PR12A	2	RUM0_SPLLT_FB_A	T	PR12A	2	RUM0_SPLLT_FB_A/RDQ15	T

LFE2M50E/SE and LFE2M70E/SE Logic Signal Connections: 900 fpBGA

LFE2M50E/SE					LFE2M70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
F26	PR11B	2	RUM0_SPLLC_IN_A	C (LVDS)*	PR11B	2	RUM0_SPLLC_IN_A/RDQ15	C (LVDS)*
G25	PR11A	2	RUM0_SPLLT_IN_A	T (LVDS)*	PR11A	2	RUM0_SPLLT_IN_A/RDQ15	T (LVDS)*
F24	PR9B	2	VREF2_2	C	PR9B	2	VREF2_2	C
VCCIO	VCCIO2	-			-	-		
GNDIO	GNDIO2	-			GNDIO2	-		
F25	PR9A	2	VREF1_2	T	PR9A	2	VREF1_2	T
VCCIO	VCCIO2	2			VCCIO2	2		
G23	XRES	-			XRES	1		
C30	URC_SQ_VCCRX0	12			URC_SQ_VCCRX0	12		
A29	URC_SQ_HDINP0	12		T	URC_SQ_HDINP0	12		T
B30	URC_SQ_VCCIB0	12			URC_SQ_VCCIB0	12		
B29	URC_SQ_HDINN0	12		C	URC_SQ_HDINN0	12		C
C27	URC_SQ_VCCTX0	12			URC_SQ_VCCTX0	12		
A26	URC_SQ_HDOUTP0	12		T	URC_SQ_HDOUTP0	12		T
A27	URC_SQ_VCCOB0	12			URC_SQ_VCCOB0	12		
B26	URC_SQ_HDOUTN0	12		C	URC_SQ_HDOUTN0	12		C
C26	URC_SQ_VCCTX1	12			URC_SQ_VCCTX1	12		
B25	URC_SQ_HDOUTN1	12		C	URC_SQ_HDOUTN1	12		C
C25	URC_SQ_VCCOB1	12			URC_SQ_VCCOB1	12		
A25	URC_SQ_HDOUTP1	12		T	URC_SQ_HDOUTP1	12		T
C29	URC_SQ_VCCRX1	12			URC_SQ_VCCRX1	12		
B28	URC_SQ_HDINN1	12		C	URC_SQ_HDINN1	12		C
C28	URC_SQ_VCCIB1	12			URC_SQ_VCCIB1	12		
A28	URC_SQ_HDINP1	12		T	URC_SQ_HDINP1	12		T
B24	URC_SQ_VCCAUX33	12			URC_SQ_VCCAUX33	12		
E24	URC_SQ_REFCLKN	12		C	URC_SQ_REFCLKN	12		C
D24	URC_SQ_REFCLKP	12		T	URC_SQ_REFCLKP	12		T
C24	URC_SQ_VCCP	12			URC_SQ_VCCP	12		
A20	URC_SQ_HDINP2	12		T	URC_SQ_HDINP2	12		T
C20	URC_SQ_VCCIB2	12			URC_SQ_VCCIB2	12		
B20	URC_SQ_HDINN2	12		C	URC_SQ_HDINN2	12		C
C19	URC_SQ_VCCRX2	12			URC_SQ_VCCRX2	12		
A23	URC_SQ_HDOUTP2	12		T	URC_SQ_HDOUTP2	12		T
C23	URC_SQ_VCCOB2	12			URC_SQ_VCCOB2	12		
B23	URC_SQ_HDOUTN2	12		C	URC_SQ_HDOUTN2	12		C
C22	URC_SQ_VCCTX2	12			URC_SQ_VCCTX2	12		
B22	URC_SQ_HDOUTN3	12		C	URC_SQ_HDOUTN3	12		C
A21	URC_SQ_VCCOB3	12			URC_SQ_VCCOB3	12		
A22	URC_SQ_HDOUTP3	12		T	URC_SQ_HDOUTP3	12		T
C21	URC_SQ_VCCTX3	12			URC_SQ_VCCTX3	12		
B19	URC_SQ_HDINN3	12		C	URC_SQ_HDINN3	12		C
B18	URC_SQ_VCCIB3	12			URC_SQ_VCCIB3	12		
A19	URC_SQ_HDINP3	12		T	URC_SQ_HDINP3	12		T
C18	URC_SQ_VCCRX3	12			URC_SQ_VCCRX3	12		
D23	PT73B	1		C	PT82B	1		C
GNDIO	GNDIO1	-			GNDIO1	-		
E21	PT73A	1		T	PT82A	1		T
D26	PT72B	1		C	PT81B	1		C
E26	PT72A	1		T	PT81A	1		T

LFE2M50E/SE and LFE2M70E/SE Logic Signal Connections: 900 fpBGA

LFE2M50E/SE					LFE2M70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
E23	PT71B	1		C	PT80B	1		C
-	-	-			VCCIO1	1		
G22	PT71A	1		T	PT80A	1		T
VCCIO	VCCIO1	1			-	-		
D22	PT70B	1		C	PT79B	1		C
F21	PT70A	1		T	PT79A	1		T
G18	PT69B	1		C	PT78B	1		C
H18	PT69A	1		T	PT78A	1		T
D20	PT68B	1		C	PT77B	1		C
GNDIO	GNDIO1	-			GNDIO1	-		
D21	PT68A	1		T	PT77A	1		T
E20	PT67B	1		C	PT76B	1		C
E19	PT67A	1		T	PT76A	1		T
D19	PT66B	1		C	PT75B	1		C
VCCIO	VCCIO1	1			VCCIO1	1		
E18	PT66A	1		T	PT75A	1		T
D18	PT65B	1		C	PT74B	1		C
C17	PT65A	1		T	PT74A	1		T
A17	PT64B	1		C	PT73B	1		C
B17	PT64A	1		T	PT73A	1		T
GNDIO	GNDIO1	-			GNDIO1	-		
VCCIO	VCCIO1	1			VCCIO1	1		
J18	NC	-			PT66B	1		C
J19	NC	-			PT66A	1		T
H17	NC	-			PT65B	1		C
J17	NC	-			PT65A	1		T
F18	NC	-			PT64B	1		C
F17	NC	-			PT64A	1		T
-	-	-			GNDIO1	-		
A16	PT54B	1		C	PT63B	1		C
B16	PT54A	1		T	PT63A	1		T
G17	PT53B	1		C	PT62B	1		C
G16	PT53A	1		T	PT62A	1		T
VCCIO	VCCIO1	1			VCCIO1	1		
H16	PT52B	1		C	PT61B	1		C
F16	PT52A	1		T	PT61A	1		T
J16	PT51B	1		C	PT60B	1		C
G15	PT51A	1		T	PT60A	1		T
GNDIO	GNDIO1	-			GNDIO1	-		
C16	PT50B	1		C	PT59B	1		C
D16	PT50A	1		T	PT59A	1		T
J15	PT49B	1		C	PT58B	1		C
H15	PT49A	1		T	PT58A	1		T
VCCIO	VCCIO1	1			VCCIO1	1		
A15	PT48B	1	VREF2_1	C	PT57B	1	VREF2_1	C
B15	PT48A	1	VREF1_1	T	PT57A	1	VREF1_1	T
F15	PT47B	1	PCLKC1_0	C	PT56B	1	PCLKC1_0	C
E16	PT47A	1	PCLKT1_0	T	PT56A	1	PCLKT1_0	T
C15	PT46B	0	PCLKC0_0	C	PT55B	0	PCLKC0_0	C

LFE2M50E/SE and LFE2M70E/SE Logic Signal Connections: 900 fpBGA

LFE2M50E/SE					LFE2M70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
GNDIO	GNDIO0	-			GNDIO0	-		
D15	PT46A	0	PCLKT0_0	T	PT55A	0	PCLKT0_0	T
C14	PT45B	0	VREF2_0	C	PT54B	0	VREF2_0	C
E15	PT45A	0	VREF1_0	T	PT54A	0	VREF1_0	T
G14	PT44B	0		C	PT53B	0		C
VCCIO	VCCIO0	0			VCCIO0	0		
J14	PT44A	0		T	PT53A	0		T
F14	PT43B	0		C	PT52B	0		C
H14	PT43A	0		T	PT52A	0		T
A14	PT42B	0		C	PT51B	0		C
B14	PT42A	0		T	PT51A	0		T
D13	PT41B	0		C	PT50B	0		C
GNDIO	GNDIO0	-			GNDIO0	-		
F13	PT41A	0		T	PT50A	0		T
G13	PT40B	0		C	PT49B	0		C
VCCIO	VCCIO0	0			VCCIO0	0		
J11	PT40A	0		T	PT49A	0		T
D4	PT38B	0		C	PT47B	0		C
D5	PT38A	0		T	PT47A	0		T
E5	PT37B	0		C	PT46B	0		C
F6	PT37A	0		T	PT46A	0		T
GNDIO	GNDIO0	-			GNDIO0	-		
VCCIO	VCCIO0	0			VCCIO0	0		
F7	PT34B	0		C	PT43B	0		C
D8	PT34A	0		T	PT43A	0		T
GNDIO	GNDIO0	-			GNDIO0	-		
J13	PT32B	0		C	PT41B	0		C
G11	PT32A	0		T	PT41A	0		T
H13	PT31B	0		C	PT40B	0		C
H12	PT31A	0		T	PT40A	0		T
VCCIO	VCCIO0	0			VCCIO0	0		
E8	PT30B	0		C	PT39B	0		C
D9	PT30A	0		T	PT39A	0		T
D12	PT28B	0		C	PT37B	0		C
GNDIO	GNDIO0	-			GNDIO0	-		
E13	PT28A	0		T	PT37A	0		T
VCCIO	VCCIO0	0			VCCIO0	0		
GNDIO	GNDIO0	-			GNDIO0	-		
J12	PT5B	0		C	PT31B	0		C
GNDIO	GNDIO0	-			-	-		
VCCIO	VCCIO0	0			VCCIO0	0		
H10	PT5A	0		T	PT31A	0		T
E12	PT4B	0		C	PT30B	0		C
D11	PT4A	0		T	PT30A	0		T
H11	PT3B	0		C	PT29B	0		C
F11	PT3A	0		T	PT29A	0		T
C13	VCC	-			ULC_SQ_VCCRX0	11		
A12	PT19A	0		T	ULC_SQ_HDINP0	11		T
B13	NC	-			ULC_SQ_VCCIB0	11		

LFE2M50E/SE and LFE2M70E/SE Logic Signal Connections: 900 fpBGA

LFE2M50E/SE					LFE2M70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
B12	PT19B	0		C	ULC_SQ_HDINN0	11		C
C10	VCC	-			ULC_SQ_VCCTX0	11		
A9	PT17A	0		T	ULC_SQ_HDOUTP0	11		T
A10	NC	-			ULC_SQ_VCCOB0	11		
B9	PT17B	0		C	ULC_SQ_HDOUTN0	11		C
C9	VCC	-			ULC_SQ_VCCTX1	11		
B8	PT18B	0		C	ULC_SQ_HDOUTN1	11		C
C8	NC	-			ULC_SQ_VCCOB1	11		
A8	PT18A	0		T	ULC_SQ_HDOUTP1	11		T
C12	VCC	-			ULC_SQ_VCCRX1	11		
B11	PT16B	0		C	ULC_SQ_HDINN1	11		C
C11	NC	-			ULC_SQ_VCCIB1	11		
A11	PT16A	0		T	ULC_SQ_HDINP1	11		T
B7	VCCAUX	-			ULC_SQ_VCCAUX33	11		
E7	PT15B	0		C	ULC_SQ_REFCLKN	11		C
D7	PT15A	0		T	ULC_SQ_REFCLKP	11		T
C7	VCC	-			ULC_SQ_VCCP	11		
A3	PT12A	0		T	ULC_SQ_HDINP2	11		T
C3	NC	-			ULC_SQ_VCCIB2	11		
B3	PT12B	0		C	ULC_SQ_HDINN2	11		C
C2	VCC	-			ULC_SQ_VCCRX2	11		
A6	PT14A	0		T	ULC_SQ_HDOUTP2	11		T
C6	NC	-			ULC_SQ_VCCOB2	11		
B6	PT14B	0		C	ULC_SQ_HDOUTN2	11		C
C5	VCC	-			ULC_SQ_VCCTX2	11		
B5	PT13B	0		C	ULC_SQ_HDOUTN3	11		C
A4	NC	-			ULC_SQ_VCCOB3	11		
A5	PT13A	0		T	ULC_SQ_HDOUTP3	11		T
C4	VCC	-			ULC_SQ_VCCTX3	11		
B2	PT11B	0		C	ULC_SQ_HDINN3	11		C
B1	NC	-			ULC_SQ_VCCIB3	11		
A2	PT11A	0		T	ULC_SQ_HDINP3	11		T
C1	VCC	-			ULC_SQ_VCCRX3	11		
L12	VCC	-			VCC	-		
L13	VCC	-			VCC	-		
L18	VCC	-			VCC	-		
L19	VCC	-			VCC	-		
M11	VCC	-			VCC	-		
M12	VCC	-			VCC	-		
M13	VCC	-			VCC	-		
M14	VCC	-			VCC	-		
M15	VCC	-			VCC	-		
M16	VCC	-			VCC	-		
M17	VCC	-			VCC	-		
M18	VCC	-			VCC	-		
M19	VCC	-			VCC	-		
M20	VCC	-			VCC	-		
N11	VCC	-			VCC	-		
N12	VCC	-			VCC	-		

LFE2M50E/SE and LFE2M70E/SE Logic Signal Connections: 900 fpBGA

LFE2M50E/SE					LFE2M70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
N19	VCC	-			VCC	-		
N20	VCC	-			VCC	-		
P12	VCC	-			VCC	-		
P19	VCC	-			VCC	-		
R12	VCC	-			VCC	-		
R19	VCC	-			VCC	-		
T12	VCC	-			VCC	-		
T19	VCC	-			VCC	-		
U12	VCC	-			VCC	-		
U19	VCC	-			VCC	-		
V11	VCC	-			VCC	-		
V12	VCC	-			VCC	-		
V19	VCC	-			VCC	-		
V20	VCC	-			VCC	-		
W11	VCC	-			VCC	-		
W12	VCC	-			VCC	-		
W13	VCC	-			VCC	-		
W14	VCC	-			VCC	-		
W15	VCC	-			VCC	-		
W16	VCC	-			VCC	-		
W17	VCC	-			VCC	-		
W18	VCC	-			VCC	-		
W19	VCC	-			VCC	-		
W20	VCC	-			VCC	-		
Y12	VCC	-			VCC	-		
Y13	VCC	-			VCC	-		
Y18	VCC	-			VCC	-		
Y19	VCC	-			VCC	-		
D14	VCCIO0	0			VCCIO0	0		
E6	VCCIO0	0			VCCIO0	0		
E9	VCCIO0	0			VCCIO0	0		
F12	VCCIO0	0			VCCIO0	0		
K12	VCCIO0	0			VCCIO0	0		
K13	VCCIO0	0			VCCIO0	0		
D17	VCCIO1	1			VCCIO1	1		
E22	VCCIO1	1			VCCIO1	1		
E25	VCCIO1	1			VCCIO1	1		
F19	VCCIO1	1			VCCIO1	1		
K18	VCCIO1	1			VCCIO1	1		
K19	VCCIO1	1			VCCIO1	1		
F28	VCCIO2	2			VCCIO2	2		
J25	VCCIO2	2			VCCIO2	2		
K28	VCCIO2	2			VCCIO2	2		
M21	VCCIO2	2			VCCIO2	2		
M24	VCCIO2	2			VCCIO2	2		
N21	VCCIO2	2			VCCIO2	2		
N28	VCCIO2	2			VCCIO2	2		
P21	VCCIO2	2			VCCIO2	2		
R25	VCCIO2	2			VCCIO2	2		

LFE2M50E/SE and LFE2M70E/SE Logic Signal Connections: 900 fpBGA

LFE2M50E/SE					LFE2M70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
AA28	VCCIO3	3			VCCIO3	3		
AB25	VCCIO3	3			VCCIO3	3		
AE28	VCCIO3	3			VCCIO3	3		
T25	VCCIO3	3			VCCIO3	3		
U21	VCCIO3	3			VCCIO3	3		
V21	VCCIO3	3			VCCIO3	3		
V28	VCCIO3	3			VCCIO3	3		
W21	VCCIO3	3			VCCIO3	3		
W24	VCCIO3	3			VCCIO3	3		
AA18	VCCIO4	4			VCCIO4	4		
AA19	VCCIO4	4			VCCIO4	4		
AE19	VCCIO4	4			VCCIO4	4		
AF22	VCCIO4	4			VCCIO4	4		
AG17	VCCIO4	4			VCCIO4	4		
AG25	VCCIO4	4			VCCIO4	4		
AA12	VCCIO5	5			VCCIO5	5		
AA13	VCCIO5	5			VCCIO5	5		
AE12	VCCIO5	5			VCCIO5	5		
AF9	VCCIO5	5			VCCIO5	5		
AG14	VCCIO5	5			VCCIO5	5		
AG6	VCCIO5	5			VCCIO5	5		
AA3	VCCIO6	6			VCCIO6	6		
AB6	VCCIO6	6			VCCIO6	6		
AE3	VCCIO6	6			VCCIO6	6		
T6	VCCIO6	6			VCCIO6	6		
U10	VCCIO6	6			VCCIO6	6		
V10	VCCIO6	6			VCCIO6	6		
V3	VCCIO6	6			VCCIO6	6		
W10	VCCIO6	6			VCCIO6	6		
W7	VCCIO6	6			VCCIO6	6		
F3	VCCIO7	7			VCCIO7	7		
J6	VCCIO7	7			VCCIO7	7		
K3	VCCIO7	7			VCCIO7	7		
M10	VCCIO7	7			VCCIO7	7		
M7	VCCIO7	7			VCCIO7	7		
N10	VCCIO7	7			VCCIO7	7		
N3	VCCIO7	7			VCCIO7	7		
P10	VCCIO7	7			VCCIO7	7		
R6	VCCIO7	7			VCCIO7	7		
AA25	VCCIO8	8			VCCIO8	8		
AD28	VCCIO8	8			VCCIO8	8		
AA10	VCCAUX	-			VCCAUX	-		
AA11	VCCAUX	-			VCCAUX	-		
AA20	VCCAUX	-			VCCAUX	-		
AA21	VCCAUX	-			VCCAUX	-		
K10	VCCAUX	-			VCCAUX	-		
K11	VCCAUX	-			VCCAUX	-		
K20	VCCAUX	-			VCCAUX	-		
K21	VCCAUX	-			VCCAUX	-		

LFE2M50E/SE and LFE2M70E/SE Logic Signal Connections: 900 fpBGA

LFE2M50E/SE					LFE2M70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
L10	VCCAUX	-			VCCAUX	-		
L11	VCCAUX	-			VCCAUX	-		
L20	VCCAUX	-			VCCAUX	-		
L21	VCCAUX	-			VCCAUX	-		
Y10	VCCAUX	-			VCCAUX	-		
Y11	VCCAUX	-			VCCAUX	-		
Y20	VCCAUX	-			VCCAUX	-		
Y21	VCCAUX	-			VCCAUX	-		
A1	GND	-			GND	-		
A13	GND	-			GND	-		
A18	GND	-			GND	-		
A24	GND	-			GND	-		
A30	GND	-			GND	-		
A7	GND	-			GND	-		
AA14	GND	-			GND	-		
AA15	GND	-			GND	-		
AA16	GND	-			GND	-		
AA17	GND	-			GND	-		
AA24	GND	-			GND	-		
AA27	GND	-			GND	-		
AA4	GND	-			GND	-		
AB24	GND	-			GND	-		
AB7	GND	-			GND	-		
AD12	GND	-			GND	-		
AD19	GND	-			GND	-		
AD27	GND	-			GND	-		
AE22	GND	-			GND	-		
AE27	GND	-			GND	-		
AE4	GND	-			GND	-		
AE9	GND	-			GND	-		
AF14	GND	-			GND	-		
AF17	GND	-			GND	-		
AF25	GND	-			GND	-		
AF6	GND	-			GND	-		
AJ10	GND	-			GND	-		
AJ21	GND	-			GND	-		
AJ27	GND	-			GND	-		
AJ4	GND	-			GND	-		
AK1	GND	-			GND	-		
AK13	GND	-			GND	-		
AK18	GND	-			GND	-		
AK24	GND	-			GND	-		
AK30	GND	-			GND	-		
AK7	GND	-			GND	-		
B10	GND	-			GND	-		
B21	GND	-			GND	-		
B27	GND	-			GND	-		
B4	GND	-			GND	-		
D25	GND	-			GND	-		

LFE2M50E/SE and LFE2M70E/SE Logic Signal Connections: 900 fpBGA

LFE2M50E/SE					LFE2M70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
D6	GND	-			GND	-		
E14	GND	-			GND	-		
E17	GND	-			GND	-		
F22	GND	-			GND	-		
F27	GND	-			GND	-		
F4	GND	-			GND	-		
F9	GND	-			GND	-		
G12	GND	-			GND	-		
G19	GND	-			GND	-		
J24	GND	-			GND	-		
J7	GND	-			GND	-		
K14	GND	-			GND	-		
K15	GND	-			GND	-		
K16	GND	-			GND	-		
K17	GND	-			GND	-		
K27	GND	-			GND	-		
K4	GND	-			GND	-		
L14	GND	-			GND	-		
L15	GND	-			GND	-		
L16	GND	-			GND	-		
L17	GND	-			GND	-		
M23	GND	-			GND	-		
M8	GND	-			GND	-		
N14	GND	-			GND	-		
N15	GND	-			GND	-		
N16	GND	-			GND	-		
N17	GND	-			GND	-		
N27	GND	-			GND	-		
N4	GND	-			GND	-		
P11	GND	-			GND	-		
P13	GND	-			GND	-		
P14	GND	-			GND	-		
P15	GND	-			GND	-		
P16	GND	-			GND	-		
P17	GND	-			GND	-		
P18	GND	-			GND	-		
P20	GND	-			GND	-		
R10	GND	-			GND	-		
R11	GND	-			GND	-		
R13	GND	-			GND	-		
R14	GND	-			GND	-		
R15	GND	-			GND	-		
R16	GND	-			GND	-		
R17	GND	-			GND	-		
R18	GND	-			GND	-		
R20	GND	-			GND	-		
R21	GND	-			GND	-		
R24	GND	-			GND	-		
R7	GND	-			GND	-		

LFE2M50E/SE and LFE2M70E/SE Logic Signal Connections: 900 fpBGA

LFE2M50E/SE					LFE2M70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
T10	GND	-			GND	-		
T11	GND	-			GND	-		
T13	GND	-			GND	-		
T14	GND	-			GND	-		
T15	GND	-			GND	-		
T16	GND	-			GND	-		
T17	GND	-			GND	-		
T18	GND	-			GND	-		
T20	GND	-			GND	-		
T21	GND	-			GND	-		
T24	GND	-			GND	-		
T7	GND	-			GND	-		
U11	GND	-			GND	-		
U13	GND	-			GND	-		
U14	GND	-			GND	-		
U15	GND	-			GND	-		
U16	GND	-			GND	-		
U17	GND	-			GND	-		
U18	GND	-			GND	-		
U20	GND	-			GND	-		
V14	GND	-			GND	-		
V15	GND	-			GND	-		
V16	GND	-			GND	-		
V17	GND	-			GND	-		
V27	GND	-			GND	-		
V4	GND	-			GND	-		
W23	GND	-			GND	-		
W8	GND	-			GND	-		
Y14	GND	-			GND	-		
Y15	GND	-			GND	-		
Y16	GND	-			GND	-		
Y17	GND	-			GND	-		
AA26	NC	-			NC	-		
AB10	PL73B	6	LDQ71	C (LVDS)*	NC	-		
AB11	NC	-			NC	-		
AB12	NC	-			NC	-		
AB13	NC	-			NC	-		
AB14	NC	-			NC	-		
AB15	NC	-			NC	-		
AB16	NC	-			NC	-		
AB17	NC	-			NC	-		
AB19	NC	-			NC	-		
AB20	NC	-			NC	-		
AB21	NC	-			NC	-		
AB9	PL73A	6	LDQ71	T (LVDS)*	NC	-		
AC10	PL74B	6	LDQ71	C	NC	-		
AC11	NC	-			NC	-		
AC21	NC	-			NC	-		
AC22	NC	-			NC	-		

LFE2M50E/SE and LFE2M70E/SE Logic Signal Connections: 900 fpBGA

LFE2M50E/SE					LFE2M70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
AC8	PL70B	6	LDQ71	C	NC	-		
AC9	PL74A	6	LDQ71	T	NC	-		
AD21	NC	-			NC	-		
AD22	NC	-			NC	-		
AD4	PL68A	6	LDQ71	T	NC	-		
AD5	PL68B	6	LDQ71	C	NC	-		
AD6	PL71A	6	LDQS71	T (LVDS)*	NC	-		
AD7	PL72A	6	LDQ71	T	NC	-		
AD8	PL72B	6	LDQ71	C	NC	-		
AE23	NC	-			NC	-		
AE5	PL69A	6	LDQ71	T (LVDS)*	NC	-		
AE6	PL70A	6	LDQ71	T	NC	-		
AE7	PL71B	6	LDQ71	C (LVDS)*	NC	-		
AF20	NC	-			NC	-		
AF23	NC	-			NC	-		
AF5	PL69B	6	LDQ71	C (LVDS)*	NC	-		
AG23	NC	-			NC	-		
AG26	NC	-			NC	-		
D10	PT10A	0		T	NC	-		
E10	PT9B	0		C	NC	-		
E11	PT10B	0		C	NC	-		
F10	PT9A	0		T	NC	-		
F20	NC	-			NC	-		
F23	NC	-			NC	-		
F8	PL6B	7	LDQ6	C (LVDS)*	NC	-		
G10	NC	-			NC	-		
G20	NC	-			NC	-		
G21	NC	-			NC	-		
G7	PL8A	7	LDQ6	T (LVDS)*	NC	-		
G8	PL6A	7	LDQS6****	T (LVDS)*	NC	-		
G9	PL5A	7	LDQ6	T	NC	-		
H19	NC	-			NC	-		
H20	NC	-			NC	-		
H21	NC	-			NC	-		
H22	NC	-			NC	-		
H6	PL8B	7	LDQ6	C (LVDS)*	NC	-		
H8	PL5B	7	LDQ6	C	NC	-		
H9	PL2A	7	LDQ6	T (LVDS)*	NC	-		
J10	PL2B	7	LDQ6	C (LVDS)*	NC	-		
J20	NC	-			NC	-		
J21	NC	-			NC	-		
J9	PL4A	7	LDQ6	T (LVDS)*	NC	-		
K9	PL4B	7	LDQ6	C (LVDS)*	NC	-		
R9	NC	-			NC	-		
U22	NC	-			NC	-		
W9	NC	-			NC	-		
N13	VCCPLL	-			VCCPLL	-		
N18	VCCPLL	-			VCCPLL	-		
V13	VCCPLL	-			VCCPLL	-		

LFE2M50E/SE and LFE2M70E/SE Logic Signal Connections: 900 fpBGA

LFE2M50E/SE					LFE2M70E/SE			
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
V18	VCCPLL	-			VCCPLL	-		

* Supports true LVDS. Other differential signals must be emulated with external resistors.
** These dedicated input pins can be used for GPLLs or GDLLs within the respective quadrant.
*** These sysCONFIG pins are dedicated I/O pins for configuration. The outpus are actively driven during normal device operation.
****Due to packaging bond out option, this DQS does not have all the necessary DQ pins bonded out for a full 8-bit data width.

Note: VCCIO and GND pads are used to determine the average DC current drawn by I/Os between GND/VCCIO connections, or between the last GND/VCCIO in an I/O bank and the end of an I/O bank. The substrate pads listed in the Pin Table do not necessarily have a one to one connection with a package ball or pin.

LFE2M100E/SE Logic Signal Connections: 900 fpBGA

LFE2M100E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
D2	PL9A	7	VREF2_7	T
D3	PL9B	7	VREF1_7	C
GNDIO	GNDIO7	-		
J8	PL11A	7	LUM0_SPLLT_IN_A/LDQ15	T (LVDS)*
H7	PL11B	7	LUM0_SPLLC_IN_A/LDQ15	C (LVDS)*
E3	PL12A	7	LUM0_SPLLT_FB_A/LDQ15	T
E4	PL12B	7	LUM0_SPLLC_FB_A/LDQ15	C
G6	PL13A	7	LDQ15	T (LVDS)*
F5	PL13B	7	LDQ15	C (LVDS)*
E2	PL14A	7	LDQ15	T
D1	PL14B	7	LDQ15	C
GNDIO	GNDIO7	-		
G5	PL15A	7	LDQS15	T (LVDS)*
G4	PL15B	7	LDQ15	C (LVDS)*
K7	PL16A	7	LDQ15	T
K8	PL16B	7	LDQ15	C
E1	PL17A	7	LDQ15	T (LVDS)*
F2	PL17B	7	LDQ15	C (LVDS)*
F1	PL18A	7	LDQ15	T
GNDIO	GNDIO7	-		
G3	PL18B	7	LDQ15	C
GNDIO	GNDIO7	-		
H5	PL25A	7	LDQ23	T (LVDS)*
H4	PL25B	7	LDQ23	C (LVDS)*
J5	PL26A	7	LDQ23	T
J4	PL26B	7	LDQ23	C
GNDIO	GNDIO7	-		
G2	PL28A	7	LDQ32	T (LVDS)*
G1	PL28B	7	LDQ32	C (LVDS)*
L9	PL29A	7	LDQ32	T
L7	PL29B	7	LDQ32	C
K6	PL30A	7	LDQ32	T (LVDS)*
K5	PL30B	7	LDQ32	C (LVDS)*
L8	PL31A	7	LDQ32	T
L6	PL31B	7	LDQ32	C
GNDIO	GNDIO7	-		
H3	PL32A	7	LDQS32	T (LVDS)*
H2	PL32B	7	LDQ32	C (LVDS)*
N8	PL33A	7	LDQ32	T
M9	PL33B	7	LDQ32	C
J3	PL34A	7	LDQ32	T (LVDS)*
-	-	-		

LFE2M100E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2M100E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
J2	PL34B	7	LDQ32	C (LVDS)*
H1	PL35A	7	LDQ32	T
GNDIO	GNDIO7	-		
J1	PL35B	7	LDQ32	C
GNDIO	GNDIO7	-		
L5	PL41A	7	LDQ45	T (LVDS)*
L4	PL41B	7	LDQ45	C (LVDS)*
N9	PL42A	7	LDQ45	T
N7	PL42B	7	LDQ45	C
K2	PL43A	7	LDQ45	T (LVDS)*
K1	PL43B	7	LDQ45	C (LVDS)*
P9	PL44A	7	LDQ45	T
P7	PL44B	7	LDQ45	C
GNDIO	GNDIO7	-		
M6	PL45A	7	LDQS45	T (LVDS)*
M5	PL45B	7	LDQ45	C (LVDS)*
N5	PL46A	7	LDQ45	T
N6	PL46B	7	LDQ45	C
M4	PL47A	7	LDQ45	T (LVDS)*
M3	PL47B	7	LDQ45	C (LVDS)*
P6	PL48A	7	LDQ45	T
GNDIO	GNDIO7	-		
P8	PL48B	7	LDQ45	C
L3	PL50A	7	LUM3_SPLLT_IN_A/LDQ54	T (LVDS)*
L2	PL50B	7	LUM3_SPLLC_IN_A/LDQ54	C (LVDS)*
P5	PL51A	7	LUM3_SPLLT_FB_A/LDQ54	T
P4	PL51B	7	LUM3_SPLLC_FB_A/LDQ54	C
L1	PL52A	7	LDQ54	T (LVDS)*
M2	PL52B	7	LDQ54	C (LVDS)*
R5	PL53A	7	LDQ54	T
R4	PL53B	7	LDQ54	C
GNDIO	GNDIO7	-		
M1	PL54A	7	LDQS54	T (LVDS)*
N2	PL54B	7	LDQ54	C (LVDS)*
R8	PL55A	7	LDQ54	T
T9	PL55B	7	LDQ54	C
P3	PL56A	7	LDQ54	T (LVDS)*
P2	PL56B	7	LDQ54	C (LVDS)*
N1	PL57A	7	PCLKT7_0/LDQ54	T
GNDIO	GNDIO7	-		
P1	PL57B	7	PCLKC7_0/LDQ54	C
T5	PL59A	6	PCLKT6_0/LDQ63	T (LVDS)*
T4	PL59B	6	PCLKC6_0/LDQ63	C (LVDS)*

LFE2M100E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2M100E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
U7	PL60A	6	VREF2_6/LDQ63	T
T8	PL60B	6	VREF1_6/LDQ63	C
R3	PL61A	6	LDQ63	T (LVDS)*
VCCIO	VCCIO6	6		
R2	PL61B	6	LDQ63	C (LVDS)*
R1	PL62A	6	LDQ63	T
T1	PL62B	6	LDQ63	C
GNDIO	GNDIO6	-		
VCCIO	VCCIO6	6		
T3	PL65A	6	LLM4_SPLLT_IN_A/LDQ63	T (LVDS)*
T2	PL65B	6	LLM4_SPLLC_IN_A/LDQ63	C (LVDS)*
U9	PL66A	6	LLM4_SPLLT_FB_A/LDQ63	T
U8	PL66B	6	LLM4_SPLLC_FB_A/LDQ63	C
GNDIO	GNDIO6	-		
U5	PL68A	6	LDQ72	T (LVDS)*
U4	PL68B	6	LDQ72	C (LVDS)*
V9	PL69A	6	LDQ72	T
V7	PL69B	6	LDQ72	C
VCCIO	VCCIO6	6		
U3	PL70A	6	LDQ72	T (LVDS)*
U2	PL70B	6	LDQ72	C (LVDS)*
V8	PL71A	6	LDQ72	T
U6	PL71B	6	LDQ72	C
GNDIO	GNDIO6	-		
U1	PL72A	6	LDQS72	T (LVDS)*
V2	PL72B	6	LDQ72	C (LVDS)*
V5	PL73A	6	LDQ72	T
VCCIO	VCCIO6	6		
V6	PL73B	6	LDQ72	C
V1	PL74A	6	LDQ72	T (LVDS)*
W1	PL74B	6	LDQ72	C (LVDS)*
W5	PL75A	6	LDQ72	T
GNDIO	GNDIO6	-		
W6	PL75B	6	LDQ72	C
W3	PL77A	6	LDQ81	T (LVDS)*
W4	PL77B	6	LDQ81	C (LVDS)*
W2	PL78A	6	LDQ81	T
Y4	PL78B	6	LDQ81	C
Y1	PL79A	6	LDQ81	T (LVDS)*
VCCIO	VCCIO6	6		
Y2	PL79B	6	LDQ81	C (LVDS)*
Y5	PL80A	6	LDQ81	T
Y6	PL80B	6	LDQ81	C

LFE2M100E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2M100E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
AA1	PL81A	6	LDQS81	T (LVDS)*
GNDIO	GNDIO6	-		
AA2	PL81B	6	LDQ81	C (LVDS)*
Y3	PL82A	6	LDQ81	T
AB1	PL82B	6	LDQ81	C
VCCIO	VCCIO6	6		
Y9	PL83A	6	LDQ81	T (LVDS)*
Y8	PL83B	6	LDQ81	C (LVDS)*
Y7	PL84A	6	LDQ81	T
AA7	PL84B	6	LDQ81	C
GNDIO	GNDIO6	-		
VCCIO	VCCIO6	6		
AB2	PL95A	6	LDQ99	T (LVDS)*
AB3	PL95B	6	LDQ99	C (LVDS)*
AA5	PL96A	6	LDQ99	T
AA6	PL96B	6	LDQ99	C
AB4	PL97A	6	LDQ99	T (LVDS)*
VCCIO	VCCIO6	6		
AB5	PL97B	6	LDQ99	C (LVDS)*
AA8	PL98A	6	LDQ99	T
AA9	PL98B	6	LDQ99	C
AC1	PL99A	6	LLM0_GPLLT_IN_A**/LDQS99	T (LVDS)*
GNDIO	GNDIO6	-		
AC2	PL99B	6	LLM0_GPLLC_IN_A**/LDQ99	C (LVDS)*
AC4	PL100A	6	LLM0_GPLLT_FB_A/LDQ99	T
AC3	PL100B	6	LLM0_GPLLC_FB_A/LDQ99	C
VCCIO	VCCIO6	6		
AC7	PL101A	6	LLM0_GDLLT_IN_A**/LDQ99	T (LVDS)*
AC6	PL101B	6	LLM0_GDLLC_IN_A**/LDQ99	C (LVDS)*
AC5	PL102A	6	LLM0_GDLLT_FB_A/LDQ99	T
AD3	PL102B	6	LLM0_GDLLC_FB_A/LDQ99	C
GNDIO	GNDIO6	-		
AB8	LLM0_PLLCAP	6		
AD2	PL104A	6		T
AD1	PL104B	6		C
AE2	TCK	-		
AE1	TDI	-		
AF2	TMS	-		
AF1	TDO	-		
AG1	VCCJ	-		
AH1	LLC_SQ_VCCRX3	14		
AK2	LLC_SQ_HDINP3	14		T
AJ1	LLC_SQ_VCCIB3	14		

LFE2M100E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2M100E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
AJ2	LLC_SQ_HDINN3	14		C
AH4	LLC_SQ_VCCTX3	14		
AK5	LLC_SQ_HDOUTP3	14		T
AK4	LLC_SQ_VCCOB3	14		
AJ5	LLC_SQ_HDOUTN3	14		C
AH5	LLC_SQ_VCCTX2	14		
AJ6	LLC_SQ_HDOUTN2	14		C
AH6	LLC_SQ_VCCOB2	14		
AK6	LLC_SQ_HDOUTP2	14		T
AH2	LLC_SQ_VCCRX2	14		
AJ3	LLC_SQ_HDINN2	14		C
AH3	LLC_SQ_VCCIB2	14		
AK3	LLC_SQ_HDINP2	14		T
AH7	LLC_SQ_VCCP	14		
AG7	LLC_SQ_REFCLKP	14		T
AF7	LLC_SQ_REFCLKN	14		C
AJ7	LLC_SQ_VCCAUX33	14		
AK11	LLC_SQ_HDINP1	14		T
AH11	LLC_SQ_VCCIB1	14		
AJ11	LLC_SQ_HDINN1	14		C
AH12	LLC_SQ_VCCRX1	14		
AK8	LLC_SQ_HDOUTP1	14		T
AH8	LLC_SQ_VCCOB1	14		
AJ8	LLC_SQ_HDOUTN1	14		C
AH9	LLC_SQ_VCCTX1	14		
AJ9	LLC_SQ_HDOUTN0	14		C
AK10	LLC_SQ_VCCOB0	14		
AK9	LLC_SQ_HDOUTP0	14		T
AH10	LLC_SQ_VCCTX0	14		
AJ12	LLC_SQ_HDINN0	14		C
AJ13	LLC_SQ_VCCIB0	14		
AK12	LLC_SQ_HDINP0	14		T
AH13	LLC_SQ_VCCRX0	14		
AF10	PB30A	5	BDQ33	T
AE8	PB30B	5	BDQ33	C
AE11	PB31A	5	BDQ33	T
VCCIO	VCCIO5	5		
AD9	PB31B	5	BDQ33	C
AE10	PB32A	5	BDQ33	T
AD10	PB32B	5	BDQ33	C
AE13	PB33A	5	BDQS33	T
GNDIO	GNDIO5	-		
AC12	PB33B	5	BDQ33	C

LFE2M100E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2M100E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
AG2	PB34A	5	BDQ33	T
AG3	PB34B	5	BDQ33	C
AD13	PB35A	5	BDQ33	T
VCCIO	VCCIO5	5		
AC13	PB35B	5	BDQ33	C
AE14	PB36A	5	BDQ33	T
AC14	PB36B	5	BDQ33	C
AF3	PB37A	5	BDQ33	T
GNDIO	GNDIO5	-		
AF4	PB37B	5	BDQ33	C
-	-	-		
AG4	PB38A	5	BDQ42	T
AG5	PB38B	5	BDQ42	C
GNDIO	GNDIO5	-		
-	-	-		
AD11	PB48A	5	BDQ51	T
AF13	PB48B	5	BDQ51	C
AF12	PB49A	5	BDQ51	T
VCCIO	VCCIO5	5		
AD14	PB49B	5	BDQ51	C
AG8	PB50A	5	BDQ51	T
AF8	PB50B	5	BDQ51	C
AE15	PB51A	5	BDQS51****	T
GNDIO	GNDIO5	-		
-	-	-		
AC15	PB51B	5	BDQ51	C
VCCIO	VCCIO5	5		
GNDIO	GNDIO5	-		
AD15	PB56A	5	BDQ60	T
AF15	PB56B	5	BDQ60	C
AG10	PB57A	5	BDQ60	T
AG9	PB57B	5	BDQ60	C
AH14	PB58A	5	BDQ60	T
AG12	PB58B	5	BDQ60	C
VCCIO	VCCIO5	5		
AG15	PB59A	5	BDQ60	T
AG13	PB59B	5	BDQ60	C
GNDIO	GNDIO5	-		
AF16	PB60A	5	BDQS60	T
AH15	PB60B	5	BDQ60	C
AC16	PB61A	5	VREF2_5/BDQ60	T
AE16	PB61B	5	VREF1_5/BDQ60	C
AG11	PB62A	5	PCLKT5_0/BDQ60	T

LFE2M100E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2M100E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
AF11	PB62B	5	PCLKC5_0/BDQ60	C
VCCIO	VCCIO5	5		
GNDIO	GNDIO5	-		
AJ14	PB67A	4	PCLKT4_0/BDQ69	T
VCCIO	VCCIO4	4		
AK14	PB67B	4	PCLKC4_0/BDQ69	C
AK15	PB68A	4	VREF2_4/BDQ69	T
AK16	PB68B	4	VREF1_4/BDQ69	C
AF18	PB69A	4	BDQS69	T
GNDIO	GNDIO4	-		
AD16	PB69B	4	BDQ69	C
AJ15	PB70A	4	BDQ69	T
AG16	PB70B	4	BDQ69	C
AE17	PB71A	4	BDQ69	T
VCCIO	VCCIO4	4		
AC17	PB71B	4	BDQ69	C
AH16	PB72A	4	BDQ69	T
AK17	PB72B	4	BDQ69	C
AG20	PB73A	4	BDQ69	T
GNDIO	GNDIO4	-		
AG21	PB73B	4	BDQ69	C
AG18	PB74A	4	BDQ78	T
AJ16	PB74B	4	BDQ78	C
AF21	PB75A	4	BDQ78	T
AG22	PB75B	4	BDQ78	C
AD17	PB76A	4	BDQ78	T
AF19	PB76B	4	BDQ78	C
VCCIO	VCCIO4	4		
GNDIO	GNDIO4	-		
AH17	PB80A	4	BDQ78	T
AJ17	PB80B	4	BDQ78	C
VCCIO	VCCIO4	4		
AF26	PB82A	4	BDQ78	T
AE25	PB82B	4	BDQ78	C
GNDIO	GNDIO4	-		
AD24	PB92A	4	BDQ96	T
AE24	PB92B	4	BDQ96	C
AD18	PB93A	4	BDQ96	T
AC18	PB93B	4	BDQ96	C
AE18	PB94A	4	BDQ96	T
AG19	PB94B	4	BDQ96	C
VCCIO	VCCIO4	4		
GNDIO	GNDIO4	-		

LFE2M100E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2M100E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
AC19	PB96A	4	BDQS96	T
AD20	PB96B	4	BDQ96	C
AB18	PB97A	4	BDQ96	T
AC20	PB97B	4	BDQ96	C
AE20	PB98A	4	BDQ96	T
AE21	PB98B	4	BDQ96	C
VCCIO	VCCIO4	4		
AC23	PB99A	4	BDQ96	T
AD23	PB99B	4	BDQ96	C
GNDIO	GNDIO4	-		
AH18	LRC_SQ_VCCRX3	13		
AK19	LRC_SQ_HDINP3	13		T
AJ18	LRC_SQ_VCCIB3	13		
AJ19	LRC_SQ_HDINN3	13		C
AH21	LRC_SQ_VCCTX3	13		
AK22	LRC_SQ_HDOUTP3	13		T
AK21	LRC_SQ_VCCOB3	13		
AJ22	LRC_SQ_HDOUTN3	13		C
AH22	LRC_SQ_VCCTX2	13		
AJ23	LRC_SQ_HDOUTN2	13		C
AH23	LRC_SQ_VCCOB2	13		
AK23	LRC_SQ_HDOUTP2	13		T
AH19	LRC_SQ_VCCRX2	13		
AJ20	LRC_SQ_HDINN2	13		C
AH20	LRC_SQ_VCCIB2	13		
AK20	LRC_SQ_HDINP2	13		T
AH24	LRC_SQ_VCCP	13		
AG24	LRC_SQ_REFCLKP	13		T
AF24	LRC_SQ_REFCLKN	13		C
AJ24	LRC_SQ_VCCAUX33	13		
AK28	LRC_SQ_HDINP1	13		T
AH28	LRC_SQ_VCCIB1	13		
AJ28	LRC_SQ_HDINN1	13		C
AH29	LRC_SQ_VCCRX1	13		
AK25	LRC_SQ_HDOUTP1	13		T
AH25	LRC_SQ_VCCOB1	13		
AJ25	LRC_SQ_HDOUTN1	13		C
AH26	LRC_SQ_VCCTX1	13		
AJ26	LRC_SQ_HDOUTN0	13		C
AK27	LRC_SQ_VCCOB0	13		
AK26	LRC_SQ_HDOUTP0	13		T
AH27	LRC_SQ_VCCTX0	13		
AJ29	LRC_SQ_HDINN0	13		C

LFE2M100E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2M100E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
AJ30	LRC_SQ_VCCIB0	13		
AK29	LRC_SQ_HDINP0	13		T
AH30	LRC_SQ_VCCRX0	13		
AG27	CFG2	8		
AD25	CFG1	8		
AG28	CFG0	8		
AG30	PROGRAMN	8		
AG29	CCLK	8		
AC24	INITN	8		
AF27	DONE	8		
GNDIO	GNDIO8	-		
AF28	WRITEN***	8		
AE26	CS1N***	8		
AB23	CSN***	8		
AF29	D0/SPIFASTN***	8		
VCCIO	VCCIO8	8		
AF30	D1***	8		
AD26	D2***	8		
AE29	D3***	8		
GNDIO	GNDIO8	-		
AE30	D4***	8		
AD29	D5***	8		
AC25	D6***	8		
AD30	D7/SPID0***	8		
VCCIO	VCCIO8	8		
AA22	DI/CSSPI0N***	8		
AC26	DOUT/CSON/CSSPI1N***	8		
AA23	BUSY/SISPI***	8		
AB22	RLM0_PLLCAP	3		
AC27	PR102B	3	RLM0_GDLLC_FB_A/RDQ99	C
GNDIO	GNDIO3	-		
AC28	PR102A	3	RLM0_GDLLT_FB_A/RDQ99	T
AC29	PR101B	3	RLM0_GDLLC_IN_A**/RDQ99	C (LVDS)*
AC30	PR101A	3	RLM0_GDLLT_IN_A**/RDQ99	T (LVDS)*
AB30	PR100B	3	RLM0_GPLLC_IN_A**/RDQ99	C
VCCIO	VCCIO3	3		
AA30	PR100A	3	RLM0_GPLLT_IN_A**/RDQ99	T
AB29	PR99B	3	RLM0_GPLLC_FB_A/RDQ99	C (LVDS)*
AB28	PR99A	3	RLM0_GPLLT_FB_A/RDQS99	T (LVDS)*
GNDIO	GNDIO3	-		
Y22	PR98B	3	RDQ99	C
Y23	PR98A	3	RDQ99	T
AB26	PR97B	3	RDQ99	C (LVDS)*

LFE2M100E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2M100E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
AB27	PR97A	3	RDQ99	T (LVDS)*
VCCIO	VCCIO3	3		
Y24	PR96B	3	RDQ99	C
Y25	PR96A	3	RDQ99	T
AA29	PR95B	3	RDQ99	C (LVDS)*
Y28	PR95A	3	RDQ99	T (LVDS)*
Y30	PR93B	3	RDQ90	C
Y29	PR93A	3	RDQ90	T
GNDIO	GNDIO3	-		
VCCIO	VCCIO3	3		
W22	PR83B	3	RDQ81	C (LVDS)*
V22	PR83A	3	RDQ81	T (LVDS)*
Y27	PR82B	3	RDQ81	C
VCCIO	VCCIO3	3		
Y26	PR82A	3	RDQ81	T
W30	PR81B	3	RDQ81	C (LVDS)*
W29	PR81A	3	RDQS81	T (LVDS)*
GNDIO	GNDIO3	-		
W25	PR80B	3	RDQ81	C
W26	PR80A	3	RDQ81	T
U29	PR79B	3	RDQ81	C (LVDS)*
V29	PR79A	3	RDQ81	T (LVDS)*
VCCIO	VCCIO3	3		
V30	PR78B	3	RDQ81	C
U30	PR78A	3	RDQ81	T
W27	PR77B	3	RDQ81	C (LVDS)*
W28	PR77A	3	RDQ81	T (LVDS)*
V24	PR75B	3	RDQ72	C
V25	PR75A	3	RDQ72	T
GNDIO	GNDIO3	-		
U28	PR74B	3	RDQ72	C (LVDS)*
U27	PR74A	3	RDQ72	T (LVDS)*
U23	PR73B	3	RDQ72	C
V23	PR73A	3	RDQ72	T
VCCIO	VCCIO3	3		
V26	PR72B	3	RDQ72	C (LVDS)*
U26	PR72A	3	RDQS72	T (LVDS)*
U25	PR71B	3	RDQ72	C
GNDIO	GNDIO3	-		
U24	PR71A	3	RDQ72	T
T30	PR70B	3	RDQ72	C (LVDS)*
R30	PR70A	3	RDQ72	T (LVDS)*
T23	PR69B	3	RDQ72	C

LFE2M100E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2M100E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
VCCIO	VCCIO3	3		
T22	PR69A	3	RDQ72	T
T29	PR68B	3	RDQ72	C (LVDS)*
T28	PR68A	3	RDQ72	T (LVDS)*
R23	PR66B	3	RLM4_SPLLC_FB_A/RDQ63	C
GNDIO	GNDIO3	-		
-	-	-		
R22	PR66A	3	RLM4_SPLLT_FB_A/RDQ63	T
P30	PR65B	3	RLM4_SPLLC_IN_A/RDQ63	C (LVDS)*
R29	PR65A	3	RLM4_SPLLT_IN_A/RDQ63	T (LVDS)*
T27	PR64B	3	RDQ63	C
VCCIO	VCCIO3	3		
T26	PR64A	3	RDQ63	T
GNDIO	GNDIO3	-		
N30	PR61B	3	RDQ63	C (LVDS)*
N29	PR61A	3	RDQ63	T (LVDS)*
VCCIO	VCCIO3	3		
R27	PR60B	3	VREF2_3/RDQ63	C
R28	PR60A	3	VREF1_3/RDQ63	T
P29	PR59B	3	PCLKC3_0/RDQ63	C (LVDS)*
P28	PR59A	3	PCLKT3_0/RDQ63	T (LVDS)*
M30	PR57B	2	PCLKC2_0/RDQ54	C
M29	PR57A	2	PCLKT2_0/RDQ54	T
GNDIO	GNDIO2	-		
P23	PR56B	2	RDQ54	C (LVDS)*
P24	PR56A	2	RDQ54	T (LVDS)*
R26	PR55B	2	RDQ54	C
P27	PR55A	2	RDQ54	T
VCCIO	VCCIO2	2		
P25	PR54B	2	RDQ54	C (LVDS)*
P26	PR54A	2	RDQS54	T (LVDS)*
K30	PR53B	2	RDQ54	C
GNDIO	GNDIO2	-		
K29	PR53A	2	RDQ54	T
N22	PR52B	2	RDQ54	C (LVDS)*
P22	PR52A	2	RDQ54	T (LVDS)*
J30	PR51B	2	RUM3_SPLLC_FB_A/RDQ54	C
VCCIO	VCCIO2	2		
J29	PR51A	2	RUM3_SPLLT_FB_A/RDQ54	T
N24	PR50B	2	RUM3_SPLLC_IN_A/RDQ54	C (LVDS)*
N23	PR50A	2	RUM3_SPLLT_IN_A/RDQ54	T (LVDS)*
N25	PR48B	2	RDQ45	C
N26	PR48A	2	RDQ45	T

LFE2M100E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2M100E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
GNDIO	GNDIO2	-		
M27	PR47B	2	RDQ45	C (LVDS)*
M28	PR47A	2	RDQ45	T (LVDS)*
H30	PR46B	2	RDQ45	C
G30	PR46A	2	RDQ45	T
VCCIO	VCCIO2	2		
M25	PR45B	2	RDQ45	C (LVDS)*
M26	PR45A	2	RDQS45	T (LVDS)*
L30	PR44B	2	RDQ45	C
GNDIO	GNDIO2	-		
L29	PR44A	2	RDQ45	T
L28	PR43B	2	RDQ45	C (LVDS)*
L27	PR43A	2	RDQ45	T (LVDS)*
H29	PR42B	2	RDQ45	C
VCCIO	VCCIO2	2		
G29	PR42A	2	RDQ45	T
L22	PR41B	2	RDQ45	C (LVDS)*
M22	PR41A	2	RDQ45	T (LVDS)*
F30	PR40B	2		C
GNDIO	GNDIO2	-		
F29	PR40A	2		T
VCCIO	VCCIO2	2		
GNDIO	GNDIO2	-		
E30	PR34B	2	RDQ32	C (LVDS)*
E29	PR34A	2	RDQ32	T (LVDS)*
-	-	-		
L25	PR33B	2	RDQ32	C
L26	PR33A	2	RDQ32	T
VCCIO	VCCIO2	2		
H28	PR32B	2	RDQ32	C (LVDS)*
J28	PR32A	2	RDQS32	T (LVDS)*
G28	PR31B	2	RDQ32	C
GNDIO	GNDIO2	-		
G27	PR31A	2	RDQ32	T
L24	PR30B	2	RDQ32	C (LVDS)*
L23	PR30A	2	RDQ32	T (LVDS)*
D30	PR29B	2	RDQ32	C
VCCIO	VCCIO2	2		
D29	PR29A	2	RDQ32	T
K24	PR28B	2	RDQ32	C (LVDS)*
K25	PR28A	2	RDQ32	T (LVDS)*
J27	PR26B	2	RDQ23	C
GNDIO	GNDIO2	-		

LFE2M100E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2M100E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
K26	PR26A	2	RDQ23	T
K23	PR25B	2	RDQ23	C (LVDS)*
K22	PR25A	2	RDQ23	T (LVDS)*
J22	PR24B	2	RDQ23	C
VCCIO	VCCIO2	2		
J23	PR24A	2	RDQ23	T
GNDIO	GNDIO2	-		
VCCIO	VCCIO2	2		
J26	PR17B	2	RDQ15	C (LVDS)*
H26	PR17A	2	RDQ15	T (LVDS)*
H27	PR16B	2	RDQ15	C
G26	PR16A	2	RDQ15	T
VCCIO	VCCIO2	2		
H23	PR15B	2	RDQ15	C (LVDS)*
H24	PR15A	2	RDQS15	T (LVDS)*
D28	PR14B	2	RDQ15	C
GNDIO	GNDIO2	-		
E28	PR14A	2	RDQ15	T
G24	PR13B	2	RDQ15	C (LVDS)*
H25	PR13A	2	RDQ15	T (LVDS)*
D27	PR12B	2	RUM0_SPLLC_FB_A/RDQ15	C
VCCIO	VCCIO2	2		
E27	PR12A	2	RUM0_SPLLT_FB_A/RDQ15	T
F26	PR11B	2	RUM0_SPLLC_IN_A/RDQ15	C (LVDS)*
G25	PR11A	2	RUM0_SPLLT_IN_A/RDQ15	T (LVDS)*
F24	PR9B	2	VREF2_2	C
-	-	-		
GNDIO	GNDIO2	-		
F25	PR9A	2	VREF1_2	T
VCCIO	VCCIO2	2		
G23	XRES	1		
C30	URC_SQ_VCCRX0	12		
A29	URC_SQ_HDINP0	12		T
B30	URC_SQ_VCCIB0	12		
B29	URC_SQ_HDINN0	12		C
C27	URC_SQ_VCCTX0	12		
A26	URC_SQ_HDOUTP0	12		T
A27	URC_SQ_VCCOB0	12		
B26	URC_SQ_HDOUTN0	12		C
C26	URC_SQ_VCCTX1	12		
B25	URC_SQ_HDOUTN1	12		C
C25	URC_SQ_VCCOB1	12		
A25	URC_SQ_HDOUTP1	12		T

LFE2M100E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2M100E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
C29	URC_SQ_VCCRX1	12		
B28	URC_SQ_HDINN1	12		C
C28	URC_SQ_VCCIB1	12		
A28	URC_SQ_HDINP1	12		T
B24	URC_SQ_VCCAUX33	12		
E24	URC_SQ_REFCLKN	12		C
D24	URC_SQ_REFCLKP	12		T
C24	URC_SQ_VCCP	12		
A20	URC_SQ_HDINP2	12		T
C20	URC_SQ_VCCIB2	12		
B20	URC_SQ_HDINN2	12		C
C19	URC_SQ_VCCRX2	12		
A23	URC_SQ_HDOUTP2	12		T
C23	URC_SQ_VCCOB2	12		
B23	URC_SQ_HDOUTN2	12		C
C22	URC_SQ_VCCTX2	12		
B22	URC_SQ_HDOUTN3	12		C
A21	URC_SQ_VCCOB3	12		
A22	URC_SQ_HDOUTP3	12		T
C21	URC_SQ_VCCTX3	12		
B19	URC_SQ_HDINN3	12		C
B18	URC_SQ_VCCIB3	12		
A19	URC_SQ_HDINP3	12		T
C18	URC_SQ_VCCRX3	12		
D23	PT100B	1		C
GNDIO	GNDIO1	-		
E21	PT100A	1		T
D26	PT99B	1		C
E26	PT99A	1		T
E23	PT98B	1		C
VCCIO	VCCIO1	1		
G22	PT98A	1		T
-	-	-		
D22	PT97B	1		C
F21	PT97A	1		T
G18	PT96B	1		C
H18	PT96A	1		T
D20	PT95B	1		C
GNDIO	GNDIO1	-		
D21	PT95A	1		T
E20	PT94B	1		C
VCCIO	VCCIO1	1		
E19	PT94A	1		T

LFE2M100E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2M100E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
D19	PT93B	1		C
E18	PT93A	1		T
D18	PT92B	1		C
C17	PT92A	1		T
A17	PT91B	1		C
B17	PT91A	1		T
GNDIO	GNDIO1	-		
VCCIO	VCCIO1	1		
J18	PT75B	1		C
J19	PT75A	1		T
H17	PT74B	1		C
J17	PT74A	1		T
F18	PT73B	1		C
F17	PT73A	1		T
GNDIO	GNDIO1	-		
A16	PT72B	1		C
B16	PT72A	1		T
G17	PT71B	1		C
G16	PT71A	1		T
VCCIO	VCCIO1	1		
H16	PT70B	1		C
F16	PT70A	1		T
J16	PT69B	1		C
G15	PT69A	1		T
GNDIO	GNDIO1	-		
C16	PT68B	1		C
D16	PT68A	1		T
J15	PT67B	1		C
H15	PT67A	1		T
VCCIO	VCCIO1	1		
A15	PT66B	1	VREF2_1	C
B15	PT66A	1	VREF1_1	T
F15	PT65B	1	PCLKC1_0	C
E16	PT65A	1	PCLKT1_0	T
C15	PT64B	0	PCLKC0_0	C
GNDIO	GNDIO0	-		
D15	PT64A	0	PCLKT0_0	T
C14	PT63B	0	VREF2_0	C
E15	PT63A	0	VREF1_0	T
G14	PT62B	0		C
VCCIO	VCCIO0	0		
J14	PT62A	0		T
F14	PT61B	0		C

LFE2M100E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2M100E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
H14	PT61A	0		T
A14	PT60B	0		C
B14	PT60A	0		T
D13	PT59B	0		C
GNDIO	GNDIO0	-		
F13	PT59A	0		T
G13	PT58B	0		C
VCCIO	VCCIO0	0		
J11	PT58A	0		T
D4	PT57B	0		
D5	PT56A	0		
E5	PT55B	0		C
F6	PT55A	0		T
GNDIO	GNDIO0	-		
VCCIO	VCCIO0	0		
F7	PT52B	0		C
D8	PT52A	0		T
GNDIO	GNDIO0	-		
J13	PT50B	0		C
G11	PT50A	0		T
H13	PT49B	0		C
H12	PT49A	0		T
VCCIO	VCCIO0	0		
E8	PT48B	0		C
D9	PT48A	0		T
D12	PT46B	0		C
GNDIO	GNDIO0	-		
E13	PT46A	0		T
VCCIO	VCCIO0	0		
GNDIO	GNDIO0	-		
J12	PT31B	0		C
-	-	-		
VCCIO	VCCIO0	0		
H10	PT31A	0		T
E12	PT30B	0		C
D11	PT30A	0		T
H11	PT29B	0		C
F11	PT29A	0		T
C13	ULC_SQ_VCCRX0	11		
A12	ULC_SQ_HDINP0	11		T
B13	ULC_SQ_VCCIB0	11		
B12	ULC_SQ_HDINN0	11		C
C10	ULC_SQ_VCCTX0	11		

LFE2M100E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2M100E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
A9	ULC_SQ_HDOUTP0	11		T
A10	ULC_SQ_VCCOB0	11		
B9	ULC_SQ_HDOUTN0	11		C
C9	ULC_SQ_VCCTX1	11		
B8	ULC_SQ_HDOUTN1	11		C
C8	ULC_SQ_VCCOB1	11		
A8	ULC_SQ_HDOUTP1	11		T
C12	ULC_SQ_VCCRX1	11		
B11	ULC_SQ_HDINN1	11		C
C11	ULC_SQ_VCCIB1	11		
A11	ULC_SQ_HDINP1	11		T
B7	ULC_SQ_VCCAUX33	11		
E7	ULC_SQ_REFCLKN	11		C
D7	ULC_SQ_REFCLKP	11		T
C7	ULC_SQ_VCCP	11		
A3	ULC_SQ_HDINP2	11		T
C3	ULC_SQ_VCCIB2	11		
B3	ULC_SQ_HDINN2	11		C
C2	ULC_SQ_VCCRX2	11		
A6	ULC_SQ_HDOUTP2	11		T
C6	ULC_SQ_VCCOB2	11		
B6	ULC_SQ_HDOUTN2	11		C
C5	ULC_SQ_VCCTX2	11		
B5	ULC_SQ_HDOUTN3	11		C
A4	ULC_SQ_VCCOB3	11		
A5	ULC_SQ_HDOUTP3	11		T
C4	ULC_SQ_VCCTX3	11		
B2	ULC_SQ_HDINN3	11		C
B1	ULC_SQ_VCCIB3	11		
A2	ULC_SQ_HDINP3	11		T
C1	ULC_SQ_VCCRX3	11		
L12	VCC	-		
L13	VCC	-		
L18	VCC	-		
L19	VCC	-		
M11	VCC	-		
M12	VCC	-		
M13	VCC	-		
M14	VCC	-		
M15	VCC	-		
M16	VCC	-		
M17	VCC	-		
M18	VCC	-		

LFE2M100E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2M100E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
M19	VCC	-		
M20	VCC	-		
N11	VCC	-		
N12	VCC	-		
N19	VCC	-		
N20	VCC	-		
P12	VCC	-		
P19	VCC	-		
R12	VCC	-		
R19	VCC	-		
T12	VCC	-		
T19	VCC	-		
U12	VCC	-		
U19	VCC	-		
V11	VCC	-		
V12	VCC	-		
V19	VCC	-		
V20	VCC	-		
W11	VCC	-		
W12	VCC	-		
W13	VCC	-		
W14	VCC	-		
W15	VCC	-		
W16	VCC	-		
W17	VCC	-		
W18	VCC	-		
W19	VCC	-		
W20	VCC	-		
Y12	VCC	-		
Y13	VCC	-		
Y18	VCC	-		
Y19	VCC	-		
D14	VCCIO0	0		
E6	VCCIO0	0		
E9	VCCIO0	0		
F12	VCCIO0	0		
K12	VCCIO0	0		
K13	VCCIO0	0		
D17	VCCIO1	1		
E22	VCCIO1	1		
E25	VCCIO1	1		
F19	VCCIO1	1		
K18	VCCIO1	1		

LFE2M100E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2M100E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
K19	VCCIO1	1		
F28	VCCIO2	2		
J25	VCCIO2	2		
K28	VCCIO2	2		
M21	VCCIO2	2		
M24	VCCIO2	2		
N21	VCCIO2	2		
N28	VCCIO2	2		
P21	VCCIO2	2		
R25	VCCIO2	2		
AA28	VCCIO3	3		
AB25	VCCIO3	3		
AE28	VCCIO3	3		
T25	VCCIO3	3		
U21	VCCIO3	3		
V21	VCCIO3	3		
V28	VCCIO3	3		
W21	VCCIO3	3		
W24	VCCIO3	3		
AA18	VCCIO4	4		
AA19	VCCIO4	4		
AE19	VCCIO4	4		
AF22	VCCIO4	4		
AG17	VCCIO4	4		
AG25	VCCIO4	4		
AA12	VCCIO5	5		
AA13	VCCIO5	5		
AE12	VCCIO5	5		
AF9	VCCIO5	5		
AG14	VCCIO5	5		
AG6	VCCIO5	5		
AA3	VCCIO6	6		
AB6	VCCIO6	6		
AE3	VCCIO6	6		
T6	VCCIO6	6		
U10	VCCIO6	6		
V10	VCCIO6	6		
V3	VCCIO6	6		
W10	VCCIO6	6		
W7	VCCIO6	6		
F3	VCCIO7	7		
J6	VCCIO7	7		
K3	VCCIO7	7		

LFE2M100E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2M100E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
M10	VCCIO7	7		
M7	VCCIO7	7		
N10	VCCIO7	7		
N3	VCCIO7	7		
P10	VCCIO7	7		
R6	VCCIO7	7		
AA25	VCCIO8	8		
AD28	VCCIO8	8		
AA10	VCCAUX	-		
AA11	VCCAUX	-		
AA20	VCCAUX	-		
AA21	VCCAUX	-		
K10	VCCAUX	-		
K11	VCCAUX	-		
K20	VCCAUX	-		
K21	VCCAUX	-		
L10	VCCAUX	-		
L11	VCCAUX	-		
L20	VCCAUX	-		
L21	VCCAUX	-		
Y10	VCCAUX	-		
Y11	VCCAUX	-		
Y20	VCCAUX	-		
Y21	VCCAUX	-		
A1	GND	-		
A13	GND	-		
A18	GND	-		
A24	GND	-		
A30	GND	-		
A7	GND	-		
AA14	GND	-		
AA15	GND	-		
AA16	GND	-		
AA17	GND	-		
AA24	GND	-		
AA27	GND	-		
AA4	GND	-		
AB24	GND	-		
AB7	GND	-		
AD12	GND	-		
AD19	GND	-		
AD27	GND	-		
AE22	GND	-		

LFE2M100E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2M100E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
AE27	GND	-		
AE4	GND	-		
AE9	GND	-		
AF14	GND	-		
AF17	GND	-		
AF25	GND	-		
AF6	GND	-		
AJ10	GND	-		
AJ21	GND	-		
AJ27	GND	-		
AJ4	GND	-		
AK1	GND	-		
AK13	GND	-		
AK18	GND	-		
AK24	GND	-		
AK30	GND	-		
AK7	GND	-		
B10	GND	-		
B21	GND	-		
B27	GND	-		
B4	GND	-		
D25	GND	-		
D6	GND	-		
E14	GND	-		
E17	GND	-		
F22	GND	-		
F27	GND	-		
F4	GND	-		
F9	GND	-		
G12	GND	-		
G19	GND	-		
J24	GND	-		
J7	GND	-		
K14	GND	-		
K15	GND	-		
K16	GND	-		
K17	GND	-		
K27	GND	-		
K4	GND	-		
L14	GND	-		
L15	GND	-		
L16	GND	-		
L17	GND	-		

LFE2M100E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2M100E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
M23	GND	-		
M8	GND	-		
N14	GND	-		
N15	GND	-		
N16	GND	-		
N17	GND	-		
N27	GND	-		
N4	GND	-		
P11	GND	-		
P13	GND	-		
P14	GND	-		
P15	GND	-		
P16	GND	-		
P17	GND	-		
P18	GND	-		
P20	GND	-		
R10	GND	-		
R11	GND	-		
R13	GND	-		
R14	GND	-		
R15	GND	-		
R16	GND	-		
R17	GND	-		
R18	GND	-		
R20	GND	-		
R21	GND	-		
R24	GND	-		
R7	GND	-		
T10	GND	-		
T11	GND	-		
T13	GND	-		
T14	GND	-		
T15	GND	-		
T16	GND	-		
T17	GND	-		
T18	GND	-		
T20	GND	-		
T21	GND	-		
T24	GND	-		
T7	GND	-		
U11	GND	-		
U13	GND	-		
U14	GND	-		

LFE2M100E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2M100E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
U15	GND	-		
U16	GND	-		
U17	GND	-		
U18	GND	-		
U20	GND	-		
V14	GND	-		
V15	GND	-		
V16	GND	-		
V17	GND	-		
V27	GND	-		
V4	GND	-		
W23	GND	-		
W8	GND	-		
Y14	GND	-		
Y15	GND	-		
Y16	GND	-		
Y17	GND	-		
AA26	NC	-		
AB10	NC	-		
AB11	NC	-		
AB12	NC	-		
AB13	NC	-		
AB14	NC	-		
AB15	NC	-		
AB16	NC	-		
AB17	NC	-		
AB19	NC	-		
AB20	NC	-		
AB21	NC	-		
AB9	NC	-		
AC10	NC	-		
AC11	NC	-		
AC21	NC	-		
AC22	NC	-		
AC8	NC	-		
AC9	NC	-		
AD21	NC	-		
AD22	NC	-		
AD4	NC	-		
AD5	NC	-		
AD6	NC	-		
AD7	NC	-		
AD8	NC	-		

LFE2M100E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2M100E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
AE23	NC	-		
AE5	NC	-		
AE6	NC	-		
AE7	NC	-		
AF20	NC	-		
AF23	NC	-		
AF5	NC	-		
AG23	NC	-		
AG26	NC	-		
D10	NC	-		
E10	NC	-		
E11	NC	-		
F10	NC	-		
F20	NC	-		
F23	NC	-		
F8	NC	-		
G10	NC	-		
G20	NC	-		
G21	NC	-		
G7	NC	-		
G8	NC	-		
G9	NC	-		
H19	NC	-		
H20	NC	-		
H21	NC	-		
H22	NC	-		
H6	NC	-		
H8	NC	-		
H9	NC	-		
J10	NC	-		
J20	NC	-		
J21	NC	-		
J9	NC	-		
K9	NC	-		
R9	NC	-		
U22	NC	-		
W9	NC	-		
N13	VCCPLL	-		
N18	VCCPLL	-		
V13	VCCPLL	-		

LFE2M100E/SE Logic Signal Connections: 900 fpBGA (Cont.)

LFE2M100E/SE				
Ball Number	**Ball/Pad Function**	**Bank**	**Dual Function**	**Differential**
V18	VCCPLL	-		

* Supports true LVDS. Other differential signals must be emulated with external resistors.
** These dedicated input pins can be used for GPLLs or GDLLs within the respective quadrant.
*** These sysCONFIG pins are dedicated I/O pins for configuration. The outpus are actively driven during normal device operation.
****Due to packaging bond out option, this DQS does not have all the necessary DQ pins bonded out for a full 8-bit data width.

Note: VCCIO and GND pads are used to determine the average DC current drawn by I/Os between GND/VCCIO connections, or between the last GND/VCCIO in an I/O bank and the end of an I/O bank. The substrate pads listed in the Pin Table do not necessarily have a one to one connection with a package ball or pin.

LFE2M70E/SE and LFE2M100E/SE Logic Signal Connections: 1152 fpBGA

LFE2M70E/SE				LFE2M100E/SE				
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
VCCIO	VCCIO7	7			VCCIO7	7		
F4	PL9A	7	VREF2_7	T	PL9A	7	VREF2_7	T
F3	PL9B	7	VREF1_7	C	PL9B	7	VREF1_7	C
GNDIO	GNDIO7	-			GNDIO7	-		
E1	PL11A	7	LUM0_SPLLT_IN_A/LDQ15	T (LVDS)*	PL11A	7	LUM0_SPLLT_IN_A/LDQ15	T (LVDS)*
E2	PL11B	7	LUM0_SPLLC_IN_A/LDQ15	C (LVDS)*	PL11B	7	LUM0_SPLLC_IN_A/LDQ15	C (LVDS)*
K9	PL12A	7	LUM0_SPLLT_FB_A/LDQ15	T	PL12A	7	LUM0_SPLLT_FB_A/LDQ15	T
H7	PL12B	7	LUM0_SPLLC_FB_A/LDQ15	C	PL12B	7	LUM0_SPLLC_FB_A/LDQ15	C
VCCIO	VCCIO7	7			VCCIO7	7		
F1	PL13A	7	LDQ15	T (LVDS)*	PL13A	7	LDQ15	T (LVDS)*
F2	PL13B	7	LDQ15	C (LVDS)*	PL13B	7	LDQ15	C (LVDS)*
J8	PL14A	7	LDQ15	T	PL14A	7	LDQ15	T
H6	PL14B	7	LDQ15	C	PL14B	7	LDQ15	C
GNDIO	GNDIO7	-			GNDIO7	-		
G2	PL15A	7	LDQS15	T (LVDS)*	PL15A	7	LDQS15	T (LVDS)*
G1	PL15B	7	LDQ15	C (LVDS)*	PL15B	7	LDQ15	C (LVDS)*
J7	PL16A	7	LDQ15	T	PL16A	7	LDQ15	T
VCCIO	VCCIO7	7			VCCIO7	7		
L8	PL16B	7	LDQ15	C	PL16B	7	LDQ15	C
L9	PL17A	7	LDQ15	T (LVDS)*	PL17A	7	LDQ15	T (LVDS)*
L10	PL17B	7	LDQ15	C (LVDS)*	PL17B	7	LDQ15	C (LVDS)*
H5	PL18A	7	LDQ15	T	PL18A	7	LDQ15	T
GNDIO	GNDIO7	-			GNDIO7	-		
J6	PL18B	7	LDQ15	C	PL18B	7	LDQ15	C
H2	NC	-			PL19A	7	LDQ23	T (LVDS)*
H1	NC	-			PL19B	7	LDQ23	C (LVDS)*
G5	NC	-			PL20A	7	LDQ23	T
G6	NC	-			PL20B	7	LDQ23	C
M9	NC	-			PL21A	7	LDQ23	T (LVDS)*
-	-	-			VCCIO7	7		
M10	NC	-			PL21B	7	LDQ23	C (LVDS)*
H3	NC	-			PL22A	7	LDQ23	T
H4	NC	-			PL22B	7	LDQ23	C
J2	PL19A	7		T (LVDS)*	PL23A	7	LDQS23	T (LVDS)*
-	-	-			GNDIO7	-		
J1	PL19B	7		C (LVDS)*	PL23B	7	LDQ23	C (LVDS)*
K2	PL20A	7		T	PL24A	7	LDQ23	T
K1	PL20B	7		C	PL24B	7	LDQ23	C
VCCIO	VCCIO7	7			VCCIO7	7		
J4	PL21A	7		T (LVDS)*	PL25A	7	LDQ23	T (LVDS)*
J3	PL21B	7		C (LVDS)*	PL25B	7	LDQ23	C (LVDS)*
J5	PL22A	7		T	PL26A	7	LDQ23	T
K5	PL22B	7		C	PL26B	7	LDQ23	C
GNDIO	GNDIO7	-			GNDIO7	-		
L2	PL24A	7	LDQ28	T (LVDS)*	PL28A	7	LDQ32	T (LVDS)*
L1	PL24B	7	LDQ28	C (LVDS)*	PL28B	7	LDQ32	C (LVDS)*
L7	PL25A	7	LDQ28	T	PL29A	7	LDQ32	T
K6	PL25B	7	LDQ28	C	PL29B	7	LDQ32	C
VCCIO	VCCIO7	7			VCCIO7	7		
M2	PL26A	7	LDQ28	T (LVDS)*	PL30A	7	LDQ32	T (LVDS)*

LFE2M70E/SE and LFE2M100E/SE Logic Signal Connections: 1152 fpBGA

LFE2M70E/SE				LFE2M100E/SE				
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
M1	PL26B	7	LDQ28	C (LVDS)*	PL30B	7	LDQ32	C (LVDS)*
L6	PL27A	7	LDQ28	T	PL31A	7	LDQ32	T
L5	PL27B	7	LDQ28	C	PL31B	7	LDQ32	C
GNDIO	GNDIO7	-			GNDIO7	-		
L3	PL28A	7	LDQS28	T (LVDS)*	PL32A	7	LDQS32	T (LVDS)*
L4	PL28B	7	LDQ28	C (LVDS)*	PL32B	7	LDQ32	C (LVDS)*
M3	PL29A	7	LDQ28	T	PL33A	7	LDQ32	T
VCCIO	VCCIO7	7			VCCIO7	7		
M4	PL29B	7	LDQ28	C	PL33B	7	LDQ32	C
N1	PL30A	7	LDQ28	T (LVDS)*	PL34A	7	LDQ32	T (LVDS)*
N2	PL30B	7	LDQ28	C (LVDS)*	PL34B	7	LDQ32	C (LVDS)*
M5	PL31A	7	LDQ28	T	PL35A	7	LDQ32	T
GNDIO	GNDIO7	-			GNDIO7	-		
N6	PL31B	7	LDQ28	C	PL35B	7	LDQ32	C
P3	NC	-			PL37A	7		T (LVDS)*
-	-	-			GNDIO7	-		
P4	NC	-			PL37B	7		C (LVDS)*
P9	NC	-			PL38A	7		T
M7	NC	-			PL38B	7		C
-	-	-			VCCIO7	7		
P1	NC	-			PL39A	7		T (LVDS)*
P2	NC	-			PL39B	7		C (LVDS)*
N7	NC	-			PL40A	7		T
P7	NC	-			PL40B	7		C
-	-	-			GNDIO7	-		
P5	PL33A	7	LDQ37	T (LVDS)*	PL41A	7	LDQ45	T (LVDS)*
N5	PL33B	7	LDQ37	C (LVDS)*	PL41B	7	LDQ45	C (LVDS)*
P8	PL34A	7	LDQ37	T	PL42A	7	LDQ45	T
P6	PL34B	7	LDQ37	C	PL42B	7	LDQ45	C
VCCIO	VCCIO7	7			VCCIO7	7		
R3	PL35A	7	LDQ37	T (LVDS)*	PL43A	7	LDQ45	T (LVDS)*
R4	PL35B	7	LDQ37	C (LVDS)*	PL43B	7	LDQ45	C (LVDS)*
R10	PL36A	7	LDQ37	T	PL44A	7	LDQ45	T
P11	PL36B	7	LDQ37	C	PL44B	7	LDQ45	C
GNDIO	GNDIO7	-			GNDIO7	-		
R7	PL37A	7	LDQS37	T (LVDS)*	PL45A	7	LDQS45	T (LVDS)*
R8	PL37B	7	LDQ37	C (LVDS)*	PL45B	7	LDQ45	C (LVDS)*
R5	PL38A	7	LDQ37	T	PL46A	7	LDQ45	T
VCCIO	VCCIO7	7			VCCIO7	7		
T5	PL38B	7	LDQ37	C	PL46B	7	LDQ45	C
R1	PL39A	7	LDQ37	T (LVDS)*	PL47A	7	LDQ45	T (LVDS)*
R2	PL39B	7	LDQ37	C (LVDS)*	PL47B	7	LDQ45	C (LVDS)*
R11	PL40A	7	LDQ37	T	PL48A	7	LDQ45	T
GNDIO	GNDIO7	-			GNDIO7	-		
T10	PL40B	7	LDQ37	C	PL48B	7	LDQ45	C
T1	PL42A	7	LUM3_SPLLT_IN_A/LDQ46	T (LVDS)*	PL50A	7	LUM3_SPLLT_IN_A/LDQ54	T (LVDS)*
T2	PL42B	7	LUM3_SPLLC_IN_A/LDQ46	C (LVDS)*	PL50B	7	LUM3_SPLLC_IN_A/LDQ54	C (LVDS)*
U10	PL43A	7	LUM3_SPLLT_FB_A/LDQ46	T	PL51A	7	LUM3_SPLLT_FB_A/LDQ54	T
U8	PL43B	7	LUM3_SPLLC_FB_A/LDQ46	C	PL51B	7	LUM3_SPLLC_FB_A/LDQ54	C
VCCIO	VCCIO7	7			VCCIO7	7		
T6	PL44A	7	LDQ46	T (LVDS)*	PL52A	7	LDQ54	T (LVDS)*

LFE2M70E/SE and LFE2M100E/SE Logic Signal Connections: 1152 fpBGA

LFE2M70E/SE				LFE2M100E/SE				
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
R6	PL44B	7	LDQ46	C (LVDS)*	PL52B	7	LDQ54	C (LVDS)*
U9	PL45A	7	LDQ46	T	PL53A	7	LDQ54	T
T7	PL45B	7	LDQ46	C	PL53B	7	LDQ54	C
GNDIO	GNDIO7	-			GNDIO7	-		
U5	PL46A	7	LDQS46	T (LVDS)*	PL54A	7	LDQS54	T (LVDS)*
U6	PL46B	7	LDQ46	C (LVDS)*	PL54B	7	LDQ54	C (LVDS)*
U7	PL47A	7	LDQ46	T	PL55A	7	LDQ54	T
VCCIO	VCCIO7	7			VCCIO7	7		
V9	PL47B	7	LDQ46	C	PL55B	7	LDQ54	C
V11	PL48A	7	LDQ46	T (LVDS)*	PL56A	7	LDQ54	T (LVDS)*
V10	PL48B	7	LDQ46	C (LVDS)*	PL56B	7	LDQ54	C (LVDS)*
U4	PL49A	7	PCLKT7_0/LDQ46	T	PL57A	7	PCLKT7_0/LDQ54	T
GNDIO	GNDIO7	-			GNDIO7	-		
U3	PL49B	7	PCLKC7_0/LDQ46	C	PL57B	7	PCLKC7_0/LDQ54	C
U2	PL51A	6	PCLKT6_0/LDQ55	T (LVDS)*	PL59A	6	PCLKT6_0/LDQ63	T (LVDS)*
U1	PL51B	6	PCLKC6_0/LDQ55	C (LVDS)*	PL59B	6	PCLKC6_0/LDQ63	C (LVDS)*
V5	PL52A	6	VREF2_6/LDQ55	T	PL60A	6	VREF2_6/LDQ63	T
V6	PL52B	6	VREF1_6/LDQ55	C	PL60B	6	VREF1_6/LDQ63	C
V7	PL53A	6	LDQ55	T (LVDS)*	PL61A	6	LDQ63	T (LVDS)*
VCCIO	VCCIO6	6			VCCIO6	6		
V8	PL53B	6	LDQ55	C (LVDS)*	PL61B	6	LDQ63	C (LVDS)*
V4	PL54A	6	LDQ55	T	PL62A	6	LDQ63	T
V3	PL54B	6	LDQ55	C	PL62B	6	LDQ63	C
V2	PL55A	6	LDQS55	T (LVDS)*	PL63A	6	LDQS63	T (LVDS)*
GNDIO	GNDIO6	-			GNDIO6	-		
V1	PL55B	6	LDQ55	C (LVDS)*	PL63B	6	LDQ63	C (LVDS)*
W7	PL56A	6	LDQ55	T	PL64A	6	LDQ63	T
W5	PL56B	6	LDQ55	C	PL64B	6	LDQ63	C
VCCIO	VCCIO6	6			VCCIO6	6		
W2	PL57A	6	LLM3_SPLLT_IN_A/LDQ55	T (LVDS)*	PL65A	6	LLM4_SPLLT_IN_A/LDQ63	T (LVDS)*
W1	PL57B	6	LLM3_SPLLC_IN_A/LDQ55	C (LVDS)*	PL65B	6	LLM4_SPLLC_IN_A/LDQ63	C (LVDS)*
Y6	PL58A	6	LLM3_SPLLT_FB_A/LDQ55	T	PL66A	6	LLM4_SPLLT_FB_A/LDQ63	T
W6	PL58B	6	LLM3_SPLLC_FB_A/LDQ55	C	PL66B	6	LLM4_SPLLC_FB_A/LDQ63	C
GNDIO	GNDIO6	-			GNDIO6	-		
Y1	PL60A	6	LDQ64	T (LVDS)*	PL68A	6	LDQ72	T (LVDS)*
Y2	PL60B	6	LDQ64	C (LVDS)*	PL68B	6	LDQ72	C (LVDS)*
Y7	PL61A	6	LDQ64	T	PL69A	6	LDQ72	T
Y5	PL61B	6	LDQ64	C	PL69B	6	LDQ72	C
VCCIO	VCCIO6	6			VCCIO6	6		
W10	PL62A	6	LDQ64	T (LVDS)*	PL70A	6	LDQ72	T (LVDS)*
Y8	PL62B	6	LDQ64	C (LVDS)*	PL70B	6	LDQ72	C (LVDS)*
Y4	PL63A	6	LDQ64	T	PL71A	6	LDQ72	T
Y3	PL63B	6	LDQ64	C	PL71B	6	LDQ72	C
GNDIO	GNDIO6	-			GNDIO6	-		
AA1	PL64A	6	LDQS64	T (LVDS)*	PL72A	6	LDQS72	T (LVDS)*
AA2	PL64B	6	LDQ64	C (LVDS)*	PL72B	6	LDQ72	C (LVDS)*
AA8	PL65A	6	LDQ64	T	PL73A	6	LDQ72	T
VCCIO	VCCIO6	6			VCCIO6	6		
Y9	PL65B	6	LDQ64	C	PL73B	6	LDQ72	C
AA6	PL66A	6	LDQ64	T (LVDS)*	PL74A	6	LDQ72	T (LVDS)*
AA7	PL66B	6	LDQ64	C (LVDS)*	PL74B	6	LDQ72	C (LVDS)*

LFE2M70E/SE and LFE2M100E/SE Logic Signal Connections: 1152 fpBGA

LFE2M70E/SE				LFE2M100E/SE				
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
AA4	PL67A	6	LDQ64	T	PL75A	6	LDQ72	T
GNDIO	GNDIO6	-			GNDIO6	-		
AA3	PL67B	6	LDQ64	C	PL75B	6	LDQ72	C
AA9	PL69A	6	LDQ73	T (LVDS)*	PL77A	6	LDQ81	T (LVDS)*
AA10	PL69B	6	LDQ73	C (LVDS)*	PL77B	6	LDQ81	C (LVDS)*
AA5	PL70A	6	LDQ73	T	PL78A	6	LDQ81	T
AB6	PL70B	6	LDQ73	C	PL78B	6	LDQ81	C
AB1	PL71A	6	LDQ73	T (LVDS)*	PL79A	6	LDQ81	T (LVDS)*
VCCIO	VCCIO6	6			VCCIO6	6		
AB2	PL71B	6	LDQ73	C (LVDS)*	PL79B	6	LDQ81	C (LVDS)*
AC8	PL72A	6	LDQ73	T	PL80A	6	LDQ81	T
AB10	PL72B	6	LDQ73	C	PL80B	6	LDQ81	C
AC1	PL73A	6	LDQS73	T (LVDS)*	PL81A	6	LDQS81	T (LVDS)*
GNDIO	GNDIO6	-			GNDIO6	-		
AC2	PL73B	6	LDQ73	C (LVDS)*	PL81B	6	LDQ81	C (LVDS)*
AB7	PL74A	6	LDQ73	T	PL82A	6	LDQ81	T
AB5	PL74B	6	LDQ73	C	PL82B	6	LDQ81	C
VCCIO	VCCIO6	6			VCCIO6	6		
AC3	PL75A	6	LDQ73	T (LVDS)*	PL83A	6	LDQ81	T (LVDS)*
AC4	PL75B	6	LDQ73	C (LVDS)*	PL83B	6	LDQ81	C (LVDS)*
AC10	PL76A	6	LDQ73	T	PL84A	6	LDQ81	T
AC9	PL76B	6	LDQ73	C	PL84B	6	LDQ81	C
GNDIO	GNDIO6	-			GNDIO6	-		
AC7	NC	-			PL86A	6	LDQ90	T (LVDS)*
AC5	NC	-			PL86B	6	LDQ90	C (LVDS)*
AC6	NC	-			PL87A	6	LDQ90	T
AD5	NC	-			PL87B	6	LDQ90	C
-	-	-			VCCIO6	6		
AD4	NC	-			PL88A	6	LDQ90	T (LVDS)*
AD3	NC	-			PL88B	6	LDQ90	C (LVDS)*
AD10	NC	-			PL89A	6	LDQ90	T
AD8	NC	-			PL89B	6	LDQ90	C
-	-	-			GNDIO6	-		
AD2	NC	-			PL90A	6	LDQS90	T (LVDS)*
AD1	NC	-			PL90B	6	LDQ90	C (LVDS)*
AD9	NC	-			PL91A	6	LDQ90	T
-	-	-			VCCIO6	6		
AC11	NC	-			PL91B	6	LDQ90	C
AD6	NC	-			PL92A	6	LDQ90	T (LVDS)*
AD7	NC	-			PL92B	6	LDQ90	C (LVDS)*
AE1	NC	-			PL93A	6	LDQ90	T
-	-	-			GNDIO6	-		
AE2	NC	-			PL93B	6	LDQ90	C
AF2	PL78A	6	LDQ82	T (LVDS)*	PL95A	6	LDQ99	T (LVDS)*
AF1	PL78B	6	LDQ82	C (LVDS)*	PL95B	6	LDQ99	C (LVDS)*
AE5	PL79A	6	LDQ82	T	PL96A	6	LDQ99	T
AE6	PL79B	6	LDQ82	C	PL96B	6	LDQ99	C
AF4	PL80A	6	LDQ82	T (LVDS)*	PL97A	6	LDQ99	T (LVDS)*
VCCIO	VCCIO6	6			VCCIO6	6		
AF3	PL80B	6	LDQ82	C (LVDS)*	PL97B	6	LDQ99	C (LVDS)*
AF5	PL81A	6	LDQ82	T	PL98A	6	LDQ99	T

LFE2M70E/SE and LFE2M100E/SE Logic Signal Connections: 1152 fpBGA

LFE2M70E/SE				LFE2M100E/SE				
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
AF6	PL81B	6	LDQ82	C	PL98B	6	LDQ99	C
AG1	PL82A	6	LLM0_GPLLT_IN_A**/LDQS82	T (LVDS)*	PL99A	6	LLM0_GPLLT_IN_A**/ LDQS99	T (LVDS)*
GNDIO	GNDIO6	-			GNDIO6	-		
AG2	PL82B	6	LLM0_GPLLC_IN_A**/LDQ82	C (LVDS)*	PL99B	6	LLM0_GPLLC_IN_A**/LDQ99	C (LVDS)*
AE9	PL83A	6	LLM0_GPLLT_FB_A/LDQ82	T	PL100A	6	LLM0_GPLLT_FB_A/LDQ99	T
AF7	PL83B	6	LLM0_GPLLC_FB_A/LDQ82	C	PL100B	6	LLM0_GPLLC_FB_A/LDQ99	C
VCCIO	VCCIO6	6			VCCIO6	6		
AH1	PL84A	6	LLM0_GDLLT_IN_A**/LDQ82	T (LVDS)*	PL101A	6	LLM0_GDLLT_IN_A**/LDQ99	T (LVDS)*
AH2	PL84B	6	LLM0_GDLLC_IN_A**/LDQ82	C (LVDS)*	PL101B	6	LLM0_GDLLC_IN_A**/ LDQ99	C (LVDS)*
AG5	PL85A	6	LLM0_GDLLT_FB_A/LDQ82	T	PL102A	6	LLM0_GDLLT_FB_A/LDQ99	T
AG4	PL85B	6	LLM0_GDLLC_FB_A/LDQ82	C	PL102B	6	LLM0_GDLLC_FB_A/LDQ99	C
GNDIO	GNDIO6	-			GNDIO6	-		
AG6	LLM0_PLLCAP	6			LLM0_PLLCAP	6		
AJ1	PL87A	6		T	PL104A	6		T
AJ2	PL87B	6		C	PL104B	6		C
AK2	TCK	-			TCK	-		
AK1	TDI	-			TDI	-		
AL1	TMS	-			TMS	-		
AF10	TDO	-			TDO	-		
AK3	VCCJ	-			VCCJ	-		
AN2	LLC_SQ_VCCRX3	14			LLC_SQ_VCCRX3	14		
AM2	LLC_SQ_HDINP3	14		T	LLC_SQ_HDINP3	14		T
AN1	LLC_SQ_VCCIB3	14			LLC_SQ_VCCIB3	14		
AM3	LLC_SQ_HDINN3	14		C	LLC_SQ_HDINN3	14		C
AN3	LLC_SQ_VCCTX3	14			LLC_SQ_VCCTX3	14		
AP2	LLC_SQ_HDOUTP3	14		T	LLC_SQ_HDOUTP3	14		T
AM1	LLC_SQ_VCCOB3	14			LLC_SQ_VCCOB3	14		
AP3	LLC_SQ_HDOUTN3	14		C	LLC_SQ_HDOUTN3	14		C
AN4	LLC_SQ_VCCTX2	14			LLC_SQ_VCCTX2	14		
AP4	LLC_SQ_HDOUTN2	14		C	LLC_SQ_HDOUTN2	14		C
AL3	LLC_SQ_VCCOB2	14			LLC_SQ_VCCOB2	14		
AP5	LLC_SQ_HDOUTP2	14		T	LLC_SQ_HDOUTP2	14		T
AN5	LLC_SQ_VCCRX2	14			LLC_SQ_VCCRX2	14		
AM4	LLC_SQ_HDINN2	14		C	LLC_SQ_HDINN2	14		C
AL4	LLC_SQ_VCCIB2	14			LLC_SQ_VCCIB2	14		
AM5	LLC_SQ_HDINP2	14		T	LLC_SQ_HDINP2	14		T
AL6	LLC_SQ_VCCP	14			LLC_SQ_VCCP	14		
AL5	LLC_SQ_REFCLKP	14		T	LLC_SQ_REFCLKP	14		T
AK5	LLC_SQ_REFCLKN	14		C	LLC_SQ_REFCLKN	14		C
AK6	LLC_SQ_VCCAUX33	14			LLC_SQ_VCCAUX33	14		
AM6	LLC_SQ_HDINP1	14		T	LLC_SQ_HDINP1	14		T
AL8	LLC_SQ_VCCIB1	14			LLC_SQ_VCCIB1	14		
AM7	LLC_SQ_HDINN1	14		C	LLC_SQ_HDINN1	14		C
AN6	LLC_SQ_VCCRX1	14			LLC_SQ_VCCRX1	14		
AP6	LLC_SQ_HDOUTP1	14		T	LLC_SQ_HDOUTP1	14		T
AK7	LLC_SQ_VCCOB1	14			LLC_SQ_VCCOB1	14		
AP7	LLC_SQ_HDOUTN1	14		C	LLC_SQ_HDOUTN1	14		C
AN7	LLC_SQ_VCCTX1	14			LLC_SQ_VCCTX1	14		
AP8	LLC_SQ_HDOUTN0	14		C	LLC_SQ_HDOUTN0	14		C

LFE2M70E/SE and LFE2M100E/SE Logic Signal Connections: 1152 fpBGA

LFE2M70E/SE				LFE2M100E/SE				
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
AL9	LLC_SQ_VCCOB0	14			LLC_SQ_VCCOB0	14		
AP9	LLC_SQ_HDOUTP0	14		T	LLC_SQ_HDOUTP0	14		T
AN8	LLC_SQ_VCCTX0	14			LLC_SQ_VCCTX0	14		
AM8	LLC_SQ_HDINN0	14		C	LLC_SQ_HDINN0	14		C
AN9	LLC_SQ_VCCIB0	14			LLC_SQ_VCCIB0	14		
AM9	LLC_SQ_HDINP0	14		T	LLC_SQ_HDINP0	14		T
AL7	LLC_SQ_VCCRX0	14			LLC_SQ_VCCRX0	14		
-	-	-			VCCIO5	5		
AJ12	NC	-			PB32A	5	BDQ33	T
AH12	NC	-			PB32B	5	BDQ33	C
-	-	-			GNDIO5	-		
-	-	-			VCCIO5	5		
AL13	NC	-			PB36A	5	BDQ33	T
AK13	NC	-			PB36B	5	BDQ33	C
-	-	-			GNDIO5	-		
AE14	NC	-			PB38A	5	BDQ42	T
AG13	NC	-			PB38B	5	BDQ42	C
AN14	PB30A	5	BDQ33	T	PB39A	5	BDQ42	T
AP14	PB30B	5	BDQ33	C	PB39B	5	BDQ42	C
AH14	PB31A	5	BDQ33	T	PB40A	5	BDQ42	T
AJ15	PB31B	5	BDQ33	C	PB40B	5	BDQ42	C
VCCIO	VCCIO5	5			VCCIO5	5		
GNDIO	GNDIO5	-			GNDIO5	-		
AL14	PB33A	5	BDQS33	T	PB42A	5	BDQS42	T
AM14	PB33B	5	BDQ33	C	PB42B	5	BDQ42	C
AF14	PB35A	5	BDQ33	T	PB44A	5	BDQ42	T
AF13	PB35B	5	BDQ33	C	PB44B	5	BDQ42	C
VCCIO	VCCIO5	5			VCCIO5	5		
AE15	PB36A	5	BDQ33	T	PB45A	5	BDQ42	T
AG14	PB36B	5	BDQ33	C	PB45B	5	BDQ42	C
AH15	PB37A	5	BDQ33	T	PB46A	5	BDQ42	T
AK15	PB37B	5	BDQ33	C	PB46B	5	BDQ42	C
GNDIO	GNDIO5	-			GNDIO5	-		
AL15	PB38A	5	BDQ42	T	PB47A	5	BDQ51	T
AM15	PB38B	5	BDQ42	C	PB47B	5	BDQ51	C
AK16	PB39A	5	BDQ42	T	PB48A	5	BDQ51	T
AJ16	PB39B	5	BDQ42	C	PB48B	5	BDQ51	C
AN15	PB40A	5	BDQ42	T	PB49A	5	BDQ51	T
VCCIO	VCCIO5	5			VCCIO5	5		
AP15	PB40B	5	BDQ42	C	PB49B	5	BDQ51	C
AG15	PB42A	5	BDQS42	T	PB51A	5	BDQS51	T
GNDIO	GNDIO5	-			GNDIO5	-		
AE16	PB42B	5	BDQ42	C	PB51B	5	BDQ51	C
AF15	PB44A	5	BDQ42	T	PB53A	5	BDQ51	T
VCCIO	VCCIO5	5			VCCIO5	5		
AD16	PB44B	5	BDQ42	C	PB53B	5	BDQ51	C
AK17	PB45A	5	BDQ42	T	PB54A	5	BDQ51	T
AH16	PB45B	5	BDQ42	C	PB54B	5	BDQ51	C
AN16	PB46A	5	BDQ42	T	PB55A	5	BDQ51	T
GNDIO	GNDIO5	-			GNDIO5	-		
AP16	PB46B	5	BDQ42	C	PB55B	5	BDQ51	C

LFE2M70E/SE and LFE2M100E/SE Logic Signal Connections: 1152 fpBGA

LFE2M70E/SE				LFE2M100E/SE				
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
AL17	PB47A	5	BDQ51	T	PB56A	5	BDQ60	T
AM17	PB47B	5	BDQ51	C	PB56B	5	BDQ60	C
AN17	PB48A	5	BDQ51	T	PB57A	5	BDQ60	T
AP17	PB48B	5	BDQ51	C	PB57B	5	BDQ60	C
AD17	PB49A	5	BDQ51	T	PB58A	5	BDQ60	T
AE17	PB49B	5	BDQ51	C	PB58B	5	BDQ60	C
VCCIO	VCCIO5	5			VCCIO5	5		
AL18	PB50A	5	BDQ51	T	PB59A	5	BDQ60	T
AM18	PB50B	5	BDQ51	C	PB59B	5	BDQ60	C
GNDIO	GNDIO5	-			GNDIO5	-		
AP18	PB51A	5	BDQS51	T	PB60A	5	BDQS60	T
AN18	PB51B	5	BDQ51	C	PB60B	5	BDQ60	C
AG17	PB52A	5	VREF2_5/BDQ51	T	PB61A	5	VREF2_5/BDQ60	T
AJ17	PB52B	5	VREF1_5/BDQ51	C	PB61B	5	VREF1_5/BDQ60	C
AF17	PB53A	5	PCLKT5_0/BDQ51	T	PB62A	5	PCLKT5_0/BDQ60	T
AH17	PB53B	5	PCLKC5_0/BDQ51	C	PB62B	5	PCLKC5_0/BDQ60	C
VCCIO	VCCIO5	5			VCCIO5	5		
GNDIO	GNDIO5	-			GNDIO5	-		
AF18	PB58A	4	PCLKT4_0/BDQ60	T	PB67A	4	PCLKT4_0/BDQ69	T
VCCIO	VCCIO4	4			VCCIO4	4		
AD18	PB58B	4	PCLKC4_0/BDQ60	C	PB67B	4	PCLKC4_0/BDQ69	C
AP19	PB59A	4	VREF2_4/BDQ60	T	PB68A	4	VREF2_4/BDQ69	T
AN19	PB59B	4	VREF1_4/BDQ60	C	PB68B	4	VREF1_4/BDQ69	C
AP20	PB60A	4	BDQS60	T	PB69A	4	BDQS69	T
GNDIO	GNDIO4	-			GNDIO4	-		
AM20	PB60B	4	BDQ60	C	PB69B	4	BDQ69	C
AN20	PB61A	4	BDQ60	T	PB70A	4	BDQ69	T
AM21	PB61B	4	BDQ60	C	PB70B	4	BDQ69	C
AG18	PB62A	4	BDQ60	T	PB71A	4	BDQ69	T
VCCIO	VCCIO4	4			VCCIO4	4		
AE18	PB62B	4	BDQ60	C	PB71B	4	BDQ69	C
AJ18	PB63A	4	BDQ60	T	PB72A	4	BDQ69	T
AH18	PB63B	4	BDQ60	C	PB72B	4	BDQ69	C
AK18	PB64A	4	BDQ60	T	PB73A	4	BDQ69	T
GNDIO	GNDIO4	-			GNDIO4	-		
AK19	PB64B	4	BDQ60	C	PB73B	4	BDQ69	C
AP21	PB65A	4	BDQ69	T	PB74A	4	BDQ78	T
AN21	PB65B	4	BDQ69	C	PB74B	4	BDQ78	C
AL20	PB66A	4	BDQ69	T	PB75A	4	BDQ78	T
AK20	PB66B	4	BDQ69	C	PB75B	4	BDQ78	C
AN22	PB67A	4	BDQ69	T	PB76A	4	BDQ78	T
AL21	PB67B	4	BDQ69	C	PB76B	4	BDQ78	C
VCCIO	VCCIO4	4			VCCIO4	4		
GNDIO	GNDIO4	-			GNDIO4	-		
AH19	PB69A	4	BDQS69	T	PB78A	4	BDQS78	T
AJ20	PB69B	4	BDQ69	C	PB78B	4	BDQ78	C
AD20	PB71A	4	BDQ69	T	PB80A	4	BDQ78	T
AF20	PB71B	4	BDQ69	C	PB80B	4	BDQ78	C
VCCIO	VCCIO4	4			VCCIO4	4		
AJ19	PB72A	4	BDQ69	T	PB81A	4	BDQ78	T
AH20	PB72B	4	BDQ69	C	PB81B	4	BDQ78	C

LFE2M70E/SE and LFE2M100E/SE Logic Signal Connections: 1152 fpBGA

LFE2M70E/SE				LFE2M100E/SE				
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
AE20	PB73A	4	BDQ69	T	PB82A	4	BDQ78	T
AG20	PB73B	4	BDQ69	C	PB82B	4	BDQ78	C
GNDIO	GNDIO4	-			GNDIO4	-		
AH22	NC	-			PB89A	4	BDQ87	T
-	-	-			VCCIO4	4		
AH21	NC	-			PB89B	4	BDQ87	C
AG22	NC	-			PB90A	4	BDQ87	T
AG21	NC	-			PB90B	4	BDQ87	C
-	-	-			GNDIO4	-		
AM22	PB74A	4	BDQ78	T	PB92A	4	BDQ96	T
AL22	PB74B	4	BDQ78	C	PB92B	4	BDQ96	C
VCCIO	VCCIO4	4			VCCIO4	4		
AP23	PB77A	4	BDQ78	T	PB95A	4	BDQ96	T
AN23	PB77B	4	BDQ78	C	PB95B	4	BDQ96	C
GNDIO	GNDIO4	-			GNDIO4	-		
AM24	PB78A	4	BDQS78	T	PB96A	4	BDQS96	T
AL24	PB78B	4	BDQ78	C	PB96B	4	BDQ96	C
AK22	PB79A	4	BDQ78	T	PB97A	4	BDQ96	T
AJ22	PB79B	4	BDQ78	C	PB97B	4	BDQ96	C
AL23	PB80A	4	BDQ78	T	PB98A	4	BDQ96	T
AK23	PB80B	4	BDQ78	C	PB98B	4	BDQ96	C
VCCIO	VCCIO4	4			VCCIO4	4		
AJ23	PB81A	4	BDQ78	T	PB99A	4	BDQ96	T
AH23	PB81B	4	BDQ78	C	PB99B	4	BDQ96	C
GNDIO	GNDIO4	-			GNDIO4	-		
AL28	LRC_SQ_VCCRX3	13			LRC_SQ_VCCRX3	13		
AM26	LRC_SQ_HDINP3	13		T	LRC_SQ_HDINP3	13		T
AN26	LRC_SQ_VCCIB3	13			LRC_SQ_VCCIB3	13		
AM27	LRC_SQ_HDINN3	13		C	LRC_SQ_HDINN3	13		C
AN27	LRC_SQ_VCCTX3	13			LRC_SQ_VCCTX3	13		
AP26	LRC_SQ_HDOUTP3	13		T	LRC_SQ_HDOUTP3	13		T
AL26	LRC_SQ_VCCOB3	13			LRC_SQ_VCCOB3	13		
AP27	LRC_SQ_HDOUTN3	13		C	LRC_SQ_HDOUTN3	13		C
AN28	LRC_SQ_VCCTX2	13			LRC_SQ_VCCTX2	13		
AP28	LRC_SQ_HDOUTN2	13		C	LRC_SQ_HDOUTN2	13		C
AK28	LRC_SQ_VCCOB2	13			LRC_SQ_VCCOB2	13		
AP29	LRC_SQ_HDOUTP2	13		T	LRC_SQ_HDOUTP2	13		T
AN29	LRC_SQ_VCCRX2	13			LRC_SQ_VCCRX2	13		
AM28	LRC_SQ_HDINN2	13		C	LRC_SQ_HDINN2	13		C
AL27	LRC_SQ_VCCIB2	13			LRC_SQ_VCCIB2	13		
AM29	LRC_SQ_HDINP2	13		T	LRC_SQ_HDINP2	13		T
AL29	LRC_SQ_VCCP	13			LRC_SQ_VCCP	13		
AL30	LRC_SQ_REFCLKP	13		T	LRC_SQ_REFCLKP	13		T
AK30	LRC_SQ_REFCLKN	13		C	LRC_SQ_REFCLKN	13		C
AK29	LRC_SQ_VCCAUX33	13			LRC_SQ_VCCAUX33	13		
AM30	LRC_SQ_HDINP1	13		T	LRC_SQ_HDINP1	13		T
AL31	LRC_SQ_VCCIB1	13			LRC_SQ_VCCIB1	13		
AM31	LRC_SQ_HDINN1	13		C	LRC_SQ_HDINN1	13		C
AN30	LRC_SQ_VCCRX1	13			LRC_SQ_VCCRX1	13		
AP30	LRC_SQ_HDOUTP1	13		T	LRC_SQ_HDOUTP1	13		T
AL32	LRC_SQ_VCCOB1	13			LRC_SQ_VCCOB1	13		

LFE2M70E/SE and LFE2M100E/SE Logic Signal Connections: 1152 fpBGA

LFE2M70E/SE				LFE2M100E/SE				
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
AP31	LRC_SQ_HDOUTN1	13		C	LRC_SQ_HDOUTN1	13		C
AN31	LRC_SQ_VCCTX1	13			LRC_SQ_VCCTX1	13		
AP32	LRC_SQ_HDOUTN0	13		C	LRC_SQ_HDOUTN0	13		C
AM34	LRC_SQ_VCCOB0	13			LRC_SQ_VCCOB0	13		
AP33	LRC_SQ_HDOUTP0	13		T	LRC_SQ_HDOUTP0	13		T
AN32	LRC_SQ_VCCTX0	13			LRC_SQ_VCCTX0	13		
AM32	LRC_SQ_HDINN0	13		C	LRC_SQ_HDINN0	13		C
AN34	LRC_SQ_VCCIB0	13			LRC_SQ_VCCIB0	13		
AM33	LRC_SQ_HDINP0	13		T	LRC_SQ_HDINP0	13		T
AN33	LRC_SQ_VCCRX0	13			LRC_SQ_VCCRX0	13		
AH28	CFG2	8			CFG2	8		
AD24	CFG1	8			CFG1	8		
AJ29	CFG0	8			CFG0	8		
AF25	PROGRAMN	8			PROGRAMN	8		
AJ28	CCLK	8			CCLK	8		
AE25	INITN	8			INITN	8		
AK31	DONE	8			DONE	8		
GNDIO	GNDIO8	-			GNDIO8	-		
AE24	WRITEN***	8			WRITEN***	8		
AJ30	CS1N***	8			CS1N***	8		
AD25	CSN***	8			CSN***	8		
AG29	D0/SPIFASTN***	8			D0/SPIFASTN***	8		
VCCIO	VCCIO8	8			VCCIO8	8		
AG28	D1***	8			D1***	8		
AG30	D2***	8			D2***	8		
AH29	D3***	8			D3***	8		
GNDIO	GNDIO8	-			GNDIO8	-		
AF26	D4***	8			D4***	8		
AH30	D5***	8			D5***	8		
AE26	D6***	8			D6***	8		
AJ31	D7/SPID0***	8			D7/SPID0***	8		
VCCIO	VCCIO8	8			VCCIO8	8		
AG27	DI/CSSPI0N***	8			DI/CSSPI0N***	8		
AK32	DOUT/CSON/ CSSPI1N***	8			DOUT/CSON/ CSSPI1N***	8		
AK33	BUSY/SISPI***	8			BUSY/SISPI***	8		
AF27	RLM0_PLLCAP	3			RLM0_PLLCAP	3		
AF28	PR85B	3	RLM0_GDLLC_FB_A	C	PR102B	3	RLM0_GDLLC_FB_A/RDQ99	C
GNDIO	GNDIO3	-			GNDIO3	-		
AD26	PR85A	3	RLM0_GDLLT_FB_A	T	PR102A	3	RLM0_GDLLT_FB_A/RDQ99	T
AJ32	PR84B	3	RLM0_GDLLC_IN_A**	C (LVDS)*	PR101B	3	RLM0_GDLLC_IN_A**/ RDQ99	C (LVDS)*
AJ33	PR84A	3	RLM0_GDLLT_IN_A**	T (LVDS)*	PR101A	3	RLM0_GDLLT_IN_A**/ RDQ99	T (LVDS)*
AJ34	PR83B	3	RLM0_GPLLC_IN_A**	C	PR100B	3	RLM0_GPLLC_IN_A**/ RDQ99	C
VCCIO	VCCIO3	3			VCCIO3	3		
AK34	PR83A	3	RLM0_GPLLT_IN_A**	T	PR100A	3	RLM0_GPLLT_IN_A**/ RDQ99	T
AH33	PR82B	3	RLM0_GPLLC_FB_A	C (LVDS)*	PR99B	3	RLM0_GPLLC_FB_A/RDQ99	C (LVDS)*
AH34	PR82A	3	RLM0_GPLLT_FB_A/RDQS82****	T (LVDS)*	PR99A	3	RLM0_GPLLT_FB_A/ RDQS99	T (LVDS)*
GNDIO	GNDIO3	-			GNDIO3	-		

LFE2M70E/SE and LFE2M100E/SE Logic Signal Connections: 1152 fpBGA

LFE2M70E/SE				LFE2M100E/SE				
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
AF29	PR81B	3	RDQ82	C	PR98B	3	RDQ99	C
AF31	PR81A	3	RDQ82	T	PR98A	3	RDQ99	T
AG33	PR80B	3	RDQ82	C (LVDS)*	PR97B	3	RDQ99	C (LVDS)*
AG34	PR80A	3	RDQ82	T (LVDS)*	PR97A	3	RDQ99	T (LVDS)*
VCCIO	VCCIO3	3			VCCIO3	3		
AF30	PR79B	3	RDQ82	C	PR96B	3	RDQ99	C
AF32	PR79A	3	RDQ82	T	PR96A	3	RDQ99	T
AE29	PR78B	3	RDQ82	C (LVDS)*	PR95B	3	RDQ99	C (LVDS)*
AE30	PR78A	3	RDQ82	T (LVDS)*	PR95A	3	RDQ99	T (LVDS)*
AF33	NC	-			PR93B	3	RDQ90	C
AF34	NC	-			PR93A	3	RDQ90	T
-	-	-			GNDIO3	-		
AC27	NC	-			PR92B	3	RDQ90	C (LVDS)*
AC28	NC	-			PR92A	3	RDQ90	T (LVDS)*
AD29	NC	-			PR91B	3	RDQ90	C
AD30	NC	-			PR91A	3	RDQ90	T
-	-	-			VCCIO3	3		
AE33	NC	-			PR90B	3	RDQ90	C (LVDS)*
AE34	NC	-			PR90A	3	RDQS90	T (LVDS)*
AD32	NC	-			PR89B	3	RDQ90	C
-	-	-			GNDIO3	-		
AD31	NC	-			PR89A	3	RDQ90	T
AB25	NC	-			PR88B	3	RDQ90	C (LVDS)*
AC25	NC	-			PR88A	3	RDQ90	T (LVDS)*
AB28	NC	-			PR87B	3	RDQ90	C
-	-	-			VCCIO3	3		
AA26	NC	-			PR87A	3	RDQ90	T
AD33	NC	-			PR86B	3	RDQ90	C (LVDS)*
AD34	NC	-			PR86A	3	RDQ90	T (LVDS)*
AC29	PR76B	3	RDQ73	C	PR84B	3	RDQ81	C
GNDIO	GNDIO3	-			GNDIO3	-		
AA27	PR76A	3	RDQ73	T	PR84A	3	RDQ81	T
AC32	PR75B	3	RDQ73	C (LVDS)*	PR83B	3	RDQ81	C (LVDS)*
AC31	PR75A	3	RDQ73	T (LVDS)*	PR83A	3	RDQ81	T (LVDS)*
AA25	PR74B	3	RDQ73	C	PR82B	3	RDQ81	C
VCCIO	VCCIO3	3			VCCIO3	3		
AC24	PR74A	3	RDQ73	T	PR82A	3	RDQ81	T
AC33	PR73B	3	RDQ73	C (LVDS)*	PR81B	3	RDQ81	C (LVDS)*
AC34	PR73A	3	RDQS73	T (LVDS)*	PR81A	3	RDQS81	T (LVDS)*
GNDIO	GNDIO3	-			GNDIO3	-		
AB24	PR72B	3	RDQ73	C	PR80B	3	RDQ81	C
Y26	PR72A	3	RDQ73	T	PR80A	3	RDQ81	T
AB33	PR71B	3	RDQ73	C (LVDS)*	PR79B	3	RDQ81	C (LVDS)*
AB34	PR71A	3	RDQ73	T (LVDS)*	PR79A	3	RDQ81	T (LVDS)*
VCCIO	VCCIO3	3			VCCIO3	3		
Y27	PR70B	3	RDQ73	C	PR78B	3	RDQ81	C
AB29	PR70A	3	RDQ73	T	PR78A	3	RDQ81	T
AA34	PR69B	3	RDQ73	C (LVDS)*	PR77B	3	RDQ81	C (LVDS)*
AA33	PR69A	3	RDQ73	T (LVDS)*	PR77A	3	RDQ81	T (LVDS)*
AA31	PR67B	3	RDQ64	C	PR75B	3	RDQ72	C
AA32	PR67A	3	RDQ64	T	PR75A	3	RDQ72	T

LFE2M70E/SE and LFE2M100E/SE Logic Signal Connections: 1152 fpBGA

LFE2M70E/SE				LFE2M100E/SE				
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
GNDIO	GNDIO3	-			GNDIO3	-		
AA28	PR66B	3	RDQ64	C (LVDS)*	PR74B	3	RDQ72	C (LVDS)*
AA29	PR66A	3	RDQ64	T (LVDS)*	PR74A	3	RDQ72	T (LVDS)*
AA30	PR65B	3	RDQ64	C	PR73B	3	RDQ72	C
AB30	PR65A	3	RDQ64	T	PR73A	3	RDQ72	T
VCCIO	VCCIO3	3			VCCIO3	3		
Y28	PR64B	3	RDQ64	C (LVDS)*	PR72B	3	RDQ72	C (LVDS)*
Y29	PR64A	3	RDQS64	T (LVDS)*	PR72A	3	RDQS72	T (LVDS)*
AA24	PR63B	3	RDQ64	C	PR71B	3	RDQ72	C
GNDIO	GNDIO3	-			GNDIO3	-		
Y25	PR63A	3	RDQ64	T	PR71A	3	RDQ72	T
Y31	PR62B	3	RDQ64	C (LVDS)*	PR70B	3	RDQ72	C (LVDS)*
Y30	PR62A	3	RDQ64	T (LVDS)*	PR70A	3	RDQ72	T (LVDS)*
Y24	PR61B	3	RDQ64	C	PR69B	3	RDQ72	C
VCCIO	VCCIO3	3			VCCIO3	3		
W25	PR61A	3	RDQ64	T	PR69A	3	RDQ72	T
Y33	PR60B	3	RDQ64	C (LVDS)*	PR68B	3	RDQ72	C (LVDS)*
Y34	PR60A	3	RDQ64	T (LVDS)*	PR68A	3	RDQ72	T (LVDS)*
W28	PR58B	3	RLM3_SPLLC_FB_A/RDQ55	C	PR66B	3	RLM4_SPLLC_FB_A/RDQ63	C
GNDIO	GNDIO3	-			GNDIO3	-		
V26	PR58A	3	RLM3_SPLLT_FB_A/RDQ55	T	PR66A	3	RLM4_SPLLT_FB_A/RDQ63	T
V28	PR57B	3	RLM3_SPLLC_IN_A/RDQ55	C (LVDS)*	PR65B	3	RLM4_SPLLC_IN_A/RDQ63	C (LVDS)*
V27	PR57A	3	RLM3_SPLLT_IN_A/RDQ55	T (LVDS)*	PR65A	3	RLM4_SPLLT_IN_A/RDQ63	T (LVDS)*
V25	PR56B	3	RDQ55	C	PR64B	3	RDQ63	C
VCCIO	VCCIO3	3			VCCIO3	3		
W24	PR56A	3	RDQ55	T	PR64A	3	RDQ63	T
W33	PR55B	3	RDQ55	C (LVDS)*	PR63B	3	RDQ63	C (LVDS)*
W34	PR55A	3	RDQS55	T (LVDS)*	PR63A	3	RDQS63	T (LVDS)*
GNDIO	GNDIO3	-			GNDIO3	-		
V24	PR54B	3	RDQ55	C	PR62B	3	RDQ63	C
U26	PR54A	3	RDQ55	T	PR62A	3	RDQ63	T
W29	PR53B	3	RDQ55	C (LVDS)*	PR61B	3	RDQ63	C (LVDS)*
W30	PR53A	3	RDQ55	T (LVDS)*	PR61A	3	RDQ63	T (LVDS)*
VCCIO	VCCIO3	3			VCCIO3	3		
U27	PR52B	3	VREF2_3/RDQ55	C	PR60B	3	VREF2_3/RDQ63	C
V29	PR52A	3	VREF1_3/RDQ55	T	PR60A	3	VREF1_3/RDQ63	T
V31	PR51B	3	PCLKC3_0/RDQ55	C (LVDS)*	PR59B	3	PCLKC3_0/RDQ63	C (LVDS)*
V32	PR51A	3	PCLKT3_0/RDQ55	T (LVDS)*	PR59A	3	PCLKT3_0/RDQ63	T (LVDS)*
V33	PR49B	2	PCLKC2_0/RDQ46	C	PR57B	2	PCLKC2_0/RDQ54	C
V34	PR49A	2	PCLKT2_0/RDQ46	T	PR57A	2	PCLKT2_0/RDQ54	T
GNDIO	GNDIO2	-			GNDIO2	-		
U24	PR48B	2	RDQ46	C (LVDS)*	PR56B	2	RDQ54	C (LVDS)*
U25	PR48A	2	RDQ46	T (LVDS)*	PR56A	2	RDQ54	T (LVDS)*
V30	PR47B	2	RDQ46	C	PR55B	2	RDQ54	C
Y32	PR47A	2	RDQ46	T	PR55A	2	RDQ54	T
VCCIO	VCCIO2	2			VCCIO2	2		
U28	PR46B	2	RDQ46	C (LVDS)*	PR54B	2	RDQ54	C (LVDS)*
U29	PR46A	2	RDQS46	T (LVDS)*	PR54A	2	RDQS54	T (LVDS)*
U33	PR45B	2	RDQ46	C	PR53B	2	RDQ54	C
GNDIO	GNDIO2	-			GNDIO2	-		
U34	PR45A	2	RDQ46	T	PR53A	2	RDQ54	T

LFE2M70E/SE and LFE2M100E/SE Logic Signal Connections: 1152 fpBGA

LFE2M70E/SE				LFE2M100E/SE				
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
T30	PR44B	2	RDQ46	C (LVDS)*	PR52B	2	RDQ54	C (LVDS)*
U30	PR44A	2	RDQ46	T (LVDS)*	PR52A	2	RDQ54	T (LVDS)*
T29	PR43B	2	RUM3_SPLLC_FB_A/RDQ46	C	PR51B	2	RUM3_SPLLC_FB_A/RDQ54	C
VCCIO	VCCIO2	2			VCCIO2	2		
T28	PR43A	2	RUM3_SPLLT_FB_A/RDQ46	T	PR51A	2	RUM3_SPLLT_FB_A/RDQ54	T
U31	PR42B	2	RUM3_SPLLC_IN_A/RDQ46	C (LVDS)*	PR50B	2	RUM3_SPLLC_IN_A/RDQ54	C (LVDS)*
U32	PR42A	2	RUM3_SPLLT_IN_A/RDQ46	T (LVDS)*	PR50A	2	RUM3_SPLLT_IN_A/RDQ54	T (LVDS)*
T33	PR40B	2	RDQ37	C	PR48B	2	RDQ45	C
T34	PR40A	2	RDQ37	T	PR48A	2	RDQ45	T
GNDIO	GNDIO2	-			GNDIO2	-		
R27	PR39B	2	RDQ37	C (LVDS)*	PR47B	2	RDQ45	C (LVDS)*
R28	PR39A	2	RDQ37	T (LVDS)*	PR47A	2	RDQ45	T (LVDS)*
R29	PR38B	2	RDQ37	C	PR46B	2	RDQ45	C
R30	PR38A	2	RDQ37	T	PR46A	2	RDQ45	T
VCCIO	VCCIO2	2			VCCIO2	2		
R33	PR37B	2	RDQ37	C (LVDS)*	PR45B	2	RDQ45	C (LVDS)*
R34	PR37A	2	RDQS37	T (LVDS)*	PR45A	2	RDQS45	T (LVDS)*
R32	PR36B	2	RDQ37	C	PR44B	2	RDQ45	C
GNDIO	GNDIO2	-			GNDIO2	-		
R31	PR36A	2	RDQ37	T	PR44A	2	RDQ45	T
P34	PR35B	2	RDQ37	C (LVDS)*	PR43B	2	RDQ45	C (LVDS)*
P33	PR35A	2	RDQ37	T (LVDS)*	PR43A	2	RDQ45	T (LVDS)*
R26	PR34B	2	RDQ37	C	PR42B	2	RDQ45	C
VCCIO	VCCIO2	2			VCCIO2	2		
T25	PR34A	2	RDQ37	T	PR42A	2	RDQ45	T
P28	PR33B	2	RDQ37	C (LVDS)*	PR41B	2	RDQ45	C (LVDS)*
P27	PR33A	2	RDQ37	T (LVDS)*	PR41A	2	RDQ45	T (LVDS)*
P30	NC	-			PR40B	2		C
-	-	-			GNDIO2	-		
P29	NC	-			PR40A	2		T
P31	NC	-			PR39B	2		C (LVDS)*
P32	NC	-			PR39A	2		T (LVDS)*
R25	NC	-			PR38B	2		C
-	-	-			VCCIO2	2		
T24	NC	-			PR38A	2		T
N34	NC	-			PR37B	2		C (LVDS)*
N33	NC	-			PR37A	2		T (LVDS)*
GNDIO	GNDIO2	-			GNDIO2	-		
M34	PR31B	2	RDQ28	C	PR35B	2	RDQ32	C
M33	PR31A	2	RDQ28	T	PR35A	2	RDQ32	T
-	-	-			GNDIO2	-		
R24	PR30B	2	RDQ28	C (LVDS)*	PR34B	2	RDQ32	C (LVDS)*
P24	PR30A	2	RDQ28	T (LVDS)*	PR34A	2	RDQ32	T (LVDS)*
N30	PR29B	2	RDQ28	C	PR33B	2	RDQ32	C
M29	PR29A	2	RDQ28	T	PR33A	2	RDQ32	T
VCCIO	VCCIO2	2			VCCIO2	2		
N28	PR28B	2	RDQ28	C (LVDS)*	PR32B	2	RDQ32	C (LVDS)*
N29	PR28A	2	RDQS28	T (LVDS)*	PR32A	2	RDQS32	T (LVDS)*
N24	PR27B	2	RDQ28	C	PR31B	2	RDQ32	C
GNDIO	GNDIO2	-			GNDIO2	-		
N25	PR27A	2	RDQ28	T	PR31A	2	RDQ32	T

LFE2M70E/SE and LFE2M100E/SE Logic Signal Connections: 1152 fpBGA

LFE2M70E/SE				LFE2M100E/SE				
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
M28	PR26B	2	RDQ28	C (LVDS)*	PR30B	2	RDQ32	C (LVDS)*
M27	PR26A	2	RDQ28	T (LVDS)*	PR30A	2	RDQ32	T (LVDS)*
L27	PR25B	2	RDQ28	C	PR29B	2	RDQ32	C
VCCIO	VCCIO2	2			VCCIO2	2		
M26	PR25A	2	RDQ28	T	PR29A	2	RDQ32	T
M32	PR24B	2	RDQ28	C (LVDS)*	PR28B	2	RDQ32	C (LVDS)*
M31	PR24A	2	RDQ28	T (LVDS)*	PR28A	2	RDQ32	T (LVDS)*
GNDIO	GNDIO2	-			GNDIO2	-		
-	-	-			VCCIO2	2		
L34	PR22B	2		C	PR22B	2	RDQ23	C
L33	PR22A	2		T	PR22A	2	RDQ23	T
L32	PR21B	2		C (LVDS)*	PR21B	2	RDQ23	C (LVDS)*
L31	PR21A	2		T (LVDS)*	PR21A	2	RDQ23	T (LVDS)*
VCCIO	VCCIO2	2			VCCIO2	2		
L28	PR20B	2		C	PR20B	2	RDQ23	C
L29	PR20A	2		T	PR20A	2	RDQ23	T
M30	PR19B	2		C (LVDS)*	PR19B	2	RDQ23	C (LVDS)*
L30	PR19A	2		T (LVDS)*	PR19A	2	RDQ23	T (LVDS)*
K34	PR18B	2	RDQ15	C	PR18B	2	RDQ15	C
K33	PR18A	2	RDQ15	T	PR18A	2	RDQ15	T
GNDIO	GNDIO2	-			GNDIO2	-		
K30	PR17B	2	RDQ15	C (LVDS)*	PR17B	2	RDQ15	C (LVDS)*
K29	PR17A	2	RDQ15	T (LVDS)*	PR17A	2	RDQ15	T (LVDS)*
J34	PR16B	2	RDQ15	C	PR16B	2	RDQ15	C
J33	PR16A	2	RDQ15	T	PR16A	2	RDQ15	T
VCCIO	VCCIO2	2			VCCIO2	2		
J32	PR15B	2	RDQ15	C (LVDS)*	PR15B	2	RDQ15	C (LVDS)*
J31	PR15A	2	RDQS15	T (LVDS)*	PR15A	2	RDQS15	T (LVDS)*
H33	PR14B	2	RDQ15	C	PR14B	2	RDQ15	C
GNDIO	GNDIO2	-			GNDIO2	-		
H34	PR14A	2	RDQ15	T	PR14A	2	RDQ15	T
J30	PR13B	2	RDQ15	C (LVDS)*	PR13B	2	RDQ15	C (LVDS)*
J29	PR13A	2	RDQ15	T (LVDS)*	PR13A	2	RDQ15	T (LVDS)*
VCCIO	VCCIO2	2			VCCIO2	2		
J27	PR11B	2	RUM0_SPLLC_IN_A/RDQ15	C (LVDS)*	PR11B	2	RUM0_SPLLC_IN_A/RDQ15	C (LVDS)*
J28	PR11A	2	RUM0_SPLLT_IN_A/RDQ15	T (LVDS)*	PR11A	2	RUM0_SPLLT_IN_A/RDQ15	T (LVDS)*
H31	PR9B	2	VREF2_2	C	PR9B	2	VREF2_2	C
GNDIO	GNDIO2	-			GNDIO2	-		
H32	PR9A	2	VREF1_2	T	PR9A	2	VREF1_2	T
VCCIO	VCCIO2	2			VCCIO2	2		
H30	XRES	1			XRES	1		
B33	URC_SQ_VCCRX0	12			URC_SQ_VCCRX0	12		
C33	URC_SQ_HDINP0	12		T	URC_SQ_HDINP0	12		T
B34	URC_SQ_VCCIB0	12			URC_SQ_VCCIB0	12		
C32	URC_SQ_HDINN0	12		C	URC_SQ_HDINN0	12		C
B32	URC_SQ_VCCTX0	12			URC_SQ_VCCTX0	12		
A33	URC_SQ_HDOUTP0	12		T	URC_SQ_HDOUTP0	12		T
C34	URC_SQ_VCCOB0	12			URC_SQ_VCCOB0	12		
A32	URC_SQ_HDOUTN0	12		C	URC_SQ_HDOUTN0	12		C
B31	URC_SQ_VCCTX1	12			URC_SQ_VCCTX1	12		
A31	URC_SQ_HDOUTN1	12		C	URC_SQ_HDOUTN1	12		C

LFE2M70E/SE and LFE2M100E/SE Logic Signal Connections: 1152 fpBGA

LFE2M70E/SE				LFE2M100E/SE				
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
D32	URC_SQ_VCCOB1	12			URC_SQ_VCCOB1	12		
A30	URC_SQ_HDOUTP1	12		T	URC_SQ_HDOUTP1	12		T
B30	URC_SQ_VCCRX1	12			URC_SQ_VCCRX1	12		
C31	URC_SQ_HDINN1	12		C	URC_SQ_HDINN1	12		C
D31	URC_SQ_VCCIB1	12			URC_SQ_VCCIB1	12		
C30	URC_SQ_HDINP1	12		T	URC_SQ_HDINP1	12		T
E29	URC_SQ_VCCAUX33	12			URC_SQ_VCCAUX33	12		
E30	URC_SQ_REFCLKN	12		C	URC_SQ_REFCLKN	12		C
D30	URC_SQ_REFCLKP	12		T	URC_SQ_REFCLKP	12		T
D29	URC_SQ_VCCP	12			URC_SQ_VCCP	12		
C29	URC_SQ_HDINP2	12		T	URC_SQ_HDINP2	12		T
D27	URC_SQ_VCCIB2	12			URC_SQ_VCCIB2	12		
C28	URC_SQ_HDINN2	12		C	URC_SQ_HDINN2	12		C
B29	URC_SQ_VCCRX2	12			URC_SQ_VCCRX2	12		
A29	URC_SQ_HDOUTP2	12		T	URC_SQ_HDOUTP2	12		T
E28	URC_SQ_VCCOB2	12			URC_SQ_VCCOB2	12		
A28	URC_SQ_HDOUTN2	12		C	URC_SQ_HDOUTN2	12		C
B28	URC_SQ_VCCTX2	12			URC_SQ_VCCTX2	12		
A27	URC_SQ_HDOUTN3	12		C	URC_SQ_HDOUTN3	12		C
D26	URC_SQ_VCCOB3	12			URC_SQ_VCCOB3	12		
A26	URC_SQ_HDOUTP3	12		T	URC_SQ_HDOUTP3	12		T
B27	URC_SQ_VCCTX3	12			URC_SQ_VCCTX3	12		
C27	URC_SQ_HDINN3	12		C	URC_SQ_HDINN3	12		C
B26	URC_SQ_VCCIB3	12			URC_SQ_VCCIB3	12		
C26	URC_SQ_HDINP3	12		T	URC_SQ_HDINP3	12		T
D28	URC_SQ_VCCRX3	12			URC_SQ_VCCRX3	12		
E23	PT82B	1		C	PT100B	1		C
GNDIO	GNDIO1	-			GNDIO1	-		
F23	PT82A	1		T	PT100A	1		T
F24	NC	-			PT99B	1		C
G23	NC	-			PT99A	1		T
D23	PT80B	1		C	PT98B	1		C
VCCIO	VCCIO1	1			VCCIO1	1		
D22	PT80A	1		T	PT98A	1		T
-	-	-			GNDIO1	-		
-	-	-			VCCIO1	1		
C21	PT79B	1		C	PT88B	1		C
D21	PT79A	1		T	PT88A	1		T
GNDIO	GNDIO1	-			GNDIO1	-		
B21	PT77B	1		C	PT86B	1		C
A21	PT77A	1		T	PT86A	1		T
F22	PT76B	1		C	PT85B	1		C
E22	PT76A	1		T	PT85A	1		T
VCCIO	VCCIO1	1			VCCIO1	1		
GNDIO	GNDIO1	-			-	-		
J22	NC	-			PT84B	1		C
G22	NC	-			PT84A	1		T
-	-	-			GNDIO1	-		
H22	PT72B	1		C	PT81B	1		C
K22	PT72A	1		T	PT81A	1		T
G21	PT71B	1		C	PT80B	1		C

LFE2M70E/SE and LFE2M100E/SE Logic Signal Connections: 1152 fpBGA

LFE2M70E/SE				LFE2M100E/SE				
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
VCCIO	VCCIO1	1			VCCIO1	1		
J21	PT71A	1		T	PT80A	1		T
H21	NC	-			PT79B	1		C
K21	NC	-			PT79A	1		T
D20	PT69B	1		C	PT78B	1		C
F20	PT69A	1		T	PT78A	1		T
C20	PT68B	1		C	PT77B	1		C
GNDIO	GNDIO1	-			GNDIO1	-		
E20	PT68A	1		T	PT77A	1		T
G20	PT67B	1		C	PT76B	1		C
VCCIO	VCCIO1	1			VCCIO1	1		
J20	PT67A	1		T	PT76A	1		T
A20	PT66B	1		C	PT75B	1		C
B20	PT66A	1		T	PT75A	1		T
GNDIO	GNDIO1	-			GNDIO1	-		
A19	PT63B	1		C	PT72B	1		C
B19	PT63A	1		T	PT72A	1		T
K20	PT62B	1		C	PT71B	1		C
H20	PT62A	1		T	PT71A	1		T
VCCIO	VCCIO1	1			VCCIO1	1		
L19	NC	-			PT70B	1		C
L20	NC	-			PT70A	1		T
E19	PT60B	1		C	PT69B	1		C
C18	PT60A	1		T	PT69A	1		T
GNDIO	GNDIO1	-			GNDIO1	-		
F19	PT59B	1		C	PT68B	1		C
D18	PT59A	1		T	PT68A	1		T
L18	NC	-			PT67B	1		C
K19	NC	-			PT67A	1		T
VCCIO	VCCIO1	1			VCCIO1	1		
A18	PT57B	1	VREF2_1	C	PT66B	1	VREF2_1	C
B18	PT57A	1	VREF1_1	T	PT66A	1	VREF1_1	T
G18	PT56B	1	PCLKC1_0	C	PT65B	1	PCLKC1_0	C
E18	PT56A	1	PCLKT1_0	T	PT65A	1	PCLKT1_0	T
F18	PT55B	0	PCLKC0_0	C	PT64B	0	PCLKC0_0	C
GNDIO	GNDIO0	-			GNDIO0	-		
G19	PT55A	0	PCLKT0_0	T	PT64A	0	PCLKT0_0	T
H18	PT54B	0	VREF2_0	C	PT63B	0	VREF2_0	C
K18	PT54A	0	VREF1_0	T	PT63A	0	VREF1_0	T
VCCIO	VCCIO0	0			VCCIO0	0		
J18	PT53B	0		C	PT60B	0		C
L17	PT53A	0		T	PT60A	0		T
G17	PT52B	0		C	PT59B	0		C
-	-	-			GNDIO0	-		
J17	PT52A	0		T	PT59A	0		T
H17	PT51B	0		C	PT58B	0		C
-	-	-			VCCIO0	0		
K17	PT51A	0		T	PT58A	0		T
B17	PT50B	0		C	PT57B	0		C
GNDIO	GNDIO0	-			-	-		
A17	PT50A	0		T	PT57A	0		T

LFE2M70E/SE and LFE2M100E/SE Logic Signal Connections: 1152 fpBGA

LFE2M70E/SE				LFE2M100E/SE				
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
D17	PT49B	0		C	PT56B	0		C
VCCIO	VCCIO0	0			-	-		
F17	PT49A	0		T	PT56A	0		T
B16	PT48B	0		C	PT55B	0		C
A16	PT48A	0		T	PT55A	0		T
-	-	-			GNDIO0	-		
-	-	-			VCCIO0	0		
E17	PT47B	0		C	PT52B	0		C
C17	PT47A	0		T	PT52A	0		T
K16	PT46B	0		C	PT51B	0		C
J15	PT46A	0		T	PT51A	0		T
GNDIO	GNDIO0	-			GNDIO0	-		
G16	PT45B	0		C	PT50B	0		C
H15	PT45A	0		T	PT50A	0		T
A15	PT44B	0		C	PT49B	0		C
B15	PT44A	0		T	PT49A	0		T
VCCIO	VCCIO0	0			VCCIO0	0		
L16	PT43B	0		C	PT48B	0		C
K15	PT43A	0		T	PT48A	0		T
F16	PT42B	0		C	PT47B	0		C
E16	PT42A	0		T	PT47A	0		T
E15	PT41B	0		C	PT46B	0		C
GNDIO	GNDIO0	-			GNDIO0	-		
G15	PT41A	0		T	PT46A	0		T
J14	NC	-			PT45B	0		C
L15	NC	-			PT45A	0		T
H14	NC	-			PT44B	0		C
VCCIO	VCCIO0	0			VCCIO0	0		
K14	NC	-			PT44A	0		T
F15	PT38B	0		C	PT42B	0		C
G14	PT38A	0		T	PT42A	0		T
C15	PT37B	0		C	PT41B	0		C
GNDIO	GNDIO0	-			GNDIO0	-		
D14	PT37A	0		T	PT41A	0		T
G13	PT36B	0		C	PT40B	0		C
-	-	-			VCCIO0	0		
J13	PT36A	0		T	PT40A	0		T
B14	PT35B	0		C	PT39B	0		C
VCCIO	VCCIO0	0			-	-		
A14	PT35A	0		T	PT39A	0		T
F13	PT34B	0		C	PT38B	0		C
H13	PT34A	0		T	PT38A	0		T
D13	PT33B	0		C	PT37B	0		C
C14	PT33A	0		T	PT37A	0		T
GNDIO	GNDIO0	-			GNDIO0	-		
E13	PT32B	0		C	PT32B	0		C
D12	PT32A	0		T	PT32A	0		T
G12	PT31B	0		C	PT31B	0		C
E12	PT31A	0		T	PT31A	0		T
VCCIO	VCCIO0	0			VCCIO0	0		
F12	NC	-			PT30B	0		C

LFE2M70E/SE and LFE2M100E/SE Logic Signal Connections: 1152 fpBGA

LFE2M70E/SE				LFE2M100E/SE				
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
D11	NC	-			PT30A	0		T
F11	NC	-			PT29B	0		C
E11	NC	-			PT29A	0		T
D7	ULC_SQ_VCCRX0	11			ULC_SQ_VCCRX0	11		
C9	ULC_SQ_HDINP0	11		T	ULC_SQ_HDINP0	11		T
B9	ULC_SQ_VCCIB0	11			ULC_SQ_VCCIB0	11		
C8	ULC_SQ_HDINN0	11		C	ULC_SQ_HDINN0	11		C
B8	ULC_SQ_VCCTX0	11			ULC_SQ_VCCTX0	11		
A9	ULC_SQ_HDOUTP0	11		T	ULC_SQ_HDOUTP0	11		T
D9	ULC_SQ_VCCOB0	11			ULC_SQ_VCCOB0	11		
A8	ULC_SQ_HDOUTN0	11		C	ULC_SQ_HDOUTN0	11		C
B7	ULC_SQ_VCCTX1	11			ULC_SQ_VCCTX1	11		
A7	ULC_SQ_HDOUTN1	11		C	ULC_SQ_HDOUTN1	11		C
E7	ULC_SQ_VCCOB1	11			ULC_SQ_VCCOB1	11		
A6	ULC_SQ_HDOUTP1	11		T	ULC_SQ_HDOUTP1	11		T
B6	ULC_SQ_VCCRX1	11			ULC_SQ_VCCRX1	11		
C7	ULC_SQ_HDINN1	11		C	ULC_SQ_HDINN1	11		C
D8	ULC_SQ_VCCIB1	11			ULC_SQ_VCCIB1	11		
C6	ULC_SQ_HDINP1	11		T	ULC_SQ_HDINP1	11		T
E6	ULC_SQ_VCCAUX33	11			ULC_SQ_VCCAUX33	11		
E5	ULC_SQ_REFCLKN	11		C	ULC_SQ_REFCLKN	11		C
D5	ULC_SQ_REFCLKP	11		T	ULC_SQ_REFCLKP	11		T
D6	ULC_SQ_VCCP	11			ULC_SQ_VCCP	11		
C5	ULC_SQ_HDINP2	11		T	ULC_SQ_HDINP2	11		T
D4	ULC_SQ_VCCIB2	11			ULC_SQ_VCCIB2	11		
C4	ULC_SQ_HDINN2	11		C	ULC_SQ_HDINN2	11		C
B5	ULC_SQ_VCCRX2	11			ULC_SQ_VCCRX2	11		
A5	ULC_SQ_HDOUTP2	11		T	ULC_SQ_HDOUTP2	11		T
D3	ULC_SQ_VCCOB2	11			ULC_SQ_VCCOB2	11		
A4	ULC_SQ_HDOUTN2	11		C	ULC_SQ_HDOUTN2	11		C
B4	ULC_SQ_VCCTX2	11			ULC_SQ_VCCTX2	11		
A3	ULC_SQ_HDOUTN3	11		C	ULC_SQ_HDOUTN3	11		C
C1	ULC_SQ_VCCOB3	11			ULC_SQ_VCCOB3	11		
A2	ULC_SQ_HDOUTP3	11		T	ULC_SQ_HDOUTP3	11		T
B3	ULC_SQ_VCCTX3	11			ULC_SQ_VCCTX3	11		
C3	ULC_SQ_HDINN3	11		C	ULC_SQ_HDINN3	11		C
B1	ULC_SQ_VCCIB3	11			ULC_SQ_VCCIB3	11		
C2	ULC_SQ_HDINP3	11		T	ULC_SQ_HDINP3	11		T
B2	ULC_SQ_VCCRX3	11			ULC_SQ_VCCRX3	11		
AA13	VCC	-			VCC	-		
AA14	VCC	-			VCC	-		
AA15	VCC	-			VCC	-		
AA16	VCC	-			VCC	-		
AA17	VCC	-			VCC	-		
AA18	VCC	-			VCC	-		
AA19	VCC	-			VCC	-		
AA20	VCC	-			VCC	-		
AA21	VCC	-			VCC	-		
AA22	VCC	-			VCC	-		
AB14	VCC	-			VCC	-		
AB15	VCC	-			VCC	-		

LFE2M70E/SE and LFE2M100E/SE Logic Signal Connections: 1152 fpBGA

LFE2M70E/SE				LFE2M100E/SE				
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
AB20	VCC	-			VCC	-		
AB21	VCC	-			VCC	-		
N14	VCC	-			VCC	-		
N15	VCC	-			VCC	-		
N20	VCC	-			VCC	-		
N21	VCC	-			VCC	-		
P13	VCC	-			VCC	-		
P14	VCC	-			VCC	-		
P15	VCC	-			VCC	-		
P16	VCC	-			VCC	-		
P17	VCC	-			VCC	-		
P18	VCC	-			VCC	-		
P19	VCC	-			VCC	-		
P20	VCC	-			VCC	-		
P21	VCC	-			VCC	-		
P22	VCC	-			VCC	-		
R13	VCC	-			VCC	-		
R14	VCC	-			VCC	-		
R21	VCC	-			VCC	-		
R22	VCC	-			VCC	-		
T14	VCC	-			VCC	-		
T21	VCC	-			VCC	-		
U14	VCC	-			VCC	-		
U21	VCC	-			VCC	-		
V14	VCC	-			VCC	-		
V21	VCC	-			VCC	-		
W14	VCC	-			VCC	-		
W21	VCC	-			VCC	-		
Y13	VCC	-			VCC	-		
Y14	VCC	-			VCC	-		
Y21	VCC	-			VCC	-		
Y22	VCC	-			VCC	-		
C12	VCCIO0	0			VCCIO0	0		
C16	VCCIO0	0			VCCIO0	0		
E14	VCCIO0	0			VCCIO0	0		
H12	VCCIO0	0			VCCIO0	0		
H16	VCCIO0	0			VCCIO0	0		
M14	VCCIO0	0			VCCIO0	0		
M15	VCCIO0	0			VCCIO0	0		
C19	VCCIO1	1			VCCIO1	1		
C23	VCCIO1	1			VCCIO1	1		
E21	VCCIO1	1			VCCIO1	1		
H19	VCCIO1	1			VCCIO1	1		
H23	VCCIO1	1			VCCIO1	1		
M20	VCCIO1	1			VCCIO1	1		
M21	VCCIO1	1			VCCIO1	1		
G32	VCCIO2	2			VCCIO2	2		
K28	VCCIO2	2			VCCIO2	2		
K32	VCCIO2	2			VCCIO2	2		
N27	VCCIO2	2			VCCIO2	2		
N32	VCCIO2	2			VCCIO2	2		

LFE2M70E/SE and LFE2M100E/SE Logic Signal Connections: 1152 fpBGA

LFE2M70E/SE				LFE2M100E/SE				
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
P23	VCCIO2	2			VCCIO2	2		
R23	VCCIO2	2			VCCIO2	2		
T27	VCCIO2	2			VCCIO2	2		
T32	VCCIO2	2			VCCIO2	2		
AA23	VCCIO3	3			VCCIO3	3		
AB27	VCCIO3	3			VCCIO3	3		
AB32	VCCIO3	3			VCCIO3	3		
AE28	VCCIO3	3			VCCIO3	3		
AE32	VCCIO3	3			VCCIO3	3		
AH32	VCCIO3	3			VCCIO3	3		
W27	VCCIO3	3			VCCIO3	3		
W32	VCCIO3	3			VCCIO3	3		
Y23	VCCIO3	3			VCCIO3	3		
AC20	VCCIO4	4			VCCIO4	4		
AC21	VCCIO4	4			VCCIO4	4		
AG19	VCCIO4	4			VCCIO4	4		
AG23	VCCIO4	4			VCCIO4	4		
AK21	VCCIO4	4			VCCIO4	4		
AM19	VCCIO4	4			VCCIO4	4		
AM23	VCCIO4	4			VCCIO4	4		
AC14	VCCIO5	5			VCCIO5	5		
AC15	VCCIO5	5			VCCIO5	5		
AG12	VCCIO5	5			VCCIO5	5		
AG16	VCCIO5	5			VCCIO5	5		
AK14	VCCIO5	5			VCCIO5	5		
AM12	VCCIO5	5			VCCIO5	5		
AM16	VCCIO5	5			VCCIO5	5		
AA12	VCCIO6	6			VCCIO6	6		
AB3	VCCIO6	6			VCCIO6	6		
AB8	VCCIO6	6			VCCIO6	6		
AE3	VCCIO6	6			VCCIO6	6		
AE7	VCCIO6	6			VCCIO6	6		
AH3	VCCIO6	6			VCCIO6	6		
W3	VCCIO6	6			VCCIO6	6		
W8	VCCIO6	6			VCCIO6	6		
Y12	VCCIO6	6			VCCIO6	6		
G3	VCCIO7	7			VCCIO7	7		
K3	VCCIO7	7			VCCIO7	7		
K7	VCCIO7	7			VCCIO7	7		
N3	VCCIO7	7			VCCIO7	7		
N8	VCCIO7	7			VCCIO7	7		
P12	VCCIO7	7			VCCIO7	7		
R12	VCCIO7	7			VCCIO7	7		
T3	VCCIO7	7			VCCIO7	7		
T8	VCCIO7	7			VCCIO7	7		
AD28	VCCIO8	8			VCCIO8	8		
AG32	VCCIO8	8			VCCIO8	8		
AB12	VCCAUX	-			VCCAUX	-		
AB13	VCCAUX	-			VCCAUX	-		
AB22	VCCAUX	-			VCCAUX	-		
AB23	VCCAUX	-			VCCAUX	-		

LFE2M70E/SE and LFE2M100E/SE Logic Signal Connections: 1152 fpBGA

LFE2M70E/SE				LFE2M100E/SE				
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
AC13	VCCAUX	-			VCCAUX	-		
AC22	VCCAUX	-			VCCAUX	-		
M13	VCCAUX	-			VCCAUX	-		
M22	VCCAUX	-			VCCAUX	-		
N12	VCCAUX	-			VCCAUX	-		
N13	VCCAUX	-			VCCAUX	-		
N22	VCCAUX	-			VCCAUX	-		
N23	VCCAUX	-			VCCAUX	-		
A1	GND	-			GND	-		
A10	GND	-			GND	-		
A13	GND	-			GND	-		
A22	GND	-			GND	-		
A25	GND	-			GND	-		
A34	GND	-			GND	-		
AB16	GND	-			GND	-		
AB17	GND	-			GND	-		
AB18	GND	-			GND	-		
AB19	GND	-			GND	-		
AB26	GND	-			GND	-		
AB31	GND	-			GND	-		
AB4	GND	-			GND	-		
AB9	GND	-			GND	-		
AC16	GND	-			GND	-		
AC17	GND	-			GND	-		
AC18	GND	-			GND	-		
AC19	GND	-			GND	-		
AD27	GND	-			GND	-		
AE27	GND	-			GND	-		
AE31	GND	-			GND	-		
AE4	GND	-			GND	-		
AE8	GND	-			GND	-		
AF12	GND	-			GND	-		
AF16	GND	-			GND	-		
AF19	GND	-			GND	-		
AF23	GND	-			GND	-		
AG31	GND	-			GND	-		
AH31	GND	-			GND	-		
AH4	GND	-			GND	-		
AJ14	GND	-			GND	-		
AJ21	GND	-			GND	-		
AK27	GND	-			GND	-		
AK8	GND	-			GND	-		
AL10	GND	-			GND	-		
AL16	GND	-			GND	-		
AL19	GND	-			GND	-		
AL2	GND	-			GND	-		
AL25	GND	-			GND	-		
AL33	GND	-			GND	-		
AP1	GND	-			GND	-		
AP10	GND	-			GND	-		
AP13	GND	-			GND	-		

LFE2M70E/SE and LFE2M100E/SE Logic Signal Connections: 1152 fpBGA

LFE2M70E/SE				LFE2M100E/SE				
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
AP22	GND	-			GND	-		
AP25	GND	-			GND	-		
AP34	GND	-			GND	-		
D10	GND	-			GND	-		
D16	GND	-			GND	-		
D19	GND	-			GND	-		
D2	GND	-			GND	-		
D25	GND	-			GND	-		
D33	GND	-			GND	-		
E27	GND	-			GND	-		
E8	GND	-			GND	-		
F14	GND	-			GND	-		
F21	GND	-			GND	-		
G31	GND	-			GND	-		
G4	GND	-			GND	-		
J12	GND	-			GND	-		
J16	GND	-			GND	-		
J19	GND	-			GND	-		
J23	GND	-			GND	-		
K27	GND	-			GND	-		
K31	GND	-			GND	-		
K4	GND	-			GND	-		
K8	GND	-			GND	-		
M16	GND	-			GND	-		
M17	GND	-			GND	-		
M18	GND	-			GND	-		
M19	GND	-			GND	-		
N16	GND	-			GND	-		
N17	GND	-			GND	-		
N18	GND	-			GND	-		
N19	GND	-			GND	-		
N26	GND	-			GND	-		
N31	GND	-			GND	-		
N4	GND	-			GND	-		
N9	GND	-			GND	-		
R16	GND	-			GND	-		
R17	GND	-			GND	-		
R18	GND	-			GND	-		
R19	GND	-			GND	-		
T12	GND	-			GND	-		
T13	GND	-			GND	-		
T15	GND	-			GND	-		
T16	GND	-			GND	-		
T17	GND	-			GND	-		
T18	GND	-			GND	-		
T19	GND	-			GND	-		
T20	GND	-			GND	-		
T22	GND	-			GND	-		
T23	GND	-			GND	-		
T26	GND	-			GND	-		
T31	GND	-			GND	-		

LFE2M70E/SE and LFE2M100E/SE Logic Signal Connections: 1152 fpBGA

LFE2M70E/SE				LFE2M100E/SE				
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
T4	GND	-			GND	-		
T9	GND	-			GND	-		
U12	GND	-			GND	-		
U13	GND	-			GND	-		
U15	GND	-			GND	-		
U16	GND	-			GND	-		
U17	GND	-			GND	-		
U18	GND	-			GND	-		
U19	GND	-			GND	-		
U20	GND	-			GND	-		
U22	GND	-			GND	-		
U23	GND	-			GND	-		
V12	GND	-			GND	-		
V13	GND	-			GND	-		
V15	GND	-			GND	-		
V16	GND	-			GND	-		
V17	GND	-			GND	-		
V18	GND	-			GND	-		
V19	GND	-			GND	-		
V20	GND	-			GND	-		
V22	GND	-			GND	-		
V23	GND	-			GND	-		
W12	GND	-			GND	-		
W13	GND	-			GND	-		
W15	GND	-			GND	-		
W16	GND	-			GND	-		
W17	GND	-			GND	-		
W18	GND	-			GND	-		
W19	GND	-			GND	-		
W20	GND	-			GND	-		
W22	GND	-			GND	-		
W23	GND	-			GND	-		
W26	GND	-			GND	-		
W31	GND	-			GND	-		
W4	GND	-			GND	-		
W9	GND	-			GND	-		
Y16	GND	-			GND	-		
Y17	GND	-			GND	-		
Y18	GND	-			GND	-		
Y19	GND	-			GND	-		
A11	NC	-			NC	-		
A12	NC	-			NC	-		
A23	NC	-			NC	-		
A24	NC	-			NC	-		
AA11	NC	-			NC	-		
AB11	NC	-			NC	-		
AC26	NC	-			NC	-		
AC30	NC	-			NC	-		
AD11	NC	-			NC	-		
AD12	NC	-			NC	-		
AD13	NC	-			NC	-		

LFE2M70E/SE and LFE2M100E/SE Logic Signal Connections: 1152 fpBGA

LFE2M70E/SE				LFE2M100E/SE				
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
AD14	NC	-			NC	-		
AD15	NC	-			NC	-		
AD19	NC	-			NC	-		
AD21	NC	-			NC	-		
AD22	NC	-			NC	-		
AD23	NC	-			NC	-		
AE10	NC	-			NC	-		
AE11	NC	-			NC	-		
AE12	NC	-			NC	-		
AE13	NC	-			NC	-		
AE19	NC	-			NC	-		
AE21	NC	-			NC	-		
AE22	NC	-			NC	-		
AE23	NC	-			NC	-		
AF11	NC	-			NC	-		
AF21	NC	-			NC	-		
AF22	NC	-			NC	-		
AF24	NC	-			NC	-		
AF8	NC	-			NC	-		
AF9	NC	-			NC	-		
AG10	NC	-			NC	-		
AG11	NC	-			NC	-		
AG24	NC	-			NC	-		
AG25	NC	-			NC	-		
AG26	NC	-			NC	-		
AG3	NC	-			NC	-		
AG7	NC	-			NC	-		
AG8	NC	-			NC	-		
AG9	NC	-			NC	-		
AH10	NC	-			NC	-		
AH11	NC	-			NC	-		
AH13	NC	-			NC	-		
AH24	NC	-			NC	-		
AH25	NC	-			NC	-		
AH26	NC	-			NC	-		
AH27	NC	-			NC	-		
AH5	NC	-			NC	-		
AH6	NC	-			NC	-		
AH7	NC	-			NC	-		
AH8	NC	-			NC	-		
AH9	NC	-			NC	-		
AJ10	NC	-			NC	-		
AJ11	NC	-			NC	-		
AJ13	NC	-			NC	-		
AJ24	NC	-			NC	-		
AJ25	NC	-			NC	-		
AJ26	NC	-			NC	-		
AJ27	NC	-			NC	-		
AJ3	NC	-			NC	-		
AJ4	NC	-			NC	-		
AJ5	NC	-			NC	-		

LFE2M70E/SE and LFE2M100E/SE Logic Signal Connections: 1152 fpBGA

LFE2M70E/SE				LFE2M100E/SE				
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
AJ6	NC	-			NC	-		
AJ7	NC	-			NC	-		
AJ8	NC	-			NC	-		
AJ9	NC	-			NC	-		
AK10	NC	-			NC	-		
AK11	NC	-			NC	-		
AK12	NC	-			NC	-		
AK24	NC	-			NC	-		
AK25	NC	-			NC	-		
AK26	NC	-			NC	-		
AK4	NC	-			NC	-		
AK9	NC	-			NC	-		
AL11	NC	-			NC	-		
AL12	NC	-			NC	-		
AL34	NC	-			NC	-		
AM10	NC	-			NC	-		
AM11	NC	-			NC	-		
AM13	NC	-			NC	-		
AM25	NC	-			NC	-		
AN10	NC	-			NC	-		
AN11	NC	-			NC	-		
AN12	NC	-			NC	-		
AN13	NC	-			NC	-		
AN24	NC	-			NC	-		
AN25	NC	-			NC	-		
AP11	NC	-			NC	-		
AP12	NC	-			NC	-		
AP24	NC	-			NC	-		
B10	NC	-			NC	-		
B11	NC	-			NC	-		
B12	NC	-			NC	-		
B13	NC	-			NC	-		
B22	NC	-			NC	-		
B23	NC	-			NC	-		
B24	NC	-			NC	-		
B25	NC	-			NC	-		
C10	NC	-			NC	-		
C11	NC	-			NC	-		
C13	NC	-			NC	-		
C22	NC	-			NC	-		
C24	NC	-			NC	-		
C25	NC	-			NC	-		
D1	NC	-			NC	-		
D15	NC	-			NC	-		
D24	NC	-			NC	-		
D34	NC	-			NC	-		
E10	NC	-			NC	-		
E24	NC	-			NC	-		
E25	NC	-			NC	-		
E26	NC	-			NC	-		
E3	NC	-			NC	-		

LFE2M70E/SE and LFE2M100E/SE Logic Signal Connections: 1152 fpBGA

LFE2M70E/SE				LFE2M100E/SE				
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
E31	NC	-			NC	-		
E32	NC	-			NC	-		
E33	NC	-			NC	-		
E34	NC	-			NC	-		
E4	NC	-			NC	-		
E9	NC	-			NC	-		
F10	NC	-			NC	-		
F25	NC	-			NC	-		
F26	NC	-			NC	-		
F27	NC	-			NC	-		
F28	NC	-			NC	-		
F29	NC	-			NC	-		
F30	NC	-			NC	-		
F31	NC	-			NC	-		
F32	NC	-			NC	-		
F33	NC	-			NC	-		
F34	NC	-			NC	-		
F5	NC	-			NC	-		
F6	NC	-			NC	-		
F7	NC	-			NC	-		
F8	NC	-			NC	-		
F9	NC	-			NC	-		
G10	NC	-			NC	-		
G11	NC	-			NC	-		
G24	NC	-			NC	-		
G25	NC	-			NC	-		
G26	NC	-			NC	-		
G27	NC	-			NC	-		
G28	NC	-			NC	-		
G29	NC	-			NC	-		
G30	NC	-			NC	-		
G33	NC	-			NC	-		
G34	NC	-			NC	-		
G7	NC	-			NC	-		
G8	NC	-			NC	-		
G9	NC	-			NC	-		
H10	NC	-			NC	-		
H11	NC	-			NC	-		
H24	NC	-			NC	-		
H25	NC	-			NC	-		
H26	NC	-			NC	-		
H27	NC	-			NC	-		
H28	NC	-			NC	-		
H29	NC	-			NC	-		
H8	NC	-			NC	-		
H9	NC	-			NC	-		
J10	NC	-			NC	-		
J11	NC	-			NC	-		
J24	NC	-			NC	-		
J25	NC	-			NC	-		
J26	NC	-			NC	-		

LFE2M70E/SE and LFE2M100E/SE Logic Signal Connections: 1152 fpBGA

LFE2M70E/SE				LFE2M100E/SE				
Ball Number	Ball/Pad Function	Bank	Dual Function	Differential	Ball/Pad Function	Bank	Dual Function	Differential
J9	NC	-			NC	-		
K10	NC	-			NC	-		
K11	NC	-			NC	-		
K12	NC	-			NC	-		
K13	NC	-			NC	-		
K23	NC	-			NC	-		
K24	NC	-			NC	-		
K25	NC	-			NC	-		
K26	NC	-			NC	-		
L11	NC	-			NC	-		
L12	NC	-			NC	-		
L13	NC	-			NC	-		
L14	NC	-			NC	-		
L21	NC	-			NC	-		
L22	NC	-			NC	-		
L23	NC	-			NC	-		
L24	NC	-			NC	-		
L25	NC	-			NC	-		
L26	NC	-			NC	-		
M11	NC	-			NC	-		
M24	NC	-			NC	-		
M25	NC	-			NC	-		
M6	NC	-			NC	-		
M8	NC	-			NC	-		
N10	NC	-			NC	-		
N11	NC	-			NC	-		
P10	NC	-			NC	-		
P25	NC	-			NC	-		
P26	NC	-			NC	-		
R9	NC	-			NC	-		
T11	NC	-			NC	-		
U11	NC	-			NC	-		
W11	NC	-			NC	-		
Y10	NC	-			NC	-		
Y11	NC	-			NC	-		
R15	VCCPLL	-			VCCPLL	-		
R20	VCCPLL	-			VCCPLL	-		
Y15	VCCPLL	-			VCCPLL	-		
Y20	VCCPLL	-			VCCPLL	-		

* Supports true LVDS. Other differential signals must be emulated with external resistors.

** These dedicated input pins can be used for GPLLs or GDLLs within the respective quadrant.

*** For density migration, board design must take into account that these sysCONFIG pins are dual function for the lower density devices (ECP2M20 and ECP2M35). They can be either sysCONFIG pins or general purpose I/Os. These pins are dedicated pins for the higher density devices (ECP2M50, ECP2M70, and ECP2M100).

****Due to packaging bond out option, this DQS does not have all the necessary DQ pins bonded out for a full 8-bit data width.

Note: VCCIO and GND pads are used to determine the average DC current drawn by I/Os between GND/VCCIO connections, or between the last GND/VCCIO in an I/O bank and the end of an I/O bank. The substrate pads listed in the Pin Table do not necessarily have a one to one connection with a package ball or pin.

LatticeECP2/M Family Data Sheet Ordering Information

November 2009 Data Sheet DS1006

LatticeECP2 Part Number Description

LFE2 – XX XE – X XXXXXX X

Device Family
ECP2 (LatticeECP2 FPGA)

Logic Capacity
6 = 6K LUTs
12 = 12K LUTs
20 = 20K LUTs
35 = 35K LUTs
50 = 50K LUTs
70 = 70K LUTs

Encryption
S = Security Series (Encryption Feature)
Blank = Standard Series (No Encryption)

Supply Voltage
E = 1.2V

Grade
C = Commercial
I = Industrial

Package
T144 = 144-pin TQFP
Q208 = 208-pin PQFP
F256 = 256-ball fpBGA
F484 = 484-ball fpBGA
F672 = 672-ball fpBGA
F900 = 900-ball fpBGA
TN144 = 144-pin Lead-Free TQFP
QN208 = 208-pin Lead-Free PQFP
FN256 = 256-ball Lead-Free fpBGA
FN484 = 484-ball Lead-Free fpBGA
FN672 = 672-ball Lead-Free fpBGA
FN900 = 900-ball Lead-Free fpBGA

Speed
5 = Slowest
6
7 = Fastest

Ordering Information

Note:þLatticeECP2 devices are dual marked. For example, the commercial speed grade LFE2-50E-7F672C is also marked with industrial grade -6I (LFE2-50E-6F672I). The commercial grade is one speed grade faster than the associated dual mark industrial grade. The slowest commercial speed grade does not have industrial markings. The markings appear as follows:

LatticeECP2 Standard Series Devices, Conventional Packaging

Commercial

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-6E-5T144C	90	1.2V	-5	TQFP	144	COM	6
LFE2-6E-6T144C	90	1.2V	-6	TQFP	144	COM	6
LFE2-6E-7T144C	90	1.2V	-7	TQFP	144	COM	6
LFE2-6E-5F256C	190	1.2V	-5	fpBGA	256	COM	6
LFE2-6E-6F256C	190	1.2V	-6	fpBGA	256	COM	6
LFE2-6E-7F256C	190	1.2V	-7	fpBGA	256	COM	6

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-12E-5T144C	93	1.2V	-5	TQFP	144	COM	12
LFE2-12E-6T144C	93	1.2V	-6	TQFP	144	COM	12
LFE2-12E-7T144C	93	1.2V	-7	TQFP	144	COM	12
LFE2-12E-5Q208C	131	1.2V	-5	PQFP	208	COM	12
LFE2-12E-6Q208C	131	1.2V	-6	PQFP	208	COM	12
LFE2-12E-7Q208C	131	1.2V	-7	PQFP	208	COM	12
LFE2-12E-5F256C	193	1.2V	-5	fpBGA	256	COM	12
LFE2-12E-6F256C	193	1.2V	-6	fpBGA	256	COM	12
LFE2-12E-7F256C	193	1.2V	-7	fpBGA	256	COM	12
LFE2-12E-5F484C	297	1.2V	-5	fpBGA	484	COM	12
LFE2-12E-6F484C	297	1.2V	-6	fpBGA	484	COM	12
LFE2-12E-7F484C	297	1.2V	-7	fpBGA	484	COM	12

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-20E-5Q208C	131	1.2V	-5	PQFP	208	COM	20
LFE2-20E-6Q208C	131	1.2V	-6	PQFP	208	COM	20
LFE2-20E-7Q208C	131	1.2V	-7	PQFP	208	COM	20
LFE2-20E-5F256C	193	1.2V	-5	fpBGA	256	COM	20
LFE2-20E-6F256C	193	1.2V	-6	fpBGA	256	COM	20
LFE2-20E-7F256C	193	1.2V	-7	fpBGA	256	COM	20
LFE2-20E-5F484C	331	1.2V	-5	fpBGA	484	COM	20
LFE2-20E-6F484C	331	1.2V	-6	fpBGA	484	COM	20
LFE2-20E-7F484C	331	1.2V	-7	fpBGA	484	COM	20
LFE2-20E-5F672C	402	1.2V	-5	fpBGA	672	COM	20
LFE2-20E-6F672C	402	1.2V	-6	fpBGA	672	COM	20
LFE2-20E-7F672C	402	1.2V	-7	fpBGA	672	COM	20

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-35E-5F484C	331	1.2V	-5	fpBGA	484	COM	35
LFE2-35E-6F484C	331	1.2V	-6	fpBGA	484	COM	35
LFE2-35E-7F484C	331	1.2V	-7	fpBGA	484	COM	35
LFE2-35E-5F672C	450	1.2V	-5	fpBGA	672	COM	35
LFE2-35E-6F672C	450	1.2V	-6	fpBGA	672	COM	35
LFE2-35E-7F672C	450	1.2V	-7	fpBGA	672	COM	35

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-50E-5F484C	339	1.2V	-5	fpBGA	484	COM	50
LFE2-50E-6F484C	339	1.2V	-6	fpBGA	484	COM	50
LFE2-50E-7F484C	339	1.2V	-7	fpBGA	484	COM	50
LFE2-50E-5F672C	500	1.2V	-5	fpBGA	672	COM	50
LFE2-50E-6F672C	500	1.2V	-6	fpBGA	672	COM	50
LFE2-50E-7F672C	500	1.2V	-7	fpBGA	672	COM	50

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-70E-5F672C	500	1.2V	-5	fpBGA	672	COM	70
LFE2-70E-6F672C	500	1.2V	-6	fpBGA	672	COM	70
LFE2-70E-7F672C	500	1.2V	-7	fpBGA	672	COM	70
LFE2-70E-5F900C	583	1.2V	-5	fpBGA	900	COM	70
LFE2-70E-6F900C	583	1.2V	-6	fpBGA	900	COM	70
LFE2-70E-7F900C	583	1.2V	-7	fpBGA	900	COM	70

Industrial

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-6E-5T144I	90	1.2V	-5	TQFP	144	IND	6
LFE2-6E-6T144I	90	1.2V	-6	TQFP	144	IND	6
LFE2-6E-5F256I	190	1.2V	-5	fpBGA	256	IND	6
LFE2-6E-6F256I	190	1.2V	-6	fpBGA	256	IND	6

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-12E-5T144I	93	1.2V	-5	TQFP	144	IND	12
LFE2-12E-6T144I	93	1.2V	-6	TQFP	144	IND	12
LFE2-12E-5Q208I	131	1.2V	-5	PQFP	208	IND	12
LFE2-12E-6Q208I	131	1.2V	-6	PQFP	208	IND	12
LFE2-12E-5F256I	193	1.2V	-5	fpBGA	256	IND	12
LFE2-12E-6F256I	193	1.2V	-6	fpBGA	256	IND	12
LFE2-12E-5F484I	297	1.2V	-5	fpBGA	484	IND	12
LFE2-12E-6F484I	297	1.2V	-6	fpBGA	484	IND	12

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-20E-5Q208I	131	1.2V	-5	PQFP	208	IND	20
LFE2-20E-6Q208I	131	1.2V	-6	PQFP	208	IND	20
LFE2-20E-5F256I	193	1.2V	-5	fpBGA	256	IND	20
LFE2-20E-6F256I	193	1.2V	-6	fpBGA	256	IND	20
LFE2-20E-5F484I	331	1.2V	-5	fpBGA	484	IND	20
LFE2-20E-6F484I	331	1.2V	-6	fpBGA	484	IND	20
LFE2-20E-5F672I	402	1.2V	-5	fpBGA	672	IND	20
LFE2-20E-6F672I	402	1.2V	-6	fpBGA	672	IND	20

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-35E-5F484I	331	1.2V	-5	fpBGA	484	IND	35
LFE2-35E-6F484I	331	1.2V	-6	fpBGA	484	IND	35
LFE2-35E-5F672I	450	1.2V	-5	fpBGA	672	IND	35
LFE2-35E-6F672I	450	1.2V	-6	fpBGA	672	IND	35

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-50E-5F484I	339	1.2V	-5	fpBGA	484	IND	50
LFE2-50E-6F484I	339	1.2V	-6	fpBGA	484	IND	50
LFE2-50E-5F672I	500	1.2V	-5	fpBGA	672	IND	50
LFE2-50E-6F672I	500	1.2V	-6	fpBGA	672	IND	50

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-70E-5F672I	500	1.2V	-5	fpBGA	672	IND	70
LFE2-70E-6F672I	500	1.2V	-6	fpBGA	672	IND	70
LFE2-70E-5F900I	583	1.2V	-5	fpBGA	900	IND	70
LFE2-70E-6F900I	583	1.2V	-6	fpBGA	900	IND	70

LatticeECP2 Standard Series Devices, Lead-Free Packaging

Commercial

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-6E-5TN144C	90	1.2V	-5	Lead-Free TQFP	144	COM	6
LFE2-6E-6TN144C	90	1.2V	-6	Lead-Free TQFP	144	COM	6
LFE2-6E-7TN144C	90	1.2V	-7	Lead-Free TQFP	144	COM	6
LFE2-6E-5FN256C	190	1.2V	-5	Lead-Free fpBGA	256	COM	6
LFE2-6E-6FN256C	190	1.2V	-6	Lead-Free fpBGA	256	COM	6
LFE2-6E-7FN256C	190	1.2V	-7	Lead-Free fpBGA	256	COM	6

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-12E-5TN144C	93	1.2V	-5	Lead-Free TQFP	144	COM	12
LFE2-12E-6TN144C	93	1.2V	-6	Lead-Free TQFP	144	COM	12
LFE2-12E-7TN144C	93	1.2V	-7	Lead-Free TQFP	144	COM	12
LFE2-12E-5QN208C	131	1.2V	-5	Lead-Free PQFP	208	COM	12
LFE2-12E-6QN208C	131	1.2V	-6	Lead-Free PQFP	208	COM	12
LFE2-12E-7QN208C	131	1.2V	-7	Lead-Free PQFP	208	COM	12
LFE2-12E-5FN256C	193	1.2V	-5	Lead-Free fpBGA	256	COM	12
LFE2-12E-6FN256C	193	1.2V	-6	Lead-Free fpBGA	256	COM	12
LFE2-12E-7FN256C	193	1.2V	-7	Lead-Free fpBGA	256	COM	12
LFE2-12E-5FN484C	297	1.2V	-5	Lead-Free fpBGA	484	COM	12
LFE2-12E-6FN484C	297	1.2V	-6	Lead-Free fpBGA	484	COM	12
LFE2-12E-7FN484C	297	1.2V	-7	Lead-Free fpBGA	484	COM	12

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-20E-5QN208C	131	1.2V	-5	Lead-Free PQFP	208	COM	20
LFE2-20E-6QN208C	131	1.2V	-6	Lead-Free PQFP	208	COM	20
LFE2-20E-7QN208C	131	1.2V	-7	Lead-Free PQFP	208	COM	20
LFE2-20E-5FN256C	193	1.2V	-5	Lead-Free fpBGA	256	COM	20
LFE2-20E-6FN256C	193	1.2V	-6	Lead-Free fpBGA	256	COM	20
LFE2-20E-7FN256C	193	1.2V	-7	Lead-Free fpBGA	256	COM	20
LFE2-20E-5FN484C	331	1.2V	-5	Lead-Free fpBGA	484	COM	20
LFE2-20E-6FN484C	331	1.2V	-6	Lead-Free fpBGA	484	COM	20
LFE2-20E-7FN484C	331	1.2V	-7	Lead-Free fpBGA	484	COM	20
LFE2-20E-5FN672C	402	1.2V	-5	Lead-Free fpBGA	672	COM	20
LFE2-20E-6FN672C	402	1.2V	-6	Lead-Free fpBGA	672	COM	20
LFE2-20E-7FN672C	402	1.2V	-7	Lead-Free fpBGA	672	COM	20

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-35E-5FN484C	331	1.2V	-5	Lead-Free fpBGA	484	COM	35
LFE2-35E-6FN484C	331	1.2V	-6	Lead-Free fpBGA	484	COM	35
LFE2-35E-7FN484C	331	1.2V	-7	Lead-Free fpBGA	484	COM	35
LFE2-35E-5FN672C	450	1.2V	-5	Lead-Free fpBGA	672	COM	35
LFE2-35E-6FN672C	450	1.2V	-6	Lead-Free fpBGA	672	COM	35
LFE2-35E-7FN672C	450	1.2V	-7	Lead-Free fpBGA	672	COM	35

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-50E-5FN484C	339	1.2V	-5	Lead-Free fpBGA	484	COM	50
LFE2-50E-6FN484C	339	1.2V	-6	Lead-Free fpBGA	484	COM	50
LFE2-50E-7FN484C	339	1.2V	-7	Lead-Free fpBGA	484	COM	50
LFE2-50E-5FN672C	500	1.2V	-5	Lead-Free fpBGA	672	COM	50
LFE2-50E-6FN672C	500	1.2V	-6	Lead-Free fpBGA	672	COM	50
LFE2-50E-7FN672C	500	1.2V	-7	Lead-Free fpBGA	672	COM	50

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-70E-5FN672C	500	1.2V	-5	Lead-Free fpBGA	672	COM	70
LFE2-70E-6FN672C	500	1.2V	-6	Lead-Free fpBGA	672	COM	70
LFE2-70E-7FN672C	500	1.2V	-7	Lead-Free fpBGA	672	COM	70
LFE2-70E-5FN900C	583	1.2V	-5	Lead-Free fpBGA	900	COM	70
LFE2-70E-6FN900C	583	1.2V	-6	Lead-Free fpBGA	900	COM	70
LFE2-70E-7FN900C	583	1.2V	-7	Lead-Free fpBGA	900	COM	70

Industrial

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-6E-5TN144I	90	1.2V	-5	Lead-Free TQFP	144	IND	6
LFE2-6E-6TN144I	90	1.2V	-6	Lead-Free TQFP	144	IND	6
LFE2-6E-5FN256I	190	1.2V	-5	Lead-Free fpBGA	256	IND	6
LFE2-6E-6FN256I	190	1.2V	-6	Lead-Free fpBGA	256	IND	6

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-12E-5TN144I	93	1.2V	-5	Lead-Free TQFP	144	IND	12
LFE2-12E-6TN144I	93	1.2V	-6	Lead-Free TQFP	144	IND	12
LFE2-12E-5QN208I	131	1.2V	-5	Lead-Free PQFP	208	IND	12
LFE2-12E-6QN208I	131	1.2V	-6	Lead-Free PQFP	208	IND	12
LFE2-12E-5FN256I	193	1.2V	-5	Lead-Free fpBGA	256	IND	12
LFE2-12E-6FN256I	193	1.2V	-6	Lead-Free fpBGA	256	IND	12
LFE2-12E-5FN484I	297	1.2V	-5	Lead-Free fpBGA	484	IND	12
LFE2-12E-6FN484I	297	1.2V	-6	Lead-Free fpBGA	484	IND	12

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-20E-5QN208I	131	1.2V	-5	Lead-Free PQFP	208	IND	20
LFE2-20E-6QN208I	131	1.2V	-6	Lead-Free PQFP	208	IND	20
LFE2-20E-5FN256I	193	1.2V	-5	Lead-Free fpBGA	256	IND	20
LFE2-20E-6FN256I	193	1.2V	-6	Lead-Free fpBGA	256	IND	20
LFE2-20E-5FN484I	331	1.2V	-5	Lead-Free fpBGA	484	IND	20
LFE2-20E-6FN484I	331	1.2V	-6	Lead-Free fpBGA	484	IND	20
LFE2-20E-5FN672I	402	1.2V	-5	Lead-Free fpBGA	672	IND	20
LFE2-20E-6FN672I	402	1.2V	-6	Lead-Free fpBGA	672	IND	20

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-35E-5FN484I	331	1.2V	-5	Lead-Free fpBGA	484	IND	35
LFE2-35E-6FN484I	331	1.2V	-6	Lead-Free fpBGA	484	IND	35
LFE2-35E-5FN672I	450	1.2V	-5	Lead-Free fpBGA	672	IND	35
LFE2-35E-6FN672I	450	1.2V	-6	Lead-Free fpBGA	672	IND	35

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-50E-5FN484I	339	1.2V	-5	Lead-Free fpBGA	484	IND	50
LFE2-50E-6FN484I	339	1.2V	-6	Lead-Free fpBGA	484	IND	50
LFE2-50E-5FN672I	500	1.2V	-5	Lead-Free fpBGA	672	IND	50
LFE2-50E-6FN672I	500	1.2V	-6	Lead-Free fpBGA	672	IND	50

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-70E-5FN672I	500	1.2V	-5	Lead-Free fpBGA	672	IND	70
LFE2-70E-6FN672I	500	1.2V	-6	Lead-Free fpBGA	672	IND	70
LFE2-70E-5FN900I	583	1.2V	-5	Lead-Free fpBGA	900	IND	70
LFE2-70E-6FN900I	583	1.2V	-6	Lead-Free fpBGA	900	IND	70

LatticeECP2 S-Series Devices, Conventional Packaging

Commercial

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-6SE-5T144C	90	1.2V	-5	TQFP	144	Com	6
LFE2-6SE-6T144C	90	1.2V	-6	TQFP	144	Com	6
LFE2-6SE-7T144C	90	1.2V	-7	TQFP	144	Com	6
LFE2-6SE-5F256C	190	1.2V	-5	fpBGA	256	Com	6
LFE2-6SE-6F256C	190	1.2V	-6	fpBGA	256	Com	6
LFE2-6SE-7F256C	190	1.2V	-7	fpBGA	256	Com	6

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-12SE-5T144C	93	1.2V	-5	TQFP	144	Com	12
LFE2-12SE-6T144C	93	1.2V	-6	TQFP	144	Com	12
LFE2-12SE-7T144C	93	1.2V	-7	TQFP	144	Com	12
LFE2-12SE-5Q208C	131	1.2V	-5	PQFP	208	Com	12
LFE2-12SE-6Q208C	131	1.2V	-6	PQFP	208	Com	12
LFE2-12SE-7Q208C	131	1.2V	-7	PQFP	208	Com	12
LFE2-12SE-5F256C	193	1.2V	-5	fpBGA	256	Com	12
LFE2-12SE-6F256C	193	1.2V	-6	fpBGA	256	Com	12
LFE2-12SE-7F256C	193	1.2V	-7	fpBGA	256	Com	12
LFE2-12SE-5F484C	297	1.2V	-5	fpBGA	484	Com	12
LFE2-12SE-6F484C	297	1.2V	-6	fpBGA	484	Com	12
LFE2-12SE-7F484C	297	1.2V	-7	fpBGA	484	Com	12

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-20SE-5Q208C	131	1.2V	-5	PQFP	208	Com	20
LFE2-20SE-6Q208C	131	1.2V	-6	PQFP	208	Com	20
LFE2-20SE-7Q208C	131	1.2V	-7	PQFP	208	Com	20
LFE2-20SE-5F256C	193	1.2V	-5	fpBGA	256	Com	20
LFE2-20SE-6F256C	193	1.2V	-6	fpBGA	256	Com	20
LFE2-20SE-7F256C	193	1.2V	-7	fpBGA	256	Com	20
LFE2-20SE-5F484C	331	1.2V	-5	fpBGA	484	Com	20
LFE2-20SE-6F484C	331	1.2V	-6	fpBGA	484	Com	20
LFE2-20SE-7F484C	331	1.2V	-7	fpBGA	484	Com	20
LFE2-20SE-5F672C	402	1.2V	-5	fpBGA	672	Com	20
LFE2-20SE-6F672C	402	1.2V	-6	fpBGA	672	Com	20
LFE2-20SE-7F672C	402	1.2V	-7	fpBGA	672	Com	20

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-35SE-5F484C	331	1.2V	-5	fpBGA	484	Com	35
LFE2-35SE-6F484C	331	1.2V	-6	fpBGA	484	Com	35
LFE2-35SE-7F484C	331	1.2V	-7	fpBGA	484	Com	35
LFE2-35SE-5F672C	450	1.2V	-5	fpBGA	672	Com	35
LFE2-35SE-6F672C	450	1.2V	-6	fpBGA	672	Com	35
LFE2-35SE-7F672C	450	1.2V	-7	fpBGA	672	Com	35

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-50SE-5F484C	339	1.2V	-5	fpBGA	484	Com	50
LFE2-50SE-6F484C	339	1.2V	-6	fpBGA	484	Com	50
LFE2-50SE-7F484C	339	1.2V	-7	fpBGA	484	Com	50
LFE2-50SE-5F672C	500	1.2V	-5	fpBGA	672	Com	50
LFE2-50SE-6F672C	500	1.2V	-6	fpBGA	672	Com	50
LFE2-50SE-7F672C	500	1.2V	-7	fpBGA	672	Com	50

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-70SE-5F672C	500	1.2V	-5	fpBGA	672	Com	70
LFE2-70SE-6F672C	500	1.2V	-6	fpBGA	672	Com	70
LFE2-70SE-7F672C	500	1.2V	-7	fpBGA	672	Com	70
LFE2-70SE-5F900C	583	1.2V	-5	fpBGA	900	Com	70
LFE2-70SE-6F900C	583	1.2V	-6	fpBGA	900	Com	70
LFE2-70SE-7F900C	583	1.2V	-7	fpBGA	900	Com	70

Industrial

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-6SE-5T144I	90	1.2V	-5	TQFP	144	Ind	6
LFE2-6SE-6T144I	90	1.2V	-6	TQFP	144	Ind	6
LFE2-6SE-5F256I	190	1.2V	-5	fpBGA	256	Ind	6
LFE2-6SE-6F256I	190	1.2V	-6	fpBGA	256	Ind	6

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-12SE-5T144I	93	1.2V	-5	TQFP	144	Ind	12
LFE2-12SE-6T144I	93	1.2V	-6	TQFP	144	Ind	12
LFE2-12SE-5Q208I	131	1.2V	-5	PQFP	208	Ind	12
LFE2-12SE-6Q208I	131	1.2V	-6	PQFP	208	Ind	12
LFE2-12SE-5F256I	193	1.2V	-5	fpBGA	256	Ind	12
LFE2-12SE-6F256I	193	1.2V	-6	fpBGA	256	Ind	12
LFE2-12SE-5F484I	297	1.2V	-5	fpBGA	484	Ind	12
LFE2-12SE-6F484I	297	1.2V	-6	fpBGA	484	Ind	12

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-20SE-5Q208I	131	1.2V	-5	PQFP	208	Ind	20
LFE2-20SE-6Q208I	131	1.2V	-6	PQFP	208	Ind	20
LFE2-20SE-5F256I	193	1.2V	-5	fpBGA	256	Ind	20
LFE2-20SE-6F256I	193	1.2V	-6	fpBGA	256	Ind	20
LFE2-20SE-5F484I	331	1.2V	-5	fpBGA	484	Ind	20
LFE2-20SE-6F484I	331	1.2V	-6	fpBGA	484	Ind	20
LFE2-20SE-5F672I	402	1.2V	-5	fpBGA	672	Ind	20
LFE2-20SE-6F672I	402	1.2V	-6	fpBGA	672	Ind	20

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-35SE-5F484I	331	1.2V	-5	fpBGA	484	Ind	35
LFE2-35SE-6F484I	331	1.2V	-6	fpBGA	484	Ind	35
LFE2-35SE-5F672I	450	1.2V	-5	fpBGA	672	Ind	35
LFE2-35SE-6F672I	450	1.2V	-6	fpBGA	672	Ind	35

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-50SE-5F484I	339	1.2V	-5	fpBGA	484	Ind	50
LFE2-50SE-6F484I	339	1.2V	-6	fpBGA	484	Ind	50
LFE2-50SE-5F672I	500	1.2V	-5	fpBGA	672	Ind	50
LFE2-50SE-6F672I	500	1.2V	-6	fpBGA	672	Ind	50

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-70SE-5F672I	500	1.2V	-5	fpBGA	672	Ind	70
LFE2-70SE-6F672I	500	1.2V	-6	fpBGA	672	Ind	70
LFE2-70SE-5F900I	583	1.2V	-5	fpBGA	900	Ind	70
LFE2-70SE-6F900I	583	1.2V	-6	fpBGA	900	Ind	70

LatticeECP2 S-Series Devices, Lead-Free Packaging

Commercial

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-6SE-5TN144C	90	1.2V	-5	Lead-Free TQFP	144	Com	6
LFE2-6SE-6TN144C	90	1.2V	-6	Lead-Free TQFP	144	Com	6
LFE2-6SE-7TN144C	90	1.2V	-7	Lead-Free TQFP	144	Com	6
LFE2-6SE-5FN256C	190	1.2V	-5	Lead-Free fpBGA	256	Com	6
LFE2-6SE-6FN256C	190	1.2V	-6	Lead-Free fpBGA	256	Com	6
LFE2-6SE-7FN256C	190	1.2V	-7	Lead-Free fpBGA	256	Com	6

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-12SE-5TN144C	93	1.2V	-5	Lead-Free TQFP	144	Com	12
LFE2-12SE-6TN144C	93	1.2V	-6	Lead-Free TQFP	144	Com	12
LFE2-12SE-7TN144C	93	1.2V	-7	Lead-Free TQFP	144	Com	12
LFE2-12SE-5QN208C	131	1.2V	-5	Lead-Free PQFP	208	Com	12
LFE2-12SE-6QN208C	131	1.2V	-6	Lead-Free PQFP	208	Com	12
LFE2-12SE-7QN208C	131	1.2V	-7	Lead-Free PQFP	208	Com	12
LFE2-12SE-5FN256C	193	1.2V	-5	Lead-Free fpBGA	256	Com	12
LFE2-12SE-6FN256C	193	1.2V	-6	Lead-Free fpBGA	256	Com	12
LFE2-12SE-7FN256C	193	1.2V	-7	Lead-Free fpBGA	256	Com	12
LFE2-12SE-5FN484C	297	1.2V	-5	Lead-Free fpBGA	484	Com	12
LFE2-12SE-6FN484C	297	1.2V	-6	Lead-Free fpBGA	484	Com	12
LFE2-12SE-7FN484C	297	1.2V	-7	Lead-Free fpBGA	484	Com	12

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-20SE-5QN208C	131	1.2V	-5	Lead-Free PQFP	208	Com	20
LFE2-20SE-6QN208C	131	1.2V	-6	Lead-Free PQFP	208	Com	20
LFE2-20SE-7QN208C	131	1.2V	-7	Lead-Free PQFP	208	Com	20
LFE2-20SE-5FN256C	193	1.2V	-5	Lead-Free fpBGA	256	Com	20
LFE2-20SE-6FN256C	193	1.2V	-6	Lead-Free fpBGA	256	Com	20
LFE2-20SE-7FN256C	193	1.2V	-7	Lead-Free fpBGA	256	Com	20
LFE2-20SE-5FN484C	331	1.2V	-5	Lead-Free fpBGA	484	Com	20
LFE2-20SE-6FN484C	331	1.2V	-6	Lead-Free fpBGA	484	Com	20
LFE2-20SE-7FN484C	331	1.2V	-7	Lead-Free fpBGA	484	Com	20
LFE2-20SE-5FN672C	402	1.2V	-5	Lead-Free fpBGA	672	Com	20
LFE2-20SE-6FN672C	402	1.2V	-6	Lead-Free fpBGA	672	Com	20
LFE2-20SE-7FN672C	402	1.2V	-7	Lead-Free fpBGA	672	Com	20

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-35SE-5FN484C	331	1.2V	-5	Lead-Free fpBGA	484	Com	35
LFE2-35SE-6FN484C	331	1.2V	-6	Lead-Free fpBGA	484	Com	35
LFE2-35SE-7FN484C	331	1.2V	-7	Lead-Free fpBGA	484	Com	35
LFE2-35SE-5FN672C	450	1.2V	-5	Lead-Free fpBGA	672	Com	35
LFE2-35SE-6FN672C	450	1.2V	-6	Lead-Free fpBGA	672	Com	35
LFE2-35SE-7FN672C	450	1.2V	-7	Lead-Free fpBGA	672	Com	35

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-50SE-5FN484C	339	1.2V	-5	Lead-Free fpBGA	484	Com	50
LFE2-50SE-6FN484C	339	1.2V	-6	Lead-Free fpBGA	484	Com	50
LFE2-50SE-7FN484C	339	1.2V	-7	Lead-Free fpBGA	484	Com	50
LFE2-50SE-5FN672C	500	1.2V	-5	Lead-Free fpBGA	672	Com	50
LFE2-50SE-6FN672C	500	1.2V	-6	Lead-Free fpBGA	672	Com	50
LFE2-50SE-7FN672C	500	1.2V	-7	Lead-Free fpBGA	672	Com	50

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-70SE-5FN672C	500	1.2V	-5	Lead-Free fpBGA	672	Com	70
LFE2-70SE-6FN672C	500	1.2V	-6	Lead-Free fpBGA	672	Com	70
LFE2-70SE-7FN672C	500	1.2V	-7	Lead-Free fpBGA	672	Com	70
LFE2-70SE-5FN900C	583	1.2V	-5	Lead-Free fpBGA	900	Com	70
LFE2-70SE-6FN900C	583	1.2V	-6	Lead-Free fpBGA	900	Com	70
LFE2-70SE-7FN900C	583	1.2V	-7	Lead-Free fpBGA	900	Com	70

Industrial

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-6SE-5TN144I	90	1.2V	-5	Lead-Free TQFP	144	Ind	6
LFE2-6SE-6TN144I	90	1.2V	-6	Lead-Free TQFP	144	Ind	6
LFE2-6SE-5FN256I	190	1.2V	-5	Lead-Free fpBGA	256	Ind	6
LFE2-6SE-6FN256I	190	1.2V	-6	Lead-Free fpBGA	256	Ind	6

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-12SE-5TN144I	93	1.2V	-5	Lead-Free TQFP	144	Ind	12
LFE2-12SE-6TN144I	93	1.2V	-6	Lead-Free TQFP	144	Ind	12
LFE2-12SE-5QN208I	131	1.2V	-5	Lead-Free PQFP	208	Ind	12
LFE2-12SE-6QN208I	131	1.2V	-6	Lead-Free PQFP	208	Ind	12
LFE2-12SE-5FN256I	193	1.2V	-5	Lead-Free fpBGA	256	Ind	12
LFE2-12SE-6FN256I	193	1.2V	-6	Lead-Free fpBGA	256	Ind	12
LFE2-12SE-5FN484I	297	1.2V	-5	Lead-Free fpBGA	484	Ind	12
LFE2-12SE-6FN484I	297	1.2V	-6	Lead-Free fpBGA	484	Ind	12

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-20SE-5QN208I	131	1.2V	-5	Lead-Free PQFP	208	Ind	20
LFE2-20SE-6QN208I	131	1.2V	-6	Lead-Free PQFP	208	Ind	20
LFE2-20SE-5FN256I	193	1.2V	-5	Lead-Free fpBGA	256	Ind	20
LFE2-20SE-6FN256I	193	1.2V	-6	Lead-Free fpBGA	256	Ind	20
LFE2-20SE-5FN484I	331	1.2V	-5	Lead-Free fpBGA	484	Ind	20
LFE2-20SE-6FN484I	331	1.2V	-6	Lead-Free fpBGA	484	Ind	20
LFE2-20SE-5FN672I	402	1.2V	-5	Lead-Free fpBGA	672	Ind	20
LFE2-20SE-6FN672I	402	1.2V	-6	Lead-Free fpBGA	672	Ind	20

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-35SE-5FN484I	331	1.2V	-5	Lead-Free fpBGA	484	Ind	35
LFE2-35SE-6FN484I	331	1.2V	-6	Lead-Free fpBGA	484	Ind	35
LFE2-35SE-5FN672I	450	1.2V	-5	Lead-Free fpBGA	672	Ind	35
LFE2-35SE-6FN672I	450	1.2V	-6	Lead-Free fpBGA	672	Ind	35

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-50SE-5FN484I	339	1.2V	-5	Lead-Free fpBGA	484	Ind	50
LFE2-50SE-6FN484I	339	1.2V	-6	Lead-Free fpBGA	484	Ind	50
LFE2-50SE-5FN672I	500	1.2V	-5	Lead-Free fpBGA	672	Ind	50
LFE2-50SE-6FN672I	500	1.2V	-6	Lead-Free fpBGA	672	Ind	50

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2-70SE-5FN672I	500	1.2V	-5	Lead-Free fpBGA	672	Ind	70
LFE2-70SE-6FN672I	500	1.2V	-6	Lead-Free fpBGA	672	Ind	70
LFE2-70SE-5FN900I	583	1.2V	-5	Lead-Free fpBGA	900	Ind	70
LFE2-70SE-6FN900I	583	1.2V	-6	Lead-Free fpBGA	900	Ind	70

LatticeECP2M Part Number Description

LFE2M XXX XE – X XXXXXX X

Device Family
ECP2M (LatticeECP2 FPGA + SERDES)

Logic Capacity
20 = 20K LUTs
35 = 35K LUTs
50 = 50K LUTs
70 = 70K LUTs
100 = 100K LUTs

Encryption
S = Security Series (Encryption Feature)
Blank = Standard Series (No Encryption)

Supply Voltage
E = 1.2V

Grade
C = Commercial
I = Industrial

Package
F256 = 256-ball fpBGA
F484 = 484-ball fpBGA
F672 = 672-ball fpBGA
F900 = 900-ball fpBGA
F1152 = 1152-ball fpBGA

FN256 = 256-ball Lead-free fpBGA
FN484 = 484-ball Lead-free fpBGA
FN672 = 672-ball Lead-free fpBGA
FN900 = 900-ball Lead-free fpBGA
FN1152 = 1152-ball Lead-free fpBGA

Speed
5 = Slowest
6
7 = Fastest

Ordering Information

Note:þLatticeECP2M devices are dual marked. For example, the commercial speed grade LFE2M50E-7F672C is also marked with industrial grade -6I (LFE2M50E-6F672I). The commercial grade is one speed grade faster than the associated dual mark industrial grade. The slowest commercial grade does not have industrial markings. The markings appear as follows:

LatticeECP2M Standard Series Devices, Conventional Packaging

Commercial

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M20E-5F484C	304	1.2V	-5	fpBGA	484	COM	20
LFE2M20E-6F484C	304	1.2V	-6	fpBGA	484	COM	20
LFE2M20E-7F484C	304	1.2V	-7	fpBGA	484	COM	20
LFE2M20E-5F256C	140	1.2V	-5	fpBGA	256	COM	20
LFE2M20E-6F256C	140	1.2V	-6	fpBGA	256	COM	20
LFE2M20E-7F256C	140	1.2V	-7	fpBGA	256	COM	20

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M35E-5F672C	410	1.2V	-5	fpBGA	672	COM	35
LFE2M35E-6F672C	410	1.2V	-6	fpBGA	672	COM	35
LFE2M35E-7F672C	410	1.2V	-7	fpBGA	672	COM	35
LFE2M35E-5F484C	303	1.2V	-5	fpBGA	484	COM	35
LFE2M35E-6F484C	303	1.2V	-6	fpBGA	484	COM	35
LFE2M35E-7F484C	303	1.2V	-7	fpBGA	484	COM	35
LFE2M35E-5F256C	140	1.2V	-5	fpBGA	256	COM	35
LFE2M35E-6F256C	140	1.2V	-6	fpBGA	256	COM	35
LFE2M35E-7F256C	140	1.2V	-7	fpBGA	256	COM	35

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M50E-5F900C	410	1.2V	-5	fpBGA	900	COM	50
LFE2M50E-6F900C	410	1.2V	-6	fpBGA	900	COM	50
LFE2M50E-7F900C	410	1.2V	-7	fpBGA	900	COM	50
LFE2M50E-5F672C	372	1.2V	-5	fpBGA	672	COM	50
LFE2M50E-6F672C	372	1.2V	-6	fpBGA	672	COM	50
LFE2M50E-7F672C	372	1.2V	-7	fpBGA	672	COM	50
LFE2M50E-5F484C	270	1.2V	-5	fpBGA	484	COM	50
LFE2M50E-6F484C	270	1.2V	-6	fpBGA	484	COM	50
LFE2M50E-7F484C	270	1.2V	-7	fpBGA	484	COM	50

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M70E-5F1152C	436	1.2V	-5	fpBGA	1152	COM	70
LFE2M70E-6F1152C	436	1.2V	-6	fpBGA	1152	COM	70
LFE2M70E-7F1152C	436	1.2V	-7	fpBGA	1152	COM	70
LFE2M70E-5F900C	416	1.2V	-5	fpBGA	900	COM	70
LFE2M70E-6F900C	416	1.2V	-6	fpBGA	900	COM	70
LFE2M70E-7F900C	416	1.2V	-7	fpBGA	900	COM	70

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M100E-5F1152C	520	1.2V	-5	fpBGA	1152	COM	100
LFE2M100E-6F1152C	520	1.2V	-6	fpBGA	1152	COM	100
LFE2M100E-7F1152C	520	1.2V	-7	fpBGA	1152	COM	100
LFE2M100E-5F900C	416	1.2V	-5	fpBGA	900	COM	100
LFE2M100E-6F900C	416	1.2V	-6	fpBGA	900	COM	100
LFE2M100E-7F900C	416	1.2V	-7	fpBGA	900	COM	100

Industrial

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M20E-5F484I	304	1.2V	-5	fpBGA	484	IND	20
LFE2M20E-6F484I	304	1.2V	-6	fpBGA	484	IND	20
LFE2M20E-5F256I	140	1.2V	-5	fpBGA	256	IND	20
LFE2M20E-6F256I	140	1.2V	-6	fpBGA	256	IND	20

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M35E-5F672I	410	1.2V	-5	fpBGA	672	IND	35
LFE2M35E-6F672I	410	1.2V	-6	fpBGA	672	IND	35
LFE2M35E-5F484I	303	1.2V	-5	fpBGA	484	IND	35
LFE2M35E-6F484I	303	1.2V	-6	fpBGA	484	IND	35
LFE2M35E-5F256I	140	1.2V	-5	fpBGA	256	IND	35
LFE2M35E-6F256I	140	1.2V	-6	fpBGA	256	IND	35

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M50E-5F900I	410	1.2V	-5	fpBGA	900	IND	50
LFE2M50E-6F900I	410	1.2V	-6	fpBGA	900	IND	50
LFE2M50E-5F672I	372	1.2V	-5	fpBGA	672	IND	50
LFE2M50E-6F672I	372	1.2V	-6	fpBGA	672	IND	50
LFE2M50E-5F484I	270	1.2V	-5	fpBGA	484	IND	50
LFE2M50E-6F484I	270	1.2V	-6	fpBGA	484	IND	50

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M70E-5F1152I	436	1.2V	-5	fpBGA	1152	IND	70
LFE2M70E-6F1152I	436	1.2V	-6	fpBGA	1152	IND	70
LFE2M70E-5F900I	416	1.2V	-5	fpBGA	900	IND	70
LFE2M70E-6F900I	416	1.2V	-6	fpBGA	900	IND	70

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M100E-5F1152I	520	1.2V	-5	fpBGA	1152	IND	100
LFE2M100E-6F1152I	520	1.2V	-6	fpBGA	1152	IND	100
LFE2M100E-5F900I	416	1.2V	-5	fpBGA	900	IND	100
LFE2M100E-6F900I	416	1.2V	-6	fpBGA	900	IND	100

LatticeECP2M Standard Series Devices, Lead-Free Packaging

Commercial

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M20E-5FN484C	304	1.2V	-5	Lead-Free fpBGA	484	COM	20
LFE2M20E-6FN484C	304	1.2V	-6	Lead-Free fpBGA	484	COM	20
LFE2M20E-7FN484C	304	1.2V	-7	Lead-Free fpBGA	484	COM	20
LFE2M20E-5FN256C	140	1.2V	-5	Lead-Free fpBGA	256	COM	20
LFE2M20E-6FN256C	140	1.2V	-6	Lead-Free fpBGA	256	COM	20
LFE2M20E-7FN256C	140	1.2V	-7	Lead-Free fpBGA	256	COM	20

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M35E-5FN672C	410	1.2V	-5	Lead-Free fpBGA	672	COM	35
LFE2M35E-6FN672C	410	1.2V	-6	Lead-Free fpBGA	672	COM	35
LFE2M35E-7FN672C	410	1.2V	-7	Lead-Free fpBGA	672	COM	35
LFE2M35E-5FN484C	303	1.2V	-5	Lead-Free fpBGA	484	COM	35
LFE2M35E-6FN484C	303	1.2V	-6	Lead-Free fpBGA	484	COM	35
LFE2M35E-7FN484C	303	1.2V	-7	Lead-Free fpBGA	484	COM	35
LFE2M35E-5FN256C	140	1.2V	-5	Lead-Free fpBGA	256	COM	35
LFE2M35E-6FN256C	140	1.2V	-6	Lead-Free fpBGA	256	COM	35
LFE2M35E-7FN256C	140	1.2V	-7	Lead-Free fpBGA	256	COM	35

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M50E-5FN900C	410	1.2V	-5	Lead-Free fpBGA	900	COM	50
LFE2M50E-6FN900C	410	1.2V	-6	Lead-Free fpBGA	900	COM	50
LFE2M50E-7FN900C	410	1.2V	-7	Lead-Free fpBGA	900	COM	50
LFE2M50E-5FN672C	372	1.2V	-5	Lead-Free fpBGA	672	COM	50
LFE2M50E-6FN672C	372	1.2V	-6	Lead-Free fpBGA	672	COM	50
LFE2M50E-7FN672C	372	1.2V	-7	Lead-Free fpBGA	672	COM	50
LFE2M50E-5FN484C	270	1.2V	-5	Lead-Free fpBGA	484	COM	50
LFE2M50E-6FN484C	270	1.2V	-6	Lead-Free fpBGA	484	COM	50
LFE2M50E-7FN484C	270	1.2V	-7	Lead-Free fpBGA	484	COM	50

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M70E-5FN1152C	436	1.2V	-5	Lead-Free fpBGA	1152	COM	70
LFE2M70E-6FN1152C	436	1.2V	-6	Lead-Free fpBGA	1152	COM	70
LFE2M70E-7FN1152C	436	1.2V	-7	Lead-Free fpBGA	1152	COM	70
LFE2M70E-5FN900C	416	1.2V	-5	Lead-Free fpBGA	900	COM	70
LFE2M70E-6FN900C	416	1.2V	-6	Lead-Free fpBGA	900	COM	70
LFE2M70E-7FN900C	416	1.2V	-7	Lead-Free fpBGA	900	COM	70

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M100E-5FN1152C	520	1.2V	-5	Lead-Free fpBGA	1152	COM	100
LFE2M100E-6FN1152C	520	1.2V	-6	Lead-Free fpBGA	1152	COM	100
LFE2M100E-7FN1152C	520	1.2V	-7	Lead-Free fpBGA	1152	COM	100
LFE2M100E-5FN900C	416	1.2V	-5	Lead-Free fpBGA	900	COM	100
LFE2M100E-6FN900C	416	1.2V	-6	Lead-Free fpBGA	900	COM	100
LFE2M100E-7FN900C	416	1.2V	-7	Lead-Free fpBGA	900	COM	100

Industrial

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M20E-5FN484I	304	1.2V	-5	Lead-Free fpBGA	484	IND	20
LFE2M20E-6FN484I	304	1.2V	-6	Lead-Free fpBGA	484	IND	20
LFE2M20E-5FN256I	140	1.2V	-5	Lead-Free fpBGA	256	IND	20
LFE2M20E-6FN256I	140	1.2V	-6	Lead-Free fpBGA	256	IND	20

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M35E-5FN672I	410	1.2V	-5	Lead-Free fpBGA	672	IND	35
LFE2M35E-6FN672I	410	1.2V	-6	Lead-Free fpBGA	672	IND	35
LFE2M35E-5FN484I	303	1.2V	-5	Lead-Free fpBGA	484	IND	35
LFE2M35E-6FN484I	303	1.2V	-6	Lead-Free fpBGA	484	IND	35
LFE2M35E-5FN256I	140	1.2V	-5	Lead-Free fpBGA	256	IND	35
LFE2M35E-6FN256I	140	1.2V	-6	Lead-Free fpBGA	256	IND	35

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M50E-5FN900I	410	1.2V	-5	Lead-Free fpBGA	900	Ind	50
LFE2M50E-6FN900I	410	1.2V	-6	Lead-Free fpBGA	900	Ind	50
LFE2M50E-5FN672I	372	1.2V	-5	Lead-Free fpBGA	672	Ind	50
LFE2M50E-6FN672I	372	1.2V	-6	Lead-Free fpBGA	672	Ind	50
LFE2M50E-5FN484I	270	1.2V	-5	Lead-Free fpBGA	484	Ind	50
LFE2M50E-6FN484I	270	1.2V	-6	Lead-Free fpBGA	484	Ind	50

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M70E-5FN1152I	436	1.2V	-5	Lead-Free fpBGA	1152	Ind	70
LFE2M70E-6FN1152I	436	1.2V	-6	Lead-Free fpBGA	1152	Ind	70
LFE2M70E-5FN900I	416	1.2V	-5	Lead-Free fpBGA	900	Ind	70
LFE2M70E-6FN900I	416	1.2V	-6	Lead-Free fpBGA	900	Ind	70

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M100E-5FN1152I	520	1.2V	-5	Lead-Free fpBGA	1152	Ind	100
LFE2M100E-6FN1152I	520	1.2V	-6	Lead-Free fpBGA	1152	Ind	100
LFE2M100E-5FN900I	416	1.2V	-5	Lead-Free fpBGA	900	Ind	100
LFE2M100E-6FN900I	416	1.2V	-6	Lead-Free fpBGA	900	Ind	100

LatticeECP2M S-Series Devices, Conventional Packaging

Commercial

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M20SE-5F484C	304	1.2V	-5	fpBGA	484	Com	20
LFE2M20SE-6F484C	304	1.2V	-6	fpBGA	484	Com	20
LFE2M20SE-7F484C	304	1.2V	-7	fpBGA	484	Com	20
LFE2M20SE-5F256C	140	1.2V	-5	fpBGA	256	Com	20
LFE2M20SE-6F256C	140	1.2V	-6	fpBGA	256	Com	20
LFE2M20SE-7F256C	140	1.2V	-7	fpBGA	256	Com	20

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M35SE-5F672C	410	1.2V	-5	fpBGA	672	Com	35
LFE2M35SE-6F672C	410	1.2V	-6	fpBGA	672	Com	35
LFE2M35SE-7F672C	410	1.2V	-7	fpBGA	672	Com	35
LFE2M35SE-5F484C	303	1.2V	-5	fpBGA	484	Com	35
LFE2M35SE-6F484C	303	1.2V	-6	fpBGA	484	Com	35
LFE2M35SE-7F484C	303	1.2V	-7	fpBGA	484	Com	35
LFE2M35SE-5F256C	140	1.2V	-5	fpBGA	256	Com	35
LFE2M35SE-6F256C	140	1.2V	-6	fpBGA	256	Com	35
LFE2M35SE-7F256C	140	1.2V	-7	fpBGA	256	Com	35

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M50SE-5F900C	410	1.2V	-5	fpBGA	900	Com	50
LFE2M50SE-6F900C	410	1.2V	-6	fpBGA	900	Com	50
LFE2M50SE-7F900C	410	1.2V	-7	fpBGA	900	Com	50
LFE2M50SE-5F672C	372	1.2V	-5	fpBGA	672	Com	50
LFE2M50SE-6F672C	372	1.2V	-6	fpBGA	672	Com	50
LFE2M50SE-7F672C	372	1.2V	-7	fpBGA	672	Com	50
LFE2M50SE-5F484C	270	1.2V	-5	fpBGA	484	Com	50
LFE2M50SE-6F484C	270	1.2V	-6	fpBGA	484	Com	50
LFE2M50SE-7F484C	270	1.2V	-7	fpBGA	484	Com	50

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M70SE-5F1152C	436	1.2V	-5	fpBGA	1152	Com	70
LFE2M70SE-6F1152C	436	1.2V	-6	fpBGA	1152	Com	70
LFE2M70SE-7F1152C	436	1.2V	-7	fpBGA	1152	Com	70
LFE2M70SE-5F900C	416	1.2V	-5	fpBGA	900	Com	70
LFE2M70SE-6F900C	416	1.2V	-6	fpBGA	900	Com	70
LFE2M70SE-7F900C	416	1.2V	-7	fpBGA	900	Com	70

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M100SE-5F1152C	520	1.2V	-5	fpBGA	1152	Com	100
LFE2M100SE-6F1152C	520	1.2V	-6	fpBGA	1152	Com	100
LFE2M100SE-7F1152C	520	1.2V	-7	fpBGA	1152	Com	100
LFE2M100SE-5F900C	416	1.2V	-5	fpBGA	900	Com	100
LFE2M100SE-6F900C	416	1.2V	-6	fpBGA	900	Com	100
LFE2M100SE-7F900C	416	1.2V	-7	fpBGA	900	Com	100

Industrial

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M20SE-5F484I	304	1.2V	-5	fpBGA	484	Ind	20
LFE2M20SE-6F484I	304	1.2V	-6	fpBGA	484	Ind	20
LFE2M20SE-5F256I	140	1.2V	-5	fpBGA	256	Ind	20
LFE2M20SE-6F256I	140	1.2V	-6	fpBGA	256	Ind	20

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M35SE-5F672I	410	1.2V	-5	fpBGA	672	Ind	35
LFE2M35SE-6F672I	410	1.2V	-6	fpBGA	672	Ind	35
LFE2M35SE-5F484I	303	1.2V	-5	fpBGA	484	Ind	35
LFE2M35SE-6F484I	303	1.2V	-6	fpBGA	484	Ind	35
LFE2M35SE-5F256I	140	1.2V	-5	fpBGA	256	Ind	35
LFE2M35SE-6F256I	140	1.2V	-6	fpBGA	256	Ind	35

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M50SE-5F900I	410	1.2V	-5	fpBGA	900	Ind	50
LFE2M50SE-6F900I	410	1.2V	-6	fpBGA	900	Ind	50
LFE2M50SE-5F672I	372	1.2V	-5	fpBGA	672	Ind	50
LFE2M50SE-6F672I	372	1.2V	-6	fpBGA	672	Ind	50
LFE2M50SE-5F484I	270	1.2V	-5	fpBGA	484	Ind	50
LFE2M50SE-6F484I	270	1.2V	-6	fpBGA	484	Ind	50

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M70SE-5F1152I	436	1.2V	-5	fpBGA	1152	Ind	70
LFE2M70SE-6F1152I	436	1.2V	-6	fpBGA	1152	Ind	70
LFE2M70SE-5F900I	416	1.2V	-5	fpBGA	900	Ind	70
LFE2M70SE-6F900I	416	1.2V	-6	fpBGA	900	Ind	70

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M100SE-5F1152I	520	1.2V	-5	fpBGA	1152	Ind	100
LFE2M100SE-6F1152I	520	1.2V	-6	fpBGA	1152	Ind	100
LFE2M100SE-5F900I	416	1.2V	-5	fpBGA	900	Ind	100
LFE2M100SE-6F900I	416	1.2V	-6	fpBGA	900	Ind	100

LatticeECP2M S-Series Devices, Lead-Free Packaging

Commercial

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M20SE-5FN484C	304	1.2V	-5	Lead-Free fpBGA	484	Com	20
LFE2M20SE-6FN484C	304	1.2V	-6	Lead-Free fpBGA	484	Com	20
LFE2M20SE-7FN484C	304	1.2V	-7	Lead-Free fpBGA	484	Com	20
LFE2M20SE-5FN256C	140	1.2V	-5	Lead-Free fpBGA	256	Com	20
LFE2M20SE-6FN256C	140	1.2V	-6	Lead-Free fpBGA	256	Com	20
LFE2M20SE-7FN256C	140	1.2V	-7	Lead-Free fpBGA	256	Com	20

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M35SE-5FN672C	410	1.2V	-5	Lead-Free fpBGA	672	Com	35
LFE2M35SE-6FN672C	410	1.2V	-6	Lead-Free fpBGA	672	Com	35
LFE2M35SE-7FN672C	410	1.2V	-7	Lead-Free fpBGA	672	Com	35
LFE2M35SE-5FN484C	303	1.2V	-5	Lead-Free fpBGA	484	Com	35
LFE2M35SE-6FN484C	303	1.2V	-6	Lead-Free fpBGA	484	Com	35
LFE2M35SE-7FN484C	303	1.2V	-7	Lead-Free fpBGA	484	Com	35
LFE2M35SE-5FN256C	140	1.2V	-5	Lead-Free fpBGA	256	Com	35
LFE2M35SE-6FN256C	140	1.2V	-6	Lead-Free fpBGA	256	Com	35
LFE2M35SE-7FN256C	140	1.2V	-7	Lead-Free fpBGA	256	Com	35

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M50SE-5FN900C	410	1.2V	-5	Lead-Free fpBGA	900	Com	50
LFE2M50SE-6FN900C	410	1.2V	-6	Lead-Free fpBGA	900	Com	50
LFE2M50SE-7FN900C	410	1.2V	-7	Lead-Free fpBGA	900	Com	50
LFE2M50SE-5FN672C	372	1.2V	-5	Lead-Free fpBGA	672	Com	50
LFE2M50SE-6FN672C	372	1.2V	-6	Lead-Free fpBGA	672	Com	50
LFE2M50SE-7FN672C	372	1.2V	-7	Lead-Free fpBGA	672	Com	50
LFE2M50SE-5FN484C	270	1.2V	-5	Lead-Free fpBGA	484	Com	50
LFE2M50SE-6FN484C	270	1.2V	-6	Lead-Free fpBGA	484	Com	50
LFE2M50SE-7FN484C	270	1.2V	-7	Lead-Free fpBGA	484	Com	50

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M70SE-5FN1152C	436	1.2V	-5	Lead-Free fpBGA	1152	Com	70
LFE2M70SE-6FN1152C	436	1.2V	-6	Lead-Free fpBGA	1152	Com	70
LFE2M70SE-7FN1152C	436	1.2V	-7	Lead-Free fpBGA	1152	Com	70
LFE2M70SE-5FN900C	416	1.2V	-5	Lead-Free fpBGA	900	Com	70
LFE2M70SE-6FN900C	416	416	-6	Lead-Free fpBGA	900	Com	70
LFE2M70SE-7FN900C	416	416	-7	Lead-Free fpBGA	900	Com	70

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M100SE-5FN1152C	520	1.2V	-5	Lead-Free fpBGA	1152	Com	100
LFE2M100SE-6FN1152C	520	1.2V	-6	Lead-Free fpBGA	1152	Com	100
LFE2M100SE-7FN1152C	520	1.2V	-7	Lead-Free fpBGA	1152	Com	100
LFE2M100SE-5FN900C	416	1.2V	-5	Lead-Free fpBGA	900	Com	100
LFE2M100SE-6FN900C	416	1.2V	-6	Lead-Free fpBGA	900	Com	100
LFE2M100SE-7FN900C	416	1.2V	-7	Lead-Free fpBGA	900	Com	100

Industrial

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M20SE-5FN484I	304	1.2V	-5	Lead-Free fpBGA	484	Ind	20
LFE2M20SE-6FN484I	304	1.2V	-6	Lead-Free fpBGA	484	Ind	20
LFE2M20SE-5FN256I	140	1.2V	-5	Lead-Free fpBGA	256	Ind	20
LFE2M20SE-6FN256I	140	1.2V	-6	Lead-Free fpBGA	256	Ind	20

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M35SE-5FN672I	410	1.2V	-5	Lead-Free fpBGA	672	Ind	35
LFE2M35SE-6FN672I	410	1.2V	-6	Lead-Free fpBGA	672	Ind	35
LFE2M35SE-5FN484I	303	1.2V	-5	Lead-Free fpBGA	484	Ind	35
LFE2M35SE-6FN484I	303	1.2V	-6	Lead-Free fpBGA	484	Ind	35
LFE2M35SE-5FN256I	140	1.2V	-5	Lead-Free fpBGA	256	Ind	35
LFE2M35SE-6FN256I	140	1.2V	-6	Lead-Free fpBGA	256	Ind	35

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M50SE-5FN900I	410	1.2V	-5	Lead-Free fpBGA	900	Ind	50
LFE2M50SE-6FN900I	410	1.2V	-6	Lead-Free fpBGA	900	Ind	50
LFE2M50SE-5FN672I	372	1.2V	-5	Lead-Free fpBGA	672	Ind	50
LFE2M50SE-6FN672I	372	1.2V	-6	Lead-Free fpBGA	672	Ind	50
LFE2M50SE-5FN484I	270	1.2V	-5	Lead-Free fpBGA	484	Ind	50
LFE2M50SE-6FN484I	270	1.2V	-6	Lead-Free fpBGA	484	Ind	50

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M70SE-5FN1152I	436	1.2V	-5	Lead-Free fpBGA	1152	Ind	70
LFE2M70SE-6FN1152I	436	1.2V	-6	Lead-Free fpBGA	1152	Ind	70
LFE2M70SE-5FN900I	416	1.2V	-5	Lead-Free fpBGA	900	Ind	70
LFE2M70SE-6FN900I	416	1.2V	-6	Lead-Free fpBGA	900	Ind	70

Part Number	I/Os	Voltage	Grade	Package	Pins	Temp.	LUTs (K)
LFE2M100SE-5FN1152I	520	1.2V	-5	Lead-Free fpBGA	1152	Ind	100
LFE2M100SE-6FN1152I	520	1.2V	-6	Lead-Free fpBGA	1152	Ind	100
LFE2M100SE-5FN900I	416	1.2V	-5	Lead-Free fpBGA	900	Ind	100
LFE2M100SE-6FN900I	416	1.2V	-6	Lead-Free fpBGA	900	Ind	100

LatticeECP2/M Family Data Sheet Supplemental Information

For Further Information

A variety of technical notes for the LatticeECP2/M family are available on the Lattice web site at www.latticesemi.com.

- TN1124, LatticeECP2M SERDES/PCS Usage Guide
- TN1102, LatticeECP2/M sysIO Usage Guide
- TN1103, LatticeECP2/M sysCLOCK PLL Design and Usage Guide
- TN1104, LatticeECP2/M Memory Usage Guide
- TN1105, LatticeECP2/M High-Speed I/O Interface
- TN1106, Power Estimation and Management for LatticeECP2/M Devices
- TN1107, LatticeECP2/M sysDSP Usage Guide
- TN1108, LatticeECP2/M sysCONFIG Usage Guide
- TN1109, LatticeECP2/M Configuration Encryption Usage Guide
- TN1113, LatticeECP2/M Soft Error Detection (SED) Usage Guide
- TN1162, LatticeECP2/M Hardware Checklist

For further information about interface standards refer to the following web sites:

- JEDEC Standards (LVTTL, LVCMOS, SSTL, HSTL): www.jedec.org
- PCI: www.pcisig.com

LatticeECP2/M Family Data Sheet Revision History

January 2012 **Data Sheet DS1006**

Date	Version	Section	Change Summary
February 2006	01.0	—	Initial release.
August 2006	01.1	Introduction	
			Updated Table 1-1 “LatticeECP2 Family Selection Guide”.
		Architecture	Updated Figure 2-2 “PFU Diagram”.
			Updated Figure 2-13 “Secondary Clock Regions ECP2-50”.
			Updated Figure 2-25 “PIC Diagram”.
			Updated Figure 2-26 “Input Register Block for Left, Right and Bottom Edges”.
			Updated Figure 2-28 “Output Register Block for Left, Right and Bottom Edges”.
			Updated Figure 2-30 “DQS Input Routing for Left and Right Edges”.
			Updated Figure 2-32 “Edge Clock, DLL Calibration and DQS Local Bus Distribution”.
			Table 2-15 Selectable Master Clock (CCLK) Frequencies - Removed frequencies 15, 20, 21, 22, 23, 30, 34, 41, 45, 51, 55, 60.
			Replaced “CLKINDEL” with “CLKO”.
			Updated SED section.
			Qualified device migration capability when using DQS banks for DDR interfaces.
		DC and Switching Characteristics	Added VCCPLL to the Recommended Operating Conditions table.
			Removed note 5 from “Hot Specifications” section.
			Added notes 7 and 8 to “Initialization Supply” Current table.
			Change note 6 - “...down to 95MHz” to “...down to 95MHz for DDR and 133MHz for DDR2” .
			New “Typical Building Block Function Performance” numbers.
			New External Switching Characteristics numbers.
			New Internal Switching Characteristics numbers.
			New Family Timing Adders numbers.
			Updated Timings for GPLLs, SPLLs and DLLs.
			Added sysCONFIG waveforms.
			Remove HSTL15D_II from sysIO Recommended Operating Conditions table.
			Updated Supply and Initialization Currents for ECP2-50.
		Pinout Information	Added VCCPLL to the Signal Descriptions table.
			Updated Logic Signal Connections tables to include 484-fpBGA for the ECP2-50.
			Added Logic Signal Connections tables for ECP2-12 devices.
			Updated Pin Information Summary table to include ECP2-12.
			Updated Power Supply and NC Connections table to include ECP2-12.
			Added note 2 to DDR Strobe (DQS) Pin table.
			Added Information on: PCI, DDR & SPI4.2 Capabilities of the device-Package combination.

Date	Version	Section	Change Summary
August 2006 (cont.)	01.1 (cont.)	Pinout Information (cont.)	Added Information on: Available Device Resources per Packaged Device table.
		Ordering Information	Updated ordering part number table to include ECP2-12.
			Updated topside mark drawing.
September 2006	02.0	Multiple	Added information regarding LatticeECP2M support throughout.
September 2006	02.1	DC and Switching Characteristics	Added Receiver Total Jitter Tolerance Specification table.
			Removed power-up requirements for proper configuration footnote in Recommended Operating Conditions table.
December 2006	02.2	Introduction	LatticeECP2M Selection Guide table has been updated.
		Architecture	Figure 2-16. Per Region Secondary Clock Selection has been updated.
			Figure 2-39. Simplified Channel Block Diagram for SERDES and PCS has been updated.
		DC and Switching	Footnotes have been added to Recommended Operating Conditions.
			DC Electrical Characteristics table has been updated.
			Supply Current (Standby) tables have been updated.
			Initialization Supply Current table have been updated.
			Updated timing numbers to include LFE2-12E (rev A 0.08).
		Pinout Information	Updated to include the entire ECP2 device information as well as 256-fpBGA and 484-fpBGA pin information for the ECP2M35E.
		Ordering Information	Updated to include the entire ECP2 and ECP2M device ordering information.
February 2007	02.3	Architecture	Updated EBR Asynchronous Reset section.
March 2007	02.4	DC and Switching Characteristics	Power-sequencing footnotes have been added to the Recommended Operating Conditions. DDR2 performance has been updated to 266MHz.
March 2007	02.5	Introduction	Added “Security Series” to the LatticeECP2 and LatticeECP2M families.
		Architecture	Enhanced Configuration Option section updated.
		DC and Switching	Recommended Operating Conditions table - footnote 4 updated.
		Ordering Information	“Security Series” ordering part numbers added.
April 2007	02.6	Introduction	LatticeECP2M family table has been updated for user I/O counts.
		Ordering Information	LatticeECP2M family ordering part number section has been updated to add 1152-fpBGA package for the ECP2M70 and ECP2M100.
July 2007	02.7	Architecture	Updated text in Ripple Mode section.
		DC and Switching	ECP2/M Supply Current information has been updated. Typical Building Block Function Performance, External Switching Characteristics, Internal Switching Characteristics, Family Timing Adders, sysCLOCK GPLL Timing, sysCLOCK SPLL Timing, DLL Timing and sysCONFIG Port Timing Specifications have been updated (timing rev. A 0.10). SERDES timing information has been updated. PCI Express timing information has been updated.
		Pinout Information	Added LatticeECP2M20 pinout information.
August 2007	02.8	Introduction	1156-fpBGA package option has been removed from the LatticeECP2M family.
		Architecture	Table 2-16. Selectable Master Clock (CCLK) Frequencies During Configuration table has been updated.
		DC and Switching	Supply Current (Standby) table has been updated.
			DSP Function timing has been updated.

Date	Version	Section	Change Summary
August 2007 (cont.)	02.8 (cont.)	DC and Switching (cont.)	sysCLOCK GPLL timing has been updated.
		Pinout Information	Added ECP2M50 (484/672/900-fpBGA), ECP2M70 (900-fpBGA) and ECP2M100 (900-fpBGA) pinout information.
		Ordering Information	1156-fpBGA package option has been removed from the LatticeECP2M family.
September 2007	02.9	Pinout Information	Added Thermal Management text section.
February 2008	03.0	Architecture	Added LVCMOS33D description.
		DC and Switching	LatticeECP2M Supply Current has been updated.
			Typical Building Block Function Performance, External Switching Characteristics, Internal Switching Characteristics, Family Timing Adders, sysCLOCK GPLL Timing, sysCLOCK SPLL Timing, DLL Timing and sysCONFIG Port Timing Specifications have been updated (timing rev. A 0.11).
			Figure 3-9. Read/Write Mode (Normal) and Figure 3-10. Read/Write Mode with Input and Output Registers have been updated.
			Table 3-8. Channel output Jitter (Max) has been updated.
		Pinout Information	Signal description has been updated.
			Added 1152-fpBGA pinouts for the ECP2M70 and ECP2M100.
April 2008	03.1	Pinout Information	Available DDR Interfaces per I/O Bank for the LFE2M35 (484/672-fpBGA) have been updated.
June 2008	03.2	Introduction	Family Selection Guide table - Updated number of EBR SRAM Blocks for the ECP2-70 device.
		Architecture	Removed Read-Before-Write sysMEM EBR mode.
			Clarification of the operation of the secondary clock regions.
		DC and Switching Characteristics	Removed Read-Before-Write sysMEM EBR mode.
August 2008	03.3	Architecture	Clarification of the operation of the secondary clock regions.
		Pinout Information	Added information for [LOC]DQ[num] to Signal Descriptions table.
January 2009	03.4	DC and Switching Characteristics	Updated typical and max. jitter numbers in Channel Output Jitter table for x10 mode.
			Added Channel Output Jitter table for x20 mode.
November 2009	03.5	DC and Switching Characteristics	Updated SPI/SPIm Configuration Waveforms diagram.
			Updated footnotes in LatticeECP2 Initialization Supply Current table.
			Updated footnotes in LatticeECP2M Initialization Supply Current table.
			Updated footnotes in SERDES High Speed Data Receiver (LatticeECP2M Family Only) table.
			Updated max. value for t_{DINIT} parameter in LatticeECP2/M sysCONFIG Port Timing Specifications table.
			Updated Serial Output Timing and Levels table.
			Updated Figure 3-5 MLVDS
			Updated Table 3-7 Serial Output Timing and Levels
			Updated Table 3-15 Power Down/Power Up Specification
		Pinout Information	Signal Descriptions table - corrected references to ULM, URM, LRM (changed to LUM, RUM and RLM), added footnote 5.

Date	Version	Section	Change Summary
November 2009 (cont.)	03.5 (cont.)	Pinout Information (cont.)	LatticeECP2M Pin Information Summary, LFE2M50, LFE2M70 and LFE2M100 table - corrected values for LFE2M50, 672 fpBGA in Available DDR-Interfaces per I/O Bank.
			Minor corrections in LFE2M20E/SE and LFE2M35E/SE Logic Signal Connections: 484 fpBGA table.
			Minor corrections in LFE2M50E/SE and LFE2M70E/SE Logic Signal Connections: 900 fpBGA table.
			Minor corrections in LFE2M100E/SE Logic Signal Connections: 900 fpBGA table.
			Updated LFE2-6E/SE and LFE2-12E/SE Logical Signal Connections (changed D1/SPIDS to D1).
		Ordering Information	Updated LatticeECP2M Part Number Description diagram.
March 2010	03.6	DC and Switching Characteristics	Footnote for SED operating frequency added to the sysCONFIG Port Timing Specifications table.
		Pinout Information	Changed Dual Function pin E7 to be D7/SPDI0 in Logic Signal Connections tables. Changed footnote (***) in Logic Signal Connections table.
July 2010	03.7	Architecture	Updated the Typical sysIO Behavior During Power-up text section.
		Pinout Information	Added reference to powerup information.
			Corrected reference to footnote for pins 131 and 132 for the LFE-20E/SE, 208 PQFP.
			Referenced footnote (***) for all D7/SPID0.
			Changed D7*** to D7/SPID0.
			Corrected *** footnote.
		All Sections	Included references to Lattice Diamond design software wherever ispLEVER and ispLeverCORE is specified.
April 2011	03.8	DC and Switching Characteristics	DC Electrical Characteristics table: - Added footnote 3 to I_{IH} - Added footnote 2 to I_{IL}, I_{IH} - Updated C1 and C2 typ. and max. data.
			DLL Timing table – Removed line for t_R and t_F.
			LatticeECP2/M sysCONFIG Port Timing Specifications table – added footnote to t_{DINIT}.
			Figure 3-18 – Corrected label to be PRGM (not PRGMRJ).
		Pinout Information	LFE2-12E/SE and LFE-20/SE Logical Signal Connections for 208 PQFP – Corrected Dual Function information for pins 112, 114, 117, 119.
January 2012	03.9	Multiple	Removed references to ispLEVER design software.
		Architecture	Corrected information regarding SED support.
		DC and Switching Characteristics	Added reference to ESD information.

Section II. LatticeECP2/M Family Technical Notes

LatticeECP2M SERDES/PCS Usage Guide

September 2011 **Technical Note TN1124**

Introduction to PCS

The LatticeECP2M™ FPGA family combines a high-performance FPGA fabric, high-performance I/Os and large embedded RAM in a single industry-leading architecture. All LatticeECP2M devices also feature up to 16 channels of embedded SERDES with associated Physical Coding Sublayer (PCS) logic. The PCS logic can be configured to support numerous industry-standard, high-speed data transfer protocols.

Each channel of PCS logic contains dedicated transmit and receive SERDES for high-speed full-duplex serial data transfers at data rates up to 3.125 Gbps. The PCS logic in each channel can be configured to support an array of popular data protocols including Ethernet (1GbE and SGMII), PCI Express, CPRI, and OBSAI. In addition, the protocol-based logic can be fully or partially bypassed in a number of configurations to allow users flexibility in designing their own high-speed data interface.

The PCS also provides bypass modes that allow for a direct 8-bit or 10-bit interface from the SERDES to the FPGA logic. Each SERDES pin can be independently DC-coupled and can allow for both high-speed and low-speed operation on the same SERDES pin for such applications as Serial Digital Video.

Features

- Up to 16 Channels of High-Speed SERDES
  - 250 Mbps to 3.125 Gbps per channel
  - 3.125Gbps operation with low 100mW power per channel
  - Receive equalization and transmit pre-emphasis for small form factor backplane operation
  - Supports PCI Express, Ethernet (1GbE and SGMII) plus multiple other standards
  - Supports user specified generic 8b10b mode
  - Beacon support for PCI Express
  - Out-of-band signal interface for low speed inputs (video application)
- Multiple Clock Rate Support
  - Separate reference clocks for each PCS quad allow easy handling of multiple protocol rates on a single device
- Full Function Embedded Physical Coding Sub-layer (PCS) Logic Supporting Industry Standard Protocols
  - Up to 16 channels of full-duplex data supported per device
  - Multiple protocol support on one chip
  - Supports popular 8b10b based packet protocols
  - SERDES Only mode allows direct 8- or 10-bit interface to FPGA logic
- Gigabit Ethernet Support
  - IEEE 1000BASE-X compliant
  - 8b10b encoding/decoding
  - Insertion of /I2/ symbols into the receive data stream for auto-negotiation support
  - Comma character word alignment
  - Clock Tolerance Compensation circuit
- PCI Express Support
  - x1 to x4 support in one PCS quad
  - Integrated Word aligner
  - 8b10b encoding/decoding
  - Clock Tolerance Compensation circuit
  - Electrical Idle and Receiver Detection support
  - Support for Beacon Transmission and Beacon Detection

- Multiple Protocol Compliant Clock Tolerance Compensation (CTC) Logic
  - Compensates for frequency differential between reference clock and received data rate
  - Allows user defined Skip pattern of 1, 2, or 4 bytes in length
- Integrated Loopback Modes for System Debugging
  - Three loopback modes are provided for system debugging

Supported Standards

The supported standards are listed in Table 8-1.

Table 8-1. Supported SERDES Standards (Fully Supported)

Standard	Rates (Mbps)	REFCLK (MHz)	FPGA CLK (MHz)	Encoding	Signal Type
PCI Express	2500	100	250	8b10b	CML
GbE/SGMII	1250	125	125	8b10b	CML
Generic 8b10b	250 to 3125	25 to 312.5	25 to 312.5	8b10b/None	CML
10-bit SERDES Only[1]	250 to 3125	25 to 312.5	25 to 312.5	None	CML
8-bit SERDES Only[1]	250 to 3125	25 to 312.5	25 to 312.5	None	CML
SD-SDI[2]	143, 177, 270, 360	14.3, 17.7, 27, 36	143, 177, 135, 180	SMPTE scrambled	CML
HD-SDI	1483.5, 1485	148.35, 148.5	148.35, 148.5	SMPTE scrambled	CML
CPRI	614.4 1228.8	61.44 122.88	61.44 122.88	8b10b	CML

1. 8-bit SERDES Only Mode and 10-bit SERDES Only Mode bypass the Link Align/Comma Align, 8b10b encoder/decoder and the CTC. It does not bypass the CDR.
2. Serial Digital Interface (SDI) for Standard Definition (SD): 143Mbps, 177Mbps bypasses the SERDES/PCS Block. The clock and data come in to the FPGA through the RX pins but get into the FPGA core via the BSCAN path. CDR is done in the FPGA core. In the transmit direction, decimation is used for these “low” bit rates. To do the CDR in the FPGA, reference clock of 14.3MHz and 17.7MHz are required respectively. 270Mbps is the most common frequency. This will go through the 10-bit data path.

It is possible to support XAUI, SRIO, OBSAI, CPRI, 1XFC, 2XFC, PICMG 3.1, PICMG 3.4, PICMG 3.5 and 3G-SDI standards with the SERDES modes specified above. Contact Lattice Semiconductor Technical Support Group for additional information.

Architecture Overview

The PCS logic is arranged in quads containing logic for four independent full-duplex data channels. Table 8-2 shows the availability of SERDES/PCS quads for each device in the LatticeECP2M family.

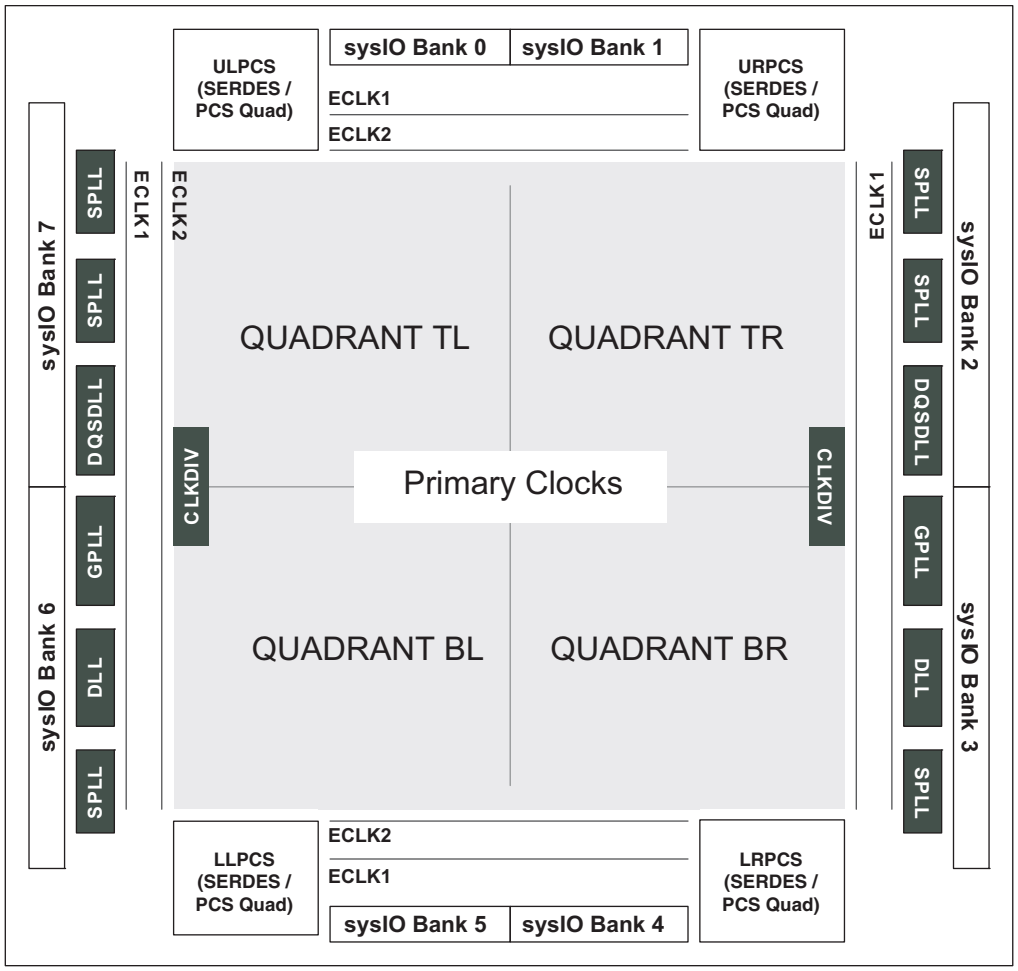
Table 8-2. SERDES/PCS Quads per LatticeECP2M Device

Device	ECP2M20	ECP2M35	ECP2M50	ECP2M70	ECP2M100
Quad URC	Yes	Yes	Yes	Yes	Yes
Quad LRC	—	—	Yes	Yes	Yes
Quad ULC	—	—	—	Yes	Yes
Quad LLC	—	—	—	Yes	Yes

PCS Quad

Figure 8-1 is a layout of a LatticeECP2M device showing the arrangement of PCS quads on the device (the largest array containing 4 quads is shown. Other devices have fewer quads).

Figure 8-1. LatticeECP2M70/100 Block Diagram



Every quad can be programmed into one of several protocol based modes. Each quad requires its own reference clock which can be sourced externally from package pins or internally from the FPGA logic.

Since each quad has its own reference clock, different quads can support different standards on the same chip. This feature makes the LatticeECP2M family of devices ideal for bridging between different standards.

PCS quads are not dedicated solely to industry standard protocols. Each quad (and each channel within a quad) can be programmed for many user defined data manipulation modes. For example, modes governing user-defined word alignment, and clock tolerance compensation can be programmed for non-protocol operation.

PCS Quad and Channels

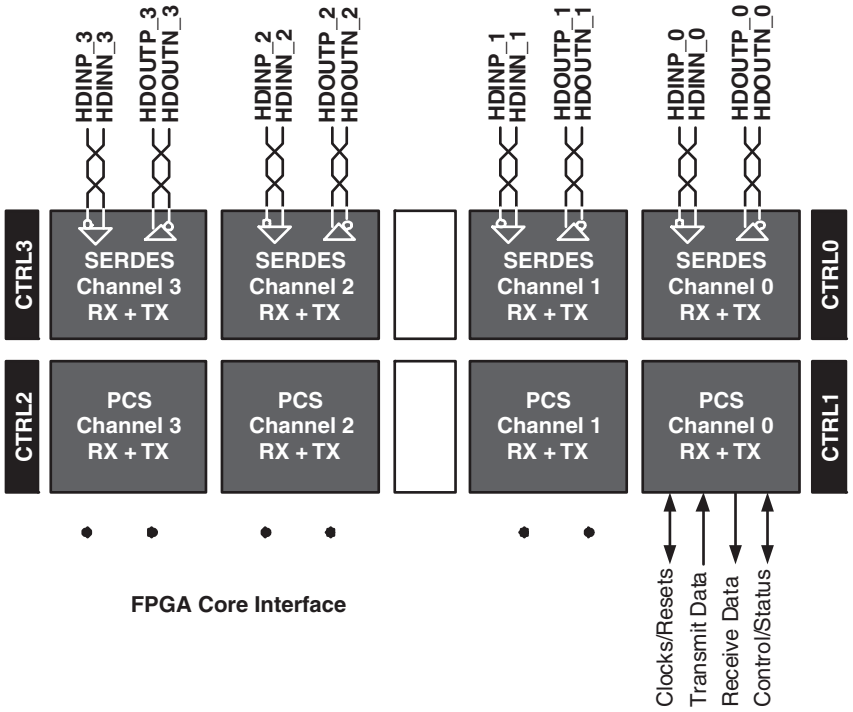
Each quad on a device supports up to four channels of full-duplex data. The user can utilize anywhere from one to four channels in a quad depending on the application. Many options can be set by the user for each channel independently within a given quad.

Figure 8-1 also shows an example of a device with four PCS quads which contain a total of 16 PCS channels. Quads are named according to the location of the respective quad on the LatticeECP2M array: URPCS (Upper Right PCS), ULPCS (Upper Left PCS), LRPCS (Lower Right PCS), LLPCS (Lower Left PCS).

Per Channel PCS/FPGA Interface Ports

All PCS quads regardless of chosen mode have the same external high speed serial interface at the package pins. However, every PCS mode has its own unique list of input/output ports from/to the FPGA logic appropriate to the protocol chosen for the quad. A detailed description of the quad input/output signals for each mode is provided in this document. A simplified diagram showing the channels within a single quad is shown in Figure 8-2.

Figure 8-2. PCS Quad Block Diagram



Locating a PCS Quad

LatticeECP2M-50 and larger devices include two to four PCS Quads.

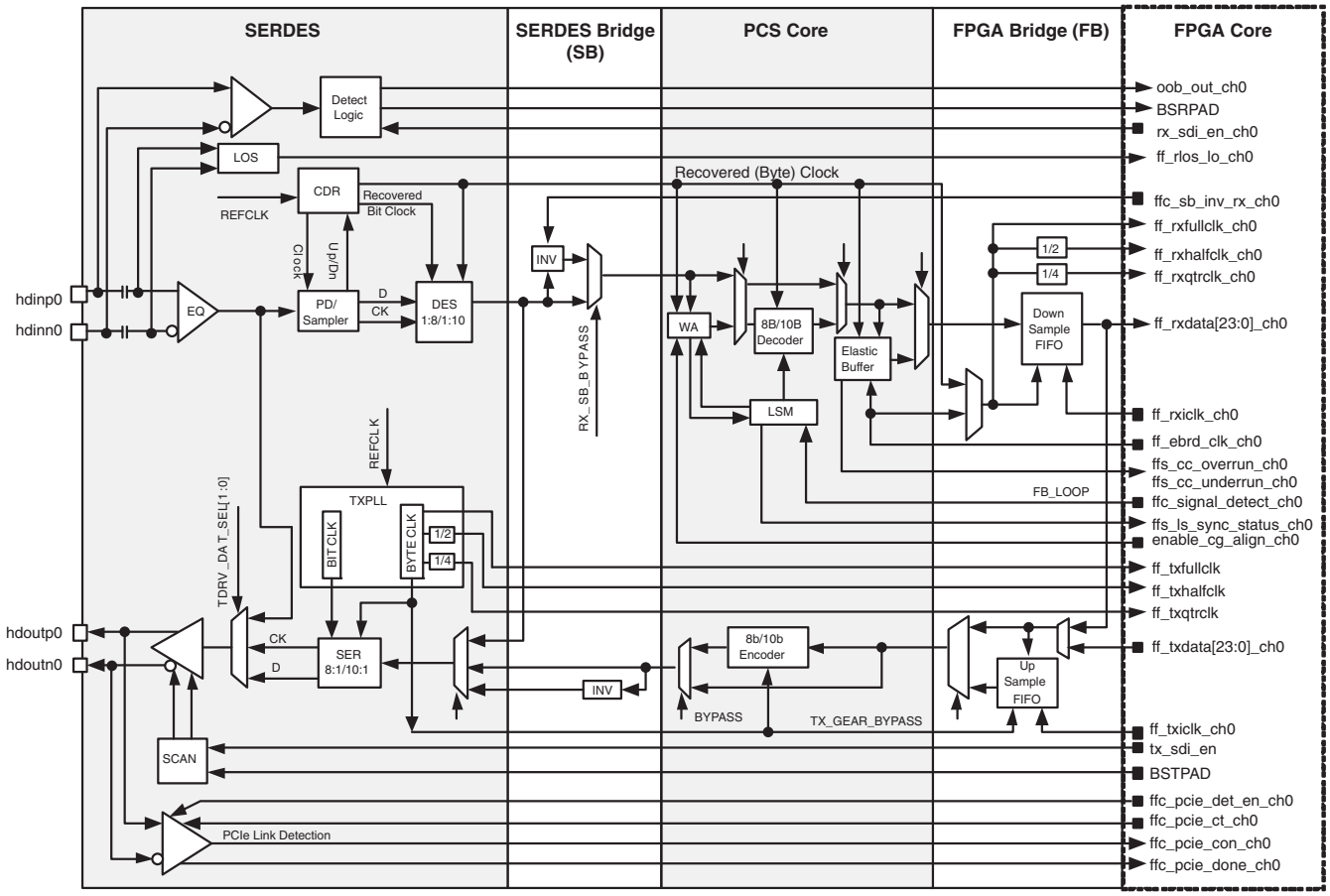
Users can locate each PCS Quad in a desired location using the LOCATE preference in the preference file. Below is an example of the preference, LOCATE.

```
LOCATE COMP "PCS_instantiation_1" SITE URPCS ;
```

Detailed Channel Block Diagram

Figure 8-3 is a detailed block diagram representation of the major functionality in a single channel of the LatticeECP2M SERDES/PCS. This diagram shows all the major blocks and the majority of the control and status signals that are visible to the user logic in the FPGA. This diagram also shows the major sub-blocks in the channel-SERDES, SERDES Bridge, PCS Core and the FPGA Bridge.

Figure 8-3. LatticeECP2M SERDES/PCS Detailed Channel Block Diagram



A general description of the FPGA interface signals follows.

Clocks and Resets

A PCS quad supplies per channel locked reference clocks and per channel recovered receive clocks to the FPGA logic interface. Each PCS quad provides these clocks on both primary and secondary FPGA clock routing. The PCS/FPGA interface also has ports for the transmit and receive clocks supplied from the FPGA fabric for all four channels in each quad.

Each quad has reset inputs to force reset of both the SERDES and PCS logic in a quad or just the SERDES. In addition, separate resets dedicated for the PCS logic are provided for each channel for both the transmit and receive directions.

Transmit Data Bus

The signals for the transmit data path are from the FPGA to the FPGA Bridge in the PCS Block. For high-speed standards, the datapath can be geared 2:1 to the internal PCS data path, which is 8 bits wide (+ control/status signals). The highest speed of the interface for PCI Express x1 is 250MHz in non-geared mode. With gearing (i.e. a 16-bit wide data path), the maximum speed is 156.25MHz (for XAUI 4x channel mode). The SERDES and PCS will support data rates up to 3.125Gbps data that correspond to an interface speed of 156.25MHz (with 2:1 gearing).

Receive Data Bus

The signals for the receive path are from the FPGA Bridge in the PCS Block to the FPGA. The data path may be geared 2:1 to the internal PCS data path which is 8 bits wide. It is possible to disable the gearing via a software register bit, in which case, the bus widths are halved. When the data is geared, the lower bits (ff_rx_d[9:0]) are the octet that has been received first and the higher bits (ff_rx_d[19:10]) are the octet that has been received second. If the data is not geared, the lower bits ((ff_rx_d[9:0]) are the active bits and the higher bits should not be used.

Table 8-3. Data Bus Usage by Mode

Data Bus PCS Cell Name[5]	G8B10B	CPRI	PCI Express	GIGE	8BSER	10BSER	SDI
FF_TX_D_0_0	ff_txdata_ch0[0]						
FF_TX_D_0_1	ff_txdata_ch0[1]						
FF_TX_D_0_2	ff_txdata_ch0[2]						
FF_TX_D_0_3	ff_txdata_ch0[3]						
FF_TX_D_0_4	ff_txdata_ch0[4]						
FF_TX_D_0_5	ff_txdata_ch0[5]						
FF_TX_D_0_6	ff_txdata_ch0[6]						
FF_TX_D_0_7	ff_txdata_ch0[7]						
FF_TX_D_0_8	ff_tx_k_cntrl_ch0[0]				GND	ff_txdata_ch0[8]	
FF_TX_D_0_9	ff_force_disp_ch0[0]][1]			GND		ff_txdata_ch0[9]	
FF_TX_D_0_10	ff_disp_sel_ch0[0]][1]			ff_xmit_ch0[0]][2]		GND	
FF_TX_D_0_11	GND		ff_pci_ei_en_ch0[0]	ff_correct_disp_ch0[0]		GND	
FF_TX_D_0_12	ff_txdata_ch0[8]					ff_txdata_ch0[10]	
FF_TX_D_0_13	ff_txdata_ch0[9]					ff_txdata_ch0[11]	
FF_TX_D_0_14	ff_txdata_ch0[10]					ff_txdata_ch0[12]	
FF_TX_D_0_15	ff_txdata_ch0[11]					ff_txdata_ch0[13]	
FF_TX_D_0_16	ff_txdata_ch0[12]					ff_txdata_ch0[14]	
FF_TX_D_0_17	ff_txdata_ch0[13]					ff_txdata_ch0[15]	
FF_TX_D_0_18	ff_txdata_ch0[14]					ff_txdata_ch0[16]	
FF_TX_D_0_19	ff_txdata_ch0[15]					ff_txdata_ch0[17]	
FF_TX_D_0_20	ff_tx_k_cntrl_ch0[1]				GND	ff_txdata_ch0[18]	
FF_TX_D_0_21	ff_force_disp_ch0[1]][1]			GND		ff_txdata_ch0[19]	
FF_TX_D_0_22	ff_disp_sel_ch0[1]][1]			ff_xmit_ch0[1]][2]		GND	
FF_TX_D_0_23	GND		ff_pci_ei_en_ch0[1]	ff_correct_disp_ch0[1]		GND	
FF_RX_D_0_0	ff_rxdata_ch0[0]						
FF_RX_D_0_1	ff_rxdata_ch0[1]						
FF_RX_D_0_2	ff_rxdata_ch0[2]						
FF_RX_D_0_3	ff_rxdata_ch0[3]						
FF_RX_D_0_4	ff_rxdata_ch0[4]						
FF_RX_D_0_5	ff_rxdata_ch0[5]						
FF_RX_D_0_6	ff_rxdata_ch0[6]						
FF_RX_D_0_7	ff_rxdata_ch0[7]						
FF_RX_D_0_8	ff_rx_k_cntrl_ch0[0]				NC	ff_rxdata_ch0[8]	
FF_RX_D_0_9	ff_disp_err_ch0[0]		ff_rxstatus0_ch0[0]	ff_disp_err_ch0[0]	NC	ff_rxdata_ch0[9]	
FF_RX_D_0_10	ff_cv_ch0[0]][3]		ff_rxstatus1_ch0[0]	ff_cv_ch0[0]][3]		NC	
FF_RX_D_0_11	NC		ff_rxstatus2_ch0[0]	ff_rx_even_ch0[0]][4]		NC	
FF_RX_D_0_12	ff_rxdata_ch0[8]					ff_rxdata_ch0[10]	
FF_RX_D_0_13	ff_rxdata_ch0[9]					ff_rxdata_ch0[11]	
FF_RX_D_0_14	ff_rxdata_ch0[10]					ff_rxdata_ch0[12]	
FF_RX_D_0_15	ff_rxdata_ch0[11]					ff_rxdata_ch0[13]	
FF_RX_D_0_16	ff_rxdata_ch0[12]					ff_rxdata_ch0[14]	
FF_RX_D_0_17	ff_rxdata_ch0[13]					ff_rxdata_ch0[15]	

Table 8-3. Data Bus Usage by Mode (Continued)

Data Bus PCS Cell Name[5]	G8B10B	CPRI	PCI Express	GIGE	8BSER	10BSER	SDI
FF_RX_D_0_18	ff_rxdata_ch0[14]					ff_rxdata_ch0[16]	
FF_RX_D_0_19	ff_rxdata_ch0[15]					ff_rxdata_ch0[17]	
FF_RX_D_0_20	ff_rx_k_cntrl_ch0[1]				NC	ff_rxdata_ch0[18]	
FF_RX_D_0_21	ff_disp_err_ch0[1]		ff_rxstatus0_ch0[1]	ff_disp_err_ch0[1]	NC	ff_rxdata_ch0[19]	
FF_RX_D_0_22	ff_cv_ch0[1]][3]		ff_rxstatus1_ch0[1]	ff_cv_ch0[1]][3]	NC		
FF_RX_D_0_23	NC		ff_rxstatus2_ch0[1]	ff_rx_even_ch0[1]][4]	NC		

1. The force_disp signal will force the disparity for the associated data word on bits [7:0] to the column selected by the tx_disp_sel signal. If disp_sel is a one, the 10-bit code is taken from the 'current RD+' column (positive disparity). If the tx_disp_sel is a zero, the 10-bit code is taken from the 'current RD-' (negative disparity) column.
2. The auto-negotiation state machine generates the signal xmit. It is used to interact with the GIGE Idle State Machine in the hard logic.
3. When there is a code violation, the packet PCS 8b10b decoder will replace the output from the decoder with hex EE and K asserted (K=1 and d=EE is not part of the 8b10b coding space).
4. rx_even is a signal generated by the GIGE Link State Machine for the use of the GIGE Auto-negotiation and Receive State Machines (which is part of the IP core).
5. FF_TX_D_0_0: FPGA Fabric Transmit Data Bus Channel 0 Bit 0.

Control

Each mode has its own set of control signals which allows direct control of various PCS features from the FPGA logic. In general, each of these control inputs duplicate the effect of writing to a corresponding control register bit or bits. The ispLEVER® design tools give the user the option to bring these ports out to the FPGA interface.

Status

Each mode has its own set of status or alarm signals that can be monitored by the FPGA logic. In general, each of these status outputs correspond to a specific status register bit or bits. The ispLEVER design tools give the user the option to bring these port out to the PCS FPGA interface. Refer to the Mode Specific Control/Status Signals section for detailed information about control and status signals.

SCI (SERDES Client Interface) Bus

The SERDES Client Interface (SCI) is a soft IP that allow the SERDES/PCS Quad block to be controlled by registers as oppose to the configuration memory cells. It is a simple register configuration interface.

Using This Technical Note

The ispLEVER design tools from Lattice support all modes of the PCS. Most modes are dedicated to applications for a specific industry standard data protocol. Other modes are more general purpose modes which allow a user to define their own custom application settings. ispLEVER design tools allow the user to define the mode for each quad in their design. This document describes operation of the SERDES and PCS for all modes supported by ispLEVER. If you are using Lattice Diamond™ design software, see Appendix E.

This document provides a thorough description of the complete functionality of the embedded SERDES and associated PCS logic. Electrical and Timing Characteristics of the embedded SERDES are provided in the LatticeECP2/M Family Data Sheet. Operation of the PCS logic is provided in the PCS section. A table of all status and control registers associated with the SERDES and PCS logic which can be accessed via the SCI Bus is provided in the Memory Map section. Package pinout information is provided in the Architecture section of the LatticeECP2/M Family Data Sheet.

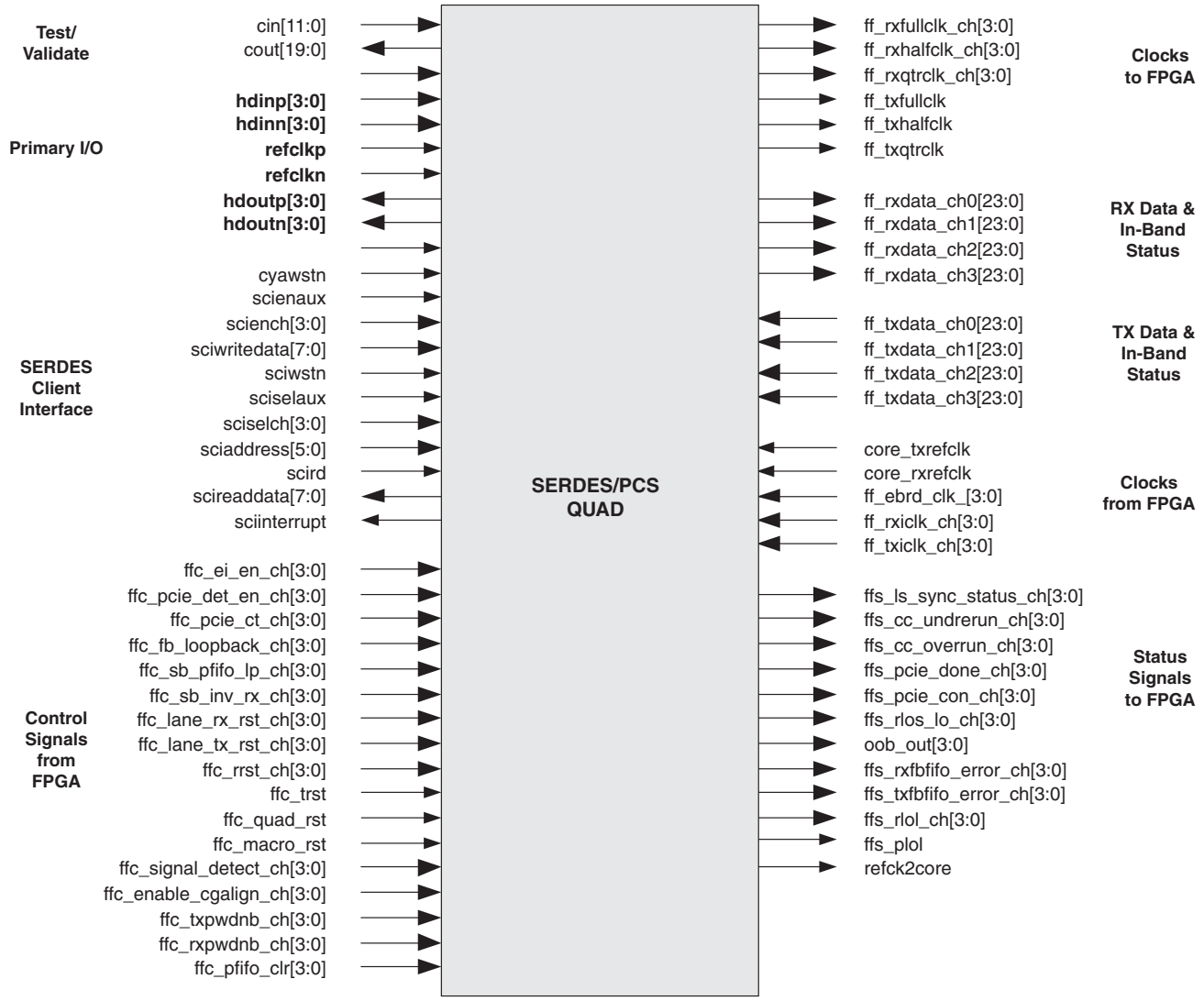
SERDES/PCS

The quad contains four channels with both Rx and Tx circuits, and an auxiliary channel that contains the Tx PLL. The reference clock to the Tx PLL can be provided either by the primary differential reference clock pins or by the FPGA core. The quad SERDES/PCS macro performs the serialization and de-serialization function for four lanes

of data. In addition, the PLL within the SERDES/PCS block provides the system clock for the FPGA logic. The quad also supports both full-data-rate and half-data-rate modes of operation on each Tx and Rx circuit independently.

The block level diagram is shown in Figure 8-4.

Figure 8-4. SERDES_PCS Block Signal Interface



I/O Definitions

Table 8-4 lists all default and optional input and outputs to/from a PCS quad. Users can choose optional ports for a PCS quad using the IPexpress™ GUI.

Table 8-4. SERDES_PCS I/O Descriptions

Signal Name	I/O	Type	Description	Default/ Optional
Primary I/O, SERDES Quad				
hdinp0	I	Channel	High-speed CML input, positive, channel 0	D
hdinn0	I	Channel	High-speed CML input, negative, channel 0	D
hdinp1	I	Channel	High-speed CML input, positive, channel 1	D
hdinn1	I	Channel	High-speed CML input, negative, channel 1	D
hdinp2	I	Channel	High-speed CML input, positive, channel 2	D
hdinn2	I	Channel	High-speed CML input, negative, channel 2	D
hdinp3	I	Channel	High-speed CML input, positive, channel 3	D
hdinn3	I	Channel	High-speed CML input, negative, channel 3	D
hdoutp0	O	Channel	High-speed CML output, positive, channel 0	D
hdoutn0	O	Channel	High-speed CML output, negative, channel 0	D
hdoutp1	O	Channel	High-speed CML output, positive, channel 1	D
hdoutn1	O	Channel	High-speed CML output, negative, channel 1	D
hdoutp2	O	Channel	High-speed CML output, positive, channel 2	D
hdoutn2	O	Channel	High-speed CML output, negative, channel 2	D
hdoutp3	O	Channel	High-speed CML output, positive, channel 3	D
hdoutn3	O	Channel	High-speed CML output, negative, channel 3	D
refclkp	I	Quad	Reference Clock input, positive, Dedicated CML input	D
refclkn	I	Quad	Reference Clock input, negative, Dedicated CML input	D
Receive / Transmit Data Bus (See Table for Detailed Data Bus Usage)				
ff_rxdata_ch0[23:0]	O	Channel	Data Signals for the channel 0 receive path	D
ff_rxdata_ch1[23:0]	O	Channel	Data Signals for the channel 1 receive path	D
ff_rxdata_ch2[23:0]	O	Channel	Data Signals for the channel 2 receive path	D
ff_rxdata_ch3[23:0]	O	Channel	Data Signals for the channel 3 receive path	D
ff_txdata_ch0[23:0]	I	Channel	Data Signals for the channel 0 transmit path	D
ff_txdata_ch1[23:0]	I	Channel	Data Signals for the channel 1 transmit path	D
ff_txdata_ch2[23:0]	I	Channel	Data Signals for the channel 2 transmit path	D
ff_txdata_ch3[23:0]	I	Channel	Data Signals for the channel 3 transmit path	D
Control Signals				
ffc_sb_inv_rx_ch[3:0]	I	Channel	Control the inversion of received data. 1 = Invert the data 0 = Do not invert the data	O
ffc_enable_cgalign_ch[3:0][4]	I	Channel	Control comma aligner. 1 = Enable comma aligner 0 = Lock comma aligner at current position.	O
ffc_signal_detect_ch[3:0][4]	I	Channel	Control Link State Machine 1 = Enable Link State Machine 0 = Disable Link State Machine	O
ffc_fb_loopback_ch[3:0]	I	Channel	FPGA Bridge Loopback. 1 = Enable loopback from Rx to Tx 0 = Normal data operation	O

Table 8-4. SERDES_PCS I/O Descriptions (Continued)

Signal Name	I/O	Type	Description	Default/ Optional
ffc_sb_pfifo_lp_ch[3:0]	I	Channel	SERDES Bridge Parallel Loopback 1 = Enable loopback from Rx to Tx, 0 = Normal data operation	O
ffc_pfifo_clr_ch[3:0]	I	Channel	SERDES Bridge Parallel Loopback FIFO Clear 1 = Reset Loopback FIFO 0 = Normal Loopback operation	D
rx_sdi_en	I	Channel	These signals are used in BSCAN mode only.	O
tx_sdi_en	I	Quad		
Reset Signals				
ffc_lane_rx_rst_ch[3:0]	I	Channel	Active high, asynchronous input. Resets individual Rx channel logic only in PCS.	D
ffc_lane_rx_tst_ch[3:0]	I	Channel	Active high, asynchronous input. Resets individual Tx channel logic only in PCS.	D
ffc_rrst_ch[3:0]	I	Channel	Active high. Resets selected digital logic in the SERDES Receive channel	D
ffc_trst	I	Quad	Active high, resets selected digital logic in all SERDES Transmit channels.	D
ffc_quad_rst	I	Quad	Active high, asynchronous input. Resets all SERDES channels including the auxiliary channel and PCS.	D
ffc_macro_rst	I	Quad	Active high, asynchronous input to SERDES quad. Resets all SERDES channels including the AUX channel but not PCS logic.	D
ffc_txpwdnb_ch[3:0]	I	Channel	Active low transmit channel power down. 0 = Transmit Channel Power Down.	D
ffc_rxpwdnb_ch[3:0]	I	Channel	Active low receive channel power down. 0 = Receive Channel Power Down.	D
Status Signals				
ffs_rlos_lo_ch[3:0]	O	Channel	Loss of signal detection for each channel. Register bits rlos_hset[2:0] are used to set the threshold. Low threshold is not user accessible. 1 = Loss of signal 0 = Signal detected	D
ffs_ls_sync_status_ch[3:0]	O	Channel	1 = Lane is synchronous to commas. 0 = Lane has not found comma.	D
ffs_cc_underrun_ch[3:0][6]	O	Channel	1 = Receive clock compensator FIFO underrun error, 0 = No FFIFO errors.	O
ffs_cc_overrun_ch[3:0][6]	O	Channel	1 = Receive clock compensator FIFO overrun error 0 = No FIFO errors.	O
ffs_rxfbfifo_error_ch[3:0]	O	Channel	1 = Receive FPGA bridge FIFO error 0 = No FIFO errors	D
ffs_txfbfifo_error_ch[3:0]	O	Channel	1 = Transmit FPGA bridge FIFO error 0 = No FIFO errors.	D
ffs_rlol_ch[3:0]	O	Channel	1 = Receive CDR loss of lock 0 = Lock maintained	D
ffs_plol	O	Quad	1 = Transmit PLL loss of lock 0 = Lock maintained	D
oob_out_ch[3:0][3]	O	Channel	Single ended outputs to video SERDES (in FPGA).	D
refck2core	O	Quad	Reference clock to FPGA core.	O

Table 8-4. SERDES_PCS I/O Descriptions (Continued)

Signal Name	I/O	Type	Description	Default/ Optional
Clock Signals to FPGA				
ff_rxfullclk_ch[3:0]	O	Channel	Receive channel recovered clock. In user mode, the source is always the channel's recovered clock. For standards such as GbE, 10 GbE that support clock compensation, the source is the respective transmit channel's system clock. For PCS bypass modes, it is also the Tx system clock, thus requiring raw mode to actually be done using either 8b10b mode with the 8b10b decoder disabled (10-bit or 20-bit data path).	D
ff_rxhalfclk_ch[3:0]	O	Channel	Receive channel recovered half clock. In 2:1 gearing mode, it is a divide-by-2 output.	D
ff_rxqtrclk_ch[3:0]	O	Channel	Receive channel recovered quarter clock. Available for further 2:1 gearing.	O
ff_txfullclk	O	Quad	Tx PLL full rate clock.	D
ff_txhalfclk	O	Quad	Tx PLL half clock.	D
ff_txqtrclk	O	Quad	Tx PLL quarter clock	O
Clock Signals from FPGA				
core_rxrefclk	I	Quad	Rx Reference clock from FPGA logic, for CDR PLL	D
core_txrefclk	I	Quad	Tx Reference clock from FPGA logic, for Tx SERDES PLL	D
ff_ebrd_clk_[3:0]	I	Channel	Receive channel clock input from FPGA for CTC FIFO (Elastic Buffer) read	D
ff_rxiclk_ch[3:0]	I	Channel	Receive channel clock input from FPGA. Used to clock the Rx FPGA Interface FIFO with a clock synchronous to the reference and/or receive reference clock.	D
ff_txiclk_ch[3:0]	I	Channel	Transmit channel clock input from FPGA.Per channel transmit clock inputs from FPGA. Used to clock the Tx FPGA. Interface FIFO with clock synchronous to the reference clock. Also used to clock the Rx FPGA Interface FIFO with a clock synchronous to the reference clock when CTC is used.	D
SERDES Client Interface (SCI)				
scienaux	I	R	1: sciwdata is written to the quad control registers 0: memory data is written to the quad control registers	O
scien_ch[3:0]	I	R	1: sciwdata is written to the channel control registers 0: memory data is written to the channel control registers	O
sciselaux	I	R	1: select quad registers	O
scisel_ch[3:0]	I	R	1: select channel registers	O
sciaddress[5:0]	I	R	Address bus input	O
scireaddata[7:0]	O	R	Read data output	O
sciwritedata[7:0]	I	R	Write data input	O
scird	I	R	1: Read data select 0: Read data not selected	O
sciwstn	I	R	Write strobe	O
sciinterrupt	O	R	Interrupt output	O
cyawstn[5]	I	R	1: Copy all memory cells to registers if sciwstn = 0 0: Default	O
SERDES Characterization / Test Bus				
cin[11:0]	I	R	Characterization test bus logic data input	D

Table 8-4. SERDES_PCS I/O Descriptions (Continued)

Signal Name	I/O	Type	Description	Default/ Optional
cout[19:0]]	O	R	Characterization test bus logic data output	D

1. During configuration, both HDOUTP and HDOUTN are pulled high to VCCOB.
2. The Generic 8b10b PCS module includes four PCI control and status signals as options. If not used, the control signals may be tied to GND and the status signals may be left float.
3. The only way to get a signal out without using the CDR is to use the OOB_OUT signal. This signal is available in SDI mode only.
4. These signals appear in the PCS port list when the external link state machine is selected. Refer to Figure 8-28.
5. For factory use only.
6. These signals are pulses. In order to properly monitor these status signals, they must be latched.

SERDES/PCS Functional Description

Devices in the LatticeECP2M family have from one to four quads of embedded SERDES/PCS logic. Each quad, in turn, supports four independent full-duplex data channels. A single channel can support a data link and each quad can support up to four such channels. Note that mode selection is done on a per quad basis. For example, the selection of Gigabit Ethernet mode for a quad dedicates all four channels in that quad to Gigabit Ethernet mode.

The embedded SERDES CDR PLLs and Tx PLLs support data rates which cover a wide range of industry standard protocols.

Figure describes the major blocks and sub-blocks in a SERDES/PCS channel.

- SERDES
  - Equalizer
  - CDR (Clock and Data Recovery)
  - Deserializer
  - PreEmphasis
  - Serializer
  - Serial Loopback
- SERDES Bridge (SB)
  - Inverter: inverts receive data. Required by PCI Express
  - SERDES Bridge Parallel Loopback
- PCS Core
  - Word Alignment
  - 8b10b Decoder
  - 8b10b Encoder
  - Link State Machine
  - Elastic Buffer (CTC)
- FPGA Bridge (FB)
  - Down-sample FIFO
  - Up-sample FIFO
  - PCS Parallel Loopback

SERDES

Equalizer

As the data rate of digital transmission advances over Gbps, frequency-dependent attenuation results in severe intersymbol interference in the received signal and makes it mandatory to use equalizer in the data transceiver to recover data correctly. Three pole positions are provided: low, medium and high frequency range.

Pre-Emphasis

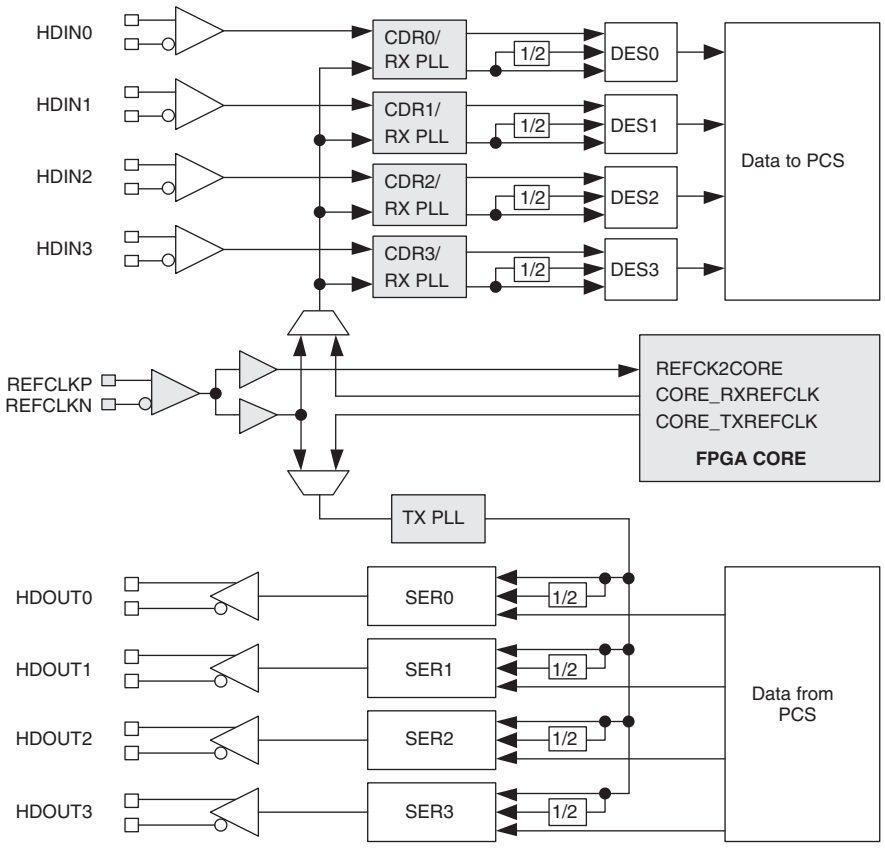
Pre-emphasis refers to a system process designed to increase the magnitude of some frequencies with respect to the magnitude of other frequencies. The goal is to improve the overall signal-to-noise ratio by minimizing the adverse effects of such phenomena as attenuation differences or saturation of recording media in subsequent parts of the system. User can select up to 80% of pre-emphasis.

Reference Clock Usage

One reference clock (REFCLK) is supported in the LatticeECP2M family. The Tx PLL and the four Rx PLLs all run at the same frequency, which is a multiple of the reference clock frequency. The Tx serializer in each channel can be independently programmed to run at this rate (full-data-rate mode) or half of this rate (half-data-rate mode). Similarly, the Rx deserializer in each channel can be independently programmed to run at this rate (full-data-rate mode) or half of this rate (half-data-rate mode). If all Tx and Rx are programmed in the same mode (normally this will be full-rate) then all four channels in the quad will run at the same frequency, both Tx and Rx.

The transmit PLL in the SERDES is able to lock to either an external reference clock from the pins, or a reference clock provided from the FPGA core (core_txrefclk). The receive CDRs in the SERDES is able to lock to either an external reference clock from the pins, or a reference clock provided by the FPGA core (core_rxrefclk).

Figure 8-5. Block Diagram, Reference Clock Usage



Reference Clock Sources

refclkp, refclkn

Dedicated CML input. This is the first choice unless different clock sources for rx and tx are used. The clock signal may be CML, LVDS or LVPECL. Refer to TN1114, Electrical Recommendations for Lattice SERDES, for example interface circuits.

core_rxrefclk, core_txrefclk

Reference clock from FPGA logic. The Primary Clock pad (PCLK) should be used as the clock input pin to the FPGA. The clock signal may be CML, LVDS, LVPECL or single-ended.

FPGA PLL

When an FPGA PLL is used as the reference clock, the reference clock to PLL should be assigned to a dedicated PLL input pad. The FPGA PLL output jitter may not meet system specifications at higher data rates. Use of an FPGA PLL is not recommended in jitter-sensitive applications.

Full-Data-Rate and Half-Data-Rate

Each Tx Serializer and Rx Deserializer can be split into full-data-rate and half-data-rate, allowing two different data rates in each direction and in each channel.

The Channel Based Protocol Mode must be selected to use this dual rate feature.

Example:

1. In the **Quad Tab window** of IPexpress (Figure 8-22), **G8B10B Mode, Channel Based Protocol Mode, Channel 0 (Full Rate) and Channel 1 (Half Rate)** are selected.
2. In the **Reference Clock (CM) window** (Figure 8-24), under the **Full Rate Channel** column, enter the following:
   - Serial Bit Clock Rate: 2.5 GHz
   - Reference Clock Multiplier: 10X
   - Leave the other three entries
3. The **Half Rate Channel** column will display calculated values for Half Rate:

	Full Rate Channel	Half Rate Channel
Serial Bit Clock Rate	2.5 GHz	1.25 GHz
Reference Clock Multiplier	10X	5X
Calculated Reference Clock Rate	250 MHz	250 MHz
FPGA Interface Data Bus Width	8	8
Calculated FPGA Interface Clock Rate	250 MHz	125 MHz

Full Clock, Half Clock and Quarter Clock Usage

In most cases, txfullclk is used for ff_rxiclk_chx, ff_txiclk_chx, ff_ebrd_clk_x as illustrated in Figure 8-32.

The IPexpress GUI automatically calculates the FPGA Interface Clock Frequency when Reference Clock Multiplier and FPGA Interface Data Bus Width is selected.

Table 8-5 illustrates clock usage examples in all possible combinations of the refclk_multiplier modes and 8-bit or 16-bit interface Data Bus widths.

Table 8-5. Clock Usage Example – G8B10B Mode, REFCLK = 120MHz

Reference Clock Multiplier	10xH	10x	20xH	20x
Bit Rate	**600 Mbps**	**1.2 Gbps**	**1.2 Gbps**	**2.4 Gbps**
8-Bit Interface Example				
rxfullclk[1]	60	120	120	240
rxhalfclk	30	60	60	120
txfullclk	120	120	240	240
txhalfclk	60	60	120	120
txqtrclk	30	30	60	60
16-Bit Interface Example				
rxfullclk	60	120	120	240
rxhalfclk[1]	30	60	60	120
txfullclk	120	120	240	240
txhalfclk	60	60	120	120
txqtrclk	30[2, 3]	30	60	60

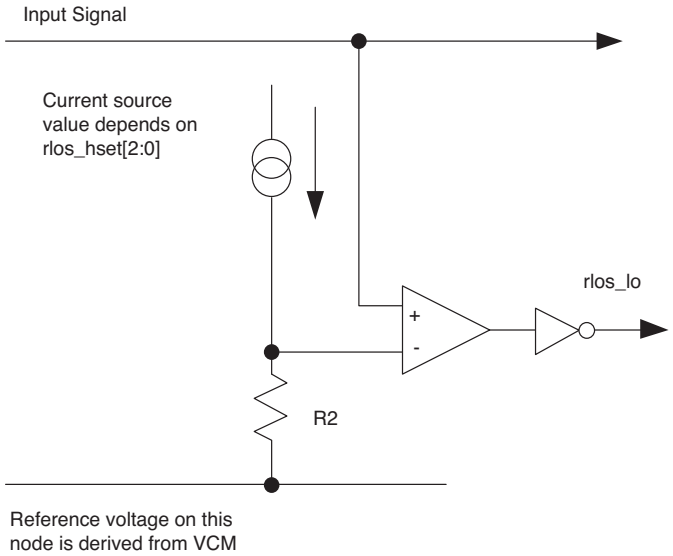
1. These recovered clocks are used as the source of rxiclk in CTC BYPASS mode. See Figures 8-34 and 8-36.
2. The clocks in the shaded cells are used as the FPGA interface clocks in each mode.
3. When this mode is selected, the 'PLL Quarter Clock' must be checked in the Optional Port tab window of the configuration GUI (see Figure 8-28).

The VCO in the Full Data Rate Channel is the same as the bit clock. In Half Data Rate Channel, the bit clock is half of the VCO.

Loss Of Signal (LOS)

Each channel contains a programmable loss-of-signal detector as shown in Figure 8-6. The loss-of-signal threshold depends on the value of the programmable current source. The current source value is chosen using the rlos_hset[2:0] control bits.

Figure 8-6. Loss-of Signal Detector



Note: rlos_lset[2:0] control bits and associated status bits are internal use only.

Loss Of Lock

Both the transmit PLL and the individual channel CDRs have digital counter-based, loss-of-lock detectors. If the transmit PLL loses lock, the loss-of-lock for the PLL is asserted and remains asserted until the PLL reacquires lock.

If a CDR loses lock, the loss-of-lock for that channel is asserted and locking to the reference clock retrains the VCO in the CDR. When this is achieved, loss-of-lock for that channel is de-asserted and the CDR is switched back over to lock to the incoming data. The CDR will either remain locked to the data, or will go back out of lock again in which case the re-training cycle will repeat.

Tx Lane-to-Lane Skew

A control bit, sync_toggle, has been added to reset all the active Tx channels to start serialization with bit0. Most multi-channel protocol standards have requirements to ensure that the Tx lane-to-lane skew is within a certain specification. This is to ensure that most of the Rx De-skew (Multi-channel alignment, which is not supported in hard PCS by Lattice ECP2M) is for channel (trace) de-skewing.

The reset to the Tx serializers is generated either by toggling the sync_toggle control bit or by a transition in PLL Loss of Lock. The reset is applied to all active Tx serializers. If both these source signals are level, then the Tx serializers are operating normally.

PCS Functional Setup

The LatticeECP2M PCS can be configured for use in various applications. Setup is chosen with the ispLEVER® IPexpress module generation tool which allows the user to select the mode and feature options for the PCS. Option selections are saved in an auto-configuration file which is subsequently used by the ispLEVER bitstream generator to write the user selections into the bitstream. To change PCS option selections it is recommended that the user re-run IPexpress to regenerate a PCS module and create a new auto-configuration file. Some options can be changed by manually editing the auto-configuration file before running the bitstream generator.

After configuration, PCS options can be changed dynamically by writing to PCS registers via the optional SERDES Client Interface (SCI) bus. The SERDES Client Interface is soft IP that allows the SERDES/PCS quad to be controlled by registers as opposed to configuration memory cells. A table of control and status registers accessible through the SCI is provided in the Memory Map section of this document.

Auto-Configuration File

Initial register setup for each PCS mode can be performed by using the autoconfiguration feature in ispLEVER. The module generator provides an auto-configuration file which contains the quad and channel register settings for the chosen mode. This file can be referred to for front-end simulation and also can be integrated into the bitstream. When an auto-configuration file is integrated into the bitstream all the quad and channel registers will be set to values defined in the auto-configuration file during configuration. The SCI (SERDES Client Interface) is therefore not needed if all quads are to be set via auto-configuration files. However, the SCI must be included in a design if the user needs to change control registers or monitor status registers during operation.

Transmit Data

The PCS quad transmit data path consists of 8b10b Encoder and Serializer per channel.

8b10b Encoder

This module implements an 8b10b encoder as described within the IEEE 802.3ae-2002 1000BASE-X specification. The encoder performs the 8-bit to 10-bit code conversion as described in the specification, along with maintaining the running disparity rules as specified. The 8b10b encoder can be bypassed on a per channel basis by setting the attribute CHx_8B10B to “BYPASS” where x is the channel number.

Serializer

The 8b10b encoded data undergoes parallel to serial conversion and is transmitted off chip via the embedded SERDES.

Receive Data

The PCS quad receive data path consists of the following sub-blocks per channel: Deserializer, Word Aligner, 8b10b Decoder, Optional Link State Machine, and Optional Receive Clock Tolerance Compensation (CTC) FIFO.

Deserializer

Data is brought on-chip to the embedded SERDES where it goes from serial to parallel.

Word Alignment (Byte Boundary Detect)
This module performs the comma codeword detection and alignment operation. The comma character is used by the receive logic to perform 10-bit word alignment upon the incoming data stream. The word aligner can be bypassed on a per channel basis by setting attribute CHx_COMMA_ALIGN to “BYPASS” where x is the channel number. The comma description can be found in section 36.2.4.9 of the 802.3.2002 1000BASE-X specification as well as section 48.2.6.3, Figure 48-7 of the 10GBASE-X specification.

A number of programmable options are supported within the word alignment module:

- Software enable control (in User Configured - UC mode).

  Note: UC_Mode refers to 8-bit SERDES Only, 10-bit SERDES Only, SD-SDI, HD-SDI.

- Ability to set two programmable word alignment characters (typically one for positive and one for negative disparity) and a programmable per bit mask register for alignment compare. Alignment characters and the mask register is set on a per quad basis. For many protocols, the word alignment characters can be set to “XXX0000011” (jhgfiedcba bits for positive running disparity comma character matching code groups K28.1, K28.5, and K28.7) and “XXX1111100” (jhgfiedcba bits for negative running disparity comma character matching code groups K28.1, K28.5, and K28.7). However the user can define any bit pattern up to 10 bits long.

- The first alignment character is defined by the 10-bit value assigned to attribute COMMA_A. This value applies to all channels in a PCS quad.

- The second alignment character is defined by the 10-bit value assigned to attribute COMMA_B. This value applies to all channels in a PCS quad.

- The mask register defines which word alignment bits to compare (a ‘1’ in a bit of the mask register means check the corresponding bit in the word alignment character register). The mask registers defined by the 10-bit value assigned to attribute COMMA_M. This value applies to all channels in a PCS quad.

When attribute CHx_COMMA_ALIGN is set to ‘AUTO’, one of the protocol based Link State machines will control word alignment. For more information on the operation of the protocol based Link State Machines, see the Protocol Specific Link State Machine description below.

8b10b Decoder
The 8b10b decoder implements an 8b10b decoder operation as described with the IEEE 802.3-2002 specification. The decoder performs the 10-bit to 8-bit code conversion along with verifying the running disparity. The 8b10b decoder can be bypassed on a per channel basis by setting attribute CHx_8B10B to “BYPASS” where x is the channel number.

When a code violation is detected, the ff_rxdata receive data is set to 0xEE with ff_rx_k_cntrl_ch set to ‘1’.

Protocol Specific Link State Machine
The PCS implements link state machines for various protocols that are used in various quad modes.

When a protocol specific Link State Machine is selected, that channel’s Link State Machine must be enabled by setting the protocol CH(0-3)_COMMA_ALIGN to “AUTO”. Selection of the specific Link State Machine that is enabled in each mode is described below and summarized in Figure 8-7.

The Link State Machine for Gigabit Ethernet is selected when attribute PROTOCOL is “GIGE”. Link synchronization is achieved after the successful detection and alignment of the required number of consecutive aligned code words. The Gigabit Ethernet link synchronization state machine implements the Synchronization State Diagram shown in Figure 36-9 of the 802.3- 2002 1000BASE-X specification.

In 'G8B10B' and '10-bit SERDES Only' protocols, the Gigabit Ethernet Link State Machine is used when COMMA_ALIGN is set to 'AUTO'.

External Link State Machine Option

When attribute CHx_COMMA_ALIGN is set to “DYNAMIC”, the protocol specific Link State Machines are bypassed. When ffc_enable_cgalign_ch(0-3) is high, the word aligner will lock alignment and stay locked. It will stop comparing incoming data to the user-defined word alignment characters and will maintain current alignment on the first successful compare to either the COMMA_A or COMMA_B. When ffc_enable_cgalign_ch(0-3) is pulsed low, the word aligner will re-lock on the next match to one of the user-defined word alignment characters. If desired, ffc_enable_cgalign_ch(0-3) can be controlled by a Link State Machine implemented externally to the PCS quad to allow a change in word alignment only under specific conditions.

Figure 8-7 illustrates the link state machine options.

Figure 8-7. PCS Word Aligner and Link State Machine Options

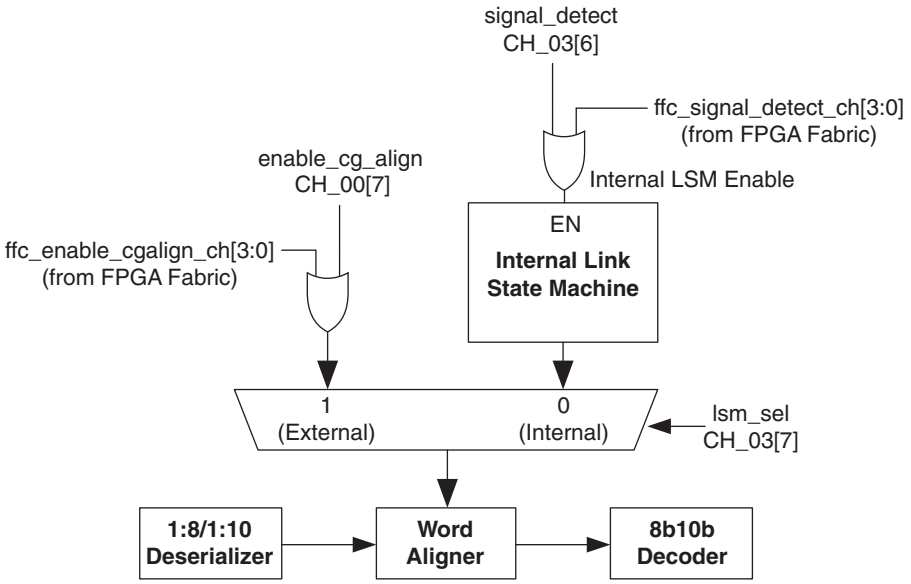


Table 8-6. Link State Machine and Word Aligner Selection

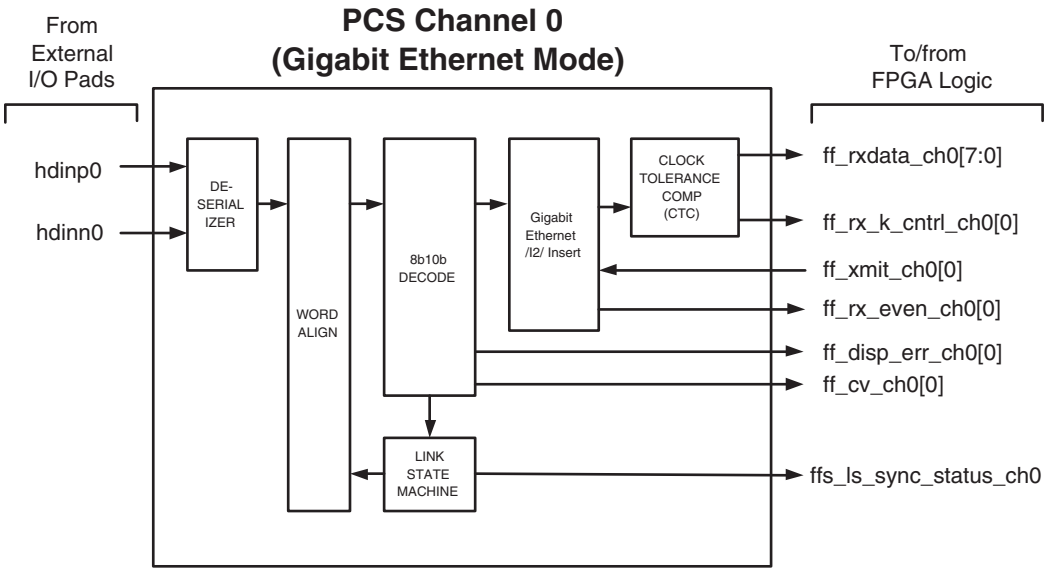
COMMA ALIGN Mode	Description
AUTO	WA enabled, LSM enabled (GbE LSM: default).
DYNAMIC	WA enabled, LSM disabled. The cg_align and sig_detect signals are set to 0 so that the potential external LSM to control both signals. The External Link State Machine option in the Optional Port tab of the GUI must be also selected.
BYPASS	WA bypassed, LSM disabled. Users may develop word aligner in the FPGA core and provide their own cg_align and sig_detect signals to the logic.

When a Link State Machine is selected and enabled, for a particular channel, that channel’s ffs_ls_sync_status_ch(0-3) status signal will go high upon successful link synchronization.

Idle Insert for Gigabit Ethernet Mode

Generic 8b10b mode also has the option to select the Link State Machine for word alignment. The PCS set to Gigabit Ethernet Mode provides for insertion of /I2/ symbols into the receive data stream for auto-negotiation. Gigabit Ethernet auto-negotiation is performed in soft logic. This function inserts a sequence of 8 /I2/ ordered sets every 2048 clock cycles. /I2/ insertion is controlled by the ff_xmit_ch(0-3) input to the PCS which is driven from the auto-negotiation soft logic. The signal ff_rx_even_ch(0-3)[0] from the PCS to the auto-negotiation soft logic is also provided. Figure 8-8 shows one channel (channel 0 in this example) of receive logic when the PCS is set to Gigabit Ethernet Mode showing these control/status signals.

Figure 8-8. PCS Receive path for Gigabit Ethernet Mode (Channel 0 Example)



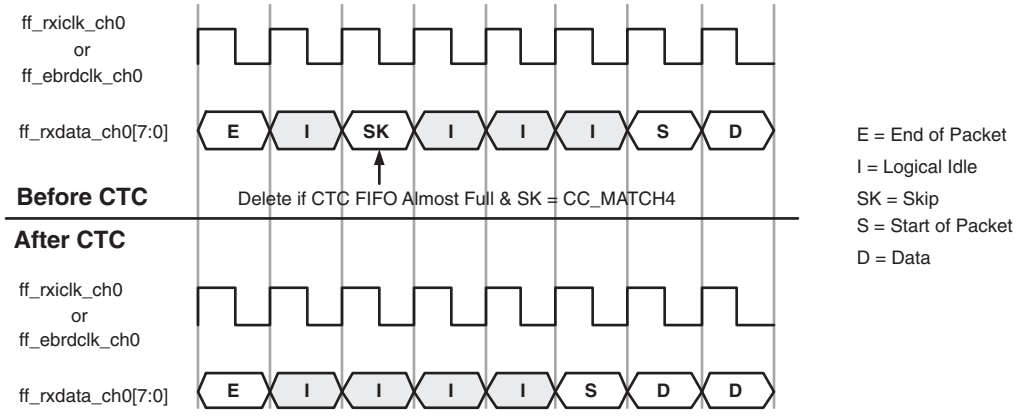
Clock Tolerance Compensation

The Clock Tolerance Compensation module performs clock rate adjustment between the recovered receive clocks and the locked reference clock. Clock compensation is performed by inserting or deleting bytes at pre-defined positions, without causing loss of packet data. A 16-Byte elasticity FIFO is used to transfer data between the two clock domains and will accommodate clock differences of up to the specified ppm tolerance for the LatticeECP2M SERDES (See DC and Switching Characteristics section of the LatticeECP2/M Family Data Sheet).

A channel has the Clock Tolerance Compensation block enable when that channel's attribute CHx_CTC_BYP is set to "NORMAL". The CTC is bypassed when that channel's attribute CHx_CTC_BYP is set to "BYPASS".

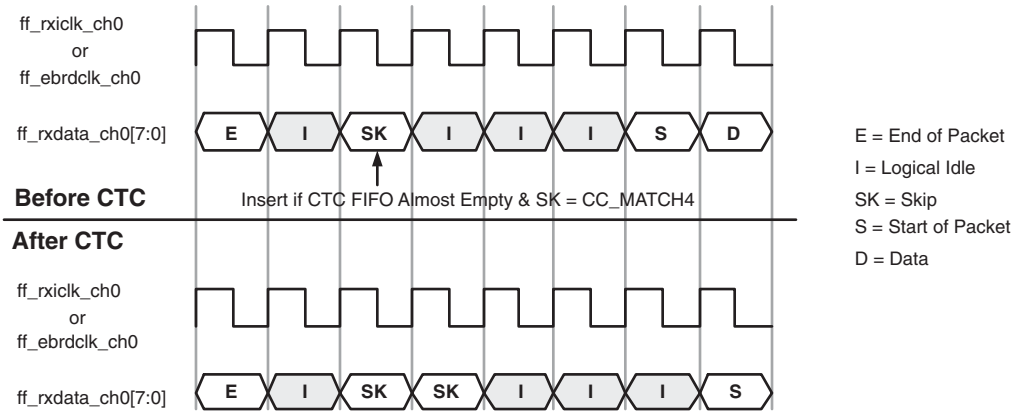
A diagram illustrating 1 byte deletion is shown in Figure 8-9:

Figure 8-9. Clock Tolerance Compensation 1 Byte Deletion Example



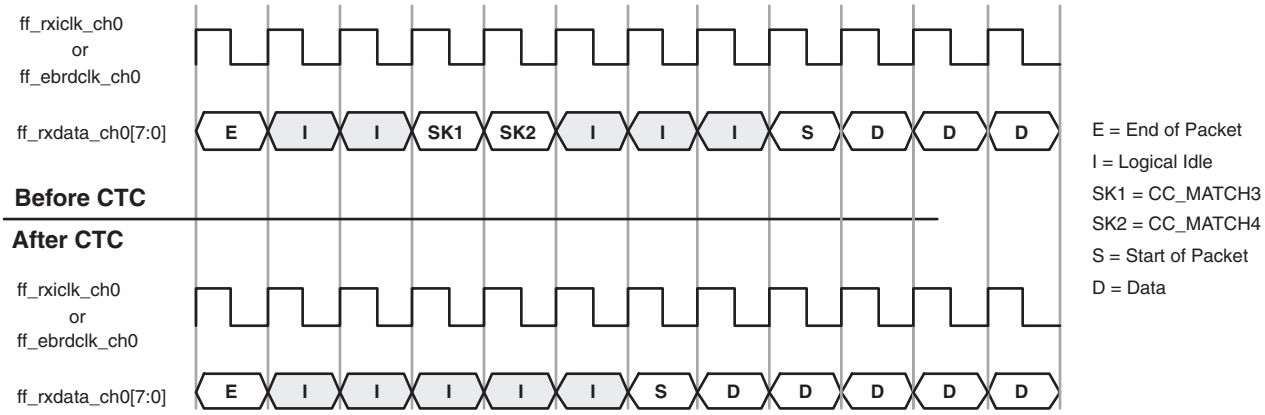
A diagram illustrating 1 byte insertion is shown in Figure 8-10:

Figure 8-10. Clock Tolerance Compensation 1 Byte Insertion Example



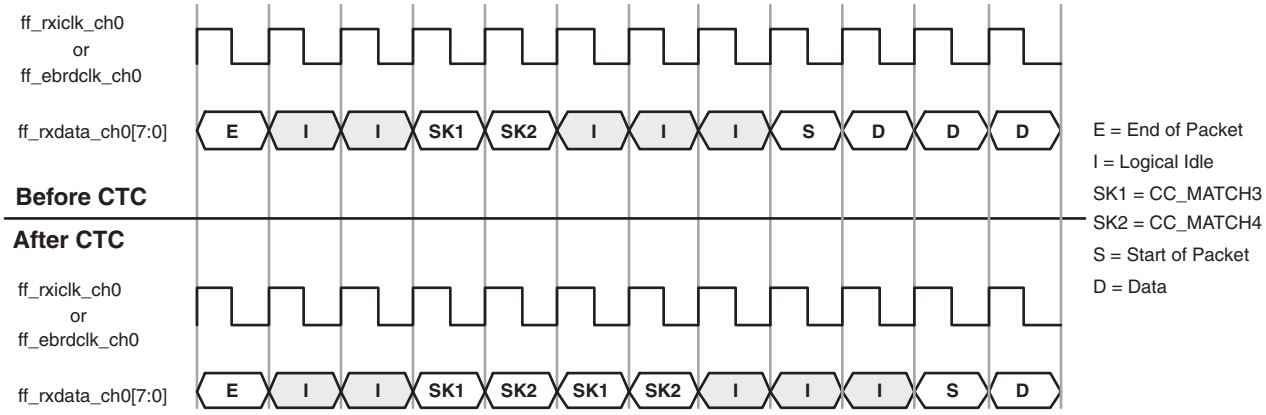
A diagram illustrating 2 byte deletion is shown in Figure 8-11:

Figure 8-11. Clock Tolerance Compensation 2 Byte Deletion Example



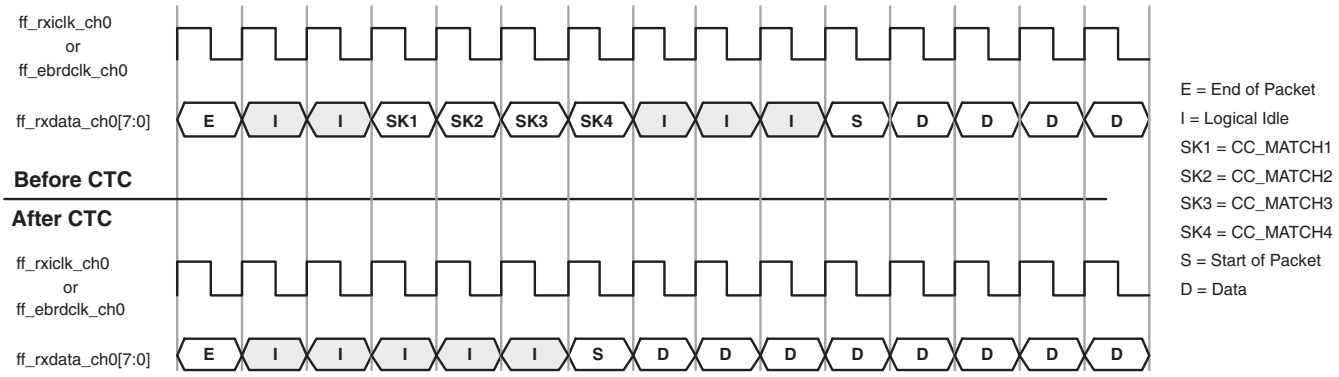
A diagram illustrating 2 byte insertion is shown in Figure 8-12:

Figure 8-12. Clock Tolerance Compensation 2 Byte Insertion Example



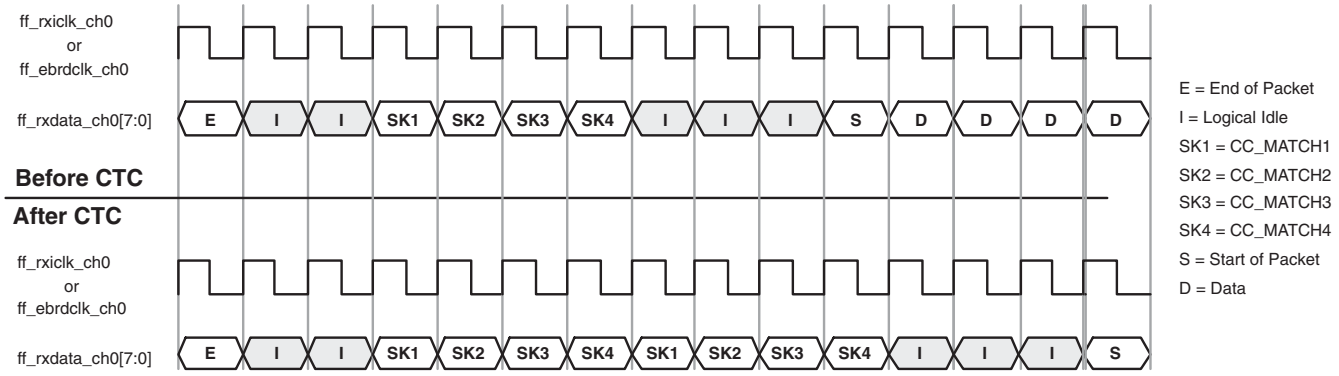
A diagram illustrating 4 byte deletion is shown in Figure 8-13:

Figure 8-13. Clock Tolerance Compensation 4 Byte Deletion Example



A diagram illustrating 4 byte insertion is shown in Figure 8-14:

Figure 8-14. Clock Tolerance Compensation 4 Byte Insertion Example



Clock compensation values are set on a quad basis. The CTC can be bypassed on a per channel basis by setting attribute CHx_CTC_BYP to “BYPASS” where x is the channel number. Setting CHx_CTC_BYP to “NORMAL” means the CTC is active. In the ispLEVER module generator, defining a channel as “Single” creates an auto-configuration file which enables CTC. Defining a channel as “MCA Group1” or “MCA Group 2” creates an auto-configuration file which bypasses CTC. When the CTC is used, the following settings for clock compensation must be set as appropriate for the intended application:

- Set the insertion/deletion pattern length using the CC_MATCHMODE attribute. This sets the number of skip bytes the CTC compares to before performing an insertion or deletion. Values for CC_MATCHMODE are “MATCH_4” (1 byte insertion/deletion), “MATCH_3_4” (2 bytes insertion/deletion), and “MATCH_1_2_3_4” (4 bytes insertion/deletion) to 1. The minimum inter-packet gap must also be set as appropriate for the targeted application. The inter-packet gap is set by assigning values to attribute CC_MIN_IPG. Allowed values for CC_MIN_IPG are “0”, “1”, “2”, and “3”. The minimum allowed inter-packet gap after skip character deletion is performed based on these attribute settings is described in Table 8-7 below.
- The Skip byte or ordered set must be set corresponding to the CC_MATCHMODE chosen. For 4 byte insertion/deletion (CC_MATCHMODE = “MATCH_1_2_3_4”), the first byte must be assigned to attribute MATCH_1, the second byte must be assigned to attribute MATCH_2, the third byte must be assigned to attribute MATCH_3, and the fourth byte must be assigned to attribute MATCH_4. Values assigned are 10 bit binary values. For example, if a 4 byte skip ordered set is /K28.5/D21.4/D21.5/D21.5, then “MATCH_1” should be “0110111100”, “MATCH_2”, = “0010010101”, and “MATCH_3” = “MATCH_4” = “0010110101”. For 2 byte insertion/deletion (CC_MATCHMODE = “MATCH_3_4”), the first byte must be assigned to attribute MATCH_3, and the second byte must be assigned to attribute MATCH_4. For 1 byte insertion/deletion (CC_MATCHMODE = “MATCH_4”), the skip byte must be assigned to attribute MATCH_4.

- The clock compensation FIFO high water and low water marks must be set to appropriate values for the targeted protocol. Values can range from 0 to 15 although the high water mark must be set to a value higher than or equal to the low water mark. The high water mark is set by assigning a value to attribute CCHMARK. Allowed values for CCHMARK are hex values ranging from “0” to “F”. The low water mark is set by assigning a value to attribute CCLMARK. Allowed values for CCLMARK are hex values ranging from “0” to “F”.
- Clock compensation FIFO overrun can be monitored on a per channel basis on the PCS/FPGA interface port labeled ffs_cc_overrun_ch(0-3) if “Error Status Ports” is selected when generating the PCS block with the isp-LEVER module generator.
- Clock compensation FIFO underrun can be monitored on a per channel basis on the PCS/FPGA interface port labeled ffs_cc_underrun_ch(0-3) if “Error Status Ports” is selected when generating the PCS block with the isp-LEVER module generator.

Calculating Minimum Inter-packet Gap

Table 8-7 shows the relationship between the user-defined values for inter-packet gap (defined by the CC_MIN_IPG attribute), and the guaranteed minimum number of bytes between packets after a skip character deletion from the PCS. The table shows the inter-packet gap as a multiplier number. The minimum number of bytes between packets is equal to the number of bytes per insertion/deletion times the multiplier number shown in the table. For example, if the number of bytes per insertion/deletion is 4 (CC_MATCHMODE is set to “MATCH_1_2_3_4”), and the minimum inter-packet gap attribute C_MIN_IPG is set to “2”, then the minimum inter-packet gap is equal to 4 (CC_MATCHMODE = “MATCH_1_2_3_4”) times 3 (Table 8-7 with CC_MIN_IPG = “2”) or 12 bytes. The PCS will not perform a skip character deletion until the minimum number of inter-packet bytes have passed through the CTC.

Table 8-7. Minimum Inter-packet Gap Multiplier

CC_MIN_IPG	Insertion/Deletion Multiplier Factor
“0”	1 X
“1”	2 X
“2”	3 X
“3”	4 X

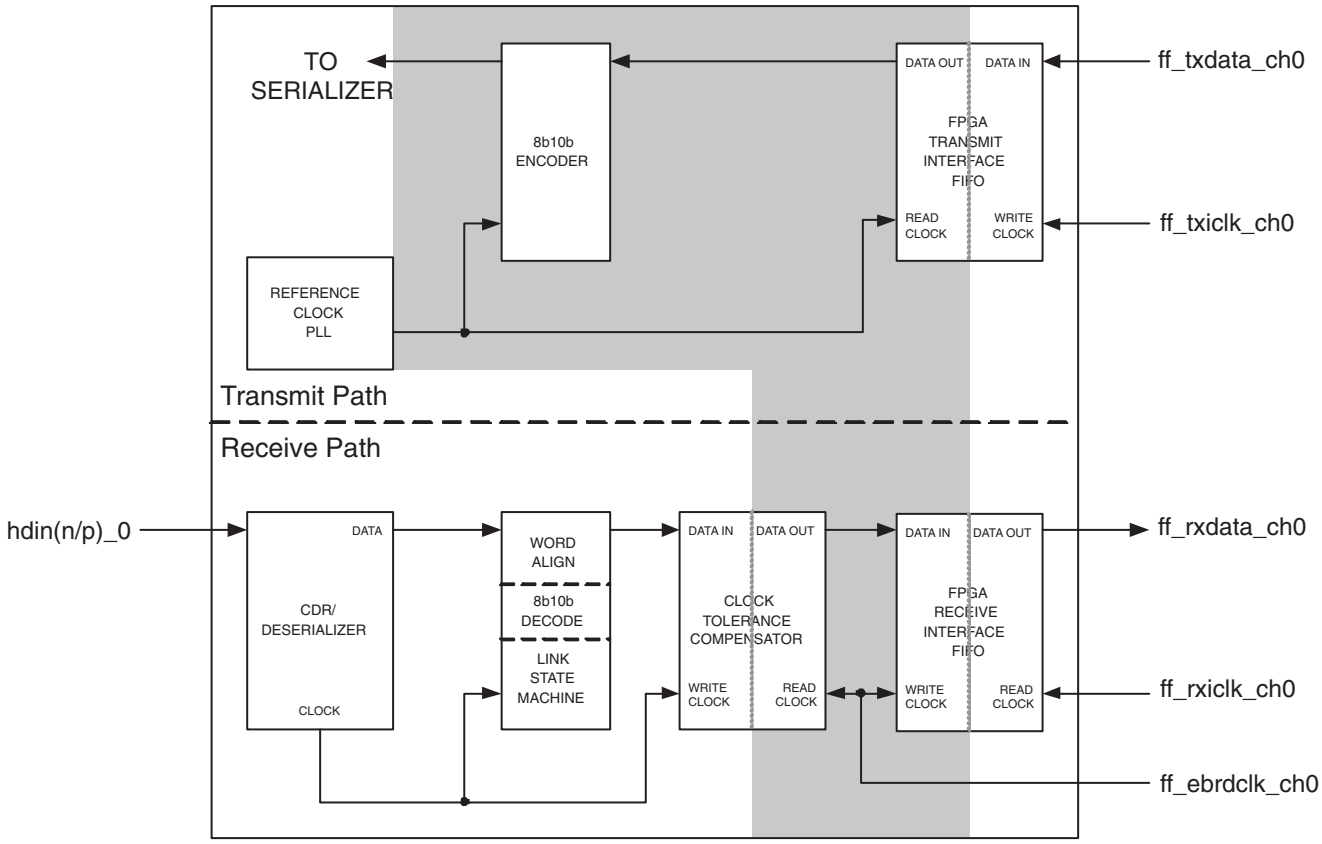
Clock Domains

Figure 8-15 shows the clock domains for both transmit and receive directions for a single channel inside the PCS for modes which utilize the Clock Tolerance Compensation (CTC) block.

On the transmit side, a clock domain transfer from the ff_txiclk_ch input at the FPGA interface to the locked reference clock occurs in the FPGA Transmit Interface FIFO. The FPGA Transmit Interface FIFO is intended to adjust for phase differences between two clocks which are of the same frequency only. These FIFOs (one per channel) cannot compensate for frequency variations.

On the receive side, a clock domain transfer occurs from the channel recovered clocks to the locked reference clock at the Clock Tolerance Compensator (CTC) block. The CTC can adjust for frequency differences between the recovered receive clock and the reference clock up to the maximum phase difference specification for the LatticeECP2M. Downstream from the CTC, another clock interface occurs between the locked reference clock and the ff_rxiclk_ch at the FPGA Receive Interface FIFO. The FPGA Receive Interface FIFO is intended to adjust for phase differences between two clocks which are of the same frequency only. The FPGA Receive Interface FIFOs (one per channel) cannot compensate for frequency variations.

Figure 8-15. PCS Clock Domain Transfers for CTC Modes



To guarantee a synchronous interface, both the input transmit clocks and input receive clock should be driven from one of the output reference clocks for a PCS quad set to Generic 8b10b mode. Figure 8-16 illustrates the possible connections that would result in a synchronous interface.

Figure 8-16. Synchronous Input Clocks to PCS Quad with CTC Used

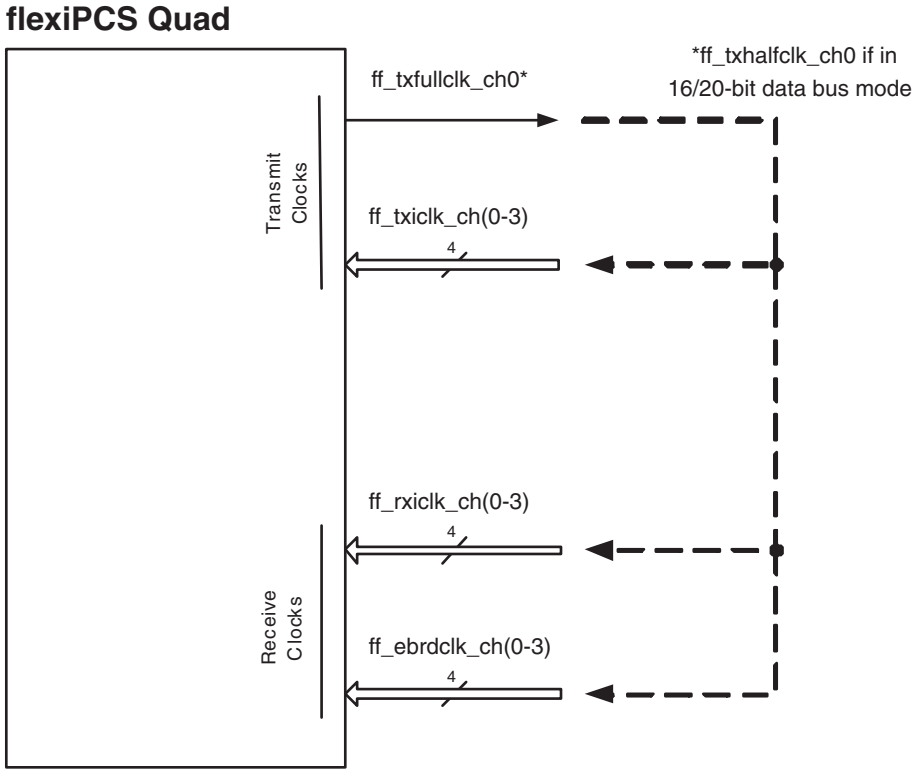
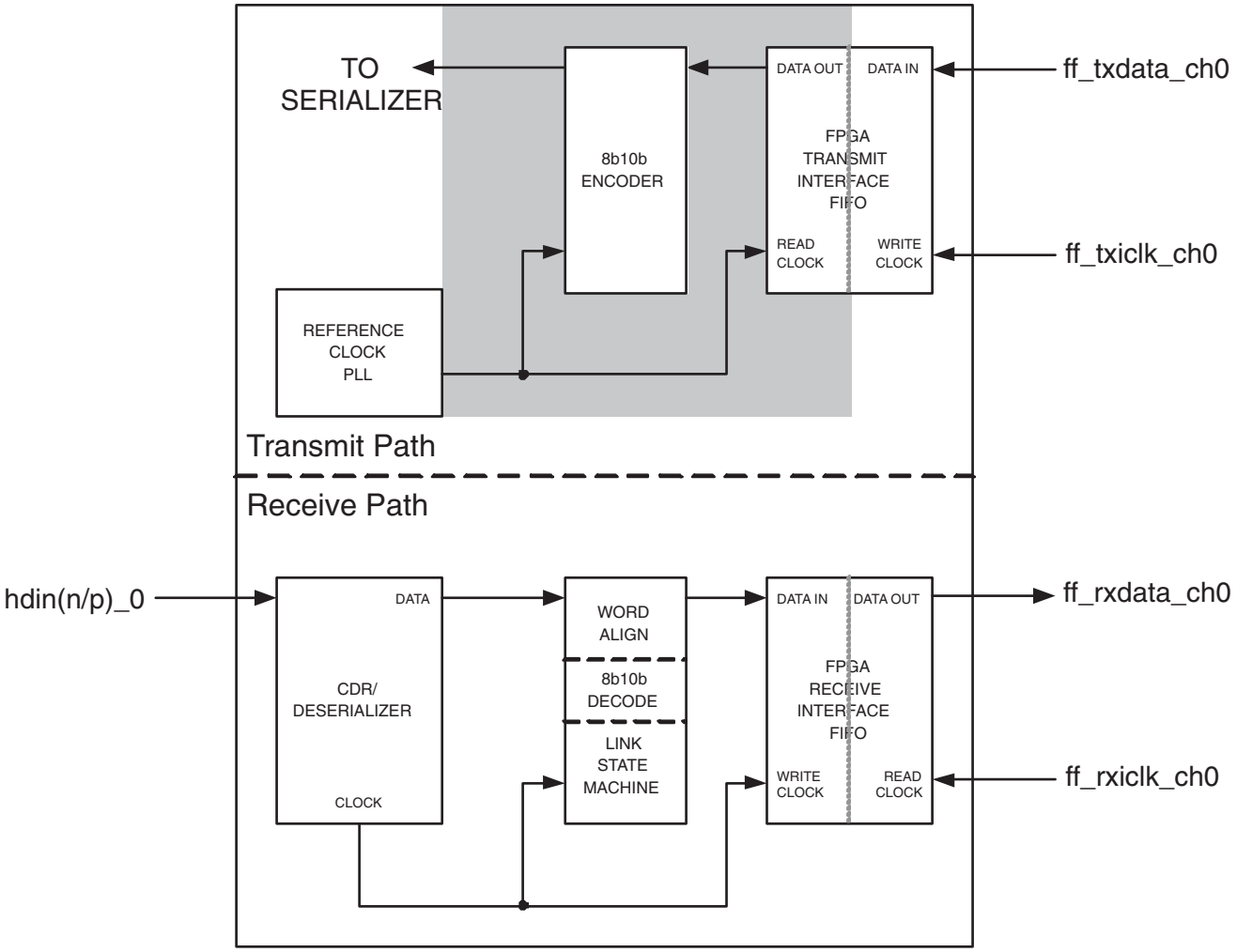


Figure 8-17 shows the clock domains for both transmit and receive directions for a single channel inside the PCS which do not utilize the Clock Tolerance Compensation (CTC) block.

On the transmit side, a clock domain transfer from the ff_txiclk_ch input at the FPGA interface to the locked reference clock occurs in the FPGA Transmit Interface FIFO. The FPGA Transmit Interface FIFO is intended to adjust for phase differences between two clocks which are of the same frequency only. These FIFOs (one per channel) cannot compensate for frequency variations.

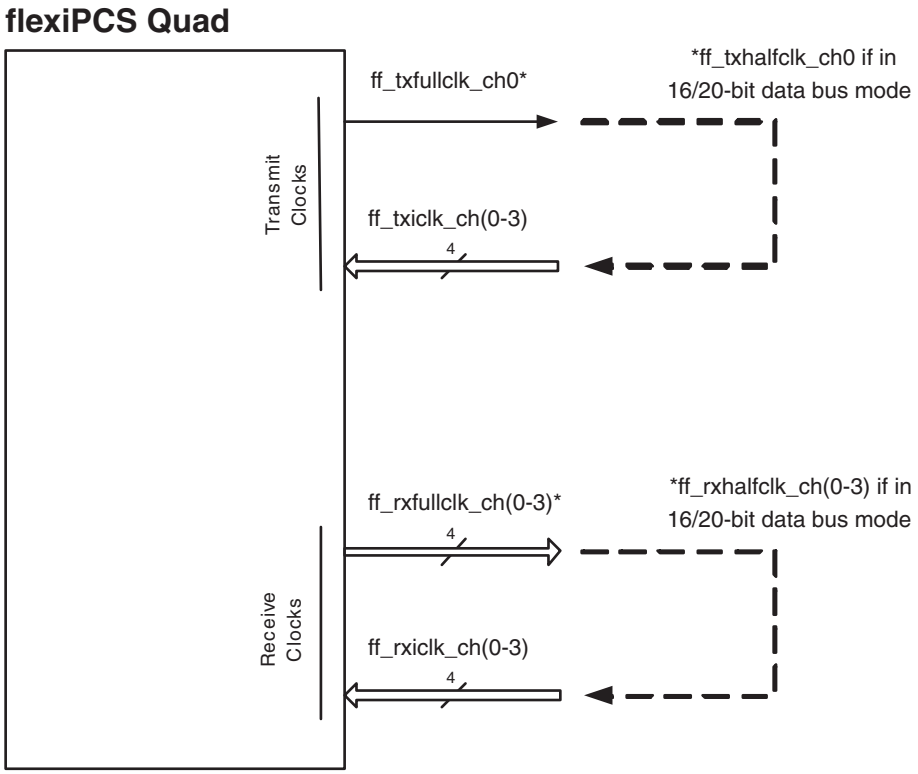
On the receive side, a clock domain transfer occurs between the recovered receive clock and the ff_rxiclk_ch at the FPGA Receive Interface FIFO. The FPGA Receive Interface FIFO is intended to adjust for phase differences between two clocks which are of the same frequency only. The FPGA Receive Interface FIFOs (one per channel) cannot compensate for frequency variations.

Figure 8-17. PCS Clock Domain Transfers for Non-CTC Modes



To guarantee a synchronous interface, both the input transmit clocks and input receive clock should be driven from one of the output reference clocks for a PCS quad set to Generic 8b10b mode. Figure 8-18 illustrates the possible connections that would result in a synchronous interface.

Figure 8-18. Synchronous Input Clocks to PCS Quad without CTC



Spread Spectrum Clocking (SSC) Support

LatticeECP2M SERDES/PCS does not have a Spread Spectrum generator but it can receive Spread Spectrum data.

SSC support in LatticeECP2M applies to all protocols that require similar support. The ports on the two ends of Link must transmit data at a rate that is within 600pm of each other at all times. This is specified to allow bit-rate clock source with a +/- 300ppm tolerance. The data rate can be modulated from +0% to -0.5% of the nominal data rate, at a modulation rate in the range of 30KHz to 33KHz. Along with the +/- 300ppm tolerance limit, both ports require the same bit rate clock when the data is modulated with an SSC.

The root complex is responsible for spreading the reference clock. The endpoint then uses that same clock to pass back the spectrum through the TX. Thus, there is no need for a separate RXREFCLK.

A predominant application of this is the add-in card. Add-in cards are not required to use the REFCLK from the connector but must receive and transmit with the same SSC as the PCI Express connector REFCLK.

Serial Digital Video and Out-Of-Band Low Speed SERDES Operation

The SERDES receiver buffers can be used to input low speed (<250Mbps: Out-Of-Band signal, OOB) by bypassing the receiver CDR and associated SERDES/PCS logic. This feature is useful for applications where the same pin is needed for both high speed and low speed data transfers down to the DC rate, as required by Serial Digital Video applications.

LatticeECP2M SERDES/PCS supports the standard definition serial digital interface, SD-SDI(143Mbps, 177Mbps, 270Mbps, 360Mbps) and the high definition serial digital interface, HD-SDI (1.485Gbps and 1.483.5Gbps).

It is required that both of these share the same receive and transmit pins. One possible implementation is shown in Figure 8-19.

The Out-Of-Band (OOB) signal port at the PCS/FPGA interface is used to input a signal slower than 250 Mbps.

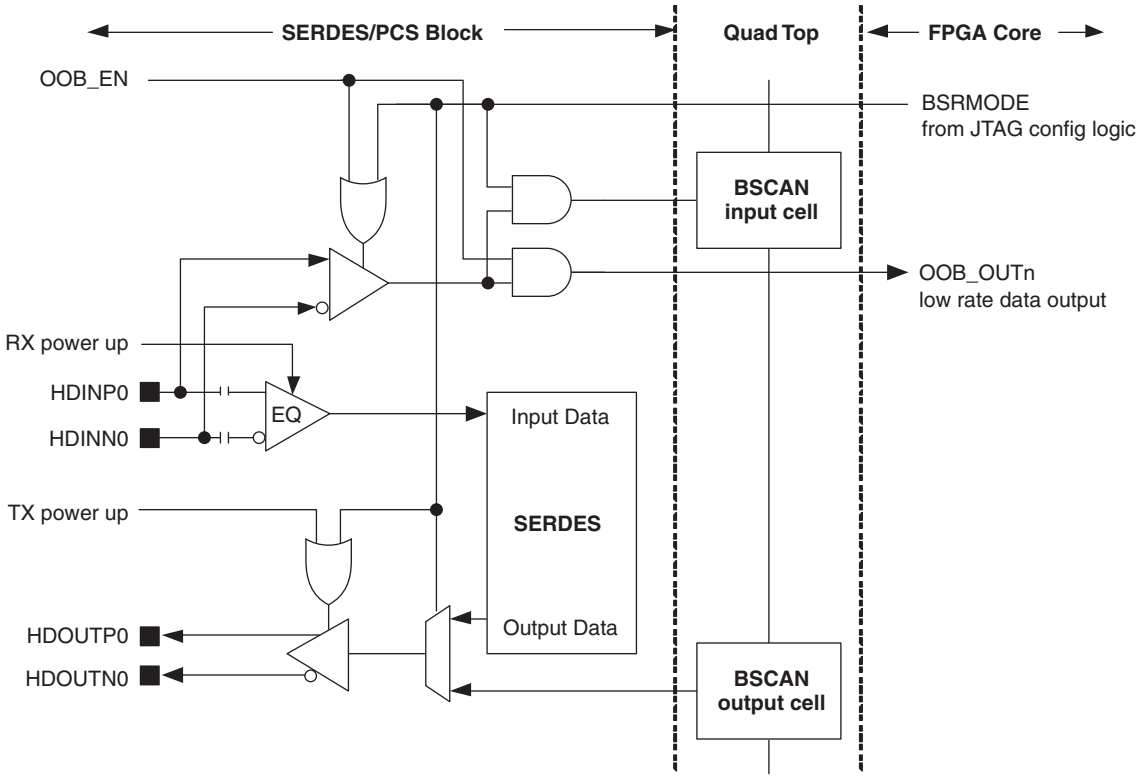
This port is available when the SD-SDI mode is selected. The OOB_OUT signal is passed directly from the SERDES input buffers to the FPGA interface and is not locked to the reference clock (see Figure 8-3).

When driving a SERDES input buffer with a low speed signal, the SERDES input buffer should be set to DC mode, which is done on a per-channel basis by selecting 'DC' in the RX I/O Coupling drop-down box of the IPexpress PCS configuration GUI (see Figure 8-25).

Though the discussion above talks about Serial Digital Video, it should be noted that similar usage is intended for any low bit rate applications that will do RX Clock Data Recovery in the FPGA logic and require decimation in the TX direction.

The input BSCAN circuit appears in parallel with the high-speed SERDES and can be used to route input data to a lower-speed Deserializer located elsewhere in the device (not in the quad). An enable signal (one per channel) is required to turn on the input BSCAN circuit independent of the BSCAN state machine.

Figure 8-19. One Possible Implementation of Serial Digital Video Support



Configuration GUIs

IPexpress is used to create and configure SERDES and PCS blocks. Designers use the graphical user interface (GUI) to select the SERDES Protocol Standard for a particular quad or channel. IPexpress takes the input from this GUI and generates a configuration file (.txt file) and HDL netlist. The HDL model is used in the simulation and synthesis flow. The configuration file contains attribute level map information. This file is input for simulation and the ispLEVER bitgen program. It is strongly recommended that designers make changes and updates in IPexpress and then regenerate the configuration file. In some exceptional occasions, users can modify the configuration file. Figure 8-20 shows the tools flow when using IPexpress to generate the SERDES/PCS block for the SERDES Protocol Standard.

When a project is saved in a different directory, this configuration file (.txt) should be manually moved to the same directory as the project file.

Figure 8-20. SERDES_PCS ispLEVER User Flow

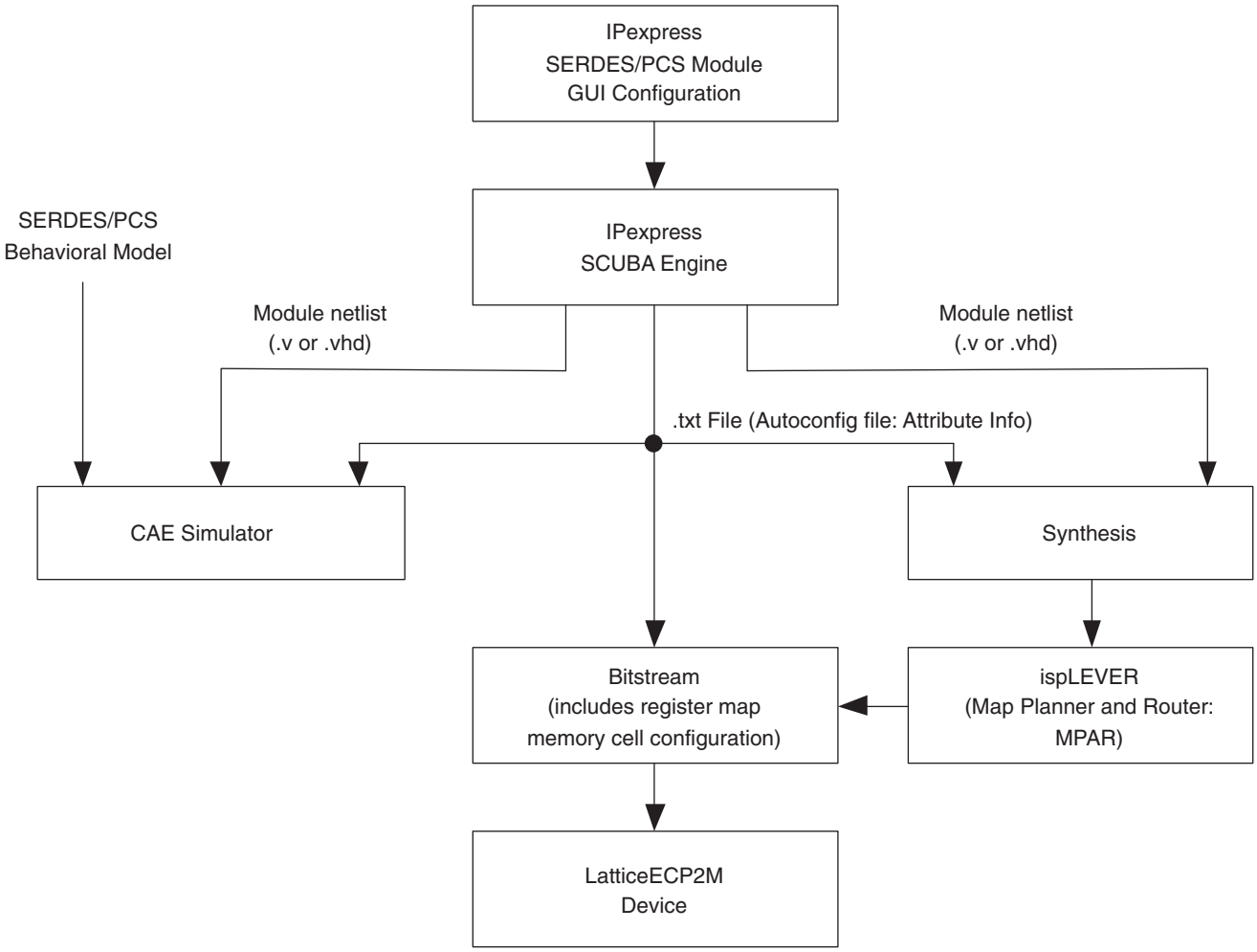
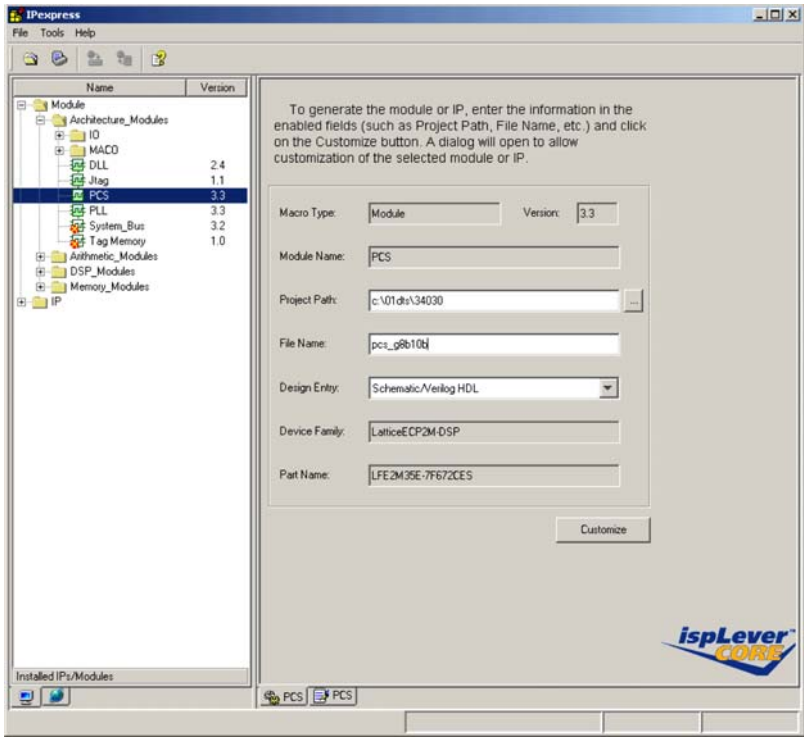


Figure 8-21 shows the main window when PCS is selected in the IPexpress GUI.

Figure 8-21. IPexpress PCS Main Window



Quad Setup Tab

Figure 8-22 shows the Quad Setup Tab window when the file name is entered and the **Customize** button is checked in the main window. The first entry required in this window is to select Protocol Mode from Quad Based Mode or Channel Based Mode. Other entries on this screen are channel selection and group selections. There are five additional tabs shown in Figures 8-23 through 8-28, listing all the user-accessible attributes with default values settings:

Figure 8-22. Configuration GUI - Quad Setup Tab

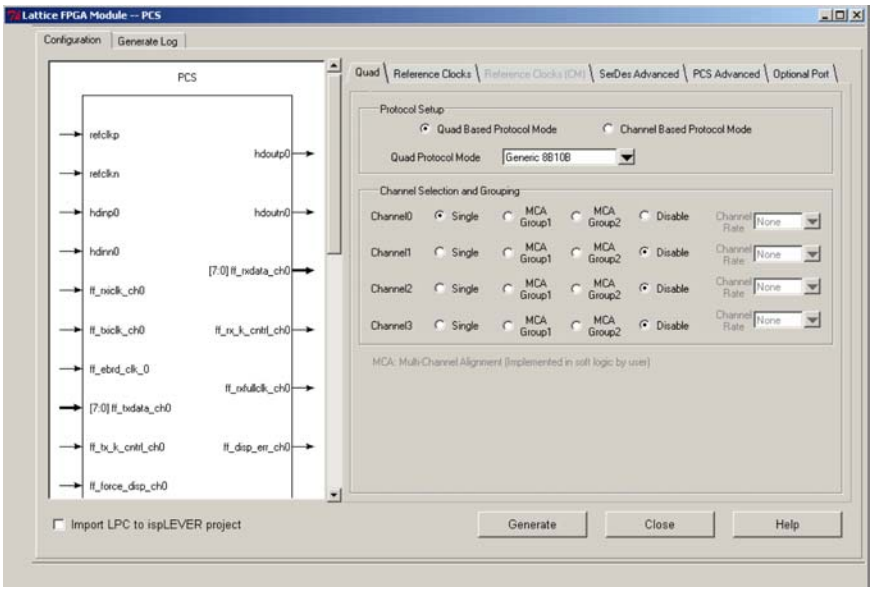


Table 8-8. SERDES_PCS GUI Attributes - Quad Tab Setup

GUI Text	Attribute Names	Range	Default Value
Protocol Setup		Quad Based Protocol Mode, Channel Based Protocol Mode	Quad Based Protocol Mode
Quad Protocol Mode	PROTOCOL	PCI Express, Gigabit Ethernet, Generic 8b10b, 10-bit SERDES Only, 8-bit SERDES Only, SD-SDI, HD-SDI[2]	Generic 8b10b
Single	CH_MODE	Enable the channel	Disable
MCA Group1[3]	CH_MODE	Multi-Channel Alignment Group 1	Disable
MCA Group2[3]	CH_MODE	Multi-Channel Alignment Group 2	Disable
Disable	CH_MODE	Disable the channel	Disable
Channel Rate[1]	CH_MODE	Full Rate, Half Rate	Full Rate

1. Channel Rate selection is applicable only in the Channel Based Protocol Mode.
2. Protocol Attribute Names: PCI Express = PCIE, Gigabit Ethernet = GIGE, Generic 8b10b = G8B10B, 8-bit SERDES Only = 8BSER, 10-bit SERDES Only = 10BSER, SD-SDI = SDSDI, HD-SDI = HDSDI.
3. Multi-Channel-Alignment is for transmitter lane-to-lane skew alignment. Receiver Multi-Channel-Alignment is not provided in hard PCS. MCA Group1 and MCA Group2 set the channels to CTC Bypass mode so that users can build the Multi-Channel Alignment in the FPGA core. The two groups are provided for identification of channels in Multi-Protocol applications.

Reference Clock Setup Tab

In this tab, the attributes of the Tx and Rx reference clock sources are selected. Users can select either a REFCLK or CORE_RXREFCLK as a Rx reference clock source and CORE_TXREFCLK as a Tx reference clock source for the quad. Similarly, in a quad all the channels use the same common Tx and Rx reference sources. Further, there is a tool to provide the required clock rate and multiplier settings for a particular data rate. In addition, for a given data bus width the tool provides the required clock rate to interface the quad to the core.

Figure 8-23. Configuration GUI - Reference Clocks Setup Tab

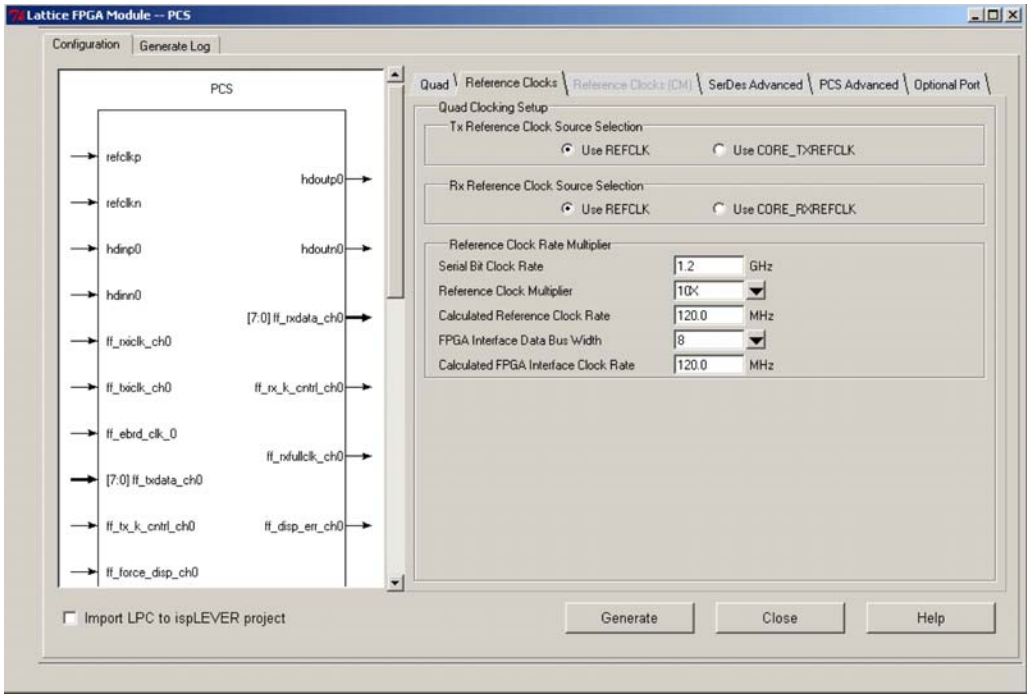


Table 8-9. SERDES_PCS GUI Attributes (LatticeECP2M) - Reference Clocks Setup Tab

GUI Text	Attribute Names	Button/ Box Type	Range	Default Value	Comment
Tx Reference Clock Source Selection	PLL_SRC	Radio	REFCLK, CORE_TXREFCLK	REFCLK	
Rx Reference Clock Source Selection	CH0_CDR_SRC CH1_CDR_SRC CH2_CDR_SRC CH3_CDR_SRC	Radio	REFCLK, CORE_RXREFCLK	REFCLK	
Serial Bit Clock Rate (GHz)	DATARANGE[1]	Check Box	0.27 to 3.125	2.5	LOW: MEDLOW: MED: MEDHIGH: HIGH:
Reference Clock Multiplier	CH0_REFCK_MULT CH1_REFCK_MULT CH2_REFCK_MULT CH3_REFCK_MULT	Drop Down	See Table 8-10	25X	
Calculated Reference Clock Rate (MHz)[2]	CP: REFCLK_RATE	Text Box	—		Not an editable field
FPGA Interface Data Bus Width	CH0_DATA_WIDTH CH1_DATA_WIDTH CH2_DATA_WIDTH CH3_DATA_WIDTH	Drop Down	See Table 8-10	8	
Calculated FPGA Interface Clock Rate (MHz)[2]	CP: FPGAINTCLK_RATE	Text Box	—		Not an editable field

1. DATARANGE:
   For Production Devices: Low ≤ 500 Mbps, 500 Mbps < Medlow ≤ 1.0 Gbps, 1.0 Gbps < Med < 2.0 Gbps, 2.0 Gbps ≤ Medhigh < 2.5 Gbps, 2.5 Gbps ≤ High ≤ 3.2 Gbps
   For Engineering Samples: Low ≤ 540 Mbps, 540 Mbps < Medlow ≤ 1.0 Gbps, 1.0 Gbps < Med < 2.0 Gbps, 2.0 Gbps ≤ Medhigh < 2.5 Gbps, 2.5 Gbps ≤ High ≤ 3.2 Gbps
   Refer to Table 8-99 for details on the control bits setting.
2. 8-bit SERDES Only Mode and 10-bit SERDES Only Mode bypass the Link Align/Comma Align, 8b10b encoder/decoder and the CTC. It does not bypass the CDR.

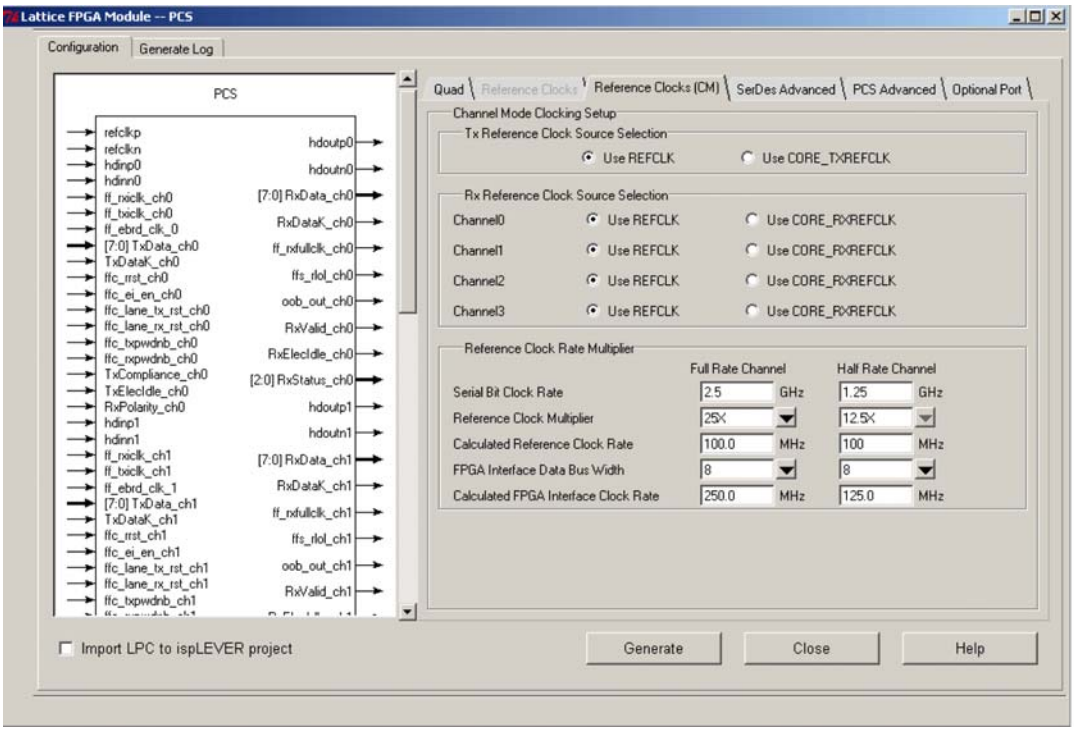
Table 8-10. Reference Clock Multiplier and FPGA Interface Data Bus Width by Protocol

Protocol	Reference Clock Multiplier	FPGA Interface Data Bus Width
PCI Express	20X, 25X	8, 16
GbE	10XH, 10X, 20X	8, 16
G8B10B	10XH, 10X, 20X	8, 16
10-bit SERDES Only	10XH, 10X, 20X	10, 20
8-bit SERDES Only	8HX, 8X, 16X	8, 16

Reference Clock Setup Tab (Channel Mode)

In this tab, the Tx reference clock selected is common to all channels but Rx reference clocks can be either REFCLK or CORE_RXREFCLK by channel.

Figure 8-24. Configuration GUI - Reference Clocks Setup Tab (Channel Mode)



When a user selects **Channel Based Protocol Mode** in the **Quad** tab and sets a channel or channels as half rate mode, this tab displays the half rate mode clock data.

SERDES Advance Setup

This tab is used to access the advanced attributes of the transmit and receive SERDES for all four channels. Transmit attributes such as PreEmphasis, termination, differential output voltage selection are selected. Receive attributes such as equalization, termination, I/O coupling are selected. Attributes for Transmit SERDES clock and PLL are also selected.

Figure 8-25. Configuration GUI - SERDES Advanced Setup Tab

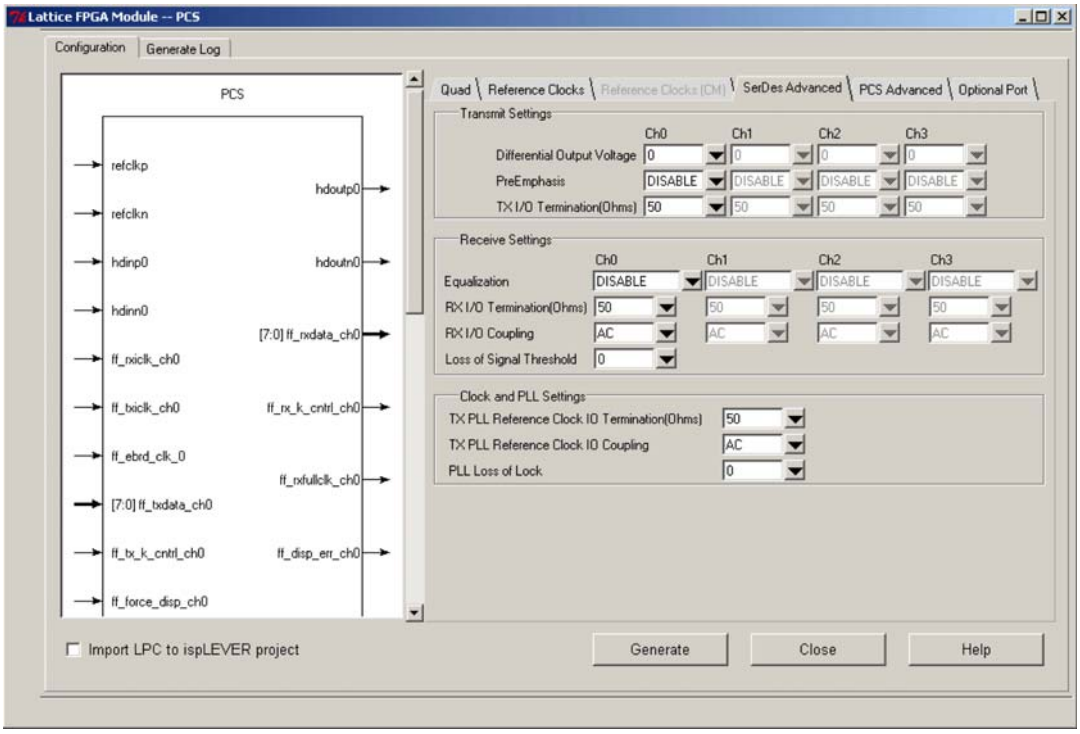


Table 8-11. SERDES_PCS GUI Attributes (LatticeECP2M) - SERDES Advanced Setup Tab

GUI Text		Attribute Names	Button/ Box Type	Range		Default Value
				PCI Express, GIGE	G8B10B, 8bSER 10bSER	
Differential Output Voltage	LatticeECP2M-35	CH0_TDRV_AMP CH1_TDRV_AMP CH2_TDRV_AMP CH3_TDRV_AMP	Drop Down	0(1040mV: default), 1(1280mV), 2 (1320mV), 3 (1360mV), 4 (640mV), 5 (760mV), 6 (870mV), 7 (990mV)		0
	All other devices			0(990mV: default), 1(1250mV), 2 (1300mV), 3 (1350mV), 4 (610mV), 5 (730mV), 6 (820mV), 7 (940mV)		
PreEmphasis	LatticeECP2M-35	CH0_TX_PRE CH1_TX_PRE CH2_TX_PRE CH3_TX_PRE	Drop Down	Disable, 0 (0%), 1 (16%), 2 (36%), 3 (40%), 4 (44%), 5 (56%), 6 (80%)		DISABLE
	All other devices			Disable, 0 (0%), 1 (12%), 2 (26%), 3 (30%), 4 (33%), 5 (40%), 6 (53%)		
Tx I/O Termination (Ohms)[3]		CH0_RTERM_TX CH1_RTERM_TX CH2_RTERM_TX CH3_RTERM_TX	Drop Down	50, 75, 5K		50
Equalization[1]		CH0_RX_EQ CH1_RX_EQ CH2_RX_EQ CH3_RX_EQ	Drop Down	Mid_High, Long_High Disable	Mid_Low, Mid_Med Mid_High, Long_Low Long_Med, Long_High Disable	DISABLE
Rx I/O Termination (Ohms)[3]		CH0_RTERM_RX CH1_RTERM_RX CH2_RTERM_RX CH3_RTERM_RX	Drop Down	50, 60, 75, High		50
Rx I/O Coupling		CH0_RX_DCC CH1_RX_DCC CH2_RX_DCC CH3_RX_DCC	Drop Down	AC, DC		AC[2]
Loss of Signal Threshold		LOS_THRESHOLD	Drop Down	0 (default), 1 (+10%), 2 (+15%),3 (+25%) 4 (-10%), 5 (-15%), 6 (-25%), 7 (-30%)		0
Tx PLL Reference Clock I/O Termination (Ohms)[3]		PLL_TERM	Drop Down	50, 2K		50
Tx PLL Reference Clock I/O Coupling		PLL_DCC	Drop Down	AC, DC		AC
PLL Loss of Lock		PLL_LOL_SET	Drop Down	Lock	Unlock	0
				0 (+/-600ppmx) 1 (+/-300ppm) 2 (+/-1500ppm) 3 (+/-4000ppm)	0 (+/-1200ppm) 1 (+/-2000ppm) 2 (+/-2200ppm) 3 (+/-6000ppm)	

1. Refer to Appendix D for details.
2. The typical capacitor value of the internal on-chip AC coupling is 5 pF.
3. Termination resistors and their usage

RX I/O Termination:

- 50: So far all of the protocols except SMTPE use 50 Ohms termination resistor.
- 60: Provided for flexibility purpose only.
- 75: SMPTE uses 75 Ohm termination resistor.
- HIGH: Such as PCI Express Rx detection.

TX I/O Termination:

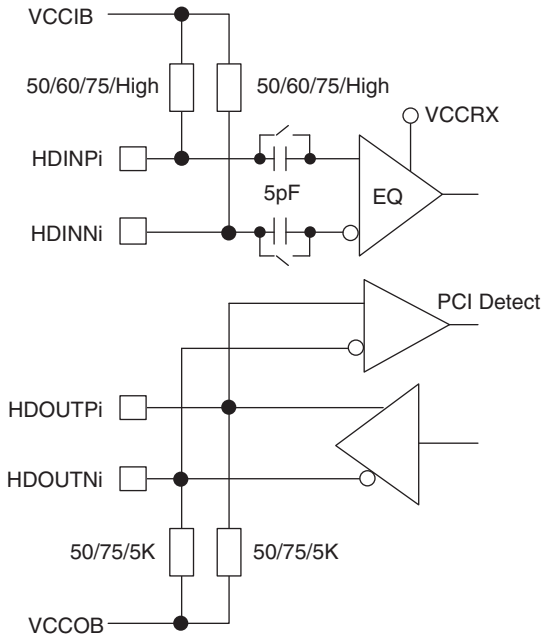
- 50: So far all of the protocols except SMTPE use 50 Ohms termination resistor.
- 75: SMPTE uses 75 Ohm termination resistor.
- 5K: Such as PCI Express electric idle and PCI Express Rx detection.

TX PLL Termination:

- 50: If there is no 50 Ohm termination resistor on PCB board
- 2K: If there is 50 Ohm termination resistor on PCB board

High speed I/O termination topology is shown in Figure 8-26.

Figure 8-26. High-Speed I/O Terminations



PCS Advanced Setup

This tab is used to access the advanced attributes of the transmit and receive PCS for all four channels. Polarity of each individual Tx and Rx channel can be individually selected. The operating mode (e.g. 8b10b) of the individual channels can be selected. In addition, word alignment values such as comma values, comma mask and comma align can be selected. This tab is also used for setting values for the Clock Tolerance Compensation block.

Figure 8-27. Configuration GUI - PCS Advanced Setup Tab

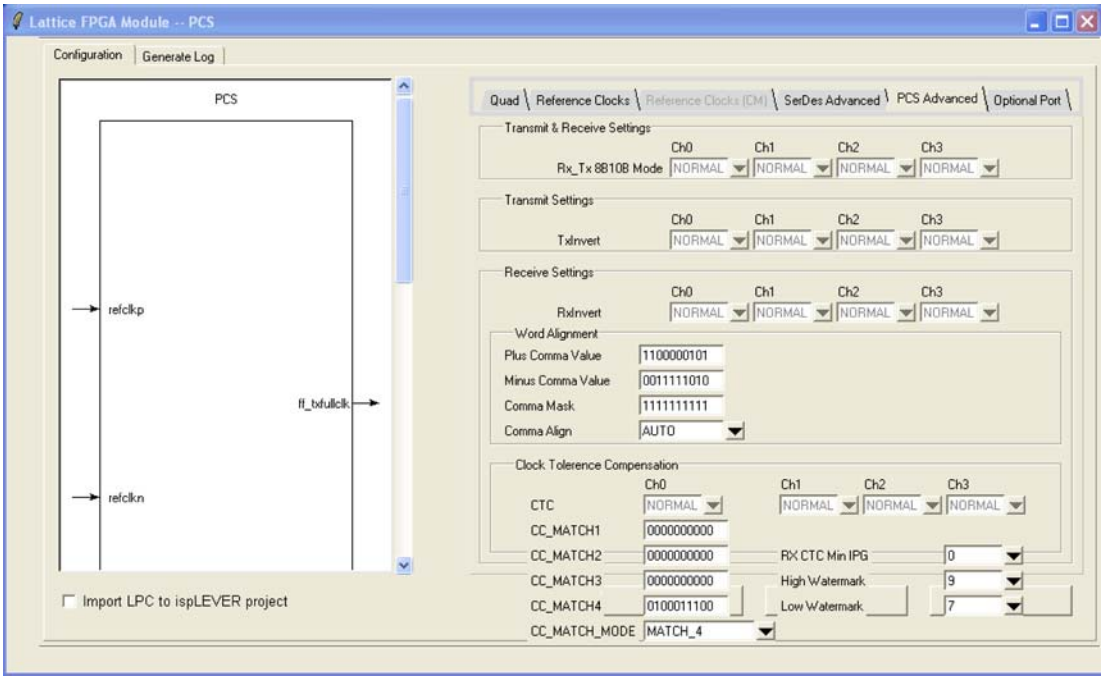


Table 8-12. SERDES/PCS GUI - PCS Advanced Setup Tab

GUI Text	Attribute Name	8-Bit SERDES Only	10-Bit SERDES Only	G8B10B	PCI Express	GIGE	SD-SDI	HD-HDI	Default
TX Invert	CHx_TX_SB	Normal, Invert							Normal
RX Invert	CHx_RX_SB								Normal
RX_TX 8b10b Mode	CHx_8B10B	Bypass		Normal			Bypass		Normal
Plus Comma Value	COMMA_A[4]	N/A	Note 1				N/A		1100000101
Minus Comma Value	COMMA_B[4]								00111111010
Comma Mask	COMMA_M								1111111111
Comma Align[2]	CHx_COMMA_ALIGN	Bypass	Auto, Dynamic, Bypass	Auto, Dynamic	Auto		Bypass		Auto
CTC[3]	CHx_CTC_BYP	Bypass[5]			Normal		Bypass		
CC_MATCH1	CC_MATCH1	N/A	Note 3				N/A		0000000000
CC_MATCH2	CC_MATCH2								0000000000
CC_MATCH3	CC_MATCH3								0100011100
CC_MATCH4	CC_MATCH4								0100011100
CC_MATCH_MODE	CC_MATCH_MODE		MATCH_3_4		MATCH_4				MATCH_4
RX CTC Min IFG	RX CTC Min IFG		0, 1, 2, 3						0
High Watermark	CCHMARK		0, 1, 2, ..., 14, 15						9
Low Watermark	CCLMARK								7

1. Refer to the Word Alignment section of this document for detailed information.
2. Refer to Table 8-6.
3. Refer to the Clock Tolerance Compensation section of this document for detailed information.
4. By definition, COMMA_A and COMM_B are one set of 8b10b encoded control character with positive and negative running disparities. For example, BC(K28.5) and FD(K29.7) cannot be used as the COMMA_A and COMM_B in a single design. The usage of COMMA_A and COMM_B in 8bit mode and 16bit mode is exactly the same in both modes. Users need to follow the rule in order to get the ls_sync. For example, 1 GbE needs K28.5+D5.6 or D16.2 as IDLE (word alignment and sync state machine).
5. GIGE or PCI Express modes may be used if CTC Normal mode is preferred. Note that the GIGE and PCI Express modes are tuned for their specific data rates.

Optional Setup

This tab allows is used to select the dynamic logic inversion and dynamic external link state machine capability per channel. In addition, users can enable the SCI, error reporting, PLL quarter clock, and loop-back capability.

Figure 8-28. Configuration GUI - Optional Setup Tab

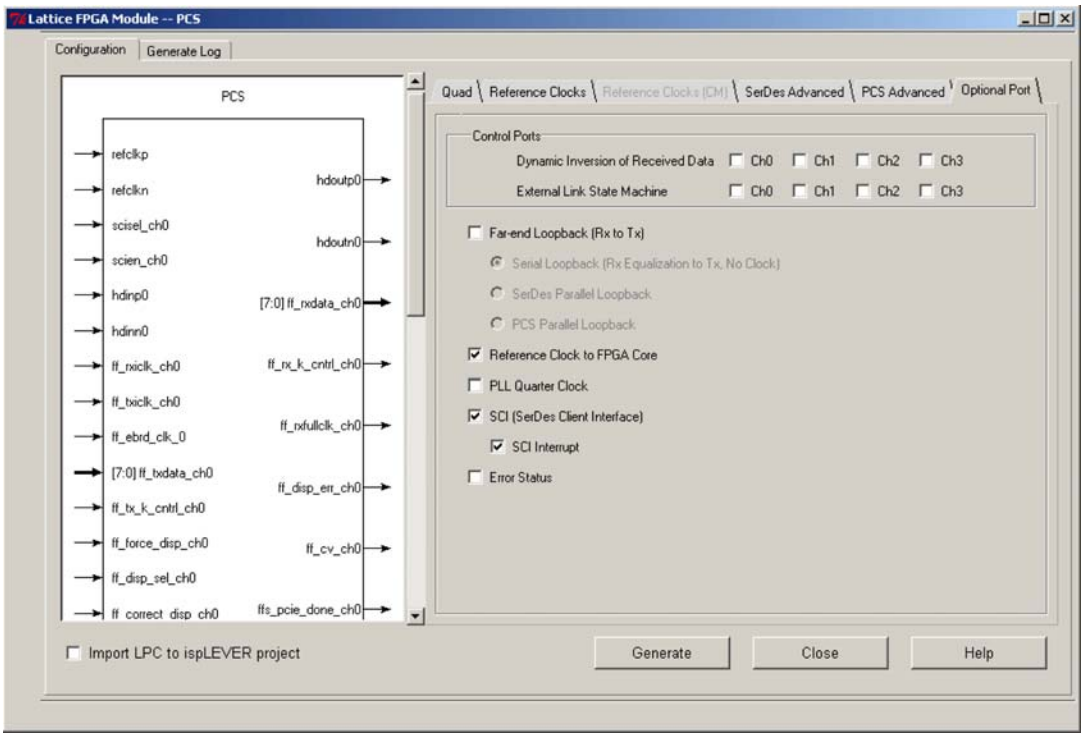


Table 8-13. Tab5. SERDES_PCS GUI Attributes (LatticeECP2M) - Optional Setup Tab

GUI Text	Attribute Names	Button/ Box Type	Range	Default Value
Dynamic Inversion of Receive Data		Check Box	TRUE, FALSE	FALSE
External Link State Machine[1]		Check Box	TRUE, FALSE	FALSE
Loopback (Rx to Tx)		Check Box	TRUE, FALSE	FALSE
Loopback Type	OS_SSLB OS_SPLBPORTS OS_PCSLBPORTS	Radio Box	Serial Loopback, SERDES Parallel Loopback, PCS Parallel Loopback	Serial Loopback
Reference Clock to FPGA Core	OS_REFCK2CORE	Check Box	TRUE, FALSE	FALSE
PLL Quarter Clock	OS_PLLQCLKPORTS	Check Box	TRUE, FALSE	FALSE
SCI		Check Box	TRUE, FALSE	FALSE
SCI Interrupt	OS_INT_ALL	Check Box	TRUE, FALSE	FALSE
Error Status		Check Box	TRUE, FALSE	FALSE

1. When Dynamic mode is selected in the Comma Align option, the External State Machine must be also selected.

Configuration File Description

IPexpress generates this file which contains attribute level map information. This file is input for simulation, synthesis and the ispLEVER bitgen program. It is strongly recommended that designers make changes in the IPexpress and then regenerate the configuration file. In some exceptional occasions, users can modify the Configuration File. The configuration file uses “txt” as the file type extension.

Below is an example of the configuration file.

```
# This file is used by the simulation model as well as the ispLEVER bitstream
# generation process to automatically initialize the PCSC quad to the mode
# selected in the IPexpress. This file is expected to be modified by the
# end user to adjust the PCSC quad to the final design requirements.

DEVICE_NAME "LFE2M35E"
PROTOCOL    "G8B10B"
CH0_MODE    "SINGLE"
CH1_MODE    "DISABLE"
CH2_MODE    "DISABLE"
CH3_MODE    "DISABLE"
PLL_SRC     "REFCLK"
DATARANGE     "HIGH"
CH0_CDR_SRC     "REFCLK"
CH0_DATA_WIDTH     "8"
CH0_REFCK_MULT     "10X"
#REFCLK_RATE     250.0
#FPGAINTCLK_RATE     250.0
CH0_TDRV_AMP     "0"
CH0_TX_PRE     "DISABLE"
CH0_RTERM_TX     "50"
CH0_RX_EQ     "DISABLE"
CH0_RTERM_RX     "50"
CH0_RX_DCC     "AC"
LOS_THRESHOLD     "0"
PLL_TERM     "50"
PLL_DCC     "AC"
PLL_LOL_SET     "0"
CH0_TX_SB     "NORMAL"
CH0_RX_SB     "NORMAL"
CH0_8B10B     "NORMAL"
COMMA_A     "1100000101"
COMMA_B     "0011111010"
COMMA_M     "1111111111"
CH0_COMMA_ALIGN     "AUTO"
CH0_CTC_BYP     "BYPASS"
CC_MATCH1     "0000000000"
CC_MATCH2     "0000000000"
CC_MATCH3     "0100011100"
CC_MATCH4     "0100011100"
CC_MATCH_MODE     "MATCH_4"
CC_MIN_IPG     "0"
CCHMARK     "4"
CCLMARK     "4"
OS_REFCK2CORE     "0"
OS_PLLQCLKPORTS     "0"
```

LatticeECP2M PCS in Gigabit Ethernet Mode

Gigabit Ethernet (1000BASE-X) Idle Insert

Idle pattern insertion is required for Clock Compensation and Auto Negotiation. Auto Negotiation is done in FPGA logic. This module automatically inserts /I2/ symbols into the receive data stream during auto-negotiation. While auto-negotiating, the link partner will continuously transmit /C1/ and /C2/ ordered sets. The clock-compensator will not delete these ordered sets as it is configured to only insert/delete /I2/ ordered sets. In order to prevent overruns and underruns in the clock-compensator, /I2/ ordered sets must be periodically inserted to provide insertion/deletion opportunities for the clock compensator.

While performing auto-negotiation, this module will insert a sequence of 8 /I2/ ordered sets (2 bytes each) every 2048 clock cycles. As this module is after the 8b10b decoder, this operation will not introduce any running disparity errors. These /I2/ ordered sets will not be passed on to the FPGA receive interface as the GMII interface is driven to IDLE by the Rx state machine during auto-negotiation. Once auto-negotiation is complete, /I2/ insertion is disabled to prevent any corruption of the received data.

Note that this state machine is active only during auto-negotiation. The auto-negotiation state machine and the GbE receive state machines are implemented in the soft logic. This state machine depends on the signal ff_xmit_ch[3:0] from the auto-negotiation state machine. This signal is provided on the TX data bus. Even though this signal is relatively static (especially after auto-negotiation) it is included in the TX data bus. It provides the signal rx_even_ch[3:0]. This is sent out on the receive data bus to the FPGA logic.

Gigabit Ethernet Idle Insert and ff_correct_disp_ch[3:0] Signal Usage

The ff_correct_disp_ch[3:0] signal is used on the transmit side of the QuadPCS to ensure that an inter-packet gap begins in the negative disparity state. Note that at the end of an Ethernet frame, the current disparity state of the transmitter can be either positive or negative, depending on the size and data content of the Ethernet frame. However, from the FPGA soft-logic side of the QuadPCS, the current disparity state of the QuadPCS transmitter is unknown. This is where the ff_correct_disp_ch[3:0] signal comes into play. If the ff_correct_disp_ch[3:0] signal is asserted for one clock cycle upon entering an interpacket gap, it will force the QuadPCS transmitter to insert an IDLE1 ordered-set into the transmit data stream if the current disparity is positive. However, if the current disparity is negative, then no change is made to the transmit data stream.

From the FPGA soft-logic side of the QuadPCS, the interpacket gap is typically characterized by the continuous transmission of the IDLE2 ordered set which is as follows:

ff_tx_k_cntrl_ch[3:0]=1, ff_txdata= 0xBC ff_tx_k_cntrl_ch[3:0]=0, ff_txdata=0x50.

Note that in the PCS channel, IDLE2s mean that current disparity is to be preserved. IDLE1s mean that the current disparity state should be flipped. Therefore, it is possible to ensure that the interpacket gap begins in a negative disparity state. If the disparity state before the interpacket gap is negative, then a continuous stream of IDLE2s are transmitted during the interpacket gap. If the disparity state before the interpacket gap is positive, then a single IDLE1 is transmitted followed by a continuous stream of IDLE2s.

In the FPGA soft-logic side of the QuadPCS, the interpacket gap is always driven with IDLE2s into the QuadPCS. The ff_correct_disp_ch[3:0] signal is asserted for one clock cycle, k_cntrl=0, data=0x50 when the interpacket gap first begins. If necessary, the QuadPCS will convert this IDLE2 into an IDLE1. For the remainder of the interpacket gap, IDLE2s should be driven into the QuadPCS and the ff_correct_disparity_chx signal should remain deasserted.

LatticeECP2M PCS in PCI Express Mode

An LatticeECP2M quad set to PCI Express Mode in IPexpress has additional ports at the FPGA interface to enable electrical functions required by the PCI Express specification such as receiver detection. Table 8-14 describes the PCI Express Mode specific ports.

Table 8-14. PCI Express Mode Specific Ports

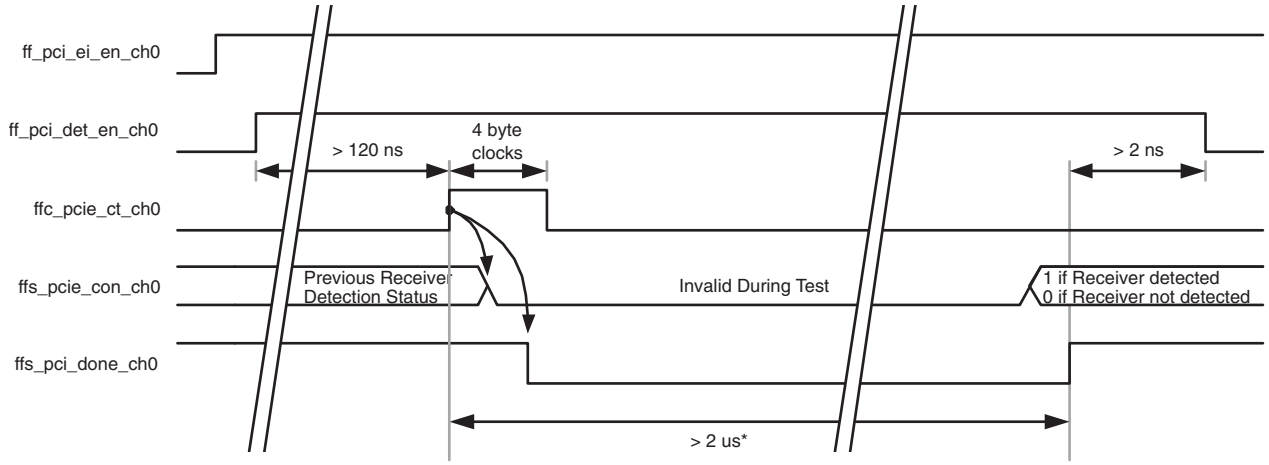
Signal	Direction	Class	Description
ffs_pcie_done_ch[3:0]	O	Channel	1 = Far-end receiver detection complete 0 = Far-end receiver detection incomplete
ffs_pcie_con_ch[3:0]	O	Channel	Result of far-end receiver detection. 1 = Far end receiver detected 0 = Far end receiver not detected.
ffc_pcie_det_en_ch[3:0]	I	Channel	FPGA logic (user logic) informs the SERDES block that it will be requesting for a PCI Express Receiver Detection operation. 1=Enable PCI Express Receiver Detect 0 = Normal operation
ffc_pcie_ct_ch[3:0]	I	Channel	1 = Request transmitter to do far-end receiver detection 0 = Normal data operation
ffc_ei_en_ch[3:0]	I	Channel	Control transmission of electrical idle by SERDES transmitter 1 = Force SERDES transmitter to output electrical idle 0 = Normal operation

Receiver Detection

Figure 8-29 shows a Receiver Detection sequence. A Receiver Detection test can be performed on each channel of a quad independently. Before starting a Receiver Detection test, the transmitter must be put into electrical idle by setting the **ff_pci_ei_en_ch** input high. The Receiver Detection test can begin 120 ns after tx_elec_idle is set high by driving the appropriate **ffc_pci_det_en_ch** high. This puts the corresponding SERDES Transmit buffer into receiver detect mode by setting the driver termination to high impedance and pulling both differential outputs to VCCOB through the high impedance driver termination.

Setting the SERDES Transmit buffer into receiver detect state takes up to 120 ns. After 120ns, the receiver detect test can be initiated by driving the channel's **ffc_pcie_ct_ch** input high for four byte (word) clock cycles. The corresponding channel's **ffc_pcie_done_ch** is then cleared asynchronously. After enough time for the receiver detect test to finish has elapsed (determined by the time constant on the Transmit side), the **ffs_pcie_done_ch** receiver detect status port will go high and the Receiver Detect status can be monitored at the **ffs_pcie_con_ch** port. If at that time the **ffs_pcie_con_ch** port is high, then a receiver has been detected on that channel. If, however, the **ffs_pcie_con_ch** port is low, then no receiver has been detected for that channel. Once the Receiver Detect test is complete, **ff_pci_ei_en_ch** can be deasserted.

Figure 8-29. PCI Express Mode Receiver Detection Sequence (Example for Channel 0)



PCI Express Beacon Support

This section highlights how the LatticeECP2M PCS can support Beacon Detection and Transmission. The PCI Express requirements for Beacon Detection are presented with the PCS support for Beacon Transmission and Beacon Detection.

- **Beacon Detection Requirements** (PCI Express Base Specification, Rev 1.0a, Chapter 4, Page 209-210)
  - Beacon is required for exit from L2 (P2) state.
  - Beacon is a DC balanced signal of periodic arbitrary data, which is required to contain some pulse widths >= 2ns (500Mhz) and < 16us (30Khz).
  - Maximum time between pulses should be < 16us.
  - DC balance must be restored within < 32us.
  - For pulse widths > 500ns, output beacon voltage level must be 6db down from VTX-DIFFp-p (800mV to 1200mV).
  - For pulse widths < 500ns, output beacon voltage level must be <= VTX-DIFFp-p and >= 3.5db down from VTX-DIFFp-p.
- **PCS Beacon Detection Support**
  - The signal loss threshold detection circuit senses if the specified voltage level exists at the receiver buffer. This is indicated by ffs_rlos_lo_ch(0-3) signal.
  - This setting can be used both for PCI Express Electrical Idle Detection and PCI Express Beacon Detection (when in power state P2).
  - The remote transmitting device can have a Beacon Output voltage of 6db down from VTX-DIFFpp (i.e 201mV). If this signal can be detected, it can be stated that Beacon is detected.
- **PCS Beacon Transmission Support**
  - Sending the K28.5 character (IDLE) (5 1's followed by 5 0's) provides a periodic pulse with of 2ns occurring every 2ns (1.0UI = 400ps, multiplied by 5 = 2ns). This meets the lower requirement. The output beacon voltage level can then be $V_{TX\text{-}DIFFp\text{-}p}$. This would be valid Beacon Transmission.

PCS Loopback Modes

The LatticeECP2M family of devices provides three loopback modes controlled by control signals at the PCS/FPGA interface for convenient testing of the external SERDES/board interface and the internal PCS/FPGA logic interface. Three loopback modes are provided to loop received data back onto the transmit data path. The loopback modes are useful for checking the high speed serial SERDES package pin connections as well as the embedded SERDES and/or PCS logic.

Serial Loopback Mode

Loops serial receive clock/data back onto the transmit buffer without passing through the CDR or de-serializer. This feature is intended for internal testing purpose but users can also use it.

Selecting the Serial Loopback option in the IPexpress GUI will set only the LB_CTL[3:0] to '0010' (refer to Table 8-77).

The TDRV_DAT_SEL[1:0] register bits should be also set to '11' via SCI to enable Serial Loopback mode. The LB_CTL[3:0] register bits are also accessible via SCI.

SERDES Parallel Loopback Mode

Loops parallel receive data back onto the transmit data path without passing through the PCS logic. SERDES parallel loopback can be selected for each channel individually by setting the appropriate ffc_sb_pfifo_lp_ch(0-3) to a "1".

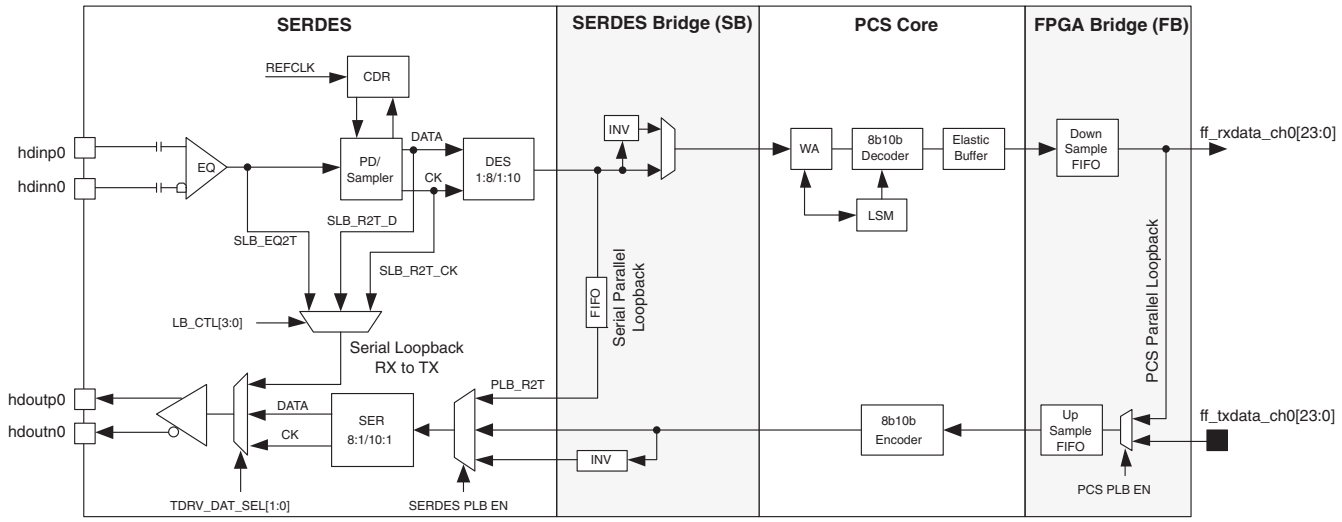
PCS Parallel Loopback Mode

Loops parallel receive data back onto the transmit data path without passing across the PCS/FPGA interface. Input serial data at the SERDES hdin package pins passes through the SERDES receive logic where it is converted to

parallel data, passes through the entire PCS receive logic path, is looped back through the entire PCS transmit logic path, is reconverted back to serial data by the SERDES transmitter and sent out onto the hdout SERDES package pins. PCS Parallel loopback can be selected for each channel individually by setting the appropriate ffc_fb_loopback_ch(0-3) port to a “1”.

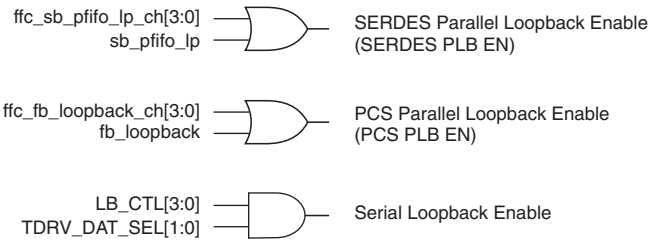
Figure 8-30 illustrates the three loopback modes for a single channel.

Figure 8-30. Three Loopback Modes



The two parallel loopback modes described above provide not only control signals from the FPGA core but also Control Register bits. When the control register bits are not set for the loopback mode, the data path can switch between Loopback mode and Normal Data Flow by control signals from the FPGA core. Figure 8-31 describes this logic. Refer to Tables 8-77 and 8-78 for detailed register settings.

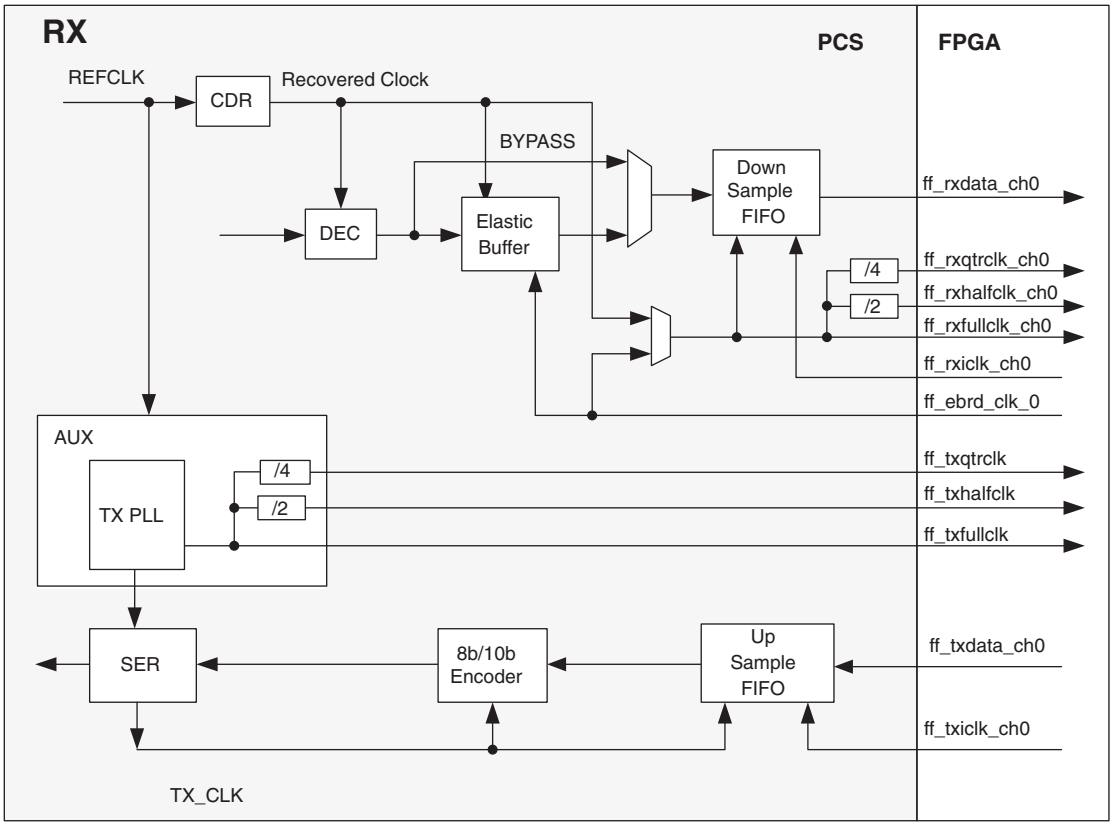
Figure 8-31. Loopback Enable Signals



FPGA Interface Clocks Usage

Figure 8-32 shows a conceptual diagram of the later stage of the PCS Core and the FPGA Bridge and the major clocks that cross the boundary between PCS and the FPGA.

Figure 8-32. Conceptual PCS/FPGA Clock Interface Diagram



In the above diagram and in the subsequent clock diagrams in this section, note that suffix “i” indicates the index [0:3] i.e., one for each channel.

None of the clock muxes have logic to guard against clock slicing or glitching. It is a requirement that if any of the selectors to the clock muxes are changed by writes to register bits, the software agent will reset the logic clocked by that muxed clock.

The PCS outputs 15 clocks. There are three transmit clocks (per quad) and 12 receive clocks (per channel). The three transmit clocks provide full rate, half rate and quarter rate clocks and are all generated from the Tx PLL. The full rate and half rate transmit clocks need to be send directly via dedicated route to the Center Clock Mux in the FPGA. There are also three clocks (full, half and quarter rates) per receive channel. All 15 clocks can be used as local (secondary) or global (primary) clocks for the FPGA logic as required. Divided-by-two clocks are used when the gearing is in 2:1 mode (the gearing is selectable only on a quad basis).

The transmit clock is used on the write port of the Up Sample FIFO (or Phase Shift FIFO, depending on the case). One of the two receive clocks is connected to the read clock of the Down Sample FIFO. The other clocks the read port of the Elastic Buffer FIFO and potentially (depending on the case) the write port of the Down Sample FIFO.

Based on the whether the Elastic Buffer and the Up Sample FIFO are bypassed or not and whether we are in 8b10b mode or 16b20b mode, four use cases are possible. The active paths are highlighted with weighted lines. It is also indicated how many and what kind of clock trees are required. There are some modes that would more commonly be preferred by the user.

This section describes the operation for the six different cases that are supported. The six cases are outlined in Table 8-15.

Table 8-15. Six Interface Cases Between SERDES/PCS Quad and FPGA Core

Interface	Data Width	Rx CTC FIFO	Rx Phase-Shift/Down-Sample FIFO	Tx Phase-Shift/ Up-Sample FIFO
Case I-a[1]	8b10b	Yes	Yes	Yes
Case I-b[1]	8b10b	Bypass	Yes	Yes
Case II-a[1]	16b20b	Yes	Yes	Yes
Case II-b[1]	16b20b	Bypass	Yes	Yes

1. In all cases in which the Tx phase-shift (up-sample) FIFO is used, it is never bypassed. Deep inside the SERDES/PCS block, the datapath width is 8/10 bits wide and the byte clock runs at 1/10 of the SERDES line rate. For example, if the SERDES line rate is 3.125Gbps, then the byte clock is 312.5MHz.

2-to-1 Gearing

For guaranteed performance of the FPGA global clock tree, it is recommended to use a 16/20 bit wide interface for SERDES line rates greater than 2.5Gbps. In this interface, the FPGA interface clocks are running at half the byte clock frequency.

Even though the 16/20 bit wide interface running at half the byte clock frequency can be used with all SERDES line rates, the 8/10 bit wide interface is preferred when the SREDES line rate is low enough to allow it (2.5Gbps and below) because this results in the most efficient implementation of IP in the FPGA core.

The decision matrix for the six interface cases is explained in Table 8-16.

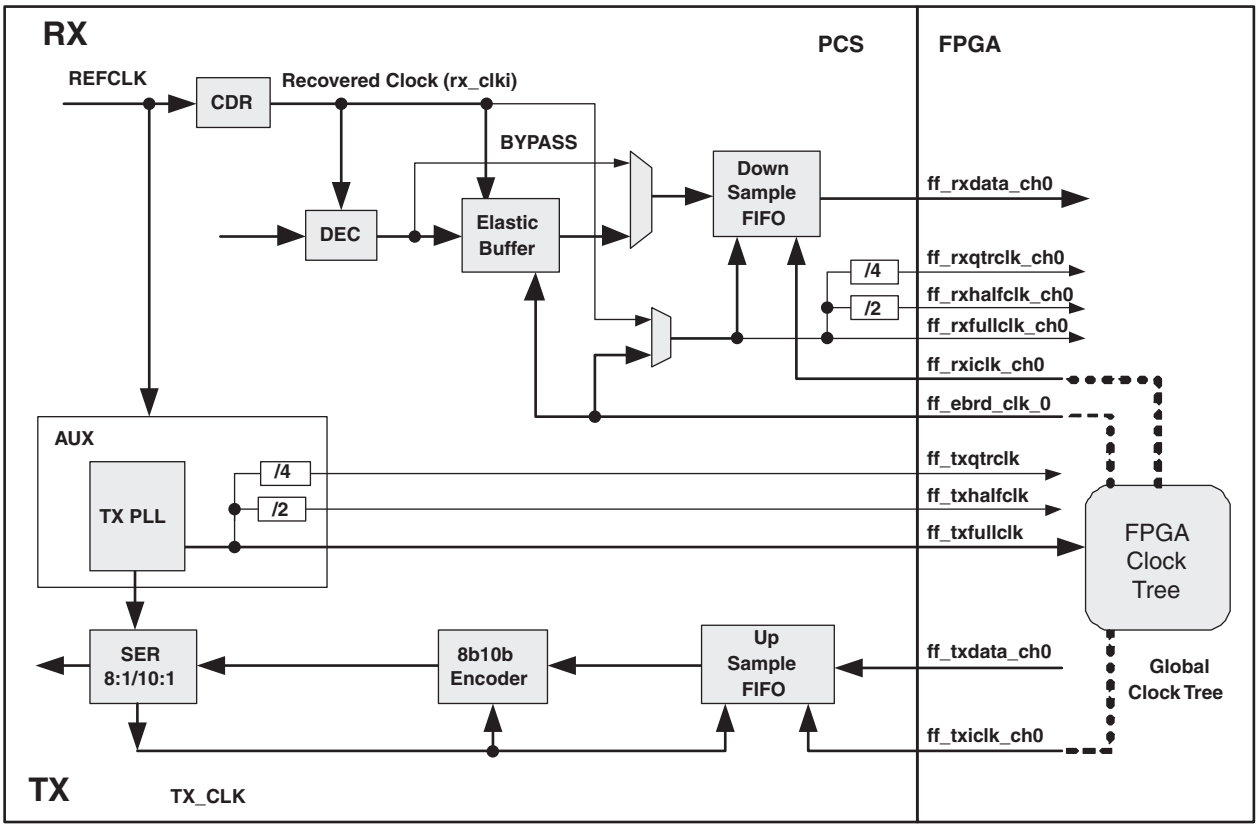
Table 8-16. Decision Matrix for Six Interface Cases

SERDES Line Rate	Datapath Width	MCA Required?	CTC Required?	Interface Case
2.5 Gbps and below	8/10 bit (1:1 gearing)	No, single-channel link	Yes	Case I_a[1]
			No	Case I_b[2]
		Yes, multi-channel link	Must bypass, not available	Case I_b[3]
3.2 Gbps and below	16/20 bit (2:1 gearing)	No, single-channel link	Yes	Case II_a[4]
			No	Case II_b[5]
		Yes, multi-channel link	Must bypass, not available	Case II_b[6]

1. This case is intended for SINGLE-channel links at line rates of 2.5Gbps and lower (8/10 bit wide interface) that require clock tolerance compensation in the quad. CTC is required when both ends of the link have separate reference clock sources that are within +/- 300ppm of each other. Case I_a is used if the IP in the core requires the Rx phase-shift FIFO. Case I_b is used if the IP does not require this FIFO.
2. This case is intended for SINGLE-channel links at line rates of 2.5Gbps and lower (8/10 bit wide interface) that do NOT require clock tolerance compensation in the quad. CTC is not required when both ends of the link are connected to the same reference clock source. This is often the case for chip-to-chip links on the same circuit board. There is exactly 0ppm difference between the reference clocks and so CTC is not required and can be bypassed. CTC is also not required in the quad when this function is performed by the IP in the core.
3. This case is intended for MULTI-channel links at line rates of 2.5Gbps and lower (8/10 bit wide interface). Multi-channel alignment MUST be done by the IP in the core, there is no provision to do MCA in the quad. Since MCA must be done prior to CTC, the CTC FIFO in the quad MUST be bypassed when MCA is required and so both MCA and CTC (if required) are done by the IP in the core.
4. This case is intended for SINGLE-channel links at line rates of 3.2Gbps and lower that require a 2:1 gearbox between the quad and the FPGA core (16/20 bit wide interface). Clock tolerance compensation is included in the quad. CTC is required when both ends of the link have separate reference clock sources that are within +/- 300ppm of each other.
5. This case is intended for SINGLE-channel links at line rates of 3.2Gbps and lower that require a 2:1 gearbox between the quad and the FPGA core (16/20 bit wide interface). Clock tolerance compensation is NOT included in the quad. CTC is not required when both ends of the link are connected to the same reference clock source. This is often the case for chip-to-chip links on the same circuit board. There is exactly 0ppm difference between the reference clocks and so CTC is not required and can be bypassed. CTC is also not required in the quad when this function is performed by the IP in the core.
6. This case is intended for MULTI-channel links at line rates of 3.2Gbps and lower that require a 2:1 gearbox between the quad and the FPGA core (16/20 bit wide interface). Multi-channel alignment MUST be done by the IP in the core, there is no provision to do MCA in the quad. Since MCA must be done prior to CTC, the CTC FIFO in the quad MUST be bypassed when MCA is required and so both MCA and CTC (if required) are done by the IP in the core.

Case I_a: 8/10bit, EB and DS FIFOs NOT Bypassed

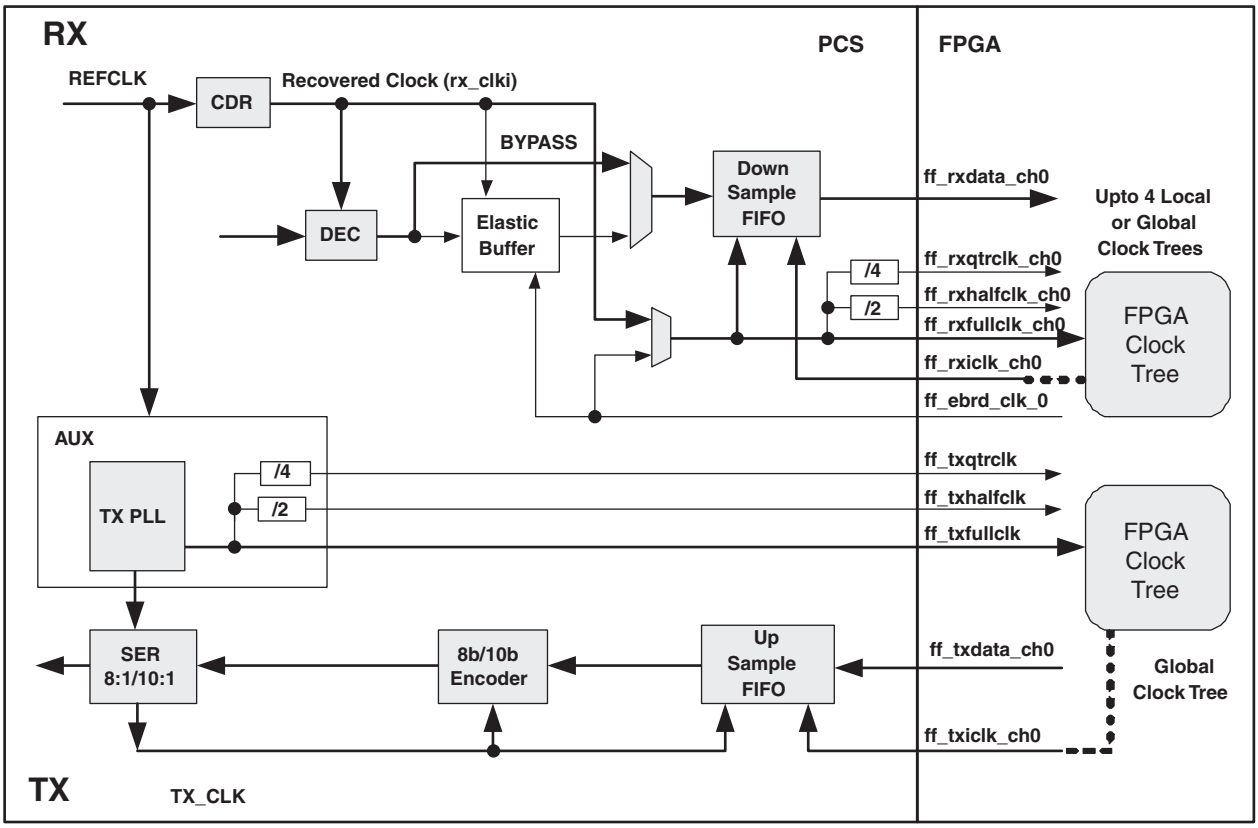
Figure 8-33. 8b10b, EB and DS FIFOs NOT Bypassed



1. The Up Sample FIFO is acting as a Phase Shift FIFO only in this case.
2. The Down Sample FIFO is acting as a Phase Shift FIFO only in this case.
3. The quad level full rate clock from the Tx PLL (ff_tx_f_clk) has direct access to the FPGA Center Clock Mux. This is a relatively higher performance path. A Global Clock Tree out of the Center Clock Mux is used to clock the user's interface logic in the FPGA. Some leaf nodes of the clock tree are connected to the FPGA Transmit Input clock (ff_txi_clki), the Elastic Buffer FIFO Read Clock per Channel (ff_ebrd_clki) via CIB Clk input and the FPGA Receive Input clock (ff_rxi_clki). This case is probably the most common single channel use case.

Case I_b: 8/10bit, EB FIFO Bypassed

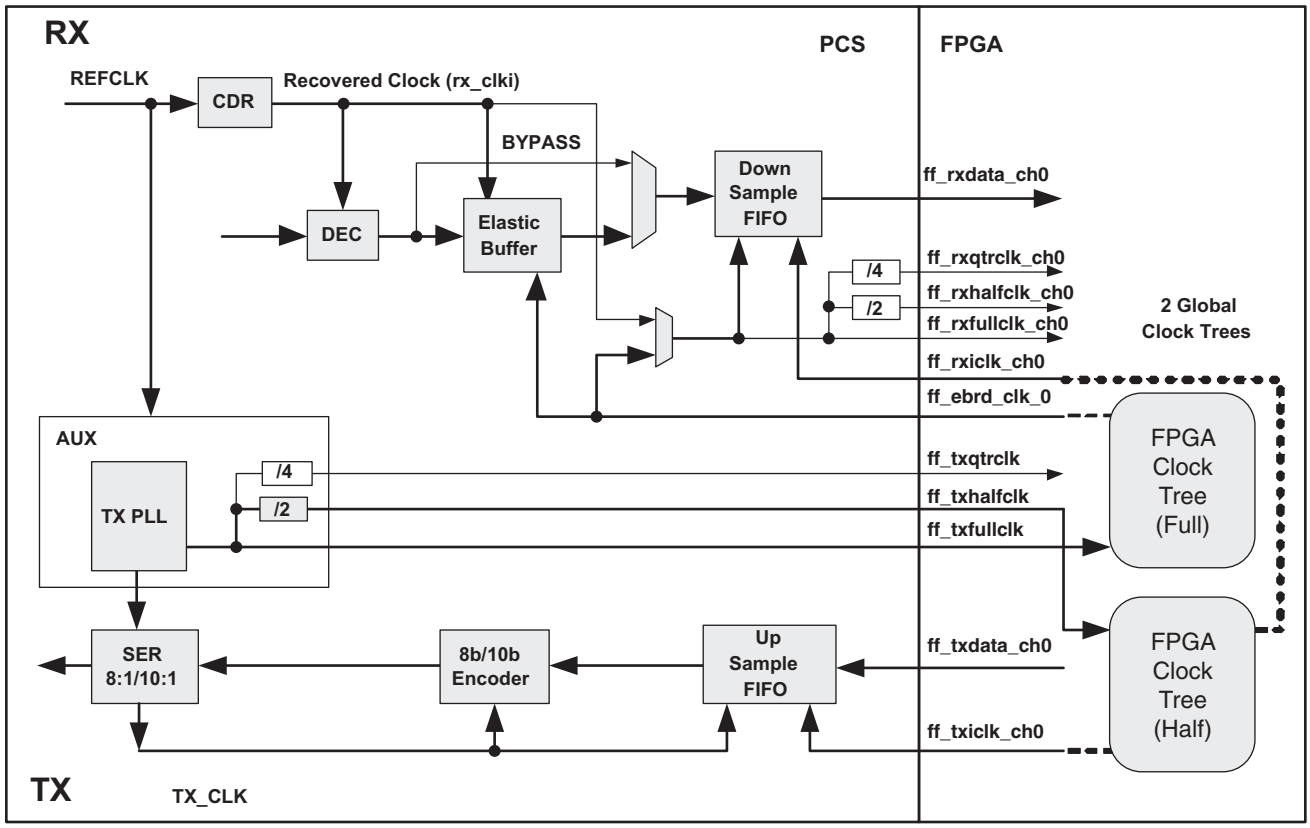
Figure 8-34. 8b10b, EB FIFO Bypassed



1. The Up Sample FIFO is acting as a Phase Shift FIFO only in this case.
2. The Down Sample FIFO is acting as a Phase Shift FIFO only in this case.
3. The Tx FPGA Channel input clock is clocked similarly as in the previous case using a clock tree driven by a direct connection of the full rate transmit FPGA output clock to the FPGA center clock mux Once the Elastic Buffer is bypassed, the recovered clock needs to control the write port of the Down Sample FIFO. The recovered clock of each channel may need to drive a separate local or global clock tree (i.e., up to 4 local or global clock trees per quad). The clock tree will then drive the FPGA receive clock input to control the read port of the Down Sample FIFO. The reason for bypassing the Elastic Buffer FIFO in this case is most likely for doing multi-channel alignment (MCA) in the FPGA core. It implies that CTC using an elastic buffer will be done in the FPGA core. The CTC FIFOs can be written by either the recovered clocks or by a master recovered clock. The read of the CTC FIFO will be done using the Tx clock via the Tx clock tree.

Case II_a: 16/20bit, EB and DS FIFOs NOT Bypassed

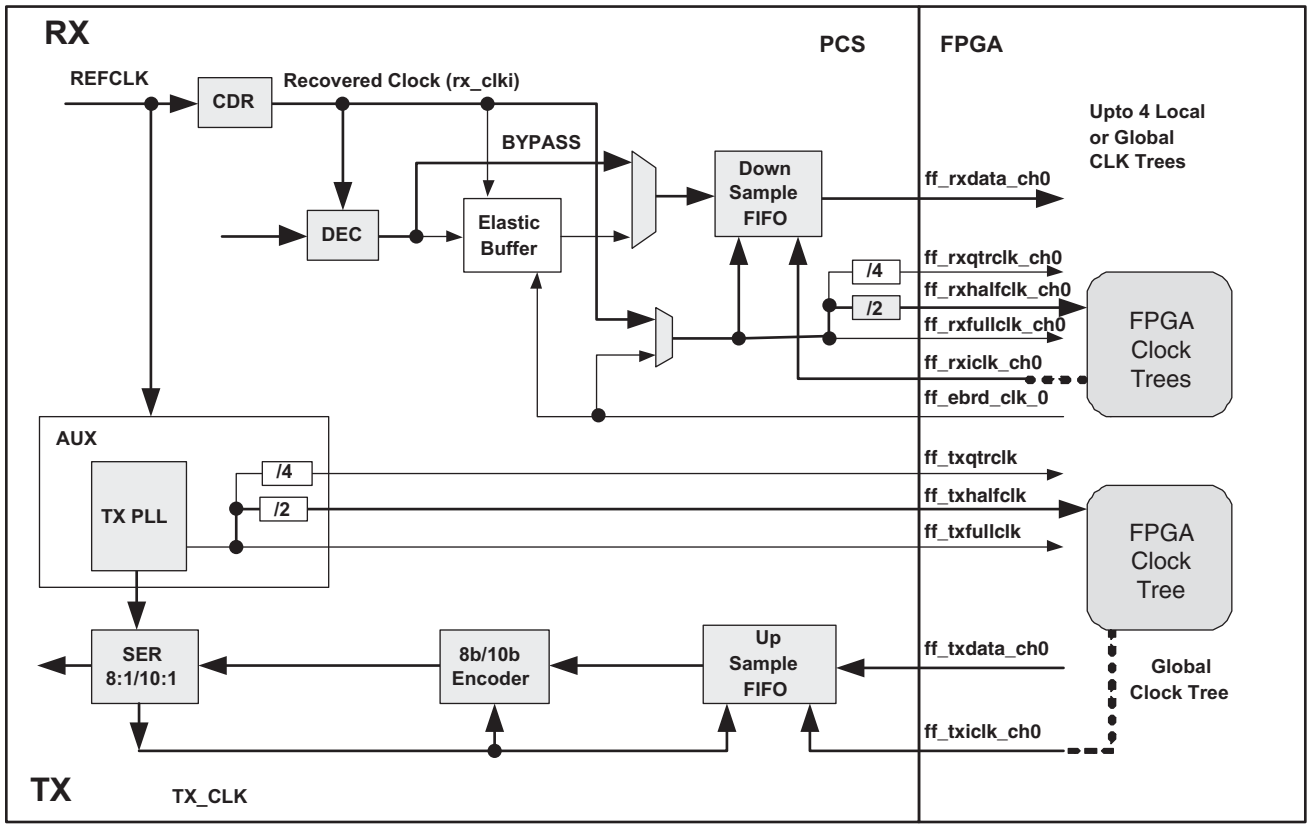
Figure 8-35. 16/20bit, EB and DS FIFOs NOT Bypassed



1. The Up Sample FIFO is acting both as a Phase Shift FIFO and Up Sample FIFO in this case.
2. The Down Sample FIFO is acting both as a Phase Shift FIFO and Down Sample FIFO in this case.
3. This is a very common single channel use case when the FPGA is unable to keep up with full byte frequency. Two clock trees are required. These clock trees are driven by direct access of transmit full rate clock and transmit half rate clock to the FPGA clock center mux. The full rate clock tree drives the Elastic Buffer read port and the Down Sample FIFO write port. The half rate clock tree drives the Down Sample FIFO and the FPGA logic.

Case II_b: 16/20bit, DS NOT Bypassed

Figure 8-36. 16/20 Bit, DS FIFO NOT Bypassed



1. The Up Sample FIFO is acting both as a Phase Shift FIFO and Up Sample FIFO in this case.
2. The Down Sample FIFO is acting both as a Phase Shift FIFO and Down Sample FIFO in this case.
3. This is a very common multi-channel alignment (MCA) use case when the FPGA is unable to keep up with full byte frequency. The receive clock trees (up to 4) can be local or global. They are running half rate clock. The transmit clock tree is driven by direct access of transmit half rate clock to the FPGA clock center mux. In this case, in FPGA logic after the MCA is done, a CTC will be required. The Tx clock tree will clock the read port of the CTC and the master channel receive clock the write port of the CTC. It is important to note that both the MCA and CTC need to be done across 16/20bits in this use case.

SERDES/PCS Block Latency

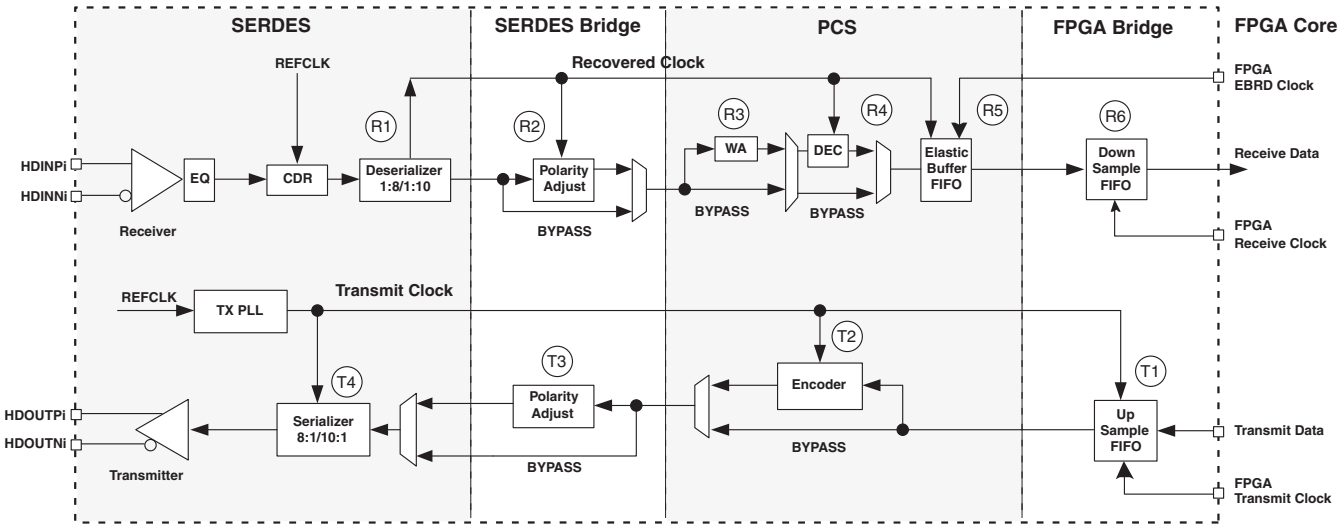
Table 8-17 describes the latency of each functional block in the transmitter and receiver. Latency is given in parallel clock cycles. Figure 8-37 shows the location of each block.

Table 8-17. SERDES/PCS Latency Breakdown (Parallel Clock Cycle)[1]

Item	Description	Min.	Average	Max.	Fixed	Bypass	Units
Transmit Data Latency							
T1	FPGA Bridge Transmit[2]	1	3	5		1	word clk
T2	8b10b Encoder	—	—	—	2	1	word clk
T3	SERDES Bridge Transmit	—	—	—	2	1	word clk
T4[3]	Serializer: 8-bit mode	—	—	—	15 + Δ1	—	UI + ps
	Serializer: 10-bit mode	—	—	—	18 + Δ1	—	UI + ps
Receive Data Latency							
R1[3]	Deserializer: 8-bit mode	—	—	—	10 + Δ2	—	UI + ps
	Deserializer: 10-bit mode	—	—	—	12 + Δ2	—	UI + ps
R2	SERDES Bridge Receive	—	—	—	2	1	word clk
R3	Word Alignment	3.1	—	4	—	0	word clk
R4	8b10b Decoder	—	—	—	1	1	word clk
R5	Clock Tolerance Compensation	7	15	23		1	word clk
R6	FPGA Bridge Receive[2]	1	3	5		1	word clk

1. PCS internal Parallel Clock. This clock rate is same as the rxfullclk in Table 8-5.
2. FPGA Bridge latency varies by UP/DOWN Sample FIFO read/write. These numbers were presented for 8bit/10bit interface. The depth of Down Sample/Up Sample FIFO is 4. The earliest read can be done after write clock cycle (1 clock) in Down Sample FIFO. The latest read will be done after the FIFO is full (4 + 1 = 5). For 16b/20b interface, the numbers become doubled. Min = 2, Max = 10. This latency depends on the internal FIFO flag operation.
3. Δ1 = -245ps, Δ2 = 700ps

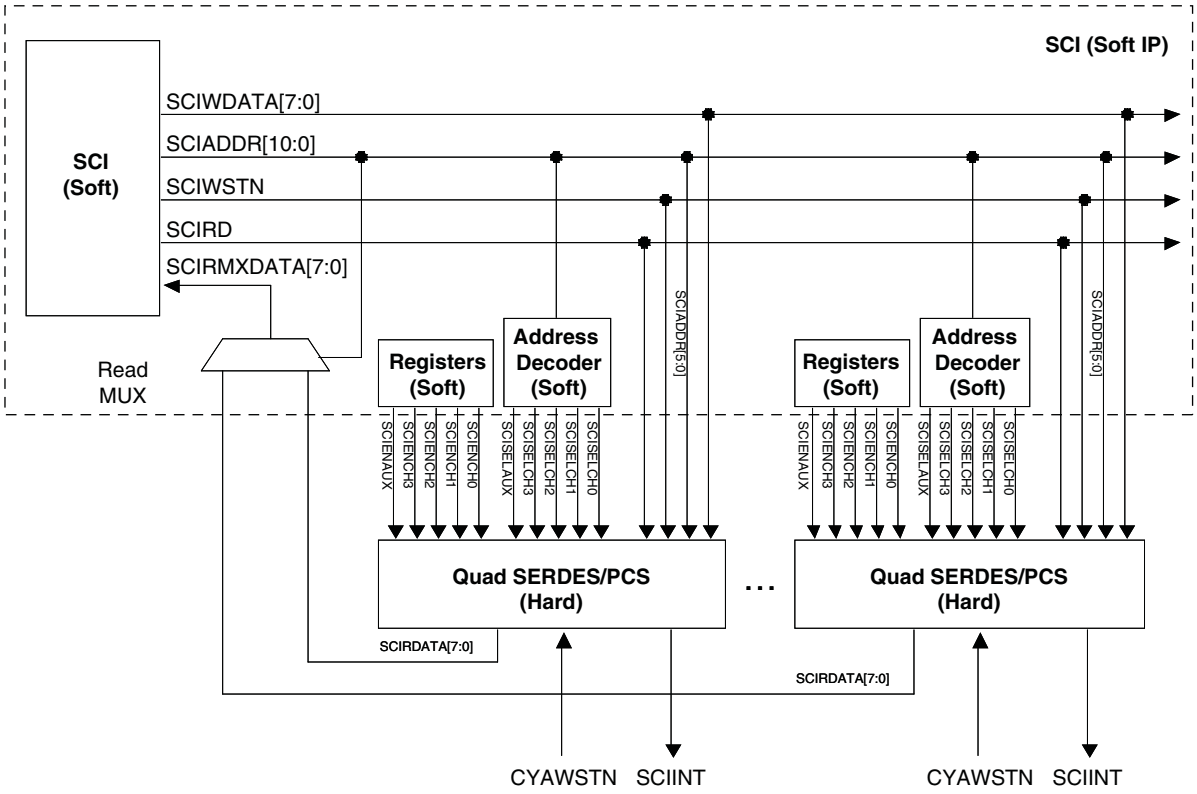
Figure 8-37. Transmitter and Receiver Block Diagram



SERDES Client Interface (SCI)

The SERDES Client Interface (SCI) consists of both soft IP that resides in the FPGA core and permanent logic that is included in the SERDES/PCS quad. The SCI allows the SERDES/PCS quad to be controlled by registers as opposed to the configuration memory cells. It is a simple register configuration interface. It shows all the major signals required. The block diagram of the soft IP part of the SCI that resides in the FPGA core is shown in Figure 8-38.

Figure 8-38. SCI Interface Block Diagram



The soft IP that resides in the FPGA core should be developed by users per their interface scheme. Contact the Lattice Technical Support Group for example code.

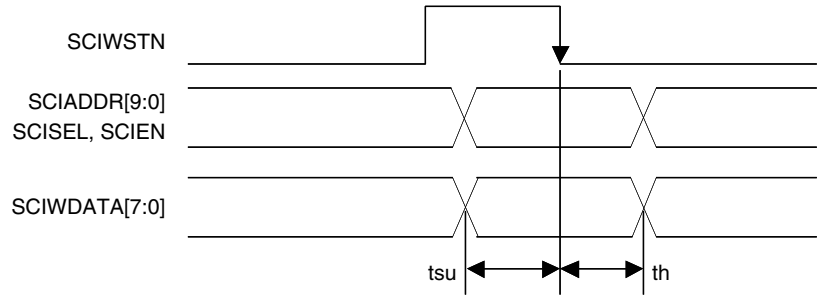
The SCIADDR bus is six bits wide within the block. The bus width at the block boundary is 11 bits. The upper five bits are used for quad block selection and channel selection. Table 8-18 shows the SCI address map for the SERDES quad.

Table 8-18. SCI Address Map for Up to Four SERDES/PCS Quads

Address Bits	Description
SCIADDR[5:0]	Register address bits 000000 = select register 0 000001 = select register 1 … 111110 = select register 62 111111 = select register 63
SCIADDR[8:6]	Channel address bits 000 = select channel 0 001 = select channel 1 010 = select channel 2 011 = select channel 3 100 = select aux channel 101 = unused 110 = unused 111 = unused
SCIADDR[10:9]	Quad address bits 00 = select quad 0 01 = select quad 1 10 = select quad 2 11 = select quad 3

Read and write operations through this interface are asynchronous. In the WRITE cycle the Write data and Write address need to be setup and held in relation to the falling edge of the SCIWSTN. In the READ cycle the timing has to be in relation with the SCIRD pulse. Figures 8-39 and 8-40 shows the WRITE and READ cycles respectively.

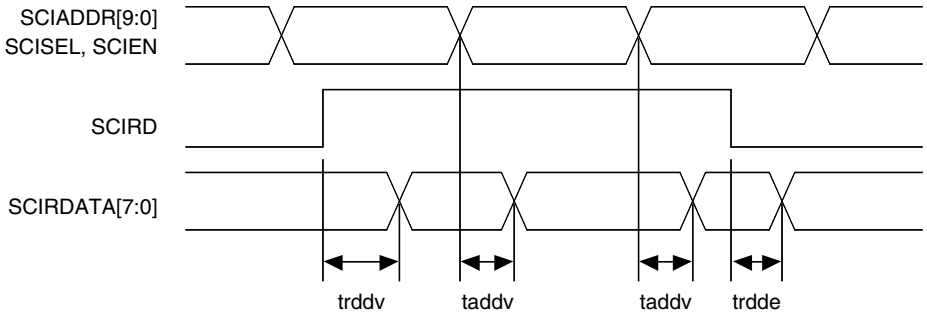
Figure 8-39. SCI WRITE Cycle, Critical Timing



Notes :
1) tsu is the setup time for address and write data prior to the falling edge of the write strobe.
2) th is the hold time for address and write data after the falling edge of the write strobe.

Note: To avoid accidental writing to control registers, registers should be used at the SCI input ports to drive them low at power-up reset.

Figure 8-40. SCI READ Cycle, Critical Timing



Notes:
1) trddv is the time from assertion of the read pulse until read data is valid.
2) taddv is the time from address change until data is valid while read pulse is asserted.
3) trdde is the hold time of the read data after the deassertion of the read pulse.

Table 8-19. Timing Parameters

Parameter	Typical Value	Units
tsu, trddv, taddv	1.127	ns
th, trdde	0.805	ns

The SCI interface is as simple as memory read/write. Here is an example of the pseudo code:

Write

```
Cycle 1 :  Set sciaddr[5:0], sciwdata[7:0], scien* = 1'b1 scisel* = 1'b1
Cycle 2 :  Set sciwstn from 0 => 1
Cycle 3 :  Set sciwstn from 1 => 0, scien* = 1'b0, scisel* = 1'b0
```

Read

```
Cycle 1 :  Set sciaddr[5:0], scisel* = 1'b1
Cycle 2 :  Set scird from 0 => 1
Cycle 3 :  Obtain reading data from scirdata[7:0]
Cycle 4 :  Set scird from 1 => 0
```

Interrupts and Status

The status bit may be read via the SCI, which is a byte wide and thus reads the status of eight interrupt status signals at a time. The SCIINT signal goes high to indicate that an interrupt event has occurred. The user is then required to read the QIF status register that indicates whether the interrupt came from the quad or one of the channels. This register is NOT cleared-on-read. It is cleared when all interrupt sources from the quad or channel is cleared.

Once the aggregated source of interrupt is determined, the user can read the registers in the associated quad or channel to determine the actual source of the interrupt. Tables 8-20 and 8-21 list all the sources of interrupt.

Table 8-20. Quad Interrupt Sources

Quad SCI_INT Source	Description	Register Name
int_quad_out	Quad Interrupt. If there is an interrupt event anywhere in the quad this register bit will be active. This register bit gets cleared when all interrupt events have been cleared.	PCS Quad Status Register 1
int_cha_out[0:3]	Channel Interrupt. If there is an interrupt event anywhere in the respective channel this register bit will be active. These register bits are cleared when all interrupt sources in the respective channel have been cleared.	PCS Quad Status Register 1
ls_sync_statusn_[0:3]_int ls_sync_status_[0:3]_int	Link Status Low (out of sync) channel interrupt Link Status High (in sync) channel interrupt	PCS Quad Status Register 3
~PLOL, PLOL	Interrupt generated on ~PLOL and PLOL - PLL Loss of Lock	SERDES Quad Status Register 2

Table 8-21. Channel Interrupt Sources

Quad SCI_INT Source	Description	Register Name
fb_tx_fifo_error_int fb_rx_fifo_error_int cc_overrun_int cc_underrun_int	FPGA Bridge Up Sample Tx FIFO Error Interrupt FPGA Bridge Down Sample Rx FIFO Error Interrupt CTC (Elastic Buffer) Overrun and Underrun Interrupts	PCS Channel General Interrupt Status Register 4
pci_det_done_int rlos_lo_int ~rlos_lo_int rlos_hi_int ~rlos_hi_int rlol_int ~rlol_int	Interrupt generated for pci_det_done Interrupt generated for rlos_lo Interrupt generated for ~rlos_lo Interrupt generated for rlos_hi Interrupt generated for ~rlos_hi Interrupt generated for rlol Interrupt generated for ~rlol	SERDES Channel Interrupt Status Register 5

SERDES Client Interface Application Example

Lattice Semiconductor's ORCAstra FPGA configuration software is a PC-based Graphical User Interface (GUI) that allows users to configure the operational mode of a Lattice FPGA by programming control bits in the FPGA registers. SERDES/PCS status information is displayed on-screen in real time, and any configuration can be saved to control registers for additional testing. Use of the GUI does not interfere with the programming of the FPGA core portion. More information and downloadable files for ORCAstra can be found on the Lattice Semiconductor website at www.latticesemi.com/products/designsoftware/orcastra.cfm.

For example:

Users can switch the reference clock multiplier mode between 10x mode and 20x mode by changing refck_mode[1](D7) and refck_mode[0](D6) bits in the Quad PCS Control Register, ser_ctl_3_qd_13. The SCI address of the register is 13(H) and ORCAstra Address = 113 (H). The Quad Reset (QD_18[5]) must be toggled to achieve this transition.

The read/write operation can be achieved in two different ways.

1. In the main window, users can read and write the content of control register read status registers.
2. In the sub-windows, users can select pull-down menu options, or check/uncheck ON/OFF options.

Table 8-22. SCI Address Map

Address Bit	Description
SCIADDR[5:0]	Register address bits 000000 = register 0 000001 = register 1 ..... 111110 = register 62 111111 = register 63
SCIADDR[8:6]	Channel address bits 000 = channel 0 001 = channel 1 010 = channel 2 011 = channel 3 100 = Aux channel
SCIADDR[10:9]	Quad address bits 00 = quad 0 01 = quad 1 10 = quad 2 11 = quad 3

Dynamic Configuration of SERDES/PCS Quad

The SERDES/PCS quad can be controlled either by the configuration memory cells or by registers that are accessed through the optional “SERDES Client Interface” (SCI). The SCI consists of both a soft IP that resides in the FPGA core and permanent logic that is included in the SERDES/PCS quad. The SCI is only available if the soft IP is present in the FPGA core.

After configuration is complete, those configuration memory cells that have associated registers are automatically copied into the registers. Subsequent changes to the contents of the registers will not affect the value stored in the configuration memory cells but will of course change the operation of the SERDES/PCS quad.

When controlled by the configuration memory cells, it is a requirement that the SERDES/PCS quads must reach a functional state after configuration is complete, without further intervention from the user. This means that any special reset sequences that are required to initialize the SERDES/PCS quad must be handled automatically by the hardware. In other words, use of the SCI is optional, the SERDES/PCS quad must NOT assume that the soft IP is present in the FPGA core.

SERDES Debug Capabilities

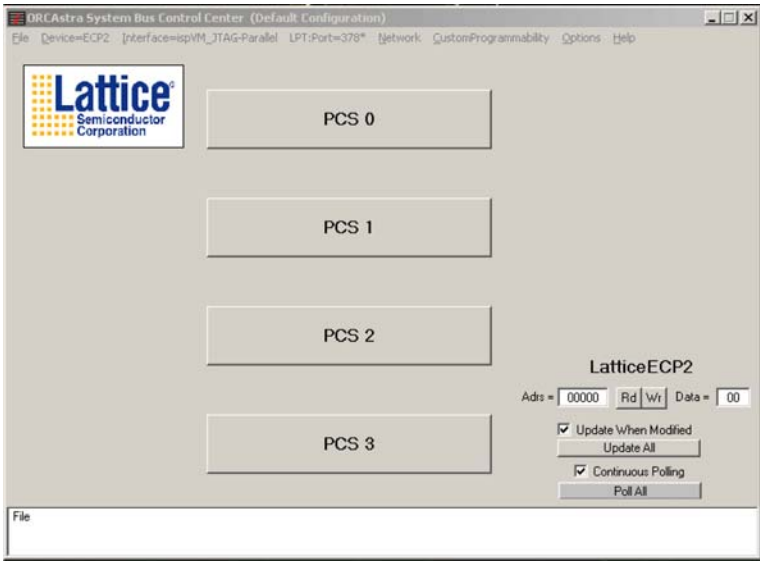
Lattice has tools to help in the debug of the SERDES/PCS operation in LatticeECP2M devices.

Lattice ORCAstra software is a PC-based graphical user interface for configuring the operational mode of a LatticeECP2M device by programming control bits in the on-chip registers. This helps you quickly explore configuration options without going through a lengthy re-compile process or making changes to your board. Configurations created in the GUI can be saved to memory and re-loaded for later use. To use ORCAstra, SCI IP must be instantiated in the user logic.

A macro capability is also available to support script-based configuration and testing. The GUI can also be used to display system status information in real time. Use of the ORCAstra software does not interfere with the programming of the FPGA.

Figure 8-41 shows the ORCAstra GUI top-level window. Users can read and write in this window without going through the subwindows for each PCS channel by read and write data at Adrs cell. When invoked, ORCAstra will automatically recognize the device type. Or, device types can be selected under the device pull-down menu.

Figure 8-41. ORCAstra Top-Level Screen Shot

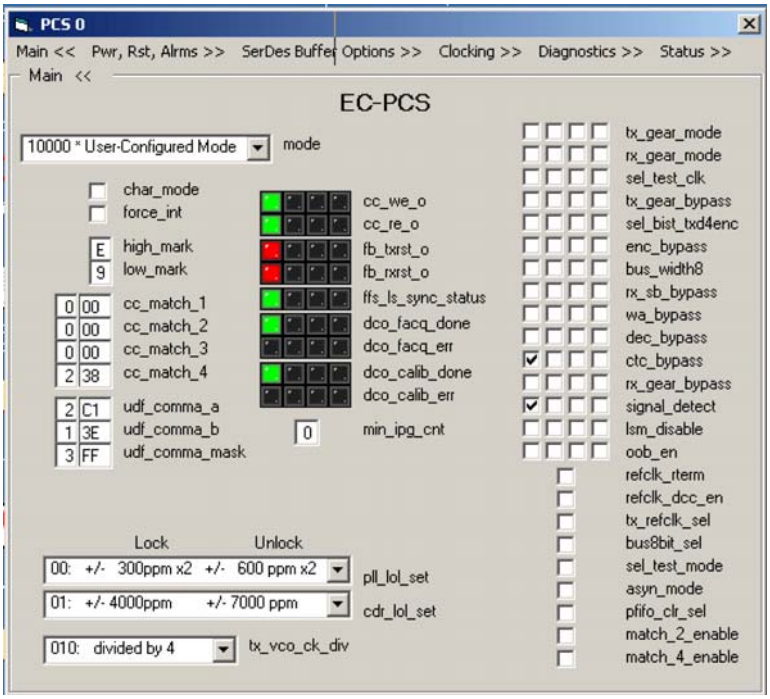


By default, the Data Box shown in Figure 8-41 follows Little Endian byte order (i.e., the most significant bit is placed on the right). Users can change to Big Endian order by selecting **Display Data reversed in Data Box** under the **Options** tab.

Double clicking on the **PCS0 (Quad 0)** button will open the main window as shown in Figure 8-42.

These standard Windows menus control the selection of the device and interface. They also support various configuration options, including setting up and saving configurations with stored files.

Figure 8-42. ORCAstra Main Window

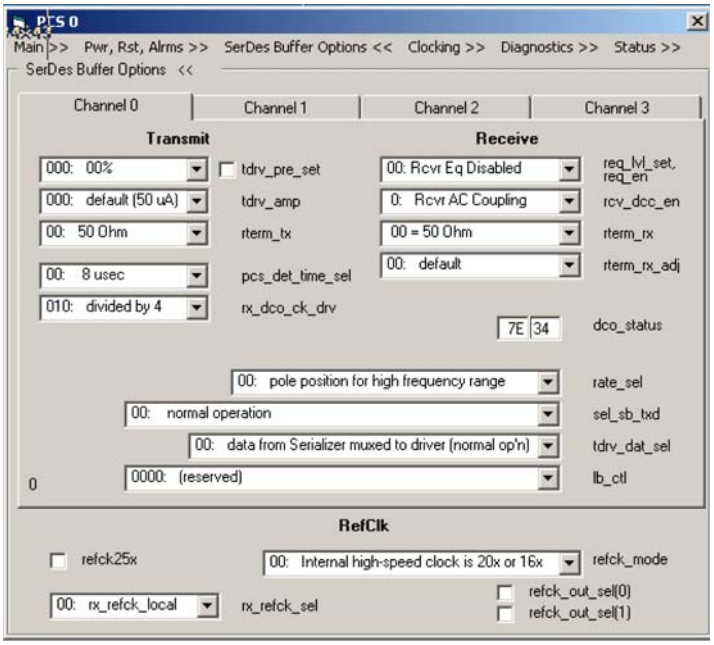


Control Boxes and Buttons, Status Boxes and the Text Window

Moving the cursor over a control box and clicking the left mouse button sets the control bits. Both the bit location and function for the selected box are displayed in the text window and will match those of the register map tables in the device's data sheet. Only the function is displayed when the cursor is over the bit name. Status boxes are similar to control boxes but have an LED appearance and a colored background.

Figure 8-43 shows SERDES Buffer Options window. Configuration options can be selected from the pull-down menu.

Figure 8-43. SERDES Buffer Options Window



More information and downloadable files for ORCAstra can be found on the Lattice Semiconductor website at the following address: www.latticesemi.com/products/designsoftware/orcastra.cfm.

Other Design Considerations

LatticeECP2M-35 vs. All Other LatticeECP2M Devices

Important Note: The SERDES settings (in the .txt file) for the LatticeECP2M-35 are different from all other LatticeECP2M devices. The difference includes default settings that are hidden in the .txt file. When there is device migration from LatticeECP2M-35 to all other LatticeECP2M devices, users must re-generate the PCS module from the IPexpress GUI.

Engineering Samples vs. Production Devices

Engineering samples and production devices have their own simulation models and libraries. The library name for the engineering sample is pcsc_ mti_work_revA.

Simulation of the SERDES/PCS

SERDES/PCS simulation support is provided by a pre-compiled model for ModelSim® and NC-Verilog®. The simulation model that is pre-compiled is a behavioral model for the SERDES and the RTL of the PCS. Table 8-23 provides the location for each of the simulation models.

Table 8-23. Simulation Model Locations

Simulator	Model Location
Cadence NC-Verilog/VHDL, NC-Sim, Synposys VCS, Mentor Graphics ModelSim, Aldec Riviera Pro	ispTools\cae_library\simulation\blackbox

Note: Model file names with “revA” are for LatticeECP2M-35 engineering samples only.

Models that are distributed in .zip files need to be decompressed before the model can be used.

Reset Usage in Simulation

If any of the PCS reset signals is not tied to GND, the designer must be careful if the same reset signal is used in the FPGA core. For example, if one of the reset signals is used to reset both the PCS and a counter in the FPGA core, and the counter uses txfullclk, the counter will not work. The counter cannot reset itself because txfullck is not running when the reset is active.

Depending on the reference clock sources, the reset assertion time can vary.

For simulation only, it is recommended to use a minimum active duration of 100ns for PCS reset signals.

16/20-bit Word Alignment

The PCS receiver can not recognize the 16-bit word boundary. With COMMA_ALIGN in 'AUTO' mode, the PCS can only do BYTE alignment. The 16-bit word alignment should be done in the FPGA fabric and is fairly straight forward. The simulation model works in the same way. It can be enhanced if users implement an alignment scheme as described below.

For example, if transmit data before at the FPGA interface are:

```
YZABCDEFGHIJKLM...  (each letter is a byte, 8-bit or 10-bit)
```

Then the incoming data in PCS after 8b10b decoder and before rx_gearbox are:

```
YZABCDEFGHIJKLM....
```

After rx_gearbox, they can become:

1. `{ZY}{BA}{DC}{FE}{HG}{JI}{LK}....`

or

2. `{AZ}{CB}{ED}{GF}{IH}{KJ}{ML}...`

Clearly sequence 2 is not aligned. It has one byte offset, but 16/20-bit alignment is needed. Let’s say the special character 'A' should be always placed in the lower byte.

Flopping one 20-bit data combines with the current 16/20-bit data to form 32/40-bit data as shown below:

1. `{DCBA}{HGFE}{LKJI}....`

```
     ^
     | **Found the A in lower 10-bit, set the offset to '0', send out aligned
     data 'BA'
```

Next clock cycle:

```
{FEDC}{JIHG}{NMLK}....
      ^
      | **send out aligned data 'DC'
```

etc.

After the 16/20-bit alignment, the output data are:

```
{ZY}{BA}{DC}{FE}{HG}{JI}{LK}....
```

2.
```
{CBAZ}{GFED}{KJIH}....
   ^
   | **Found the A in upper 10-bit, set the offset to '10', send out aligned
   data 'BA'
```

Next clock cycle:

```
{EDCB}{IHGF}{MLKJ}....
  ^
  | **send out aligned data 'DC'
```

etc.

After the 20-bit alignment, the output data are:

```
{ZY}{BA}{DC}{FE}{HG}{JI}{LK}....
```

Note: The LSB of a 8/10-bit byte or a 16/20-bit word is always transmitted first and received first.

Switching Between 10XH, 10X and 20X Reference Clock Multiplier Modes Using SCI

Designers can change the bit rates between 10XH, 10X and 20X reference clock multiplier modes using the SERDES Client Interface. This is a useful feature in a system where different rates are received at times and the receiver is able to sweep between different rates and lock to one of them. For example, a system can sweep three different rates of 622Mbps, 1.244Gbps and 2.488Gbps and lock to one of them whichever is received.

There are a few control register bits associated with different rates.

```
CH_0A[D1]: rate_mode_tx
CH_0B[D1]: rate_mode_rx
QD_13[D6]: refclk_mode[0]
QD_18[D5]: quad_rst
```

Switching Between 20X to 20XH or 10X to 10XH Mode in 16-Bit Interface

In addition to the control register settings described above, the transmit interface clock(ff_txiclk_chn) input must be switched from txhalfclk to txqtrclk as described in Table 8-5. ff_rxiclk_chn is not affected in this case. The CDR PLL will automatically tune to the incoming data rate and provide the correct rxhalfclk as described in Table 8-5.

Contact the Lattice Semiconductor Technical Support Group for detailed information.

Off-Chip AC Coupling

When off-chip AC coupling is required, the recommended capacitor values are shown in Table 8-24.

Table 8-24. Off-chip AC Coupling Capacitor Values

Protocol	Min.	Typ.	Max.	Remark	Units
8b10b	4.7	10		@ 3.125Gbps	nF
PCI Express rev1.1	75		200		nF

When a DC balanced pattern is used, such as 8b10b encoding, a capacitor of minimum 4.7nF, is required to reduce the edge degradation.

Data patterns with longer run lengths require larger capacitance values to reduce pattern dependent jitter.

Refer to Lattice technical note TN1114, Electrical Recommendations for Lattice SERDES, for more information.

Unused Quad/Channel and Power Supply

On unused channels, VCCTX, VCCRX, VCCP and VCCAUX33 should be powered up. VCCIB, VCCOB, HDINP/N, HDOUTP/N and REFCLKP/N should be left floating. Unused channel outputs are tristated, with approximately 10 KOhm internal resistor connecting between the differential output pair.

VCCAUX33 supplies power to termination resistors. It is recommended to have the PI filter like the other power supplies. VCCAUX33 noise will directly be coupled to high-speed I/O, HDIN/HDOUT. If VCCAUX(FPGA core power supply) is very clean, it can be connected to VCCAUX33.

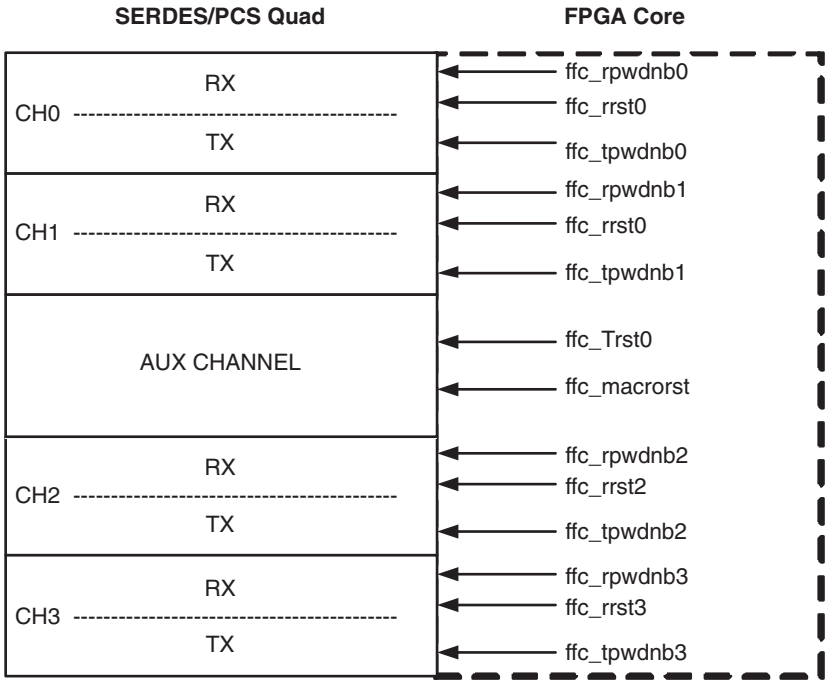
Also, unused SERDES is configured in power down mode by default.

Reset and Power-Down Control

The SERDES quad has reset and power-down controls for the whole macro as well as individual reset and power-down controls for each transmitter and receiver as shown in Figure 8-44. The reset signals are active high and the power-down signals are active low. The operation of the various reset and power-down controls are described in the following sections.

Note: When the device is powering up and the chip level power-on-reset is active, the SERDES control bits (in the PCS) will be cleared (or will take on their default value). This will put the SERDES quad into the power-down state.

Figure 8-44. SERDES/PCS Quad Reset and Power-Down Controls



Reset generation and distribution is handled by the clocks and resets block. Typically, all resets are via hard reset and various FPGA fabric resets. Reset pulse widths are a function of the source. However, all should be at least a clock wide. The reset logic is shown in Figure 8-45 and the corresponding table is in Table 8-25.

Table 8-25. SERDES/PCS Reset Table

Reset Signals		PCS[1] TX	PCS[1] RX	SERDES TX	SERDES RX	PCS CTRL Registers	TX PLL	CDR PLL
FPGA	Control Register							
ffc_lane_tx_rst_ch[3:0]	lane_tx_rst[3:0]	X						
ffc_lane_rx_rst_ch[3:0]	lane_rx_rst[3:0]		X					
ffc_quad_rst	quad_rst	X	X	X	X		X	X
ffc_macro_rst	macro_rst			X	X		X	X
ffc_rrst_ch[3:0][2]	rrst[3:0][2]				X			
ffc_trst_ch[3:0][3]	trst[3:0][3]						X	
TRI_ION (configuration)		X	X	X	X	X		

1. Includes SB (SERDES Bridge), PCS core and FB (FPGA Bridge) sub-blocks.
2. For internal use only. This reset should always be tied to '0'.
3. Only resets TX PLL loss of lock (ffx_plol).

Figure 8-45. SERDES/PCS Reset Diagram

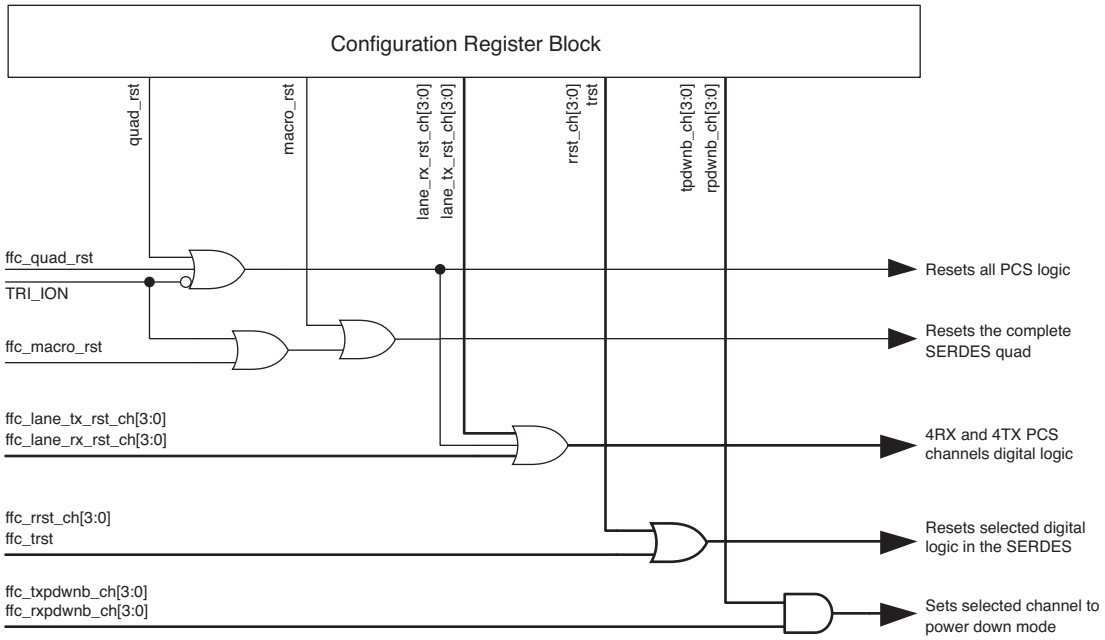


Table 8-26. Reset Controls Description[1, 2, 3]

Rest Signal		Description
FPGA	Control Register	
ffc_quad_rst	quad_rst	Active high, asynchronous input. Resets all SERDES channels including the auxiliary channel and PCS. This reset includes macro_rst, txpll, cdr, lane_tx_rst, and lane_rx_rst.
ffc_macro_rst	macro_rst	Active high, asynchronous input to the SERDES quad. Gated with software register bit. This reset is for the SERDES block only and TXPLL and CDRPLL are included.
ffc_lane_tx/rx_rst_ch[0:3]	lane_tx/rx_rst[0:3]	Active high, asynchronous input. Resets individual TX/RX channel in SB, PCS core and FB blocks.
ffc_rrst_ch[0:3]	rrst[0:3]	Resets loss-of-lock (rlol) and loss-of-signal circuits. Does not reset CDR PLL.
ffc_trst	trst	Resets the loss-of-lock of AUX PLL (plol).

1. For all channels in the quad running in full-data-rate mode, parallel side clocks are guaranteed to be in-phase.
2. For all channels in the quad running in half-data-rate mode, each channel has a separate divide-by-two circuit. Since there is no mechanism in the quad to guarantee that these divide by two circuits are in phase after de-assertion of “macrorst”, the PCS design should assume that the dividers (and therefore the parallel side clocks) are NOT in phase.
3. In half-data-rate mode, since there is no guarantee that the parallel side clocks are in phase, this may add channel-to-channel skew to both transmit and receive sides of a multi-channel link.

Power-Down Controls Description

Each Rx and Tx channel can be individually powered-down by a software register bit or a control signal from the FPGA. The individual channel power-down control bits will only power down selected blocks within the SERDES macro and the high-speed I/O buffers.

Table 8-27. Power-Down Controls Descriptions

Signal		Description
FPGA	Register	
macropdb		Active low asynchronous input to the SERDES quad, acts on all channels including the auxiliary channel. When driven low, it powers down the whole macro including the transmit PLL. All clocks are stopped and the macro power dissipation is minimized.
ffc_txpwdnb_ch[0:3]	tpwdnb[0:3]	Active Low Transmit Channel Power Down – Powers down the serilizer and output driver.
ffc_rxpwdnb_ch[0:3]	rpwdnb[0:3]	Active Low Receive Channel Power Down – Powers down CDR, input buffer (equalizer and amplifier) and loss-of-signal detector.

Table 8-28. Reset Pulse Specification

Parameter	Description	Min.	Typ.	Max.	Units
$t_{MACRORST}$	Macro reset high time	1			µs
t_{RRST}	Channel RX reset high time	3			ns
t_{TRST}	Quad TX reset high time	3			ns

Table 8-29. Power-Down/Power-Up Timing Specification

Parameter	Description	Min.	Typ.	Max.	Unit
t_{PWRDN}	Power-down time after macropdb			10	µs
t_{PWRUP}	Power-up tim after macropdb			100	µs

SERDES/PCS Reset

Reset Sequence and Reset State Diagram

After power-up and configuration, all SERDES resets and FPGA resets are applied.

Lock Status Signals Definitions

ffs_plol : 1 = TX PLL loss of lock.
: 0 = TX PLL lock.
It takes 1,400,000 UI to declare the lock of the TX PLL.

ffs_rlol_ch[3:0] : 1 = CDR loss of lock.
: 0 = Lock maintained.
It takes 400,000 reference clock cycles (worst case) to declare the lock of the CDR PLL.

ffs_rlos_lo_ch[3:0] : 1 = Loss of signal detection for each channel.
: 0 = Signal detected.

The ffs_rlol_ch[3:0] status signal is an indicator of the CDR lock status as defined above. However, during the CDR locking process, the CDR PLL will lock to the reference clock when there is no input data present. This purpose of this feature is to avoid ignoring the input data when it is restored.

In order to ensure the presence of input data during CDR lock status checking, it is recommended to use the ffs_rlos_lo_ch[3:0] signal in conjunction with the ffs_rlol_ch[3:0] signal.

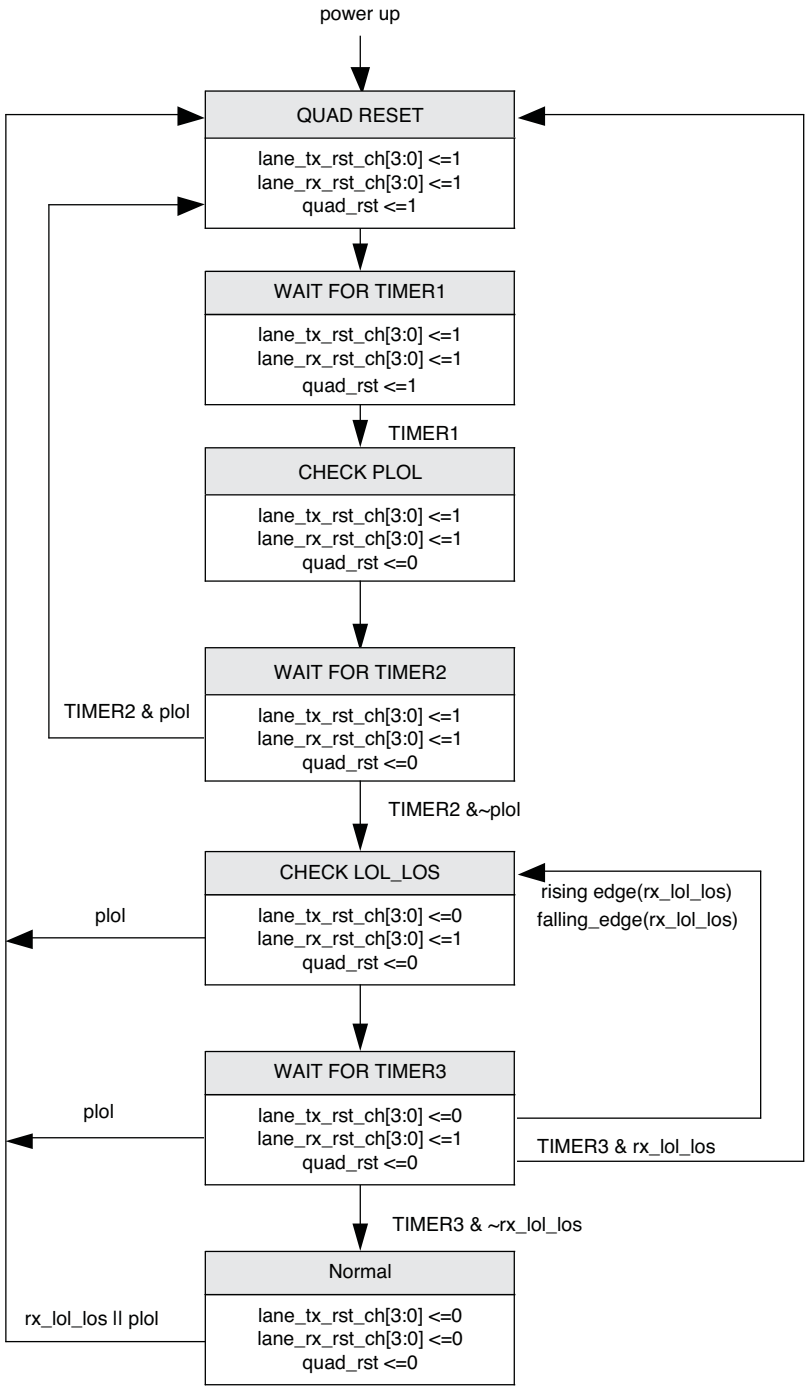
In most applications, combining the two status signals, ffs_rlol_ch[3:0]_s and ffs_rlos_lo_ch[3:0], will work as the indicator of loss of CDR lock. But in applications where high-speed traffic is heavily loaded (i.e. multi-channel, multi-quad), ffs_rlos_lo_ch[3:0] may not represent the correct signal status due to the sensitive nature of the detection circuit. It is strongly recommended to use the protocol-level CDR lock status signal in the RX reset sequence instead of ffs_rlol_ch[3:0].

Reset Sequence

0. QUAD_RESET – At power up, assert ffc_quad_rst, ffc_lane_tx_rst_ch[3:0] and ffc_lane_rx_rst_ch[3:0].
1. WAIT_FOR_TIMER1 – Start TIMER1. Wait a minimum of 20 ns.
2. CHECK_PLOL – Release ffc_quad_rst.
3. WAIT_FOR_TIMER2 – Start TIMER2. If TIMER2 expires and the TX PLL is not locked, go to step 0.
4. CHECK_LOL_LOS – Wait for rx_lol_los[3:0] to go low. Reset TIMER3.
5. WAIT_FOR_TIMER3 – Wait for both ffs_rlol_ch[3:0] and ffs_rlos_lo_ch[3:0] to go low
   If there is a transition in rx_lol_los (ffs_rlol_ch[3:0] || ffs_rlos_lo_ch[3:0]), go to step 4.
   If TIMER2 expires with rx_lol_los = 1, go to step 0.
6. NORMAL – Release ffc_lane_tx_rst_ch[3:0]. If ffs_plol goes high during normal operation, go to step 0.

The reset sequence state diagram is described in Figure 8-46.

Figure 8-46. Reset State Diagram



Notes:
TIMER1: ffc_quad_rst asserted for a minimum of 20 ns.

TIMER2: Time to declare TX PLL lock : 1,400,000 UI.

TIMER 3: Time for rx_lol_los signal to stay low (400,000 reference clock cycles).
Any FPGA clock can be used to satisfy the timer requirement.

In Figure 8-46, rx_lol_los is defined as ffs_rlol_ch[3:0] || ffs_rlos_lo_ch[3:0]. If a protocol-level CDR lock status signal is provided, it can replace the ffs_rlol_ch[3:0] signal.

Power Supply Sequencing Requirements

When using the SERDES with 1.5V VCCIB or VCCOB, the SERDES should not be left in a steady state condition with the 1.5V power applied and the 1.2V power not applied. Both the 1.2V and the 1.5V power should be applied to the SERDES at nominally the same time. The normal variation in ramp_up times of power supplies and voltage regulators is not a concern.

References

- Technical notes:
  - TN1029, FPSC SERDES CML Buffer Interface
  - TN1084, LatticeSC SERDES Jitter
  - TN1114, Electrical Recommendations for Lattice SERDES
  - TN1159, LatticeECP2/M Pin Assignment Recommendations
- HB1003, LatticeECP2/M Family Handbook

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
September 2006	01.0	Initial release.
March 2007	02.0	Re-formatted and updated majority of the document to clarify the SERDES functionality in detail.
August 2007	02.1	SD-SDI and HD-SDI are added in full-support list. Reset sequence, Simulation Consideration, 16-bit word alignment sections added. Plus various updates.
September 2007	02.2	Updated Supported SERDES Standards table.
		Added Simulation of the SERDES/PCS section.
December 2007	02.3	TDRV_DAT_SEL tables updated for ECP2M-35 and the rest of ECP2M families.
February 2008	02.4	Added detailed Reset Sequence diagram.
May 2008	02.5	SCI Address Map updated.
		Clock usage example table updated.
		Control register CH_07[2:0] corrected.
August 2008	02.6	Updated High Speed I/O Terminations figure.
October 2008	02.7	Updated Serializer/Deserializer Blocks latency
		Corrected footnote 5 in the SERDES_PCS I/O Descriptions table.
		Changed title of Per Channel Register Settings for Different Standards table to K Characters Recognized for Different Standards.
November 2008	02.8	Detailed definitions of reset signals added.
January 2009	02.9	Removed references to PIPE Mode.
March 2009	03.0	Mode-specific control and status signals are now arranged in one table (Data Bus Usage by Mode table).
		In all user modes, CTC is bypassed by hardware (SERDES/PCS GUI - PCS Advanced Setup Tab table).
May 2009	03.1	Word Aligner Latency value updated.
July 2009	03.2	ffc_txpwdnb description corrected. rlol settling time changed to 4ms.
February 2010	03.3	SERDES/PCS reset sequence updated.
June 2010	03.4	Added Appendix E.
September 2011	03.5	SERDES_PCI I/O Descriptions table – Updated description for ffc_macro_rst signal.
		SERDES/PCS Reset diagram – Replaced lowest OR gate with an AND gate.
		Reset Pulse Specification table – Units for $t_{MACRORST}$ changed from ns to µs.
		SERDES Control Register 3 (QD_13) table – Default for CDR_LOL_SET [1:0] bit changed from 2'b10 to 2'b00.

Appendix A. Memory Map

Configuration Register Definition

There are specific quad-level registers and channel-level registers. In each category, there are SERDES specific registers and PCS specific registers. Within these sub-categories there are:

- Control registers
- Status registers
- Status registers with clear-on-read
- Interrupt Control registers
- Interrupt Status registers
- Interrupt Source registers (clear-on-read)

All register bits shown below are with D0 as the LEAST significant bit on the right.

The following indications are the nomenclature for what types of register each is: (R/W = Read/Write, RO = Read Only, CR = Clear on a read).

Each of these register bits is “shadowed” with a memory cell. These memory cells are only programmable through bitstream control. After configuration is complete, the configuration memory cells that have associated registers are automatically copied into the registers. Subsequent changes to the contents of the registers will not affect the value stored in the configuration memory cells but will change the operation of the SERDES/PCS quad.

All “reserved” bits are written to zero.

Per Quad Register Overview

Table 8-30. Quad Interface Register Map

BA[1]	Register Name	D7	D6	D5	D4	D3	D2	D1	D0
1. PER QUAD CONTROL REGISTERS (28)									
Per Quad PCS Control Registers (17)									
00	qd_00	sync_toggle	force_int	char_mode	xge_mode	rio_mode	pcie_mode	fc_mode	uc_mode[5]
01	qd_01	bist_rpt_ch_sel[1][6]	bist_rpt_ch_sel[0]	bist_res_sel[1]	bist_res_sel[0]	bist_time_sel[1]	bist_time_sel[0]	bist_head_sel[1]	bist_head_sel[0]
02	qd_02	high_mark[3]	high_mark[2]	high_mark[1]	high_mark[0]	low_mark[3]	low_mark[2]	low_mark[1]	low_mark[0]
03	qd_03	min_ipg_cnt[1]	min_ipg_cnt[0]	match_4_enable	match_2_enable		pfifo_clr_sel	asyn_mode	sel_test_clk
04	qd_04	cc_match_1[7]	cc_match_1[6]	cc_match_1[5]	cc_match_1[4]	cc_match_1[3]	cc_match_1[2]	cc_match_1[1]	cc_match_1[0]
05	qd_05	cc_match_2[7]	cc_match_2[6]	cc_match_2[5]	cc_match_2[4]	cc_match_2[3]	cc_match_2[2]	cc_match_2[1]	cc_match_2[0]
06	qd_06	cc_match_3[7]	cc_match_3[6]	cc_match_3[5]	cc_match_3[4]	cc_match_3[3]	cc_match_3[2]	cc_match_3[1]	cc_match_3[0]
07	qd_07	cc_match_4[7]	cc_match_4[6]	cc_match_4[5]	cc_match_4[4]	cc_match_4[3]	cc_match_4[2]	cc_match_4[1]	cc_match_4[0]
08	qd_08	cc_match_4[9]	cc_match_4[8]	cc_match_3[9]	cc_match_3[8]	cc_match_2[9]	cc_match_2[8]	cc_match_1[9]	cc_match_1[8]
09	qd_09	udf_comma_mask[7]	udf_comma_mask[6]	udf_comma_mask[5]	udf_comma_mask[4]	udf_comma_mask[3]	udf_comma_mask[2]	udf_comma_mask[1]	udf_comma_mask[0]
0A	qd_0a	udf_comma_a[7]	udf_comma_a[6]	udf_comma_a[5]	udf_comma_a[4]	udf_comma_a[3]	udf_comma_a[2]	udf_comma_a[1]	udf_comma_a[0]
0B	qd_0b	udf_comma_b[7]	udf_comma_b[6]	udf_comma_b[5]	udf_comma_b[4]	udf_comma_b[3]	udf_comma_b[2]	udf_comma_b[1]	udf_comma_b[0]
0C	qd_0c	udf_comma_a[9]	udf_comma_a[8]	udf_comma_b[9]	udf_comma_b[8]	udf_comma_mask[9]	udf_comma_mask[8]	bist_mode	bist_en
0D	qd_0d	bist_udf_def_header [7]	bist_udf_def_header [6]	bist_udf_def_header [5]	bist_udf_def_header [4]	bist_udf_def_header [3]	bist_udf_def_header [2]	bist_udf_def_header [1]	bist_udf_def_header [0]
0E	qd_0e	bist_udf_def_header [15]	bist_udf_def_header [14]	bist_udf_def_header [13]	bist_udf_def_header [12]	bist_udf_def_header [11]	bist_udf_def_header [10]	bist_udf_def_header [9]	bist_udf_def_header [8]
0F	qd_0f	bist_bus8bit_sel	bist_ptn_sel[2]	bist_ptn_sel[1]	bist_ptn_sel[0]	bist_udf_def_header [19]	bist_udf_def_header [18]	bist_udf_def_header [17]	bist_udf_def_header [16]
10	qd_int_10[2]	ls_sync_status_3_int _ctl	ls_sync_status_2_int _ctl	ls_sync_status_1_int _ctl	ls_sync_status_0_int _ctl	ls_sync_statusn_3_ int_ctl	ls_sync_statusn_2_ int_ctl	ls_sync_statusn_1_ int_ctl	ls_sync_statusn_0_ int_ctl
Per Quad SERDES Control Registers (6)									
11	qd_11	reserved	reserved	tx_refck_sel	refck_dcc_en	refck_rterm	refck_out_sel[2]	refck_out_sel[1]	refck_out_sel[0]
12	qd_12	refck25x	bus8bit_sel	rlos_hset[2]	rlos_hset[1]	rlos_hset[0]	rlos_lset[2]	rlos_lset[1]	rlos_lset[0]
13	qd_13	refck_mode[1]	refck_mode[0]	reserved	reserved	reserved	reserved	cdr_lol_set[1]	cdr_lol_set[0]
14	qd_14	reserved	reserved	reserved	pll_lol_set[1]	pll_lol_set[0]	tx_vco_ck_div[2]	tx_vco_ck_div[1]	tx_vco_ck_div[0]
15	qd_15	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
16	qd_int_16[2]	plol_int_ctl	~plol_int_ctl	reserved	reserved	reserved	reserved	reserved	reserved
Per Quad Clock Reset Registers (5)									
17	qd_17	lane_rx_rst3	lane_rx_rst2	lane_rx_rst1	lane_rx_rst0	lane_tx_rst3	lane_tx_rst2	lane_tx_rst1	lane_tx_rst0
18	qd_18	macropdb	macro_rst	quad_rst	trst	rrst[3]	rrst[2]	rrst[1]	rrst[0]
19	qd_19	bist_rx_data_sel	bist_bypass_tx_gate	bist_sync_head_req [1]	bist_sync_head_req [0]	sel_sd_rx_clk3	sel_sd_rx_clk2	sel_sd_rx_clk1	sel_sd_rx_clk0
1A	qd_1a	ff_rx_clk_sel_2[3]	ff_rx_clk_sel_2[2]	ff_rx_clk_sel_2[1]	ff_rx_clk_sel_2[0]	ff_rx_clk_sel_1[3]	ff_rx_clk_sel_1[2]	ff_rx_clk_sel_1[1]	ff_rx_clk_sel_1[0]
1B	qd_1b	ff_tx_clk_sel[2]	ff_tx_clk_sel[1]	ff_tx_clk_sel[0]	reserved	ff_rx_clk_sel_0[3]	ff_rx_clk_sel_0[2]	ff_rx_clk_sel_0[1]	ff_rx_clk_sel_0[0]
2. PER QUAD STATUS REGISTERS (9)									
Per Quad PCS Status Registers (5)									
20	qd_20				int_qua_out	int_cha[3]	int_cha[2]	int_cha[1]	int_cha[0]
21	qd_21[3]	ls_sync_status_3	ls_sync_status_2	ls_sync_status_1	ls_sync_status_0	ls_sync_statusn_3	ls_sync_statusn_2	ls_sync_statusn_1	ls_sync_statusn_0
22	qd_int_22[4]	ls_sync_status_3_int	ls_sync_status_2_int	ls_sync_status_1_int	ls_sync_status_0_int	ls_sync_statusn_3_ int	ls_sync_statusn_2_ int	ls_sync_statusn_1_ int	ls_sync_statusn_0_ int
23	qd_23	bist_report[7]	bist_report[6]	bist_report[5]	bist_report[4]	bist_report[3]	bist_report[2]	bist_report[1]	bist_report[0]
24	qd_24	bist_report[15]	bist_report[14]	bist_report[13]	bist_report[12]	bist_report[11]	bist_report[10]	bist_report[9]	bist_report[8]
Per Quad SERDES Status Registers (4)									
25	qd_25[3]	plol	~plol	reserved	reserved	reserved	reserved	reserved	reserved
26	qd_int_26[4]	plol_int	~plol_int	reserved	reserved	reserved	reserved	reserved	reserved
27	qd_27	reserved	reserved	pll_calib_status[5]	pll_calib_status[4]	pll_calib_status[3]	pll_calib_status[2]	pll_calib_status[1]	pll_calib_status[0]
28	qd_28	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved

1. BA = Base Address (Hex)
2. Interrupt control register related to an interruptible status (int_sts_x) register.
3. Status register which has an associated interruptible status (int_sts_x) register.
4. Interruptible status register; clear on read (has associated control register and status register)
5. uc_mode: 8-bit SERDES Only and 10-bit SERDES Only.
6. BIST is a built-in PRBS generator and checker for internal use only.

Note: Default value is "0", unless otherwise specified.

Per Quad PCS Control Register Details

Table 8-31. PCS Control Register 1 (QD_00)

Bit	Name	Description	Type	Default
7	sync_toggle	Transition = Reset the 4 Tx Serializer to minimize Tx Lane-to-Lane Skew Level = Normal operation of Tx Serializers		
6	force_int	1 = Force to generate interrupt signal 0 = Normal operation	R/W	0
5	char_mode	1 = Enable SERDES characterization mode 0 = Disable SERDES characterization mode	R/W	0
4	xge_mode	1 = Selects 10Gb Ethernet 0 = Selects 1Gb Ethernet mode	R/W	0
3	rio_mode	1 = Selects Rapid-IO mode 0 = Selects other mode (10GbE, 1GbE)	R/W	0
2	pcie_mode	1 = PCI Express mode of operation 0 = Selects other mode (RapidIO, 10GbE, 1GbE)	R/W	0
1	fc_mode	1 = Select Fibre Channel mode 0 = Selects other mode (PCI Express, RapidIO, 10GbE, 1GbE)	R/W	0
0	uc_mode	1 = Selects User Configured mode 0 = Selects other mode (PCI Express, RapidIO, 10GbE, 1GbE)		

Note: Bits 0 to 3 are mutually exclusive. Only one of the bits can be set.

Table 8-32. PCS Control Register 2 (QD_01)

Bit	Name	Description	Type	Default
7:6	bist_rpt_ch_sel [1:0]	00 = BIST report from channel 0 01 = BIST report from channel 1 10 = BIST report from channel 2 11 = BIST report from channel 3	R/W	00
5:4	bist_res_sel [1:0]	BIST resolution selection 00 = no error 01 < 2 errors 10 < 16 errors 11 < 128 errors	R/W	00
3:2	bist_time_sel [1:0]	BIST time selection: 00 = 5e+8 cycles 01 = 5e+9 cycles 10 = 5e+6 cycles 11 = 100K cycles	R/W	00
1:0	bist_head_sel [1:0]	BIST header selection 00 = K28_5 (K28_5=10'h305 K28_5_=10'h0FA) 01 = A1A2 (MA1=10'h1F6; MA2=10'h128) 10 = 10'h1BC 11 = user defined	R/W	0

Table 8-33. PCS Control Register 3 (QD_02)

Bit	Name	Description	Type	Default
7:4	high_mark [3:0]	Clock compensation FIFO high water mark. Mean is 4'b1000	R/W	4'b0111
3:0	low_mark [3:0]	Clock compensation FIFO low water mark. Mean is 4'b1000	R/W	4'b1001

Table 8-34. PCS Control Register 4 (QD_03)

Bit	Name	Description	Type	Default
7:6	min_ipg_cnt [1:0]	Minimum IPG to enforce	R/W	2'b11
5	match_4_enable	1 = enable four character skip matching (using match 4, 3, 2, 1)	R/W	0
4	match_2_enable	1 = enable two character skip matching (using match 4,3)	R/W	1
3	Reserved			
2	pfifo_clr_sel	1 = pfifo_clr signal or channel register bit clears the FIFO 0 = pfifo_error internal signal self clears the FIFO	R/W	0
1	asyn_mode	For test only, select asynchronous reset	R/W	0
0	sel_test_clk	For test only, select test clock	R/W	0

Table 8-35. PCS Control Register 5 - CC match 1 LO (QD_04)

Bit	Name	Description	Type	Default
7:0	cc_match_1 [7:0]	Lower bits of user defined clock compensator skip pattern 1	R/W	8'h00

Table 8-36. PCS Control Register 6 - CC match 2 LO (QD_05)

Bit	Name	Description	Type	Default
7:0	cc_match_2 [7:0]	Lower bits of user defined clock compensator skip pattern 2	R/W	8'h00

Table 8-37. PCS Control Register 7 - CC match 3 LO (QD_06)

Bit	Name	Description	Type	Default
7:0	cc_match_3 [7:0]	Lower bits of user defined clock compensator skip pattern 3	R/W	8'hBC

Table 8-38. PCS Control Register 8 - CC match 4 LO (QD_07)

Bit	Name	Description	Type	Default
7:0	cc_match_4 [7:0]	Lower bits of user defined clock compensator skip pattern 4	R/W	8'h50

Table 8-39. PCS Control Register 9 - CC match HI (QD_08)

Bit	Name	Description	Type	Default
7:6	cc_match_4 [9:8]	Upper bits of user defined clock compensator skip pattern 4 [9] = Disparity error [8] = K control	R/W	2'b01
5:4	cc_match_3 [9:8]	Upper bits of user defined clock compensator skip pattern 3 [9] = Disparity error [8] = K control	R/W	2'b01
3:2	cc_match_2 [9:8]	Upper bits of user defined clock compensator skip pattern 2 [9] = Disparity error [8] = K control	R/W	2'b00
1:0	cc_match_1 [9:8]	Upper bits of user defined clock compensator skip pattern 1 [9] = Disparity error [8] = K control	R/W	2'b00

Table 8-40. PCS Control Register 10 - UDF comma mask LO (QD_09)

Bit	Name	Description	Type	Default
7:0	udf_comma_mask [7:0]	Lower bits of user defined comma mask	R/W	8'hFF

Table 8-41. PCS Control Register 11 - UDF comma a LO (QD_0A)

Bit	Name	Description	Type	Default
7:0	udf_comma_a [7:0]	Lower bits of user defined comma character ‘a’	R/W	8’h83

Table 8-42. PCS Control Register 12 - UDF comma b LO (QD_0B)

Bit	Name	Description	Type	Default
7:0	udf_comma_b [7:0]	Lower bits of user defined comma character ‘b’	R/W	8’h7C

Table 8-43. PCS Control Register 13 - UDF comma HI (QD_0C)

Bit	Name	Description	Type	Default
7:6	udf_comma_a [9:8]	Upper bits of user defined comma character ‘a’	R/W	2’b10
5:4	udf_comma_b [9:8]	Upper bits of user defined comma character ‘b’	R/W	2’b01
3:2	udf_comma_mask [9:8]	Upper bits of user defined comma mask	R/W	2’b11
1	bist_mode	1 = Continuous Bist Mode 0 = Timed BIST mode	R/W	0
0	bist_en	1 = Enable PCS BIST 0 = Normal operation.	R/W	0

Table 8-44. PCS Control Register 14 - UDF BIST header LO (QD_0D)

Bit	Name	Description	Type	Default
7:0	bist_udf_def_header [7:0]	Lower bits of user defined header for BIST	R/W	8’h00

Table 8-45. PCS Control Register 15 - UDF BIST header MD (QD_0E)

Bit	Name	Description	Type	Default
7:0	bist_udf_def_header [15:8]	Middle bits of user defined header for BIST	R/W	8’h00

Table 8-46. PCS Control Register 16 - UDF BIST header HI (QD_0F)

Bit	Name	Description	Type	Default
7	bist_bus8bit_sel	1 = 8 bit data BIST 0 = 10 bit data BIST	R/W	0
6:4	bist_ptn_sel[2:0]	BIST pattern selection: 000 = PRBS11 001 = max data rate 010 = PRBS31 011 = PRBS21100 = K28_5 101 = A1A2110 = 5150 (repeat) 111 = 2120 (repeat)	R/W	0
3:0	bist_udf_def_header [19:16]	High bits of user defined header for BIST	R/W	8’h00

Table 8-47. PCS Interrupt Control Register 17 (QD_10)

Bit	Name	Description	Type	Default
7	ls_sync_status_3_int_ctl	1 = Enable interrupt for ls_sync_status_3 (in sync) 0 = Disable interrupt for ls_sync_status_3 (in sync)	R/W	0
6	ls_sync_status_2_int_ctl	1 = Enable interrupt for ls_sync_status_2 (in sync) 0 = Disable interrupt for ls_sync_status_2 (in sync)	R/W	0
5	ls_sync_status_1_int_ctl	1 = Enable interrupt for ls_sync_status_1 (in sync) 0 = Disable interrupt for ls_sync_status_1 (in sync)	R/W	0
4	ls_sync_statusn_3_int_ctl	1 = Enable interrupt for ls_sync_status_3 when it goes low (out of sync) 0 = Disable interrupt for ls_sync_status_3 when it goes low (out of sync)	R/W	0
4	ls_sync_status_0_int_ctl	1 = Enable interrupt for ls_sync_status_0 (in sync) 0 = Disable interrupt for ls_sync_status_0 (in sync)	R/W	0
3	ls_sync_statusn_2_int_ctl	1 = Enable interrupt for ls_sync_status_2 when it goes low (out of sync) 0 = Disable interrupt for ls_sync_status_2 when it goes low (out of sync)	R/W	0
1	ls_sync_statusn_1_int_ctl	1 = Enable interrupt for ls_sync_status_1 when it goes low (out of sync) 0 = Disable interrupt for ls_sync_status_1 when it goes low (out of sync)	R/W	0
0	ls_sync_statusn_0_int_ctl	1 = Enable interrupt for ls_sync_status_0 when it goes low (out of sync) 0 = Disable interrupt for ls_sync_status_0 when it goes low (out of sync)	R/W	0

Per Quad SERDES Control Register Details

Note: Except indicated, all channels must be reset after writing any SERDES control register.

Table 8-48. SERDES Control Register 1 (QD_11)

Bit	Name	Description	Type	Default
7:6	Reserved			
5	TX_REFCK_SEL	TXPLL reference clock select 0 = refclk (differential input) 1 = core_txrefclk	R/W	0
4	REFCK_DCC_EN	1 = Reference clock DC coupling enable 0 = Reference clock DC coupling disable (default)	R/W	0
3	REFCK_RTERM	Termination at reference clock input buffer 0 = 50 ohm (default) 1 = high impedance	R/W	0
2:0	REFCL_OUT_SEL[2:0]	Refck control [0]: 0 = refck buffer enable, 1 = refck buffer disable [1]: 0 = refck2core disable, 1 = refck2core enable [2]: reserved Refer to Figure 8-47.	R/W	3'b000

Figure 8-47. Reference Clock Select Control

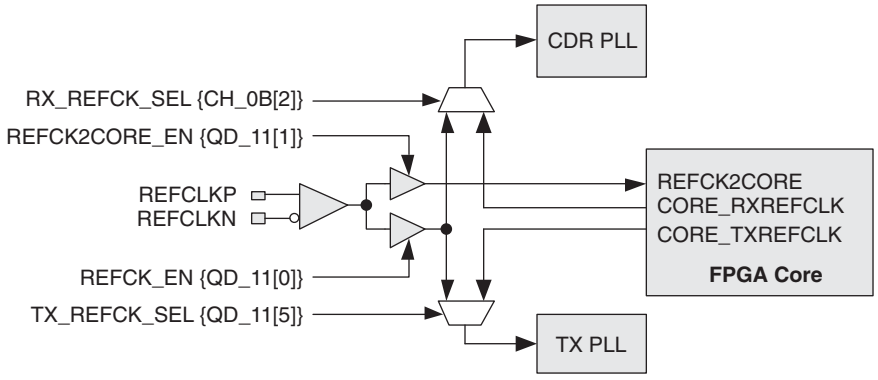


Table 8-49. SERDES Control Register 2 (QD_12)

Bit	Name	Description	Type	Default
7	REFCK25X	1 = Internal high speed bit clock is 25x (for reference clock = 100MHz only) 0 = See REFCK_MODE	R/W	0
6	BUS8BIT_SEL	1 = Select 8-bit bus width 0 = Select 10-bit bus width (default)	R/W	0
5:3	RLOS_HSET [2:0]	LOS detector reference current adjustment for larger swing 000 = default 001 = +10% 010 = +15% 011 = +25% 100 = -10% 101 = -15% 110 = -25% 111 = -30%	R/W	3'b000
2:0	RLOS_LSET [2:0]	(Internal use only)	R/W	3'b000

Table 8-50. SERDES Control Register 3 (QD_13)

Bit	Name	Description	R/W	Default
7:6	REFCK_MODE [1:0]	00 = Internal high speed bit clock is 20x or 16x 01 = Internal high speed bit clock is 10x or 8x 1X = Reserved	R/W	2'b00
5:2	Reserved			
1:0	CDR_LOL_SET [1:0]	CDR loss of lock setting: Lock / Unlock 00 = +/-1000ppm x2 / +/-1500ppm x2 01 = +/-2000ppm x2 / +/-2500ppm x2 10 = +/-4000ppm / +/-7000ppm 11 = +/-300ppm / +/-450ppm	R/W	2'b00[1]

1. This is the default value used in ispLEVER 8.1.

Table 8-51. SERDES Control Register 4 (QD_14)

Bit	Name	Description	R/W	Default
7:5	Reserved			
4:3	PLL_LOL_SET [1:0]	TxPLL loss of lock setting Lock / Unlock 00 = +/-300ppm x2 / +/-600ppm x2 01 = +/-300ppm / +/-2000ppm 10 = +/-1500ppm / +/-2200ppm 11 = +/-4000ppm / +/-6000ppm	R/W	2'b00
2:0	TX_VCO_CK_DIV[2:0]	VCO output frequency select: 000 = divided by 1 / 001 = reserved 010 = divided by 2 / 011 = reserved 100 = divided by 4 / 101 = divided by 8 110 = divided by 16 / 111 = divided by 32	R/W	3'b000

Table 8-52. SERDES Control Register 5 (QD_15)

Bit	Name	Description	R/W	Default
7:0	Reserved			

Table 8-53. SERDES Interrupt Control Register 6 (QD_16)

Bit	Name	Description	R/W	Default
7	PLOL_INT_CTL	1 = Interrupt enabled for loss of lock on PLOL. 0 = Interrupt not enabled for loss of lock PLOL.	RO CR	0
6	~PLOL_INT_CTL	1 = Interrupt enabled for obtaining lock on PLOL. 0 = Interrupt not enabled for obtaining lock PLOL.	RO CR	0
5:0	Reserved			

Per Quad Reset and Clock Control Register Details

Table 8-54. Reset and Clock Control Register 1 (QD_17)

Bit	Name	Description	R/W	Default
7	lane_rx_rst3	1 = Assert reset signal to channel 3 receive logic	R/W	0
6	lane_rx_rst2	1 = Assert reset signal to channel 2 receive logic	R/W	0
5	lane_rx_rst1	1 = Assert reset signal to channel 1 receive logic	R/W	0
4	lane_rx_rst0	1 = Assert reset signal to channel 0 receive logic	R/W	0
3	lane_tx_rst3	1 = Assert reset signal to channel 3 transmit logic	R/W	0
2	lane_tx_rst2	1 = Assert reset signal to channel 2 transmit logic	R/W	0
1	lane_tx_rst1	1 = Assert reset signal to channel 1 transmit logic	R/W	0
0	lane_tx_rst0	1 = Assert reset signal to channel 0 transmit logic	R/W	0

Table 8-55. Reset and Clock Control Register 2 (QD_18)

Bit	Name	Description	R/W	Default
7	macropdb	0 = Assert power down	R/W	1
6	macro_rst	1 = Assert macro reset	R/W	0
5	quad_rst	1 = Assert quad reset	R/W	0
4	trst	1 = Tx Reset	R/W	0
3:0	rrst [3:0]	1 = Rx channel-based reset	R/W	0

Table 8-56. Reset and Clock Control Register 3 (QD_19)

Bit	Name	Description	R/W	Default
7	bist_rx_data_sel	0 = data from SERDES 1 = data from after 8b10b decoder	R/W	0
6	bist_bypass_tx_gate	0 = start to send BIST data after finding the head 1 = force to send BIST data whether the head is found or not	R/W	0
5:4	bist_sync_head_req [1:0]	BIST sync header counter selection 00 = 5'd5 01 = 5'd8 10 = 5'd14 11 = 5'd24	R/W	00
3	sel_sd_rx_clk3	1 = Select rx_clk3[1] for DS FIFO write clock 0 = Select ff_ebrd_clk_3 for DS FIFO write clock	R/W	0
2	sel_sd_rx_clk2	1 = Select rx_clk2 for DS FIFO write clock 0 = Select ff_ebrd_clk_2 for DS FIFO write clock	R/W	0
1	sel_sd_rx_clk1	1 = Select rx_clk1 for DS FIFO write clock 0 = Select ff_ebrd_clk_1 for DS FIFO write clock	R/W	0
0	sel_sd_rx_clk0	1 = Select rx_clk0 for DS FIFO write clock 0 = Select ff_ebrd_clk_0 for DS FIFO write clock	R/W	0

1. rx_clk3 is channel3 recovered clock.

Table 8-57. Reset and Clock Control Register 4 (QD_1A)

Bit	Name	Description	R/W	Default
7	ff_rx_clk_sel3_2	1 = Disable ff_rx_f_clk for channel 3	R/W	0
6	ff_rx_clk_sel2_2	1 = Disable ff_rx_f_clk for channel 2	R/W	0
5	ff_rx_clk_sel1_2	1 = Disable ff_rx_f_clk for channel 1	R/W	0
4	ff_rx_clk_sel0_2	1 = Disable ff_rx_f_clk for channel 0	R/W	0
3	ff_rx_clk_sel3_1	1 = Enable ff_rx_h_clk for channel 3	R/W	0
2	ff_rx_clk_sel2_1	1 = Enable ff_rx_h_clk for channel 2	R/W	0
1	ff_rx_clk_sel1_1	1 = Enable ff_rx_h_clk for channel 1	R/W	0
0	ff_rx_clk_sel0_1	1 = Enable ff_rx_h_clk for channel 0	R/W	0

Table 8-58. Reset and Clock Control Register 5 (QD_1B)

Bit	Name	Description	R/W	Default
7	ff_tx_clk_sel2	1 = Disable ff_tx_f_clk for quad	R/W	0
6	ff_tx_clk_sel1	1 = Enable ff_tx_h_clk for quad	R/W	1
5	ff_tx_clk_sel0	1 = Enable ff_tx_q_clk for quad	R/W	0
4	Reserved			
3	ff_rx_clk_sel3_0	1 = Enable ff_rx_q_clk for channel 3	R/W	0
2	ff_rx_clk_sel2_0	1 = Enable ff_rx_q_clk for channel 2	R/W	0
1	ff_rx_clk_sel1_0	1 = Enable ff_rx_q_clk for channel 1	R/W	0
0	ff_rx_clk_sel0_0	1 = Enable ff_rx_q_clk for channel 0	R/W	0

Per Quad PCS Status Register Details

Table 8-59. PCS Status Register 1 (QD_20)

Bit	Name	Description	R/W	Int?
7:6	Reserved			
5	ion_delay	Delayed global resetn from tri_ion	RO	N
4	int_qua_out	Per Quad Interrupt status	RO	N
3:0	int_cha_out [3:0]	Per Channel Interrupt status	RO	N

Table 8-60. PCS Status Register 2 (QD_21)

Bit	Name	Description	R/W	Int?
7	ls_sync_status_3	1 = Alarm generated on sync_status_3. 0 = Alarm not generated on sync_status_3.	RO	Y
6	ls_sync_status_2	1 = Alarm generated on sync_status_2. 0 = Alarm not generated on sync_status_2.	RO	Y
5	ls_sync_status_1	1 = Alarm generated on sync_status_1. 0 = Alarm not generated on sync_status_1.	RO	Y
4	ls_sync_status_0	1 = Alarm generated on sync_status_0. 0 = Alarm not generated on sync_status_0.	RO	Y
3	ls_sync_statusn_3	1 = Alarm generated on sync_status_3 when it goes low (out of sync) 0 = Alarm not generated on sync_status_3 when it goes low (out of sync)	RO	Y
2	ls_sync_statusn_2	1 = Alarm generated on sync_status_2 when it goes low (out of sync) 0 = Alarm not generated on sync_status_2 when it goes low (out of sync)	RO	Y
1	ls_sync_statusn_1	1 = Alarm generated on sync_status_1 when it goes low (out of sync) 0 = Alarm not generated on sync_status_1 when it goes low (out of sync)	RO	Y
0	ls_sync_statusn_0	1 = Alarm generated on sync_status_0 when it goes low (out of sync) 0 = Alarm not generated on sync_status_0 when it goes low (out of sync)	RO	Y

Table 8-61. Packet Interrupt Status Register 3 (QD_22)

Bit	Name	Description	R/W	Int?
7	ls_sync_status_3_int	1 = Interrupt generated on sync_status_0 (in sync) 0 = Interrupt not generated on sync_status_0 (in sync)	RO CR	Y
6	ls_sync_status_2_int	1 = Interrupt generated on sync_status_1 (in sync) 0 = Interrupt not generated on sync_status_1 (in sync)	RO CR	Y
5	ls_sync_status_1_int	1 = Interrupt generated on sync_status_2 (in sync) 0 = Interrupt not generated on sync_status_2 (in sync)	RO CR	Y
4	ls_sync_status_0_int	1 = Interrupt generated on sync_status_3 (in sync) 0 = Interrupt not generated on sync_status_3 (in sync)	RO CR	Y
3	ls_sync_statusn_3_int	1 = Interrupt generated on sync_status_3 when it goes low (out of sync) 0 = Interrupt not generated on sync_status_3 when it goes low (out of sync)	RO CR	Y
2	ls_sync_statusn_2_int	1 = Interrupt generated on sync_status_2 when it goes low (out of sync) 0 = Interrupt not generated on sync_status_2 when it goes low (out of sync)	RO CR	Y
1	ls_sync_statusn_1_int	1 = Interrupt generated on sync_status_1 when it goes low (out of sync) 0 = Interrupt not generated on sync_status_1 when it goes low (out of sync)	RO CR	Y
0	ls_sync_statusn_0_int	1 = Interrupt generated on sync_status_0 when it goes low (out of sync) 0 = Interrupt not generated on sync_status_0 when it goes low (out of sync)	RO CR	Y

Table 8-62. PCS BIST Status Register 4 (QD_23)

Bit	Name	Description	R/W	Int?
7:0	bist_report [7:0]	Lower bits of BIST report	RO	N

Table 8-63. PCS BIST Status Register 5 (QD_24)

Bit	Name	Description	R/W	Int?
7:0	bist_report [15:8]	Higher bits of BIST report	RO	N

Per Quad SERDES Status Register Details

Table 8-64. SERDES Status Register 1 (QD_25)

Bit	Name	Description	R/W	Int?
7	PLOL	1 = PLL Loss of lock	RO	Y
6	~PLOL	1 = PLL lock obtained	RO	Y
5:0	Reserved			

Table 8-65. SERDES Interrupt Status Register 2 (QD_26)

Bit	Name	Description	R/W	Int?
7	PLOL_INT	1 = Interrupt generated on PLOL. 0 = Interrupt not generated PLOL.	RO CR	Y
6	~PLOL_INT	1 = Interrupt generated on ~PLOL. 0 = Interrupt not generated ~PLOL.	RO CR	Y
5:0	Reserved			

Table 8-66. SERDES Status Register 3 (QD_27)

Bit	Name	Description	R/W	Int?
7:6	Reserved			
5:0	PLL_CALIB_ STATUS[5:0]	TxPLL VCO calibration status output	RO	N

Table 8-67. SERDES Status Register 4 (QD_28)

Bit	Name	Description	R/W	Int?
7:0	Reserved			

Per Channel Register Overview

Table 8-68. Channel Interface Register Map

BA[1]	Register Name	D7	D6	D5	D4	D3	D2	D1	D0
1. CONTROL REGISTERS (13)									
Per Channel General Registers (7)									
00	ch_00	enable_cg_align	prbs_enable[6]	prbs_lock	ge_an_enable			invert_tx	invert_rx
01	ch_01	pfifo_clr	pcie_ei_en	pcs_det_time_sel[1]	pcs_det_time_sel[0]	rx_gear_mode	tx_gear_mode	rx_ch	tx_ch
02	ch_02	bus_width8	sb_bypass	sb_pfifo_lp	sb_bist_sel	enc_bypass	sel_bist_txd4enc	tx_gear_bypass	fb_loopback
03	ch_03	lsm_sel	signal_detect	rx_gear_bypass	ctc_bypass	dec_bypass	wa_bypass	rx_sb_bypass	sb_loopback
04	ch_int_04[2]					cc_underrun_int_ctl	cc_overrun_int_ctl	fb_rx_fifo_error_int_ctl	fb_tx_fifo_error_int_ctl
05	ch_05								los_hi_sel
06	ch_06								
Per Channel SERDES Registers (6)									
07	ch_07	req_en	req_lvl_set	rcv_dcc_en	rate_sel[1]	rate_sel[0]	rx_dco_ck_div[2]	rx_dco_ck_div[1]	rx_dco_ck_div[0]
08	ch_08	lb_ctl[3]	lb_ctl[2]	lb_ctl[1]	lb_ctl[0]	rterm_rxadj[1]	rterm_rxadj[0]	rterm_rx[1]	rterm_rx[0]
09	ch_09	tdrv_amp[2]	tdrv_amp[1]	tdrv_amp[0]	tdrv_pre_set[2]	tdrv_pre_set[1]	tdrv_pre_set[0]	tdrv_dat_sel[1]	tdrv_dat_sel[0]
0A	ch_0a				tdrv_pre_en	rterm_tx[1]	rterm_tx[0]	rate_mode_tx	tpwdnb
0B	ch_0b				oob_en	rx_refck_sel[1]	rx_refck_sel[0]	rate_mode_rx	rpwdnb
0C	ch_int_0c[2]		pci_det_done_int_ctl	rlos_lo_int_ctl	~rlos_lo_int_ctl	rlos_hi_int_ctl	~rlos_hi_int_ctl	rlol_int_ctl	~rlol_int_ctl
2. STATUS REGISTERS (13)									
Per Channel General Registers (6)									
20	ch_20[3]					cc_underrun	cc_overrun	fb_rx_fifo_error	fb_tx_fifo_error
21	ch_21[5]	prbs_error_cnt[7]	prbs_error_cnt[6]	prbs_error_count[5]	prbs_error_cnt[4]	prbs_error_cnt[3]	prbs_error_cnt[2]	prbs_error_cnt[1]	prbs_error_cnt[0]
22	ch_22								
23	ch_int_23	fb_tx_fifo_error_int	fb_rx_fifo_error_int			cc_underrun_int	cc_overrun_int	fb_rx_fifo_error_int	fb_tx_fifo_error_int
24	ch_24		ffs_ls_sync_status	fb_rxrst_o	fb_txrst_o			cc_re_o	cc_we_o
25	ch_25								
Per Channel SERDES Registers (7)									
26	ch_26[3]		pci_det_done	rlos_lo	~rlos_lo	rlos_hi	~rlos_hi	rlol	~rlol
27	ch_27	dco_calib_err	dco_calib_done	dco_facq_err	dco_facq_done				pci_connect
28	ch_28	dco_status[7]	dco_status[6]	dco_status[5]	dco_status[4]	dco_status[3]	dco_status[2]	dco_status[1]	dco_status[0]
29	ch_29	dco_status[15]	dco_status[14]	dco_status[13]	dco_status[12]	dco_status[11]	dco_status[10]	dco_status[9]	dco_status[8]
2A	ch_int_2a[4]		pci_det_done_int	rlos_lo_int	~rlos_lo_int	rlos_hi_int	~rlos_hi_int	rlol_int	~rlol_int
2B	ch_2b								
2C	ch_2c								

1. BA = Base Address (Hex)
2. Interrupt control register related to an interruptible status (int_sts_x) register.
3. Status register which has an associated interruptible status (int_sts_x) register.
4. Interruptible status register; clear on read (has associated control register and status register).
5. Status register with clear on read.
6. PRBS is generated by built-in logic and is for internal test only.

Note: Default value is “0”, unless otherwise specified.

Table 8-69. PCS Control Register 1 (CH_00)

Bit	Name	Description	R/W	Default
7	enable_cg_align	Only valid when operating in uc_mode 1 = Enable continuous comma alignment 0 = Disable continuous comma alignment	R/W	0
6	prbs_enable	1 = Enable PRBS generator & checker 0 = Normal operational mode.	R/W	0
5	prbs_lock	1 = Lock receive PRBS checker 0 = Unlock receive PRBS checker	R/W	0
4	ge_an_enable	1 = Enable GIGE Auto Negotiation 0 = Disable GIGE Auto Negotiation	R/W	0
3:2	Reserved			
1	invert_tx	1 = Invert transmitted data 0 = Do not invert transmitted data.	R/W	0
0	invert_rx	1 = Invert received data 0 = Do not invert received data.	R/W	0

Table 8-70. PCS Control Register 2 (CH_01)

Bit	Name	Description	R/W	Default
7	pfifo_clr	1 = Clears PFIFO if quad register bit pfifo_clr_sel is set to 1. This signal is ORed with interface signal pfifo_clr. 0 = Normal operation	R/W	0
6	pcie_ei_en	1 = PCI Express Electrical Idle 0 = Normal operation	R/W	0
5:4	pcs_det_time_sel[1:0]	PCS connection detection time 11 = 16us 10 = 4us 01 = 2us 00 = 8us	R/W	0
3	rx_gear_mode	1 = Enable 2:1 gearing for receive path on all channels 0 = Disable 2:1 gearing for receive path on all channels (no gearing)	R/W	0
2	tx_gear_mode	1 = Enable 2:1 gearing for transmit path on all channels 0 = Disable 2:1 gearing for transmit path on all channels (no gearing)	R/W	0
1	rx_ch	1 = Receive outputs can be monitored on the test characterization pins. The test characterization mode (bit 6 in pcs_ctl_4_qd_03) should be set to ‘1’.	R/W	0
0	tx_ch	1 = Transmit PCS inputs are sourced from test characterization ports. The test characterization mode should be enabled.	R/W	0

Table 8-71. PCS Control Register 3 (CH_02)

Bit	Name	Description	R/W	Default
7	bus_width8	1 = 8-bit bus between PCS and SER 0 = 10-bit bus between PCS and SER.	R/W	0
6	sb_bypass	1 = Bypass Tx SERDES Bridge 0 = Normal operation	R/W	0
5	sb_pfifo_lp	1 = Enable Parallel Loopback from Rx to Tx via parallel FIFO 0 = Normal data operation	R/W	0
4	sb_bist_sel	1 = Select BIST data 0 = Select Normal data	R/W	0
3	enc_bypass	1 = Bypass 8b10b encoder 0 = Normal operation	R/W	0
2	sel_bist_txd4enc	1 = Enable BIST data to Tx before 8b10b ENC 0 = Normal operation	R/W	0
1	tx_gear_bypass	1 = Bypass PCS Tx gear box 0 = Normal operation	R/W	0
0	fb_loopback	1 = Enable loopback in the PCS just before FPGA bridge from Rx to Tx. 0 = Normal data operation.	R/W	0

Table 8-72. PCS Control Register 4 (CH_03)

Bit	Name	Description	R/W	Default
7	lsm_sel	1 = selects External LSM for Word Alignment 0 = selects Internal LSM for Word Alignment	R/W	0
6	signal_detect	1 = force enabling the Rx link state machine 0 = dependent of ffc_signal_detect to enable the Rx link state machine	R/W	0
5	rx_gear_bypass	1 = Bypass PCS Rx gear box 0 = Normal operation	R/W	0
4	ctc_bypass	1 = Bypass clock toleration compensation 0 = Normal operation	R/W	0
3	dec_bypass	1 = Bypass 8b10b decoder 0 = Normal operation	R/W	0
2	wa_bypass	1 = Bypass word alignment 0 = Normal operation	R/W	0
1	rx_sb_bypass	1 = Bypass Rx SERDES Bridge 0 = Normal operation	R/W	0
0	sb_loopback	1 = Enable loopback in the PCS from Tx to Rx in SERDES bridge. 0 = Normal data operation.	R/W	0

Table 8-73. PCS Interrupt Control Register 5 (CH_04)

Bit	Name	Description	R/W	Default
7:4	Reserved			
3	cc_underrun_int_ctl	1 = Enable interrupt for cc_underrun 0 = Disable interrupt for cc_underrun.	RW	0
2	cc_overrun_int_ctl	1 = Enable interrupt for cc_overrun 0 = Disable interrupt for cc_overrun.	RW	0
1	fb_rx_fifo_error_int_ctl	1 = Enable interrupt on empty/full condition in the receive FPGA bridge FIFO.	R/W	0
0	fb_tx_fifo_error_int_ctl	1 = Enable interrupt on empty/full condition in the transmit FPGA bridge FIFO.	R/W	0

Table 8-74. PCS Control Register 6 (CH_05)

Bit	Name	Description	R/W	Default
7:1	Reserved			
0	low_hi_sel	1 = Select rlos_hi 0 = Select rlos_lo (this option is for internal use only)	R/W	0

Table 8-75. PCS Control Register 7 (CH_06)

Bit	Name	Description	R/W	Default
7:0	Reserved			

Per Channel SERDES Control Register Details

Note: Except indicated, all channels must be reset after writing any SERDES control register.

Table 8-76. SERDES Control Register 1 (CH_07)

Bit	Name	Description	R/W	Default
7	REQ_EN	1 = Receiver equalization enable 0 = Receiver equalization disable	R/W	0
6	REQ_LVL_SET	Level setting for equalization 1 = long-reach equalization 0 = mid-length route equalization	R/W	0
5	RCV_DCC_EN	1 = Receiver DC coupling enable. 0 = AC coupling (default)	R/W	0
4:3	RATE_SEL [1:0]	Equalizer pole position select: 00 = pole position for high frequency range 01 = pole position for medium frequency range 10 = pole position for low frequency range 11 = not used	R/W	2'b00
2:0	RX_DCO_CK_DIV[2:0]	VCO output frequency select: 000 = divided by 1 001 = reserved 010 = divided by 2 011 = reserved 100 = divided by 4 101 = divided by 8 110 = divided by 16 111 = divided by 32	R/W	3'b000

Table 8-77. SERDES Control Register 2 (CH_08)

Bit	Name	Description	R/W	Default
7:4	LB_CTL [3:0]	Loop back control: [3] = slb_r2t_dat_en, serial rx to tx LB enable(CDR data) [2] = slb_r2t_ck_en, serial rx to tx LB enable(CDR clock) [1] = slb_eq2t_en, serial LB from equalizer to driver enable [0] = slb_t2r_en, serial tx_to rx LB enable	R/W	4'h0
3:2	RTERM_RXADJ [1:0]	Termination resistor compensation 00 = default 01= -7% 10= -14% 11= -20%	R/W	2'b00
1:0	RTERM_RX [1:0]	00 = 50 ohm 01 = 75 ohm 10 = 2K ohm 11 = 60 ohm	R/W	2'b00

Table 8-78. SERDES Control Register 3 (CH_09)

Bit	Name	Description			R/W	Default
SERDES Control Register 3(CH_09) for LatticeECP2M-35						
7:5	TDRV_AMP[2:0]	CML driver amplitude setting (mV), VCCOB = 1.2V 000 = 1040(default) 001 = 1280 010 = 1320 011 = 1360 100 = 640 101 = 760 110 = 870 111 = 990			R/W	0
4:2	TDRV_PRE_SET[2:0]	Tx_driver pre-emphasis level setting 0 1 2 3 4 5 6	% 0 16 36 40 44 56 80		R/W	0
1:0	TDRV_DAT_SEL[1:0]	Driver output select: 00 = data from Serializer muxed to driver (normal operation) 01 = data rate clock from Serializer muxed to driver 10 = serial Rx to Tx LB (data) if slb_r2t_dat_en='1' 10 = serial Rx to Tx LB (clock) if slb_r2t_ck_en='1' 11 = serial LB from equalizer to driver if slb_eq2t_en='1'			R/W	0
SERDES Control Register 3(CH_09) for LatticeECP2M-20/50/70/100						
7:5	TDRV_AMP[2:0]	See Table 8-81			R/W	0
4.0	TDRV_PRE_SET[4:0]	Tx_driver pre-emphasis level setting			R/W	0
		3-bit setting in GUI 0 1 2 3 4 5 6	5-bit setting in register 00000 00001 00010 01010 10010 00011 00100	% 0 12 26 30 33 40 53		

Table 8-79. SERDES Control Register 4 (CH_0A)

Bit	Name	Description	R/W	Default
7	Reserved			
6:5	TDRV_DAT_SEL[1:0][1]	Driver output select: 00 = data from Serializer muxed to driver (normal operation) 01 = data rate clock from Serialzer muxed to driver 10 = serial Rx to Tx LB (data) if slb_r2t_data_en='1' 10 = serial Rx to Tx LB (clock) if slb_r2t_ck_en='1' 11 = serial LB from equalizer to driver if slb_eq2t_en='1'	R/W	0
4	TDRV_PRE_EN	1 = Tx driver pre-emphasis enable 0 = Tx driver pre-emphasis disable.	R/W	0
3:2	RTERM_TX [1:0]	Tx resistor termination select. Disabled when PCI Express feature is enabled 00 = 50 ohm (default) 01 = 75 ohm 10 = 5K ohm	R/W	2'b00
1	RATE_MODE_TX	0 = Full rate selection for transmit 1 = Half rate selection for transmit	R/W	0
0	tpwdnb	0 = Power down transmit channel 1= Power up transmit channel	R/W	0

1. For LatticeECP2M-20/50/70/100.

Table 8-80. SERDES Control Register 4 (CH_0B)

Bit	Name	Description	R/W	Default
TDRV_AMP Setting for LatticeECP2M-35				
7:6	TDRV_AMP[4:3]	See Table 8-81.	R/W	0
7:5	Reserved	For LatticeECP2M-35 only		
4	oob_en	1=Enables boundary scan input path for routing the high speed receive inputs to a lower speed SERDES in the FPGA (for out of band application).	R/W	0
3:2	rx_refck_sel [1:0]	Rx CDR Reference Clock Select 00 = refclk (differential input) 01 = core_rxrefclk 1x = reserved	R/W	2'b00
1	RATE_MODE_RX	0 = Full rate selection for receive 1 = Half rate selection for receive	R/W	0
0	rpwdnb	0 = Power down receive channel. 1= Power up receive channel	R/W	0

Table 8-81. TDRV_AMP Setting for LatticeECP2M-20/50/70/100

Bit	Name	Description			R/W	Default
CH_0B [7:6]	TDRV_AMP[4:3]	Tx_driver pre-emphasis level setting			R/W	0
CH_09 [7:5]	TDRV_AMP[2:0]	**3-bit setting in GUI**	**5-bit setting in register**	**mV**		
		0	00101	990		
		1	00001	1250		
		2	00010	1300		
		3	00011	1350		
		4	11100	610		
		5	10100	730		
		6	10110	820		
		7	01110	940		

Table 8-82. SERDES Interrupt Control Register 1 (CH_0C)

Bit	Name	Description	R/W	Default
7	Reserved			
6	pci_det_done_int_ctl	1 = Enable interrupt for detection of far-end receiver for PCI Express	R/W	0
5	rlos_lo_int_ctl	1 = Enable interrupt for Rx Loss of Signal when input levels fall below the programmed LOW threshold (using rlos_lset)	RW	0
4	~rlos_lo_int_ctl	1 = Enable interrupt for receiver Rx Loss of Signal when input level meets or is greater than programmed LOW threshold	RW	0
3	rlos_hi_int_ctl	1 = Enable interrupt for Rx Loss of Signal when input levels fall below the programmed HIGH threshold (using rlos_lset)	RW	0
2	~rlos_hi_int_ctl	1 = Enable interrupt for Rx Loss of Signal when input level meets or is greater than programmed HIGH threshold	RW	0
1	rlol_int_ctl	1= Enable interrupt for receiver loss of lock	R/W	0
0	~rlol_int_ctl	1= Enable interrupt when receiver recovers from loss of lock	R/W	0

Per Channel PCS Status Register Details

Table 8-83. PCS Status Register 1 (CH_20)

Bit	Name	Description	R/W	Int?
7:5	Reserved			
4	pfifo_error	1 = Parallel FIFO error 0 = No Parallel FIFO error	RO	Y
3	cc_underrun	1 = CC FIFO underrun 0 = CC FIFO not underrun	RO	Y
2	cc_overrun	1 = CC FIFO overrun 0 = CC FIFO not overrun	RO	Y
1	fb_rx_fifo_error	1 = FPGA bridge (FB) Rx FIFO overrun 0 = FB Rx FIFO not overrun	RO	Y
0	fb_tx_fifo_error	1 = FPGA bridge (FB) Tx FIFO overrun 0 = FB Tx FIFO not overrun	RO	Y

Table 8-84. PCS Status Register 2 (CH_21)

Bit	Name	Description	R/W	Int?
7:0	prbs_errors	Count of the number of PRBS errors. Clears to zero on read. Sticks at FF.	RO CR	N

Table 8-85. PCS Status Register 3 (CH_22)

Bit	Name	Description	R/W	Int?
7:0	Reserved			

Table 8-86. PCS General Interrupt Status Register 4 (CH_23)

Bit	Name	Description	R/W	Int?
7:4	Reserved			
3	cc_underrun_int	1 = Interrupt generated on cc_underrun 0 = Interrupt not generated on cc_underrun	RO CR	Y
2	cc_overrun_int	1 = Interrupt generated on cc_overrun 0 = Interrupt not generated on cc_overrun	RO CR	Y
1	fb_rx_fifo_error_int	1 = Interrupt generated on fb_rx_fifo_error. 0 = Interrupt not generated fb_rx_fifo_error.	RO CR	Y
0	fb_tx_fifo_error_int	1 = Interrupt generated on fb_tx_fifo_error 0 = Interrupt not generated fb_tx_fifo_error.	RO CR	Y

Table 8-87. PCS Status Register 5 (CH_24)

Bit	Name	Description	R/W	Int?
7	Reserved			
6	ffs_ls_sync_status	1 = Sync in the link state machine. 0 = Not sync in the LSM.	RO	N
5	fb_rxrst_o	1 = Normal operation 0 = FPGA bridge Rx reset	RO	N
4	fb_txrst_o	1 = Normal operation 0 = FPGA bridge Tx reset	RO	N
3	Reserved			
2	Reserved			
1	cc_re_o	1 = Elastic FIFO read enable 0 = Elastic FIFO read disable	RO	N
0	cc_we_o	1 = Elastic FIFO write enable 0 = Elastic FIFO write disable	RO	N

Table 8-88. PCS Status Register 6 (CH_25)

Bit	Name	Description	R/W	Int?
7:0	Reserved			

Per Channel SERDES Status Register Details

Table 8-89. SERDES Status Register 1 (CH_26)

Bit	Name	Description	R/W	Int?
7	Reserved			
6	pci_det_done	1 = Receiver detection process completed by SERDES transmitter. 0 = Receiver detection process not completed by SERDES transmitter.	RO CR	Y
5	rlos_lo	1= Indicates that the input signal detected by receiver is below the programmed LOW threshold	RO CR	Y
4	~rlos_lo	1= Indicates that the input signal detected by receiver is greater than or equal to the programmed LOW threshold	RO CR	Y
3	rlos_hi	1= Indicates that the input signal detected by receiver is below the programmed HIGH threshold	RO CR	Y
2	~rlos_hi	1= Indicates that the input signal detected by receiver is greater than or equal to the programmed HIGH threshold	RO CR	Y
1	rlol	1=Indicates CDR loss of lock to data. CDR is locked to reference clock.	RO	Y
0	~rlol	1 = Indicates that CDR has locked to data.	RO	Y

Table 8-90. SERDES Status Register 2 (CH_27)

Bit	Name	Description	R/W	Int?
7	DCO_CALIB_ERR	1 = indicates DCO calibration might be wrong (L/H boundary band selected)	RO	N
6	DCO_CALIB_DONE	1 = indicates DCO calibration done	RO	N
5	DCO_FACQ_ERR	1 = indicates DCO frequency acquisition error (>300ppm)	RO	N
4	DCO_FACQ_DONE	1 = indicates DCO frequency acquisition done	RO	N
3:1	Reserved			
0	pci_connect	1 = Receiver detected by SERDES transmitter (at the transmitter device). 0 = Receiver not detected by SERDES transmitter (at the transmitter device).	RO	N

Table 8-91. SERDES Status Register 3 (CH_28)

Bit	Name	Description	R/W	Int?
7:0	DCO_STATUS[7:0]	setdcoidac[7:0]	RO	N

Table 8-92. SERDES Status Register 4 (CH_29)

Bit	Name	Description	R/W	Int?
7:0	DCO_STATUS[15:8]	[1:0] = setdcoidac[9:8][7:2] = setdcoband[5:0]	RO	N

Table 8-93. SERDES Interrupt Status Register 5 (CH_2A)

Bit	Name	Description	R/W	Int?
7	Reserved			
6	pci_det_done_int	1 = Interrupt generated for pci_det_done	RO CR	Y
5	rlos_lo_int	1 = Interrupt generated for rlos_lo	RO CR	Y
4	~rlos_lo_int	1 = Interrupt generated for ~rlos_lo	RO CR	Y
3	rlos_hi_int	1 = Interrupt generated for rlos_hi	CO CR	Y
2	~rlos_hi_int	1 = Interrupt generated for ~rlos_hi	CO CR	Y
1	rlol_int	1 = Interrupt generated for rlol	CO CR	Y
0	~rlol_int	1 = Interrupt generated for ~rlol	CO CR	Y

Table 8-94. SERDES Status Register 6 (CH_2B)

Bit	Name	Description	R/W	Default
7:0	Reserved			

Table 8-95. SERDES Status Register 7 (CH_2C)

Bit	Name	Description	R/W	Default
7:0	Reserved			

Table 8-96. K Characters Recognized for Different Standards

Character	Gbe	XAUI	1xFC	PCI Express	RapidIO
K23.7 (F7)	Carrier extend			PAD	
K27.7 (FB)	SOP	ST		Start TLP	A (align)
K28.0 (1C)		SKIP R		SKIP	SC
K28.1 (3C)				FTS	
K28.2 (5C)		SoS		Start DLLP	
K28.3 (7C)		ALIGN A		IDLE	PD
K28.4 (9C)		SEQ			
K28.5 (BC)	+D5.6 or D16.2 = IDLE	SYNC K	+D21.4 +D21.5 +D21.5 = IDLE	COMMA (used for alignment)	K
K28.6 (DC)					
K28.7 (FC)					
K29.7 (FD)	EOP	T		END	R (skip)
K30.7 (FE)	ERR	ERR		END BAD	

Note: Refer to each standard specification for detailed information.

Appendix B. 8b10b Symbol Codes

Table 8-97. 8b10b Symbol Codes

Symbol Name (Mode Combination Table)	Symbol Code (8b10b Code)	10-Bit GUI Representation
K28.0	8`b000_11100	0100011100
K28.5	8`b101_11100	0110111100
K29.7	8`b111_11101	0011111101
D16.2	8`b010_10000	0001010000
D21.4	8`b100_10101	0010010101
D21.5	8`b101_10101	0010110101
K28.5+	10`b110000_0101	1100000101
K28.5-	10`b001111_1010	0011111010

Table 8-98. Lattice Mask for Symbol Code

Symbol Name (Mode Combination Table)	Symbol Code (8b10b Code)	10-Bit GUI Representation
28.5 (Mask)	10`b111111_1111	1111111111

Appendix C. Attribute Cross-Reference Table

Table 8-99. Attribute Cross-Reference Table

Independent Attribute Name	Dependent Attribute Names	Attribute Value	Register Map[3]
PROTOCOL	[QAUD_MODE, CHn_RX_DET, CHn_OOB_EN, CHn_GE_AN_EN]	GIGE : [00000,00,0,1] PCIE : [00100,00.0,0] G8B10B : [00001,00,0,0] 10BSER : [00001,00,0,0] 8BSER : [00001,00,0,0] SDSDI :[00001,00,1,0] HDSDI :[00001,00,0,0]	{QD_00[4:0], CH_01[5:4], CH_0B[4], CH_00[4]}
CH[0,1,2,3]_MODE	[CHn_TXPWDNB, CHn_RXPWDNB]	SINGLE : [11] GROUP1 : [11] GROUP2 : [11] DISABLE: [00]	{CH_0A[0], CH_0B[0]}
DATARANGE (ECP2M35 ES Device)	[PLL_DIV, CHn_CDR_DIV]	LOW : [101,101] MEDLOW : [100,100] MED : [010,010] MEDHIGH : [010,000] HIGH : [000,000]	{QD_14[2:0], CH_07[2:0]}
DATARANGE (All other Devices)		LOW : [101,101] MEDLOW : [100,100] MED : [010,010] MEDHIGH : [000,000] HIGH : [000,000]	
CH[0,1,2,3]_REFCK_MULT	[BUS8BIT_SEL, REFCK25X, REFCK_MODE, CHn_RATE_MODE_RX, CHn_RATE_MODE_TX]	8X : [1,0,1,0,0] 8XH : [1,0,1,1,1] 10X : [0,0,1,0,0] 10XH : [0,0,1,1,1] 16X : [1,0,0,0,0] 16XH : [1,0,0,1,1] 20X : [0,0,0,0,0] 20XH : [0,0,0,1,1] 25X : [0,1,0,0,0] 25XH : [0,1,0,1,1]	{QD_12[6], QD_12[7], QD_13[6], CH_0B[1], CH_0A[1]}
CH[0,1,2,3]_DATA_WIDTH	[TXCLKF, TXCLKH, CHn_RXCLKF, CHn_RXCLKH, CHn_TX_GEAR, CHn_RX_GEAR, CHn_BUS_WIDTH8]	8,BYPASS :[0,1,0000,0000,0,0,1] 8, NORMAL; 10, BYPASS; 10, NORMAL :[0,1,0000.0000,0,0,0] 16,BYPASS :[01,1111,1111,1,1,1] 16,NORMAL;20,BYPASS;20,NORMAL :[0,1,1111,1111,1,1,0]	{QD_1B[7], QD_1B[6], QD_1A[7:4], QD_1A[3:0], CH_01[2], CH_01[3], CH_02[7]}
PLL_SRC		REFCLK :[0] CORE_TXREFCLK :[1]	{QD_11[5]}
CH[0,1,2,3]_CDR_SRC		REFCLK :[00] CORE_RXREFCLK :[01]	{CH_0B[3:2]}

Table 8-99. Attribute Cross-Reference Table (Continued)

Independent Attribute Name	Dependent Attribute Names	Attribute Value	Register Map[3]
CH_[0,1,2,3]_TRDV_AMP (ECP2M35 all Device)		0:[000] 1:[001] 2:[010] 3:[011] 4:[100] 5:[101] 6:[110] 7:[111]	CH_09[7:5]
CH_[0,1,2,3]_TRDV_AMP (All other Devices)		0:[00101] 1:[00001] 2:[00010] 3:[00011] 4:[11100] 5:[10100] 6:[10110] 7:[01110]	CH_0B[7:6] CH_09[7:5]
CH_[0,1,2,3]_TX_PRE[5] (ECP2M35 ES Device)	[CHn_TRDV_PRE_EN, CHn_TRDV_PRE_SET]	DISABLE:[0,000] 0:[1,000] 1:[1,001] 2:[1,010] 3:[1,011] 4:[1,100] 5:[1,101]	{CH_0A[4], CH_09[4:2]}
CH_[0,1,2,3]_TX_PRE[6] (ECP2M35 non-ES device)		DISABLE:[0,000] 0:[1,000] 1:[1,001] 2:[1,010] 3:[1,011] 4:[1,100] 5:[1,101] 6:[1,110]	
CH_[0,1,2,3]_TX_PRE[6] (All other Devices)		DISABLE:[0,00000] 0:[1,00000] 1:[1,00001] 2:[1,00010] 3:[1,01010] 4:[1,10010] 5:[1,00011] 6:[1,00100]	{CH_0A[4], CH_09[4:0]}
CH[0,1,2,3]_RTERM_TX		50:[00] 75:[01] 5K:[10]	{CH_0A[3:2]}
CH[0,1,2,3]_RX_EQ	[CHn_REQ_EN, CHn_REQ_LVL_SET, CHn_RATE_SEL]	DISABLE :[0,0,00] MID_LOW :[1,0,10] MID_MED :[1,0,01] MID_HIGH :[1,0,00] LONG_LOW :[1,1,10] LONG_MED :[1,1,01] LONG_HIGH :[1,1,00]	{CH_07[7], CH_07[6], CH_07[4:3]}
CH[0,1,2,3]_RTERM_RX	[CHn_RX_RTERM]	50 :[00] 60 :[11] 75 :[01] HIGH[4] :[10]	{CH_08[1:0]}
CH[0,1,2,3]_RX_DCC		AC:[0] DC:[1]	{CH_07[5]}

Table 8-99. Attribute Cross-Reference Table (Continued)

Independent Attribute Name	Dependent Attribute Names	Attribute Value	Register Map[3]
LOS_THRESHOLD (ECP2M35 ES device)	[RLOS_LO, RLOS_HI]	0:[000,000] 1:[001,000] 2:[010,000] 3:[011,000] 4:[100,000] 5:[101,000] 6:[110,000] 7:[111,000]	{QD_12[2:0], QD_12[5:3]}
LOS_THRESHOLD (All other devices)	[LOS_HI_SEL, RLOS_LO, RLOS_HI]	0:[1111,000,000] 1:[1111,000,001] 2:[1111,000,010] 3:[1111,000,011] 4:[1111,000,100] 5:[1111,000,101] 6:[1111,000,110] 7:[1111,000,111]	{CH_05[0], QD_12[2:0], QD_12[5:3]}
PLL_TERM		50:[0] 2K:[1]	{QD_11[3]}
PLL_DCC		AC:[0] DC:[1]	{QD_11[4]}
PLL_LOL_SET		0:[00] 1:[01] 2:[10] 3:[11]	{QD_14[4:3]}
CH[0,1,2,3]_TX_SB	[CHn_TXPOL, CHn_TXSBBYP]	NORMAL: [0,0] INV : [1,0]	{CH_00[1], CH_02[6]}
CH[0,1,2,3]_RX_SB	[CHn_RXPOL, CHn_RXWSBBYP]	NORMAL: [0,0] INV : [1,0]	{CH_00[0], CH_03[1]}
CH[0,1,2,3]_8B10B	[CHn_TXENC, CHn_RXDEC]	NORMAL: [0,0] BYPASS: [1,1]	{CH_02[3], CH_03[3]}
COMMA_A		Note 1	{QD_0A[0:7], QD_0C[6:7]}
COMMA_B		Note 1	{QD_0B[0:7], QD_0C[4:5]}
COMMA_M		Note 1	{QD_09[0:7], QD_0C[2:3]}
CH[0,1,2,3]_COMMA_ALIGN	[CHn_RXWA, CHn_LSM_SEL, CHn_C_ALIGN, CHn_SIG_DET]	AUTO : [0,0,0,1] DYNAMIC : [0,1,0,0] BYPASS : [1,1,0,0]	{CH_03[2], CH_03[7], CH_00[7], CH_03[6]}
CH[0,1,2,3]_CTC_BYP	[CHn_RXRECCLK]	NORMAL : [0,0,0,0,0] BYPASS : [1,1,1,1,1]	{QD_19[3:0], CH_03[4]}
CC_MATCH1			{QD_04[7:0], QD_08[1:0]}
CC_MATCH2			{QD_05[7:0], QD_08[3:2]}
CC_MATCH3			{QD_06[7:0], QD_08[5:4]}
CC_MATCH4			{QD_07[7:0], QD_08[7:6]}
CC_MATCH_MODE	[MATCH_2_EN, MATCH_4_EN]	MATCH_3_4 : [1,0] MATCH_4 : [0,0] MATCH_1_2_3_4 : [0,1]	{QD_03[4], QD_03[5]}

Table 8-99. Attribute Cross-Reference Table (Continued)

Independent Attribute Name	Dependent Attribute Names	Attribute Value	Register Map[3]
CC_MIN_IPG		0: [00] 1: [01] 2: [10] 3: [11]	{QD_03[7:6]}
CCHMARK		0:[0000] 1:[0001] 2:[0010] 3:[0011] 4:[0100] 5:[0101] 6:[0110] 7:[0111] 8:[1000] 9:[1001] 10:[1010] 11:[1011] 12:[1100] 13:[1101] 14:[1110] 15:[1111]	{QD_02[7:4]}
CCLMARK		0:[0000] 1:[0001] 2:[0010] 3:[0011] 4:[0100] 5:[0101] 6:[0110] 7:[0111] 8:[1000] 9:[1001] 10:[1010] 11:[1011] 12:[1100] 13:[1101] 14:[1110] 15:[1111]	{QD_02[3:0]}
[PLL_SRC, CHn_CDR_SRC]	[REFCKLOCAL]	[CORE_TXREFCLK, CORE_RXREFCLK, CORE_RXREFCLK, CORE_RXREFCLK, CORE_RXREFCLK]:[1] [REFCLK,X,X,X,X]:[0]note2 [X,REFCLK,X,X,X]:[0] [X,X,REFCLK,X,X]:[0] [X,X,X,REFCLK,X]:[0] [X,X,X,X,REFCLK]:[0]	{QD_11[0]}
OS_SSLB	[CHn_EQ2T_EN]	0:[0] 1:[1]	{CH_08[5]}
OS_SPLBPORTS	[PFIFO_CLR_SEL, CHn_SB_PFIFO_LP]	0:[0,0] 1:[1,1]	{QD_03[2], CH_02[5]}
OS_PCSLBPORTS	[FB_LOOPBACK]	0:[0] 1:[1]	{CH_02[0]}
OS_REFCK2CORE	[REFCK2CORE]	0:[0] 1:[1]	{QD_11[1]}
OS_PLLQCLKPORTS	[TXCLKQ, CHn_RXCLKQ]	0:[0,0000] 1:[1,1111]	{QD_QB[5], QD_1B[3:0]}

Table 8-99. Attribute Cross-Reference Table (Continued)

Independent Attribute Name	Dependent Attribute Names	Attribute Value	Register Map[3]
OS_INT_ALL	[PLOL_INT, PLOLN_INT, CHn_PCIDETINT, CHn_RLOSLINT, CHn_RLOSLNINT, CHn_RLOSHINT, CHn_RLOSHNINT, CHn-RLOLINT, CHn_RLOLNINT, CHn_LSSYNCINT, CHn_LSSYNCNINT, CHn_TXFIFOINT, CHn_RXFIFOINT, CHn_CCORUNINT, CHn_CCURUNINT]	0:[0,0,0,0,0,0,0,0,0,0000, 0000,0,0,0,0] 1:[1,1,1,1,1,1,1,1,1,1111, 1111,1,1,1,1]	{QD_16[7], QD_16[6], QD_0C[1], QD_0C[2], QD_0C[3], QD_0C[4], QD_0C[5], QD_0C[6], QD_0C[7], QD_10[7:4], QD_10[3:0], CH_04[0], CH_04[1], CH_04[2], CH_04[3]}

1. 10-bit symbol code default value or specified by user in the 'PCS Advanced setup' configuration GUI. Since the 10-bit symbol code representation in the GUI is LSB to MSB, the bit representation is swapped appropriately in the table.
2. X = Don't care.
3. QD_xx : Quad Register with Base Address = xx
   CH_yy : Channel Register with Base Address = yy
4. HIGH = 2K for LatticeECP2M35 ES devices and infinity for all other devices.
5. 0: 0%; 1: 16%; 2: 32%; 3:48%; 4:64%; 5:80%.
6. 0: 0%; 1: 16%; 2: 36%; 3:40%; 4:44%; 5:56%; 6:80%.

Appendix D. Protocol Specific SERDES Setup Options

Table 8-100. Protocol Specific SERDES Setup Options

<table>
<tr><th>Protocol</th><th>DATARATE</th><th>DATARATE Range</th><th>REFCK Multiplier</th><th>DATA WIDTH</th><th>Rx Equalization[1]</th></tr>
<tr><td>GbE</td><td>1.25</td><td>MED</td><td>20x, 10x, 5x</td><td rowspan="2">8, 16</td><td>DISABLE, MID_MED, LONG_MED</td></tr>
<tr><td>PCI Express</td><td>2.5</td><td>HIGH</td><td>25x, 20x</td><td rowspan="2">DISABLE, MID_HIGH, LONG_HIGH</td></tr>
<tr><td>XAUI</td><td>3.125</td><td>HIGH</td><td>20x</td><td>16</td></tr>
<tr><td>Generic 8b10b</td><td rowspan="3">ANY_VALUE</td><td rowspan="3">LOW, MEDLOW, MED, MEDHIGH, HIGH</td><td>20x, 10x, 5x</td><td rowspan="2">8, 16</td><td rowspan="3">DISABLE, MID_LOW, MID_MED, MID_HIGH, LONG_LOW, LONG_MED, LONG_HIGH</td></tr>
<tr><td>8-bit SERDES_Only</td><td>16x, 8x, 4x</td></tr>
<tr><td>10-bit SERDES_Only</td><td>20x, 10x, 5x</td><td>10, 20</td></tr>
</table>

1. MID: about 20" in length
   LONG: about 40" in length
   LOW: less than 1.2 Gbps
   MED: between 1.2 Gbps and 2 Gbps
   HIGH: over 2 Gbps

Appendix E. Lattice Diamond Usage Overview

This appendix discusses the use of Lattice Diamond design software for projects that include the LatticeECP2M SERDES/PCS module .

For general information about the use of Lattice Diamond, refer to the Lattice Diamond Tutorial.

If you have been using ispLEVER software for your FPGA design projects, Lattice Diamond may look like a big change. But if you look closer, you will find many similarities because Lattice Diamond is based on the same toolset and work flow as ispLEVER. The changes are intended to provide a simpler, more integrated, and more enhanced user interface.

Converting an ispLEVER Project to Lattice Diamond

Design projects created in ispLEVER can easily be imported into Lattice Diamond. The process is automatic except for the ispLEVER process properties, which are similar to the Diamond strategy settings, and PCS modules. After importing a project, you need to set up a strategy for it and regenerate any PCS modules.

Importing an ispLEVER Design Project

Make a backup copy of the ispLEVER project or make a new copy that will become the Diamond project.

1. In Diamond, choose **File > Open > Import ispLEVER Project**.
2. In the ispLEVER Project dialog box, browse to the project's .syn file and open it.
3. If desired, change the base file name or location for the Diamond project. If you change the location, the new Diamond files will go into the new location, but the original source files will not move or be copied. The Diamond project will reference the source files in the original location.

The project files are converted to Diamond format with the default strategy settings.

Adjusting PCS Modules

PCS modules created with IPexpress have an unusual file structure and need additional adjustment when importing a project from ispLEVER. There are two ways to do this adjustment. The preferred method is to regenerate the module in Diamond. However this may upgrade the module to a more recent version. An upgrade is usually desirable but if, for some reason, you do not want to upgrade the PCS module, you can manually adjust the module by copying its .txt file into the implementation folder. If you use this method, you must remember to copy the .txt file into any future implementation folders.

Regenerate PCS Modules

1. Find the PCS module in the Input Files folder of File List view. The module may be represented by an .lpc, .v, or .vhd file.
2. If the File List view shows the Verilog or VHDL file for the module, and you want to regenerate the module, import the module's .lpc file:
   a. In the File List view, right-click the implementation folder () and choose **Add > Existing File**.
   b. Browse for the module's .lpc file, **<*module_name*>.lpc**, and select it.
   c. Click **Add**. The .lpc file is added to the File List view.
   d. Right-click the module's Verilog or VHDL file and choose **Remove**.
3. In File List, double-click the module's .lpc file. The module's IPexpress dialog box opens.
4. In the bottom of the dialog box, click **Generate**. The Generate Log tab is displayed. Check for errors and close.

In File List, the .lpc file is replaced with an .ipx file. The IPexpress manifest (.ipx) file is new with Diamond. The .ipx file keeps track of the files needed for complex modules.

Using IPexpress with Lattice Diamond

Using IPexpress with Lattice Diamond is essentially same as with ispLEVER.

The configuration GUI tabs are all the same except for the Generation Options tab. Figure 8-48 shows the Generation Options tab window.

Figure 8-48. Generation Options Tab

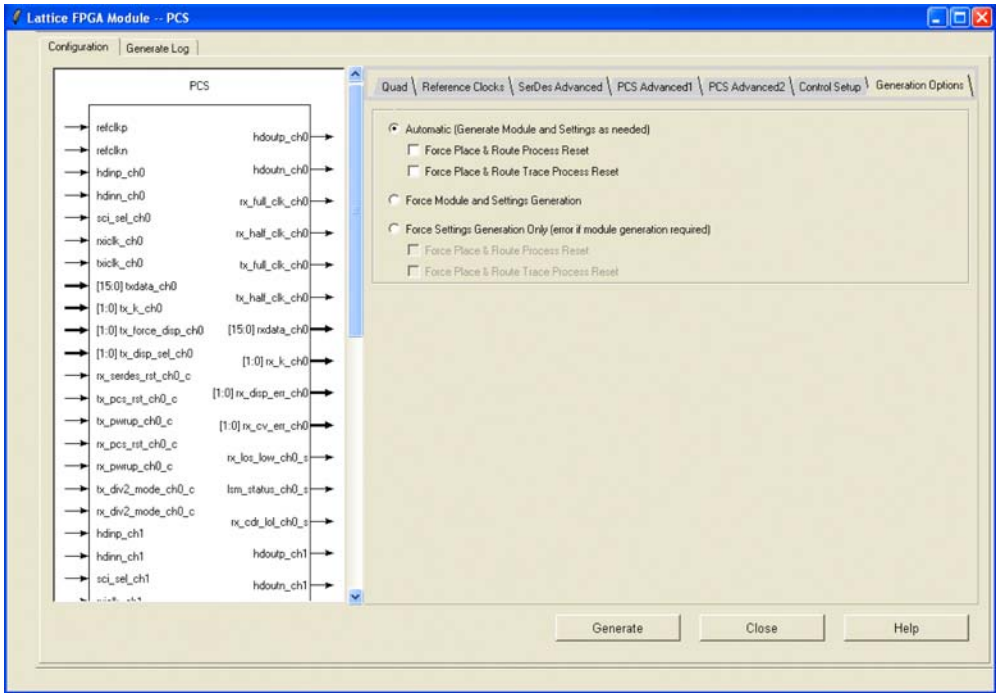


Table 8-101. SERDES_PCS GUI Attributes – Generation Options Tab

GUI Text	Description
Automatic	Automatically generates the HDL and configuration(.txt) files as needed. Some changes do not require regenerating both files.
Force Module and Settings Generation	Generates both the HDL and configuration files.
Force Settings Generation Only	Generates only the attributes file. You get an error message if the HDL file also needs to be generated.
Force Place & Route Process Reset	Resets the Place & Route Design process, forcing it to be run again with the newly generated PCS module.
Force Place & Route Trace Process Reset	Resets the Place & Route Trace process, forcing it to be run again with the newly generated PCS module.

Note:
Automatic is set as the default option. If either Automatic or Force Settings Generation Only and no sub-options (Process Reset Options) are checked and the HDL module is not generated, the reset pointer is set to Bitstream generation automatically.

After the Generation is finished, the reset marks in the process window will be reset accordingly.

Creating a New Simulation Project Using Simulation Wizard

This section describes how to use the Simulation Wizard to create a simulation project (.spf) file so you can import it into a standalone simulator.

1. In Project Navigator, click **Tools > Simulation Wizard**. The Simulation Wizard opens.
2. In the Preparing the Simulator Interface page click **Next**.
3. In the Simulator Project Name page, enter the name of your project in the Project Name text box and browse to the file path location where you want to put your simulation project using the Project Location text box and Browse button.

   When you designate a project name in this wizard page, a corresponding folder will be created in the file path you choose. Click **Yes** in the popup dialog that asks you if you wish to create a new folder.
4. Click either the Active-HDL® or ModelSim® simulator check box and click **Next**.
5. In the Process Stage page choose which type of Process Stage of simulation project you wish to create Valid types are RTL, Post-Synthesis Gate-Level, Post-Map Gate-Level, and Post-Route Gate-level+Timing. Only those process stages that are available are activated.

   Note that you can make a new selection for the current strategy if you have more than one defined in your project.

   The software supports multiple strategies per project implementation which allow you to experiment with alternative optimization options across a common set of source files. Since each strategy may have been processed to different stages, this dialog allows you to specify which stage you wish to load.
6. In the Add Source page, select from the source files listed in the Source Files list box or use the browse button on the right to choose another desired source file. Note that if you wish to keep the source files in the local simulation project directory you just created, check the **Copy Source to Simulation Directory** option.
7. Click **Next** and a Summary page appears and provides information on the project selections including the simulation libraries. By default, the Run Simulator check box is enabled and will launch the simulation tool you chose earlier in the wizard in the Simulator Project Name page.
8. Click **Finish**.

The Simulation Wizard Project (.spf) file and a simulation script DO file are generated after running the wizard. You can import the DO file into your current project if desired. If you are using Active-HDL, the wizard will generate an .ado file and if you are using ModelSim, it creates and .mdo file.

Note: PCS configuration file, (.txt) must be added in step 6.

LatticeECP2/M sysIO Usage Guide

March 2011 Technical Note TN1102

Introduction

The LatticeECP2™ and LatticeECP2M™ sysIO™ buffers give the designer the ability to easily interface with other devices using advanced system I/O standards. This technical note describes the sysIO standards available and how they can be implemented using Lattice's ispLEVER® design software.

sysIO Buffer Overview

LatticeECP2/M sysIO interface contains multiple Programmable I/O Cells (PIC) blocks. Each PIC contains two Programmable I/Os (PIO), PIOA and PIOB, connected to their respective sysIO Buffers. Two adjacent PIOs can be joined to provide a differential I/O pair (labeled as "T" and "C").

Each Programmable I/O (PIO) includes a sysIO Buffer and I/O Logic (IOLOGIC). The LatticeECP2/M sysIO buffers supports a variety of single-ended and differential signaling standards. The sysIO buffer also supports the DQS strobe signal that is required for interfacing with the DDR memory. One of every 16/18 PIOs in the LatticeECP2/M contains a delay element to facilitate the generation of DQS signals. The DQS signal from the bus is used to strobe the DDR data from the memory into input register blocks. For more information on the architecture of the sysIO buffer please refer to the LatticeECP2/M Family Data Sheet.

The IOLOGIC includes input, output and tristate registers that implement both single data rate (SDR) and double data rate (DDR) applications along with the necessary clock and data selection logic. Programmable delay lines and dedicated logic within the IOLOGIC are used to provide the required shift to incoming clock and data signals and the delay required by DQS inputs in DDR memory. The DDR implementation in the IOLOGIC and the DDR memory interface support are discussed in more detail in TN1105, LatticeECP2/M High-Speed I/O Interface.

Supported sysIO Standards

The LatticeECP2/M sysIO buffer supports both single-ended and differential standards. Single-ended standards can be further subdivided into internally ratioed standard such as LVCMOS, LVTTL and PCI; and externally referenced standards such as HSTL and SSTL. The buffers support the LVTTL, LVCMOS 1.2, 1.5, 1.8, 2.5 and 3.3V standards. In the LVCMOS and LVTTL modes, the buffer has individually configurable options for drive strength, bus maintenance (weak pull-up, weak pull-down, or a bus-keeper latch). Other single-ended standards supported include SSTL and HSTL. Differential standards supported include LVDS, RSDS, BLVDS, LVPECL, differential SSTL and differential HSTL. Tables 1 and 2 list the sysIO standards supported in LatticeECP2/M devices.

Table 9-1. Supported Input Standards

Input Standard	V_{REF} (Nom.)	V_{CCIO}[1] (Nom.)
Single Ended Interfaces		
LVTTL	—	—
LVCMOS33	—	—
LVCMOS25	—	—
LVCMOS18	—	1.8
LVCMOS15	—	1.5
LVCMOS12	—	—
PCI 33	—	3.3
HSTL18 Class I, II	0.9	—
HSTL15 Class I	0.75	—
SSTL3 Class I, II	1.5	—

Table 9-1. Supported Input Standards (Continued)

Input Standard	V_{REF} (Nom.)	V_{CCIO}[1] (Nom.)
SSTL2 Class I, II	1.25	—
SSTL18 Class I, II	0.9	—
GTL+[2]	1.0	—
Differential Interfaces		
Differential SSTL18 Class I, II	—	—
Differential SSTL2 Class I, II	—	—
Differential SSTL3 Class I, II	—	—
Differential HSTL15 Class I	—	—
Differential HSTL18 Class I, II	—	—
LVDS, MLVDS, LVPECL, BLVDS, RSDS	—	—

1 When not specified, V_{CCIO} can be set anywhere in the valid operating range.
2. GTL+ inputs can be supported using HSTL15 Class I inputs with VREF set to 1.0V and external VTT termination to 1.5V. Please see "sysIO Buffer Configurations" on page 6 for implementation details.

Table 9-2. Supported Output Standards

Output Standard	Drive	V_{CCIO} (Nom.)
Single-ended Interfaces		
LVTTL	4mA, 8mA, 12mA, 16mA, 20mA	3.3
LVCMOS33	4mA, 8mA, 12mA 16mA, 20mA	3.3
LVCMOS25	4mA, 8mA, 12mA, 16mA, 20mA	2.5
LVCMOS18	4mA, 8mA, 12mA, 16mA	1.8
LVCMOS15	4mA, 8mA	1.5
LVCMOS12	2mA, 6mA	1.2
LVCMOS33, Open Drain	4mA, 8mA, 12mA 16mA, 20mA	—
LVCMOS25, Open Drain	4mA, 8mA, 12mA 16mA, 20mA	—
LVCMOS18, Open Drain	4mA, 8mA, 12mA 16mA	—
LVCMOS15, Open Drain	4mA, 8mA	—
LVCMOS12, Open Drain	2mA, 6mA	—
PCI33/PCIX	N/A	3.3
HSTL18 Class I	8mA, 12mA	1.8
HSTL18 Class II	N/A	1.8
HSTL15 Class I	4mA, 8mA	1.5
SSTL3 Class I, II	N/A	3.3
SSTL2 Class I	8mA, 12mA	2.5
SSTL2 Class II	16mA, 20mA	2.5
SSTL18 Class I	N/A	1.8
SSTL18 Class II	8mA, 12mA	1.8
Differential Interfaces		
Differential SSTL3, Class I, II	N/A	3.3
Differential SSTL2, Class I	8mA, 12mA	2.5
Differential SSTL2, Class II	16mA, 20mA	2.5
Differential SSTL18, Class I	N/A	1.8
Differential SSTL18, Class II	8mA, 12mA	1.8
Differential HSTL18, Class I	8mA, 12mA	1.8

Table 9-2. Supported Output Standards (Continued)

Output Standard	Drive	V_{CCIO} (Nom.)
Differential HSTL18, Class II	N/A	1.8
Differential HSTL15, Class I	4mA, 8mA	1.5
LVDS	N/A	2.5
MLVDS[1]	N/A	2.5
BLVDS[1]	N/A	2.5
LVPECL[1]	N/A	3.3
RSDS[1]	N/A	2.5

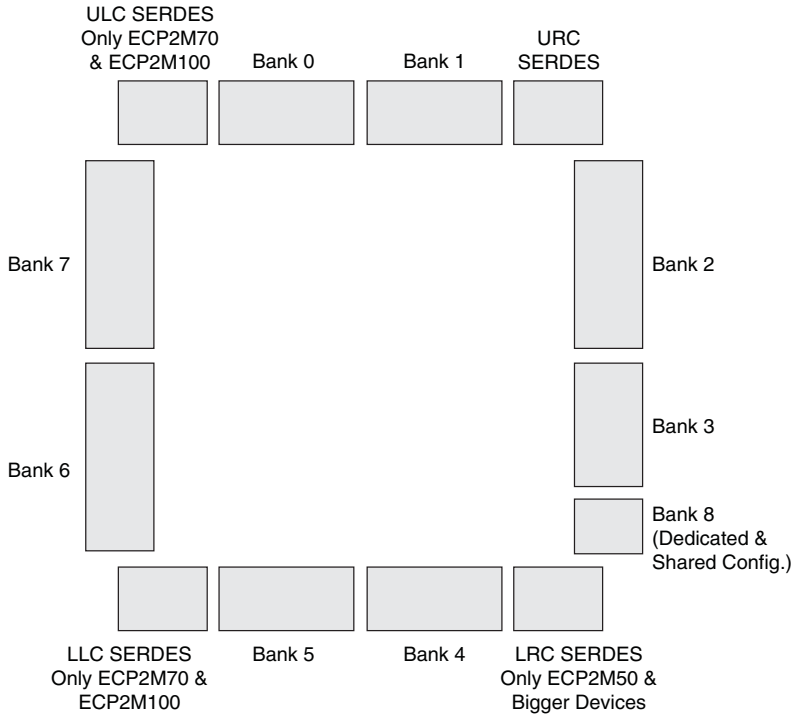
1. Emulated with external resistors.

sysIO Banking Scheme

LatticeECP2/M devices have eight general purpose programmable sysIO banks and a ninth configuration bank. Each of the eight general purpose sysIO banks has a V_{CCIO} supply voltage, and two reference voltages, V_{REF1} and V_{REF2}. Figure 9-1 shows the eight general purpose banks and the configuration bank with associated supplies. Bank 8 is a bank dedicated to configuration logic and has seven dedicated configuration I/Os and 14 multiplexed configuration I/Os. Bank 8 does have the power supply pads (V_{CCIO} and V_{CCAUX}) but does not have any independent V_{REF} pads. The I/Os in Bank 8 are connected to V_{REF} from Bank 3.

On the top and bottom banks, the sysIO buffer pair consists of two single-ended output drivers and two sets of single-ended input buffers (both ratioed and referenced). The left and right sysIO buffer pair consists of two single-ended output drivers and two sets of single-ended input buffers (both ratioed and referenced). The referenced input buffer can also be configured as a differential input. In 50% of the pairs there is also one differential output driver. The two pads in the pair are described as “true” and “comp”, where the true pad is associated with the positive side of the differential input buffer and the comp (complementary) pad is associated with the negative side of the differential input buffer.

Figure 9-1. LatticeECP2M sysIO Banking



Note: URC = upper right corner, LRC = lower right corner, ULC = upper left corner and LLC = lower left corner.

V_{CCIO} (1.2V/1.5V/1.8V/2.5V/3.3V)

There are a total of eight V_{CCIO} supplies, V_{CCIO0} - V_{CCIO7}. Each bank has a separate V_{CCIO} supply that powers the single-ended output drivers and the ratioed input buffers such as LVTTL, LVCMOS, and PCI. LVTTL, LVCMOS3.3, LVCMOS2.5 and LVCMOS1.2 also have fixed threshold options allowing them to be placed in any bank. The V_{CCIO} voltage applied to the bank determines the ratioed input standards that can be supported in that bank. It is also used to power the differential output drivers. In addition, V_{CCIO8} is used to supply power to the sysCONFIG™ signals.

V_{CCAUX} (3.3V)

In addition to the bank V_{CCIO} supplies, devices have a V_{CC} core logic power supply and a V_{CCAUX} auxiliary supply that powers the differential and referenced input buffers. V_{CCAUX} is used to supply I/O reference voltage requiring 3.3V to satisfy the common-mode range of the drivers and input buffers.

V_{CCJ} (1.2V/1.5V/1.8V/2.5V/3.3V)

The JTAG pins have a separate V_{CCJ} power supply that is independent of the bank V_{CCIO} supplies. V_{CCJ} determines the electrical characteristics of the LVCMOS JTAG pins, both the output high level and the input threshold.

Table 9-3 shows a summary of all the required power supplies.

Table 9-3. Power Supplies

Power Supply	Description	Value[1]
V_{CC}	Core Power Supply	1.2V
V_{CCIO}	Power Supply for the I/O and Configuration Banks	1.2V/1.5V/1.8V/2.5V/3.3V
V_{CCAUX}	Auxiliary Power Supply	3.3V
V_{CCJ}	Power Supply for JTAG Pins	1.2V/1.5V/1.8V/2.5V/3.3V

1. Refer to LatticeECP2/M Family Data Sheet for recommended min. and max. values.

Input Reference Voltage (V_{REF1}, V_{REF2})

Each bank can support up to two separate V_{REF} input voltages, V_{REF1} and V_{REF2}, that are used to set the threshold for the referenced input buffers. The locations of these V_{REF} pins are pre-determined within the bank. These pins can be used as regular I/Os if the bank does not require a V_{REF} voltage.

V_{REF1} for DDR Memory Interface

When interfacing to DDR memory, the V_{REF1} input must be used as the reference voltage for the DQS and DQ input from the memory. A voltage divider between V_{REF1} and GND is used to generate an on-chip reference voltage that is used by the DQS transition detector circuit. This voltage divider is only present on V_{REF1} it is not available on V_{REF2}. For more information on the DQS transition detect logic and its implementation, please refer to TN1105, LatticeECP2/M High-Speed I/O Interface. DDR1 follows the SSTL25_II signaling specification and DDR2 follows the SSTL18_II signaling specification.

Mixed Voltage Support in a Bank

The LatticeECP2/M sysIO buffer is connected to three parallel ratioed input buffers. These three parallel buffers are connected to V_{CCIO}, V_{CCAUX} and V_{CC}, giving support for thresholds that track with V_{CCIO} as well as fixed thresholds for 3.3V (V_{CCAUX}) and 1.2V (V_{CC}) inputs. This allows the input threshold for ratioed buffers to be assigned on a pin-by-pin basis rather than tracking with V_{CCIO}. This option is available for all 1.2V, 2.5V and 3.3V ratioed inputs and is independent of the bank V_{CCIO} voltage. For example, if the bank V_{CCIO} is 1.8V, it is possible to have 1.2V and 3.3V ratioed input buffers with fixed thresholds, as well as 2.5V ratioed inputs with tracking thresholds.

Prior to device configuration, the ratioed input thresholds always tracks the bank V_{CCIO}. This option only takes effect after configuration. Output standards within a bank are always set by V_{CCIO}. Table 9-4 shows the sysIO standards that can be mixed in the same bank.

Table 9-4. Mixed Voltage Support

	Input sysIO Standards					Output sysIO Standards				
V_{CCIO}	1.2V	1.5V	1.8V	2.5V	3.3V	1.2V	1.5V	1.8V	2.5V	3.3V
1.2V	Yes			Yes	Yes	Yes				
1.5V	Yes	Yes		Yes	Yes		Yes			
1.8V	Yes		Yes	Yes	Yes			Yes		
2.5V	Yes			Yes	Yes				Yes	
3.3V	Yes			Yes	Yes					Yes

sysIO Standards Supported by Bank

Table 9-5. I/O Standards Supported by Bank

Description	Top Side Banks 0-1	Right Side Banks 2-3	Bottom Side Banks 4-5	Left Side Banks 6-7
I/O Buffers	Single-ended	Single-ended and Differential	Single-ended	Single-ended and Differential
Output Standards Supported	LVTTL LVCMOS33 LVCMOS25 LVCMOS18 LVCMOS15 LVCMOS12 SSTL18 Class I, II SSTL25 Class I, II SSTL33 Class I, II HSTL15 Class I HSTL18_I, II SSTL18D Class I, II SSTL25D Class I, II SSTL33D Class I, II HSTL15D Class I HSTL18D Class I, II LVDS25E[1] LVPECL[1] BLVDS[1] RSDS[1]	LVTTL LVCMOS33 LVCMOS25 LVCMOS18 LVCMOS15 LVCMOS12 SSTL18 Class I, II SSTL25 Class I, II SSTL33 Class I, II HSTL15 Class I HSTL18 Class I, II SSTL18D Class I, II SSTL25D Class I, II SSTL33D Class I, II HSTL15D Class I HSTL18D Class I, II LVDS LVDS25E[1] LVPECL[1] BLVDS[1] RSDS[1]	LVTTL LVCMOS33 LVCMOS25 LVCMOS18 LVCMOS15 LVCMOS12 SSTL18 Class I, II SSTL2 Class I, II SSTL3 Class I, II HSTL15 Class I HSTL18 Class I, II SSTL18D Class I, II SSTL25D Class I, II, SSTL33D Class I, II HSTL15D Class I HSTL18D Class I, II PCI33 LVDS25E[1] LVPECL[1] BLVDS[1] RSDS[1]	LVTTL LVCMOS33 LVCMOS25 LVCMOS18 LVCMOS15 LVCMOS12 SSTL18 Class I, II SSTL2 Class I, II SSTL3 Class I, II HSTL15 Class I, III HSTL18 Class I, II, III SSTL18D Class I, II SSTL25D Class I, II, SSTL33D_I, II HSTL15D Class I HSTL18D Class I, II LVDS LVDS25E[1] LVPECL[1] BLVDS[1] RSDS[1]
Inputs	All Single-ended, Differential	All Single-ended, Differential	All Single-ended, Differential	All Single-ended, Differential
Clock Inputs	All Single-ended, Differential	All Single-ended, Differential	All Single-ended, Differential	All Single-ended, Differential
PCI Support	PCI33 without clamp	PCI33 without clamp	PCI33 with clamp	PCI33 without clamp PCI33 with clamp (ECP2M)
LVDS Output Buffers		LVDS (3.5mA) Buffers[2]		LVDS (3.5mA) Buffers[2]

1. These differential standards are implemented by using a complementary LVCMOS driver with external resistor pack.
2. Available only on 50% of the I/Os in the bank.

sysIO Buffer Configurations

All LVCMOS buffer have programmable pull, programmable drive and programmable slew configurations that can be set in the software.

Bus Maintenance Circuit

Each pad has a weak pull-up, weak pull-down and weak buskeeper capability. The pull-up and pull-down settings offer a fixed characteristic, which is useful in creating wired logic such as wired ORs. However, current can be slightly higher than other options, depending on the signal state. The bus-keeper option latches the signal in the last driven state, holding it at a valid level with minimal power dissipation. Users can also choose to turn off the bus maintenance circuitry, minimizing power dissipation and input leakage. Note that in this case, it is important to ensure that inputs are driven to a known state to avoid unnecessary power dissipation in the input buffer.

Programmable Drive

Each LVCMOS or LVTTL, as well as some of the referenced (SSTL and HSTL) output buffers, has a programmable drive strength option. This option can be set for each I/O independently. The drive strength settings available are 2mA, 4mA, 6mA, 8mA, 12mA, 16mA and 20mA. Actual options available vary by the I/O voltage. The user must consider the maximum allowable current per bank and the package thermal limit current when selecting the drive strength. Table 9-6 shows the available drive settings for each out the output standards.

Table 9-6. Programmable Drive Values for Single-ended Buffers

Single Ended I/O Standards	Programmable Drive (mA)
HSTL15_I/ HSTL15D_I	4, 8
HSTL18_I/ HSTL18D_I	8, 12
SSTL25_I/ SSTL25D_I	8, 12
SSTL25_II/ SSTL25D_II	16, 20
SSTL18_II/SSTL18D_II	8, 12
LVCMOS12	2, 6
LVCMOS15	4, 8
LVCMOS18	4, 8, 12, 16
LVCMOS25	4, 8, 12, 16, 20
LVCMOS33	4, 8, 12, 16, 20
LVTTL	4, 8, 12, 16, 20

Programmable Slew Rate

Each LVCMOS or LVTTL output buffer pin also has a programmable output slew rate control that can be configured for either low noise or high-speed performance. Each I/O pin has an individual slew rate control. This allows designers to specify slew rate control on a pin-by-pin basis. This slew rate control affects both the rising and falling edges.

Open-Drain Control

All LVCMOS and LVTTL output buffers can be configured to function as open drain outputs. The user can implement an open drain output by turning on the OPENDRAIN attribute in the software.

Differential SSTL and HSTL Support

The single-ended driver associated with the complementary ‘C’ pad can optionally be driven by the complement of the data that drives the single-ended driver associated with the true pad. This allows a pair of single-ended drivers to be used to drive complementary outputs with the lowest possible skew between the signals. This is used for driving complementary SSTL and HSTL signals (as required by the differential SSTL and HSTL clock inputs on synchronous DRAM and synchronous SRAM devices respectively). This capability is also used in conjunction with off-chip resistors to emulate LVPECL, and BLVDS output drivers.

PCI Support with Programmable PCICLAMP

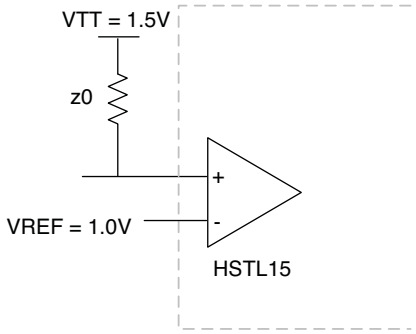
Each sysIO buffer can be configured to support PCI33. The buffers on the bottom of the device (for LatticeECP2) or on the left and bottom sides of the device (for LatticeECP2M) have an optional PCI clamp diode that may optionally be specified in the ispLEVER design tools.

Programmable PCICLAMP can be turned ON or OFF. This option is available on each I/O independently on the bottom side banks (for LatticeECP2) or on the left and bottom side banks (for LatticeECP2M).

GTL+ Input Support

GTL+ inputs are supported using the HSTL15_I input standard with VREF set to 1.0V and external VTT termination set to 1.5V. GTL+ inputs implemented using this method can support the maximum speed listed for the HSTL standard in the LatticeECP2/M Family Data Sheet . GTL+ outputs are not supported in the LatticeECP2 device.

Figure 9-2. GTL+ Input Buffer Emulation Using HSTL15 Input



Programmable Input Delay

Each input can optionally be delayed before it is passed to the core logic or input registers. The primary use for the input delay is to achieve zero hold time for the input registers when using a direct drive primary clock. To arrive at zero hold time, the input delay will delay the data by at least as much as the primary clock injection delay. This option can be turned ON or OFF for each I/O independently in the software using the FIXEDDELAY attribute. This attribute is described in more detail in the software sysIO attribute section. Appendix A shows how this feature can be enabled in the software using HDL attributes.

Software sysIO Attributes

sysIO attributes can be specified in the HDL, using the Preference Editor GUI or in the ASCII preference file (.prf) file directly. Appendices A, B and C list examples of how these can be assigned using each of these methods. This section describes each of these attributes in detail.

IO_TYPE

This is used to set the sysIO standard for an I/O. The V_{CCIO} required to set these I/O standards are embedded in the attribute names itself. There is no separate attribute to set the V_{CCIO} requirements. Table 9-7 lists the available I/O types.

Table 9-7. IO_TYPE Attribute Values

sysIO Signaling Standard	IO_TYPE
DEFAULT	LVCMOS25
LVDS 2.5V	LVDS25
RSDS	RSDS
Emulated LVDS 2.5V	LVDS25E[1]
Bus LVDS 2.5V	BLVDS25[1]
LVPECL 3.3V	LVPECL33[1]
HSTL18 Class I and II	HSTL18_I, HTSL18_II
Differential HSTL 18 Class I and II	HSTL18D_I, HSTL18D_II
HSTL 15 Class I	HSTL15_I
Differential HSTL 15 Class I	HSTL15D_I
SSTL 33 Class I and II	SSTL33_I, SSTL33_II
Differential SSTL 33 Class I and II	SSTL33D_I, SSTL3D_II
SSTL 25 Class I and II	SSTL25_I,SSTL25_II
Differential SSTL 25 Class I and II	SSTL25D_I, SSTL25D_II
SSTL 18 Class I and II	SSTL18_I, SSTL18_II
Differential SSTL 18 Class I	SSTL18D_I,SSTL18D_II
LVTTL	LVTTL33
3.3V LVCMOS	LVCMOS33
2.5V LVCMOS	LVCMOS25
1.8V LVCMOS	LVCMOS18
1.5V LVCMOS	LVCMOS15
1.2V LVCMOS	LVCMOS12
3.3V PCI	PCI33

1. These differential standards are implemented by using a complementary LVCMOS driver with external resistor pack.

OPENDRAIN

LVCMOS and LVTTL I/O standards can be set to open drain configuration by using the OPENDRAIN attribute.

Values: ON, OFF
Default: OFF

DRIVE

The DRIVE attribute will set the programmable drive strength for the output standards that have programmable drive capability

Table 9-8. DRIVE Settings

Output Standard	DRIVE (mA)	Default (mA)
HSTL15_I/ HSTL15D_I	4, 8	8
HSTL18_I/ HSTL18D_I	8, 12	12
SSTL25_I/ SSTL25D_I	8, 12	8
SSTL25_II/ SSTL25D_II	16, 20	16
SSTL18_II/SSTL18D_II	8, 12	12
LVCMOS12	2, 6	6
LVCMOS15	4, 8	8
LVCMOS18	4, 8, 12, 16	12
LVCMOS25	4, 8, 12, 16, 20	12
LVCMOS33	4, 8, 12, 16, 20	12
LVTTL	4, 8, 12, 16, 20	12

PULLMODE

The PULLMODE attribute is available for all the LVTLL and LVCMOS inputs and outputs. This attribute can be enabled for each I/O independently.

Values: UP, DOWN, NONE, KEEPER
Default: UP

Table 9-9. PULLMODE Values

PULL Options	PULLMODE Value
Pull-up (Default)	UP
Pull-down	DOWN
Bus Keeper	KEEPER
Pull Off	NONE

PCICLAMP

PCI33 inputs on the bottom of the device (for LatticeECP2) or on the left and bottom sides of the device (for LatticeECP2M) have an optional PCI clamp that is enabled via the PCICLAMP attribute. The PCICLAMP is also available for all LVCMOS33 and LVTTL inputs.

Values: ON, OFF
Default: OFF

Table 9-10. PCICLAMP Values

Input Type	PCICLAMP Value
PCI33	ON
LVCMOS33	OFF (default), ON
LVTTL	OFF (default), ON

SLEWRATE

The SLEWRATE attribute is available for all LVTTL and LVCMOS output drivers. Each I/O pin has an individual slew rate control. This allows a designer to specify slew rate control on a pin-by-pin basis.

Values: FAST, SLOW
Default: FAST

FIXEDDELAY

The FIXEDDELAY attribute is available to each input pin. This attribute, when enabled, is used to achieve zero hold time for the input registers when using global clock. This attribute can only be assigned in the HDL source.

Values: TRUE, FALSE
Default: FALSE

INBUF

By default, all the unused input buffers are disabled. The INBUF attribute is used to enable the unused input buffers when performing a boundary scan test. This is a global attribute and can be globally set to ON or OFF.

Values: ON, OFF
Default: OFF

DIN/DOUT

This attribute can be used to assign I/O registers. Using DIN will assert an input register and using the DOUT attribute will assert an output register. By default, the software will try to assign the I/O registers, if applicable. The user can turn this OFF by using the synthesis attribute or by using the Preference Editor of the ispLEVER software. These attributes can only be applied to registers.

LOC

This attribute can be used to make pin assignments to the I/O ports in the design. This attributes is only used when the pin assignments are made in HDL source. Designers can also assign pins directly using the GUI in the Preference Editor of the ispLEVER software. The appendices explain this in further detail.

Design Considerations and Usage

This section discusses some of the design rules and considerations that must be taken into account when designing with the LatticeECP2/M sysIO buffer

Banking Rules

- If V_{CCIO} or V_{CCJ} for any bank is set to 3.3 V, it is recommended that it be connected to the same power supply as $V_{CCAUX,}$ thus minimizing leakage.
- If V_{CCIO} or V_{CCJ} for any bank is set to 1.2V, it is recommended that it be connected to the same power supply as V_{CC}, thus minimizing leakage.
- When implementing DDR memory interfaces, the V_{REF1} of the bank is used to provide reference to the interface pins and cannot be used to power any other referenced inputs.
- Only the bottom banks for LatticeECP2 (Banks 4 and 5) or left and bottom banks for LatticeECP2M (Banks 4, 5, 6 and 7) will support PCI clamps.
- All legal input buffers should be independent of bank $V_{CCIO,}$ except for 1.8V and 1.5V buffers, which require a bank V_{CCIO} of 1.8V and 1.5V.

Differential I/O Rules

- All banks can support LVDS input buffers. Only the banks on the right and left sides (Banks 2, 3, 6 and 7) will support True Differential output buffers. The banks on the top and bottom will support the LVDS input buffers but will not support True LVDS outputs. The user can use emulated LVDS output buffers on these banks.
- All banks support emulated differential buffers using external resistor pack and complementary LVCMOS drivers.
- Only 50% of the I/Os on the left and right sides can provide LVDS output buffer capability. LVDS can only be assigned to the TRUE pad. The ispLEVER design tool will automatically assign the other I/Os of the differential pair to the complementary pad. Refer to the device data sheet to see the pin listings for all LVDS pairs.

Assigning V_{REF1}/ V_{REF2} Groups for Referenced Inputs

Each bank has two dedicated V_{REF} input pins, V_{REF1} and V_{REF2}. Designers can group buffers to a particular V_{REF} rail, V_{REF1} or V_{REF2}. This grouping is done by assigning a PGROUP VREF preference along with the LOCATE PGROUP preference.

Preference Syntax

```
PGROUP <pgrp_name> [(VREF <vref_name>)+] (COMP <comp_name>)+;
LOCATE PGROUP <pgrp_name> BANK <bank_num>;
LOCATE VREF <vref_name> SITE <site_name>;
```

Example Showing VREF Groups

```
PGROUP "vref_pg1" VREF "ref1" COMP "ah(0)" COMP "ah(1)" COMP "ah(2)" COMP "ah(3)"
COMP "ah(4)" COMP "ah(5)" COMP "ah(6)" COMP "ah(7)";

PGROUP "vref_pg2" VREF "ref2" COMP "al(0)" COMP "al(1)" COMP "al(2)" COMP "al(3)"
COMP "al(4)" COMP "al(5)" COMP "al(6)" COMP "al(7)";

LOCATE VREF "ref1" SITE PR29C;
LOCATE VREF "ref2" SITE PR48B;
```

or

```
LOCATE PGROUP " vref_pg1" BANK 2;
LOCATE PGROUP " vref_pg2" BANK 2;
```

The example shows two V_{REF} groups, "vref_pg1" assigned to VREF "ref1" and "vref_pg2" assigned to "ref2". The user must lock these V_{REF} to either V_{REF1} or V_{REF2} using the LOCATE preference. Alternatively, users can designate to which bank the V_{REF} group should be located. The software will then assign these to either the V_{REF1} or V_{REF2} of the bank.

If the PGROUP VREF is not used, the software will automatically group all pins that need the same V_{REF} reference voltage. This preference is most useful when there is more than one bus that uses the same reference voltage and the user wishes to associate each of these busses to different V_{REF} resources.

Differential I/O Implementation

The LatticeECP2/M devices support a variety of differential standards as detailed in the following sections.

LVDS

True LVDS (LVDS25) drivers are available on 50% of the I/Os on the left and right side of the devices. LVDS input support is provided on all sides of the device. All four sides of the device support LVDS using complementary LVCMOS drivers with external resistors (LVDS25E). Refer to the LatticeECP2/M Family Data Sheet for a detailed explanation of these LVDS implementations.

BLVDS

All single-ended sysIO buffers pairs support the Bus-LVDS standard using complementary LVCMOS drivers with external resistors. Please refer to the LatticeECP2/M Family Data Sheet for a detailed explanation of BLVDS implementation.

RSDS

All single-ended sysIO buffers pairs support RSDS standard using complementary LVCMOS drivers with external resistors. Please refer to the LatticeECP2/M Family Data Sheet for a detailed explanation of RSDS implementation.

LVPECL

All the sysIO buffers will support LVPECL inputs. LVPECL outputs are supported using complementary LVCMOS driver with external resistors. Please refer to the LatticeECP2/M Family Data Sheet for a detailed explanation of LVPECL implementation.

Differential SSTL and HSTL

All single-ended sysIO buffers pairs support differential SSTL and HSTL. Please refer to the LatticeECP2/M Family Data Sheet for a detailed explanation of Differential HSTL and SSTL implementation.

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
February 2006	01.0	Initial release.
September 2006	01.1	Updated to include LatticeECP2M support.
April 2007	01.2	Updated Supported Output Standards table.
June 2007	01.3	Updated V_{REF1} for DDR Memory Interface section.
June 2007	01.4	Updated sysIO Standards Supported by Bank table.
		Updated Banking Rules bullet list.
June 2007	01.5	Updated Power Supplies table.
April 2008	01.6	Updated LatticeECP2M sysIO Banking diagram.
February 2009	01.7	Updated I/O Standards Supported by Bank table.
June 2010	01.8	Updated screen shots in Appendix B.
		Updated document for Lattice Diamond design software support.
March 2011	01.9	Added support for GTL+ input standard using HSTL input buffer.

Appendix A. HDL Attributes for Synplicity® and Precision® RTL Synthesis

Using these HDL attributes, designers can assign the sysIO attributes directly in their source. The attribute definition and syntax for the appropriate synthesis vendor must be used. Below are a list of all the sysIO attributes, syntax and examples for Precision RTL Synthesis and Synplicity. This section only lists the sysIO buffer attributes for these devices. You can refer to the Precision RTL Synthesis and Synplicity user manuals for a complete list of synthesis attributes. These manuals are available through the ispLEVER software Help system.

VHDL Synplicity/Precision RTL Synthesis

This section lists syntax and examples for all the sysIO Attributes in VHDL when using the Precision RTL Synthesis or Synplicity synthesis tools.

Syntax

Table 9-11. VHDL Attribute Syntax for Synplicity and Precision RTL Synthesis

Attribute	Syntax
IO_TYPE	attribute IO_TYPE: string; attribute IO_TYPE of *Pinname*: signal is *"IO_TYPE Value";*
OPENDRAIN	attribute OPENDRAIN: string; attribute OPENDRAIN of *Pinname:* signal is *"OpenDrain Value";*
DRIVE	attribute DRIVE: string; attribute DRIVE of *Pinname:* signal is *"Drive Value";*
PULLMODE	attribute PULLMODE: string; attribute PULLMODE of *Pinname:* signal is *"Pullmode Value";*
PCICLAMP	attribute PCICLAMP: string; attribute PCICLAMP of *Pinname:* signal is *"PCIClamp Value";*
SLEWRATE	attribute PULLMODE: string; attribute PULLMODE of *Pinname:* signal is *"Slewrate Value";*
FIXEDDELAY	attribute FIXEDDELAY: string; attribute FIXEDDELAY of *Pinname:* signal is "*Fixeddelay Value*";
DIN	attribute DIN: string; attribute DIN of *Pinname*: signal is " ";
DOUT	attribute DOUT: string; attribute DOUT of *Pinname:* signal is " ";
LOC	attribute LOC: string; attribute LOC of *Pinname:* signal is "pin_locations";

Examples

IO_TYPE

```
--***Attribute Declaration***
ATTRIBUTE IO_TYPE: string;
--***IO_TYPE assignment for I/O Pin***
ATTRIBUTE IO_TYPE OF portA:    SIGNAL IS "PCI33";
ATTRIBUTE IO_TYPE OF portB:    SIGNAL IS "LVCMOS33";
ATTRIBUTE IO_TYPE OF portC:    SIGNAL IS "LVDS25";
```

OPENDRAIN

```
--***Attribute Declaration***
ATTRIBUTE OPENDRAIN: string;
--***DRIVE assignment for I/O Pin***
ATTRIBUTE OPENDRAIN OF portB: SIGNAL IS "ON";
```

DRIVE

```
--***Attribute Declaration***
ATTRIBUTE DRIVE: string;
--***DRIVE assignment for I/O Pin***
ATTRIBUTE DRIVE OF portB: SIGNAL IS “20”;
```

PULLMODE

```
--***Attribute Declaration***
ATTRIBUTE PULLMODE : string;
--***PULLMODE assignment for I/O Pin***
ATTRIBUTE PULLMODE OF portA:    SIGNAL IS “DOWN”;
ATTRIBUTE PULLMODE OF portB:    SIGNAL IS “UP”;
```

PCICLAMP

```
--***Attribute Declaration***
ATTRIBUTE PCICLAMP: string;
--***PULLMODE assignment for I/O Pin***
ATTRIBUTE PCICLAMP OF portA:    SIGNAL IS “ON”;
```

SLEWRATE

```
--***Attribute Declaration***
ATTRIBUTE SLEWRATE : string;
--*** SLEWRATE assignment for I/O Pin***
ATTRIBUTE SLEWRATE OF portB: SIGNAL IS “FAST”;
```

FIXEDDELAY

```
--***Attribute Declaration***
ATTRIBUTE FIXEDDELAY: string;
--*** SLEWRATE assignment for I/O Pin***
ATTRIBUTE FIXEDDELAY OF portB: SIGNAL IS “TRUE”;
```

DIN/DOUT

```
--***Attribute Declaration***
ATTRIBUTE din : string;
ATTRIBUTE dout : string;
--*** din/dout assignment for I/O Pin***
ATTRIBUTE din OF input_vector: SIGNAL IS “ “;
ATTRIBUTE dout OF output_vector: SIGNAL IS “ “;
```

LOC

```
--***Attribute Declaration***
ATTRIBUTE LOC : string;
--*** LOC assignment for I/O Pin***
ATTRIBUTE LOC OF input_vector: SIGNAL IS “E3,B3,C3 “;
```

Verilog Synplicity

This section lists syntax and examples for all the sysIO Attributes in Verilog using the Synplicity synthesis tool.

Syntax

Table 9-12. Verilog Synplicity Attribute Syntax

Attribute	Syntax
IO_TYPE	*PinType PinName* /* synthesis IO_TYPE=”IO_Type Value”*/;
OPENDRAIN	*PinType PinName* /* synthesis OPENDRAIN =”OpenDrain Value”*/;
DRIVE	*PinType PinName* /* synthesis DRIVE=”Drive Value”*/;
PULLMODE	*PinType PinName* /* synthesis PULLMODE=”Pullmode Value”*/;
PCICLAMP	*PinType PinName* /* synthesis PCICLAMP =” PCIClamp Value”*/;
SLEWRATE	*PinType PinName* /* synthesis SLEWRATE=”Slewrate Value”*/;
FIXEDDELAY	*PinType PinName* /* synthesis FIXEDDELAY=”Fixeddelay Value”*/;
DIN	*PinType PinName* /* synthesis DIN=” “*/;
DOUT	*PinType PinName* /* synthesis DOUT=” “*/;
LOC	*PinType PinName* /* synthesis LOC=”pin_locations “*/;

Examples

```
//IO_TYPE, PULLMODE, SLEWRATE and DRIVE assignment
output portB /*synthesis IO_TYPE=”LVCMOS33” PULLMODE =”UP” SLEWRATE =”FAST”
DRIVE =”20”*/;
output portC /*synthesis IO_TYPE=”LVDS25” */;

//OPENDRAIN
output portA /*synthesis OPENDRAIN =”ON”*/;

//PCICLAMP
output portA /*synthesis IO_TYPE=”PCI33” PULLMODE =”PCICLAMP”*/;

// Fixeddelay
input load /* synthesis FIXEDDELAY=”TRUE” */;

// Place the flip-flops near the load input
input load /* synthesis din=”” */;

// Place the flip-flops near the outload output
output outload /* synthesis dout=”” */;

//I/O pin location
input [3:0] DATA0 /* synthesis loc=”E3,B1,F3”*/;

//Register pin location
reg data_in_ch1_buf_reg3 /* synthesis loc=”R40C47” */;

//Vectored internal bus
reg [3:0] data_in_ch1_reg /*synthesis loc =”R40C47,R40C46,R40C45,R40C44” */;
```

Verilog Precision

This section lists syntax and examples for all the sysIO Attributes in Verilog using the Precision RTL Synthesis tool.

Syntax

Table 9-13. Verilog Precision Attribute Syntax

Attribute	Syntax
IO_TYPE	//pragma attribute *PinName* IO_TYPE IO_TYPE Value
OPENDRAIN	// pragma attribute *PinName* OPENDRAIN OpenDrain Value
DRIVE	// pragma attribute *PinName* DRIVE Drive Value
PULLMODE	// pragma attribute *PinName* IO_TYPE Pullmode Value
PCICLAMP	// pragma attribute *PinName* PCICLAMP PCIClamp Value
SLEWRATE	// pragma attribute *PinName* IO_TYPE Slewrate Value
FIXEDDELAY	// pragma attribute *PinName* IO_TYPE Fixeddelay Value
LOC	// pragma attribute *PinName* LOC pin_location

Examples

```
//****IO_TYPE ***
//pragma  attribute portA IO_TYPE PCI33
//pragma  attribute portB IO_TYPE LVCMOS33
//pragma  attribute portC IO_TYPE SSTL25_II

//*** Opendrain ***
//pragma  attribute portB OPENDRAIN ON
//pragma  attribute portD OPENDRAIN OFF

//*** Drive ***
//pragma  attribute portB DRIVE 20
//pragma  attribute portD DRIVE 8

//*** Pullmode***
//pragma  attribute portB PULLMODE UP

//*** PCIClamp***
//pragma  attribute portB PCICLAMP ON

//*** Slewrate ***
//pragma  attribute portB SLEWRATE FAST
//pragma  attribute portD SLEWRATE SLOW

// ***Fixeddelay***
// pragma  attribute load FIXEDDELAY TRUE

//***LOC***
//pragma  attribute portB loc E3
```

Appendix B. sysIO Attributes Using the ispLEVER Design Planner User Interface

Designers can assign sysIO buffer attributes using the Design Planner Spreadsheet View GUI available in the ispLEVER design tool. If you are using Lattice Diamond™ design software, refer to Appendix D. The Pin Attribute Sheet list all the ports in a design and all the available sysIO attributes as preferences. By clicking on each of these cells, a list of all the valid I/O preference for that port is displayed. Each column takes precedence over the next. Therefore, when a particular IO_TYPE is chosen, the DRIVE, PULLMODE and SLEWRATE columns will only list the valid combinations for that IO_TYPE. The pin locations can be locked using the pin location column of the Pin Attribute sheet. Right-clicking on a cell will list the available pin locations. The Preference Editor will also conduct a DRC check to search for any incorrect pin assignments.

Designers can enter DIN/DOUT preferences using the Cell Attributes sheet of the Preference Editor. All the preferences assigned using the Preference Editor are written into the preference file (.prf).

Figures 2 and 3 show the Pin Attribute sheet and the Cell Attribute sheet views of the preference editor. For further information on how to use the Preference Editor, refer to the ispLEVER Help documentation in the Help menu option of the software.

Figure 9-3. Pin Attributes Tab

	Type	Name	Group by	Pin	Bank	Vref	IO_TYPE	PULLMODE	DRIVE	SLEWRATE	PCICLAMP	OPENDRAIN
1	Clock Input	clk	N/A			N/A	LVCMOS25	UP	NA	FAST	OFF	OFF
2	Input Port	d0_0	N/A			N/A	LVCMOS25	UP	NA	FAST	OFF	OFF
3	Input Port	d0_1	N/A			N/A	LVCMOS25	UP	NA	FAST	OFF	OFF
4	Input Port	d0_2	N/A			N/A	LVCMOS25	UP	NA	FAST	OFF	OFF
5	Input Port	d0_3	N/A			N/A	LVCMOS25	UP	NA	FAST	OFF	OFF
6	Input Port	d0_4	N/A			N/A	LVCMOS25	UP	NA	FAST	OFF	OFF
7	Input Port	d0_5	N/A			N/A	LVCMOS25	UP	NA	FAST	OFF	OFF
8	Input Port	d0_6	N/A			N/A	LVCMOS25	UP	NA	FAST	OFF	OFF
9	Input Port	d0_7	N/A			N/A	LVCMOS25	UP	NA	FAST	OFF	OFF
10	Input Port	d1_0	N/A			N/A	LVCMOS25	UP	NA	FAST	OFF	OFF
11	Input Port	d1_1	N/A			N/A	LVCMOS25	UP	NA	FAST	OFF	OFF
12	Input Port	d1_2	N/A			N/A	LVCMOS25	UP	NA	FAST	OFF	OFF

Port Attributes | Clock Attributes | Net Attributes | Cell Attributes | Global | Block | Period/Frequency | In/Out Clock | MultiCycle/MaxDelay | Derating

Figure 9-4. Cell Attributes Tab

	Type	Name	Din/Dout	PIO Register
1	FlipFlops	q_i_7	DIN	True
2	FlipFlops	q_reg_0	DIN	True
3	FlipFlops	q_reg_1	DIN	True
4	**FlipFlops**	**q_reg_2**	DIN	True
5	FlipFlops	q_reg_3	DIN	True
6	FlipFlops	q_reg_4	DIN	True
7	FlipFlops	q_reg_5	DIN	True
8	FlipFlops	q_reg_6	DIN	True
9	FlipFlops	q_reg_7	DIN	True
10	FlipFlops	q_i_0	DIN	True
11	FlipFlops	q_i_1	DIN	True
12	FlipFlops	q_i_2	DIN	True
13	FlipFlops	q_i_3	DIN	True

Port Attributes | Clock Attributes | Net Attributes | Cell Attributes | Global | Block | Period

Appendix C. sysIO Attributes Using Preference File (ASCII File)

Designers can enter sysIO attributes directly in the preference (.prf) file as sysIO buffer preferences. The PRF file is an ASCII file containing two separate sections: a schematic section for those preferences created by the mapper or translator, and a user section for preferences entered by the user. User preferences can be written directly into this file. The synthesis attributes appear between the schematic start and schematic end of the file. The sysIO buffer preferences can be entered after the schematic end line using the preference file syntax. Below are a list of sysIO buffer preference syntax and examples.

IOBUF

This preference is used to assign the attribute IO_TYPE, PULLMODE, SLEWRATE and DRIVE.

Syntax

```
IOBUF [ALLPORTS | PORT <port_name> | GROUP <group_name>] (keyword=<value>)+;
```

where:

<port_name> = These are not the actual top-level port names, but should be the signal name attached to the port. PIOs in the physical design (.ncd) file are named using this convention. Any multiple listings or wildcarding should be done using GROUPs

Keyword = IO_TYPE, OPENDRAIN, DRIVE, PULLMODE, PCICLAMP, SLEWRATE.

Example

```
IOBUF PORT "port1" IO_TYPE=LVTTL33 OPENDRAIN=ON DRIVE=8 PULLMODE=UP
PCICLAMP =OFF SLEWRATE=FAST;
DEFINE GROUP "bank1" "in*" "out_[0-31]";
IOBUF GROUP "bank1" IO_TYPE=SSTL18_II;
```

LOCATE

When this preference is applied to a specified component, it places the component at a specified site and locks the component to the site. If applied to a specified macro instance, it places the macro's reference component at a specified site, places all of the macro's pre-placed components (that is, all components that were placed in the macro's library file) in sites relative to the reference component, and locks all of these placed components at their sites. This can also be applied to a specified PGROUP.

Syntax

```
LOCATE [COMP <comp_name> | MACRO <macro_name>] SITE <site_name>;
LOCATE PGROUP <pgroup_name> [SITE <site_name>; | REGION <region_name>;]
LOCATE PGROUP <pgroup_name> RANGE <site_1> [<site_2> | <count>] [<direction>] |
RANGE <chip_side> [<direction>];
LOCATE BUS < bus_name> ROW|COL <number>;
<bus_name> := string
<number>  := integer
```

Note: If the comp_name, macro_name, or site_name begins with anything other than an alpha character (for example, "11C7"), you must enclose the name in quotes. Wildcard expressions are allowed in <comp_name>.

Examples

This command places the port Clk0 on the site A4:

```
LOCATE COMP "Clk0" SITE "A4";
```

This command places the component PFU1 on the site named R1C7:

```
LOCATE COMP "PFU1" SITE "R1C7";
```

This command places bus1 on ROW 3 and bus2 on COL4

```
LOCATE BUS "bus1" ROW 3;
LOCATE BUS "bus2" COL 4;
```

USE DIN CELL

This preference specifies the given register to be used as an input flip-flop.

Syntax

```
USE DIN CELL <cell_name>;
```

where:

```
<cell_name> := string
```

Example

```
USE DIN CELL "din0";
```

USE DOUT CELL

Specifies the given register to be used as an output flip-flop.

Syntax

```
USE DOUT CELL <cell_name>;
```

where:

```
<cell_name> := string
```

Example

```
USE DOUT CELL "dout1";
```

PGROUP VREF

This preference is used to group all the components that need to be associated to one V_{REF} pin within a bank.

Syntax

```
PGROUP <pgrp_name> [(VREF <vref_name>)+] (COMP <comp_name>)+;
LOCATE PGROUP <pgrp_name> BANK <bank_num>;
LOCATE VREF <vref_name> SITE <site_name>;
```

Example

```
PGROUP "vref_pg1" VREF "ref1" COMP "ah(0)" COMP "ah(1)" COMP "ah(2)" COMP "ah(3)"
COMP "ah(4)" COMP "ah(5)" COMP "ah(6)" COMP "ah(7)";
PGROUP "vref_pg2" VREF "ref2" COMP "al(0)" COMP "al(1)" COMP "al(2)" COMP "al(3)"
COMP "al(4)" COMP "al(5)" COMP "al(6)" COMP "al(7)";
LOCATE VREF "ref1" SITE PR29C;
LOCATE VREF "ref2" SITE PR48B;
```

or:

```
LOCATE PGROUP " vref_pg1" BANK 2;
LOCATE PGROUP " vref_pg2" BANK 2;
```

Appendix D. Assigning sysIO Attributes Using Lattice Diamond Spreadsheet View

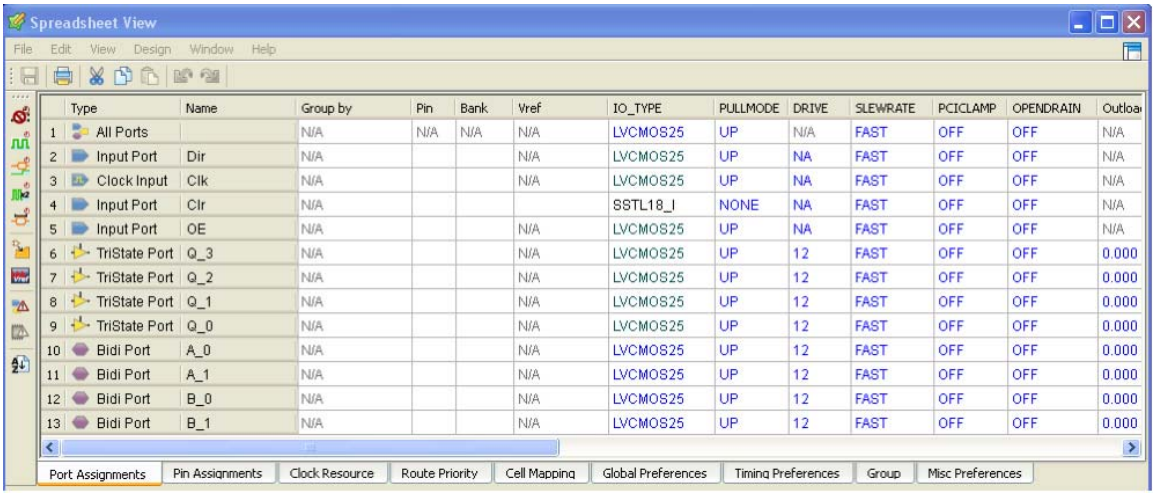
sysIO buffer attributes can be assigned using the Spreadsheet View available in the Lattice Diamond design software. The Port Assignments Sheet lists all the ports in a design and all the available sysIO attributes in multiple columns. Click on each of these cells for a list of all the valid I/O preferences for that port. Each column takes precedence over the next. Therefore, when you choose a particular IO_TYPE, the columns for the DRIVE, PULLMODE, SLEWRATE and other attributes will only list the valid entries for that IO_TYPE.

Pin locations can be locked using the Pin column of the Port Assignments sheet or using the Pin Assignments sheet. You can right-click on a cell and go to Assign Pins to see a list of available pins.

The Spreadsheet View also has an option to run a DRC check to check for any incorrect pin assignments. You can enter the DIN/ DOUT preferences using the Cell Mapping Tab. All the preferences assigned using the Spreadsheet View are written into the logical preference file (.lpf).

Figure 9-5 shows the Port Assignments Sheet of the Spreadsheet View. For further information on how to use the Spreadsheet View, refer to the Diamond Help documentation, available in the Help menu option of the software.

Figure 9-5. Port Attributes Tab of the SpreadSheet View



	Type	Name	Group by	Pin	Bank	Vref	IO_TYPE	PULLMODE	DRIVE	SLEWRATE	PCICLAMP	OPENDRAIN	Outloa
1	All Ports		N/A	N/A	N/A	N/A	LVCMOS25	UP	N/A	FAST	OFF	OFF	N/A
2	Input Port	Dir	N/A			N/A	LVCMOS25	UP	NA	FAST	OFF	OFF	N/A
3	Clock Input	Clk	N/A			N/A	LVCMOS25	UP	NA	FAST	OFF	OFF	N/A
4	Input Port	Clr	N/A				SSTL18_I	NONE	NA	FAST	OFF	OFF	N/A
5	Input Port	OE	N/A			N/A	LVCMOS25	UP	NA	FAST	OFF	OFF	N/A
6	TriState Port	Q_3	N/A			N/A	LVCMOS25	UP	12	FAST	OFF	OFF	0.000
7	TriState Port	Q_2	N/A			N/A	LVCMOS25	UP	12	FAST	OFF	OFF	0.000
8	TriState Port	Q_1	N/A			N/A	LVCMOS25	UP	12	FAST	OFF	OFF	0.000
9	TriState Port	Q_0	N/A			N/A	LVCMOS25	UP	12	FAST	OFF	OFF	0.000
10	Bidi Port	A_0	N/A			N/A	LVCMOS25	UP	12	FAST	OFF	OFF	0.000
11	Bidi Port	A_1	N/A			N/A	LVCMOS25	UP	12	FAST	OFF	OFF	0.000
12	Bidi Port	B_0	N/A			N/A	LVCMOS25	UP	12	FAST	OFF	OFF	0.000
13	Bidi Port	B_1	N/A			N/A	LVCMOS25	UP	12	FAST	OFF	OFF	0.000

Users can create a VREF pin using the Spreadsheet View as shown in Figure 9-6 and then assign VREF for a bank using the VREF Column in the Ports Assignment Tab of the Spreadsheet View as shown in Figure 9-7. See the Diamond online help for a detailed description of this setting.

Figure 9-6. Creating a VREF in Spreadsheet View

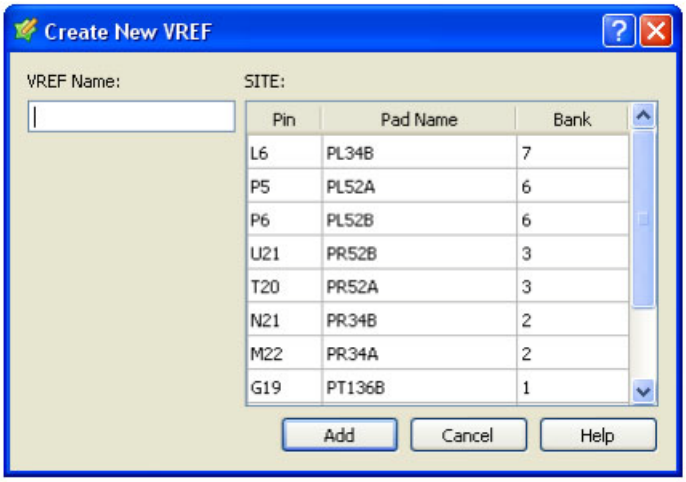
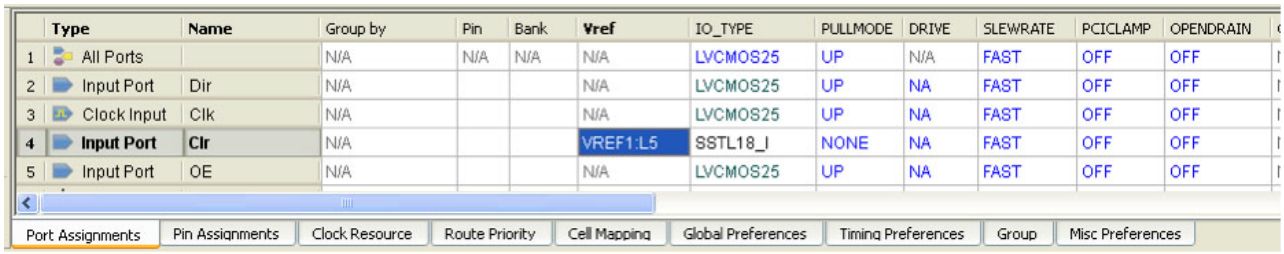


Figure 9-7. Assigning VREF for an Input Port in Spreadsheet View



LatticeECP2/M sysCLOCK PLL/DLL Design and Usage Guide

June 2010 Technical Note TN1103

Introduction

This user's guide describes the clock resources available in the LatticeECP2™ and LatticeECP2M™ device architectures. Details are provided for primary clocks, secondary clocks and edge clocks, as well as clock elements such as PLLs, DLLs, Clock Dividers and more.

The number of PLLs and DLLs for each package can be found in Tables 10-1 and 10-2.

Table 10-1. Number of PLLs and DLLs: LatticeECP2 Family

Device	Description	ECP2-6	ECP2-12	ECP2-20	ECP2-35	ECP2-50	ECP2-70
Number of SPLLs	Standard PLL (Subset of GPLL)	0	0	0	0	2	4
Number of GPLLs	General Purpose PLL	2	2	2	2	2	2
Number of DLLs	General Purpose DLL	2	2	2	2	2	2
Number of DQSDLLs	DLL for DDR Applications	2	2	2	2	2	2

Table 10-2. Number of PLLs, DLLs and SERDES: LatticeECP2M Family

Device	Description	ECP2M-20	ECP2M-35	ECP2M-50	ECP2M-70	ECP2M-100
Number of SPLLs	Standard PLL (Subset of GPLL)	6	6	6	6	6
Number of GPLLs	General Purpose PLL	2	2	2	2	2
Number of DLLs	General Purpose DLL	2	2	2	2	2
Number of DQSDLLs	DLL for DDR Applications	2	2	2	2	2
SERDES	4-Channel Quad SERDES	1	1	2	4	4

Clock/Control Distribution Network

The LatticeECP2/M family provides global clock distribution in the form of eight quadrant-based primary clocks and flexible secondary clocks. The devices also provide two edge clocks on each edge of the device. Other clock sources include clock input pins, internal nodes, PLLs, DLLs, Slave Delay Lines and Clock Dividers.

LatticeECP2/M Top Level View

Figure 10-1 shows the primary clocking structure of the LatticeECP2-50 device.

Figure 10-1. LatticeECP2-50 Clocking Structure

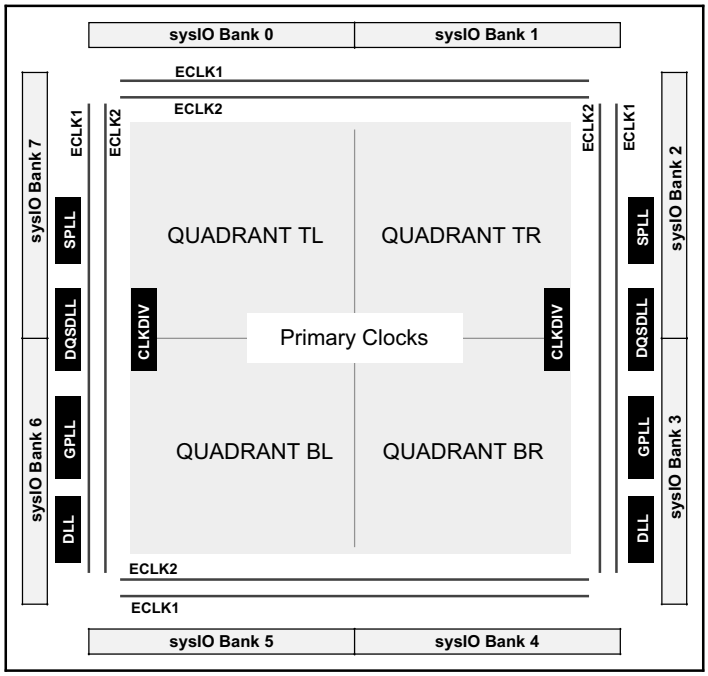
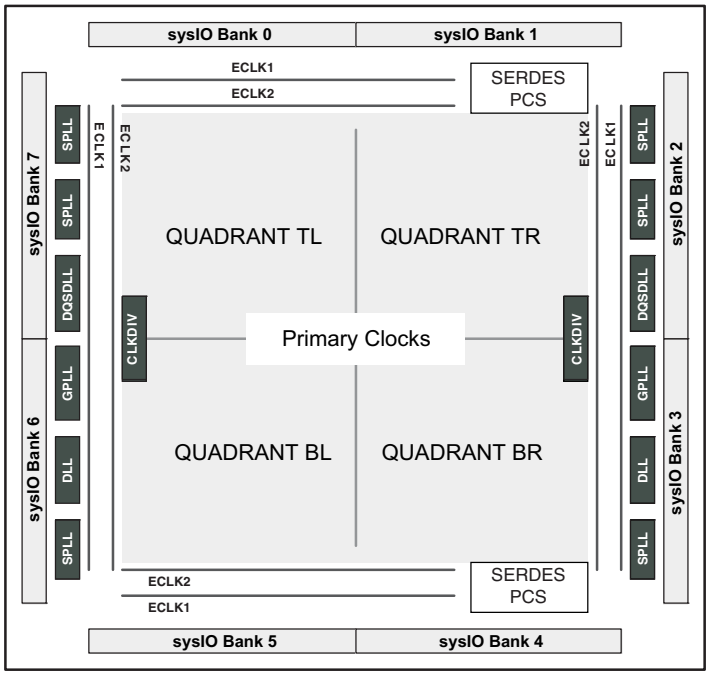


Figure 8-2 illustrates the primary clocking structure of the LatticeECP2M-50 device. The figure shows two SERDES blocks. The edge clocks on the top and bottom sides stop when they reach the SERDES block boundary. Other members of the LatticeECP2M family have a similar structure, except for the number of SERDES blocks.

Figure 10-2. LatticeECP2M-50 Clocking Structure



Primary Clocks

Each quadrant receives up to eight primary clocks. Two of these clocks provide the dynamic clock selection (DCS) feature. The six primary clocks without DCS can be specified in the Spreadsheet View in the ispLEVER Design Planner (or **Tools > Spreadsheet View** in the Lattice Diamond™ design software) as 'Primary Pure' and the two DCS clocks as 'Primary-DCS'.

The sources of the primary clocks are:

- PLL outputs
- DLL outputs
- CLKDIV outputs
- Dedicated clock pins
- Internal nodes
- SERDES TX_H_CLK (LatticeECP2M only)

Secondary Clocks

The LatticeECP2/M secondary clocks are a flexible region-based clocking resource. Each region can have four independent clock inputs. As a regional resource, it can cross the primary clock quadrant boundaries.

There are eight secondary clock muxes per region. Each mux has inputs from four different sources. Three of these are from internal nodes. The fourth input comes from a primary clock pin. The input sources are not necessarily located in the same region as the Mux. This structure enables global usage of secondary clocks.

The sources of secondary clocks are:

- Dedicated clock pins on right and left sides of device (PCLKT2, PCLKT3, PCLKT6, PCLKT7)
- Internal nodes

Edge Clocks

The LatticeECP2/M has two edge clocks per side. These clocks, which have low injection times and skew, are used to clock I/O registers. The edge clock (ECLK) resources are designed for high speed I/O interfaces with high fanout capability. Refer to Appendix B for detailed connectivity information.

The sources of the edge clocks are:

- Left and Right Edge Clocks
  - Dedicated clock pins
  - PLL outputs
  - DLL outputs
  - Internal nodes
- Top and Bottom Edge Clocks
  - Dedicated clock pins
  - Internal nodes

ECLK can directly drive the secondary clock resources and general routing resources. This means that an ECLK source clock can also route to the Primary Clock Net through general routing at the same time.

Figure 10-3 describes the secondary clock and edge clock structure.

Figure 10-3. LatticeECP2-50 Secondary Clocks and Edge Clocks

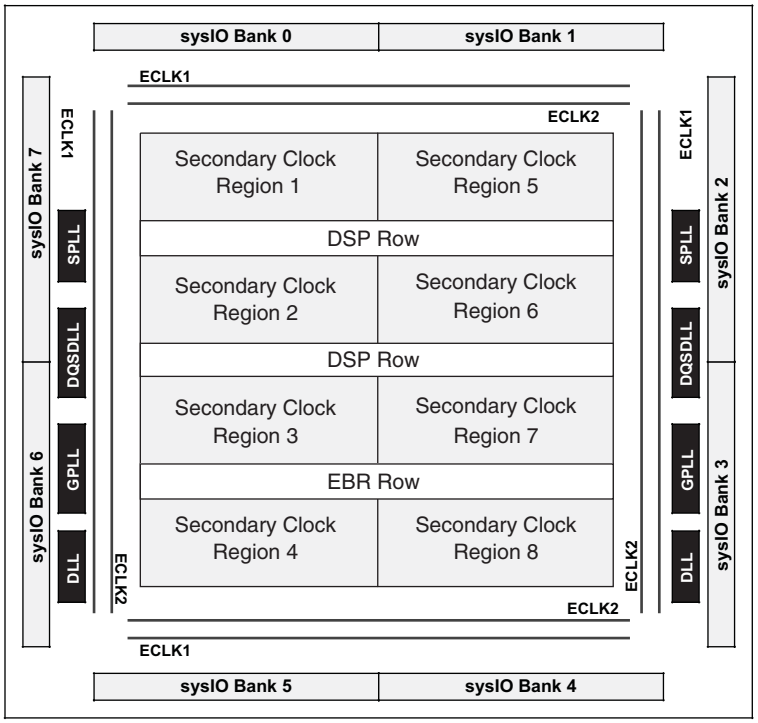
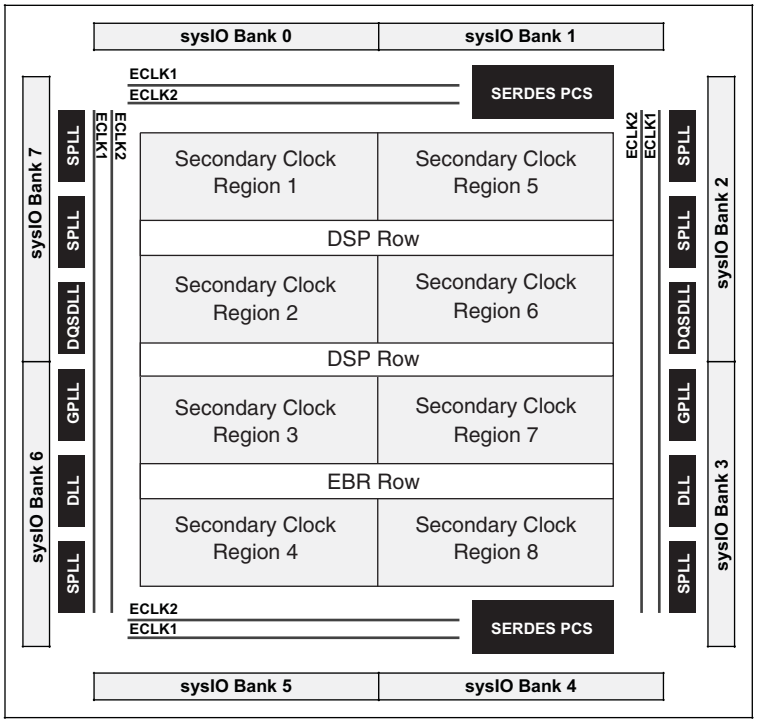


Figure 10-4. LatticeECP2M-50 Secondary Clocks and Edge Clocks



Note on Primary Clocks

The CLKOP must be used as the feedback source to optimize the PLL performance.

Most designers use PLL for clock tree injection removal mode and the CLKOP should be assigned to the Primary Clock. This is done automatically by the software unless the user specifies otherwise.

CLKOP can route to CLK0 to CLK5 only and CLKOS/CLKOK can route to all Primary Clocks (CLK0 to CLK7).

When CLK6 or CLK7 is used as a Primary Clock and there is only one clock input to the DCS, the DCS is assigned as a buffer mode by the software. See the DCS section of this document for further information.

Specifying Clocks in the Design Tools

If desired, designers can specify the clock resources, primary, secondary or edge to be used to distribute a given clock source. Figure 10-4 illustrates how this can be done in the Spreadsheet View in the ispLEVER Design Planner (or **Tools > Spreadsheet View** in Diamond). Alternatively the Preference file can be used, as discussed in Appendix C.

Primary-Pure and Primary-DCS

Primary Clock Net can be assigned to either Primary-Pure (CLK0 to CLK5) or Primary-DCS (CLK6 and CLK7).

Global Primary Clock and Quadrant Primary Clock

Global Primary Clock

If a primary clock is not assigned as a quadrant clock, the software assumes it is a Global Clock.

There are six Global Primary/Pure Clocks and two Global Primary/DCS Clocks available.

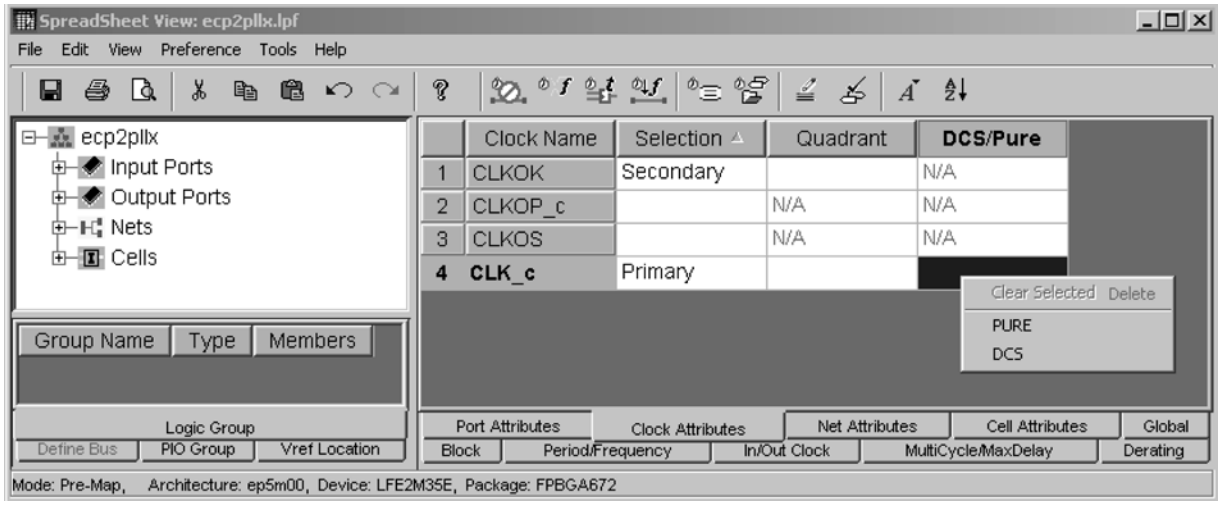
Quadrant Primary Clock

Any Primary Clock may be assigned to a Quadrant Clock. The clock may be assigned to a single quadrant or to two adjacent quadrants (not diagonally adjacent).

When a quadrant clock net is used, the user must ensure that the registers each clock drives can be assigned in that quadrant without any routing issues.

In the Quadrant Primary Clocking scheme, the maximum number of Primary Clocks is 32, as long as all the Primary Clock sources are available.

Figure 10-5. Design Planner Spreadsheet View (see Appendix D Figure 10-39 for Diamond Equivalent)



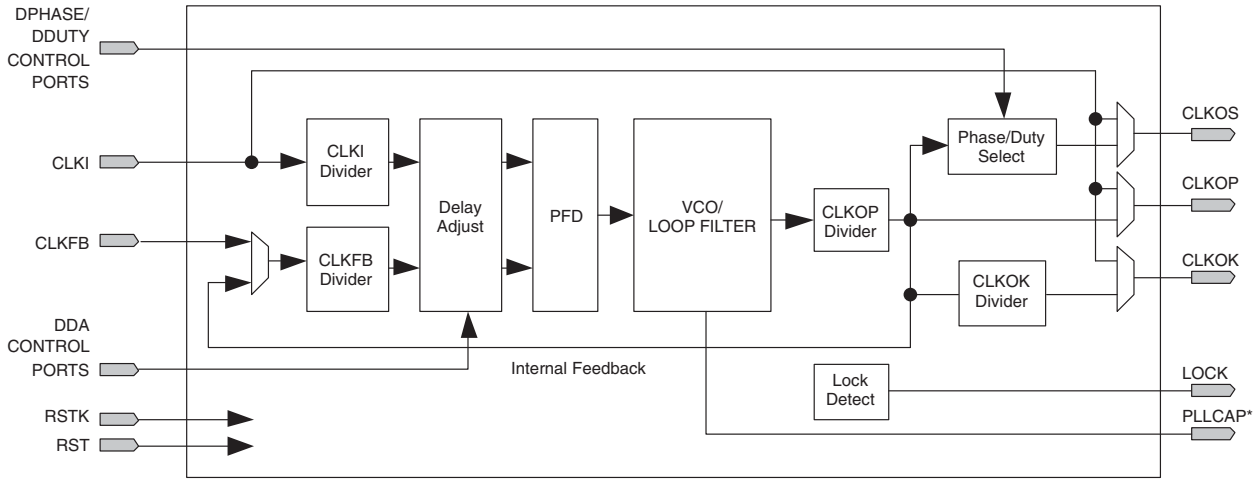
Note on Edge Clocks

Refer to Appendix A for detailed clock network diagrams.

sysCLOCK PLL

The LatticeECP2/M PLL provides features such as clock injection delay removal, frequency synthesis, phase/duty cycle adjustment, and dynamic delay adjustment. Figure 10-6 shows the block diagram of the PLL.

Figure 10-6. PLL Block Diagram



Functional Description

PLL Divider and Delay Blocks

Input Clock (CLKI) Divider

The CLKI divider is used to control the input clock frequency into the PLL block. The divider setting directly corresponds to the divisor of the output clock. The input and output of the input divider must be within the input and output frequency ranges specified in the device data sheet.

Feedback Loop (CLKFB) Divider

The CLKFB divider is used to divide the feedback signal. Effectively, this multiplies the output clock, because the divided feedback must speed up to match the input frequency into the PLL block. The PLL block increases the output frequency until the divided feedback frequency equals the input frequency. The input and output of the feedback divider must be within the input and output frequency ranges specified in the device data sheet.

Delay Adjustment

The delay adjust circuit provides programmable clock delay. The programmable clock delay allows for step delays in increments of 130ps (nominal) for a total of 1.04ns, lagging or leading. The time delay setting has a tolerance. See the device data sheet for details. Under this mode, CLKOP, CLKOS and CLKOK are identically affected. The delay adjustment has two modes of operation:

- **Static Delay Adjustment**: In this mode, the user-selected delay is configured at power-up.
- **Dynamic Delay Adjustment (DDA)**: In this mode, a simple bus is used to configure the delay. The bus signals are available to the general purpose FPGA.

Output Clock (CLKOP) Divider

The CLKOP divider serves the dual purposes of squaring the duty cycle of the VCO output and scaling up the VCO frequency into the 640MHz to 1280MHz range to minimize jitter. The CLKOP Divider values are the same whether or not CLKOS is used.

CLKOK Divider

The CLKOK divider acts as a source for the global clock nets. It divides the CLKOP signal of the PLL by the value of the divider to produce a lower frequency clock.

Phase Adjustment and Duty Cycle Select

Users can program CLKOS with Phase and Duty Cycle options. Phase adjustment can be done in 22.5° steps. The Duty Cycle resolution is 1/16th of a period. However, 1/16th and 15/16th duty cycle options are not supported to avoid minimum pulse violation.

Dynamic Phase Adjustment (DPHASE) and Dynamic Duty Cycle (DDUTY) Select
With LatticeECP2/M device families, users can control the Phase Adjustment and Duty Cycle Select in dynamic mode. When this mode is selected, both the Phase Adjustment and Duty Cycle Select must be in Dynamic mode. If only one of the features is to be used in Dynamic mode, the other control inputs can be set with the fixed logic levels desired.

External Capacitor
An optional external capacitor can be used with PLLs to accommodate low frequency input clocks. See the Optional External Capacitor section of this document for further information.

PLL Inputs and Outputs

CLKI Input
The CLKI signal is the reference clock for the PLL. It must conform to the specifications in the LatticeECP2/M Family Data Sheet for the PLL to operate correctly. The CLKI can be derived from a dedicated dual-purpose pin or from routing.

RST Input
The PLL reset occurs under two conditions. At power-up an internal power-up reset signal from the configuration block resets the PLL. The user-controlled PLL reset signal RST is provided as part of the PLL module that can be driven by an internally generated reset function or a pin. This RST signal resets all internal PLL counters, flip-flops (including M-Dividers), and the charge pump. The M-Divider reset synchronizes the M-Divider output to the input clock. When RST goes inactive, the PLL will start the lock-in process, and will take the t_{LOCK} time to complete the PLL lock. Figure 10-7 shows the timing diagram of the RST Input. RST is active high.

The RESET signal is optional.

Figure 10-7. RST Input Timing Diagram

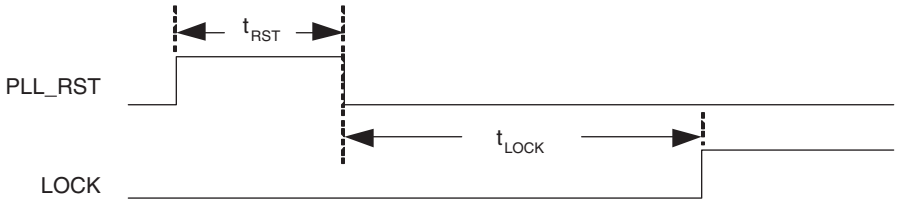
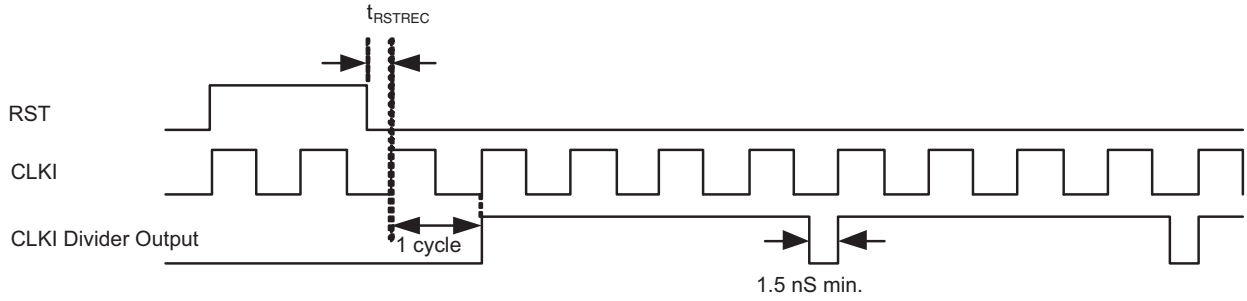


Figure 10-8 shows the timing relationship between RST and the CLKI Divider Output.

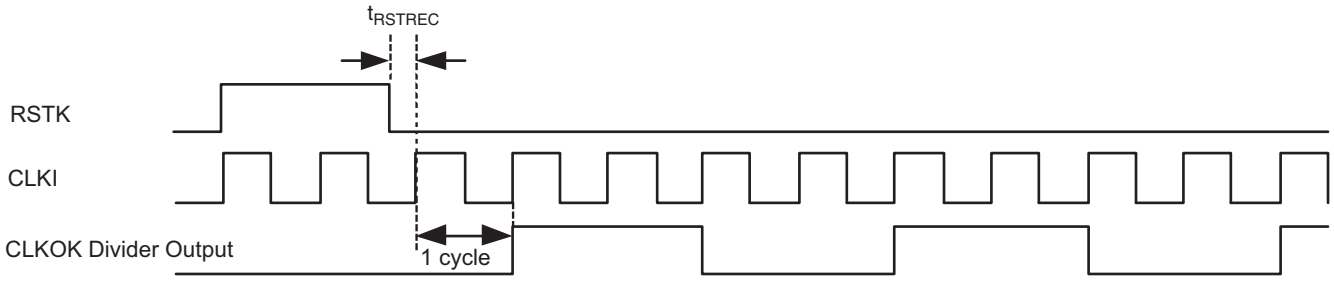
Figure 10-8. RST Input and CLKI Divider Output Timing Diagram (Example: CLKI_DIV = 4)



RSTK Input
RSTK is the reset input for the K-Divider. The K-Divider reset is used to synchronize the K-Divider output clock to the input clock. The LatticeECP2/M has an optional gearbox in the I/O cell for both outputs and inputs. The K-Divider reset is useful for the gearbox implementation. RSTK is active high.

Figure 10-9 shows the timing relationship between RSTK and CLKOK (example: CLKOK_DIV = 4)

Figure 10-9. RSTK Input and CLKOK Divider Output Timing Diagram (Example: CLKOK_DIV = 4)



CLKFB Input
The feedback signal to the PLL, which is fed through the feedback divider, can be derived from the Primary Clock net (CLKOP), a preferred pin, directly from the CLKOP divider (internal feedback) or from general routing. External feedback allows the designer to compensate for board-level clock alignment.

CLKOP Output
The sysCLOCK PLL main clock output, CLKOP, is a signal available for selection as a primary clock and an edge clock. This clock signal is available at the CLK_OUT pin.

CLKOS Output with Phase and Duty Cycle Select
The sysCLOCK PLL auxiliary clock output, CLKOS, is a signal available for selection as a primary clock and an edge clock. The CLKOS is used when phase shift and/or duty cycle adjustment is desired. The programmable phase shift allows for different phases in increments of 22.5°. The duty select feature provides duty selection in 1/16th of the clock period. This feature is also supported in Dynamic Control Mode.

CLKOK Output with Lower Frequency
The CLKOK is used when a lower frequency is desired. This signal is available for selection as a primary clock.

Dynamic Delay Control/Dynamic Phase Adjustment/Dynamic Duty Cycle
Detailed information about these features are described later in this document. The I/O ports for these features are illustrated in Table 10-3.

Table 10-3. Dynamic Delay Adjust and Dynamic Phase and Duty Cycle Adjust Ports

Parameter	I/O	Description
DDAMODE	I	DDA (Dynamic Delay Adjust) Mode. 1": Pin control (dynamic), “0”: Fuse Control (static)
DDAIZR	I	DDA Delay Zero. “1”: delay = 0, “0”: delay = on
DDAILAG	I	DDA Lag/Lead. “1”: Lead, “0”: Lag
DDAIDEL[2:0}	I	DDA Delay Step value
DPAMODE	I	DPA (Dynamic Phase Adjust/Duty Cycle Select) mode. 1": Pin Pin control (dynamic), “0”: Fuse Control (static)
DPHASE[3:0]	I	DPA Phase Adjust inputs
DDUTY[3:0]	I	DPA Duty Cycle Select inputs

LOCK Output
The LOCK output provides information about the status of the PLL. After the device is powered up and the input clock is valid, the PLL will achieve lock within the specified lock time. Once lock is achieved, the PLL lock signal will

be asserted. If, during operation, the input clock or feedback signals to the PLL become invalid, the PLL will lose lock. However, when the input clock completely stops, the LOCK output will remain in its last state, since it is internally registered by this clock. It is recommended to assert PLL RST to re-synchronize the PLL to the reference clock. The LOCK signal is available to the FPGA routing to implement generation of RST. Simulation models take several reference clock cycles from RST release to LOCK high.

PLLCAP
This port is not included in the software module. Instead, it is hard-wired to the PLLCAP pin of the device. See the Optional External Capacitor section of this document for further information.

PLL Attributes

The PLL utilizes several attributes that allow the configuration of the PLL through source constraints and preference files. The following section details these attributes and their usage.

FIN
The input frequency can be any value within the specified frequency range based on the divider settings.

CLKI_DIV, CLKFB_DIV, CLKOP_DIV, CLKOK_DIV
These dividers determine the output frequencies of each output clock. The user is not allowed to input an invalid combination. This is determined by the input frequency, the dividers and the PLL specifications.

Note: Unlike PLLs in the LatticeECP™, LatticeEC™, LatticeXP™ and MacoXO™ devices, the CLKOP Divider values are the same whether or not CLKOS is used. The CLKOP_DIV value is calculated to maximize the f_{VCO} within the specified range based on FIN and CLKOP_FREQ in conjunction with CLKI_DIV and CLKFB_DIV values. These value settings are designed so that the output clock duty cycle is as close to 50% as possible.

FREQUENCY_PIN_CLKI, FREQUENCY_PIN_CLKOP, FREQUENCY_PIN_CLKOK
These input and output clock frequencies determine the divider values.

CLKOP Frequency Tolerance
When the desired output frequency is not achievable, the frequency tolerance of the clock output may be entered.

PHASEADJ (Phase Shift Adjust)
The PHASEADJ attribute is used to select Phase Shift for the CLKOS output. The phase adjustment is programmable in 22.5° increments.

DUTY (Duty Cycle)
The DUTY attribute is used to select the Duty Cycle for CLKOS output. The Duty Cycle is programmable at 1/16th of the period increment. Steps 2 to 14 are supported. 1/16th and 15/16th duty cycles are not supported to avoid the minimum pulse width violation.

FB_MODE
There are three sources of feedback signals that can drive the CLKFB Divider: Internal, CLKOP (Clock Tree) and User Clock. CLKOP (Clock Tree) feedback is used by default. Internal feedback takes the CLKOP output at the CLKOP Divider output (CLKINTFB) before the Clock Tree to minimize the feedback path delay. User Clock feedback is driven from the dedicated pin, clock pin or user specified internal logic.

DELAY_CNTL
This attribute is designed to select the Delay Adjustment mode. If the attribute is set to “DYNAMIC” the delay control switches between dynamic and static, depending upon the input logic of the DDAMODE pin. If the attribute is set to “STATIC”, Dynamic Delay inputs are ignored in this mode.

PHASE/DUTY_CNTL
This attribute is designed to select the Phase Adjustment/Duty Cycle Select mode. If the attribute is set to "DYNAMIC" the Phase Adjustment/Duty Cycle Select control switches between dynamic and static, depending upon the input logic of the DPAMODE pin. If the attribute is set to "STATIC", Dynamic Phase Adjustment/Duty Cycle Select inputs are ignored in this mode.

CLKOS/CLKOK Select
Users select these output clocks only when they are used in the design.

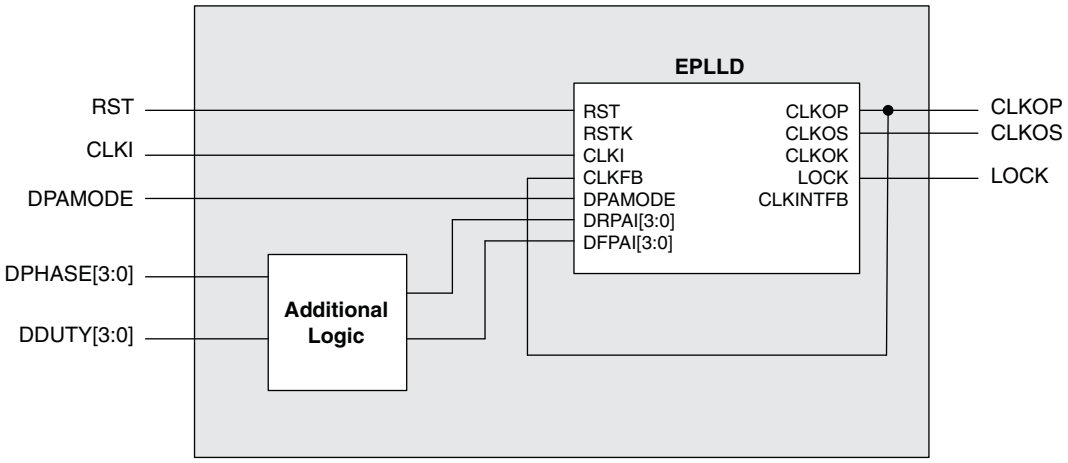
CLKOP/CLKOS/CLKOK BYPASS
These bypasses are enabled if set. The CLKI is routed directly to the corresponding output clock.

RESET/RSTK Select
Users select these reset signals only when they are used in the design.

LatticeECP2/M PLL Modules

When the user creates a PLL module using IPexpress, the module will consist of a wrapper around the PLL library element and any additional logic required for the module. Figure 10-10 shows a diagram of a typical PLL module. The module port names can be different than the library element is some cases. The user will see the module port names in the IPexpress window and also in the source code file for the generated module. These are the ports that will be connected in the user's design. IPexpress also creates an instantiation template file that shows the user how to instantiate the PLL module in their design. The user can import the *.LPC (or *.IPX for Diamond) file into their project or the generated source code file.

Figure 10-10. LatticeECP2/M Typical PLL Module Generated by IPexpress



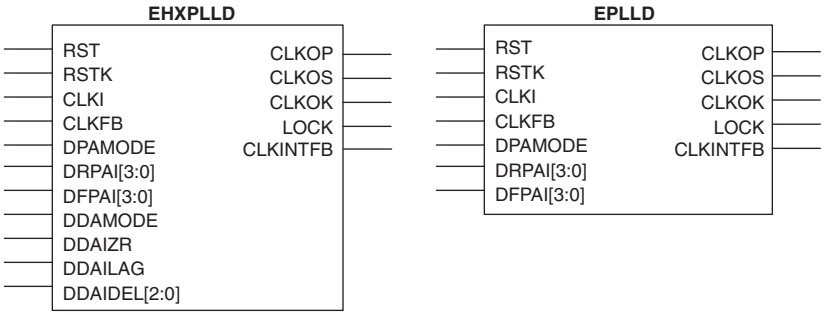
The PLL module shown in Figure 10-10 represents an example where the user has chosen to use the CLKOP and CLKOS ports, with a PLL reset signal, PLL lock signal, and dynamic phase and dynamic duty cycle. It also uses CLKOP feedback so the software will connect the CLKOP signal to the CLKFB port and use the primary clock tree to route this signal. The user would connect their signals to the CLKI, RST, DPAMODE, DPHASE[3:0], DDUTY[3:0], CLKOP, CLKOS, and LOCK signals.

LatticeECP2/M PLL Library Definitions

All LatticeECP2/M devices support two General Purpose PLLs (GPLLs) which are full-featured PLLs. In addition, some of the larger devices have two to six Standard PLLs (SPLLs) that have a subset of the GPLL functionalities.

Two PLL library elements are defined for LatticeECP2/M PLL implementation. Figure 10-11 shows the LatticeECP2/M PLL library symbols. The GPLL may be configured as either EPLLD or EHXPLLD. The SPLL can be configured as EPLLD only.

Figure 10-11. LatticeECP2/M PLL Library Symbols



Dynamic Delay Adjustment (EHXPLLD Only)

The Dynamic Delay Adjustment is controlled by the DDAMODE input. When the DDAMODE input is set to "1", the delay control is done through the inputs, DDAIZR, DDAILAG and DDAIDEL(2:0). For this mode, the attribute "DELAY_CNTL" must be set to "DYNAMIC". Table 10-4 shows the delay adjustment values based on the attribute/input settings.

In this mode, the PLL may come out of lock due to the abrupt change of phase. RST must be asserted to re-lock the PLL. Upon de-assertion of RST, the PLL will start the lock-in process and will take the t_{LOCK} time to complete the PLL lock.

Table 10-4. Delay Adjustment

DDAMODE = 1: Dynamic Delay Adjustment				DDAMODE = 0
DDAIZR	DDAILAG	DDAIDEL[2:0]	Delay 1 Tdly = 130 ps (nominal)	Equivalent FDEL Value
0	1	111	Lead 8 Tdly	-8
0	1	110	Lead 7 Tdly	-7
0	1	101	Lead 6 Tdly	-6
0	1	100	Lead 5 Tdly	-5
0	1	011	Lead 4 Tdly	-4
0	1	010	Lead 3 Tdly	-3
0	1	001	Lead 2 Tdly	-2
0	1	000	Lead 1 Tdly	-1
1	Don't Care	Don't Care	no delay	0
0	0	000	Lag 1 Tdly	1
0	0	001	Lag 2 Tdly	2
0	0	010	Lag 3 Tdly	3
0	0	011	Lag 4 Tdly	4
0	0	100	Lag 5 Tdly	5
0	0	101	Lag 6 Tdly	6
0	0	110	Lag 7 Tdly	7
0	0	111	Lag 8 Tdly	8

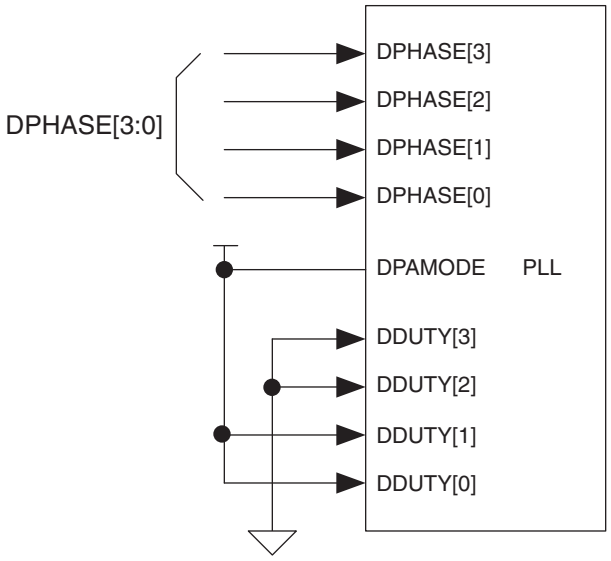
Dynamic Phase/Duty Mode

This mode sets both Dynamic Phase Adjustment and Dynamic Duty Select at the same time.

There are two modes, "Dynamic Phase and Dynamic Duty" and "Dynamic Phase and 50% Duty".

To use Dynamic Phase Adjustment with a fixed Duty Cycle, simply set the DDUTY[3:0] inputs to the desired Duty Cycle value. Figure 10-12 illustrates an example circuit. This example assumes the user-desired Duty Cycle is 3/16.

Figure 10-12. Example Dynamic Phase Adjustment Set-up with Duty Cycle Fixed to 3/16



Dynamic Phase Adjustment/Duty Cycle Select

Phase Adjustment settings are described in Table 10-5.

Table 10-5. Dynamic Phase Adjustment Settings

DPHASE[3:0]	Equivalent to PHASEADJ in Static Mode
0000	0
0001	22.5
0010	45
0011	67.5
0100	90
0101	112.5
0110	135
0111	157.5
1000	180
1001	202.5
1010	225
1011	247.5
1100	270
1101	292.5
1110	315
1111	337.5

Duty Cycle Select settings are described in Table 10-6.

Table 10-6. Dynamic Duty Cycle Select Settings

DDUTY[3:0]	Equivalent to DUTY in Static Mode (1/16 of Period)	Comment
0000	0	Not Supported
0001	1	Not Supported
0010	2	
0011	3	
0100	4	
0101	5	
0110	6	
0111	7	
1000	8	
1001	9	
1010	10	
1011	11	
1100	12	
1101	13	
1110	14	
1111	15	Not Supported

Optional External Capacitor

An optional external capacitor can be used with both the EHXPLLD and the EPLLD to change the frequency response of the on-chip loop filter. When an external capacitor is used, the frequency at the phase detector inputs (Fpd) can be as low as 2MHz, allowing the PLLs to extend the low-end of their operating ranges. Using the external capacitor will limit the high end of the PLL operating range as shown in the LatticeECP2/M Family Data Sheet. IPexpress™ checks the phase detector frequency to determine if an external capacitor is required.

The allowable ranges for the PLL parameters with and without the external capacitor are described in the LatticeECP2/M Family Data Sheet.

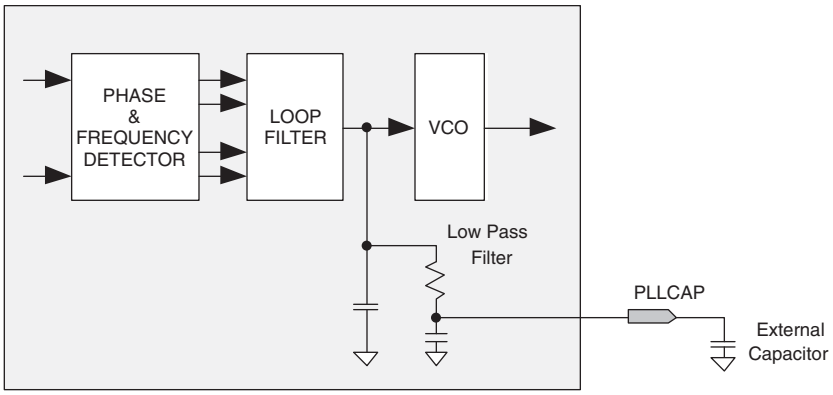
Recommended Optional External Capacitor Specifications

Value: 5.6 nF, +/- 20%
Type: Ceramic chip capacitor, NPO dielectric
Package: 1206 or smaller

Each device has two external capacitor pins, one for the left side PLLs and one for the right side PLLs. These pins are in fixed locations. They are dedicated function pins that are NOT shared with user I/Os.

When an external capacitor pin is used by a PLL on one side of the device, it cannot be used by any other PLLs on the same side of the device. This means that a maximum of two PLLs per device, one on the left side and one on the right side, can have external capacitors attached.

Figure 10-13. External Capacitor Usage

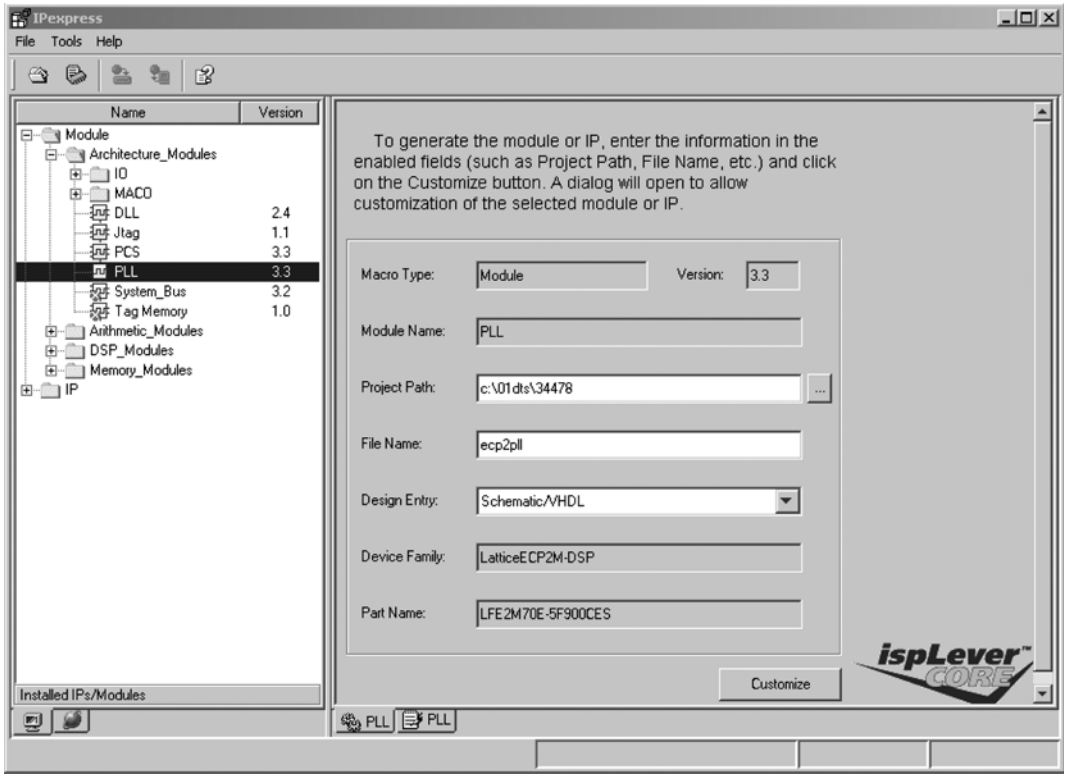


PLL Usage in IPexpress

IPexpress is used to create and configure a PLL. Designers use the graphical user interface to select parameters for the PLL. The result is an HDL model to be used in the simulation and synthesis flow.

Figure 10-14 shows the main window when PLL is selected. The only entry required in this window is the module name. Other entries are set to the project settings. These entries may be changed if desired. After entering the module name, click on **Customize** to open the **Configuration Tab** window as shown in Figure 10-15.

Figure 10-14. IPexpress Main Window (see Appendix D Figure 10-40 for Diamond Equivalent)



Configuration Tab

The Configuration Tab lists all user-accessible attributes with default values set. Upon completion, click **Generate** to generate source and constraint files. The user may choose to use the *.LPC file (or *.IPX file for Diamond projects) to load parameters.

Modes

There are two modes for configuring the PLL in the Configuration Tab: Frequency Mode and Divider Mode.

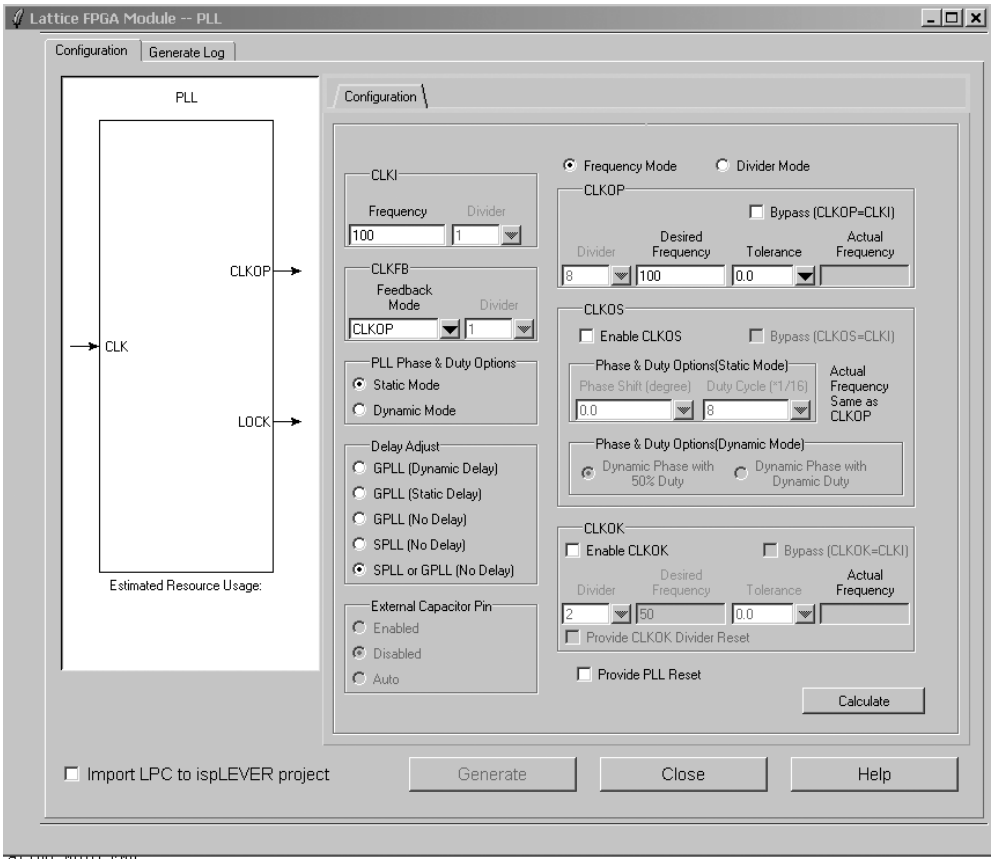
Frequency Mode

In this mode, the user enters input and output clock frequencies and the software calculates the divider settings. If the output frequency entered is not achievable, the nearest frequency will be displayed in the 'Actual' text box. After input and output frequencies are entered, clicking the **Calculate** button will display the divider values.

Divider Mode

In this mode, the user sets the divider settings with the input frequency. The user must choose the CLKOP Divider value in order to maximize the f_{VCO} and achieve optimum PLL performance. After setting the input frequency and divider settings, click **Calculate** to display the frequencies. Figure 10-15 shows the Configuration Tab.

Figure 10-15. LatticeECP2/M PLL Configuration Tab



Note: In the External Capacitor Pin, the grayed out text will automatically indicate the requirement for the external loop capacitor based upon the PLL settings. This is used to alert the user that the external loop capacitor may be required. The Auto setting indicates that the software will determine if the external loop capacitor is required after the PLL is placed into an SPLL or GPLL by the Place and Route (PAR) step. Other grayed-out sections of this dialog box turn on as their sections are enabled.

Table 10-7 describes all user parameters in the IPexpress GUI.

Table 10-7. User Parameters in the Configuration GUI

User Parameters		Description	Range	Default
Frequency Mode		Desired input/output frequency	ON/OFF	ON
Divider Mode		Desired input frequency and divider settings	ON/OFF	OFF
CLKI	Frequency	Without external capacitor	25 (33[1]) MHz to 420 MHz	100 MHz
		With external capacitor	2 MHz to 420 MHz[2]	—
	Divider	Without external capacitor	1 to 16(12[1])	1
		With external capacitor	1 to 64	—
CLKFB	Feedback Mode	Feedback Mode	Internal, CLKOP, User clock	CLKOP
	Divider	Without external capacitor	1 to 16 (12[1])	1
		With external capacitor	1 to 10	-
CLKOP	Bypass	Bypass PLL: CLKOP = CLKI	ON/OFF	OFF
	Desired Frequency	Without external capacitor	25(331) MHz to 420 MHz	100 MHz
		With external capacitor	5 MHz to 50 MHz[3]	—
	Divider	CLKOP Divider Setting (Divider Mode)	2,4,8,16,32,48, 64,80,96,112,128	8
	Tolerance	CLKOP tolerance users can tolerate	0.0, 0.1, 0.2, 0.5, 1.0, 2.0, 5.0, 10.0	0.0
	Actual Frequency	Actual frequency achievable, Read only	—	—
CLKOS	Enable	Enable CLKOS output clock	ON/OFF	OFF
	Bypass	Bypass PLL: CLKOS = CLKI	ON/OFF	OFF
	Phase - Static	CLKOS Static Phase Shift	0°, 22.5°, 45°, ... , 337.5°	—
	Duty - Static	CLKOS Static Duty Cycle Select	2 to 14	8
	Dynamic Phase with 50% Duty	Dynamic Phase and 50% Duty Cycle	ON/OFF	ON
	Dynamic Phase with Dynamic Duty	Dynamic Phase and Dynamic Duty Cycle	ON/OFF	ON
CLKOK	Enable	Enable CLKOK output clock	ON/OFF	OFF
	Bypass	Bypass PLL: CLKOK = CLKI	ON/OFF	OFF
	Desired Frequency	Without external capacitor	0.195 MHz to 210 MHz	50 MHz
		With external capacitor	0.016 MHz to 25 MHz[3]	—
	Divider	CLKOK Divider Setting (Divider Mode)	2 to 128 (all even numbers)	2
	Tolerance	CLKOK tolerance users can tolerate	0.0, 0.1, 0.2, 0.5, 1.0, 2.0, 5.0, 10.0	0.0
	Actual Frequency	Actual frequency achievable, Read only	—	—
PLL Phase & Duty Option		Dynamic/Static Mode selection	Dynamic Mode/Static Mode	Static Mode
Delay Adjust		Dynamic/Static/No delay selection	Dynamic/Static/No Delay	No Delay[4]
Provide PLL Reset		Provide PLL Reset Port	ON/OFF	OFF
Provide CLKOK Divider Reset		Provide CLKOK Reset Port	ON/OFF	OFF
Import LPC to ispLEVER project		Import .lpc file to ispLEVER project	ON/OFF	OFF

1. These values apply to SPLL. All other values apply to both GPLL and SPLL.
2. Phase Detector Input Frequency range 2 MHz to 50MHz.
3. For f_{IN} < 5MHz, f_{OUT_max} = 10 * f_{IN}.
4. IPexpress gives the user the ability to select the GPLL, SPLL, or to let the software choose, based upon the settings in the Delay Adjust section.

Frequency Calculation

Table 10-8 illustrates the Frequency limits at the Phase Detector inputs. Users must select CLKI Divider and CLKFB Divider values so that the Phase Detector Frequency falls within the range.

Let M = CLKI divider value
N = CLKFB divider value
V = CLKOP divider value

The basic equations are:

CLKOP Frequency = CLKI Frequency * N/M
f_{VCO} (VCO Frequency) = CLKOP Frequency * V
f_{PFD} (PFD Frequency) = CLKI Frequency / M = CLKFB Frequency (= CLKOP Frequency) / N

Example: If CLKI frequency is 25 MHz without external capacitor, the CLKI divider value can be only 1.

Table 10-8. Phase Detector Frequency (f_{PFD}) Range

PLL Type	External Capacitor	Frequency Range
GPLL	Without external capacitor	25 MHz to 420 MHz
	With external capacitor	2 MHz to 50 MHz[1]
SPLL	Without external capacitor	33 MHz to 420 MHz
	With external capacitor	2 MHz to 50 MHz[1]

1. For f_{IN} < 5MHz, f_{OUT_max} = 10 * f_{IN}.

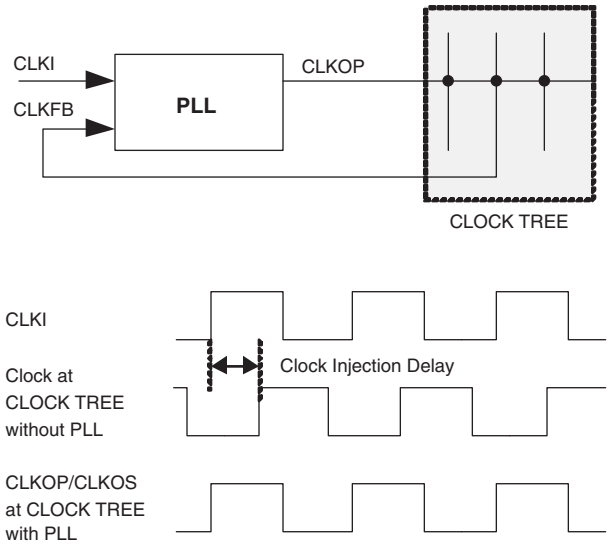
PLL Modes of Operation

PLLs have many uses within logic design. The two most popular are clock injection removal and clock phase adjustment. These two modes of operation are described below.

PLL Clock Injection Removal

In this mode, the PLLs are used to reduce clock injection delay. Clock injection delay is the delay from the input pin of the device to a destination element such as a flip-flop. The phase detector of the PLL aligns the CLKI with CLKFB. If the CLKFB signal comes from the clock tree (CLKOP), then the PLL delay and the clock tree delay is removed. Figure 10-16 Illustrates an example block diagram and waveform.

Figure 10-16. Clock Injection Delay Removal Application

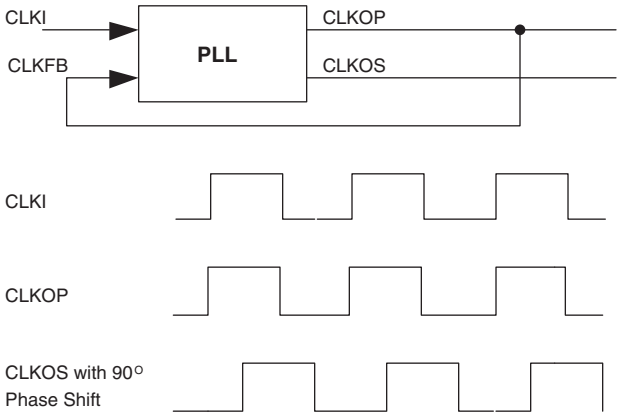


PLL Clock Phase Adjustment

In this mode, the PLLs are used to create fixed phase relationships in 22.5° increments. Creating fixed phase relationships are useful for forward clock interfaces where a specific relationship between the clock and data is required.

The fixed phase relationship can be used between CLKI and CLKOS or between CLKOP and CLKOS.

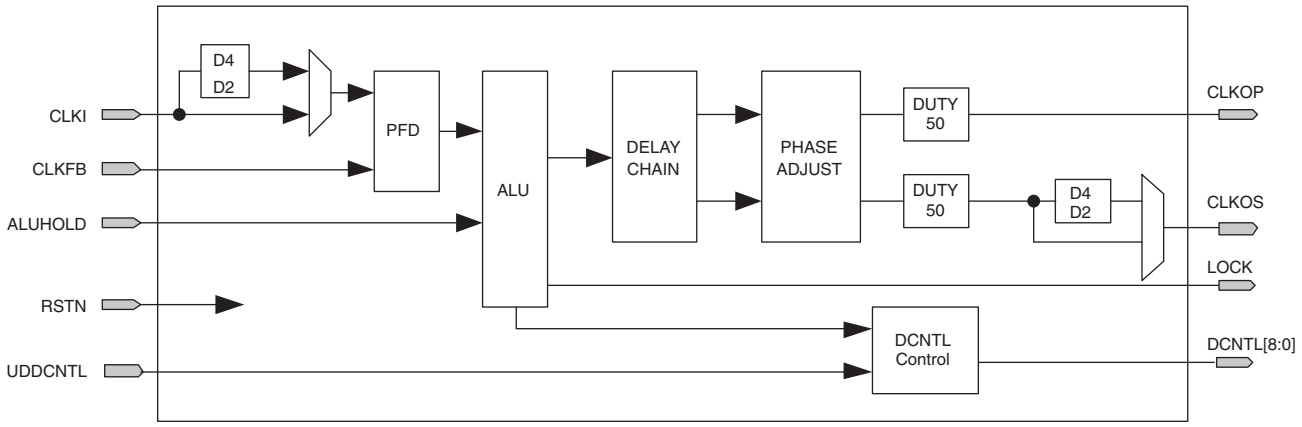
Figure 10-17. CLKOS Phase Adjustment from CLKOP



sysCLOCK DLL

The LatticeECP2/M DLL provides features such as clock injection delay removal, delay match, time reference delay (90° phase delay), and output phase adjustment. The DLL performs clock manipulation by adding delay to the CLKI input signal to create specific phase relationships. There are two types of outputs of the DLL. The first are clock signals similar to the PLL CLKOP and CLKOS. The other type of output is a delay control vector (DCNTL[8:0]). The delay control vector is connected to a Slave Delay Line (DLLDEL) element located in the I/O logic which matches the delay cells in the DLL. This delay vector allows the DLL to dynamically delay an input signal by a specific amount. Figure 10-18 provides a block diagram of the LatticeECP2/M DLL.

Figure 10-18. LatticeECP2/M DLL Block Diagram



Both clock injection delay removal and output phase adjustment use only the clock outputs of the DLL. Time reference delay and delay match modes use the delay control vector output. Specific examples of these features are discussed later in this document.

DLL Overview

The LatticeECP2/M DLL is created and configured by IPexpress. The following is a list of port names and descriptions for the DLL. There are two library elements used to implement the DLL: CIDDLLA (Clock Injection Delay), and TRDDLLA (Time Reference Delay). IPexpress will wrap one of these library elements to create a customized DLL module based on user selections.

DLL Inputs and Outputs

CLKI Input

The CLKI signal is the reference clock for the DLL. The CLKI input can be sourced from any type of FPGA routing and pin. The DLL CLKI input has a preferred pin per DLL which provides the lowest latency and best case performance.

CLKFB Input

The CLKFB input is available only if the user chooses to use a user clock signal for the feedback or in clock delay match mode. If internal feedback or CLKOS/CLKOP is used for the feedback, this connection will be made inside the module. In Clock Injection Delay Removal mode, the DLL will align the input clock phase with the feedback clock phase by delaying the input clock.

In Clock Injection Delay Match mode, the DLL will calculate the delta between the CLKI and CLKFB signals. This delay value is then output on the DCNTL vector. The DLL CLKFB input has a preferred pin per DLL which is discussed later in this document. The preferred pin provides the lowest latency and best case performance.

CLKOP Output

An output of the DLL based on the CLKI rate. The CLKOP output can drive primary and edge clock routing.

CLKOS Output

An output of the PLL based on the CLKI rate which can be divided and/or phase shifted. The CLKOS output can drive the primary and edge clock routing.

DCNTL[8:0] Output

This output of the DLL is used to delay a signal by a specific amount. The DCNTL[8:0] vector connects to a Slave Delay Line element. The DLL can then control multiple input delays from a single DLL.

UDDCNTL Input
This input is used to enable or disable updating of the DCNTL[8:0]. To ensure that the signal is captured by the synchronizer in the DLL block, it must be driven high for a time equal to at least two clock cycles when an update is required. If the signal is driven high and held in that state, the DCNTL[8:0] outputs are continuously updated.

ALUHOLD Input
This active high input stops the DLL from adding and subtracting delays to the CLKI signal. The DCNTL[8:0], CLKOP, and CLKOS outputs will still be valid, but will not change from the current delay setting.

LOCK Output
Active high lock indicator output. The LOCK output will be high when the CLKI and CLKFB signal are in phase. If the CLKI input stops the LOCK output will remain asserted. The clock is stopped so there is no clock to de-assert the LOCK output. Note that this is different from the operation of the PLL where the VCO continues to run when the input clock stops.

RSTN
Active low reset input to reset the DLL. The DLL can optionally be reset by the GSRN as well. It is recommended that if the DLL requires a reset, the reset should not be the same as the FPGA logic reset. Typically, logic requires that a clock is running during a reset condition. If the data path reset also resets the DLL, the source of the logic clock will stop and this may cause problems in the logic.

DLL Attributes

The LatticeECP2/M DLL utilizes several attributes that allow the configuration of the DLL through source constraints, IPexpress and preference files. The following section details these attributes and their usage.

DLL Lock on Divide by 2 or Divide by 4 CLKOS Output
Usually, the DLL is a 'times one' device, allowing neither frequency multiplication or division. But the LatticeECP2/M DLL allows 'divide by 2' or 'divide by 4' CLKOS outputs. Two optional 'divide by 2' and 'divide by 4' blocks are placed at the CLKI input as well as the CLKOS and this enables the use of divided CLKOS in the DLL feedback path. This allows the DLL to perform clock injection removal on a 'divide by 2' or 'divide by 4' clock, which is useful for DDRX2 and DDRX4 modes of I/O buffer operation.

When this optional clock divider is used only in the CLKOS output path, it allows the DLL to output two time-aligned clocks at different frequencies. When the divider is set to divide by 2 or divide by 4, a 'dummy' delay is inserted in the CLKOP output path to match the clock to Q delay of the CLKOS divider.

Optional Clock Fine Phase Shift
The optional fine phase shift in the CLKOS output path is built from a delay block that matches the other four blocks in the main delay chain. This delay block allows the CLKOS output to phase shifted a further 45, 22.5 or 11.25 degrees relative to its normal position.

GSR
The DLL can be reset by the GSR, if enabled. The GSR keyword can be set to ENABLED/DISABLED. This option is provided in the IPexpress GUI. Below is an example of the use of this preference.

```
ASIC "dll/dll_0_0" TYPE "CIDDLLA" GSR=DISABLED;
```

DLL Lock Time Control
The DLL will lock when the CLKI and CLKFB phases are aligned. In a simulation environment, the lock time is fixed to 100µs (default). This value can be changed through an HDL parameter or preference (for the back annotation simulation). The DLL contains a parameter named LOCK_DELAY which accepts an integer value for the total time in µs until the lock output goes high. Below is an example of how to set this value for front-end simulation.

Verilog:

```
defparam mydll.mypll_0_0.LOCK_DELAY=500;
mydll dll_inst(.CLKI(clkin), .CLKOP(clk1), .CLKOS(clk2),
```

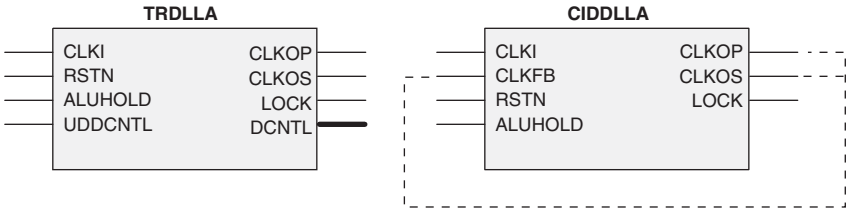
VHDL:

Not supported. For back annotation simulation LOCK_DELAY needs to be set in the preference file. Below is an example for the PLL.

```
ASIC "pll/pll_0_0" TYPE "EHXPLLA" LOCK_DELAY=200;
```

DLL Library Symbols

Figure 10-19. DLL Library Symbols



DLL Library Definitions

The Lattice library contains library elements to allow designers to utilize the DLL. These library elements use the DLL attributes defined in the "DLL Attributes" section.

The two modes of operation are presented as library elements as listed below.

Table 10-9. DLL Library Elements

Library Element Name	Mode of Operation	Description
TRDLLA	Time Reference Delay DLL	This mode generates four phases of the clock, 0°, 90°, 180°, 270°, along with the control setting used to generate these phases.
CIDDLLA	Clock Injection Delay DLL (Four Delay Cell Mode)	This mode removes the clock tree delay, aligning the external feedback clock to the reference clock. It has a single output coming from the fourth delay block.

DLL Library Element I/Os

Table 10-10. DLL Library Element I/O Descriptions

Signal	I/O	Description
CLKI	I	Clock input pin from dedicated clock input pin, other I/O or logic block.
CLKFB	I	Clock feedback input pin from dedicated feedback input pin, internal feedback, other I/O or logic block. This signal is not user selectable.
RSTN	I	Active low synchronous reset. From dedicated pin or internal node.
ALUHOLD	I	"1" freezes the ALU. For TRDLLA and CIDDLLA.
UDDCNTL	I	Active high synchronous enable signal from CIB for updating digital control to PIC delay. It must be driven high at least two clock cycles.
DCNTL[8:0]	O	Digital delay control code signals.
CLKOP	O	The primary clock output for all possible modes.
CLKOS	O	The secondary clock output with finer phase shift and/or division by 2 or by 4.
LOCK	O	Active high phase lock indicator. Lock means the reference and feedback clocks are in phase.

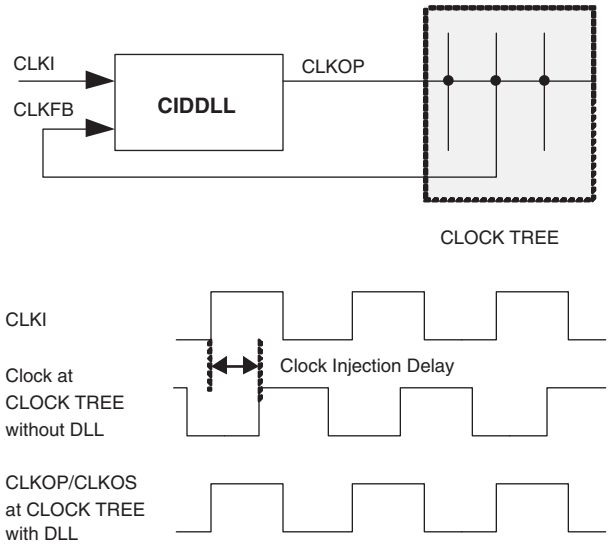
Note: Refer to device data sheet for frequency specifications.

DLL Modes of Operation

Clock Injection Removal Mode (CIDDLLA)

The DLL can be used to reduce clock injection delay (CIDDLLA). Clock injection delay is the delay from the input pin of the device to a destination element such as a flip-flop. The DLL will add delay to the CLKI input to align CLKI to CLKFB. If the CLKFB signal comes from the clock tree (CLKOP, CLKOS) then the delay of the DLL and the clock tree will be removed from the overall clock path. Figure 10-20 shows a circuit example and waveform.

Figure 10-20. Clock Injection Delay Removal via DLL



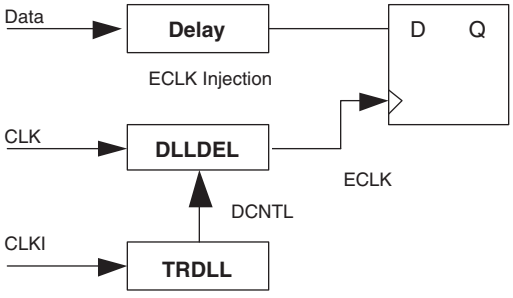
Clock injection removal mode can also provide a DCNTL port. In this mode, the delay added to the CLKI signal is output on the DCNTL port so that other input signals can be delayed by the same amount. This is very useful if several clocks are used in the same circuit to minimize the number of DLLs required. When using the DCNTL, the DLL delay will be limited to the range of the DCNTL vector. Therefore, IPexpress will restrict the CLKI rate from 300MHz to 700MHz.

Time Reference Delay Mode (TRDLLA: 90-Degree Phase Delay)

The Time Reference Delay (TRDDLLA) mode of the DLL is used to calculate 90 degrees of delay to be placed on the DCNTL vector. This is a useful mode in delaying a clock 90 degrees for use in clocking a DDR type interface.

Figure 10-21 provides a circuit example of this mode.

Figure 10-21. Time Reference Delay Circuit Example



In this mode, CLKI accepts a clock input. The DLL produces a DCNTL vector that will delay an input signal by 90 degrees of a full period of the CLKI signal. This DCNTL vector can then be connected to a Slave Delay Line (DLLDELA) to delay the signal by 90 degrees of the full period of CLKI.

DLL Usage in IPexpress

IPexpress is used to create and configure a DLL. The IPexpress graphical user interface allows users to select parameters for the DLL. The result is an HDL model to be used in the simulation and synthesis flow.

Configuration Tab

- **Usage Mode** – Select the mode of the DLL (Time Reference Delay [TRDLLA] or Clock Injection Delay Removal [CIDDLLA]). This selection will enable or disable further options in the GUI.
- **CLKI Frequency** – The rate of the CLKI input in MHz.
- **CLKOS Divider** – Set the divider for the CLKOS output to be either no divide, divide by 2, or divide by 4.
- **CLKOS Phase Shift** – Set the phase offset of the CLKOS to the CLKOP output. CLKOP will lead CLKOS by the amount of phase shift selected. The phase increment is 11.25 degrees. The pull-down list numbers are abbreviated to a decimal point.
- **CLKFB Feedback Mode** – Sets the feedback mode of the DLL to either CLKOP, CLKOS or User Clock. If CLKOP/CLKOS is selected, the clock tree injection delay for the specific output clock will be removed. If User Clock is selected, the user will be provided with the CLKFB port on the DLL.
- **CLKFB Frequency** – This is used only with User Clock Feedback Mode.
- **Provide RSTN Port** – The RSTN port allows the user to reset the DLL through a user signal.

PLL/DLL Cascading

It is possible to connect several arrangements of PLLs and DLLs. There are three possible cascading schemes:

- PLL to PLL
- PLL to DLL
- DLL to DLL

It is not possible to connect the DLL to a PLL. The DLL produces abrupt changes on its output clocks when changing delay settings. The PLL sees this as radical phase changes that prevent the PLL from locking correctly.

IPexpress Output

There are two outputs of IPexpress that are important for use in the design. The first is the <module_name>.[v|vhd] file. This is the user-named module that was generated by the tool to be used in both synthesis and simulation flows. The second file is a template file <module_name>_tmpl.[v|vhd]. This file contains a sample instantiation of the module. This file is only provided for the user to copy/paste the instance and is not intended to be used in the synthesis or simulation flows directly.

For the PLL/DLL, IPexpress sets attributes in the HDL module created that are specific to the data rate selected. Although these attributes can easily be changed, they should only be modified by re-running the GUI so that the performance of the PLL/DLL is maintained. After the map stage in the design flow, FREQUENCY preferences will be included in the preference file to automatically constrain the clocks produced from the PLL/DLL.

Clock Dividers (CLKDIV)

The clock divider divides the high-speed clock by 2, 4 and 8. The divided output can then be used as a primary clock or secondary clock input. The clock dividers are used for providing the low-speed FPGA clocks for shift registers (x2, x4, x8) and DDR/SPI4 I/O logic interfaces. There are two clock dividers, one on each side.

CLKDIV Library Element Definition

Users can instantiate CLKDIV in the source code as defined in this section. Figure 10-22, Table 10-11 and Table 10-12 describe the definition of CLKDIVB.

Figure 10-22. CLKDIV Library Symbol

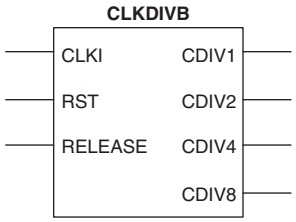


Table 10-11. CLKDIVB Port Definitions

Name	Description
CLKI	Clock input
RST[1]	Reset input, asynchronously forces all outputs low.
RELEASE[1]	Releases outputs synchronously to input clock.
CDIV1	Divided by 1 output
CDIV2	Divided by 2 output
CDIV4	Divided by 4 output
CDIV8	Divided by 8 output

1. Note: Unused RST must be tied to ground. Unused RELEASE must be tied to V_{CC}.

Table 10-12. CLKDIVB Attribute Definition

Name	Description	Value	Default
GSR	GSR Enable	ENABLED/DISABLED	DISABLED

CLKDIV Declaration in VHDL Source Code

```
COMPONENT CLKDIVB
-- synthesis translate_off
         GENERIC (
         GSR             : in String);
-- synthesis translate_on
         PORT (
         CLKI,RST, RELEASE:IN    std_logic;
         CDIV1, CDIV2, CDIV4, CDIV8:OUT   std_logic);
END COMPONENT;

     attribute GSR : string;
     attribute GSR of CLKDIVinst0 : label is “DISABLED”;

begin

CLKDIVinst0:CLKDIVB
-- synthesis translate_off
         GENERIC MAP(
         GSR             => “disabled”
         );
-- synthesis translate_on
         PORT MAP(
         CLKI            => CLKIsig,
         RST             => RSTsig,
         RELEASE         => RELEASEsig,
```

```
        CDIV1           => CDIV1sig,
        CDIV2           => CDIV2sig,
        CDIV4           => CDIV4sig,
        CDIV8           => CDIV8sig
        );

        end
```

CLKDIV Usage with Verilog - Example

```
module clkdiv_top(RST,CLKI,RELEASE,CDIV1,CDIV2,CDIV4,CDIV8);

input CLKI,RST,RELEASE;
output CDIV1,CDIV2,CDIV4,CDIV8;

CLKDIVB CLKDIBinst0 (.RST(RST),.CLKI(CLKI),.RELEASE(RELEASE),
             .CDIV1(CDIV1),.CDIV2(CDIV2),.CDIV4(CDIV4),.CDIV8(CDIV8));

defparam CLKDIBint0.GSR = "DISABLED";

endmodule
```

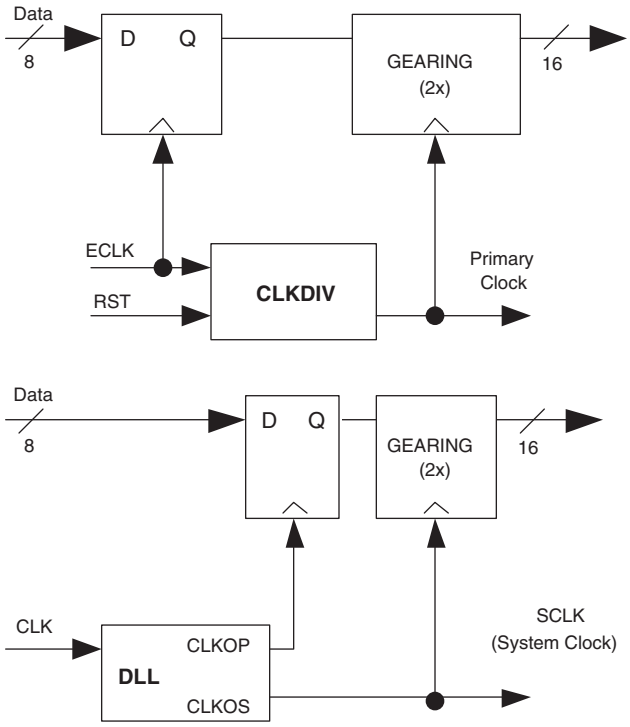
CLKDIV Example Circuits

The clock divider (CLKDIV) can divide a clock by 2 or 4 and drives a primary clock network. The clock dividers are useful for providing the low-speed FPGA clocks for I/O shift registers (x2, x4) and DDR (x2, x4) I/O logic interfaces. Divide by 8 is provided for slow speed/low power operation.

To guarantee a synchronous transfer in the I/O logic, the CLKDIV input clock must come from an edge clock and the output drives a primary clock. In this case, they are phase matched. This is especially useful for synchronously resetting the I/O logic when Mux/DeMux gearing is used in order to synchronize the entire data bus as shown in Figure 10-23. Using the low skew characteristics of the edge clock routing, a reset can be provided to all bits of the data bus to synchronize the Mux/DeMux gearing.

The second circuit shows that a DLL can replace CLKDIV for x2 and x4 applications.

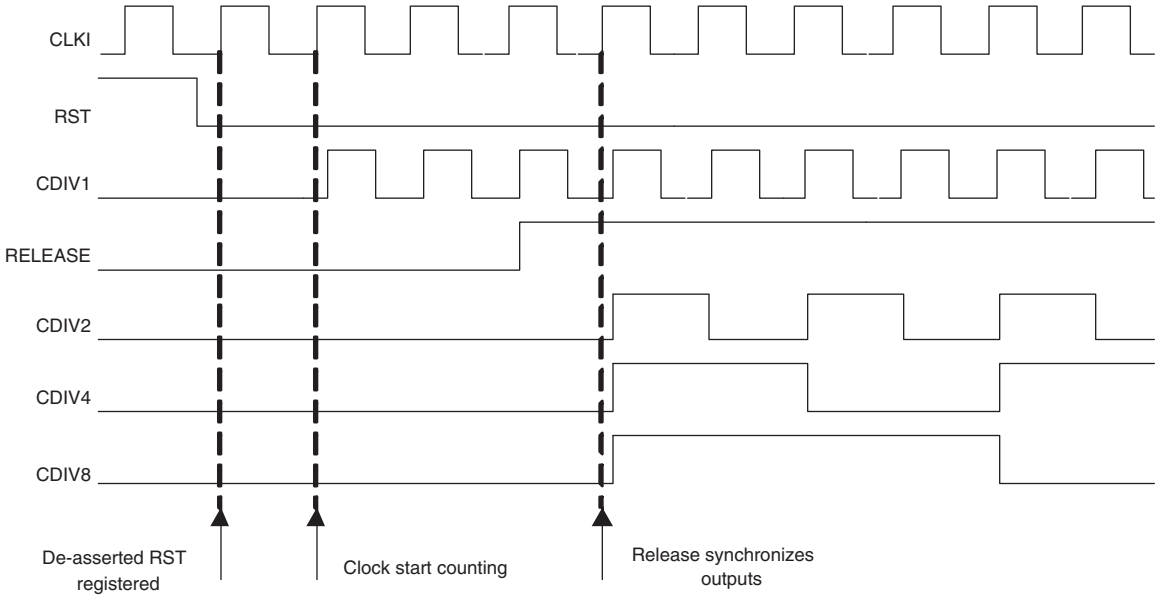
Figure 10-23. CLKDIV Application Examples



Release Behavior

The port “Release” is used to synchronize all outputs after RST is de-asserted. Figure 10-24 illustrates the Release behavior.

Figure 10-24. CLKDIV Release Behavior



DLLDEL (Slave Delay Line)

The Slave Delay line is designed to generate the desired delay in DDR/SPI4 applications. The delay control inputs (DCNTL[8:0]) are fed from the general purpose DLL outputs. The library element definitions are described in Figure 10-25 and Table 10-13.

Figure 10-25. DLLDELA Library Symbol

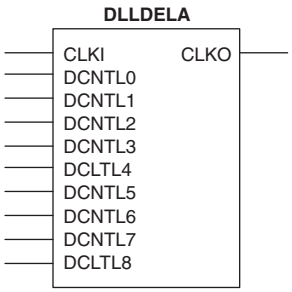


Table 10-13. DLLDELA I/O

Name	I/O	Description
CLKI	I	Clock Input
DCNTL[8:0]	I	Delay Control Bits
CLKO	O	Clock Output

DLLDELA Declaration in VHDL Source Code

```
COMPONENT DLLDELA
      PORT     (
               CLKI :IN std_logic;
               DCNTL0 :IN std_logic;
               DCNTL1 :IN std_logic;
               DCNTL2 :IN std_logic;
               DCNTL3 :IN std_logic;
               DCNTL4 :IN std_logic;
               DCNTL5 :IN std_logic;
               DCNTL6 :IN std_logic;
               DCNTL7 :IN std_logic;
               DCNTL8 :IN std_logic;
               CLKO :OUT std_logic
               );
END COMPONENT;

begin
DLLDELAinst0: DLLDELA1
      PORT MAP (
               CLKI => clkisig,
               DCNTL0 => dcntl0sig,
               DCNTL1 => dcntl1sig,
               DCNTL2 => dcntl2sig,
               DCNTL3 => dcntl3sig,
               DCNTL4 => dcntl4sig,
               DCNTL5 => dcntl5sig,
               DCNTL6 => dcntl6sig,
               DCNTL7 => dcntl7sig,
               DCNTL8 => dcntl8sig,
```

```
                CLKO => clkosig
                );

   end
```

DLLDELA Usage with TRDLLA - Verilog - Example

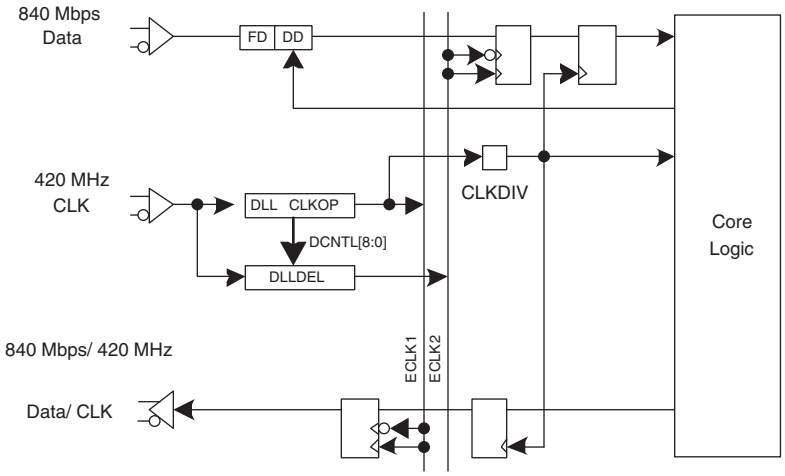
Note: DLL0(TRDLLA) must be generated by IPexpress as a sub-module

```
module ddldel_top (rst,d,clkin,clkin2,clkout,aluhold,uddcntl,q);

input rst,d,clkin,clkin2,aluhold,uddcntl;
output clkout,q;

wire [8:0]DCntl_int;
reg qint;

DLL0   dllinst0   (.clk(clkin2),   .aluhold(aluhold),   .uddcntl(uddcntl),   .clkop(),
.clkos(),
          .dcntl(DCntl_int),.lock());
DLLDELA delinst0 (.CLKI(clkin),.DCNTL0(DCntl_int[0]),.DCNTL1(DCntl_int[1]),
            .DCNTL2(DCntl_int[2]), .DCNTL3(DCntl_int[3]), .DCNTL4(DCntl_int[4]),
            .DCNTL5(DCntl_int[5]), .DCNTL6(DCntl_int[6]), .DCNTL7(DCntl_int[7]),
            .DCNTL8(DCntl_int[8]), .CLKO(clk90)); //synthesis syn_black_box

assign clkout = clk90;
assign q = qint;

always@(posedge clk90 or negedge rst)
      if (~rst)
        qint =1'b0;
      else
        qint = d;

endmodule
```

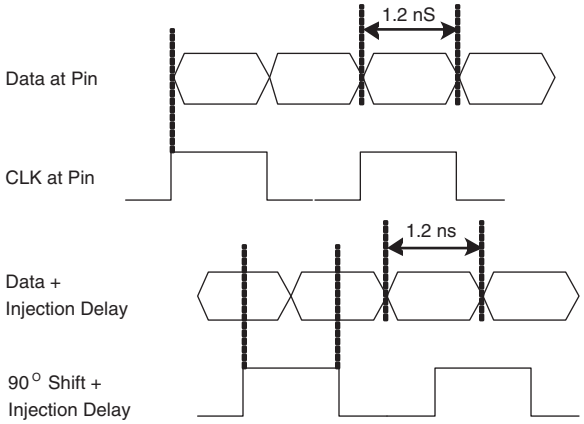
DLLDELA Application Example

Figure 10-26 shows an example DLLDEL application. As shown in the timing diagram, DLLDEL shifts the clock by 90 degrees to center both edges in the middle of data window.

Figure 10-26. SPI4.2 and DDR Registers Interface Application



FD: Fixed Delay
DD: Dynamic Delay
Users can select the delay setting in IPexpress.



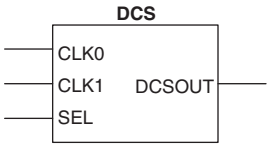
DQSDLL and DQSDEL

There is another combination of DLL and Slave Delay Line, DQSDLL and DQSDEL, in the LatticeECP2/M device family. This pair is similar in design and function to DLL and DLLDEL, but usage is limited to DDR implementation. For additional information, see TN1102, LatticeECP2/M sysIO Usage Guide.

DCS (Dynamic Clock Select)

DCS is a global clock buffer incorporating a smart multiplexer function that takes two independent input clock sources and avoids glitches or runt pulses on the output clock, regardless of where the enable signal is toggled. There are two DCSs for each quadrant. The outputs of the DCS then reach primary clock distribution via the feed-lines. Figure 10-27 shows the block diagram of the DCS.

Figure 10-27. DCS Library Symbol



DCS Library Element Definition

Table 10-14 defines the I/O ports of the DCS block. There are eight modes available. Table 10-15 describes how each mode is configured.

Table 10-14. DCS I/O Definition

I/O	Name	Description
Input	SEL	Input Clock Select
	CLK0	Clock input 0
	CLK1	Clock Input 1
Output	DCSOUT	Clock Output

Table 10-15. DCS Modes of Operations

Attribute Name	Description	Output		Value
		SEL = 0	SEL = 1	
DCS MODE	Rising edge triggered, latched state is high	CLK0	CLK1	POS
	Falling edge triggered, latched state is low	CLK0	CLK1	NEG
	Sel is active high, disabled output is low	0	CLK1	HIGH_LOW
	Sel is active high, disabled output is high	1	CLK1	HIGH_HIGH
	Sel is active low, disabled output is low	CLK0	0	LOW_LOW
	Sel is active low, disabled output is high	CLK0	1	LOW_HIGH
	Buffer for CLK0	CLK0	CLK0	CLK0
	Buffer for CLK1	CLK1	CLK1	CLK1

DCS Timing Diagrams

Each mode performs its unique operation. Clock output timing is determined by input clocks and the edge of the SEL signal. Figure 10-28 describes the timing of each mode.

Figure 10-28. Timing Diagrams by DCS MODE

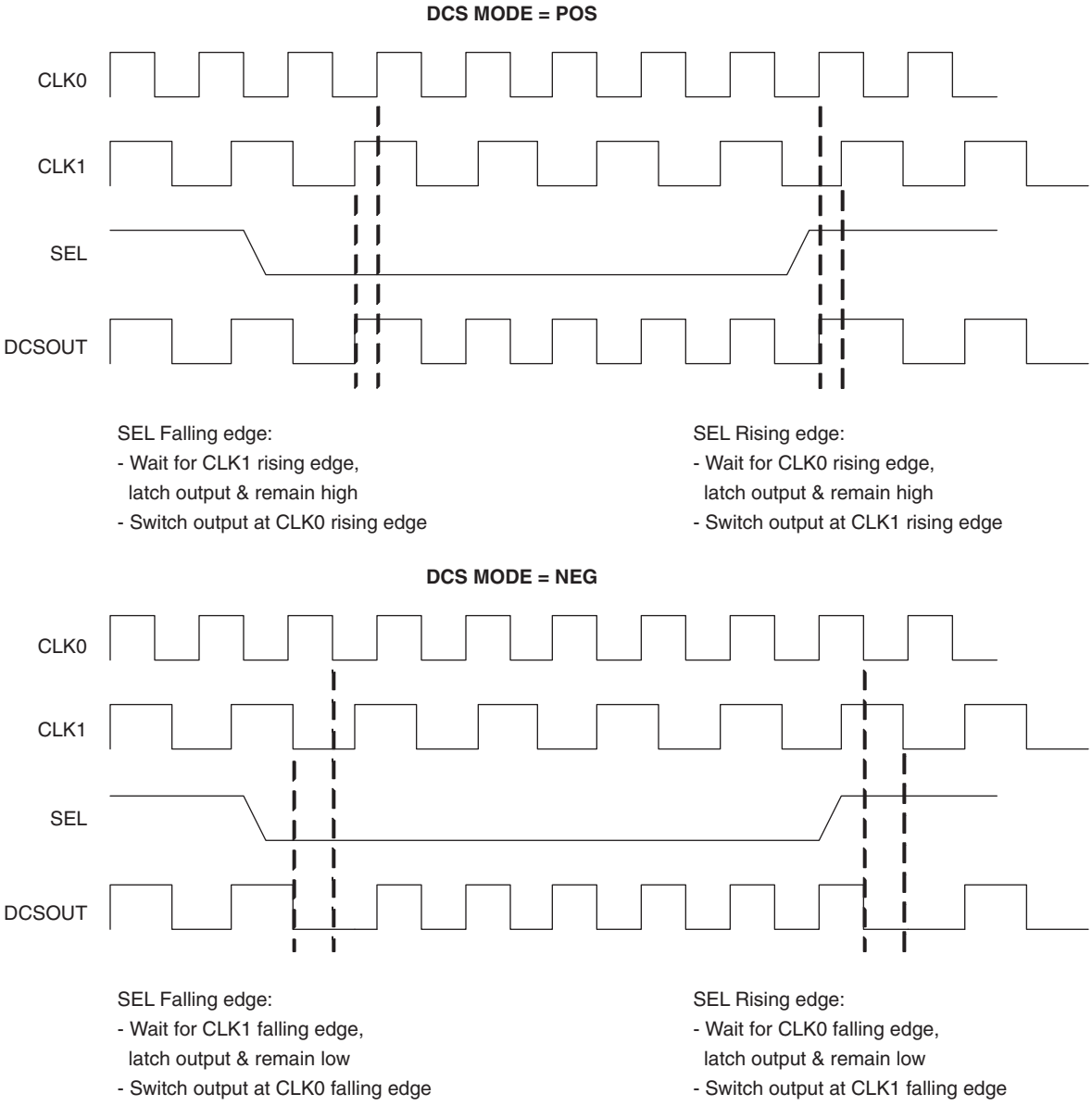
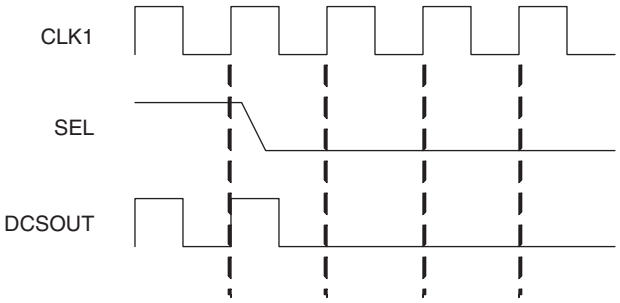


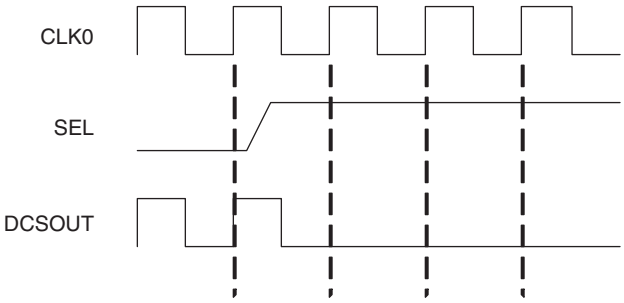
Figure 10-28. Timing Diagrams by DCS MODE (Cont.)

DCS MODE = HIGH_LOW



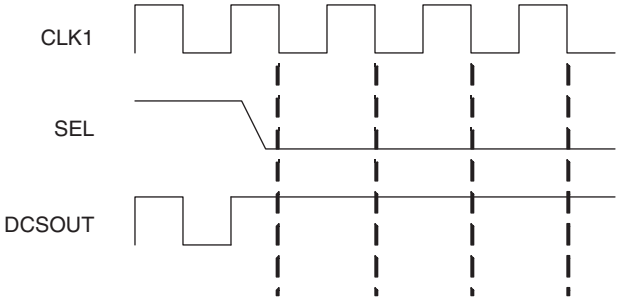
- Switch low @CLK1 falling edge.
- If SEL is low, output stays low at on CLK1 rising edge. SEL must not change during setup prior to rising clock.

DCS MODE = LOW_LOW



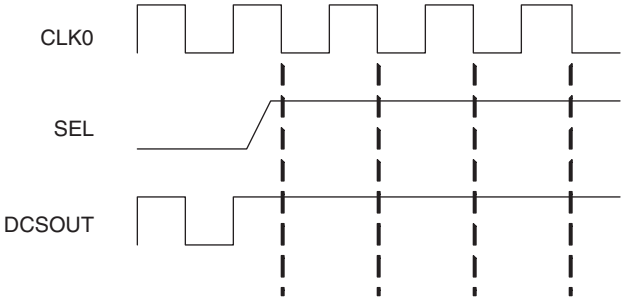
- Switch low @CLK0 falling edge.
- If SEL is high, output stays low at on CLK0 rising edge.

DCS MODE = HIGH_HIGH



- Switch high @CLK1 rising edge.
- If SEL is low, output stays low high on CLK1 falling edge.

DCS MODE = LOW_HIGH



- Switch high @ CLK0 rising edge.
- If SEL is high, output stays high on CLK0 falling edge.

DCS Usage with VHDL - Example

```
library ecp2m;
use ecp2m.components.all;
library IEEE;
use IEEE.std_logic_1164.all;

entity DCStest is port
( clksel  : in  std_logic;
  dcsclk0 : in  std_logic;
  sysclk1 : in  std_logic;
  dcsclk  : out std_logic
);
end DCStest;
architecture DCStest_arch of DCStest is
  COMPONENT DCS
  -- synthesis translate_off
  GENERIC
  ( DCSMODE : string := "POS"
  );
  -- synthesis translate_on
  PORT
  ( CLK0   :IN  std_logic;
    CLK1   :IN  std_logic;
    SEL    :IN  std_logic;
    DCSOUT :OUT std_logic
  );
  END COMPONENT;

  attribute DCSMODE             : string;
  attribute DCSMODE of DCSinst0 : label is "POS";

begin
  DCSInst0: DCS
  -- synthesis translate_off
  GENERIC MAP
  ( DCSMODE => "POS"
  )
  -- synthesis translate_on
  PORT MAP
  ( SEL    => clksel,
    CLK0   => dcsclk0,
    CLK1   => sysclk1,
    DCSOUT => dcsclk
  );
end DCStest_arch;
```

DCS Usage with Verilog - Example

```
module dcs(clk0,clk1,sel,dcsout);

input clk0, clk1, sel;
output dcsout;

DCS DCSInst0 (.SEL(sel),.CLK0(clk0),.CLK1(clk1),.DCSOUT(dcsout));
defparam DCSInst0.DCSMODE = "CLK0";

endmodule
```

Oscillator (OSCD)

There is a dedicated oscillator in the LatticeECP2/M devices whose output is made available for users. The oscillator frequency is programmable with a range of 2.5 to 130MHz. The output of the oscillator can also be routed as an input clock to the clock tree. The oscillator frequency output can be further divided by internal logic (user logic) for lower frequencies, if desired. The oscillator is powered down when not in use. The output of this oscillator is not a precision clock. It is intended for use as an extra clock that does not require accurate clocking.

Library Element Name: OSCD

Table 10-16. OSCD Port Definition

I/O	Name	Description
Output	OSC	Oscillator Clock Output

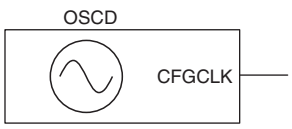
Table 10-17. OSCD Attribute Definition

User Attribute	Attribute Name	Value (MHz)	Default Value
Nominal Frequency	NOM_FREQ	2.5, 4.3, 5.4, 6.9, 8.1, 9.2, 10.0, 13.0, 15.0, 20.0, 26.0, 30.0, 34.0, 41.0, 45.0, 55.0, 60.0, 130.0	2.5

Please refer to the LatticeECP2/M Family Data Sheet for detailed specifications.

OSC Library Symbol (OSCD)

Figure 10-29. OCS Symbol



OSC Usage with VHDL - Example

```
COMPONENT OSCD
-- synthesis translate_off
      GENERIC(NOM_FREQ: string);
-- synthesis translate_on
PORT (CFGCLK:OUT   std_logic);

END COMPONENT;

      attribute NOM_FREQ : string;
      attribute NOM_FREQ of OSCins0 : label is “2.5”;
begin

OSCInst0: OSCD
-- synthesis translate_off
      GENERIC MAP (NOM_FREQ => “2.5”)
-- synthesis translate_on
      PORT MAP ( CFGCLK=>  osc_int);

end
```

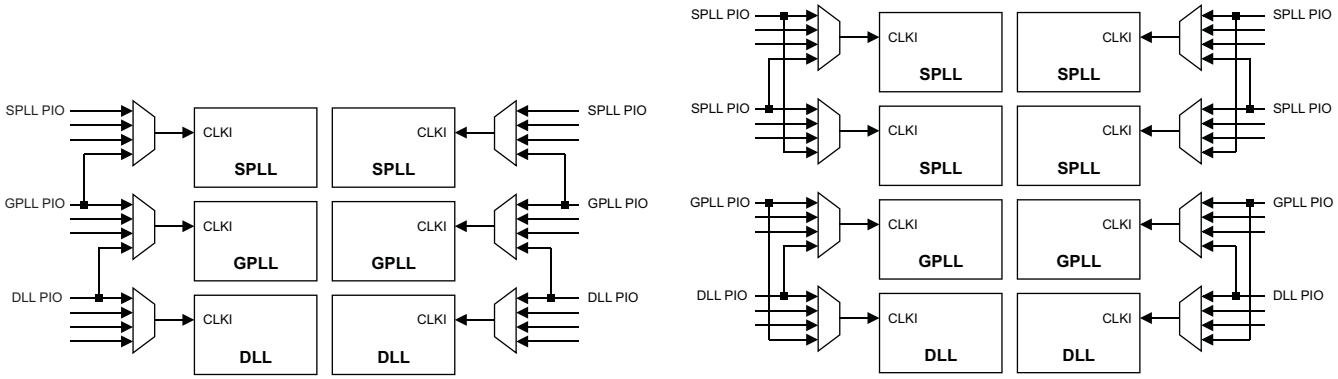
OSC Usage with Verilog - Example

```
module OSC_TOP(OSC_CLK);

output OSC_CLK;

OSCD OSCinst0 (.CFGCLK(OSC_CLK));

defparam OSCinst0.NOM_FREQ = "2.5";

endmodule
```

Input Clock Sharing

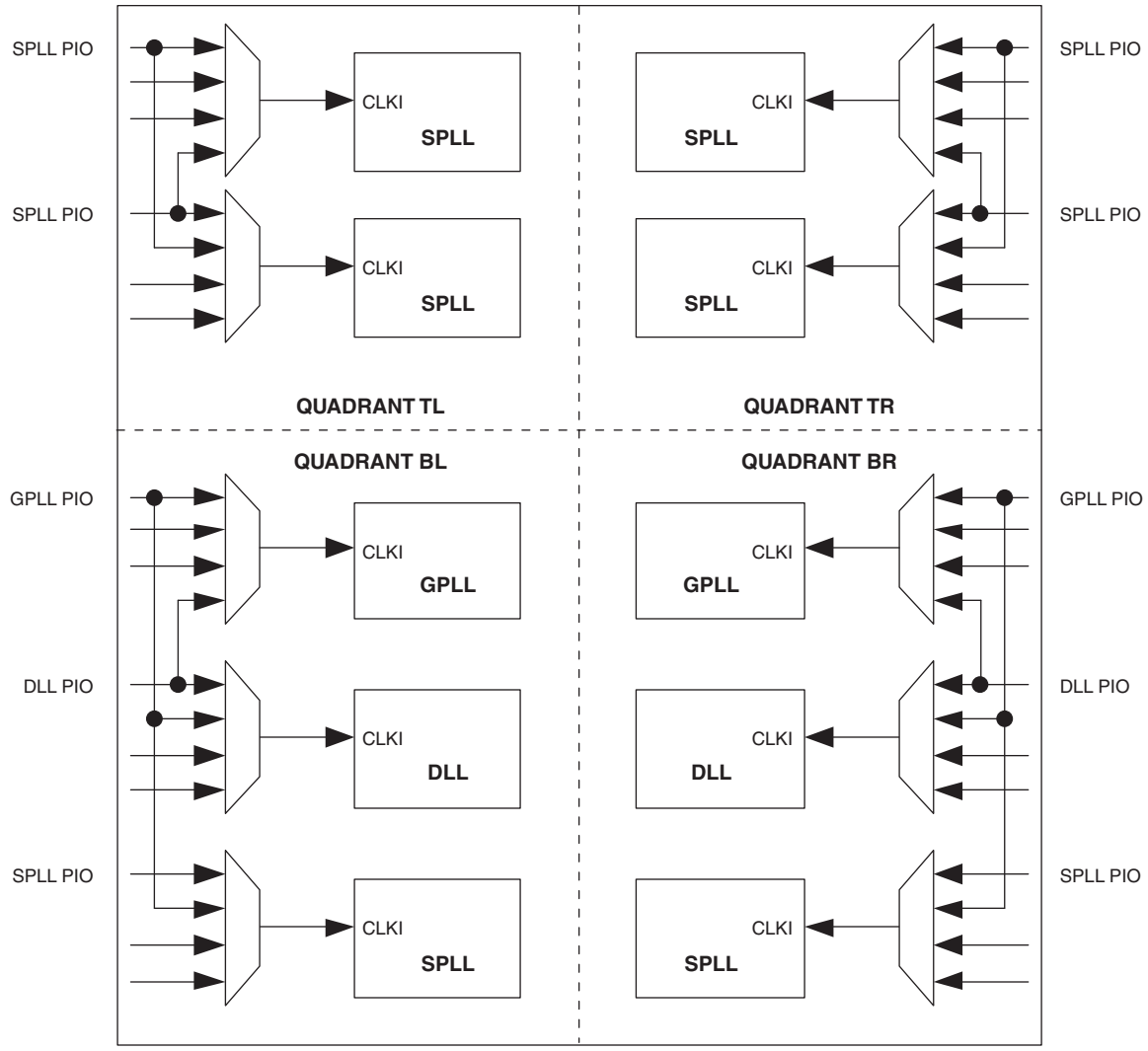
The reference clock from the pads can be shared in LatticeECP2/M PLLS and DLLS as shown in Figures 10-30 and 10-31. This feature is useful when only one clock source is available for multiple PLLs/DLLs.

Figure 10-30. Input Clock Sharing (LatticeECP2-50 and LatticeECP2-70)



LatticeECP2-50 **LatticeECP2-70**

Figure 10-31. Input Clock Sharing (LatticeECP2M with Six SPLLs)



Setting Clock Preferences

Clock preferences allow designers to implement clocks to the desired performance. Preferences can be set in the ispLEVER Design Planner Spreadsheet View (or **Tools > Spreadsheet View** in Diamond) or in preference files. Frequently used preferences are described in Appendix C. For additional information see the ispLEVER or Diamond on-line Help system.

Power Supplies

Each PLL has its own power supply. On the LatticeECP2-6, LatticeECP2-12, and LatticeECP2-20 devices the PLL power supply has been combined with the package VCC for better performance. There will only be VCC pins on these devices.

On the larger LatticeECP2 and all LatticeECP2M devices, the PLL power supply has its own power supply pins, VCCPLL. Since VCC and VCCPLL are normally the same 1.2V, it is recommended that they are driven from the same power supply on the circuit board, thus minimizing leakage. In addition, each of these supplies should be independently isolated from the main 1.2V supply on the board using proper board filtering techniques to minimize the noise coupling between them.

The DLL is powered from the FPGA core power supply.

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
February 2006	01.0	Initial release.
April 2006	01.1	Removed unsupported devices, removed DLL SMI phrases, rephrased DDUTY support due to software incomplete.
September 2006	01.2	Added detailed clock network descriptions.
		Added IPexpress GUI quick reference table.
		Added LatticeECP2M information throughout.
		OSC divider value range updated.
January 2007	01.3	Updated Frequency range.
		Described CLKOP and CLKOK synchronous timing relationship with respective reset signals.
January 2007	01.4	Updated IPexpress Main Window screen shot.
		Updated LatticeECP2/M Configuration Tab screen shot.
		Updated User Parameters in the Configuration GUI table.
June 2007	01.5	Corrected reference to EXHPLLD in Optional External Capacitor section (changed to EHXPLLD).
		Updated GSR section of Attributes.
August 2008	01.6	Updated the LatticeECP2/M PLL Configuration tab screen shot.
		Updated the Port names for the LatticeECP2/M PLL Library symbols.
		Added LatticeECP2/M PLL Modules section.
		Corrected the External Capacitor section.
		Removed the DLL operation mode CIMDLLA since it is not available on LatticeECP2/M.
		Updated Power Supplies section.
March 2009	01.7	Updated “DCS Usage with VHDL - Example” code.
October 2009	01.8	Updated Input Clock Sharing (LatticeECP2-50 and LatticeECP2-70) figure.
February 2010	01.9	Reconciled LOCK description among MachXO, LatticeXP2, LatticeECP2/M and LatticeECP3.
May 2010	02.0	Specified dedicated clock pins in the Secondary Clocks text section.
June 2010	02.1	Updated for Lattice Diamond design software support.

Appendix A. Primary Clock Sources and Distribution

Figure 10-32. LatticeECP2 Primary Clock Sources and Distribution

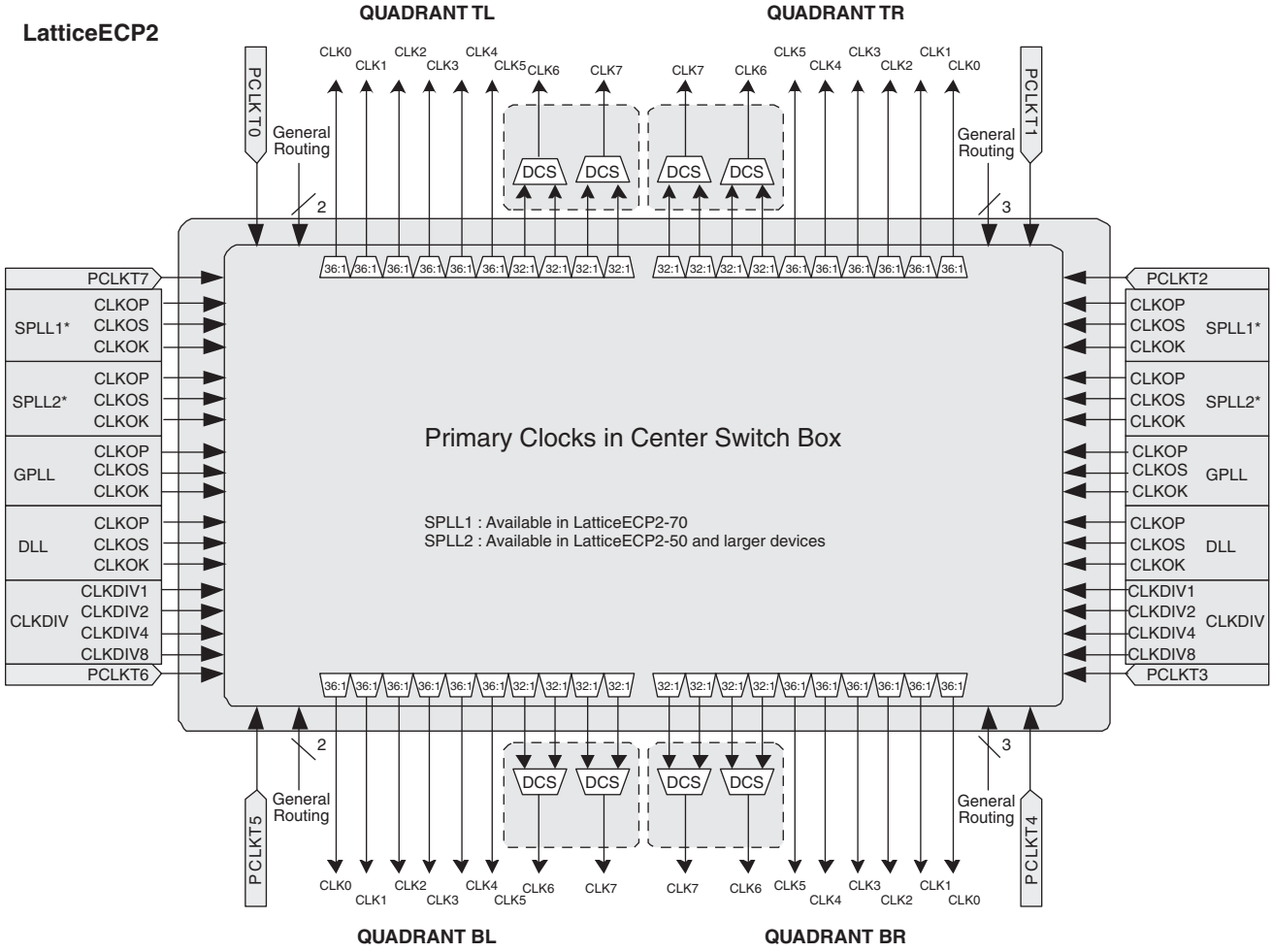
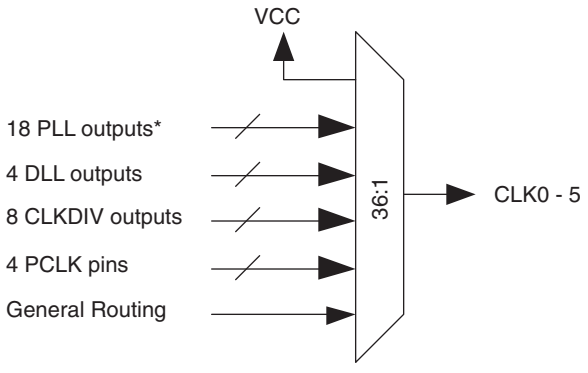
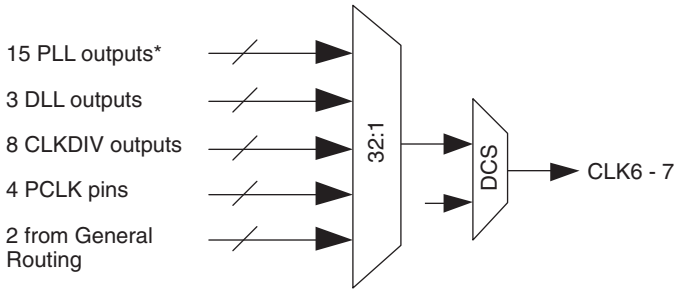


Figure 10-33. LatticeECP2 Primary Clock Muxes



*LatticeECP2-50 has twelve PLL outputs



*LatticeECP2-50 has ten PLL outputs

Figure 10-34. LatticeECP2M Primary Clock Sources and Distribution

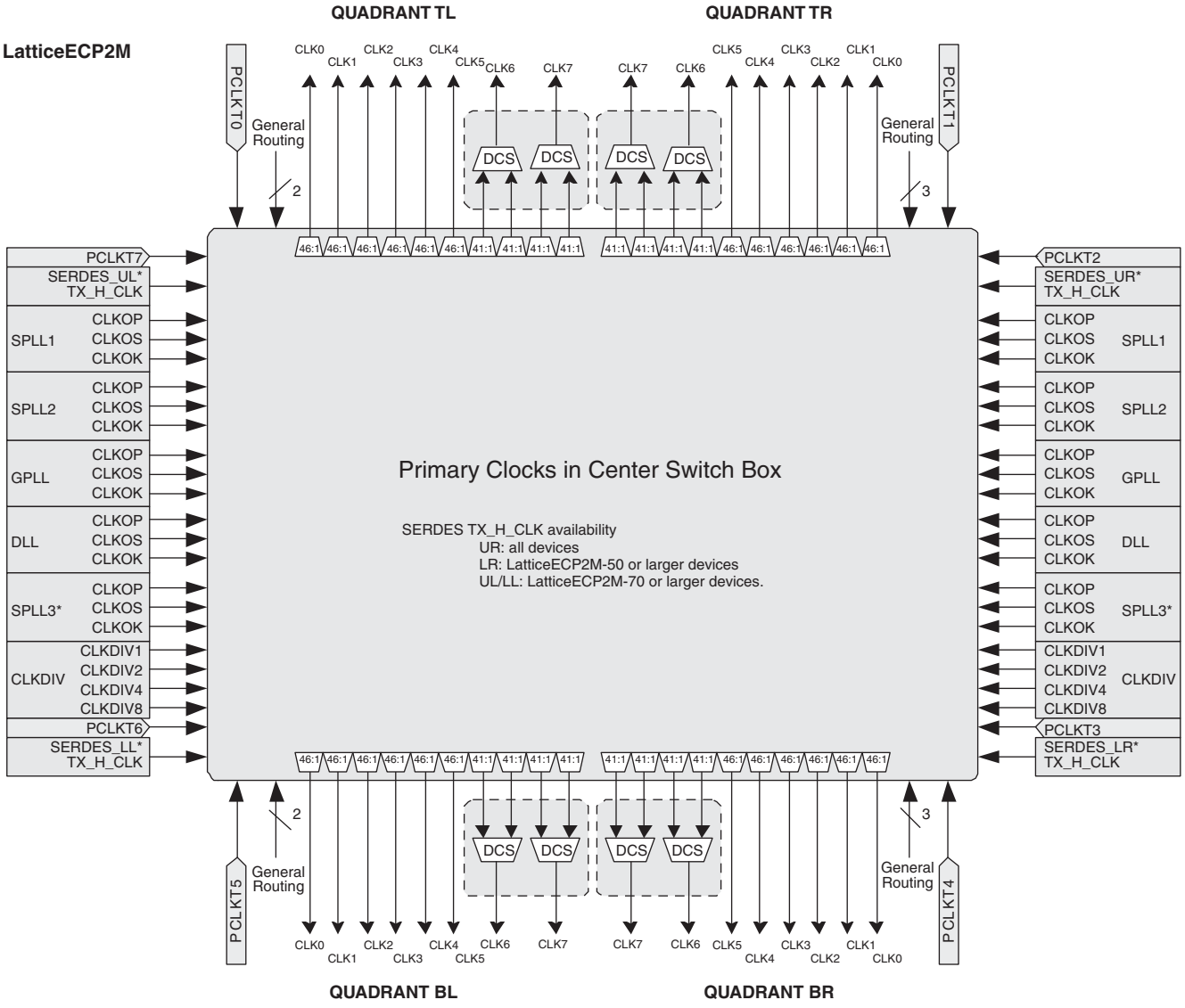
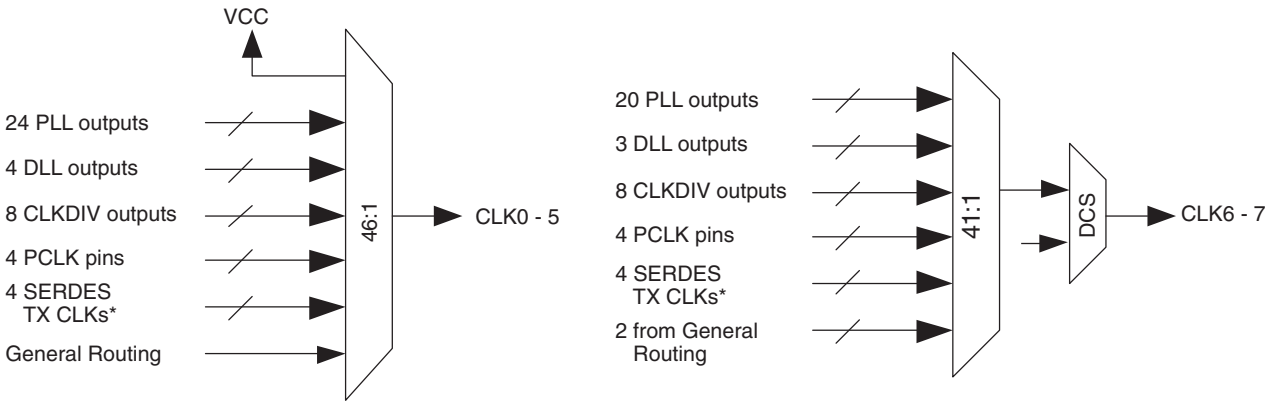


Figure 10-35. LatticeECP2M Primary Clock Muxes

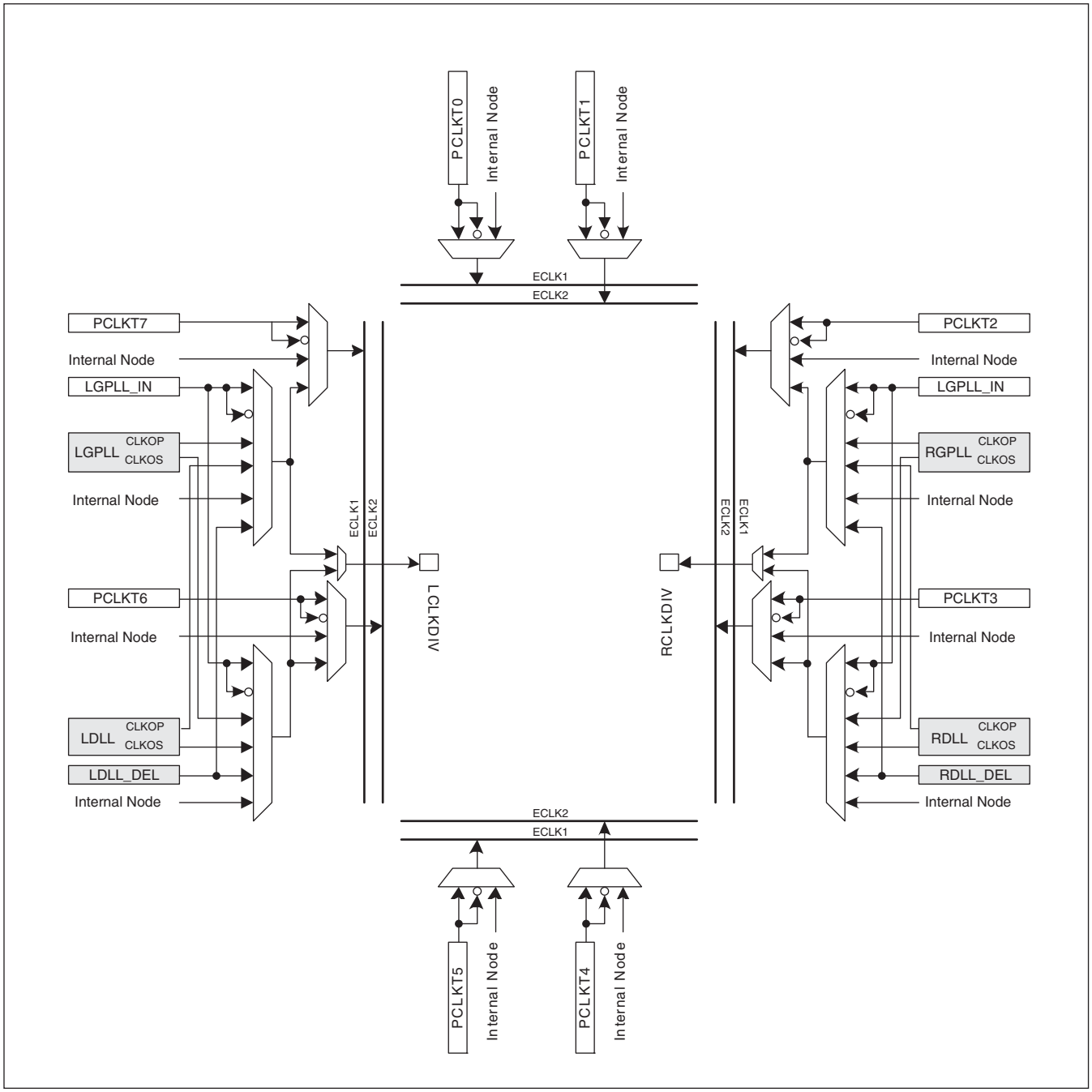


*LatticeECP2M-50 has two SERDES TX CLK outputs

Appendix B. PLL, DLL, CLKIDV and ECLK Locations and Connectivity

Figure 10-36 shows the locations, site names, and connectivity of the PLLs, DLLs, CLKDIVs and ECLKs

Figure 10-36. PLL, DLL, CLKIDV and ECLK Locations and Connectivity



Appendix C. Clock Preferences

A few key clock preferences are introduced below. Refer to the software “Help” file for other preferences and detailed information.

ASIC
The following preference command assigns a phase of 90° to the TRDLLA_CLKOS:

```
ASIC "my_dll" TYPE "TRDLLA" CLKOS_PHASE=90;
```

FREQUENCY
The following physical preference command assigns a frequency of 100 MHz to a net named clk1:

```
FREQUENCY NET "clk1" 100 MHz;
```

The following preference specifies a hold margin value for each clock domain:

```
FREQUENCY NET "RX_CLKA_CMOS_c" 100.000 MHz HOLD_MARGIN 1 ns;
```

MAXSKEW
The following command assigns a maximum skew of 5ns to a net named NetB:

```
MAXSKEW NET "NetB" 5 NS;
```

MULTICYCLE
The following command will relax the period to 50ns for the path starting at COMPA to COMPB (NET1):

```
MULTICYCLE "PATH1" START COMP "COMPA" END COMP "COMPB" NET "NET1" 50 NS ;
```

PERIOD
The following command assigns a clock period of 30ns to the port named Clk1:

```
PERIOD PORT "Clk1" 30 NS;
```

PROHIBIT
This command prohibits the use of a primary clock to route a clock net named bf_clk:

```
PROHIBIT PRIMARY NET "bf_clk";
```

USE PRIMARY
Use a primary clock resource to route the specified net.

```
USE PRIMARY NET clk_fast;
USE PRIMARY DCS NET "bf_clk";
USE PRIMARY PURE NET “bf_clk” QUADRANT_TL;
```

USE SECONDARY
Use a secondary clock resource to route the specified net.

```
USE SECONDARY NET "clk_lessfast" QUADRANT_TL;
```

USE EDGE
Use a edge clock resource to route the specified net.

```
USE EDGE NET “clk_fast”;
```

CLOCK_TO_OUT
Specifies a maximum allowable output delay relative to a clock.

Here are two preferences using both the CLKPORT and CLKNET keywords showing the corresponding scope of TRACE reporting.

The CLKNET will stop tracing the path before the PLL, so you will not get PLL compensation timing numbers.

```
CLOCK_TO_OUT PORT "RxAddr_0" 6.000000 ns CLKNET "pll_rxclk" ;
```

The above preference will yield the following clock path:

```
Physical Path Details:
     Clock path pll_inst/pll_utp_0_0 to PFU_33:
  Name    Fanout   Delay (ns)          Site               Resource
ROUTE        49     2.892    ULPPLL.MCLK to     R3C14.CLK0 pll_rxclk
                 --------
                   2.892   (0.0% logic, 100.0% route), 0 logic levels.
```

If CLKPORT is used, the trace is complete back to the clock port resource and provides PLL compensation timing numbers.

```
CLOCK_TO_OUT PORT "RxAddr_0" 6.000000 ns CLKPORT "RxClk" ;
```

The above preference will yield the following clock path:

```
Clock path RxClk to PFU_33:
  Name    Fanout   Delay (ns)          Site               Resource
IN_DEL      ---     1.431         D5.PAD to        D5.INCK RxClk
ROUTE         1     0.843        D5.INCK to   ULPPLL.CLKIN RxClk_c
MCLK_DEL    ---     3.605   ULPPLL.CLKIN to    ULPPLL.MCLK pll_inst/pll_utp_0_0
ROUTE        49     2.892    ULPPLL.MCLK to     R3C14.CLK0 pll_rxclk
                 --------
                   8.771   (57.4% logic, 42.6% route), 2 logic levels.
```

INPUT_SETUP
Specifies an setup time requirement for input ports relative to a clock net.

```
INPUT_SETUP PORT "datain" 2.000000 ns HOLD 1.000000 ns CLKPORT "clk"
PLL_PHASE_BACK ;
```

PLL_PHASE_BACK
This preference is used with INPUT_SETUP when the user needs a Trace calculation based on the previous clock edge.

This preference is useful when setting the PLL output Phase Adjustment. Since there is no negative phase adjustment provided, the PLL_PHASE_BACK preference works as if negative phase adjustment is available.

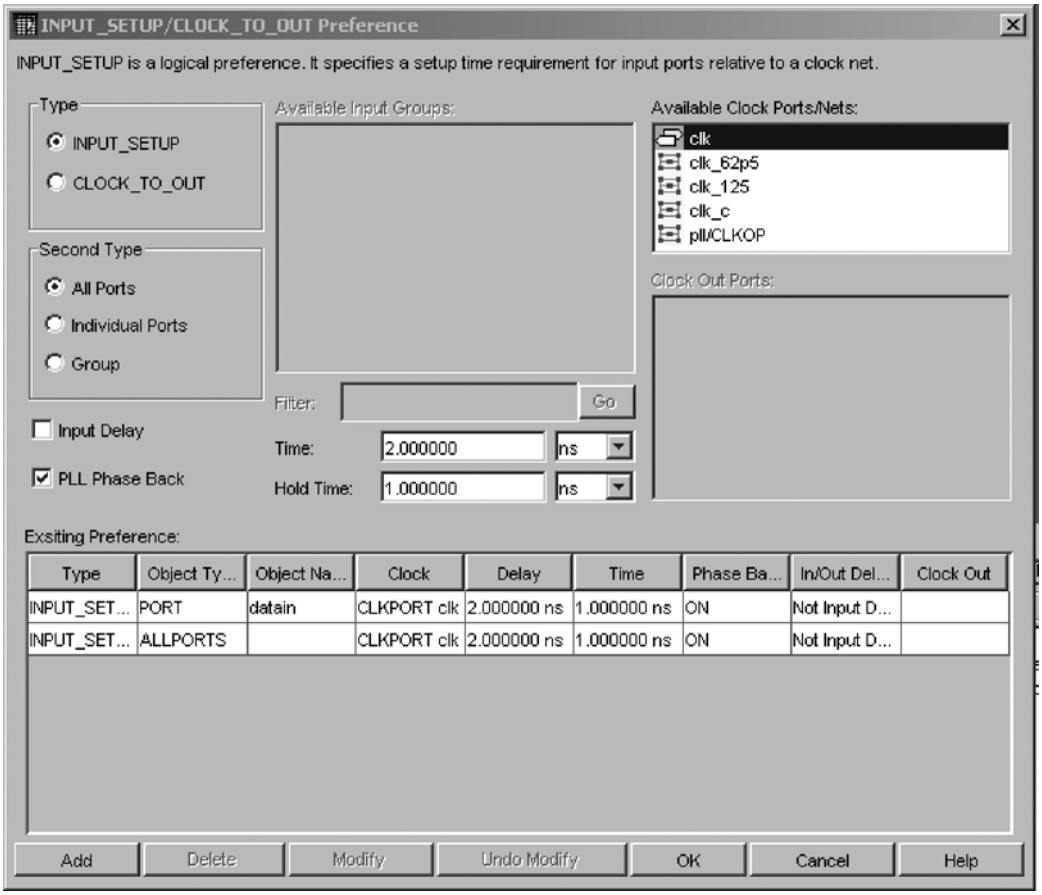
For example:

If a Phase Adjustment of -90° of CLKOS is desired, the user can set the Phase to 270° and set the INPUT_SETUP preference with PLL_PHASE_BACK.

PLL_PHASE_BACK Usage in Pre-Map Preference Editor: The Pre-Map Preference Editor can be used to set the PLL_PHASE_BACK attribute.

1. Open the Design Planner (Pre-Map).
2. In the Design Planner control window, select **Spreadsheet View** under **View**.
3. In the Spreadsheet View window, select **Input_setup/Clock_to_out…**
4. The INPUT_SETUP/CLOCK_TO_OUT Preference window is shown in Figure 10-37.

Figure 10-37. INPUT_SETUP/CLOCK_TO_OUT Preference Window (see Appendix D Figure 10-41 for Diamond Equivalent)



Appendix D. Lattice Diamond Usage Overview

This appendix discusses the use of Lattice Diamond design software for projects that include the LatticeECP2M SERDES/PCS module .

For general information about the use of Lattice Diamond, refer to the Lattice Diamond Tutorial.

If you have been using ispLEVER software for your FPGA design projects, Lattice Diamond may look like a big change. But if you look closer, you will find many similarities because Lattice Diamond is based on the same toolset and work flow as ispLEVER. The changes are intended to provide a simpler, more integrated, and more enhanced user interface.

Converting an ispLEVER Project to Lattice Diamond

Design projects created in ispLEVER can easily be imported into Lattice Diamond. The process is automatic except for the ispLEVER process properties, which are similar to the Diamond strategy settings, and PCS modules. After importing a project, you need to set up a strategy for it and regenerate any PCS modules.

Importing an ispLEVER Design Project

Make a backup copy of the ispLEVER project or make a new copy that will become the Diamond project.

1. In Diamond, choose **File > Open > Import ispLEVER Project**.
2. In the ispLEVER Project dialog box, browse to the project's .syn file and open it.
3. If desired, change the base file name or location for the Diamond project. If you change the location, the new Diamond files will go into the new location, but the original source files will not move or be copied. The Diamond project will reference the source files in the original location.

The project files are converted to Diamond format with the default strategy settings.

Adjusting PCS Modules

PCS modules created with IPexpress have an unusual file structure and need additional adjustment when importing a project from ispLEVER. There are two ways to do this adjustment. The preferred method is to regenerate the module in Diamond. However this may upgrade the module to a more recent version. An upgrade is usually desirable but if, for some reason, you do not want to upgrade the PCS module, you can manually adjust the module by copying its .txt file into the implementation folder. If you use this method, you must remember to copy the .txt file into any future implementation folders.

Regenerate PCS Modules

1. Find the PCS module in the Input Files folder of File List view. The module may be represented by an .lpc, .v, or .vhd file.
2. If the File List view shows the Verilog or VHDL file for the module, and you want to regenerate the module, import the module's .lpc file:
   a. In the File List view, right-click the implementation folder () and choose **Add > Existing File**.
   b. Browse for the module's .lpc file, **<*module_name*>.lpc**, and select it.
   c. Click **Add**. The .lpc file is added to the File List view.
   d. Right-click the module's Verilog or VHDL file and choose **Remove**.
3. In File List, double-click the module's .lpc file. The module's IPexpress dialog box opens.
4. In the bottom of the dialog box, click **Generate**. The Generate Log tab is displayed. Check for errors and close.

In File List, the .lpc file is replaced with an .ipx file. The IPexpress manifest (.ipx) file is new with Diamond. The .ipx file keeps track of the files needed for complex modules.

Using IPexpress with Lattice Diamond

Using IPexpress with Lattice Diamond is essentially same as with ispLEVER.

The configuration GUI tabs are all the same except for the Generation Options tab. Figure 10-38 shows the Generation Options tab window.

Figure 10-38. Generation Options Tab

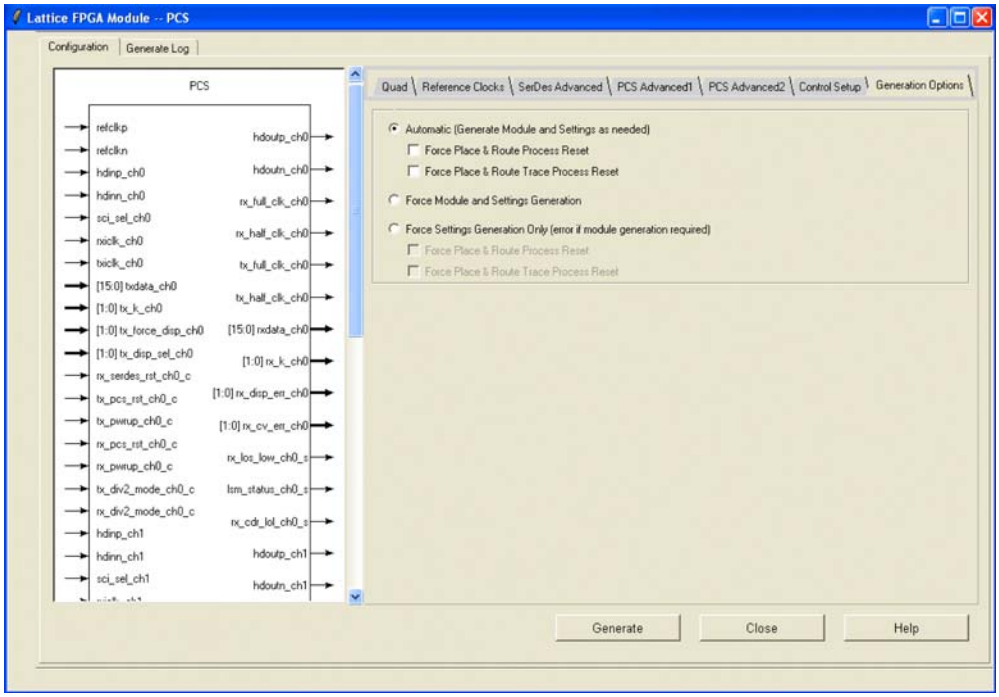


Table 10-18. SERDES_PCS GUI Attributes – Generation Options Tab

GUI Text	Description
Automatic	Automatically generates the HDL and configuration(.txt) files as needed. Some changes do not require regenerating both files.
Force Module and Settings Generation	Generates both the HDL and configuration files.
Force Settings Generation Only	Generates only the attributes file. You get an error message if the HDL file also needs to be generated.
Force Place & Route Process Reset	Resets the Place & Route Design process, forcing it to be run again with the newly generated PCS module.
Force Place & Route Trace Process Reset	Resets the Place & Route Trace process, forcing it to be run again with the newly generated PCS module.

Note:
Automatic is set as the default option. If either Automatic or Force Settings Generation Only and no sub-options (Process Reset Options) are checked and the HDL module is not generated, the reset pointer is set to Bitstream generation automatically.

After the Generation is finished, the reset marks in the process window will be reset accordingly.

Creating a New Simulation Project Using Simulation Wizard

This section describes how to use the Simulation Wizard to create a simulation project (.spf) file so you can import it into a standalone simulator.

1. In Project Navigator, click **Tools > Simulation Wizard**. The Simulation Wizard opens.
2. In the Preparing the Simulator Interface page click **Next**.
3. In the Simulator Project Name page, enter the name of your project in the Project Name text box and browse to the file path location where you want to put your simulation project using the Project Location text box and Browse button.

   When you designate a project name in this wizard page, a corresponding folder will be created in the file path you choose. Click **Yes** in the popup dialog that asks you if you wish to create a new folder.
4. Click either the Active-HDL® or ModelSim® simulator check box and click **Next**.
5. In the Process Stage page choose which type of Process Stage of simulation project you wish to create Valid types are RTL, Post-Synthesis Gate-Level, Post-Map Gate-Level, and Post-Route Gate-level+Timing. Only those process stages that are available are activated.

   Note that you can make a new selection for the current strategy if you have more than one defined in your project.

   The software supports multiple strategies per project implementation which allow you to experiment with alternative optimization options across a common set of source files. Since each strategy may have been processed to different stages, this dialog allows you to specify which stage you wish to load.
6. In the Add Source page, select from the source files listed in the Source Files list box or use the browse button on the right to choose another desired source file. Note that if you wish to keep the source files in the local simulation project directory you just created, check the **Copy Source to Simulation Directory** option.
7. Click **Next** and a Summary page appears and provides information on the project selections including the simulation libraries. By default, the Run Simulator check box is enabled and will launch the simulation tool you chose earlier in the wizard in the Simulator Project Name page.
8. Click **Finish**.

The Simulation Wizard Project (.spf) file and a simulation script DO file are generated after running the wizard. You can import the DO file into your current project if desired. If you are using Active-HDL, the wizard will generate an .ado file and if you are using ModelSim, it creates and .mdo file.

Note: PCS configuration file, (.txt) must be added in step 6.

Figure 10-39. Diamond Spreadsheet View (see Figure 10-5 for ispLEVER Equivalent)

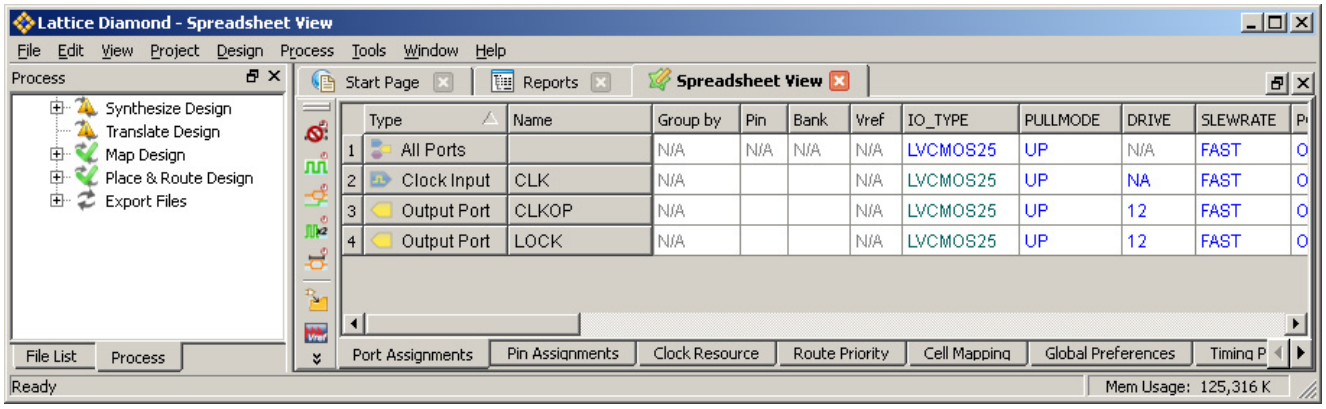


Figure 10-40. Diamond IPexpress Main Window (see Figure 10-14 for ispLEVER Equivalent)

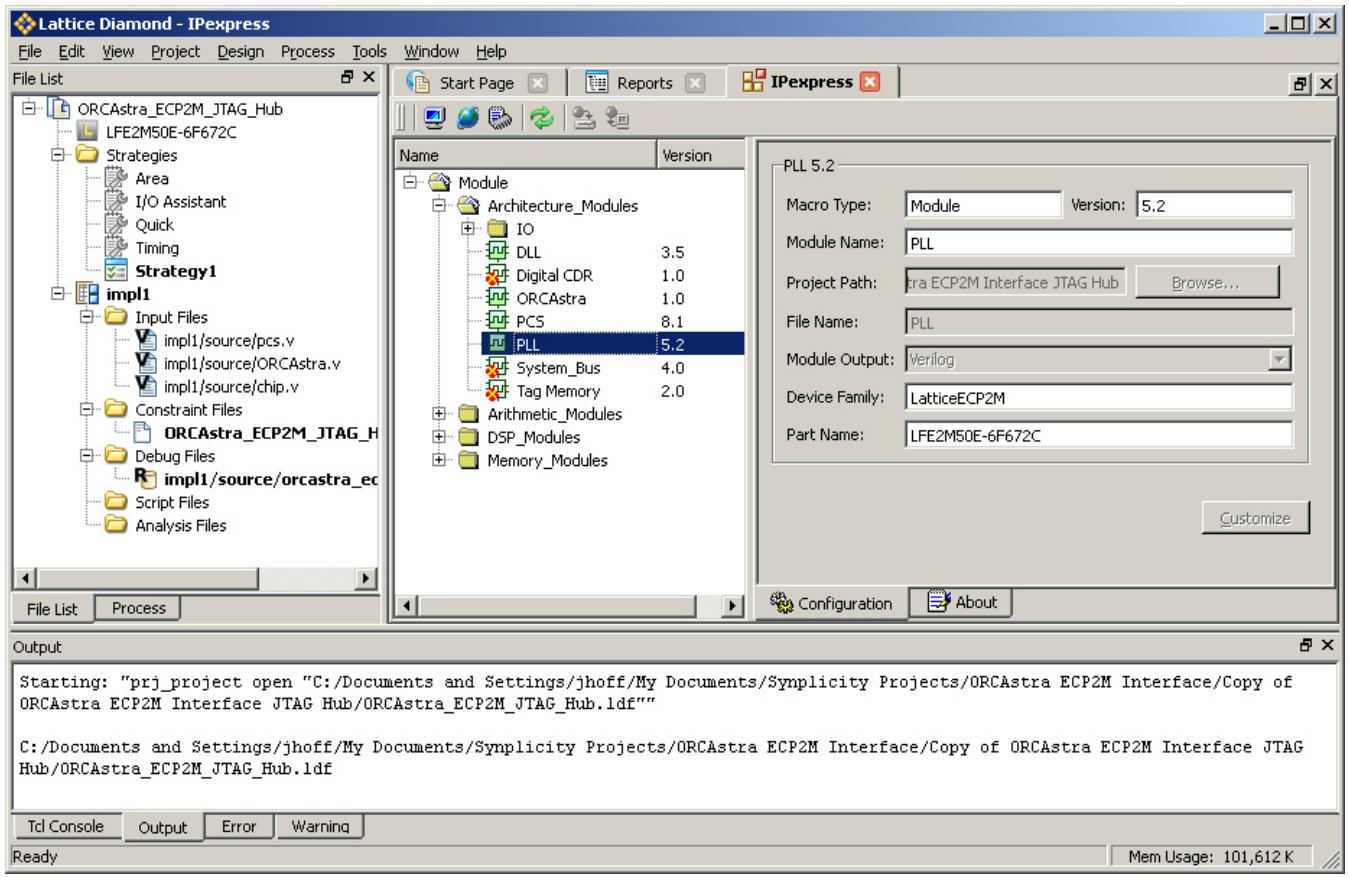
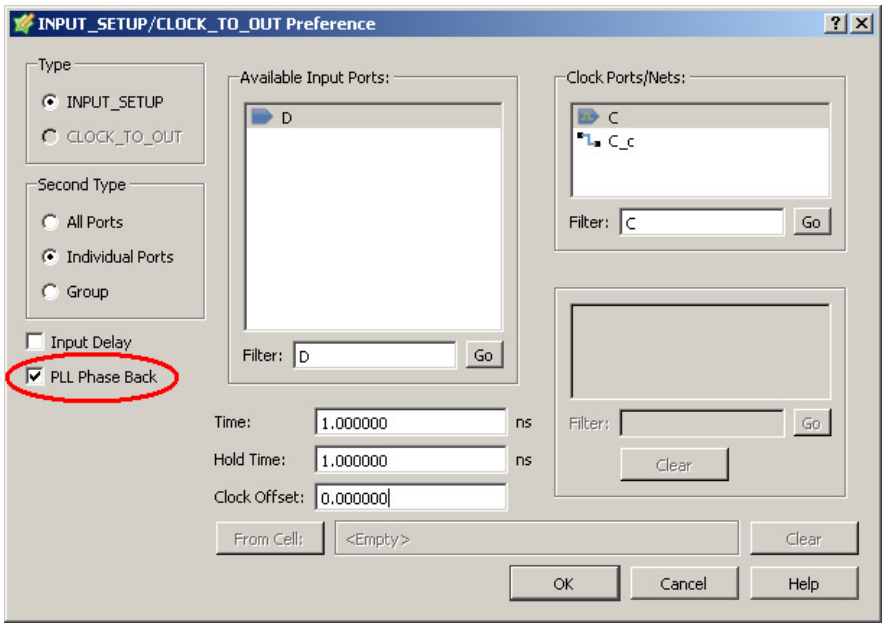


Figure 10-41. Diamond INPUT_SETUP Preference Window (see Figure 10-37 for ispLEVER Equivalent)



LatticeECP2/M Memory Usage Guide

July 2011 Technical Note TN1104

Introduction

This technical note discusses memory usage for the LatticeECP2™ and LatticeECP2M™ device families. It is intended to be used by design engineers as a guide for integrating the EBR- (Embedded Block RAM) and PFU-based memories in this device family using the ispLEVER® design tool.

The architecture of these devices provides resources for FPGA on-chip memory applications. The sysMEM™ EBR complements the distributed PFU-based memory. Single-Port RAM, Dual-Port RAM, Pseudo Dual-Port RAM, FIFO and ROM memories can be constructed using the EBR. LUTs and PFU can implement Distributed Single-Port RAM, Dual-Port RAM and ROM.

The capabilities of the EBR RAM and PFU RAM are referred to as primitives and are described later in this document. Designers can utilize the memory primitives in two ways via the IPexpress™ tool in the ispLEVER software. The IPexpress GUI allows users to specify the memory type and size required. IPexpress takes this specification and constructs a netlist to implement the desired memory by using one or more of the memory primitives.

The remainder of this document discusses the use of IPexpress, memory modules and memory primitives.

Memories in LatticeECP2/M Devices

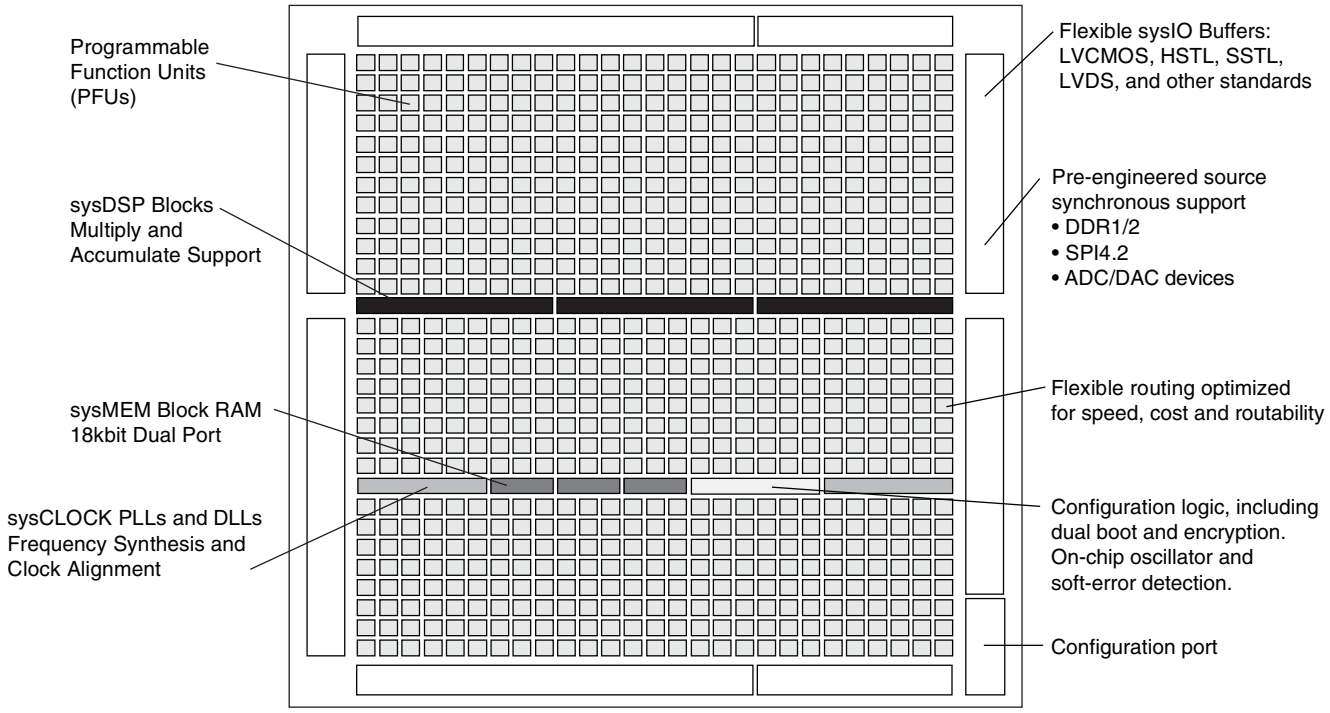
There are two kinds of logic blocks, the Programmable Functional Unit (PFU) and Programmable Functional Unit without RAM (PFF). The PFU contains the building blocks for logic, arithmetic, RAM, ROM and register functions. The PFF block contains building blocks for logic, arithmetic and ROM functions. Both PFU and PFF blocks are optimized for flexibility allowing complex designs to be implemented quickly and efficiently. Logic Blocks are arranged in a two-dimensional array. Only one type of block is used per row.

The LatticeECP2 family of devices contains up to two rows of sysMEM EBR blocks and the LatticeECP2M family of devices contains up to seven rows of sysMEM EBR blocks. sysMEM EBRs are large, dedicated 18K fast memory blocks. Each sysMEM block can be configured in a variety of depths and widths of RAM or ROM. In addition, LatticeECP2/M devices contain up to two rows of sysDSP™ blocks. Each sysDSP block has multipliers and accumulators, which are the building blocks for complex signal processing capabilities

Table 11-1. LatticeECP2/M LUT and Memory Densities

LUTs	6K	12K	20K		35K		50K		70K		100K
Device	**ECP2-6**	**ECP2-12**	**ECP2-20**	**ECP2M-20**	**ECP2-35**	**ECP2M-35**	**ECP2-50**	**ECP2M-50**	**ECP2-70**	**ECP2M-70**	**ECP2M-100**
LUTs (K)	6	12	21	19	32	34	48	48	68	67	95
Distributed RAM (Kbits)	12	24	42	41	65	71	96	101	136	145	202
EBR SRAM Blocks	3	12	15	66	18	114	21	225	60	246	288
EBR SRAM (Kbits)	55	221	276	1217	332	2101	387	4147	1106	4534	5308

Figure 11-1. Simplified Block Diagram, LatticeECP2-6 Device (Top Level)



Utilizing IPexpress

Designers can utilize IPexpress to easily specify a variety of memories in their designs. These modules are constructed using one or more memory primitives along with general purpose routing and LUTs, as required. The available primitives are:

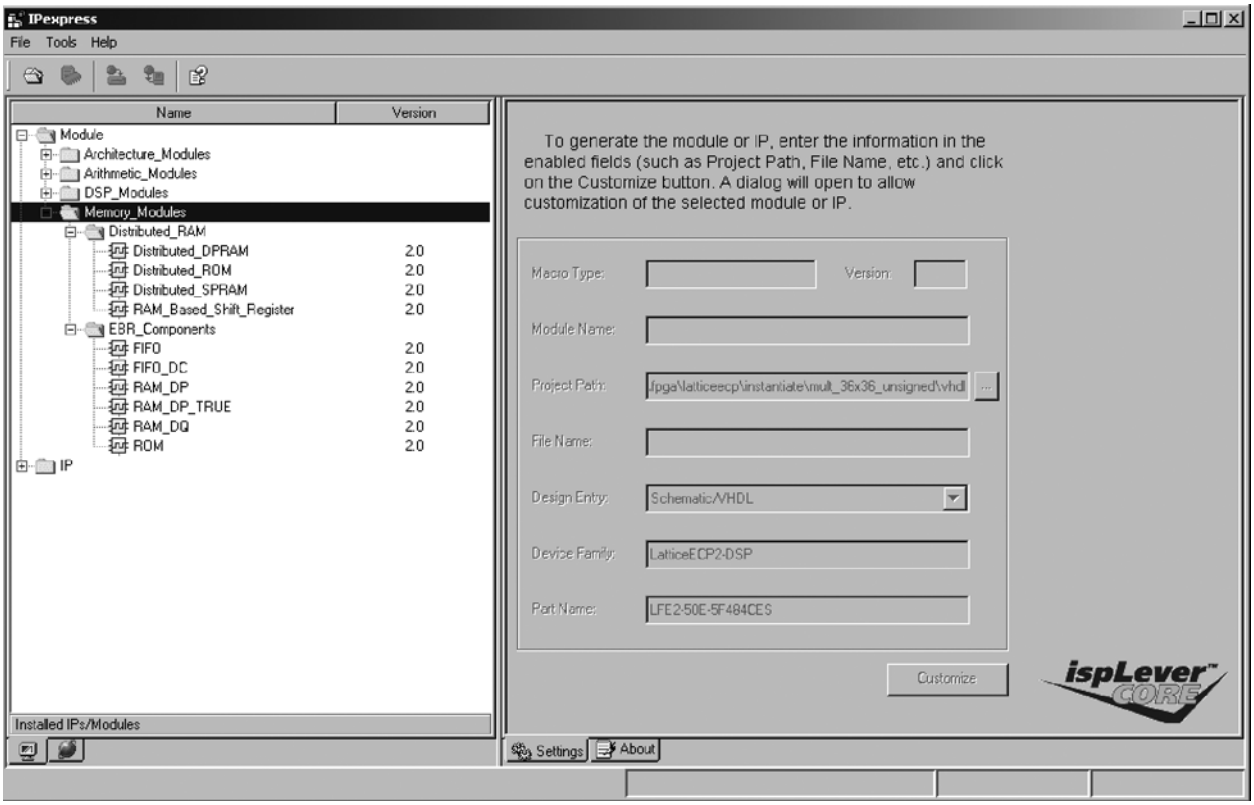
- Single Port RAM (RAM_DQ) – EBR-based
- Dual PORT RAM (RAM_DP_TRUE) – EBR-based
- Pseudo Dual Port RAM (RAM_DP) – EBR-based
- Read Only Memory (ROM) – EBR-Based
- First In First Out Memory (Dual Clock) (FIFO_DC) – EBR-based
- Distributed Single Port RAM (Distributed_SPRAM) – PFU-based
- Distributed Dual Port RAM (Distributed_DPRAM) – PFU-based
- Distributed ROM (Distributed_ROM) – PFU/PFF-based
- Distributed Shift Register (RAM_Based_Shift_Register) – PFU-based (see IPexpress Help for details)

IPexpress Flow

For generating any of these memories, create (or open) a project for the LatticeECP2/M devices.

From the Project Navigator, select **Tools > IPexpress** or click on the button in the toolbar when LatticeECP2/M devices are targeted in the project. This opens the IPexpress main window as shown in Figure 11-2.

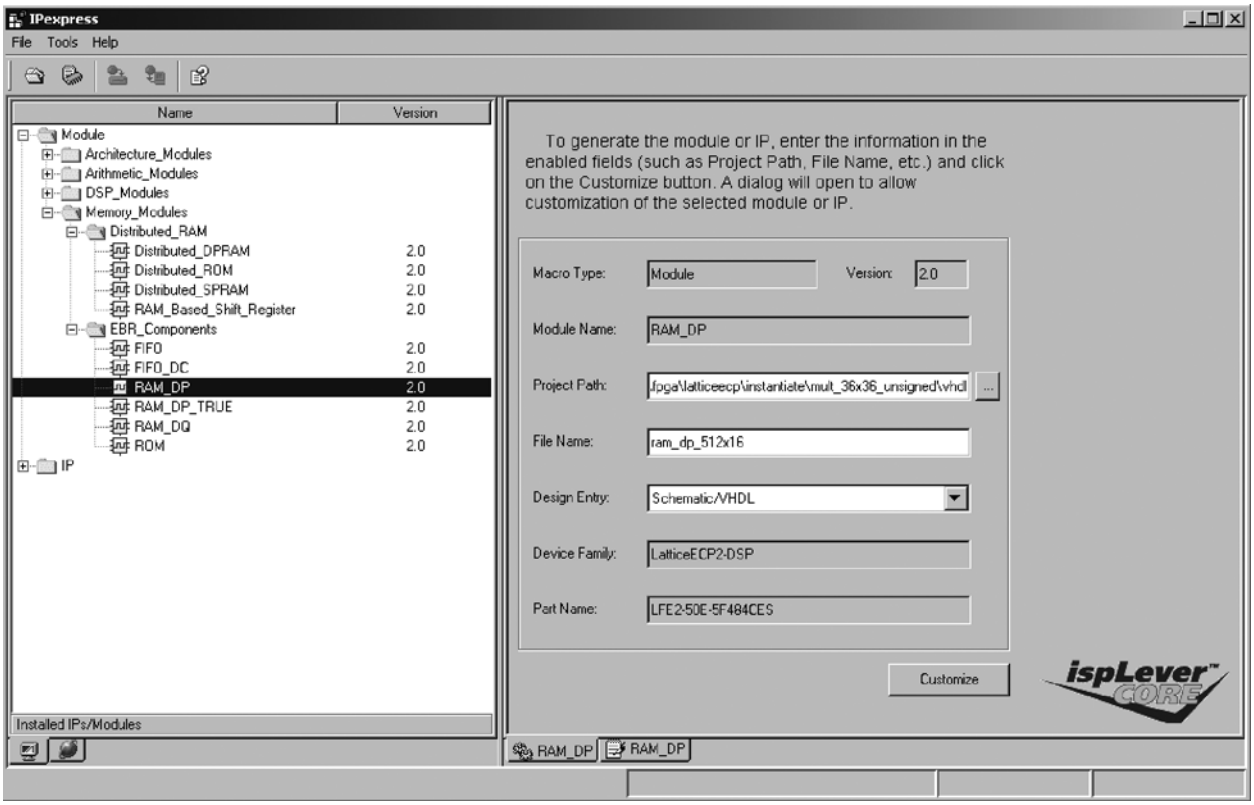
Figure 11-2. IPexpress – Main Window



The left pane of this window includes the Module Tree. The EBR-based Memory Modules are under the **EBR_Components** and the PFU-based Distributed Memory Modules are under **Storage_Components**, as shown in Figure 11-2.

As an example, let us consider generating an EBR-based Pseudo Dual Port RAM of size 512x16. Select **RAM_DP** under **EBR_Components**. The right pane changes as shown in Figure 11-3.

Figure 11-3. Example Generating Pseudo Dual Port RAM (RAM_DP) Using IPexpress



In the right pane, options like the **Device Family**, **Macro Type**, **Category**, and **Module_Name** are device and selected module dependent. These cannot be changed in IPexpress.

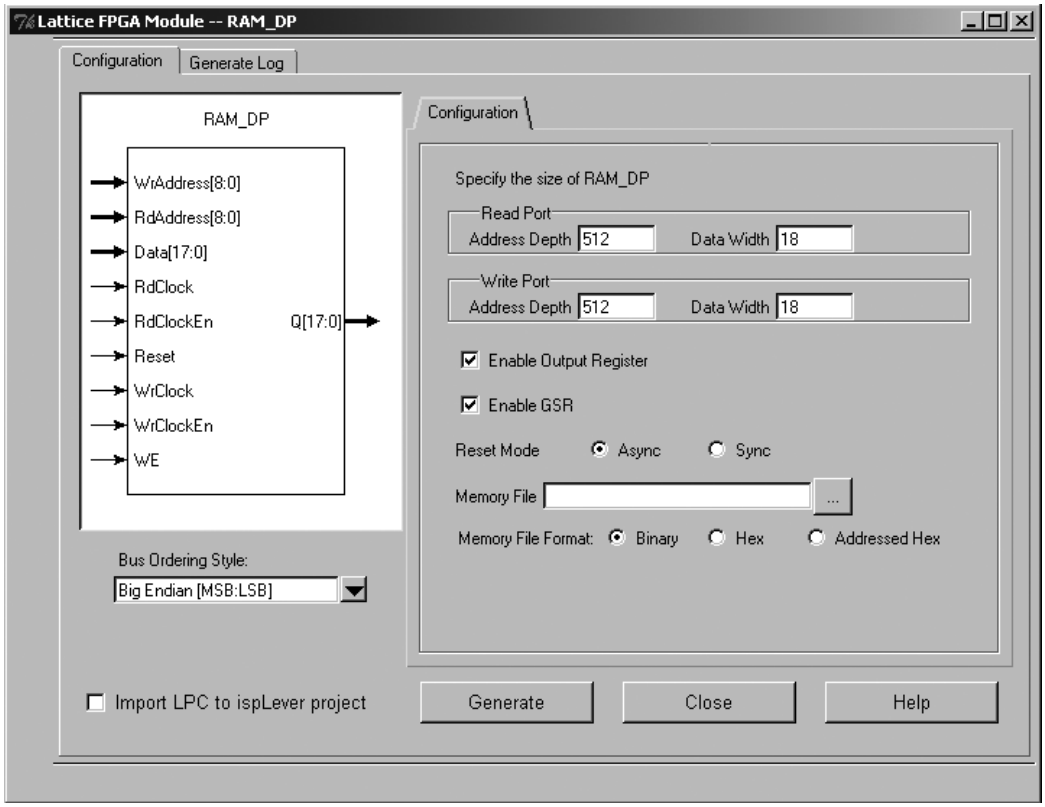
Users can change the directory where the generated module files will be placed by clicking the **Browse** button in the **Project Path**.

The **Module Name** text box allows users to specify an entity name for the module they are about to generate. Users must provide this entity name.

Design entry, Verilog or VHDL, by default, is the same as the project type. If the project is a VHDL project, the selected design entry option will be “Schematic/ VHDL”, and “Schematic/ Verilog-HDL” if the project type is Verilog-HDL.

The **Device** pull-down menu allows users to select different devices within the same family, LatticeECP2/M in this example. By clicking the **Customize** button, another window opens where users can customize the RAM (Figure 11-4).

Figure 11-4. Example Generating Pseudo Dual Port RAM (RAM_DP) Module Customization



The left side of this window shows the block diagram of the module. The right side includes the **Configuration** tab where users can choose options to customize the RAM_DP (e.g. specify the address port sizes and data widths).

Users can specify the address depth and data width for the **Read Port** and the **Write Port** in the text boxes provided. In this example, we are generating a Pseudo Dual Port RAM of size 512 x 16. Users can also create RAMs of different port widths for Pseudo Dual Port and True Dual Port RAMs.

The Input Data and the Address Control are always registered, as the hardware only supports the clocked write operation for the EBR based RAMs. The check box **Enable Output Registers** inserts the output registers in the Read Data Port. Output registers are optional for EBR-based RAMs.

Users have the option to set the **Reset Mode** as Asynchronous Reset or Synchronous Reset. **Enable GSR** can be checked to enable the Global Set Reset. If an EBR is pre-loaded during configuration, the GSR input must be disabled or the release of the GSR during device Wake Up must occur before the release of the device I/Os becomes active. These instructions apply to all EBR RAM and ROM implementations. Note that there are no reset restrictions if the EBR synchronous reset is used and the EBR GSR input is disabled.

Users can also pre-initialize their memory with the contents specified in the **Memory File**. It is optional to provide this file in the RAM; however for ROM, the Memory File is required. These files can be of Binary, Hex or Addresses Hex format. The details of these formats are discussed in the Initialization File section of this document.

At this point, users can click the **Generate** button to generate the module they have customized. A VHDL or Verilog netlist is then generated and placed in the specified location. Users can incorporate this netlist in their designs.

Another important button is the **Load Parameters** button. IPexpress stores the parameters specified in a <module_name>.lpc file. This file is generated along with the module. Users can click on the Load Parameters button to load the parameters of a previously generated module to re-visit or make changes to them.

Once the module is generated, users can either instantiate the *.lpc or the Verilog-HDL/ VHDL file in top-level module of their design.

The various memory modules, both EBR and distributed, are discussed in detail in this document.

Memory Modules

ECC is supported in most memories. If you choose to use ECC, you will have a 2-bit error signal.

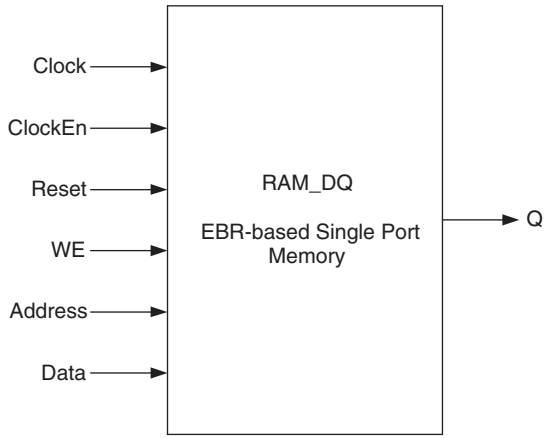
- When Error[1:0]=00, there is no error.
- When Error[0]=1, it indicates that there was a 1 bit error which was fixed.
- When Error[1]=1, it indicates that there was a 2-bit error which cannot be corrected.

Single Port RAM (RAM_DQ) – EBR Based

The EBR blocks in LatticeECP2/M devices can be configured as Single Port RAM or RAM_DQ. IPexpress allows users to generate the Verilog-HDL or VHDL along EDIF netlist for the memory size as per design requirements.

IPexpress generates the memory module as shown in Figure 11-5.

Figure 11-5. Single Port Memory Module Generated by IPexpress



Since the device has a number of EBR blocks, the generated module makes use of these EBR blocks, or primitives, and cascades them to create the memory sizes specified by the user in the IPexpress GUI. For memory sizes smaller than an EBR block, the module will be created in one EBR block. For memory sizes larger than one EBR block, multiple EBR blocks can be cascaded in depth or width (as required to create these sizes).

In Single Port RAM mode, the input data and address for the ports are registered at the input of the memory array. The output data of the memory is optionally registered at the output.

The various ports and their definitions for the Single Port Memory are listed in Table 11-2. The table lists the corresponding ports for the module generated by IPexpress and for the EBR RAM_DQ primitive.

Table 11-2. EBR-based Single Port Memory Port Definitions

Port Name in Generated Module	Port Name in the EBR Block Primitive	Description	Active State
Clock	CLK	Clock	Rising Clock Edge
ClockEn	CE	Clock Enable	Active High
Address	AD[x:0]	Address Bus	—
Data	DI[y:0]	Data In	—
Q	DO[y:0]	Data Out	—
WE	WE	Write Enable	Active High
Reset	RST	Reset	Active High
—	CS[2:0]	Chip Select	—

Reset (or RST) resets only the input and output registers of the RAM. It does not reset the contents of the memory.

Chip Select (CS) is a useful port in the EBR primitive when multiple cascaded EBR blocks are required by the memory. The CS signal forms the MSB for the address when multiple EBR blocks are cascaded. CS is a 3-bit bus, so it can cascade eight memories easily. If the memory size specified by the user requires more than eight EBR blocks, the ispLEVER software automatically generates the additional address decoding logic, which is implemented in the PFU (external to the EBR blocks).

Each EBR block consists of 18,432 bits of RAM. The values for x (address) and y (data) for each EBR block for the devices are listed in Table 11-3.

Table 11-3. Single Port Memory Sizes for 16K Memories for LatticeECP2/M

Single Port Memory Size	Input Data	Output Data	Address [MSB:LSB]
16K x 1	DI	DO	AD[13:0]
8K x 2	DI[1:0]	DO[1:0]	AD[12:0]
4K x 4	DI[3:0]	DO[3:0]	AD[11:0]
2K x 9	DI[8:0]	DO[8:0]	AD[10:0]
1K x 18	DI[17:0]	DO[17:0]	AD[9:0]
512 x 36	DI[35:0]	DO[35:0]	AD[8:0]

Table 11-4 shows the various attributes available for the Single Port Memory (RAM_DQ). Some of these attributes are user-selectable through the IPexpress GUI. For detailed attribute definitions, refer to Appendix A.

Table 11-4. Single Port RAM Attributes for LatticeECP2/M

Attribute	Description	Values	Default Value	User Selectable Through IPexpress
Address depth	Address Depth Read Port	16K, 8K, 4K, 2K, 1K, 512		YES
Data Width	Data Word Width Read Port	1, 2, 4, 9, 18, 36	1	YES
Enable Output Registers	Register Mode (Pipelining) for Write Port	NOREG, OUTREG	NOREG	YES
Enable GSR	Enables Global Set Reset	ENABLE, DISABLE	ENABLE	YES
Reset Mode	Selects the Reset type	ASYNC, SYNC	ASYNC	YES
Memory File Format		BINARY, HEX, ADDRESSED HEX		YES
Write Mode	Read / Write Mode for Write Port	NORMAL, WRITE-THROUGH	NORMAL	YES
Chip Select Decode	Chip Select Decode for Read Port	0b000, 0b001, 0b010, 0b011, 0b100, 0b101, 0b110, 0b111	0b000	NO
Init Value	Initialization value	0x00000000000000000000 00000000000000000000 00000000000000000000 0000000000000.....0xFFFF FFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFF	0x000000000 00000000000 00000000000 00000000000 00000000000 00000000000 00000000000 00000	NO

The Single Port RAM (RAM_DQ) can be configured as NORMAL or WRITE THROUGH modes. Each of these modes affects the data coming out of port Q of the memory during the write operation followed by the read operation at the same memory location.

Additionally, users can select to enable the output registers for RAM_DQ. Figures 11-6-11-9 show the internal timing waveforms for the Single Port RAM (RAM_DQ) with these options.

It is important that no setup and hold time violations occur on the address registers (Address). Failing to meet these requirements can result in corruption of memory contents. This applies to both read and write operations.

A Post Place and Route timing report in Lattice Diamond® or ispLEVER design software can be run to verify that no such timing errors occur. Refer to the timing preferences in the Online Help documents.

Figure 11-6. Single Port RAM Timing Waveform – NORMAL Mode, without Output Registers

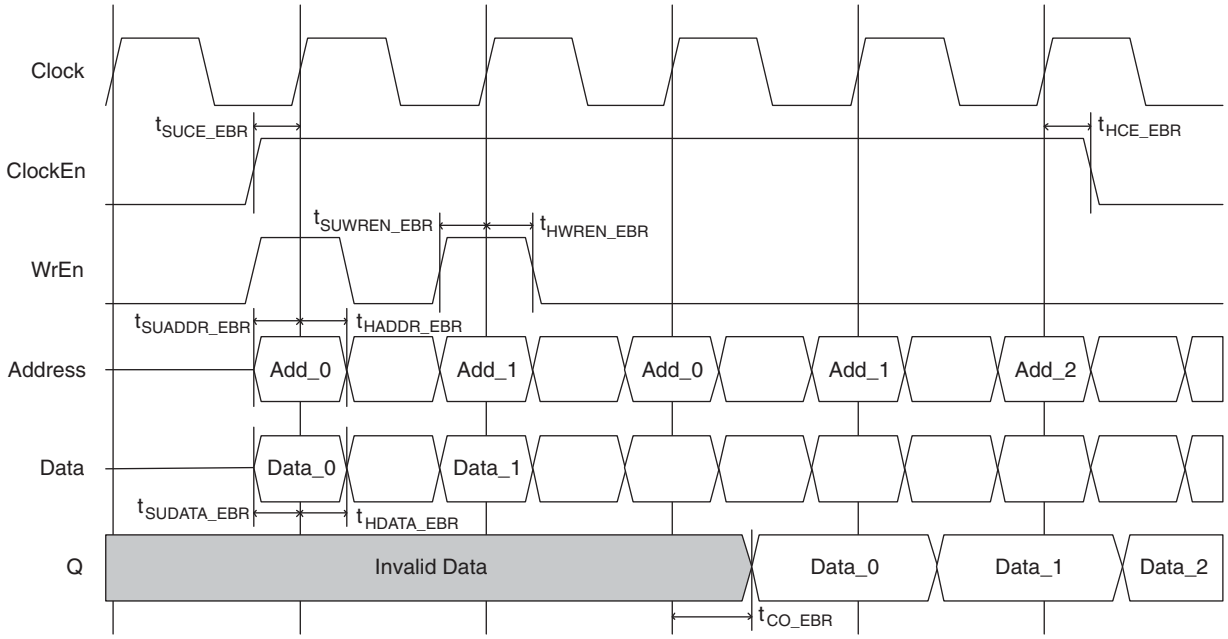


Figure 11-7. Single Port RAM Timing Waveform – NORMAL Mode, with Output Registers

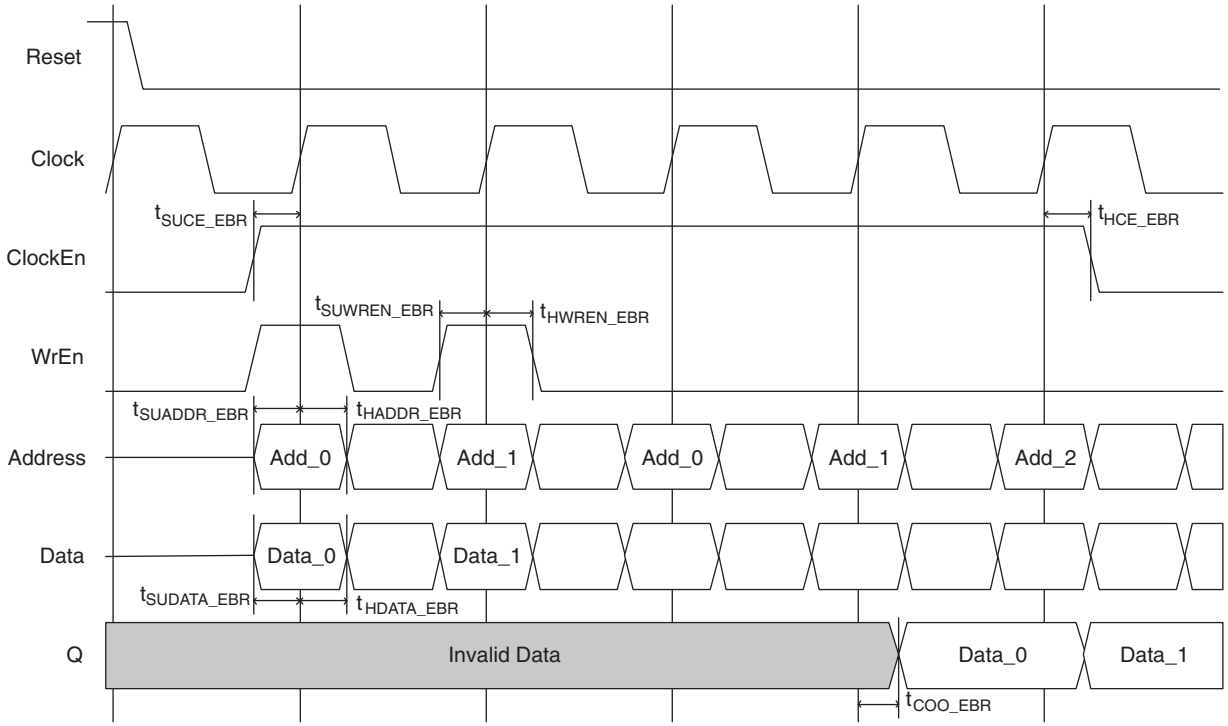


Figure 11-8. Single Port RAM Timing Waveform – WRITE THROUGH Mode, without Output Registers

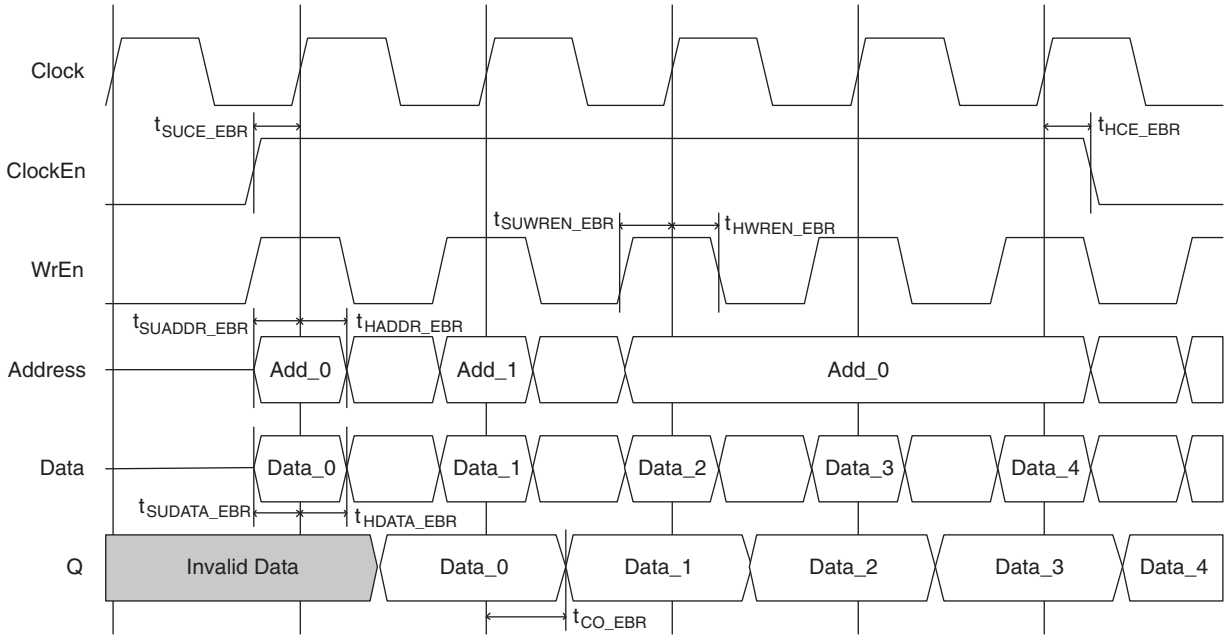
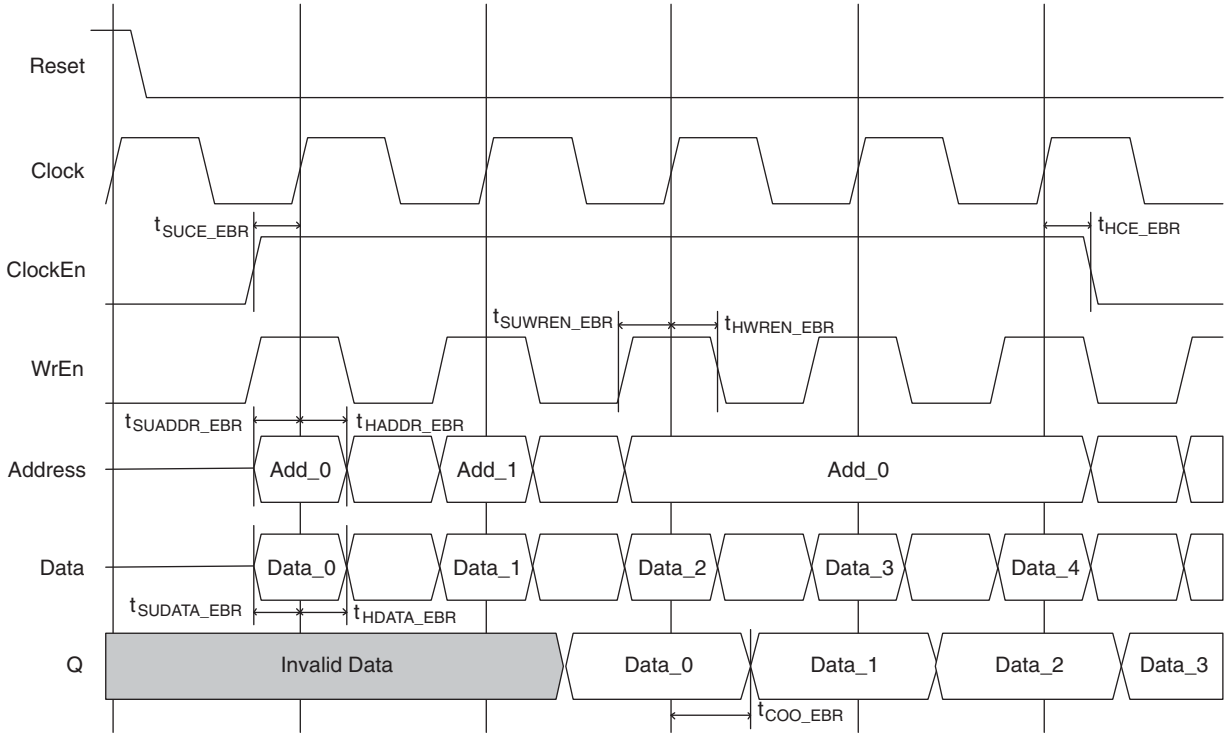


Figure 11-9. Single Port RAM Timing Waveform – WRITE THROUGH Mode, with Output Registers

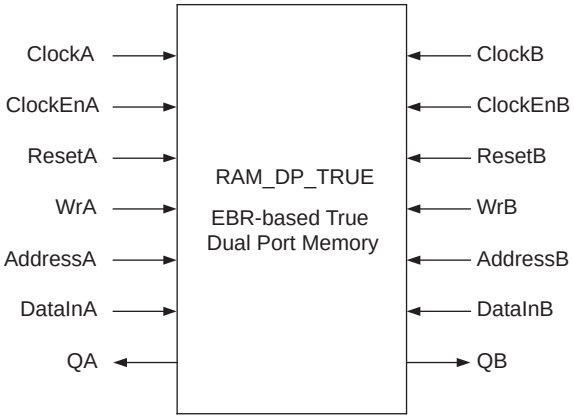


True Dual Port RAM (RAM_DP_TRUE) – EBR Based

The EBR blocks in the LatticeECP2/M devices can be configured as True-Dual Port RAM or RAM_DP_TRUE. IPexpress allows users to generate the Verilog-HDL, VHDL or EDIF netlists for the memory size as per design requirements.

IPexpress generates the memory module as shown in Figure 11-10.

Figure 11-10. True Dual Port Memory Module Generated by IPexpress



The generated module makes use of these EBR blocks or primitives. For memory sizes smaller than an EBR block, the module will be created in one EBR block. When the specified memory is larger than one EBR block, multiple EBR blocks can be cascaded in depth or width (as required to create these sizes).

In True Dual Port RAM mode, the input data and address for the ports are registered at the input of the memory array. The output data of the memory is optionally registered at the output.

The various ports and their definitions for Single Port Memory are listed in Table 11-5. The table lists the corresponding ports for the module generated by IPexpress and for the EBR RAM_DP_TRUE primitive.

Table 11-5. EBR-based True Dual Port Memory Port Definitions

Port Name in Generated Module	Port Name in the EBR Block Primitive	Description	Active State
ClockA, ClockB	CLKA, CLKB	Clock for PortA and PortB	Rising Clock Edge
ClockEnA, ClockEnB	CEA, CEB	Clock Enables for Port CLKA and CLKB	Active High
AddressA, AddressB	ADA[18:19-x1], ADB[18:19-x2]	Address Bus port A and port B	—
DataA, DataB	DIA[y1:0], DIB[y2:0]	Input Data port A and port B	—
QA, QB	DOA[y1:0], DOB[y2:0]	Output Data port A and port B	—
WrA, WrB	WEA, WEB	Write enable port A and port B	Active High
ResetA, ResetB	RSTA, RSTB	Reset for PortA and PortB	Active High
—	CSA[2:0], CSB[2:0]	Chip Selects for each port	—

Reset (or RST) resets only the input and output registers of the RAM. It does not reset the contents of the memory.

Chip Select (CS) is a useful port in the EBR primitive when multiple cascaded EBR blocks are required by the memory. The CS signal forms the MSB for the address when multiple EBR blocks are cascaded. Since CS is a 3-bit bus, it can cascade eight memories easily. However, if the memory size specified by the user requires more than eight EBR blocks, the ispLEVER software automatically generates the additional address decoding logic, which is implemented in the PFU external to the EBR blocks.

Each EBR block consists of 18,432 bits of RAM. The values for x's (for address) and y's (data) for each EBR block for the devices are listed in Table 11-6.

Table 11-6. True Dual Port Memory Sizes for 16K Memory for LatticeECP2/M

Dual Port Memory Size	Input Data Port A	Input Data Port B	Output Data Port A	Output Data Port B	Address Port A [MSB:LSB]	Address Port B [MSB:LSB]
16K x 1	DIA	DIB	DOA	DOB	ADA[13:0]	ADB[13:0]
8K x 2	DIA[1:0]	DIB[1:0]	DOA[1:0]	DOB[1:0]	ADA[12:0]	ADB[12:0]
4K x 4	DIA[3:0]	DIB[3:0]	DOA[3:0]	DOB[3:0]	ADA[11:0]	ADB[11:0]
2K x 9	DIA[8:0]	DIB[8:0]	DOA[8:0]	DOB[8:0]	ADA[10:0]	ADB[10:0]
1K x 18	DIA[17:0]	DIB[17:0]	DOA[17:0]	DOB[17:0]	ADA[9:0]	ADB[9:0]

Table 11-7 shows the various attributes available for the Single Port Memory (RAM_DQ). Some of these attributes are user-selectable through the IPexpress GUI. For detailed attribute definitions, refer to the Appendix A.

Table 11-7. True Dual Port RAM Attributes for LatticeECP2/M

Attribute	Description	Values	Default Value	User Selectable Through IPexpress
Port A Address depth	Address Depth Port A	16K, 8K, 4K, 2K, 1K		YES
Port A Data Width	Data Word Width Port A	1, 2, 4, 9, 18	1	YES
Port B Address depth	Address Depth Port B	16K, 8K, 4K, 2K, 1K		YES
Port B Data Width	Data Word Width Port B	1, 2, 4, 9, 18	1	YES
Port A Enable Output Registers	Register Mode (Pipelining) for Port A	NOREG, OUTREG	NOREG	YES
Port B Enable Output Registers	Register Mode (Pipelining) for Port B	NOREG, OUTREG	NOREG	YES
Enable GSR	Enables Global Set Reset	ENABLE, DISABLE	ENABLE	YES
Reset Mode	Selects the Reset type	ASYNC, SYNC	ASYNC	YES
Memory File Format		BINARY, HEX, ADDRESSED HEX		YES
Port A Write Mode	Read / Write Mode for Port A	NORMAL, WRITE-THROUGH	NORMAL	YES
Port B Write Mode	Read / Write Mode for Port B	NORMAL, WRITE-THROUGH	NORMAL	YES
Chip Select Decode for Port A	Chip Select Decode for Port A	0b000, 0b001, 0b010, 0b011, 0b100, 0b101, 0b110, 0b111	0b000	NO
Chip Select Decode for Port B	Chip Select Decode for Port B	0b000, 0b001, 0b010, 0b011, 0b100, 0b101, 0b110, 0b111	0b000	NO
Init Value	Initialization value	0x0000000000000000000000000000000000000000000000000000000000000000000000000000000000......0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	0x000000000000000000000000000000000000000000000000000000000000000000000000000000000000000	NO

The True Dual Port RAM (RAM_DP_TRUE) can be configured as NORMAL or WRITE THROUGH modes. Each of these modes affects what data comes out of the port Q of the memory during the write operation followed by the read operation at the same memory location. The detailed discussions of the WRITE modes and the constraints of the True Dual Port can be found in Appendix A.

Additionally, users can select to enable the output registers for RAM_DP_TRUE. Figures 11-11 through 11-14 show the internal timing waveforms for the True Dual Port RAM (RAM_DP_TRUE) with these options.

It is important that no setup and hold time violations occur on the address registers (AddressA and AddressB). Failing to meet these requirements can result in corruption of memory contents. This applies to both read and write operations.

A Post Place and Route timing report in Lattice Diamond or ispLEVER design software can be run to verify that no such timing errors occur. Refer to the timing preferences in the Online Help documents.

Figure 11-11. True Dual Port RAM Timing Waveform – NORMAL Mode, without Output Registers

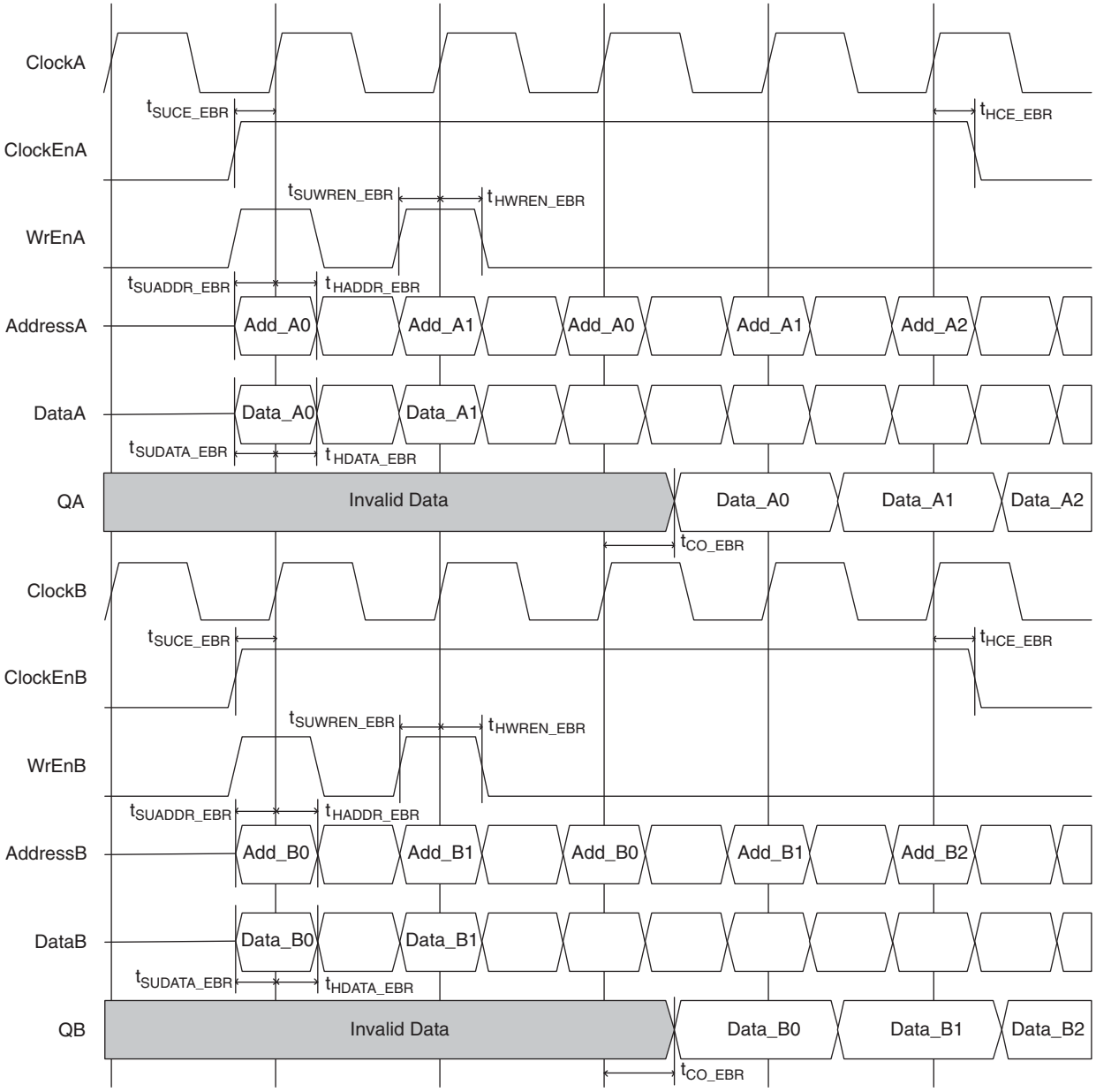


Figure 11-12. True Dual Port RAM Timing Waveform – NORMAL Mode with Output Registers

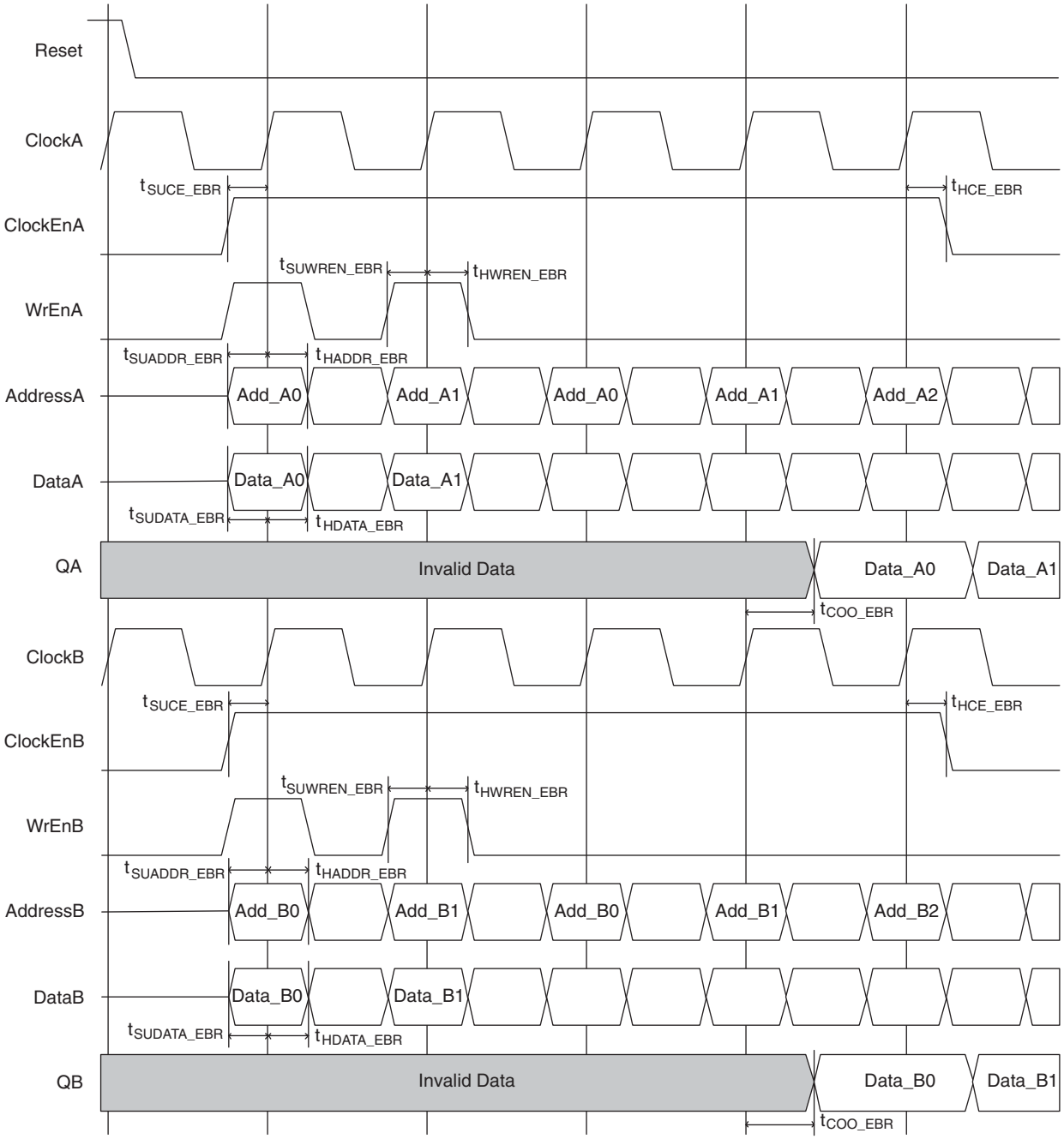


Figure 11-13. True Dual Port RAM Timing Waveform – WRITE THROUGH Mode, without Output Registers

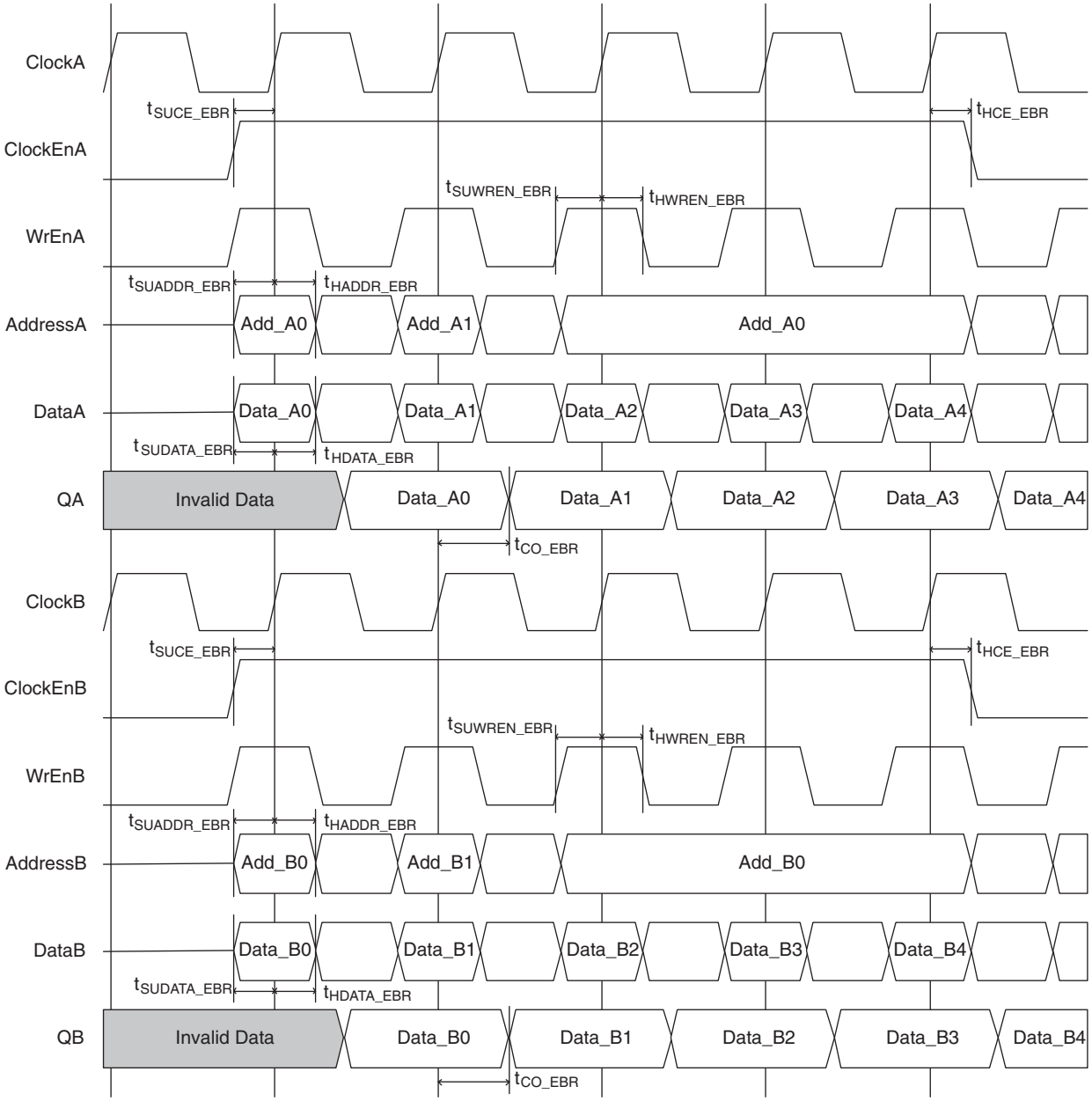
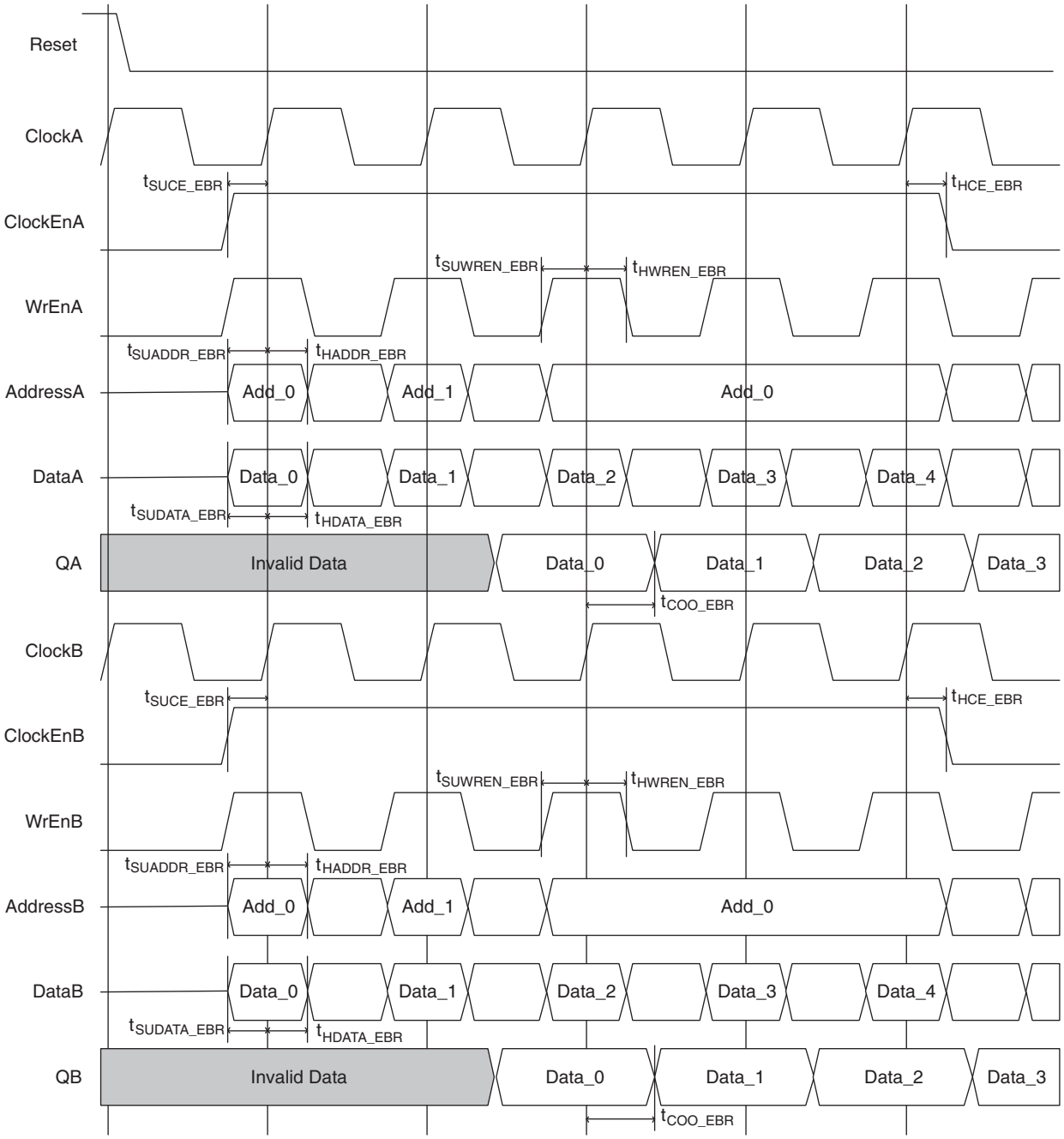


Figure 11-14. True Dual Port RAM Timing Waveform – WRITE THROUGH Mode, with Output Registers

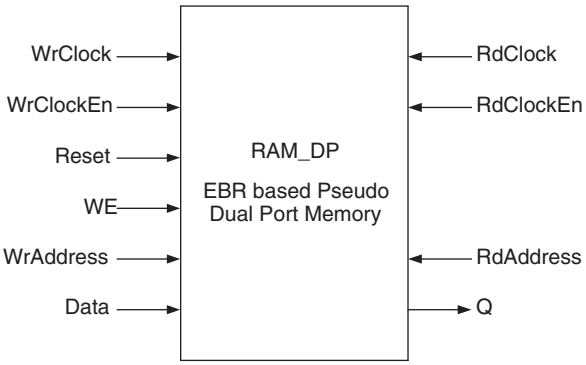


Pseudo Dual Port RAM (RAM_DP) – EBR Based

The EBR blocks in LatticeECP2/M devices can be configured as Pseudo-Dual Port RAM or RAM_DP. IPexpress allows users to generate the Verilog-HDL or VHDL along with EDIF netlists for the memory size as per design requirements.

IPexpress generates the memory module as shown in Figure 11-15.

Figure 11-15. Pseudo Dual Port Memory Module Generated by IPexpress



The generated module makes use of these EBR blocks or primitives. For memory sizes smaller than an EBR block, the module will be created in one EBR block. If the specified memory is larger than one EBR block, multiple EBR blocks can be cascaded in depth or width (as required to create these sizes).

In Pseudo Dual Port RAM mode, the input data and address for the ports are registered at the input of the memory array. The output data of the memory is optionally registered at the output.

The various ports and their definitions for the Single Port Memory are listed in Table 11-8. The table lists the corresponding ports for the module generated by IPexpress and for the EBR RAM_DP primitive.

Table 11-8. EBR-based Pseudo-Dual Port Memory Port Definitions

Port Name in Generated Module	Port Name in the EBR Block Primitive	Description	Active State
RdAddress	ADR[x1:0]	Read Address	—
WrAddress	ADW[x2:0]	Write Address	—
RdClock	CLKR	Read Clock	Rising Clock Edge
WrClock	CLKW	Write Clock	Rising Clock Edge
RdClockEn	CER	Read Clock Enable	Active High
WrClockEn	CEW	Write Clock Enable	Active High
Q	DO[y1:0]	Read Data	—
Data	DI[y2:0]	Write Data	—
WE	WE	Write Enable	Active High
Reset	RST	Reset	Active High
—	CS[2:0]	Chip Select	—

Reset (RST) resets only the input and output registers of the RAM. It does not reset the contents of the memory.

Chip Select (CS) is a useful port when multiple cascaded EBR blocks are required by the memory. The CS signal forms the MSB for the address when multiple EBR blocks are cascaded. Since CS is a 3-bit bus, it can cascade eight memories easily. However, if the memory size specified by the user requires more than eight EBR blocks, the ispLEVER software automatically generates the additional address decoding logic, which is implemented in the PFU external to the EBR blocks.

Each EBR block consists of 18,432 bits of RAM. The values for x's (for address) and y's (data) for each EBR block for the devices are as in Table 11-9.

Table 11-9. Pseudo-Dual Port Memory Sizes for 16K Memory for LatticeECP2/M

Pseudo-Dual Port Memory Size	Input Data Port A	Input Data Port B	Output Data Port A	Output Data Port B	Read Address Port A [MSB:LSB]	Write Address Port B [MSB:LSB]
16K x 1	DIA	DIB	DOA	DOB	RAD[13:0]	WAD[13:0]
8K x 2	DIA[1:0]	DIB[1:0]	DOA[1:0]	DOB[1:0]	RAD[12:0]	WAD[12:0]
4K x 4	DIA[3:0]	DIB[3:0]	DOA[3:0]	DOB[3:0]	RAD[11:0]	WAD[11:0]
2K x 9	DIA[8:0]	DIB[8:0]	DOA[8:0]	DOB[8:0]	RAD[10:0]	WAD[10:0]
1K x 18	DIA[17:0]	DIB[17:0]	DOA[17:0]	DOB[17:0]	RAD[9:0]	WAD[9:0]
512 x 36	DIA[35:0]	DIB[35:0]	DOA[35:0]	DOB[35:0]	RAD[8:0]	WAD[8:0]

Table 11-10 shows the various attributes available for the Pseudo-Dual Port Memory (RAM_DP). Some of these attributes are user-selectable through the IPexpress GUI. For detailed attribute definitions, refer to Appendix A.

Table 11-10. Pseudo-Dual Port RAM Attributes for LatticeECP2/M

Attribute	Description	Values	Default Value	User Selectable Through IPexpress
Read Port Address Depth	Address Depth Read Port	16K, 8K, 4K, 2K, 1K, 512		YES
Read Port Data Width	Data Word Width Read Port	1, 2, 4, 9, 18, 36	1	YES
Write Port Address Depth	Address Depth Write Port	16K, 8K, 4K, 2K, 1K		YES
Write Port Data Width	Data Word Width Write Port	1, 2, 4, 9, 18, 36	1	YES
Write Port Enable Output Registers	Register Mode (Pipelining) for Write Port	NOREG, OUTREG	NOREG	YES
Enable GSR	Enables Global Set Reset	ENABLE, DISABLE	ENABLE	YES
Reset Mode	Selects the Reset type	ASYNC, SYNC	ASYNC	YES
Memory File Format		BINARY, HEX, ADDRESSED HEX		YES
Read Port Write Mode	Read / Write Mode for Read Port	NORMAL	NORMAL	YES
Write Port Write Mode	Read / Write Mode for Write Port	NORMAL	NORMAL	YES
Chip Select Decode for Read Port	Chip Select Decode for Read Port	0b000, 0b001, 0b010, 0b011, 0b100, 0b101, 0b110, 0b111	0b000	NO
Chip Select Decode for Write Port	Chip Select Decode for Write Port	0b000, 0b001, 0b010, 0b011, 0b100, 0b101, 0b110, 0b111	0b000	NO
Init Value	Initialization value	0x0000000000000000000000000 000000000000000000000000000 000000000000000000000000000 0......0xFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFF	0x000000000000 00000000000000 00000000000000 00000000000000 00000000000000 000000000000	NO

Users have the option to enable the output registers for Pseudo-Dual Port RAM (RAM_DP). Figures 11-16 and 11-17 show the internal timing waveforms for Pseudo-Dual Port RAM (RAM_DP) with these options. It is important that no setup and hold time violations occur on the address registers (RdAddress and WrAddress). Failing to meet these requirements can result in corruption of memory contents. This applies to both read and write operations.

A Post Place and Route timing report in Lattice Diamond or ispLEVER design software can be run to verify that no such timing errors occur. Refer to the timing preferences in the Online Help documents.

Figure 11-16. PSEUDO DUAL PORT RAM Timing Diagram – without Output Registers

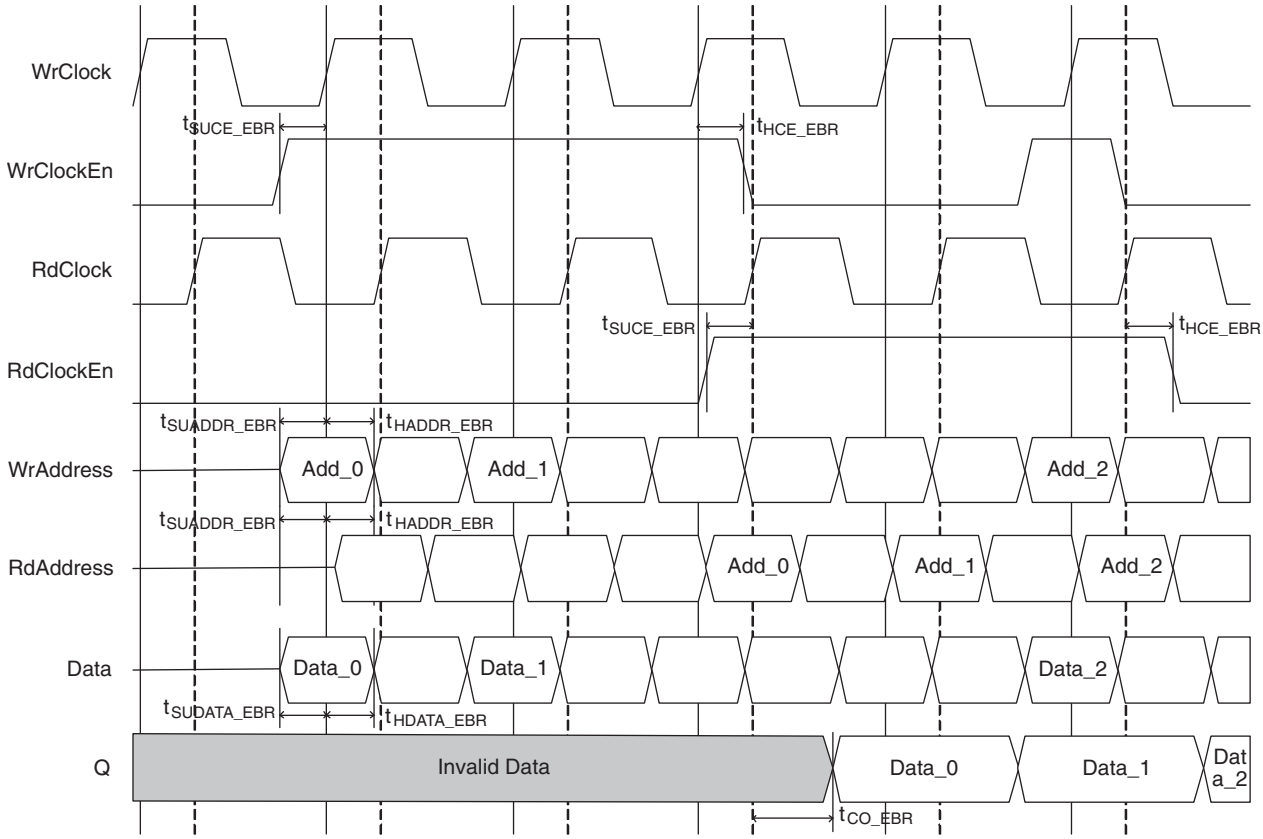
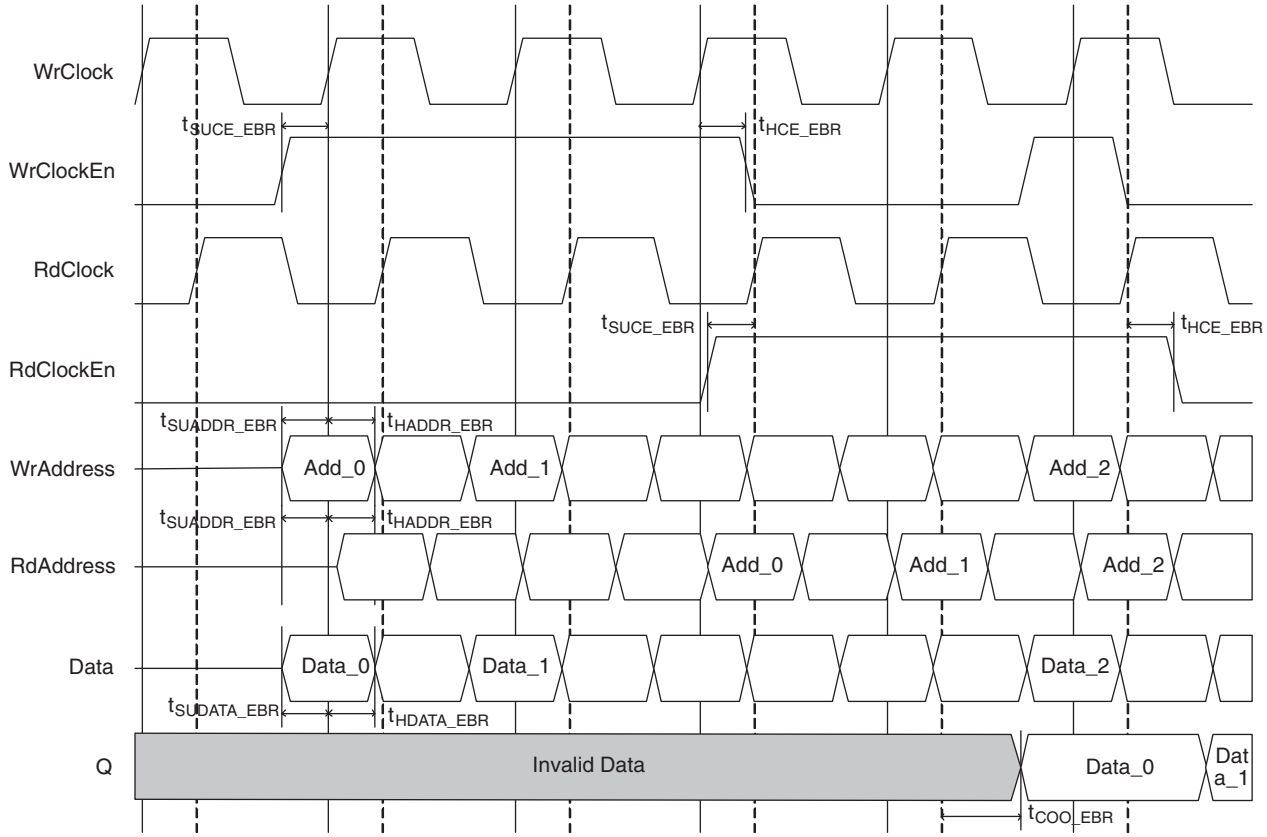


Figure 11-17. PSEUDO DUAL PORT RAM Timing Diagram – with Output Registers

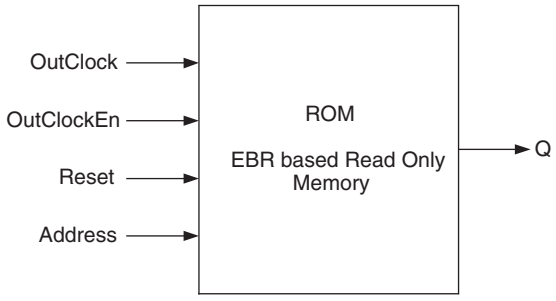


Read Only Memory (ROM) – EBR Based

The EBR blocks in the LatticeECP2/M devices can be configured as Read Only Memory or ROM. IPexpress allows users to generate the Verilog-HDL or VHDL and the EDIF netlist for the memory size, as per design requirements. Users are required to provide the ROM memory content in the form of an initialization file.

IPexpress generates the memory module as shown in Figure 11-18.

Figure 11-18. Read-Only Memory Module Generated by IPexpress



The generated module makes use of these EBR blocks or primitives. For memory sizes smaller than an EBR block, the module will be created in one EBR block. If the specified memory is larger than one EBR block, multiple EBR blocks can be cascaded, in depth or width (as required to create these sizes).

In ROM mode, the address for the port is registered at the input of the memory array. The output data of the memory is optionally registered at the output.

The various ports and their definitions for the ROM are listed in Table 11-11. The table lists the corresponding ports for the module generated by IPexpress and for the ROM primitive.

Table 11-11. EBR-based ROM Port Definitions

Port Name in Generated Module	Port Name in the EBR block Primitive	Description	Active State
Address	AD[x:0]	Read Address	—
OutClock	CLK	Clock	Rising Clock Edge
OutClockEn	CE	Clock Enable	Active High
Reset	RST	Reset	Active High
—	CS[2:0]	Chip Select	—

Reset (RST) resets only the input and output registers of the RAM. It does not reset the contents of the memory.

Chip Select (CS) is a useful port when multiple cascaded EBR blocks are required by the memory. The CS signal forms the MSB for the address when multiple EBR blocks are cascaded. Since CS is a 3-bit bus, it can cascade eight memories easily. However, if the memory size specified by the user requires more than eight EBR blocks, the ispLEVER software automatically generates the additional address decoding logic, which is implemented in the PFU external to the EBR blocks.

While generating the ROM using IPexpress, the user must provide the initialization file to pre-initialize the contents of the ROM. These files are the *.mem files and they can be of Binary, Hex or the Addressed Hex formats. The initialization files are discussed in detail in the Initializing Memory section of this document.

Users have the option of enabling the output registers for Read Only Memory (ROM). Figures 11-19 and 11-20 show the internal timing waveforms for the Read Only Memory (ROM) with these options.

Each EBR block consists of 18,432 bits of RAM. The values for x's (for address) and y's (data) for each EBR block for the devices are as per Table 11-12.

Table 11-12. ROM Memory Sizes for 16K Memory for LatticeECP2/M

ROM	Output Data	Address Port [MSB:LSB]
16K x 1	DOA	WAD[13:0]
8K x 2	DOA[1:0]	WAD[12:0]
4K x 4	DOA[3:0]	WAD[11:0]
2K x 9	DOA[8:0]	WAD[10:0]
1K x 18	DOA[17:0]	WAD[9:0]
512 x 36	DOA[35:0]	WAD[8:0]

Table 11-13 shows the various attributes available for the Read Only Memory (ROM). Some of these attributes are user-selectable through the IPexpress GUI. For detailed attribute definitions, refer to Appendix A.

Users have the option to enable the output registers for Read Only Memory (ROM). Figures 11-19 and 11-20 show the internal timing waveforms for ROM with these options.

It is important that no setup and hold time violations occur on the address registers (Address). Failing to meet these requirements can result in corruption of memory contents. This applies to both read operations in this case.

A Post Place and Route timing report in Lattice Diamond or ispLEVER design software can be run to verify that no such timing errors occur. Refer to the timing preferences in the Online Help documents.

Table 11-13. ROM Attributes for LatticeECP2/M

Attribute	Description	Values	Default Value	User Selectable Through IPexpress
Address depth	Address Depth Read Port	16K, 8K, 4K, 2K, 1K, 512		YES
Data Width	Data Word Width Read Port	1, 2, 4, 9, 18, 36	1	YES
Enable Output Registers	Register Mode (Pipelining) for Write Port	NOREG, OUTREG	NOREG	YES
Enable GSR	Enables Global Set Reset	ENABLE, DISABLE	ENABLE	YES
Reset Mode	Selects the Reset type	ASYNC, SYNC	ASYNC	YES
Memory File Format		BINARY, HEX, ADDRESSED HEX		YES
Chip Select Decode	Chip Select Decode for Read Port	0b000, 0b001, 0b010, 0b011, 0b100, 0b101, 0b110, 0b111	0b000	NO

Figure 11-19. ROM Timing Waveform – without Output Registers

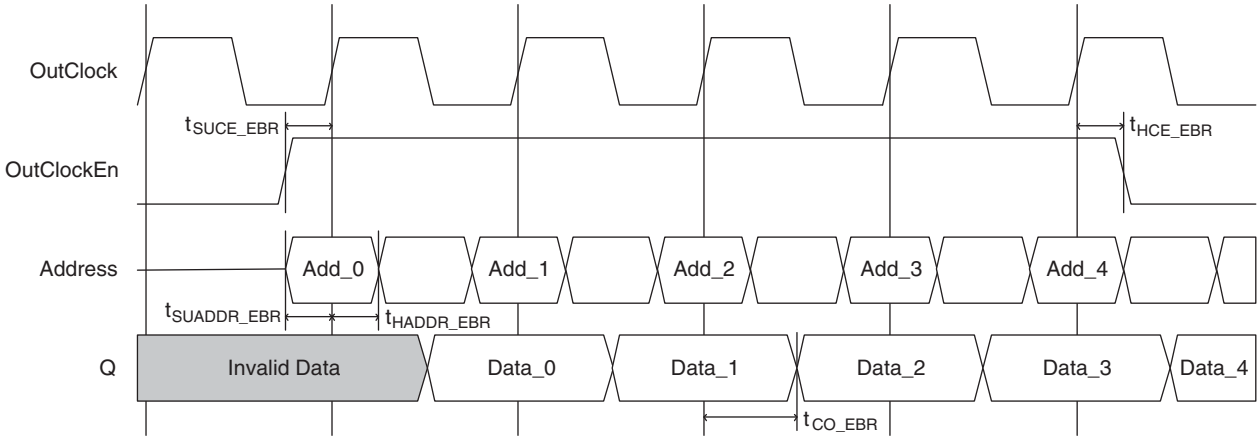
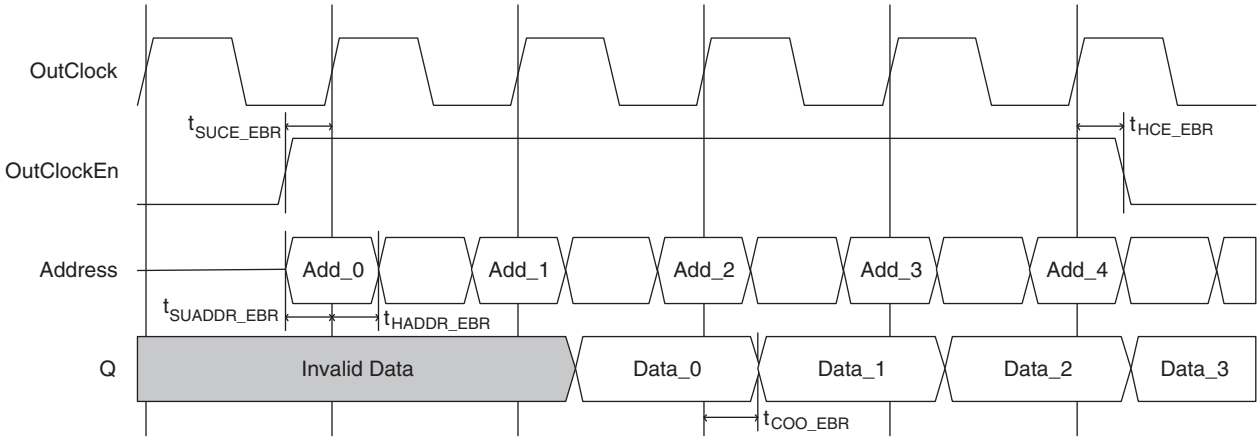


Figure 11-20. ROM Timing Waveform – with Output Registers



First In First Out (FIFO, FIFO_DC) – EBR Based

The hardware has Embedded Block RAM (EBR) which can be configured in Single Port (RAM_DQ), Pseudo-Dual Port (RAM_DP) and True Dual Port (RAM_DP_TRUE) RAMs. The FIFOs in these devices can be built using these RAMs. The IPexpress point tool in the ispLEVER design software allows users to build a FIFO and FIFO_DC around Pseudo Dual Port RAM (or DP_RAM).

Each of these FIFOs can be configured with (pipelined) and without (non-pipelined) output registers. In the pipelined mode users have an extra option to enable the output registers by the RdEn signal. We will discuss the operation in the following sections.

Let us take a look at the operation of these FIFOs.

First In First Out (FIFO) Memory

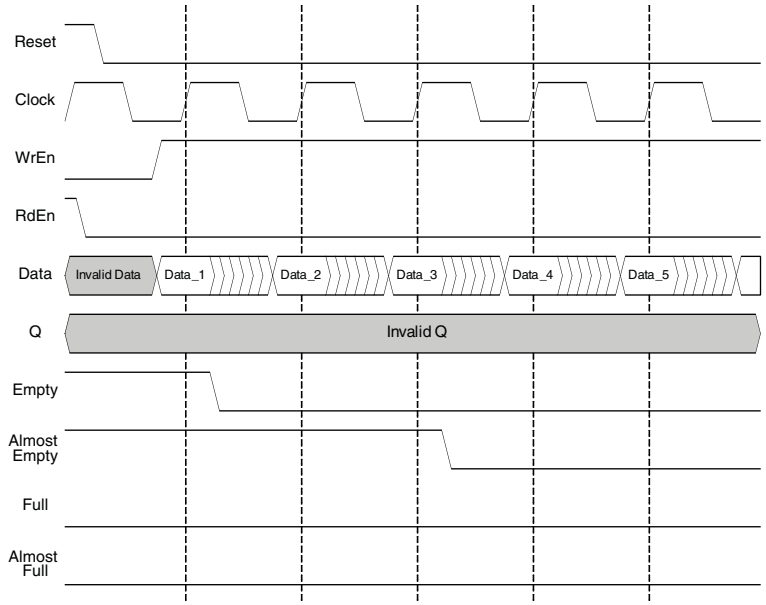
The FIFO, or the single clock FIFO, is an emulated FIFO. The address logic and the flag logic is implemented in the FPGA fabric around the RAM.

The ports available on the FIFO are:

- Reset
- Clock
- WrEn
- RdEn
- Data
- Q
- Full Flag
- Almost Full Flag
- Empty Flag
- Almost Empty Flag

Let us first discuss the non-pipelined or the FIFO without output registers. Figure 11-21 shows the operation of the FIFO when it is empty and the data starts to get written into it.

Figure 11-21. FIFO without Output Registers, Start of Data Write Cycle

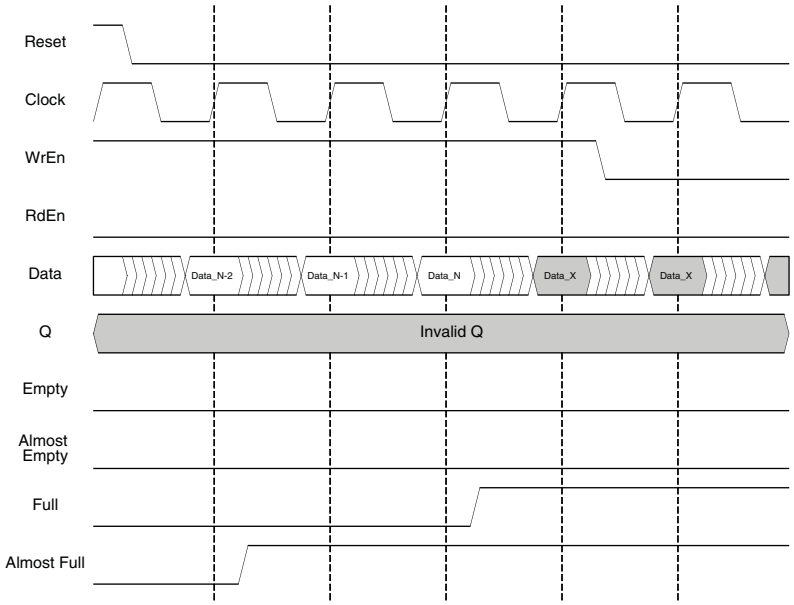


The WrEn signal must be high to start writing into the FIFO. The Empty and Almost Empty flags are high to begin and Full and Almost Full are low.

When the first data is written into the FIFO, the Empty flag de-asserts (or goes low), as the FIFO is no longer empty. In this figure we assume that the Almost Empty setting flag setting is 3 (address location 3). So the Almost Empty flag gets de-asserted when the third address location is filled.

Now let us assume that we continue to write into the FIFO to fill it. When the FIFO is filled, the Almost Full and Full Flags are asserted. Figure 11-22 shows the behavior of these flags. In this figure we assume that FIFO depth is 'N'.

Figure 11-22. FIFO without Output Registers, End of Data Write Cycle

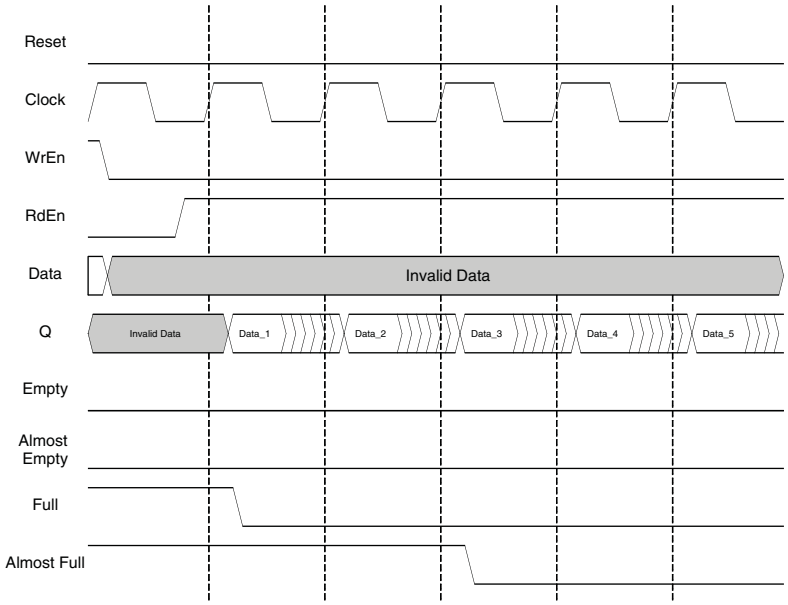


In this case, the Almost Full flag is in the 2 location before the FIFO is filled. The Almost Full flag is asserted when the N-2 location is written, and the Full flag is asserted when the last word is written into the FIFO.

Data_X data inputs do not get written as the FIFO is full (the Full flag is high).

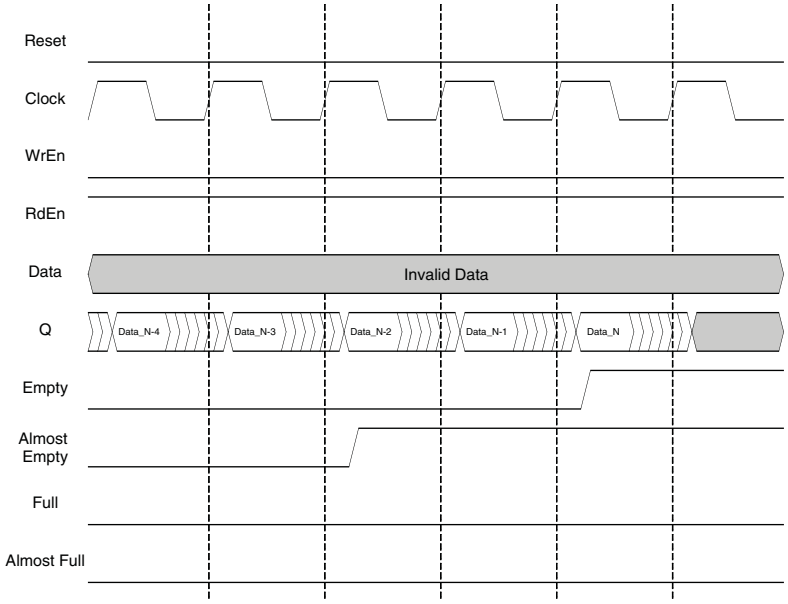
Now let us look at the waveforms when the contents of the FIFO are read out. Figure 11-23 shows the start of the read cycle. RdEn goes high and the data read starts. The Full and Almost Full flags are de-asserted, as shown.

Figure 11-23. FIFO without Output Registers, Start of Data Read Cycle



Similarly, as the data is read out and FIFO is emptied, the Almost Empty and Empty flags are asserted.

Figure 11-24. FIFO without Output Registers, End of Data Read Cycle



Figures 11-21 to 11-24 show the behavior of non-pipelined FIFO or FIFO without output registers. When we pipeline the registers, the output data is delayed by one clock cycle. There is also the extra option for Output registers to be enabled by the RdEn signal.

Figures 11-25 to 11-28 show the similar waveforms for the FIFO with output register and with output register enable with RdEn. It should be noted that flags are asserted and de-asserted with similar timing to the FIFO without output registers. However, it is only the data out 'Q' that is delayed by one clock cycle.

Figure 11-25. FIFO with Output Registers, Start of Data Write Cycle

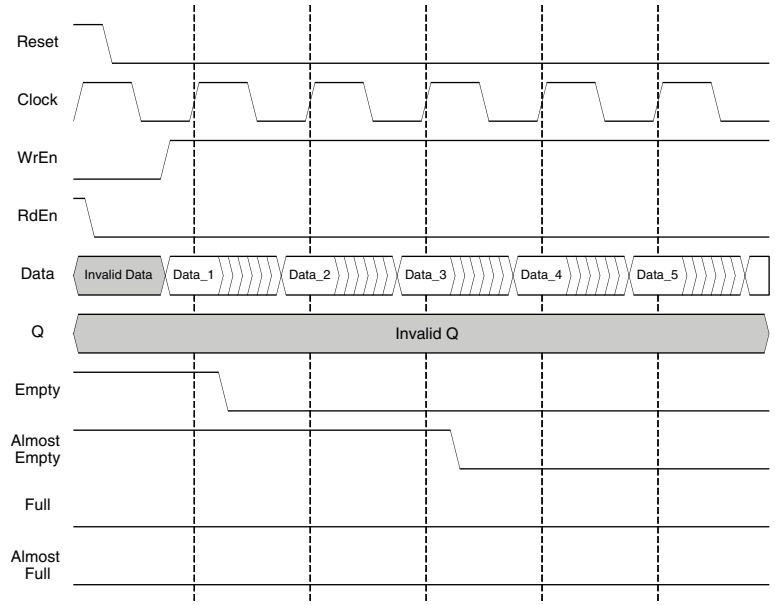


Figure 11-26. FIFO with Output Registers, End of Data Write Cycle

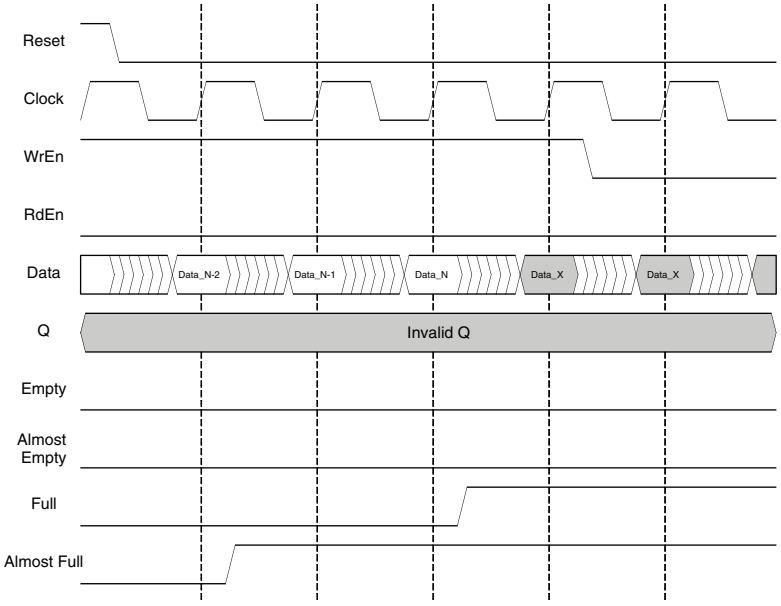


Figure 11-27. FIFO with Output Registers, Start of Data Read Cycle

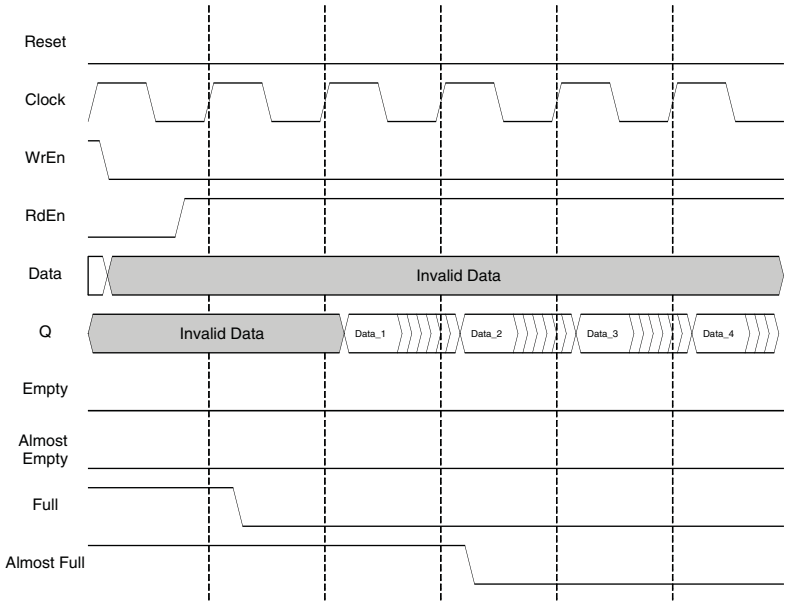
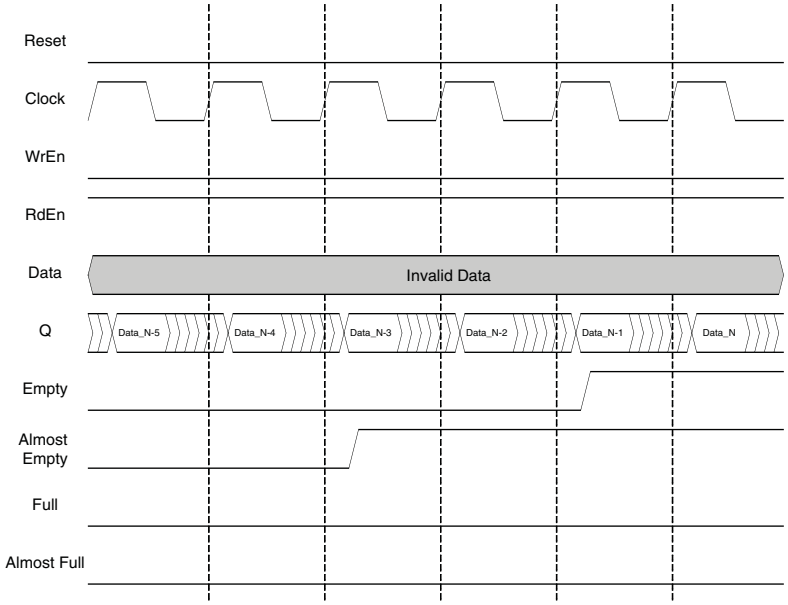
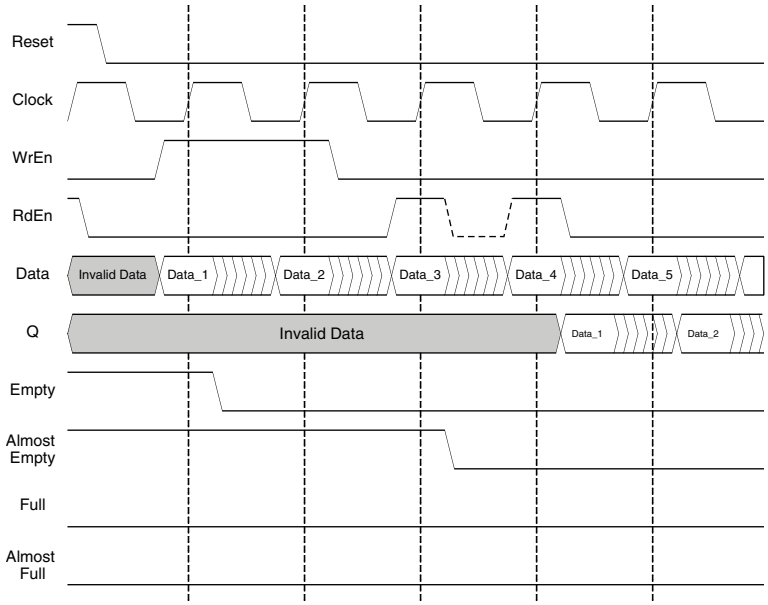


Figure 11-28. FIFO with Output Registers, End of Data Read Cycle



And finally, if you select the option enable output register with RdEn, it still delays the data out by one clock cycle (as compared to the non-pipelined FIFO). The RdEn should also be high during that clock cycle, otherwise the data takes an extra clock cycle when the RdEn goes true.

Figure 11-29. FIFO with Output Registers and RdEn on Output Registers



Dual Clock First In First Out (FIFO_DC) Memory:
The FIFO_DC or the dual clock FIFO is also an emulated FIFO. Again, the address logic and the flag logic is implemented in the FPGA fabric around the RAM.

The ports available on the FIFO_DC are:

- Reset
- RPReset
- WrClock
- RdClock
- WrEn
- RdEn
- Data
- Q
- Full Flag
- Almost Full Flag
- Empty Flag
- Almost Empty Flag

FIFO_DC Flags
The FIFO_DC, as an emulated FIFO, required the flags to be implemented in the FPGA logic around the block RAM. Because of the two clocks, the flags were required to change clock domains from read clock to write clock and vice versa. This adds latency to the flags either during assertion or de-assertion. The latency can be avoided only in one of the cases (either assertion or de-assertion) or distributed among these cases.

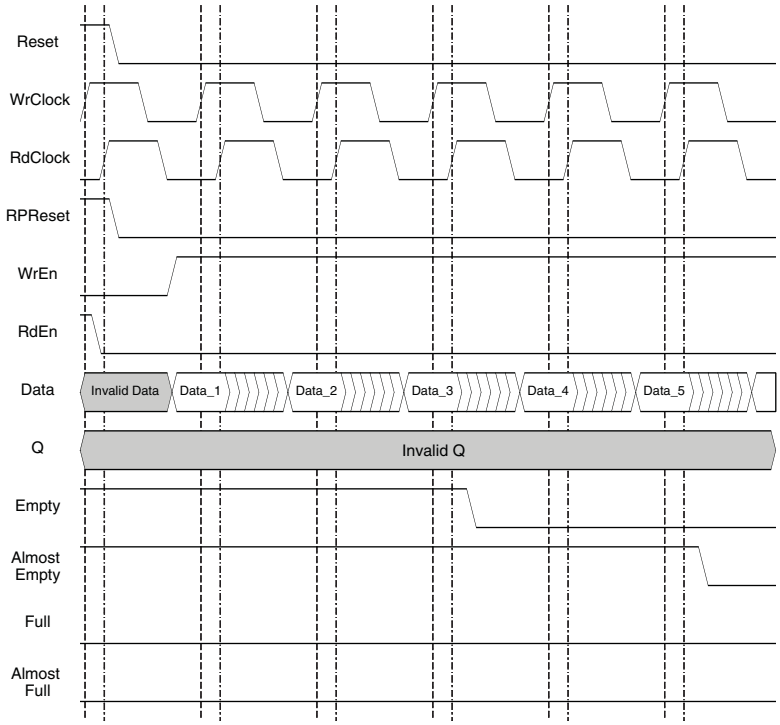
In the current emulated FIFO_DC, there is no latency during assertion of these flags which we feel is more important. Thus, when these flags are required to go true, there is no latency. However, due to the design of the flag logic running on two clock domains, there is latency during the de-assertion.

Let us assume that we start to write into the FIFO_DC to fill it. The write operation is controlled by WrClock and WrEn, however it takes extra RdClock cycles for de-assertion of the Empty and Almost Empty flags.

On the other hand, de-assertion of Full and Almost Full result in the reading out of the data from the FIFO_DC. It takes extra WrClock cycles, after reading this data, for the flags to come out.

With this in mind, let us look at the FIFO_DC without output registers waveforms. Figure 11-30 shows the operation of the FIFO_DC when it is empty and the data starts to be written into it.

Figure 11-30. FIFO_DC without Output Registers, Start of Data Write Cycle

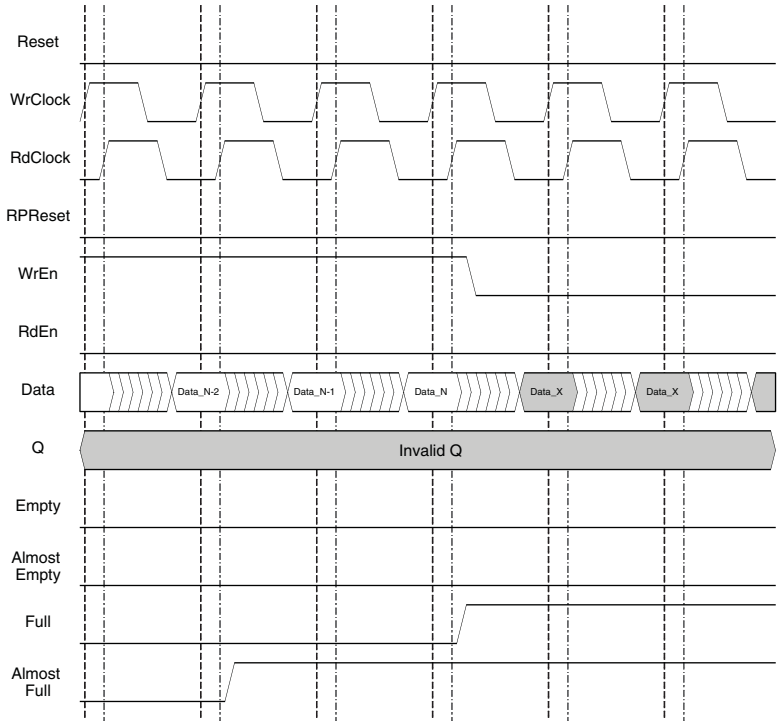


The WrEn signal must be high to start writing into the FIFO_DC. The Empty and Almost Empty flags are high to begin and Full and Almost full are low.

When the first data is written into the FIFO_DC, the Empty flag de-asserts (or goes low), as the FIFO_DC is no longer empty. In this figure we assume that the Almost Empty setting flag setting is 3 (address location 3). So the Almost Empty flag is de-asserted when the third address location is filled.

Now let us assume that we continue to write into the FIFO_DC to fill it. When the FIFO_DC is filled, the Almost Full and Full Flags are asserted. Figure 11-31 shows the behavior of these flags. In this figure the FIFO_DC depth is 'N'.

Figure 11-31. FIFO_DC without Output Registers, End of Data Write Cycle



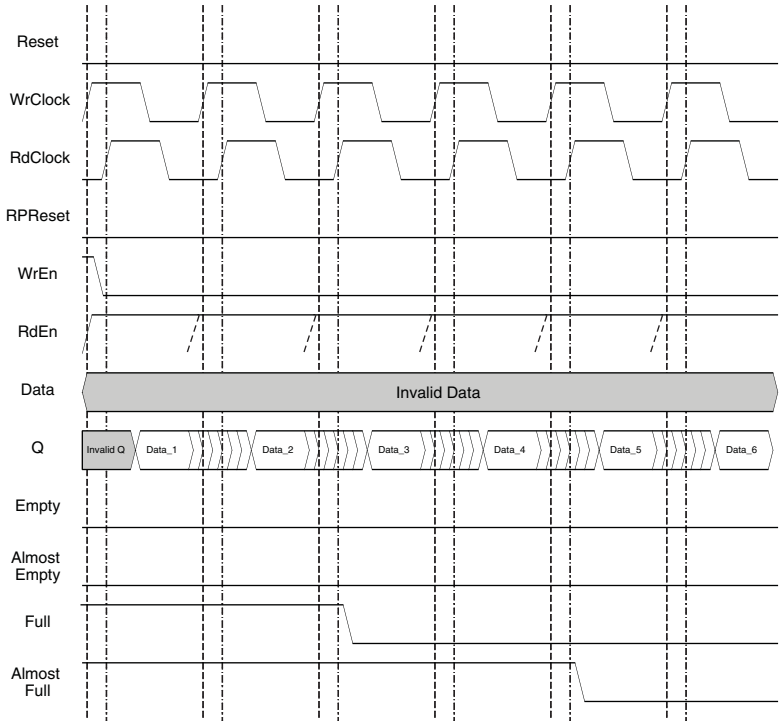
In this case, the Almost Full flag is in the 2 location before the FIFO_DC is filled. The Almost Full flag is asserted when the N-2 location is written, and the Full flag is asserted when the last word is written into the FIFO_DC.

Data_X data inputs do not get written as the FIFO_DC is full (the Full flag is high).

Note that the assertion of these flags is immediate and there is no latency when they go true.

Now let us look at the waveforms when the contents of the FIFO_DC are read out. Figure 11-32 shows the start of the read cycle. RdEn goes high and the data read starts. The Full and Almost Full flags are de-asserted, as shown. In this case, note that the de-assertion is delayed by two clock cycles.

Figure 11-32. FIFO_DC without Output Registers, Start of Data Read Cycle



Similarly, as the data is read out, and FIFO_DC is emptied, the Almost Empty and Empty flags are asserted.

Figure 11-33. FIFO_DC without Output Registers, End of Data Read Cycle

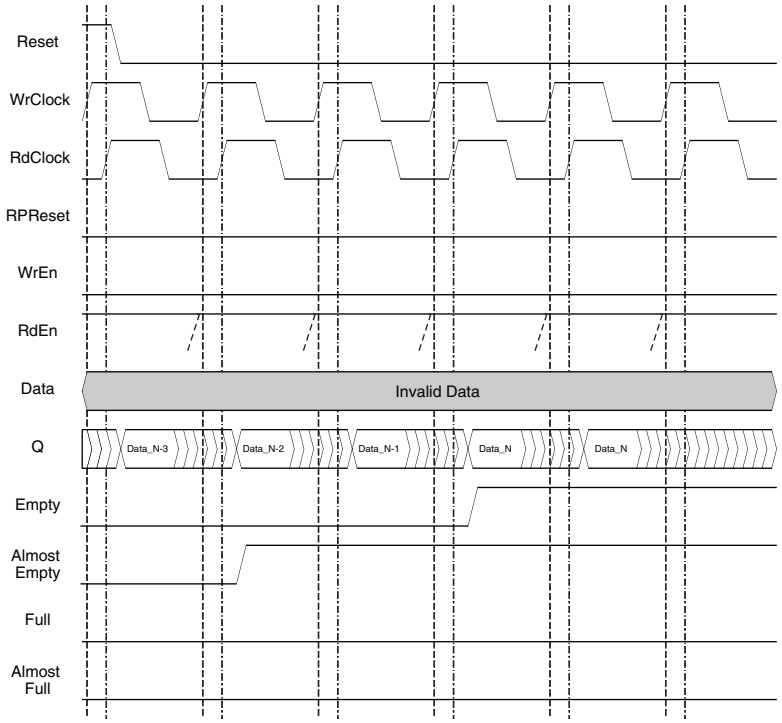


Figure 11-33 show the behavior of non-pipelined FIFO_DC or FIFO_DC without output registers. When we pipeline the registers, the output data is delayed by one clock cycle. There is an extra option for the output registers to be enabled by the RdEn signal.

Figures 11-34 to 11-37 show the similar waveforms for the FIFO_DC with output register and without output register enable with RdEn. Note that flags are asserted and de-asserted with similar timing to the FIFO_DC without output registers. However, it is only the data out 'Q' that is delayed by one clock cycle.

Figure 11-34. FIFO_DC with Output Registers, Start of Data Write Cycle

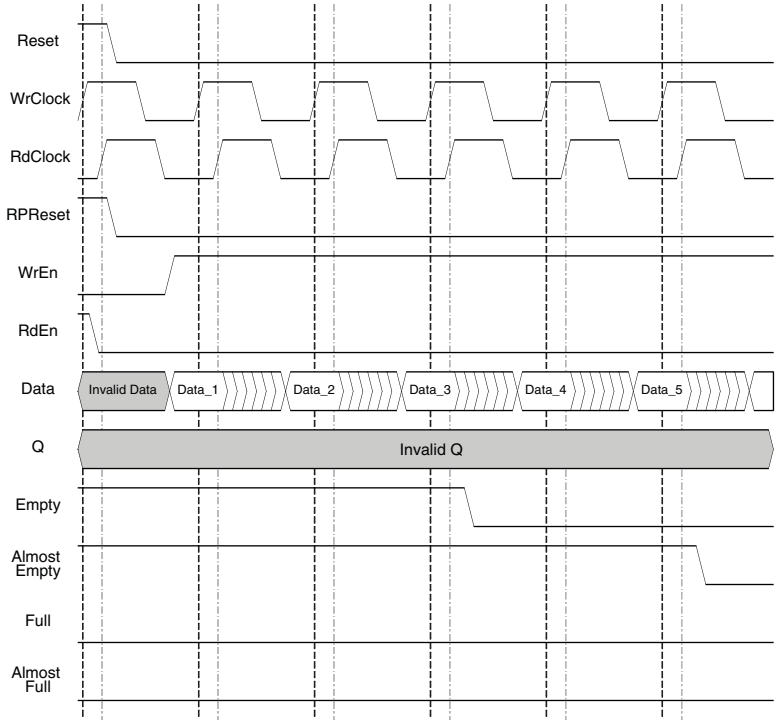


Figure 11-35. FIFO_DC with Output Registers, End of Data Write Cycle

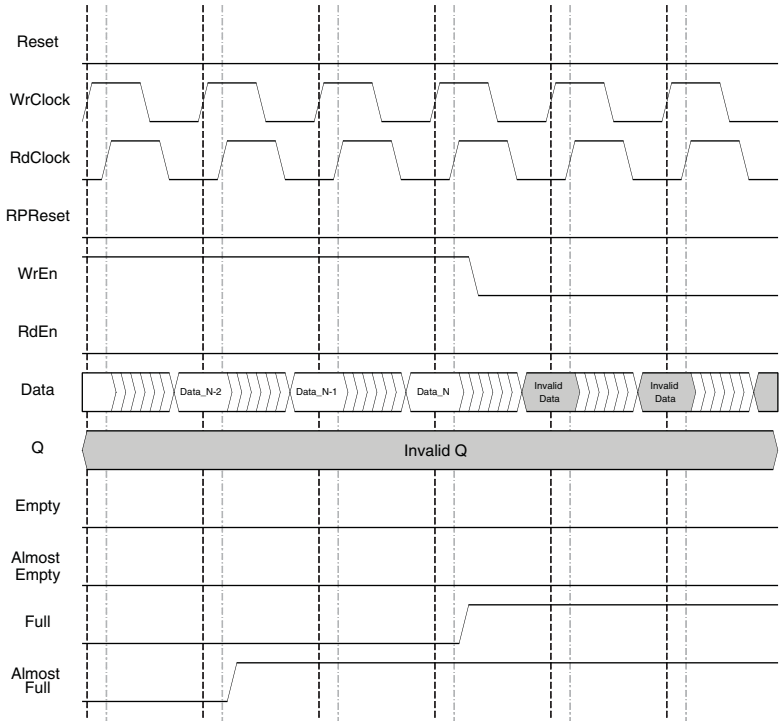


Figure 11-36. FIFO_DC with Output Registers, Start of Data Read Cycle

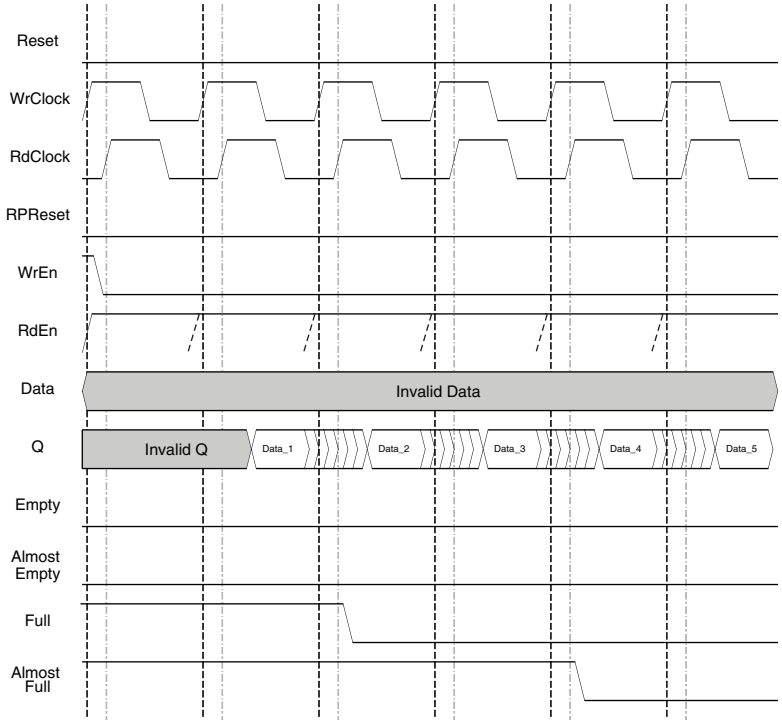
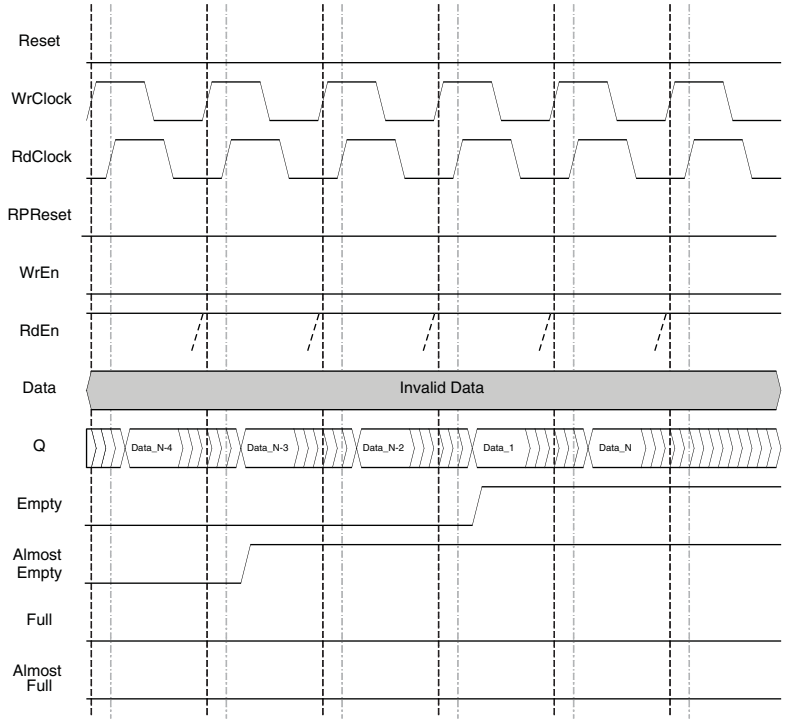
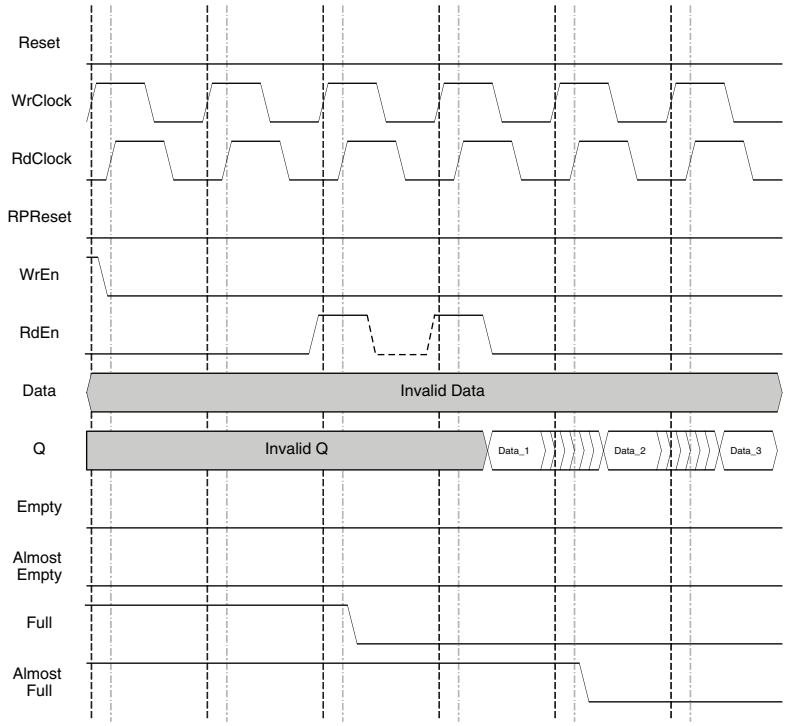


Figure 11-37. FIFO_DC with Output Registers, End of Data Read Cycle



And finally, if you select the option to enable the output register with RdEn, it still delays the data out by one clock cycle (as compared to the non-pipelined FIFO_DC). The RdEn should also be high during that clock cycle, otherwise the data takes an extra clock cycle when the RdEn is goes true.

Figure 11-38. FIFO_DC with Output Registers and RdEn on Output Registers

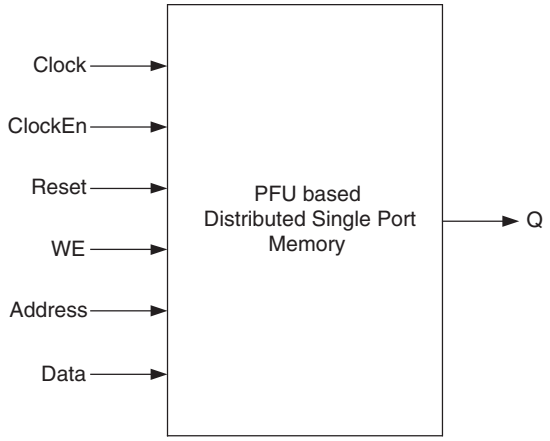


Distributed Single Port RAM (Distributed_SPRAM) – PFU Based

PFU-based Distributed Single Port RAM is created using the 4-input LUT (Look-Up Table) available in the PFU. These LUTs can be cascaded to create larger Distributed Memory sizes.

Figure 11-39 shows the Distributed Single Port RAM module as generated by IPexpress.

Figure 11-39. Distributed Single Port RAM Module Generated by IPexpress



The generated module makes use 4-input LUT available in the PFU. Additional logic like Clock, Reset is generated by utilizing the resources available in the PFU.

Ports such as Read Clock (RdClock) and Read Clock Enable (RdClockEn), are not available in the hardware primitive. These are generated by IPexpress when the user wants the to enable the output registers in their IPexpress configuration.

The various ports and their definitions for the memory are as per Table 11-14. The table lists the corresponding ports for the module generated by IPexpress and for the primitive.

Table 11-14. PFU-based Distributed Single Port RAM Port Definitions

Port Name in Generated Module	Port Name in the PFU Primitive	Description	Active State
Clock	CK	Clock	Rising Clock Edge
ClockEn	—	Clock Enable	Active High
Reset	—	Reset	Active High
WE	WRE	Write Enable	Active High
Address	AD[3:0]	Address	—
Data	DI[1:0]	Data In	—
Q	DO[1:0]	Data Out	—

Ports such as Clock Enable (ClockEn) are not available in the hardware primitive. These are generated by IPexpress when the user wishes to enable the output registers in the IPexpress configuration.

Users have the option of enabling the output registers for Distributed Single Port RAM (Distributed_SPRAM). Figures 11-40 and 11-41 show the internal timing waveforms for the Distributed Single Port RAM (Distributed_SPRAM) with these options.

Figure 11-40. PFU Based Distributed Single Port RAM Timing Waveform – without Output Registers

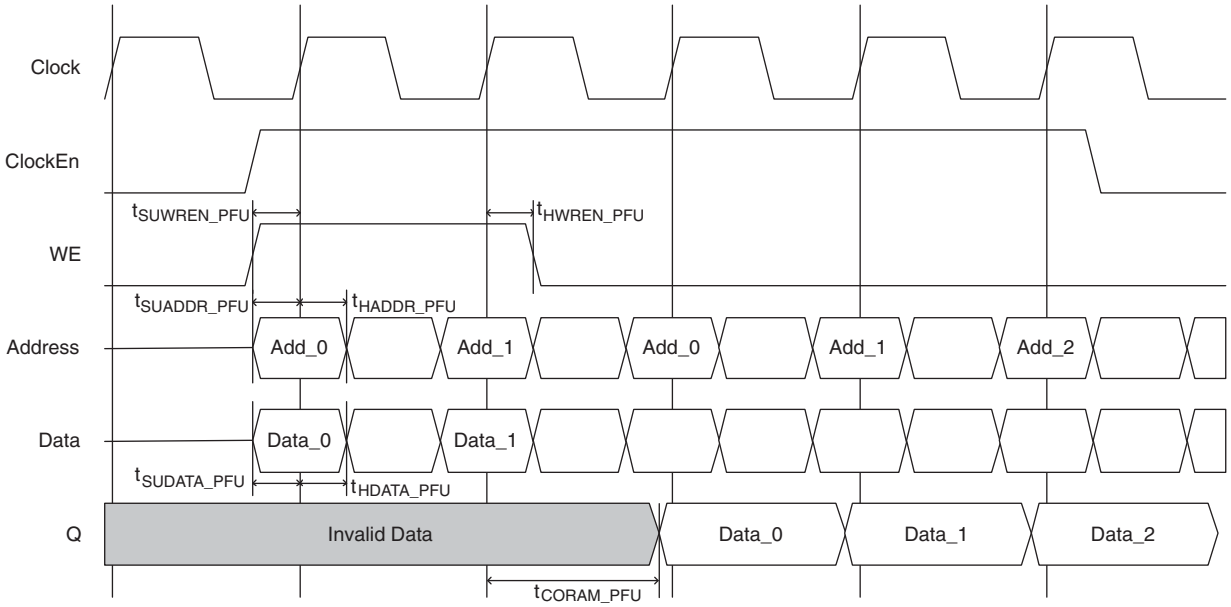
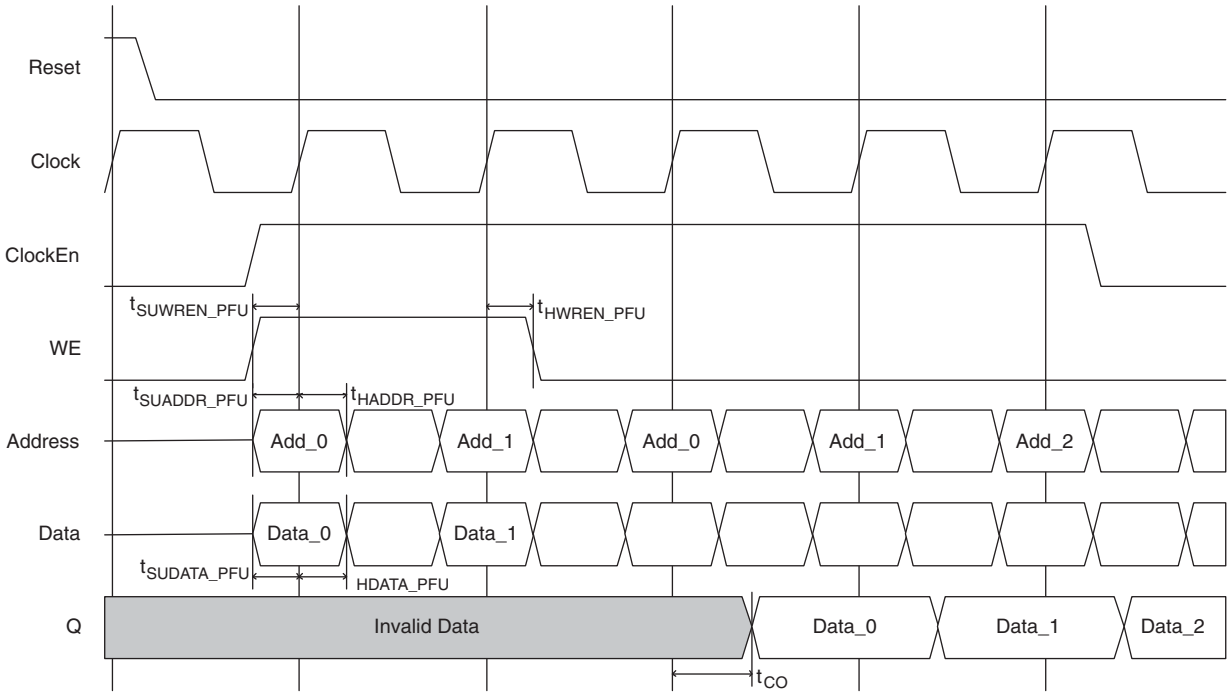


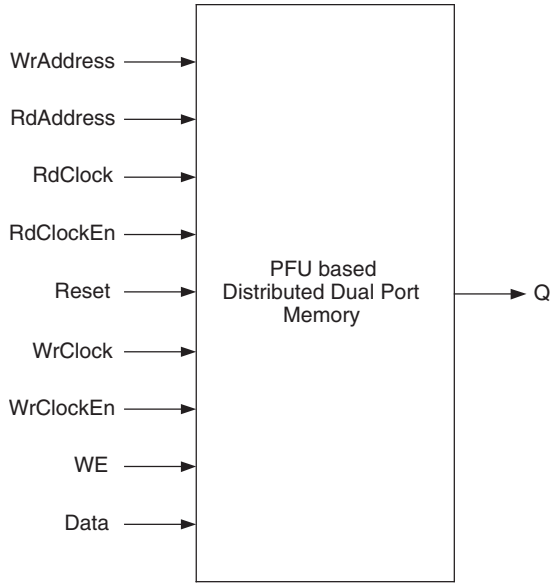
Figure 11-41. PFU Based Distributed Single Port RAM Timing Waveform – with Output Registers



Distributed Dual Port RAM (Distributed_DPRAM) – PFU Based

PFU-based Distributed Dual Port RAM is also created using the 4-input LUT (Look-Up Table) available in the PFU. These LUTs can be cascaded to create a larger Distributed Memory sizes.

Figure 11-42. Distributed Dual Port RAM Module Generated by IPexpress



The generated module makes use of the 4-input LUT available in the PFU. Additional logic like Clock and Reset is generated by utilizing the resources available in the PFU.

Ports such as Read Clock (RdClock) and Read Clock Enable (RdClockEn), are not available in the hardware primitive. These are generated by IPexpress when the user wants the to enable the output registers in the IPexpress configuration.

The various ports and their definitions for memory are as per Table 11-15. The table lists the corresponding ports for the module generated by IPexpress and for the primitive.

Table 11-15. PFU-based Distributed Dual-Port RAM Port Definitions

Port Name in Generated Module	Port Name in the EBR Block Primitive	Description	Active State
WrAddress	WAD[3:0]	Write Address	—
RdAddress	RAD[3:0]	Read Address	—
RdClock	—	Read Clock	Rising Clock Edge
RdClockEn	—	Read Clock Enable	Active High
WrClock	WCK	Write Clock	Rising Clock Edge
WrClockEn	—	Write Clock Enable	Active High
WE	WRE	Write Enable	Active High
Data	DI[1:0]	Data Input	—
Q	RDO[1:0]	Data Out	—

Ports such as Read Clock (RdClock) and Read Clock Enable (RdClockEn) are not available in the hardware primitive. These are generated by IPexpress when the user wants to enable the output registers in the IPexpress configuration.

Users have the option of enabling the output registers for Distributed Dual Port RAM (Distributed_DPRAM). Figures 11-43 and 11-44 show the internal timing waveforms for the Distributed Dual Port RAM (Distributed_DPRAM) with these options.

Figure 11-43. PFU Based Distributed Dual Port RAM Timing Waveform – without Output Registers

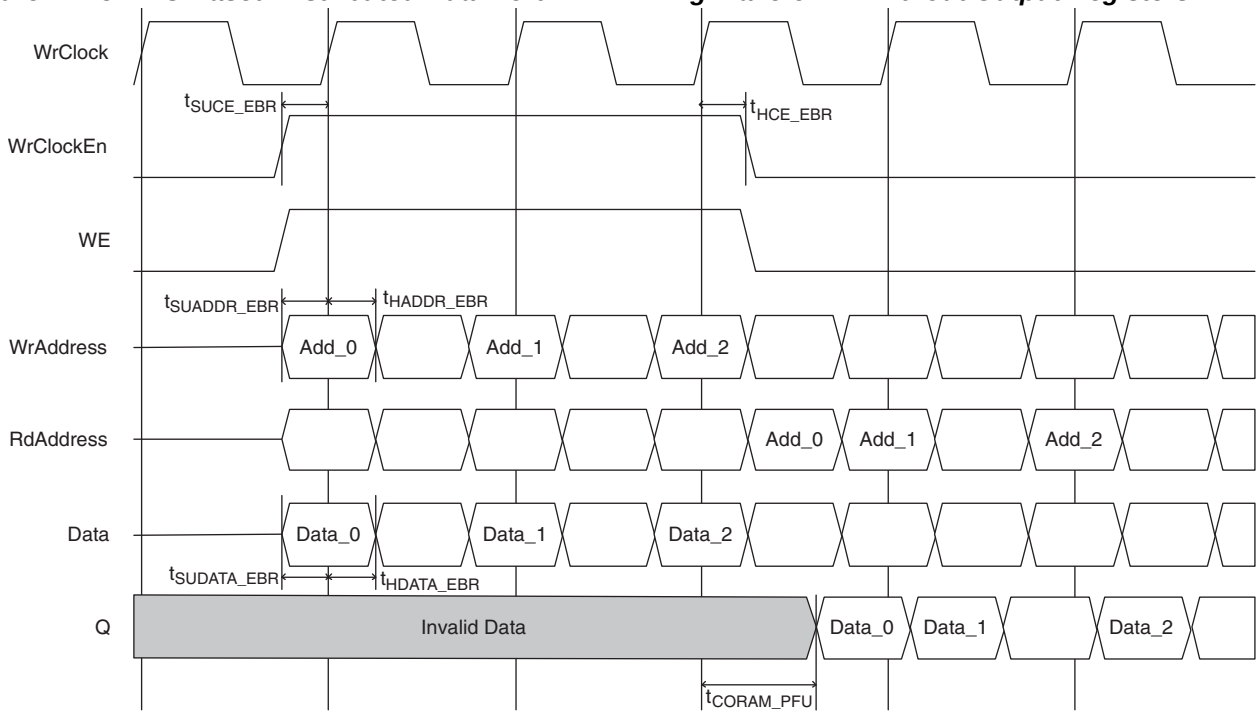
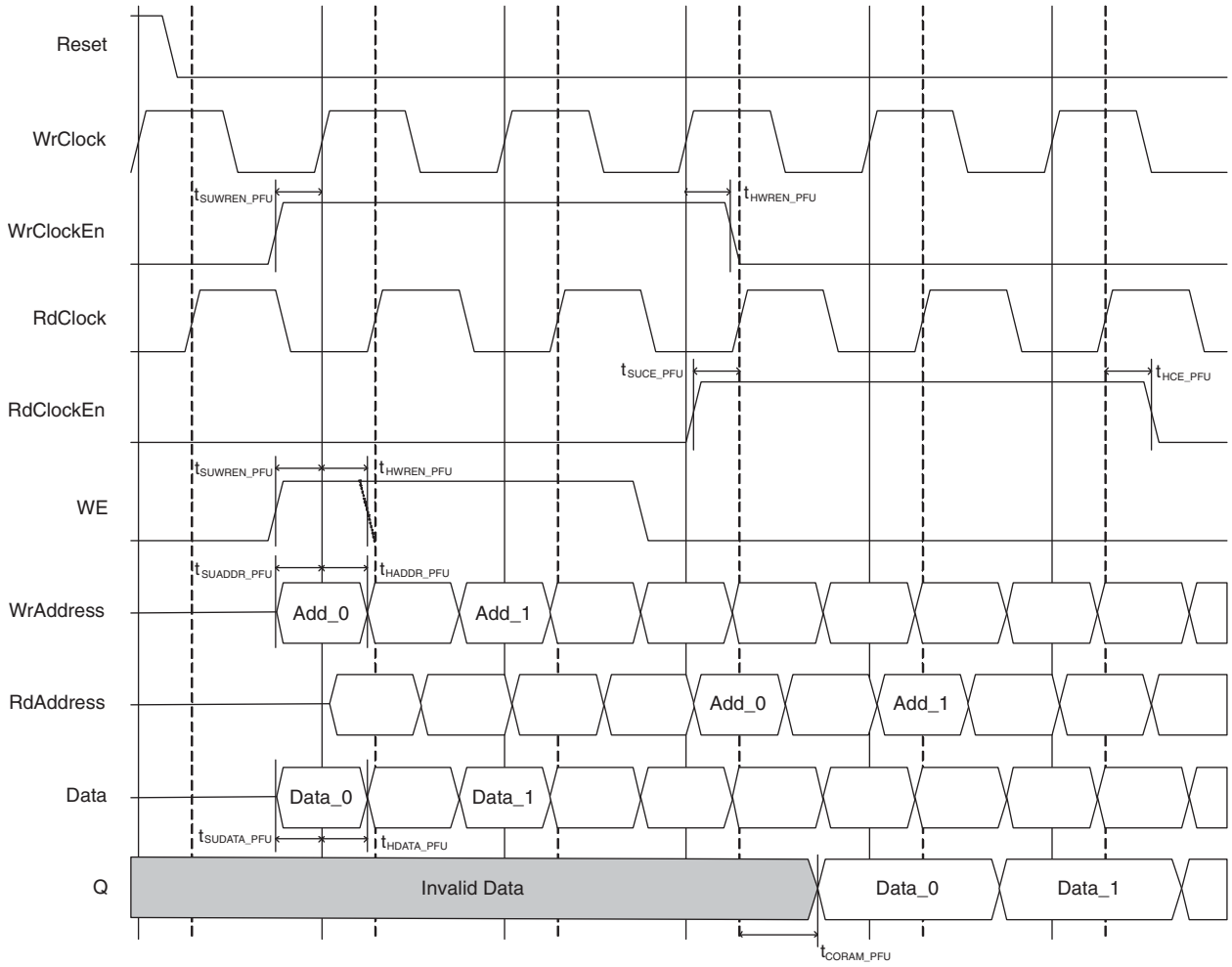


Figure 11-44. PFU Based Distributed Dual Port RAM Timing Waveform – with Output Registers

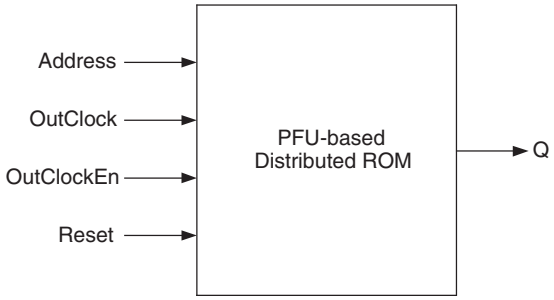


Distributed ROM (Distributed_ROM) – PFU Based

PFU-based Distributed ROM is also created using the 4-input LUT (Look-Up Table) available in the PFU. These LUTs can be cascaded to create larger Distributed Memory sizes.

Figure 11-45 shows the Distributed ROM module as generated by IPexpress.

Figure 11-45. Distributed ROM Generated by IPexpress



The generated module makes use of the 4-input LUT available in the PFU. Additional logic like Clock and Reset is generated by utilizing the resources available in the PFU.

Ports such as Out Clock (OutClock) and Out Clock Enable (OutClockEn) are not available in the hardware primitive. These are generated by IPexpress when the user wants to enable the output registers in the IPexpress configuration.

The various ports and their definitions for memory are as per Table 11-16. The table lists the corresponding ports for the module generated by IPexpress and for the primitive.

Table 11-16. PFU-based Distributed ROM Port Definitions

Port Name in Generated Module	Port Name in the PFU Block Primitive	Description	Active State
Address	AD[3:0]	Address	—
OutClock	—	Out Clock	Rising Clock Edge
OutClockEn	—	Out Clock Enable	Active High
Reset	—	Reset	Active High
Q	DO	Data Out	—

Users have the option to enable the output registers for Distributed ROM (Distributed_ROM). Figures 11-46 and 11-47 show the internal timing waveforms for the Distributed ROM with these options.

Figure 11-46. PFU Based ROM Timing Waveform – without Output Registers

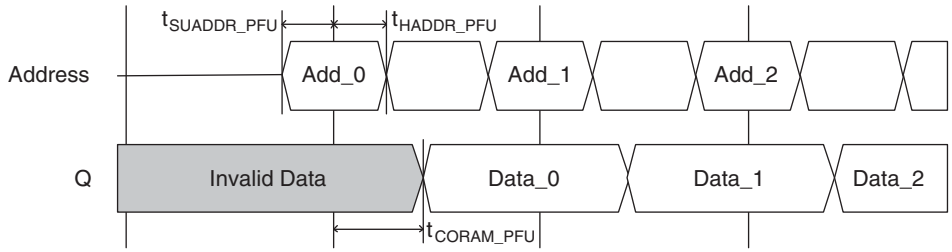
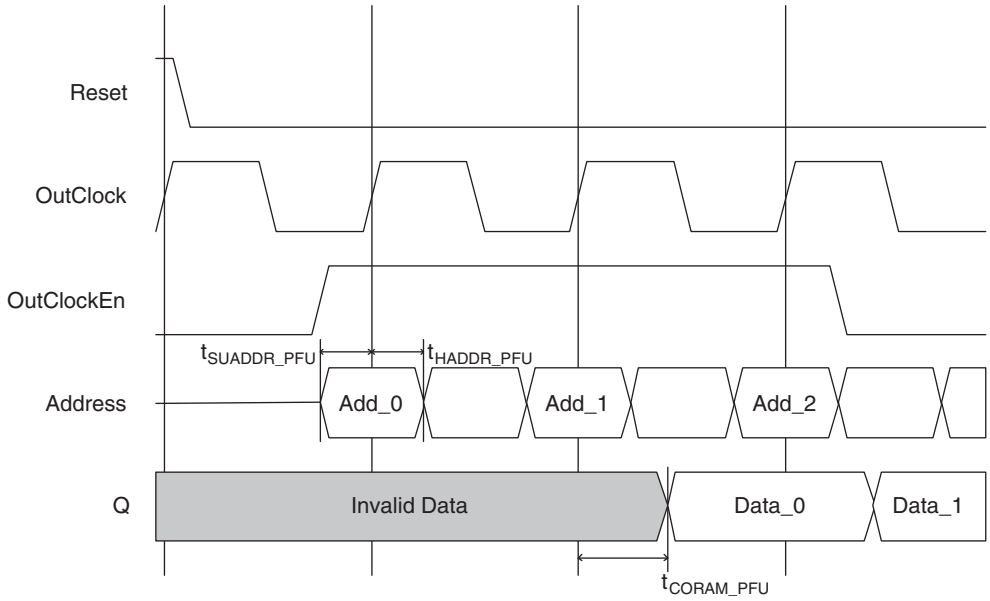


Figure 11-47. PFU Based ROM Timing Waveform – with Output Registers



Initializing Memory

In the EBR based ROM or RAM memory modes and the PFU based ROM memory mode, it is possible to specify the power-on state of each bit in the memory array. Each bit in the memory array can have one of two values: 0 or 1.

Initialization File Format

The initialization file is an ASCII file, which users can create or edit using any ASCII editor. IPexpress supports three types of memory file formats:

- Binary file
- Hex File
- Addressed Hex

The file name for the memory initialization file is *.mem (<file_name>.mem). Each row depicts the value to be stored in a particular memory location and the number of characters (or the number of columns) represents the number of bits for each address (or the width of the memory module).

The Initialization File is primarily used for configuring the ROMs. The EBR in RAM mode can optionally use this Initialization File also to preload the memory contents.

Binary File

The file is essentially a text file of 0's and 1's. The rows indicate the number of words and columns indicate the width of the memory.

```
Memory Size 20x32
00100000010000000010000001000000
00000001000000010000000100000001
00000010000000100000001000000010
00000011000000110000001100000011
00000100000001000000010000000100
00000101000001010000010100000101
00000110000001100000011000000110
00000111000001110000011100000111
00001000010010000000100001001000
00001001010010010000100101001001
00001010010010100000101001001010
00001011010010110000101101001011
00001100000011000000110000001100
00001101001011010000110100101101
00001110001111100000111000111110
00001111001111110000111100111111
00010000000100000001000000010000
00010001000100010001000100010001
00010010000100100001001000010010
00010011000100110001001100010011
```

Hex File

The Hex file is essentially a text file of Hex characters arranged in a similar row-column arrangement. The number of rows in the file is same as the number of address locations, with each row indicating the content of the memory location.

```
Memory Size 8x16
A001
0B03
1004
CE06
0007
040A
0017
02A4
```

Addressed Hex

Addressed Hex consists of lines of address and data. Each line starts with an address, followed by a colon, and any number of data. The format of memfile is address: data data data data ... where address and data are hexadecimal numbers.

```
-A0 : 03 F3 3E 4F
-B2 : 3B 9F
```

The first line puts 03 at address A0, F3 at address A1, 3E at address A2,and 4F at address A3. The second line puts 3B at address B2 and 9F at address B3.

There is no limitation on the values of address and data. The value range is automatically checked based on the values of addr_width and data_width. If there is an error in an address or data value, an error message is printed. Users need not specify data at all address locations. If data is not specified at certain address, the data at that location is initialized to 0. IPexpress makes memory initialization possible both through the synthesis and simulation flows.

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
February 2006	01.0	Initial release.
April 2006	01.1	Updated the Initializing Memory section
September 2006	01.2	Added LatticeECP2M device information. Added dual port memory access notes.
April 2007	01.3	Updated Utilizing IPexpress section.
February 2008	01.4	Updated Pseudo-Dual Port RAM Attributes for LatticeECP2/M table.
March 2008	01.5	Updated FIFO_DC without Output Registers (Non-Pipelined) figure.
June 2008	01.6	Updated First In First Out (FIFO, FIFO_DC) – EBR Based section.
		Removed Read-Before-Write sysMEM EBR mode.
August 2008	01.7	Corrected AddressA, AddressB information in EBR-based True Dual Port Memory Port Definitions table.
September 2008	01.8	Updated IPexpress Flow text section.
March 2009	01.9	Updated Memory Modules text section.
July 2011	02.0	Added the setup and hold requirements for addresses to EBR-based memories.

Appendix A. Attribute Definitions

DATA_WIDTH

Data width is associated with the RAM and FIFO elements. The DATA_WIDTH attribute will define the number of bits in each word. It takes the values as defined in the RAM size tables in each memory module.

REGMODE

REGMODE or the Register mode attribute is used to enable pipelining in the memory. This attribute is associated with the RAM and FIFO elements. The REGMODE attribute takes the NOREG or OUTREG mode parameter that disables and enables the output pipeline registers.

RESETMODE

The RESETMODE attribute allows users to select the mode of reset in the memory. This attribute is associated with the block RAM elements. RESETMODE takes two parameters: SYNC and ASYNC. SYNC means that the memory reset is synchronized with the clock. ASYNC means that the memory reset is asynchronous to clock.

CSDECODE

CSDECODE or the Chip Select Decode attributes are associated to block RAM elements. CS, or Chip Select, is the port available in the EBR primitive that is useful when memory requires multiple EBR blocks cascaded. The CS signal forms the MSB for the address when multiple EBR blocks are cascaded. CS is a 3-bit bus, so it can cascade eight memories easily. CSDECODE takes the following parameters: “000”, “001”, “010”, “011”, “100”, “101”, “110”, and “111”. CSDECODE values determine the decoding value of CS[2:0]. CSDECODE_W is chip select decode for write and CSDECODE_R is chip select decode for read for Pseudo Dual Port RAM. CSDECODE_A and CSDECODE_B are used for true dual port RAM elements and refer to the A and B ports.

WRITEMODE

The WRITEMODE attribute is associated with the block RAM elements. It takes the NORMAL and WRITETHROUGH mode parameters.

In NORMAL mode, the output data does not change or get updated, during the write operation. This mode is supported for all data widths.

In WRITETHROUGH mode, the output data is updated with the input data during the write cycle. This mode is supported for all data widths.

WRITEMODE_A and WRITEMODE_B are used for dual port RAM elements and refer to the A and B ports in case of a True Dual Port RAM.

For all modes (of the True Dual Port module), simultaneous read access from one port and write access from the other port to the same memory address is not recommended. The read data may be unknown in this situation. Also, simultaneous write access to the same address from both ports is not recommended. (When this occurs, the data stored in the address becomes undetermined when one port tries to write a ‘H’ and the other tries to write a ‘L’).

It is recommended that the designer implements control logic to identify this situation if it occurs and either:

1. Implement status signals to flag the read data as possibly invalid, or
2. Implement control logic to prevent the simultaneous access from both ports.

GSR

GSR or the Global Set/ Reset attribute is used to enable or disable the global set/reset for RAM element.

LatticeECP2/M High-Speed I/O Interface

June 2010 Technical Note TN1105

Introduction

LatticeECP2™ and LatticeECP2M™ devices support Double Data Rate (DDR) and Single Data Rate (SDR) interfaces using the logic built into the Programmable I/O (PIO). SDR applications capture data on one edge of a clock while the DDR interfaces capture data on both the rising and falling edges of the clock, thus doubling performance. The LatticeECP2/M I/Os also have dedicated circuitry to support DDR and DDR2 SDRAM memory interfaces. This technical note details the use of LatticeECP2/M devices to implement both a high-speed generic DDR interface and DDR and DDR2 memory interfaces.

DDR and DDR2 SDRAM Interfaces Overview

A DDR SDRAM interface will transfer data at both the rising and falling edges of the clock. The DDR2 is the second generation of the DDR SRDRAM memory.

The DDR and DDR2 SDRAM interfaces rely on the use of a data strobe signal, called DQS, for high-speed operation. The DDR SDRAM interface uses a single-ended DQS strobe signal, whereas the DDR2 interface uses a differential DQS strobe. Figures 12-1 and 12-2 show typical DDR and DDR2 SDRAM interface signals. SDRAM interfaces are typically implemented with eight DQ data bits per DQS. An x16 interface will use two DQS signals and each DQS is associated with eight DQ bits. Both the DQ and DQS are bi-directional ports used to both read and write to the memory.

When reading data from the external memory device, data coming into the device is edge-aligned with respect to the DQS signal. This DQS strobe signal needs to be phase-shifted 90 degrees before FPGA logic can sample the read data. When writing to a DDR/DDR2 SDRAM, the memory controller (FPGA) must shift the DQS by 90 degrees to center-align with the data signals (DQ). A clock signal is also provided to the memory. This clock is provided as a differential clock (CLKP and CLKN) to minimize duty cycle variations. The memory also uses these clock signals to generate the DQS signal during a read via a DLL inside the memory. Figures 12-3 and 12-4 show DQ and DQS timing relationships for read and write cycles. For other detailed timing requirements, please refer to the DDR SDRAM JEDEC specification (JESD79C).

During read, the DQS signal is LOW for some duration after it comes out of tristate. This state is called Preamble. The state when the DQS is LOW before it goes into Tristate is the Postamble state. This is the state after the last valid data transition.

DDR SDRAM also requires a Data Mask (DM) signal to mask data bits during write cycles. Note that the ratio of DQS to data bits is independent of the overall width of the memory. An 8-bit interface will have one strobe signal.

DDR SDRAM interfaces use the SSTL25 Class I/II I/O standards whereas the DDR2 SDRAM interface uses the SSTL18 Class I/II I/O standards. The DDR2 SDRAM interface also supports differential DQS (DQS and DQS#).

Figure 12-1. Typical DDR SDRAM Interface

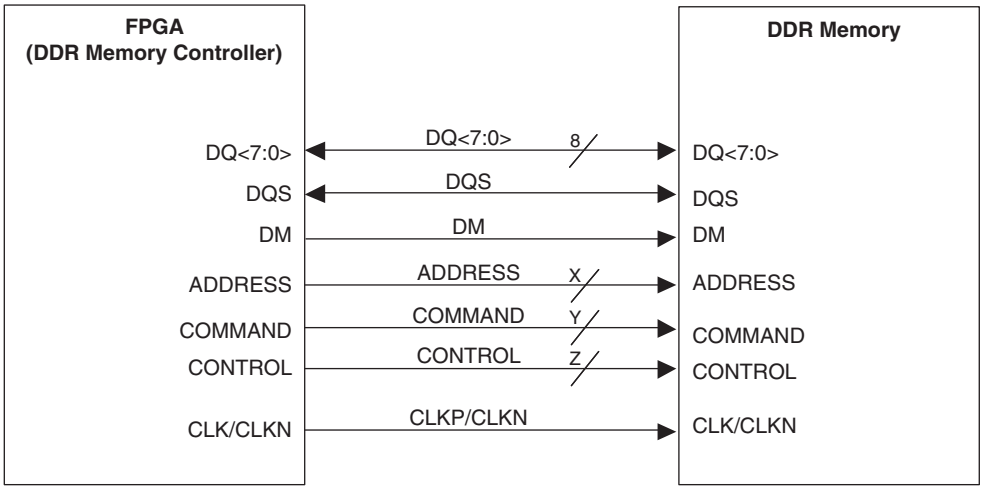
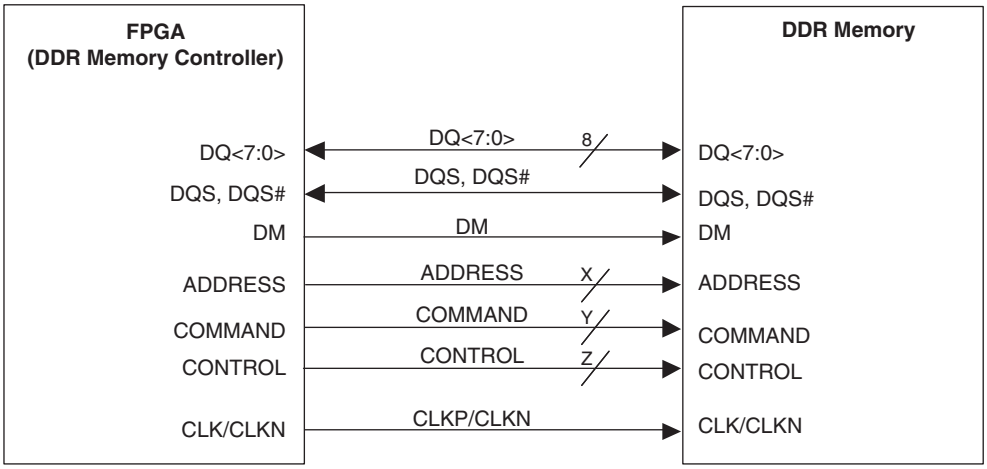


Figure 12-2. Typical DDR2 SDRAM Interface



The following two figures show the DQ and DQS relationship for memory read and write interfaces.

Figure 12-3. DQ-DQS During READ

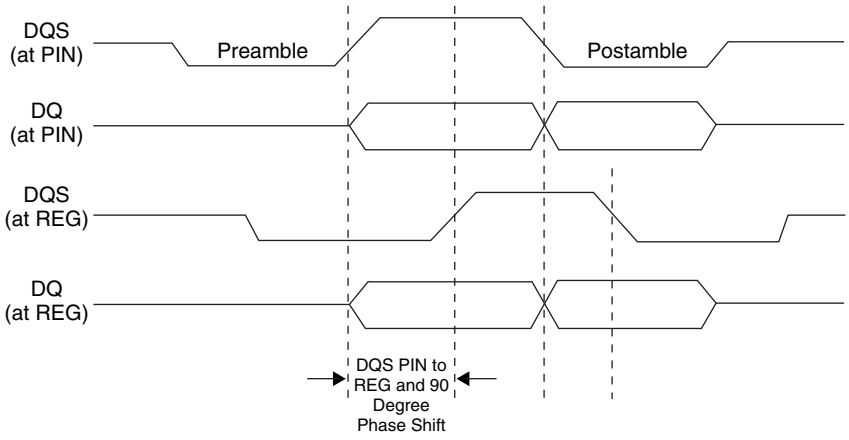
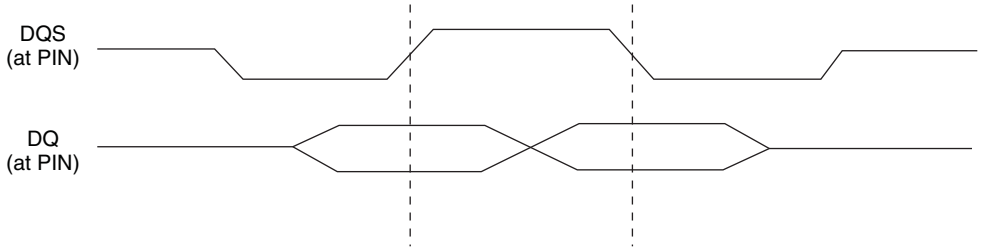


Figure 12-4. DQ-DQS During WRITE



Implementing DDR Memory Interfaces with LatticeECP2/M Devices

As described in the DDRSDRAM overview section, the DDR SDRAM interfaces rely primarily on the use of a data strobe signal called DQS for high-speed operation. When reading data from the external memory device, data coming into the LatticeECP2/M device is edge-aligned with respect to the DQS signal. Therefore, the LatticeECP2/M device needs to shift the DQS (a 90-degree phase shift) before using it to sample the read data. When writing to a DDR SDRAM, the memory controller from the LatticeECP2/M device must generate a DQS signal that is center-aligned with the DQ, the data signals. This is accomplished by ensuring the DQS strobe is 90 degrees ahead relative to DQ data.

LatticeECP2/M devices have dedicated DQS support circuitry for generating the appropriate phase shifting for DQS. The DQS phase shift circuit uses a frequency reference DLL to generate delay control signals associated with each of the dedicated DQS pins and is designed to compensate for process, voltage and temperature (PVT) variations. The frequency reference is provided through one of the global clock pins.

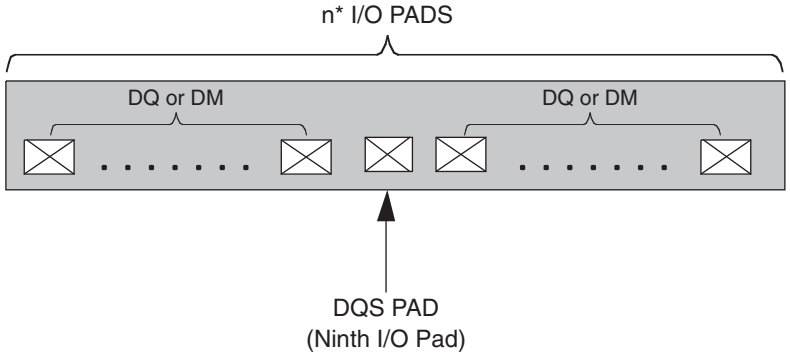
The dedicated DDR support circuit is also designed to provide comfortable and consistent margins for data sampling window.

This section describes how to implement the read and write sections of a DDR memory interface. It also provides details of the DQ and DQS grouping rules associated with the LatticeECP2/M devices.

DQS Grouping

Each DQS group generally consists of at least 10 I/Os (one DQS, eight DQ and one DM) to implement a complete 8-bit DDR memory interface. LatticeECP2/M devices support DQS signals on the bottom, left and right sides of the device. Each DQS signal on the bottom half of the device will span across 18 I/Os and on the left and right sides of the device will span across 16 I/Os. Any 10 of these I/Os spanned by the DQS can be used to implement an 8-bit DDR memory interface.

Figure 12-5. DQ-DQS Grouping



*n=18 on bottom banks and n=16 on the left and right side banks.

Figure 12-5 shows a typical DQ-DQS group for a LatticeECP2/M device. The ninth I/O of this group of 16 I/Os (on the left and right side banks) and 18 I/Os (on the bottom bank) is the dedicated DQS pin. All eight pads before of the DQS and 7 (on the left and right side) and 9 (on the bottom bank) pads after the DQS are covered by this DQS bus span. The user can assign any eight of these I/O pads to be DQ data pins. Therefore, to implement a 32-bit wide memory interface you would need to use four such DQ-DQS groups.

When not interfacing with the memory, the dedicated DQS pin can be used as a general purpose I/O. Each of the dedicated DQS pins is internally connected to the DQS phase shift circuitry. The pinout information contained in the LatticeECP2/M Family Data Sheet shows pin locations for the DQS pads. Table 12-1 shows an extract from the data sheet.

In this case, the DQS is marked as LDQS8 (L = left side, 8 = associated PFU row/column). Since DQS is always the fifth True Pad in the DQ-DQS group, counting from low to high PFU Row/Column number, LDQS6 will cover PL2A to PL11B. Following this convention, there are eight pads before and seven pads after DQS for DQ available following counter-clockwise for the left and bottom sides of the device and following clockwise for the top and right sides of the device. The user can assign any eight of these pads to be DQ data signals.

Table 12-1. ECP2-50 672 fpBGA Pinout from LatticeECP2/M Family Data Sheet

Ball Number	Ball Function	Bank	Dual Function	Differential
D2	PL2A	7	VREF2_7	T*
D1	PL2B	7	VREF1_7	C*
GND	GNDIO	7		
F6	PL5A	7		T
F5	PL5B	7		C
VCCIO	VCCIO	7		
E4	PL6A	7		T*
E3	PL6B	7		C*
VCC	VCC	7		
E2	PL7A	7		T
E1	PL7B	7		C
GND	GNDIO	7		
GND	GND	7		
H6	PL8A	7	LDQS8	T*
H5	PL8B	7		C*
F2	PL9A	7		T
VCCIO	VCCIO	7		
F1	PL9B	7		C
H8	PL10A	7		T*
J9	PL10B	7		C*
G4	PL11A	7		T
GND	GNDIO	7		
G3	PL11B	7		C
H7	PL12A	7		T*
VCCAUX	VCCAUX	7		

DDR Software Primitives

This section describes the software primitives that can be used to implement DDR interfaces. These primitives include:

- **DQSDLL** – The DQS delay calibration DLL
- **DQSBUFC** – The DQS delay function and the clock polarity selection logic
- **IDDRMX1A** – The DDR input and DQS to system clock transfer registers with half clock cycle transfer
- **IDDRMFX1A** – The DDR input and DQS to system clock transfer registers with full clock cycle transfer
- **ODDRMXA** – The DDR output registers

HDL usage examples for each of these primitives are listed in Appendices A and B.

DQSDLL

The DQSDLL generates a 90-degree phase shift required for the DQS signal. This primitive implements the on-chip DQSDLL. Only one DQSDLL should be instantiated for all the DDR implementations on one half of the device. The clock input to this DLL should be at the same frequency as the DDR interface. The DLL generates the delay based on this clock frequency and the update control input to this block. The DLL updates the dynamic delay control to the DQS delay block when this update control (UDDCNTL) input is asserted. Figure 12-6 shows the primitive symbol. The active low signal on UDDCNTL updates the DQS phase alignment and should be initiated at the beginning of READ cycles.

Figure 12-6. DQSDLL Symbol

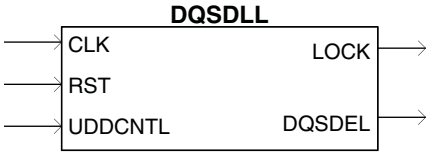


Table 12-2 provides a description of the ports.

Table 12-2. DQSDLL Ports

Port Name	I/O	Description
CLK	I	System CLK should be at the frequency of the DDR interface from the FPGA core.
RST	I	Resets the DQSDLL
UDDCNTL	I	Provides update signal to the DLL that will update the dynamic delay.
LOCK	O	Indicates when the DLL is in phase.
DQSDEL	O	The digital delay generated by the DLL, should be connected to the DQSBUF primitive.

DQSDLL Update Control: The DQS Delay can be updated for PVT variation using the UDDCNTL input. The DQSDEL is updated when the when the UDDCNTL is held LOW. The DQSDEL can be updated when variations are expected. DQSDEL can be updated anytime, except when the memory controller is receiving data from the memory.

DQSDLL Configuration: By default, this DLL will generate a 90-degree phase shift for the DQS strobe based on the frequency of the input reference clock to the DLL. The user can control the sensitivity to jitter by using the LOCK_SENSITIVITY attribute. This configuration bit can be programmed to be either “HIGH” or “LOW”.

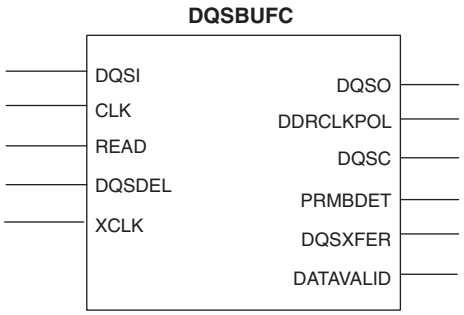
The DLL Lock Detect circuit has two modes of operation controlled by the LOCK_SENSITIVITY bit, which selects more or less sensitivity to jitter. If this DLL is operated at or above 150 MHz, it is recommended that the LOCK_SENSITIVITY bit be programmed “HIGH” (more sensitive). When running at or below 100 MHz, it is recom-

mended that the bit be programmed “LOW” (more tolerant). For 133 MHz, the LOCK_SENSITIVITY bit can go either way.

DQSBUFC

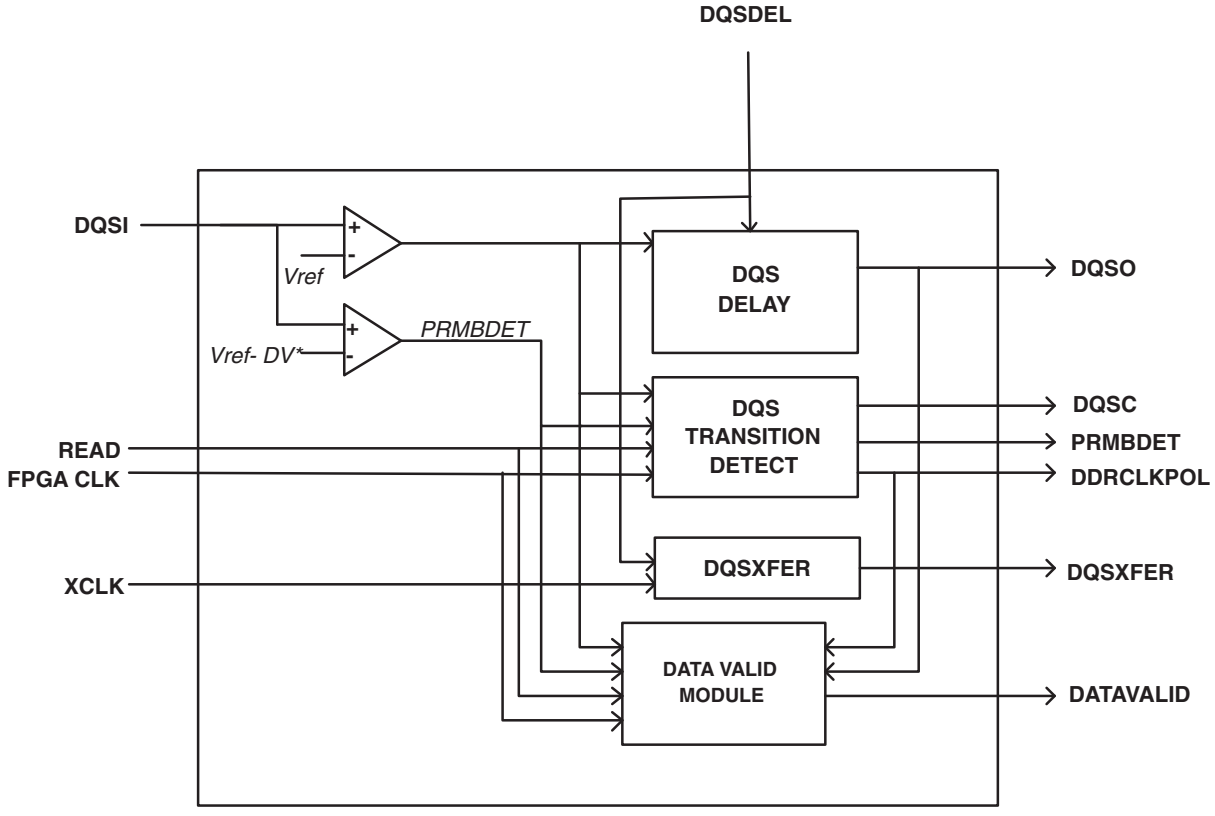
This primitive implements the DQS delay and the DQS transition detector logic. Figure 12-7 shows the primitive symbol.

Figure 12-7. DQSBUFC Symbol



The DQSBUFC is composed of the DQS Delay, the DQS Transition Detect and the DQSXFER block as shown in Figure 12-8. This block inputs the DQS and delays it by 90 degrees. It also generates the DDR Clock Polarity and the DQSXFER signal. The preamble detect (PRMBDET) signal is generated from the DQSI input using a voltage divider circuit.

Figure 12-8. DQSBUFC Function



**DV ~ 170mV for DDR1 (SSTL25 signaling)*

**DV ~ 120mV for DDR2 (SSTL18 signaling)*

DQS Delay Block: The DQS Delay block receives the digital control delay line (DQSDEL) coming from one of the two DQSDLL blocks. These control signals are used to delay the DQSI by 90 degrees. DQSO is the delayed DQS and is connected to the clock input of the first set of DDR registers.

DQS Transition Detect: The DQS Transition Detect block generates the DDR Clock Polarity signal based on the phase of the FPGA clock at the first DQS transition. The DDR READ control signal and FPGA CLK inputs to this coming and should be coming from the FPGA core.

DQSXFER: This block generates the 90-degree phase shifted clock to for the DDR Write interface. The input to this block is the XCLK. The user can choose to connect this either to the edge clock or the FPGA clocks. The DQSXFER is routed using the DQSXFER tree to all the I/Os spanned by that DQS.

Data Valid Module: The data valid module generates a DATAVALID signal. This signal indicates to the FPGA that valid data is transmitted out of the input DDR registers to the FPGA core.

Table 12-3 provides a description of the I/O ports associated with the DQSBUFC primitive.

Table 12-3. DQSBUFC Ports

Port Name	I/O	Description
DQSI	I	DQS Strobe signal from memory
CLK	I	System CLK
READ	I	Read generated from the FPGA core
DQSDEL	I	DQS Delay from the DQSDLL primitive
XCLK	I	Edge Clock or System CLK
DQSO	O	Delayed DQS Strobe signal, to the input capture register block
DQSC	O	DQS Strobe signal before delay, going to the FPGA core logic
DDRCLKPOL	O	DDR Clock Polarity signal
PRMBDET	O	Preamble detect signal, going to the FPGA core logic
DQSXFER	O	90 degree shifted clock going to the Output DDR register Block
DATAVALID	O	Signal indicating transmission of Valid data to the FPGA core

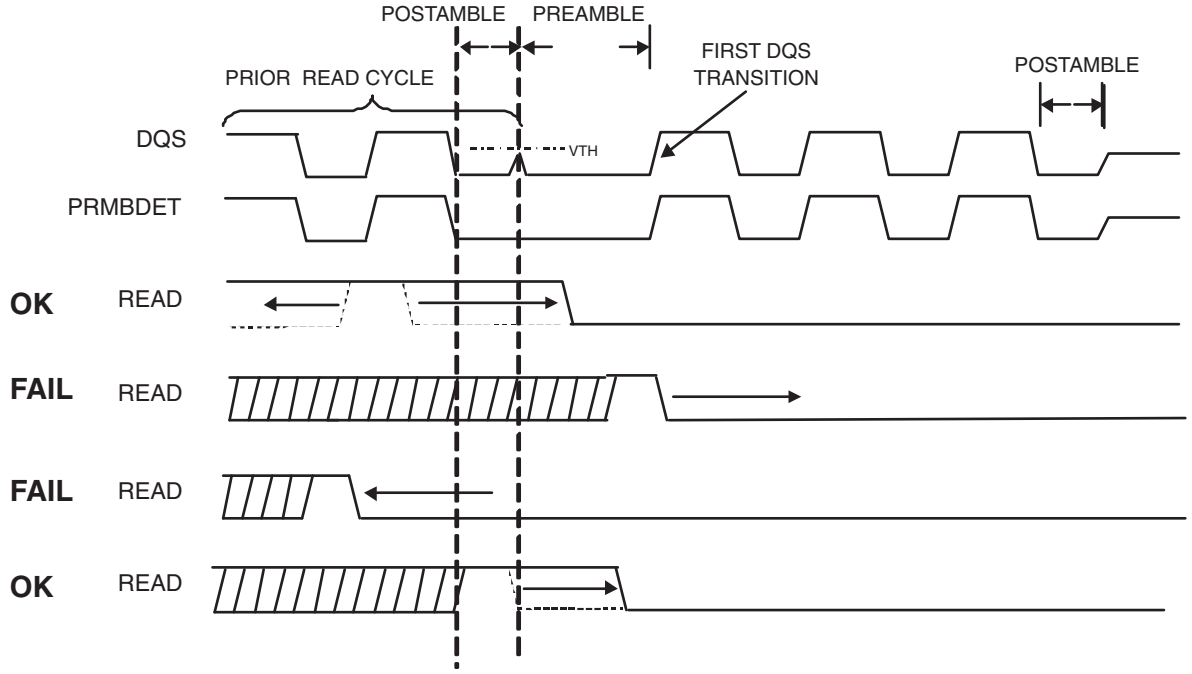
READ Pulse Generation

The READ signal to the DQSBUFC block is internally generated in the FPGA core. The READ signal goes high when the READ command to control the DDR-SDRAM is initially asserted. This precedes the DQS preamble by one cycle, yet may overlap the trailing bits of a prior read cycle. The DQS Detect circuitry of the LatticeECP2/M device requires the falling edge of the READ signal to be placed within the preamble stage.

The preamble state of the DQS can be detected using the CAS latency and the round trip delay for the signals between the FPGA and the memory device. Note that the internal FPGA core generates the READ pulse. The rise of the READ pulse should coincide with the initial READ command of the Read Burst and need to go low before the Preamble goes high.

Figure 12-9 shows a READ Pulse timing example with respect to the PRMBDET signal.

Figure 12-9. READ Pulse Generation



IDDRMX1A

This primitive will implement the input register block in memory mode. The DDR registers are designed to use edge clock routing on the I/O side and the primary clock on the FPGA side. The ECLK input is used to connect to the DQS strobe coming from the DQS delay block (DQSBUFC primitive). The SCLK input is connected to the system (FPGA) clock. DDRCLKPOL is an input from the DQS Clock Polarity tree. This signal is generated by the DQS Transition detect circuit in the hardware. The DDRCLKPOL signal is used to choose the polarity of the SCLK to the synchronization registers.

Figure 12-10. IDDRMX1A Symbol

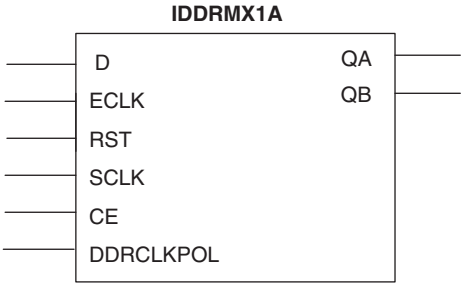


Table 12-4 provides a description of all I/O ports associated with the IDDRMX1A primitive.

Table 12-4. IDDRMX1A Ports

Port Name	I/O	Definition
D	I	DDR Data
ECLK	I	The phase shifted DQS should be connected to this input
RST	I	Reset
SCLK	I	System CLK
CE	I	Clock enable
DDRCLKPOL	I	DDR clock polarity signal
QA	O	Data at Positive edge of the CLK
QB	O	Data at the negative edge of the CLK

Note: The DDRCLKPOL input to IDDRMX1A should be connected to the DDRCLKPOL output of DQSBUFC.

Figure 12-11 shows the Input Register Block configured in the IDDRMX1A mode.

Figure 12-11. Input Register Block in IDDRMX1A Mode

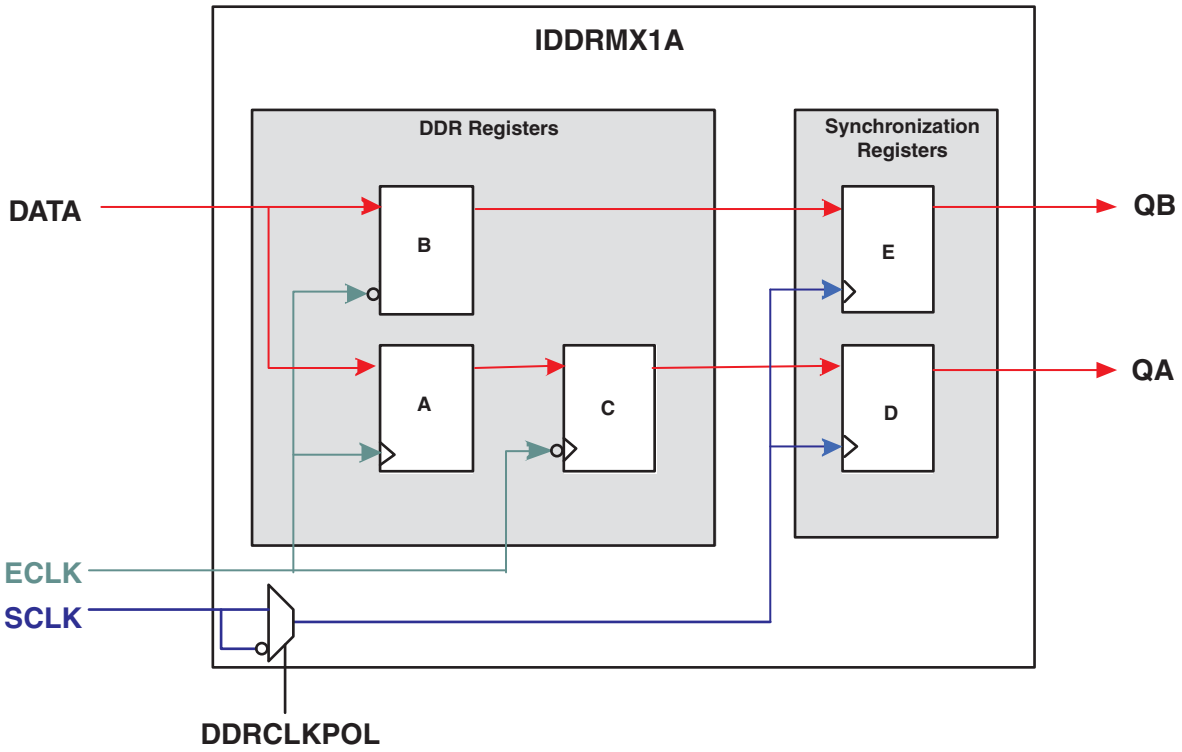
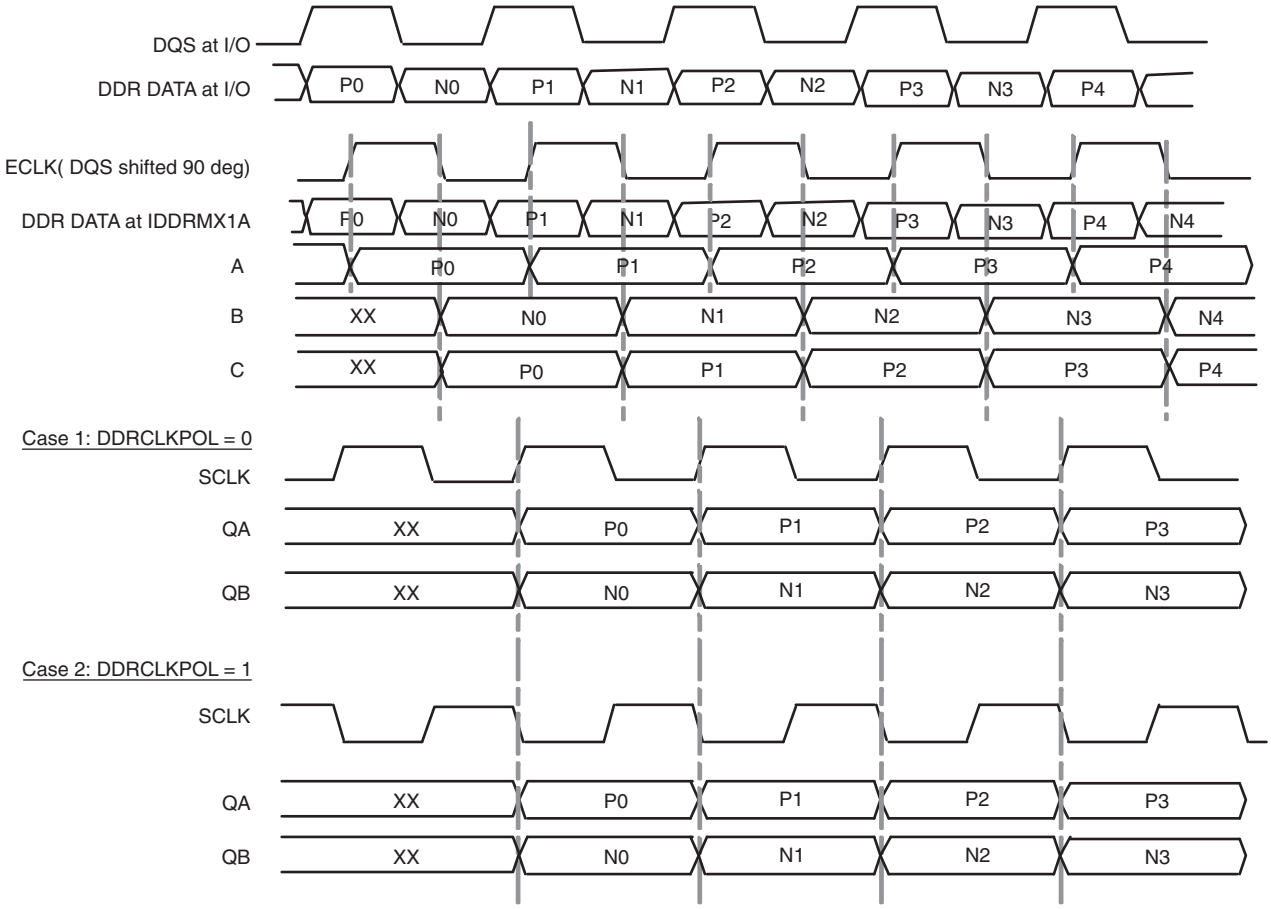


Figure 12-12 shows the IDDRMX1A timing waveform.

Figure 12-12. IDDRMX1A Waveform



IDDRMFX1A

With the IDDRMX1A, the data can enter the FPGA at either the positive or negative edge of the SCLK depending on the state of the DDRCLKPOL signal. The IDDRMFX1A module includes an additional clock transfer stage that ensures that the data is transferred at a known edge of the system clock.

Figure 12-13. IDDRMFX1A Symbol

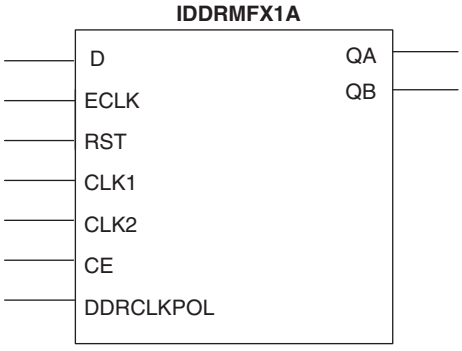


Table 12-5 provides a description of all I/O ports associated with the IDDRMFX1A primitive.

Table 12-5. IDDRMFX1A Ports

Port Name	I/O	Description
D	I	DDR Data
ECLK	I	The phase shifted DQS should be connected to this input
RST	I	Reset
CLK1	I	Slow FPGA CLK
CLK2	I	Slow FPGA CLK
CE	I	Clock enable
DDRCLKPOL	I	DDR clock polarity signal
QA	O	Data at the positive edge of the CLK
QB	O	Data at the negative edge of the CLK

Note: The DDRCLKPOL input to IDDRMFX1A should be connected to the DDRCLKPOL output of DQSBUFC.

Figure 12-14 shows the LatticeECP2 Input Register Block configured to function in the IDDRXMFX1A mode.

The DDR registers are designed to use Edge clock routing on the I/O side and the primary clock on the FPGA side. The ECLK input is used to connect to the DQS strobe coming from the DQS delay block (DQSBUFC primitive). The CLK1 and CLK2 inputs should be connected to the slow system (FPGA) clock. DDRCLKPOL is an input from the DQS Clock Polarity tree. This signal is generated by the DQS Transition detect circuit in the hardware. The additional clock transfer registers are shared with the output register block.

Figure 12-14. Input Register Block in IDDRMFX1A Mode

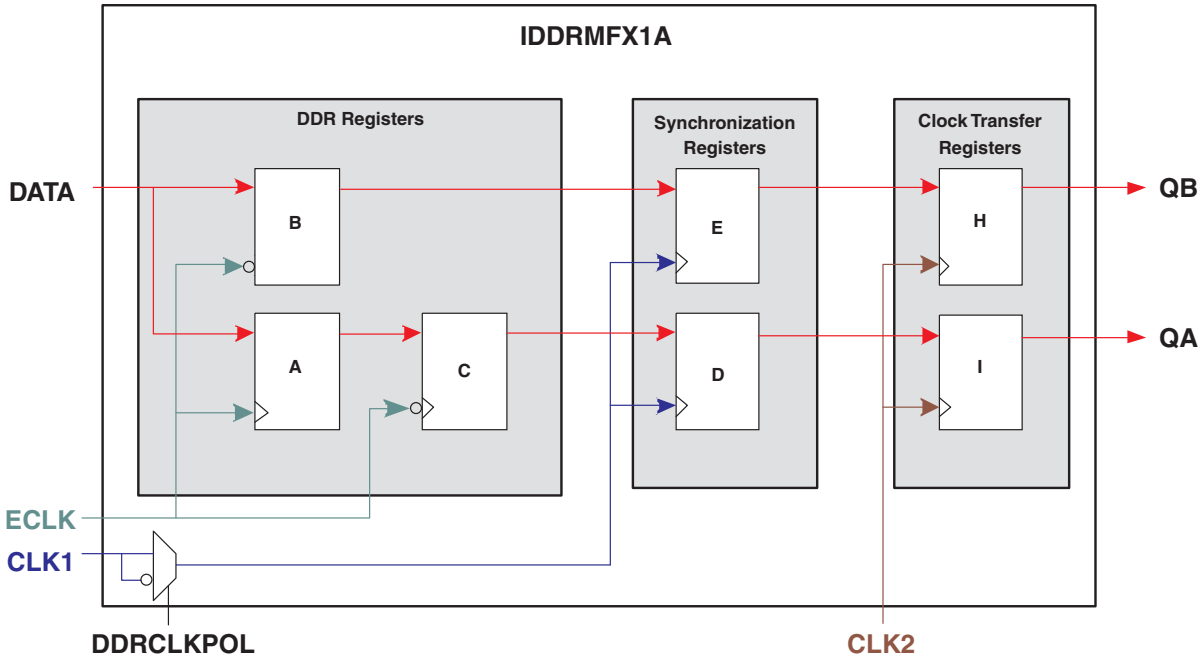
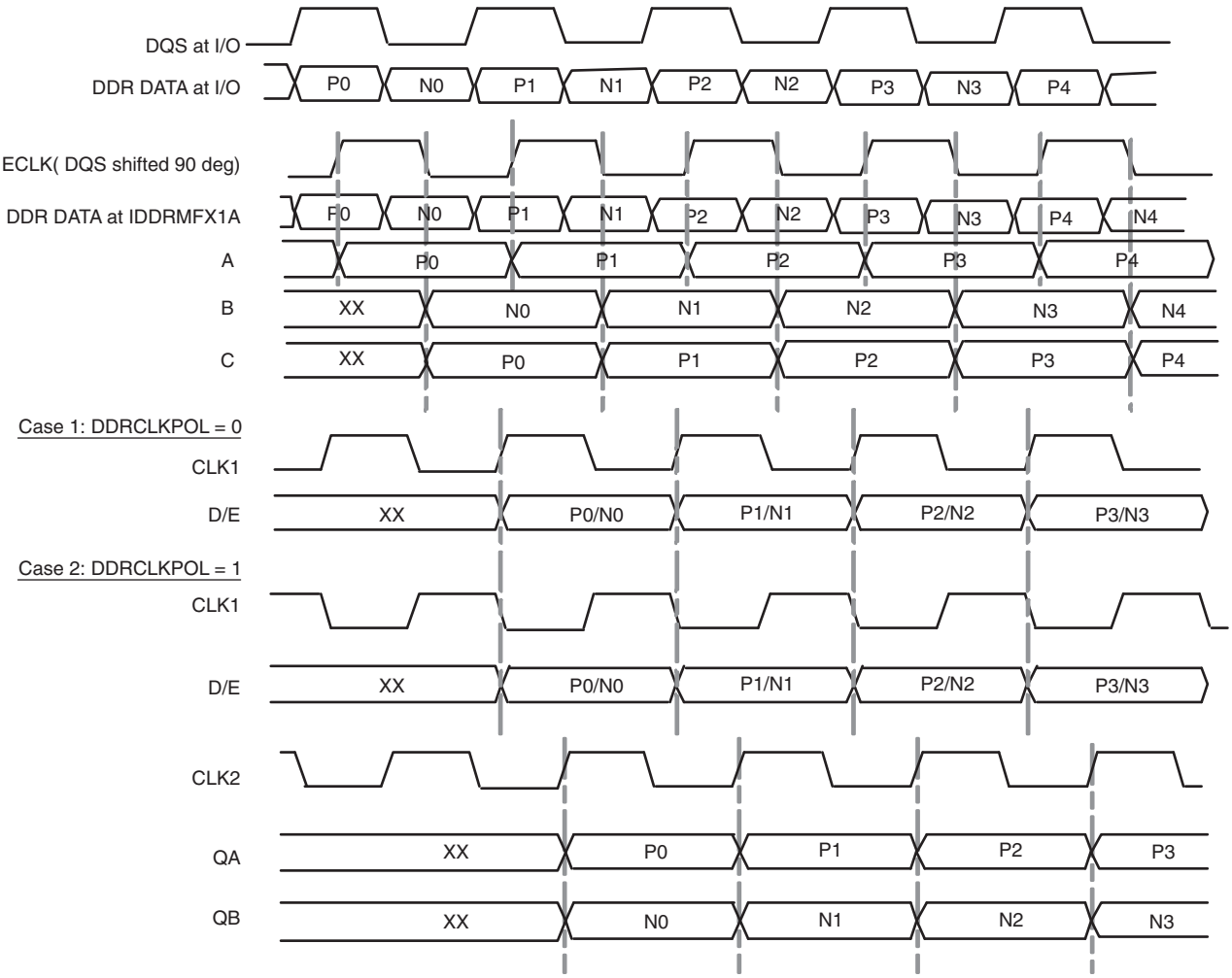


Figure 12-15 shows the IDDRMFX1A timing waveform.

Figure 12-15. IDDRMFX1A Waveform



ODDRMXA

The ODDRMXA primitive implements the output register for both the write and the tristate functions. This primitive is used to output DDR data and the DQS strobe to the memory. All the DDR output tristate functions are also implemented using this primitive.

Figure 12-16 shows the ODDRMXA primitive symbol and its I/O ports.

Figure 12-16. ODDRMXA Symbol

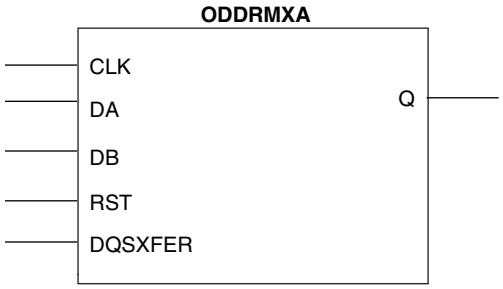


Table 12-6 provides a description of all I/O ports associated with the ODDRMXA primitive.

Table 12-6. ODDRMXA Ports

Port Name	I/O	Description
CLK	I	System CLK or ECLK
DA	I	Data at the negative edge of the clock
DB	I	Data at the positive edge of the clock
RST	I	Reset
DQSXFER	I	90-degree phase shifted clock coming from the DQSBUFC block
Q	I	DDR data to the memory

Notes:
1. RST should be held low during DDR Write operation.
2. DDR output and tristate registers do not have CE support. RST is available for the tristate DDRX mode (while reading). The LSR will default to set when used in the tristate mode.
3. When asserting reset during DDR writes, it is important to keep in mind that this only resets the flip-flops and not the latches.

Figure 12-17 shows the LatticeECP2 Output Register Block configured in the ODDRXMA mode.

Figure 12-17. Output Register Block in ODDRXMA Mode

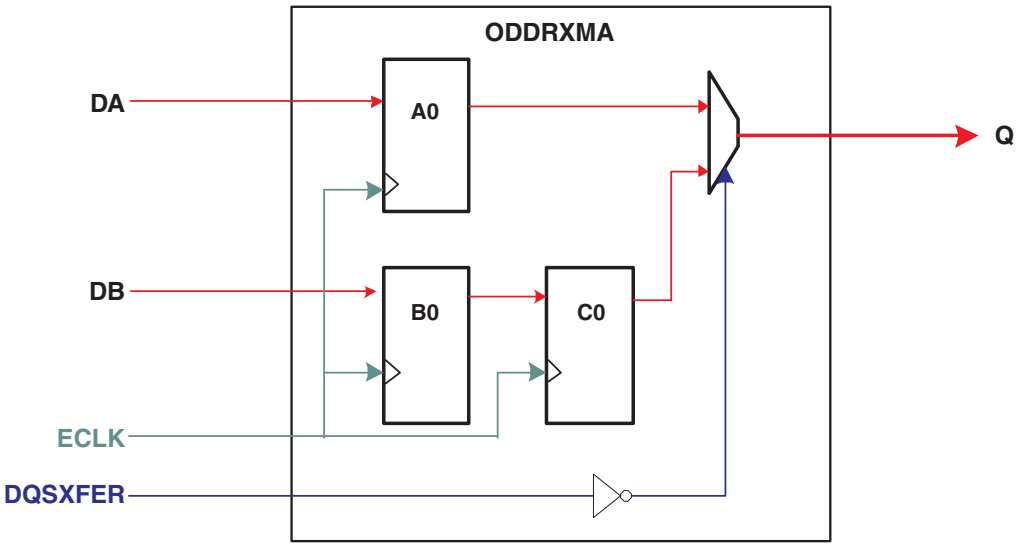
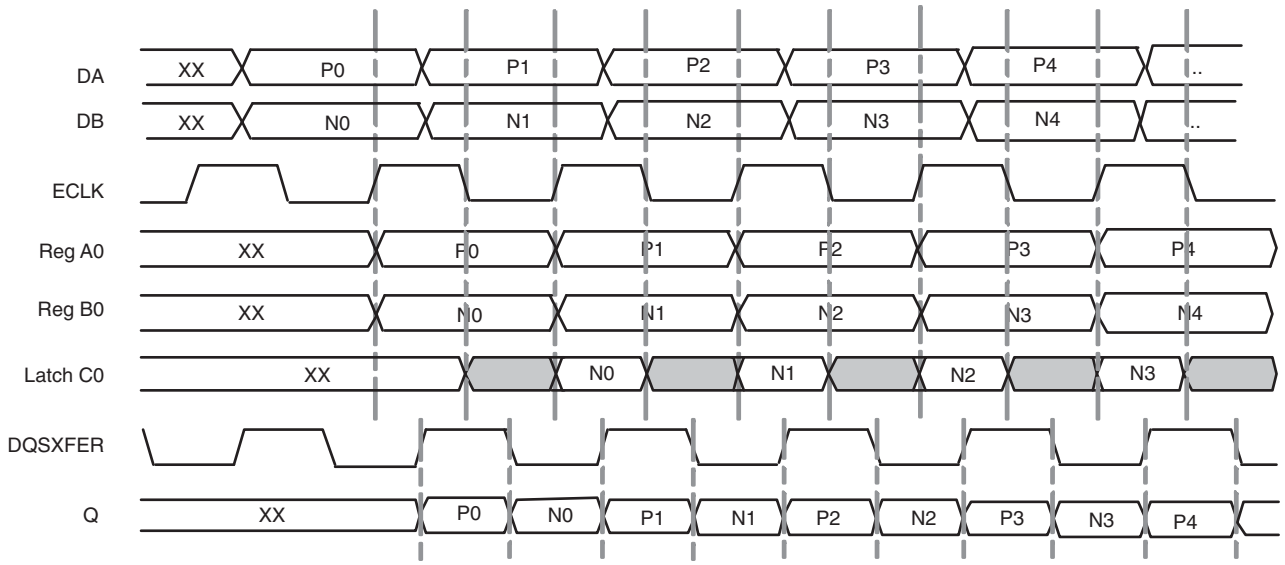


Figure 12-18 shows the ODDRMXA timing waveform.

Figure 12-18. ODDRMXA Waveform



Note that the DQSXFER is inverted inside the ODDRXMA. This will cause the data coming out of the ODDRXMA to be -90° in phase with the output of the ODDRXC module.

Memory Read Implementation

LatticeECP2/M devices contain a variety of features to simplify implementation of the read portion of a DDR interface:

- DLL compensated DQS delay elements
- DDR input registers
- Automatic DQS to system clock domain transfer circuitry
- Data Valid Module

DLL Compensated DQS Delay Elements

The DQS from the memory is connected to the DQS Delay element. The DQS Delay block receives a 6-bit delay control from the on-chip DQSDLL. This 6-bit delay is modeled as a single bit in the software. The LatticeECP2/M devices support two DQSDLL, one on the left and one on the right side of the device. The DQSDEL generated by the DQSDLL on the left side is routed to all the DQS blocks on the left and bottom half of the device. The delay generated by the DQSDLL on the right side is distributed to all the DQS Delay blocks on the right side and the other bottom half of the device. There are no DQS pins on the top banks of the device. These digital delay control signals are used to delay the DQS from the memory by 90 degrees.

The DQS received from the memory is delayed in each of the DQS Delay blocks and this delay DQS is used to clock the first set stage DDR input registers.

DQS Transition Detect or Automatic Clock Polarity Select

In a typical DDR memory interface design, the phase relation between the incoming delayed DQS strobe and the internal system clock (during the READ cycle) is unknown. Prior to the READ operation in DDR memories, DQS is in tristate (pulled by termination). Coming out of tristate, the DDR memory device drives DQS low in the Preamble State. The DQS Transition Detect block detects the first DQS rising edge after a preamble transition and generates a signal indicating the required polarity for the FPGA system clock (DDRCLKPOL). This signal is used to control the polarity of the clock to the synchronizing registers.

Data Valid Module

The data valid module generates a DATAVALID signal. This signal indicates to the FPGA that valid data is transmitted out the input DDR registers to the FPGA core.

DDR I/O Register Implementation

The first set of DDR registers is used to de-mux the DDR data at the positive and negative edge of the phase shifted DQS signal. The register that captures the positive-edge data is followed by a negative-edge triggered register. This register transfers the positive edge data from the first register to the negative edge of DQS so that both the positive and negative portions of the data are now aligned to the negative edge of DQS.

The second stage of registers is clocked by the FPGA clock, the polarity of this clock is selected by the DDR Clock Polarity signal generated by the DQS Transition Detect Block.

The I/O Logic registers can be implemented in two modes:

- Half Clock Transfer Mode
- Full Clock Transfer Mode

In Half Clock Transfer mode the data is transferred to the FPGA core after the second stage of the register. In Full Clock Transfer mode, an additional stage of I/O registers clocked by the FPGA clock is used to transfer the data to the FPGA core.

The LatticeECP2/M Family Data Sheet explains each of these circuit elements in more detail.

Memory Read Implementation in Software

Three primitives in the ispLEVER® design tools represent the capability of these three elements. The DQSDLL represents the DLL used for calibration. The IDDRMX1A/IDDRMFX1A primitive represents the DDR input registers and clock domain transfer registers with or without full clock transfer. Finally, the DQSBUFC represents the DQS delay block, the clock polarity control logic and the Data Valid module. Figures 12-19 and 12-20 show the READ interface block generated using the IPexpress™ tool in the ispLEVER software.

Figure 12-19. Software Primitive Implementation for Memory READ (Half Clock Transfer)

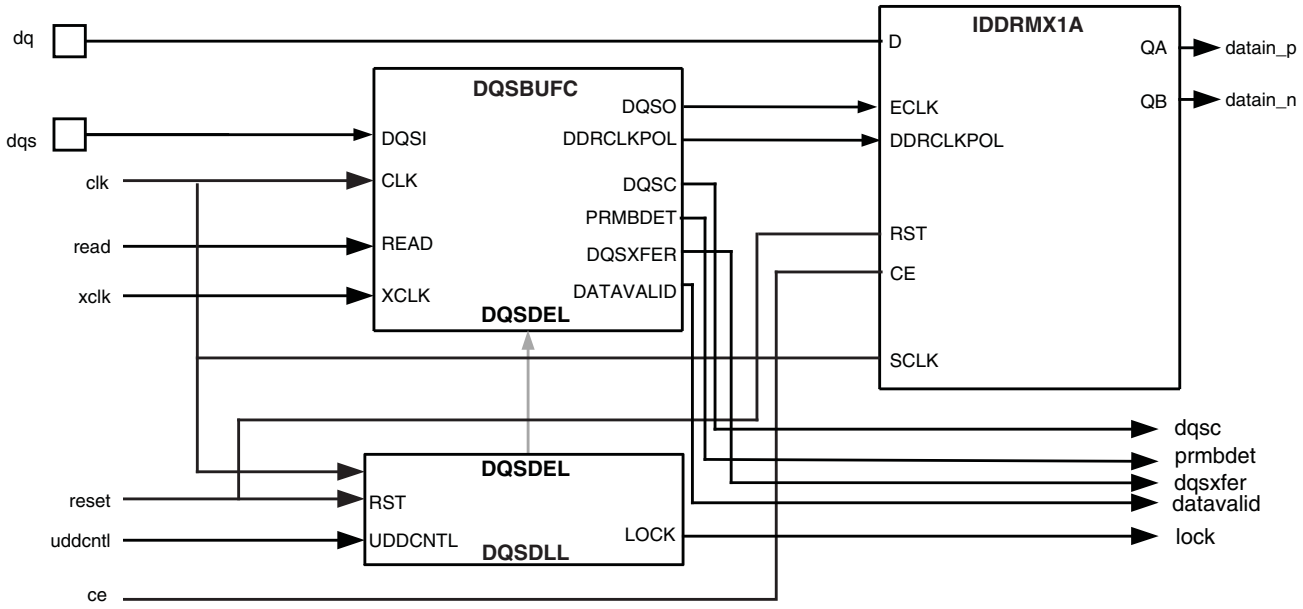
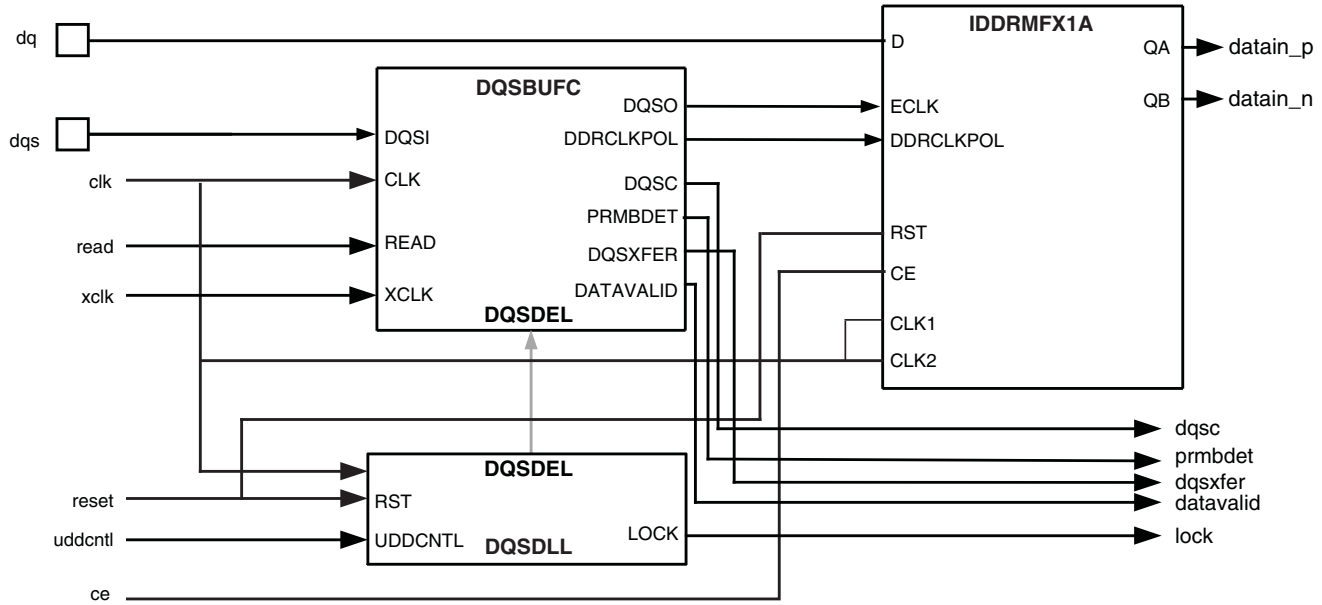


Figure 12-20. Software Primitive Implementation for Memory READ (Full Clock Transfer)



Read Timing Waveforms

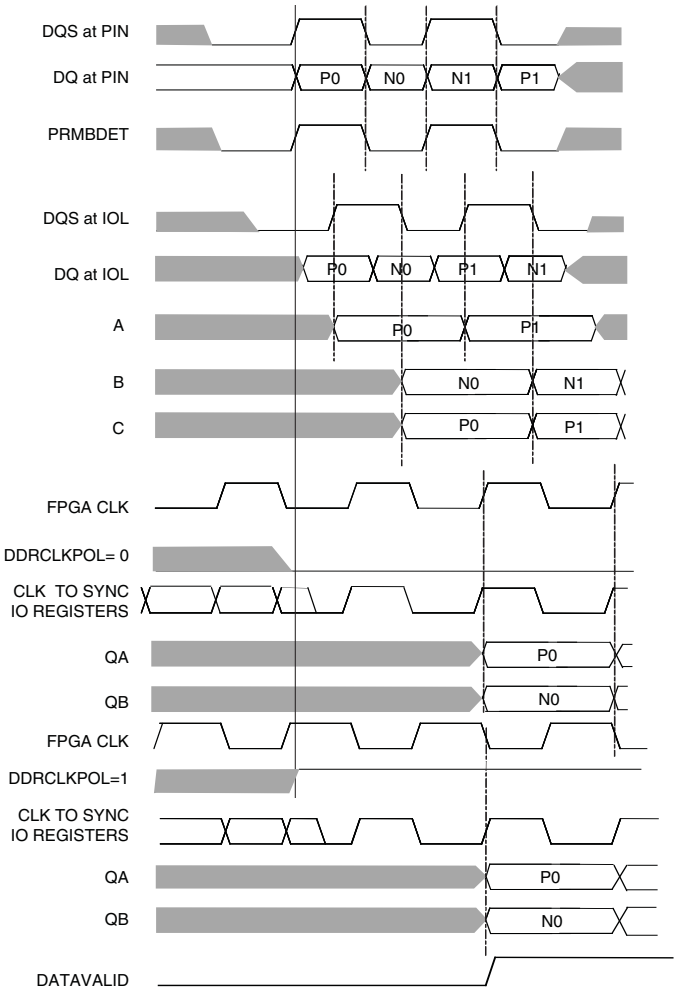
Figures 12-21 and 12-22 show READ data transfer for half and full clock cycle data transfer based on the results of the DQS Transition detector logic. This circuitry decides whether or not to invert the phase of FPGA system CLK to the synchronization registers based on the relative phases of PRMBDET and CLK.

- **Case 1:** If the FPGA clock is low on the first PRMBDET transition, then DDRCLKPOL is low and no inversion is required.
- **Case 2:** If the FPGA clock is high on the first PRMBDET, then DDRCLKPOL is high and the FPGA clock (CLK) needs to be inverted before it is used for synchronization.

Figure 12-21 illustrates the DDR data timing using half clock transfer mode at different stages of the IDDRMX1A registers. The first stage of the register captures data on the positive edge as shown by signal A and the negative edge as shown by signal B. Data stream A goes through an additional half clock cycle transfer shown by signal C. Phase-aligned data streams B and C are presented to the next stage registers clocked by the FPGA clock.

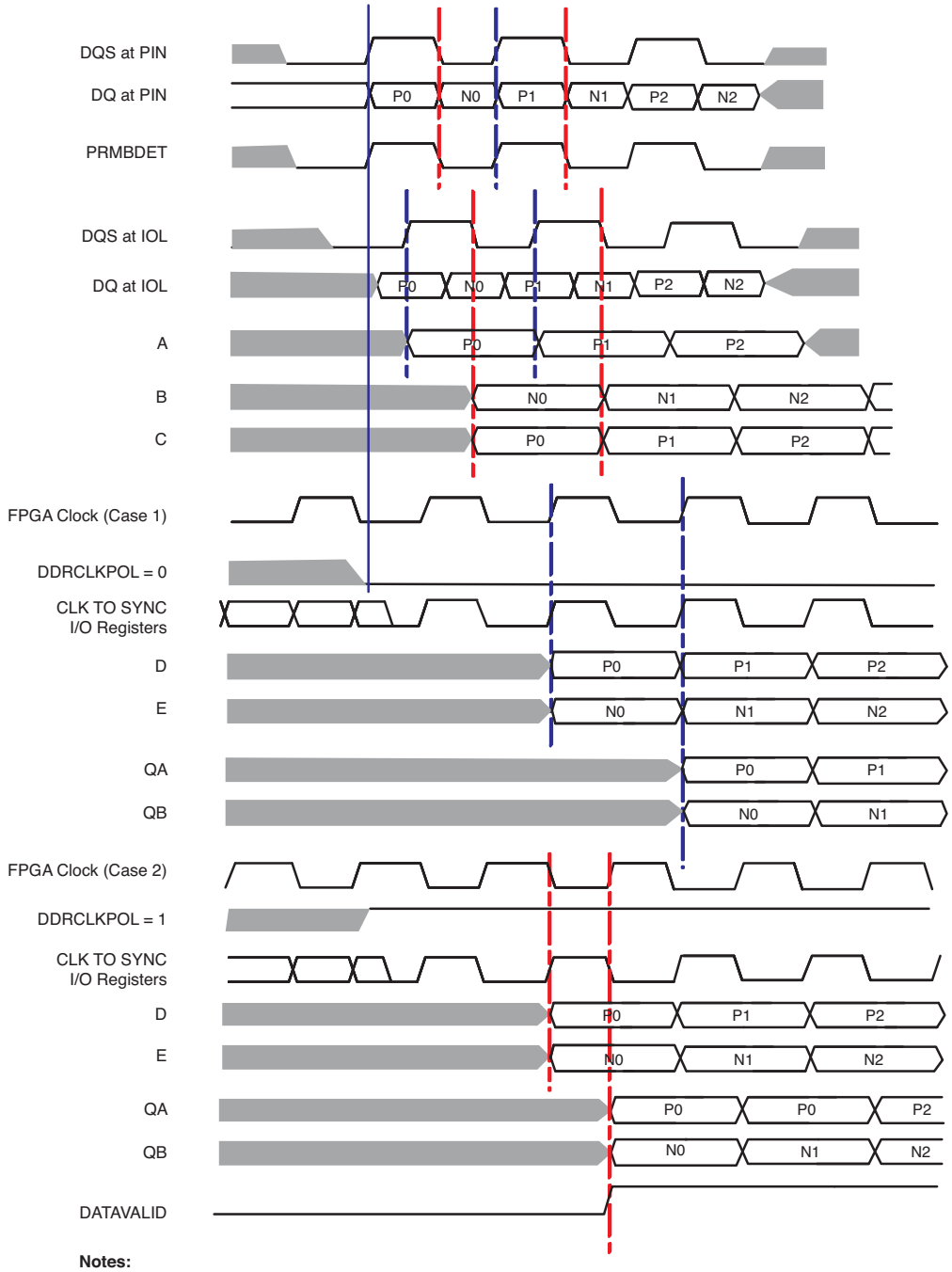
Figure 12-22 illustrates the DDR data timing using full clock transfer mode at different stages of IDDRMFX1A registers. In addition to the first two register stages in the half clock mode, the full clock transfer mode has an additional stage register clocked by the FPGA clock. In this case, D and E are the data streams after the second register stage presented to the final stage of registers clocked by the FPGA clock.

Figure 12-21. READ Data Transfer When Using IDDRMX1A



Notes:
1. DDR memory sends DQ aligned to DQS Strobe.
2. The DQS Strobe is delayed by 90 degrees using the dedicated DQS logic.
3. DQ is now center aligned to DQS Strobe.
4. PRMBDET is the Preamble detect signal generated using the DQSBUFB primitive. This is used to generate the DDRCLKPOL signal.
5. The first set of I/O registers, A and B, capture data on the positive and negative edges of DQS.
6. I/O Register C transfers data so that both data are now aligned to negative edge of DQS.
7. DDCLKPOL signal generated will determine if the FPGA CLK going into the synchronization registers need to be inverted. The DDRCLK-POL=0 when the FPGA CLK is LOW at the first rising edge of PRMBDET. The clock to the synchronization registers is not inverted. The DDRCLKPOL=1 when the FPGA CLK is HIGH at the first rising edge of PRMBDET. In this case the clock to the synchronization register is inverted.
8. The I/O synchronization registers capture data on either the rising or falling edge of the FPGA clock.
9. The DATAVALID signal goes HIGH when valid data enters the FPGA core. Once DATA VALID is asserted, it stays high until the next READ pulse.

Figure 12-22. Read Data Transfer When Using IDDRMFX1A



Notes:

1. DDR memory sends DQ aligned to DQS strobe.
2. The DQS strobe is delayed by 90 degress, using the dedicated DQS logic.
3. DQ is now center-aligned to the DQS strobe.
4. PRMBDET is the preamble detect signal generated usin the DQSBUFB primitive. This is used to generate the DDRCLKPOL signal.
5. The first set of I/O registers, A and B, capture data on the positive and negative edges of DQS.
6. I/O register C transfers data such that both data are aligned to the negative edge of DQS.
7. The DDCLKPOL signal generated will determine whether the FPGA clock going into the synchronization registers needs to be inverted. The DDRCLKPOL = 0 when the FPGA clock is LOW at the first rising edge of PRMBDET. So, the clock to the synchronization registers is not inverted. The DDRCLKPOL = 1 when the FPGA clock is HIGH at the first rising edge of PRMBDET. In this case, the clock to the synchronization register is inverted.
8. Registers D and E capture data at the FPGA clock.
9. The data again registers at the FPGA clock to ensure a full clock cycle transfer.
10. The DATAVALID signal goes HIGH when valid data enters the FPGA core. Once DATAVALID is asserted, it stays HIGH until the next READ pulse.

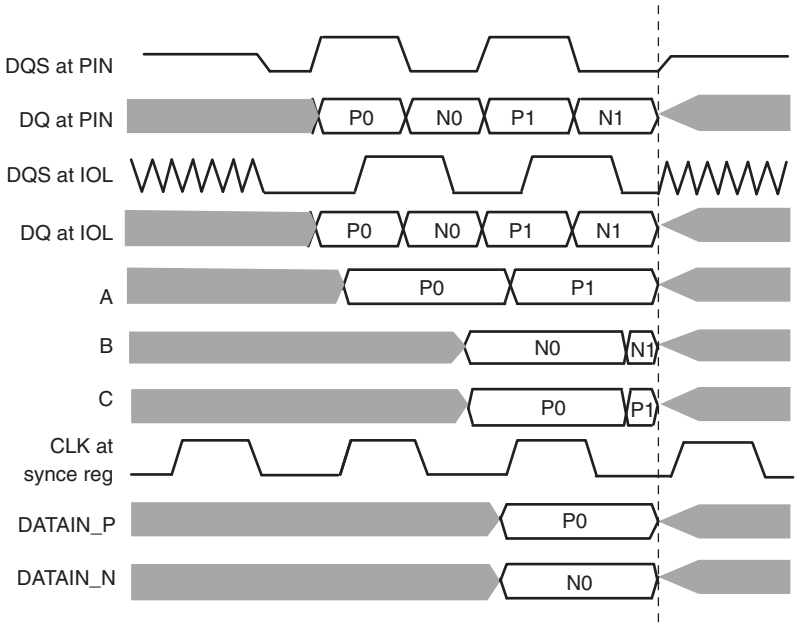
Data Read Critical Path

When using IDDRXM1A, the data in the second stage DDR registers can be registered either on the positive edge or on the falling edge of the FPGA clock depending on the DDRCLKPOL signal. In order to ensure that the data transferred to the FPGA core registers is aligned to the rising edge of the system clock, this path should be constrained with a half clock transfer. This half clock transfer can be forced in the software by assigning a multi-cycle constraint (multi-cycle of 0.5 X) on all the data paths to first PFU register. When using IDDRXMFX1A, there is an additional stage of registers inside the I/O block that transfers data to the positive edge of the FPGA clock. Hence no constraint is required for this case.

DQS Postamble

At the end of a READ cycle, the DDR SDRAM device executes the READ cycle postamble and then immediately tristates both the DQ and DQS output drivers. Since neither the memory controller (FPGA) nor the DDR SDRAM device are driving DQ or DQS at that time, these signals float to a level determined by the off-chip termination resistors. While these signals are floating, noise on the DQS strobe may be interpreted as a valid strobe signal by the FPGA input buffer. This can cause the last READ data captured in the IOL input DDR registers to be overwritten before the data has been transferred to the free running resynchronization registers inside the FPGA.

Figure 12-23. Postamble Effect on READ



LatticeECP2/M devices have extra dedicated logic in the in the DQS Delay Block that prevents this postamble problem. The DQS postamble logic is automatically implemented when the user instantiates the DQS Delay logic (DQSBUFC software primitive) in the design.

Memory Write Implementation

To implement the write portion of a DDR memory interface, two streams of single data rate data must be multiplexed together with data transitioning on both edges of the clock. In addition, during a write cycle, DQS must arrive at the memory pins center-aligned with the data, DQ. Along with the DQS strobe and data this portion of the interface must also provide the CLKP, CLKN Address/Command and Data Mask (DM) signals to the memory.

It is the responsibility of the FPGA output control to edge-align the DDR output signals (ADDR,CMD, DQS, but not DQ, DM) to the rising edge of the outgoing differential clock (CLKP/CLKN).

Challenges encountered by the during Memory WRITE:

1. DQS needs to be center-aligned with the outgoing DDR Data, DQ.

2. Differential CLK signals (CLKP and CLKN) need to be generated.
3. The controller must meet the DDR interface specification for t_{DSS} and t_{DSH} parameters, defined as DQS falling to CLKP rising setup and hold times.
4. The DDR output data must be muxed from two SDR streams into a single outgoing DDR data stream.

All DDR output signals ("ADDR, CMD", DQS, DQ, DM) are initially aligned to the rising edge of the FPGA clock inside the FPGA core. The relative phase of the signals may be adjusted in the IOL logic before departing the FPGA. These adjustments are shown in Figure 12-24.

LatticeECP2/M devices contain DDR output and tri-state registers along with the DQSXFER signal generated by the DQSBUFC that allows easy implementation of the write portion of the DDR memory interfaces. The DDR output registers can be accessed in the design tools via the ODDRMXA and the ODDRXC primitives.

The DQS signal and the DDR clock outputs are generated using the ODDRXC primitive. As shown in the figure, the CLKP and DQS signals are generated so that they are 180 degrees in phase with the clock. This is done by connecting "1" to the DA input and "0" to the DB inputs of the ODDRXC primitive. Refer to the DDR Generic Software Primitive section of this document to see the ODDRXC timing waveforms.

The DDR clock output is then fed into a SSTL differential output buffer to generate CLKP and CLKN differential clocks. Generating the CLKN in this manner prevents any skew between the two signals. When interfacing to DDR1, SDRAM memory CLKP should be connected to the SSTL25D I/O standard. When interfacing to DDR2 memory, it should be connected to the SSTL18D I/O standard.

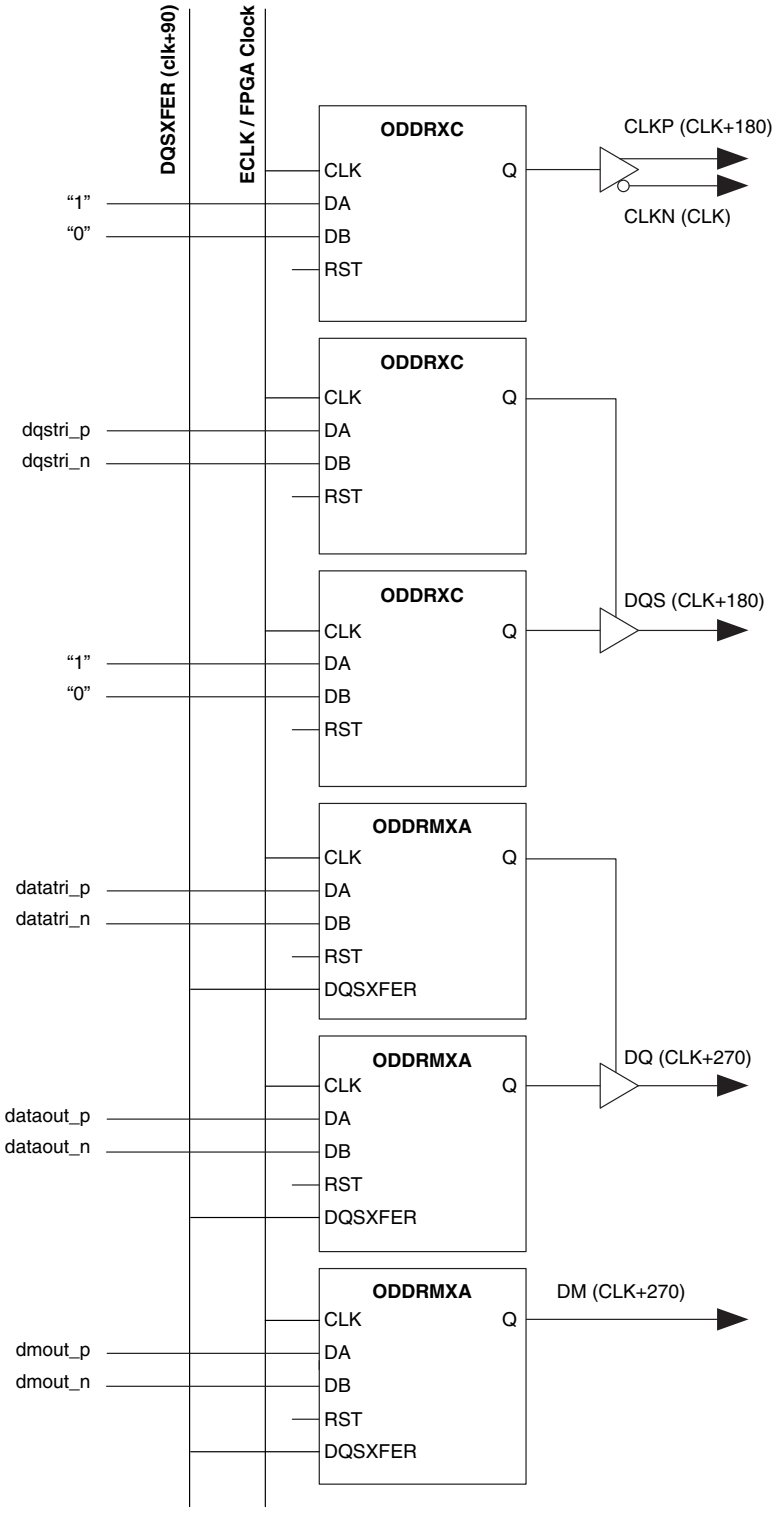
The DQSXFER output from the DQSBUFC block is the 90-degree phase shifted clock. This 90-degree phase shifted clock is used as an input to the ODDRMXA block. The ODDRMXA is used to generate the DQ and DM data outputs going to the memory. In the ODDRMXA module, the data is first registered using the ECLK or FPGA clock input and then shifted out using the DQSXFER signal. To ensure that the data going to the memory is center-aligned to the DQS, the DQSXFER is inverted inside the ODDRXMA primitive. This will generate data that is center-aligned to the DQS. Refer to the Software Primitives section of this document for the ODDRXMA timing waveforms.

The DDR interface specification for t_{DSS} and t_{DSH} parameters defined as DQS falling to CLKP rising setup and hold times must be met. This is accomplished by ensuring that the CLKP and DQS signals are identical in phase.

The tristate control for the DQS and DQ outputs can also be implemented using the ODDRXC primitive.

Figure 12-24 shows the DDR Write implementation using the DDR primitives.

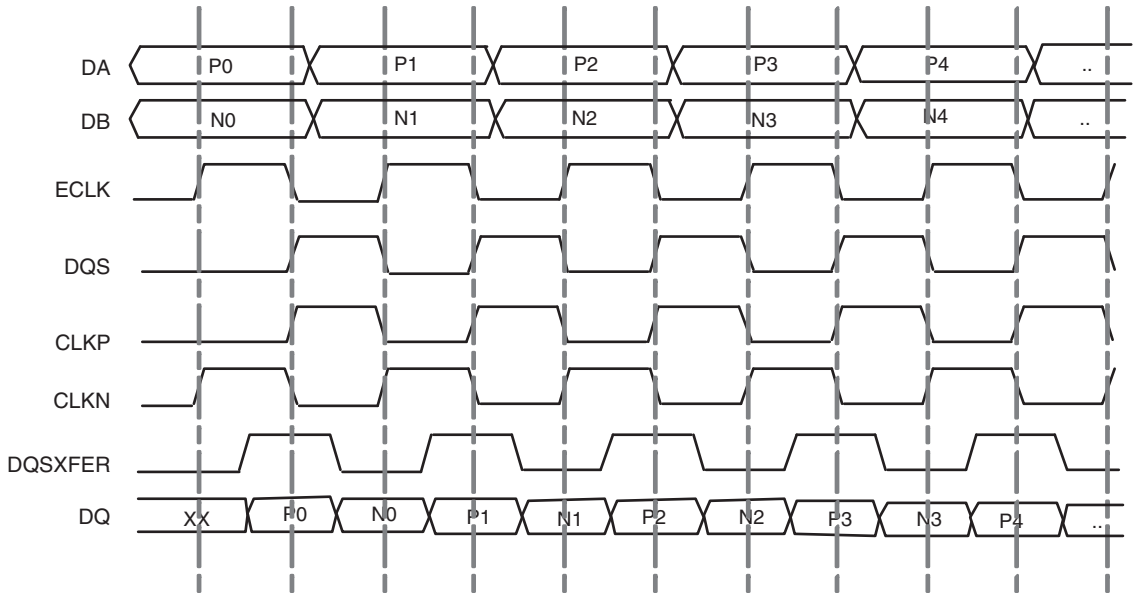
Figure 12-24. Software Implementation for Memory Write



Write Timing Waveforms

Figure 12-25 shows the DDR write side data transfer timing for the DQ Data pad and the DQS Strobe Pad. When writing to the DDR memory device, the DM (Data Mask) and the ADDR/ CMD (Address and Command) signals are also sent to the memory device along with the data and strobe signals.

Figure 12-25. DDR Write Data Transfer for DQ Data



Design Rules/Guidelines

Listed below are some rules and guidelines to keep in mind when implementing DDR memory interfaces in the LatticeECP2/M devices.

- The LatticeECP2/M devices have dedicated DQ-DQS banks. Please refer to the logical signal connections of the groups in the LatticeECP2/M Family Data Sheet before locking these pins.
- There are two DQSDLL on the device, one for the left half and one for the right half of the device. Therefore, only one DQSDLL primitive should be instantiated for each half of the device. Since there is only one DQSDLL on each half of the device, all the DDR memory interfaces on that half of the device should run at the same frequency. Each of the DQSDLL will generate 90-degree digital delay bits for all the DQS delay blocks on that half of the device based on the reference clock input to the DLL.
- When implementing a DDR SDRAM interface, all interface signals should be connected to the SSTL25 I/O standard. In the case of the DDR2 SDRAM interface, the interface signal should be connected to SSTL18 I/O standard.
- For DDR2, the differential DQS signals need to be connected to SSTL18 the Differential I/O standard.
- When implementing the DDR interface, the VREF1 of the bank is used to provide the reference voltage for the interface pins.

Generic High Speed DDR Implementation

In addition to the DDR memory interface, the I/O logic DDR registers can be used to implement high speed DDR interfaces. The Input DDR registers can operate in full clock transfer and half clock transfer modes. The DDR input and output register also support x1 and x2 gearing ratios. A gearing capability is provided to Mux/DeMux the I/O data rate (ECLK) to the FPGA clock rate (SCLK). For DDR interfaces, this ratio is slightly different than the SDR ratio. A basic 2x DDR element provides four FPGA side bits for two I/O side bits at half the clock rate on the FPGA side.

The data going to the DDR registers can be optionally delayed before going to the DDR register block.

Generic DDR Software Primitives

The IPexpress tool in the ispLEVER software can be used to generate the DDR modules. The various DDR modes described below can be configured in the IPexpress tool. The various modes are implemented using the following software primitives.

- IDDRXC – DDR Generic Input
- IDDRFXA – DDR Generic Input with full clock transfer (x1 gearbox)
- IDDRX2B – DDR Generic Input with 2x gearing ratio. DDRX2 inputs a double data rate signal as four data streams. Two stages of DDR registers are used to convert serial DDR data at input pad into four SDR data streams entering FPGA core logic.
- ODDRXC – DDR Generic Output
- ODDRX2B – DDR Generic Output with 2x gearing ratio. The DDRX2 inputs four separate data streams and outputs a single data stream to the I/O buffer.
- DELAYB – The DDR input can be optionally delayed before it is input to the DDR registers. The user can choose to implement a fixed delay value or use a dynamic delay.

IDDRXC

This primitive inputs DDR data at both edges of the CLK and generates two streams of data. The CLK to this module can be connected to either the edge clock or the primary FPGA clock.

Figure 12-26 shows the primitive symbol for IDDRXC mode.

Figure 12-26. IDDRXC Symbol

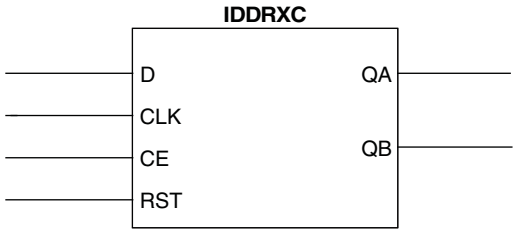


Table 12-7 lists the port names and descriptions for the IDDRXC primitive.

Table 12-7. IDDRXC Port Names

Port Name	I/O	Definition
D	I	DDR data
CLK	I	This clock can be connected to the ECLK or the FPGA clock
CE	I	Clock enable signal
RST	I	Reset to the DDR register
QA	O	Data at the positive edge of the clock
QB	O	Data at the negative edge of the clock

Figure 12-27 shows the LatticeECP2 Input Register Block configured in the IDDRXFC mode.

Figure 12-27. Input Register Block Configured as IDDRXC

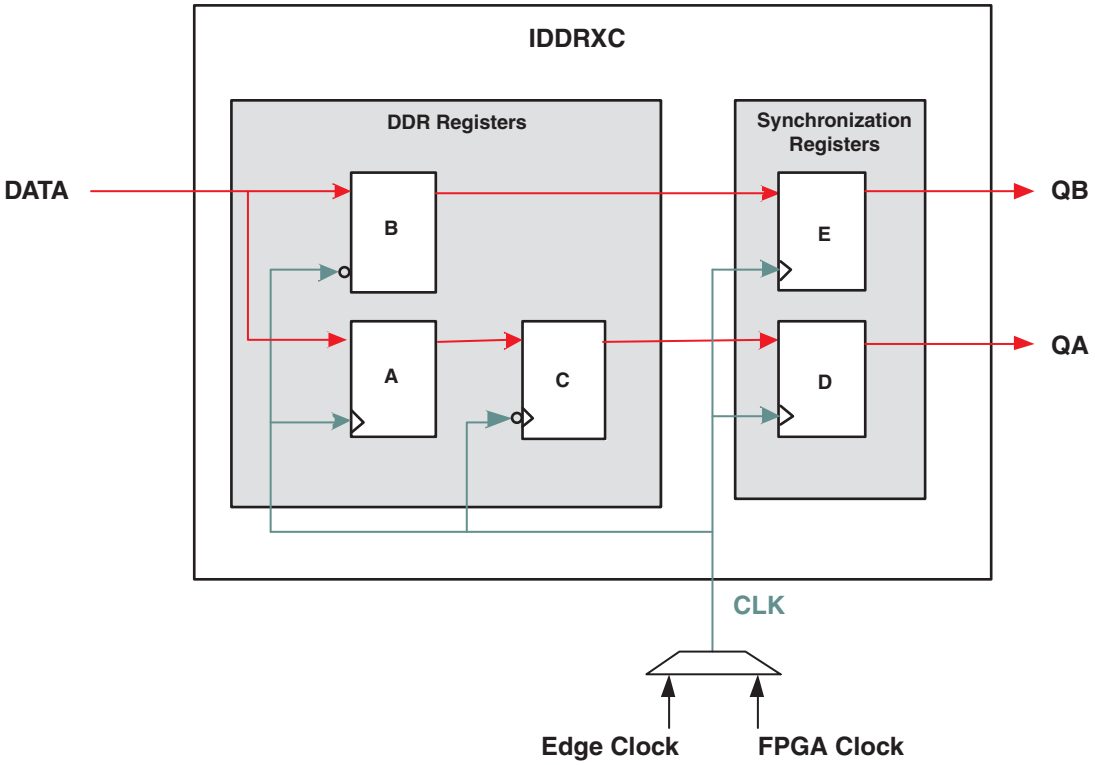
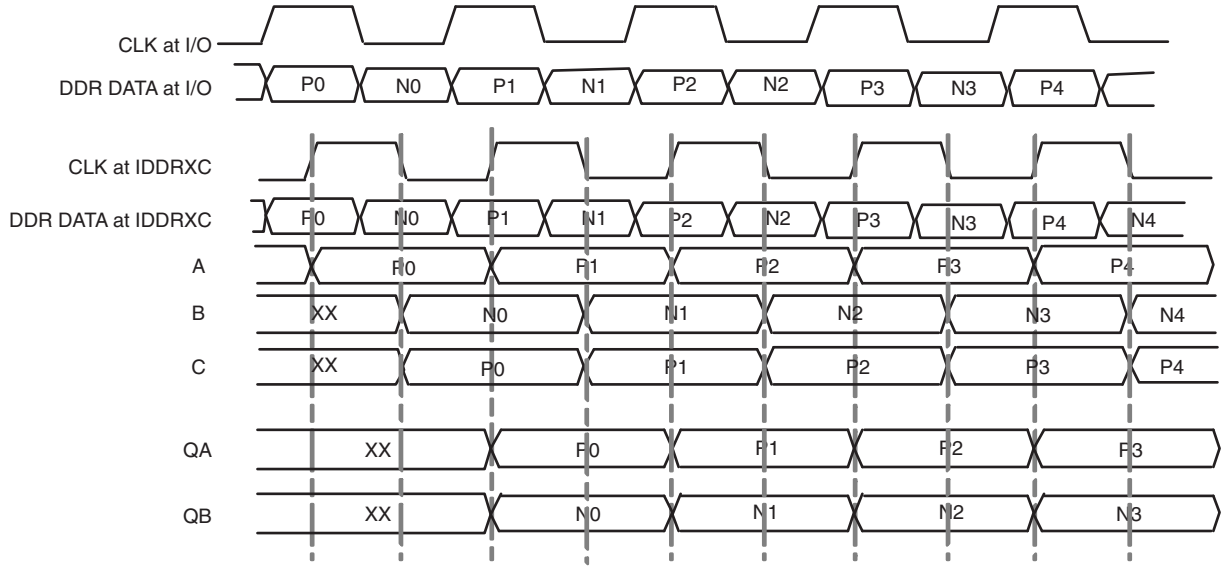


Figure 12-28 shows the timing waveform when using the IDDRXC module.

Figure 12-28. IDDRXC Waveform



IDDRFXA

This primitive inputs DDR data at both edges of clock CLK1 and generates two streams of data aligned to clock CLK2. CLK1 can be connected either to the edge clock or the internal FPGA clock. If the Edge clock input is used for CLK1 then CLK2 should be generated from the same clock going to CLK1.

Figure 12-29 shows the primitive symbol for the IDDRFXA mode.

Figure 12-29. IDDRFXA Symbol

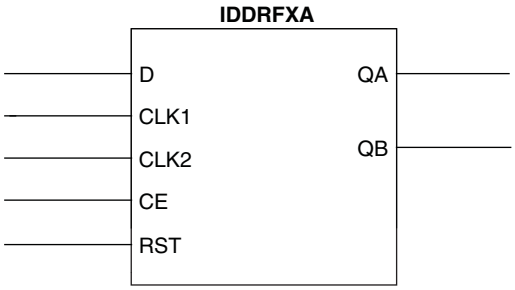


Table 12-8 lists the port names and descriptions for the IDDRFXA primitive.

Table 12-8. IDDRFXA Port Names

Port Name	I/O	Description
D	I	DDR data
CLK1	I	This clock can be connected to the ECLK or the FPGA clock
CLK2	I	This clock should be connected to the FPGA clock
CE	I	Clock Enable signal
RST	I	Reset to the DDR register
QA	O	Data at the positive edge of the clock
QB	O	Data at the negative edge of the clock

Figure 12-31 shows the LatticeECP2 Input Register Block configured in the IDDRXFXA mode. CLK1 used to register the DDR registers and the first set of synchronization registers. CLK2 is used by the third stage of registers and should be clocked by the FPGA clock. These clock transfer registers are shared with the output register block.

Figure 12-30. Input Register Block configured as IDDRFXA

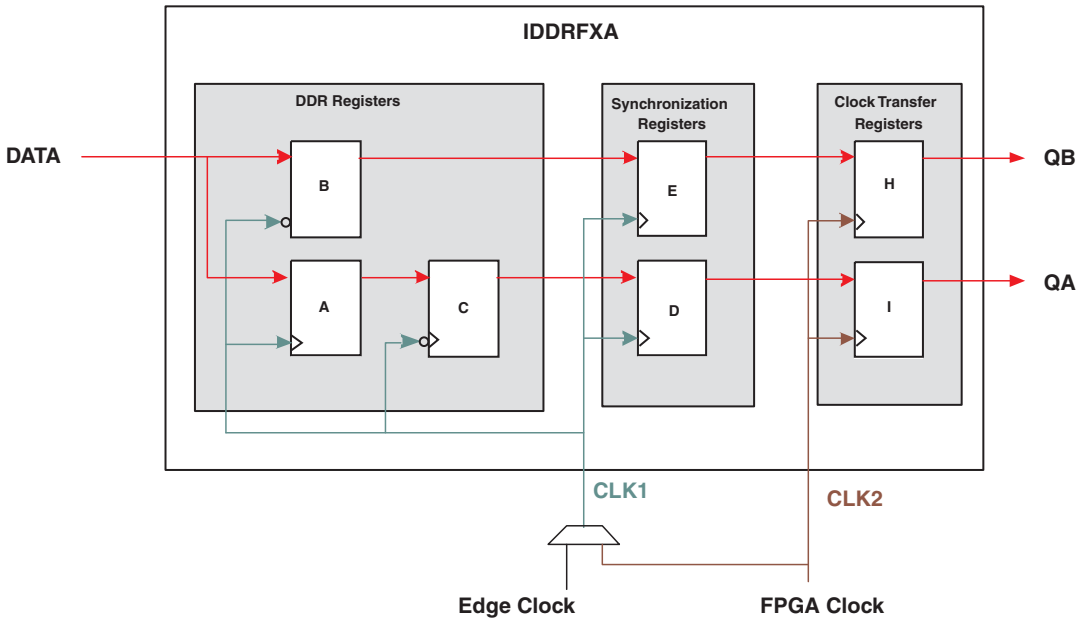
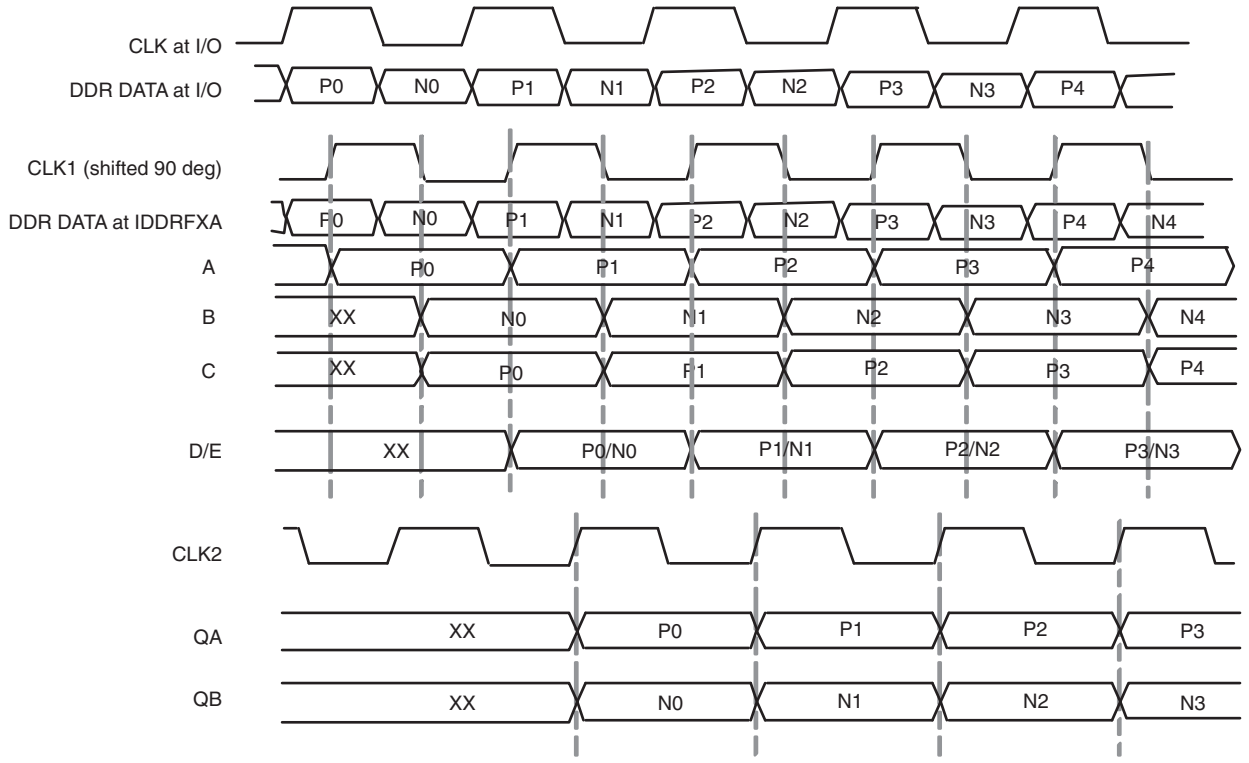


Figure 12-31 shows the timing waveform when using the IDDRFXA module.

Figure 12-31. IDDRFXA Waveform



IDDRX2B

This module is used when a gearing function is required. This primitive inputs the DDR data at both edges of the edge clock and generates four streams of data aligned to SCLK. SCLK is always half the frequency of ECLK. It is recommended that the CLKDIV module or PLL be used to generate the SCLK from the ECLK.

Figure 12-32 shows the primitive symbol for the IDDRX2B mode.

Figure 12-32. IDDRX2B Symbol

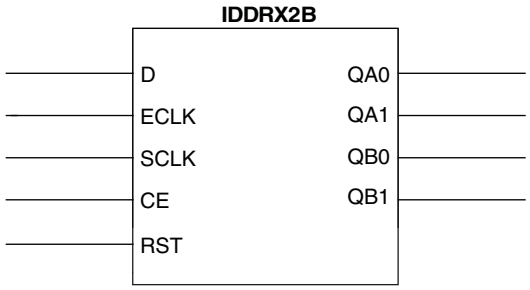


Table 12-9 lists the port names and descriptions for the IDDRX2B primitive.

Table 12-9. IDDRX2B Port Names

Port Name	I/O	Description
D	I	DDR data
ECLK	I	This clock can be connected to the fast edge clock
SCLK	I	This clock should be connected to the FPGA clock
CE	I	Clock enable signal
RST	I	Reset to the DDR register
QA0, QA1	O	Data at the positive edge of the clock
QB0, QB1	O	Data at the negative edge of the clock

Figure 12-33 shows the LatticeECP2 Input Register Block configured in the IDDRX2B mode. The DDR registers and the first set of synchronization registers are clocked by the ECLK input. The SCLK is used to clock the third stage of register. This primitive will output four streams of data. The 2x gearing function is implemented by using the synchronization registers of the complementary PIO. The clock transfer registers are shared with the output register block.

Figure 12-33. Input Register Block Configured as IDDRX2B

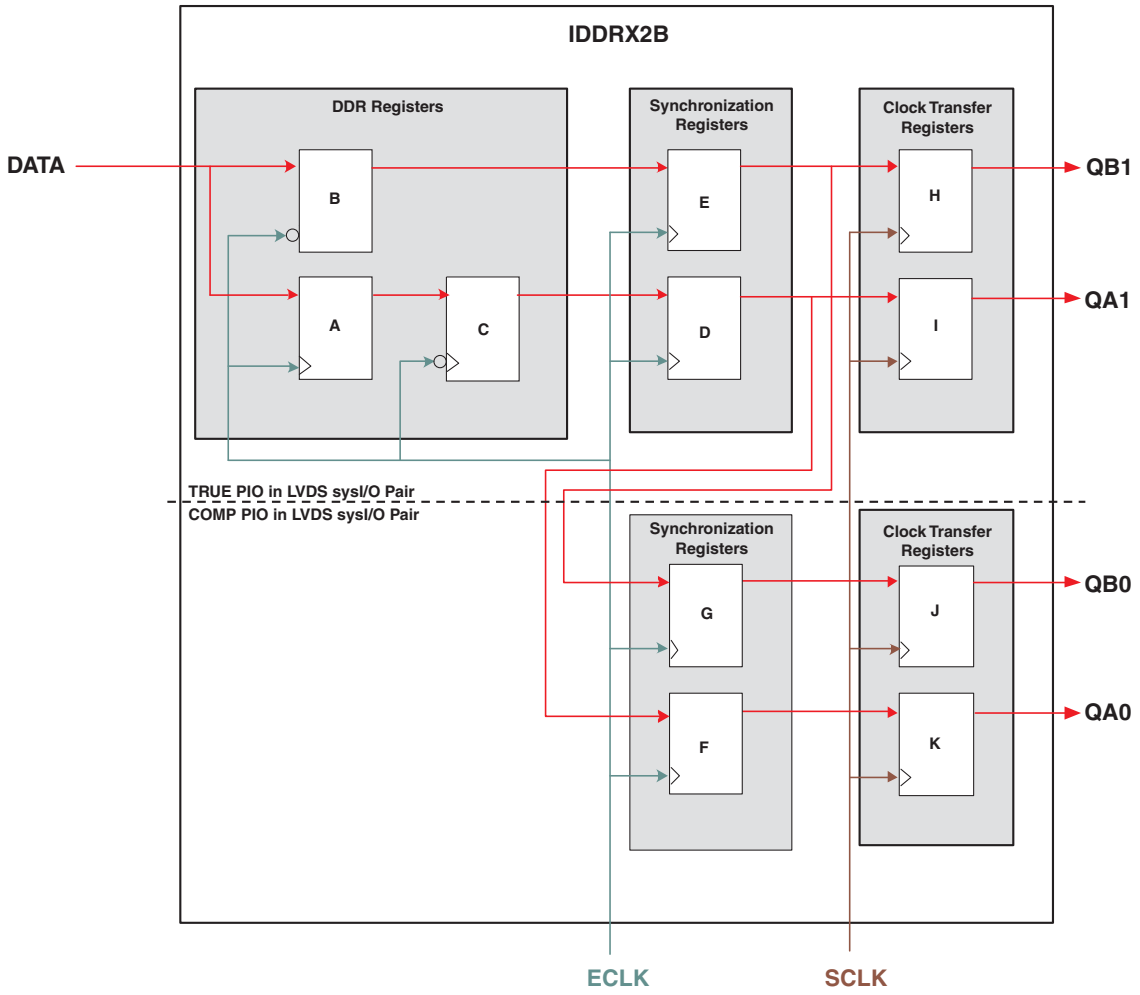
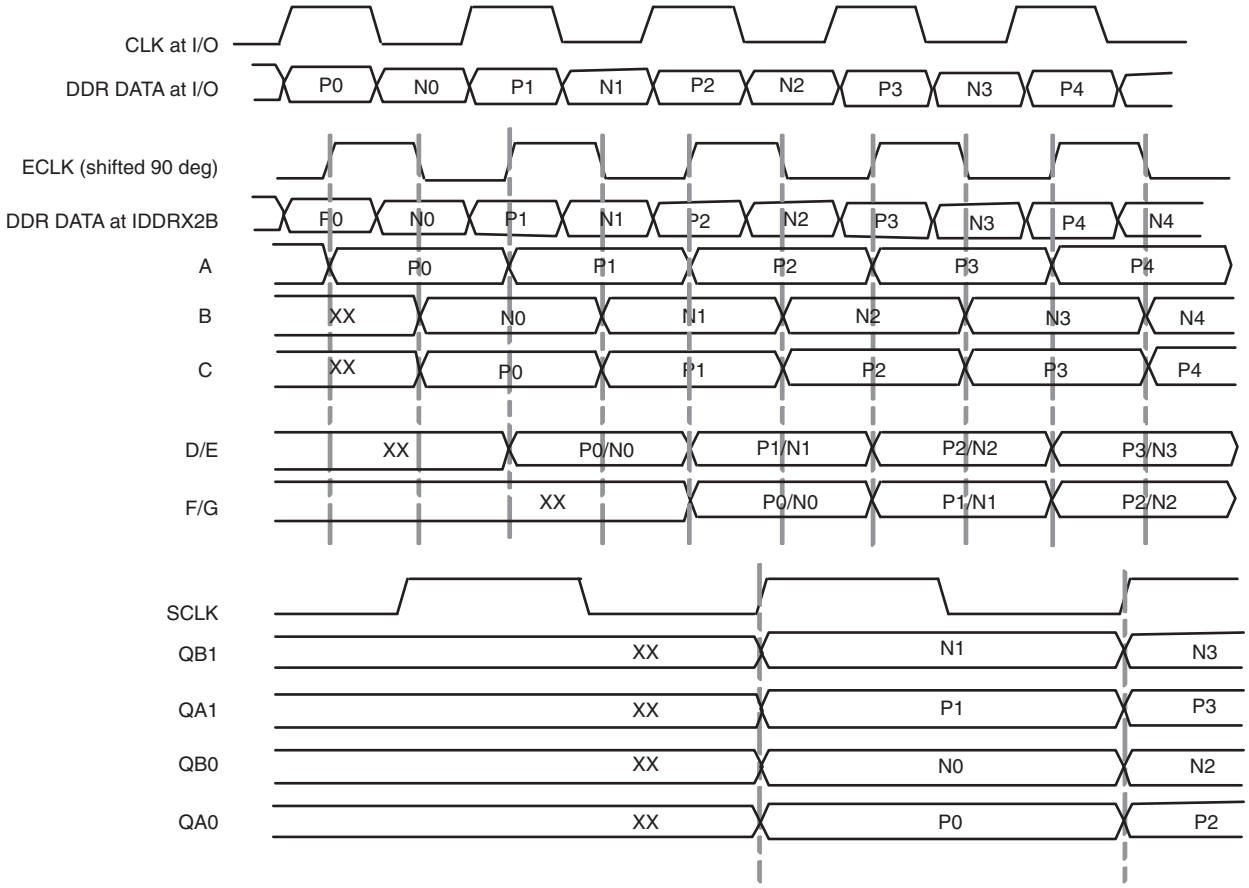


Figure 12-34 shows the timing waveform using the IDDRX2B module.

Figure 12-34. IDDRX2B Waveform



ODDRXC

This is the DDR output module. This primitive will input two data streams and mux them together to generate a single stream of data going to the sysIO™ buffer. The CLK to this module can be connected to the edge clock or to the FPGA clock. This primitive is also used for when DDR function is required for the tristate signal.

Figure 12-35 shows the primitive symbol for the ODDRXC mode.

Figure 12-35. ODDRXC Symbol

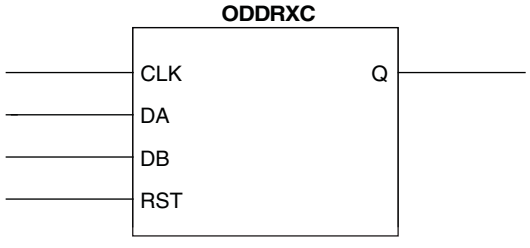


Table 12-10 lists the port names and descriptions for the ODDRXC primitive.

Table 12-10. ODDRXC Port Names

Port Name	I/O	Definition
DA	I	Data at the negative edge of the clock
DB	I	Data at the positive edge of the clock
CLK	I	This clock can be connected to the edge clock or to the FPGA clock
RST	I	Reset signal
Q	O	DDR data output

Figure 12-36 shows the Output Register Block of the LatticeECP2 device configured in ODDRXC mode.

Figure 12-36. Output Register Block in ODDRC Mode

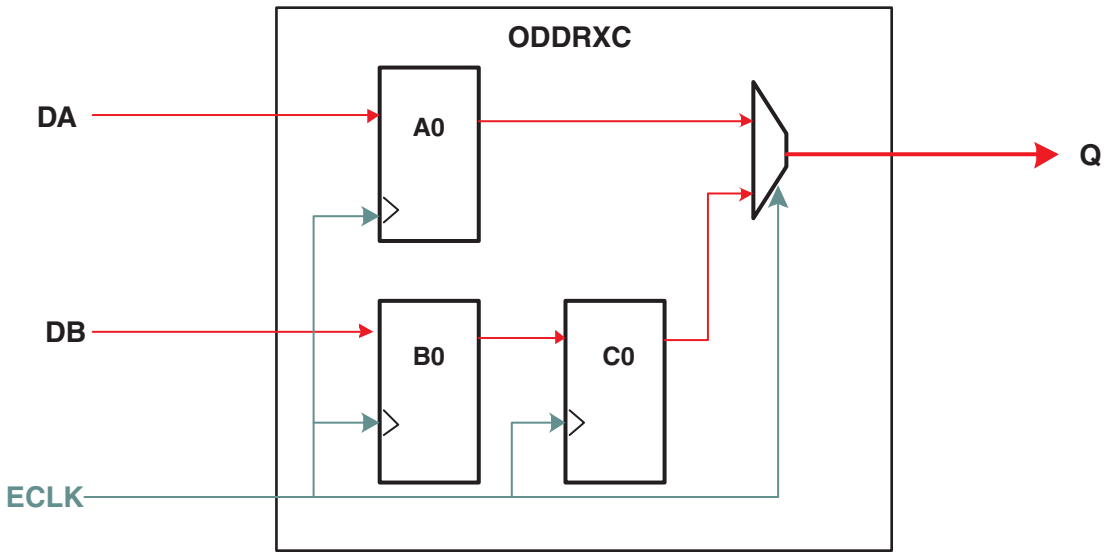
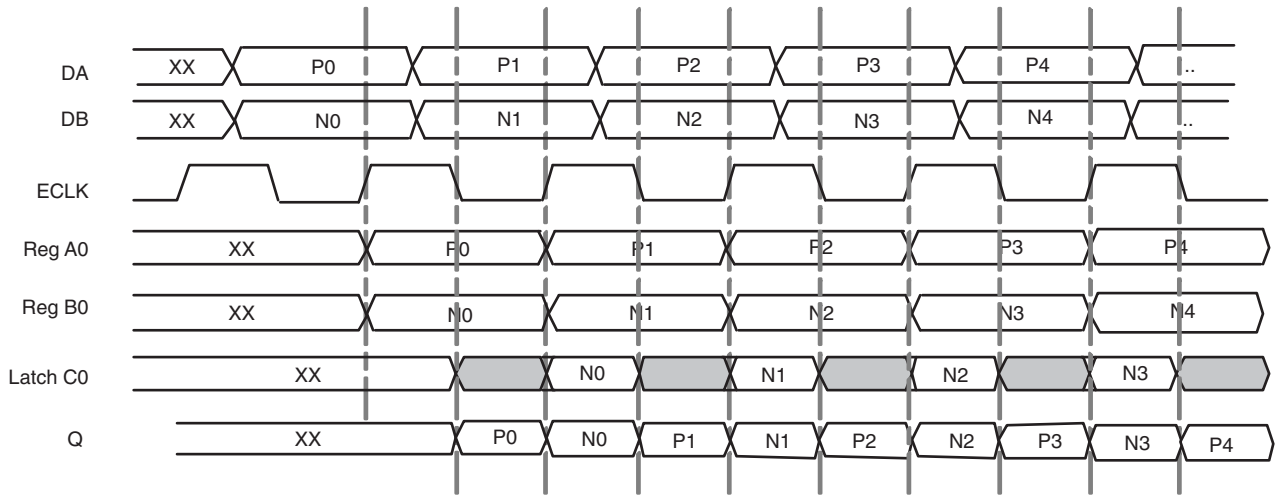


Figure 12-37 shows the timing waveform when using the ODDRXC module.

Figure 12-37. ODDRXC Waveform



ODDRX2B

This DDR output module can be used when a gearbox function is required. This primitive inputs four data streams and muxes them together to generate a single stream of data going to the sysIO buffer.

DDR registers of the complementary PIO are used in this mode. The complementary PIO register can no longer be used to perform the DDR function. There are two clocks going to this primitive. The ECLK is connected to the faster edge clock and the SCLK is connected to the slower FPGA clock. The DDR data output of this primitive is aligned to the faster edge clock.

Figure 12-38 shows the primitive symbol for the ODDRX2B mode.

Figure 12-38. ODDRX2B Symbol

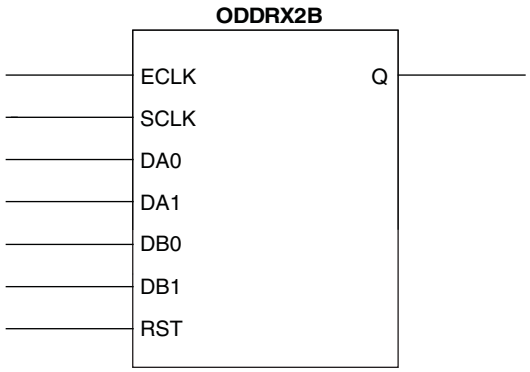


Table 12-11 lists the port names and descriptions for the ODDRX2B primitive.

Table 12-11. ODDRX2B Port Names

Port Name	I/O	Description
DA0, DB0	I	Data at the negative edge of the clock
DA1, DB1	I	Data at the positive edge of the clock
ECLK	I	This clock should be connected to the faster edge clock
SCLK	I	This clock should be connected to the slower FPGA clock
RST	I	Reset signal
Q	O	DDR data output

Figure 12-39 shows the LatticeECP2 Output Register Block in the ODDRX2B mode.

Figure 12-39. Output Register Block Configured in ODDRX2B Mode

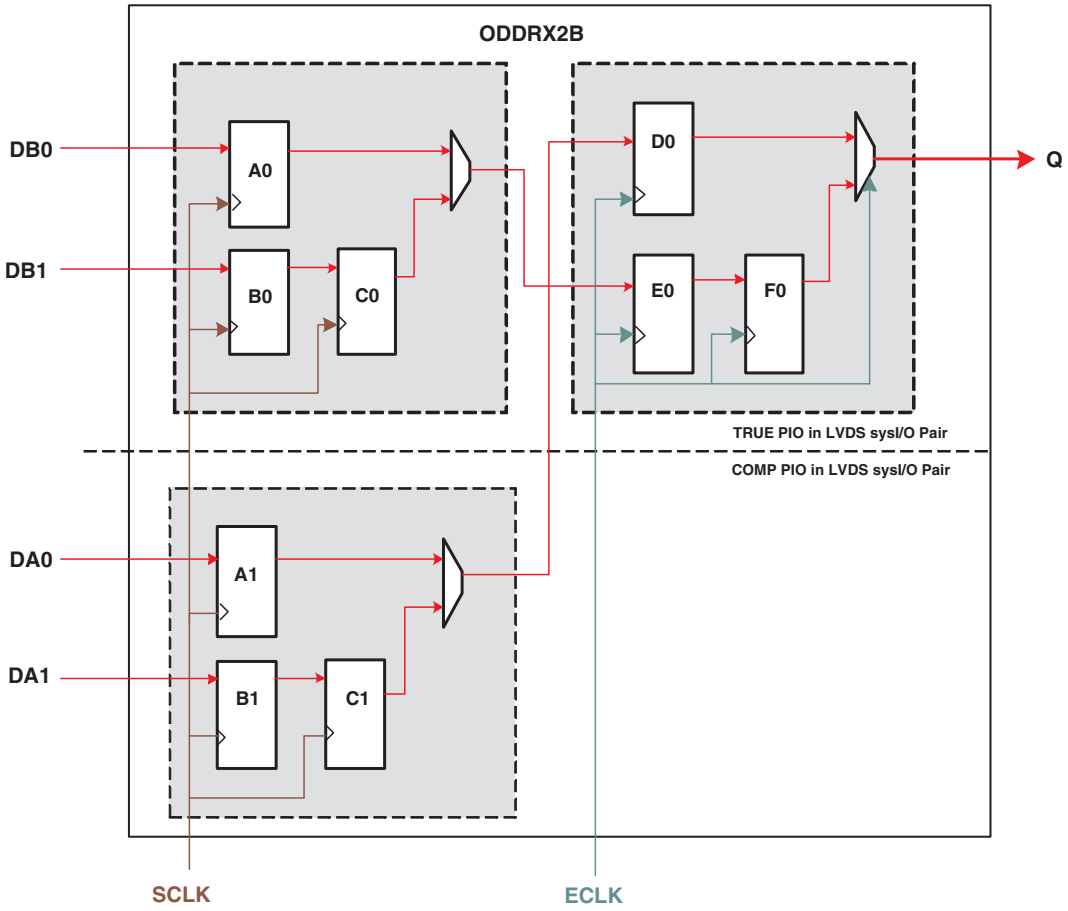
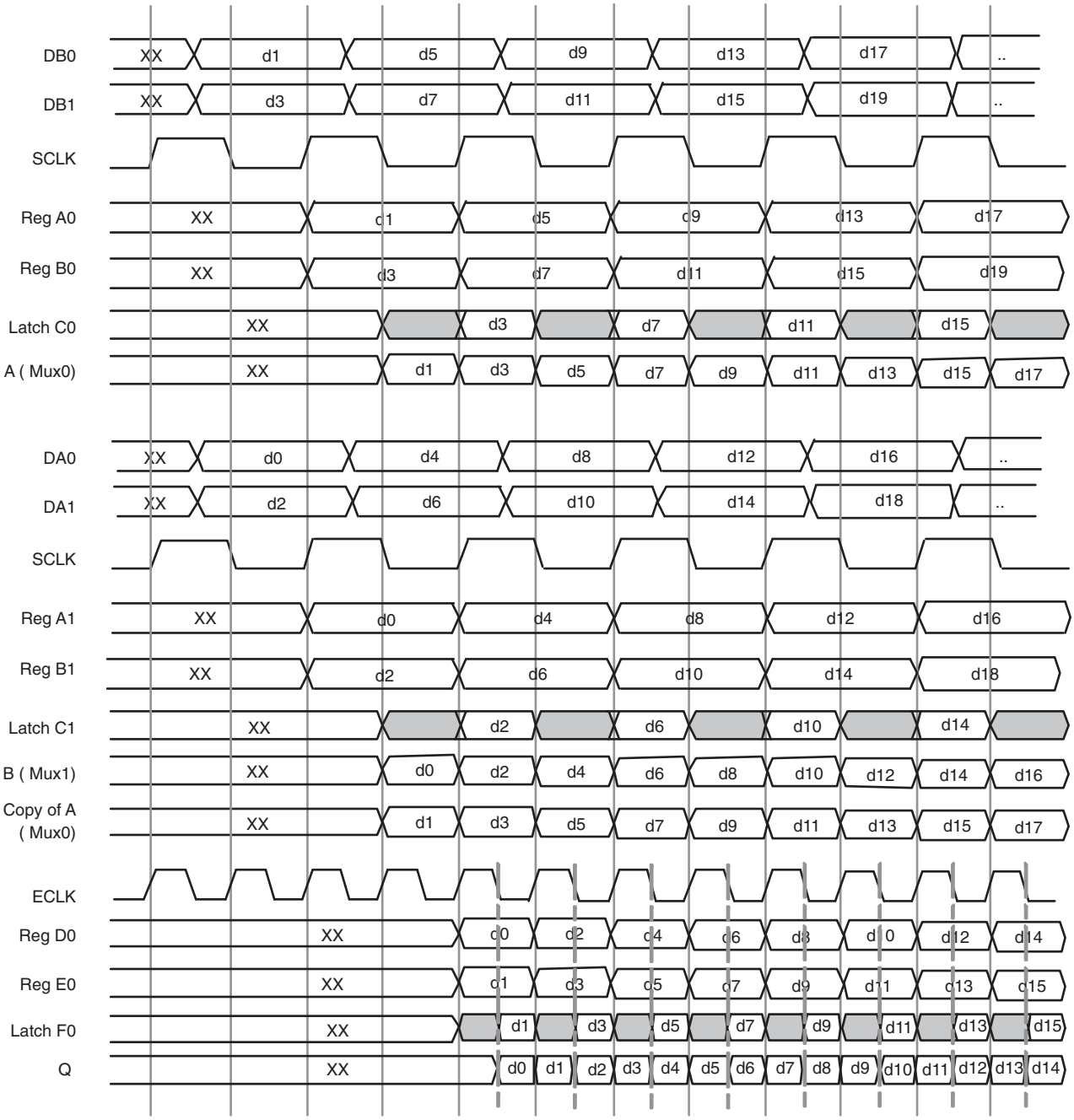


Figure 12-40 shows the timing waveform when using the ODDRXC module.

Figure 12-40. ODDRX2B Waveform



DELAYB

Data going to the DDR registers can be optionally delayed using the Delay block. The Delay block receives 4-bit delay control. The 4-bit delay can be set using fixed multiplier values or it can be controlled by the user. The DELAYB block is available for use with the input DDR registers.

The DELAYB block can be configured when generating the DDR input modules in the IPexpress tool of the software. The delay can be adjusted in 35ps steps. Users can choose from three types of delay values:

1. Dynamic – The delay value is controlled by the user logic using the DEL[3:0] input of the DELAYB block.

2. Fixed – When choosing the fixed value, the user will also need to choose from one of the 16 multiplier values. This will tie the inputs DEL[3:0] of the DELAYB block to a fixed value depending on the multiplier value chosen.
3. FIXED_XGMII – The DEL [3:0] will be configured with the delay value required when implementing a XGMII interface.

Figure 12-41 shows the primitive symbol for the DELAYB mode.

Figure 12-41. DELAYB Symbol

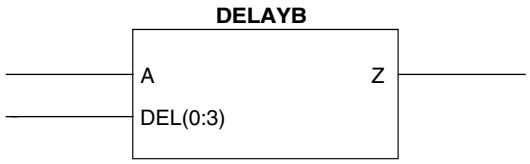


Table 12-12 lists the port names and descriptions for the DELAYB primitive.

Table 12-12. DELAYB Port Names

Port Name	I/O	Definition
A	I	DDR input from the sysIO buffer
DEL (0:3)	I	Delay inputs
Z	O	Delay DDR data

Design Rules/Guidelines

Listed below are some rules and guidelines for implementing generic DDR interfaces in LatticeECP2/M devices.

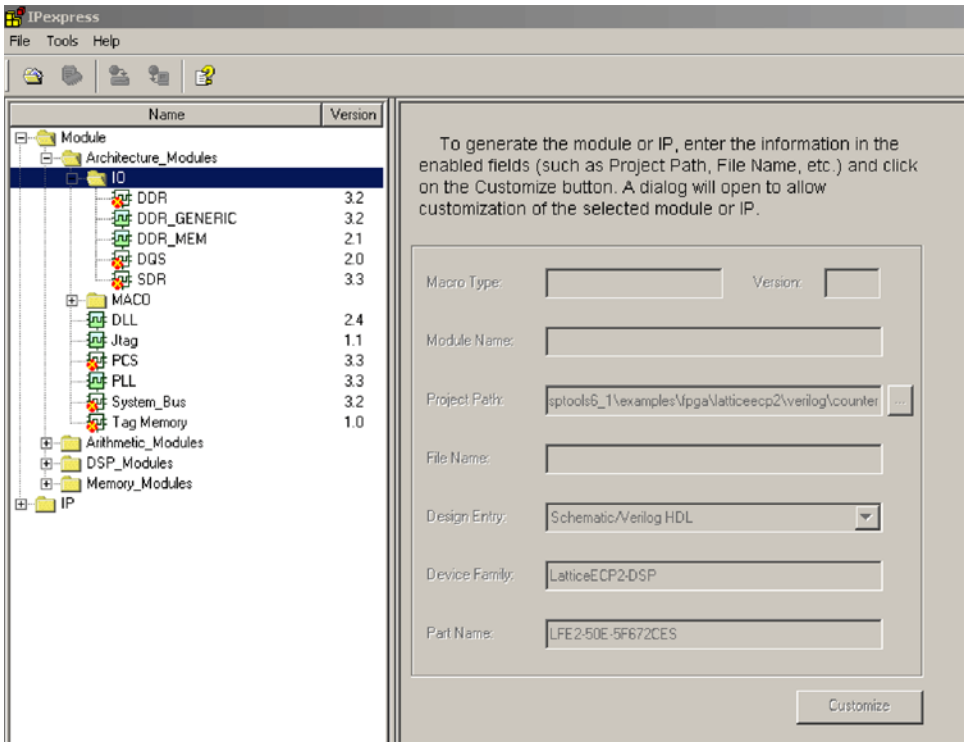
- When implementing a 2x gearing mode, the complement PIO registers are used. This complementary PIO register can no longer be used and should not be connected.
- DDR registers are available on the Left, Right and Bottom sides of the device. The top side does not support DDR registers.

DDR Usage in ispLEVER IPexpress

This section describes how IPexpress in ispLEVER is used to generate the DDR modules. If you are using Lattice Diamond™ design software, refer to Appendix A to see how DDR modules are generated in Diamond. IPexpress can be used to configure and generate the DDR Memory Interface and Generic DDR Module. The tool will generate an HDL module that will contain the DDR primitives. This module can be using in the top level design.

Figure 12-42 shows the main window of IPexpress. The DDR_Generic and DDR_MEM options under **Architecture->IO** are used to configure the DDR modules.

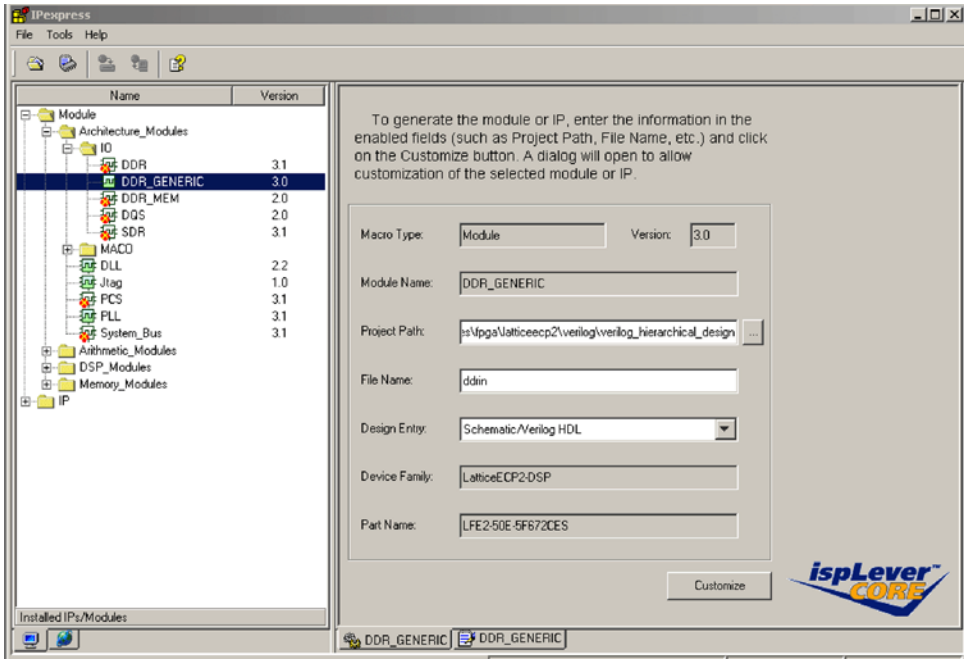
Figure 12-42. IPexpress Main Window



DDR Generic

Figure 12-43 shows the main window when DDR_Generic is selected. The only entry required in this window is the module name. Other entries are set to the project settings. The user may change these entries if desired. After entering the module name, click on **Customize** to open the **Configuration Tab** window as shown in Figure 12-44.

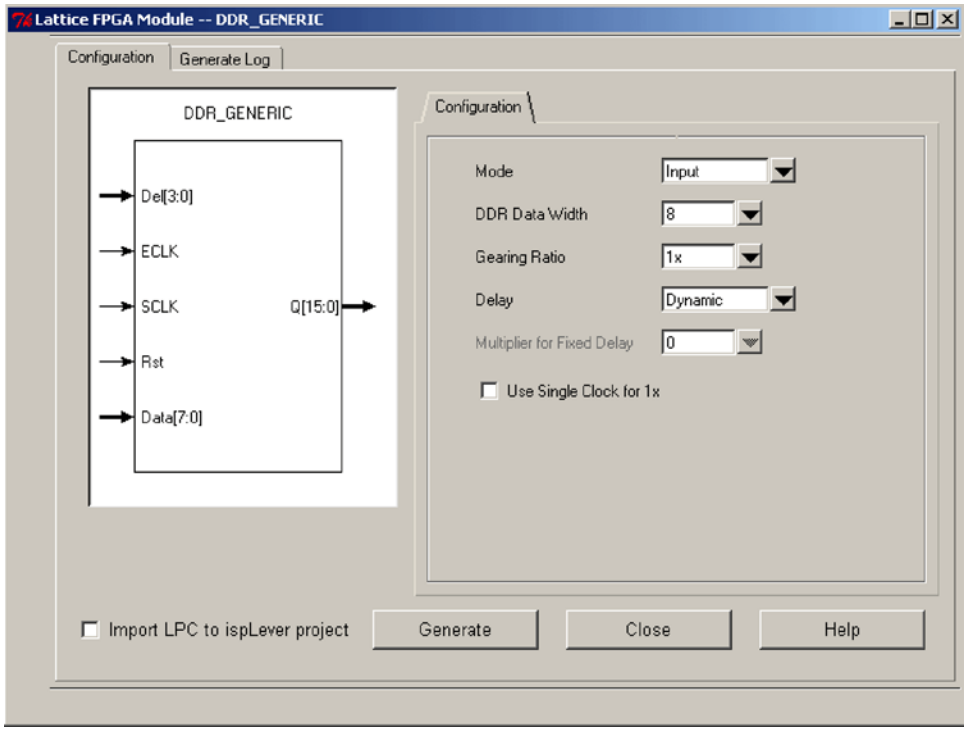
Figure 12-43. IPexpress Main Window for DDR_Generic



Configuration Tab

The Configuration Tab lists all user-accessible attributes with default values set. Upon completion, click **Generate** to generate source and constraint files. The user may choose to use the .lpc file to load parameters.

Figure 12-44. Configuration Tab for DDR_Generic



The user can change the Mode parameter to choose either Input, Output, Bidirection or Tristate DDR module. The other configuration parameters will change according to the mode selected. The Delay parameter is only available for Input and Bidirectional modes. Similarly the Multiplier for Fixed Delay parameter is only available when the Delay parameter is configured to Fixed.

Table 12-13. User Parameters in the IPexpress GUI

User Parameters	Description	Values/Range	Default
Mode	Mode selection for the DDR block.	Input, Output, Bidirectional, Tristate	Input
Data Width	Width of the data bus.	1-64	8
Gearing Ratio	Gearing ratio selection.	1x, 2x[1]	1x
Delay	Input delay configuration.	Dynamic, Fixed, Fixed XGMII	Dynamic
Multiplier for Fixed Delay	Fixed delay setting. Available only when delay is configured as Fixed.	0-15	0
Use Single Clk for 1x	Allows the selection of a single clock for the gearing logic.	On/Off	Off

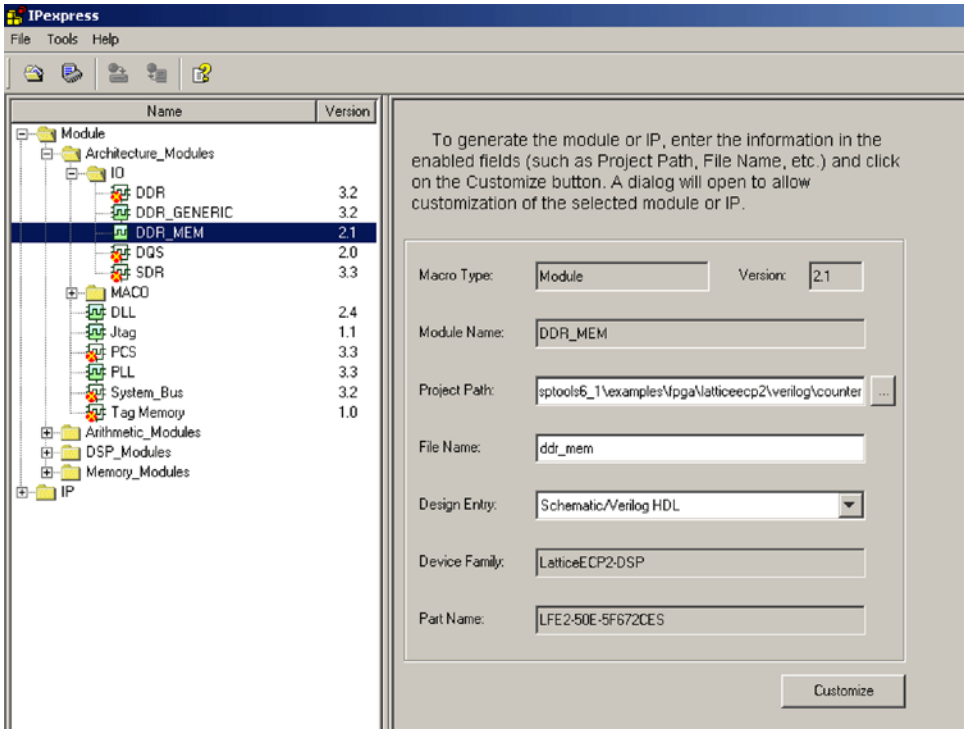
1. Only 1x available when Mode is Bidirection or Tristate.

DDR_MEM

Figure 12-45 shows the main window when DDR_MEM is selected. Similar to the DDR_Generic, the only entry required here is the module name. Other entries are set to the project settings. The user may change these entries

if desired. After entering the module name, click on **Customize** to open the **Configuration Tab** window as shown in Figure 12-46.

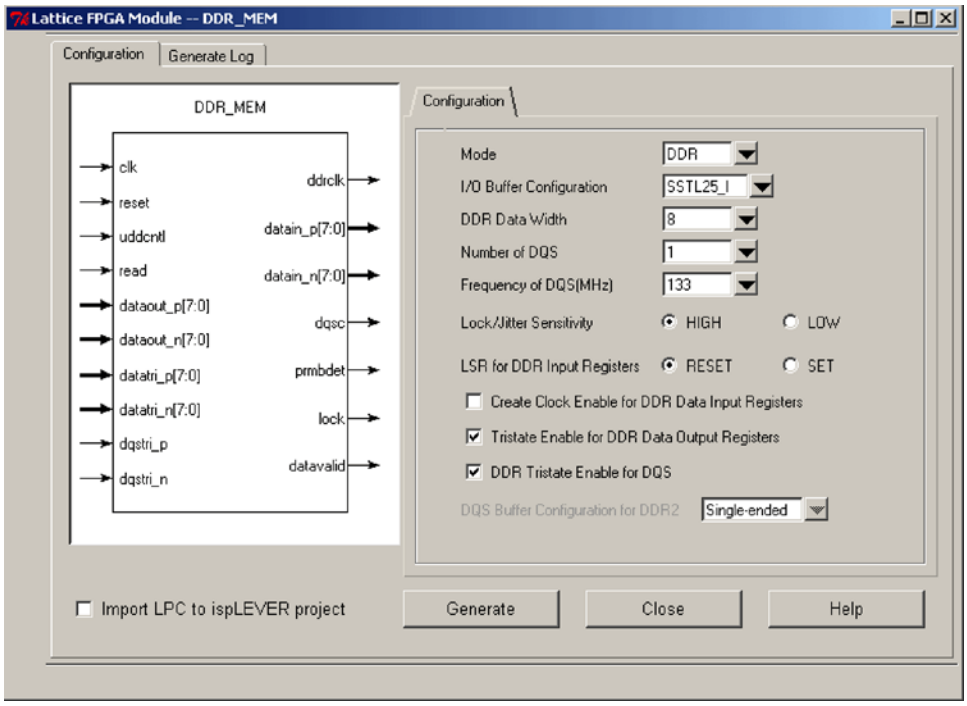
Figure 12-45. IPexpress Main Window for DDR_MEM



Configuration Tab

The Configuration Tab lists all user-accessible attributes with default values set. Upon completion, click **Generate** to generate source and constraint files. The user may choose to use the .lpc file to load parameters.

Figure 12-46. Configuration Tab for DDR_MEM



The user can change the Mode parameter to choose either the DDR or DDR2 interface. The other configuration parameters will change according to the Mode selected. The Number of DQS parameter determines the number of DDR interfaces. The software will assume there are eight data bits for every DQS. The user can also choose the frequency of operation and the DDR DLL will be configured to this frequency.

The user has an option to enable the clock enable and tristate enables for the DDR registers. It is recommend that the Lock/Jitter be enabled if the DDR interface is running at 150MHz or higher.

The parameters available depend on the mode selected. Tables 12-14 and 12-15 describe all user parameters in the IPexpress GUI and their usage for modes DDR and DDR2.

Table 12-14. User Parameters in the IPexpress GUI when in DDR Mode

User Parameters	Description	Values/Range	Default
I/O Buffer Configuration	I/O Standard used for the Interface. This will also depend on the Mode selected.	SSTL25_I, SSTL25_II	SSTL25_I
Data Width	Width of the Data bus	8-64	8
Number of DQS	Number of DQS will determine the number of DQS Groups	1, 2, 4, 8	1
Frequency of DQS	DDR Interface Frequency. This is also input to the DDR DLL. The values will depend on the mode selected.	100MHz, 133MHz, 166MHz, 200MHz	200MHz
Lock/Jitter Sensitivity	DLL Sensitivity to Jitter	High, Low	High
LSR for DDR Input Register	LSR Control	RESET, SET	RESET
Create Clock Enable for DDR Input Register	Create Clock enable inputs to the block	On/Off	Off
Tri-state Enable for DDR Output Registers	Creates Tri-state control for the DDR data output registers.	On/Off	On
DDR Tristate enable for the DQS output	Creates Tristate control for DQS output	On/Off	On

Table 12-15. User Parameters in the IPexpress GUI when in DDR2 Mode

User Parameters	Description	Values/Range	Default
I/O Buffer Configuration	I/O Standard used for the Interface. This will also depend on the Mode selected.	SSTL18_I, SSTL18_II	SSTL18_I
Data Width	Width of the Data bus	8-64	8
Number of DQS	Number of DQS will determine the number of DQS Groups	1, 2, 4, 8	1
Frequency of DQS	DDR Interface Frequency. This is also input to the DDR DLL. The values will depend on the mode selected.	166MHz, 200MHz, 266MHz	200MHz
Lock/Jitter Sensitivity	DLL Sensitivity to Jitter	High, Low	High
LSR for DDR Input Register	LSR Control	RESET, SET	RESET
Create Clock Enable for DDR Input Register	Create Clock enable inputs to the block	On/Off	Off
Tri-state Enable for DDR Output Registers	Creates Tri-state control for the DDR data output registers.	On/Off	On
DDR Tristate enable for the DQS output	Creates Tristate control for DQS output	On/Off	On
DQS Buffer Configuration for DDR2	DQS Buffer can be configured as Differential	On/Off	Off

FCRAM ("Fast Cycle Random Access Memory") Interface

FCRAM is a DDR-type DRAM, which performs data output at both the rising and falling edges of the clock. FCRAM devices operate at a core voltage of 2.5V with SSTL Class II I/O. It has enhanced both the core and peripheral logic of the SDRAM. In FCRAM the address and command signals are synchronized with the clock input, and the data pins are synchronized with the DQS signal. Data output takes place at both the rising and falling edges of the DQS. DQS is in phase with the clock input of the device. The DDR SDRAM and DDR FCRAM controller will have different pinouts.

LatticeECP2/M devices can implement the FCRAM interface using dedicated DQS logic, input DDR registers and output DDR registers, as described in the Implementing Memory Interfaces section of this document. Generation of address and control signals for FCRAM are different than in DDR SDRAM devices. Please refer to the FCRAM data sheets to see detailed specifications. Toshiba, Inc. and Fujitsu, Inc. offer FCRAM devices in 256Mb densities. They are available in x8 or x16 configurations.

Board Design Guidelines

The most common challenge associated with implementing DDR memory interfaces is the board design and layout. It is required that users strictly follow the guidelines recommended by memory device vendors.

Some of the common recommendations include matching trace lengths of interface signals to avoid skew, proper DQ-DQS signal grouping, proper termination of the SSTL2 or SSTL18 I/O Standard, proper VREF and VTT generation decoupling and proper PCB routing.

The following documents include board layout guidelines:

- www.idt.com, IDT, PCB Design for Double Data Rate Memory
- www.motorola.com, AN2582, Hardware and Layout Design Considerations for DDR Interfaces

References

- www.jedec.org, JEDEC Standard 79, Double Data Rate (DDR) SDRAM Specification
- www.micron.com, DDR SDRAM Data Sheets

- www.infinion.com, DDR SDRAM Data Sheets
- www.samsung.com, DDR SDRAM Data Sheets
- www.toshiba.com, DDR FCRAM Data Sheet
- www.fujitsu.com, DDR FCRAM Data Sheet
- RD1019, QDR Memory Controller Reference Design
- EB23, LatticeECP2 Advanced Evaluation Board User's Guide
- IPUG35, DDR1 & DDR2 SDRAM Controller (Pipelined Versions) User's Guide

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Revision History

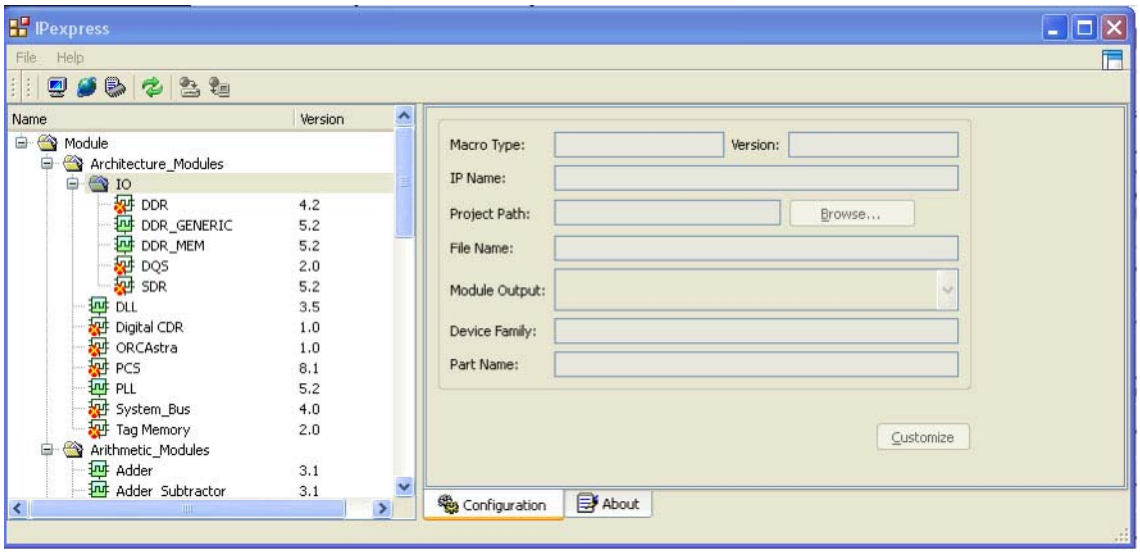
Date	Version	Change Summary
February 2006	01.0	Original version.
September 2006	01.1	Some figures updated. Added information on DELAYB block.
September 2006	01.2	Updated for LatticeECP2M. Added "DDR Generic Usage in IPexpress" section.
February 2007	01.3	Updated DDR and DDR2 SDRAM Interfaces Overview section.
June 2007	01.4	Updated the port names on the input DDR block diagrams.
		Updated text in the DQS Transition Detect under Memory Read Implementation section.
		Updated the DQSDEL from 6-bit bus to single bit in figures 12-19 and 12-20.
		Updated text in the DLL Compensated DQS Delay Elements section under Memory Read Implementation.
October 2007	01.5	Updated DDRX2B Waveform diagram.
June 2009	01.6	Updated DQSDLL Update Control text section.
January 2010	01.7	Updated Read Data Transfer When Using IDDRMFX1A figure.
		Updated Data Read Critical Path text section.
June 2010	01.8	Added Appendix A.

Appendix A. DDR Generation Using IPexpress with Lattice Diamond

The IPexpress tool in the Lattice Diamond design software can be used to configure and generate the DDR Memory Interface and Generic DDR Module. IPexpress will generate an HDL module that will contain the DDR primitives. This module can be using in the top level design.

Figure 12-47 shows the main window of IPexpress. The DDR_Generic and DDR_MEM options under **Architecture > IO** are used to configure the DDR modules.

Figure 12-47. IPexpress Main Window

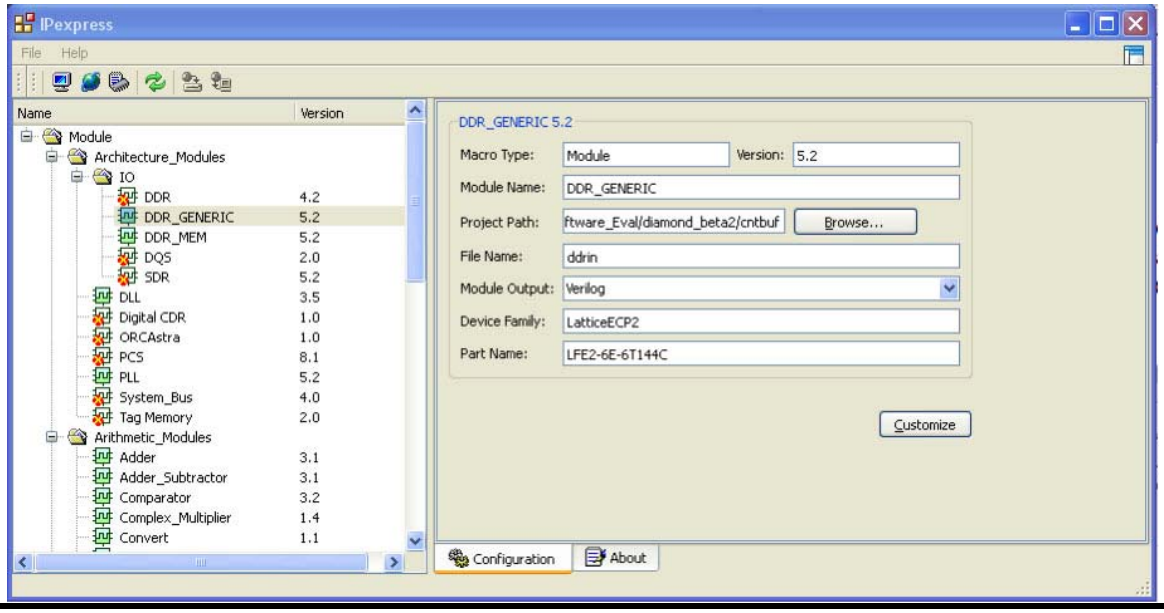


DDR Generic

Figure 12-48 shows the main window when DDR_Generic is selected. The only entry required in this window is the module name. Other entries are set to the project settings.

The user may change these entries if desired. After entering the module name, click on **Customize** to open the Configuration Tab window as shown in Figure 12-49.

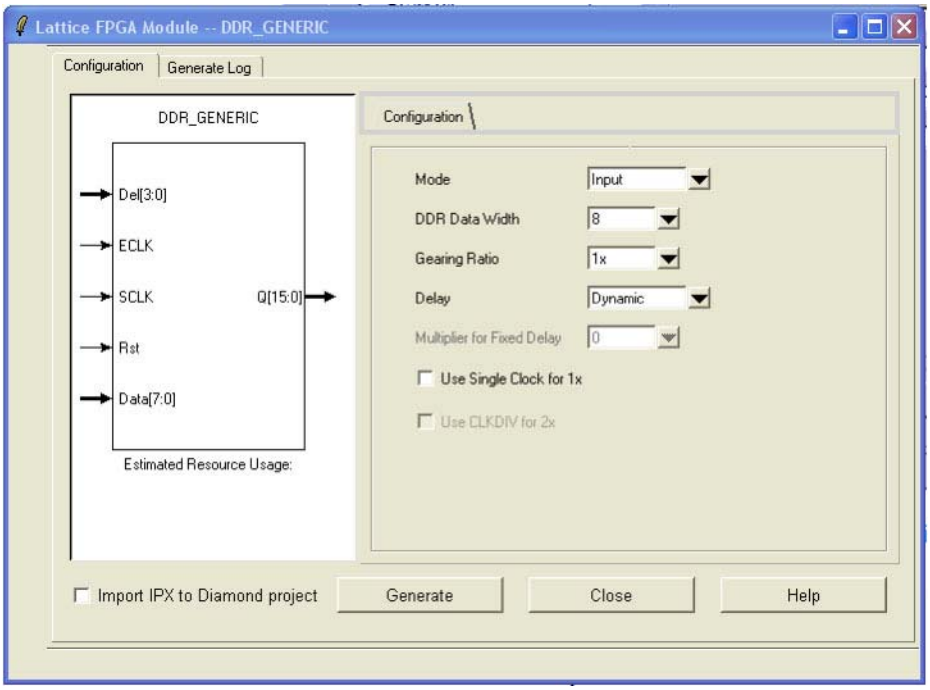
Figure 12-48. IPexpress Main Window for DDR_Generic



Configuration Tab

The Configuration Tab lists all user-accessible attributes with default values set. Upon completion, click **Generate** to generate source and constraint files. The user may choose to use the .lpc file to load parameters.

Figure 12-49. Configuration Tab for DDR_Generic



The user can change the Mode parameter to choose either the Input, Output, Bi-directional or Tristate DDR modules. The other configuration parameters will change according to the mode selected. The Delay parameter is only available for Input and Bi-directional modes. Similarly, the Multiplier for Fixed Delay parameter is only available when the Delay parameter is configured as Fixed.

Table 12-16. User Parameters in the IPexpress GUI

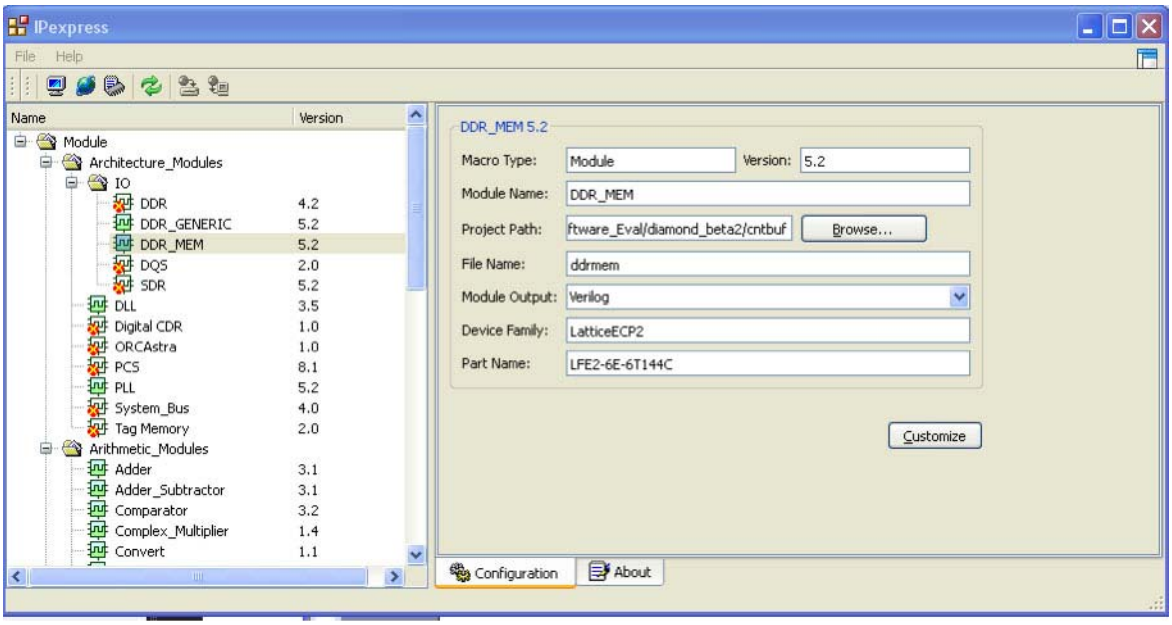
User Parameters	Description	Values/Range	Default
Mode	Mode selection for the DDR block.	Input, Output, Bidirectional, Tristate	Input
Data Width	Width of the data bus.	1-64	8
Gearing Ratio	Gearing ratio selection.	1x, 2x[1]	1x
Delay	Input delay configuration.	Dynamic, Fixed, Fixed XGMII	Dynamic
Multiplier for Fixed Delay	Fixed delay setting. Available only when delay is configured as Fixed.	0-15	0
Use Single Clk for 1x	Allows the selection of a single clock for the gearing logic.	On/Off	Off

1. Only 1x available when Mode is Bi-directional or Tristate.

DDR_MEM

Figure 12-50 shows the main window when DDR_MEM is selected. Similar to DDR_Generic, the only entry required here is the module name. Other entries are set to the project settings. The user may change these entries if desired. After entering the module name, click on **Customize** to open the Configuration Tab window as shown in Figure 12-51.

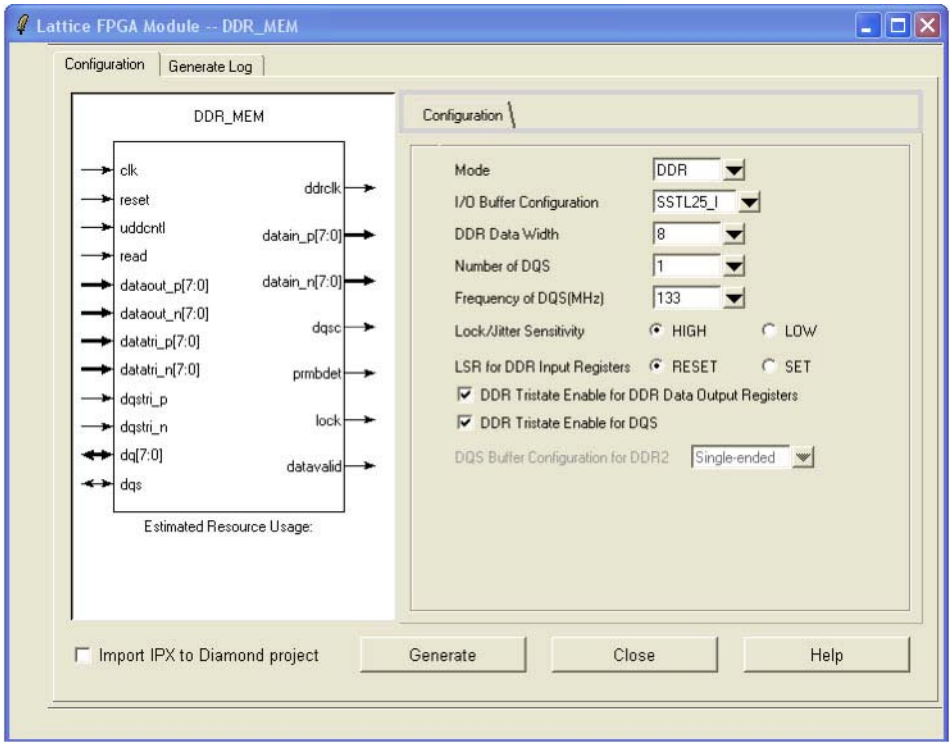
Figure 12-50. IPexpress Main Window for DDR_MEM



Configuration Tab

The Configuration Tab lists all user-accessible attributes with default values set. Upon completion, click **Generate** to generate source and constraint files. The user may choose to use the .lpc file to load parameters.

Figure 12-51. Configuration Tab for DDR_MEM



The user can change the Mode parameter to choose either the DDR or DDR2 interface. The other configuration parameters will change according to the Mode selected. The Number of DQS parameters determines the number

of DDR interfaces. The software will assume there are eight data bits for every DQS. The user can also choose the frequency of operation and the DDR DLL will be configured to this frequency.

The user has the option to enable the clock enable and tristate enables for the DDR registers. It is recommend that the Lock/Jitter be enabled if the DDR interface is running at 150 MHz or higher.

The parameters available depend on the mode selected. Tables 12-17 and 12-18 describe all user parameters in the IPexpress GUI and their usage for modes DDR and DDR2.

Table 12-17. User Parameters in the IPexpress GUI when in DDR Mode

User Parameters	Description	Values/Range	Default
I/O Buffer Configuration	I/O Standard used for the Interface. This will also depend on the Mode selected.	SSTL25_I, SSTL25_II	SSTL25_I
Data Width	Width of the Data bus	8-64	8
Number of DQS	Number of DQS will determine the number of DQS Groups	1, 2, 4, 8	1
Frequency of DQS	DDR Interface Frequency. This is also input to the DDR DLL. The values will depend on the mode selected.	100MHz, 133MHz, 166MHz, 200MHz	200MHz
Lock/Jitter Sensitivity	DLL Sensitivity to Jitter	High, Low	High
LSR for DDR Input Register	LSR Control	RESET, SET	RESET
Create Clock Enable for DDR Input Register	Create Clock enable inputs to the block	On/Off	Off
Tri-state Enable for DDR Output Registers	Creates Tri-state control for the DDR data output registers.	On/Off	On
DDR Tristate enable for the DQS output	Creates Tristate control for DQS output	On/Off	On

Table 12-18. User Parameters in the IPexpress GUI when in DDR2 Mode

User Parameters	Description	Values/Range	Default
I/O Buffer Configuration	I/O Standard used for the Interface. This will also depend on the Mode selected.	SSTL18_I, SSTL18_II	SSTL18_I
Data Width	Width of the Data bus	8-64	8
Number of DQS	Number of DQS will determine the number of DQS Groups	1, 2, 4, 8	1
Frequency of DQS	DDR Interface Frequency. This is also input to the DDR DLL. The values will depend on the mode selected.	166MHz, 200MHz, 266MHz	200MHz
Lock/Jitter Sensitivity	DLL Sensitivity to Jitter	High, Low	High
LSR for DDR Input Register	LSR Control	RESET, SET	RESET
Create Clock Enable for DDR Input Register	Create Clock enable inputs to the block	On/Off	Off
Tri-state Enable for DDR Output Registers	Creates Tri-state control for the DDR data output registers.	On/Off	On
DDR Tristate enable for the DQS output	Creates Tristate control for DQS output	On/Off	On
DQS Buffer Configuration for DDR2	DQS Buffer can be configured as Differential	On/Off	Off

Power Estimation and Management for LatticeECP2/M Devices

November 2009 Technical Note TN1106

Introduction

Power considerations in FPGA design are critical for determining the maximum system power requirements and sequencing requirements of the FPGA on the board. This technical note provides users with detailed power considerations such as sequencing. Also included are instructions for calculating power consumption in LatticeECP2™ and LatticeECP2M™ devices using the Power Calculator available in the Lattice ispLEVER® design tool. General guidelines for reducing power consumption are also discussed.

Power Supply Sequencing

Power-Up Sequencing

There are three main power supplies that are required to power-up the LatticeECP2/M device for proper operation: V_{CC}, V_{CCAUX} and V_{CCIO8}. Bank 8, or V_{CCIO8}, powers the sysCONFIG™ port and configuration circuitry and is therefore required during power-up.

The nominal voltages for these power supplies are 1.2V for V_{CC}, 3.3V for V_{CCAUX} and 1.2V to 3.3V for V_{CCIO8}. The nominal trip points for these power supplies are 0.6V to 0.8 V for V_{CC} and V_{CCIO8}, and 2.2V to 2.5V for V_{CCAUX}. For power supply sequencing, refer to the Recommended Operation Conditions section of the LatticeECP2/M Family Data Sheet.

Each power supply must follow a monotonically clean ramp between the trip points and the minimum required supply voltage. Note that for slow ramps (when the power-up ramp rate is 10s or 100s milliseconds) it is critical that the ramp is clean and monotonic. The device may go in and out of the power-up reset if the ramp is unclean and non-monotonic, especially around the trip point. This also applies when powering down the device. A clean, monotonic ramp will ensure that the device will power up and power down properly.

After initialization is complete, if any V_{CC}, V_{CCAUX} or V_{CCIO8} drops below its power-down trip point, the device will reset. Any $V_{CCIO[7:0]}$ can be removed without resetting the device after initialization is complete.

Refer to the LatticeECP2/M Family Data Sheet and TN1108, LatticeECP2/M sysCONFIG Usage Guide, for configuration timing and power-up information.

Power-Down Sequencing

During power-down, power should be removed from one of the supplies' V_{CC}, V_{CCAUX} or V_{CCIO8} first to ensure that no high currents are seen on the input pins as the other V_{CCIO} supplies are removed. This only applies when input signals are still being driven, such as in hot-socketing applications.

For non-hot-socketing applications, the input signals are likely to be powered from the same supply as V_{CCIO}. Therefore, they will usually be less than or equal to V_{CCIO} during power down.

Power Sequencing Recommendations

LatticeECP2/M devices do not have a power-up sequence requirement. The supplies can be brought up in any sequence.

In order to minimize the transients and hot socketing currents during power up, Lattice recommends that the V_{CC} be brought up before V_{CCAUX} or V_{CCIO8}. Additionally, V_{CC} should reach its minimum voltage value before V_{CCAUX} and V_{CCIO8} reach their minimum values. When removing the supplies, V_{CCAUX} or V_{CCIO8} must be removed before V_{CC} is turned off.

Note that this this sequence is not a requirement for LatticeECP2/M devices.

For LatticeECP2MS power-up sequencing, refer to the Recommended Operation Conditions section of the LatticeECP2/M Family Data Sheet.

Power Calculator Hardware Assumptions

The Power Calculator reports the power dissipation in terms of:

1. DC portion of the power consumption
2. AC portion of the power consumption

Total power dissipation is the sum of the static (DC) and dynamic (AC) power dissipations of a device. While DC power depends upon voltage, temperature and process variation, AC power is a strong function of the frequency and the activity of the resources and a weak function of voltage, temperature and process.

Static Power or DC Power

DC power can be further subdivided into the power consumption of the used and unused resources. Another important term is Quiescent Power, the DC power for a blank (BE or Bulk Erase) device. In the Bulk Erase mode, none of the resources are used, so it is the total DC power of an unused device.

The AC portion of the power consumption, associated with used resources, is the dynamic part of the power consumption. AC power dissipation is directly proportional to the frequency and activity at which the resource is running and the number of resource units used.

Junction Temperature

For a fixed temperature, voltage and device package combination, quiescent power is fixed.

Ambient temperature that affects the junction temperature is a factor that contributes to the final power consumption.

Power Calculator models this ambient-to-junction temperature dependency. When a user provides an ambient temperature, it is rolled into an algorithm which calculates the junction temperature and quiescent power through an iterative process.

Typical and Worst Case Process Power/I_{CC}

Another factor that affects the DC power is process variation. This variation is turn causes variation in quiescent power.

Power Calculator takes these factors into account and allows users to specify either a typical process or a worst case process.

- A typical process selection under Device Variation allows users to calculate the power dissipation of a design using a typical process device.
- The worst case selection under the same option provides the maximum power dissipation for the device and package combination. This information is particularly useful for FPGA power budgeting for the entire system.

Dynamic Power Budgets and Maximum Operating Temperature

When designing a system, designers must make sure a device operates at specified temperatures within the system environment. This is particularly important to consider before a system is designed. With Power Calculator, users can predict device thermodynamics and estimate the dynamic power budget. The ability to estimate a device's operating temperature prior to board design also allows the designer to better plan for power budgeting and airflow.

Although total power, ambient temperature, thermal resistance and airflow all contribute to device thermodynamics, the junction temperature (as specified in the device data sheet) is the key to device operation. The allowed junction temperature range is 0°C to 85°C for commercial devices and -40°C to 100°C for industrial devices. Any time the junction temperature of the die falls out of these ranges, the performance and reliability of the device's operation

must be evaluated. The reliability limit of junction temperature, on the other hand, for this generation of device technology is 125°C.

Let us consider an example for how to determine and use the Power Calculator for thermal analysis. Once the user has imported or provided all the required information in the Power Calculator, the software will provide the power estimation and predict the Junction Temperature (T_J). Any time this junction temperature is outside the limits specified in the device data sheet, the viability of operating the device at this junction temperature must be re-evaluated. A commercial device is likely to show speed degradation with a junction temperature above 85°C and an industrial device at a junction temperature will degrade above 100°C. It is required that the die temperature be kept below these limits to achieve the guaranteed speed operation.

Operating a device at a higher temperature also means a higher SICC. The difference between the SICC and the total ICC (both Static ICC and Dynamic ICC) at a given temperature provides the dynamic budget available. If the device runs at a dynamic ICC higher than this budget, the total ICC is also higher. This causes the die temperature to rise above the specified operating conditions.

There are a number of ways to handle this situation. Some of these are discussed in the Power Management section of this document. The four factors listed earlier in this section, namely power, ambient temperature, thermal resistance and airflow, can also be varied and controlled to reduce the junction temperature of the device.

Power Calculator is a powerful tool to help system designers properly budget the FPGA power that in turn helps improve the overall system reliability.

Power Calculator

Power Calculator is a powerful tool that allows users to estimate power consumption at two different levels:

1. Estimate of the utilized resources before completing place and route
2. Post place and route design

At a coarse level of estimation, the user provides estimates of device usage in the Power Calculator Wizard and the tool provides a rough estimate of the power consumption.

For a more accurate approach, a designer can import actual device utilization by importing the post Place and Route netlist (NCD) file.

Power Calculation Equations

The following are the power equations used in the Power Calculator:

Total DC Power (Resource)
= Total DC Power of Used Portion + Total DC Power of Unused Portion
= [DC Leakage per Resource when Used * NRESOURCE]
+ [DC Leakage per Resource when Unused * ($N_{TOTAL\ RESOURCE}$ - $N_{RESOURCE}$)]

Where:
$N_{TOTAL\ RESOURCE}$ is the total number of Resources in a device.
$N_{RESOURCE}$ is the number of Resources used in the design.

The total DC power consumption for all the resources as per the design data is the sum of the quiescent power and the individual DC power of the resources in the Power Calculator.

Total DC Power (I_{CCAUX})
= $K_{RESOURCE}$ * 525 µA + Typical Standby I_{CCAUX}

Where:

$K_{RESOURCE}$	is the number of reference input I/O such as HSTL/SSTL. For LVDS KRESOURCE is number of inputs divided by two.
I_{CCAUX}	is a DC current that does not change with I/O toggle rate or temperature.

Typical Standby I_{CCAUX} is found in the data sheet.

The AC power, on the other hand, is governed by the following equation:

Total AC Power (Resource)
$= K_{RESOURCE} * f_{MAX} * AF_{RESOURCE} * N_{RESOURCE}$

Where:

$N_{RESOURCE}$	is the number of resources used in the design.
$K_{RESOURCE}$	is the power constant for the resource in mW/MHz.
f_{MAX}	is the max. frequency at which the resource is running. Frequency is measured in MHz.
$AF_{RESOURCE}$	is the activity factor for the resource group.The Activity Factor is a percentage of the switching frequency.

For example, the power consumption of the LUT is calculated as per the following equation,

Total AC Power (LUT)
$= K_{LUT} * f_{MAX} * AF_{LUT} * N_{LUT}$

Where:

N_{LUT}	is the number of LUTs used in the design.
K_{LUT}	is the Power constant for the LUT blocks in mW/MHz.
f_{MAX}	is the max. frequency of the LUT clock measured in MHz.
AF_{LUT}	is the activity factor for the LUT. The Activity Factor is a percentage of the switching frequency.

Another example is the power consumption of the EBR block, which is calculated as follows:

Total AC Power (EBR)
$= K_{EBR} * f_{MAX} * AF_{EBR} * N_{EBR}$

Where:

NEBR	is the number of EBR blocks used in the design.
KEBR	is the power constant for the EBR blocks in mW/MHz.
FMAX	is the max. frequency of the EBR clock measured in MHz.
AFEBR	is the activity factor for the Read and Write ports of the EBR. The Activity Factor is a percentage of the switching frequency.

Also note that the LUT can be configured in Logic, Ripple or Distributed RAM modes. Each of these modes has a different power constant/power coefficient. However, the equations stay the same.

The AC power of some of the dedicated blocks can be calculated using the following equation:

Total AC Power (Dedicated Resource)
$= K_{RESOURCE} * f_{MAX} * N_{RESOURCE}$

Where:

$N_{RESOURCE}$	is the number of resources used in the design.
$K_{RESOURCE}$	is the power constant for the resource in mW/MHz.
f_{MAX}	is the max. frequency at which the resource is running measured in MHz.

Activity Factor Calculation

The Activity Factor % (or AF%) is defined as the percentage of frequency (or time) that a signal is active or toggling the output.

Most resources associated with a clock domain are running or toggling at some percentage of the frequency at which the clock is running. Users must provide this value as a percentage under the AF% column in the Power Calculator tool.

Another term for I/Os is the I/O Toggle Rate. The AF% is applicable to the PFU, Routing, and Memory Read Write Ports, etc. The activity of I/Os is determined by the signals provided by the user (in the case of inputs) or as an output of the design (in the case of outputs). The rates at which I/Os toggle define their activity. The I/O Toggle Rate or the I/O Toggle Frequency is a better measure of their activity.

The Toggle Rate (or TR) in MHz of the output is defined in the following equation:

$$\text{Toggle Rate (MHz)} = 1/2 * f_{MAX} * \text{AF\%}$$

Users are required to provide the TR (MHz) value for the I/O instead of providing the frequency and AF% for other resources.

AF can be calculated for each routing resource, output or PFU. However, this involves long calculations. The general recommendation for a design occupying roughly 30% to 70% of the device is an AF% between 15% and 25%. This is an average value. The accurate value of an AF depends upon clock frequency, stimulus to the design and the final output.

Ambient and Junction Temperatures and Airflow

A common method of characterizing a packaged device's thermal performance is with Thermal Resistance, θ. In a semiconductor device, thermal resistance indicates the steady state temperature rise of the die junction above a given reference for each watt of power (heat) dissipated at the die surface. Its units are °C/W.

The most common examples are θ_{JA}, Thermal Resistance Junction-to-Ambient (in °C/W) and θ_{JC}, Thermal Resistance Junction-to-Case (also in °C/W). Another factor is θ_{JB}, Thermal Resistance Junction-to-Board (in °C/W).

Knowing the reference (i.e. ambient, case or board) temperature, the power and the relevant θ value, the junction temperature can be calculated as follows.

$$T_J = T_A + \theta_{JA} * P \quad (1)$$
$$T_J = T_C + \theta_{JC} * P \quad (2)$$
$$T_J = T_B + \theta_{JB} * P \quad (3)$$

Where T_J, T_A, T_C and T_B are the Junction, Ambient, Case (or Package) and Board temperatures (in °C) respectively. P is the total power dissipation of the device.

θ_{JA} is commonly used with natural and forced convection air-cooled systems. θ_{JC} is useful when the package has a high conductivity case mounted directly to a PCB or heat sink. θ_{JB} applies when the board temperature adjacent to the package is known.

Power Calculator utilizes the ambient temperature (°C) to calculate the junction temperature (°C) based on the θ_{JA} for the targeted device, per equation 1 above. Users can also provide the airflow values (in LFM) to get a more accurate value of the junction temperature.

Managing Power Consumption

One of the most critical factors in design today is reducing the system power consumption. Low power consumption is especially important for hand-held devices and other modern electronic products. There are several design techniques that can significantly reduce overall system power consumption. These include:

1. Reducing the operating voltage.
2. Operating within the specified package temperature limitations.
3. Using optimum clock frequency to reduce power consumption, as the dynamic power is directly proportional to the frequency of operation. Designers must determine if a portion of their design can be clocked at a lower rate, which will reduce power.
4. Reducing the span of the design across the device. A more closely placed design utilizes fewer routing resources for less power consumption.
5. Reducing the voltage swing of the I/Os where possible.
6. Using optimum encoding where possible. For example, a 16-bit binary counter has, on average, only 12% Activity Factor and a 7-bit binary counter has an average of 28% Activity Factor. On the other hand, a 7-bit Linear Feedback Shift Register can toggle as much as 50% Activity Factor, which causes higher power consumption. A gray code counter, where only one bit changes at each clock edge, will use the least amount of power, as the Activity Factor is less than 10%.
7. Minimizing the operating temperature, by the following methods:
   a. Use packages that can better dissipate heat. For example, packages with lower thermal impedance.
   b. Place heat sinks and thermal planes around the device on the PCB.
   c. Better airflow techniques using mechanical airflow guides and fans (both system fans and device-mounted fans).

Power Calculator Assumptions

The following are the assumptions made by the Power Calculator:

1. The Power Calculator tool uses equations with constants based on a room temperature of 25°C.
2. The user can define the Ambient Temperature (T_A) for device Junction Temperature (T_J) calculation based on the power estimation. T_J is calculated from the user-entered T_A and the power calculation of typical room temperature.
3. I/O power consumption is based on an output loading of 5pF. Users have the ability to change this capacitive loading.
4. Users can estimate power dissipation and current for each type of power supplies that are V_{CC}, V_{CCIO}, V_{CCJ}, and V_{CCAUX}. For V_{CCAUX}, only static I_{CCAUX} values are provided in the Power Calculator.

   Additional V_{CCAUX} contributions due to differential output buffers, differential input buffers and reference input buffers must be added per pair for differential buffers or per pin for reference input buffers according to the user's design. See the equation given in this technical note for Total DC Power (I_{CCAUX}).
5. The nominal V_{CC} is used by default to calculate the power consumption. A lower or higher V_{CC} can be chosen from a list of available values.
6. Users can enter an Airflow in Linear Feet per Minute (LFM) along with a Heat Sink option to calculate the Junction Temperature.
7. The default value of the I/O types for the LatticeECP2/M devices is LVCMOS12, 6mA.
8. The Activity Factor is defined as the toggle rate of the registered output. For example, assuming that the input of a flip-flop is changing at every clock cycle, 100% AF of a flip-flop running at 100MHz is 50MHz.

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
February 2006	01.0	Initial release.
September 2006	01.1	Updated for LatticeECP2/M.
		Added discussion on Dynamic Power Budgets and Junction Temperature.
November 2006	01.2	Added calculation of I_{CCAUX} in Power Calculation Equations section.
January 2007	01.3	Updated Power Supply Sequencing section.
November 2009	01.4	Updated Power-up Sequencing text section.
		Updated Power Sequencing Recommendations text section.

LatticeECP2/M sysDSP Usage Guide

June 2010 Technical Note TN1107

Introduction

This technical note discusses how to access the features of the LatticeECP2™ and LatticeECP2M™ sysDSP™ (Digital Signal Processing) Block described in the LatticeECP2/M Family Data Sheet. Designs targeting the sysDSP Block can offer significant improvement over traditional LUT-based implementations. Table 14-1 provides an example of the performance and area benefits of this approach:

Table 14-1. sysDSP Block vs. LUT-based Multipliers

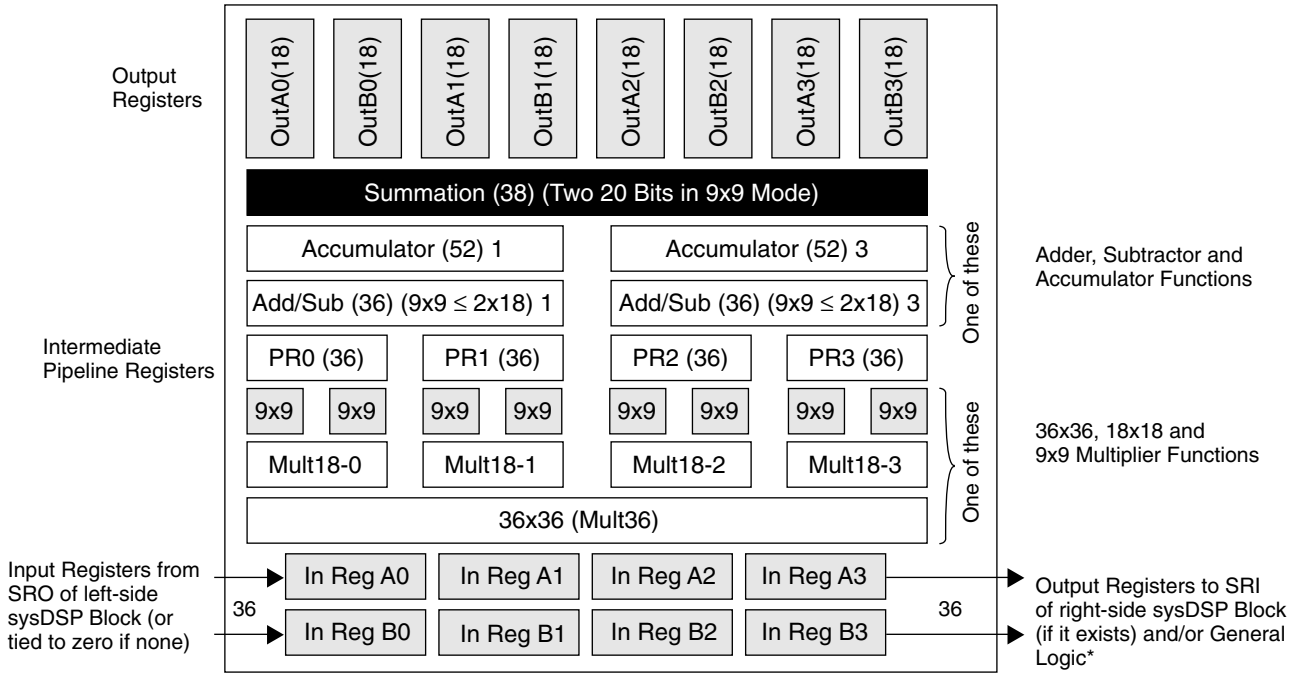
Multiplier Width	Register Pipelining	LatticeECP2-50-7 Uses One sysDSP Block		LatticeECP2-50-7 Uses LUTs	
		f_{MAX} (MHz)[1]	LUTs	f_{MAX} (MHz)[1]	LUTs
9x9	Input, Multiplier, Output	404	0	124	192
18x18	Input, Multiplier, Output	420	0	95	698
36x36	Input, Multiplier, Output	371	0	61	2732

1. These timing numbers were generated using the ispLEVER® design tool. Exact performance may vary with design and tool version.

sysDSP Block Hardware

The LatticeECP2/M sysDSP Blocks are located in rows throughout device. Below is a block diagram of one of the sysDSP Blocks:

Figure 14-1. LatticeECP2/M sysDSP Block



*Can only be routed to general logic routing when configured with less than three MULT18X18.

Note: Each sysDSP Block spans nine columns of PFUs.

The sysDSP Block can be configured as:

- One 36x36 Multiplier
  - Basic multiplier, no add/sub/accum/sum blocks
- Four 18x18 Multipliers
  - Two add/sub/accum blocks
  - One summation block for adding four multipliers
- Eight 9x9 Multipliers
  - Four add/sub blocks
  - Two summation blocks

Note that a sysDSP block can only be configured in one mode at a time.

sysDSP Block Software

Overview

The sysDSP Block of the LatticeECP2/M device can be targeted in a number of ways.

- The IPexpress™ tool in the ispLEVER or Lattice Diamond™ design software allows the rapid creation of modules implementing sysDSP elements. These modules can then be used in HDL designs as appropriate.
- The coding of certain functions into a design's HDL and allowing the synthesis tools to Inference the use of a sysDSP block.
- The implementation of designs in The MathWorks® Simulink® tool using a Lattice block set. The ispLEVER sysDSP design tools will then convert these blocks into HDL as appropriate.
- Instantiation of sysDSP primitives directly in the source code

Targeting sysDSP Block Using IPexpress

IPexpress allows you to graphically specify sysDSP elements. Once the element is specified, a HDL file is generated, which can be instantiated in a design. IPexpress allows users to configure all ports and set all available parameters. The following modules target the sysDSP Block in IPexpress:

- MULT (Multiplier)
- MAC (Multiplier Accumulate)
- MULTADDSUB (Multiplier Add/Subtract)
- MULTADDSUBSUM (Multiply Add/Subtract and SUM)

Note: See Appendix B for information about targeting a sysDSP Block using Lattice Diamond design software and IPexpress.

MULT Module

The MULT Module configures elements to be packed into the sysDSP primitives. The Basic mode screen illustrated in Figure 14-2 consists of an optional one clock, one clock enable and one reset tied to all registers. Multiple sysDSP Blocks can be spanned to accommodate large multiplications. Additional LUTs may be required if multiple sysDSP blocks are needed. Select **Area/Speed** to determine the LUT implementation. The input data format can be selected as Parallel, Shift or Dynamic. The Shift format can only be enabled if inputs are less than 18 bits. The Shift format enables a sample/shift register, which is useful in applications such as the FIR filter. The Advanced mode screen, illustrated in Figure 14-3, allows finer control over the register. In the Advanced mode, users can control each register with independent clocks, clock enables and resets. MULT inputs can be from 2 to 72 bits.

Figure 14-2. MULT Mode Basic Set-up

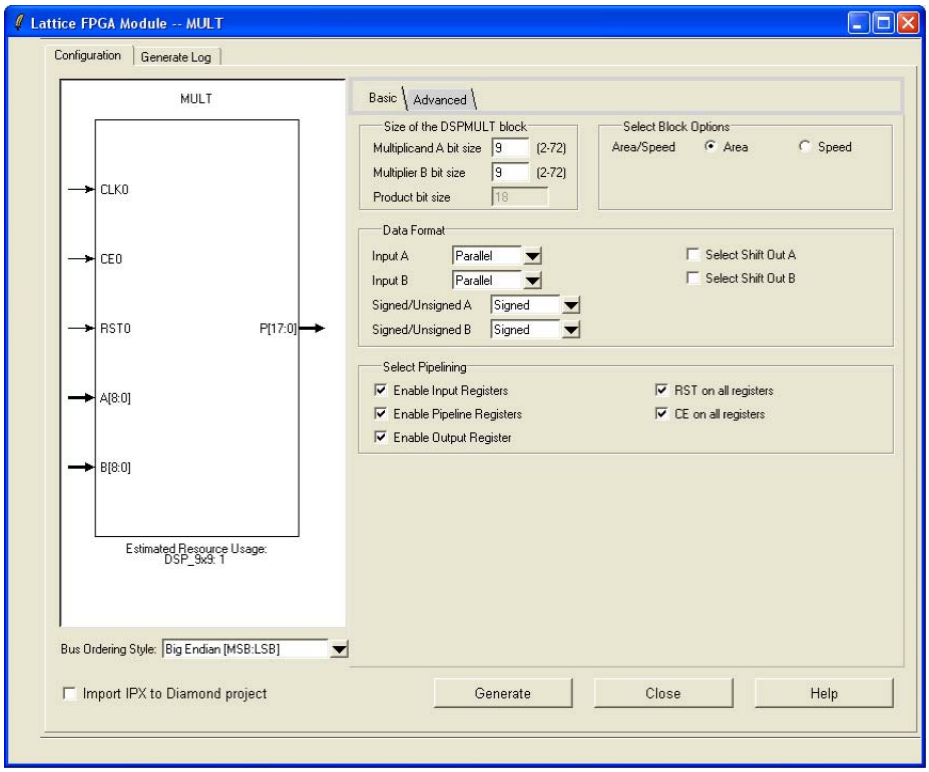
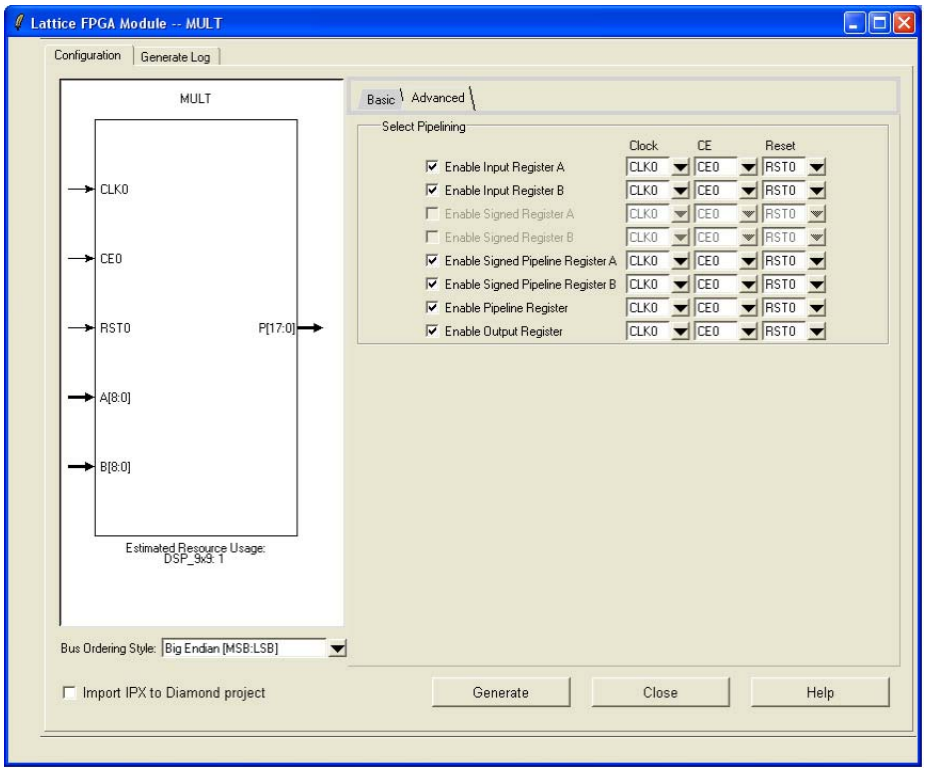


Figure 14-3. MULT Mode Advanced Set-up



MAC Module

The MAC Module configures multiply accumulate elements to be packed into the primitive MULT18X18MACB. The Basic mode, shown in Figure 14-4, consists of an optional one clock, one clock enable and one reset tied to all registers. Because of the accumulator, the output register is automatically enabled. Multiple sysDSP Blocks can be spanned to accommodate large multiplications. The accumulator of the sysDSP block is 52 bits deep and additional LUTs can be used if a larger accumulation is required. If sysDSP blocks are spanned, additional LUT logic may be required. Select **Area/Speed** to determine the LUT implementation. The input data format can be selected as Parallel, Shift or Dynamic. The Shift format can only be enabled if inputs are less than 18 bits. The Shift format enables a sample/shift register. The Accumsload loads the accumulator with the value from the LD port. This is required to initialize and load the first value of the accumulation. The Advanced mode, shown in Figure 14-5, allows finer control over the registers. In the advanced mode, users can control each register with independent clocks, clock enables and resets. MAC inputs can be from 2 to 72 bits.

Figure 14-4. MAC Mode Basic Set-up

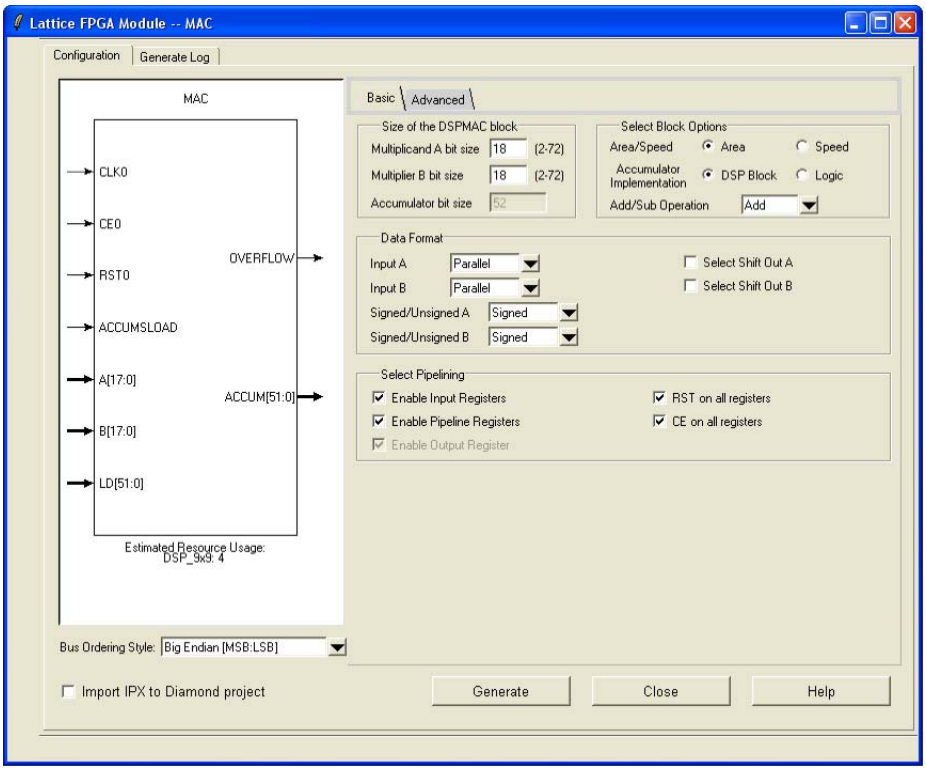
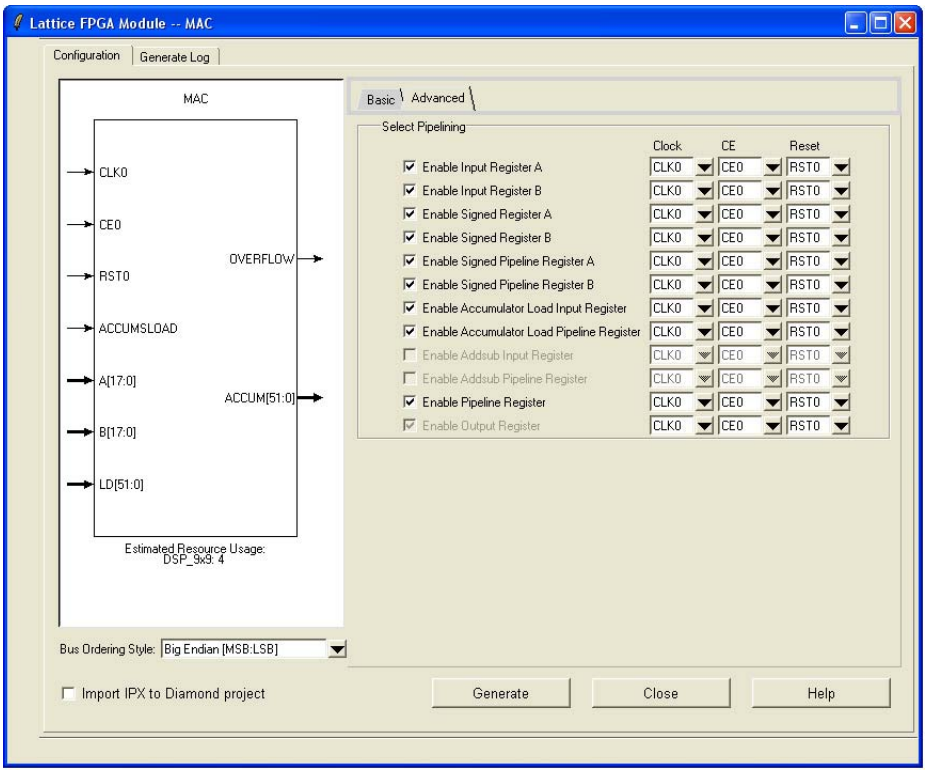


Figure 14-5. MAC Mode Advanced Set-up



MULTADDSUB Module

The MULTADDSUB GUI configures multiplier addition/subtraction elements to be packed into the primitives MULT18X18ADDSUBB or MULT9X9ADDSUBB. The Basic mode, shown in Figure 14-6, consists of an optional one clock, one clock enable and one reset tied to all registers. Multiple sysDSP Blocks can be spanned to accommodate large multiplications. If sysDSP blocks are spanned, additional LUT logic may be required. Select **Area/Speed** to determine the LUT implementation. The input data format can be selected as Parallel, Shift or Dynamic. The Shift format can only be enabled if inputs are less than 18 bits. The Shift format enables a sample/shift register, which is useful in applications such as the FIR filter. The Advanced mode, shown in Figure 14-7, provides finer control over the registers. In the advanced mode, users can control each register with independent clocks, clock enables and resets. MULTADDSUB inputs can be from 2 to 72 bits.

Figure 14-6. MULTADDSUB Mode Basic Set-up

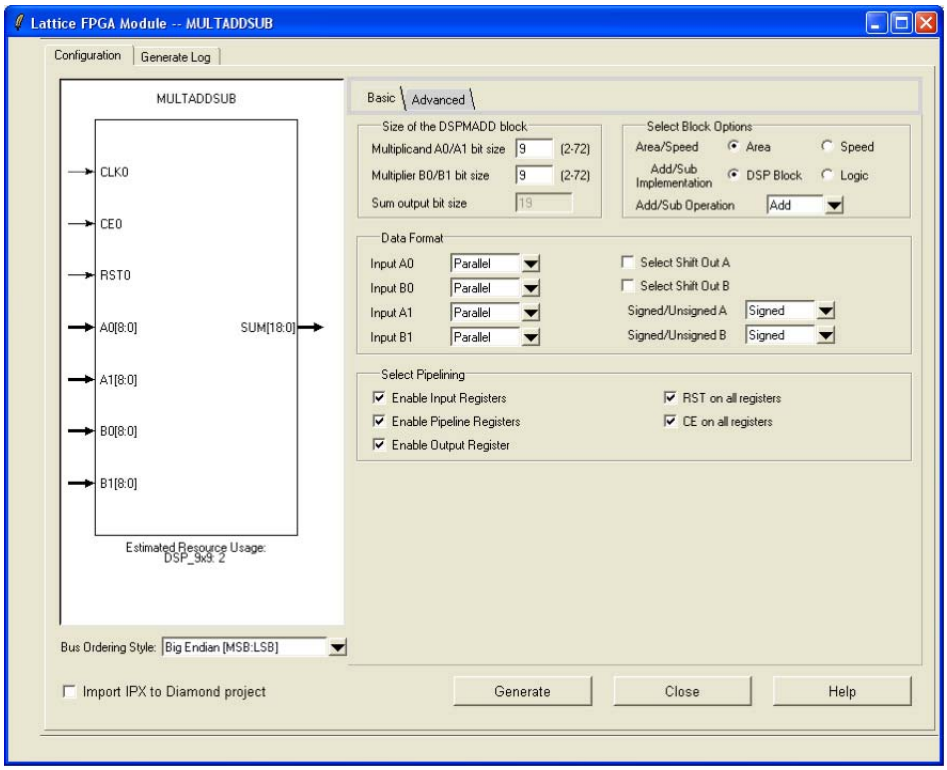
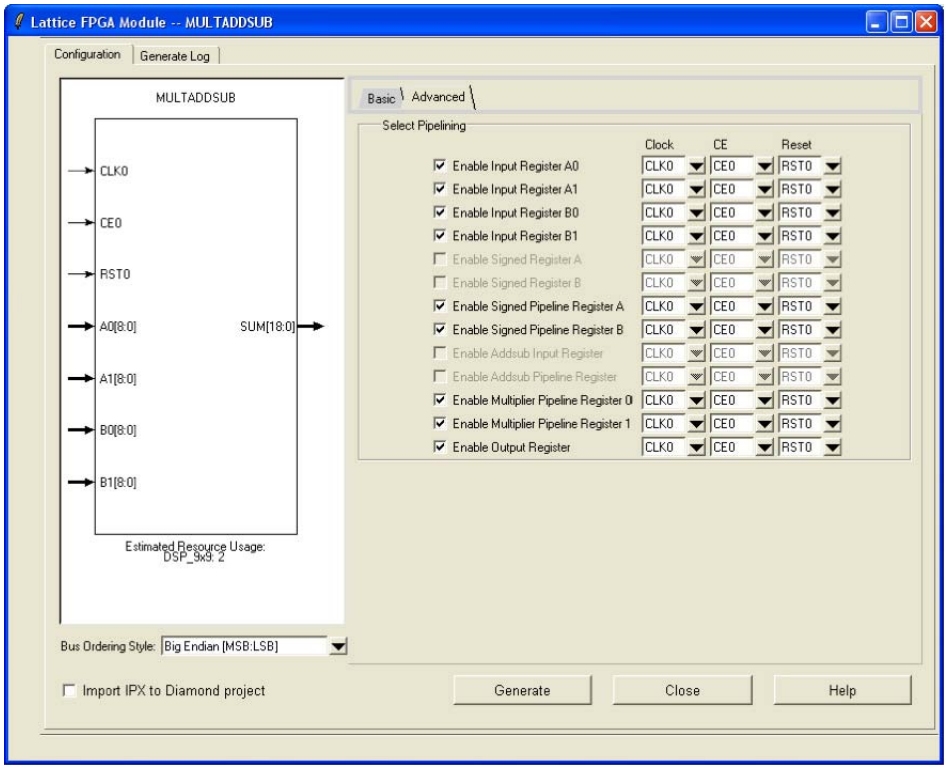


Figure 14-7. MULTADDSUB Mode Advanced Set-up



MULTADDSUBSUM Module

The MULTADDSUBSUM GUI configures Multiplier Addition/Subtraction Addition elements to be packed into the primitives MULT18X18ADDSUBSUMB or MULT9X9ADDSUBSUMB. The Basic mode, shown in Figure 14-8, consists of an optional one clock, one clock enable and one reset tied to all registers. Multiple sysDSP Blocks can be spanned to accommodate large multiplications. If sysDSP blocks are spanned, additional LUT logic may be required. Select **Area/Speed** to determine the LUT implementation. The input data format can be selected as Parallel, Shift or Dynamic. The Shift format is can only be enabled if inputs are less than 18 bits. The Shift format enables a sample/shift register, which is useful in applications such as the FIR filter. The Advanced mode, shown in Figure 14-9, provides finer control over the registers. In the advanced mode, users can control each register with independent clocks, clock enables and resets. MULTADDSUBSUM inputs can be from 2 to 72 bits.

Figure 14-8. MULTADDSUBSUM Mode Basic Set-up

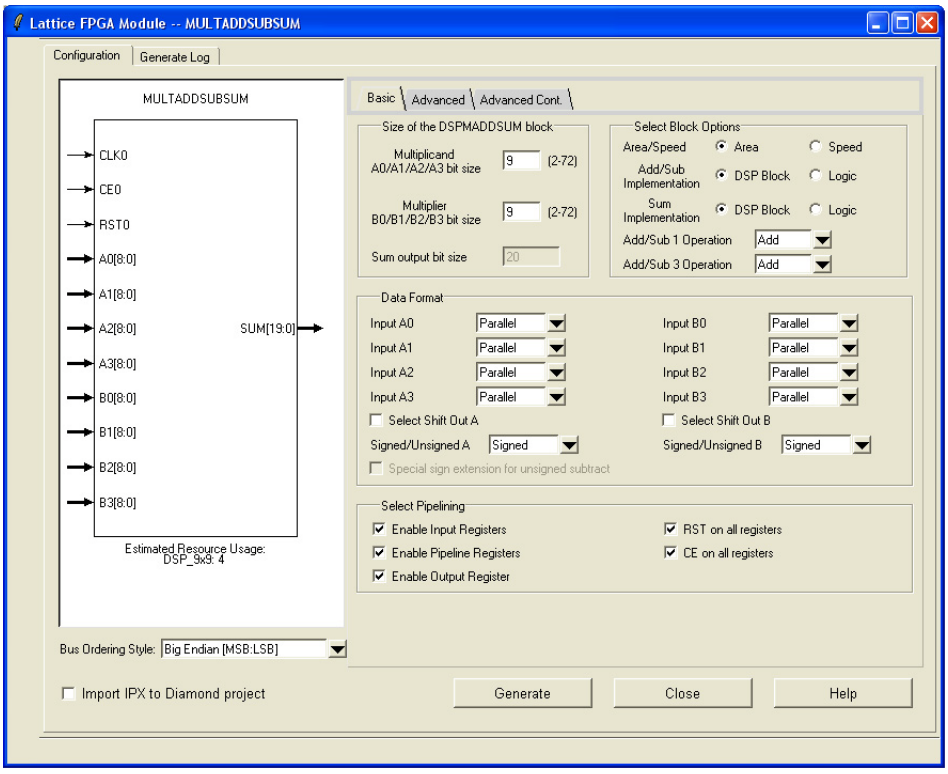


Figure 14-9. MULTADDSUBSUM Mode Advanced

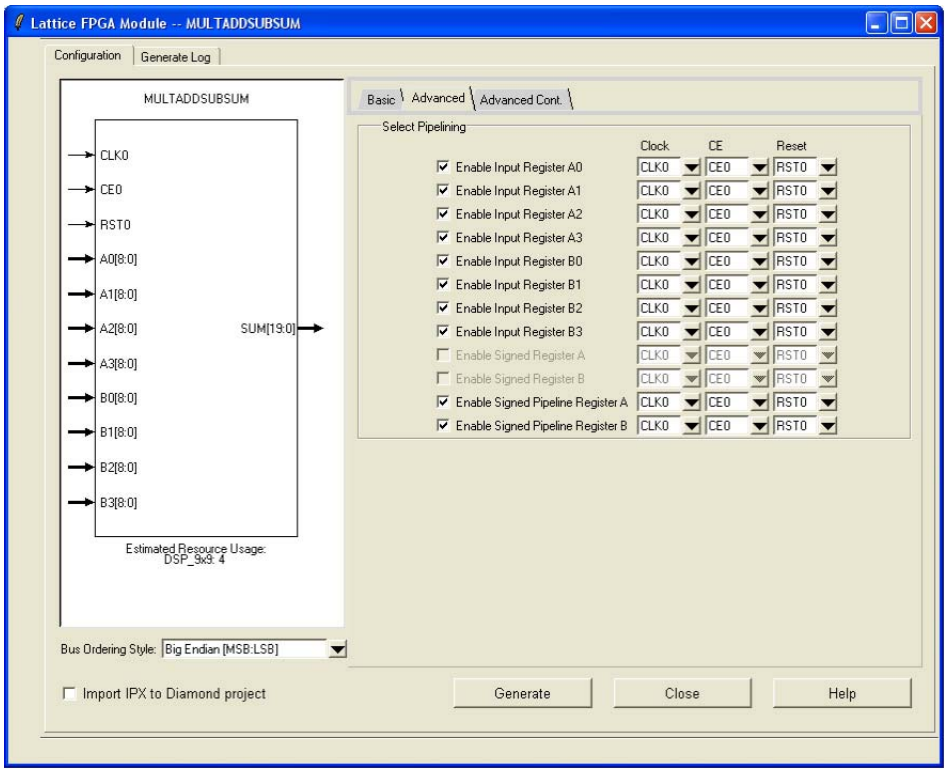
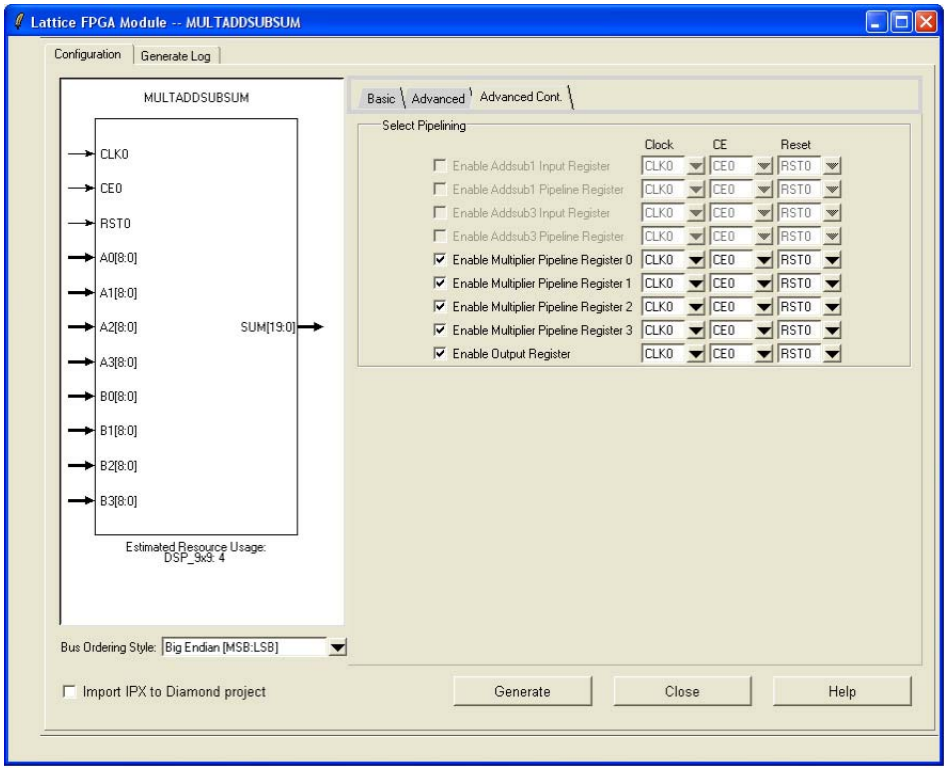


Figure 14-10. MULTADDSUBSUM Mode Advanced



Targeting the sysDSP Block by Inference

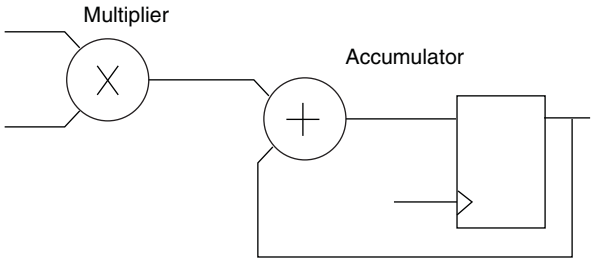
The Inferencing flow enables the design tools to infer sysDSP Blocks from a HDL design. It is important to note that when using the Inferencing flow, unless the code style matches the sysDSP Block, results will not be optimal. Consider the following Verilog and VHDL examples:

```
// This Verilog example will be mapped into single MULT18X18MACB with the output register enabled
module mult_acc (dataout, dataax, dataay, clk);
   output [16:0] dataout;
   input [7:0] dataax, dataay;
   input clk;
   reg  [16:0] dataout;

   wire [15:0] multa = dataax * dataay; // 9x9 Multiplier
   wire [16:0] adder_out;
   assign adder_out = multa + dataout; // Accumulator
   always @(posedge clk)
   begin
     dataout <= adder_out; // Output Register of the Accumulator
   end
endmodule

-- This VHDL example will be mapped into single MULT18X18MACB with all the registers enabled
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity mac is
port (clk, reset : in std_logic;
      dataax, dataay : in std_logic_vector(8 downto 0);
      dataout : out std_logic_vector(17 downto 0));
end;

architecture arch of mac is
signal dataax_reg, dataay_reg : std_logic_vector(8 downto 0);
signal multout, multout_reg : std_logic_vector(17 downto 0);
signal addout : std_logic_vector(17 downto 0);
signal dataout_reg : std_logic_vector(17 downto 0);
begin

dataout <= dataout_reg;

process (clk, reset)
begin
if (reset = '1') then
  dataax_reg <= (others => '0');
  dataay_reg <= (others => '0');
elsif (clk'event and clk='1') then
  dataax_reg <= dataax;
  dataay_reg <= dataay;
end if;
end process;

multout <= dataax_reg * dataay_reg;

process (clk, reset)
begin
if (reset = '1') then
  multout_reg <= (others => '0');
elsif (clk'event and clk='1') then
  multout_reg <= multout;
end if;
```

```
end process;

addout <= multout_reg + dataout_reg;

process (clk, reset)
begin
if (reset = '1') then
  dataout_reg <= (others => '0');
elsif (clk'event and clk='1') then
  dataout_reg <= addout;
end if;
end process;
end arch;
```

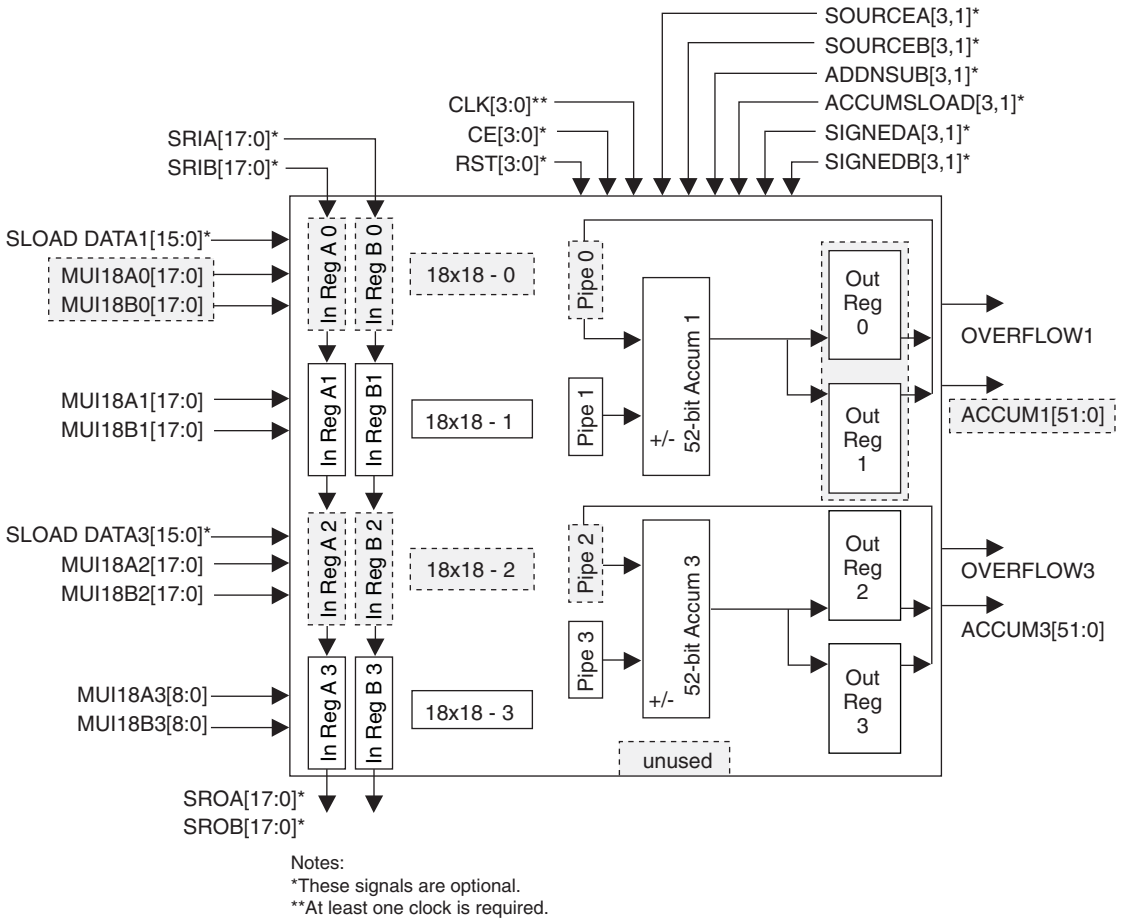
The above RTL will infer the following block diagram:

Figure 14-11. MULT18X18MACB Block Diagram



This block diagram can be mapped directly into the sysDSP primitives. Note that if a test point were added between the multiplier and the accumulator, or two output registers, etc. the code could not be mapped into a MULT18X18MACB of a sysDSP Block. Therefore, options that could be included in a design are input registers, pipeline registers, etc.

Figure 14-12. MAC18X18MACB Packed into a sysDSP Block



Notes:
*These signals are optional.
**At least one clock is required.

sysDSP Blocks in the Report File

To check the configuration of the sysDSP Blocks in your design you can look at the MAP and Place & Route (PAR) report files. The MAP report file shows the mapped sysDSP components/primitives in your design. The Place & Route (PAR) report file shows the number of components in each sysDSP Block. The report files that follow show how the inferred MAC was used.

MAP Report File

```
. MULT18X18MACB  addout_17_0:

Multiplier
        Operation               Unsigned
        Operation Registers     CLK     CE      RST
        ------------------------------------------
                Input
                Pipeline
        Operation Registers     CLK     CE      RST
        ------------------------------------------
                Input
                Pipeline
```

```
AddSub
        Operation               Add
        Operation Registers     CLK     CE      RST
        ------------------------------------------
                Input
                Pipeline
Data
        Input Registers         CLK     CE      RST
        ------------------------------------------
                A               CLK0    CE0     RST0
                B               CLK0    CE0     RST0
        Pipeline Registers      CLK     CE      RST
        ------------------------------------------
                Pipe            CLK0    CE0     RST0
        Output Register         CLK     CE      RST
        ------------------------------------------
                Output          CLK0    CE0     RST0
Other
        GSR     ENABLED
   Number Of Mapped DSP Components:
   --------------------------------
   MULT36X36B          0
   MULT18X18B          0
   MULT18X18MACB       1
   MULT18X18ADDSUBB    0
   MULT18X18ADDSUBSUMB 0
   MULT9X9B            0
   MULT9X9ADDSUBB      0
   MULT9X9ADDSUBSUMB   0
   --------------------------------
```

Place & Route (PAR) Report File

```
DSP Utilization Summary:
DSP Block #: 1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18
# of MULT36X36B
# of MULT18X18B
# of MULT18X18MACB                                        1
# of MULT18X18ADDSUBB
# of MULT18X18ADDSUBSUMB
# of MULT9X9B
# of MULT9X9ADDSUBB
# of MULT9X9ADDSUBSUMB
DSP Block  1    Component_Type         Instance_Name
DSP Block  2    Component_Type         Instance_Name
DSP Block  3    Component_Type         Instance_Name
DSP Block  4    Component_Type         Instance_Name
DSP Block  5    Component_Type         Instance_Name
DSP Block  6    Component_Type         Instance_Name
DSP Block  7    Component_Type         Instance_Name
DSP Block  8    Component_Type         Instance_Name
DSP Block  9    Component_Type         Instance_Name
  R45C81        MULT18X18MACB           addout_17_0
DSP Block 10    Component_Type         Instance_Name
DSP Block 11    Component_Type         Instance_Name
DSP Block 12    Component_Type         Instance_Name
DSP Block 13    Component_Type         Instance_Name
DSP Block 14    Component_Type         Instance_Name
DSP Block 15    Component_Type         Instance_Name
DSP Block 16    Component_Type         Instance_Name
DSP Block 17    Component_Type         Instance_Name
DSP Block 18    Component_Type         Instance_Name
```

Targeting the sysDSP Block Using Simulink

Simulink Overview

Simulink is a graphical add-on (similar to schematic entry) for Matlab®, which is produced by The MathWorks. For more information, refer to the Simulink web page at www.mathworks.com/products/simulink/.

Why is Simulink used?

- It allows users to create algorithms using floating point numbers.
- It helps users convert floating point algorithms into fixed point algorithms.

How does Simulink fit into the normal ispLEVER or Diamond design/process flow?

- Once you have converted have your algorithm working in fixed point. You can use the Lattice ispDSP block to create HDL files, which can be instantiated in your HDL design.

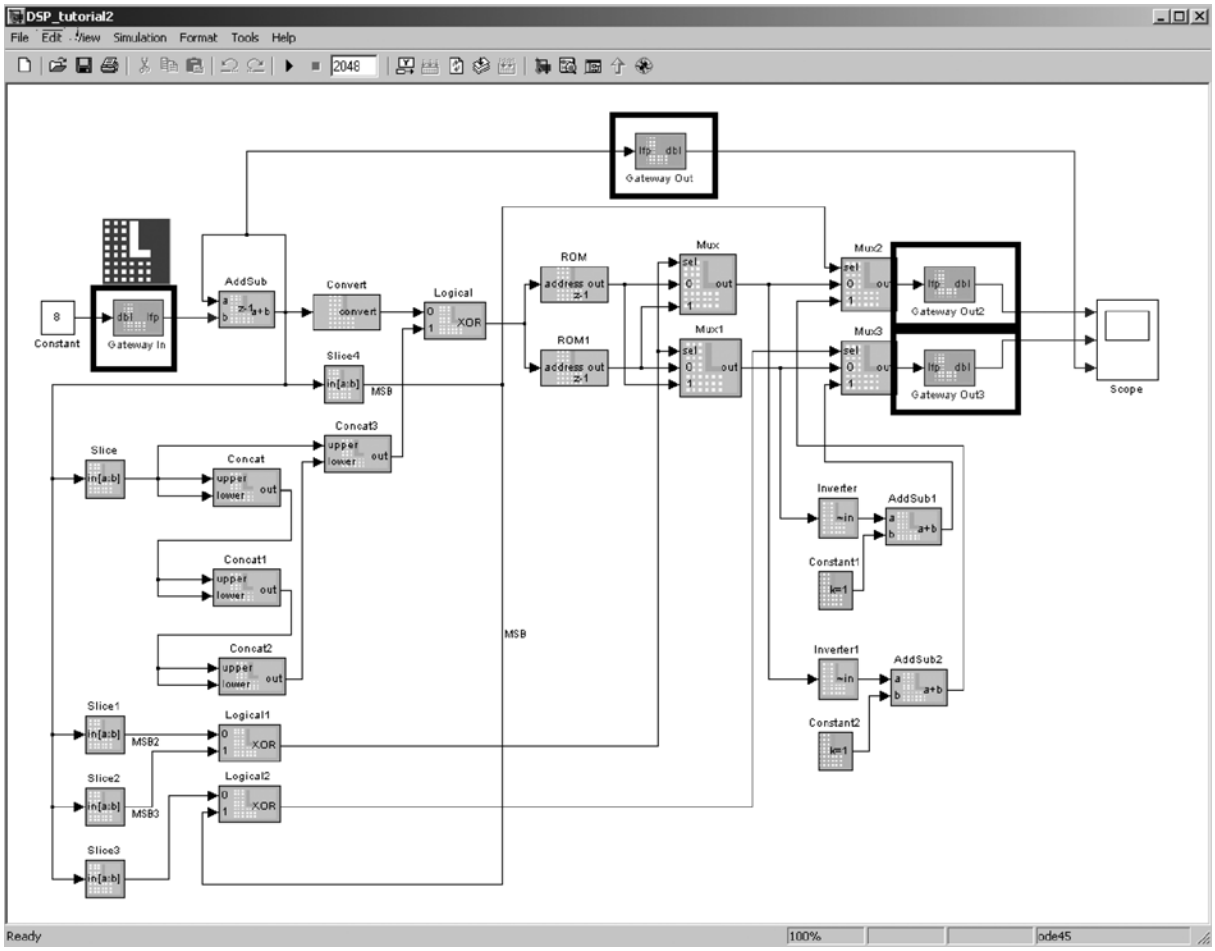
What does Lattice provide?

- Lattice provides a library of blocks for the Simulink tool, which include Multipliers, Adders, Registers, and other standard building blocks. Besides the basic building blocks there are a couple unique Lattice blocks:

  Gateways In and Out
  Everything between Gateways In and Out represents the HDL code. Everything before a Gateway In is the stimulus your test bench. Everything after the Gateway Out are the signals you will be monitoring in the test bench. Below is an example. The box on the left contains Gateway In blocks and the three boxes on the right contain Gateway Out blocks in Figure 14-13.

  Generate
  The Generate block is used to convert Fixed point Simulink design into HDL files which can be instantiated in a HDL design. The Generate block is identified by the Lattice logo and can be seen in Figure 14-13.

Figure 14-13. Simulink Design



Targeting the sysDSP Block by Instantiating Primitives

The sysDSP Block can be targeted by instantiating the sysDSP Block primitives into the design. The advantage of instantiating primitives is that it provides access to all ports and sets all available parameters. The disadvantage of this flow is that all this customization requires extra coding by the user. Appendix A details the syntax for the sysDSP Block primitives.

sysDSP Block Control Signal and Data Signal Descriptions

RST	Asynchronous reset of selected registers
CE	Clock Enable 1 = enabled, 0 = disabled
SIGNEDA	Dynamic signal: 0 = unsigned, 1 = signed
SIGNEDB	Dynamic signal: 0 = unsigned, 1 = signed
ACCUMSLOAD	Dynamic signal: 0 = accumulate, 1 = load
ADDNSUB	Dynamic signal: 0 = subtract, 1 = add
SOURCEA	Dynamic signal: 0 = parallel input, 1 = shift input
SOURCEB	Dynamic signal: 0 = parallel input, 1 = shift input

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
February 2006	01.0	Initial release.
September 2006	01.1	Updated table 1 with new software numbers.
		Updated figure 1 summation size from 37 to 38
November 2008	01.2	Updated sysDSP Block Control Signal and Data Signal Descriptions.
June 2010	01.3	Updated for Lattice Diamond design software support.

Appendix A. DSP Block Primitives

MULT18X18B

```
input A17,A16,A15,A14,A13,A12,A11,A10,A9,A8,A7,A6,A5,A4,A3,A2,A1,A0;
input B17,B16,B15,B14,B13,B12,B11,B10,B9,B8,B7,B6,B5,B4,B3,B2,B1,B0;
input SIGNEDA, SIGNEDB, SOURCEA, SOURCEB;
input CE0,CE1,CE2,CE3,CLK0,CLK1,CLK2,CLK3,RST0,RST1,RST2,RST3;
input SRIA17,SRIA16,SRIA15,SRIA14,SRIA13,SRIA12,SRIA11,SRIA10,SRIA9;
input SRIA8,SRIA7,SRIA6,SRIA5,SRIA4,SRIA3,SRIA2,SRIA1,SRIA0;
input SRIB17,SRIB16,SRIB15,SRIB14,SRIB13,SRIB12,SRIB11,SRIB10,SRIB9;
input SRIB8,SRIB7,SRIB6,SRIB5,SRIB4,SRIB3,SRIB2,SRIB1,SRIB0;
output SROA17,SROA16,SROA15,SROA14,SROA13,SROA12,SROA11,SROA10,SROA9;
output SROA8,SROA7,SROA6,SROA5,SROA4,SROA3,SROA2,SROA1,SROA0;
output SROB17,SROB16,SROB15,SROB14,SROB13,SROB12,SROB11,SROB10,SROB9;
output SROB8,SROB7,SROB6,SROB5,SROB4,SROB3,SROB2,SROB1,SROB0;
output P35,P34,P33,P32,P31,P30,P29,P28,P27,P26,P25,P24,P23,P22,P21,P20,P19,P18;
output P17,P16,P15,P14,P13,P12,P11,P10,P9,P8,P7,P6,P5,P4,P3,P2,P1,P0;
parameter REG_INPUTA_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_INPUTA_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_INPUTA_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_INPUTB_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_INPUTB_CE = = " CE0, CE1, CE2, CE3 ";
parameter REG_INPUTB_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_PIPELINE_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_PIPELINE_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_PIPELINE_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_OUTPUT_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_OUTPUT_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_OUTPUT_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_SIGNEDA_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_SIGNEDA_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_SIGNEDA_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_SIGNEDB_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_SIGNEDB_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_SIGNEDB_RST = " RST0, RST1, RST2, RST3 ";
parameter GSR = " Enabled, Disabled ";
```

MULT18X18ADDSUBB

```
input A017,A016,A015,A014,A013,A012,A011,A010,A09;
input A08,A07,A06,A05,A04,A03,A02,A01,A00;
input A117,A116,A115,A114,A113,A112,A111,A110,A19;
input A18,A17,A16,A15,A14,A13,A12,A11,A10;
input B017,B016,B015,B014,B013,B012,B011,B010,B09;
input B08,B07,B06,B05,B04,B03,B02,B01,B00;
input B117,B116,B115,B114,B113,B112,B111,B110,B19;
input B18,B17,B16,B15,B14,B13,B12,B11,B10;
input SIGNEDA, SIGNEDB, SOURCEA0, SOURCEA1, SOURCEB0, SOURCEB1, ADDNSUB;
input CE0,CE1,CE2,CE3,CLK0,CLK1,CLK2,CLK3,RST0,RST1,RST2,RST3;
input SRIA17,SRIA16,SRIA15,SRIA14,SRIA13,SRIA12,SRIA11,SRIA10,SRIA9;
input SRIA8,SRIA7,SRIA6,SRIA5,SRIA4,SRIA3,SRIA2,SRIA1,SRIA0;
input SRIB17,SRIB16,SRIB15,SRIB14,SRIB13,SRIB12,SRIB11,SRIB10,SRIB9;
input SRIB8,SRIB7,SRIB6,SRIB5,SRIB4,SRIB3,SRIB2,SRIB1,SRIB0;
output SROA17,SROA16,SROA15,SROA14,SROA13,SROA12,SROA11,SROA10,SROA9;
output SROA8,SROA7,SROA6,SROA5,SROA4,SROA3,SROA2,SROA1,SROA0;
output SROB17,SROB16,SROB15,SROB14,SROB13,SROB12,SROB11,SROB10,SROB9;
output SROB8,SROB7,SROB6,SROB5,SROB4,SROB3,SROB2,SROB1,SROB0;
output
SUM36,SUM35,SUM34,SUM33,SUM32,SUM31,SUM30,SUM29,SUM28,SUM27,SUM26,SUM25,SUM24,SUM23,SUM22,SUM21,SU
M20,SUM19,SUM18,SUM17,SUM16,SUM15,SUM14,SUM13,SUM12,SUM11,SUM10,SUM9,SUM8,SUM7,SUM6,SUM5,SUM4,SUM3
,SUM2,SUM1,SUM0;
```

```
parameter REG_INPUTA0_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_INPUTA0_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_INPUTA0_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_INPUTA1_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_INPUTA1_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_INPUTA1_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_INPUTB0_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_INPUTB0_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_INPUTB0_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_INPUTB1_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_INPUTB1_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_INPUTB1_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_PIPELINE0_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_PIPELINE0_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_PIPELINE0_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_PIPELINE1_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_PIPELINE1_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_PIPELINE1_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_OUTPUT_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_OUTPUT_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_OUTPUT_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_SIGNEDA_0_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_SIGNEDA_0_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_SIGNEDA_0_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_SIGNEDA_1_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_SIGNEDA_1_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_SIGNEDA_1_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_SIGNEDB_0_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_SIGNEDB_0_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_SIGNEDB_0_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_SIGNEDB_1_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_SIGNEDB_1_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_SIGNEDB_1_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_ADDNSUB_0_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_ADDNSUB_0_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_ADDNSUB_0_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_ADDNSUB_1_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_ADDNSUB_1_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_ADDNSUB_1_RST = " RST0, RST1, RST2, RST3 ";
parameter GSR = " Enabled, Disabled ";
```

MULT18X18ADDSUBSUMB

```
input A017,A016,A015,A014,A013,A012,A011,A010,A09;
input A08,A07,A06,A05,A04,A03,A02,A01,A00;
input A117,A116,A115,A114,A113,A112,A111,A110,A19;
input A18,A17,A16,A15,A14,A13,A12,A11,A10;
input A217,A216,A215,A214,A213,A212,A211,A210,A29;
input A28,A27,A26,A25,A24,A23,A22,A21,A20;
input A317,A316,A315,A314,A313,A312,A311,A310,A39;
input A38,A37,A36,A35,A34,A33,A32,A31,A30;
input B017,B016,B015,B014,B013,B012,B011,B010,B09;
input B08,B07,B06,B05,B04,B03,B02,B01,B00;
input B117,B116,B115,B114,B113,B112,B111,B110,B19;
input B18,B17,B16,B15,B14,B13,B12,B11,B10;
input B217,B216,B215,B214,B213,B212,B211,B210,B29;
input B28,B27,B26,B25,B24,B23,B22,B21,B20;
input B317,B316,B315,B314,B313,B312,B311,B310,B39;
input B38,B37,B36,B35,B34,B33,B32,B31,B30;
input SIGNEDA, SIGNEDB,ADDNSUB1,ADDNSUB3;
input SOURCEA0, SOURCEA1, SOURCEA2, SOURCEA3;
input SOURCEB0, SOURCEB1, SOURCEB2, SOURCEB3;
```

```
input CE0,CE1,CE2,CE3,CLK0,CLK1,CLK2,CLK3,RST0,RST1,RST2,RST3;
input SRIA17,SRIA16,SRIA15,SRIA14,SRIA13,SRIA12,SRIA11,SRIA10,SRIA9;
input SRIA8,SRIA7,SRIA6,SRIA5,SRIA4,SRIA3,SRIA2,SRIA1,SRIA0;
input SRIB17,SRIB16,SRIB15,SRIB14,SRIB13,SRIB12,SRIB11,SRIB10,SRIB9;
input SRIB8,SRIB7,SRIB6,SRIB5,SRIB4,SRIB3,SRIB2,SRIB1,SRIB0;
output SROA17,SROA16,SROA15,SROA14,SROA13,SROA12,SROA11,SROA10,SROA9;
output SROA8,SROA7,SROA6,SROA5,SROA4,SROA3,SROA2,SROA1,SROA0;
output SROB17,SROB16,SROB15,SROB14,SROB13,SROB12,SROB11,SROB10,SROB9;
output SROB8,SROB7,SROB6,SROB5,SROB4,SROB3,SROB2,SROB1,SROB0;
output
SUM37,SUM36,SUM35,SUM34,SUM33,SUM32,SUM31,SUM30,SUM29,SUM28,SUM27,SUM26,SUM25,SUM24,SUM23,SUM22,SU
M21,SUM20,SUM19,SUM18,SUM17,SUM16,SUM15,SUM14,SUM13,SUM12,SUM11,SUM10,SUM9,SUM8,SUM7,SUM6,SUM5,SUM
4,SUM3,SUM2,SUM1,SUM0;
parameter REG_INPUTA0_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_INPUTA0_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_INPUTA0_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_INPUTA1_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_INPUTA1_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_INPUTA1_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_INPUTA2_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_INPUTA2_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_INPUTA2_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_INPUTA3_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_INPUTA3_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_INPUTA3_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_INPUTB0_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_INPUTB0_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_INPUTB0_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_INPUTB1_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_INPUTB1_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_INPUTB1_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_INPUTB2_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_INPUTB2_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_INPUTB2_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_INPUTB3_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_INPUTB3_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_INPUTB3_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_PIPELINE0_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_PIPELINE0_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_PIPELINE0_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_PIPELINE1_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_PIPELINE1_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_PIPELINE1_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_PIPELINE2_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_PIPELINE2_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_PIPELINE2_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_PIPELINE3_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_PIPELINE3_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_PIPELINE3_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_OUTPUT_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_OUTPUT_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_OUTPUT_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_SIGNEDA_0_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_SIGNEDA_0_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_SIGNEDA_0_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_SIGNEDA_1_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_SIGNEDA_1_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_SIGNEDA_1_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_SIGNEDB_0_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_SIGNEDB_0_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_SIGNEDB_0_RST = " RST0, RST1, RST2, RST3 ";
```

```
parameter REG_SIGNEDB_1_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_SIGNEDB_1_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_SIGNEDB_1_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_ADDNSUB1_0_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_ADDNSUB1_0_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_ADDNSUB1_0_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_ADDNSUB1_1_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_ADDNSUB1_1_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_ADDNSUB1_1_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_ADDNSUB3_0_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_ADDNSUB3_0_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_ADDNSUB3_0_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_ADDNSUB3_1_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_ADDNSUB3_1_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_ADDNSUB3_1_RST = " RST0, RST1, RST2, RST3 ";
parameter GSR = " Enabled, Disabled ";
```

MULT18X18MACB

```
input A17,A16,A15,A14,A13,A12,A11,A10,A9;
input A8,A7,A6,A5,A4,A3,A2,A1,A0;
input B17,B16,B15,B14,B13,B12,B11,B10,B9;
input B8,B7,B6,B5,B4,B3,B2,B1,B0;
input ADDNSUB, SIGNEDA, SIGNEDB,ACCUMSLOAD;
input SOURCEA, SOURCEB;
input CE0,CE1,CE2,CE3,CLK0,CLK1,CLK2,CLK3,RST0,RST1,RST2,RST3;
input LD51, LD50, LD49, LD48, LD47, LD46, LD45, LD44, LD43, LD42, LD41, LD40
input LD39, LD38, LD37, LD36, LD35, LD34, LD33, LD32, LD31, LD30
input LD29, LD28, LD27, LD26, LD25, LD24, LD23, LD22, LD21, LD20
input LD19, LD18, LD17, LD16, LD15, LD14, LD13, LD12, LD11, LD10
input LD9, LD8, LD7, LD6, LD5, LD4, LD3, LD2, LD1, LD0;
input SRIA17,SRIA16,SRIA15,SRIA14,SRIA13,SRIA12,SRIA11,SRIA10,SRIA9;
input SRIA8,SRIA7,SRIA6,SRIA5,SRIA4,SRIA3,SRIA2,SRIA1,SRIA0;
input SRIB17,SRIB16,SRIB15,SRIB14,SRIB13,SRIB12,SRIB11,SRIB10,SRIB9;
input SRIB8,SRIB7,SRIB6,SRIB5,SRIB4,SRIB3,SRIB2,SRIB1,SRIB0;
output SROA17,SROA16,SROA15,SROA14,SROA13,SROA12,SROA11,SROA10,SROA9;
output SROA8,SROA7,SROA6,SROA5,SROA4,SROA3,SROA2,SROA1,SROA0;
output SROB17,SROB16,SROB15,SROB14,SROB13,SROB12,SROB11,SROB10,SROB9;
output SROB8,SROB7,SROB6,SROB5,SROB4,SROB3,SROB2,SROB1,SROB0;
output
ACCUM51,ACCUM50,ACCUM49,ACCUM48,ACCUM47,ACCUM46,ACCUM45,ACCUM44,ACCUM43,ACCUM42,ACCUM41,ACCUM40,AC
CUM39,ACCUM38,ACCUM37,ACCUM36,ACCUM35,ACCUM34,ACCUM33,ACCUM32,ACCUM31,ACCUM30,ACCUM29,ACCUM28,ACCU
M27,ACCUM26,ACCUM25,ACCUM24,ACCUM23,ACCUM22,ACCUM21,ACCUM20,ACCUM19,ACCUM18,ACCUM17,ACCUM16,ACCUM1
5,ACCUM14,ACCUM13,ACCUM12,ACCUM11,ACCUM10,ACCUM9,ACCUM8,ACCUM7,ACCUM6,ACCUM5,ACCUM4,ACCUM3,ACCUM2,
ACCUM1,ACCUM0,OVERFLOW;
parameter REG_INPUTA_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_INPUTA_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_INPUTA_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_INPUTB_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_INPUTB_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_INPUTB_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_PIPELINE_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_PIPELINE_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_PIPELINE_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_OUTPUT_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_OUTPUT_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_OUTPUT_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_SIGNEDA_0_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_SIGNEDA_0_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_SIGNEDA_0_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_SIGNEDA_1_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_SIGNEDA_1_CE = " CE0, CE1, CE2, CE3 ";
```

```
parameter REG_SIGNEDA_1_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_SIGNEDB_0_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_SIGNEDB_0_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_SIGNEDB_0_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_SIGNEDB_1_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_SIGNEDB_1_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_SIGNEDB_1_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_ACCUMSLOAD_0_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_ACCUMSLOAD_0_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_ACCUMSLOAD_0_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_ACCUMSLOAD_1_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_ACCUMSLOAD_1_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_ACCUMSLOAD_1_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_ADDNSUB_0_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_ADDNSUB_0_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_ADDNSUB_0_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_ADDNSUB_1_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_ADDNSUB_1_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_ADDNSUB_1_RST = " RST0, RST1, RST2, RST3 ";
parameter GSR = " Enabled, Disabled ";
```

MULT36X36B

```
input A35,A34,A33,A32,A31,A30,A29,A28,A27,A26,A25,A24,A23,A22,A21,A20,A19,A18;
input A17,A16,A15,A14,A13,A12,A11,A10,A9,A8,A7,A6,A5,A4,A3,A2,A1,A0;
input B35,B34,B33,B32,B31,B30,B29,B28,B27,B26,B25,B24,B23,B22,B21,B20,B19,B18;
input B17,B16,B15,B14,B13,B12,B11,B10,B9,B8,B7,B6,B5,B4,B3,B2,B1,B0;
input SIGNEDA, SIGNEDB;
input CE0,CE1,CE2,CE3,CLK0,CLK1,CLK2,CLK3,RST0,RST1,RST2,RST3;
output P71,P70,P69,P68,P67,P66,P65,P64,P63,P62,P61,P60,P59,P58,P57,P56,P55,P54;
output P53,P52,P51,P50,P49,P48,P47,P46,P45,P44,P43,P42,P41,P40,P39,P38,P37,P36;
output P35,P34,P33,P32,P31,P30,P29,P28,P27,P26,P25,P24,P23,P22,P21,P20,P19,P18;
output P17,P16,P15,P14,P13,P12,P11,P10,P9,P8,P7,P6,P5,P4,P3,P2,P1,P0;
parameter REG_INPUTA_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_INPUTA_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_INPUTA_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_INPUTB_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_INPUTB_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_INPUTB_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_PIPELINE_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_PIPELINE_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_PIPELINE_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_OUTPUT_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_OUTPUT_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_OUTPUT_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_SIGNEDA_0_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_SIGNEDA_0_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_SIGNEDA_0_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_SIGNEDA_1_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_SIGNEDA_1_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_SIGNEDA_1_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_SIGNEDB_0_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_SIGNEDB_0_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_SIGNEDB_0_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_SIGNEDB_1_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_SIGNEDB_1_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_SIGNEDB_1_RST = " RST0, RST1, RST2, RST3 ";
parameter GSR = " Enabled, Disabled ";
```

MULT9X9B

```
input A8,A7,A6,A5,A4,A3,A2,A1,A0;
input B8,B7,B6,B5,B4,B3,B2,B1,B0;
input SIGNEDA, SIGNEDB, SOURCEA, SOURCEB;
input CE0,CE1,CE2,CE3,CLK0,CLK1,CLK2,CLK3,RST0,RST1,RST2,RST3;
input SRIA8,SRIA7,SRIA6,SRIA5,SRIA4,SRIA3,SRIA2,SRIA1,SRIA0;
input SRIB8,SRIB7,SRIB6,SRIB5,SRIB4,SRIB3,SRIB2,SRIB1,SRIB0;
output SROA8,SROA7,SROA6,SROA5,SROA4,SROA3,SROA2,SROA1,SROA0;
output SROB8,SROB7,SROB6,SROB5,SROB4,SROB3,SROB2,SROB1,SROB0;
output P17,P16,P15,P14,P13,P12,P11,P10,P9,P8,P7,P6,P5,P4,P3,P2,P1,P0;
parameter REG_INPUTA_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_INPUTA_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_INPUTA_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_INPUTB_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_INPUTB_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_INPUTB_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_PIPELINE_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_PIPELINE_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_PIPELINE_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_OUTPUT_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_OUTPUT_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_OUTPUT_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_SIGNEDA_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_SIGNEDA_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_SIGNEDA_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_SIGNEDB_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_SIGNEDB_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_SIGNEDB_RST = " RST0, RST1, RST2, RST3 ";
parameter GSR = " Enabled, Disabled ";
```

MULT9X9ADDSUBB

```
input A08,A07,A06,A05,A04,A03,A02,A01,A00;
input A18,A17,A16,A15,A14,A13,A12,A11,A10;
input B08,B07,B06,B05,B04,B03,B02,B01,B00;
input B18,B17,B16,B15,B14,B13,B12,B11,B10;
input SIGNEDA, SIGNEDB,ADDNSUB;
input SOURCEA0, SOURCEA1, SOURCEB0, SOURCEB1;
input CE0,CE1,CE2,CE3,CLK0,CLK1,CLK2,CLK3,RST0,RST1,RST2,RST3;
input SRIA8,SRIA7,SRIA6,SRIA5,SRIA4,SRIA3,SRIA2,SRIA1,SRIA0;
input SRIB8,SRIB7,SRIB6,SRIB5,SRIB4,SRIB3,SRIB2,SRIB1,SRIB0;
output SROA8,SROA7,SROA6,SROA5,SROA4,SROA3,SROA2,SROA1,SROA0;
output SROB8,SROB7,SROB6,SROB5,SROB4,SROB3,SROB2,SROB1,SROB0;
output
SUM18,SUM17,SUM16,SUM15,SUM14,SUM13,SUM12,SUM11,SUM10,SUM9,SUM8,SUM7,SUM6,SUM5,SUM4,SUM3,SUM2,SUM1
,SUM0;
parameter REG_INPUTA0_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_INPUTA0_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_INPUTA0_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_INPUTA1_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_INPUTA1_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_INPUTA1_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_INPUTB0_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_INPUTB0_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_INPUTB0_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_INPUTB1_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_INPUTB1_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_INPUTB1_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_PIPELINE0_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_PIPELINE0_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_PIPELINE0_RST = " RST0, RST1, RST2, RST3 ";
```

```
parameter REG_PIPELINE1_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_PIPELINE1_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_PIPELINE1_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_OUTPUT_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_OUTPUT_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_OUTPUT_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_SIGNEDA_0_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_SIGNEDA_0_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_SIGNEDA_0_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_SIGNEDA_1_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_SIGNEDA_1_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_SIGNEDA_1_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_SIGNEDB_0_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_SIGNEDB_0_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_SIGNEDB_0_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_SIGNEDB_1_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_SIGNEDB_1_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_SIGNEDB_1_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_ADDNSUB_0_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_ADDNSUB_0_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_ADDNSUB_0_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_ADDNSUB_1_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_ADDNSUB_1_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_ADDNSUB_1_RST = " RST0, RST1, RST2, RST3 ";
parameter GSR = " Enabled, Disabled ";
```

MULT9X9ADDSUBSUMB

```
input A08,A07,A06,A05,A04,A03,A02,A01,A00;
input A18,A17,A16,A15,A14,A13,A12,A11,A10;
input A28,A27,A26,A25,A24,A23,A22,A21,A20;
input A38,A37,A36,A35,A34,A33,A32,A31,A30;
input B08,B07,B06,B05,B04,B03,B02,B01,B00;
input B18,B17,B16,B15,B14,B13,B12,B11,B10;
input B28,B27,B26,B25,B24,B23,B22,B21,B20;
input B38,B37,B36,B35,B34,B33,B32,B31,B30;
input SIGNEDA, SIGNEDB,ADDNSUB1,ADDNSUB3;
input SOURCEA0, SOURCEA1, SOURCEA2, SOURCEA3;
input SOURCEB0, SOURCEB1, SOURCEB2, SOURCEB3;
input CE0,CE1,CE2,CE3,CLK0,CLK1,CLK2,CLK3,RST0,RST1,RST2,RST3;
input SRIA8,SRIA7,SRIA6,SRIA5,SRIA4,SRIA3,SRIA2,SRIA1,SRIA0;
input SRIB8,SRIB7,SRIB6,SRIB5,SRIB4,SRIB3,SRIB2,SRIB1,SRIB0;
output SROA8,SROA7,SROA6,SROA5,SROA4,SROA3,SROA2,SROA1,SROA0;
output SROB8,SROB7,SROB6,SROB5,SROB4,SROB3,SROB2,SROB1,SROB0;
output
SUM19,SUM18,SUM17,SUM16,SUM15,SUM14,SUM13,SUM12,SUM11,SUM10,SUM9,SUM8,SUM7,SUM6,SUM5,SUM4,SUM3,SUM
2,SUM1,SUM0;
parameter REG_INPUTA0_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_INPUTA0_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_INPUTA0_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_INPUTA1_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_INPUTA1_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_INPUTA1_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_INPUTA2_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_INPUTA2_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_INPUTA2_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_INPUTA3_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_INPUTA3_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_INPUTA3_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_INPUTB0_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_INPUTB0_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_INPUTB0_RST = " RST0, RST1, RST2, RST3 ";
```

```
parameter REG_INPUTB1_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_INPUTB1_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_INPUTB1_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_INPUTB2_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_INPUTB2_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_INPUTB2_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_INPUTB3_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_INPUTB3_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_INPUTB3_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_PIPELINE0_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_PIPELINE0_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_PIPELINE0_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_PIPELINE1_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_PIPELINE1_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_PIPELINE1_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_PIPELINE2_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_PIPELINE2_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_PIPELINE2_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_PIPELINE3_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_PIPELINE3_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_PIPELINE3_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_OUTPUT_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_OUTPUT_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_OUTPUT_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_SIGNEDA_0_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_SIGNEDA_0_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_SIGNEDA_0_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_SIGNEDA_1_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_SIGNEDA_1_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_SIGNEDA_1_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_SIGNEDB_0_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_SIGNEDB_0_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_SIGNEDB_0_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_SIGNEDB_1_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_SIGNEDB_1_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_SIGNEDB_1_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_ADDNSUB1_0_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_ADDNSUB1_0_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_ADDNSUB1_0_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_ADDNSUB1_1_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_ADDNSUB1_1_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_ADDNSUB1_1_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_ADDNSUB3_0_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_ADDNSUB3_0_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_ADDNSUB3_0_RST = " RST0, RST1, RST2, RST3 ";
parameter REG_ADDNSUB3_1_CLK = " NONE, CLK0, CLK1, CLK2, CLK3 ";
parameter REG_ADDNSUB3_1_CE = " CE0, CE1, CE2, CE3 ";
parameter REG_ADDNSUB3_1_RST = " RST0, RST1, RST2, RST3 ";
parameter GSR = " Enabled, Disabled ";
```

Appendix B. Using IPexpress for Diamond

Invoking IPexpress for Diamond

There are several ways IPexpress can be invoked. To invoke IPexpress from the Start menu, select:

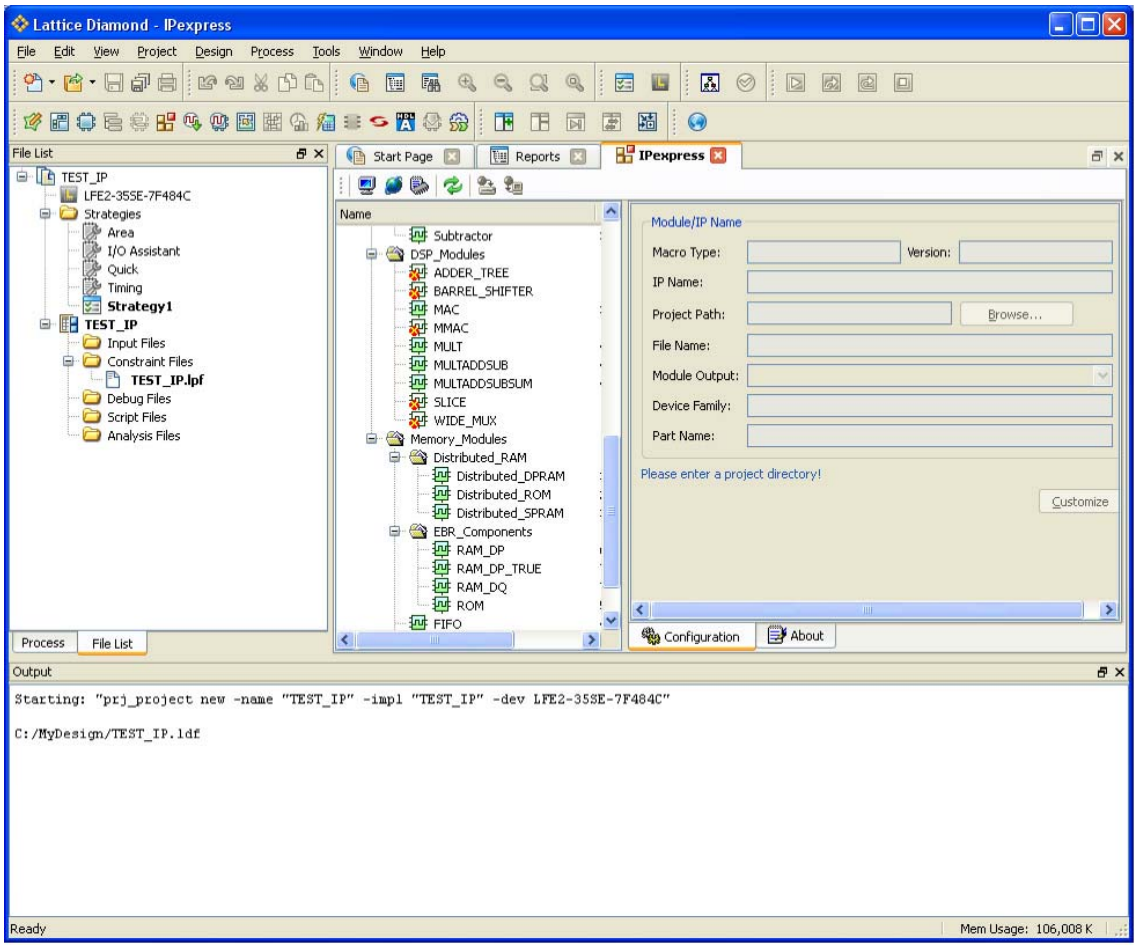
Start > Programs > Lattice Diamond 1.0 > Accessories > IPexpress

To invoke IPexpress from within Diamond, a project must be opened, then invoke the IPexpress icon or select:

Tools > IPexpress

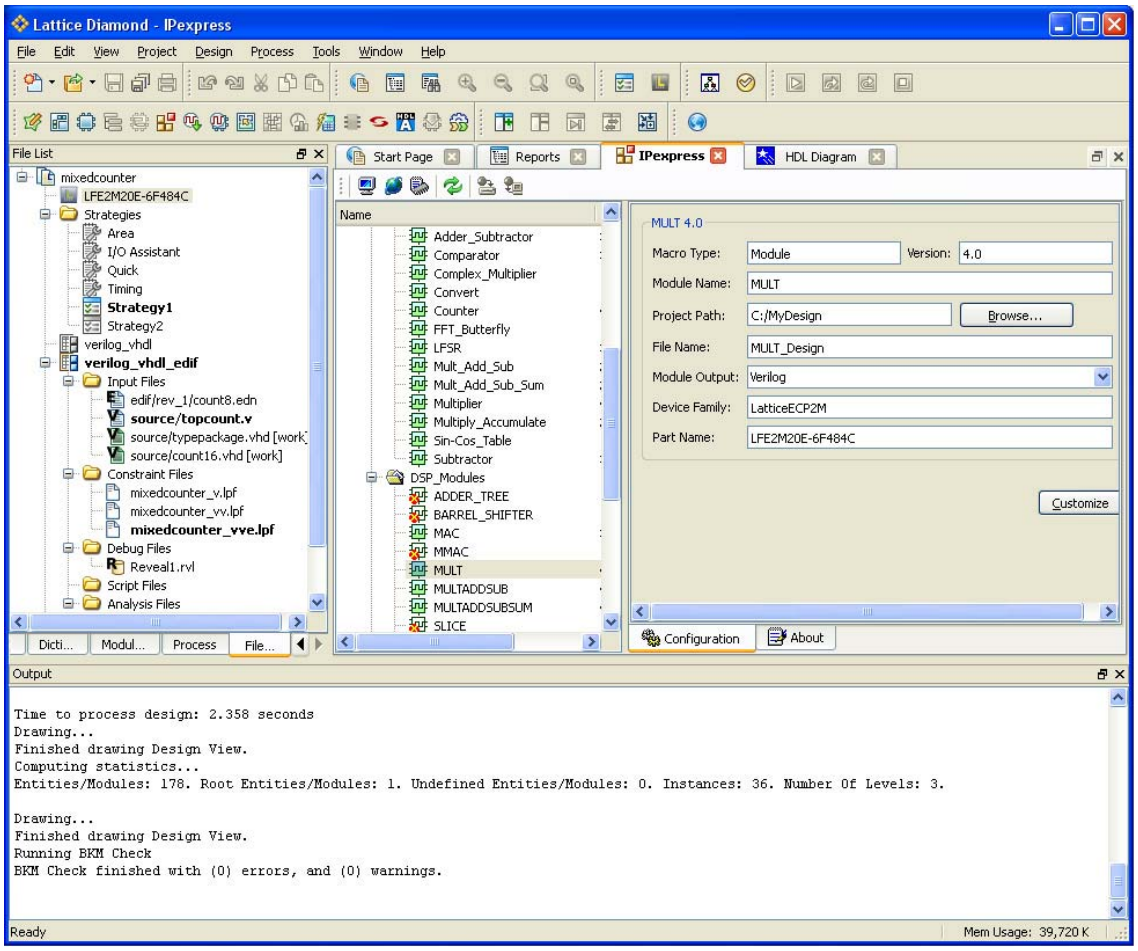
The IPexpress interface appears as shown in Figure 14-14.

Figure 14-14. IPexpress Interface



Scroll to the modules and left-click on the module you wish to use. Enter the information into the IPexpress interface as shown in the example Figure 14-15.

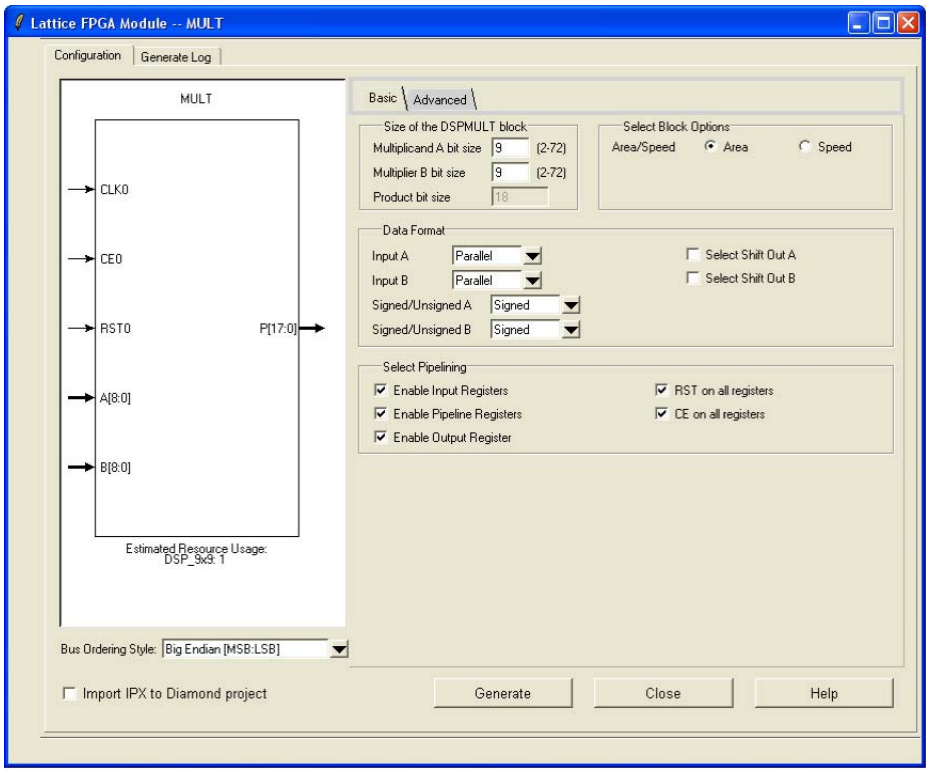
Figure 14-15. Creating the Module Instance



Select **Customize**.

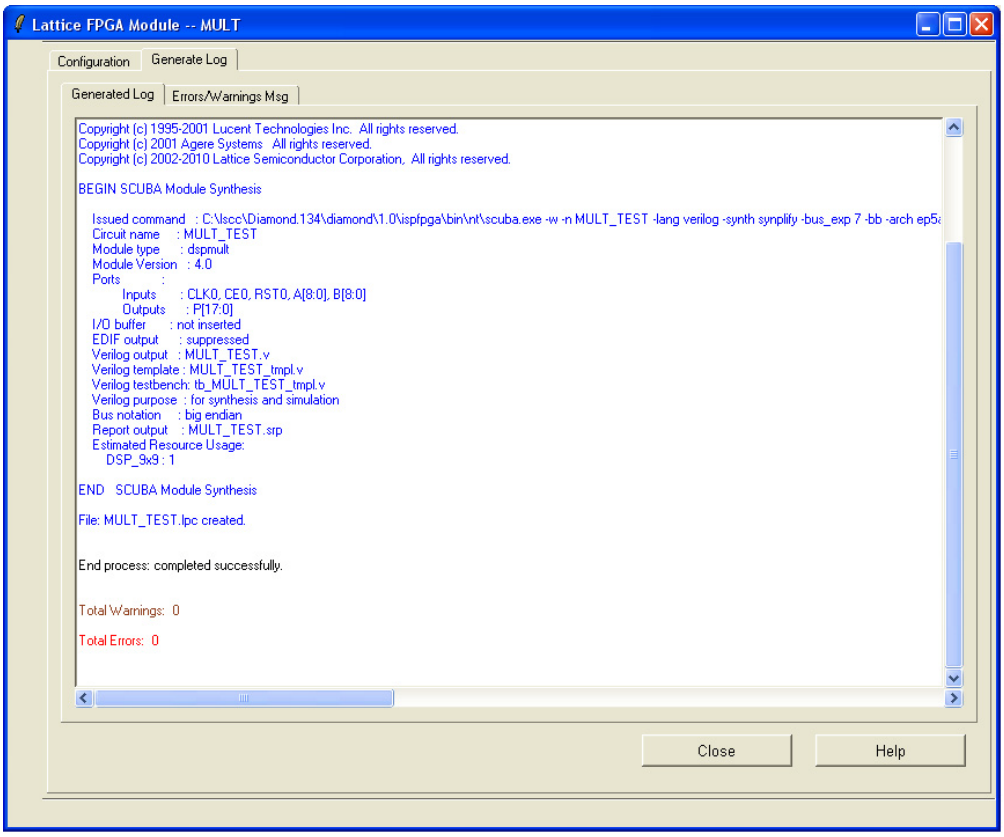
An example of a MULT Module Dialog window appears as shown in Figure 14-16. Select **Help** for information about the fields in this window.

Figure 14-16. MULT Mode Basic Set-up



When you are finished selecting your options, select **Generate**. A log window will appear similar to what is shown in Figure 14-17.

Figure 14-17. IP Generation Log Window



LatticeECP2/M sysCONFIG Usage Guide

March 2011 Technical Note TN1108

Introduction

The configuration memory in the LatticeECP2™ and LatticeECP2M™ FPGAs is built using volatile SRAM; therefore, an external non-volatile configuration memory is required to maintain the configuration data when the power is removed. This non-volatile memory supplies the configuration data to the LatticeECP2/M when it powers-up, or any other time the device needs to be updated.

To support multiple configuration options the LatticeECP2/M supports the Lattice sysCONFIG™ interface, as well as the dedicated ispJTAG™ port. The available configuration options, or ports, are listed in Table 15-1.

Table 15-1. Supported Configuration Ports

Interface	Port
sysCONFIG	SPI
	SPIm
	Slave Serial
	Slave Parallel
ispJTAG	JTAG (IEEE 1149.1 and IEEE 1532 compliant)

This technical note covers all of the configuration options available for LatticeECP2/M.

General Configuration Flow

The LatticeECP2/M will enter configuration mode when one of three things happens, power is applied to the chip, the PROGRAMN pin is driven low, or when a JTAG Refresh instruction is issued. Upon entering configuration mode the INITN pin and the DONE pin are driven low to indicate that the device is initializing, i.e. getting ready to receive configuration data.

Once the LatticeECP2/M has finished initializing, the INITN pin will be driven high. The low to high transition of the INITN pin causes the CFG pins to be sampled, telling the LatticeECP2/M which port it is going to configure from. The LatticeECP2/M then begins reading data from the selected port and starts looking for the preamble, BDB3 (hex). All data after the preamble is valid configuration data.

When the LatticeECP2/M has finished reading all of the configuration data, assuming there have been no errors, the DONE pin goes high and the LatticeECP2/M enters user mode, in other words the device begins to function according to the user's design.

Note that the LatticeECP2/M may also be programmed via JTAG. When programming via JTAG, the INITN and DONE signals have no meaning, because JTAG, per the IEEE standard, takes complete control of the chip and it's I/Os.

The Lattice ECP2/M devices are also available in an "S" version which supports the use of an encrypted bitstream configuration file. These versions have the same configuration options as the standard versions, except where noted in this document. When using these devices, the user should refer to the LatticeECP2/M Family Data Sheet and TN1109, LatticeECP2/M Configuration Encryption Usage Guide, in addition to this document, to understand the configuration requirements.

The following sections define each configuration pin, each configuration mode, and all of the configuration options for the LatticeECP2/M.

Configuration Pins

The LatticeECP2/M supports two types of configuration pins, dedicated and dual-purpose. The dedicated pins are used exclusively for configuration; the dual-purpose pins, when not being used for configuration, are available as extra I/O pins. If a dual-purpose pin is to be used both for configuration and as a general purpose I/O (GPIO) the user must adhere to the following:

- The I/O type must remain the same, in other words if the pin is a 3.3V CMOS pin (LVCMOS33) during configuration it must remain a 3.3V CMOS pin as a GPIO.
- The user must select the correct CONFIG_MODE setting and set the PERSISTENT bit to OFF in order to use the dual-purpose sysCONFIG pins as GPIO after configuration. In ispLEVER® these preferences can be set in the Design Planner. If you are using Lattice Diamond™ design software, select **Tools > Spreadsheet View** and then select the **Global Preferences** tab in the Spreadsheet View.
- The user is responsible for insuring that no internal or external logic will interfere with device configuration.

Also, if slave parallel configuration mode is not being used then one or both of the parallel port chip selects (CSN, CS1N) must be high or tri-state during configuration.

Programmable options control the direction and type of each dual-purpose configuration pin. These options are controlled via pin preferences in Lattice ispLEVER and Diamond software, or as HDL source file attributes.

The LatticeECP2/M also supports ispJTAG for configuration, transparent read back, and JTAG testing. The following sections describe the function of the various sysCONFIG and JTAG pins. Table 15-2 is provided for reference.

Table 15-2. Configuration Pins for the LatticeECP2/M

Pin Name	I/O Type	Pin Type	Mode Used
CFG[2:0]	Input, weak pull-up	Dedicated	All
PROGRAMN	Input, weak pull-up	Dedicated	All
INITN	Bi-Directional Open Drain, weak pull-up	Dedicated	All
DONE	Bi-Directional Open Drain with weak pull-up, or Active Drive	Dedicated	All
CCLK	Input or Output	Dedicated	All
DI/CSSPI0N[2]	Input, weak pull up	Dual-Purpose	Serial, SPI, SPIm
DOUT/CSON[2]	Output	Dual-Purpose	Parallel, Serial, SPI
CSN[2]	Input, weak pull-up	Dual-Purpose	Parallel
CS1N[2]	Input, weak pull-up	Dual-Purpose	Parallel
WRITEN[2]	Input, weak pull-up	Dual-Purpose	Parallel
BUSY/SISPI[2]	Output, tri-state, weak pull-up	Dual-Purpose	Serial, SPI, SPIm
D[0]/SPIFASTN[2]	Input or Output	Dual-Purpose	Parallel, SPI, SPIm
D[1:6][2]			Parallel
D[7]/SPID0[2]			Parallel, SPI, SPIm
TDI	Input, weak pull-up	Dedicated	JTAG
TDO	Output, weak pull-up	Dedicated	JTAG
TCK	Input with Hysteresis	Dedicated	JTAG
TMS	Input, weak pull-up	Dedicated	JTAG

1. Weak pull-ups consist of a current source of 30µA to 150µA. The pull-ups for sysCONFIG dedicated and dual-purpose pins track V_{CCIO8}; the pull-ups for TDI, TDO, and TMS track V_{CCJ}.
2. The sysCONFIG pins on the LatticeECP2M50/M70/M100 are dedicated sysCONFIG pins. The sysCONFIG output pins are actively driven during normal device operation.

Dedicated Control Pins

The following sub-sections describe the LatticeECP2/M dedicated sysCONFIG pins. These pins are powered by V_{CCIO8}.

While the device is under IEEE 1149.1 or 1532 JTAG control the dedicated programming pins have no meaning. This is because a boundary scan cell will control each pin, per JTAG 1149.1, rather than normal internal logic.

CFG[2:0]
The Configuration Mode pins, CFG[2:0], are dedicated inputs with weak pull-ups. The CFG pins are sampled on the rising edge of INITN and are used to select the configuration mode, i.e. what type of device the LatticeECP2/M will configure from. As a consequence the CFG pins determine which groups of dual-purpose pins will be used for device configuration (see the right-most column in Table 15-2). See Table 15-3 for a list of Configuration Modes.

Table 15-3. Configuration Modes

Configuration Mode		CFG[2]	CFG[1]	CFG[0]	D[0]/SPIFASTN
SPI	Normal (0x03)	0	0	0	Pull-Up
	Fast (0x0B)	0	0	0	Pull-Down
Reserved		0	0	1	X
SPIm	Normal (0x03)	0	1	0	Pull-Up
	Fast (0x0B)	0	1	0	Pull-Down
Reserved		0	1	1	X
Reserved		1	0	0	X
Slave Serial		1	0	1	X
Reserved		1	1	0	X
Slave Parallel		1	1	1	D0

Notes:
JTAG is always available for IEEE 1149.1 and 1532 support.

PROGRAMN
The PROGRAMN pin is a dedicated input with a weak pull-up. This pin is used to initiate a non-JTAG SRAM configuration sequence.

A high to low signal applied to PROGRAMN takes the device out of user mode and sets it into configuration mode. The low to high transition will initiate the configuration process. The PROGRAMN pin can be used to trigger configuration at any time.

INITN
The INITN pin is a bidirectional open drain control pin. INITN is capable of driving a low pulse out as well as detecting a low pulse driven in.

When the PROGRAMN pin is driven low, or a JTAG Reset instruction is received, or after the internal Power-On-Reset signal is released during power-up, the INITN pin will be driven low to reset the internal configuration circuitry. Once the PROGRAMN pin is driven high, the configuration initialization begins. Once the configuration initialization is completed, the INITN pin will go high. To delay configuration the INITN pin can be held low externally. The device will not enter configuration mode as long as the INITN pin is held low. A low to high transition on INITN causes the CFG pins to be sampled, telling the LatticeECP2/M which port to use, and starts configuration.

During configuration the INITN pin becomes an error detection pin. If a CRC error is detected during configuration INITN will be driven low. The error will be cleared at the beginning of the next configuration.

DONE
The DONE pin is a dedicated bi-directional open drain with a weak pull-up (default), or it is an actively driven pin.

DONE goes low when INITN goes low, when INITN and PROGRAMN go high, and the internal Done bit is programmed at the end of configuration, the DONE pin will be released (or driven high, if it is an actively driven pin). The DONE pin can be held low externally and, depending on the wake-up sequence selected, the device will not become functional until the DONE pin is externally brought high. Externally delaying the wake-up sequence using the DONE pin is a good way to synchronize the wake-up of multiple FPGAs; it is also required when configuring multiple FPGAs from a single configuration device.

Sampling the DONE pin is a good way for an external device to tell if the FPGA has finished configuration. However, when using IEEE 1532 JTAG to configure SRAM the DONE pin is driven by a boundary scan cell, so the state of the DONE pin has no meaning during IEEE 1532 JTAG configuration (once configuration is complete, DONE reverts to internal logic and will be high).

CCLK
CCLK is a dedicated bi-directional pin; direction depends on whether a Master or Slave mode is selected. If a Master mode (SPI or SPIm) is selected, via the CFG pins, the CCLK pin becomes an output; otherwise CCLK is an input.

If the CCLK pin becomes an output, the internal programmable oscillator is connected to CCLK and is driven out to slave devices. CCLK will stop 120 clock cycles after the DONE pin is brought high. The extra clock cycles ensure that enough clocks are provided to wake-up other devices in the chain. When stopped, CCLK becomes an input (tri-stated output). CCLK will restart (become an output again) on the next configuration initialization sequence.

The MCCLK_FREQ parameter (one of the global preferences in the Design Planner of ispLEVER or the Spreadsheet View in Diamond) controls the CCLK master frequency (see data sheet On-Chip Oscillator section for the frequency selection). The software default setting for the configuration CCLK is 2.5 MHz. For a complete list of the supported Master Clock frequencies, please see the LatticeECP2/M Family Data Sheet. One of the first operations during configuration is the MCCLK_FREQ parameter; once this parameter is loaded the frequency changes to the selected value. Care should be exercised not to exceed the frequency specification of the slave devices or the signal integrity capabilities of the PCB layout.

When downloading an encrypted bitstream file to the LatticeECP2/M S-Series devices, the user must adhere to the appropriate conditions for the CCLK signal. These conditions are shown in TN1109, LatticeECP2/M Configuration Encryption Usage Guide.

Dual-Purpose sysCONFIG Pins

The following is a list of the dual-purpose sysCONFIG pins. If any of these pins are used for configuration and for user I/O, the user must adhere to the requirements listed at the start of the Configuration pin sections. On LatticeECP2M50/M70/M100 devices, the sysCONFIG pins described below are dedicated pins. When using the same pins to access the external boot Flash, the system design must take care of tri-stating these output pins while driving these pins from a different I/O pin.

These pins are powered by V_{CCIO8}.

DI/CSSPI0N
The DI/CSSPI0N dual-purpose pin is designated as DI (Data Input) for Serial configurations. DI has an internal weak pull-up. DI captures data on the rising edge of CCLK.

In SPI or SPIm mode the DI/CSSPI0N becomes a low true Chip Select output that drives the SPI Serial Flash chip select.

DOUT/CSON
The DOUT/CSON pin is an output pin and has two purposes.

For serial and parallel configuration modes, when BYPASS mode is selected, this pin becomes DOUT (see Figure 15-9). When the device is fully configured a Bypass instruction in the bitstream is executed and the data on DI, or D[0:7] in the case of a parallel configuration mode, will then be routed to the DOUT pin. This allows data to be passed, serially, to the next device. In a parallel configuration mode D0 will be shifted out first followed by D1, D2, and so on.

For parallel configuration mode there is a Flowthrough option as well. When Flowthrough mode is selected this pin becomes Chip Select Out (CSON). When the device is fully configured, and the Flowthrough instruction in the bitstream is executed, the CSON pin is driven low to enable the next device. The data pins, D[0:7], are wired in parallel to each device in the chain (see Figure 15-10).

In SPIm mode, the sysCONFIG daisy chaining mode of configuration is not supported.

The DOUT/CSON drives out a high on power-up and will continue to do so until the execution of the Bypass/Flow Through instruction within the bitstream, or until the I/O Type is changed by the user code.

CSN and CS1N
Both CSN and CS1N are active low input pins with weak pull-ups and are used in parallel mode only. These inputs are OR'ed and used to enable the D[0:7] data pins to receive or output a byte of data. *Note: In the 144-pin TQFP and 208-pin PQFP packages, CSN, CS1N and WRITEN are not bonded out.*

When CSN or CS1N is high, the D[0:7], and BUSY pins are tri-stated. CSN and CS1N are interchangeable when controlling the D[0:7], and BUSY pins. Driving both CSN and CS1N high causes the LatticeECP2/M to exit Bypass or Flowthrough mode and resets the Bypass register. If Bypass or Flowthrough mode will not be used then CSN or CS1N may be tied low, i.e. in this case it is only required that one of these pins be driven. The CSN and CS1N pins must remain low while the configuration bitstream is being sent to the device or the configuration will fail.

If SRAM (configuration memory) needs to be accessed using the parallel pins while the part is in user mode (the DONE pin is high) then the PERSISTENT preference must be set to ON to preserve these pins as CSN and CS1N. CSN and CS1N are not connected in the 100-pin TQFP and 208-pin PQFP devices. Note that SRAM may only be read using JTAG or Slave Parallel mode.

WRITEN
The WRITEN pin is an active low input with a weak pull-up and used for parallel mode only. *Note: In the 144-pin TQFP and 208-pin PQFP packages, CSN, CS1N and WRITEN are not bonded out.*

The WRITEN pin is used to determine the direction of the data pins D[0:7]. The WRITEN pin must be driven low in order to clock a byte of data into the device and driven high to clock data out of the device.

If SRAM (configuration memory) needs to be accessed using the parallel pins while the part is in user mode (the DONE pin is high) then the PERSISTENT preference must be set to ON to preserve this pin as WRITEN. WRITEN is not connected in the 100-pin TQFP and 208-pin PQFP devices. Note that SRAM may only be read using JTAG or Slave Parallel mode.

BUSY/SISPI
The BUSY/SISPI pin has two functions.

In parallel configuration mode, the BUSY pin is a tri-stated output. The BUSY pin will be driven low by the device only when it is ready to receive a byte of data on D[0:7] or a byte of data is ready for reading. The BUSY pin allows the LatticeECP2/M to pause transfers on the parallel port.

If SRAM (configuration memory) needs to be accessed using the parallel pins while the part is in user mode (the DONE pin is high) then the PERSISTENT preference must be set to ON to preserve this pin as BUSY. Note that SRAM may only be read using JTAG or Slave Parallel mode.

In SPI or SPIm configuration modes, the BUSY/SISPI pin becomes an output that drives control and data to the SPI Serial Flash. Control and data are output on the falling edge of CCLK. If SPI memory needs to be accessed

using the SPI port while the part is in user mode (the DONE pin is high) then the PERSISTENT preference must be set to ON to preserve this pin as SISPI.

D[0]/SPIFASTN
The D[0]/SPIFASTN pin has two functions.

In parallel mode this pin is D[0] and operates in the same way as D[1:6] below. Taken together D[0:7] form the parallel data bus, D[0] is the most significant bit in the byte. As with D[1:6], if SRAM (configuration memory) needs to be accessed using the parallel pins while the part is in user mode (the DONE pin is high) then the PERSISTENT preference must be set to ON to preserve this pin as D[0]. Note that SRAM may only be read using JTAG or Slave Parallel mode.

In SPI or SPIm mode the D[0]/SPIFASTN pin becomes an input. SPIFASTN is sampled on the rising edge of INITN. If SPIFASTN is high the LatticeECP2/M will use SPI Serial Flash read op-code 03 (hex). Read op-code 03 (hex) is the standard read command used by all “25” series SPI Serial Flash. If SPIFASTN is low the LatticeECP2/M will use SPI Serial Flash fast read op-code 0B (hex). The fast read op-code 0B (hex) accommodates higher frequency read clocks, exact clock speeds can be found in the SPI Serial Flash manufacturer’s data sheet.

If SPI memory needs to be accessed using the SPI port while the part is in user mode (the DONE pin is high) then the PERSISTENT preference must be set to ON to preserve this pin as SPIFASTN.

When using the SPI or SPIM mode the SPIFASTN pin should either be tied high or low. It must not be left floating or configuration problems will occur.

Not all SPI Serial Flash support the 0B (hex) fast read op-code, consult the manufacturer’s data sheet. Care must also be taken not to exceed the signal integrity capabilities of the PCB layout.

D[1:6]
The D[1:6] pins support parallel mode only. The D[1:6] pins are tri-statable bi-directional I/O pins used for data write and read. When the WRITEN signal is low, and the CSN and CS1N pins are low, the D[1:6] pins become data inputs. When the WRITEN signal is driven high, and the CSN and CS1N pins are low, the D[1:6] pins become data outputs. If either CSN or CS1N is high D[1:6] will be tri-state.

If SRAM (configuration memory) needs to be accessed using the parallel pins while the part is in user mode (the DONE pin is high) then the PERSISTENT preference must be set to ON to preserve this pin as D[1:6]. Note that SRAM may only be read using JTAG or Slave Parallel mode.

Care must be exercised during read back of EBR or PFU memory. It is up to the user to ensure that reading these RAMs will not cause data corruption; corruption may be caused when these RAMs are read while being accessed by user code.

D[7]/SPID0
The D[7]/SPID0 pin has two functions.

In parallel mode this pin is D[7] and operates in the same way as D[1:6] above. Taken together D[0:7] form the parallel data bus, D[7] is the least significant bit in the byte. As with D[1:6], if SRAM (configuration memory) needs to be accessed using the parallel pins while the part is in user mode (the DONE pin is high) then the PERSISTENT preference must be set to ON to preserve this pin as D[7]. Note that SRAM may only be read using JTAG or Slave Parallel mode.

In SPI or SPIm mode the D[7]/SPID0 pin becomes an input and should be wired to the output data pin of the SPI Serial Flash. The data on SPID0 is clocked in on the rising edge of CCLK. If SPI memory needs to be accessed using the SPI port while the part is in user mode (the DONE pin is high) then the PERSISTENT preference must be set to ON to preserve this pin as SPID0.

ispJTAG Pins

The ispJTAG pins are standard IEEE 1149.1 TAP (Test Access Port) pins. The ispJTAG pins are dedicated pins and are always accessible when the LatticeECP2/M device is powered up. While the device is under 1149.1 or 1532

JTAG control the dedicated programming pins INITN, DONE, and CCLK have no meaning. This is because a boundary scan cell will control each pin, per the IEEE standard, rather than normal internal logic. While the LatticeECP2/M is under JTAG control the PROGRAMN pin will be ignored.

These pins are powered by V_{CCJ}.

TDO
The Test Data Output pin is used to shift out serial test instructions and data. When TDO is not being driven by the internal circuitry, the pin will be in a high impedance state. This pin should be wired to TDO of the JTAG connector, or to TDI of a downstream device in a JTAG chain. An internal pull-up resistor on the TDO pin is provided. The internal resistor is pulled up to V_{CCJ}.

TDI
The Test Data Input pin is used to shift in serial test instructions and data. This pin should be wired to TDI of the JTAG connector, or to TDO of an upstream device in a JTAG chain. An internal pull-up resistor on the TDI pin is provided. The internal resistor is pulled up to V_{CCJ}.

TMS
The Test Mode Select pin controls test operations on the TAP controller. On the falling edge of TCK, depending on the state of TMS, a transition will be made in the TAP controller state machine. An internal pull-up resistor on the TMS pin is provided. The internal resistor is pulled up to V_{CCJ}.

TCK
The test clock pin, TCK, provides the clock to run the TAP controller state machine, which loads and unloads the JTAG data and instruction registers. TCK can be stopped in either the high or low state and can be clocked at frequencies up to that indicated in the device data sheet. The TCK pin supports hysteresis; the typical hysteresis is approximately 100mV when V_{CCJ} = 3.3V. The TCK pin does not have a pull-up. A pull-down resistor between TCK and ground on the PCB of 4.7 K is recommended to avoid inadvertent clocking of the TAP controller as V_{CC} ramps up.

When downloading an encrypted bitstream file to the LatticeECP2/M S-Series devices, the user must adhere to the appropriate conditions for the TCK signal. These conditions are shown in TN1109, LatticeECP2/M Configuration Encryption Usage Guide.

Optional TRST
Test Reset, TRST, in not supported on the LatticeECP2/M.

V_{CCJ}
Having a separate JTAG V_{CC} (V_{CCJ}) pin lets the user apply a voltage level to the JTAG port that is independent from the rest of the device. Valid voltage levels are 3.3V, 2.5V, 1.8V, 1.5V, and 1.2V, but the voltage used must match the other voltages in the JTAG chain. V_{CCJ} must be connected even if JTAG is not used.

Please see In-System Programming Design Guidelines for ispJTAG Devices for further information.

Configuration and JTAG Pin Physical Description
All of the sysCONFIG dedicated and dual-purpose pins are part of Bank 8. Bank 8 V_{CCIO} determines the output voltage level of these pins, input thresholds are determined by the I/O Type selected in the ispLEVER Design Planner (default is 3.3V LVCMOS) or Diamond Spreadsheet View.

JTAG voltage levels and thresholds are determined by the V_{CCJ} pin, allowing the LatticeECP2/M to accommodate JTAG chain voltages from 1.2V to 3.3V.

Configuration Modes

The LatticeECP2/M devices support many different configuration modes, utilizing either serial or parallel data paths. On power-up, when a JTAG Refresh instruction is issued, or when the PROGRAMN pin is toggled, the CFG[2:0] pins are sampled to determine the configuration mode. See Table 15-3 above for a list of available configuration modes.

The following sub-sections break down each configuration mode. For more information on the options for each mode, see the section below entitled Configuration Options.

SPI Mode

The LatticeECP2/M offers a direct connection to memories that support the SPI Serial Flash standard (see Table 15-6). By setting the configuration pins, CFG[2:0], to all zeros the LatticeECP2/M will configure from the SPI interface. The SPI interface supports two configuration topologies:

- One FPGA configured from one SPI Serial Flash
- Multiple FPGAs configured from one SPI Serial Flash

The required boot memory size for each of the ECP2/M device sizes is shown in Table 15-4. The values shown are for a single LatticeECP2/M device. The size for a dual-boot application would be twice that shown.

Table 15-4. Maximum Configuration Bits - SPI Flash Mode Bitstream File

Device	Bitstream Size (Mb)	SPI Flash (Mb)	Dual Boot SPI Flash (Mb)
ECP2-6	1.5	2	4 / 8[2]
ECP2-12	2.9	4	8
ECP2-20	4.5	8	16
ECP2-35	6.3	8	16
ECP2-50	8.9	16	32
ECP2-70	13.3	16	32
ECP2M-20	5.9	8	16
ECP2M-35	9.8	16	32
ECP2M-50	15.8	16	64
ECP2M-70	19.8	32	64
ECP2M-100	25.6	32	64

1. These values apply for both encrypted and unencrypted bitstream files in the SPI Flash mode except as noted below.
2. For the LatticeECP2-6 S-Series device, the dual Boot Flash size required is 8 Mb due to the number of sectors required.

Table 15-5. Maximum Configuration Bits - Serial and Parallel Mode Bitstream File

	All Modes	Slave Serial Mode	Slave Parallel Mode
Device	Unencrypted Bitstream Size (Mb)	Encrypted Bitstream Size (Mb)	Encrypted Bitstream Size (Mb)
ECP2-6	1.5	2.3	7.5
ECP2-12	2.9	4.3	14.2
ECP2-20	4.5	6.7	22.0
ECP2-35	6.3	9.4	31.2
ECP2-50	8.9	13.4	44.3
ECP2-70	13.3	20.0	66.1
ECP2M-20	5.9	8.9	29.5
ECP2M-35	9.8	14.8	48.9
ECP2M-50	15.8	23.7	78.6
ECP2M-70	19.8	29.7	98.6
ECP2M-100	25.6	38.5	127.6

The estimated time for configuration can be calculated by dividing the bitstream size (in bits) from Table 15-4 by the CCLK frequency. The CCLK frequency can be set using the global preferences tab within the ispLEVER Design Planner or the Spreadsheet View (Global Preferences tab) in Diamond. For more information on setting the CCLK frequency, please see the Master Clock section and the D[0]/SPIFASTN pin section of this document.

When downloading an encrypted bitstream file to the LatticeECP2/M S-Series devices, the user must adhere to the appropriate conditions for the CCLK signal. These conditions are shown in TN1109, LatticeECP2/M Configuration Encryption Usage Guide.

Table 15-6. SPI Serial Flash Vendor List

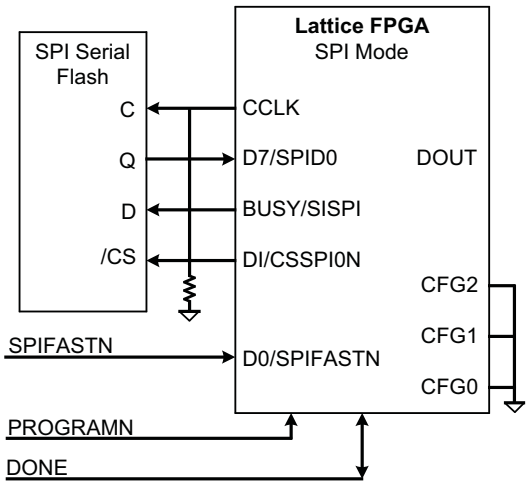
Vendor	Part Number
ST Microelectronics	M25Pxx
Winbond	W25Pxx
Silicon Storage Technology	SST25VFxx, SST25LFxx
Spansion	S25FLxx
Atmel	AT25Fxx
NexFlash	NX25Pxx
Macronix	MX25Lxx

Note: This is not meant to be an exhaustive list and may be updated from time to time.

One FPGA, One SPI Flash

The simplest SPI configuration consists of one SPI Serial Flash connected to one LatticeECP2/M, as shown in Figure 15-1. This is also the recommended method for use when downloading an encrypted bitstream file to the LatticeECP2/M S-Series devices.

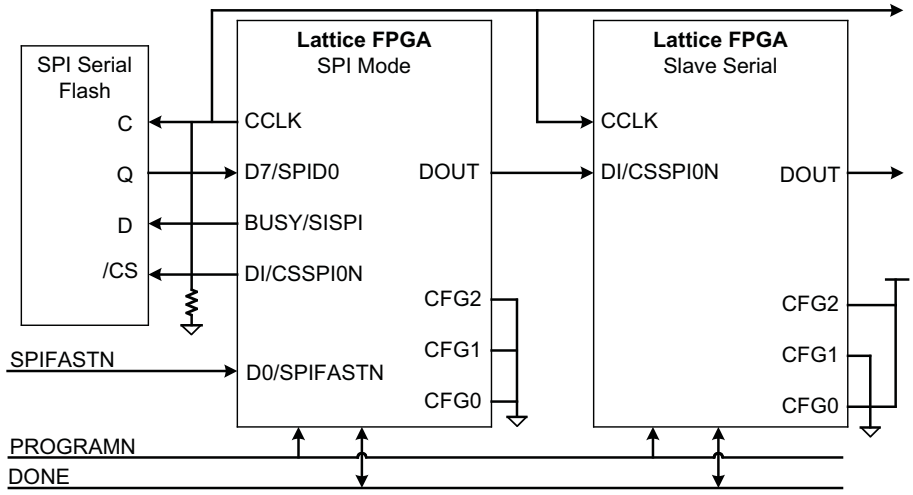
Figure 15-1. One FPGA, One SPI Serial Flash



Multiple FPGA, One SPI Flash

With a sufficiently large SPI Flash multiple FPGAs can be configured as shown in Figure 15-2. The first FPGA is configured in SPI mode; the following FPGAs are configured in Slave Serial mode.

Figure 15-2. Multiple FPGAs, One SPI Serial Flash



Note: This method is not available when using encrypted bitstream files with the LatticeECP2/M S-Series devices. Please refer to the *LatticeECP2/M S-Series Configuration Encryption Usage Guide*, TN1109, for more information about using encrypted bitstream files.

SPIm Mode

Externally, except for the CFG pins, SPIm mode looks like SPI mode, they use the same SPI Serial Flash devices, and are wired to the FPGA in the same way (see Figure 15-6). The SPIm mode does not support multiple FPGAs being configured from one SPI serial Flash as shown in Figure 15-2. Internally the two modes are treated differently.

SPI mode treats the SPI Serial Flash as a single block of storage starting at address zero. The SPIm mode treats the SPI flash memory as discrete blocks of memory rather than a single block of memory. This allows the storage of separate configuration images in separate blocks of memory. SPIm supports dual configuration (or boot) images, here referred to as a “golden” image and a primary image.

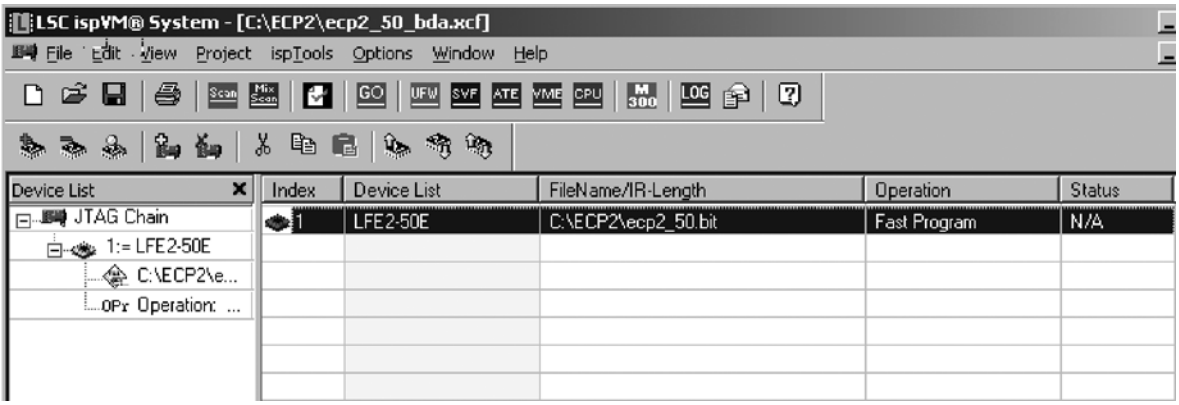
A golden image is used when there is the possibility that corrupt data could be inadvertently loaded into the SPI Serial Flash, such as during remote updates. For instance, if the Flash erase or program procedure is interrupted, perhaps due to a power failure, then the Flash will contain corrupt data and the system could be rendered inoperable. Ideally the FPGA should detect that the data is corrupt and boot from a known good, or golden, boot image. This is exactly what SPIm does.

The golden image is stored at the beginning of the Flash address space; the updatable, or primary, image is above the golden image. During configuration, if the FPGA detects data corruption in the primary image, it will automatically reboot from the golden image. Each time the FPGA powers up, the PROGRAMN pin is toggled, or a JTAG Refresh instruction is issued, it will try to configure from the primary image first. Note that if the LatticeECP2/M detects that the data in the golden image is corrupt as well INITN will be driven low and the part will stop trying to configure.

Dual Boot Image Setup

In order to use dual configuration files the files must first be properly stored within the SPI Serial Flash. Lattice ispVM™ System software makes this easy.

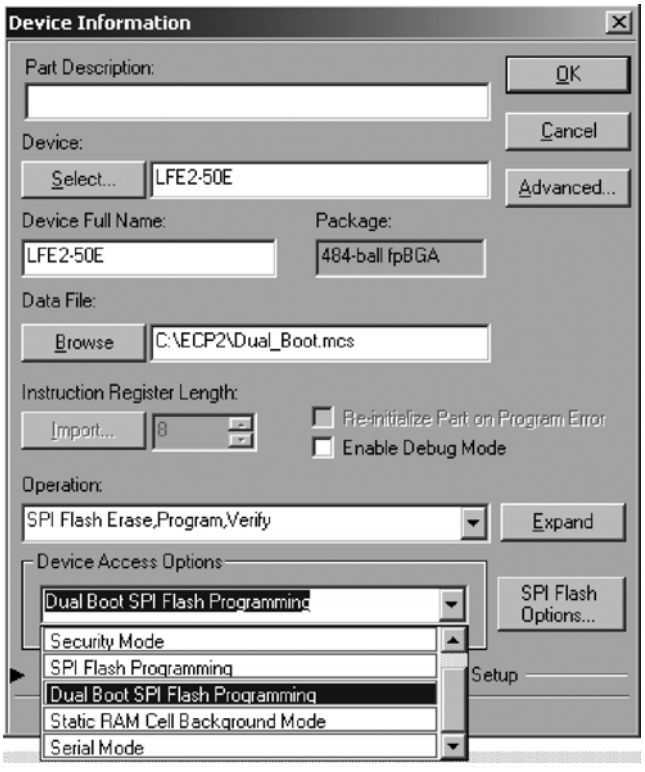
Figure 15-3. ispVM Main Window



First, create the desired files using ispLEVER or Diamond. Depending on the application, these files might be the same design or different designs. There is nothing special about these files, in other words they contain all of the information needed to fully configure the FPGA.

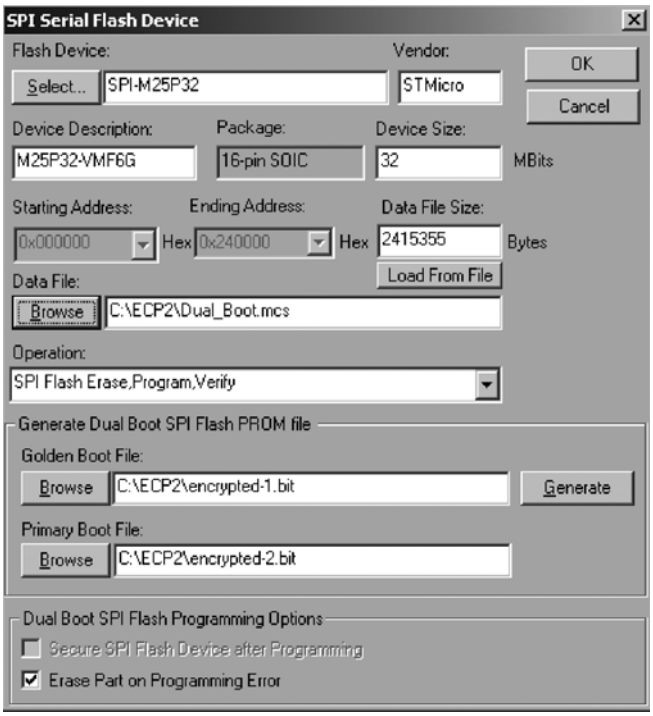
Next, open ispVM (see Figure 15-3), do a scan of the board by clicking on the **SCAN** button on the toolbar, and then double click on the row in the chain that has the LatticeECP2/M. You will now see the Device Information window (see Figure 15-4).

Figure 15-4. Device Information Window



From the Device Options drop-down box select **Dual Boot SPI Flash Programming**. Then click on the **SPI Flash Options** button to open the SPI Serial Flash Device window (see Figure 15-5). In this window click on **Select** and choose the SPI Flash that's mounted on the board.

Figure 15-5. SPI Serial Flash Device Window



Second, click on **Browse** to select an output file name.

Third, select the operation, such as **SPI Flash Erase**, **Program**, **Verify**.

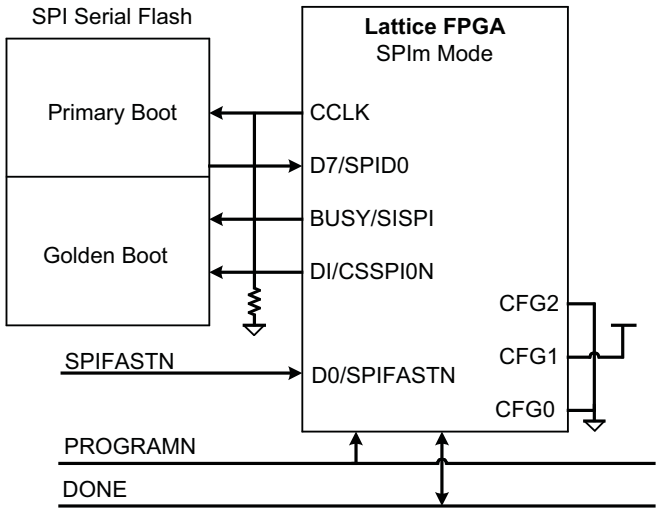
Fourth, select the golden file and primary file. Note that these may be encrypted for the LatticeECP2/M "S" version or non-encrypted files. If they are encrypted files they must both be encrypted with the same key.

And finally, click on the **Generate** button. When file generation is complete, click on **OK** to get back to the main ispVM window. To program the file into the SPI Serial Flash just click on the **GO** button on the toolbar.

At power-up, when the PROGRAMN pin is toggled, or when a JTAG Refresh instruction is issued, the LatticeECP2/M reads the primary file from SPI Serial Flash. If a CRC error is found then the LatticeECP2/M will re-boot automatically, reading configuration data from the golden file instead of the primary file. Note that if an error is found in the golden file the LatticeECP2/M will drive the INITN pin low and stop trying to configure.

Note that the flow specified here is for initial programming of the SPI Serial Flash. During a field update of the SPI Serial Flash it is expected that only the primary configuration file will be updated, not the golden file. This is important because it guarantees the availability of a known good configuration file.

Figure 15-6. SPIm, Dual Configuration Images



Note: The CSSPI1N pin should not be connected.

Programming SPI Serial Flash

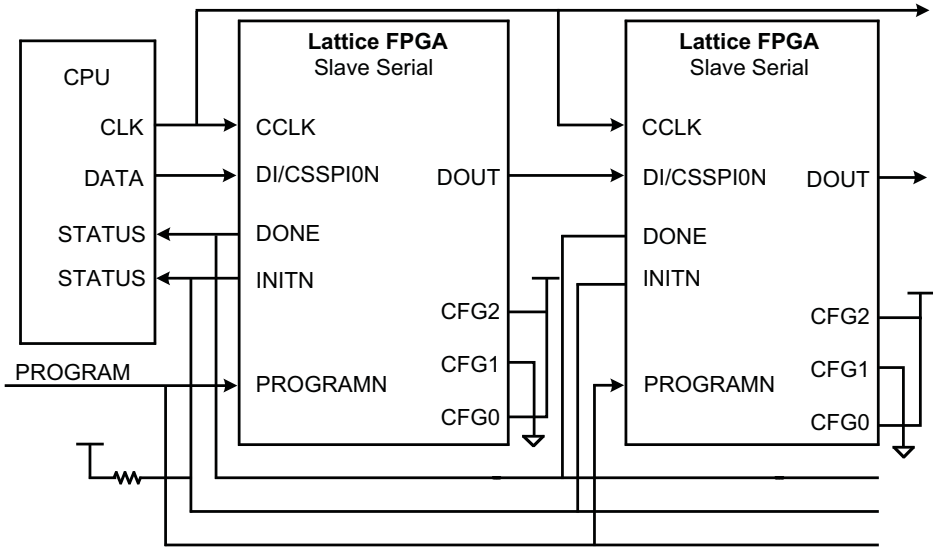
The LatticeECP2/M contains dedicated hardware that allows JTAG to access the SPI port, allowing ispVM, embedded hardware, or ATE equipment to program the Flash while it is on the board. In order to program SPI Serial Flash using JTAG, the CFG pins must be set to SPI or SPIm (see Table 15-3). Please refer to the ispVM Help system for more information.

Slave Serial Mode

In Slave Serial mode the CCLK pin becomes an input, receiving the clock from an external device. The LatticeECP2/M accepts data on the DI pin on the rising edge of CCLK. Slave Serial only supports writes to the FPGA, it does not support reading from the FPGA.

After the device is fully configured, if the Bypass option has been set, any additional data clocked into DI will be presented to the next device via the DOUT pin, as shown in Figure 15-7.

Figure 15-7. Serial Mode Daisy Chain



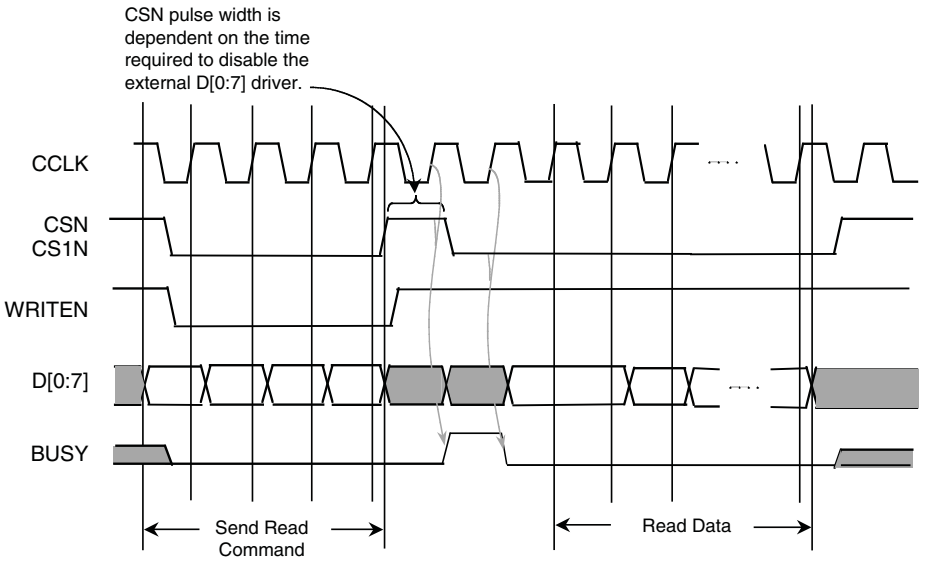
Note: In the Slave Serial Mode, the Bypass option is not supported when using encrypted bitstream files with the LatticeECP2/M S-Series devices. Please refer to the *LatticeECP2/M S-Series Configuration Encryption Usage Guide*, TN1109, for more information about using encrypted bitstream files.

Slave Parallel Mode

In Slave Parallel mode a host system sends the configuration data in a byte-wide stream to the LatticeECP2/M. The CCLK, CSN, CS1N, and WRITEN pins are driven by the host system. WRITEN, CSN, and CS1N must be held low to write to the device; data is input from D[0:7]. D0 is the MSb and D7 is the LSb.

Slave Parallel mode can also be used for readback of the internal configuration. By driving the WRITEN pin low, and CSN and CS1N low, the device will input the readback instructions on the D[0:7] pins; WRITEN is then driven high and read data is output on D[0:7] (see Figure 15-8). In order to support readback, the PERSISTENT bit must be set to ON in ispLEVER Design Planner or in the Diamond Spreadsheet View (Global Preferences tab). *Note: SLAVE PARALLEL Mode is not available in the 100-pin TQFP and 208-pin PQFP package offerings since the CSN, CS1N and WRITEN pins are not bonded out for these packages.*

Figure 15-8. Parallel Port Read Timing Diagram



The host sends the preamble, BDB3 (hex), and then sends the read command. The LatticeECP2/M sends read data on D[0:7], driving BUSY high as needed to pause data flow. For an example bitstream see Table 15-7. Table 15-8 lists the various read commands. Note that the sample bitstream and the list of read commands are for reference only; to help the user better understand the flow. The actual bitstream, containing the read commands, is generated by ispLEVER or Diamond and ispVM, the host, toggles the control signals and sends the bitstream.

Table 15-7. Parallel Port Read Bitstream Example

Frame	Contents	Description
Header	1111...1111	2 Dummy Bytes
	10111101 10110011	2-byte Preamble (BDB3)
Verify ID		8 bytes of command and data
Reset Address		4 bytes of command and data
Read Increment		4 bytes of command and data

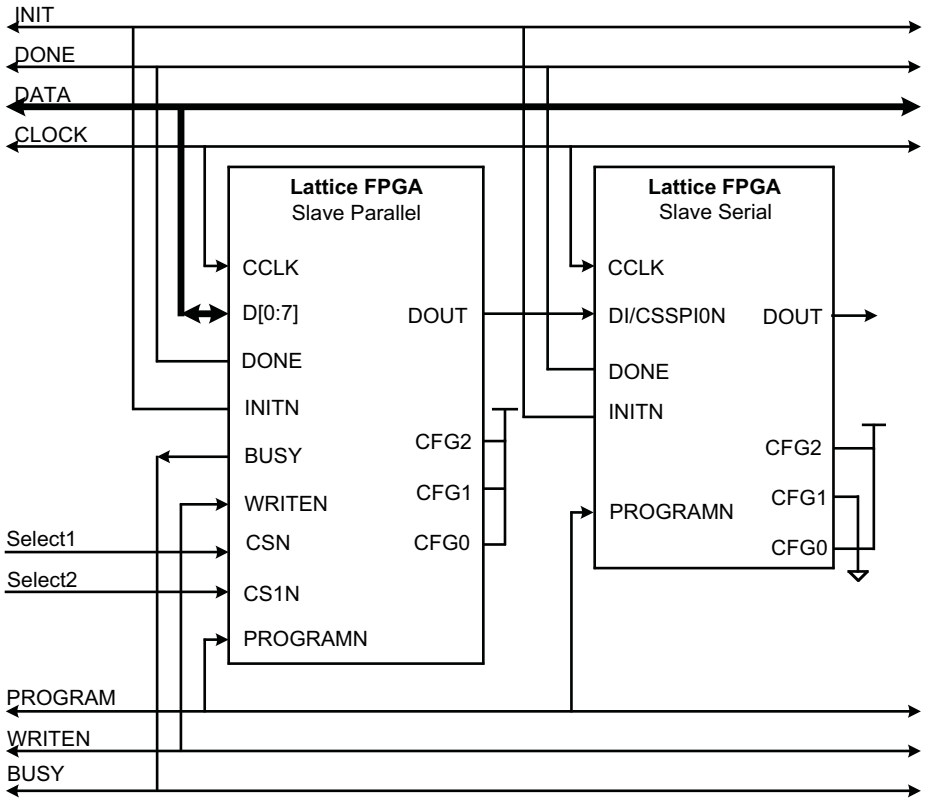
Table 15-8. Parallel Port Read Commands

Command	32-bit Opcode	Function
Reset Address	E2000000	Reset address register to point to the first data frame
Read Increment	01vvvvvv	Read back the configuration memory frame selected by the address register and post increment the address
Read Usercode	83000000	Read the content of the USERCODE register
Read Ctrl Reg 0	84000000	Read the content of Control Register 0
Read CRC	86000000	Read CRC register content
Read ID Code	87000000	Read ID code
NO OP	FF	No operation. This is an 8-bit Opcode. Extra bits should not be appended to this Opcode as this could cause INITN to go low during configuration.

Note: x = don't care, v = variable.

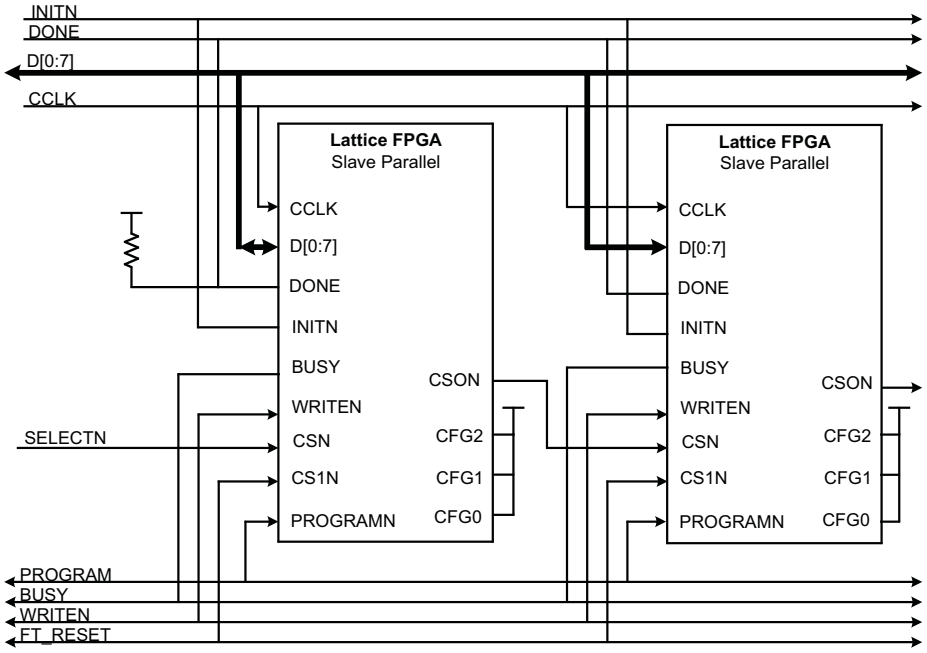
Slave Parallel mode can support two types of overflow, Bypass and Flowthrough. After the first device has received all of it's configuration data, and the Bypass command is detected in the bitstream, the data presented to the D[0:7] pins will be serialized and bypassed to the DOUT pin (see Figure 15-9). If the Flowthrough command is detected in the bitstream, instead of the bypass command, the CSON signal will drive the following parallel mode device's chip select low as shown in Figure 15-10. If either type of overflow is active, driving both the CSN and CS1N pins high will reset overflow, i.e. take the device out of overflow.

Figure 15-9. Slave Parallel with Bypass Option



Note: In Slave Parallel mode, the Bypass option is not supported when using encrypted bitstream files with the LatticeECP2/M S-Series devices. Please refer to the *LatticeECP2/M S-Series Configuration Encryption Usage Guide*, TN1109, for more information about using encrypted bitstream files.

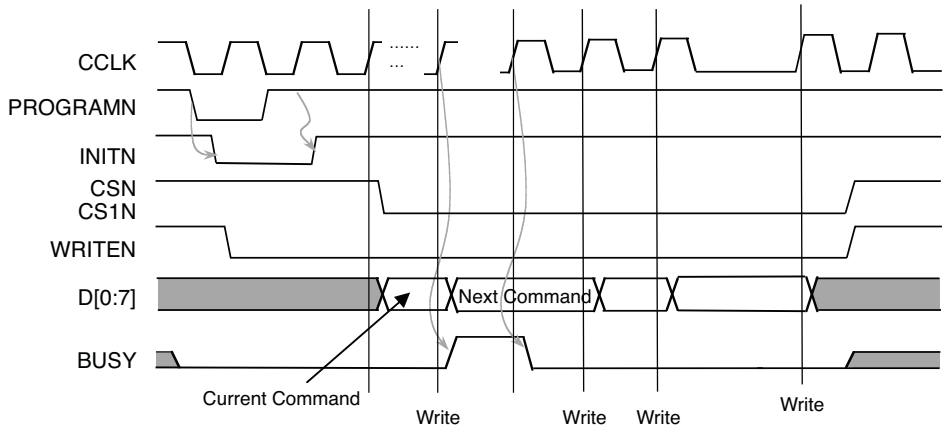
Figure 15-10. Slave Parallel with Flowthrough



Note: In Slave Parallel mode, the Flowthrough option is not supported when using encrypted bitstream files with the LatticeECP2/M S-Series devices. Please refer to the *LatticeECP2/M S-Series Configuration Encryption Usage Guide*, TN1109, for more information about using encrypted bitstream files.

To support asynchronous configuration, where the host may provide data faster than the FPGA can accept it, Slave Parallel mode can use the BUSY signal. By driving the BUSY signal high the Slave Parallel device tells the host to pause sending data. Please note that all data and control are still synchronous with CCLK, asynchronous refers to the ability to throttle the data transfer using BUSY. See Figure 15-11.

Figure 15-11. Parallel Port Write Timing Diagram



Note: When downloading an encrypted bitstream file to the LatticeECP2/M S-Series devices, the user must adhere to the appropriate conditions for the CCLK signal. These conditions are shown in the *LatticeECP2/M S-Series Configuration Encryption Usage Guide*, TN1109.

The CSN and CS1N pins must remain low while the configuration bitstream is being sent to the device or the configuration will fail. To temporarily stop the write process, the user can pause the CCLK signal and the data. An example of this is shown in Figure 15-11.

ispJTAG Mode

The LatticeECP2/M device can be configured through the ispJTAG port using either Fast Program or IEEE 1532 mode. The JTAG port is always on and available, regardless of the configuration mode selected. The IEEE 1532 mode is called Erase, Program, Verify in ispVM System.

Fast Program

Fast Program can be thought of as serial configuration using the JTAG port. The data file used for Fast Program is the same as a file used for a sysCONFIG Serial mode configuration, in other words there is a header, a preamble, and configuration data. Fast Program will result in a faster configuration time since the bitstream contains CRC checking so a time consuming post programming bit by bit verification is not required. The SVF and VME file formats also support Fast programming.

During JTAG configuration the Boundary Scan cells take control of the LatticeECP2/M I/Os. The Boundary Scan cells will usually drive the I/Os to a tri-state level but this can be controlled and even customized using ispVM. Note that PROGRAMN, INITN, and DONE have no meaning since they are controlled by the Boundary Scan cells during JTAG configuration. However, the INITN and DONE do indicate configuration status after JTAG configuration is completed.

IEEE 1532

Besides Fast Program the LatticeECP2/M can also be configured through JTAG using the IEEE 1532 Standard. The IEEE 1532 mode is called Erase, Program, Verify in ispVM System. IEEE 1532 configuration files contain JTAG instructions, as well as the configuration data. IEEE 1532 files, including ISC, SVF, and VME, can be created using ispVM's Universal File Writer (UFW). These files can be used by ispVM or by third party ATE equipment. These files can also be used in embedded situations, where an on-board processor provides the data while controlling the JTAG signals (this is called ispVM Embedded, more information can be found in ispVM's help facility). IEEE 1532 programming will be slower than the Fast Program mode since it requires a post programming bit by bit verification.

During JTAG configuration the Boundary Scan cells take control of the LatticeECP2/M I/Os. The Boundary Scan cells will usually drive the I/Os to a tri-state level but this can be controlled and even customized using ispVM.

Note that PROGRAMN, INITN, and DONE have no meaning since they are controlled by the Boundary Scan cells during JTAG configuration. However, the DONE pin will indicate configuration status after IEEE 1532 configuration is completed.

Transparent Read Back

The ispJTAG transparent read back mode allows the user to read the content of the device while the device remains fully functional. All I/O, as well at the non-JTAG configuration pins, remain under internal logic control during a Transparent Read Back. The device enters the Transparent Read Back mode through a JTAG instruction. The user must ensure that Transparent Read Back does not access EBR or distributed RAM at the same time internal logic is accessing these resources or corruption of the RAM may occur.

Boundary Scan and BSDL Files

The LatticeECP2/M BSDL files can be found on the Lattice Semiconductor web site. The boundary scan ring covers all of the I/O pins, as well as the dedicated and dual-purpose sysCONFIG pins. Note that PROGRAMN, CCLK, and the CFG pins are observe only (BC4, JTAG read-only) boundary scan cells.

Configuration Options

Several configuration options are available for each configuration mode.

- When daisy chaining multiple FPGA devices an overflow option is provided for serial and parallel configuration modes
- When using SPI or SPIm mode, the master clock frequency can be set
- A security bit can be set to prevent SRAM readback

- The bitstream can be compressed
- The Persistent option can be set
- Configuration pins can be protected
- DONE pin options can be selected

By setting the proper parameter in the Lattice design software the selected configuration options are set in the generated bitstream. As the bitstream is loaded into the device the selected configuration options take effect. These options are described in the following sections.

Bypass Option

In ispLEVER, the Bypass option can be set by using the Bitgen properties. Or, a chain of bitstreams can be assembled and Bypass set using ispVM. To set the Bypass option in Diamond, see Appendix A. The Bypass option can be used in parallel and serial mode daisy chains. The Bypass option is not supported when using SPIm configuration. The Bypass option is not supported when using encrypted bitstream files with the LatticeECP2/M S-Series devices. Please refer to TN1109, LatticeECP2/M Configuration Encryption Usage Guide, for more information about using encrypted bitstream files.

When the first device completes configuration, and a Bypass command is input from the bitstream, any additional data coming into the FPGA configuration port will overflow serially on DOUT. This data is applied to the DI pin of the next device (downstream devices must be set to slave serial mode).

In serial configuration mode the Bypass option connects DI to DOUT via a bypass register. The bypass register is initialized with a '1' at the beginning of configuration and will stay at that value until the Bypass command is executed. In parallel configuration mode the Bypass option causes the excess data coming in on D[0:7] to be serially shifted to DOUT. The serialized data is shifted to DOUT through a bypass register. D0 will be shifted out first followed by D1, D2, and so on. Once the Bypass option starts the device will remain in Bypass until the Wake-up sequence completes. In parallel mode, if Bypass needs to be aborted, drive both CSN and CS1N high, this acts as a Bypass reset signal.

Flowthough Option

As with Bypass, Flowthrough can be set in the Bitgen properties in ispLEVER and in the Strategy Process Options settings. To set the Flowthrough option in Diamond, see Appendix A. The Flowthrough option can be used with parallel daisy chains only. The Flowthrough option is not supported when using SPIm configuration. The Flowthrough option not supported when using encrypted bitstream files with the LatticeECP2/M S-Series devices. Please refer to TN1109, LatticeECP2/M Configuration Encryption Usage Guide, for more information about using encrypted bitstream files.

When the first device completes configuration, and a Flowthrough command is input from the bitstream, the CSON pin is driven low. In addition to driving CSON low, Flowthrough also tri-states the device's D[0:7] and BUSY pins in order to avoid contention with the other daisy-chained devices. Once the Flowthrough option starts the device will remain in Flowthrough until the Wake-up sequence completes. If Flowthrough needs to be aborted drive both CSN and CS1N high, this acts as a Flowthrough reset signal.

Master Clock

If the CFG pins indicate an SPI or SPIm mode the CCLK pin will become an output, with the frequency set by the user. The default Master Clock Frequency is 3.1 MHz. For a complete list of the supported Master Clock frequencies, please see the LatticeECP2/M Family Data Sheet. When using the LatticeECP2/M S-Series devices, the available frequencies are restricted, as shown in the data sheet.

The user can change the Master Clock frequency by setting the MCCLK_FREQ global preference in the Lattice ispLEVER Design Planner. To set this option in Diamond, see Appendix A. One of the first things loaded during configuration is the MCCLK_FREQ parameter; once this parameter is loaded the frequency changes to the selected value using a glitchless switch. Care should be exercised not to exceed the frequency specification of the slave devices or the signal integrity capabilities of the PCB layout.

Configuration time is computed by dividing the maximum number of configuration bits, as given in Table 15-4 above, by the Master Clock frequency.

Security Bit
Setting the CONFIG_SECURE option to ON prevents readback of the SRAM from JTAG or the sysCONFIG pins. When CONFIG_SECURE is set to ON the only operations available are erase and write. The security fuse is updated as the last operation of SRAM configuration. If a secured device is read it will output all zeros.

For LatticeECP2/M devices the CONFIG_SECURE option is accessed via the Design Planner in ispLEVER. To set this option in Diamond, see Appendix A. The default is OFF.

For the LatticeECP2/M S-Series devices (part numbers ECP2-XXES and ECP2MXXES) the CONFIG_SECURE option is accessed using the **Security Setting** option under the **Tools** menu.

Compress Bitstream
Setting the global COMPRESS_CONFIG option to ON in ispLEVER Design Planner will cause the software to generate a compressed bitstream. To set this option in Diamond, see Appendix A. The LatticeECP2/M will automatically decompress the bitstream as it comes into the device. The actual amount of compression varies according to the data pattern in the uncompressed bitstream. Though unlikely, it is theoretically possible for the compressed bitstream to be larger than the uncompressed bitstream.

Compressing the bitstream can result in faster configuration. The default setting is OFF.

Persistent Option
The PERSISTENT Option is set using ispLEVER Design Planner (default is OFF). To set this option in Diamond, see Appendix A. PERSISTENT serves two purposes.

Setting PERSISTENT ON tells the place and route tools that it may not use any of the sysCONFIG pins associated with the parallel port (all of the dual-purpose pins except DI) or the SPI Port pins.

Setting PERSISTENT ON also sets a hardware fuse. So, not only are the pins reserved in software, they are also reserved in hardware.

PERSISTENT is set to ON when the user wants to be able to read the SRAM configuration memory using the Slave Parallel port. In order to perform a read using the parallel port the user must first send a read command, setting PERSISTENT ON allows the parallel port to listen for this command while in user mode (the DONE pin is high). If the design does not require this function, the PERSISTENT option should be set to OFF.

The PERSISTENT preference must also be set to ON if the SPI memory needs to be accessed using the SPI port while the part is in user mode (the DONE pin is high) to preserve the SPI port pins.

Configuration Mode
Just as the CFG pins tell the hardware which port to configure from; the CONFIG_MODE option tells the software which port will be used. CONFIG_MODE allows the user to protect dual-purpose sysCONFIG pins. For example setting CONFIG_MODE to SPI will keep the Place and Route tools from using the SPI pins as general purpose I/O. The user, however, is still free to assign these pins as GPIO, but a warning will be generated as a reminder that there are certain precautions (see the section above entitled Configuration Pins).

Available options are NONE, JTAG, SPI, SPIm, Slave Serial, and Slave Parallel. The default is Slave Serial.

DONE_OD, DONE_EX
During configuration the DONE pin is low. Once configuration is complete, indicated by the setting of the internal Done bit, the device wake-up sequence takes place and then the DONE pin goes high. Under most circumstances, this flow is exactly what is needed, however, if there are several devices in one configuration chain, delay of the wake-up sequence may be desirable in order to “synchronize” the wake-up of all devices in the chain. There are two options that allow for this synchronization. These options are set in ispLEVER Design Planner. To set these options in Diamond, see Appendix A.

DONE_OD defaults to ON. DONE_OD ON forces the DONE pin type to be open- drain. When connecting multiple DONE pins together all of the pins should be open- drain, however it may be advantageous to have an actively driven DONE pin on the last device. Setting DONE_OD to OFF makes the DONE pin an actively driven pin, rather than open-drain.

DONE_EX defaults to OFF. Setting DONE_EX to ON will cause LatticeECP2/M to sample the DONE pin and, if the DONE pin is held low externally, delay the wake-up sequence. Setting DONE_EX to OFF will cause the device to wake-up as soon as the internal Done bit is set.

Device Wake-Up

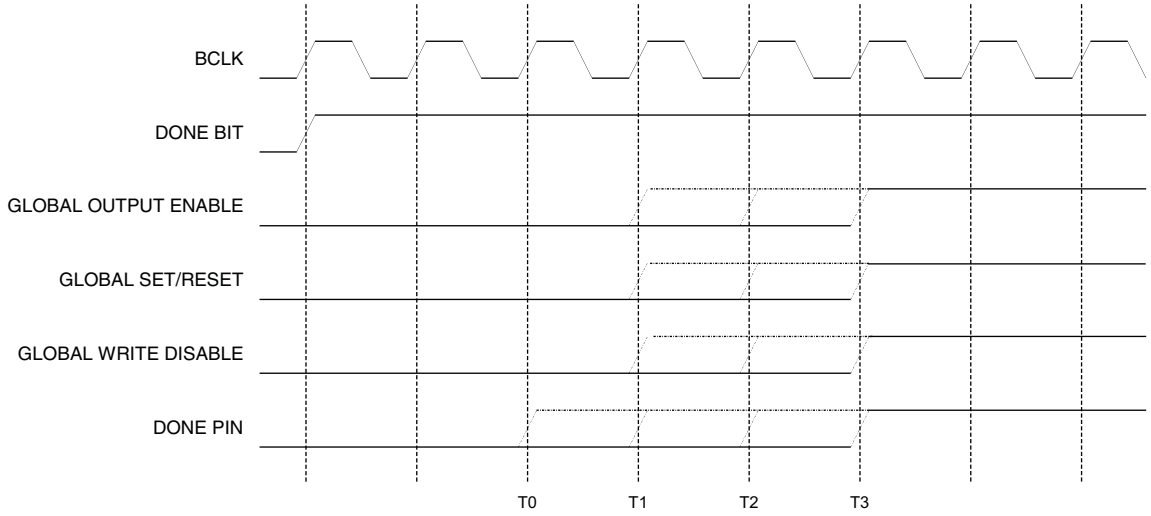
When configuration is complete the device will wake up in a predictable fashion. Wake-Up occurs after successful configuration, without errors, and provides the transition from Configuration Mode to User Mode. The Wake-Up process begins when the internal Done bit is set.

Table 15-9 provides a list of the Wake-Up sequences supported by the LatticeECP2/M; Figure 15-12 shows the Wake-Up timing. The WAKE_UP defaults work fine for the vast majority of applications. To set the WAKE_UP options in Diamond, see Appendix A.

Table 15-9. Wake-Up Options

Sequence	Phase T0	Phase T1	Phase T2	Phase T3
1	DONE	GOE, GWDIS, GSR		
2	DONE		GOE, GWDIS, GSR	
3	DONE			GOE, GWDIS, GSR
4	DONE	GOE	GWDIS, GSR	
5	DONE	GOE		GWDIS, GSR
6	DONE	GOE	GWDIS	GSR
7	DONE	GOE	GSR	GWDIS
8		DONE	GOE, GWDIS, GSR	
9		DONE		GOE, GWDIS, GSR
10		DONE	GWDIS, GSR	GOE
11		DONE	GOE	GWDIS, GSR
12			DONE	GOE, GWDIS, GSR
13		GOE, GWDIS, GSR	DONE	
14		GOE	DONE	GWDIS, GSR
15		GOE, GWDIS	DONE	GSR
16		GWDIS	DONE	GOE, GSR
17		GWDIS, GSR	DONE	GOE
18		GOE, GSR	DONE	GWDIS
19			GOE, GWDIS, GSR	DONE
20		GOE, GWDIS, GSR		DONE
21 (Default)		GOE	GWDIS, GSR	DONE
22		GOE, GWDIS	GSR	DONE
23		GWDIS	GOE, GSR	DONE
24		GWDIS, GSR	GOE	DONE
25		GOE, GSR	GWDIS	DONE

Figure 15-12. Wake-Up Timing Diagram



Synchronizing Wake-Up

The internal Wake-Up sequence clock source can be chosen as well as how the device wakes up relative to other devices.

Wake-Up Clock Selection

The Wake-Up sequence is synchronized to a clock source that is user selectable. The clock sources are External (default) and User Clock.

When External is selected the LatticeECP2/M will use one of two clocks during the Wake-Up sequence, depending on the configuration data source. If the LatticeECP2/M is being configured from JTAG then JTAG's TCK will be used for the Wake-Up sequence, if configuration data is coming from a sysCONFIG port (serial, parallel, or SPI) then CCLK will be used.

When User is selected, any of the clock signals in the design can be used as the clock source.

Synchronous to Internal Done Bit

If the LatticeECP2/M is the only device in the configuration chain, or the last device in the chain, DONE_EX should be set to the default value (OFF). The Wake-Up process will be initiated by setting of the internal Done bit on successful completion of configuration.

Synchronous to External DONE Pin

The DONE pin can be used to synchronize Wake-Up to other devices in the configuration chain. If DONE_EX (see the DONE_OD, DONE_EX section above) is ON then the DONE pin is a bi-directional pin. If an external device drives the DONE pin low then the Wake-Up sequence will be delayed; configuration can complete but Wake-Up is delayed. Once the DONE pin goes high the device will follow the selected WAKE_UP sequence.

In a configuration chain, a chain of devices configuring from one source (such as Figure 15-2), it is usually desirable, or even necessary, to delay wake-up of all of the devices until the last device finishes configuration. This is accomplished by setting DONE_OD to OFF and DONE_EX to OFF on the last device while setting DONE_OD to ON and DONE_EX to ON for the other devices.

Wake-Up Sequence Options

The Wake-Up sequence options shown in Table determine the order of application for three internal signals, GSR, GWDIS, and GOE, and one external signal, DONE.

- GSR is used to set and reset the core of the device. GSR is asserted (low) during configuration and de-asserted (high) in the Wake-Up sequence.

- When the GWDIS signal is low it safeguards the integrity of the RAM Blocks and LUTs in the device. This signal is low before the device wakes up.
- When low, GOE prevents the device's I/O buffers from driving the pins.
- When high, the DONE pin indicates that configuration is complete and that no errors were detected.

If DONE_EX (see DONE_OD, DONE_EX above) is OFF then sequence 21 is the default, but the user can select any sequence from 8 to 25; if DONE_EX is ON the default sequence is 4, but the user can select any sequence from 1 to 7.

Configuration FAQs

Here are some of the more common questions regarding device configuration.

General

- **Q. Other than JTAG, what is the least expensive method of configuration?**

  A. If you already have a processor and extra storage on the board you can use the processor to feed configuration data to the LatticeECP2/M. The least expensive stand-alone configuration option is SPI Serial Flash.

- **Q. I have created my bitstream, now how do I load the bitstream into the LatticeECP2/M?**

  A. Use the free Lattice ispVM tool (from www.latticesemi.com), and a Lattice ispDOWNLOAD® cable.

- **Q. I can't read the LatticeECP2/M device ID using JTAG. What could be wrong?**

  A. This is the most basic of JTAG operations. If you are having trouble reading the device ID then something basic is wrong. Check that the JTAG connections are correct, and that V_{CCJ} and the download cable V_{CC} are correct (and the same). Make sure that the XRES pin is connected to ground through a 10K resistor. Check that all LatticeECP2/M V_{CC} and ground pins are properly connected. Check for noise on the JTAG signals. Sometimes touching a properly grounded 'scope probe to TCK will change the symptoms; if so, you have signal noise issues. Check for excessive noise on the V_{CC} pins.

- **Q. Is there anything I can do during board design that will make debugging easier?**

  A. Bring the dedicated configurations pins, PROGRAMN, INITN, DONE, and CCLK out to accessible test points.

- **Q. Do I need external pull-up resistors on PROGRAMN, INITN, or DONE?**

  A. All of these signals have internal weak pull-ups, however if you have a noisy environment, or have several devices connected to these pins, adding a 10K pull-up to the signal is recommended.

- **Q. How can I get assistance with configuration issues?**

  A. Use the On-line Assistant feature in ispVM. You will find it under the Help menu.

Mode Specific

SPI/SPIm

- **Q. How do I program the SPI Serial Flash once it's on the board?**

  A. Connect the SPI Serial Flash to the LatticeECP2/M as shown in this document, then use ispVM, and a Lattice ispDOWNLOAD cable connected to the JTAG port, to program the bitstream into the Flash.

  LatticeECP2/M devices have a JTAG instruction for programming the SPI Serial Flash. This JTAG instruction connects the JTAG TCK internally to the CCLK which drives the SPI clock during programming. After launching ispVM, click on the **Scan** button to scan the devices in the JTAG chain, select the LatticeECP2/M device, then click **Edit>Edit Device** to launch the Device Information window. In the Device Information window, select **SPI Flash Programming** for the Device Access Options to open the SPI Serial Flash Device window. Select the SPI Serial Flash device and the bitstream data file for programming the SPI Serial Flash.

- **Q. Are there any special requirements for wiring the SPI Flash to the LatticeECP2/M?**

  A. Other than connecting the Flash to the right pins the only other suggestion is to add a 4.7K pull-down resistor between CCLK and ground. This keeps CCLK quite during V_{CC} ramp-up.

- **Q. Can I use 2.5V to power the SPI Flash?**

  A. Today all SPI Serial Flash of the "25" type are 3.3V, so the Flash, and V_{CCIO8}, must be connected to 3.3V.

- **Q. Can I use something other than a "25" type SPI Serial Flash?**

  A. Only devices that recognize a read op-code of 03h may be used with the LatticeECP2/M. Please refer to Table 15-6 for a list of vendors.

- **Q. My design is small, can I use a smaller-than-recommended SPI Flash?**

  A. The state of all of the device fuses is contained in the bitstream, whether they are part of the design or not. The size of the design does not affect the size of the bitstream.

Serial

- **Q. Can I use a free running clock for Slave Serial mode?**

  A. The LatticeECP2/M clocks data in on every rising edge of CCLK so there should only be one rising clock edge for each data bit.

- **Q. Is the bitstream for the serial modes different from the bitstream for other modes?**

  A. All sysCONFIG bitstreams are the same, they can be different file types, such as hex or binary, but the data is the same.

Parallel

- **Q. My processor is generating all of the proper control signals but the LatticeECP2/M won't configure, and INITN goes high and stays high while DONE stays low. What's wrong?**

  A. D0 is the MSB and D7 is the LSB. Try reversing the bit order for each byte in the bitstream. You can do this using your processor or you can generate a bit mirrored file using ispVM. Lattice recommends using your processor so that you don't have to remember to bit mirror the file.

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
February 2006	01.0	Initial release.
February 2006	01.1	Added ECP2-6 to Maximum Configuration Bits table and changed ECP2-22 to ECP2-20.
April 2006	01.2	Updated section on SPIm. Also added screen shots to SPIm section.
September 2006	01.3	Updated Maximum Configuration Bits table with ECP2M35 data.
February 2007	01.4	Updated Maximum Configuration Bits table with ECP2M data.
March 2007	01.5	Added information about LatticeECP2/M “S” series to the Dual Boot Image Setup section.
April 2007	01.6	Added more information about SPI programming using ispVM to the SPI/SPIm Q&A section.
June 2007	01.7	Added hysteresis value. Removed support for booting from multiple SPI Flash devices in the SPIm mode. Added note for SPIFASTN pin.
August 2007	01.8	Added sizes for encrypted bitstream files and new conditions for their use. Removed restriction on I/O direction for dual-purpose configuration pins.
September 2007	01.9	Updated Configuration Pins text section.
		Updated PROGRAMN text section.
January 2008	02.0	Changed PROGRAMN and INITN pin descriptions. Changed CSN and CS1N pin description. Added CSN and CS1N pin restrictions in Slave Parallel Mode section.
September 2008	02.1	Updated CCLK text section.
June 2010	02.2	Updated document for Lattice Diamond design software support.
March 2011	02.3	Added information for 100-pin TQFP and 208-pin PQFP device configuration limitations.
		Added notes to the Dual-purpose sysCONFIG pins section for CSN and CS1N, and WRITEN.
		Added note to the Slave Parallel Mode section for CSN and CS1N and WRITEN.

Appendix A. Lattice Diamond Usage Overview

This appendix discusses the use of Lattice Diamond design software for projects that include the LatticeECP2M SERDES/PCS module .

For general information about the use of Lattice Diamond, refer to the Lattice Diamond Tutorial.

If you have been using ispLEVER software for your FPGA design projects, Lattice Diamond may look like a big change. But if you look closer, you will find many similarities because Lattice Diamond is based on the same toolset and work flow as ispLEVER. The changes are intended to provide a simpler, more integrated, and more enhanced user interface.

Converting an ispLEVER Project to Lattice Diamond

Design projects created in ispLEVER can easily be imported into Lattice Diamond. The process is automatic except for the ispLEVER process properties, which are similar to the Diamond strategy settings, and PCS modules. After importing a project, you need to set up a strategy for it and regenerate any PCS modules.

Importing an ispLEVER Design Project

Make a backup copy of the ispLEVER project or make a new copy that will become the Diamond project.

1. In Diamond, choose **File > Open > Import ispLEVER Project**.
2. In the ispLEVER Project dialog box, browse to the project's .syn file and open it.
3. If desired, change the base file name or location for the Diamond project. If you change the location, the new Diamond files will go into the new location, but the original source files will not move or be copied. The Diamond project will reference the source files in the original location.

The project files are converted to Diamond format with the default strategy settings.

Adjusting PCS Modules

PCS modules created with IPexpress have an unusual file structure and need additional adjustment when importing a project from ispLEVER. There are two ways to do this adjustment. The preferred method is to regenerate the module in Diamond. However this may upgrade the module to a more recent version. An upgrade is usually desirable but if, for some reason, you do not want to upgrade the PCS module, you can manually adjust the module by copying its .txt file into the implementation folder. If you use this method, you must remember to copy the .txt file into any future implementation folders.

Regenerate PCS Modules

1. Find the PCS module in the Input Files folder of File List view. The module may be represented by an .lpc, .v, or .vhd file.
2. If the File List view shows the Verilog or VHDL file for the module, and you want to regenerate the module, import the module's .lpc file:
   a. In the File List view, right-click the implementation folder () and choose **Add > Existing File**.
   b. Browse for the module's .lpc file, ***<module_name>*.lpc**, and select it.
   c. Click **Add**. The .lpc file is added to the File List view.
   d. Right-click the module's Verilog or VHDL file and choose **Remove**.
3. In File List, double-click the module's .lpc file. The module's IPexpress dialog box opens.
4. In the bottom of the dialog box, click **Generate**. The Generate Log tab is displayed. Check for errors and close.

In File List, the .lpc file is replaced with an .ipx file. The IPexpress manifest (.ipx) file is new with Diamond. The .ipx file keeps track of the files needed for complex modules.

Using IPexpress with Lattice Diamond

Using IPexpress with Lattice Diamond is essentially same as with ispLEVER.

The configuration GUI tabs are all the same except for the Generation Options tab. Figure 15-13 shows the Generation Options tab window.

Figure 15-13. Generation Options Tab

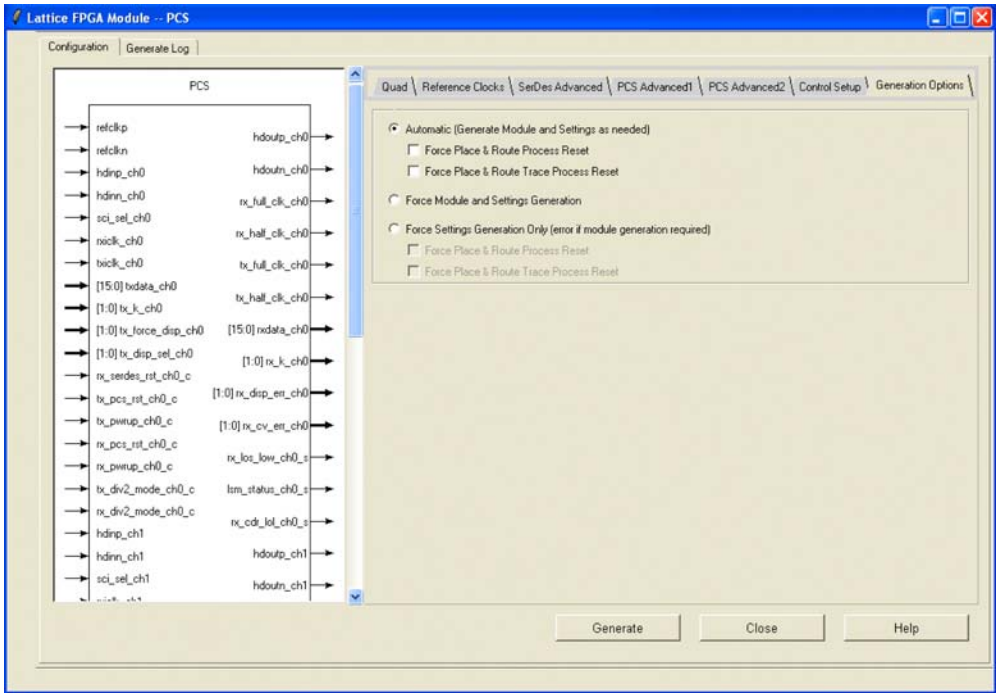


Table 15-10. SERDES_PCS GUI Attributes – Generation Options Tab

GUI Text	Description
Automatic	Automatically generates the HDL and configuration(.txt) files as needed. Some changes do not require regenerating both files.
Force Module and Settings Generation	Generates both the HDL and configuration files.
Force Settings Generation Only	Generates only the attributes file. You get an error message if the HDL file also needs to be generated.
Force Place & Route Process Reset	Resets the Place & Route Design process, forcing it to be run again with the newly generated PCS module.
Force Place & Route Trace Process Reset	Resets the Place & Route Trace process, forcing it to be run again with the newly generated PCS module.

Note:
Automatic is set as the default option. If either Automatic or Force Settings Generation Only and no sub-options (Process Reset Options) are checked and the HDL module is not generated, the reset pointer is set to Bitstream generation automatically.

After the Generation is finished, the reset marks in the process window will be reset accordingly.

Creating a New Simulation Project Using Simulation Wizard

This section describes how to use the Simulation Wizard to create a simulation project (.spf) file so you can import it into a standalone simulator.

1. In Project Navigator, click **Tools > Simulation Wizard**. The Simulation Wizard opens.
2. In the Preparing the Simulator Interface page click **Next**.
3. In the Simulator Project Name page, enter the name of your project in the Project Name text box and browse to the file path location where you want to put your simulation project using the Project Location text box and Browse button.

   When you designate a project name in this wizard page, a corresponding folder will be created in the file path you choose. Click **Yes** in the popup dialog that asks you if you wish to create a new folder.
4. Click either the Active-HDL® or ModelSim® simulator check box and click **Next**.
5. In the Process Stage page choose which type of Process Stage of simulation project you wish to create Valid types are RTL, Post-Synthesis Gate-Level, Post-Map Gate-Level, and Post-Route Gate-level+Timing. Only those process stages that are available are activated.

   Note that you can make a new selection for the current strategy if you have more than one defined in your project.

   The software supports multiple strategies per project implementation which allow you to experiment with alternative optimization options across a common set of source files. Since each strategy may have been processed to different stages, this dialog allows you to specify which stage you wish to load.
6. In the Add Source page, select from the source files listed in the Source Files list box or use the browse button on the right to choose another desired source file. Note that if you wish to keep the source files in the local simulation project directory you just created, check the **Copy Source to Simulation Directory** option.
7. Click **Next** and a Summary page appears and provides information on the project selections including the simulation libraries. By default, the Run Simulator check box is enabled and will launch the simulation tool you chose earlier in the wizard in the Simulator Project Name page.
8. Click **Finish**.

The Simulation Wizard Project (.spf) file and a simulation script DO file are generated after running the wizard. You can import the DO file into your current project if desired. If you are using Active-HDL, the wizard will generate an .ado file and if you are using ModelSim, it creates and .mdo file.

Note: PCS configuration file, (.txt) must be added in step 6.

Setting Global Preferences in Diamond

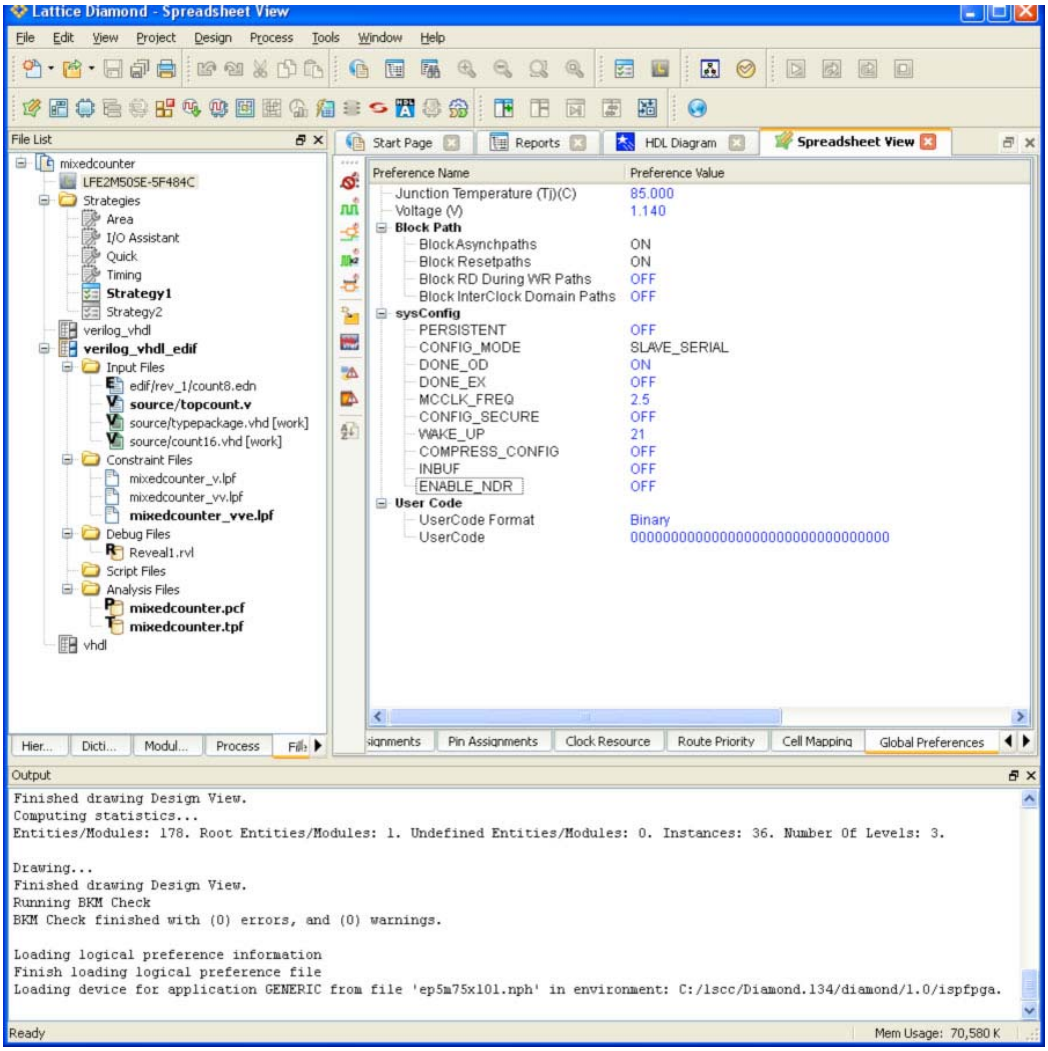
To set any of the Global preferences in Table 15-11, do the following in Diamond:

- Invoke the Spreadsheet View by selecting **Tools > Spreadsheet View**.
- Select the **Global Preferences Tab** beneath the Spreadsheet View pane as shown in Figure 15-14.
- Right-click on the **Preference Value** to be set. In the drop-down menu, select the desired value.

Table 15-11. Global Preferences

Preference Name	Values
PERSISTENT	ON Off
CONFIG_MODE	SLAVE_SERIAL JTAG NONE SLAVE_PARALLEL SPI SPIm
DONE_OD	ON Off
DONE_EX	OFF
MCCLK_FREQ	2.5 5.4 10 34 41 45
CONFIG_SECURE	OFF ON
WAKE_UP	An integer between 1 and 25
COMPRESS_CONFIG	OFF ON
INBUF	OFF ON
ENABLE_NDR	OFF ON

Figure 15-14. Global Preferences Tab



Setting Bitstream Generation Options in Diamond

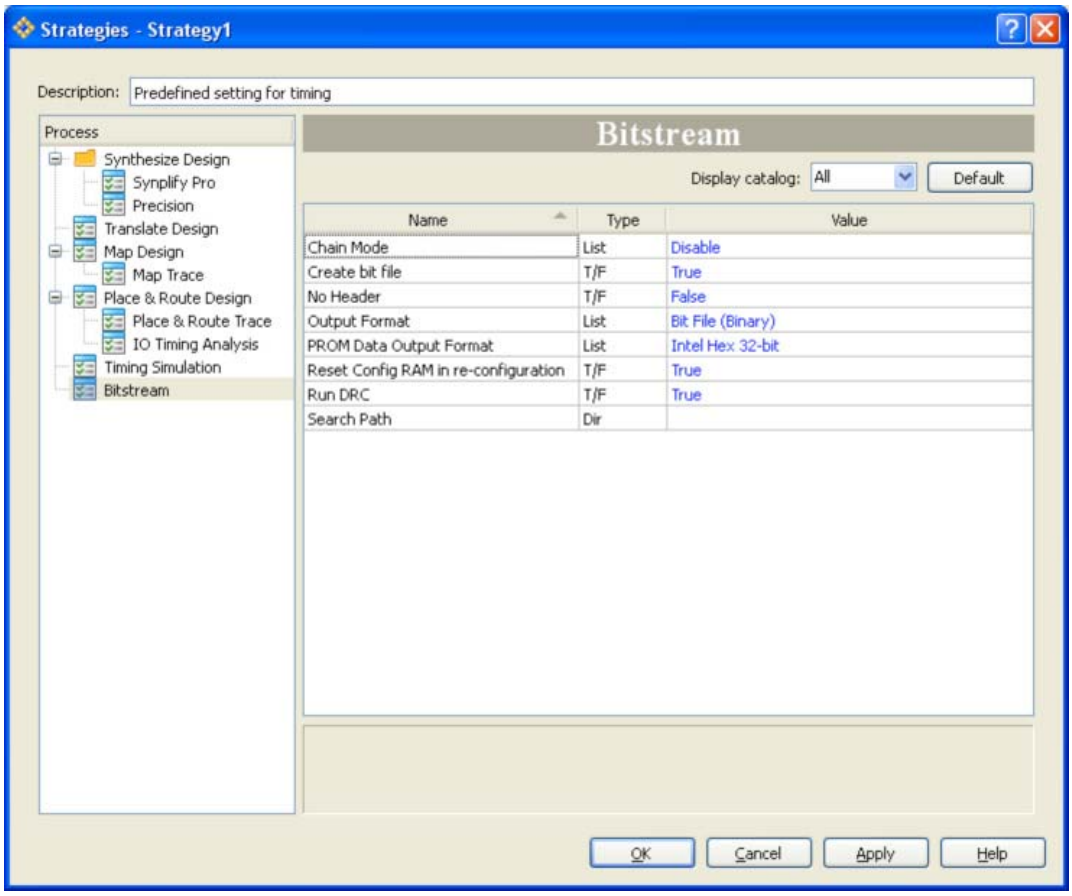
To set any of the Bitstream Generation options listed in Table 15-12, do the following:

- In the **File List** pane, double-click the left mouse button on a **Strategy** to invoke the Strategy settings window.
- In the **Process** pane, left-click on **Bitstream**. All options related to generating a bitstream can be set in this window.

Table 15-12. Bitstream Generation Options

Preference Name	Values
Chain Mode	Disable (default) Bypass Flowthrough
Create bit file	True
No Header	True False
Output Format	Bit File (Binary) Mask and Readback File (ASCII) Mask and Radback File (Binary) Raw Bit file (ASCII)
PROM Data Output Format	Intel Hex 32-bit Motorola Hex 32-bit
Reset Config RAM in re-configuation	True False
Run DRC	True False
Search Path	Enter a value or browse to specify the search path

Figure 15-15. Setting Bitstream Options in Diamond



- Double-click the left mouse button on the **Value** you want to set. Select the desired value from the drop-down menu.

*Note: An explanation of the option is displayed at the bottom of the window. The **Help** button also invokes online help for the option*

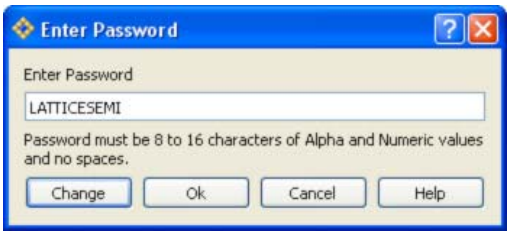
- Select **OK**. You can then run the Bitstream File process.

Setting Security Options in Diamond

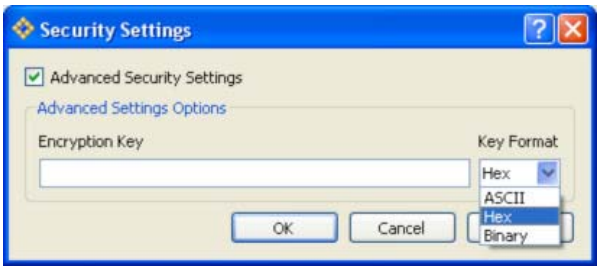
Prior to setting security options in Diamond, you must have installed the Encryption Control Pack. You must also have selected an encrypted device in your project.

To Set Security Settings, do the following:

- Select the **Tools > Security Setting** option. The following dialog box appears:



- If desired, select **Change** and enter a password.
- Select **OK**. A dialog window appears to enter an encryption key.
- If you do not want to enable an encryption key, select **OK**.
- If you do want to enable an encryption key, select the **Advanced Security Settings** checkbox, enter the **Key Format**, and then enter the **Encryption Key**.



- Select **OK** to create the encryption files.

LatticeECP2/M S-Series Configuration Encryption Usage Guide

June 2010 Technical Note TN1109

Introduction

All Lattice FPGAs provide configuration data read security, meaning that a fuse can be set so that when the device is read all zeros will be output instead of the actual configuration data. This kind of protection is common in the industry and provides very good security if the configuration data storage is on-chip, such as with the LatticeXP™ and MachXO™ device families. However, if the configuration bitstream comes from an external boot device it is quite easy to read the configuration data, allowing access to the FPGA design.

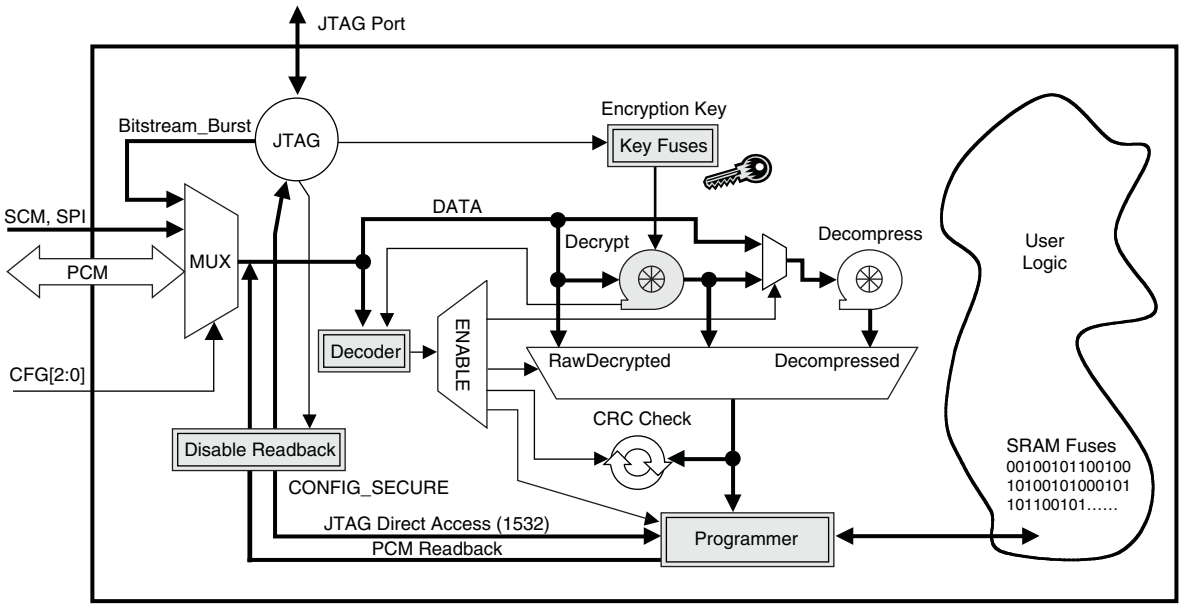
For this reason the "S" versions of LatticeECP2™ and LatticeECP2M™ offer the 128-bit Advanced Encryption Standard (AES) to protect the bitstream. The user selects and has total control over the 128-bit key and no special voltages are required to maintain the key within the FPGA.

This document explains the capabilities of this new security feature and how to take advantage of it.

General Configuration Process

Figure 16-1 is a block diagram describing the LatticeECP2/M "S" version bitstream encryption data paths. Refer to this figure as you read the following sections.

Figure 16-1. LatticeECP2/M "S" Version Bitstream Encryption Block Diagram



Lattice FPGAs are configured by using the sysCONFIG™ interface or the JTAG interface (see Table 16-1).

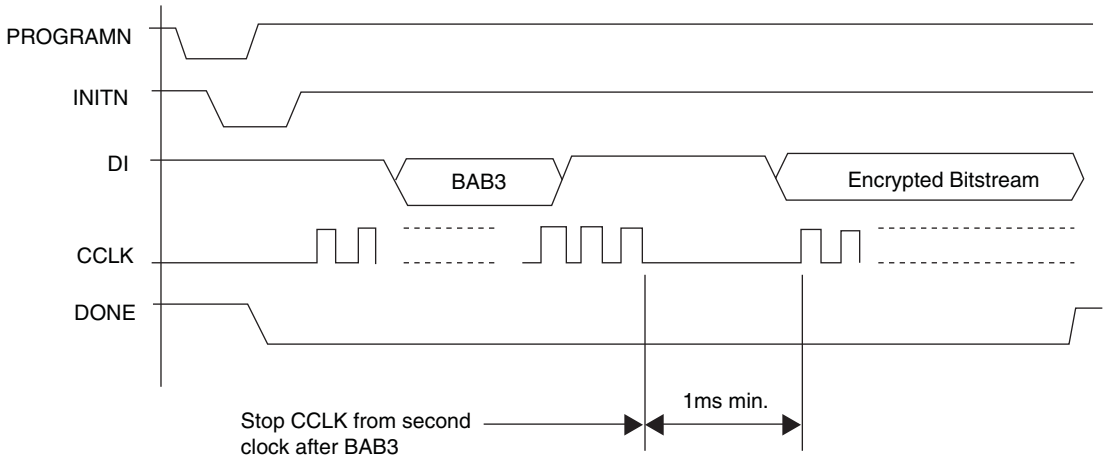
Table 16-1. Configuration Ports

Interface	Port
sysCONFIG	SPI[1]
	SPIm[1]
	Slave Serial (SCM)[2]
	Slave Parallel (PCM)[2]
ispJTAG™	IEEE 1532 (Erase, Program, Verify)[3]
	Bitstream Burst (Fast Program)

1. Supports subset of the CCLK frequencies specified in the LatticeECP2/M Family Data Sheet.
2. Users must adhere to the appropriate conditions for the CCLK signal as described below.
3. Does not support encrypted bitstreams.

The sysCONFIG interface allows the user to input data serially, using serial configuration mode (SCM) or SPI Serial Flash, or in parallel, using the parallel configuration mode (PCM). In general, the connection between the FPGA and the configuration device consists of a clock, chip select(s), a write signal (in PCM), and data. During configuration all data written to the FPGA is ignored until a special preamble is detected in the bitstream. Everything after the preamble is configuration data. The normal preamble is BDB3 (hex), however encrypted bitstreams contain a different preamble. When using SCM mode, any CCLK frequencies from 2.5MHz to 45MHz are supported but it is required to stop the CCLK after loading the BAB3 (hex) encryption preamble as shown below.

Figure 16-2. CCLK Timing



To ensure proper programming when using encrypted bitstreams in SPI mode only a subset of CCLK frequencies are supported. The supported frequencies are shown in the LatticeECP2/M Family Data Sheet.

The JTAG port, which conforms to IEEE 1149.1 and IEEE 1532 standards, can input data in Bitstream-Burst mode (Fast Program) or 1532 mode. Configuration bitstreams created for Bitstream-Burst mode (Fast Program) are identical to the configuration bitstreams created for sysCONFIG mode. The bitstream contains a header, a preamble, configuration data, and frame data CRC. However, 1532 mode makes use of standard JTAG instructions to configure the device. In other words, the configuration data file contains configuration data only. Because 1532 mode data files do not contain a preamble, they cannot be used to input encrypted configuration files.

In addition to being a configuration interface, JTAG also allows the user to program the 128-bit encryption key. In fact, JTAG is the only way to program the key.

For detailed information on LatticeECP2/M configuration including bitstream file sizes, refer to TN1108, LatticeECP2/M sysCONFIG Usage Guide. Note that bitstream sizes vary depending on the configuration mode.

Bitstream Encryption/Decryption Flow

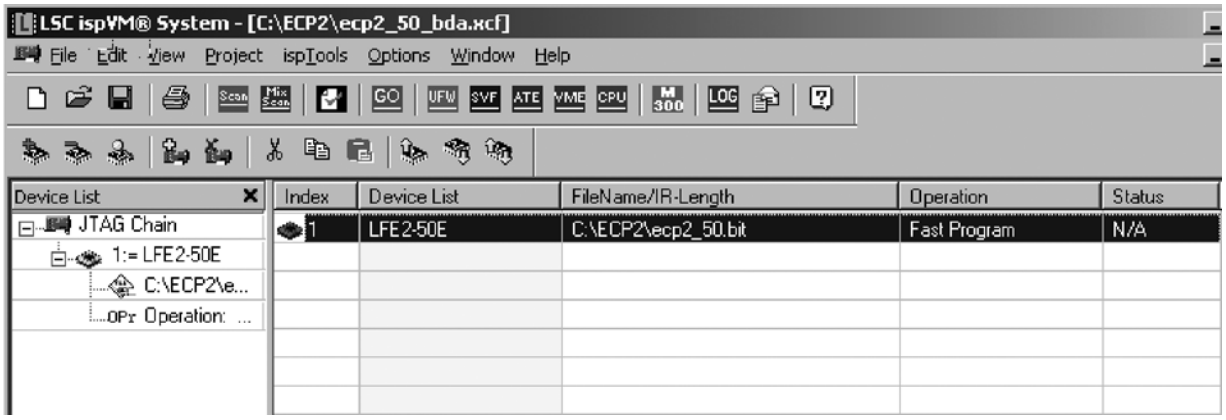
The LatticeECP2/M “S” versions support both encrypted and non-encrypted bitstreams. Since the non-encrypted flow is covered in TN1108, LatticeECP2/M sysCONFIG Usage Guide, this document will concentrate on the additional steps needed for the encrypted flow. The encrypted flow adds only two steps to the normal FPGA design flow, encryption of the configuration bitstream and programming the encryption key into the LatticeECP2/M “S” version devices.

Encrypting the Bitstream

As with any other Lattice FPGA design flow, the engineer must first create the design using a version of ispLEVER® or Lattice Diamond™ design software which supports the encryption feature. You may need to request the Encryption Installer to enable access to the additional encryption software. The design is synthesized, mapped, placed and routed, and verified. Once the engineer is satisfied with the design a bitstream is created and loaded into the FPGA for final debug. After the design has been debugged it is time to secure the design.

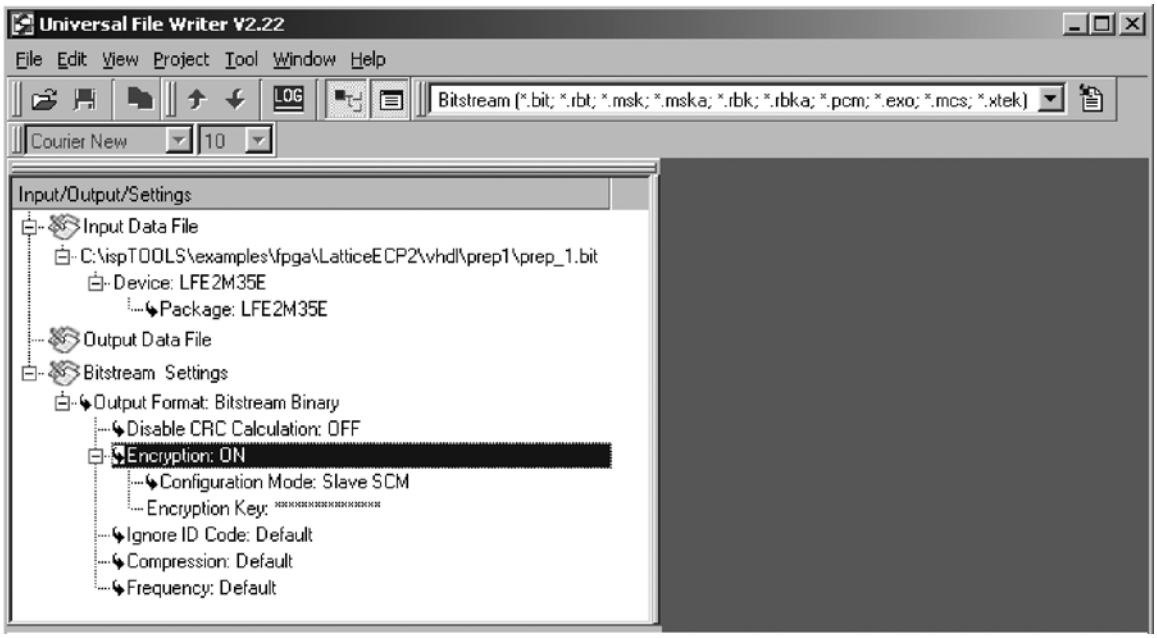
The bitstream can be encrypted using an appropriate version of ispLEVER by going to the Tools pull-down menu and selecting Security features or by using the Universal File Writer (UFW), which is part of the Lattice ispVM® System tool suite. The file is encrypted using ispVM as follows. To encrypt a bitstream in Diamond, refer to Appendix A.

Figure 16-3. ispVM Main Window



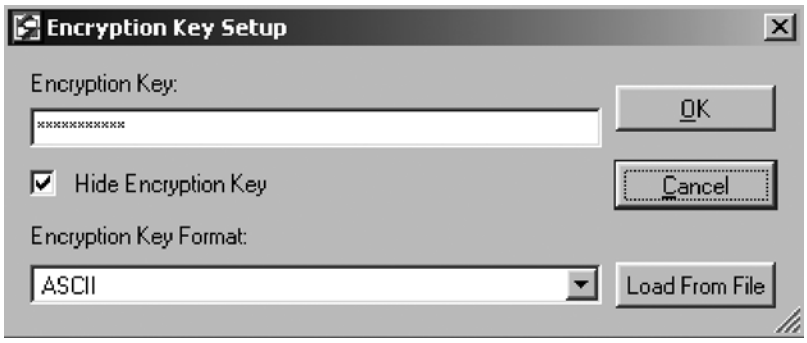
1. Start ispVM. You can start ispVM from within ispLEVER or from the **Start -> Programs** menu in Windows. You should see a window that looks similar to Figure 16-3. Click on the **UFW** button on the toolbar. You will see a window similar to Figure 16-4. ispVM cannot be invoked within Diamond.

Figure 16-4. Universal File Writer (Encryption Option)



2. Double click on **Input Data File** and browse to the non-encrypted bitstream created in ispLEVER or Diamond. Double-click on **Output Data File** and select an output file name. Right-click on **Encryption** and select **ON**. Right-click on **Configuration Mode** and select the type of device the FPGA will be configuring from, such as SPI Serial Flash. Right-click on **Encryption Key** and select **Edit Encryption Key**. You will see a window that looks similar to Figure 16-5.

Figure 16-5. Encryption Key Dialog Window



3. Enter the desired 128-bit key. The key can be entered in Hexadecimal or ASCII. Hex supports 0 through f and is not case sensitive. ASCII supports all alphanumeric characters, as well as spaces, and is case sensitive. Note: be sure to remember this key. Lattice cannot recover an encrypted file if the key is lost. Click on **OK** to go back to the main UFW window.
4. From the menu bar, click on **Project -> Generate** to create the encrypted bitstream file.
5. The bitstream can now be loaded directly into non-volatile configuration storage (such as SPI Serial Flash) using a Lattice ispDOWNLOAD® Cable, a third-party programmer, or any other method normally used to program a non-encrypted bitstream. However, before the LatticeECP2/M can configure from the encrypted file the 128-bit key used to encrypt the file must be programmed into the one-time programmable fuses on the FPGA.

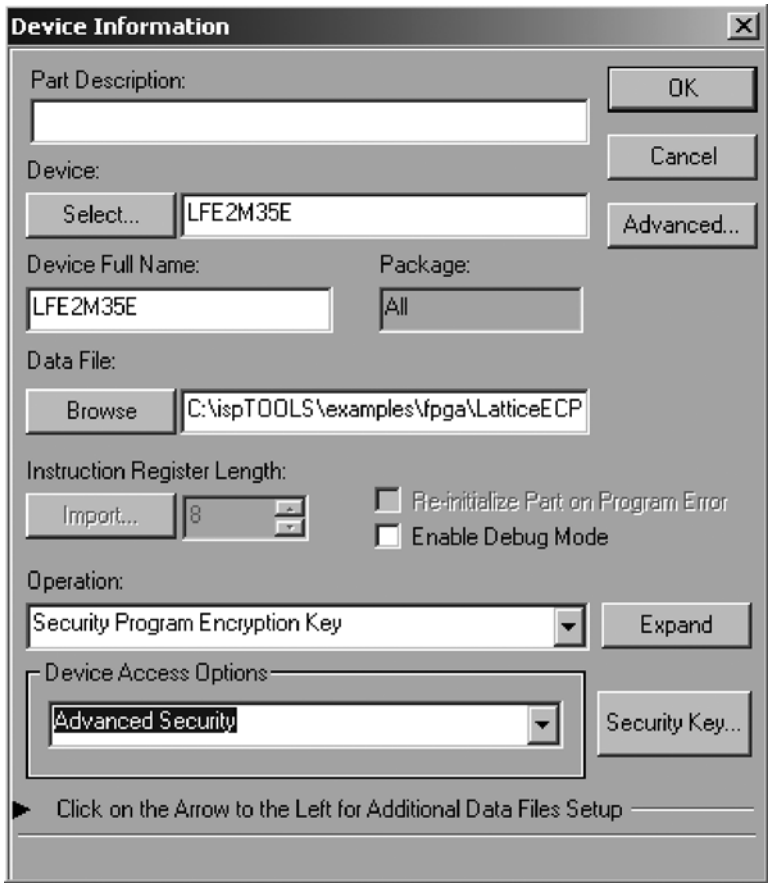
Programming the 128-bit Key

The next step is to program the 128-bit encryption key into the one-time programmable fuses on the LatticeECP2/M. Note that this step is separated from file encryption to allow flexibility in the manufacturing flow. For instance, the board manufacturer might program the encrypted file into the SPI Serial Flash, but the key might be programmed at the user's facility. This flow adds to design security and it allows the user to control over-building of a design. Over-building occurs when a third party builds more boards than are authorized and sells them to grey market customers. If the key is programmed at the factory, then the factory controls the number of working boards that enter the market. The LatticeECP2/M "S" version will only configure from a file that has been encrypted with the same 128-bit key that is programmed into the FPGA.

To program the key into the LatticeECP2/M "S" version, proceed as follows.

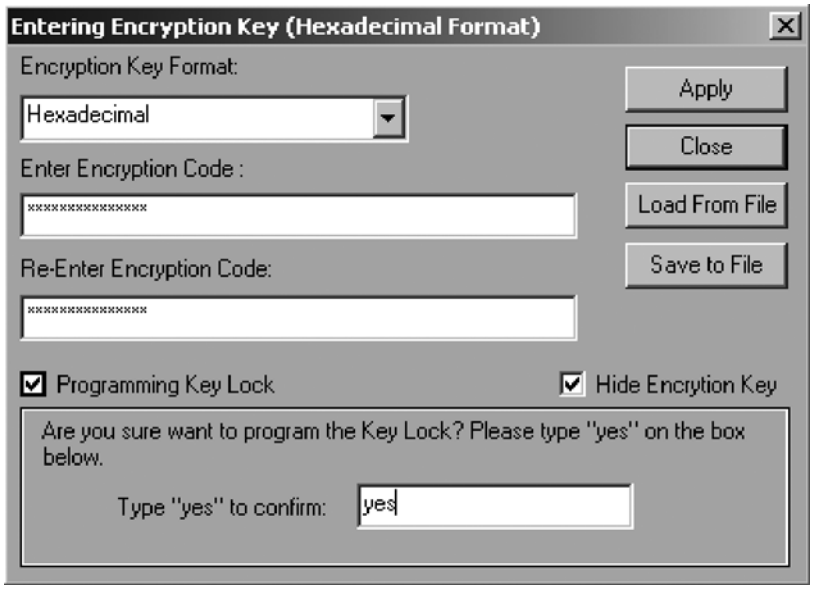
1. Attach a Lattice ispDOWNLOAD cable from a PC to the JTAG connector wired to the LatticeECP2/M (note that the 128-bit key can only be programmed into the LatticeECP2/M using the JTAG port). Apply power to the board.
2. Start the ispVM System software. ispVM can be started from within the ispLEVER design tool (ispVM cannot be invoked from within Diamond) or from the **Start -> Programs** menu in Windows. You should see a window that looks similar to Figure 16-3. If the window does not show the board's JTAG chain then proceed as follows. Otherwise, proceed to step 3.
    a. Click the **SCAN** button in the toolbar to find all Lattice devices in the JTAG chain. The chain shown in Figure 16-3 has only one device, the LatticeECP2.

Figure 16-6. Device Information Window (Encryption Option)



3. Double-click on the line in the chain containing the LatticeECP2. This will open the Device Information window (see Figure 5). From the Device Access Options drop-down box select **Security Mode**, then click on the **Security Key** button to the right. The window will look similar to Figure 16-7.

Figure 16-7. Enter the Encryption Key



4. Enter the desired 128-bit key. The key can be entered in Hexadecimal or ASCII. Hex supports 0 through f and is not case sensitive. ASCII supports all alphanumeric characters, as well as spaces, and is case sensitive. This key must be the same as the key used to encrypt the bitstream. The LatticeECP2/M will only configure from an encrypted file whose encryption key matches the one loaded into the FPGA's one-time programmable fuses. *Note: be sure to remember this key. Once the Key Lock is programmed, Lattice Semiconductor cannot read back the one-time programmable key.*

    a. The key can be saved to a file using the **Save to File** button. The key will be encrypted using an 8-character password that the user selects. The name of the file will be <project_name>.bek. In the future, instead of entering the 128-bit key, simply click on **Load from File** and provide the password.

5. Programming the Key Lock secures the 128-bit encryption key. Once the Key Lock is programmed and the device is power cycled, the 128-bit encryption key cannot be read out of the device. When satisfied, type **Yes** to confirm, then click **Apply**.

6. From the main ispVM window (Figure 16-3) click on the green **GO** button on the toolbar to program the key into the LatticeECP2/M one-time programmable fuses. When complete, the LatticeECP2/M will only configure from a bitstream encrypted with a key that exactly matches the one just programmed.

Verifying a Configuration

As an additional security step when an encrypted bitstream is used, the readback path from the SRAM fabric is automatically blocked. In this case, for all ports, a read operation will produce all zeros. However, even when the configuration bitstream has been encrypted and readback disabled, there are still ways to verify that the bitstream was successfully downloaded into the FPGA.

If the SRAM fabric is programmed directly, the data is first decrypted and then the FPGA performs a CRC on the data. If all CRCs pass, configuration was successful. If a CRC does not pass, the DONE pin will stay low and INITN will go from high to low (for more information on this type of error, refer to TN1108, LatticeECP2/M sysCONFIG Usage Guide.

If the encrypted data is stored in non-volatile configuration memory, such as SPI Serial Flash, the data is stored encrypted. A bit-for-bit verify can be performed between the encrypted configuration file and the stored data.

File Formats

The base binary file format is the same for all non-encrypted, non-1532 configuration modes. Different file types (hex, binary, ASCII, etc.) may ultimately be used to configure the device, but the data in the file is the same. Table 16-2 shows the format of a non-encrypted bitstream. The bitstream consists of a comment field, a header, the preamble, and the configuration setup and data.

Table 16-2. Non-Encrypted Configuration Data

Frame	Contents	Description
Comments	(Comment String)	ASCII Comment (Argument) String and Terminator
Header	1111...1111	16 Dummy bits
	1011110110110011	16-bit Standard Bitstream Preamble (0xBDB3)
Verify ID		64 bits of command and data
Control Register 0		64 bits of command and data
Reset Address		32 bits of command and data
Write Increment		32 bits of command and data
Data 0		Data, 16-bit CRC, and Stop bits
Data 1		Data, 16-bit CRC, and Stop bits
. . .	. . .	. . .
Data n-1		Data, 16-bit CRC, and Stop bits
End	1111...1111	Terminator bits and 16-bit CRC
Usercode		64 bits of command and data
SED CRC		64 bits of command and data
Program Security		32 bits of command and data
Program Done		32 bits of command and data, 16-bit CRC
NOOP	1111...1111	64 bits of NOOP data
End	1111...1111	32-bit Terminator (all ones)

Note: The data in this table is intended for reference only.

Table 16-3 shows a bitstream that is built for encryption but has not yet been encrypted. The highlighted areas will be encrypted. The changes between Table 16-2 and Table 16-3 include the following:

- The Program Security frame (readback disable) has been moved to the beginning of the file so that readback is turned off at the very beginning of configuration. This is an important security feature that prevents someone from interrupting the configuration before completion and reading back unsecured data.
- A copy of the usercode is placed in the non-encrypted comment string. This has been done to allow the user a method to identify an encrypted file. For example, the usercode could be used as a file index or a “hint”. Note that the usercode itself, while encrypted in the configuration data file, is not encrypted on the device. At configuration the usercode is decrypted and placed in the JTAG Usercode register. This allows the user a method to identify the data in the device. The JTAG Usercode register can be read back at any time, even when all SRAM readback paths have been turned off. The usercode can be set to any 32-bit value. For information on how to set usercode, see the ispLEVER or Diamond help facility.
- A copy of CONFIG_MODE, one of the global preferences, is placed in the non-encrypted comment string. CONFIG_MODE can be SPI/SPIm, Slave SCM, or Slave PCM.

Note that if the global COMPRESS_CONFIG option is turned ON using ispLEVER Design Planner or UFW, data compression will be performed before encryption. To set this configuration option in Diamond, see Appendix A.

Table 16-3. Configuration File Just Before Encryption

Frame	Contents	Description
Comments	(Comment String)	ASCII Comment (Argument) String and Terminator
Header	1111...1111	16 Dummy Bits
		16-bit Standard Bitstream Preamble
Verify ID		64 bits of Command and Data
Control Register 0		64 bits of Command and Data
Program Security		32 bits of Command and Data
Reset Address		32 bits of Command and Data
Write Increment		32 bits of Command and Data
Data 0		Data, 16-bit CRC, and Stop Bits
Data 1		Data, 16-bit CRC, and Stop Bits
. . .	. . .	. . .
Data n-1		Data, 16-bit CRC and Stop Bits
End	1111...1111	Terminator Bits and 16-bit CRC
Usercode		64 Bits of Command and Data
SED CRC		64 Bits of Command and Data
Program Done		32 Bits of Command and Data, 16-bit CRC
NOOP	1111...1111	64 bits of NOOP data
End	1111...1111	32-bit Terminator (All Ones).

Note: The data in this table is intended for reference only. The shaded areas will be encrypted.

Once encrypted, besides the obvious encryption of the data itself, the file will have additional differences from a non-encrypted file (refer to Tables 16-4, 16-5, and 16-6).

- There are three preambles, the encryption preamble, alignment preamble, and the bitstream preamble. The alignment preamble marks the beginning of the encrypted data. The entire original bitstream, including the bitstream preamble are all encrypted, per Table 16-3. The comment string, the encryption preamble, dummy data, and alignment preamble are not encrypted.
- The decryption engine within the FPGA takes some time to perform its task; extra time is provided in one of two ways. For master configuration modes (SPI and SPIm) the FPGA drives the configuration clock, so when extra time is needed the FPGA stops sending configuration clocks. For slave configuration modes (Bitstream-Burst, Slave Serial, and Slave Parallel) the data must be padded to create the extra time. Because of this there are several different file formats for encrypted data (see Tables 16-4, 16-5, and 16-6). Note that because of the time needed to decrypt the bitstream it takes longer to configure from an encrypted data file than it does from a non-encrypted file. The bitstream sizes may vary depending on the configuration mode. For exact file sizes, refer to TN1108, LatticeECP2/M sysCONFIG Usage Guide.

Table 16-4. Encrypted File Format for a Master Mode

Frame	Contents	Description
Comments	(Comment String)	ASCII Comment (Argument) String and Terminator.
Header	1111...1111	16 Dummy bits.
		16-bit Encryption Preamble.
30,000 Filler Bits		This allows time for the device to load and hash the 128-bit encryption key.
Alignment Preamble		16-bit Alignment Preamble.
	1	1-bit Dummy Data.
Data		There are no dummy filler bits when the bitstream is generated for master programming modes. The CCLK of the master device stops the clock when it needs time to decrypt the data. It resumes the clock when ready for new data - Encrypted.
Program Done		32-bit Program Done Command - Encrypted.
End	1111...1111	32-bit Terminator (all ones) - Encrypted.
Filler Bits		Filler to meet the bound requirement.
Dummy Data	1111...1111	200 bits of Dummy Data (all ones). Provides a delay to turn off the decryption engine.

Note:The data in this table is intended for reference only. The shaded area is encrypted data.

Table 16-5. Encrypted File Format for a Slave Serial Mode

Frame	Contents	Description
Comments	(Comment String)	ASCII Comment (Argument) String and Terminator.
Header	1111...1111	2 Dummy Bytes.
		16-bit Encryption Preamble
30,000 Filler Bits		This allows time for the device to load and hash the 128-bit encryption key.
Alignment Preamble		16-bit Alignment Preamble.
	1	1-bit Dummy Data.
Data		128 bits of Configuration Data.
		64 bits of all ones data. Provides a delay for the decryption engine to decrypt the 128 bits of data just received. If the peripheral device can provide the needed 64 clocks while pausing data, then the 64 bits of dummy data are not required, saving file size.
	...	
		Last 128 bits of the last Frame of Configuration Data.
		64 bits of all ones data. Provides a delay for the decryption engine to decrypt the 128 bits of data just received. If the peripheral device can provide the needed 64 clocks while pausing data, then the 64 bits of dummy data are not required, saving file size.
Program Done		32-bit Program Done Command - Encrypted.
End		32-bit Terminator (all ones) - Encrypted.
Filler Bits		Filler to meet the bound requirement.
Delay		64 bits of all ones data. Delay to decrypt the Program Done command and the filler.
Dummy Data	1111...1111	200 bits of Dummy Data (all ones), to provide delay to turn off the decryption engine.

Note:The data in this table is intended for reference only. The shaded area is encrypted data.

Table 16-6. Encrypted File Format for a Slave Parallel Mode

Frame	Contents	Description
Comments	(Comment String)	ASCII Comment (Argument) String and Terminator.
Header	1111...1111	2 Dummy Bytes.
		2-byte Encryption Preamble.
30,000 Filler Bytes		This allows time for the device to load and hash the 128-bit encryption key.
Alignment Preamble		2-byte Alignment Preamble.
	11111111	1-byte Dummy Data.
Data		16 bytes of Configuration Data.
		64 bytes (clocks) of all ones data. Provides a delay for the decryption engine to decrypt the 16 bytes of data just received. If the peripheral device can provide the needed 64 clocks while pausing data, then the 64 bytes of dummy data are not required, saving file size.
	...	
		16 bytes of Configuration Data.
		64 bytes (clocks) of all ones data. Provides a delay for the decryption engine to decrypt the 16 bytes of data just received. If the peripheral device can provide the needed 64 clocks while pausing data, then the 64 bytes of dummy data are not required, saving file size.
Program Done		4-byte Program Done Command - Encrypted.
End		4-byte Terminator (all ones) - Encrypted.
Filler Bits		Filler to meet the bound requirement.
Delay		64 bytes of all ones data. Delay to decrypt the Program Done command and the filler.
Dummy Data	1111...1111	200 bytes of Dummy Data (all ones), to provide delay to turn off the decryption engine.

Note:The data in this table is intended for reference only. The shaded area is encrypted data.

Decryption Flow

From the user's point of view, as compared to the encryption flow just discussed, the decryption flow is much simpler.

When data comes into the FPGA the decoder starts looking for the preamble (see Figure 16-1) and all information before the preamble is ignored. The preamble, along with the compression bit in Control Register 0, determines the path of the configuration data.

If the decoder detects a standard bitstream preamble in the bitstream it knows that this is a non-encrypted data file. The decoder then examines Control Register 0 in the bitstream to determine if the file has been compressed. If the file has not been compressed then the Raw data path is selected (see Figure 16-1). If the file has been compressed then the Decompressed path is selected; CRC is then checked and the SRAM fuses programmed.

If the decoder detects an encryption preamble in the bitstream it knows that this is an encrypted data file. If an encryption key has not been programmed, the encrypted data is blocked and configuration fails (the DONE pin stays low), if the proper key has been programmed then configuration can continue. The next block read contains 30,000 clocks of filler data. This delay allows time for the FPGA to read the key fuses and prepare the decryption engine. The decoder keeps reading the filler data looking for the alignment preamble. Once found, it knows that the following data needs to go through the decryption engine. It first looks for the standard preamble. Once found, then it reads the Control Register 0 frame. The decoder then examines the decrypted Control Register 0 contents to determine if the file has been compressed. If the file has not been compressed then the Decrypted data path is used, if the file has been compressed then the decrypted data is passed through the decompression engine and the Decompressed path is selected (refer to the block diagram, Figure 16-1). CRC is then checked and the SRAM fuses programmed once the bitstream preamble is read. The decryption and decompression engines are turned off

when the internal Done bit is set at the end of configuration. This is done so that if there is any data overflow (to other devices in a chain) the downstream devices will receive raw data from configuration storage.

But what happens if the key in the FPGA does not match the key used to encrypt the file? Once the data is decrypted, the FPGA expects to find a valid standard bitstream preamble (BDB3), along with proper commands and data that pass CRC checks. If the keys do not match then the decryption engine will not produce a proper configuration bitstream; either configuration will not start because the preamble was not found (the INITN pin stays high and the DONE pin stays low) or CRC errors will occur, causing the INITN pin to go low to indicate the error (see TN1108, LatticeECP2/M sysCONFIG Usage Guide, for more information on INITN and DONE).

References

- TN1108, LatticeECP2/M sysCONFIG Usage Guide
- Federal Information Processing Standard Publication 197, Nov. 26, 2001. Advanced Encryption Standard (AES)

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
April 2006	01.0	Initial release.
September 2006	01.1	Added information throughout for LatticeECP2M support.
		Updated screen shots based on the latest software version.
		Provided clarification in Table 1.
		Changed Bitstream Preamble to Alignment Preamble through out the document.
		Reworded sections of the document to provide additional information/clarification.
March 2007	01.2	Added “S” series encryption information throughout.
August 2007	01.3	Updated for “S” series reduced frequency support and special requirement for CCLK and TCK on LatticeSCM, and PCM and JTAG configuration modes.
June 2010	01.4	Updated for Lattice Diamond design software support.

Appendix A. Lattice Diamond Usage Overview

Setting Global Preferences in Diamond

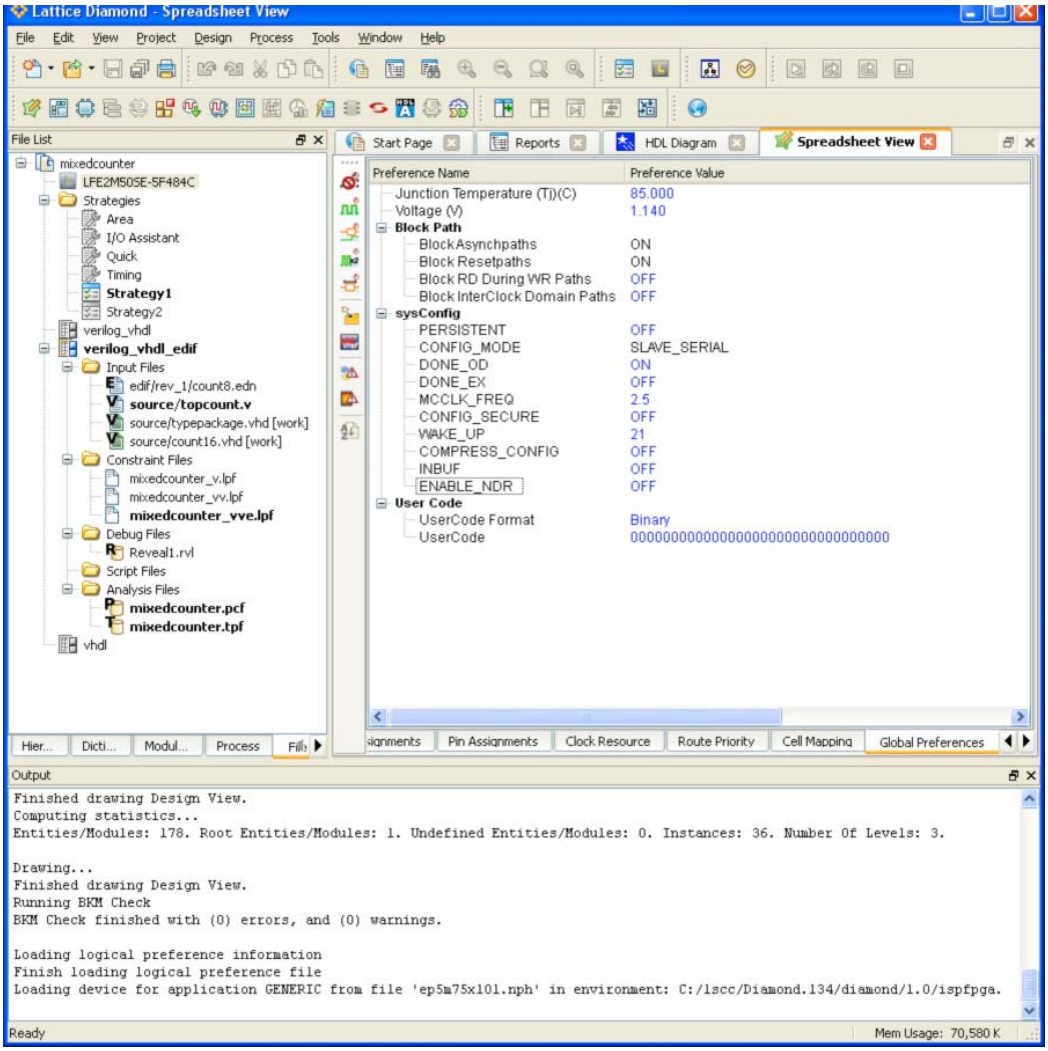
To set any of the Global preferences in Table 16-7, do the following in Diamond:

- Invoke the Spreadsheet View by selecting **Tools > Spreadsheet View**.
- Select the **Global Preferences Tab** beneath the Spreadsheet View pane as shown in Figure 16-8.
- Right-click on the **Preference Value** to be set. In the drop-down menu, select the desired value.

Table 16-7. Global Preferences

Preference Name	Values
PERSISTENT	ON Off
CONFIG_MODE	SLAVE_SERIAL JTAG NONE SLAVE_PARALLEL SPI SPIm
DONE_OD	ON Off
DONE_EX	OFF
MCCLK_FREQ	2.5 5.4 10 34 41 45
CONFIG_SECURE	OFF ON
WAKE_UP	An integer between 1 and 25
COMPRESS_CONFIG	OFF ON
INBUF	OFF ON
ENABLE_NDR	OFF ON

Figure 16-8. Global Preferences Tab



Setting Bitstream Generation Options in Diamond

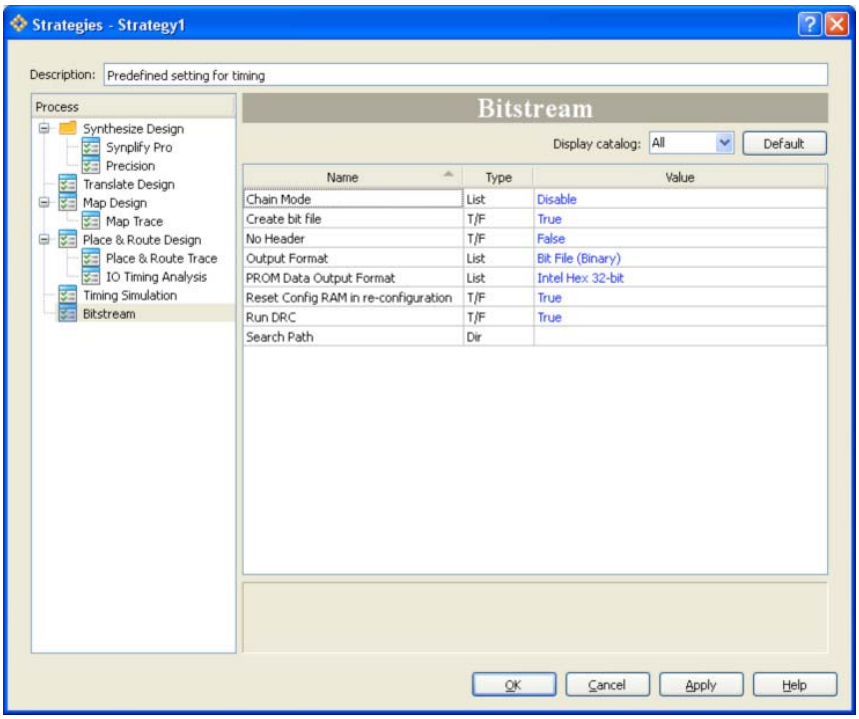
To set any of the Bitstream Generation options listed in Table 16-8, do the following:

- In the **File List** pane, double-click the left mouse button on a **Strategy** to invoke the Strategy settings window.
- In the **Process** pane, left-click on **Bitstream**. All options related to generating a bitstream can be set in this window.

Table 16-8. Bitstream Generation Options

Preference Name	Values
Chain Mode	Disable (default) Bypass Flowthrough
Create bit file	True
No Header	True False
Output Format	Bit File (Binary) Mask and Readback File (ASCII) Mask and Radback File (Binary) Raw Bit file (ASCII)
PROM Data Output Format	Intel Hex 32-bit Motorola Hex 32-bit
Reset Config RAM in re-configuation	True False
Run DRC	True False
Search Path	(Enter a value or browse to specify the search path)

Figure 16-9. Bitstream Generation Options



- Double-click the left mouse button on the **Value** you want to set. Select the desired value from the drop-down menu.

  *Note: An explanation of the option is displayed at the bottom of the window. The **Help** button also invokes online help for the option.*

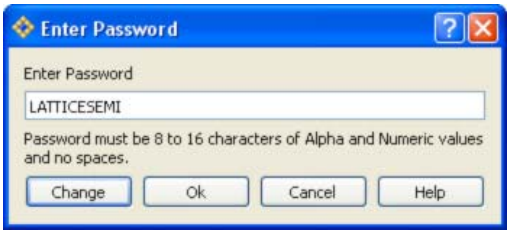
- Select **OK**. You can then run the Bitstream File process.

Setting Security Options in Diamond

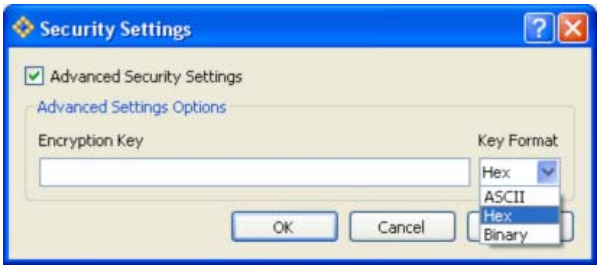
Prior to setting security options in Diamond, you must have installed the Encryption Control Pack. You must also have selected an encrypted device in your project.

To Set Security Settings, do the following:

- Select the **Tools > Security Setting** option. The following dialog box appears:



- If desired, select **Change** and enter a password.
- Select **OK**. A dialog window appears to enter an encryption key.
- If you do not want to enable an encryption key, select **OK**.
- If you do want to enable an encryption key, select the **Advanced Security Settings** checkbox, enter the **Key Format**, and then enter the **Encryption Key**.



- Select **OK** to create the encryption files.

LatticeECP2/M Soft Error Detection (SED) Usage Guide

January 2012 Technical Note TN1113

Introduction

Soft errors occur when high-energy charged particles alter the stored charge in a memory cell in an electronic circuit. The phenomenon first became an issue in DRAM, requiring error detection and correction for large memory systems in high-reliability applications. As device geometries have continued to shrink, the probability of soft errors in SRAM has become significant for some systems. Designers are using a variety of approaches to minimize the effects of soft errors on system behavior.

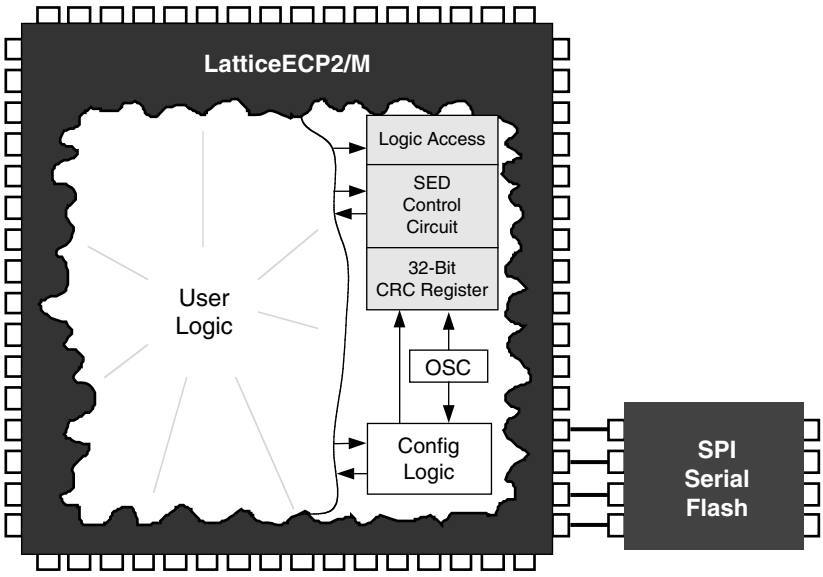
SRAM-based FPGAs store logic configuration data in SRAM cells. As the number and density of SRAM cells in an FPGA increase, the probability that a soft error will alter the programmed logical behavior of the system increases. A number of approaches have been taken to address this issue, but most involve Intellectual Property (IP) cores that the user instantiates into the logic of their design, using valuable resources and possibly affecting design performance.

This document describes the hardware based soft error detect (SED) approach taken by Lattice Semiconductor for LatticeECP2™ and LatticeECP2M™ FPGAs.

SED Overview

The SED hardware in the LatticeECP2/M devices consists of an access point to FPGA configuration memory, a controller circuit, and a 32-bit register to store the CRC for a given bitstream (see Figure 17-1). The SED hardware reads serial data from the FPGA's configuration memory and calculates a CRC. The data that is read, and the CRC that is calculated, does not include EBR memory or PFUs used as RAM. The calculated CRC is then compared with the expected CRC that was stored in the 32-bit register. If the CRC values match it indicates that there has been no configuration memory corruption, but if the values differ an error signal is generated. SED checking does not impact the performance or operation of the user logic.

Figure 17-1. System Block Diagram[1]



1. Any kind of configuration memory can be used, including the SPI configuration shown.

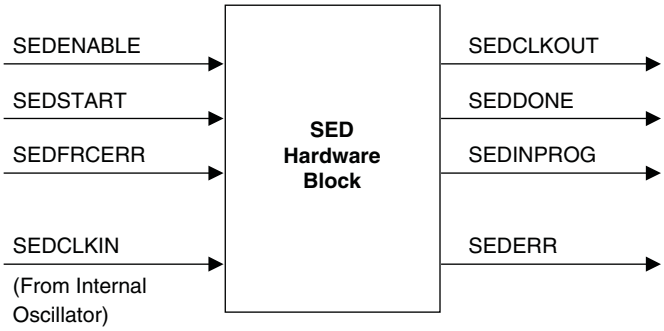
Note that the calculated CRC is based on the particular arrangement of configuration memory for a particular design. Consequently, the expected CRC results cannot be specified until after the design is placed and routed. The Lattice Diamond® bitstream generation software analyzes the configuration of a placed and routed design and updates the 32-bit SED CRC register contents during bitstream generation.

The following sections describe the LatticeECP2/M SED implementation and flow, along with some sample code to get started with.

Hardware Description

As shown in Figure 17-2, the LatticeECP2/M SED hardware has several inputs and outputs that allow the user to control, and monitor, SED behavior.

Figure 17-2. Signal Block Diagram



Signal Description

Table 17-1. SED Signal Description

Signal Name	Direction	Active	Description
SEDCLKIN	Input	N/A	Clock
SEDENABLE	Input	High	SED enable
SEDCLKOUT	Output	N/A	Output clock
SEDSTART	Input	High	Start SED cycle
SEDINPROG	Output	High	SED cycle is in progress
SEDDONE	Output	High	SED cycle is complete
SEDFRCERR	Input	High	Force an SED error flag
SEDERR	Output	High	SED error flag

SEDCLKIN

Clock input to the SED hardware.

This clock is derived from the LatticeECP2/M on-chip oscillator. The on-chip oscillator output goes through a divider to create MCCLK. MCCLK goes through another divider to create SEDCLKIN.

The software default for MCCLK is 2.5 MHz, but this can be modified using the MCCLK_FREQ global preference in the Global Preferences tab of the Diamond Spreadsheet View (see TN1108, LatticeECP2/M sysCONFIG Usage Guide, for possible values of MCCLK).

The divider for SEDCLKIN can be set to 1, 2, 4, 8, 16 or 32. The software default is 1, so the default SEDCLKIN frequency is 2.5 MHz. The divider value can be set using a parameter, see the example code at the end of this document. Care must be taken to ensure that the SEDCLKIN setting is at least 20 MHz. Refer to Appendix A for details on MCCLK and SEDCLKIN frequencies.

Note that SEDCLKIN is an internally generated signal, so it should not be included as an input in the user design. See the examples at the end of this document. Also note that while inputs to the SED block are clocked using SEDCLKIN, no attempt has been made to synchronize between clock domains. If this is a concern for a particular design then the designer will need to provide synchronization.

SEDENABLE

Active high input to the SED hardware, sampled on the rising edge of SEDCLKIN.

Table 17-2. SEDENABLE

State	Description
1	Enables output of SEDCLKOUT, arms SED hardware.
0	Aborts SED and forces all SED hardware outputs low.

SEDCLKOUT

Gated version of SEDCLKIN, SEDCLKOUT is gated by SEDENABLE.

SEDSTART

Active high input to the SED hardware, sampled on the rising edge of SEDCLKIN.

Table 17-3. SEDSTART

State	Description
1	Start error detection. Must be high a minimum of one SEDCLKIN period.
0	No action.

SEDFRCERR

Active high input to the SED hardware, sampled on the rising edge of SEDCLKIN.

Table 17-4. SEDFRCERR

State	Description
1	Forces SEDERR high, simulating an SED error.
0	No action.

SEDINPROG

Active high output from the SED hardware, clocked out on the rising edge of SEDCLKOUT.

Table 17-5. SEDINPROG

State	Description
1	SED checking is in progress, goes high on the clock following SEDSTART high.
0	SED checking is not active.

SEDDONE

Active high output from the SED hardware, clocked out on the rising edge of SEDCLKOUT.

Table 17-6. SEDDONE

State	Description
1	SED checking is complete. Reset by a high on SEDSTART or a low on SEDENABLE.
0	SED checking is not complete.

SEDERR

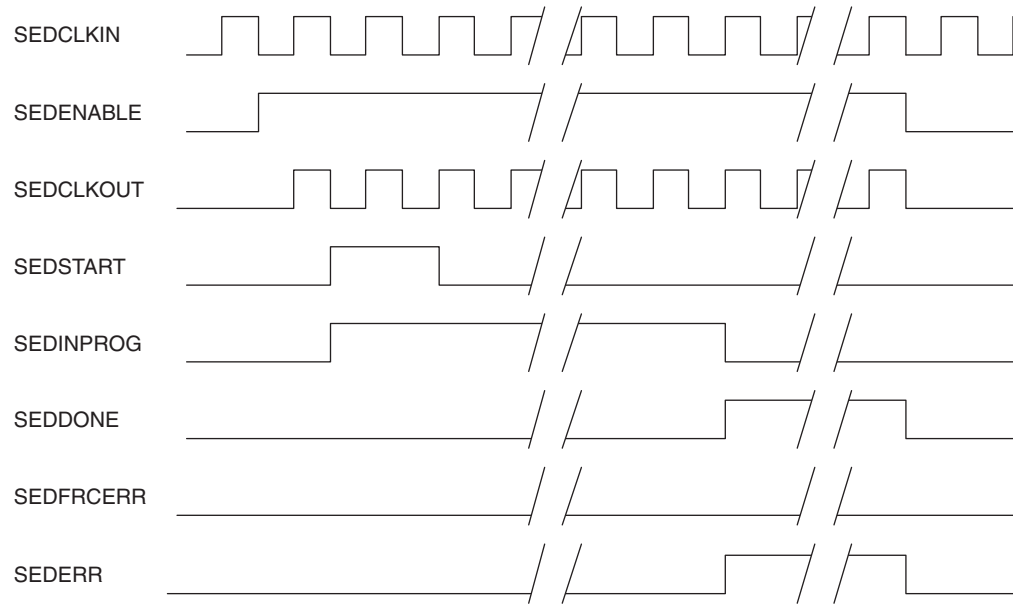
Active high output from the SED hardware, clocked out on the rising edge of SEDCLKOUT.

Table 17-7. SEDERR

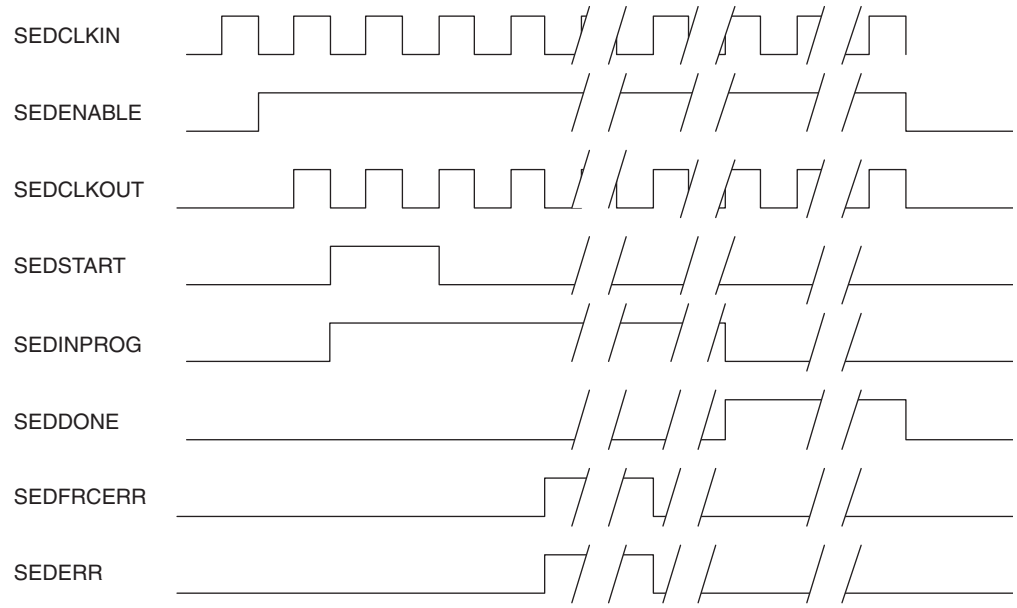
State	Description
1	SED has detected an error. Reset by SEDENABLE going low.
0	SED has not detected an error.

SED Flow

Figure 17-3. Timing Diagram



Normal Failure



Failure Forced With SEDFRCERR

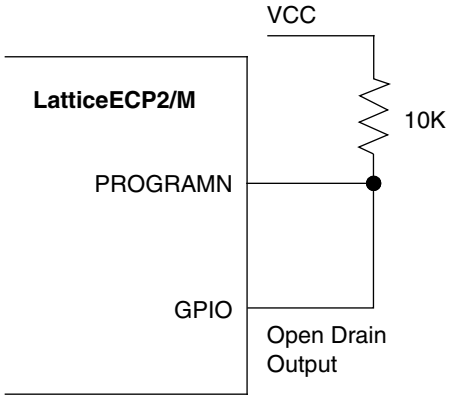
The general SED flow is as follows.

1. User logic sets SEDENABLE high. This signal may be tied high if desired.
2. User logic sets SEDSTART high. SEDINPROG goes high. If SEDDONE is already high it is driven low. SEDSTART may be tied high to enable continuous SED checking.
3. SED starts reading back data from the configuration SRAM.

4. SED finishes checking. SEDERR is updated, SEDINPROG goes low, and SEDDONE goes high.
5. If SEDERR is driven high there are only two ways to reset it, drive SEDENABLE low or reconfigure the FPGA.

The user has two choices when an error is detected, ignore the error, and possibly log it, or reconfigure the FPGA. Reconfiguration can be accomplished by driving the PROGRAMN pin low; this can be done with external logic or by wiring one of the FPGA's general purpose I/Os to the PROGRAMN pin and toggling the pin with user logic, perhaps something as simple as inverting SEDERR. If a general purpose I/O is tied to PROGRAMN it is recommended that the I/O Type be set to open drain and an external pull-up resistor be connected to the pin.

Figure 17-4. Example Schematic



SED Run Time

The amount of time needed to perform an SED check depends on the density of the device and the frequency of SEDCLKIN. There will also be some overhead time for calculation, but it is fairly short in comparison. An approximation of the time required can be found by using the following formula:

Maxbits / SEDCLKIN = Time

Maxbits is in mega-bits and depends on the density of the FPGA (see Table 17-8). SEDCLKIN is frequency in MHz. Time is in seconds

For example, if the design is using a LatticeECP2 with 50K look-up tables and the SEDCLKIN is set in the software to be 20 MHz:

8.9 Mbits / 20 MHz = 0.445 seconds

In this example, SED checking will take approximately 0.445 seconds. Remember that this happens in the background and does not affect user logic performance.

Note that the internal oscillator used to generate SEDCLKIN can vary by ±30%.

Table 17-8. SED Run Time

Density	Bitstream Size (Mb)	Run Time[1] (ms)
ECP2-6	1.5	75
ECP2-12	2.9	145
ECP2-20	4.5	225
ECP2-35	6.3	315
ECP2-50	8.9	445
ECP2-70	13.3	665
ECP2M-20	5.9	295
ECP2M-35	9.8	490
ECP2M-50	15.8	790
ECP2M-70	19.8	990
ECP2M-100	25.6	1280

1. Based on SEDCLKIN = 20 MHz.

Sample Code

The following simple example code shows how to instantiate the SED. In the example the SED is always on and always running, and the outputs of the SED hardware have been routed to FPGA output pins.

Note that the SEDAA primitive is part of ispLEVER 6.0 or later.

VHDL Example

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity example is
   port (
        Sed_Done     : out std_logic;
        Sed_In_Prog  : out std_logic;
        Sed_Clk_out  : out std_logic;
        Sed_out      : out std_logic);
end;

architecture behavioral of example is

   component  SEDAA -- SED component
      generic (OSC_DIV : integer := 1); -- set SEDCLKIN divider
      port (
        SEDENABLE    : in std_logic;
        SEDSTART     : in std_logic;
        SEDFRCERR    : in std_logic;
        SEDERR       : out std_logic;
        SEDDONE      : out std_logic;
        SEDINPROG    : out std_logic;
        SEDCLKOUT    : out std_logic) ;
   end component;

   begin
```

```
      isnt1: SEDAA
      generic map (OSC_DIV=> "1")
      port map (
         SEDENABLE    => '1',    -- tied high
         SEDSTART     => '1',    -- tied high
         SEDFRCERR    => '0',  -- tied low
         SEDERR       => Sed_out,  -- wired to an output
         SEDDONE      => Sed_Done,  -- wired to an output
         SEDINPROG    => Sed_In_Prog,  -- wired to an output
         SEDCLKOUT    => Sed_Clk_out ) ;    -- wired to an output

end behavioral ;
```

Verilog Example

```
module example (
   Sed_Done,
   Sed_In_Prog,
   Sed_Clk_out,
   Sed_out) ;

output Sed_Done;
output Sed_In_Prog;
output Sed_Clk_out;
output Sed_out;

assign V_hi = 1'b1;
assign V_lo = 1'b0;

SEDAA
     #(.OSC_DIV(1))

SED_IP(
      .SEDENABLE(V_hi), // always high
      .SEDSTART(V_hi),    // always high
      .SEDFRCERR(V_lo), // always low
      .SEDERR(Sed_out),   // wired to an output
      .SEDDONE(Sed_Done), // wired to an output
      .SEDINPROG(Sed_In_Prog), // wired to an output
      .SEDCLKOUT(Sed_Clk_out)); // wired to an output

endmodule
```

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
April 2006	01.0	Initial release.
April 2006	01.1	Fixed VHDL code
September 2006	01.2	Changed FPGA family naming to show support for LatticeECP2M.
April 2007	01.3	Updated SED Flow timing diagram.
September 2007	01.4	Updated SEDCLKIN frequency setting.
February 2008	01.5	Updated Timing Diagram.
July 2008	01.6	Added note to SED Flow timing diagram.
January 2009	01.7	Updated Verilog Example code.
September 2009	01.8	Updated VHDL Example in the Sample Code section.
April 2010	01.9	Changed minimum operating frequency of SEDCLKIN to be at least 20 MHz.
		Updated SED Run Time table.
		Corrected formula for data in SED Run Time table to use SEDCLKIN = 20 MHz.
January 2012	02.0	Added Appendix A.

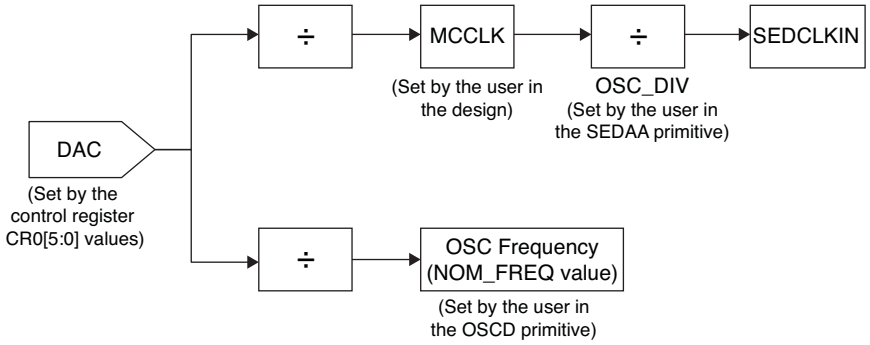
Appendix A. Calculating Exact MCCLK and SEDCLKIN Values

This appendix deals with the interdependency of the SEDCLKIN, MCCLK and the internal oscillator frequency. Figure 17-5 shows the hardware block diagram, including the internal oscillator, the MCCLK divider and the SEDCLKIN divider. The base frequency is generated by the DAC, which generates one out of four base frequencies. The internal oscillator frequency and the MCCLK and SEDCLKIN frequencies are generated by applying proper divider values to the DAC base frequency. The DAC is capable of generating only one of the four values: 260 MHz, 270 MHz, 300 MHz or 330 MHz. The selection of the base DAC frequency depends upon the internal oscillator frequency and this, in turn, affects the MCCLK and SEDCLKIN frequencies. The DAC base frequency is set by the control register CR0[5:0] values. The CR0 values can be observed in the Lattice ispVM™ System software. To see CR0 values in the ispVM tool, select **ispTools > ispVM Editors > "Control Register0 Editor.."**. The CR0[5:0] values corresponding to the DAC base frequency include:

- 011110 for 260 MHz
- 010001 for 270 MHZ
- 001001 for 300 MHz
- 000001 for 330 MHz

See the ispVM Help system for more details on the ControlRegister0 Editor.

Figure 17-5. Internal Clocking Scheme



The CR0 values are selected based on the selection of valid NOM_FREQ values of the OSCD primitive for the internal oscillator. See the LatticeECP2/M Family Data Sheet for more details on the OSCD primitive.

Tables 17-9 to 17-12 show valid NOM_FREQ values and corresponding CR0[5:0] values, resulting in one of the four DAC base frequencies.

Table 17-9. NOM_FREQ Values and CR0[5:0] Values that Result in a 260 MHz DAC Base Frequency

NOM_FREQ in OSCD Primitive (MHz)	Resulting CR0[5:0] Setting	Resulting Frequency (MHz)
2.5	011110	260
4.3		
5.4		
6.9		
8.1		
9.2		
10		
13		
26		

Table 17-10. NOM_FREQ Values and CR0[5:0] Values that Result in a 270 MHz DAC Base Frequency

NOM_FREQ in OSCD Primitive (MHz)	Resulting CR0[5:0] Setting	Resulting Frequency (MHz)
15	010001	270
34		
45		

Table 17-11. NOM_FREQ Values and CR0[5:0] Values that Result in a 300 MHz DAC Base Frequency

NOM_FREQ in OSCD Primitive (MHz)	Resulting CR0[5:0] Setting	Resulting Frequency (MHz)
30	001001	300

Table 17-12. NOM_FREQ Values and CR0[5:0] Values that Result in a 330 MHz DAC Base Frequency

NOM_FREQ in OSCD Primitive (MHz)	Resulting CR0[5:0] Setting	Resulting Frequency (MHz)
20	000001	330
41		
55		

The default value of NOM_FREQ is 2.5 MHz, or when the OSCD primitive is not explicitly instantiated. In that case, the CR0[5:0] value is 011110, resulting in 260 MHz of DAC base frequency. If the OSCD primitive is instantiated in the design, and the NOM_FREQ value is 30 MHz, then as per Table 17-10, the tool will set the CR[5:0] value as 001001, which results in a DAC base frequency of 300 MHz.

Calculating MCCLK Frequency

The MCCLK_FREQ is 2.5 MHz by default; however, this can be set in the attributes of the SED HDL component (see sample code section or Diamond help files) and Global Constraints tab of the Diamond Spreadsheet View. Whenever the user selects a particular MCCLK frequency, the most appropriate divider value is selected, based on the DAC base frequency. As mentioned in the previous section, the DAC base frequency depends upon the OSCD NOM_FREQ value selected. Table 17-13 shows the divider values to generate the nearest MCCLK frequency, based on the DAC frequency.

Table 17-13. Divider Values to Generate Nearest MCCLK Frequency, Based on DAC Frequency

	CR0[5:0] Setting	011110	010001	001001	000001
	DAC Base Frequency (MHz)	260	270	300	330
MCCLK_FREQ (MHz)	2.5	104	108	120	128
	4.3	60	62	70	76
	5.4	48	50	56	62
	6.9	38	40	44	48
	8.1	32	34	38	40
	9.2	28	30	32	36
	10	26	28	30	34
	13	20	20	24	26
	15	18	18	20	22
	20	14	14	16	16
	26	10	10	12	12
	30	8	10	10	12
	34	8	8	8	10
	41	6	6	6	8
	45	6	6	6	8
	55	6	6	6	8
	60	4	4	6	6
	130	2	2	2	2

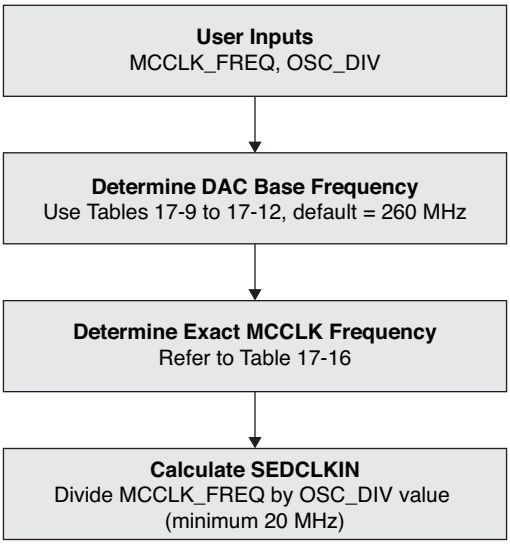
Table 17-13 shows all the valid divider values which exist in the LatticeECP2/M device. Because of these integer divider values, the exact MCCLK frequencies may not be observed in several cases. The actual MCCLK frequency is obtained by dividing the DAC base frequency by the divider value under that column, while the row corresponds to the desired MCCLK frequency.

For example, if the desired MCCLK value is 9.2 MHz, and the DAC base frequency is 300 MHz (CR0[5:0] = 001001), then the divider value would be 32. Thus, the MCCLK value would be 300/32 = 9. Similarly, with a DAC base frequency of 260 MHz, if MCCLK_ FREQ is 41, 45 or 55, the same divider value is selected, resulting in the same output frequency.

Calculating the SEDCLKIN

Once the MCCLK frequency has been calculated, the SEDCLKIN is calculated by dividing the MCCLK by a valid set of divider values, as defined by OSC_DIV parameter of the SEDAA primitive. In effect, SEDCLKIN = MCCLK/OSC_DIV. Due to the discrete nature of divider values and four different DAC Base frequencies, the exact SEDCLKIN cannot be expected. The variation can go up to ±30% as mentioned in the DC and Switching Characteristics section of the LatticeECP2/M Family Data Sheet. Figure 17-6 shows the steps required to calculate the SEDCLKIN frequency.

Figure 17-6. Calculating SEDCLKIN Frequency



LatticeECP2/M Hardware Checklist

September 2007 Technical Note TN1162

Introduction

When designing complex hardware using the LatticeECP2/M FPGA, designers must pay special attention to critical hardware configuration requirements. This technical note steps through these critical hardware implementation items relative to the LatticeECP2/M device. The device family consists of FPGA LUT densities ranging from 6K to 100K. This technical note assumes that the reader is familiar with the LatticeECP2/M device features as described in the LatticeECP2/M Family Data Sheet.

The critical hardware areas covered in this technical note are:

- Power supplies as they relate to the LatticeECP2/M supply rails and how to connect them to the PCB and the associated system
- Configuration mode selection for proper power-up behavior
- Device I/O interface and critical signals

Power Supplies

The V_{CC}, V_{CCIO8} and V_{CCAUX} power supplies determine the LatticeECP2/M internal "power good" condition. In addition to the three power supplies, there are $V_{CCIO0-7}$, V_{CCPLL} and V_{CCJ} supplies that power the I/O banks, PLL and JTAG port. All power supplies are required for proper device operation but since V_{CC}, V_{CCIO8} and V_{CCAUX} determine the device power-on condition, it is recommended that one of these supplies should be the final power supply to power up the LatticeECP2/M device after all other power supplies are stable. Table 18-1 shows the power supplies and the appropriate voltage levels for each supply.

Table 18-1. Power Supply Description and Voltage Levels

Supply	Voltage (Typical)	Description
V_{CC}	1.2V	Core power supply. A nominal trip point for Vcc power supply is 1.0V.
V_{CCAUX}	3.3V	Auxiliary power supply. A 3.3V supply that provides an internal reference to the input buffers. A nominal trip point for VCCAUX is 2.9V.
V_{CCPLL}	1.2V	Power supply for PLL. Available on larger devices only.
$V_{CCIO0-7}$	1.2V to 3.3V	I/O power supply. There are eight general purpose I/O banks and each bank has its own supply V_{CCIO0} - V_{CCIO7}.
V_{CCIO8}	1.2V to 3.3V	Configuration I/O bank power supply. A nominal trip point for V_{CCIO8} is 1.0V.
V_{CCJ}	1.2V to 3.3V	JTAG power supply for the TAP controller port.

LatticeECP2M SERDES/PCS Power Supplies

When using the SERDES with 1.5V VCCIB or VCCOB, the SERDES should not be left in a steady state condition with the 1.5V power applied and the 1.2V power not applied. Both the 1.2V and the 1.5V power should be applied to the SERDES at nominally the same time. The normal variation in ramp_up times of power supplies and voltage regulators is not a concern.

All VCCTX, VCCRX, VCCP and VCCAUX33 supply pins must always be powered to the recommended operating voltage range regardless of the SERDES use. When SERDES channels are not used, the required supplies can be connected to the standard FPGA 1.2V or 3.3V power supplies since the noise levels of these supplies are not critical. VCCIB and VCCOB could be left floating for unused SERDES channels. Unused channel outputs are tristated, with approximately 10 KOhm internal resistor connecting between the differential output pair.

VCCAUX33 supplies power to termination resistors. As a result, noise on VCCAUX33 is directly coupled with the high-speed I/O, HDIN/HDOUT. A clean FPGA core VCCAUX (power supply) can be used to supply the SERDES VCCAUX33. Unused SERDES channels are configured in power-down mode by default.

Table 18-2 shows the power supplies and the appropriate voltage levels for each supply.

Table 18-2. Power Supply Description and Voltage Levels

Supply	Voltage (Typical)	Description
V_{CCTX}	1.2V	Transmit power supply
V_{CCRX}	1.2V	Receive power supply
V_{CCP}	1.2V	PLL and reference clock buffer power
V_{CCIB}	1.2V/1.5V	Input buffer power supply
V_{CCOB}	1.2V/1.5V	Output buffer power supply
$V_{CCAUX33}$	3.3V	Termination resistor switching power supply

Power Supply Sequencing

There are three main power supplies that are required to power-up the LatticeECP2/M device for proper operation: V_{CC}, V_{CCAUX} and V_{CCIO8}. There is no specific power sequencing requirement for the LatticeECP2/M device family. If the user's system has the option to design for power sequencing, a practical sequencing is V_{CC} before V_{CCAUX} or V_{CCIO8}. V_{CC} should reach its minimum voltage value before V_{CCAUX} and V_{CCIO8} reach their minimum values. For the LatticeECP2/M "S" version only, V_{CC} must reach its valid minimum value before powering up V_{CCAUX}. Power sequencing considerations should also consider that common supplies are generally tied together to the same rail. For example, If there is a 3.3V V_{CCIO}, it should be tied to the same supply as the 3.3V rail for V_{CCAUX}, thus minimizing leakage.

Power Supply Ramp

For the LatticeECP2/M, it is important to make sure that the power supply ramp times stay within a reasonable range. Each power supply must follow a monotonically clean ramp between the trip points and the minimum required supply voltage. Slow power supply ramps in the tens of milliseconds to hundreds of milliseconds are critical to ensure that the transitions around trip points are monotonic.

Multiple transitions through the trip point may cause multiple internal power-on reset sequencing.

Power Estimation

Once the LatticeECP2/M device density, package and logic implementation is decided, power estimation for the system environment should be determined based on the software Power Calculator provided as part of the isp-LEVER® design tool. When estimating power, the designer should keep two goals in mind:

1. Power supply budgeting should be based on the maximum of the power-up in-rush current, configuration current or maximum DC and AC current for the given system's environmental conditions.
2. The ability for the system environment and LatticeECP2/M device packaging to be able to support the specified maximum operating junction temperature.

By determining these two criteria, LatticeECP2/M power requirements are taken into consideration early in the design phase.

Configuration

All LatticeECP2/M devices contain two ports that can be used for device configuration. The Test Access Port (TAP), which supports bit-wide configuration, and the sysCONFIG™ port, which supports both byte-wide and serial configuration.

Table 18-3 shows the associated CFG pin definitions.

Table 18-3. Configuration Mode Selection

Configuration Mode	CF[2]	CFG[1]	CFG[0]	D[0]/SPIFASTN
SPI (Normal, 0x03)	0	0	0	Pull-Up
SPI (Fast, 0x0B)	0	0	0	Pull-Down
SPIm (Normal, 0x03)	0	1	0	Pull-Up
SPIm (Fast, 0x0B)	0	1	0	Pull-Down
Slave Serial	1	0	1	X
Slave Parallel	1	1	1	D0

The configuration port resides in I/O bank 8 and has dedicated/shared I/Os for configuration. Shared pins are available as a user I/O after configuration, if PERSISTENT is OFF.

V_{CCIO8} must match the supply voltage of the SPI Flash. For example, if the external SPI Flash operates at 3.3V, V_{CCIO8} must be tied to the 3.3V supply rail as well.

Table 18-4 lists the sysCONFIG pins. If any of these pins are used for configuration or user I/O, the designer must adhere to the requirements listed in TN1108, LatticeECP2/M sysCONFIG Usage Guide.

Table 18-4. Configuration Pin Descriptions

Pin Name	I/O Type	Pin Type	Description
CFG[2:0]	Input, weak pull-up	Dedicated	FPGA configuration mode selection
PROGRAMN	Input, weak pull-up	Dedicated	FPGA Configuration control and status signals
INITN	Bi-Directional Open Drain, weak pull-up	Dedicated	
DONE	Bi-Directional Open Drain with weak pull-up or Active Drive	Dedicated	
CCLK	Input or Output	Dedicated	Configuration clock
DI/CSSPI0N	Input, weak pull up	Dual-purpose	SPI control and data signals
DOUT/CSON	Output	Dual-purpose	
CSN	Input, weak pull up	Dual-purpose	
CS1N	Input, weak pull up	Dual-purpose	
WRITEN	Input, weak pull up	Dual-purpose	
BUSY/SISPI	Output, tri-state, weak pull-up	Dual-purpose	
D[0]/SPIFASTN	Input or Output, weak pull-up	Dual-purpose	
D[1:6]			
D[7]/SPID0			
TDI	Input, weak pull-up	Dedicated	JTAG
TDO	Output, weak pull-up		
TCK	Input with Hysteresis		
TMS	Input, weak pull-up		

JTAG Interface

The JTAG interface pins are referenced to V_{CCJ}. Typically, JTAG pins are referenced to a 3.3V supply. V_{CCJ} can support supplies from 1.2V to 3.3V. In cases where V_{CCJ} is connected to supplies other than 3.3V, validate that the JTAG interface cable or tester can support an I/O interface with the same I/O voltage standard.

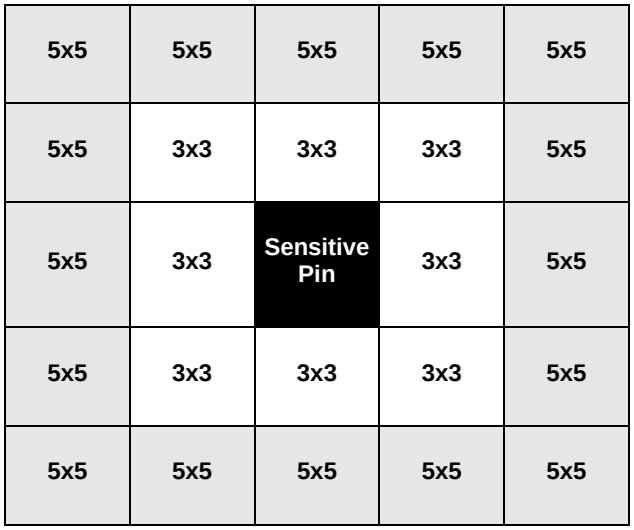
I/O Interface and Critical Pins

There are nine I/O banks on every LatticeECP2/M device. V_{CCIO8} is the configuration I/O bank and as such, the configuration requirements should have the highest priority to determine the supply voltage levels.

I/O Pin Assignments Around V_{CCPLL}

The V_{CCPLL} provides a “quiet” supply for the internal PLLs. For the best PLL jitter performance, careful pin assignment will keep “noisy” I/O pins away from “sensitive” pins, as shown in the BGA ball locations identified in Figure 18-1. In this case, the sensitive pin is one of the V_{CCPLL} supply pins. The noisy”I/O pins generally have the highest switching frequency, the highest V_{CCIO} standard and the fastest output slew rates. For example, using Figure 18-1, one can identify the “keep out” ball locations for potentially noisy signals.

Figure 18-1. “Quiet” Pin Assignment Considerations for BGA Packages



PLLCAP

An optional external capacitor can be used with both EXHPLLD and EPLLD to change the frequency response of the on-chip loop filter. When an external capacitor is used, it allows the PLLs to extend the low-end of their operating ranges. IPexpress™ checks the phase detector frequency to determine if an external capacitor is required. The allowable ranges for the PLL parameters with and without the external capacitor are described in the LatticeECP2/M Family Data Sheet.

Recommended optional external capacitor specifications:

- **Value:** 5.6 nF, +/- 20%
- **Type:** Ceramic chip capacitor, NPO dielectric
- **Package:** 1206 or smaller

Each device has two external capacitor pins, one for the left-side PLLs and one for the right-side PLLs. These pins are in fixed locations. They are dedicated function pins that are NOT shared with user I/Os. When an external capacitor pin is used by a PLL on one side of the device, it cannot be used by any other PLLs on the same side of the device. This means that a maximum of two PLLs per device, one on the left side and one on the right side, can have external capacitors attached.

Placing the capacitors at the PLLCAP pins only affects the PLL response when the software enables this feature. This allows a designer to provide the capacitors (or unpopulated PCB pads) to the PLLCAP pins to utilize the lower PLL frequencies if it becomes necessary for future changes to the design.

DDR/DDR2 Memory Interface Pin Assignments

The DDR Memory interface on the LatticeECP2/M device family is provided with a pre-engineered I/O register along with the precision I/O DLL timing control. There are two I/O DLLs specifically assigned to the two halves of the device. One I/O DLL supports I/O banks 2, 3 and 4; another I/O DLL supports I/O banks 5, 6 and 7.

In addition to the I/O DLL assignments, there are pre-defined data strobe (DQS) signals that can support a span of I/O pins as part of the memory data lanes. When assigning DDR memory interface I/O pins, the FPGA designer must insure that there are enough I/O pins to assign DDR memory data pins for each of the assigned DQS signals.

When interfacing to the DDR memory, the I/O type used is SSTL18 for DDR2 memory or SSTL25 for the DDR1 memory interface. The VREF required for these SSTL buffers should be assigned to VREF1 of the bank.

True-LVDS Output Pin Assignments

True-LVDS outputs are available on 50% of the I/O pins on the left and right sides of the device. The left- and right-side I/O banks are banks 2, 3, 6 and 7. When using the LVDS outputs, a 2.5V supply must be connected to these V_{CCIO} supply rails.

HSTL and SSTL Pin Assignments

These externally referenced I/O standards require an external reference voltage. Each of the LatticeECP2/M device family I/O banks allow up to two pre-defined V_{REF} pins. The V_{REF} pin(s) should get the highest priority when assigning pins.

PCI Clamp Pin Assignments

In LatticeECP2 devices, only the I/Os on the bottom banks have programmable PCI clamps. In LatticeECP2M devices, the I/Os on the left and bottom banks have the programmable PCI clamps. When the system design calls for a PCI clamp, those pins should be assigned to I/O banks 4 and 5 for LatticeECP2 and banks 4, 5, 6 and 7 for LatticeECP2M. For clamp characteristics, refer to the IBIS buffer models either on the Lattice website or ispLEVER design tool.

Checklist

	LatticeECP2/M Hardware Checklist Items	OK	N/A
1	**Power Supply**		
1.1	Core Supply VCC @1.2V		
1.2	Auxiliary Supply VCCAUX @3.3V		
1.3	PLL Supply VCCPLL @1.2V		
1.4	JTAG Supply VCCJ from 1.2V-3.3V		
1.5	I/O Supply VCCIO0-8 from 1.2V-3.3V		
1.6	10K +/-1% Pull down on XRES		
1.6	Supply Sequencing considerations		
1.7	Supply Ramp considerations		
1.8	Power Estimation		
1.9	Capacitor on PLLCAP pins (optional if using lower PLL frequencies)		
2	**Configuration**		
2.1	Consistency of VCCIO8 Supply when external SPI Flash is used		
2.2	Configuration control and status selections		
2.2.1	Pull-up or Pull-down on CFG2, CFG1, CFG0		
2.2.4	Pull-up on PROGRAMN, INITN, DONE		
2.2.5	Pull-up or pull-down on SPIFASTN (SPI mode)		
2.2.5	Pull-down on TCK		
2.2	JTAG Supply and default logic levels		

Checklist (Continued)

	LatticeECP2/M Hardware Checklist Items	OK	N/A
3	**I/O Pin Assignments**		
3.1	I/O pin assignments around VCCPLL		
3.2	DDR Memory pin assignment considerations		
3.3	True-LVDS pin assignment considerations		
3.4	HSTL and SSTL pin assignment considerations		
3.5	PCI clamp requirement considerations		
4	**LatticeECP2M SERDES**		
4.1	Transmitter power supply VCCTX@1.2V		
4.2	Receiver power supply VCCRX@1.2V		
4.3	TXPLL & reference clock buffer power VCCP@1.2V		
4.4	VCCIB & VCCOB (floating if Serdes are not used)@1.2V/1..5V		
4.5	Termination resistor switching power supply VCCAUX33@3.3V		

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
July 2007	01.0	Initial release.
September 2007	01.1	Updated Power Supply Sequencing text section.

LatticeECP3 and LatticeECP2M High-Speed Backplane Measurements

May 2010 Technical Note TN1149

Introduction

The LatticeECP3™ and LatticeECP2M™ families are low-cost FPGA product lines offering high-end features such as high-speed, embedded SERDES (SERializer/DESerializer) interfaces. These devices feature up to 16 SERDES channels with data rates of up to 3.125 Gbps.

This technical note outlines two experiments that measure the SERDES backplane transmission performance thresholds of the LatticeECP3 and LatticeECP2M devices. These experiments include:

- **Eye Diagram Experiment**: Uses eye diagrams to explore FPGA performance over a standard reference backplane.
- **Data Rate Experiment**: Uses bit error rate measurement techniques to verify FPGA backplane data rate limits at varying speeds and trace lengths.

Both experiments use a bit error rate tester (BERT) for pattern generation, a Tyco HM-Zd as the backplane and a LatticeECP3 or LatticeECP2M FPGA as the device under test. Both experiments collect data at 2.5 Gbps and 3.125 Gbps and use pre-emphasis adjustments to optimize transmitter performance.

Eye Diagram Experiment

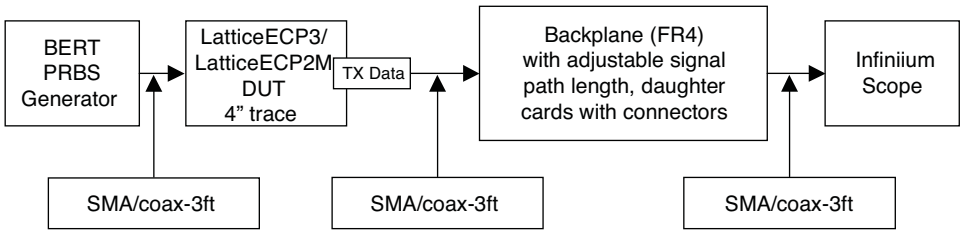
The eye diagram experiment checks the ability of the LatticeECP3 or LatticeECP2M FPGA to drive SERDES signals through a backplane at different pre-emphasis levels and data rates. It provides a visual, qualitative measurement of link performance by persistently sampling the signal at the receive end of the backplane.

In this configuration, a pseudo-random bit sequence (PRBS) pattern is generated and then sent into the LatticeECP3 or LatticeECP2M DUT. The PRBS is looped back and transmitted by the DUT into the backplane. The signal is routed out of the backplane, where it is sampled on an oscilloscope and eye diagrams are assembled. This experiment was performed at 2.5 Gbps and 3.125 Gbps with different levels of pre-emphasis. See Figure 19-1 for an illustration.

The equipment used in this test includes:

- Tyco HM-Zd Quad Route Test backplane with two daughter cards
- Agilent Infiniium DSO81304B 13 GHz oscilloscope
- High-quality coaxial cabling (rated for >3.125 Gbps) with SMA connectors
- Agilent 81250 3.7 GHz Parallel Bit Error Tester (ParBERT)
- Agilent 81130A function/pulse generator
- Agilent E3610A power supply
- Thermonics Thermostream for temperature control
- Internal LatticeECP3 and LatticeECP2M evaluation boards

Figure 19-1. Eye Diagram Experiment Setup



Backplane Specifications

- Tyco Electronics HM-Zd Quad Route Test backplane with daughter cards
- Backplane – 200 mils thick with 14 layers, made from Nalco 4000-FR4 material
- Signal layers 10 mil wide (1/2 copper thickness) designed for 100-ohm differential impedance traces
- Daughter cards – 93 mil thick with 14 layers
- Daughter cards – 6 mil wide, 100-ohm differential impedance traces
- Backplane trace length 40 inches

Test Setup Parameters

- DUT (LatticeECP3 or LatticeECP2M fpBGA)
- SERDES VCC supply values: 1.2V -5%
- Ambient temperature: 125°C
- Data pattern = PRBS (7-bit polynomial)
- Socketed, nominal device

Eye Diagram Measurements

Receiver end eye diagrams are an excellent measurement of expected link performance. This experiment tested a 40-inch length trace path, whereas most applications will have a 12 to 24 inch path. Eye diagrams were taken at 2.5 Gbps and 3.125 Gbps at varied pre-emphasis levels.

Pre-emphasis compensates for signal losses that occur with higher speeds and longer trace lengths. In general, increasing pre-emphasis will increase the signal quality for an improved eye diagram. The FPGA provides eight programmable pre-emphasis levels, available on a per-channel basis. To illustrate the progressive advantages of increased pre-emphasis, Tables 19-1 and 19-2 show eye diagrams taken at six of these pre-emphasis settings, each sampled at both 2.5 Gbps and 3.125 Gbps. Eye opening widths (measured in pS) and peak-to-peak voltage swings (in mV) are listed with each sample. For each of these measurements, larger is better.

Table 19-1. LatticeECP3 Eye Diagram Measurements

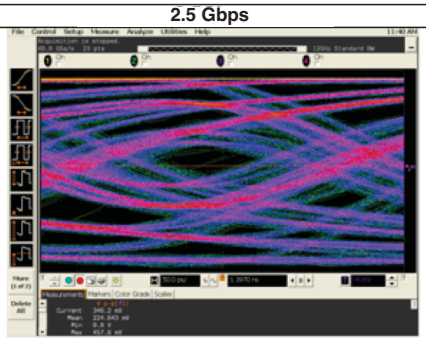
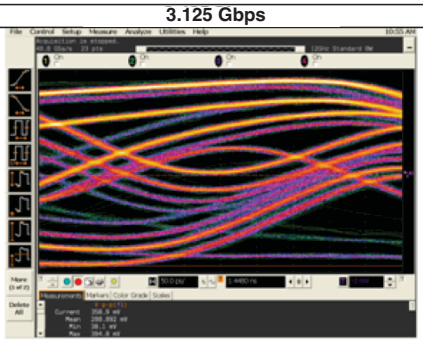
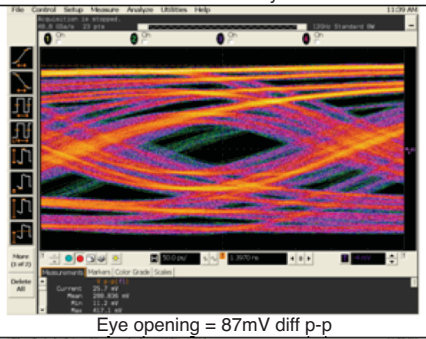
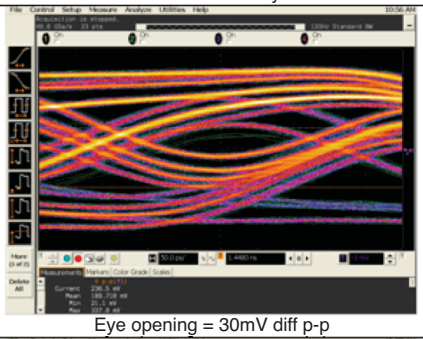
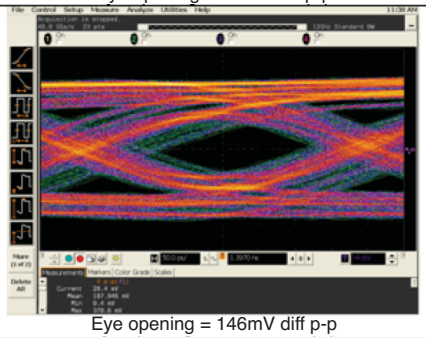
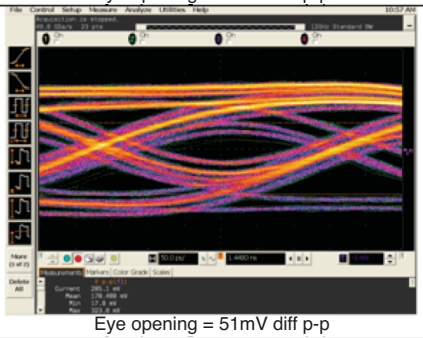
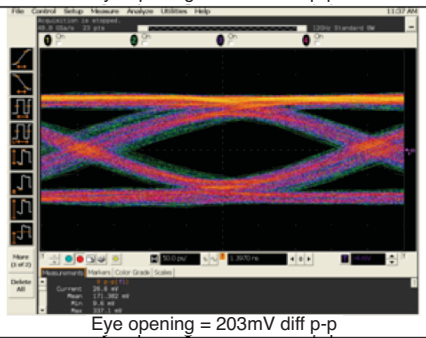
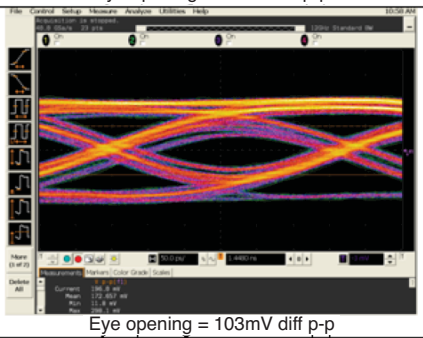
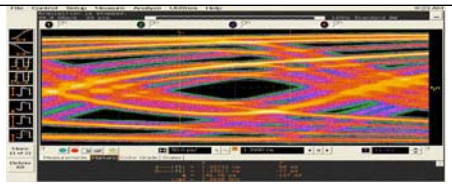
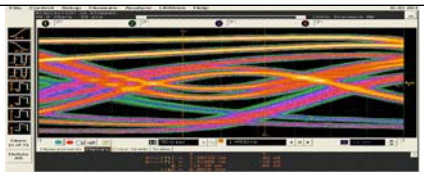
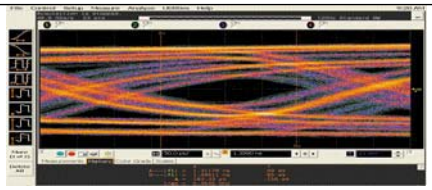
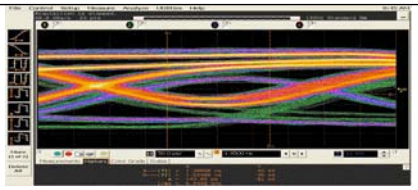
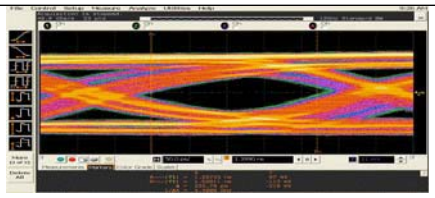
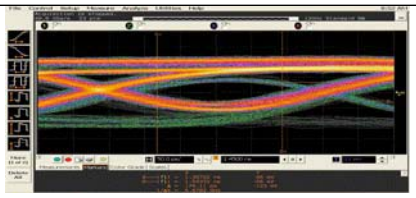
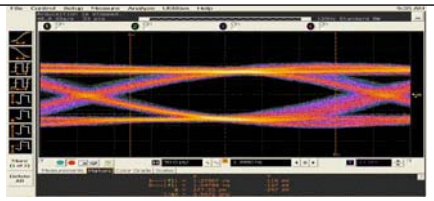
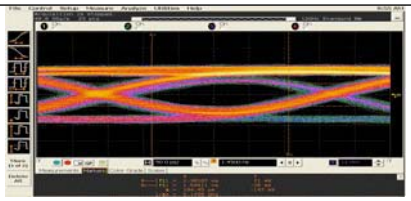
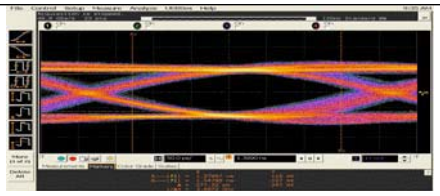
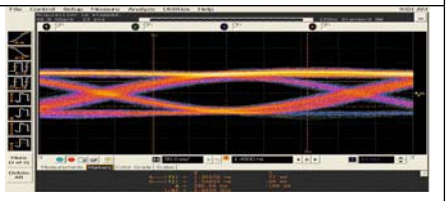
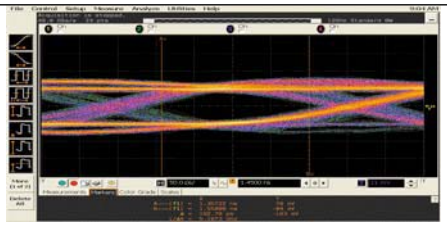
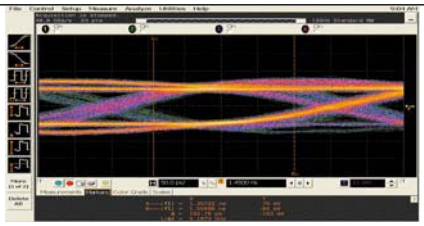
Pre-emphasis	2.5 Gbps	3.125 Gbps
Disabled 0%	 No detectable eye	 No detectable eye
1 (5%)	 Eye opening = 87mV diff p-p	 Eye opening = 30mV diff p-p
2 (12%)	 Eye opening = 146mV diff p-p	 Eye opening = 51mV diff p-p
3 (18%)	 Eye opening = 203mV diff p-p	 Eye opening = 103mV diff p-p

Table 19-2. LatticeECP2M Eye Diagram Measurements

Pre-Emphasis Setting	2.5 Gbps	3.125 Gbps
Pre-Emphasis Disabled	 Eye opening= 140pS, Diff. Amp. = 130mV pk-pk	 Eye opening= 110pS, Diff. Amp. = 45mV pk-pk
1 (16%)	 Eye opening= 180pS, Diff. Amp. = 155mV pk-pk	 Eye opening= 140pS, Diff. Amp. = 80mV pk-pk
2 (36%)	 Eye opening= 220pS, Diff. Amp. = 210mV pk-pk	 Eye opening= 176pS, Diff. Amp. = 123mV pk-pk
4 (44%)	 Eye opening= 265pS, Diff. Amp. = 240mV pk-pk	 Eye opening= 194pS, Diff. Amp. = 145mV pk-pk
5 (56%)	 Eye opening= 270pS, Diff. Amp. = 250mV pk-pk	 Eye opening= 210pS, Diff. Amp. = 165mV pk-pk
6 (80%)	 Eye opening= 270pS, Diff. Amp. = 210mV pk-pk	 Eye opening= 195pS, Diff. Amp. = 160mV pk-pk

Results and Conclusion

Tables 19-1 and 19-2 show that for both data rates the eye opening width and the pk-pk height were maximized at a pre-emphasis of 5. However, if the receiver sensitivity requirements are met, the user might choose a lower pre-emphasis setting in the interest of power savings and lower EM radiation. For example, the LatticeECP3 and LatticeECP2M SERDES receive ports have a minimum input differential sensitivity of 100 mV, so even with pre-emphasis disabled, this requirement would be met at 2.5 Gps (130 mV pk-pk).

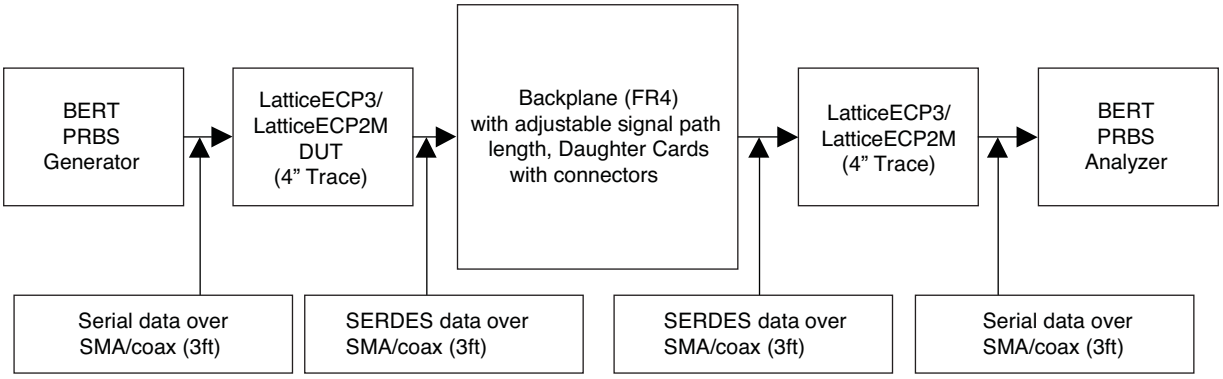
The eye diagrams in this experiment demonstrate that the LatticeECP3 and LatticeECP2M effectively provide high-quality signaling in a long trace, backplane design. These diagrams also show that pre-emphasis settings can be used to optimize SERDES performance.

Data Rate Experiment

The data rate experiment checks the FPGA SERDES transmission performance using a Bit Error Rate Tester (BERT) for expected/received data comparisons in a pass/fail context. The configuration begins by generating a PRBS sequence at a BERT, then sending the pattern into a LatticeECP3 or LatticeECP2M DUT. The FPGA loops the data back to the backplane as a SERDES bitstream. The SERDES data then exits the backplane and is received by the LatticeECP3 or LatticeECP2M (not necessarily the same device as the earlier DUT), which loops the data back to a BERT analyzer. Finally, the BERT compares the incoming bitstream to an expected data pattern and keeps track of how many mismatches are found. A typical expected bit error rate (BER) is less than 1x10-12 errors per second. See Figure 19-2 for an illustration.

The equipment used for this test was the same as used for the eye diagram experiment.

Figure 19-2. Data Rate Experiment



Backplane Specifications

- Backplane specifications are the same as for the eye diagram experiment
- Total trace length: 60 inches for 2.5 Gbps testing and 40 inches for 3.125 Gbps testing

Test Setup Parameters

- DUT (LatticeECP3 or LatticeECP2M fpBGA)
- Ambient temperature: 125°C
- SERDES VCC supply values: 1.2V -5%
- Socketed device
- All process variations
- Pre-emphasis setting: 4
- Equalization: 8 db

Data Rate Measurements

The data rate experiment was performed against samples from five process variations. Each sample was tested at 2.5 Gbps and 3.125 Gbps. The pass criterion was BER of less than 1x10-12 errors per second.

Results and Conclusions

For all process splits, the FPGA samples achieved a BER of better than 1x10-12 at 3.125 Gbps and 2.5 Gbps. This indicates that with a pre-emphasis setting of 4, the LatticeECP3 and LatticeECP2M can drive SERDES data error-free up to 40 inches of backplane at 3.125 Gbps and 40 inches of backplane with margin at 2.5 Gbps.

Conclusions and Design Guidelines

The experiments detailed in this technical note measured the ability of the LatticeECP3 and LatticeECP2M to reliably transmit SERDES data streams in a typical backplane design. The data-rate experiment used a statistical pass/fail scenario, whereas the eye-diagram experiment provided a visual, qualitative measurement. Both experiments concluded that the LatticeECP3 and LatticeECP2M deliver high-quality SERDES transmission over long backplane distances and over a broad range of data rates.

In both experiments, pre-emphasis was used to optimize LatticeECP3 and LatticeECP2M backplane performance. The eye diagram experiment went further to demonstrate that increased pre-emphasis provides a larger eye opening. Pre-emphasis settings are available on a per-channel basis.

Both experiments proved LatticeECP3 and LatticeECP2M performance at 2.5 Gbps and 3.125 Gbps. The data-rate experiment went further to show that devices from all process variations meet these performance goals.

Due to the number of factors in a high-speed PCB design, it is difficult to predict system performance in all scenarios. The data rate experiment illustrated robust performance at 3.125 Gbps over 40 inches in typical conditions, even for the slowest speed grades of the devices. The maximum trace length that can be implemented for an individual system environment may vary and should be evaluated by the individual designer.

The following suggestions will help designers optimize their high-speed LatticeECP3 and LatticeECP2M applications:

1. It is critical that all SERDES path cables and connectors be carefully selected. Cabling should be high-quality coaxial and connectors should be SMA. Both should be characterized for the intended frequency range. The designer should pay close attention to the parasitic performance of these devices.
2. Backplane and port cards should be implemented with good high-speed design practices in mind. For more details see TN1033, High-Speed PCB Design Considerations.
3. To improve the performance of receivers, use the LatticeECP3 or LatticeECP2M programmable equalization settings. For example, 8 db was used in the data rate experiment in this technical note.
4. Use analog circuit simulation tools to assess backplane performance signal integrity issues prior to building models. Contact Lattice for information about obtaining LatticeECP3 or LatticeECP2M SERDES HSPICE models for your simulations.
5. Early lab experimentation of long paths are recommended prior to full system model design in order to reduce technical risk. Eye diagram and bit error rate experiments are recommended.

References

- TN1118, LatticeSC High-Speed Backplane Measurements
- TN1176, LatticeECP3 SERDES/PCS Usage Guide
- TN1124, LatticeECP2M SERDES/PCS Usage Guide
- TN1033, High-Speed PCB Design Considerations
- TN1114, Electrical Recommendations for Lattice SERDES

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
February 2007	01.0	Initial release.
March 2007	01.1	Updated LatticeECP2M Eye-Diagram Measurements table.
April 2007	01.2	Updated Eye-Diagram Measurements section.
May 2010	01.3	Added LatticeECP3 FPGA data.

Section III. LatticeECP2/M Family Handbook Revision History

LatticeECP2/M Family Handbook Revision History

February 2012 **Handbook HB1003**

Revision History

Date	Handbook Revison Number	Change Summary
February 2006	01.0	Initial release.
August 2006	01.1	LatticeECP2 Family Data Sheet updated to version 01.1.
		Technical note TN1103 updated to version 01.1.
		Technical note TN1104 updated to version 01.1.
		Technical note TN1108 updated to version 01.2.
September 2006	02.0	LatticeECP2 Family Data Sheet updated to version 02.0.
		Technical note TN1102 updated to version 02.0.
		Technical note TN1103 updated to version 01.2.
		Technical note TN1104 updated to version 01.2.
		Technical note TN1105 updated to version 02.0.
		Technical note TN1106 updated to version 01.1.
		Technical note TN1107 updated to version 01.1.
		Technical note TN1108 updated to version 01.3.
		Added technical note TN1109.
		Added technical note TN1113.
		Added technical note TN1124.
December 2006	02.1	LatticeECP2/M Family Data Sheet updated to version 02.2.
		Technical note TN1106 updated to version 01.2.
		Technical note TN1103 updated to version 01.3.
February 2007	02.2	Technical note TN1103 updated to version 01.4.
		Technical note TN1106 updated to version 01.3.
		Added technical note TN1149.
February 2007	02.3	LatticeECP2/M Family Data Sheet updated to version 02.3.
March 2007	02.4	LatticeECP2/M Family Data Sheet updated to version 02.4.
		Technical note TN1105 updated to version 01.3.
		Technical note TN1108 updated to version 01.4.
March 2007	02.5	Technical note TN1124 updated to version 02.0.
		Technical note TN1149 updated to version 01.1.
March 2007	02.6	LatticeECP2/M Family Data Sheet updated to version 02.5.
		Technical note TN1108 updated to version 01.5.
		Technical note TN1109 updated to version 01.2.
March 2007	02.7	Technical note TN1124 updated to version 02.0.
April 2007	02.8	LatticeECP2/M Family Data Sheet updated to version 02.6.
		Technical note TN1102 updated to version 01.2.
		Technical note TN1104 updated to version 01.3.
		Technical note TN1108 updated to version 01.6.
		Technical note TN1113 updated to version 01.3.

Date	Handbook Revison Number	Change Summary
April 2007 (cont.)	02.8 (cont).	Technical note TN1149 updated to version 01.2.
July 2007	02.9	LatticeECP2/M Family Data Sheet updated to version 02.7.
		Technical note TN1102 updated to version 01.5.
		Technical note TN1103 updated to version 01.5.
		Technical note TN1105 updated to version 01.4.
		Technical note TN1108 updated to version 01.7.
August 2007	03.0	Technical note TN1124 updated to version 02.1.
August 2007	03.1	LatticeECP2/M Family Data Sheet updated to version 02.8.
		Technical note TN1108 updated to version 01.8.
		Technical note TN1109 updated to version 01.3.
September 2007	03.2	Technical note TN1124 updated to version 02.2.
		Technical note TN1108 updated to version 01.9.
September 2007	03.3	LatticeECP2/M Family Data Sheet updated to version 02.9.
		Technical note TN1113 updated to version 01.4.
December 2007	03.4	Technical note TN1105 updated to version 01.5.
		Technical note TN1124 updated to version 02.3.
February 2008	03.5	LatticeECP2/M Family Data Sheet updated to version 03.0.
		Technical note TN1104 updated to version 01.4.
		Technical note TN1108 updated to version 02.0.
		Technical note TN1113 updated to version 01.5.
		Technical note TN1124 updated to version 02.4.
April 2008	03.6	Technical note TN1102 updated to version 01.6.
		Technical note TN1104 updated to version 01.5.
April 2008	03.7	LatticeECP2/M Family Data Sheet updated to version 03.1.
June 2008	03.8	LatticeECP2/M Family Data Sheet updated to version 03.2.
		Technical note TN1124 updated to version 02.5.
		Technical note TN1104 updated to version 01.6.
July 2008	03.9	Technical note TN1113 updated to version 01.6.
September 2008	04.0	LatticeECP2/M Family Data Sheet updated to version 03.3.
		Technical note TN1124 updated to version 02.6.
		Technical note TN1103 updated to version 01.6.
		Technical note TN1104 updated to version 01.7.
November 2008	04.1	Technical note TN1104 updated to version 01.8.
		Technical note TN1108 updated to version 02.1.
		Technical note TN1124 updated to version 02.7.
		Added technical note TN1162, LatticeECP2/M Hardware Checklist.
January 2009	04.2	LatticeECP2/M Family Data Sheet updated to version 03.4.
		Technical note TN1124 updated to version 02.8.
March 2009	04.3	Technical note TN1102 updated to version 01.7.
		Technical note TN1104 updated to version 01.9.
		Technical note TN1107 updated to version 01.2.
		Technical note TN1113 updated to version 01.7.
		Technical note TN1124 updated to version 02.9.

Date	Handbook Revison Number	Change Summary
March 2010	04.4	LatticeECP2/M Family Data Sheet updated to version 03.5.
		Technical note TN1103 updated to version 01.9.
		Technical note TN1105 updated to version 01.7.
		Technical note TN1106 updated to version 01.4.
		Technical note TN1113 updated to version 01.8.
		Technical note TN1124 updated to version 03.3.
April 2010	04.5	LatticeECP2/M Family Data Sheet updated to version 03.6.
		Technical note TN1113 updated to version 01.9.
May 2010	04.6	Technical note TN1149 updated to version 01.3.
June 2010	04.7	Technical note TN1102 updated to version 01.8.
		Technical note TN1103 updated to version 02.1.
		Technical note TN1105 updated to version 01.7.
		Technical note TN1107 updated to version 01.3.
		Technical note TN1108 updated to version 02.2.
		Technical note TN1109 updated to version 01.4.
		Technical note TN1124 updated to version 03.4.
July 2010	04.8	LatticeECP2/M Family Data Sheet updated to version 03.7.
April 2011	04.9	LatticeECP2/M Family Data Sheet updated to version 03.8.
		Technical note TN1102 updated to version 01.9.
July 2011	05.0	Technical note TN1104 updated to version 02.0.
September 2011	05.1	Technical note TN1108 updated to version 02.3.
		Technical note TN1124 updated to version 03.5.
January 2012	05.2	Technical note TN1113 updated to version 02.0.
February 2012	05.3	LatticeECP2/M Family Data Sheet updated to version 03.9.

Note: For detailed revision changes, please refer to the revision history for each document.