MEC1632

Low Power 32-Bit Mobile Embedded Controller

Highlights

- 3.3V Operation
- ACPI Compliant
- LPC Interface
- VTR (standby) and VBAT Power Planes
  - Low Standby Current in Sleep Mode
- Configuration Register Set
  - Compatible with ISA Plug-and-Play Standard
  - EC-Programmable Base Address
- ARC-625D Embedded Controller (EC)
  - 16 KB Single Cycle 32-bit Wide Dual-ported SRAM, Accessible as Closely Coupled Data Memory and Instruction Memory
  - 4KB Boot ROM
  - 32 x 32 → 64 Fast Multiply
  - Divide Assist and Saturation Arithmetic
  - Maskable Interrupt Aggregator/Accelerator Interface
  - Maskable Hardware Wake-Up Events
  - Sleep mode
  - JTAG Debug Port, Includes JTAG Master
  - MCU Serial Debug Port
  - 1μS Delay Register
  - 10-Channel DMA Interface Supports SMBus Controllers and EC/Host GP-SPI Controllers
- Embedded Flash
  - 192 KB user space, 32-bit Access, 10 K Cycles Endurance
  - Flash Security Enhancements
    - –4K Boot Block Protection
    - –Direct JTAG and Direct LPC-protected (2) Pages at or Near Top of Memory for Password Protection
  - Multiple Flash Programming Options
    - –JTAG programmable
    - –BIOS programmable
    - –Programmable by EC at Power-on Using UART
    - –Programmable on a Gang Programmer via Gang-programmer Interface
- Embedded Non-volatile Read/Write Memory
  - 2 KB of EEPROM, Single Byte Access, 250K Cycles Endurance
  - 8-byte Block Erasable, 128 Blocks
  - Independent of main Flash memory
- Legacy Support
  - Fast GATEA20 & Fast CPU_RESET
- System to EC Message Interface
  - 8042 Style Host Interface
  - Embedded Memory Interface
    - Host Serial or Parallel IRQ Source
    - Provides Two Windows to On-Chip SRAM for Host Access
    - Two Register Mailbox Command Interface
    - Host Access of Virtual Registers Without EC Intervention
  - Mailbox Registers Interface
    - Thirty-two 8-Bit Scratch Registers
    - Two Register Mailbox Command Interface
    - Two Register SMI Source Interface
  - ACPI Embedded Controller Interface
    - Four Instances
    - 1 or 4 Byte Data transfer capable
    - Full-duplex Register Access
  - ACPI Power Management Interface
    - SCI Event-Generating Functions
- Battery Backed Resources
  - Power-Fail Status Register
  - 32 KHz Clock Generator
  - Week Alarm Timer Interface with Programmable Wake-up from 1ms to 45 Days
  - VBAT-Powered Control Interface
    - Six Wake-up Input Signals
    - Optional Latching of Wake-up Inputs
  - VBAT-Backed 64 Byte Memory
- Four EC-based SMBus 2.0 Host Controllers
  - Allows Master or Dual Slave Operation
  - Controllers are Fully Operational on Standby Power
  - DMA-driven I^2C Network Layer Hardware
  - I^2C Datalink Compatibility Mode
  - Multi-Master Capable
  - Supports Clock Stretching
  - Programmable Bus Speed up to 400KHz
  - Hardware Bus Access “Fairness” Interface
  - SMBus Time-outs Interface
  - AMD-TSI Port
  - 12 Ports Assignable to Any Controller
  - 3 SMBus Isolation Switches
    - Three Pairs of Ports Can Be Joined
- PECI Interface 3.0

© 2012-2018 Microchip Technology Inc.

- 18 x 8 Interrupt Capable Multiplexed Keyboard Scan Matrix
  - Optional Push-Pull Drive for Fast Signal Switching
- Three independent Hardware Driven PS/2 Ports
  - Fully functional on Main and/or Suspend Power
  - PS/2 Edge Wake Capable
- General Purpose I/O Pins
  - 142 GPIOs
  - 8 GPIO Pass-Through Port (GPTP)
  - Glitch protection on all GPIO pins
  - 6 Battery-powered General Purpose Outputs
- Low Power Programmable LED Interface
  - Supports three modes of operation:
    - Blinking Mode with Programmable Blink Rates
    - Breathing LED Output
    - 8-bit PWM
  - Breathing LED Supports Piecewise-linear Brightness Curves, Symmetric or Asymmetric
  - Supports Low Power Operation in Blinking and Breathing Modes
    - Operates on Standby Power
    - Operates in Chip's System Deepest Sleep State on 32kHz standby clock
    - Operational in EC Sleep State
  - Provides Three LED pins
    - LED pin buffers capable of sinking up to 20 mA
- Programmable 16-bit Counter/Timer Interface
  - Four Wake-capable 16-bit Auto-reloading Counter/Timer Instances
  - Four Operating Modes per Instance: Timer, One-shot, Event and Measurement
    - 4 External Inputs, 4 External Outputs
- Hibernation Timer Interface
  - Two 32.768 KHz Driven Timers
  - Programmable Wake-up from 0.5ms to 128 Minutes
- System Watch Dog Timer (WDT)
- Input Capture and Compare Timer
  - 32-bit Free-running timer
  - Six 32-bit Capture Registers
  - Two 32-bit Compare Registers
  - Capture, Compare and Overflow Interrupts
- BC-Link Interconnection Bus
  - Two High Speed and one Low Speed Bus Masters Controllers
- Two General Purpose Serial Peripheral Interface Controllers (ECGP-SPI)
  - One 3-pin EC-driven Full Duplex Serial Communication Interface
  - One 4-pin EC/Host-driven Full Duplex Serial Communication Interface to SPI Flash Interface
  - Flexible Clock Rates
  - SPI Burst Capable
- FAN Support
  - Six Programmable Pulse-Width Modulator (PWM) Outputs
    - Multiple Clock Rates
    - 16-Bit 'On' & 16-Bit 'Off' Counters
  - Six Fan Tachometer Inputs
  - 6 x 2 Capture/Compare Timer Interface
- ADC Interface
  - 10-bit Conversion in 10µs
  - 16 Channels
  - Integral Non-Linearity of ±0.5 LSB; Differential Non-Linearity of ±0.5 LSB
- 2-Pin Debug Port with Standard 16C550 Register Interface
  - Accessible from Host and EC
  - Programmable Input/output Pin Polarity Inversion
  - Programmable Main Power or Standby Power Functionality
- Port 80h Debug Ports for BIOS Debug
  - Two Ports, Assignable to Any LPC IO Address
  - 24-bit Timestamp with Adjustable Timebase
  - 16-Entry FIFO
- Resistor/Capacitor Identification Detection (RC_ID)
  - Single Pin Interface to External Inexpensive RC Circuit
  - Replacement for Multiple GPIO's
  - Provides 8 Quantized States on One Pin
- Integrated Standby Power Reset Generator
  - Reset Input Pin
  - Reset Output Pin
- HDMI Consumer Electronics Control (CEC) Bus Controller
- Clock Generator
  - 32.768KHz Clock Source
    - Low power 32KHz crystal oscillator
    - Optional use of a crystal-free silicon oscillator with ±2% Accuracy
    - Optional use of 32.768 KHz input Clock
    - Operational on Suspend Power
  - Programmable Clock Power Management Control & Distribution
  - 20.27 MHz silicon oscillator, ±2% Accuracy
- Real Time Clock
- Package
  - 169 Pin LFBGA RoHS Compliant package

© 2012-2018 Microchip Technology Inc.

Tool Requirements

For information on the latest version of the Metaware Development system, please see Application Note #26.14, “ARC Metaware Development System.”

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@microchip.com**. We welcome your feedback.

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000000A is version A of document DS30000000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip’s Worldwide Web site; http://www.microchip.com
- Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include -literature number) you are using.

Customer Notification System

Register on our web site at **www.microchip.com** to receive the most current information on all of our products.

© 2012-2018 Microchip Technology Inc.

Table of Contents

1.0 Change List 5
2.0 General Description 7
3.0 MEC1632 Pin Configuration 10
4.0 Bus Hierarchy 55
5.0 Logical Device Configuration 63
6.0 Host Interface 79
7.0 Power, Clocks, and Resets 95
8.0 Embedded Memory Interface 151
9.0 ACPI Embedded Controller Interface (ACPI-ECI) 170
10.0 8042 Emulated Keyboard Controller 190
11.0 ACPI PM1 Block Interface 205
12.0 MailBox Register Interface 213
13.0 Two Pin Serial Port (UART) 220
14.0 Embedded Flash Subsystem 240
15.0 EEPROM 266
16.0 ARC 625D Embedded Controller 280
17.0 EC Interrupt Aggregator 287
18.0 Watchdog Timer Interface 330
19.0 HDMI-CEC Interface Controller 337
20.0 16-Bit Timer Interface 351
21.0 Hibernation Timer 366
22.0 Week Alarm Interface 370
23.0 RTC With Date and DST Adjustment 375
24.0 GPIO Interface 388
25.0 Input Capture and Compare Timer 406
26.0 DMA Controller 419
27.0 SMB Device Interface 432
28.0 PECI Interface 437
29.0 Analog to Digital Converter 439
30.0 TACH Monitor 451
31.0 PWM Controller 460
32.0 RC Identification Detection (RC_ID) 466
33.0 General Purpose Serial Peripheral Interface (GP-SPI) 475
34.0 VBAT-Powered Control Interface 499
35.0 VBAT Powered RAM 508
36.0 Blinking/Breathing PWM 510
37.0 PS/2 Device Interface 530
38.0 Keyboard Matrix Scan Support 539
39.0 BC-Link Master 547
40.0 Port 80 BIOS Debug Port 555
41.0 Trace FIFO Debug Port (TFDP) 562
42.0 Boot ROM 567
43.0 Gang Programmer Interface 569
44.0 JTAG and XNOR 580
45.0 Electrical Specifications 602
46.0 Timing Diagrams 611
47.0 Reference Documents 631
Appendix A: Data Sheet Revision History 632
The Microchip Web Site 633
Customer Change Notification Service 633
Customer Support 633
Product Identification System 634

© 2012-2018 Microchip Technology Inc.

1.0 CHANGE LIST

This chapter lists the changes from the MEC1618.

1.1 New Package

- The 156 pin package is replaced by a 169 pin package.

1.2 Additional Clock Trees

- The number of EC clock trees is increased from 4 independent trees to 6 independent trees.
- See Table 7-12, “Master Clock Tree Block Allocation,” on page 109.

1.3 Additional SMBus Controller

- A fourth SMBus controller is added. It can be directed to any of the current 12 ports. See Table 27-2, “SMB Device Interface Base Address Table,” on page 433.
- Two additional DMA channels are added to support the additional SMBus controller. See Table 26-3, “DMA Device Selection,” on page 423, Table 26-4, “DMA Controller Base Address Table,” on page 426 and Section 26.8.1, "DMA Control Register," on page 427.

1.4 Additional GPIO pins

- There are 7 additional GPIO pins available.

1.5 Additional BGPO pins

- Five additional BGPO pins are added, for a total of 6. See the description in Section 22.8.1, "Week Timer Control Register," on page 372.

1.6 UART-based Flash Reprogram strap

- A strap option is added that will be checked at VTR POR. If asserted, the UART-based Flash reprogram function in the ROM will execute unconditionally, ignoring the contents of the Flash.

See Section 42.4, "UART Boot Strap Option," on page 568.

1.7 Increased LED Drive Strength

- The pins for the 3 blinking/breathing LEDs are changed from 8ma to 20ma pads.
- Optional configurable LED to LED skew control is added to the Blinking LED block, in order to minimize current drain when switching the higher-current signals.

1.8 EEPROM Enhancements

- Reading, writing and erasing blocks of the EEPROM will be independent of the main Flash array.
- The size of the EEPROM is increased to 2K Bytes.

The EEPROM is described Section 15.0, "EEPROM".

1.9 Crystal Oscillator and Real Time Clock

- A 32.768KHz crystal oscillator, with two crystal pins and a private power/ground pair of pins, can be used to source the internal 32KHz clock domain, in lieu of the silicon oscillator or an external pin.
- A standard battery-powered Real Time Clock is added. The RTC will not include CMOS memory. See Section 23.0, "RTC With Date and DST Adjustment," on page 375.

There are changes to the Clock Enable Register (See Section 7.9.2, "Clock Enable Register," on page 149).

A description of the Crystal Oscillator and the 32KHz clock domain switching can be found in Section 7.4.4, "32.768 KHz Crystal Oscillator," on page 101 and Section 7.4.6, "32 KHz Clock Domain Switching," on page 103.

© 2012-2018 Microchip Technology Inc.

1.10 LPC Bus Enhancements

- The LPC interface operates from 24MHz to 33MHz.
- An optional mode is added to the LPC interface that will reduce wait states when operating at 33MHz.

1.11 VCI modifications

- VCI inputs have programmable polarity. See
- Status latches on VCI pins capture rising edges and falling edges. See

1.12 Base Address Registers for LPC Memory Accesses

- The following logical devices on the MEC1632 can be accessed via LPC Memory cycles as well as LPC IO cycles:
  - EMI
  - ACPI EC interfaces
  - Embedded Flash controller
  - Mailbox interface
- The following logical devices can only be accessed via LP IO cycles:
  - 8042 Emulation port
  - Legacy GateA20
  - ACPI PM1
  - 16550 UART
  - General Purpose SPI
  - Port 80 Debug port
  - Real Time Clock

See Section 5.6.5, "Memory Base Address Registers," on page 70.

© 2012-2018 Microchip Technology Inc.

2.0 GENERAL DESCRIPTION

The MEC1632 is the mixed signal base component of a multi-device advanced I/O controller architecture. The MEC1632 incorporates a high-performance 32-bit ARC 625D embedded microcontroller with a 192 Kilobyte Embedded Flash Subsystem, 16 Kilobytes of SRAM, 1 Kilobyte EEPROM emulation, and a 2 Kilobyte EEPROM. The MEC1632 communicates with the system host using the Intel® Low Pin Count bus.

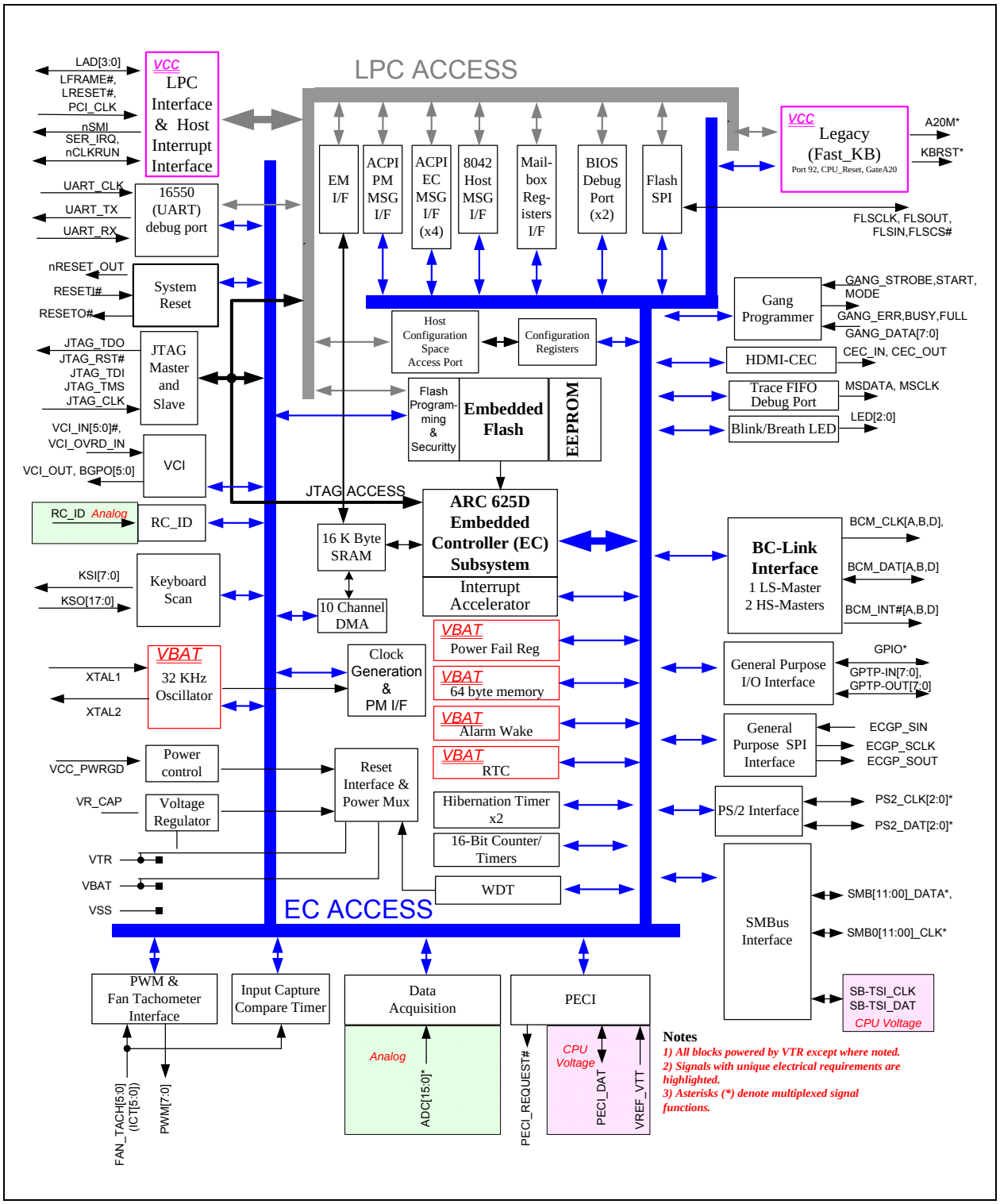
The MEC1632 is the EC Base Component of a split-architecture Advanced I/O Controller system which uses BC-Link communication protocol to access up to three companion components. The BC-Link protocol is peer-to-peer providing communication between the MEC1632 embedded controller and registers located in a companion.

The MEC1632 is directly powered by two separate suspend supply planes (VBAT and VTR) and senses a third runtime power plane (VCC) to provide “instant on” and system power management functions. The MEC1632 also contains an integrated VTR Reset Interface and a system Power Management Interface that supports low-power states and can drive state changes as a result of hardware wake events as defined by the MEC1632 Wake Interface.

The MEC1632 defines a software development system interface that includes an MCU Serial Debug Port, a two pin serial debug port with a 16C550A register interface that is accessible to the EC or to the LPC host and can operate up to 2 MB/s, a flexible Flash programming interface, a Port 80 BIOS Debug Port, Gang Programmer Interface, and a JTAG interface. The EC can also drive the JTAG interface as a master.

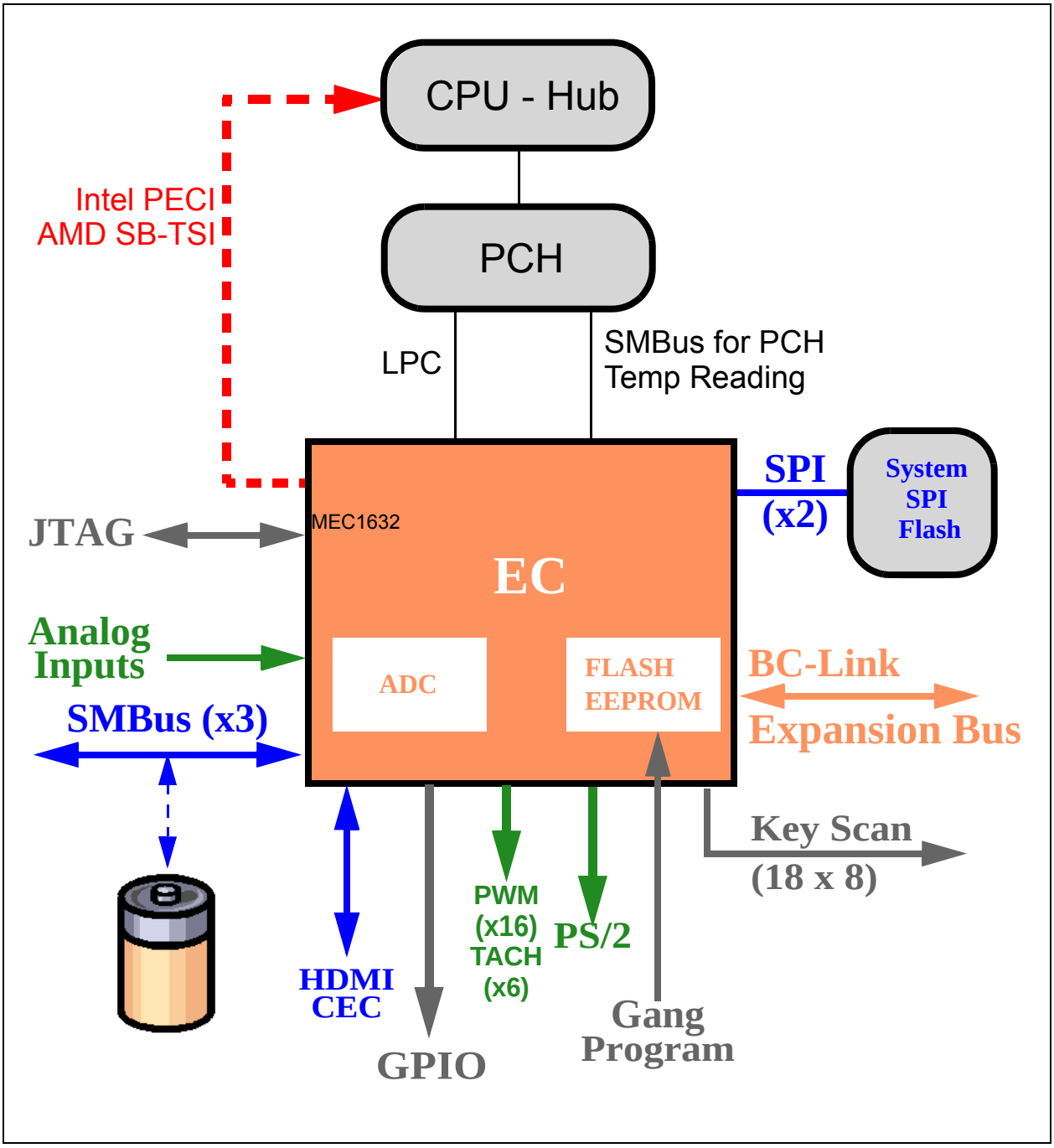
A top-level block diagram of the MEC1632 is shown below in Figure 2-1. An example of system level connection is shown in Figure 2-2. A detailed description of the Bus Hierarchy can be found in Section 4.0, "Bus Hierarchy," on page 55.

FIGURE 2-1: MEC1632 TOP-LEVEL BLOCK DIAGRAM



© 2012-2018 Microchip Technology Inc.

FIGURE 2-2: EXAMPLE OF MEC1632 CONNECTIONS TO SYSTEM COMPONENTS



3.0 MEC1632 PIN CONFIGURATION

3.1 Description

The MEC1632 Pin Configuration chapter includes a Pin List, General System/Layout Issues, Pin Description, Pin Multiplexing, Notes for Tables in this Chapter, Strapping Options and Package Outline.

3.2 Terminology and Symbols for Pins/Buffers

Term	Definition
#	The '#' sign at the end of a signal name indicates that this is an active-low signal
n	The lowercase 'n preceding a signal name indicates that this is an active-low signal
PWR	Power
I	Digital Input
IS	Input with Schmitt Trigger
I_AN	Analog Input
O	Push-Pull Output
OD	Open Drain Output
IO	Bi-directional pin
IOD	Bi-directional pin with Open Drain Output
PCI_I	Input. These pins meet the PCI 3.3V AC and DC Characteristics. (Note 3-1
PCI_O	Output. These pins meet the PCI 3.3V AC and DC Characteristics. (Note 3-1)
PCI_OD	Open Drain Output. These pins meet the PCI 3.3V AC and DC Characteristics. (Note 3-1)
PCI_IO	Input/Output These pins meet the PCI 3.3V AC and DC Characteristics. (Note 3-1)
PCI_ICLK	Clock Input. These pins meet the PCI 3.3V AC and DC Characteristics and timing. (Note 3-2)
SB-TSI	Input/Output (Open Drain), These pins are at the SB-TSI V_{REF} level. See , .
IO_PECI	PECI Input/Output. These pins are at the PECI V_{REF} level. See Section 45.0, "Electrical Specifications," on page 602.

Note 3-1 See the "PCI Local Bus Specification," Revision 2.1, Section 4.2.2.

Note 3-2 See the "PCI Local Bus Specification," Revision 2.1, Section 4.2.2 and 4.2.3.

© 2012-2018 Microchip Technology Inc.

3.3 Pin List

The MEC1632 Pin List is illustrated below in Table 3-1. The BGA package ball mapping to MEC1632 Pin Names is shown in Figure 3-1.

TABLE 3-1: MEC1632 PIN CONFIGURATION

Pin Ref. Number	Ball Number	Pin Name	Pin Ref. Number	Ball Number	Pin Name
1	D4	GPIO165/32KHZ_IN	43	N1	ADC5/GPIO205
2	A3	BGND	44	N2	ADC13/GPIO215
3	F4	VBAT	45	M3	ADC6/GPIO206
4	C1	BGPO0	46	M4	ADC14/GPIO216
5	B2	GPIO101/BGPO1	47	N3	ADC7/GPIO207
6	C4	GPIO102/BGPO2	48	N4	ADC15/GPIO217
7	E3	GPIO172/BGPO3	49	K4	VSS_ADC
8	B4	GPIO173/BGPO4	50	J3	VREF_ADC
9	E4	GPIO174/BGPO5	51	M5	LRESET#
10	D1	VCI_OUT	52	L5	CLKRUN#
11	C2	VCI_IN2#/GPIO161	53	L6	LFRAME#
12	G1	VCI_IN1#/GPIO162	54	N5	SER_IRQ
13	D2	VCI_IN0#/GPIO163	55	N6	PCI_CLK
14	E2	VCI_OVRD_IN/GPIO164	56	M6	LAD0
15	E1	VCI_IN3#/GPIO000	57	M7	LAD1
16	C3	VCI_IN4#/GPIO234	58	H8	VTR2
17	H2	VCI_IN5#/GPIO235	59	L7	LAD2
18	F1	RESETI#	60	K7	LAD3
19	F2	GPIO062/RESETO#/GANG_START	61	K6	GPIO100/nEC_SCI
20	F3	VCC_PWRGD/GANG_MODE	62	N7	GPIO011/nSMI
21	G2	GPIO106/nRESET_OUT	63	H6	GPIO061/LPCPD#
22	G4	VSS_RO	64	J6	GPIO050/FAN_TACH0
23	G6	VTR0	65	M8	GPIO222
24	H5	VSS0	66	N8	GPIO051/FAN_TACH1
25	G3	GPIO033/RC_ID/GANG_STROBE	67	H7	GPIO223
26	H1	GPIO021/KSI2/GANG_FULL	68	J7	GPIO052/FAN_TACH2
27	H3	VTR_REG	69	G7	GPIO224
28	H4	VR_CAP	70	L8	GPIO016/GPTP-IN7/FAN_TACH3
29	J5	GPIO060/KBRST/GANG_ERROR	71	J9	GPIO230/ECGP_SCLK
30	K5	GPIO166	72	M9	GPIO053/PWM0
31	J1	AVTR_ADC	73	K9	GPIO231/ECGP_SOUT
32	K3	ADC0/GPIO200	74	N9	GPIO054/PWM1
33	J2	ADC8/GPIO210	75	K8	GPIO233/ECGP_SIN
34	K1	ADC1/GPIO201	76	J8	GPIO055/PWM2
35	K2	ADC9/GPIO211	77	N10	GPIO056/PWM3
36	J4	VTR3	78	L9	GPIO001/PWM4
37	L2	ADC2/GPIO202	79	L10	GPIO002/PWM5
38	L1	ADC10/GPIO212	80	N11	GPIO014/GPTP-IN6/PWM6
39	M1	ADC3/GPIO203	81	M10	GPIO015/GPTP-OUT6/PWM7
40	L3	ADC11/GPIO213	82	N12	GPIO151/GPTP-IN3/FAN_TACH4/KSO15
41	L4	ADC4/GPIO204	83	M11	GPIO152/GPTP-OUT3/FAN_TACH5/KSO16
42	M2	ADC12/GPIO214	84	N13	GPIO017/GPTP-OUT7/TOUT3/KSI0

© 2012-2018 Microchip Technology Inc.

Pin Ref. Number	Ball Number	Pin Name	Pin Ref. Number	Ball Number	Pin Name
85	L11	GPIO040/GPTP-OUT2/TOUT2/KSO0	128	B11	GPIO044/VREF_VTT
86	M13	GPIO032/GPTP-IN2/TOUT1/KSI7	129	C11	GPIO145/SMB09_DATA/JTAG_TDI
87	M12	GPIO031/GPTP-OUT1/TOUT0/KSI6	130	A11	GPIO146/SMB09_CLK/JTAG_TDO
88	L12	GPIO012/SMB07_DATA	131	C10	GPIO147/SMB08_DATA/JTAG_CLK
89	K11	GPIO013/SMB07_CLK	132	B10	GPIO150/SMB08_CLK/JTAG_TMS
90	J10	GPIO130/SMB10_DATA	133	A10	JTAG_RST#
91	K10	GPIO131/SMB10_CLK	134	F9	GPIO104/UART_TX
92	L13	GPIO132/SMB06_DATA/KSO14	135	A9	GPIO105/UART_RX
93	K12	GPIO140/SMB06_CLK/PWM13/GANG_DATA7	136	E9	GPIO141/SMB05_DATA/PWM14/FLSCLK
94	K13	VTR_FLASH	137	B9	GPIO142/SMB05_CLK/PWM15/FLSOUT/GANG_DATA6
95	H9	GPIO025/UART_CLK/TIN0/nEM_INT	138	C9	GPIO143/SMB04_DATA/PWM12/FLSIN/GANG_BUSY
96	J11	GPIO026/GPTP-IN0/TIN1/KSI3/scan_test	139	A8	GPIO144/SMB04_CLK/FLSCS#/GANG_DATA5
97	J12	GPIO027/GPTP-OUT0/TIN2/KSI4/scan_test	140	D9	GPIO007/SMB03_DATA/PS2_CLK0B
98	J13	GPIO030/GPTP-IN1/TIN3/KSI5/scan_test	141	F8	GPIO010/SMB03_CLK/PS2_DAT0B/GANG_DATA4
99	H12	GPIO107/KSO4	142	E8	GPIO154/SMB02_DATA/PS2_CLK1B
100	H13	GPIO120/KSO7	143	B8	GPIO155/SMB02_CLK/PS2_DAT1B/GANG_DATA3
101	H10	GPIO124/GPTP-OUT4/KSO11	144	C8	GPIO006/SMB01_CLK
102	H11	GPIO125/GPTP-IN4/KSO12	145	D8	GPIO005/SMB01_DATA
103	G11	GPIO110/PS2_CLK2/GPTP-IN5	146	A7	GPIO004/SMB00_CLK
104	G12	GPIO111/PS2_DAT2/GPTP-OUT5/GANG_DATA2	147	E7	GPIO003/SMB00_DATA
105	G10	GPIO112/PS2_CLK1A/KSO5	148	D5	GPIO022/BCM_B_CLK
106	G9	GPIO113/PS2_DAT1A/KSO6/GANG_DATA1	149	C5	GPIO023/BCM_B_DAT
107	F11	GPIO114/PS2_CLK0A	150	E5	GPIO024/BCM_B_INT#
108	F12	GPIO115/PS2_DAT0A/GANG_DATA0	151	D7	GPIO127/A20M
109	G8	VTR1	152	F6	GPIO034/CEC_OUT
110	G5	VSS1	153	A6	GPIO036/CEC_IN
111	G13	GPIO070	154	B7	GPIO045/LSBCM_D_INT#/KSO1
112	F13	GPIO071	155	E6	GPIO046/LSBCM_D_DAT/KSO2
113	E13	GPIO072	156	B6	GPIO047/LSBCM_D_CLK/KSO3
114	D13	GPIO073	157	C6	GPIO121/BCM_A_INT#/KSO8
115	C13	GPIO074	158	C7	GPIO122/BCM_A_DAT/KSO9
116	B13	GPIO075	159	B5	GPIO123/BCM_A_CLK/KSO10
117	E12	GPIO041	160	A5	GPIO126/KSO13
118	E11	GPIO076	161	D6	GPIO020/KSI1
119	D11	GPIO220	162	A4	GPIO156/LED0
120	D12	GPIO035/PWM8	163	F5	GPIO157/LED1
121	E10	GPIO170/MSCLK	164	D3	GPIO153/LED2
122	D10	GPIO171/MSDATA	165	B3	GPIO175/32KHZ_OUT/KSO17
123	A13	GPIO133/PWM9	166	F7	VSS2
124	C12	GPIO134/PWM10	167	A1	XTAL1
125	F10	GPIO135/PWM11	168	B1	XTAL2
126	B12	GPIO042/PECI_DAT/SB-TSI_DAT	169	A2	VSS_XTAL
127	A12	GPIO043/SB-TSI_CLK			

© 2012-2018 Microchip Technology Inc.

FIGURE 3-1: MEC1632 PIN NAME TO 169-PIN LFBGA BALL MAPPING (TOP)

	1	2	3	4	5	6	7
A	XTAL1	VSS_XTAL	BGND	GPIO156/LED0	GPIO126/KSO13	GPIO036/CEC_IN	GPIO004/SMB00_CLK
B	XTAL2	GPIO101/BGPO1	GPIO175/32KHZ_OUT/KSO17	GPIO173/BGPO4	GPIO123/BCM_A_CLK/KSO10	GPIO047/LSBCM_D_CLK/KSO3	GPIO045/LSBCM_D_INT#/KSO1
C	BGPO0	VCI_IN2#/GPIO161	VCI_IN4#/GPIO234	GPIO102/BGPO2	GPIO023/BCM_B_DAT	GPIO121/BCM_A_INT#/KSO8	GPIO122/BCM_A_DAT/KSO9
D	VCI_OUT	VCI_IN0#/GPIO163	GPIO153/LED2	GPIO165/32KHZ_IN	GPIO022/BCM_B_CLK	GPIO020/KSI1	GPIO127/A20M
E	VCI_IN3#/GPIO000	VCI_OVRD_IN/GPIO164	GPIO172/BGPO3	GPIO174/BGPO5	GPIO024/BCM_B_INT#	GPIO046/LSBCM_D_DAT/KSO2	GPIO003/SMB00_DATA
F	RESETI#	GPIO062/RESETO#/GANG_START	VCC_PWRGD/GANG_MODE	VBAT	GPIO157/LED1	GPIO034/CEC_OUT	VSS2
G	VCI_IN1#/GPIO162	GPIO106/nRESET_OUT	GPIO033/RC_ID/GANG_STROBE	VSS_RO	VSS1	VTR0	GPIO224
H	GPIO021/KSI2/GANG_FULL	VCI_IN5#/GPIO235	VTR_REG	VR_CAP	VSS0	GPIO061/LPCPD#	GPIO223
J	AVTR_ADC	ADC8/GPIO210	VREF_ADC	VTR3	GPIO060/KBRST/GANG_ERROR	GPIO050/FAN_TACH0	GPIO052/FAN_TACH2
K	ADC1/GPIO201	ADC9/GPIO211	ADC0/GPIO200	VSS_ADC	GPIO166	GPIO100/nEC_SCI	LAD3
L	ADC10/GPIO212	ADC2/GPIO202	ADC11/GPIO213	ADC4/GPIO204	CLKRUN#	LFRAME#	LAD2
M	ADC3/GPIO203	ADC12/GPIO214	ADC6/GPIO206	ADC14/GPIO216	LRESET#	LAD0	LAD1
N	ADC5/GPIO205	ADC13/GPIO215	ADC7/GPIO207	ADC15/GPIO217	SER_IRQ	PCI_CLK	GPIO011/nSMI

THIS PINOUT IS FINAL - MAY BE USED FOR LAYOUT

8	9	10	11	12	13	
GPIO144/SMB04_CLK/FLSCS#/GANG_DATA5	GPIO105/UART_RX	JTAG_RST#	GPIO146/SMB09_CLK/JTAG_TDO	GPIO043/SB-TSI_CLK	GPIO133/PWM9	A
GPIO155/SMB02_CLK/PS2_DAT1B/GANG_DATA3	GPIO142/SMB05_CLK/PWM15/FLSOUT/GANG_DATA6	GPIO150/SMB08_CLK/JTAG_TMS	GPIO044/VREF_VTT	GPIO042/PECI_DAT/SB-TSI_DAT	GPIO075	B
GPIO006/SMB01_CLK	GPIO143/SMB04_DATA/PWM12/FLSIN/GANG_BUSY	GPIO147/SMB08_DATA/JTAG_CLK	GPIO145/SMB09_DATA/JTAG_TDI	GPIO134/PWM10	GPIO074	C
GPIO005/SMB01_DATA	GPIO007/SMB03_DATA/PS2_CLK0B	GPIO171/MSDATA	GPIO220	GPIO035/PWM8	GPIO073	D
GPIO154/SMB02_DATA/PS2_CLK1B	GPIO141/SMB05_DATA/PWM14/FLSCLK	GPIO170/MSCLK	GPIO076	GPIO041	GPIO072	E
GPIO010/SMB03_CLK/PS2_DAT0B/GANG_DATA4	GPIO104/UART_TX	GPIO135/PWM11	GPIO114/PS2_CLK0A	GPIO115/PS2_DAT0A/GANG_DATA0	GPIO071	F
VTR1	GPIO113/PS2_DAT1A/KSO6/GANG_DATA1	GPIO112/PS2_CLK1A/KSO5	GPIO110/PS2_CLK2/GPTP-IN5	GPIO111/PS2_DAT2/GPTP-OUT5/GANG_DATA2	GPIO070	G
VTR2	GPIO025/UART_CLK/TIN0/nEM_INT	GPIO124/GPTP-OUT4/KSO11	GPIO125/GPTP-IN4/KSO12	GPIO107/KSO4	GPIO120/KSO7	H
GPIO055/PWM2	GPIO230/ECGP_SCLK	GPIO130/SMB10_DATA	GPIO026/GPTP-IN0/TIN1/KSI3/scan_test	GPIO027/GPTP-OUT0/TIN2/KSI4/scan_test	GPIO030/GPTP-IN1/TIN3/KSI5/scan_test	J
GPIO233/ECGP_SIN	GPIO231/ECGP_SOUT	GPIO131/SMB10_CLK	GPIO013/SMB07_CLK	GPIO140/SMB06_CLK/PWM13/GANG_DATA7	VTR_FLASH	K
GPIO016/GPTP-IN7/FAN_TACH3	GPIO001/PWM4	GPIO002/PWM5	GPIO040/GPTP-OUT2/TOUT2/KSO0	GPIO012/SMB07_DATA	GPIO132/SMB06_DATA/KSO14	L
GPIO222	GPIO053/PWM0	GPIO015/GPTP-OUT6/PWM7	GPIO152/GPTP-OUT3/FAN_TACH5/KSO16	GPIO031/GPTP-OUT1/TOUT0/KSI6	GPIO032/GPTP-IN2/TOUT1/KSI7	M
GPIO051/FAN_TACH1	GPIO054/PWM1	GPIO056/PWM3	GPIO014/GPTP-IN6/PWM6	GPIO151/GPTP-IN3/FAN_TACH4/KSO15	GPIO017/GPTP-OUT7/TOUT3/KSI0	N

THIS PINOUT IS FINAL - MAY BE USED FOR LAYOUT

© 2012-2018 Microchip Technology Inc.

3.4 General System/Layout Issues

3.4.1 PIN DEFAULT STATE THROUGH POWER TRANSITIONS

The power state and power state transitions illustrated in Table 3-2 are defined in , . Pin behavior in this table assumes no specific programming to change the pin state. All GPIO default pins have the same behavior described in Table 3-2 as generically as GPIOXXX.

TABLE 3-2: Pin Default State Through Power Transitions

Signal	VBAT applied	VBAT STABLE	VTR applied	nSYS_RST de-asserted	VCC_PWRGD asserted	VCC_PWRGD de-asserted	nSYS_RST asserted	VTR un-powered	VBAT un-powered	Notes
GPIO042	unpowered	unpowered	low	*In*	*In*	*In*	*Z*	glitch	unpowered	
GPIO043	unpowered	unpowered	low	*In*	*In*	*In*	*Z*	glitch	unpowered	
GPIO062	unpowered	unpowered	low	*Out=0*	*Out*	*Out*	*Out*	glitch	unpowered	
GPIOXXX	unpowered	unpowered	Z	*In*	*In*	*In*	*Z*	glitch	unpowered	Note E
SER_IRQ	unpowered	unpowered	glitch	*In*	Z>I/O (*P*)>Z	*In*	*In*	glitch	unpowered	Note A
LRESET#	unpowered	unpowered	glitch	*In*	*In*	*In*	Z	glitch	unpowered	Note A
PCI_CLK	unpowered	unpowered	glitch	*In*	*In*	*In*	Z	glitch	unpowered	
LFRAME#	unpowered	unpowered	glitch	*In*	*In*	*In*	Z	glitch	unpowered	
LAD0	unpowered	unpowered	glitch	*In*	In>I/O (*P*)>In	*In*	Z	glitch	unpowered	
LAD1	unpowered	unpowered	glitch	*In*	In>I/O (*P*)>In	*In*	Z	glitch	unpowered	
LAD2	unpowered	unpowered	glitch	*In*	In>I/O (*P*)>In	*In*	Z	glitch	unpowered	
LAD3	unpowered	unpowered	glitch	*In*	In>I/O (*P*)>In	*In*	Z	glitch	unpowered	
CLKRUN#	unpowered	unpowered	glitch	*In*	Z>I/O (*P*)>Z	*In*	Z	glitch	unpowered	
BGPO0	Out=0	Out=0	*Retain*	*Retain*	*Retain*	*Retain*	*Retain*	*Retain*	unpowered	*Note B*
VCI_INx#	*In*	*In*	*In*	*In*	*In*	*In*	*In*	*In*	unpowered	
VCI_OUT	*Out logic*	*Out logic*	*Out logic*	*Out logic*	*Out logic*	*Out logic*	*Out logic*	*Out logic*	unpowered	*Note C*
VCI_OVRD_IN	*In*	*In*	*In*	*In*	*In*	*In*	*In*	*In*	unpowered	
XTAL1	Crystal In	Crystal In	Crystal In	Crystal In	Crystal In	Crystal In	Crystal In	Crystal In	Crystal In	
XTAL2	*Crystal Out*	*Crystal Out*	*Crystal Out*	*Crystal Out*	*Crystal Out*	*Crystal Out*	*Crystal Out*	*Crystal Out*	*Crystal Out*	

Legend	**Notes**
(P) = I/O state is driven by protocol while power is applied.	**Note A:** Pin exhibits "VCC" power domain emulation.
Z = Tristate	**Note B:** Pin is programmable by the EC and retains its value through a VTR power cycle.
	Note C: Pin is programmable by the EC and affected by other VBAT inputs pins.
	Note D: Pin exhibits "VTR" power domain emulation.
	Note E: Does not include GPIO042, GPIO043, and GPIO062

3.4.2 ALTERNATE FUNCTION PIN STATE THROUGH POWER TRANSITIONS

The power state and power state transitions illustrated in Table 3-3 are defined in Section 7.0, "Power, Clocks, and Resets". Pin behavior in this table assumes that the EC programs alternate function pin state (see Section 3.6, "Pin Multiplexing," on page 30).

TABLE 3-3: Alternate Function Pin State Through Power Transitions

Signal	VBAT applied	VBAT STABLE	VTR applied	nSYS_RST de-asserted Note E	VCC_ PWRGD asserted	VCC_ PWRGD de-asserted	nSYS_ RST asserted	VTR un-powered	VBAT un-powered	Notes
nSMI	N/A	N/A	N/A	N/A	1>OD(P)>1	*OD(1)*	*In*	glitch	N/A	
KBRST	N/A	N/A	N/A	N/A	1>OD(P)>1	*Z*	*Z>In*	glitch	N/A	Note F
A20M	N/A	N/A	N/A	N/A	1>OD(P)>1	*Z*	Z	glitch	N/A	Note F
LPCPD#	N/A	N/A	N/A	N/A	In	*Z*	Z	glitch	N/A	Note F

Legend	**Notes**
(P) = I/O state is driven by protocol while power is applied. **Z** = Tristate **OD** = Open Drain Output Undriven (1) or driven (0)	**Note E:** Transition occurs due to EC selecting alternate function. **Note F:** Pin is programmable by the EC and retains its value through a VTR power cycle.

© 2012-2018 Microchip Technology Inc.

3.4.3 NON 5 VOLT TOLERANT PINS

Table 3-4 lists all signal pins which are not 5.5 Volt tolerant; all other signal pins are 5 Volt tolerant. Signals in Table 3-4 refer to Pin Reference Numbers as defined in Table 3-1.

TABLE 3-4: NON 5 VOLT TOLERANT PINS

Pin Reference Number	Pin Name
32	ADC0/GPIO200
33	ADC8/GPIO210
34	ADC1/GPIO201
35	ADC9/GPIO211
37	ADC2/GPIO202
38	ADC10/GPIO212
39	ADC3/GPIO203
40	ADC11/GPIO213
41	ADC4/GPIO204
42	ADC12/GPIO214
43	ADC5/GPIO205
44	ADC13/GPIO215
45	ADC6/GPIO206
46	ADC14/GPIO216
47	ADC7/GPIO207
48	ADC15/GPIO217
51	LRESET#
52	CLKRUN#
53	LFRAME#
54	SER_IRQ
55	PCI_CLK
56	LAD0
57	LAD1
59	LAD2
60	LAD3
119	GPIO220
167	XTAL1
168	XTAL2

© 2012-2018 Microchip Technology Inc.

3.4.4 NON GLITCH PROTECTED PINS

Table 3-5 lists pins which do not have POR output glitch protection. POR output glitch protection guarantees that pins will have a steady-state output during VTR POR. Pins without POR output glitch protection may be susceptible to transitory changes as VTR power is applied. Signals in Table 3-5 refer to Pin Reference Numbers as defined in Table 3-1.

TABLE 3-5: NON GLITCH PROTECTED PINS

Pin Reference Number	Pin Name
1	GPIO165/32KHZ_IN
4	BGPO0
5	GPIO101/BGPO1
6	GPIO102/BGPO2
7	GPIO172/BGPO3
8	GPIO173/BGPO4
9	GPIO174/BGPO5
10	VCI_OUT
11	VCI_IN2#/GPIO161
12	VCI_IN1#/GPIO162
13	VCI_IN0#/GPIO163
14	VCI_OVRD_IN/GPIO164
15	VCI_IN3#/GPIO000
16	VCI_IN4#/GPIO234
17	VCI_IN5#/GPIO235
167	XTAL1
168	XTAL2

© 2012-2018 Microchip Technology Inc.

3.4.5 NON BACKDRIVE PROTECTED PINS

Table 3-6 lists pins which do not have backdrive protection. Signals in Table 3-6 refer to Pin Reference Numbers as defined in Table 3-1.

TABLE 3-6: NON BACKDRIVE PROTECTED PINS

Pin Reference Number	Pin Name
32	ADC0/GPIO200
33	ADC8/GPIO210
34	ADC1/GPIO201
35	ADC9/GPIO211
37	ADC2/GPIO202
38	ADC10/GPIO212
39	ADC3/GPIO203
40	ADC11/GPIO213
41	ADC4/GPIO204
42	ADC12/GPIO214
43	ADC5/GPIO205
44	ADC13/GPIO215
45	ADC6/GPIO206
46	ADC14/GPIO216
47	ADC7/GPIO207
48	ADC15/GPIO217
51	LRESET#
52	CLKRUN#
53	LFRAME#
54	SER_IRQ
55	PCI_CLK
56	LAD0
57	LAD1
59	LAD2
60	LAD3
119	GPIO220
167	XTAL1
168	XTAL2

© 2012-2018 Microchip Technology Inc.

3.5 Pin Description

3.5.1 OVERVIEW

The following tables describe the signal functions in the MEC1632 pin configuration. See Section 3.7, "Notes for Tables in this Chapter," on page 53 for notes that are referenced in the Pin Description tables.

3.5.2 HOST INTERFACE

TABLE 3-7: HOST INTERFACE

HOST INTERFACE			(13 Pins)
Pin Ref. Number	**Signal Name**	**Description**	**Notes**
54	SER_IRQ	Serial IRQ	Note 2, Note 8
51	LRESET#	LPC Reset. LRESET# is the same as the system PCI reset, PCIRST#	Note 8
55	PCI_CLK	PCI Clock	Note 8
53	LFRAME#	Frame signal. Indicates start of new cycle and termination of broken cycle	Note 8
56	LAD0	LPC Multiplexed command, address and data bus Bit 0.	Note 2
57	LAD1	LPC Multiplexed command, address and data bus Bit 1.	Note 2
59	LAD2	LPC Multiplexed command, address and data bus Bit 2.	Note 2
60	LAD3	LPC Multiplexed command, address and data bus Bit 3.	Note 2
52	CLKRUN#	PCI Clock Control	Note 8
61	nEC_SCI	Power Management Event	Note 1
63	LPCPD#	The LPC Bus Powerdown Signal.	
62	nSMI	SMI Output	
95	nEM_INT	EM Interface Interrupt Output	

3.5.3 BC-LINK INTERFACE

TABLE 3-8: BC-LINK INTERFACE

BC-Link Interface			(9 Pins)
Pin Ref. Number	**Signal Name**	**Description**	**Notes**
159	BCM_A_CLK	BC-Link Master clock	
158	BCM_A_DAT	BC-Link Master data I/O	Note 3
157	BCM_A_INT#	BC-Link Master interrupt	
148	BCM_B_CLK	BC-Link Master clock	
149	BCM_B_DAT	BC-Link Master data I/O	Note 3
150	BCM_B_INT#	BC-Link Master interrupt	
155	LSBCM_D_DAT	BC-Link Master data I/O	Note 3
156	LSBCM_D_CLK	BC-Link Master clock	
154	LSBCM_D_INT#	BC-Link Master interrupt	

© 2012-2018 Microchip Technology Inc.

Note 3-3 For ribbon cable applications, the Low Speed BC-Link Master maximum clock frequency is 3 MHz. The High Speed BC-Link Master maximum clock frequency is 20.27 MHz. The clock frequency is set with the BC Clock Select. register.

Note 3-4 the BCM DAT pins require a weak pull up resistor (100 K Ohms).

3.5.4 JTAG INTERFACE

TABLE 3-9: JTAG INTERFACE

JTAG Interface			(5 Pins)
Pin Ref. Number	**Signal Name**	**Description**	**Notes**
131	JTAG_CLK	JTAG Test Clock	
133	JTAG_RST#	JTAG Test Reset (active low)	Note 14
129	JTAG_TDI	JTAG Test Data In	
130	JTAG_TDO	JTAG Test Data Out	
132	JTAG_TMS	JTAG Test Mode Select	

3.5.5 MASTER CLOCK INTERFACE

TABLE 3-10: MASTER CLOCK INTERFACE

Master Clock Interface			(4 Pins)
Pin Ref. Number	**Signal Name**	**Description**	**Notes**
1	32KHZ_IN	32.768 KHz Digital Input	
165	32KHZ_OUT	32.768 KHz Digital Output	
167	XTAL1	32.768 KHz Crystal Input	
168	XTAL2	32.768 KHz Crystal Output (single-ended 32.768 kHz clock input)	

3.5.6 ANALOG DATA ACQUISITION INTERFACE

TABLE 3-11: ANALOG DATA ACQUISITION

Analog Data Acquisition Interface			(17 Pins)
Pin Ref. Number	**Signal Name**	**Description**	**Notes**
32	ADC0	ADC channel 1	
34	ADC1	ADC channel 2	
37	ADC2	ADC channel 3	
39	ADC3	ADC channel 4	
41	ADC4	ADC channel 5	
43	ADC5	ADC channel 6	
45	ADC6	ADC channel 7	
47	ADC7	ADC channel 8	
33	ADC8	ADC channel 9	
35	ADC9	ADC channel 10	
38	ADC10	ADC channel 11	
40	ADC11	ADC channel 12	
42	ADC12	ADC channel 13	
44	ADC13	ADC channel 14	
46	ADC14	ADC channel 15	
48	ADC15	ADC channel 16	
50	VREF_ADC	ADC Voltage Reference Pin	

Note: The voltage on the pins in Table 3-11 must not exceed 3.6 V or damage to the device will occur.

© 2012-2018 Microchip Technology Inc.

3.5.7 FAN TACHOMETER, PWM AND INPUT CAPTURE TIMER INTERFACE

TABLE 3-12: FAN TACHOMETER AND PWM INTERFACE

FAN PWM & TACHOMETER			(22 Pins)
Pin Ref. Number	**Signal Name**	**Description**	**Notes**
64	FAN_TACH0	Fan Tachometer Input 1/Input Capture Timer Input 0	
66	FAN_TACH1	Fan Tachometer Input 2/Input Capture Timer Input 1	
68	FAN_TACH2	Fan Tachometer Input 3/Input Capture Timer Input 2	
70	FAN_TACH3	Fan Tachometer Input 4/Input Capture Timer Input 3	
82	FAN_TACH4	Fan Tachometer Input 5/Input Capture Timer Input 4	
83	FAN_TACH5	Fan Tachometer Input 6/Input Capture Timer Input 5	
72	PWM0	Pulse Width Modulator Output 0	
74	PWM1	Pulse Width Modulator Output 1	
76	PWM2	Pulse Width Modulator Output 2	
77	PWM3	Pulse Width Modulator Output 3	
78	PWM4	Pulse Width Modulator Output 4	
79	PWM5	Pulse Width Modulator Output 5	
80	PWM6	Pulse Width Modulator Output 6	
81	PWM7	Pulse Width Modulator Output 7	
120	PWM8	Pulse Width Modulator Output 8	
123	PWM9	Pulse Width Modulator Output 9	
124	PWM10	Pulse Width Modulator Output 10	
125	PWM11	Pulse Width Modulator Output 11	
138	PWM12	Pulse Width Modulator Output 12	
93	PWM13	Pulse Width Modulator Output 13	
136	PWM14	Pulse Width Modulator Output 14	
137	PWM15	Pulse Width Modulator Output 15	

3.5.8 GENERAL PURPOSE I/O INTERFACE

TABLE 3-13: GPIO INTERFACE

GPIO Interface			(142 Pins)
Pin Ref. Number	**Signal Name**	**Description**	**Notes**
See Pin Configuration Table	GPIO	General Purpose Input Output Pins	

3.5.9 GENERAL PURPOSE PASS-THROUGH PORTS INTERFACE

TABLE 3-14: GPIO PASS-THROUGH PORTS

General Purpose Pass-Through Ports			(16 Pins)
Pin Ref. Number	**Signal Name**	**Description**	**Notes**
96	GPTP-IN0	General Purpose Pass Through Port Input 0	
98	GPTP-IN1	General Purpose Pass Through Port Input 1	
86	GPTP-IN2	General Purpose Pass Through Port Input 2	
82	GPTP-IN3	General Purpose Pass Through Port Input 3	
102	GPTP-IN4	General Purpose Pass Through Port Input 4	
103	GPTP-IN5	General Purpose Pass Through Port Input 5	
80	GPTP-IN6	General Purpose Pass Through Port Input 6	
70	GPTP-IN7	General Purpose Pass Through Port Input 7	
97	GPTP-OUT0	General Purpose Pass Through Port Output 0	
87	GPTP-OUT1	General Purpose Pass Through Port Output 1	
85	GPTP-OUT2	General Purpose Pass Through Port Output 2	
83	GPTP-OUT3	General Purpose Pass Through Port Output 3	
101	GPTP-OUT4	General Purpose Pass Through Port Output 4	
104	GPTP-OUT5	General Purpose Pass Through Port Output 5	
81	GPTP-OUT6	General Purpose Pass Through Port Output 6	
84	GPTP-OUT7	General Purpose Pass Through Port Output 7	

© 2012-2018 Microchip Technology Inc.

3.5.10 MISCELLANEOUS FUNCTIONS

TABLE 3-15: MISCELLANEOUS FUNCTIONS

MISC Functions			(17 Pins)
Pin Ref. Number	**Signal Name**	**Description**	**Notes**
151	A20M	KBD GATEA20 Output	
29	KBRST	CPU_RESET	
162	LED0	LED Output 1	Note 9
163	LED1	LED Output 2	Note 9
164	LED2	LED Output 3	Note 9
121	MSCLK	Microchip Proprietary EC debug port	
122	MSDATA	Microchip Proprietary EC debug port	
95	UART_CLK	UART CLK input	
135	UART_RX	UART RX Input	Note 4
134	UART_TX	UART TX Output	Note 4
20	VCC_PWRGD	System Main Power Indication	Note 8
25	RC_ID	RC Identification Detection	
18	RESETI#	System Reset Input	
19	RESETO#	System Reset Output	Note 7
153	CEC_IN	HDMI-CEC control bus input	
152	CEC_OUT	HDMI-CEC control bus output	
21	nRESET_OUT	EC-driven External System Reset	

Note 3-5 The KBRST pin function is the output of CPU_RESET described in Section 10.14, "CPU_RESET Hardware Speed-Up," on page 203.

Note 3-6 When the CLK_SRC bit is '1' in the UARTConfiguration Select register (See page 238), the baud clock is externally sourced from the UART_CLK pin. UART_CLK requires a frequency of 1.8432 MHz ± 2%.

Note 3-7 The nRESET_OUT pin function is an external output signal version of the internal signal nSIO_RESET. See the iRESET OUT bit in the Block Sleep Enable Registers on page 136.

3.5.11 PS/2 INTERFACE

TABLE 3-16: PS/2 INTERFACE

PS/2 Interface			(10 Pins)
Pin Ref. Number	**Signal Name**	**Description**	**Notes**
140	PS2_CLK0B	PS/2 clock ouput	Note 6
141	PS2_DAT0B	PS/2 data	Note 6
103	PS2_CLK2	PS/2 clock ouput	
104	PS2_DAT2	PS/2 data	
105	PS2_CLK1A	PS/2 clock ouput	Note 6
106	PS2_DAT1A	PS/2 data	Note 6
107	PS2_CLK0A	PS/2 clock ouput	Note 6
108	PS2_DAT0A	PS/2 data	Note 6
142	PS2_CLK1B	PS/2 clock ouput	Note 6
143	PS2_DAT1B	PS/2 data	Note 6

3.5.12 POWER INTERFACE

TABLE 3-17: POWER INTERFACE

Pin Ref. Number	Signal Name	Description	Notes
2	BGND	VBAT associated ground	
3	VBAT	VBAT supply	
28	VR_CAP	Internal Voltage Regulator Output (Capacitor Required)	
24	VSS0	VTR associated ground 0	
110	VSS1	VTR associated ground 1	
166	VSS2	VTR associated ground 2	
22	VSS_RO	VTR associated ground used for ring ocsillator.	
23	VTR0	VTR supply 0	
109	VTR1	VTR supply 1	
58	VTR2	VTR supply 2	
36	VTR3	VTR supply 3	
49	VSS_ADC	Analog ADC VTR associated ground	
27	VTR_REG	VTR Internal Voltage Regulator Supply	
94	VTR_FLASH	VTR Internal Flash Supply	
31	AVTR_ADC	Analog ADC VTR associated Supply	
169	VSS_XTAL	Ground associated with XTAL pins	

APPLICATION NOTE: VBAT to VTR switching must be done externally as described in Section 7.7.1, "Power Mux," on page 129.

© 2012-2018 Microchip Technology Inc.

3.5.13 SMBUS INTERFACE

TABLE 3-18: SMBUS INTERFACE

SMBus Interface			(24 Pins)
Pin Ref. Number	**Signal Name**	**Description**	**Notes**
146	SMB00_CLK	SMBus Controller Port 0 Clock	
147	SMB00_DATA	SMBus Controller Port 0 Data	
144	SMB01_CLK	SMBus Controller Port 1 Clock	
145	SMB01_DATA	SMBus Controller Port 1 Data	
143	SMB02_CLK	SMBus Controller Port 2 Clock	
142	SMB02_DATA	SMBus Controller Port 2 Data	
141	SMB03_CLK	SMBus Controller Port 3 Clock	
140	SMB03_DATA	SMBus Controller Port 3 Data	
139	SMB04_CLK	SMBus Controller Port 4 Clock	
138	SMB04_DATA	SMBus Controller Port 4 Data	
137	SMB05_CLK	SMBus Controller Port 5 Clock	
136	SMB05_DATA	SMBus Controller Port 5 Data	
93	SMB06_CLK	SMBus Controller Port 6 Clock	
92	SMB06_DATA	SMBus Controller Port 6 Data	
89	SMB07_CLK	SMBus Controller Port 7 Clock	
88	SMB07_DATA	SMBus Controller Port 7 Data	
132	SMB08_CLK	SMBus Controller Port 8 Clock	
131	SMB08_DATA	SMBus Controller Port 8 Data	
130	SMB09_CLK	SMBus Controller Port 9 Clock	
129	SMB09_DATA	SMBus Controller Port 9 Data	
91	SMB10_CLK	SMBus Controller Port 10 Clock	
90	SMB10_DATA	SMBus Controller Port 10 Data	
127	SB-TSI_CLK	SMBus Controller AMD-TSI Port Clock	
126	SB-TSI_DAT	SMBus Controller AMD-TSI Port Data	

3.5.14 PECI INTERFACE

TABLE 3-19: PECI INTERFACE

PECI Interface			(2 Pins)
Pin Ref. Number	**Signal Name**	**Description**	**Notes**
126	PECI_DAT	PECI Bus	
128	VREF_VTT	Processor Interface Voltage Reference	

3.5.15 16-BIT COUNTER/TIMER INTERFACE

TABLE 3-20: 16-BIT COUNTER/TIMER INTERFACE

16-Bit Counter/Timer Interface			(8 Pins)
Pin Ref. Number	**Signal Name**	**Description**	**Notes**
95	TIN0	16-Bit Counter/Timer Input 1	
96	TIN1	16-Bit Counter/Timer Input 2	
97	TIN2	16-Bit Counter/Timer Input 3	
98	TIN3	16-Bit Counter/Timer Input 4	
87	TOUT0	16-Bit Counter/Timer Output 1	
86	TOUT1	16-Bit Counter/Timer Output 2	
85	TOUT2	16-Bit Counter/Timer Output 3	
84	TOUT3	16-Bit Counter/Timer Output 4	

3.5.16 KEYBOARD SCAN INTERFACE

TABLE 3-21: KEYBOARD SCAN INTERFACE

Keyboard Scan Interface			(26 Pins)
Pin Ref. Number	**Signal Name**	**Description**	**Notes**
84	KSI0	Keyboard Scan Matrix Input 0	
161	KSI1	Keyboard Scan Matrix Input 1	
26	KSI2	Keyboard Scan Matrix Input 2	
96	KSI3	Keyboard Scan Matrix Input 3	
97	KSI4	Keyboard Scan Matrix Input 4	
98	KSI5	Keyboard Scan Matrix Input 5	
87	KSI6	Keyboard Scan Matrix Input 6	
86	KSI7	Keyboard Scan Matrix Input 7	
85	KSO0	Keyboard Scan Matrix Output 0	
154	KSO1	Keyboard Scan Matrix Output 1	
155	KSO2	Keyboard Scan Matrix Output 2	
156	KSO3	Keyboard Scan Matrix Output 3	
99	KSO4	Keyboard Scan Matrix Output 4	
105	KSO5	Keyboard Scan Matrix Output 5	
106	KSO6	Keyboard Scan Matrix Output 6	
100	KSO7	Keyboard Scan Matrix Output 7	
157	KSO8	Keyboard Scan Matrix Output 8	
158	KSO9	Keyboard Scan Matrix Output 9	
159	KSO10	Keyboard Scan Matrix Output 10	
101	KSO11	Keyboard Scan Matrix Output 11	
102	KSO12	Keyboard Scan Matrix Output 12	
160	KSO13	Keyboard Scan Matrix Output 13	
92	KSO14	Keyboard Scan Matrix Output 14	
82	KSO15	Keyboard Scan Matrix Output 15	
83	KSO16	Keyboard Scan Matrix Output 16	
165	KSO17	Keyboard Scan Matrix Output 17	

© 2012-2018 Microchip Technology Inc.

3.5.17 VCI

TABLE 3-22: VCI INTERFACE

VBAT-Powered Control Interface			(14 Pins)
Pin Ref. Number	**Signal Name**	**Description**	**Notes**
14	VCI_OVRD_IN	Input can cause wakeup or interrupt event	
10	VCI_OUT	OUTPUT from combinational logic and/or EC	
4	BGPO0	VBAT driven GPO	
5	BGPO1	VBAT driven GPO	
6	BGPO2	VBAT driven GPO	
7	BGPO3	VBAT driven GPO	
8	BGPO4	VBAT driven GPO	
9	BGPO5	VBAT driven GPO	
13	VCI_IN0#	Input can cause wakeup or interrupt event	
12	VCI_IN1#	Input can cause wakeup or interrupt event	
11	VCI_IN2#	Input can cause wakeup or interrupt event	
15	VCI_IN3#	Input can cause wakeup or interrupt event	
16	VCI_IN4#	Input can cause wakeup or interrupt event	
17	VCI_IN5#	Input can cause wakeup or interrupt event	

The signals BGP1 through BGPO5 are activated on their pins by control bits in the Week Timer Control Register, rather than the mux control field in the associated GPIO mux control register. When a pin is selected as a BGPOx output, its buffer type is O-8ma, independent of the buffer type of the GPIO.

The following table defines pin multiplexing for the BGPOx outputs:

TABLE 3-23: BGPOX PIN MULTIPLEXING

Pin Ref. Number	Signal Name	Buffer Type	Associated GPIO	Control Bit in Week Time Control Register
5	BGPO1	O-8mA	GPIO101	D16
6	BGPO2	O-8mA	GPIO102	D17
7	BGPO3	O-8mA	GPIO172	D18
8	BGPO4	O-8mA	GPIO173	D10
9	BGPO5	O-8mA	GPIO174	D20

3.5.18 SPI CONTROLLERS INTERFACE

TABLE 3-24: SPI CONTROLLERS INTERFACE

SPI Controllers Interface			(7 Pins)
Pin Ref. Number	**Signal Name**	**Description**	**Notes**
71	ECGP_SCLK	General Purpose SPI Clock	
73	ECGP_SOUT	General Purpose SPI Output	
75	ECGP_SIN	General Purpose SPI Input	
136	FLSCLK	Flash Interface SPI Clock	
137	FLSOUT	Flash Interface SPI Output	
138	FLSIN	Flash Interface SPI Input	
139	FLSCS#	Flash Interface SPI Chip Select	

Note 3-8 For General Purpose Serial Peripheral Interface (GP-SPI) interface pins with 8 mA buffers the maximum SPCLK pin clock frequency is 16.13 MHz for all modes. Limited functionality is available

at higher frequencies, but performance is not guaranteed (see Table 33-14, “SPI_CLK Frequencies,” on page 493 and Section 33.9.5.5, "Limits of SPI configurations," on page 487).

3.5.19 GANG PROGRAMMING INTERFACE

TABLE 3-25: GANG PROGRAMMING INTERFACE

Gang Programmer Interface			(14 Pins)
Pin Ref. Number	**Signal Name**	**Description**	**Notes**
20	GANG_MODE	Gang Programmer operating mode control	
19	GANG_START	Gang Programmer start control	
25	GANG_STROBE	Gang Programmer data latch strobe	
26	GANG_FULL	Gang Programmer data flow control	
138	GANG_BUSY	Gang Programmer operational status	
29	GANG_ERROR	Gang Programmer error status	
108	GANG_DATA0	Gang Programmer Flash program data 0	
106	GANG_DATA1	Gang Programmer Flash program data 1	
104	GANG_DATA2	Gang Programmer Flash program data 2	
143	GANG_DATA3	Gang Programmer Flash program data 3	
141	GANG_DATA4	Gang Programmer Flash program data 4	
139	GANG_DATA5	Gang Programmer Flash program data 5	
137	GANG_DATA6	Gang Programmer Flash program data 6	
93	GANG_DATA7	Gang Programmer Flash program data 7	

3.6 Pin Multiplexing

Multifunction Pin Multiplexing in the MEC1632 is controlled by the GPIO Interface and illustrated in the Multiplexing Tables that follow. See Section 3.7, "Notes for Tables in this Chapter," on page 53 for notes that are referenced in the Pin Multiplexing tables. See Section 24.9.1, "Pin Control Register," on page 396 for pin multiplexing programming details. See also Section 24.7, "Pin Multiplexing Control," on page 391.

Pin signal functions that exhibit power domain emulation have a different power supply designation in the “Emulated Power Well” column and “Signal Power Well“ columns of the multiplexing tables in Section 3.6.2. See also Section 3.4.1, "Pin Default State Through Power Transitions," on page 15 for a description of pin states through power transitions.

Several signals that are alternate functions for GPIOs are not selected using the GPIO mux controls. These signals include the Gang Programmer interface. These signals are not listed in the following tables.

3.6.1 VCC POWER DOMAIN EMULATION

Pin signal functions that exhibit VCC Power Domain Emulation are documented in the multiplexing tables as “Signal Power Well“= VTR and “Emulated Power Well” = VCC. The System Runtime Supply power VCC is not connected to the MEC1632. The VCC_PWRGD signal is used to indicate when power is applied to the System Runtime Supply. All pin signal functions that exhibit VCC power domain emulation are powered by VTR and controlled by the VCC_PWRGD signal input. Outputs on VCC power domain emulation pin signal functions are tri-stated when VCC_PWRGD is not asserted and are functional when VCC_PWRGD is active. Inputs on VCC power domain emulation pin signal functions are gated according as defined by the Gated State column in the following tables.

3.6.2 MULTIPLEXING TABLES

In the following tables, the columns have the following meanings:

3.6.2.1 Pin Ref. Number

Every pin has a reference number. The mapping between pin reference numbers and signal names is summarized in the first table in this chapter.

© 2012-2018 Microchip Technology Inc.

3.6.2.2 MUX

If the pin has an associated GPIO, then the MUX column refers to the Mux Control field in the GPIO Pin Control Register. Setting the Mux Control field to value listed in the row will configure the pin for the signal listed in the Signal column on the same row. The row marked “Default” is the setting that is assigned on system reset.

If there is no GPIO associated with a pin, then the pin has a single function.

3.6.2.3 Signal

This column lists the signals that can appear on each pin, as configured by the MUX control.

3.6.2.4 Buffer Type

Pin buffer types are defined in Table 45-4, “DC Electrical Characteristics,” on page 604.

GPIO pins with the (I/O/OD) buffer type are configurable to operate as an I, an IO, or an IOD type buffer. The user may use this GPIO signal as an input only, an output only, or an I/O pin. Following the closing parenthesis is a value expressed in terms of mA, which indicates the output drive strength when the pin is configured as a buffer of type O or OD. The buffer is configured by the Output Buffer Type field in the GPIO Pin Control Register.

For signals that are multiplexed with GPIOs the following rules apply

- **Buffer Type Notation (O/OD)-mA**. Output buffer configuration options are programmed in the GPIO Pin Control Register.
- **All other Buffer Type Notations.** The buffer configuration is controlled by the logic driving the signal. When the GPIO mux control is configured for these signals, the Output Buffer Type field must be configured for Push-Pull.

3.6.2.5 Signal Power Well

This column defines the power well that powers the pin.

3.6.2.6 Emulated Power Well

For GPIOs, the emulated power well is defined by the Power Gating Signals field in the GPIO Pin Control Register. All GPIOs can be configured for VCC power well emulation, as defined in Section 3.6.1, "VCC Power Domain Emulation".

Power well emulation for signals that are multiplexed with GPIO signals is controlled by the GPIO Power Gating configuration. Entries for these signals in the following tables represent typical applications.

Power well emulating for signals that are not multiplexed with GPIO signals is defined by the entries in this column. When VCC_PWRGD is low outputs for these signals are tri-stated; internal values for these signals are gated according to the entries in the Gated State column.

3.6.2.7 Gated State

This column defines the internal value of a signal when either its emulated power well is inactive or it is not selected by the GPIO alternate function multiplexor. A value of “No Gate” means that the internal signal always follows the pin even when the emulated power well is inactive.

TABLE 3-26: MULTIPLEXING TABLE (1 OF 21)

Pin Ref. Number	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Gated State	Notes
1	Default: 0	GPIO165	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	
1	1	32KHZ_IN	IS	VBAT	VBAT	Low	
1	2	Reserved	Reserved	Reserved	Reserved		
1	3	Reserved	Reserved	Reserved	Reserved		
2		BGND	PWR	PWR	PWR		
2							
2							
2							
3		VBAT	PWR	PWR	PWR		
3							
3							
3							
4	Default: 0	BGPO0	O-8 mA	VBAT	VBAT	No Gate	
4	1	Reserved	Reserved	Reserved	Reserved		
4	2	Reserved	Reserved	Reserved	Reserved		
4	3	Reserved	Reserved	Reserved	Reserved		
5	Default: 0	GPIO101	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	Note 10
5	1	Reserved	Reserved	Reserved	Reserved		
5	2	Reserved	Reserved	Reserved	Reserved		
5	3	Reserved	Reserved	Reserved	Reserved		
6	Default: 0	GPIO102	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	Note 10
6	1	Reserved	Reserved	Reserved	Reserved		
6	2	Reserved	Reserved	Reserved	Reserved		
6	3	Reserved	Reserved	Reserved	Reserved		
7	Default: 0	GPIO172	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	Note 10
7	1	Reserved	Reserved	Reserved	Reserved		
7	2	Reserved	Reserved	Reserved	Reserved		
7	3	Reserved	Reserved	Reserved	Reserved		
8	Default: 0	GPIO173	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	Note 10
8	1	Reserved	Reserved	Reserved	Reserved		
8	2	Reserved	Reserved	Reserved	Reserved		
8	3	Reserved	Reserved	Reserved	Reserved		

© 2012-2018 Microchip Technology Inc.

TABLE 3-27: MULTIPLEXING TABLE (2 OF 21)

Pin Ref. Number	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Gated State	Notes
9	Default: 0	GPIO174	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	Note 10
9	1	Reserved	Reserved	Reserved	Reserved		
9	2	Reserved	Reserved	Reserved	Reserved		
9	3	Reserved	Reserved	Reserved	Reserved		
10	Default: 0	VCI_OUT	O-8 mA	VBAT	VBAT	No Gate	
10	1	Reserved	Reserved	Reserved	Reserved		
10	2	Reserved	Reserved	Reserved	Reserved		
10	3	Reserved	Reserved	Reserved	Reserved		
11	0	GPIO161	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	
11	Default: 1	VCI_IN2#	IS	VBAT	VBAT	Low	
11	2	Reserved	Reserved	Reserved	Reserved		
11	3	Reserved	Reserved	Reserved	Reserved		
12	0	GPIO162	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	
12	Default: 1	VCI_IN1#	IS	VBAT	VBAT	Low	
12	2	Reserved	Reserved	Reserved	Reserved		
12	3	Reserved	Reserved	Reserved	Reserved		
13	0	GPIO163	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	
13	Default: 1	VCI_IN0#	IS	VBAT	VBAT	Low	
13	2	Reserved	Reserved	Reserved	Reserved		
13	3	Reserved	Reserved	Reserved	Reserved		
14	0	GPIO164	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	
14	Default: 1	VCI_OVRD_IN	IS	VBAT	VBAT	Low	
14	2	Reserved	Reserved	Reserved	Reserved		
14	3	Reserved	Reserved	Reserved	Reserved		
15	0	GPIO000	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	
15	Default: 1	VCI_IN3#	IS	VBAT	VBAT	Low	
15	2	Reserved	Reserved	Reserved	Reserved		
15	3	Reserved	Reserved	Reserved	Reserved		
16	0	GPIO234	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	
16	Default: 1	VCI_IN4#	IS	VBAT	VBAT	Low	
16	2	Reserved	Reserved	Reserved	Reserved		
16	3	Reserved	Reserved	Reserved	Reserved		

TABLE 3-28: MULTIPLEXING TABLE (3 OF 21)

Pin Ref. Number	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Gated State	Notes
17	0	GPIO235	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	
17	Default: 1	VCI_IN5#	IS	VBAT	VBAT	Low	
17	2	Reserved	Reserved	Reserved	Reserved		
17	3	Reserved	Reserved	Reserved	Reserved		
18	Default: 0	RESETI#	I	VTR	Reserved	No Gate	
18	1	Reserved	Reserved	Reserved	Reserved		
18	2	Reserved	Reserved	Reserved	Reserved		
18	3	Reserved	Reserved	Reserved	Reserved		
19	Default: 0	GPIO062	(I/O/OD)-4 mA	VTR	VTR/VCC	No Gate	Note 7
19	1	Reserved	Reserved	Reserved	Reserved		
19	2	Reserved	Reserved	Reserved	Reserved		
19	3	Reserved	Reserved	Reserved	Reserved		
20	0	Reserved	Reserved	Reserved	Reserved		
20	Default: 1	VCC_PWRGD	I	VTR	VTR	high	Note 8
20	2	Reserved	Reserved	Reserved	Reserved		
20	3	Reserved	Reserved	Reserved	Reserved		
21	Default: 0	GPIO106	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
21	1	nRESET_OUT	O-8T mA	VTR	VTR		
21	2	Reserved	Reserved	Reserved	Reserved		
21	3	Reserved	Reserved	Reserved	Reserved		
22		VSS_RO	PWR	PWR	PWR		
22							
22							
22							
23		VTR0	PWR	PWR	PWR		
23							
23							
23							
24		VSS0	PWR	PWR	PWR		
24							
24							
24							

© 2012-2018 Microchip Technology Inc.

TABLE 3-29: MULTIPLEXING TABLE (4 OF 21)

Pin Ref. Number	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Gated State	Notes
25	Default: 0	GPIO033	(I/O/OD)-12 mA	VTR	VTR/VCC	No Gate	
25	1	RC_ID	I_AN-OD12 mA	VTR	VTR	Low	
25	2	Reserved	Reserved	Reserved	Reserved		
25	3	Reserved	Reserved	Reserved	Reserved		
26	Default: 0	GPIO021	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
26	1	KSI2	IS	VTR	VTR	Low	
26	2	Reserved	Reserved	Reserved	Reserved		
26	3	Reserved	Reserved	Reserved	Reserved		
27		VTR_REG	PWR	PWR	PWR		
27							
27							
27							
28		VR_CAP	PWR	PWR	PWR		
28							
28							
28							
29	Default: 0	GPIO060	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
29	1	KBRST	OD-8T mA	VTR	VCC		
29	2	Reserved	Reserved	Reserved	Reserved		
29	3	Reserved	Reserved	Reserved	Reserved		
30	Default: 0	GPIO166	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
30	1	Reserved	Reserved	Reserved	Reserved		
30	2	Reserved	Reserved	Reserved	Reserved		
30	3	Reserved	Reserved	Reserved	Reserved		
31		AVTR_ADC	PWR	PWR	PWR		
31							
31							
31							
32	0	GPIO200	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
32	Default: 1	ADC0	I_AN	AVTR_ADC	AVTR_ADC	Low	
32	2	Reserved	Reserved	Reserved	Reserved		
32	3	Reserved	Reserved	Reserved	Reserved		

TABLE 3-30: MULTIPLEXING TABLE (5 OF 21)

Pin Ref. Number	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Gated State	Notes
33	0	GPIO210	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
33	Default: 1	ADC8	I_AN	AVTR_ADC	AVTR_ADC	Low	
33	2	Reserved	Reserved	Reserved	Reserved		
33	3	Reserved	Reserved	Reserved	Reserved		
34	0	GPIO201	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
34	Default: 1	ADC1	I_AN	AVTR_ADC	AVTR_ADC	Low	
34	2	Reserved	Reserved	Reserved	Reserved		
34	3	Reserved	Reserved	Reserved	Reserved		
35	0	GPIO211	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
35	Default: 1	ADC9	I_AN	AVTR_ADC	AVTR_ADC	Low	
35	2	Reserved	Reserved	Reserved	Reserved		
35	3	Reserved	Reserved	Reserved	Reserved		
36		VTR3	PWR	PWR	PWR		
36							
36							
36							
37	0	GPIO202	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
37	Default: 1	ADC2	I_AN	AVTR_ADC	AVTR_ADC	Low	
37	2	Reserved	Reserved	Reserved	Reserved		
37	3	Reserved	Reserved	Reserved	Reserved		
38	0	GPIO212	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
38	Default: 1	ADC10	I_AN	AVTR_ADC	AVTR_ADC	Low	
38	2	Reserved	Reserved	Reserved	Reserved		
38	3	Reserved	Reserved	Reserved	Reserved		
39	0	GPIO203	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
39	Default: 1	ADC3	I_AN	AVTR_ADC	AVTR_ADC	Low	
39	2	Reserved	Reserved	Reserved	Reserved		
39	3	Reserved	Reserved	Reserved	Reserved		
40	0	GPIO213	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
40	Default: 1	ADC11	I_AN	AVTR_ADC	AVTR_ADC	Low	
40	2	Reserved	Reserved	Reserved	Reserved		
40	3	Reserved	Reserved	Reserved	Reserved		

© 2012-2018 Microchip Technology Inc.

TABLE 3-31: MULTIPLEXING TABLE (6 OF 21)

Pin Ref. Number	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Gated State	Notes
41	0	GPIO204	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
41	Default: 1	ADC4	I_AN	AVTR_ADC	AVTR_ADC	Low	
41	2	Reserved	Reserved	Reserved	Reserved		
41	3	Reserved	Reserved	Reserved	Reserved		
42	0	GPIO214	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
42	Default: 1	ADC12	I_AN	AVTR_ADC	AVTR_ADC	Low	
42	2	Reserved	Reserved	Reserved	Reserved		
42	3	Reserved	Reserved	Reserved	Reserved		
43	0	GPIO205	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
43	Default: 1	ADC5	I_AN	AVTR_ADC	AVTR_ADC	Low	
43	2	Reserved	Reserved	Reserved	Reserved		
43	3	Reserved	Reserved	Reserved	Reserved		
44	0	GPIO215	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
44	Default: 1	ADC13	I_AN	AVTR_ADC	AVTR_ADC	Low	
44	2	Reserved	Reserved	Reserved	Reserved		
44	3	Reserved	Reserved	Reserved	Reserved		
45	0	GPIO206	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
45	Default: 1	ADC6	I_AN	AVTR_ADC	AVTR_ADC	Low	
45	2	Reserved	Reserved	Reserved	Reserved		
45	3	Reserved	Reserved	Reserved	Reserved		
46	0	GPIO216	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
46	Default: 1	ADC14	I_AN	AVTR_ADC	AVTR_ADC	Low	
46	2	Reserved	Reserved	Reserved	Reserved		
46	3	Reserved	Reserved	Reserved	Reserved		
47	0	GPIO207	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
47	Default: 1	ADC7	I_AN	AVTR_ADC	AVTR_ADC	Low	
47	2	Reserved	Reserved	Reserved	Reserved		
47	3	Reserved	Reserved	Reserved	Reserved		
48	0	GPIO217	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
48	Default: 1	ADC15	I_AN	AVTR_ADC	AVTR_ADC	Low	
48	2	Reserved	Reserved	Reserved	Reserved		
48	3	Reserved	Reserved	Reserved	Reserved		

TABLE 3-32: MULTIPLEXING TABLE (7 OF 21)

Pin Ref. Number	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Gated State	Notes
49		VSS_ADC	PWR	PWR	PWR		
49							
49							
49							
50		VREF_ADC	PWR	PWR	PWR		
50							
50							
50							
51	0	Reserved	Reserved	Reserved	Reserved		
51	Default: 1	LRESET#	PCI_I	VTR	VCC	Low	Note 8
51	2	Reserved	Reserved	Reserved	Reserved		
51	3	Reserved	Reserved	Reserved	Reserved		
52	0	Reserved	Reserved	Reserved	Reserved		
52	Default: 1	CLKRUN#	PCI_OD	VTR	VCC		Note 8
52	2	Reserved	Reserved	Reserved	Reserved		
52	3	Reserved	Reserved	Reserved	Reserved		
53	0	Reserved	Reserved	Reserved	Reserved		
53	Default: 1	LFRAME#	PCI_I	VTR	VCC	Low	Note 8
53	2	Reserved	Reserved	Reserved	Reserved		
53	3	Reserved	Reserved	Reserved	Reserved		
54	0	Reserved	Reserved	Reserved	Reserved		
54	Default: 1	SER_IRQ	PCI_IO	VTR	VCC	Low	Note 2, Note 8
54	2	Reserved	Reserved	Reserved	Reserved		
54	3	Reserved	Reserved	Reserved	Reserved		
55	0	Reserved	Reserved	Reserved	Reserved		
55	Default: 1	PCI_CLK	PCI_ICLK	VTR	VCC	Low	Note 8
55	2	Reserved	Reserved	Reserved	Reserved		
55	3	Reserved	Reserved	Reserved	Reserved		
56	Default: 0	LAD0	PCI_IO	VTR	VCC	No Gate	Note 2
56	1	Reserved	Reserved	Reserved	Reserved		
56	2	Reserved	Reserved	Reserved	Reserved		
56	3	Reserved	Reserved	Reserved	Reserved		

© 2012-2018 Microchip Technology Inc.

TABLE 3-33: MULTIPLEXING TABLE (8 OF 21)

Pin Ref. Number	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Gated State	Notes
57	Default: 0	LAD1	PCI_IO	VTR	VCC	No Gate	Note 2
57	1	Reserved	Reserved	Reserved	Reserved		
57	2	Reserved	Reserved	Reserved	Reserved		
57	3	Reserved	Reserved	Reserved	Reserved		
58		VTR2	PWR	PWR	PWR		
58							
58							
58							
59	Default: 0	LAD2	PCI_IO	VTR	VCC	No Gate	Note 2
59	1	Reserved	Reserved	Reserved	Reserved		
59	2	Reserved	Reserved	Reserved	Reserved		
59	3	Reserved	Reserved	Reserved	Reserved		
60	Default: 0	LAD3	PCI_IO	VTR	VCC	No Gate	Note 2
60	1	Reserved	Reserved	Reserved	Reserved		
60	2	Reserved	Reserved	Reserved	Reserved		
60	3	Reserved	Reserved	Reserved	Reserved		
61	Default: 0	GPIO100	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
61	1	nEC_SCI	OD-8T mA	VTR	VTR		Note 1
61	2	Reserved	Reserved	Reserved	Reserved		
61	3	Reserved	Reserved	Reserved	Reserved		
62	Default: 0	GPIO011	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
62	1	nSMI	OD-8T mA	VTR	VTR		
62	2	Reserved	Reserved	Reserved	Reserved		
62	3	Reserved	Reserved	Reserved	Reserved		
63	Default: 0	GPIO061	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
63	1	LPCPD#	I	VTR	VTR	Low	
63	2	Reserved	Reserved	Reserved	Reserved		
63	3	Reserved	Reserved	Reserved	Reserved		
64	Default: 0	GPIO050	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
64	1	FAN_TACH0	I	VTR	VTR	Low	
64	2	Reserved	Reserved	Reserved	Reserved		
64	3	Reserved	Reserved	Reserved	Reserved		

TABLE 3-34: MULTIPLEXING TABLE (9 OF 21)

Pin Ref. Number	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Gated State	Notes
65	Default: 0	GPIO222	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
65	1	Reserved	Reserved	Reserved	Reserved		
65	2	Reserved	Reserved	Reserved	Reserved		
65	3	Reserved	Reserved	Reserved	Reserved		
66	Default: 0	GPIO051	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
66	1	FAN_TACH1	I	VTR	VTR	Low	
66	2	Reserved	Reserved	Reserved	Reserved		
66	3	Reserved	Reserved	Reserved	Reserved		
67	Default: 0	GPIO223	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
67	1	Reserved	Reserved	Reserved	Reserved		
67	2	Reserved	Reserved	Reserved	Reserved		
67	3	Reserved	Reserved	Reserved	Reserved		
68	Default: 0	GPIO052	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
68	1	FAN_TACH2	I	VTR	VTR	Low	
68	2	Reserved	Reserved	Reserved	Reserved		
68	3	Reserved	Reserved	Reserved	Reserved		
69	Default: 0	GPIO224	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
69	1	Reserved	Reserved	Reserved	Reserved		
69	2	Reserved	Reserved	Reserved	Reserved		
69	3	Reserved	Reserved	Reserved	Reserved		
70	Default: 0	GPIO016	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
70	1	GPTP-IN7	I	VTR	VTR	Low	
70	2	FAN_TACH3	I	VTR	VTR	Low	
70	3	Reserved	Reserved	Reserved	Reserved		
71	Default: 0	GPIO230	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	
71	1	ECGP_SCLK	O-8 mA	VTR	VTR		
71	2	Reserved	Reserved	Reserved	Reserved		
71	3	Reserved	Reserved	Reserved	Reserved		
72	Default: 0	GPIO053	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
72	1	PWM0	(O/OD)-8T mA	VTR	VTR		
72	2	Reserved	Reserved	Reserved	Reserved		
72	3	Reserved	Reserved	Reserved	Reserved		

© 2012-2018 Microchip Technology Inc.

TABLE 3-35: MULTIPLEXING TABLE (10 OF 21)

Pin Ref. Number	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Gated State	Notes
73	Default: 0	GPIO231	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	
73	1	ECGP_SOUT	IO-8 mA	VTR	VTR	Low	
73	2	Reserved	Reserved	Reserved	Reserved		
73	3	Reserved	Reserved	Reserved	Reserved		
74	Default: 0	GPIO054	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
74	1	PWM1	(O/OD)-8T mA	VTR	VTR		
74	2	Reserved	Reserved	Reserved	Reserved		
74	3	Reserved	Reserved	Reserved	Reserved		
75	Default: 0	GPIO233	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	
75	1	ECGP_SIN	IO-8 mA	VTR	VTR	Low	
75	2	Reserved	Reserved	Reserved	Reserved		
75	3	Reserved	Reserved	Reserved	Reserved		
76	Default: 0	GPIO055	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
76	1	PWM2	(O/OD)-8T mA	VTR	VTR		
76	2	Reserved	Reserved	Reserved	Reserved		
76	3	Reserved	Reserved	Reserved	Reserved		
77	Default: 0	GPIO056	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
77	1	PWM3	(O/OD)-8T mA	VTR	VTR		
77	2	Reserved	Reserved	Reserved	Reserved		
77	3	Reserved	Reserved	Reserved	Reserved		
78	Default: 0	GPIO001	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
78	1	PWM4	(O/OD)-8T mA	VTR	VTR		
78	2	Reserved	Reserved	Reserved	Reserved		
78	3	Reserved	Reserved	Reserved	Reserved		
79	Default: 0	GPIO002	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
79	1	PWM5	(O/OD)-8T mA	VTR	VTR		
79	2	Reserved	Reserved	Reserved	Reserved		
79	3	Reserved	Reserved	Reserved	Reserved		
80	Default: 0	GPIO014	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
80	1	GPTP-IN6	I	VTR	VTR	Low	
80	2	PWM6	(O/OD)-8T mA	VTR	VTR		
80	3	Reserved	Reserved	Reserved	Reserved		

TABLE 3-36: MULTIPLEXING TABLE (11 OF 21)

Pin Ref. Number	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Gated State	Notes
81	Default: 0	GPIO015	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
81	1	GPTP-OUT6	(O/OD)-8T mA	VTR	VTR		
81	2	PWM7	(O/OD)-8T mA	VTR	VTR		
81	3	Reserved	Reserved	Reserved	Reserved		
82	Default: 0	GPIO151	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
82	1	GPTP-IN3	I	VTR	VTR	Low	
82	2	FAN_TACH4	I	VTR	VTR	Low	
82	3	KSO15	(O/OD)-8T mA	VTR	VTR		
83	Default: 0	GPIO152	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
83	1	GPTP-OUT3	(O/OD)-8T mA	VTR	VTR		
83	2	FAN_TACH5	I	VTR	VTR	Low	
83	3	KSO16	(O/OD)-8T mA	VTR	VTR		
84	Default: 0	GPIO017	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
84	1	GPTP-OUT7	(O/OD)-8T mA	VTR	VTR		
84	2	TOUT3	O-8T mA	VTR	VTR		
84	3	KSI0	IS	VTR	VTR	Low	
85	Default: 0	GPIO040	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
85	1	GPTP-OUT2	(O/OD)-8T mA	VTR	VTR		
85	2	TOUT2	O-8T mA	VTR	VTR		
85	3	KSO0	(O/OD)-8T mA	VTR	VTR		
86	Default: 0	GPIO032	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
86	1	GPTP-IN2	I	VTR	VTR/VCC	Low	
86	2	TOUT1	O-8T mA	VTR	VTR		
86	3	KSI7	IS	VTR	VTR	Low	
87	Default: 0	GPIO031	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
87	1	GPTP-OUT1	(O/OD)-8T mA	VTR	VTR/VCC		
87	2	TOUT0	O-8T mA	VTR	VTR		
87	3	KSI6	IS	VTR	VTR	Low	
88	Default: 0	GPIO012	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	
88	1	SMB07_DATA	IOD-8 mA	VTR	VTR/VCC	High	
88	2	Reserved	Reserved	Reserved	Reserved		
88	3	Reserved	Reserved	Reserved	Reserved		

© 2012-2018 Microchip Technology Inc.

TABLE 3-37: MULTIPLEXING TABLE (12 OF 21)

Pin Ref. Number	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Gated State	Notes
89	Default: 0	GPIO013	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	
89	1	SMB07_CLK	IOD-8 mA	VTR	VTR/VCC	High	
89	2	Reserved	Reserved	Reserved	Reserved		
89	3	Reserved	Reserved	Reserved	Reserved		
90	Default: 0	GPIO130	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	
90	1	SMB10_DATA	IOD-8 mA	VTR	VTR/VCC	High	
90	2	Reserved	Reserved	Reserved	Reserved		
90	3	Reserved	Reserved	Reserved	Reserved		
91	Default: 0	GPIO131	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	
91	1	SMB10_CLK	IOD-8 mA	VTR	VTR/VCC	High	
91	2	Reserved	Reserved	Reserved	Reserved		
91	3	Reserved	Reserved	Reserved	Reserved		
92	Default: 0	GPIO132	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	
92	1	SMB06_DATA	IOD-8 mA	VTR	VTR	High	
92	2	KSO14	(O/OD)-8 mA	VTR	VTR		
92	3	Reserved	Reserved	Reserved	Reserved		
93	Default: 0	GPIO140	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	
93	1	SMB06_CLK	IOD-8 mA	VTR	VTR	High	
93	2	PWM13	(O/OD)-8 mA	VTR	VTR		
93	3	Reserved	Reserved	Reserved	Reserved		
94		VTR_FLASH	PWR	PWR	PWR		
94							
94							
94							
95	Default: 0	GPIO025	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
95	1	UART_CLK	I	VTR	VTR	Low	
95	2	TIN0	I	VTR	VTR	Low	
95	3	nEM_INT	(O/OD)-8T mA	VTR	VCC		
96	Default: 0	GPIO026	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
96	1	GPTP-IN0	I	VTR	VTR	Low	
96	2	TIN1	I	VTR	VTR	Low	
96	3	KSI3	IS	VTR	VTR	Low	

TABLE 3-38: MULTIPLEXING TABLE (13 OF 21)

Pin Ref. Number	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Gated State	Notes
97	Default: 0	GPIO027	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
97	1	GPTP-OUT0	(O/OD)-8T mA	VTR	VTR		
97	2	TIN2	I	VTR	VTR	Low	
97	3	KSI4	IS	VTR	VTR	Low	
98	Default: 0	GPIO030	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
98	1	GPTP-IN1	I	VTR	VTR	Low	
98	2	TIN3	I	VTR	VTR	Low	
98	3	KSI5	IS	VTR	VTR	Low	
99	Default: 0	GPIO107	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
99	1	KSO4	(O/OD)-8T mA	VTR	VTR		
99	2	Reserved	Reserved	Reserved	Reserved		
99	3	Reserved	Reserved	Reserved	Reserved		
100	Default: 0	GPIO120	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
100	1	KSO7	(O/OD)-8T mA	VTR	VTR		
100	2	Reserved	Reserved	Reserved	Reserved		
100	3	Reserved	Reserved	Reserved	Reserved		
101	Default: 0	GPIO124	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
101	1	GPTP-OUT4	(O/OD)-8T mA	VTR	VTR/VCC		
101	2	KSO11	(O/OD)-8T mA	VTR	VTR		
101	3	Reserved	Reserved	Reserved	Reserved		
102	Default: 0	GPIO125	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
102	1	GPTP-IN4	I	VTR	VTR/VCC	Low	
102	2	KSO12	(O/OD)-8T mA	VTR	VTR		
102	3	Reserved	Reserved	Reserved	Reserved		
103	Default: 0	GPIO110	(I/O/OD)-12 mA	VTR	VTR/VCC	No Gate	
103	1	PS2_CLK2	IOD-12 mA	VTR	VTR/VCC	Low	
103	2	GPTP-IN5	I	VTR	VTR	Low	
103	3	Reserved	Reserved	Reserved	Reserved		
104	Default: 0	GPIO111	(I/O/OD)-12 mA	VTR	VTR/VCC	No Gate	
104	1	PS2_DAT2	IOD-12 mA	VTR	VTR/VCC	Low	
104	2	GPTP-OUT5	(O/OD)-12 mA	VTR	VTR		
104	3	Reserved	Reserved	Reserved	Reserved		

© 2012-2018 Microchip Technology Inc.

TABLE 3-39: MULTIPLEXING TABLE (14 OF 21)

Pin Ref. Number	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Gated State	Notes
105	Default: 0	GPIO112	(I/O/OD)-12 mA	VTR	VTR/VCC	No Gate	
105	1	PS2_CLK1A	IOD-12 mA	VTR	VTR/VCC	Low	Note 6
105	2	KSO5	(O/OD)-12 mA	VTR	VTR		
105	3	Reserved	Reserved	Reserved	Reserved		
106	Default: 0	GPIO113	(I/O/OD)-12 mA	VTR	VTR/VCC	No Gate	
106	1	PS2_DAT1A	IOD-12 mA	VTR	VTR/VCC	Low	Note 6
106	2	KSO6	(O/OD)-12 mA	VTR	VTR		
106	3	Reserved	Reserved	Reserved	Reserved		
107	Default: 0	GPIO114	(I/O/OD)-12 mA	VTR	VTR/VCC	No Gate	
107	1	PS2_CLK0A	IOD-12 mA	VTR	VTR/VCC	Low	Note 6
107	2	Reserved	Reserved	Reserved	Reserved		
107	3	Reserved	Reserved	Reserved	Reserved		
108	Default: 0	GPIO115	(I/O/OD)-12 mA	VTR	VTR/VCC	No Gate	
108	1	PS2_DAT0A	IOD-12 mA	VTR	VTR/VCC	Low	Note 6
108	2	Reserved	Reserved	Reserved	Reserved		
108	3	Reserved	Reserved	Reserved	Reserved		
109		VTR1	PWR	PWR	PWR		
109							
109							
109							
110		VSS1	PWR	PWR	PWR		
110							
110							
110							
111	Default: 0	GPIO070	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
111	1	Reserved	Reserved	Reserved	Reserved		
111	2	Reserved	Reserved	Reserved	Reserved		
111	3	Reserved	Reserved	Reserved	Reserved		
112	Default: 0	GPIO071	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
112	1	Reserved	Reserved	Reserved	Reserved		
112	2	Reserved	Reserved	Reserved	Reserved		
112	3	Reserved	Reserved	Reserved	Reserved		

TABLE 3-40: MULTIPLEXING TABLE (15 OF 21)

Pin Ref. Number	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Gated State	Notes
113	Default: 0	GPIO072	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
113	1	Reserved	Reserved	Reserved	Reserved		
113	2	Reserved	Reserved	Reserved	Reserved		
113	3	Reserved	Reserved	Reserved	Reserved		
114	Default: 0	GPIO073	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
114	1	Reserved	Reserved	Reserved	Reserved		
114	2	Reserved	Reserved	Reserved	Reserved		
114	3	Reserved	Reserved	Reserved	Reserved		
115	Default: 0	GPIO074	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
115	1	Reserved	Reserved	Reserved	Reserved		
115	2	Reserved	Reserved	Reserved	Reserved		
115	3	Reserved	Reserved	Reserved	Reserved		
116	Default: 0	GPIO075	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
116	1	Reserved	Reserved	Reserved	Reserved		
116	2	Reserved	Reserved	Reserved	Reserved		
116	3	Reserved	Reserved	Reserved	Reserved		
117	Default: 0	GPIO041	(I/O/OD)-8T mA	VTR	VTR	No Gate	
117	1	Reserved	Reserved	Reserved	Reserved		
117	2	Reserved	Reserved	Reserved	Reserved		
117	3	Reserved	Reserved	Reserved	Reserved		
118	Default: 0	GPIO076	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
118	1	Reserved	Reserved	Reserved	Reserved		
118	2	Reserved	Reserved	Reserved	Reserved		
118	3	Reserved	Reserved	Reserved	Reserved		
119	Default: 0	GPIO220	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
119	1	Reserved	Reserved	Reserved	Reserved		
119	2	Reserved	Reserved	Reserved	Reserved		
119	3	Reserved	Reserved	Reserved	Reserved		
120	Default: 0	GPIO035	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
120	1	PWM8	(O/OD)-8T mA	VTR	VTR		
120	2	Reserved	Reserved	Reserved	Reserved		
120	3	Reserved	Reserved	Reserved	Reserved		

© 2012-2018 Microchip Technology Inc.

TABLE 3-41: MULTIPLEXING TABLE (16 OF 21)

Pin Ref. Number	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Gated State	Notes
121	Default: 0	GPIO170	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	
121	1	MSCLK	O-8 mA	VTR	VTR		
121	2	Reserved	Reserved	Reserved	Reserved		
121	3	Reserved	Reserved	Reserved	Reserved		
122	Default: 0	GPIO171	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	
122	1	MSDATA	O-8 mA	VTR	VTR		
122	2	Reserved	Reserved	Reserved	Reserved		
122	3	Reserved	Reserved	Reserved	Reserved		
123	Default: 0	GPIO133	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
123	1	PWM9	(O/OD)-8T mA	VTR	VTR		
123	2	Reserved	Reserved	Reserved	Reserved		
123	3	Reserved	Reserved	Reserved	Reserved		
124	Default: 0	GPIO134	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
124	1	PWM10	(O/OD)-8T mA	VTR	VTR		
124	2	Reserved	Reserved	Reserved	Reserved		
124	3	Reserved	Reserved	Reserved	Reserved		
125	Default: 0	GPIO135	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
125	1	PWM11	(O/OD)-8T mA	VTR	VTR		
125	2	Reserved	Reserved	Reserved	Reserved		
125	3	Reserved	Reserved	Reserved	Reserved		
126	Default: 0	GPIO042	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	
126	1	Reserved	Reserved	Reserved	Reserved		
126	2	PECI_DAT	IO-PECI	VREF_VTT	VREF_VTT	Low	
126	3	SB-TSI_DAT	SB-TSI	VREF_VTT	VREF_VTT	High	
127	Default: 0	GPIO043	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	
127	1	Reserved	Reserved	Reserved	Reserved		
127	2	Reserved	Reserved	Reserved	Reserved		
127	3	SB-TSI_CLK	SB-TSI	VREF_VTT	VREF_VTT	High	
128	Default: 0	GPIO044	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
128	1	VREF_VTT	I	VTR	VTR	No Gate	
128	2	Reserved	Reserved	Reserved	Reserved		
128	3	Reserved	Reserved	Reserved	Reserved		

TABLE 3-42: MULTIPLEXING TABLE (17 OF 21)

Pin Ref. Number	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Gated State	Notes
129	Default: 0	GPIO145	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	
129	1	SMB09_DATA	IOD-8 mA	VTR	VTR	High	
129	2	JTAG_TDI	I	VTR	VTR	Low	
129	3	Reserved	Reserved	Reserved	Reserved		
130	Default: 0	GPIO146	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	
130	1	SMB09_CLK	IOD-8 mA	VTR	VTR	High	
130	2	JTAG_TDO	O-8 mA	VTR	VTR		
130	3	Reserved	Reserved	Reserved	Reserved		
131	Default: 0	GPIO147	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	
131	1	SMB08_DATA	IOD-8 mA	VTR	VTR	High	
131	2	Reserved	Reserved	Reserved	Reserved		
131	3	JTAG_CLK	I	VTR	VTR	Low	
132	Default: 0	GPIO150	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	
132	1	SMB08_CLK	IOD-8 mA	VTR	VTR	High	
132	2	Reserved	Reserved	Reserved	Reserved		
132	3	JTAG_TMS	I	VTR	VTR	Low	
133	Default: 0	JTAG_RST#	IS	VTR	VTR	No Gate	Note 5
133	1	Reserved	Reserved	Reserved	Reserved		
133	2	Reserved	Reserved	Reserved	Reserved		
133	3	Reserved	Reserved	Reserved	Reserved		
134	Default: 0	GPIO104	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	
134	1	UART_TX	O-8 mA	VTR	VTR/VCC		Note 4
134	2	Reserved	Reserved	Reserved	Reserved		
134	3	Reserved	Reserved	Reserved	Reserved		
135	Default: 0	GPIO105	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
135	1	UART_RX	I	VTR	VTR/VCC	Low	Note 4
135	2	Reserved	Reserved	Reserved	Reserved		
135	3	Reserved	Reserved	Reserved	Reserved		
136	Default: 0	GPIO141	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	
136	1	SMB05_DATA	IOD-8 mA	VTR	VTR	High	
136	2	PWM14	(O/OD)-8 mA	VTR	VTR		
136	3	FLSCLK	O-8 mA	VTR	VTR		

© 2012-2018 Microchip Technology Inc.

TABLE 3-43: MULTIPLEXING TABLE (18 OF 21)

Pin Ref. Number	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Gated State	Notes
137	Default: 0	GPIO142	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	
137	1	SMB05_CLK	IOD-8 mA	VTR	VTR	High	
137	2	PWM15	(O/OD)-8 mA	VTR	VTR		
137	3	FLSOUT	(I/O/OD)-8 mA	VTR	VTR	Low	
138	Default: 0	GPIO143	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	
138	1	SMB04_DATA	IOD-8 mA	VTR	VTR	High	
138	2	PWM12	(O/OD)-8 mA	VTR	VTR		
138	3	FLSIN	(I/O/OD)-8 mA	VTR	VTR	Low	
139	Default: 0	GPIO144	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	
139	1	SMB04_CLK	IOD-8 mA	VTR	VTR	High	
139	2	Reserved	Reserved	Reserved	Reserved		
139	3	FLSCS#	O-8 mA	VTR	VTR		
140	Default: 0	GPIO007	(I/O/OD)-12 mA	VTR	VTR/VCC	No Gate	
140	1	SMB03_DATA	IOD-12 mA	VTR	VTR	High	
140	2	PS2_CLK0B	IOD-12 mA	VTR	VTR	Low	Note 6
140	3	Reserved	Reserved	Reserved	Reserved		
141	Default: 0	GPIO010	(I/O/OD)-12 mA	VTR	VTR/VCC	No Gate	
141	1	SMB03_CLK	IOD-12 mA	VTR	VTR	High	
141	2	PS2_DAT0B	IOD-12 mA	VTR	VTR	Low	Note 6
141	3	Reserved	Reserved	Reserved	Reserved		
142	Default: 0	GPIO154	(I/O/OD)-12 mA	VTR	VTR/VCC	No Gate	
142	1	SMB02_DATA	IOD-12 mA	VTR	VTR	High	
142	2	PS2_CLK1B	IOD-12 mA	VTR	VTR	Low	Note 6
142	3	Reserved	Reserved	Reserved	Reserved		
143	Default: 0	GPIO155	(I/O/OD)-12 mA	VTR	VTR/VCC	No Gate	
143	1	SMB02_CLK	IOD-12 mA	VTR	VTR	High	
143	2	PS2_DAT1B	IOD-12 mA	VTR	VTR	Low	Note 6
143	3	Reserved	Reserved	Reserved	Reserved		
144	Default: 0	GPIO006	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	
144	1	SMB01_CLK	IOD-8 mA	VTR	VTR	High	
144	2	Reserved	Reserved	Reserved	Reserved		
144	3	Reserved	Reserved	Reserved	Reserved		

TABLE 3-44: MULTIPLEXING TABLE (19 OF 21)

Pin Ref. Number	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Gated State	Notes
145	Default: 0	GPIO005	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	
145	1	SMB01_DATA	IOD-8 mA	VTR	VTR	High	
145	2	Reserved	Reserved	Reserved	Reserved		
145	3	Reserved	Reserved	Reserved	Reserved		
146	Default: 0	GPIO004	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	
146	1	SMB00_CLK	IOD-8 mA	VTR	VTR	High	
146	2	Reserved	Reserved	Reserved	Reserved		
146	3	Reserved	Reserved	Reserved	Reserved		
147	Default: 0	GPIO003	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	
147	1	SMB00_DATA	IOD-8 mA	VTR	VTR	High	
147	2	Reserved	Reserved	Reserved	Reserved		
147	3	Reserved	Reserved	Reserved	Reserved		
148	Default: 0	GPIO022	(I/O/OD)-16 mA	VTR	VTR/VCC	No Gate	
148	1	BCM_B_CLK	O-16 mA	VTR	VTR		
148	2	Reserved	Reserved	Reserved	Reserved		
148	3	Reserved	Reserved	Reserved	Reserved		
149	Default: 0	GPIO023	(I/O/OD)-16 mA	VTR	VTR/VCC	No Gate	
149	1	BCM_B_DAT	IO-16 mA	VTR	VTR	Low	Note 3
149	2	Reserved	Reserved	Reserved	Reserved		
149	3	Reserved	Reserved	Reserved	Reserved		
150	Default: 0	GPIO024	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
150	1	BCM_B_INT#	IO-8T mA	VTR	VTR	Low	
150	2	Reserved	Reserved	Reserved	Reserved		
150	3	Reserved	Reserved	Reserved	Reserved		
151	Default: 0	GPIO127	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
151	1	A20M	O-8T mA	VTR	VTR		
151	2	Reserved	Reserved	Reserved	Reserved		
151	3	Reserved	Reserved	Reserved	Reserved		
152	Default: 0	GPIO034	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	
152	1	CEC_OUT	(O/OD)-8 mA	VTR	VTR		
152	2	Reserved	Reserved	Reserved	Reserved		
152	3	Reserved	Reserved	Reserved	Reserved		

© 2012-2018 Microchip Technology Inc.

TABLE 3-45: MULTIPLEXING TABLE (20 OF 21)

Pin Ref. Number	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Gated State	Notes
153	Default: 0	GPIO036	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
153	1	CEC_IN	I	VTR	VTR	Low	
153	2	Reserved	Reserved	Reserved	Reserved		
153	3	Reserved	Reserved	Reserved	Reserved		
154	Default: 0	GPIO045	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
154	1	LSBCM_D_INT#	I	VTR	VTR	Low	
154	2	KSO1	(O/OD)-8T mA	VTR	VTR		
154	3	Reserved	Reserved	Reserved	Reserved		
155	Default: 0	GPIO046	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	
155	1	LSBCM_D_DAT	IO-8 mA	VTR	VTR	Low	Note 3
155	2	KSO2	(O/OD)-8 mA	VTR	VTR		
155	3	Reserved	Reserved	Reserved	Reserved		
156	Default: 0	GPIO047	(I/O/OD)-8 mA	VTR	VTR/VCC	No Gate	
156	1	LSBCM_D_CLK	O-8 mA	VTR	VTR		
156	2	KSO3	(O/OD)-8 mA	VTR	VTR		
156	3	Reserved	Reserved	Reserved	Reserved		
157	Default: 0	GPIO121	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
157	1	BCM_A_INT#	I	VTR	VTR	Low	
157	2	KSO8	(O/OD)-8T mA	VTR	VTR		
157	3	Reserved	Reserved	Reserved	Reserved		
158	Default: 0	GPIO122	(I/O/OD)-16 mA	VTR	VTR/VCC	No Gate	
158	1	BCM_A_DAT	IO-16 mA	VTR	VTR	Low	Note 3
158	2	KSO9	(O/OD)-16 mA	VTR	VTR		
158	3	Reserved	Reserved	Reserved	Reserved		
159	Default: 0	GPIO123	(I/O/OD)-16 mA	VTR	VTR/VCC	No Gate	
159	1	BCM_A_CLK	O-16 mA	VTR	VTR		
159	2	KSO10	(O/OD)-16 mA	VTR	VTR		
159	3	Reserved	Reserved	Reserved	Reserved		
160	Default: 0	GPIO126	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
160	1	KSO13	(O/OD)-8T mA	VTR	VTR		
160	2	Reserved	Reserved	Reserved	Reserved		
160	3	Reserved	Reserved	Reserved	Reserved		

TABLE 3-46: MULTIPLEXING TABLE (21 OF 21)

Pin Ref. Number	MUX	Signal	Buffer Type	Signal Power Well	Emulated Power Well	Gated State	Notes
161	Default: 0	GPIO020	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
161	1	KSI1	I	VTR	VTR	Low	
161	2	Reserved	Reserved	Reserved	Reserved		
161	3	Reserved	Reserved	Reserved	Reserved		
162	Default: 0	GPIO156	(I/O/OD)-20 mA	VTR	VTR/VCC	No Gate	
162	1	LED0	(O/OD)-20 mA	VTR	VTR		Note 9
162	2	Reserved	Reserved	Reserved	Reserved		
162	3	Reserved	Reserved	Reserved	Reserved		
163	Default: 0	GPIO157	(I/O/OD)-20 mA	VTR	VTR/VCC	No Gate	
163	1	LED1	(O/OD)-20 mA	VTR	VTR		Note 9
163	2	Reserved	Reserved	Reserved	Reserved		
163	3	Reserved	Reserved	Reserved	Reserved		
164	Default: 0	GPIO153	(I/O/OD)-20 mA	VTR	VTR/VCC	No Gate	
164	1	LED2	(O/OD)-20 mA	VTR	VTR		Note 9
164	2	Reserved	Reserved	Reserved	Reserved		
164	3	Reserved	Reserved	Reserved	Reserved		
165	Default: 0	GPIO175	(I/O/OD)-8T mA	VTR	VTR/VCC	No Gate	
165	1	32KHZ_OUT	O-8T mA	VTR	VTR		
165	2	KSO17	(O/OD)-8T mA	VTR	VTR		
165	3	Reserved	Reserved	Reserved	Reserved		
166		VSS2	PWR	PWR	PWR		
166							
166							
166							
167	Default: 0	XTAL1	I_AN	VBAT	VBAT	No Gate	
167	1	Reserved	Reserved	Reserved	Reserved		
167	2	Reserved	Reserved	Reserved	Reserved		
167	3	Reserved	Reserved	Reserved	Reserved		
168	Default: 0	XTAL2	I_AN	VBAT	VBAT	No Gate	
168	1	Reserved	Reserved	Reserved	Reserved		
168	2	Reserved	Reserved	Reserved	Reserved		
168	3	Reserved	Reserved	Reserved	Reserved		
169		VSS_XTAL	PWR	PWR	PWR		
169							
169							
169							

© 2012-2018 Microchip Technology Inc.

3.7 Notes for Tables in this Chapter

Note 1	The nEC_SCI pin can be controlled by hardware and EC firmware. The nEC_SCI pin can drive either the ACPI Run-time GPE Chipset input or the Wake GPE Chipset input. Depending how the nEC_SCI pin is used, other ACPI-related SCI functions may be best supplied by other general purpose outputs that can be configured as open-drain drivers.
Note 2	These pins require an external weak pull-up resistors of 10k-100k ohms.
Note 3	A pull-up is not needed on this BC-Link DATA pin as long as the voltage remains above the logic-high threshold during the second turnaround cycle.
Note 4	The two pin debug port UART can be used by the Host or EC. This pin can be VCC protected or not VCC portected under program control by the POWER bit in the Configuration Select Register in Host configuration space (also accessible by the EC).
Note 5	When the JTAG_RST# pin is not asserted (logic'1'), the JTAG_TDI, JTAG_TDO, JTAG_CLK, JTAG_TMS signal functions in the JTAG interface are unconditionally routed to the interface; the Pin Control register for these pins has no effect. When the JTAG_RST# pin is asserted (logic'0'), the JTAG_TDI, JTAG_TDO, JTAG_CLK, JTAG_TMS signal functions in the JTAG interface are not routed to the interface and the Pin Control Register for these pins controls the muxing. The pin control registers can not route the JTAG interface to the pins. System Board Designer should terminate this pin in all functional state using jumpers and pull-up or pull down resistors, etc .
Note 6	PS/2 ports ending with signal functions ending with "A" or "B" are muxed to a single controller. Only one set of clock and data are intended to used at a time (either "A" or "B" not both. The unused port segment should have its associated pin control register's, Mux Control Field programmed away from the PS2 controller.
Note 7	The GPIO062 pin defaults to output 'low' on nSYS_RST to support the firmware controlled RESETO# feature. RESETO# is not a GPIO alternate function; it is controlled by firmware as a GPIO function.
Note 8	This pin has a GPIO signal for interrupt/wake-only signal functions.
Note 9	The LED pins do not have internal pull-downs.
Note 10	This pin is muxed with a BGPO signal that is configured by programming the Week Timer Control register. When configured as the BGPO function the pad is powered by VBAT.

3.8 Strapping Options

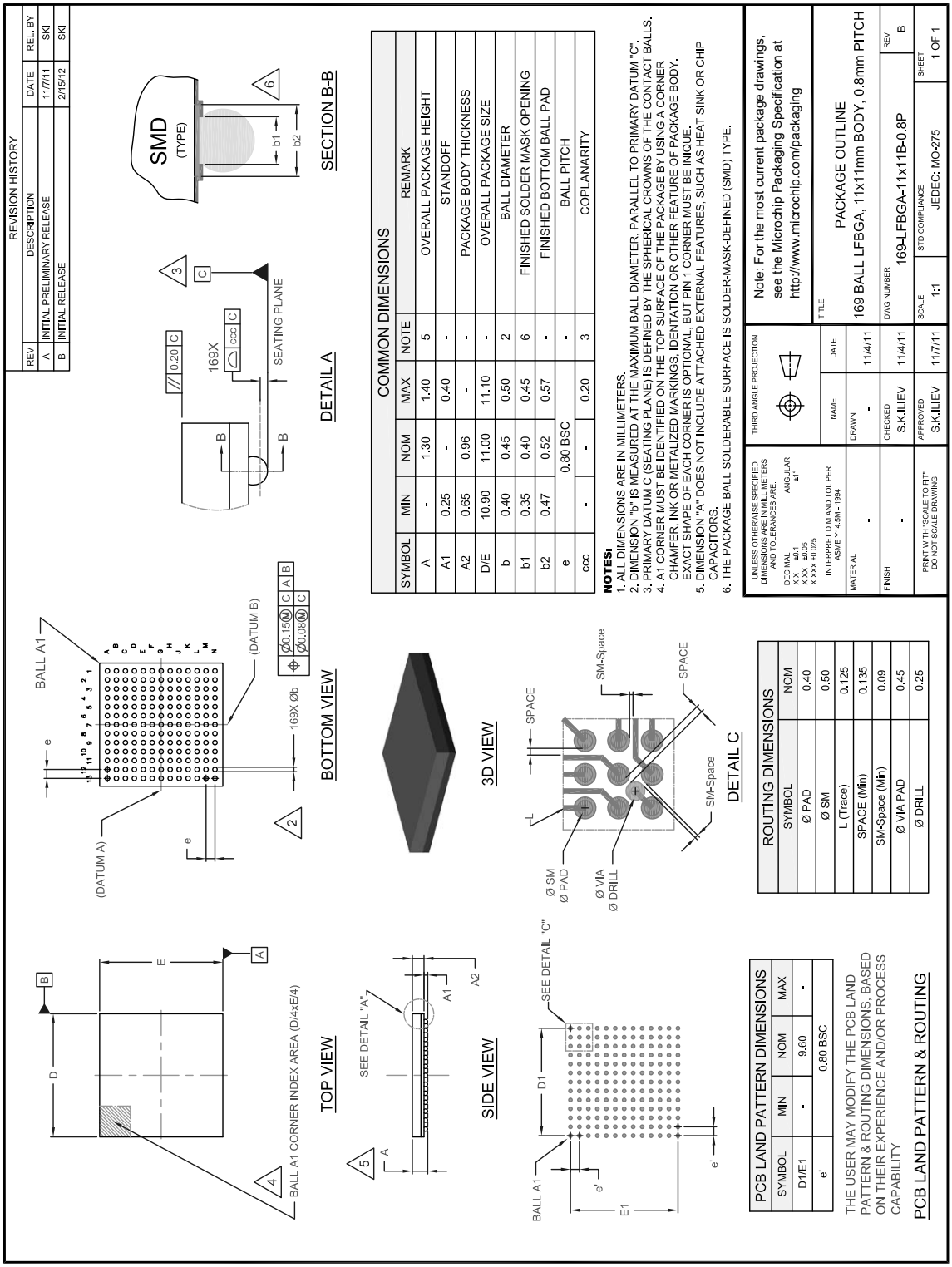
GPIO171 is used for the TAP Controller select strap (see Section 44.2.1, "TAP Controller Select Strap Option," on page 580). If any of the MEC1632 JTAG TAP controllers are used, GPIO171 must only be configured as an output to a VTR powered external function. GPIO171 may only be configured as an input when the JTAG TAP controllers are not needed or when an external driver does not violate the Slave Select Timing as defined in Section 44.2.2, "Slave Select Timing," on page 580.

GPIO166 is used for the UART_BOOT_STRAP option (see Section 42.4, "UART Boot Strap Option," on page 568.

See also Section 7.6.2, "Strap Options," on page 124.

3.9 Package Outline

FIGURE 3-2: 169-Ball LFBGA 11 mm x 11 mm x 0.8 mm Package Outline



© 2012-2018 Microchip Technology Inc.

4.0 BUS HIERARCHY

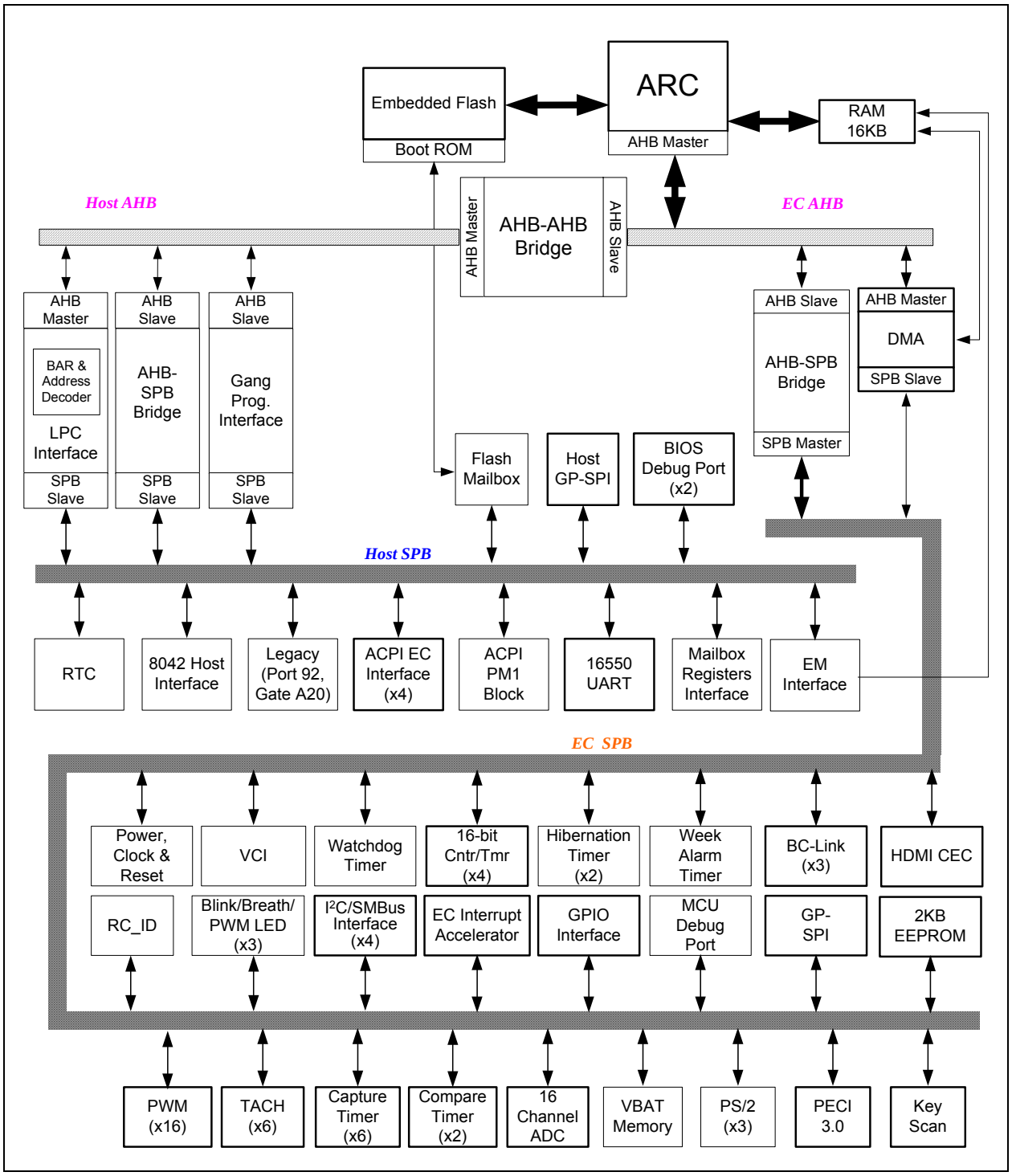
4.1 General Description

All devices in the MEC1632 are maintained in a common address space. All communication with on-chip functions is through registers that have addresses in this common address space. The ARC Embedded Controller (EC) can reference all devices through the address space, while the Host can only reference a subset.

4.2 Block Diagram

FIGURE 4-1: MEC1632 Bus Hierarchy on page 56 shows, in graphic form, the inter connectivity of devices on the MEC1632, including the EC, the principal lower buses and most of the peripherals.

FIGURE 4-1: MEC1632 BUS HIERARCHY



© 2012-2018 Microchip Technology Inc.

4.3 Address Space

The ARC EC has a 24-bit address space. Addresses in the lower half of the range, 0h through 7F_FFFFh, can be used for both instruction access and data access. The address range 0h through 4_FFFFh (which includes the 192KB Flash Memory Array is mapped to the Embedded Flash Subsystem. The address range 10_0000h through 10_0FFFh is mapped to the Boot ROM. Addresses in the range 80_0000h through 80_3FFFh are mapped to the Closely Coupled Data Memory. These memories are shown in Figure 4-2. Software can change the address mapping so that the Closely Coupled Data Memory is mapped to 6_0000h through 6_0FFFh, where it can be used for both instructions and data, and the Embedded Flash becomes accessible only through a register interface. Addresses greater or equal to 80_1000h are propagated through the AHB interface on the ARC processor. References, by either the EC or the Host via the LPC bus, to addresses that are not mapped to any device register or memory will cause a bus error; see Section 4.4.3, "AHB Bus Errors," on page 62.

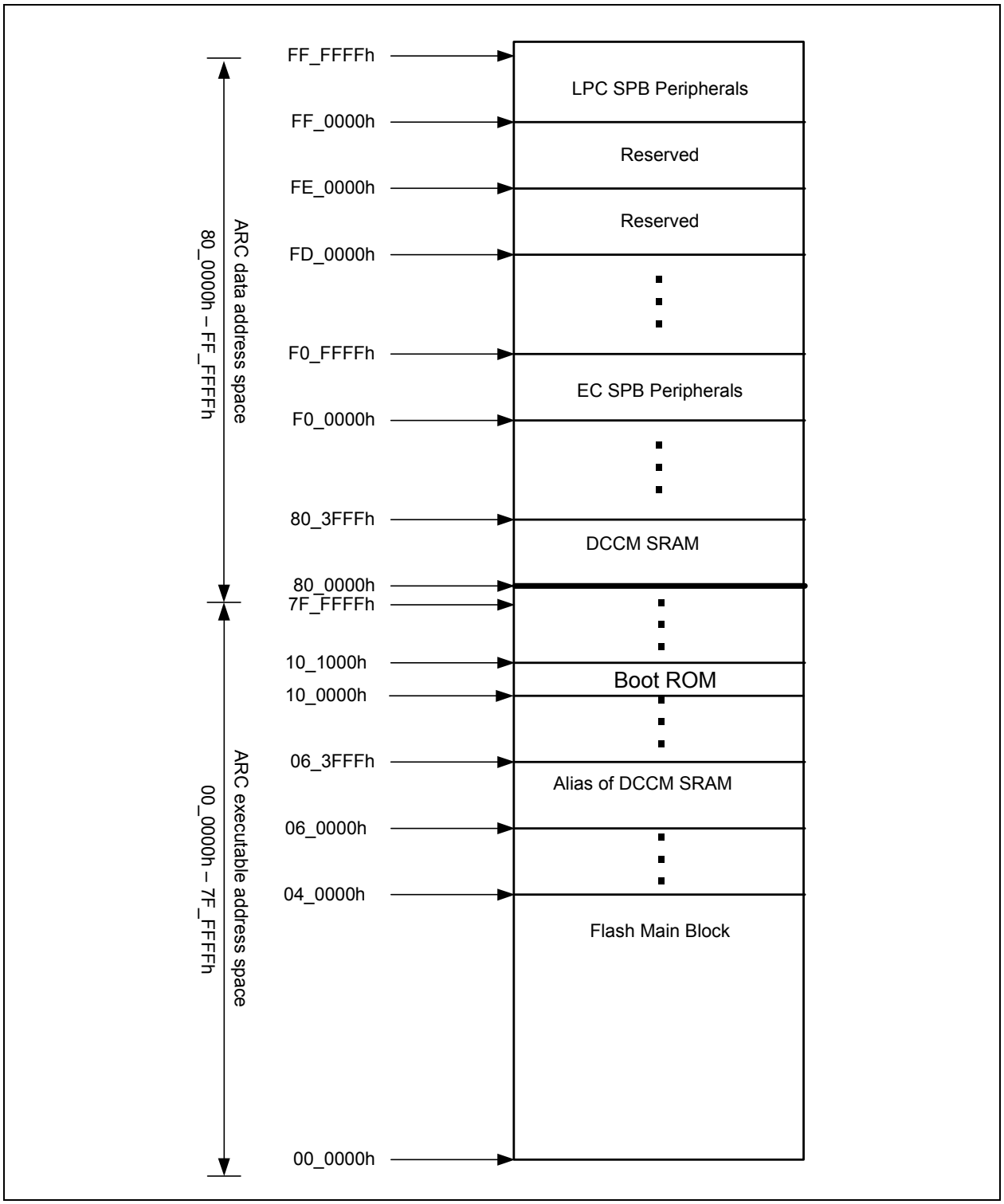
4.3.1 ARC ADDRESS SPACE

The ARC Address Space is illustrated in Figure 4-2, "MEC1632 EC Memory Map" & Figure 4-1, "MEC1632 Bus Hierarchy". EC Instruction Memory occupies the lower half of the address space, 00_0000h through 7F_FFFFh. The ARC processor can only execute instructions that are located in the Instruction Memory portion of the address space. Only a portion of the Instruction space is populated.

The upper half of the address space is used solely for data references. The region contains a general-purpose EC Data Memory SRAM as well as the address space of the two SPB peripheral buses.

The contents of the Instruction memory can be read and written by the ARC processor through regular load and store instructions. However, there may be a one cycle instruction fetch penalty whenever a data load or store instruction to Instruction memory is executed.

FIGURE 4-2: MEC1632 EC MEMORY MAP



© 2012-2018 Microchip Technology Inc.

4.3.2 AHB ADDRESS SPACE

The components on the AHB subsystem and the two SPB bridges define a set of addresses that are accessible by the EC. This address space is shown in Table 4-1, "MEC1632 Peripheral Address Space". As shown in the table, the Host can access much of the address space, but not all.

TABLE 4-1: MEC1632 PERIPHERAL ADDRESS SPACE

Address Range	Device	Accessible by EC	Accessible by Host
F0_0000h - F0_FFFFh	ARC-only SPB Bridge	Yes	No
FD_0000h - FE_FFFFh	Reserved	No	No
FF_0000h - FF_BFFFh and FF_D000h - FF_FFFFh	LPC SPB Bridge	Yes	Yes (limited by LPC interface map)

The address range of FD_0000h through FE_FFFFh is reserved in MEC1632.

The 64KB address space of an SPB Bridge is divided into 1KB Frames. Peripherals are grouped into Logical Devices; each Logical Device corresponds to a 1KB Frame. Logical Devices addressable by the Host are listed in Table 4-2, "Host Logical Devices in the MEC1632". Logical device numbers not listed in Table 4-2 are reserved.

The 1KB Address Frame of a Host Logical Device is divided into four subregions, as described in Section 4.4.2, "Address Framing," on page 61.

TABLE 4-2: HOST LOGICAL DEVICES IN THE MEC1632

Logical Device Number	Logical Devices	AHB Address of Frame	Runtime Registers	Configuration Registers	EC-only Registers
0h	Mailbox Interface	FF_0000h	yes	yes	yes
1h	Keyboard Controller (8042)	FF_0400h	yes	yes	no
2h	ACPI EC Channel 0	FF_0800h	yes	yes	no
3h	ACPI EC Channel 1	FF_0C00h	yes	yes	no
4h	ACPI EC Channel 2	FF_1000h	yes	yes	no
5h	ACPI EC Channel 3	FF_1400h	yes	yes	no
6h	ACPI PM1	FF_1800h	yes	yes	no
7h	UART	FF_1C00h	yes	yes	no
8h	Legacy (Fast KB)	FF_2000h	yes	yes	no
Ch	LPC Interface	FF_3000h	no	yes	yes
Dh	Reserved	FF_3400h	no	yes	no
Eh	Embedded Flash Interface	FF_3800h	yes	yes	yes
Fh	Flash SPI	FF_3C00h	yes	yes	no
10h	EM Interface	FF_4000h	yes	yes	yes
11h	Real Time Clock	FF_4400h	yes	no	yes
20h	BIOS Debug Port 0	FF_8000h	yes	yes	no
21h	BIOS Debug Port 1	FF_8400h	yes	yes	yes
3Fh	Global Configuration	FF_FC00h	no	no	yes

The 64KB address space of the ARC-only SPB Bridge is divided into 1KB Frames. Peripherals are grouped into Logical Devices; each Logical Device corresponds to a 1KB Frame. These EC Logical Devices addressable by the ARC-only are listed in Table 4-3, "EC Logical Devices on MEC1632". Multiple instantiations of the same block are in a single 1KB Frame. Each instantiation is separated by 128 bytes.

© 2012-2018 Microchip Technology Inc.

Note 4-1 All VBAT powered registers are in a Single 1KB Frame separated by 128 bytes (see Table 4-3, Logical Device Number 33h.)

Note that the ARC-only SPB Bridge address space uses the terminology of Logical Device; however, there is no host access to these blocks.

TABLE 4-3: EC LOGICAL DEVICES ON MEC1632

Logical Device Number	AHB Address for Base of Frame	Logical Devices	Notes
0h	F0_0000h	Hibernation Timer	
1h	F0_0400h	Watchdog Timer	
2h	**F0_0800h**	**Input Capture and Compare Timer**	
3h	F0_0C00h	16-bit Timer	
4h	F0_1000h	RC ID	
5h	F0_1400h	BC Bus Master	
6h	F0_1800h	SMBus	
7h	F0_1C00h	EC GP-SPI	
8h	**F0_2000h**	**Keyscan Interface**	
9h	**F0_2400h**	**DMA**	
Ah	F0_2800h	SPI Flash Read	
Bh	F0_2C00h	EEPROM	
Ch - Fh	F0_3000h - F0_3FFFh	Reserved	
10h	F0_4000h	CEC Interface Controller	
11h	F0_4400h	Reserved	
12h	F0_4800h	Reserved	
14h - 15h	F0_5000h - F0_57FFh	Reserved	
16h	F0_5800h	PWM(7-0)	
17h	F0_5C00h	**PWM(15-8)**	Note 4-2
18h	**F0_6000h**	TACH	
19h	F0_6400h	**PECI**	
1Ah	**F0_6800h**	**ADC**	
1Bh - 20h	F0_6C00h - F0_83FFh	Reserved	
21h	F0_8400h	LED	
22h	F0_8800h	PS/2	
23h	F0_8C00h	MCU Debug Port	
24h - 2Fh	F0_9000h - F0_BCFFh	Reserved	
30h	F0_C000h	EC Interrupt Aggregator	
31h	F0_C400h	GPIOs	
32h	F0_C800h	Power, Clock & Reset (VTR PWR'ed)	
33h	F0_CC00h	Power, Clock & Reset (VBAT PWR'ed)	
	F0_CC80h	Week Alarm Timer	
	F0_CD00h	VBAT Backed Memory	
34h	F0_D000h	VBAT-Powered Control Interface	
35h - 3Dh	F0_D400h - F0_F7FFh	Reserved	
3Eh	F0_F800h	Reserved	
3Fh	F0_FC00h	System Registers	

Note 4-2 PWM register addressing within Logical device 17h, PWM(15-8), is similar to the addressing within the PWM(7-0) logical address space.

© 2012-2018 Microchip Technology Inc.

4.4 AHB Buses

Addresses and internal buses in the MEC1632 are compatible with ARM Limited's *Advanced Microprocessor Bus Architecture* (AMBA), as specified in *AMBA™ Specification (Rev 2.0),* 1999.

As seen in FIGURE 4-1: MEC1632 Bus Hierarchy on page 56, there are two separate AHB buses, the EC AHB and the Host AHB. The first has one master, the EC, and two slaves, while the second has two masters, the LPC interface and the AHB-AHB bridge, and four slaves. The bus connections are summarized in Table 4-4, "MEC1632 AHB Buses" and can be seen in Figure 4-1.

TABLE 4-4: MEC1632 AHB BUSES

AHB Bus	Master Interfaces	Slave Interfaces
EC AHB	EC	EC SPB bridge AHB-AHB bridge
Host AHB	LPC Interface AHB-AHB bridge	LPC SPB bridge SPI bridge

The AHB-to-AHB bridge is a one-way device: addresses generated on the EC AHB can be propagated to the Host AHB bus, but addresses on the Host AHB bus cannot be propagated to the EC AHB bus. This is the reason that the Host is restricted from accessing the address range F0_0000h to F0_FFFFh (the address range of the EC SPB bridge). The bridge maps address from FD_0000h through FF_FFFFh. Multi-word block transfers on the EC AHB are converted by the bridge into a series of sequential 4-byte transfers.

Both the EC and the LPC interface can have at most one outstanding AHB bus request at one time. On the Host AHB, the LPC interface always has priority over the AHB-AHB bridge. Both AHB buses support byte, halfword and word AHB transfers. The only Burst mode supported on the EC AHB is an incrementing burst of 4 beats of 4 bytes per beat. Burst mode is not supported on the Host AHB. AHB bus locking or early burst termination are not supported.

4.4.1 BUS CLOCKING

The Host AHB runs at the MEC1632 system clock rate (MCLK). The bus clock and the bus arbiter will be shut down when there are no transactions active on the bus. The bus clock and arbiter will be restarted as soon as an address is acquired from the LPC bus, or when an EC AHB bus transaction is mapped, via the AHB-to-AHB bridge, to the address space of the Host AHB. The EC AHB clock can be programmed to run at any of the available rates as defined by Block Sleep Enable Registers (see Section 7.8.4, "Block Sleep Enable Registers," on page 136). The EC bus clock is shut down when the EC is idle.

If the Host AHB clock is idle when an LPC transaction arrives at the LPC interface, the LPC interface will restart the AHB bus clock and arbiter early enough so that as soon as an I/O address is translated to an AHB address the I/O transaction can be placed on the AHB without delay. If the I/O address is not claimed by the MEC1632 then the LPC interface will drop its bus request. If the EC is not requesting the Host AHB at the same time, the Host AHB bus clock will again shut down.

4.4.2 ADDRESS FRAMING

The EC can directly address all peripherals on the MEC1632. The Host, by contrast, is restricted in what it can address. The Host accesses the MEC1632 through the LPC bus, using I/O cycles, Memory Cycles and Firmware Hub cycles (see Section 6.0, "Host Interface," on page 79). These cycles are mapped into the MEC1632 address space accessible by the LPC interface.

The mapping function forces some of the address bits to preset values, as shown in Table 4-5, "LPC to MEC1632 Address Mapping". This mapping has the effect of creating four contiguous 256-byte regions: LPC I/O, EC-only, LPC DMA and LPC Configuration. Together, the four regions create a 1024 byte "frame" for each logical device that is accessible to the Host. There is therefore a maximum of 64 Logical Devices, each with a 1KB frame.

LPC I/O cycles for Runtime Registers are mapped into the first 256 bytes of the 1024 byte frame. Configuration Registers, which are accessed through a Configuration portal typically located at addresses 2Eh and 2Fh in the LPC I/O space, are restricted to the highest 256 bytes of the 1024 byte frame.

DMA FIFO addresses are restricted to offsets between 200h and 2FFh within a Logical Device frame. In addition, DMA FIFO addresses are restricted to 32-bit aligned addresses, that is, address bits [1:0] are both 0. The low 10 bits of a DMA FIFO are thus in the form '10xxxxxx00b'.

© 2012-2018 Microchip Technology Inc.

LPC Firmware Hub cycles and LPC Memory cycles are mapped into the address range 80_0000h to FF_FFFFh. This range is mapped into the off-chip SPI Flash memory.

Because the EC does not require the mapping mechanism required for translating LPC Runtime Registers, Configuration Registers and DMA channels, it accesses each 1KB frame uniformly. All Logical Devices located on the EC-only AHB (those in the address range F0_0000h through F0_FFFFh) have a flat, 1KB frame.

For details on LPC address mapping to the MEC1632 address space, see Section 6.0, "Host Interface," on page 79.

TABLE 4-5: LPC TO MEC1632 ADDRESS MAPPING

Type of Access	Address Bit Mapping
LPC I/O Access Runtime Registers	Address bits[9:8] = 00b Address bits[23:10] set from map
No LPC Access EC-only registers	Address bits[9:8] = 01b
LPC I/O Access through Configuration Access Port Configuration Registers	Address bits[9:8] = 11b Address bits[23:10] set from map
LPC DMA Access	Address bits[9:8] = 10b Address bits[23:10] set from map

4.4.3 AHB BUS ERRORS

AHB bus requests by both the Host, through the LPC bus, and the EC, can be terminated with an AHB bus error. The handling of bus errors by the EC is described in the *ARCompact™ Instruction Set Architecture: Programmer's Reference*.

Bus errors may be caused by:

- EC I/O requests that lie outside the range of the Data Closely Coupled Memory, the EC SPB or the AHB-AHB bridge
- I/O requests to either the Host SPB or the EC SPB that map to a non-existent Logical Device
- I/O requests to an invalid register address within a valid Logical Device on either the EC SPB or the Host SPB

4.5 Peripheral Buses (SPB)

The Peripheral Bus (SPB) is a byte-addressable bus with a 16-bit address space and a 32-bit data path. All accesses must be aligned to the data-size boundaries. The SPB supports 32-bit, 16-bit and 8-bit accesses. All peripheral accesses are 32-bits wide, so that if a peripheral cannot transfer more than 8 bits of a register in one I/O access, register addresses should all be on 32-bit boundaries even though the upper bits (bits 31 through 8) are always zero. The SPB will not assemble a 32-bit word out of multiple 8-bit accesses.

SPB transfers take two cycles, so that a read or write on the AHB to a register on the SPB will take a total of three cycles. The SPB contains an 8-bit word address, four byte lane strobes and up to 64 logical device select strobes. Data on the SPB read and write buses are always 32-bit aligned, with the byte strobes indicating which byte lane or lanes should be active during a transaction. An SPB peripheral must steer bytes, based on the byte strobes, to the correct byte lane.

There are two SPB bridges forwarded to two SPB (busses):

LPC SPB

The LPC SPB address range is FF_0000h - FF_BFFFh and FF_C400h - FF_FFFFh.

EC SPB

The EC SPB address range is F0_0000h through F0_FFFFh

© 2012-2018 Microchip Technology Inc.

5.0 LOGICAL DEVICE CONFIGURATION

5.1 Description

The Configuration of the MEC1632 is very flexible and is based on the configuration architecture implemented in typical Plug-and-Play components.

The MEC1632 is designed for motherboard designs in which the resources required by their components are known. With its flexible resource allocation architecture, the MEC1632 allows the BIOS to assign resources at POST.

5.2 Location of Configuration Registers

Configuration Registers for Logical Devices accessible by the Host are located on the LPC SPB in the address range FF_0000h through FF_FFFFh. All Configuration Registers are located at addresses where address bits 9 and 8 are both '1b' (that is, at offsets 300h through 3FFh from the base of a 1KB address frame). Configuration registers are accessible by the Embedded Controller with 8-bit, 16-bit or 32-bit accesses. The Host can access the registers only with 8-bit accesses.

The Configuration Registers for the LPC Logical Device are located on the LPC SPB in the address range FF_3300h through FF_33F0h. The Global Configuration Registers are located within the Global Configuration Logical Device.

5.3 Logical Devices

Logical devices described in this section are peripherals that are located on the MEC1632 and are accessible to the Host over the LPC bus.

Each logical device on the MEC1632 can have a set of Runtime Register and a set of Configuration Registers. The distinction between Runtime and Configuration registers is that the Host can access Runtime Registers by a direct I/O address, while it can only access Configuration Registers through a configuration port. The Embedded Controller (EC) can access all Configuration Registers and all Runtime Registers directly. The Logical Device Numbers for the Logical Devices resident in the MEC1632 are listed in Table 5-14, "MEC1632 Configuration Register Map," on page 75.

TABLE 5-1: LOGICAL DEVICES

Logical Device Number	Logical Device	Logical Device CR Map on Table 5-14
0h	Mailbox Interface	on page 75
1h	Keyboard Controller (8042)	on page 75
2h	ACPI EC Channel 0	on page 75
3h	ACPI EC Channel 1	on page 75
4h	ACPI EC Channel 2	on page 75
5h	ACPI EC Channel 3	on page 75
6h	ACPI PM1	on page 75
7h	UART	on page 75
8h	Legacy (Fast KB)	on page 75
Ch	LPC Interface	on page 75
Eh	Embedded Flash Interface	on page 77
Fh	Flash SPI	on page 77
10h	EM Interface	on page 77
11h	Real Time Clock	on page 77
20h	BIOS Debug Port 0	on page 77
21h	BIOS Debug Port 1	on page 77
3Fh	Global Configuration	on page 77

All Configuration and Runtime Registers in the MEC1632 have an assigned AHB address between FF 0000h and FF FFFFh. Unless indicated otherwise, the EC can issue reads and writes to any register in that AHB address range. The EC can access 8-bit registers with 8-bit reads and writes, 16-bit registers with either 8-bit or 16-bit reads and writes and 32-bit registers with 8-bit, 16-bit and 32-bit reads and writes.

© 2012-2018 Microchip Technology Inc.

The Host can only access a subset of the AHB address space, and within that space all registers are treated as 8-bit registers, although a register may be implemented as a 32-bit register and accessible to the EC as a 32-bit register. The Host accesses registers in the FF_0000h through FF_FFFFh range through LPC I/O cycles. I/O cycles are mapped according to rules described in Section 5.5, on page 65 and Section 5.6, on page 66.

5.4 Registers

The Host Interface has its own Logical Device Number and Base Address as indicated in Table 5-2. The Host LPC I/O addresses for the Logical Device Configuration are selected via a Base Address Register. LPC access to configuration registers is through the Host Access Configuration Port

The Logical Device Configuration also has a Global Configuration block which has a separate Logical Device Number and Base Address Register as indicated in Table 5-2. The Base Address Register for the Global Configuration has only one writable bit, the Valid Bit, since the only I/O accessible Register has a fixed address.

Table 5-3 is a register summary for the LPC Interface block and Table 5-15, “Chip-Level (Global) Control/Configuration Registers,” on page 78 is a register summary for the Global Configuration block.

TABLE 5-2: Host Interface BASE ADDRESS TABLE

Host Interface Blocks	LDN from (Table 4-2 on page 59)	AHB Base Address
LPC Interface	Ch	FF_3000h
Global Configuration	3Fh	FF_FC00h

Note: The Host LPC I/O addresses for this instance are selected via a Base Address Register (see Section 5.6.2, on page 66). LPC access to configuration registers is through the Host Access Configuration Port (see Section 5.5.1, on page 65).

Table 5-3 is a register summary for the Host Access Port block. The LPC I/O address for each Run-Time Register is described below as an offset from its Base Address Register. Each EC address is indicated as an SPB Offset from its AHB base address. Each Configuration register access through the Host Access Port address via its LDN indicated in Table 5-2 on page 64 and its Host Access Port index which is described as “Host Config Index” in the tables below.

TABLE 5-3: Host Access Port REGISTER SUMMARY

	Host I/O Access			EC Interface			Notes
Port Name	**Host I/O Index**	**SPB Offset**	**Host Type**	**SPB Offset**	**Byte Lane**	**EC Type**	
CONFIG PORT	00h	00h	W				
INDEX PORT	00h	00h	R/W				
DATA PORT	01h	01h	R/W				

Note: The EC does not have access to the Host Access Port; however, the EC can access registers with AHB addresses.

© 2012-2018 Microchip Technology Inc.

5.5 Configuration Registers

5.5.1 HOST ACCESS PORT

The Host can access Configuration Registers through a port described in Section 5.5.2, on page 65. Host accesses are limited to 8 bits. There are 48 8-bit Global Configuration Registers (at offsets 00h through 2Fh), plus up to 208 8-bit registers associated with each Logical Device. The Logical Device is selected with the Logical Device Number Register (Global Configuration Register 07h). The INDEX PORT is used to select a specific logical device register. These registers are then accessed through the DATA PORT. The Logical Device registers are accessible only when the device is in the Configuration State.

Only two states are defined (Run and Configuration). In the Run State, the chip will always be ready to enter the Configuration State.

The desired configuration registers are accessed in two steps:

a) Write the index of the Logical Device Number Configuration Register (i.e., 07h) to the INDEX PORT and then write the number of the desired logical device to the DATA PORT

b) Write the address of the desired configuration register within the logical device to the INDEX PORT and then write or read the configuration register through the DATA PORT.

Note:

- If accessing the Global Configuration Registers, step (a) is not required.
- Any write to an undefined or reserved Configuration register is terminated normally on the LPC bus without any modification of state in the MEC1632. Any read to an undefined or reserved Configuration register returns FFh.

5.5.2 PRIMARY CONFIGURATION ADDRESS DECODER

The logical are configured through three Configuration Access Ports (CONFIG, INDEX and DATA). The BIOS uses these ports to initialize the logical devices at POST (Table 5-4).

The Base Address of the Configuration Access Ports is determined by the BAR that corresponds to Logical Device Ch, the LPC Interface. This is the first BAR in the table, at AHB address FF_3360h. The Configuration Access Port BAR is unique in that an LPC I/O access that matches this BAR does not directly generate an AHB read or write. Instead, the Device and Frame values in the BAR indicates that the LPC I/O should be handled locally in the LPC Logical Device. The Configuration map will issue an AHB read or write, the results of which will be used to complete the LPC access.

TABLE 5-4: MEC1632 CONFIGURATION ACCESS PORTS

Port Name	Relative Address	Type	Port Name
CONFIG PORT	Configuration Access Ports Base Address + 0	Write	CONFIG PORT
INDEX PORT	Configuration Access Ports Base Address + 0	Read/Write	INDEX PORT
DATA PORT	Configuration Access Ports Base Address + 1		DATA PORT

5.5.2.1 Entering the Configuration State

The INDEX and DATA ports are effective only when the chip is in the Configuration State. The device enters the Configuration State when the Config Entry Key is successfully written to the CONFIG PORT.

Config Entry Key = < 55h>

5.5.2.2 Exiting the Configuration State

The device exits the Configuration State when the following Config Exit Key is successfully written to the CONFIG PORT address.

Config Exit Key = < AAh>

5.5.2.3 Read Accessing Configuration Port

The data read from the Configuration Port is undefined when not in the Configuration State. Writing the Config Entry Key puts the chip in the Configuration State. Once in the Configuration State, reading the Configuration Port will return the last value written to the Configuration Index. If no value was written the Configuration Port reads 00h.

5.5.3 CONFIGURATION SEQUENCE EXAMPLE

To program the configuration registers, the following sequence must be followed:

1. Enter Configuration State
2. Program the Configuration Registers
3. Exit Configuration State.

The following is an example of a configuration program in Intel 8086 assembly language.

```
;----------------------------.
; ENTER CONFIGURATION STATE  |
;----------------------------'
MOV DX,CONFIG_PORT_BASE_ADDRESS
MOV AX,055H ; Config Entry Key
OUT DX,AL
;----------------------------.
; CONFIGURE BASE ADDRESS,    |
; LOGICAL DEVICE 8           |
;----------------------------'
MOV DX,CONFIG_PORT_BASE_ADDRESS
MOV AL,07H
OUT DX,AL ; Point to LD# Config Reg
MOV DX,CONFIG_PORT_BASE_ADDRESS+1
MOV AL, 08H
OUT DX,AL ; Point to Logical Device 8
;
MOV DX,CONFIG_PORT_BASE_ADDRESS
MOV AL,60H
OUT DX,AL ; Point to BASE ADDRESS REGISTER
MOV DX,CONFIG_PORT_BASE_ADDRESS+1
MOV AL,02H
OUT DX,AL ; Update BASE ADDRESS REGISTER
;----------------------------.
; EXIT CONFIGURATION STATE   |
;----------------------------'
MOV DX,CONFIG_PORT_BASE_ADDRESS
MOV AX,0AAH ; Config Exit Key
OUT DX,AL.
```

5.5.4 CONFIGURATION REGISTER ADDRESS MAPPING

The INDEX PORT defines 256 bytes for configuration. The first 48 of these bytes are Global Configuration registers, which reside in the first 48 bytes of the Configuration part of the address frame for Logical Device 3Fh. Values of INDEX greater than 48 map into registers that are specific to the Logical Device specified in the Global Configuration Logical Device Number Register 7h. These registers reside in upper 20 bytes of the Logical Device address frame. See Section 5.9.2, on page 74 for details.

5.6 Configuring Runtime Register Addresses

5.6.1 RUNTIME REGISTERS

Runtime Registers are registers that are accessible to the Host within the Host I/O address space. These Host I/O accesses are all mapped into the MEC1632 AHB address space onto devices located on the LPC SPB. Runtime registers all reside within the first 256 bytes of a 1KB Logical Device address frame. The Host accesses these registers with 8-bit LPC I/O accesses. Each 8-bit I/O address is mapped into an 8-bit address in the AHB address space, so the first 256 bytes of the Logical Device frame can accommodate 256 LPC Runtime Registers per Logical Device. The Host I/O addresses are determined by a block of Base Address Registers located in the LPC Logical Device. The Embedded Controller can access all the Runtime Registers as well, using loads and stores to full AHB addresses.

5.6.2 BASE ADDRESS REGISTERS

Each Logical Device has a Base Address Register (BAR). These BARs are located in blocks of Configuration Registers in Logical Device 0Ch, in the AHB address range FF_3360h through FF_3384h. On every LPC bus I/O access all Base Address Registers are checked in parallel and if any matches the LPC I/O address the MEC1632 claims the bus cycle.

© 2012-2018 Microchip Technology Inc.

Note: Software should that insure that no two BARs map the same LPC I/O address. If two BARs do map to the same address, the BAR_Conflict bit in the Host Bus Error Register is set when an LPC access targeting the BARConflict address. An EC interrupt can be generated.

Each BAR is 32 bits wide. The format of each BAR is summarized in Table 5-5, "Base Address Register Format". An LPC I/O request is translated by the BAR into an 8-bit read or write transaction on the AHB bus. The 16-bit LPC I/O address is translated into a 24-bit AHB address

The Base Address Register Table is itself part of the AHB address space. It resides in the Configuration quadrant of Logical Device Ch, the LPC Interface.

TABLE 5-5: BASE ADDRESS REGISTER FORMAT

BYTE3 BIT	D31	D30	D29	D28	D27	D26	D25	D24
BIT NAME	LPC Host Address, most significant bits							
BYTE2 BIT	D23	D22	D21	D20	D19	D18	D17	D16
BIT NAME	LPC Host Address, least significant bits							
BYTE1 BIT	D15	D14	D13	D12	D11	D10	D9	D8
BIT NAME	Valid	Reserved	Frame					
BYTE0 BIT	D7	D6	D5	D4	D3	D2	D1	D0
BIT NAME	Reserved	Mask						

MASK

These 7 bits are used to mask off address bits in the address match between an LPC I/O address and the Host Address field of the BARs, as described in Section 5.6.3, "Mapping LPC I/O Addresses". A block of up to 128 8-bit registers can be assigned to one base address.

FRAME

These 6 bits are used to specify a logical device frame number within a bus. This field is multiplied by 400h to provide the frame address within the peripheral bus address. In the MEC1632 Frame values for frames corresponding to logical devices that are not present on the MEC1632 are invalid.

VALID

If this bit is 1, the BAR is valid and will participate in LPC matches. If it is 0 this BAR is ignored

HOST_ADDRESS

These 16 bits are used to match LPC I/O addresses

5.6.3 MAPPING LPC I/O ADDRESSES

A Base Address Register will match an LPC I/O address, and thus the MEC1632 will claim the LPC bus cycle, if the following relation holds:

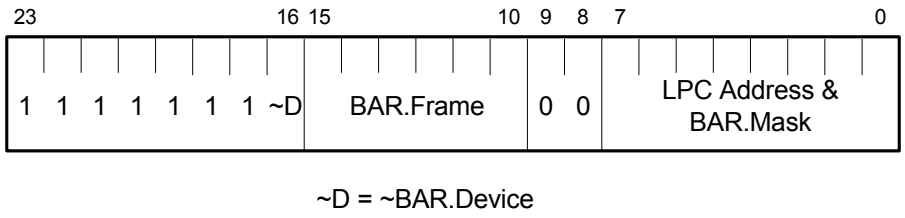
(LPC Address & ~BAR.MASK) == (BAR.LPC_Address & ~BAR.MASK) && (BAR.Valid == 1)

If one of the BARs match, the LPC cycle will be claimed and the LPC request will be translated to an AHB address according to the following formula:

AHB Address = FF_000h | (BAR.Frame<<10) | (LPC_Address & BAR.MASK)

The formula is illustrated in Figure 5-1, "LPC BAR Mapping".

FIGURE 5-1: LPC BAR MAPPING



When matching LPC I/O addresses, the MEC1632 ignores address bits that correspond to '1b' bits in the MASK field. When forming the AHB address from the LPC I/O address, the LPC I/O address bits that correspond to the '1b' bits in the MASK are passed through to the AHB address. For example, the Keyboard Controller (8042) Base Address Register has 60h in the LPC Address field, the Frame field is 01h, and the MASK field is 04h. Because of the single '1b' bit in MASK, the BAR will match LPC I/O patterns in the form '00000000011000b', so both 60h and 64h will be matched and claimed by the MEC1632. When forming the AHB address, the Frame number 01h will be put in bit positions 15 through 10 and concatenated with FF_0000h, to form the Keyboard Controller Frame base address of FF_0400h. If the Host reads from address 60h, a read access will be initiated to AHB address FF_0400h (the base address OR'd with bit 2 from 60h). If the Host reads from address 64h, a read access will be initiated to AHB address FF_0404h (again, bit 2 from the LPC I/O address is OR'd into the AHB address).

As another example, if a standard 16550 UART was located at LPC I/O address 238h, then the UART Receive buffer would appear at address 238h and the Line Status register at 23Dh. If the BAR for the UART was set to 0238_8047h, then the UART will be matched at I/O address 238h, the UART is located in Logical Device 6h, the UART is enabled and the UART device includes 8 registers. The Receive buffer would map to AHB address FF_1800h and the Line Status register would map to AHB address FF_1805h.

5.6.4 BASE ADDRESS REGISTER TABLE

Table 5-6, "Base Address Registers Default Values", lists the Base Address Registers for all logical devices on the MEC1632. The BAR EC Offsets are relative to the base address of the LPC Logical Device, which is located at AHB address FF_3000h. The columns to the right of the heavy line show the field definitions for the default values listed in the column labeled "Reset Default". Shaded fields in Table 5-6 are read-only.

The EC can read and write the BAR table entries with 32-bit, 16-bit or 8-bit accesses. The Host accesses the BAR table using 8-bit reads and writes.

Note: The BAR at offset 360h associated with the LPC Interface is different in that LPC I/O accesses that are claimed by this BAR does not translate directly into a AHB addresses. Addresses that are claimed by this BAR, the Configuration Port BAR, are used to manage Configuration Registers. These accesses are described in Section 5.5, on page 65. This BAR is also special in that a byte write to offset 362h (bits[7:0] of the LPC Host Address field) does not directly write into the BAR. Instead, the byte is held in a buffer. A byte write to offset 363h will both write the byte at offset 363h and will copy the buffer into offset 362h. This is done to insure that the 16-bit LPC Host Address field of the Configuration Port BAR is always completely updated in one cycle.

The shaded LPC I/O Address, VALID, FRAME, MASK fields are read-only Table 5-6. The unshaded fields has read/write access.

© 2012-2018 Microchip Technology Inc.

TABLE 5-6: BASE ADDRESS REGISTERS DEFAULT VALUES

Bar EC Offset	LPC Offset	Reset Default	LPC I/O Address	Valid	Frame	Mask	Description
360h	60h	002E_0C01h	002Eh	0	C	1	Logical Device 0Ch: LPC Interface (Configuration Port)
364h	64h	0000_0001h	0000h	0	0	1	Logical Device 00h: Mailbox Register I/F
368h	68h	0060_0104h	0000h	0	1	4	Logical Device 01h: Keyboard Controller (8042 Interface)
36Ch	6Ch	0062_0204h	0062h	0	2	4	Logical Device 02h: ACPI EC Interface 0
370h	70h	0062_0307h	0062h	0	3	7	Logical Device 03h: ACPI EC Interface 1
374h	74h	0062_0407h	0062h	0	4	7	Logical Device 04h: ACPI EC Interface 2
378h	78h	0062_0507h	0062h	0	5	7	Logical Device 05h: ACPI EC Interface 3
37Ch	7Ch	0000_0607h	0000h	0	6	7	Logical Device 06h: ACPI PM1 Interface
380h	80h	0000_0707h	0000h	0	7	7	Logical Device 07h: UART
384h	84h	0092_0800h	0092h	0	8	0	Logical Device 08h: Legacy (GATEA20) I/F
388h	88h	0000_0E04h	0000h	0	E	4	Logical Device 0Eh: Embedded Flash Interface
38Ch	8Ch	0000_0F07h	0000h	0	F	07h	Logical Device 0Fh: GPSPI for Flash Interface
390h	90h	0000_100Fh	0000h	0	10h	Fh	Logical Device 10h. EM Interface
398h	98h	0000_2000h	0000h	0	20h	0	Logical Device 20h. BIOS Debug Port 0
39Ch	9Ch	0000_2100h	0000h	0	21h	0	Logical Device 21h. BIOS Debug Port 1
3A0h	A0h	0000_111Fh	0000h	0	11h	1Fh	Logical Device 11h. Real Time Clock
3B0h-3ECh	B0h - ECh	0000_0000h	0000h	0	0	0	Reserved

Note 5-1 In order to avoid address conflict with the Keyboard Controller (8042) LDN 1h at legacy at LPC I/O address 60h/64h, only ACPI EC Channel 0 LDN 2h at AHB base address FF_0800h should be located at the legacy LPC I/O address 62h/66h. When operating both the Keyboard Controller (8042) and ACPI EC Channel 0 at legacy LPC I/O addresses 60h/62h/64h/66h, the Mask value in ACPI EC Channel 0 BAR should be programmed to 4h and the ACPI EC Channel 0, Byte Control EC-Register should remain 0. To utilize the ACPI EC Channel 0 in Non Legacy Operation the ACPI EC Channel 0 BAR must be placed on LPC I/O 8 byte boundary. Only bits[3:0] in the Mask field of ACPI EC Channel 0 BAR are programmable, bits[6:4] are read-only '000'.

Note 5-2 The ACPI EC Channel 1**,** ACPI EC Channel 2 and ACPI EC Channel 3 **(**LDN 3h, 4h and 5h) can not be located at the legacy LPC I/O address 62h/66h. The ACPI EC Channel 1, ACPI EC Channel 2 and ACPI EC Channel 3 BAR's must be placed on LPC I/O 8 byte boundaries.

© 2012-2018 Microchip Technology Inc.

5.6.5 MEMORY BASE ADDRESS REGISTERS

Each Logical Device has a Memory Base Address Register (M-BAR). These M-BARs are located in blocks of Configuration Registers in Logical Device 0Ch, in the AHB address range FF_33C0h through FF_33FFh. On every LPC bus Memory access all Base Address Registers are checked in parallel and if any matches the LPC memory address the MEC1632 claims the bus cycle.

Note: Software should that insure that no two BARs map the same LPC memory address. If two BARs do map to the same address, the BAR_Conflict bit in the Host Bus Error Register is set when an LPC access targeting the BARConflict address. An EC interrupt can be generated.

Each M-BAR is 48 bits wide. The format of each M-BAR is summarized in Table 5-7, "Memory Base Address Register Format". An LPC memory request is translated by the M-BAR into an 8-bit read or write transaction on the AHB bus. The 32-bit LPC memory address is translated into a 24-bit AHB address

The Base Address Register Table is itself part of the AHB address space. It resides in the Configuration quadrant of Logical Device Ch, the LPC Interface.

TABLE 5-7: MEMORY BASE ADDRESS REGISTER FORMAT

BYTE5 BIT	D47	D46	D45	D44	D43	D42	D41	D40
BIT NAME	LPC Host Address Bits[31:24]							
BYTE4 BIT	D39	D38	D37	D36	D35	D34	D33	D32
BIT NAME	LPC Host Address Bits[23:16]							
BYTE3 BIT	D31	D30	D29	D28	D27	D26	D25	D24
BIT NAME	LPC Host Address Bits[15:08]							
BYTE2 BIT	D23	D22	D21	D20	D19	D18	D17	D16
BIT NAME	LPC Host Address Bits[07:00]							
BYTE1 BIT	D15	D14	D13	D12	D11	D10	D9	D8
BIT NAME	Valid	Reserved	Frame					
BYTE0 BIT	D7	D6	D5	D4	D3	D2	D1	D0
BIT NAME	Mask							

MASK

These bits are used to mask off address bits in the address match between an LPC memory address and the Host Address field of the BARs, as described in Section 5.6.6, "Mapping LPC Memory Addresses".

FRAME

These 6 bits are used to specify a logical device frame number within a bus. This field is multiplied by 400h to provide the frame address within the peripheral bus address. In the MEC1632 Frame values for frames corresponding to logical devices that are not present on the MEC1632 are invalid.

VALID

If this bit is 1, the BAR is valid and will participate in LPC matches. If it is 0 this BAR is ignored.

HOST_ADDRESS

These 32 bits are used to match LPC memory addresses.

© 2012-2018 Microchip Technology Inc.

5.6.6 MAPPING LPC MEMORY ADDRESSES

This section is the same as Mapping LPC I/O Addresses on page 67 except that the LPC memory cycle address is 32 bit instead of the I/O cycle 16 bit.

5.6.7 MEMORY BASE ADDRESS REGISTER TABLE

Table 5-6, "Base Address Registers Default Values", lists the Base Address Registers for logical devices which have LPC memory access in the MEC1632.

- LPC Memory cycle access is controlled by LPC Memory Base Address Registers. LPC Memory BAR registers are located in LDN Ch (LPC Interface) at AHB base address FF_3000h starting at offset 3C0h.

TABLE 5-8: MEMORY BASE ADDRESS REGISTERS DEFAULT VALUES

Block	LDN	Memory Bar's AHB Address in CR Space (Hex)	Memory Bar's LPC Offset in CR Space (Hex)	Memory Bar's Default Value (Hex)	Memory Bar Decodes AHB ADDR (HEX)
Mailbox	0	FF33C0h	C0h	0000_0000_0001h	FF0000
ACPIEC0	2	FF33C6h	C6h	0000_0062_0204h	FF0800
ACPIEC1	3	FF33CCh	CCh	0000_0062_0307h	FF0C00
ACPIEC2	4	FF33D2h	D2h	0000_0062_0407h	FF1000
ACPIEC3	5	FF33D8h	D8h	0000_0062_0507h	FF1400
Flash	E	FF33DEh	DEh	0000_0000_0E04h	FF3800
EMI	10	FF33E4h	E4h	0000_0000_100Fh	FF4000

Note 5-3 The Mask and Frame fields of all logical devices are read-only except for 2h **(**ACPI EC Channel 0**).**

5.7 SERIRQ Interrupts

The MEC1632 can routes Logical Device interrupts onto SIRQ stream frames IRQ[0:15]. Routing is controlled by the SIRQ Interrupt Configuration Registers. There is one SIRQ Interrupt Configuration Register for each accessible SIRQ Frame (IRQ); all 16 registers are listed in Table 5-9, "SIRQ Interrupt Configuration Register Map". Each SIRQ Interrupt Configuration Register controls a series of multiplexors which route to a single Logical Device interrupt as illustrated in FIGURE 5-2: SIRQ Routing Internal Logical Devices on page 73. The format for each SIRQ Interrupt Configuration Register is described in Table 5-10. Each Logical Device can have up to two LPC SERIRQ interrupts. When the MEC1632 is polled by the host, each SIRQ frame routes the level of the Logical Device interrupt (selected by the corresponding SIRQ Interrupt Configuration Register) to the SIRQ stream. See

Note: Two Logical Devices cannot share a Serial IRQ.

The SIRQ Interrupt Configuration Register The Host can access the Interrupt Configuration registers with 8-bit accesses. The EC can access the Interrupt Configuration registers as four 32-bit registers, eight 16-bit registers or sixteen 8-bit registers.

Note: An interrupt is deactivated by setting an entry in the SIRQ Interrupt Configuration Register Map to FFh, which is the default reset value.

5.7.1 SERIRQ CONFIGURATION REGISTERS

TABLE 5-9: SIRQ INTERRUPT CONFIGURATION REGISTER MAP

Address	Type	Reset	Configuration Register Name
FF_3340h	R/W	FFh	IRQ0
FF_3341h	R/W	FFh	IRQ1
FF_3342h	R/W	FFh	IRQ2 (nSMI)
FF_3343h	R/W	FFh	IRQ3
FF_3344h	R/W	FFh	IRQ4
FF_3345h	R/W	FFh	IRQ5
FF_3346h	R/W	FFh	IRQ6
FF_3347h	R/W	FFh	IRQ7
FF_3348h	R/W	FFh	IRQ8
FF_3349h	R/W	FFh	IRQ9
FF_334Ah	R/W	FFh	IRQ10
FF_334Bh	R/W	FFh	IRQ11
FF_334Ch	R/W	FFh	IRQ12
FF_334Dh	R/W	FFh	IRQ13
FF_334Eh	R/W	FFh	IRQ14
FF_334Fh	R/W	FFh	IRQ15

Note 5-4 The SIRQ Interrupt Configuration Registers are through the Host Access Port as 8-bit accesses. The EC can access the SIRQ Interrupt Configuration Registers as 32-bit, 16-bit across 8-bit boundary or as individual 8-bit accesses.

TABLE 5-10: SIRQ INTERRUPT CONFIGURATION REGISTER FORMAT

BYTE0 BIT	D7	D6	D5	D4	D3	D2	D1	D0
BIT NAME	Select	Reserved	Frame					

FRAME

These six bits select the Logical Device as the source for the interrupt.

This field defaults to 3Fh.

Note: The LPC Logical Device (Logical Device Number 0Ch) can be used by the Embedded Controller to generate a Serial Interrupt Request to the Host under software control.

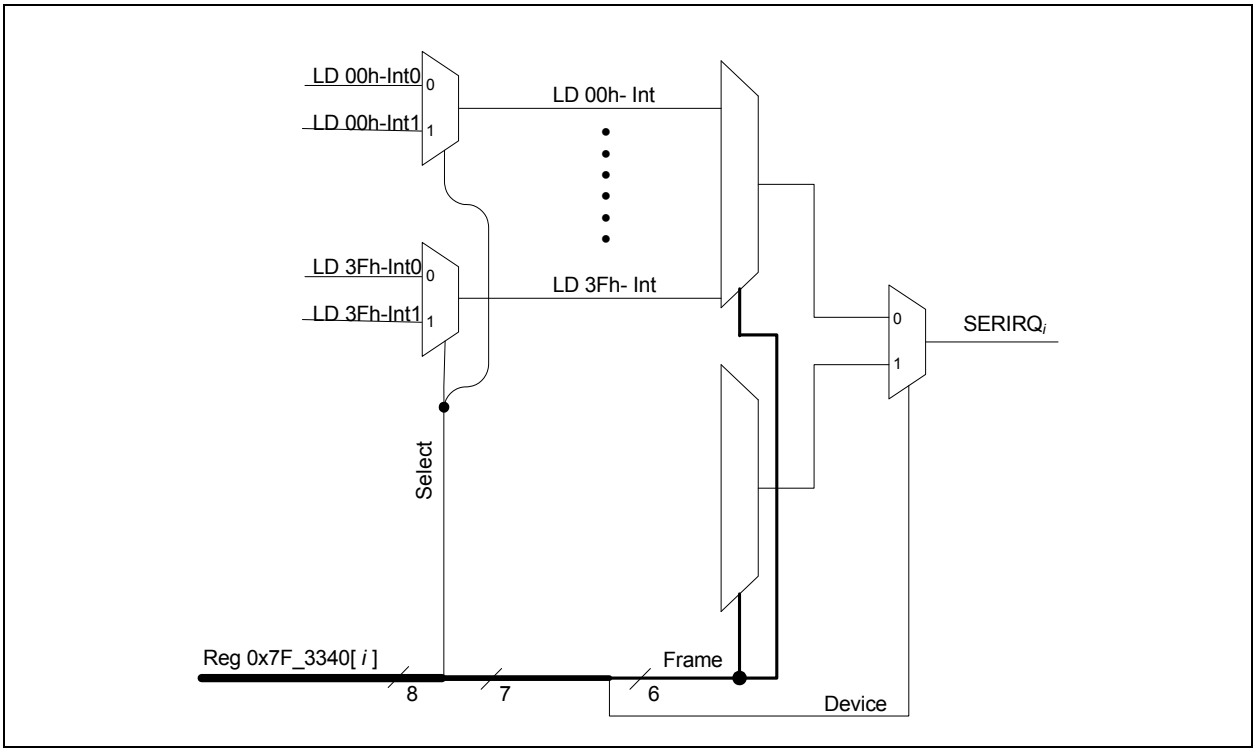
SELECT

If this bit is 0, the first interrupt signal from the Logical Device is selected for the SERIRQ vector. If this bit is 1, the second interrupt signal from the Logical Device is selected. The KYBD controller is the only Logical Devices on the MEC1632 which has a second interrupt signal. For all other Logical Devices this field is ignored.

This field defaults to 1.

© 2012-2018 Microchip Technology Inc.

FIGURE 5-2: SIRQ ROUTING INTERNAL LOGICAL DEVICES



5.7.1.1 MEC1632 SIRQ Routing

TABLE 5-11: MEC1632 LOGICAL DEVICE SIRQ ROUTING

SIRQ Interrupt Configuration Register		Logical Device Interrupt Source
SELECT	FRAME	
0	00h	MailBox SIRQ - See Section 12.3.1, on page 214
1	00h	SMI - Section 12.3.1, on page 214
0	01h	Keyboard SIRQ - Section 10.4.1, on page 191
1	01h	Mouse SIRQ - Section 10.4.1, on page 191
0	06h	UART SIRQ - Section 13.3.2, on page 223
0	0Ch	Serial IRQ - Section 6.10.3, on page 92

5.8 Configuration Register Reset Conditions

There are two reset conditions that will cause Configuration Registers on the MEC1632 to reset to default values. A reset can be caused by a VTR Power On Reset condition or a VCC-Powergood Power on Reset condition. In addition, firmware running on the Embedded Controller can set all Configuration Registers to a default condition. The Host can request that the Configuration Registers be reset through a request to the Embedded Controller sent via the Mailbox interface.

5.9 Logical Device Configuration/Control Registers

A separate set of control and configuration registers exist for each Logical Device and is selected with the Logical Device # Register (07h). The Logical Devices are listed in Table 5-1, “Logical Devices,” on page 63, and the registers within each Logical Device are listed in Section 5.9.2, on page 74.

5.9.1 LOGICAL DEVICE ACTIVATION

Many Logical Devices have a register, called Activate, that is used to activate the Logical Device. When a Logical Device is inactive, it is powered down and will not respond to an AHB request. The format for the Activate Register is shown in Table 5-12, "Activate Register". The Activate Register for the LPC Logical Device is shown in Table 6-6, “Activate Register,” on page 89.

TABLE 5-12: ACTIVATE REGISTER

HOST OFFSET	BYTE0: 30h			8-bit			**HOST SIZE**	
EC OFFSET	Frame offset 330h			8-bit			**EC SIZE**	
POWER	VTR			00b			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R	R	R	R	R	R/W
EC TYPE	R	R	R	R	R	R	R	R/W
BIT NAME	Reserved							Activate

ACTIVATE

When this bit is 1, the logical device is powered and functional. When this bit is 0, the logical device is powered down and inactive.

5.9.2 CONFIGURATION REGISTER MAP

The MEC1632 Configuration register map is shown in Table 5-14, "MEC1632 Configuration Register Map". Logical Device numbers are in hexadecimal. All Logical Devices are accessible by both the Host and the EC. Logical Devices values between 00h and 3Fh are defined for the MEC1632; however some are reserved.

The EC has access to all Configuration Registers and all Global Control registers directly on the AHB bus. The address of a Global Control register is FF FF00h+*offset*, where *offset* is the offset listed in Column 2 of Table 5-15, "Chip-Level (Global) Control/Configuration Registers". The other Configuration Registers are accessible at the INDEX (Table 5-14, Column 1) plus the AHB Address Base (Table 5-13, "Configuration Register AHB Mapping", Column 2). LDN is the Logical Device Number.

Note: The Global Configuration Registers are the first 48 bytes of the configuration space Logical Device 3Fh.

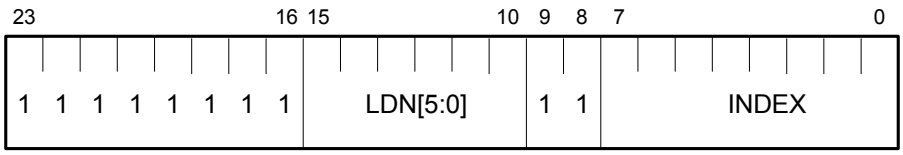
TABLE 5-13: CONFIGURATION REGISTER AHB MAPPING

Logical Device Range	AHB Address Base
00h - 3Fh	FF_0300h + (LDN << 10)

© 2012-2018 Microchip Technology Inc.

The Configuration Register address mapping is illustrated in:

FIGURE 5-3: CONFIGURATION REGISTER MAPPING



~L = ~LDN[6]

TABLE 5-14: MEC1632 CONFIGURATION REGISTER MAP

LPC CR Index	AHB Offset	Type Note 5-5	Reset Note 5-6	Configuration Register Name
Configuration Registers for LDN 0h (Mailbox Interface) at AHB base address FF_0000h				
-	-	-	-	None
Configuration Registers for LDN 1h (Keyboard Controller (8042)) at AHB base address FF_0400h				
30h	330h	R/W	00h on nSYS_RST	Activate Register
Configuration Registers for LDN 2h (ACPI EC Channel 0) at AHB base address FF_0800h				
-	-	-	-	None
Configuration Registers for LDN 3h (ACPI EC Channel 1) at AHB base address FF_0C00h				
-	-	-	-	None
Configuration Registers for LDN 4h (ACPI EC Channel 2) at AHB base address FF_1000h				
-	-	-	-	None
Configuration Registers for LDN 5h (ACPI EC Channel 3) at AHB base address FF_1400h				
-	-	-	-	None
Configuration Registers for LDN 6h (ACPI PM1) at AHB base address FF_1800h				
-	-	-	-	None
Configuration Registers for LDN 7h (UART) at AHB base address FF_1C00h				
30h	330h	R/W	00h on nSYS_RST	Activate Register
F0h	3F0h	R/W	00h on nSYS_RST	Configuration Select Register
Configuration Registers for LDN 8h (Legacy (Fast KB)) at AHB base address FF_2000h				
30h	330h	R/W	00h on nSYS_RST	PORT92 Enable Register
Configuration Registers for LDN Ch (LPC Interface) at AHB base address FF_3000h				
30h	330h	R/W	00h on nSYS_RST	Activate Register
40h	340h	R/W	FFh on nSIO_RESET	SIRQ IRQ0 Configuration Register
41h	341h	R/W	FFh on nSIO_RESET	SIRQ IRQ1 Configuration Register
42h	342h	R/W	FFh on nSIO_RESET	SIRQ IRQ2 (nSMI) Configuration Register
43h	343h	R/W	FFh on nSIO_RESET	SIRQ IRQ3 Configuration Register
44h	344h	R/W	FFh on nSIO_RESET	SIRQ IRQ4 Configuration Register
45h	345h	R/W	FFh on nSIO_RESET	SIRQ IRQ5 Configuration Register
46h	346h	R/W	FFh on nSIO_RESET	SIRQ IRQ6 Configuration Register

© 2012-2018 Microchip Technology Inc.

TABLE 5-14: MEC1632 CONFIGURATION REGISTER MAP (CONTINUED)

LPC CR Index	AHB Offset	Type Note 5-5	Reset Note 5-6	Configuration Register Name
47h	347h	R/W	FFh on nSIO_RESET	SIRQ IRQ7 Configuration Register
48h	348h	R/W	FFh on nSIO_RESET	SIRQ IRQ8 Configuration Register
49h	349h	R/W	FFh on nSIO_RESET	SIRQ IRQ9 Configuration Register
4Ah	34Ah	R/W	FFh on nSIO_RESET	SIRQ IRQ10 Configuration Register
4Bh	34Bh	R/W	FFh on nSIO_RESET	SIRQ IRQ11 Configuration Register
4Ch	34Ch	R/W	FFh on nSIO_RESET	SIRQ IRQ12 Configuration Register
4Dh	34Dh	R/W	FFh on nSIO_RESET	SIRQ IRQ13 Configuration Register
4Eh	34Eh	R/W	FFh on nSIO_RESET	SIRQ IRQ14 Configuration Register
4Fh	34Fh	R/W	FFh on nSIO_RESET	SIRQ IRQ15 Configuration Register
50h - 5Fh	350h	R/W	00h on nSIO_RESET	Reserved
60h - 63h	360h	R/W / R	002E_0C01h on nSIO_RESET	BAR for Configuration Port
64h - 67h	364h	R/W / R	0000_0001h on nSIO_RESET	BAR for Mailbox
68h - 6Bh	368h	R/W / R	0060_0104h on nSIO_RESET	BAR for 8042/Keyboard Interface
6Ch - 6Fh	36Ch	R/W / R	0062_0204h on nSIO_RESET	BAR for ACPI EC Interface 0 (See Note 5-1 on page 69)
70h - 73h	370h	R/W / R	0062_0307h on nSIO_RESET	BAR for ACPI EC Interface 1 (See Note 5-2 on page 69)
74h - 77h	374h	R/W / R	0062_0407h on nSIO_RESET	BAR for ACPI EC Interface 2 (See Note 5-2 on page 69)
78h - 7Bh	378h	R/W / R	0062_0507h on nSIO_RESET	BAR for ACPI EC Interface 3 (See Note 5-2 on page 69)
7Ch - 7Fh	37Ch	R/W / R	0000_0607h on nSIO_RESET	BAR for ACPI PM1 Interface
80h - 83h	380h	R/W / R	0000_0707h on nSIO_RESET	BAR for UART
84h - 87h	384h	R/W / R	0092_0800h on nSIO_RESET	BAR for Legacy (Fast KYBD) Interface
88h - 8Bh	388h	R/W / R	0000_0E04h on nSIO_RESET	BAR for Embedded Flash Interface
8Ch - 8Fh	38Ch	R/W / R	0000_0F07h on nSIO_RESET	BAR for GP-SPI Interface
90h - 93h	390h	R/W / R	0000_100F on nSIO_RESET	BAR for EM Interface
94h - 97h	394h	R/W / R	15E0_301F on nSIO_RESET	Reserved
98h - 9Bh	398h	R/W / R	0000_2000 on nSIO_RESET	BAR for Port 80 BIOS Debug Port 0
9Ch - 9Fh	39Ch	R/W / R	0000_2100 on nSIO_RESET	BAR for Port 80 BIOS Debug Port 1
A0h - A3h	3A0h	R/W / R	0000_111F on nSIO_RESET	BAR for RTC
B0h - BFh	3B0h - 3BFh	R	0000_0000 on nSIO_RESET	Reserved
C0h - C5h	3C0h	R/W / R	0000_0000_0001h on nSIO_RESET	Memory BAR for Mailbox

© 2012-2018 Microchip Technology Inc.

TABLE 5-14: MEC1632 CONFIGURATION REGISTER MAP (CONTINUED)

LPC CR Index	AHB Offset	Type Note 5-5	Reset Note 5-6	Configuration Register Name
C6h - CBh	3C6h	R/W / R	0000_0062_0204h on nSIO_RESET	Memory BAR for ACPI EC Interface 0
CCh - D1h	3CCh	R/W / R	0000_0062_0307h on nSIO_RESET	Memory BAR for ACPI EC Interface 1
D2h - D7h	3D2h	R/W / R	0000_0062_0407h on nSIO_RESET	Memory BAR for ACPI EC Interface 2
D8h - DDh	3D8h	R/W / R	0000_0062_0507h on nSIO_RESET	Memory BAR for ACPI EC Interface 3
DEh - E3h	3DEh	R/W / R	0000_0000_0E04h on nSIO_RESET	Memory BAR for Embedded Flash Interface
E4h - E9h	3E4h	R/W / R	0000_0000_100F on nSIO_RESET	Memory BAR for EM Interface
EAh - EFh	3EAh - 3EFh	R	0000_0000 on nSIO_RESET	Reserved
Configuration Registers for LDN Eh (Embedded Flash Interface) at AHB base address FF_3800h				
-	-	-	-	None
Configuration Registers for LDN Fh (Flash SPI) at AHB base address FF_3C00h				
-	-	-	-	None
Configuration Registers for LDN 10h (EM Interface) at AHB base address FF_4000h				
-	-	-	-	None
Configuration Registers for LDN 11h (Real Time Clock) at AHB base address FF_4400h				
-	-	-	-	None
Configuration Registers for LDN 20h (BIOS Debug Port 0) at AHB base address FF_8000h				
30h	330h	R/W	00h on nSYS_RST	Activate Register
Configuration Registers for LDN 21h (BIOS Debug Port 1) at AHB base address FF_8400h				
30h	330h	R/W	00h on nSYS_RST	Activate Register
Configuration Registers for LDN 3Fh (Global Configuration) at AHB base address FF_FC00h				
00h - 02h	FF00h - FF02h	-		Reserved
03H	FF03	-	-	MCHP Reserved
04h - 06h	FF04h - FF06h	-	00h on nSIO_RESET	Reserved
07h	FF07h	R/W	-	Logical Device Number
08h - 1Fh	FF08h- FF1Fh	-		Reserved
20h	FF20h	R	17h	Device ID
21h	FF21h	R	Current Revision hardwired	Device Revision A read-only register which provides device revision information
22h– 23h	FF22h- FF23h	-	04h on nSIO_RESET	MCHP Reserved
24h	FF24h	R/W	00h	Device Mode
25h – 2Fh	FF25h- FF2Fh	-		MCHP Reserved

Note 5-5 R/W / R means that some parts of a register are read/write and some parts are read-only.

Note 5-6 Resets are defined in Section 7.0, "Power, Clocks, and Resets": nSYS_RST on page 128 and nSIO_RESET on page 97.

5.10 Chip-Level (Global) Control/Configuration Registers [00h - 2Fh]

The chip-level (global) registers reside in Logical Device 3Fh at AHB addresses FF_FF00h through FF_FF2Fh. All unimplemented registers and bits ignore writes and return zero when read. The global registers are accessed in the configuration address range [00h - 2Fh] in all Logical Devices. There is no Activate associated with Logical Device 3Fh: the Global Configuration Registers are always accessible.

As with all Configuration Registers, the INDEX PORT is used to select a Global Configuration Register in the chip. The DATA PORT is then used to access the selected register.

The Host can access all the Global Configuration registers at the offsets listed in Table 5-15, "Chip-Level (Global) Control/Configuration Registers" through the INDEX PORT and the DATA PORT. The EC can access all these registers at the listed offsets from the AHB Base Address shown in Table 5-14, "MEC1632 Configuration Register Map".

TABLE 5-15: CHIP-LEVEL (GLOBAL) CONTROL/CONFIGURATION REGISTERS

Register	Offset	Description
CHIP (GLOBAL) CONTROL REGISTERS		
Reserved	00h-03h	Reserved, Writes are ignored, reads return 0.
Reserved	04h - 06h	Reserved - Writes are ignored, reads return 0.
Logical Device Number	07h	A write to this register selects the current logical device. This allows access to the control and configuration registers for each logical device. **Note:** The Activate command operates only on the selected logical device.
Reserved	08h - 1Fh	Reserved - Writes are ignored, reads return 0.
Device ID Hard Wired	20h	A read-only register which provides device identification.
Device Revision Hard Wired	21h	A read-only register which provides device revision information. Bits[7:0] = current revision when read
Reserved	22h - 23h	Reserved - Writes are ignored, reads return 0.
Device Mode	24h	Bit [1:0] Reserved – writes ignored, reads return “0”. Bit[2] SerIRQ Mode (Note 5-7) = 0: Serial IRQ Disabled. = 1: Serial IRQ Enabled (Default). Bit [7:3] Reserved – writes ignored, reads return “0”.
Reserved	25h - 27h	Reserved - Writes are ignored, reads return 0.
Test Register	28h	MCHP Test Mode Register, Reserved for MCHP
Test Register	29h	MCHP Test Mode Register, Reserved for MCHP
Reserved	2Ah - 2Bh	Reserved - Writes are ignored, reads return 0.
Test Register	2Ch	MCHP Test Mode Register, Reserved for MCHP
Reserved	2Dh - 2Fh	Reserved - Writes are ignored, reads return 0.

Note 5-7 The SerIRQ Mode bit controls the SER_IRQ pin, the CLKRUN# pin and the affects of LPC DMA requests on CLKRUN# (See Section 6.7, on page 84 and Section 6.8, on page 86.

© 2012-2018 Microchip Technology Inc.

6.0 HOST INTERFACE

6.1 General Description

6.1.1 OVERVIEW

The host processor communicates with the MEC1632 via the LPC Bus Interface. The host processor communicates through a series of read/write registers in the MEC1632. Register access is accomplished through programmed I/O cycles. All I/O transfer cycles are 8 bits wide.

The Logical Devices physically located in the MEC1632 are identified in Table 4-2, “Host Logical Devices in the MEC1632,” on page 59 and Table 5-1, “Logical Devices,” on page 63. The base addresses of logical devices with registers located in LPC I/O space, including the Keyboard Controller, can be moved via the configuration registers located in the LPC Interface Configuration Register Space.

All configuration register access for the MEC1632 are accessed indirectly through the LPC I/O Configuration Register Port (IOCR-Port.) The default I/O address is 2Eh and 2Fh, but the IOCR-Port can be relocated by either the host or the EC. The relocation value can be stored in the BAR Init Register so that the updated IOCR-Port address is used when the main power rail cycles. Detailed description of the MEC1632 Configuration Space is in Section 5.0, "Logical Device Configuration," on page 63.

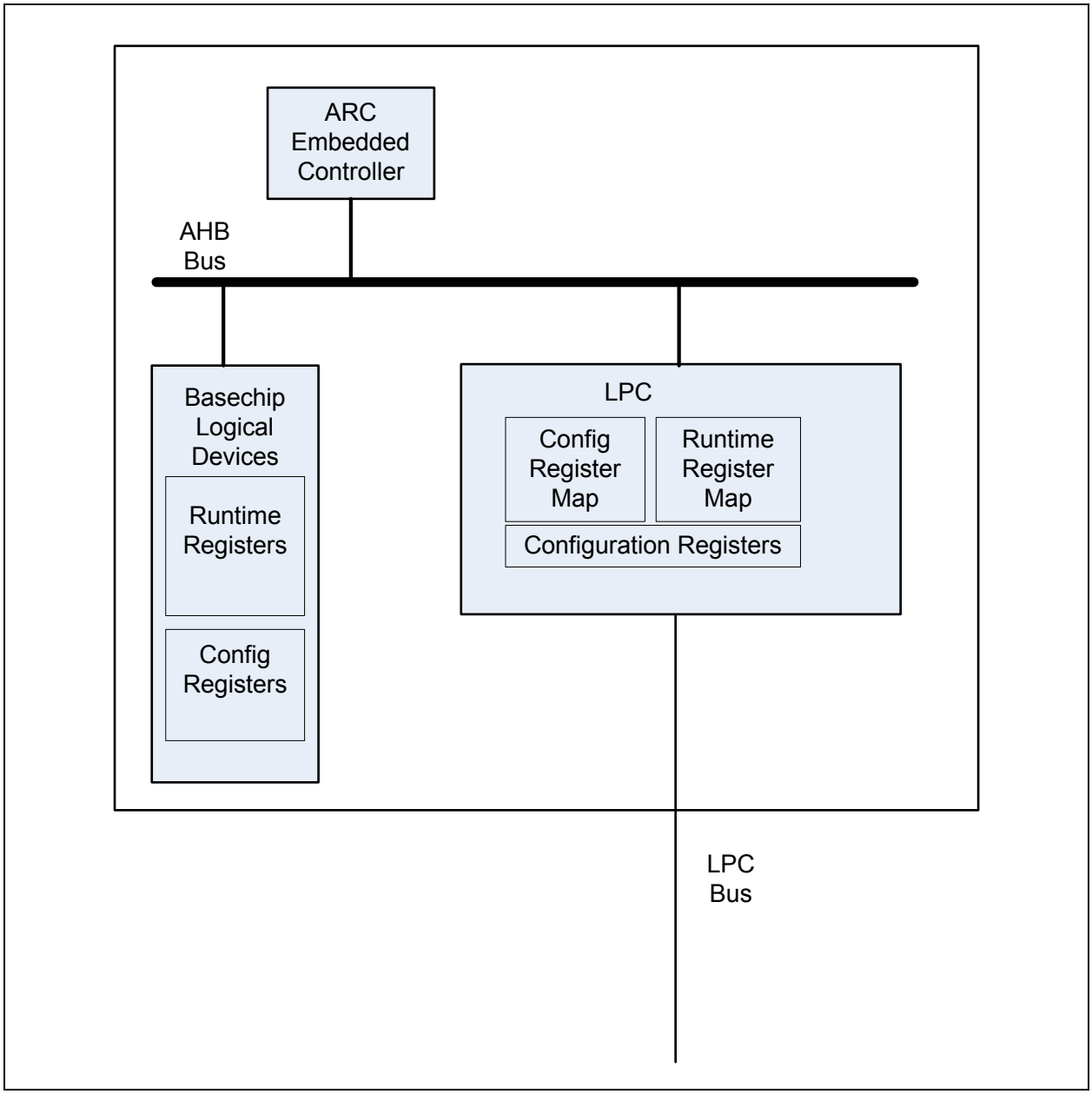
All LPC transactions that are claimed by the MEC1632 are mapped by the LPC interface to an address in the MEC1632’s AHB address space. All these addresses can also be accessed by the Embedded Controller in the MEC1632.

TABLE 6-1: TARGETS OF LBC CYCLES CLAIMED BY THE MEC1632

Target	Acronym	Description	LPC Type
LPC IO Configuration Register Port	IOCR-Port	Standard LPC 2Eh/2Fh Port which permits BIOS access. This port can be relocated by the ARC.	I/O
Logical Devices	LD	Targets physically located in the MEC1632. The Keyboard Controller Interface uses a Port at 60h/64h	I/O
Configuration Register	CR	256 byte space per Logical Device accessed by BIOS through the IOCR-Port.	I/O through CR-Port

6.1.2 BLOCK DIAGRAM

FIGURE 6-1: LPC INTERFACE IN MEC1632



© 2012-2018 Microchip Technology Inc.

6.2 Power, Clocks and Resets

6.2.1 POWER DOMAIN

This block is powered by VTR. Although the block is not powered by VCC, the block is also controlled by VCC_PWRGD. When VCC_PWRGD is de-asserted, the LPC bus pins are placed in the same state they assume when VTR is off. LAD[3:0] and SERIRQ are tri-stated, CLKRUN# is unpowered and LFRAME#, LRESET# and LPCPD# are gated high; see Table 6-2, “LPC Bus Pin Behavior on Reset,” on page 83.The LPC block is also placed in a minimal power state.

See Section 7.1.1, "Power Configuration," on page 95 for details on power domains.

6.2.2 CLOCKS

LPC Logical Device Configuration Registers and LPC Logical Device EC-only Registers in this block are clocked by the LPC Bus Clock. The LPC interface itself is clocked by the PCI_CLK clock input.

The clock rate of the LPC Bus Clock runs at the MCLK rate.

See Section 7.4, "Clock Generator," on page 98 for details on clocks.

6.2.3 RESETS

This block is affected by nSYS_RST, VCC Power Good and LPC RESET.

The assertion of nSYS_RST resets the LPC state machine and all registers to their default values. The AHB Master state machine is also reset to its default value.

VCC Power Good going low resets the LPC state machine. The AHB Master interface that is part of the Host Interface will go to its idle state. Any transaction that is active on the AHB Master when the VCC POR occurs will be terminated in such a way that the AHB subsystem will not be locked up.

The assertion of LPC RESET# resets the LPC state machine but does not otherwise affect register values. An interrupt to the EC will be generated on either edge of LRESET#. See Section 6.5, "Host Interrupts to EC," on page 84.

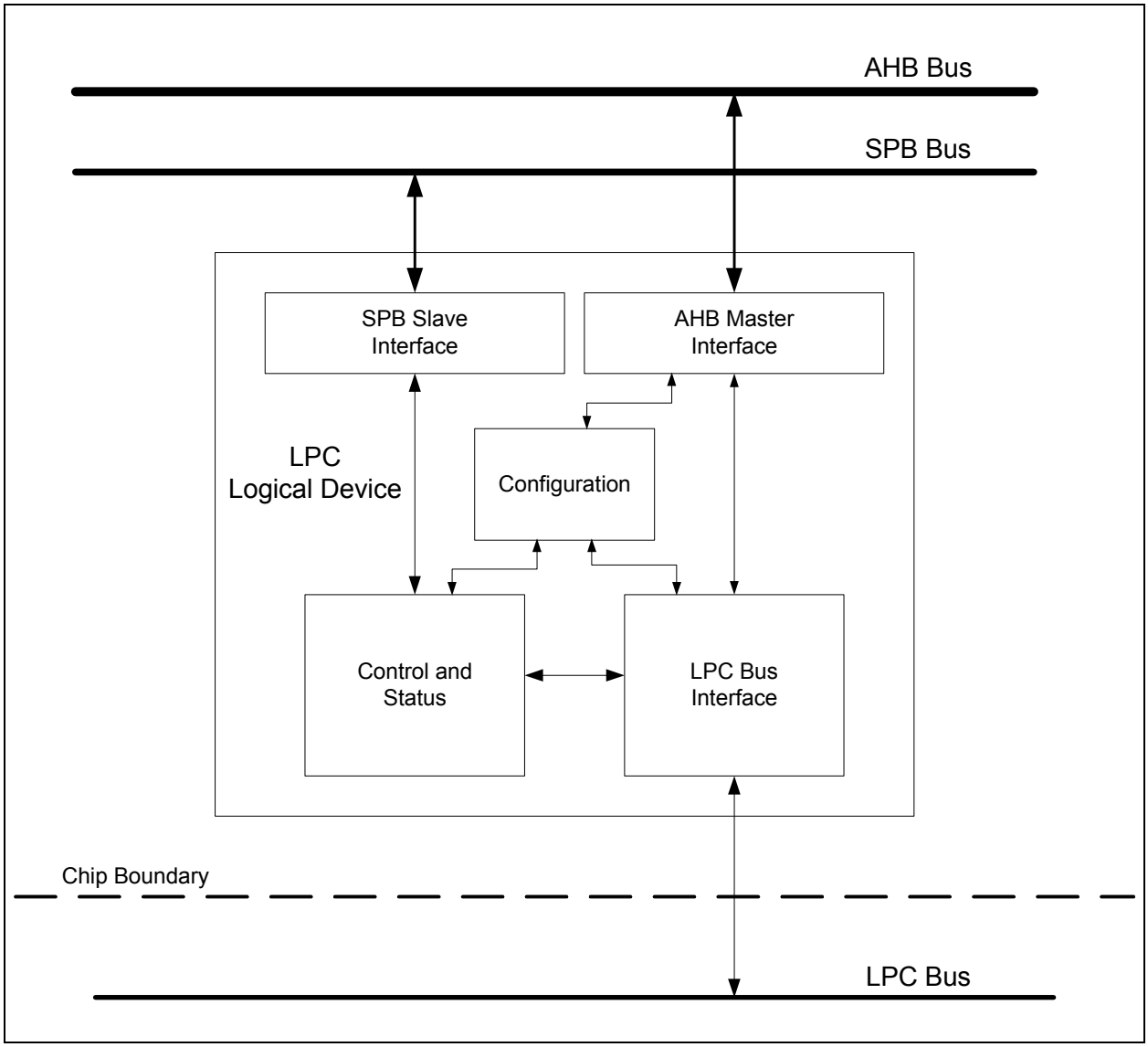
See Section 7.6, "Reset Interface," on page 124 for details on reset.

© 2012-2018 Microchip Technology Inc.

6.3 LPC Logical Device

The LPC Logical Device structure is illustrated in Figure 6-1, "LPC Interface in MEC1632".

FIGURE 6-2: LPC LOGICAL DEVICE



The LPC Logical Device is directly connected to two internal buses, the 32-bit AHB as well as the SPB. The SPB interface is 32-bits. In addition, it is connected to the external LPC bus.

Host accesses to Configuration Registers for each Logical Device on the MEC1632 are managed by a Configuration block described in Section 5.0, "Logical Device Configuration," on page 63. Configuration registers are accessed through the LPC IO Configuration Register Port. The LPC Logical Device translates the Configuration address to an AHB address and the Host LPC access is converted into an AHB transaction inside the MEC1632.

Host I/O accesses to Configuring Runtime Register Addresses are converted directly to AHB accesses. The LPC address is translated by the LPC Bus Interface to an AHB address inside the MEC1632 and the access becomes an access on the AHB bus.

© 2012-2018 Microchip Technology Inc.

6.3.1 LPC BUS INTERFACE

The MEC1632 communicates with the host over a Low Pin Count (LPC) interface. The LPC interface uses 3.3V signaling. For detailed specifications, see the *Intel Low Pin Count Specification* and the P*CI Local Bus Specification*, Section 4.2.2.

The following cycle types are supported by the LPC Bus protocol.

- 8-bit I/O Read
- 8-bit I/O Write
- 8-bit Memory Reads
- 8-bit Memory Writes

LPC transactions that claimed by the MEC1632 require a minimum of two wait SYNCs on the LPC bus. The number of SYNCs may be larger if the AHB bus is in use by the embedded controller, of if the data referenced by the host is not present in a MEC1632 register. The MEC1632 always uses Long Wait SYNCs, rather than Short Wait SYNCs, when responding to an LPC bus request.

The PCI_CLK input can run at 24MHz or 33MHz. When the PCI_CLK input is 24MHz the Hand shake bit in the EC Clock Control Register must be set to a '1' to capture LPC transactions properly. See Section 6.10.4, "EC Clock Control Register," on page 92.

Table 6-2, "LPC Bus Pin Behavior on Reset", shows the behavior of LPC outputs and input/outputs under reset conditions in accordance with the *Intel Low Pin Count Specification* and the P*CI Local Bus*. See Section 6.7, "LPC Clock Run and LPC Power Down Behavior," on page 84 for LPC protocol dependent pin state transitions requirements.

TABLE 6-2: LPC BUS PIN BEHAVIOR ON RESET

Pins	nSYS_RST	VCC POR	LPCPD# Asserted	LRESET# Asserted
LAD[3:0]	Tri-state	Tri-state	Tri-State	Tri-State
SERIRQ	Tri-state	Tri-state	Tri-State	Tri-State
CLKRUN#	Tri-state	Tri-state	Tri-State	Tri-State

6.3.2 LPC I/O CYCLES

LPC 8-bit I/O Read cycles and 8-bit I/O Write cycles are mapped directly to addresses in the MEC1632 AHB address space. The mapping will be to the range FF_0000h through FF_FFFFh. For information on how addresses map between the LPC bus and the MEC1632, see Section 4.0, "Bus Hierarchy," on page 55 and Section 5.0, "Logical Device Configuration," on page 63. For a list of all Configuration Registers accessible to the Host, see Section 5.0, "Logical Device Configuration," on page 63.

6.3.3 LPC FIRMWARE HUB CYCLES

The MEC1632 does not support LPC Firmware Hub cycles on the LPC Bus.

6.3.4 LPC MEMORY CYCLES

The system host may use LPC memory cycles to access memory mapped registers and internal RAMs implemented in this device. See the Intel® Low Pin Count (LPC) Interface Specification, v1.1, Section 5.1 for definition of LPC Memory Cycles.

6.3.5 LPC DMA CYCLES

The MEC1632 does not support LPC DMA cycles.

6.4 LPC Bus Configuration

The mapping from LPC Bus cycles to AHB read/write cycles is managed by the LPC Logical Device. The mapping is defined by a series of configuration registers which are defined in Section 5.0, "Logical Device Configuration," on page 63, in Section 6.4, "LPC Bus Configuration," on page 83.

6.5 Host Interrupts to EC

The LRESET# reset signal and the LPCPD# power down signal can be used to generate EC interrupts and wake-up events. The edge detection of the interrupt and wake events are controlled by their associated Pin Control Register on page 396. The interrupts are routed to the LRESET# and LPCPD# bits in the GIRQ14 Source Register on page 311.

The LPC Logical Device can generate an additional interrupt to the EC when a Host access is mapped to the AHB bus. Bit LPC_AHB_ERR in the Host Bus Error Register is set when an LPC-sourced AHB bus access causes an error; it is also routed to the LPC_AHB_ERR bit in the GIRQ14 Source Register on page 311. For details see Section 6.10.2, "Host Bus Error Register," on page 91.

6.6 EC Interrupts to Host

The Embedded Controller can send an interrupt to the Host on any Serial Interrupt Request channel using the EC SERIRQ Register in conjunction with the SERIRQ Configuration Registers.

6.7 LPC Clock Run and LPC Power Down Behavior

The LPCPD# signal (see the *Intel Low Pin Count Specification*, Section 8.1) and the CLKRUN# signal (see the Intel Low Pin Count Specification, Section 8.2) are implemented in the MEC1632.

6.7.1 USING LPCPD#

The MEC1632 tolerates the LPCPD# signal going active and then inactive again without LRESET# going active. This is a requirement for notebook power management functions.

The *LPC Bus Specification, Rev. 1.0*, Section 8.2 states that “After LPCPD# goes back inactive, the LPC I/F will always be reset using LRST#”. This text must be qualified for mobile systems where it is possible that when exiting a “light” sleep state (ACPI S1, APM POS), LPCPD# may be asserted but the LPC Bus power may not be removed, in which case LRESET# will not occur. When exiting a “deeper” sleep state (ACPI S3-S5, APM STR, STD, soft-off), LRESET# will occur.

The LPCPD# pin is implemented as a “local” powergood for the LPC bus in the MEC1632. It is not to be used as a global powergood for the chip. It is used to minimize the LPC power dissipation.

Prior to going to a low-power state, the system asserts the LPCPD# signal. LPCPD# goes active at least 30 microseconds prior to the LCLK signal stopping low and power being shut to the other LPC interface signals. Upon recognizing LPCPD# active, there are no further transactions on the LPC interface.

6.7.2 USING CLKRUN#

CLKRUN# is used to indicate the PCI clock status as well as to request that a stopped clock be started. See FIGURE 6-3: CLKRUN# System Implementation Example on page 85, an example of a typical system implementation using CLKRUN#.

PCI Clock Run Support can be enabled and disabled using the Bit[2] SerIRQ Mode in the Device Mode register, Global Configuration Register 24h (see Table 5-15, “Chip-Level (Global) Control/Configuration Registers,” on page 78). When the Bit[2] SerIRQ Mode is ‘0,’ Serial IRQs are disabled, the CLKRUN# pin is disabled, and the affects of Interrupts on CLKRUN# are ignored. When the Bit[2] SerIRQ Mode is ‘1,’ Serial IRQs are enabled, the CLKRUN# pin is enabled, and the CLKRUN# support related to Interrupts as described in the section below is enabled.

The CLKRUN# pin is an open drain output and input. Refer to the *PCI Mobile Design Guide Rev 1.0* for a description of the CLKRUN# function. If CLKRUN# is sampled “high”, the PCI clock is stopped or stopping. If CLKRUN# is sampled “low”, the PCI clock is starting or started (running). CLKRUN# in the MEC1632 supports Serial IRQ cycles.

6.7.2.1 CLKRUN# Support for Serial IRQ Cycle

If a device in the MEC1632 asserts or de-asserts an interrupt and CLKRUN# is sampled “high”, the MEC1632 can request the restoration of the clock by asserting the CLKRUN# signal asynchronously (Table 6-3). The MEC1632 holds CLKRUN# low until it detects two rising edges of the clock. After the second clock edge, the MEC1632 must disable the open drain driver (Figure 6-4).

The MEC1632 must not assert CLKRUN# if it is already driven low by the central resource; i.e., the PCI CLOCK GENERATOR in Figure 6-4. The MEC1632 will not assert CLKRUN# under any conditions if the Serial IRQs are disabled.

The MEC1632 must not assert CLKRUN# unless the line has been de-asserted for two successive clocks; i.e., before the clock was stopped (Figure 6-4).

© 2012-2018 Microchip Technology Inc.

TABLE 6-3: MEC1632 CLKRUN# FUNCTION

SIRQ_MODE (Bit[2] SerIRQ Mode in Device Mode Register)	Internal Interrupt Request	CLKRUN#	Action
0	X	X	None
1	NO CHANGE	X	None
	CHANGE (Note 6-1)	0	None
		1	Assert CLKRUN#

Note 6-1 "Change" means either-edge change on any or all parallel IRQs routed to the Serial IRQ block. The "change" detection logic must run asynchronously to the PCI Clock and regardless of the Serial IRQ mode; i.e., "continuous" or "quiet".

FIGURE 6-3: CLKRUN# SYSTEM IMPLEMENTATION EXAMPLE

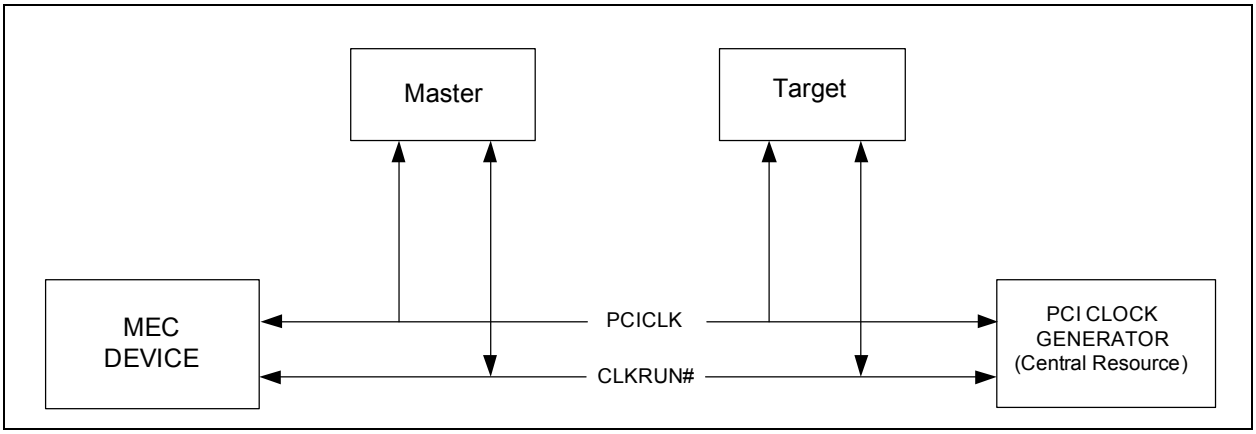
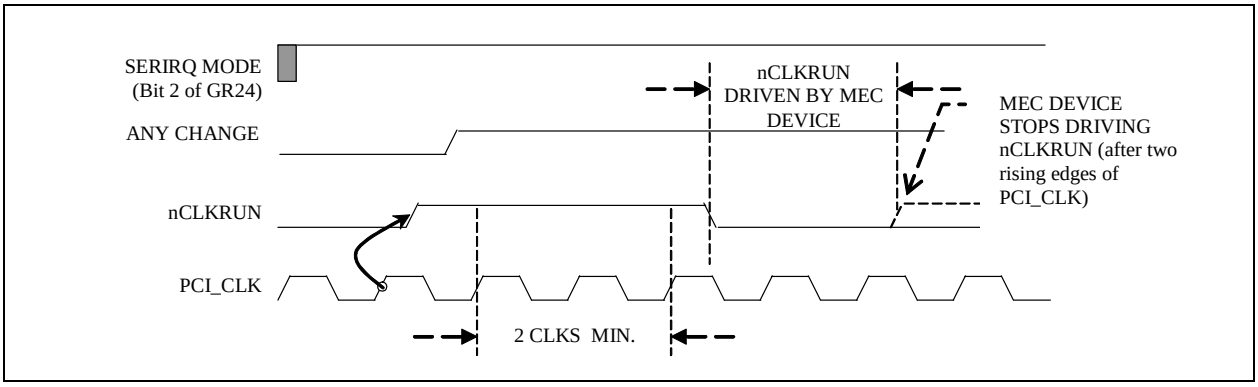


FIGURE 6-4: CLOCK START ILLUSTRATION



Note:

- The signal "ANY CHANGE" is the same as "CHANGE" in Table 6-3.
- The MEC1632 must continually monitor the state of CLKRUN# to maintain the PCI Clock until an active "any IRQ change" condition has been transferred to the host in a Serial IRQ cycle. For example, if "any IRQ change" is asserted before CLKRUN# is de-asserted (not shown in Figure 6-4), the MEC1632 must assert CLKRUN# as needed until the Serial IRQ cycle has completed.

6.8 Using Serial Interrupts

The MEC1632 will support the serial interrupt scheme, which is adopted by several companies, to transmit interrupt information to the system. The serial interrupt scheme adheres to the *Serial IRQ Specification for PCI Systems Version 6.0*.

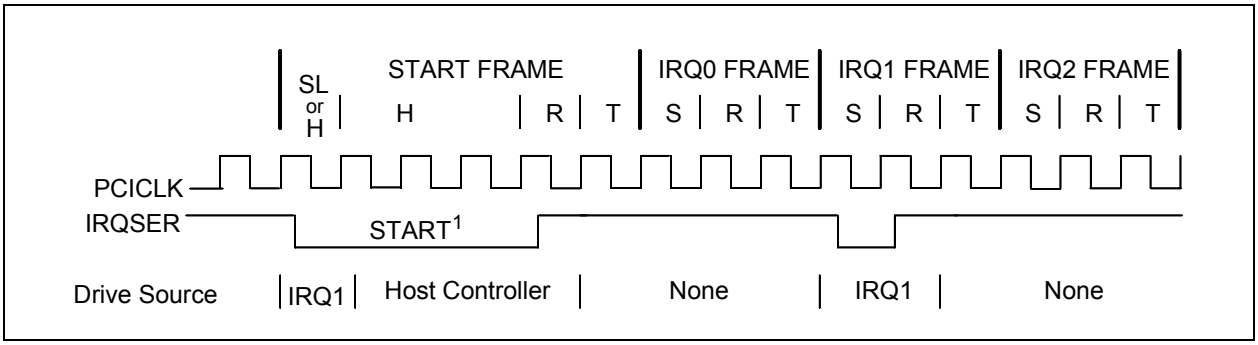
TIMING DIAGRAMS for IRQSER CYCLE

PCICLK = 33 MHz_IN pin

IRQSER = SIRQ pin

Start Frame timing with source sampled a low pulse on IRQ1

FIGURE 6-5: SERIAL INTERRUPTS WAVEFORM "START FRAME"

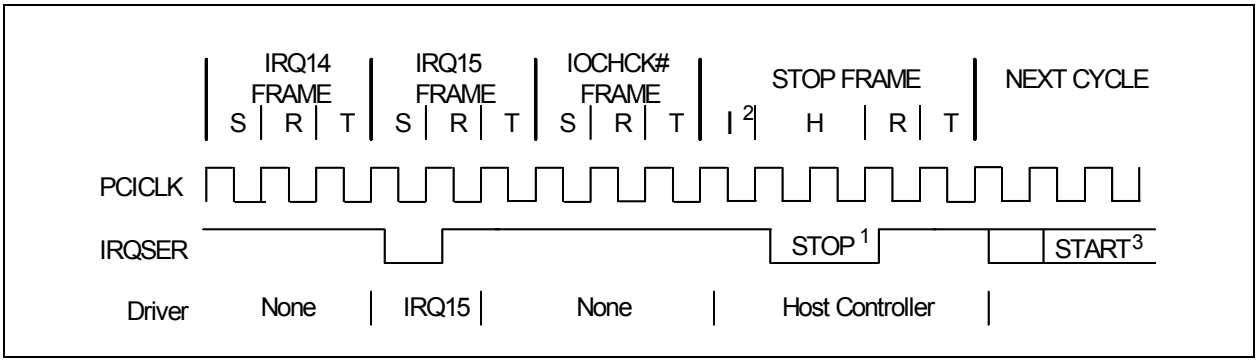


H=Host Control SL=Slave Control R=Recovery T=Turn-around S=Sample

Start Frame pulse can be 4-8 clocks wide.

Stop Frame Timing with Host using 17 IRQSER sampling period

FIGURE 6-6: SERIAL INTERRUPT WAVEFORM "STOP FRAME"



H=Host Control R=Recovery T=Turn-around S=Sample I= Idle

Stop pulse is two clocks wide for Quiet mode, three clocks wide for Continuous mode.

There may be none, one, or more Idle states during the Stop Frame.

The next IRQSER cycle's Start Frame pulse may or may not start immediately after the turn-around clock of the Stop Frame.

© 2012-2018 Microchip Technology Inc.

6.8.1 SERIRQ MODE BIT FUNCTION

TABLE 6-4: SERIRQ_EN CONFIGURATION CONTROL

CR25 Bit[2]	Name	Description
0	SERIRQ_EN	Serial IRQ Disabled
1		Serial IRQ Enabled (Default)

6.8.1.1 IRQSER Cycle Control

There are two modes of operation for the IRQSER Start Frame.

Quiet (Active) Mode

Any device may initiate a Start Frame by driving the IRQSER low for one clock, while the IRQSER is Idle. After driving low for one clock, the IRQSER must immediately be tri-stated without at any time driving high. A Start Frame may not be initiated while the IRQSER is active. The IRQSER is Idle between Stop and Start Frames. The IRQSER is active between Start and Stop Frames. This mode of operation allows the IRQSER to be Idle when there are no IRQ/Data transitions which should be most of the time.

Once a Start Frame has been initiated, the host controller will take over driving the IRQSER low in the next clock and will continue driving the IRQSER low for a programmable period of three to seven clocks. This makes a total low pulse width of four to eight clocks. Finally, the host controller will drive the IRQSER back high for one clock then tri-state.

Any IRQSER Device (i.e., The MEC1632) which detects any transition on an IRQ/Data line for which it is responsible must initiate a Start Frame in order to update the host controller unless the IRQSER is already in an IRQSER Cycle and the IRQ/Data transition can be delivered in that IRQSER Cycle.

Continuous (Idle) Mode

Only the Host controller can initiate a Start Frame to update IRQ/Data line information. All other IRQSER agents become passive and may not initiate a Start Frame. IRQSER will be driven low for four to eight clocks by host controller. This mode has two functions. It can be used to stop or idle the IRQSER or the host controller can operate IRQSER in a continuous mode by initiating a Start Frame at the end of every Stop Frame.

An IRQSER mode transition can only occur during the Stop Frame. Upon reset, IRQSER bus is defaulted to continuous mode, therefore only the host controller can initiate the first Start Frame. Slaves must continuously sample the Stop Frames pulse width to determine the next IRQSER Cycle's mode.

IRQSER Data Frame

Once a Start Frame has been initiated, the MEC1632 will watch for the rising edge of the Start Pulse and start counting IRQ/Data Frames from there. Each IRQ/Data Frame is three clocks: Sample phase, Recovery phase, and Turn-around phase. During the sample phase, the MEC1632 must drive the IRQSER (SIRQ pin) low, if and only if, its last detected IRQ/Data value was low. If its detected IRQ/Data value is high, IRQSER must be left tri-stated. During the recovery phase, the MEC1632 must drive the SERIRQ high, if and only if, it had driven the IRQSER low during the previous sample phase. During the turn-around phase, the MEC1632 must tri-state the SERIRQ. The MEC1632 drives the IRQSER line low at the appropriate sample point if its associated IRQ/Data line is low, regardless of which device initiated the start frame.

The Sample phase for each IRQ/Data follows the low to high transition of the Start Frame pulse by a number of clocks equal to the IRQ/Data Frame times three, minus one e.g. The IRQ5 Sample clock is the sixth IRQ/Data Frame, then the sample phase is {(6 x 3) - 1 = 17} the seventeenth clock after the rising edge of the Start Pulse.

TABLE 6-5: IRQSER SAMPLING PERIODS

IRQSER Period	Signal Sampled	# of Clocks Past Start
1	Not Used	2
2	IRQ1	5
3	nSMI/IRQ2	8
4	IRQ3	11
5	IRQ4	14
6	IRQ5	17
7	IRQ6	20

TABLE 6-5: IRQSER SAMPLING PERIODS (CONTINUED)

IRQSER Period	Signal Sampled	# of Clocks Past Start
8	IRQ7	23
9	IRQ8	26
10	IRQ9	29
11	IRQ10	32
12	IRQ11	35
13	IRQ12	38
14	IRQ13	41
15	IRQ14	44
16	IRQ15	47

The SIRQ data frame will now support IRQ2 from a logical device; previously IRQSER Period 3 was reserved for use by the System Management Interrupt (nSMI). When using Period 3 for IRQ2, the user should mask off the MEC1632's SMI via the ESMI Mask Register. Likewise, when using Period 3 for nSMI, the user should not configure any logical devices as using IRQ2.

IRQSER Period 14 is used to transfer IRQ13. Each Logical devices will have IRQ13 as a choice for their primary interrupt.

Stop Cycle Control

Once all IRQ/Data Frames have completed, the host controller will terminate IRQSER activity by initiating a Stop Frame. Only the host controller can initiate the Stop Frame. A Stop Frame is indicated when the IRQSER is low for two or three clocks. If the Stop Frame's low time is two clocks, then the next IRQSER cycle's sampled mode is the Quiet mode; and any IRQSER device may initiate a Start Frame in the second clock or more after the rising edge of the Stop Frame's pulse. If the Stop Frame's low time is three clocks, then the next IRQSER cycle's sampled mode is the continuous mode, and only the host controller may initiate a Start Frame in the second clock or more after the rising edge of the Stop Frame's pulse.

Latency

Latency for IRQ/Data updates over the IRQSER bus in bridge-less systems with the minimum IRQ/Data Frames of 17 will range up to 96 clocks (3.84μS with a 25 MHz PCI Bus or 2.88μs with a 33 MHz PCI Bus). If one or more PCI to PCI Bridge is added to a system, the latency for IRQ/Data updates from the secondary or tertiary buses will be a few clocks longer for synchronous buses, and approximately double for asynchronous buses.

EOI/ISR Read Latency

Any serialized IRQ scheme has a potential implementation issue related to IRQ latency. IRQ latency could cause an EOI or ISR Read to precede an IRQ transition that it should have followed. This could cause a system fault. The host interrupt controller is responsible for ensuring that these latency issues are mitigated. The recommended solution is to delay EOIs and ISR Reads to the interrupt controller by the same amount as the IRQSER Cycle latency in order to ensure that these events do not occur out of order.

AC/DC Specification Issue

All IRQSER agents must drive/sample IRQSER synchronously related to the rising edge of the PCI bus clock. The IRQSER (SIRQ) pin uses the electrical specification of the PCI bus. Electrical parameters will follow the PCI Specification Section 4, sustained tri-state.

Reset and Initialization

The IRQSER bus uses LRESET as its reset signal and follows the PCI bus reset mechanism. The IRQSER pin is tri-stated by all agents while LRESET is active. With reset, IRQSER slaves and bridges are put into the (continuous) Idle mode. The host controller is responsible for starting the initial IRQSER cycle to collect system's IRQ/Data default values. The system then follows with the Continuous/Quiet mode protocol (Stop Frame pulse width) for subsequent IRQSER cycles. It is the host controller's responsibility to provide the default values to the 8259's and other system logic before the first IRQSER cycle is performed. For IRQSER system suspend, insertion, or removal application, the host controller should be programmed into Continuous (IDLE) mode first. This is to ensure the IRQSER bus is in Idle state before the system configuration changes.

© 2012-2018 Microchip Technology Inc.

6.9 LPC Logical Device Configuration Registers

The configuration registers in the LPC Logical Device are described in Section 5.0, "Logical Device Configuration," on page 63. These registers control the activity of all the Logical Devices in the MEC1632. The Activate Register controls the LPC device itself. The Host can shut down the LPC Logical Device by clearing the Activate bit, but it cannot restart the LPC interface, since once the LPC interface is inactive the Host has no access to any registers on the MEC1632. The Embedded Controller can set or clear the Activate bit at any time.

TABLE 6-6: ACTIVATE REGISTER

HOST OFFSET	BYTE0: 30h			8-bit			**HOST SIZE**	
EC ADDRESS	FF_3330h			8-bit			**EC SIZE**	
POWER	VTR			00b			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R	R	R	R	R	R/W
EC TYPE	R	R	R	R	R	R	R	R/W
BIT NAME	Reserved							Activate

ACTIVATE

When this bit is 1, the LPC Logical Device is powered and functional.

When this bit is 0, the logical device is powered down and inactive. Except for the Activate Register itself, clocks to the block are gated and the LPC Logical Device will permit the ring oscillator to be shut down (see Section 6.10.4, "EC Clock Control Register," on page 92). LPC bus output pads will be tri-stated. Serial IRQ activation is separately controlled by the Device Mode register in the Chip-Level (Global) Control/Configuration Registers.

APPLICATION NOTE: The Activate bit in the Activate Register should not be written '0' to by the Host over LPC.

6.10 LPC Logical Device EC-only Registers

Table 6-7, "LPC EC-only Registers" summarizes the registers in the Host Interface block that are only accessible by the EC. In addition to these registers, the Host Interface also contains Configuration registers, described in Section 5.0, "Logical Device Configuration," on page 63.

TABLE 6-7: LPC EC-ONLY REGISTERS

AHB Address	Name	VTR POR (nSYS_RST) Default
FF_3100h	Reserved	0000_0000h
FF_3104h	LPC Bus Monitor Register	0000_0000h
FF_3108h	Host Bus Error Register	0000_0000h
FF_310Ch	EC SERIRQ Register	0000_0000h
FF_3110h	EC Clock Control Register	0000_0000h
FF_3120h	BAR Inhibit Register	0000_0000_0000_0000h
FF_3130h	BAR Init Register	0000_002Eh
FF_3140h	Mem BAR Inhibit Register	0000_0000_0000_0000h

Because their addresses are in the part of the LPC Logical Device address frame that is not addressable from the LPC bus, the following registers are accessible only to the EC.

6.10.1 LPC BUS MONITOR REGISTER

TABLE 6-8: LPC BUS MONITOR REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC ADDRESS	FF_3104h			32-bit			**EC SIZE**	
POWER	VTR			00h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE[3:1] BIT	**D31**	**D30**	**D29**	**...**		**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved						LRESET_ Status	LPCPD_ Status

LPCPD_STATUS

This bit reflects the state of the LPCPD# input pin. The LPCPD_Status is the inverse of the LPCPD# pin (see Section 6.7, "LPC Clock Run and LPC Power Down Behavior," on page 84).

When the LPCPD_Status bit is ‘0b’, the LPCPD# input pin is de-asserted (that is, the pin has the value ‘1b’). When the LPCPD_Status bit is ‘1b’, the LPCPD# input pin is asserted (that is, the pin has the value ‘0b’).

An interrupt to the EC will be generated on either edge of LPCPD#. See Section 6.5, "Host Interrupts to EC," on page 84.

LRESET_STATUS

This bit reflects the state of the LRESET# input pin. The LRESET_Status is the inverse of the LRESET# pin (see Section 6.2.3, "Resets," on page 81).

When the LRESET_Status bit is ‘0b’, the LRESET# input pin is de-asserted (that is, the pin has the value ‘1b’). When the LRESET_Status bit is ‘1b’, the LRESET# input pin is asserted (that is, the pin has the value ‘0b’).

An interrupt to the EC will be generated on either edge of LRESET#. See Section 6.5, "Host Interrupts to EC," on page 84.

© 2012-2018 Microchip Technology Inc.

6.10.2 HOST BUS ERROR REGISTER

TABLE 6-9: HOST BUS ERROR REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC ADDRESS	FF_3108h			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D31**	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	ErrorAddress[23:16]							
BYTE2 BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	ErrorAddress[15:8]							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	ErrorAddress[7:0]							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/WC	R/WC	R/WC	R/WC	R/WC	R/W	R/WC
BIT NAME	Reserved	Reserved	Reserved	Config Err	Runtime Err	BAR_ Conflict	En_ AHB_Err	LPC_ AHB_Err

LPC_AHB_ERR

This bit can be used to generate an EC interrupt. It is set whenever either a BAR conflict or an AHB bus error occurs as a result of an LPC access. Once set, it remains set until cleared by being written with a 1.

EN_AHB_ERR

When this bit is 0, only a BAR conflict, which occurs when two BARs match the same LPC I/O address, will cause LPC_AHB_ERR to be set. When this bit is 1, AHB bus errors will also cause LPC_AHB_ERR to be set.

BAR_CONFLICT

This bit is set to 1 whenever a BAR conflict occurs on an LPC address. A Bar conflict occurs when more than one BAR matches the address during of an LPC cycle access, and can occur either in LPC I/O addresses or in LPC memory addresses. Once this bit is set, it remains set until cleared by being written with a 1.

RUNTIME_ERR

This bit is set to 1 whenever En_AHB_ERR is 1 and an LPC I/O access causes an AHB bus error. This error will only occur if a BAR is misconfigured. Once set, it remains set until cleared by being written with a 1.

CONFIG_ERR

This bit is set to 1 whenever En_AHB_ERR is 1 and an LPC Configuration access causes an AHB bus error. Once set, it remains set until cleared by being written with a 1.

© 2012-2018 Microchip Technology Inc.

ERROR ADDRESS

This 24-bit field captures the 24-bit AHB address of every LPC transaction whenever the bit LPC_AHB_ERR in this register is 0. When LPC_AHB_ERR is 1 this register is not updated but retains its previous value. When bus errors occur this field saves the address of the first address that caused an error.

6.10.3 EC SERIRQ REGISTER

TABLE 6-10: EC SERIRQ REGISTER

HOST OFFSET	N/A				N/A		**HOST SIZE**	
EC ADDRESS	FF_310Ch				32-bit		**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE[3:1] BIT	**D31**	**D30**	**D29**	**...**		**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R/W
BIT NAME	Reserved							EC_IRQ

EC_IRQ

If the LPC Logical Device is selected as the source for a Serial Interrupt Request by an Interrupt Configuration register (see Section 5.7, "SERIRQ Interrupts," on page 71), this bit is used as the interrupt source.

6.10.4 EC CLOCK CONTROL REGISTER

TABLE 6-11: EC CLOCK CONTROL REGISTER

HOST OFFSET	N/A				N/A		**HOST SIZE**	
EC ADDRESS	FF_3110h				32-bit		**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE[3:1] BIT	**D31**	**D30**	**D29**	**...**		**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R/W	R/W	R/W
BIT NAME	Reserved					Handshake	Clock_Control	

© 2012-2018 Microchip Technology Inc.

CLOCK_CONTROL

This field controls when the host interface will permit the internal ring oscillator to be shut down. The choices are as follows:

0h: The host interface will permit the ring oscillator to be shut down if the LPCPD# signal is asserted (sampled low)

1h: The host interface will permit the ring oscillator to be shut down if the CLKRUN# signals “CLOCK STOP” and there are no pending serial interrupt requests from devices associated with the MEC1632. The CLKRUN# signals “CLOCK STOP” by CLKRUN# being high for 5 LPCCLK’s after the raising edge of CLKRUN#.

2h: The host interface will permit the ring oscillator to be shut down after the completion of every LPC transaction. This mode may cause an increase in the time to respond to LPC transactions if the ring oscillator is off when the LPC transaction arrives at the MEC1632

3h: The ring oscillator is not permitted to shut down as long as the host interface is active.

When the Activate bit in the Activate Register on page 89 is 0, the Host Interface will permit the ring oscillator to be shut down and the Clock_Control Field is ignored. The Clock_Control Field only effects the Host Interface when the Activate bit in the Activate Register is 1.

Although the Host Interface can permit the internal oscillator to shut down, it cannot turn the oscillator on in response to an LPC transaction that occurs while the oscillator is off. In order to restart the oscillator in order to complete an LPC transaction, EC firmware must enable a wake interrupt on the LPC LFRAME# input. See the Application Note in Section 17.3.5, "Wake Capable Interrupts" for details.

HAND SHAKE

This bit controls throughput of LPC transactions.

When this bit is a ‘0’. the MEC1632 only supports a 33MHz PCI Clock. When this bit is a ‘1’, the MEC1632 supports a PCI Clock from 24MHz to 33MHz.

6.10.5 BAR INHIBIT REGISTER

TABLE 6-12: BAR INHIBIT REGISTER

HOST OFFSET	n/a			n/a			**HOST SIZE**	
EC ADDRESS	FF_3120h			64-bit			**EC SIZE**	
POWER	VTR			0000_0000_0000_0000h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D63**	**D62**	**D61**	**...**		**D2**	**D1**	**D0**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	BAR_Inhibit[63:0]							

BAR_INHIBIT[63:0]

When bit *i* of the BAR_Inhibit[63:0] field is asserted (‘1’), where *i* is the logical device number of one of the Host Logical Devices in the MEC1632, the BAR for the associated device is disabled and its addresses will not be claimed on the LPC bus, independent of the value of the Valid bit in the BAR. When bit *i* is not asserted (default), BAR activity for the Logical Device is based on the Valid bit in the BAR.

All of the BAR_Inhibit[63:0] bits are R/W and have no affect on reserved logical device numbers. The Host Logical Devices in the MEC1632 are shown in Table 4-2 in Section 4.3.2, "AHB Address Space," on page 59.

6.10.6 BAR INIT REGISTER

TABLE 6-13: BAR INIT REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC ADDRESS	FF_3130h			16-bit			**EC SIZE**	
POWER	VTR			02Eh			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D15**	**D14**	**D13**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	BAR_Init							

BAR_INIT

This field is loaded into the LPC BAR located at LPC Logical Device Configuration Register offset at offset 60h on nSIO_RESET.

6.10.7 MEM BAR INHIBIT REGISTER

TABLE 6-14: MEM BAR INHIBIT REGISTER

HOST OFFSET	n/a			n/a			**HOST SIZE**	
EC ADDRESS	FF_3140h			64-bit			**EC SIZE**	
POWER	VTR			0000_0000_0000_0000h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D63**	**D62**	**D61**	**...**		**D2**	**D1**	**D0**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Mem BAR_Inhibit[63:0]							

MEM BAR_INHIBIT[63:0]

When bit *i* of the Mem BAR_Inhibit[63:0] field is asserted (‘1’), where *i* is the logical device number of one of the Host Logical Devices in the MEC1632, the BAR for the associated device is disabled and its LPC Memory addresses will not be claimed on the LPC bus, independent of the value of the Valid bit in the BAR. When bit *i* is not asserted (default), BAR activity for the Logical Device is based on the Valid bit in the BAR.

All of the Mem BAR_Inhibit[63:0] bits are R/W and have no affect on reserved logical device numbers. The Host Logical Devices in the MEC1632 are shown in Table 4-2 in Section 4.3.2, "AHB Address Space," on page 59.

© 2012-2018 Microchip Technology Inc.

7.0 POWER, CLOCKS, AND RESETS

7.1 General Description

The Power, Clocks, and Resets chapter includes descriptions of the MEC1632 Clock Generator, Power Configuration and Reset Interface. The Clock Generator, in addition to describing clock sources, also features a Generic Block Clocking Model and a Power Management Interface. The Reset Interface description includes internal and external reset sources, as well as descriptions of an internal 1.8V Regulator and Power Mux. Other descriptions in this chapter include References, a Port List, Interrupt Interface and a Registers Interface.

The Power Configuration, Clock Generator and Reset circuits have the following features:

7.1.1 Power Configuration

TABLE 7-1: Power Configuration FEATURES

Feature	Page
Description of Power Supplies and Clocks ACPI Context	121
Enumerated Power Supply Configurations	123
Power-Up Sequence Definition	124
1.8V Regulator	128
Power Mux	129

7.1.2 Clock Generator

TABLE 7-2: Clock Generator FEATURES

Feature	Page
Three asynchronous clock sources: 20 MHz Oscillator, 32K Clock Domain and PCI_CLK	100
Efficient Logic Design and Controllable Master Clock Trees to Minimize Power Consumption	108
Independent EC-driven Power Management Interface	111
20 MHz Oscillator Optimized for 115.2K baud 16C550A UART Support	–
Generic Block Clocking Model	105
EC-accessible Registers Interface	130

7.1.3 Reset Interface

TABLE 7-3: Reset Interface FEATURES

Feature	Page
VTR and VBAT Reset Signaling (VTRGD, VBAT_POR, nSYS_RST, nEC_RST)	124
RESET Pin Interface	126
VCC Reset Signaling (VCC Power Good)	129
Watch-Dog Timer Forced Reset	130
Interrupt Interface	150

7.2 References

1. Advanced Configuration and Power Interface Specification, Revision 1.0b, February 2, 1999.
2. Intel® 82801DBM I/O Controller Hub 4 Mobile (ICH4-M), Data Sheet, Order Number: 252337-001, Intel Corp., January 2003.
3. PCI Mobile Design Guide, Version 1.1, PCI-SIG, December 18, 1998.

© 2012-2018 Microchip Technology Inc.

7.3 Port List

TABLE 7-4: Power, Clocks, and Resets Port List

Signal Name	Direction	Source	Destination	Description
ARC_CLK_DISABLE	Input	External - Embedded Controller Core (EC)	Internal - EC Power State Controls	Indication for the Power Management Interface (see Section 7.4.10 on page 111) that an EC Sleep instruction has occurred and the processor is sleeping or halted.
SLEEP_STATE	Output	Internal - EC Power State Controls	Internal - 20 MHz Oscillator Control and External Functions as Needed.	System Sleeping State status indicator as described in "EC Power State Controls," on page 115.
SLEEP_FLAG	Output	Clock Control Register	Internal - 20 MHz Oscillator Control, Block Sleep Enables and External Functions as Needed.	Sleep indicator from the Clock Control Register. See also "EC Power State Controls," on page 115 and "Block Sleep Enables," on page 117.
PCI_CLK	Input	External	Internal Test Functions	33 MHz PCI Clock Input (also TEST_CLK_IN).
LRESET#	Input	External	Internal	PCI Reset. See Section 7.7.3, "LPC RESET," on page 129.
WAKE	Input	External	Internal - Wake Interface	Aggregated Wake indicator from the Section 17.0, "EC Interrupt Aggregator," on page 287 for the Power Management Interface (see Section 7.4.10 on page 111).
WDT_ALRT	Input	External	Internal - Reset Interface	Causes a Watch-Dog Timer Forced Reset as described in Section 7.7.4, "Watch-Dog Timer Forced Reset," on page 130.
ALL BLOCK SLEEP ENABLE OUTPUTS	Output	Internal - Block Sleep Enables	External - Sleepable Blocks	To all blocks as defined in Section 7.8.4, "Block Sleep Enable Registers," on page 136. See also "Block Sleep Enables," on page 117
MCLK, EC_BUS_CLK_EN, LPC_BUS_CLK_EN, MCLK_DIV1_EN, MCLK_DIV2_EN, MCLK_DIV4_EN, MCLK_DIV8_EN, MCLK_DIV10_EN, MCLK_DIV16_EN, MCLK_DIV20_EN, MCLK_DIV32_EN, MCLK_DIV64_EN, MCLK_DIV128_EN, MCLK_DIV203_EN, MCLK_5HZ_EN, MCLK_DIV20_EN_HST, X32K_CLK	Output	Internal - MCLK Sourced Clocking and 32K Clock Domain	External	See Section 7.4.11, "MCLK Sourced Clocking," on page 120 and Section 7.4.12, "32K Clock Domain," on page 121.
XTAL1	Input	External	Internal - 32.768 KHz Crystal Oscillator	32.768 KHz Crystal Oscillator crystal input pin.
XTAL2	Output	External	Internal - 32.768 KHz Crystal Oscillator	32.768 KHz Crystal Oscillator crystal output pin.

© 2012-2018 Microchip Technology Inc.

TABLE 7-4: Power, Clocks, and Resets Port List (CONTINUED)

Signal Name	Direction	Source	Destination	Description
32KHZ_OUT	Output	Internal	External	Off-Chip 32.768KHz Oscillator Output (see "32KHz OUTPUT," on page 135).
ALL BLOCK "CLOCK REQUIRED" STATUS BITS	Input	External - Sleep-able Blocks	Internal - 20 MHz Oscillator Control	see Section 7.8.5, "Clock Required Status Registers," on page 140.
nSIO_RESET	Output	Internal	Internal/ External	EC-driven SIO Reset and External System Reset (nRESET_OUT pin). (see "iRESET OUT," on page 133).
VCC_PWRGD	Input	External	Internal -Host Clock Domain	VCC Power Good Input. See Section 7.4.11.4, "Host Clock Domain," on page 120 and Section 7.7.2, "VCC Power Good," on page 129. The EC can determine the state of the VCC_PWRGD signal using VCC PWRGD bit in the Block Sleep Enable Registers. See also Section 7.7.3, "LPC RESET," on page 129.
VCC_PWRGD_BUFF	Output	Internal	External - Pad Buffers	Buffered VCC_PWRGD output used to tri-state VCC-related Pads.
PCR_INT	Output	Internal	External ("EC Interrupt Aggregator," on page 287)	see Section 7.10, "Interrupt Interface," on page 150.
FLASH_PGM	Input	External	Internal - Power-Fail and Reset Status Register	see "FLASH," on page 148.
LPC_RST#	Output	Internal	External (LPC Interface)	see Section 7.7.3, "LPC RESET," on page 129.
VTR	Power Well	External	–	Suspend Supply
VBAT	Power Well	External	–	Battery Supply
VSS	Power Well	External	–	Digital Ground
AGND	Power Well	External	–	Analog Ground for the 32.768 KHz Silicon Oscillator.
VTR_1.8	Power Well	Internal	–	Output of the internal 1.8 V regulator (see Section 7.7, "1.8V Regulator," on page 128).
VTR1.8_BAT	Power Well	Internal	–	Output of the internal Power Mux for VBAT-backed logic (see Section 7.7.1, "Power Mux," on page 129).
nSYS_RST	Output	Internal	Internal/ External	Synchronized VTR Power Good (Section 7.6.6, "nSYS_RST," on page 128).
nEC_RST	Output	Internal	Internal/ External	Stretched nSYS_RST used for EC reset and Registers Interface (see Section 7.6, "Reset Interface," on page 124).
VBAT_POR	Output	Internal	Internal/ External	VBAT Power On Reset (Section 7.6.5, "VBAT_POR," on page 128)
VR_CAP	Power Well	Internal	–	Capacitor Connection for Internal Voltage Regulator (4.7µF ±20%, ESR 2 Ohms, max.) (see also Section 7.7, "1.8V Regulator," on page 128).

TABLE 7-4: Power, Clocks, and Resets Port List (CONTINUED)

Signal Name	Direction	Source	Destination	Description
RESETI#	Input	External	RESET Pin Interface	See Section 7.6.3, "RESET Pin Interface," on page 126.
AVTRGD	Output	Internal	VBAT-Powered Control Interface	Delayed ALLGD signal in FIGURE 7-18: Reset Interface Block Diagram on page 125.

7.4 Clock Generator

7.4.1 OVERVIEW

The MEC1632 Clock Generator includes clock sources as illustrated in Figure 7-1, "Clock Generator Block Diagram": the 20 MHz Oscillator and 32K Clock Domain, as well as the PCI Clock (PCI_CLK in Table 7-4). The relationship of these clock sources to the system power supplies is described in Section 7.4.2, "Power Supplies and Clocking," on page 99; their relationship to the ACPI power states is described in Section 7.5.1, "Power Supplies and Clocks ACPI Context," on page 121.

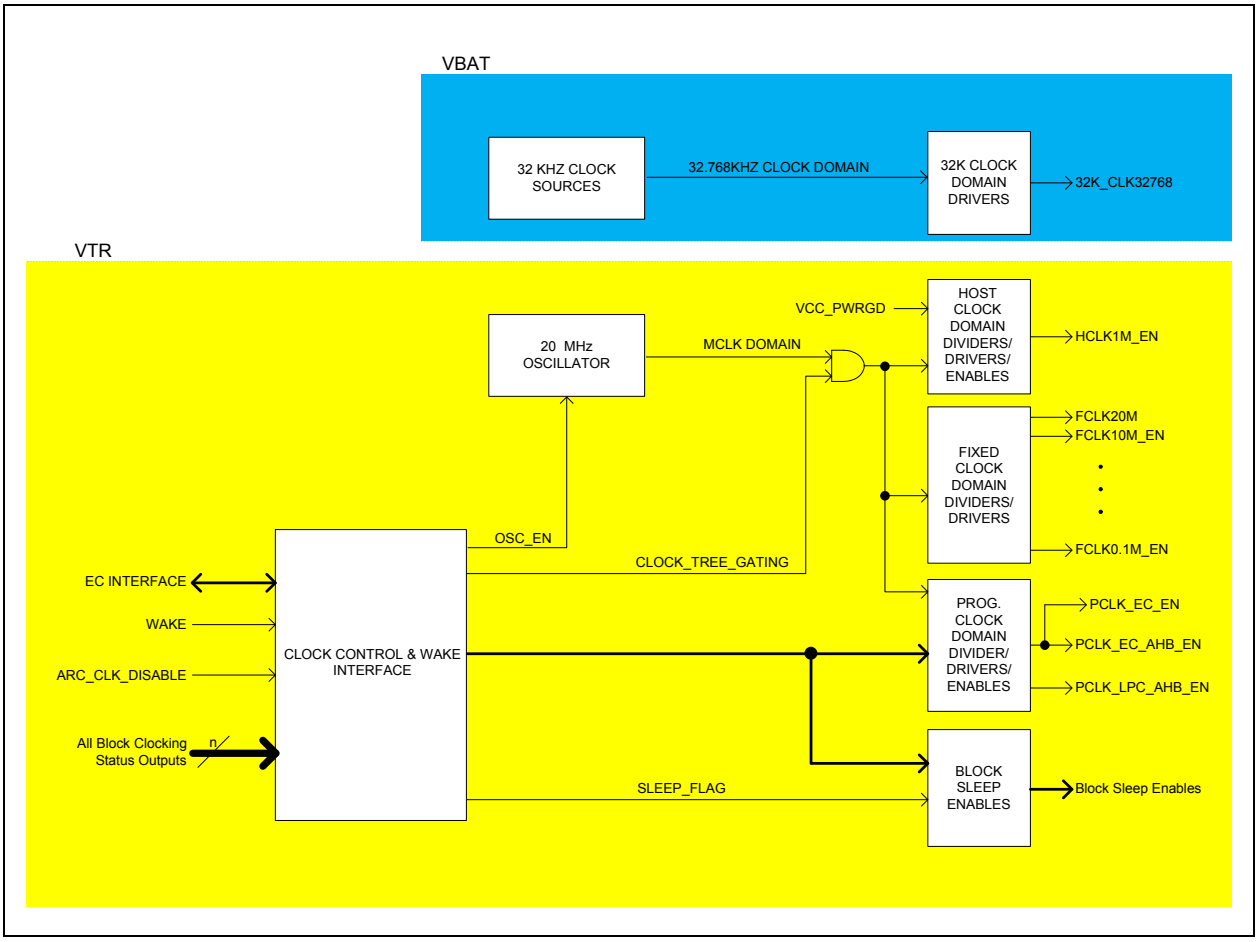
MCLK Sourced Clocking includes Programmable Clock Domains, a Fixed Clock Domain and a Host Clock Domain. The output from the 32.768 KHz Silicon Oscillator, or 32.768 KHz Crystal Oscillator or a 32 KHz external input clock defines a 32K Clock Domain.

The Clock Generator also includes the definition of a Generic Block Clocking Model that provides the foundation for a Power Management Interface. This interface defines several EC Controlled Dynamic Power States that can influence power consumption at the block level and within the Clock Generator.

The Clock Generator includes an EC accessible Registers Interface.

© 2012-2018 Microchip Technology Inc.

FIGURE 7-1: Clock Generator BLOCK DIAGRAM



7.4.2 POWER SUPPLIES AND CLOCKING

Table 7-5 illustrates clocking capabilities versus power supply availability. For more information, see Section 7.5, "Power Configuration," on page 121.

TABLE 7-5: CLOCKS VS. POWER SUPPLIES

Power Supply States (Note 7-1)			Clocks				
VBAT	**VTR**	**VCC**	**PCI Clock**	32.768 KHz Silicon Oscillator	32.768 KHz Crystal Oscillator	**32K External**	20 MHz Oscillator
OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
ON	OFF	OFF		ON/OFF (Note 7-2) OFF	ON/OFF (Note 7-4) OFF	ON/OFF (Note 7-5) OFF	
ON	ON	OFF					ON/OFF (Note 7-3)
ON	ON	ON	ON/OFF				

Note 7-1 power supply states not illustrated in Table 7-5 are undefined (see also Section 7.5, "Power Configuration," on page 121).

Note 7-2 this depends on the 32K_EN bit in the Clock Enable Register.

Note 7-3 there is accuracy adjustment latency as described in Section 7.4.3, "20 MHz Oscillator," on page 100. The 20 MHz Oscillator can be disabled using the Power Management Interface.

Note 7-4 this depends on the XTL_SEL bit in the Clock Enable Register.

Note 7-5 an external single-ended 32.768 KHz clock source may be VTR or VBAT powered. Note that higher than normal VBAT current may occur when VTR transitions from unpowered to powered if the switching threshold on the external single-ended 32.768 KHz clock source is different than the internal Power Mux switch threshold.

7.4.3 20 MHZ OSCILLATOR

The MEC1632 Clock Generator includes a high-accuracy, low power, low start-up latency 20 MHz Oscillator. The 20 MHz Oscillator is always enabled except during power up and in the SYSTEM HEAVY SLEEP 3 and SYSTEM DEEPEST SLEEP state when the oscillator is stopped by hardware as described in Section 7.4.10, "Power Management Interface," on page 111. The 20 MHz Oscillator timing parameters are shown in Table 7-6.

The 20 MHz Oscillator is reset by VTRGD as described in Section 7.6, "Reset Interface," on page 124 and powered by a dedicated voltage regulator and power cycled as described in Section 7.4.10.6, "Wake Interface," on page 118.

FIGURE 7-2: 20 MHz Oscillator BLOCK DIAGRAM

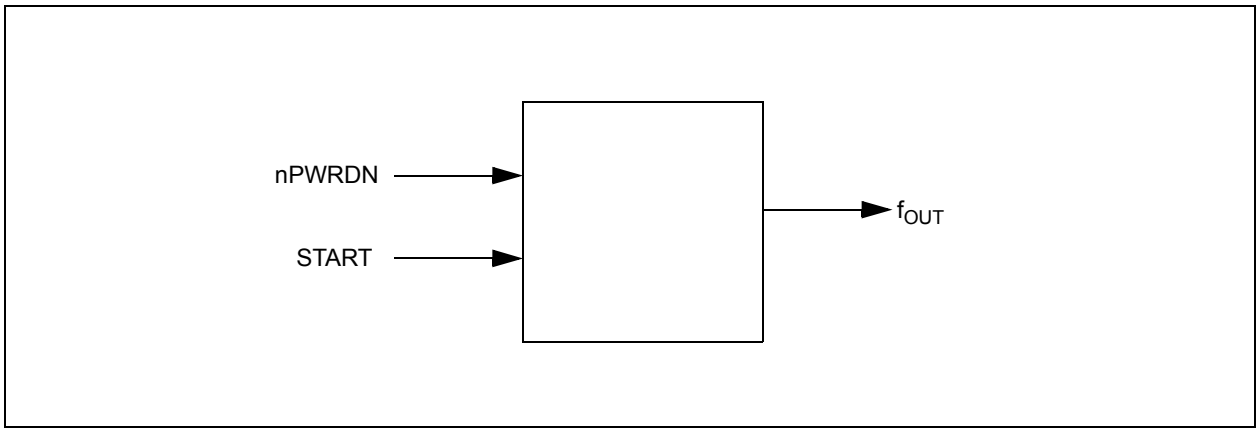
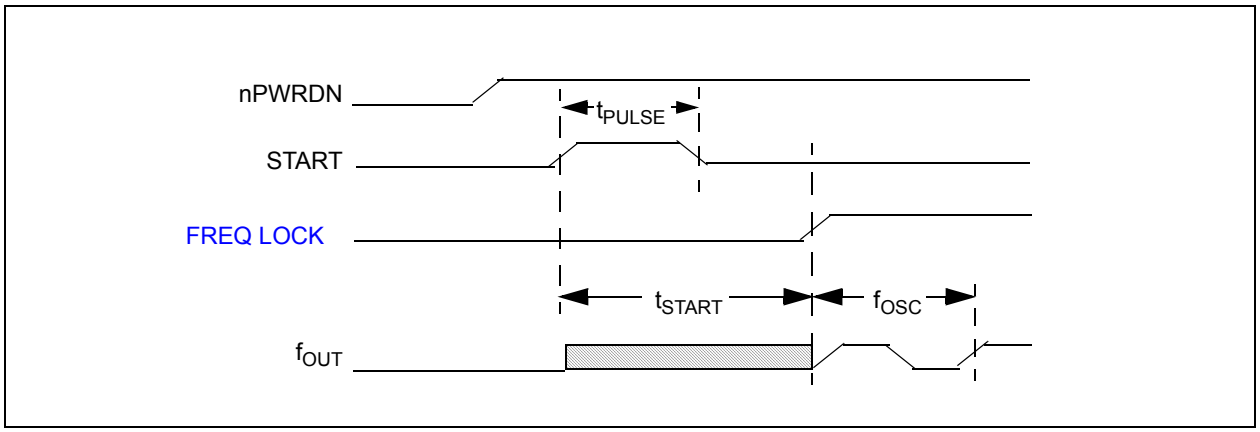


FIGURE 7-3: 20 MHz Oscillator LOCK TIME



© 2012-2018 Microchip Technology Inc.

TABLE 7-6: 20 MHz Oscillator TIMING PARAMETERS

Symbol	Parameter	Limits			Units	Comments
		MIN	NOM	MAX		
f_{OSC}	Oscillator Output Frequency	19.76	20.27	20.78	MHz	± 2% (Commercial) FREQ LOCK = '1' (Note 7-6)
t_{START}	Oscillator Lock Time	–	–	300	µs	
t_{PULSE}	Oscillator Start Pulse Time	2	–	–	µs	

Note 7-6 When FREQ LOCK = '0', MCLK Sourced Clocking is defined by the 10 MHz Ring Oscillator.

7.4.4 32.768 KHZ CRYSTAL OSCILLATOR

The 32.768 KHz Crystal Oscillator provides a high accuracy alternative clock source for the 32K Clock Domain (Figure 7-4). The clock source switching for the 32K Clock Domain is described in Section 7.4.6, "32 KHz Clock Domain Switching," on page 103.

The 32.768 KHz Crystal Oscillator is controlled by the XTL32K_EN bit in the Clock Enable Register (see Section 7.9.2, "Clock Enable Register," on page 149). When XTL32K_EN is asserted, there is a start-up delay (t_{SU}) for the 32.768 KHz Crystal Oscillator as shown in Table 7-7.

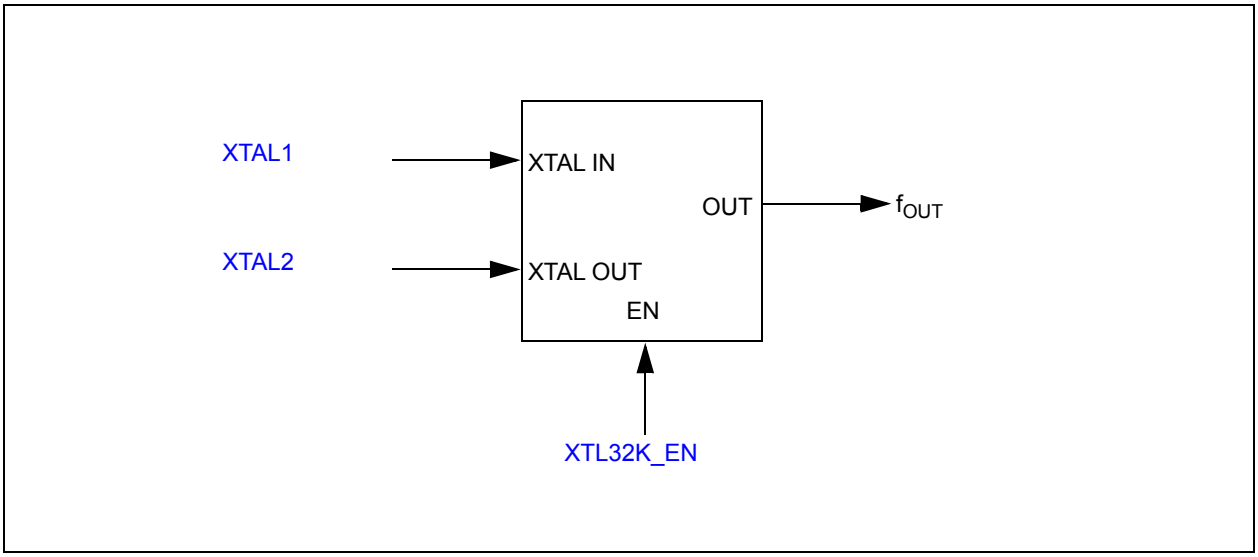
The 32.768 KHz Crystal Oscillator can be driven by an external 32 KHz single-ended clock source as defined by the XOSEL bit in the Clock Enable Register.

If the 32.768 KHz Crystal Oscillator is not used, the XTL32K_EN bit must be '0' and the XTAL1 and the XTAL2 pins should be tied to VSS.

TABLE 7-7: 32.768 KHz Crystal Oscillator TIMING PARAMETERS

Parameter	Symbol	MIN	TYP	MAX	Units
32.768 KHz Crystal Oscillator Start-up Delay	t_{SU}			5	sec.

FIGURE 7-4: 32.768 KHz Crystal Oscillator BLOCK DIAGRAM



7.4.5 32.768 KHZ SILICON OSCILLATOR

7.4.5.1 Description

The 32.768 KHz Silicon Oscillator provides a clock source for the 32K Clock Domain that does not require an external crystal (see also Section 7.4.6, "32 KHz Clock Domain Switching," on page 103).

The 32.768 KHz Silicon Oscillator is controlled by the 32K_EN bit in the Clock Enable Register (see Section 7.9.2, "Clock Enable Register," on page 149). When 32K_EN is asserted, there is a start-up delay (t_{START}) for the 32.768 KHz Silicon Oscillator as shown in Table 7-8.

The clocks sourced by the 32.768 KHz Silicon Oscillator in the 32K Clock Domain operate as described in Table 7-23, "Typical MEC1632 Clocks vs. ACPI Power States" (see Section 7.5.1, "Power Supplies and Clocks ACPI Context," on page 121.

FIGURE 7-5: 32.768 KHz Silicon Oscillator BLOCK DIAGRAM

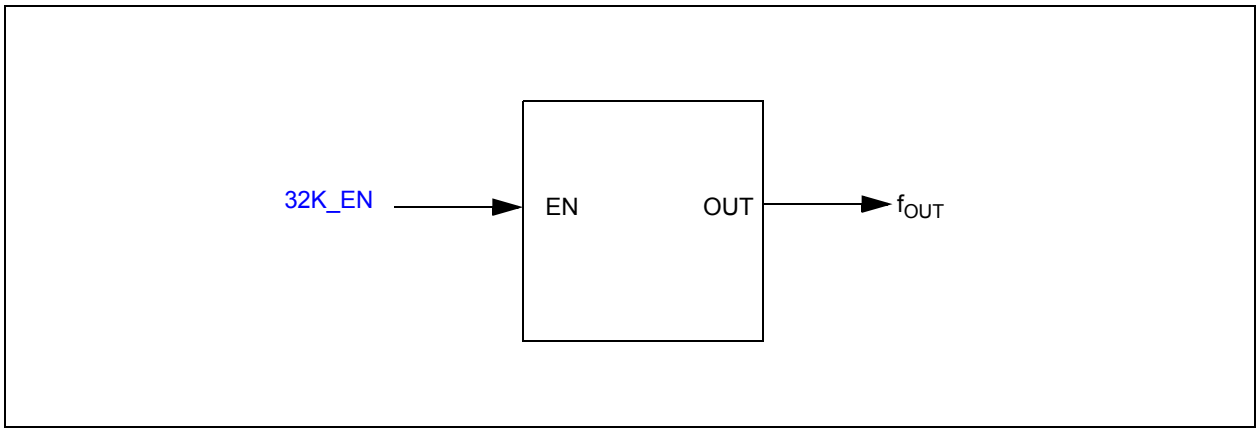


FIGURE 7-6: 32.768 KHz Silicon Oscillator TIMING

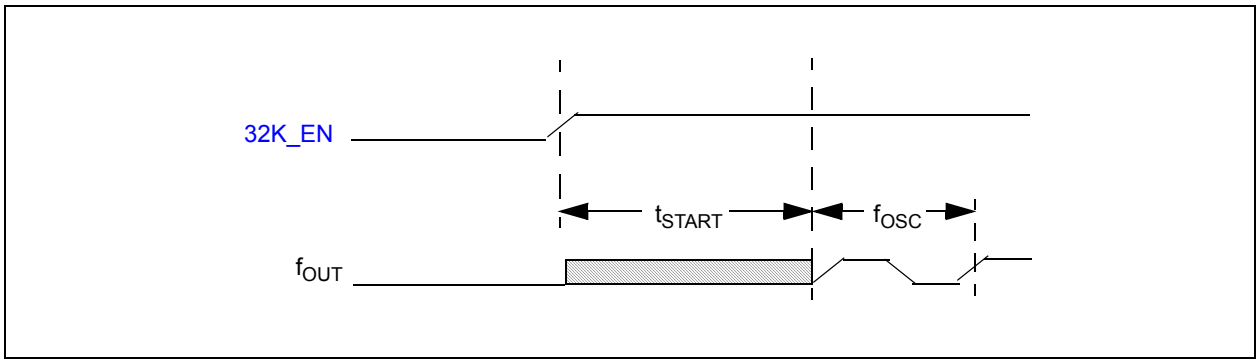


TABLE 7-8: 32.768 KHz Silicon Oscillator TIMING PARAMETERS

Symbol	Parameter	Limits			Units	Comments
		MIN	NOM	MAX		
f_{OSC}	Oscillator Output Frequency	32.113	32.768	33.423	KHz	
t_{START}	Oscillator Lock Time	–	–	300	µs	

© 2012-2018 Microchip Technology Inc.

7.4.6 32 KHZ CLOCK DOMAIN SWITCHING

Clock sources for the 32K Clock Domain are illustrated in Figure 7-7 includes both an internal 32.768 KHz Silicon Oscillator, 32.768 KHz Crystal Oscillator and an external 32 KHz clock source. 32 KHz Clock Domain Switching is controlled as defined in Table 7-9.

The external 32 KHz clock source is configured using the EXT32K_VBAT bit D0 in the Clock Enable Register.

TABLE 7-9: 32KHZ CLOCK DOMAIN CONTROL

XTL_SEL (Note 7-11)	XTL32K_EN (Note 7-10)	AD_EN (Note 7-9)	32K_EN (Note 7-8)	Mode (Note 7-7)	Description	Notes
X	0	0	0	Off	32 KHz clocking (internal/external) disabled (default)	
0	X	0	1	Internal	Internal 32.768 KHz Silicon Oscillator only	
1	1	0	X	Internal	Internal 32.768 KHz Crystal Oscillator only	Note 7-13
X	0	1	0	External	External 32 KHz input (32KHZ_IN pin) only	Note 7-12
0	X	1	1	Auto Switch	Internal 32 KHz clock source used only if external not detected.	
1	1		X			

Note 7-7 The internal 32kHz Clock Source must first be enabled via the Clock Enable Register before switching to the External 32kHz clock source either in the Forced mode (external only) or the auto-switch mode.

Note 7-8 32K_EN is bit D1 in the Clock Enable Register.

Note 7-9 AD_EN is bit D2 in the Clock Enable Register.

Note 7-10 XTL32K_EN is bit D3 in the Clock Enable Register.

Note 7-11 XTL_SEL is bit D4 in the Clock Enable Register.

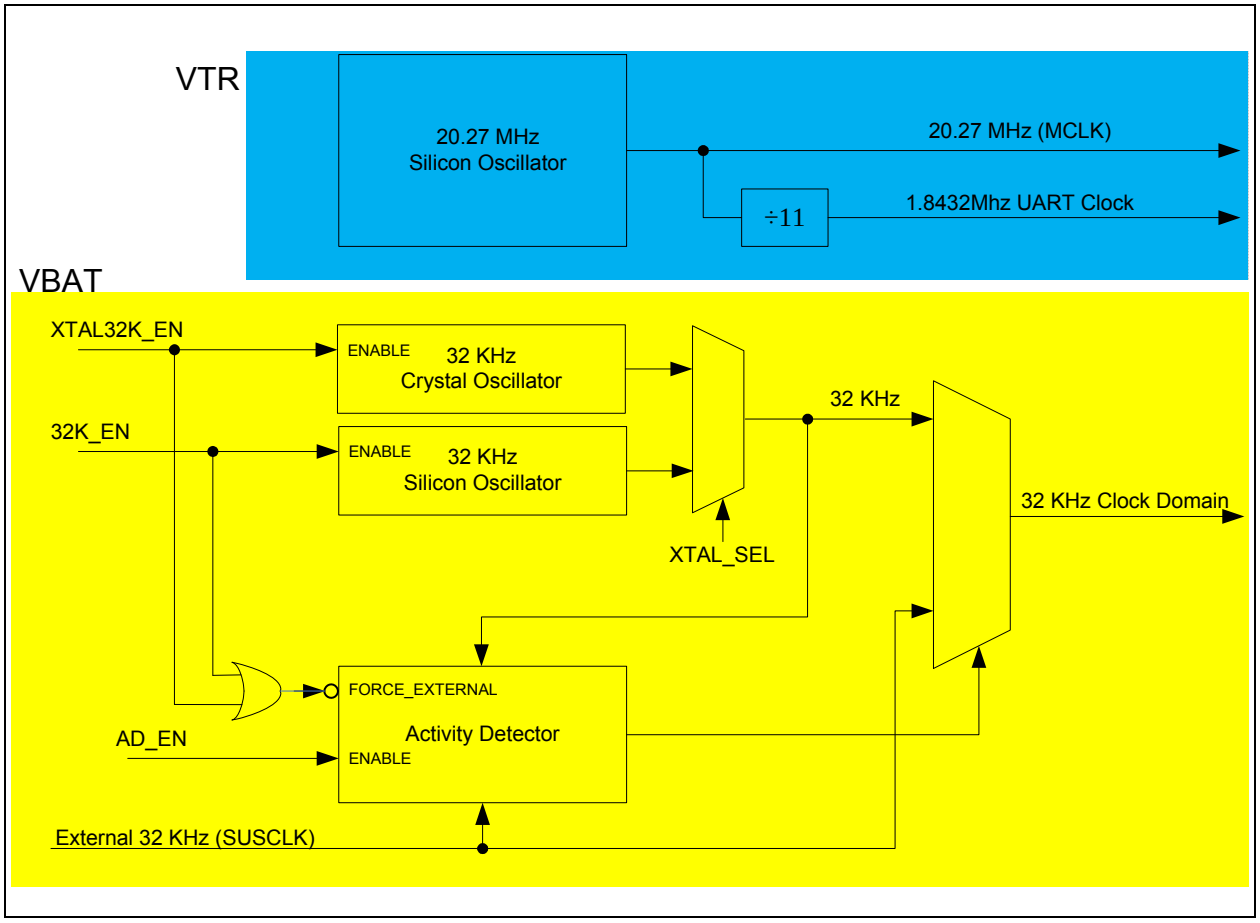
Note 7-12 If AD_EN is asserted (‘1’), the external 32 KHz Oscillator is selected and the external 32 KHz clock stops (i.e., stays high or low for longer than ~31 µs), the external clock must remain stopped for a minimum of 300 µs.

If AD_EN is asserted (‘1’), and the 32K_EN changes from asserted (‘1’) to de-asserted, AD_EN must remain asserted for a minimum of 300 µs.

At least one of 32K_EN or XTL32K_EN must be set to ‘1’ when switching AD_EN from ‘0’ to ‘1’ in order for clock domain switching to take place. Once the 32K domain is switched, 32K_EN or XTL32K_EN can be set to ‘0’.

Note 7-13 The 32.768 KHz Crystal Oscillator can be driven by a crystal or an external 32 KHz single-ended clock source as defined by the XOSEL bit in the Clock Enable Register.

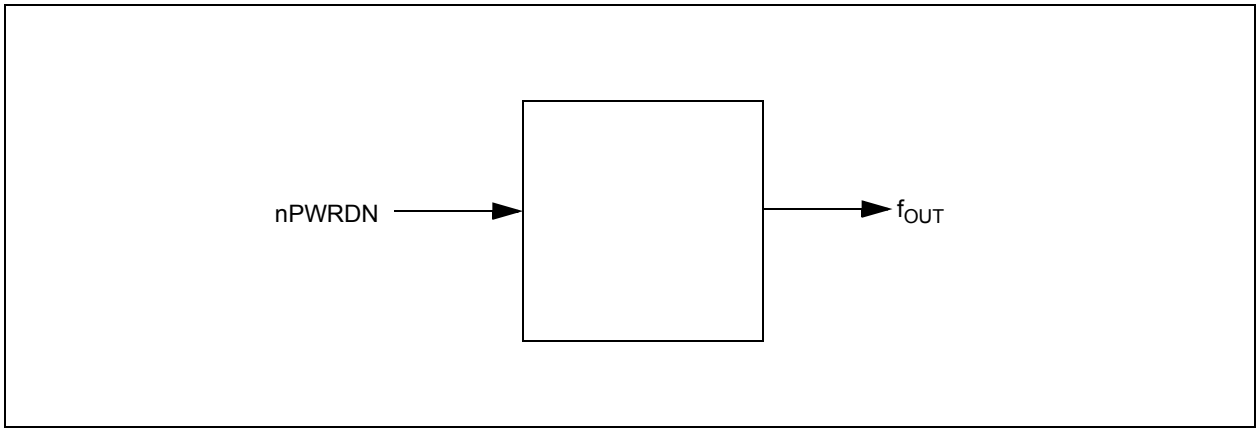
FIGURE 7-7: MEC1632 CLOCKING BLOCK DIAGRAM



7.4.7 10 MHZ RING OSCILLATOR

The MEC1632 Clock Generator includes a low start-up latency 10 MHz Ring Oscillator. The 10 MHz Ring Oscillator is enabled during power up. The 10 MHz Ring Oscillator timing parameters are shown in Table 7-10.

FIGURE 7-8: 10 MHz Ring Oscillator BLOCK DIAGRAM



© 2012-2018 Microchip Technology Inc.

FIGURE 7-9: 10 MHz Ring Oscillator LOCK TIME

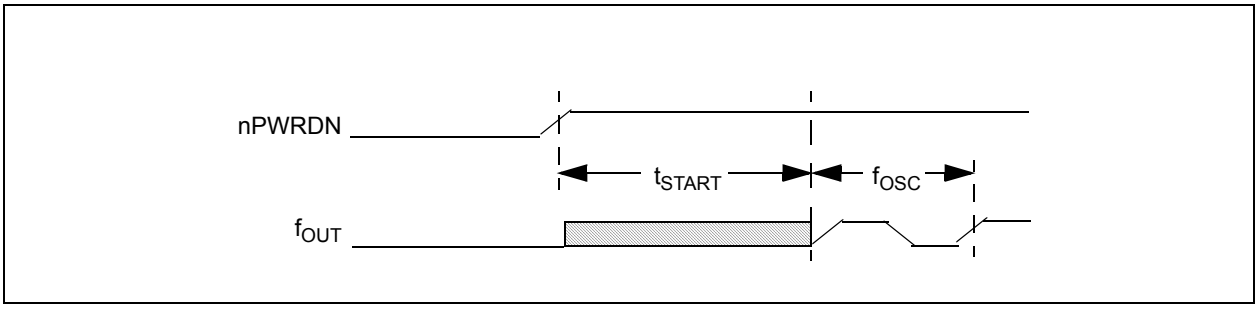


TABLE 7-10: 10 MHz Ring Oscillator TIMING PARAMETERS

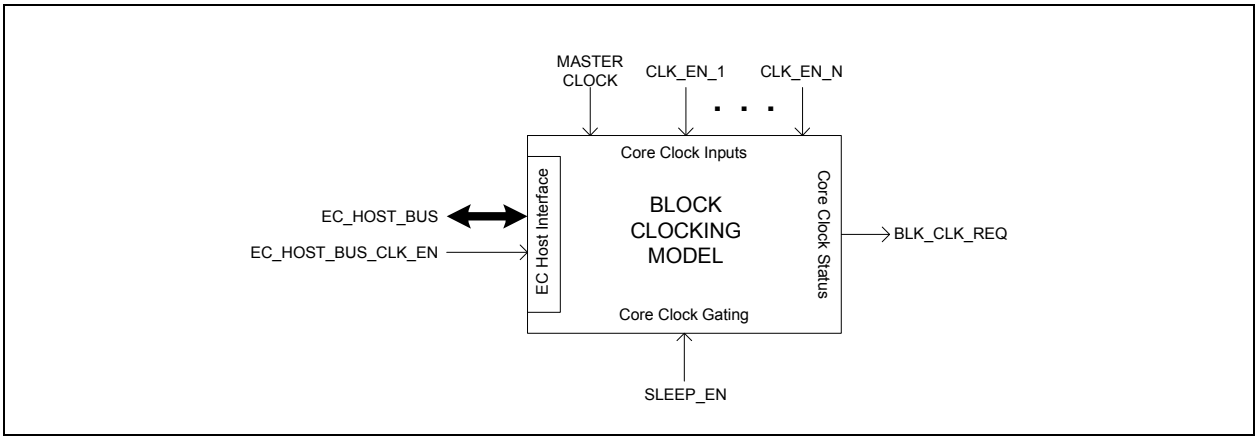
Symbol	Parameter	Limits			Units	Comments
		MIN	NOM	MAX		
f_{OSC}	Oscillator Output Frequency	5	10	15	MHz	
t_{START}	Oscillator Startup Time	–	–	10	µs	

7.4.8 GENERIC BLOCK CLOCKING MODEL

7.4.8.1 Overview

The Generic Block Clocking Model defines the block Clock Gating interface that is assumed by the Clock Generator for all of the MEC1632 internal blocks identified in the Block Sleep Enable Registers. Components of this model are illustrated in Figure 7-10 and described in Section 7.4.8.2. The response of this model to the actions of the Power Management Interface is described in Section 7.4.8.3, "Behavior," on page 107.

FIGURE 7-10: Generic Block Clocking Model ILLUSTRATION



7.4.8.2 Components

As shown in Figure 7-10 the external interface for the Generic Block Clocking Model includes an EC (or other available bus master) host interface, core clock Inputs (which may be clocks or clock enables), a logical core clock gating control and a core clock status output. Not shown in Figure 7-10 is the internal interface for the Generic Block Clocking Model that includes a block enable bit and may also include a block idle status indicator. Each of the Generic Block Clocking Model internal and external interface elements and operational states are described in Table 7-11, "Generic Block Clocking Model Components".

When firmware de-asserts the internal block enable bit the block is disabled and in a minimum power consumption state. Depending on the implementation, the host may need to ensure that the block is not in use before the internal enable bit is de-asserted because it may also function as a reset. Transitions to a minimum block power consumption state while the internal enable bit remains asserted may be requested by the Power Management Interface using an external sleep enable input (see also Section 7.4.8.3, "Behavior," on page 107). In both cases (i.e., when the block is disabled or sleeping), the core clock required status indicator output (BLK_CLK_REQ in Figure 7-10) is de-asserted.

When firmware asserts the internal block enable and the external sleep enable input is not asserted, or the external sleep enable input is asserted but the internal idle indicator is not asserted, the block is operational and in a maximum power consumption state. In both of these cases, the core clock required status indicator output is asserted.

TABLE 7-11: Generic Block Clocking Model COMPONENTS

Internal Enable Bit (Note 7-15)	External SLEEP_EN Input (Note 7-16)	Block Idle Status	Core Clock Required Status Output (Note 7-14)	State	Power	Description
0	X	X	0	DISABLED	MINIMUM	Block is disabled by firmware and the core clock is not needed and gated 'off' internally. Note: it may be up to the host to ensure that the block is not in use before the internal enable bit is de-asserted because the internal enable may also function as a reset when not asserted.
1	0	NOT IDLE	1	FULL POWER	MAXIMUM	The full power state identifies the block normal operation mode where the block is neither disabled by firmware nor commanded to sleep by the Power Management Interface.
		IDLE				
	1	NOT IDLE		PREPARING TO SLEEP		A sleep command has been asserted but the core clock is still required because the block is not idle.
		IDLE	0	SLEEPING	MINIMUM	A sleep command has been asserted, the block is idle and the core clocks are stopped (Note 7-17).

Note 7-14 the "Block Clock Required Status Output" (BLK_CLK_REQ in Figure 7-10) only reflects the core clock requirement; i.e. independent of the host interface clock enable (EC_HOST_BUS_CLK_EN in Figure 7-10). The MEC1632 Generic Block Clocking Model assumes that the block may not remain operational without the host interface clock enable which will not be stopped unless the 20 MHz Oscillator is disabled (see Section 7.4.10, "Power Management Interface," on page 111). The "Clock Required Status" for each block can be seen in the Clock Required Status Registers.

Note 7-15 the internal enable bit (not shown in Figure 7-10) is accessible through the EC Host Interface shown in Figure 7-10 and provides a reset to each block. Typically, as soon as the internal enable bit is de-asserted, the block may be immediately reset and held in the lowest power consumption state.

Note 7-16 The external sleep enables are configured using the Block Sleep Enables as described in "Block Sleep Enables," on page 117.

© 2012-2018 Microchip Technology Inc.

Note 7-17 State transitions on the internal enable bit are undefined in the SLEEPING state and may produce undesirable results.

7.4.8.3 Behavior

The affects of the Power Management Interface on the block sleep Behavior as a result of transitions on the sleep enable input are ignored when the block is disabled. Blocks are typically disabled following a power-on reset (Figure 7-12, "Mode Transitions for Dynamic Power Management").

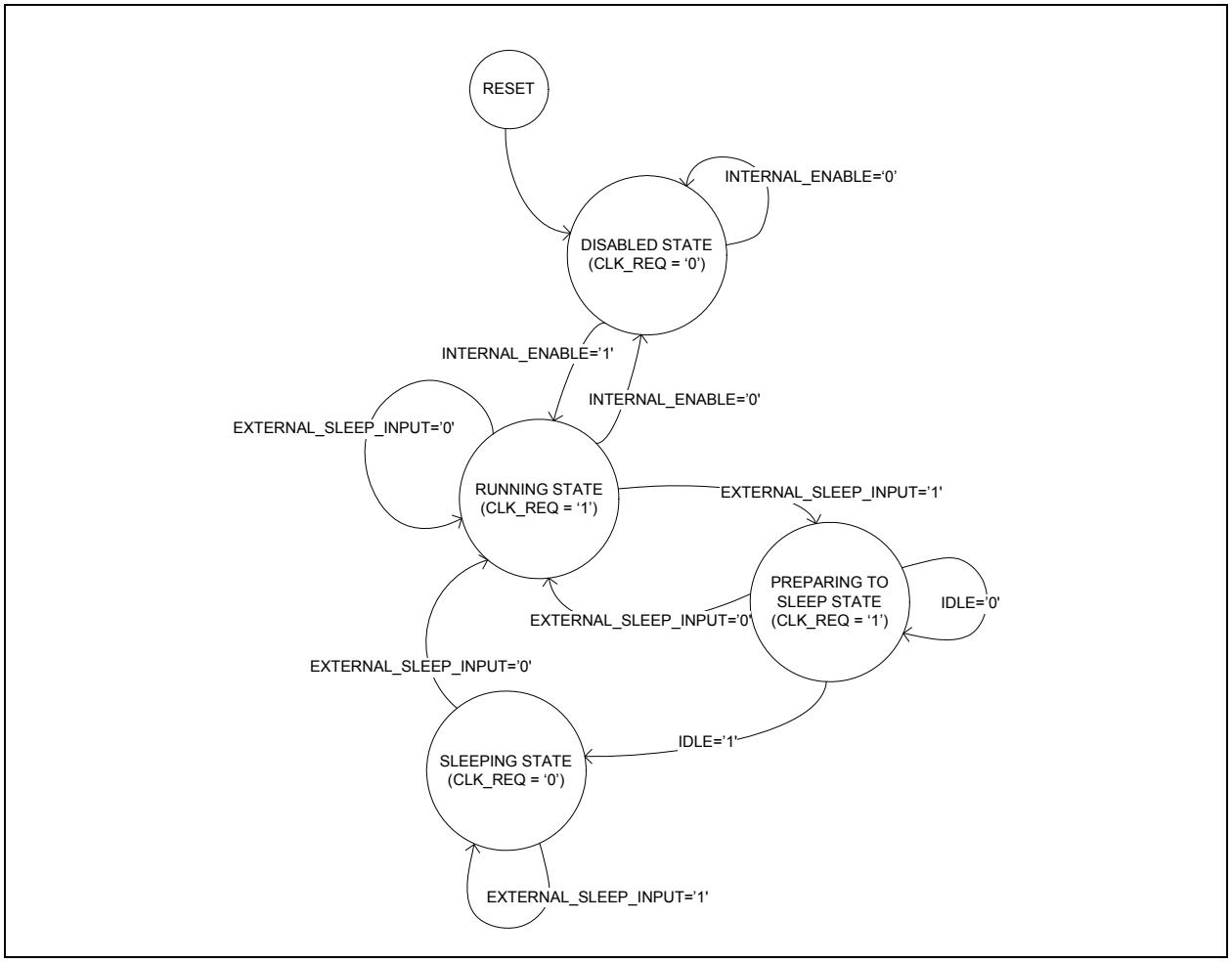
When a block is enabled, a sleep state request as a result of a transition on the sleep enable input must not adversely affect block operation; e.g., by performing illegal operations on an external circuit or corrupting transaction data. As a result, there may be transition latency from the running state to the sleeping state, depending on the nature of the block and its operational state as defined by an internal block 'idle' indicator when the sleep enable is asserted. It is possible that a block may never enter the sleeping state if the block does not idle before the sleep enable input is de-asserted as a result of a wake event.

Once a block enters the sleeping state, internal clocks are gated 'off' and the block is inactive; i.e., outputs are static and the block cannot respond to transitions on inputs, except as defined by the Wake Interface. The transition from the sleeping state to the running state can only occur once the system clocks are running and the sleep enable input is de-asserted.

As described in Section 7.4.8.2, "Components," on page 105, transitions from the running state or the preparing to sleep state to the disabled state may occur without latency depending on the implementation (not shown in Figure 7-12). Transitions from the sleeping state to the disabled state are undefined when the 20 MHz Oscillator is stopped. Transitions from the sleeping state to the disabled state can occur without latency when the 20 MHz Oscillator is running (also not shown in Figure 7-12).

© 2012-2018 Microchip Technology Inc.

FIGURE 7-11: Generic Block Clocking Model CLOCK GATING STATE DIAGRAM EXAMPLE



7.4.9 MASTER CLOCK TREES

7.4.9.1 Description

The master clock derived from the 20 MHz Oscillator is branched into six internal clock trees in the MEC1632. These Master Clock Trees are controlled by register bits in the Clock Tree Control 0 Register and Clock Tree Control 1 Register and function as defined in Section 7.4.9.4, "Control Bit Encoding," on page 110 and Section 7.4.9.2, "Dynamic Clock Tree Gating," on page 108. The Block Allocation Per Clock Tree is defined in Section 7.4.9.3.

Each master clock tree can be forced 'on,' forced 'off,' or dynamically controlled by hardware to help minimize power consumption throughout the EC Controlled Dynamic Power States. See also Section 7.4.10.3, "Clock tree Gating in Heavy Sleep," on page 114.

7.4.9.2 Dynamic Clock Tree Gating

7.4.9.2.1 Overview

Dynamic Clock Tree Gating specifies that a master clock tree is 'on' if any one of the blocks allocated to that tree requires a clock, or if a bus master is accessing registers in that tree. Clock trees are 'off' when the bits in the Clock Required Status Registers that are associated with the blocks in a tree are not asserted and a bus master is not accessing registers in the tree.

© 2012-2018 Microchip Technology Inc.

Dynamic Clock Tree Gating is enabled as described in Table 7-13, “Clock Tree Force Control Bits Encoding,” on page 110 and is enabled by default for the Master Clock Trees (see also Note 7-19).

The EC Interrupt Aggregator and the GPIO Interface are special cases as described in Section 7.4.9.2.2, "INTERRUPT AGGREGATOR CLOCK TREE" and Section 7.4.9.2.3, "GPIO CLOCK TREE".

7.4.9.2.2 INTERRUPT AGGREGATOR CLOCK TREE

The EC Interrupt Aggregator does not conform to the Generic Block Clocking Model, but instead dynamically gates its master clock ‘off’ when all source interrupts and wake-up events are *not* asserted. For the INTERRUPT AGGREGATOR CLOCK TREE to remain ‘off,’ all active interrupts must be cleared at the source.

For example, the PCR_INT interrupt is asserted following a VBAT_POR and must be cleared to enable Dynamic Clock Tree Gating in the INTERRUPT AGGREGATOR CLOCK TREE.

Additionally, when an interrupt from the GPIO Interface is configured as level-sensitive, this interrupt will be continuously asserted when the voltage at the pin changes to the specified level, which will also inhibit Dynamic Clock Tree Gating in this interface. In this instance, Dynamic Clock Tree Gating can only be achieved by setting the GPIO Interface interrupt(s) to edge detect mode.

As shown in Table 7-12, the EC Interrupt Aggregator is part of EC Clock Tree 0. Dynamic Clock Tree Gating in EC Clock Tree 0 cannot occur until the INTERRUPT AGGREGATOR CLOCK TREE is gated ‘off.’

Note: The need for clocking within the EC Interrupt Aggregator is not considered a requirement for keeping the 20 MHz Oscillator running. As a result, the EC Interrupt Aggregator does not include a clock required output to the Clock Generator. If an interrupt is asserted and enabled, the EC cannot sleep, which by default keeps the 20 MHz Oscillator running. If an interrupt is asserted, but not enabled, the EC may sleep when needed and the 20 MHz Oscillator will stop if no other block requires MCLK.

7.4.9.2.3 GPIO CLOCK TREE

The GPIO Interface does not conform to the Generic Block Clocking Model, but instead dynamically gates its master clock ‘off’ when all GPIO Interface interrupts and wake-up events are *not* asserted. To enable Dynamic Clock Tree Gating for the GPIO CLOCK TREE, follow the procedure described for the INTERRUPT AGGREGATOR CLOCK TREE to set the GPIO interrupts to edge detect mode.

As shown in Table 7-12, the GPIO Interface is part of EC Clock Tree 0. Dynamic Clock Tree Gating in EC Clock Tree 0 cannot occur until the GPIO CLOCK TREE is gated ‘off.’

7.4.9.3 Block Allocation Per Clock Tree

The Block Allocation Per Clock Tree is defined in Table 7-12.

Note that clocking for the ARC 625D Embedded Controller is also dynamically controlled as defined in Section 7.4.10, "Power Management Interface," on page 111, but is not considered one of the Master Clock Trees as described here; however, EC Clock Tree 0 remains ‘on’ whenever the ARC 625D Embedded Controller clock is running.

TABLE 7-12: MASTER CLOCK TREE BLOCK ALLOCATION

Clock Tree							
EC Clock Tree 0	**EC Clock Tree 1**	**EC Clock Tree 2**	**EC Clock Tree 3**	**EC Clock Tree 4**	**EC Clock Tree 5**	**Host Clock Tree 0**	**Host Clock Tree 1**
Blinking/Breathing PWM (LED 0-2)	VBAT Register Bank	PWM Controller 2 - 4	Watchdog Timer Interface Register Bank	SMB Device Interface 0 - 3	JTAG Master	Logical Device Configuration Global Register Bank	ACPI Embedded Controller Interface (ACPI-ECI) 2 - 3
GPIO Interface	VBAT Powered RAM	TACH Monitor 0 - 1	PS/2 Device Interface 2	PWM Controller 0 - 1	BC-Link Master A & B	Embedded Flash Subsystem Host Controller	MailBox Register Interface

TABLE 7-12: MASTER CLOCK TREE BLOCK ALLOCATION (CONTINUED)

Clock Tree							
EC Clock Tree 0	**EC Clock Tree 1**	**EC Clock Tree 2**	**EC Clock Tree 3**	**EC Clock Tree 4**	**EC Clock Tree 5**	**Host Clock Tree 0**	**Host Clock Tree 1**
Power, Clocks, and Resets Register Bank	Week Alarm Interface	Analog to Digital Converter	16-Bit Timer Interface Timer 3	16-Bit Timer Interface Timer 1 - 2	RC Identification Detection (RC_ID)	Host Interface (LPC)	Two Pin Serial Port (UART)
Hibernation Timer 0/1 Register Bank	VBAT-Powered Control Interface	PECI Interface	JTAG and Boundary Scan	DMA Controller	PS/2 Device Interface1	ACPI Embedded Controller Interface (ACPI-ECI) 0 - 1	Embedded Memory Interface
16-Bit Timer Interface Timer 0	–	Keyboard Matrix Scan Support	Trace FIFO Debug Port (TFDP)	–	–	8042 Emulated Keyboard Controller	RTC With Date and DST Adjustment
EC Interrupt Aggregator	–	PS/2 Device Interface 0	PWM Controller 5 - 15	–	–	ACPI PM1 Block Interface	Host General Purpose Serial Peripheral Interface (GP-SPI)
Flash Memory	–	EC General Purpose Serial Peripheral Interface (GP-SPI)	TACH Monitor 2 - 5	–	–	Port 80 BIOS Debug Port 0 and 1	–
EEPROM	–	–	Input Capture and Compare Timer	–	–	–	–
SRAM	–	–	HDMI-CEC Interface Controller	–	–	–	–
–	–	–	BC-Link Master D	–	–	–	–

Note 7-18 EC Clock Tree 0 remains ‘on’ whenever the ARC 625D Embedded Controller clock is running. None of the blocks allocated to EC Clock Tree 0 have “Clock Required” Power Management Interface outputs. EC Clock Tree 0 remains ‘off’ whenever the ARC 625D Embedded Controller is sleeping.

7.4.9.4 Control Bit Encoding

TABLE 7-13: CLOCK TREE FORCE CONTROL BITS ENCODING

Force ‘OFF’ Bit	Force ‘ON’ Bit	Description
0	0	Dynamic Clock Tree Gating enabled (Default) (Note 7-19).
0	1	Clock Tree Forced ‘On’
1	0	Clock Tree Forced ‘Off’
1	1	

© 2012-2018 Microchip Technology Inc.

Note 7-19 to shut down EC Clock Tree 0, Host Clock Tree 0 and Host Clock Tree 1 drive the VCC_PWRGD input pin to ground while the VCC_PWRGD signal function is selected (the default setting for this pin).

7.4.10 POWER MANAGEMENT INTERFACE

7.4.10.1 Overview

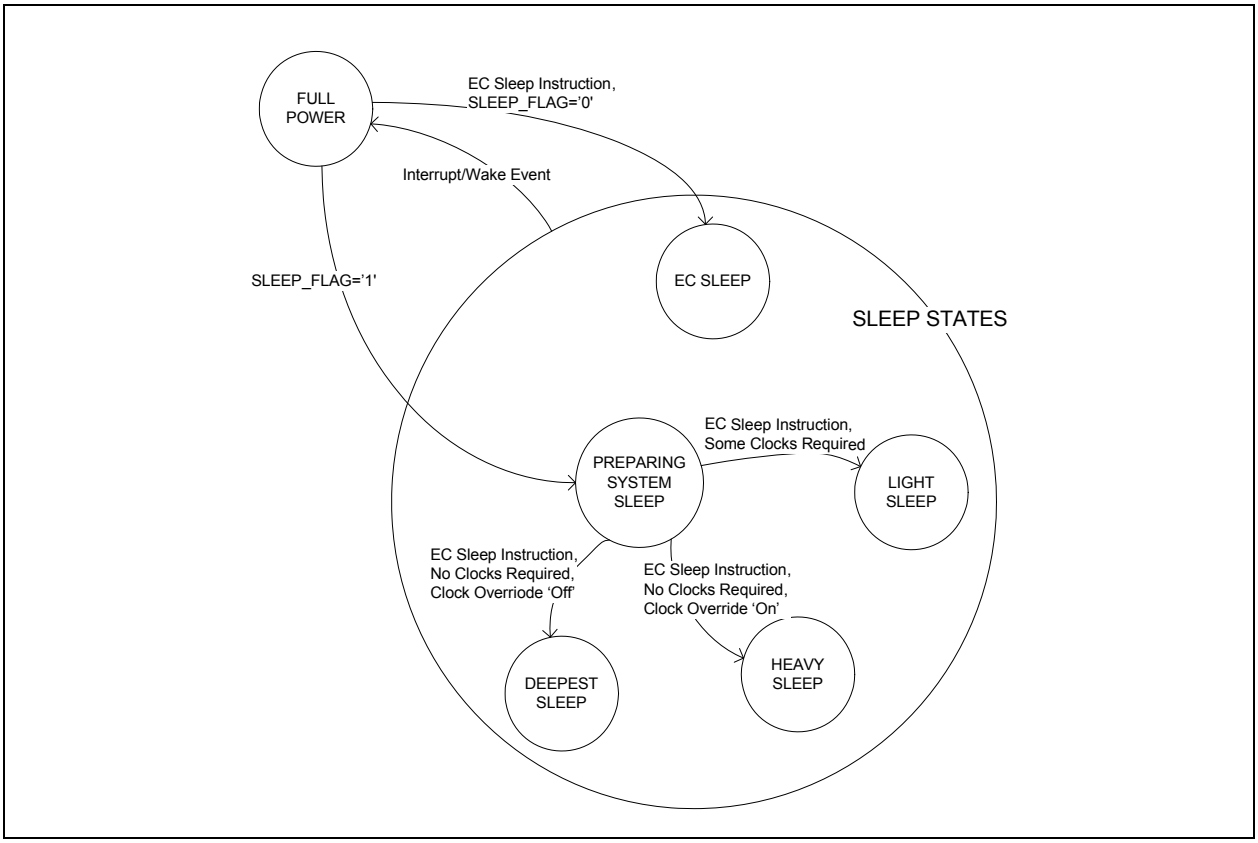
The MEC1632 includes several features to help minimize power consumption, the most intrinsic of which is the application of advanced gate-level low-power design techniques. The EC can also establish the upper run-time power consumption limit by asserting individual internal block enables (i.e., as described in the Generic Block Clocking Model) only for functions that are absolutely required during normal operation. Finally, the EC can also dynamically minimize power consumption by modulating clocks at the block level and within the Clock Generator using the clock gating feature of the Power Management Interface.

APPLICATION NOTE: implementing dynamic power management using individual internal block enables alone is complicated by the fact that, depending on the block implementation, block enables may also perform a reset function.

There are six EC Controlled Dynamic Power States as described in Section 7.4.10.2, "EC Controlled Dynamic Power States". These states are achieved using clock gating as described in Section 7.4.10.5, "Clock Gating," on page 115, which can also affect the 20 MHz Oscillator.

Running and sleeping are the two basic operational modes when considering dynamic power management as defined by this interface (Figure 7-12). Transitions between these modes are deliberate and persistent. For example, to exit the full power state the EC must issue a sleep command as described in "EC Controlled Sleep State Activation," on page 117. Exiting sleep states requires the Wake Interface.

FIGURE 7-12: MODE TRANSITIONS FOR DYNAMIC POWER MANAGEMENT



7.4.10.2 EC Controlled Dynamic Power States

Table 7-14 illustrates the EC Controlled Dynamic Power States that can be achieved using the clock gating feature of Power Management Interface. The EC Controlled Dynamic Power States closely mirror the system power states defined by the Generic Block Clocking Model, but 1) redefine the “preparing to sleep” state to include the affects of the EC sleep state, 2) define additional implementation-specific sleep states that affect Wake Interface latency and 3) indicate the aggregated response of all the MEC1632 power-managed blocks. Typically, the higher the state number in Table 7-14. the greater the system power savings. Traversing the EC Controlled Dynamic Power States requires the SLEEP_FLAG**,** ARC_CLK_DISABLE signal, WAKE signal, the SLEEP CONFIG field, the Block Sleep Enable Registers and the Clock Required Status Registers.

TABLE 7-14: EC CONTROLLED DYNAMIC POWER STATES

Power State Name	SLEEP_FLAG **(**Note 7-20**)**	ARC_CLK_ DISABLE **(**Note 7-21**)**	**Global Block Clock Status (**Note 7-22**)**	SLEEP CONFIG **(**Note 7-23**)**	**Description**
FULL POWER	0	0	X	X	The system is running. This is the highest power consumption state.
EC SLEEP	0	1	X	X	EC has executed a sleep instruction. The rest of the system is unaffected by the EC SLEEP state.
PREPARING SYSTEM SLEEP	1	0	X	X	Sleep commands issued to all sleep-enabled blocks (see "EC Controlled Sleep State Activation," on page 117). The EC can return to the FULL POWER state from PREPARING SYSTEM SLEEP by de-asserting (‘0’) SLEEP_FLAG (see also Note 7-24).

© 2012-2018 Microchip Technology Inc.

TABLE 7-14: EC CONTROLLED DYNAMIC POWER STATES (CONTINUED)

Power State Name	SLEEP_FLAG (Note 7-20)	ARC_CLK_DISABLE (Note 7-21)	Global Block Clock Status (Note 7-22)	SLEEP CONFIG (Note 7-23)	Description
SYSTEM LIGHT SLEEP	1	1	Clock Required	X	System is in a sleeping mode but the 20 MHz Oscillator must remain operating because one or more blocks require a clock.
SYSTEM HEAVY SLEEP 1			Clock Not Required	0	System is in a sleeping mode and no blocks require the clock. There is no clock start up latency following a wake event in SYSTEM HEAVY SLEEP 1 (see Table 7-41, “SLEEP CONFIG Bit Encoding,” on page 135).
SYSTEM HEAVY SLEEP 2				1	System is in a sleeping mode and no blocks require the clock. There is minimal clock start up latency following a wake event in SYSTEM HEAVY SLEEP 2 (see Table 7-41, “SLEEP CONFIG Bit Encoding,” on page 135).
SYSTEM HEAVY SLEEP 3				2	System is in a sleeping mode and no blocks require the clock. There is more clock start up latency following a wake event in SYSTEM HEAVY SLEEP 3 than from SYSTEM HEAVY SLEEP 2 (see Table 7-41, “SLEEP CONFIG Bit Encoding,” on page 135).
SYSTEM DEEPEST SLEEP				3	System is in a sleeping mode and no blocks require the clock. Clock start up latency is the longest and power consumption the lowest in SYSTEM DEEPEST SLEEP (see Table 7-41, “SLEEP CONFIG Bit Encoding,” on page 135).

Note 7-20 SLEEP_FLAG is bit D1 in the Clock Control Register.

Note 7-21 the ARC_CLK_DISABLE signal is described in Table 7-4, “Power, Clocks, and Resets Port List,” on page 96.

Note 7-22 includes the sum of all the “Core Clock Required Status” outputs defined in the Generic Block Clocking Model and aggregated in the Clock Required Status Registers.

Note 7-23 this column refers to the SLEEP CONFIG field in the Clock Control Register.

Note 7-24 In the PREPARING SYSTEM SLEEP state when the sleep enable to a block is asserted, EC register accesses to that block are undefined and should be avoided.

7.4.10.3 Clock tree Gating in Heavy Sleep

The output of the 20 MHz Oscillator is gated 'off' in some of the System Heavy Sleep states as defined in Table 7-15 to conserve power beyond gating the Master Clock Trees as described in Section 7.4.9.

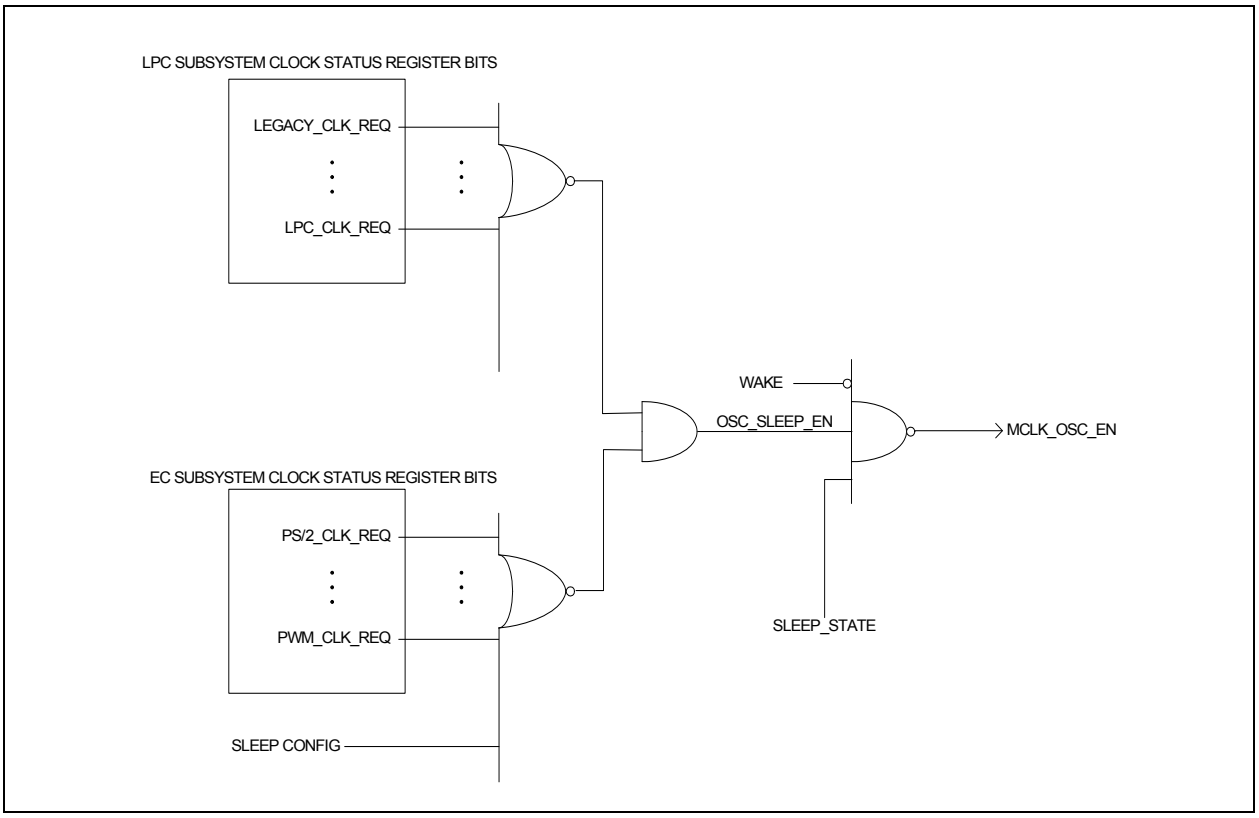
TABLE 7-15: MEC1632 Clock tree Gating in Heavy Sleep IN HEAVY SLEEP STATE

Power States	Clock Required Status Registers	SLEEP CONFIG (Note 7-23)	Description
SYSTEM HEAVY SLEEP 2	No blocks require a clock.	1	Sleep commands issued to all blocks, the EC is sleeping and no blocks require a clock, but the 20 MHz Oscillator remains running to limit start-up latency.
SYSTEM HEAVY SLEEP 3		2	

7.4.10.4 20 MHz Oscillator Control

The 20 MHz Oscillator can only be controlled by the EC using the Power Management Interface, as illustrated in FIGURE 7-13: 20 MHz Ring Oscillator Controls on page 114 and described in Table 7-16, "MCLK_OSC_EN Control".

FIGURE 7-13: 20 MHZ RING OSCILLATOR CONTROLS



© 2012-2018 Microchip Technology Inc.

TABLE 7-16: MCLK_OSC_EN CONTROL

OSC_SLEEP_ENABLE (Note 7-25)	SLEEP_STATE (Note 7-26)	WAKE (Note 7-27)	MCLK_OSC_EN (Note 7-28)	Description
0	X	X	1	The 20 MHz Oscillator is always enabled (“on”) when at least one block requires the clock or the EC has overridden the 20 MHz Oscillator sleep state using the SLEEP CONFIG field in the Clock Control Register.
X	0	X		The 20 MHz Oscillator is always enabled (“on”) when SLEEP_STATE is not asserted.
X	X	1		The 20 MHz Oscillator is always enabled (“on”) when WAKE is asserted.
1	1	0	0	20 MHz Oscillator is disabled (“off”) when SLEEP_STATE is asserted, all blocks do not require the 20 MHz Oscillator, the EC has not overridden the 20 MHz Oscillator sleep state using the SLEEP CONFIG field and WAKE is not asserted.

Note 7-25 OSC_SLEEP_ENABLE is illustrated in Figure 7-13 and indicates the status of all the “Core Clock Required Status” outputs defined in the Generic Block Clocking Model and aggregated in the Clock Required Status Registers and the SLEEP CONFIG field. For a description of the SLEEP CONFIG signaling, see the SLEEP CONFIG field in Section 7.8.3, "Clock Control Register," on page 134.

Note 7-26 see "EC Power State Controls," on page 115 and Table 7-4 for a description of the SLEEP_STATE signal.

Note 7-27 see Section 7.4.10.6, "Wake Interface," on page 118 and Table 7-4 for a description of the WAKE signal.

Note 7-28 MCLK_OSC_EN is the 20 MHz Oscillator enable control illustrated in Figure 7-13.

7.4.10.5 Clock Gating

7.4.10.5.1 Overview

Power savings using the Power Management Interface comes from Clock Gating as described above and in the sub-sections that follow. The magnitude of the power savings depends on the configuration of the Block Sleep Enables, the SLEEP CONFIG field in the Clock Control Register, the configuration of the Master Clock Trees and the operational status of the individual blocks as defined by the Clock Required Status Registers.

7.4.10.5.2 EC Power State Controls

Assuming WAKE is not asserted, when the SLEEP_FLAG bit in the Clock Control Register is asserted by the EC (see also "EC Controlled Sleep State Activation," on page 117), a sleep command is sent to all blocks that are configured for sleep as defined by the Block Sleep Enables. Clock Gating occurs at the block level following the sleep command depending on the state of the block clocking requirement (see Section 7.8.5, "Clock Required Status Registers," on page 140).

Once the SLEEP_FLAG bit is asserted, Clock Gating can only occur within the Clock Generator when the ARC_CLK_DISABLE signal (see Table 7-4, “Power, Clocks, and Resets Port List,” on page 96) is asserted, at which time the SLEEP_STATE signal that can affect 20 MHz Oscillator Control is asserted (Table 7-17). Timing for the EC Power State Controls is defined in Figure 6.7 and Table 6.11.

If WAKE is asserted when EC Controlled Sleep State Activation is attempted, the 20 MHz Oscillator will remain enabled, the SLEEP_FLAG will be automatically de-asserted by hardware as described in Section 7.4.10.6, "Wake Interface," on page 118 and an EC interrupt will occur (not shown in Table 7-17).

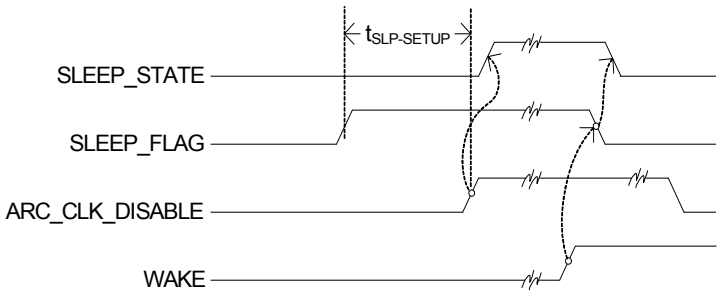
TABLE 7-17: EC Power State Controls DESCRIPTION

ARC_CLK_DISABLE (Note 7-21)	SLEEP_FLAG (Note 7-20)	SLEEP_STATE (Note 7-30)	Status (Note 7-29)	Description
0	0	0	FULL POWER	–
0	1	0	PREPARING SYSTEM SLEEP	SLEEP_FLAG has been asserted by the EC which asserts sleep enables to blocks that have been enabled for sleeping; i.e., clocks in these blocks are turning 'off' in this state. The EC and the 20 MHz Oscillator clock remain active. The PREPARING SYSTEM SLEEP state may represent a lower than FULL POWER system power consumption state.
1	0	0	EC SLEEP	The Clock Generator is unaffected in the EC SLEEP state. The EC SLEEP state may or may not represent a lower system power state than the PREPARING SYSTEM SLEEP state depending on the Block Sleep Enables and the system operational state.
1	1	1	SYSTEM LIGHT SLEEP, SYSTEM HEAVY SLEEP 1, SYSTEM HEAVY SLEEP 2, SYSTEM HEAVY SLEEP 3, SYSTEM DEEPEST SLEEP	The 20 MHz Oscillator may be stopped as described in Section 7.4.10.4, "20 MHz Oscillator Control," on page 114.

Note 7-29 see the system state definitions in Section 7.4.10.2, "EC Controlled Dynamic Power States," on page 112.

Note 7-30 SLEEP_STATE affects the 20 MHz Oscillator as described in Section 7.4.10.4, "20 MHz Oscillator Control," on page 114 and appears in Table 7-4, "Power, Clocks, and Resets Port List," on page 96.

FIGURE 7-14: EC Power State Controls TIMING



© 2012-2018 Microchip Technology Inc.

TABLE 7-18: EC Power State Controls TIMING PARAMETERS

Parameter	Symbol	MIN	TYP	MAX	Units
System Sleep Setup Time	$t_{SLP-SETUP}$	1	–	–	EC Clock

7.4.10.5.3 EC Controlled Sleep State Activation

OVERVIEW

EC Controlled Sleep State Activation depends upon the assertion of the ARC_CLK_DISABLE signal which occurs when the EC executes a sleep instruction. The dynamic sleep states that can be achieved through EC Controlled Sleep State Activation in part depend on the state of the SLEEP_FLAG in the Clock Control Register as defined in the sections below, “Entering and Exiting EC Sleep State” and “Entering and Exiting System Sleep States.” Simultaneous assertions of the WAKE input and EC Controlled Sleep State Activation immediately terminate system sleeps states as defined in Section 7.4.10.6, "Wake Interface," on page 118.

ENTERING AND EXITING EC SLEEP STATE

As illustrated in Section TABLE 7-14:, "EC Controlled Dynamic Power States," on page 112, transitions to the EC SLEEP state occur when the ARC_CLK_DISABLE signal is asserted (i.e., the EC enters the sleep state) while the SLEEP_FLAG in the Clock Control Register is not asserted (‘0’). The EC SLEEP state is terminated when an interrupt to the EC occurs, as described in Section 7.4.10.6, "Wake Interface," on page 118.

The EC enters the sleep state when the SLEEP instruction is issued. The SLEEP instruction halts the CPU pipeline and gates the processor clocks. The clocks to the ARC interrupt logic are not gated. As long as interrupts are enabled in the ARC core STATUS register and at least one interrupt is enabled in the Interrupt Accelerator when the SLEEP instruction is issued, the processor will wake up and process the interrupt service routine when the interrupt triggers. On return from interrupt, the processor will execute the instruction immediately following the SLEEP instruction.

If it is necessary to enable a wakeup interrupt just before entering the sleep state, some care is required to insure that the interrupt does not fire before the SLEEP instruction is issued. The ARC FLAG instruction should be used to enable and disable interrupts and the FLAG and SLEEP instructions should be contiguous to insure that the SLEEP is in the processor pipeline when interrupts are enabled and therefore executes before an interrupt can fire.

For more information on the FLAG and SLEEP instructions, see the *ARCompactä ISA Instruction Set Architecture Programmer’s Reference*.

ENTERING AND EXITING SYSTEM SLEEP STATES

As illustrated in Section TABLE 7-14:, "EC Controlled Dynamic Power States," on page 112, transitions to the system sleep states (i.e., sleep states other than EC SLEEP) occur when the ARC_CLK_DISABLE signal is asserted (i.e., the EC enters the sleep state) while the SLEEP_FLAG in the Clock Control Register is asserted (‘1’). These states include SYSTEM LIGHT SLEEP, SYSTEM HEAVY SLEEP 1 and SYSTEM DEEPEST SLEEP.

The system sleep states are terminated when a wake event occurs as described in Section 7.4.10.6, "Wake Interface," on page 118 Entering and Exiting System Sleep States is similar to Entering and Exiting EC Sleep State. Refer to the section Entering and Exiting EC Sleep State for information about sleeping the EC.

Note that as described in Section 7.4.10.2, "EC Controlled Dynamic Power States," on page 112, the PREPARING SYSTEM SLEEP state in Table 7-14 occurs as soon as the SLEEP_FLAG is asserted, independent of the state of ARC_CLK_DISABLE. The EC can optionally interrogate the Clock Required Status Registers to estimate the depth of the sleep state, for example, before executing a sleep instruction. The EC can return to the FULL POWER state from the PREPARING SYSTEM SLEEP state at any time before executing a sleep instruction by de-asserting the SLEEP_FLAG.

When the ring oscillator is shut down, the oscillator can only be restarted by the Wake Interface or by power cycling the system. In order for the Wake Interface to operate, at least one wake capable interrupt must be enabled in the Interrupt Aggregator. This class of interrupts is described in Section 17.3.5, "Wake Capable Interrupts," on page 294.

7.4.10.5.4 Block Sleep Enables

The Block Sleep Enables allow the EC firmware to determine which blocks will receive sleep commands as a result of EC Controlled Sleep State Activation (see "Entering and Exiting System Sleep States," on page 117). The Block Sleep Enables are configured using the Block Sleep Enable Registers (see Section 7.8.4, "Block Sleep Enable Registers," on

page 136) and behave as illustrated in Figure 7-15, "Block Sleep Enables Example" and Table 7-19, "Block Sleep Enables Definition". There are three Block Sleep Enable Registers: two EC Blocks Sleep Enables Registers (see Section 7.8.4.2 on page 137) and one LPC Blocks Sleep Enables Register (Section 7.8.4.1 on page 136).

FIGURE 7-15: Block Sleep Enables EXAMPLE

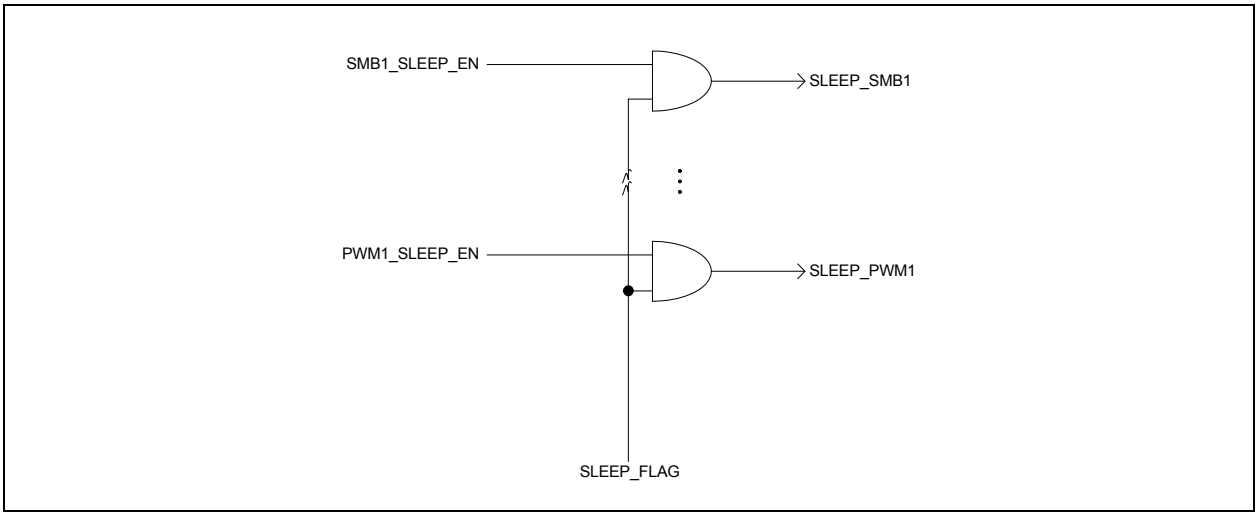


TABLE 7-19: Block Sleep Enables DEFINITION

Sleep Enable (Note 7-31)	SLEEP_FLAG (Note 7-20)	Block SLEEP_EN Signal (Note 7-32)	Description
0	0	0	Block not enabled for sleep, system is in FULL POWER or EC SLEEP state.
	1		Block not enabled for sleep, system is in PREPARING SYSTEM SLEEP or SYSTEM LIGHT SLEEP state.
1	0		Block enabled for sleep, system is in FULL POWER or EC SLEEP state.
	1	1	Block enabled for sleep, system is in PREPARING SYSTEM SLEEP, SYSTEM LIGHT SLEEP, SYSTEM HEAVY SLEEP 1, SYSTEM HEAVY SLEEP 2, SYSTEM HEAVY SLEEP 3 or SYSTEM DEEPEST SLEEP state.

Note 7-31 sleep enable for a single block as defined in Section 7.8.4, "Block Sleep Enable Registers," on page 136.

Note 7-32 Clock Generator sleep enable output signal to a single block (SLEEP_EN) as defined in the Generic Block Clocking Model.

7.4.10.6 Wake Interface

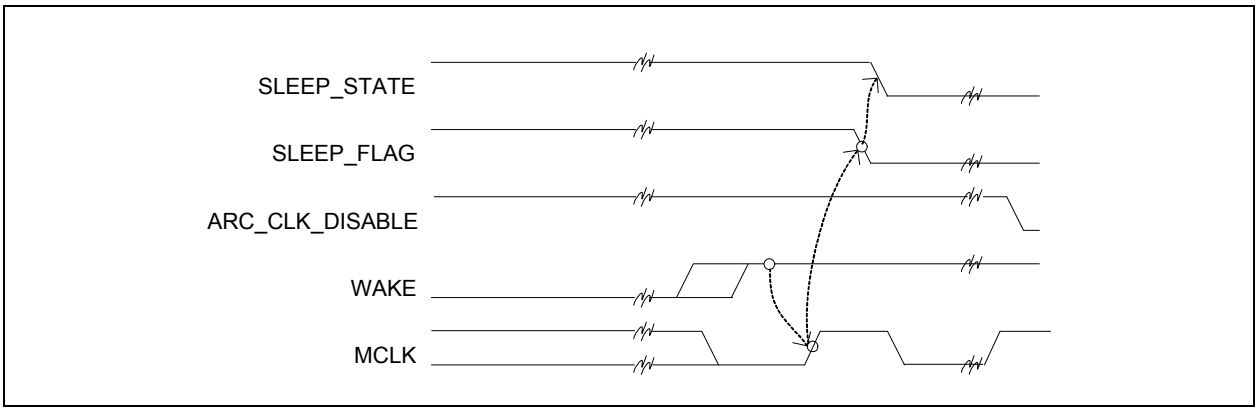
The Wake Interface terminates the sleep states illustrated in Table 7-14, “EC Controlled Dynamic Power States,” on page 1121 and includes interrupts to the EC to transition from the EC SLEEP state, as well as wake events that can restart the 20 MHz Oscillator and terminate the SYSTEM LIGHT SLEEP, SYSTEM HEAVY SLEEP 1, SYSTEM HEAVY SLEEP 2, SYSTEM HEAVY SLEEP 3 and SYSTEM DEEPEST SLEEP states.

© 2012-2018 Microchip Technology Inc.

The WAKE signal shown in Table 7-4, “Power, Clocks, and Resets Port List,” on page 96 is the only required input to the Clock Generator for the Wake Interface. Wake Capable Interrupts, although technically part of the Wake Interface, are external to the Power, Clocks, and Resets function and are not illustrated in this definition. The minimum pulse-width for wake events is characterized in Table 24-1, “GPIO Interrupt/Wake Event Timing Parameters,” on page 390.

When the WAKE signal is asserted while a dynamic system sleep state is enabled, the SLEEP_FLAG in the Clock Control Register is automatically de-asserted by hardware within the first one or two clocks that occur when the 20 MHz Oscillator is re-enabled as illustrated in Figure 7-16. As described in "EC Power State Controls," on page 115 and in "EC Controlled Sleep State Activation," on page 117, system sleep states are essentially ignored if the SLEEP_STATE signal and the WAKE signal are simultaneously asserted.

FIGURE 7-16: Wake Interface TIMING



7.4.10.6.1 Treatment Of Non-wake Capable Interrupts During Sleep Transitions

There are two categories of Interrupts which effect the Wake Interface differently: Wake Capable Interrupts and Non-Wake Capable Interrupts. See Section 17.3.5, "Wake Capable Interrupts," on page 294 in Section 17.0, "EC Interrupt Aggregator".

The lowest power states in which Non-Wake Capable Interrupts can be asserted are PREPARING SYSTEM SLEEP or SYSTEM LIGHT SLEEP. This is because the source of these interrupts require clocks to generate the interrupts. In the event that the EC Controlled Dynamic Power States enters the PREPARING SYSTEM SLEEP or SYSTEM LIGHT SLEEP state by the ARC executing a sleep instruction while a least one non-wake interrupt associated block is active, the EC Controlled Dynamic Power States will not enter a lower state until a non-wake interrupt occurs. The non-wake-able interrupt changes the ARC from a halt state to a running state to process the interrupt. Firmware needs to toggle the state of the SLEEP_FLAG bit in the Clock Control Register. If these bit were Wake-up events, the firmware would not need to de-assert the SLEEP_FLAG bit since hardware will have done so already.

APPLICATION NOTE: if an aggressive sleep policy is implemented and firmware attempts to enter sleep when at least one non-wake interrupt associated block is active, then firmware can treat all interrupt identically (this includes both Wake Capable Interrupts or Non-Wake Capable Interrupts). Firmware can service both Wake Capable Interrupts or Non-Wake Capable Interrupts by attempting to toggle the SLEEP_FLAG in the Clock Control Register first ‘0’ then ‘1’ to prepare to sleep.

APPLICATION NOTE: if a less aggressive sleep policy is implemented and firmware never attempts to enter sleep when any non-wake interrupt associated block is active, then firmware can treat all Wake Capable Interrupts identically. Firmware can service all Wake Capable Interrupts by attempting setting the SLEEP_FLAG bit to ‘1’ in the Clock Control Register to prepare to sleep.

© 2012-2018 Microchip Technology Inc.

7.4.11 MCLK SOURCED CLOCKING

7.4.11.1 Overview

MCLK Sourced Clocking includes all of the Fixed Clock Domain, Host Clock Domain and Programmable Clock Domains that are derived from the 20 MHz Oscillator. Ring Oscillator Sourced Clocking remains active as long as the 20 MHz Oscillator is running (see Section 7.4.10.4, "20 MHz Oscillator Control," on page 114).

All MCLK Sourced Clocking and 32K Clock Domain clocking is summarized in Table 7-20.

TABLE 7-20: ALL Clock Generator OUTPUT PORTS SUMMARY

Symbol	Type (Clock or Enable)	Nominal Frequency	Reference
MCLK	CLOCK	20.27 MHz	Section 7.4.3, "20 MHz Oscillator," on page 100
EC_BUS_CLK_EN	ENABLE	Programmable	"EC Bus Clock," on page 120
LPC_BUS_CLK_EN	ENABLE	20.27 MHz	"LPC Bus Clock," on page 120
MCLK_DIV1_EN	ENABLE	20.27 MHz	Section 7.4.11.3, "Fixed Clock Domain," on page 120
MCLK_DIV2_EN	ENABLE	10.14 MHz	
MCLK_DIV4_EN	ENABLE	5.07 MHz	
MCLK_DIV8_EN	ENABLE	2.53 MHz	
MCLK_DIV10_EN	ENABLE	2.03 MHz	
MCLK_DIV16_EN	ENABLE	1.27 MHz	
MCLK_DIV20_EN	ENABLE	1.01 MHz	
MCLK_DIV32_EN	ENABLE	0.633 MHz	
MCLK_DIV64_EN	ENABLE	317 KHz	
MCLK_DIV128_EN	ENABLE	158 KHz	
MCLK_DIV203_EN	ENABLE	99.852 KHz	
MCLK_5HZ_EN	ENABLE	5 Hz	
MCLK_DIV20_EN_HST	ENABLE	1.01 MHz	Section 7.4.11.4, "Host Clock Domain," on page 120
X32K_CLK	CLOCK	32.768 KHz	Section 7.4.5, "32.768 KHz Silicon Oscillator," on page 102

7.4.11.2 Programmable Clock Domains

7.4.11.2.1 EC Bus Clock

The EC Bus Clock (EC_BUS_CLK_EN) is a programmable clock that is derived from the Block Sleep Enable Registers as described in Section 7.8.4 on page 136.

7.4.11.2.2 LPC Bus Clock

The LPC Bus Clock (LPC_BUS_CLK_EN) is a fixed clock.

7.4.11.3 Fixed Clock Domain

The Fixed Clock Domain represents non-programmable clocks that are derived from the 20 MHz Oscillator. The Fixed Clock Domain outputs as shown in Table 7-20 are MCLK, MCLK_DIV2_EN, MCLK_DIV4_EN, MCLK_DIV10_EN, and MCLK_DIV203_EN.

7.4.11.4 Host Clock Domain

The Host Clock Domain includes clocking for the Legacy Port Functions. Host Clock Domain clock gating is controlled by VCC_PWRGD such that when VCC_PWRGD is not asserted (‘0’), clocks in the Host Clock Domain are ‘off.’

When VCC_PWRGD is asserted (‘1’) clocks in the Host Clock Domain are ‘on’ and the Legacy Port Functions may be affected by the EC Controlled Dynamic Power States as described in Section 7.4.10, "Power Management Interface," on page 111.

© 2012-2018 Microchip Technology Inc.

The Host Clock Domain output as shown in Table 7-20 is MCLK_DIV20_EN_HST.

7.4.12 32K CLOCK DOMAIN

7.4.12.1 Overview

The 32K Clock Domain represents all of the clocking derived from the 32.768 KHz Silicon Oscillator, the 32.768 KHz Crystal Oscillator or the external single-ended 32KHz clock input. The output 32.768 KHz clock source is synchronized to the 20 MHz Oscillator as described below in Section 7.4.12.2, "Synchronization".

The 32K Clock Domain remains active as long as the selected 32.768 KHz clock source is running (see "32K_EN," on page 149). The blocks driven by the 32K Clock Domain are summarized in Table 7-21 and are not affected by Clock Gating in the Power Management Interface. Typically, blocks driven by the 32K Clock Domain can generate wake events, even when the 20 MHz Oscillator is disabled.

7.4.12.2 Synchronization

The 32K Clock Domain X32K_CLK output is synchronized to the 20 MHz Oscillator. Synchronization is disabled under two conditions: 1) when the 20 MHz Oscillator is disabled (e.g., in the SYSTEM DEEPEST SLEEP state or during a test mode) and 2) when VTRGD is not asserted (see Section 7.6.4, "VTRGD," on page 126).

7.4.12.3 Summary

The distribution of the 32K Clock Domain throughout the MEC1632 is illustrated in Table 7-21, below.

TABLE 7-21: 32K Clock Domain DRIVEN BLOCKS

Block Name	Block Cross Reference
Week Timer	Section 22.0, "Week Alarm Interface," on page 370
Watch-Dog Timer	Section 18.0, "Watchdog Timer Interface," on page 330
Hibernation Timer 0	Section 21.0, "Hibernation Timer," on page 366
Hibernation Timer 1	
LED Interface	Section 36.0, "Blinking/Breathing PWM," on page 510
Watch-Dog Timer Forced Reset	Section 7.7.4, "Watch-Dog Timer Forced Reset," on page 130
20 MHz Oscillator	Section 7.4.3, "20 MHz Oscillator," on page 100 (Note 7-33).

Note 7-33 Synchronization as described in Section 7.4.12.2 does not apply to the output of the 32.768 KHz Silicon Oscillator that is applied to the 20 MHz Oscillator.

7.5 Power Configuration

7.5.1 POWER SUPPLIES AND CLOCKS ACPI CONTEXT

The MEC1632 is influenced by three separate power planes, VBAT, VTR, and VCC, as described in Table 6.15. The VBAT and VTR power planes provide power directly to the MEC1632 through the VBAT and VTR pins shown in Table 7-4, "Power, Clocks, and Resets Port List". The MEC1632 senses the VCC power state using the VCC_PWRGD input pin. The VBAT, VTR, and VCC power sequencing requirements are as follows (see also Section 7.5.3, "Power-Up Sequence," on page 124):

1. VCC power can be applied simultaneously with or after VTR power.
2. VTR power can be applied simultaneously with or after VBAT.

The typical relationships of the MEC1632 power supplies to the system power states is shown below in Table 6.15. The distribution of the MEC1632 power supplies to the various functional blocks is illustrated in FIGURE 2-1: MEC1632 Top-level Block Diagram on page 8.

The typical relationships of the MEC1632 clocks to the system power states is shown below in Table 6.16. Descriptions of the various clock domains can be found in Section 7.4, "Clock Generator," on page 98.

TABLE 7-22: TYPICAL MEC1632 POWER SUPPLIES VS. ACPI POWER STATES

	ACPI Power State						
Supply Name	**S0 (FULL ON)**	**S1 (POS)**	**S3 (STR)**	**S4 (STD)**	**S5 (SOFT OFF)**	**G3 (MECH OFF)**	**Description**
VBAT	ON	ON	ON	ON	ON	ON	MEC1632 VBAT Well Supply (assuming a TYPE 2 configuration as described in Section 7.5.2, "Power Supply Configurations," on page 123)
VTR	ON	ON	ON	ON/OFF	ON/OFF	OFF	MEC1632 Suspend Supply. (Note 7-34)
VCC	ON	ON	OFF	OFF	OFF	OFF	MEC1632 Runtime Supply (Note 7-35)

Note 7-34 VTR availability in S4 - S5 may depend, for example, on whether AC power is available.

Note 7-35 the MEC1632 senses the VCC power state using the VCC_PWRGD input pin; i.e., VCC power is not directly applied to this device.

TABLE 7-23: TYPICAL MEC1632 CLOCKS VS. ACPI POWER STATES

	ACPI Power State						
Clock Name	**S0 (FULL ON)**	**S1 (POS)**	**S3 (STR)**	**S4 (STD)**	**S5 (SOFT OFF)**	**G3 (MECH OFF)**	**Description**
32K Clock Domain (External Clock Source)	ON	ON	ON	ON	ON	OFF/ ON	32K Clock Domain is driven by an external source, for example, by an Intel ICH4M SUSCLK, the 32K Clock Domain is running whenever RSMRST is not asserted. (see Ref[6]).
32K Clock Domain (Internal Clock Source)	ON	ON	ON	ON	ON	ON	32K Clock Domain is running whenever VBAT is fully powered except following a VBAT_POR as described in "32K_EN," on page 149.
Host Clock Domain	ON	ON	OFF	OFF	OFF	OFF	The Host Clock Domain is gated by the MEC1632 runtime supply (VCC) as described in Section 7.4.11.4, "Host Clock Domain," on page 120.
PCI_CLK	ON/OFF	ON/OFF	OFF	OFF	OFF	OFF	33MHz LPC Bus clock input powered by the MEC1632 runtime supply (VCC). (Note 7-36)
Programmable Clock Domains and Fixed Clock Domain	ON/OFF	ON/OFF	ON/OFF	OFF/ON	OFF/ON	OFF	These clocks are powered by the MEC1632 suspend supply (VTR) but may start and stop as described in Section 7.4.10, "Power Management Interface," on page 111. (see also Note 7-34)

Note 7-36 the PCI_CLK can start and stop in S0/S1 as defined in Ref[7].

© 2012-2018 Microchip Technology Inc.

7.5.2 POWER SUPPLY CONFIGURATIONS

There are two acceptable types of MEC1632 power supply configuration that fundamentally differ based on the need for a backup battery connection to VBAT. In both cases VTR is connected to the suspend supply as described in Section 7.5.1, "Power Supplies and Clocks ACPI Context," on page 121.

7.5.2.1 TYPE 1

TYPE 1 configurations do not use a VBAT backup battery connection, VBAT is tied to VTR and the Reset Interface generates a VBAT_POR whenever VTR powers up as described in Section 7.6.5, "VBAT_POR," on page 128. In TYPE 1 configurations the VBAT-Powered Control Interface is fully operational when VTR is powered.

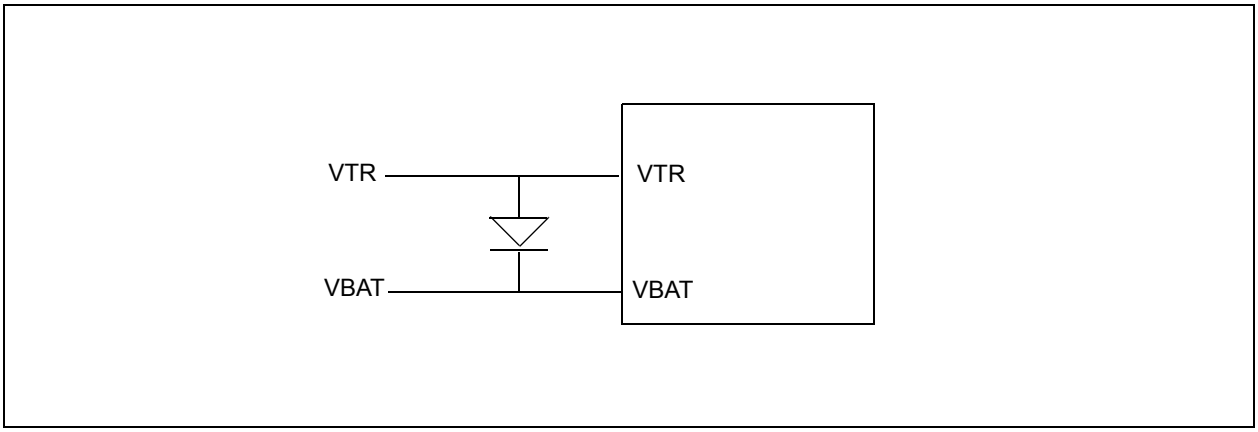
7.5.2.2 TYPE 2

TYPE 2 configurations use a VBAT backup battery connection. Power supply requirements for TYPE 2 configurations are as follows: VBAT is connected to a backup battery that is externally switched through a diode with VTR (Figure 7-17).

In this configuration some internal components that utilize the VBAT power plane are switched internally to VTR using a Power Mux when VTR power is applied as described in Section 7.7.1, "Power Mux," on page 129.

In TYPE 2 configurations, the VBAT-Powered Control Interface can be used to power-on an unpowered system.

FIGURE 7-17: TYPE 2 POWER SUPPLY CONFIGURATION



7.5.3 POWER-UP SEQUENCE

Table 7-24 summarizes the MEC1632 Power-Up Sequence. For information regarding the typical relationships of the MEC1632 power supplies to the system power states see Section 7.5, "Power Configuration," on page 121.

TABLE 7-24: Power-Up Sequence

	VBAT	VTR	VCC	Reset Interface	Description
1.	OFF	OFF	OFF	–	MEC1632 fully unpowered
2.	ON	OFF	OFF	–	32.768 KHz Silicon Oscillator may be disabled as described in Section 7.6.5, "VBAT_POR," on page 128.
3.	ON	ON	OFF	VBAT_POR, VTRGD, nSYS_RST. nEC_RST, nSIO_RESET	VBAT-powered registers may be reset as described in Section 7.6.5, "VBAT_POR," on page 128. VTR-powered registers and blocks reset (nSYS_RST). EC held in reset as described in Section 7.6.7, "nEC_RST," on page 128.
					nSIO_RESET asserted.
					20 MHz Oscillator enabled.
					EC begins code execution following a delay as described in Section 7.6.7, "nEC_RST," on page 128.
4.	ON	ON	ON	VCC_PWRGD, nSIO_RESET	Registers affected by VCC_PWRGD are reset (Note 7-37)
					Firmware de-asserts nSIO_RESET as described in "iRESET OUT," on page 133.

Note 7-37 for VTR-powered on-chip registers that are reset by VCC_PWRGD, it is important that firmware not write to any of these registers until 1 ms following the assertion of VCC_PWRGD (‘1’).

7.6 Reset Interface

7.6.1 OVERVIEW

The primary function of the Reset Interface (Figure 7-18) is to generate VTR and VBAT reset signaling; including, VTRGD, VBAT_POR, nSYS_RST, nEC_RST and Watch-Dog Timer Forced Reset (Table 7-25). The Reset Interface also includes section regarding a RESET Pin Interface, and the 1.8V Regulator. There is other VCC-related Reset Interface functionality not shown in Figure 7-18 that is described Section 7.7.2, "VCC Power Good," on page 129 and Section 7.7.3, "LPC RESET," on page 129.

Also included in the Reset Interface are descriptions of the Power Mux and Registers Interface. These are related Power, Clocks, and Resets functions that are not described anywhere else in this chapter.

TABLE 7-25: VTR/VBAT RESET THRESHOLDS

Parameter	MIN	TYP	MAX	Units	Notes
VTRGD Reset Threshold	0.5	1.8	2.7	Volts	
VBAT_POR Reset Threshold	0.5	1.25	1.9		

7.6.2 STRAP OPTIONS

TABLE 7-26: STRAP OPTIONS

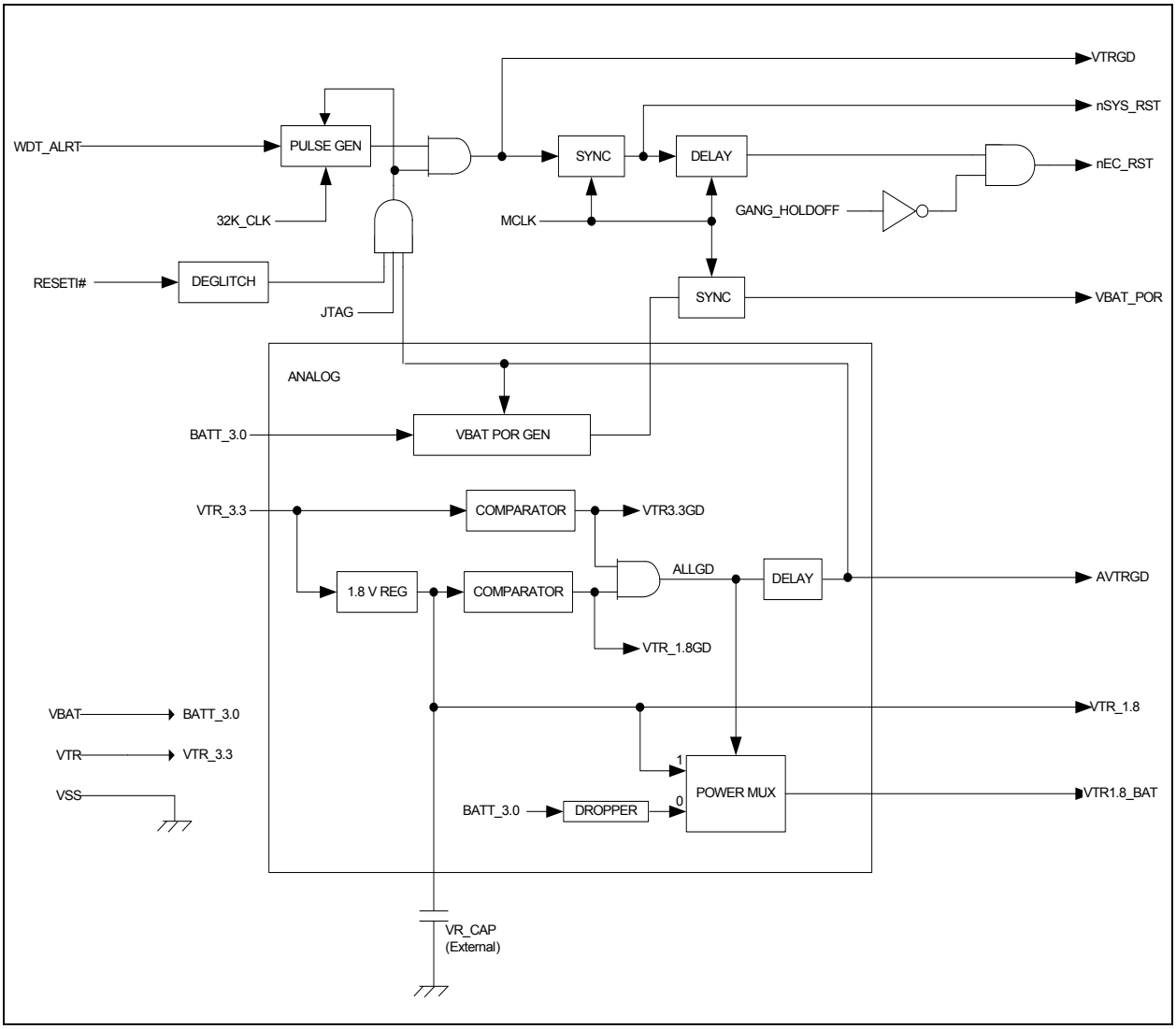
Name	Pin	Notes
Boundary Scan	GPIO171/MSDATA	Note 7-38
UART_BOOT_STRAP	GPIO166	Note 7-39

Note 7-38 This strap option is sampled on VTR power up, and is not affected by RESETI# or a WDT reset.

Note 7-39 This GPIO pin is sampled by the Boot ROM code following a VTR power up, and when RESETI# is asserted.

© 2012-2018 Microchip Technology Inc.

FIGURE 7-18: Reset Interface BLOCK DIAGRAM



7.6.3 RESET PIN INTERFACE

7.6.3.1 RESETI#

As illustrated in Figure 7-19, when asserted ('0') the RESETI# input can force the equivalent of a VTRGD to the MEC1632. The RESETI# input includes an analog glitch filter (Note 7-40). Timing for the RESETI# input is shown in Figure 7-19 and Table 7-27.

The RESETI# input also affects the JTAG interface as defined in Section 44.7.3.1, "Async JTAG RESET," on page 586.

FIGURE 7-19: RESETI# TIMING

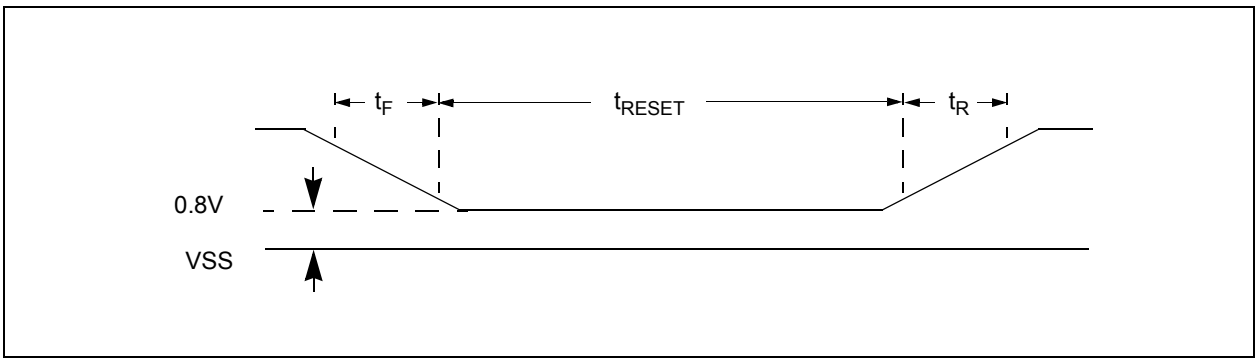


TABLE 7-27: RESETI# TIMING PARAMETERS

Symbol	Parameter	Limits		Units	Comments
		MIN	MAX		
t_F	RESETI# Fall time	0	10	µs	
t_R	RESETI# Rise time	0	10	µs	
t_{RESET}	Minimum Reset Time	1		µs	Note 7-40

Note 7-40 The RESETI# input can tolerate glitches of no more than 50ns.

7.6.3.2 RESETO#

The RESETO# output pin GPIO062/RESETO# defaults to an output 'low' following a VTR power up, or assertion of the RESETI# input pin. Firmware de-asserts RESETO# by making the GPIO062 output 'high'.

7.6.4 VTRGD

VTRGD is the reset signal for the 20 MHz Oscillator and the source for nSYS_RST and nEC_RST.

As shown in Figure 7-18, Figure 7-20 and in Table 7-28, VTRGD is asserted following a delay after the VTR and VTR_1.8 power supplies exceed preset voltage thresholds as defined in Table 7-25, "VTR/VBAT Reset Thresholds," on page 124. VTRGD is de-asserted as soon as either the VTR or VTR_1.8 power supplies drop below these thresholds (see Figure 7-21, "VTR Power-Down Timing").

VTRGD can also be asserted as a result of a Watch-Dog Timer Forced Reset as described in Section 7.7.4, "Watch-Dog Timer Forced Reset," on page 130, and by the RESETI# pin in the RESET Pin Interface.

© 2012-2018 Microchip Technology Inc.

FIGURE 7-20: VTR POWER-UP TIMING

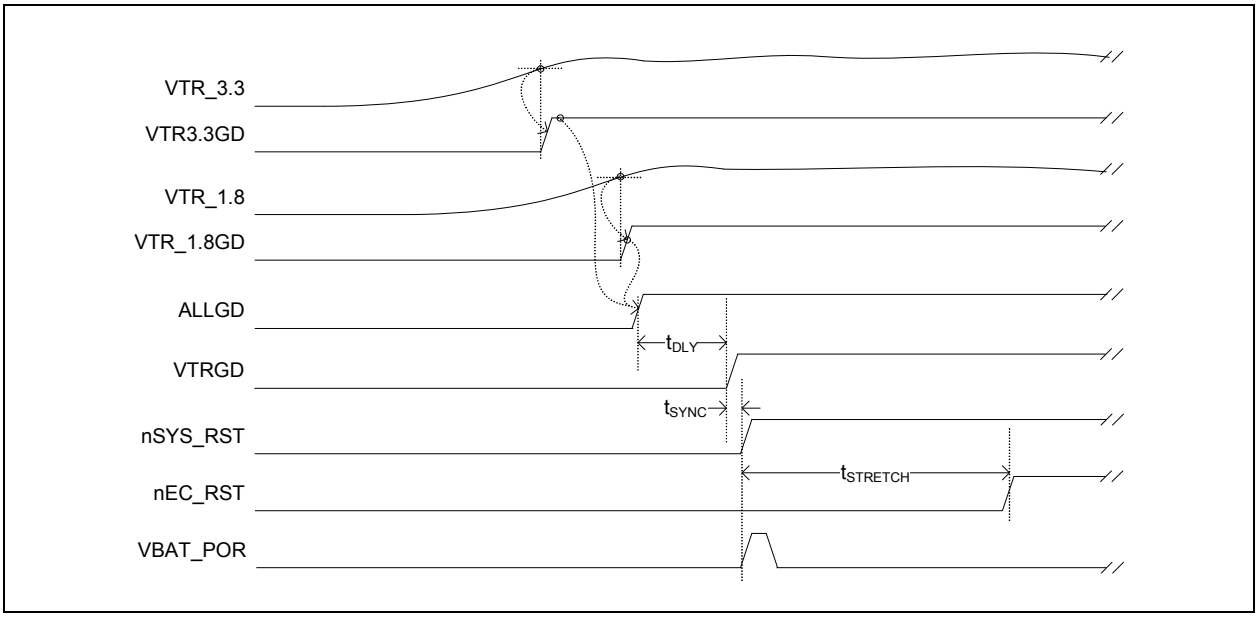
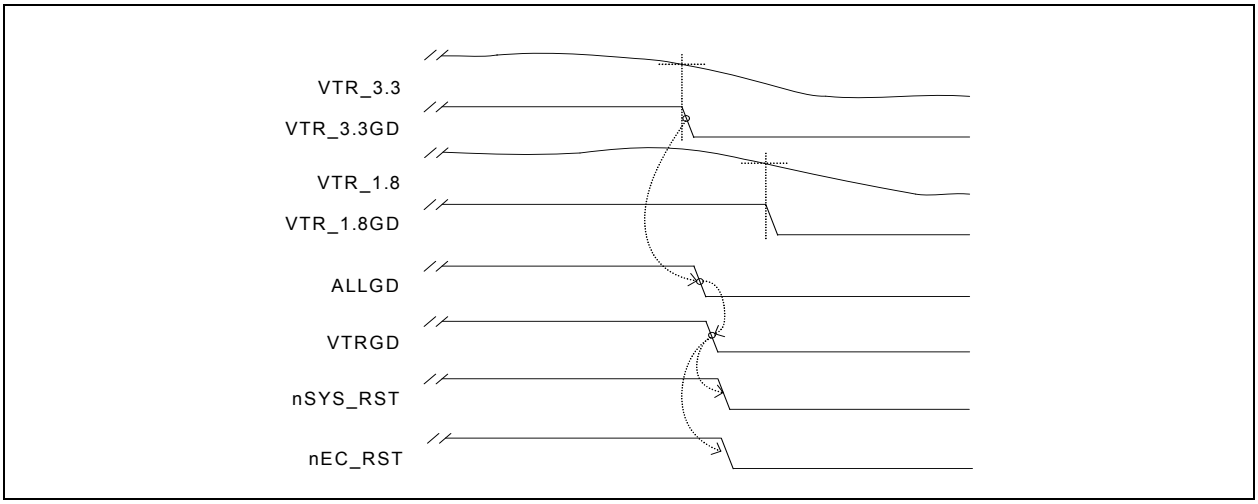


TABLE 7-28: VTR POWER-UP TIMING

Parameter	Symbol	MIN	TYP	MAX	Units	Notes
VTRGD Delay Time	t_{DLY}	–	600	800	µs	
nSYS_RST Delay Time	t_{SYNC}	2	–	3	20 MHz Oscillator Clocks	Note 7-41
nEC_RST Delay Time	$t_{STRETCH}$	–	–	5	ms	

Note 7-41 this interval is determined using a Fixed Clock Domain from the 20 MHz Oscillator.

FIGURE 7-21: VTR POWER-DOWN TIMING



7.6.5 VBAT_POR

VBAT_POR occurs within the Reset Interface whenever the coin cell is replaced, or the VBAT voltage falls below 1.25 V nominal (Table 7-25).

There is also a VBAT_POR pulse external to the Reset Interface (Figure 7-20) that is used to reset VBAT powered status bits in VTR powered registers. This pulse is asserted at the rising edge of nSYS_RST if a VBAT_POR was detected within the Reset Interface when VTRGD was not asserted.

Note that both the 32.768 KHz Silicon Oscillator and the 32.768 KHz Crystal Oscillator are stopped if the coin cell is replaced, or the VBAT voltage falls below 1.25 V nominal while VTRGD is not asserted. No action is taken if the coin cell is replaced, or the VBAT voltage falls below 1.25 V nominal while VTRGD is asserted.

VBAT_POR is used as described throughout this specification to reset registers and functional device blocks. VBAT_POR events are registered in the Power-Fail and Reset Status Register.

7.6.6 NSYS_RST

nSYS_RST is a combination of the following reset sources as described below: VTRGD, Watch-Dog Timer Forced Reset, the RESETI# pin.

nSYS_RST is VTRGD synchronized to the 20 MHz Oscillator. Note that VTRGD and nSYS_RST have the same logical sense (uninverted); however, because of nomenclature, the asserted states are opposite. Note that VTRGD is defined in Section 7.6.4, "VTRGD," on page 126.

nSYS_RST is de-asserted as defined in Figure 7-20, "VTR Power-Up Timing" and in Table 7-28, "VTR Power-Up Timing". nSYS_RST can also be asserted as a result of a Watch-Dog Timer Forced Reset as described in Section 7.7.4, "Watch-Dog Timer Forced Reset," on page 130, or when the RESETI# pin is asserted as described in Section 7.6.3, "RESET Pin Interface," on page 126. nSYS_RST is asserted as soon as either the VTR or VTR_1.8 power supplies drop below preset voltage thresholds (see FIGURE 7-21: VTR Power-Down Timing on page 127).

nSYS_RST is the reset signal for all VTR-powered blocks except for the 20 MHz Oscillator and the Embedded Controller. nSYS_RST also affects the VBAT-Powered Control Interface as described in Table 34-2, “VCI Output Truth Table,” on page 501.

7.6.7 NEC_RST

nEC_RST is a delayed version of nSYS_RST that is used to reset the Embedded Controller and for Registers Interface as described in Section 7.8, "Registers Interface," on page 130.

nEC_RST can be asserted as a result of a Watch-Dog Timer Forced Reset as described in Section 7.7.4, "Watch-Dog Timer Forced Reset," on page 130, or when the RESETI# pin is asserted as described in Section 7.6.3, "RESET Pin Interface," on page 126. Like nSYS_RST, nEC_RST is asserted as soon as either the VTR or VTR_1.8 power supplies drop below preset voltage thresholds (see FIGURE 7-21: VTR Power-Down Timing on page 127).

nEC_RST is de-asserted as defined in Figure 7-20, "VTR Power-Up Timing" and in Table 7-28, "VTR Power-Up Timing". nEC_RST is held asserted while the Gang Programmer Interfaceis activated.

7.7 1.8V Regulator

The 1.8V Regulator generates the MEC1632 core power well. As illustrated in Figure 7-18, "Reset Interface Block Diagram", the input to the 1.8V Regulator is VTR, the output is VTR_1.8 (see also Table 7-4). The 1.8V Regulator is not used when VTR is inactive, as described in Section 7.7.1, "Power Mux" below.

The stability of the 1.8V Regulator amplifier depends on an external capacitor, VR_CAP as described in Table 7-4. The choice of capacitor can be either ceramic or low ESR tantalum. Ceramics are the recommended choice due to their superior AC performance (below 100 mΩ ESR), but X5R dielectrics should be used to prevent greater than 20% capacitance variation over temperature and voltage. Low ESR tantalum capacitors will work but care should be taken because the ESR can vary 2x at low temperatures.

APPLICATION NOTE: the 1.8V Regulator can be suspended in the SYSTEM DEEPEST SLEEP state using the VREG SUS bit in the VREG Control Register on page 147.

© 2012-2018 Microchip Technology Inc.

7.7.1 POWER MUX

To provide the highest reliability and lowest possible power consumption, the Power Mux switches between the 1.8V Regulator and a level-shifted VBAT voltage to produce the 1.8V internal supply for VBAT-backed logic (VTR1.8_BAT in Table 7-4).

Power Mux switching depends on the voltage level of the 1.8V Regulator and the VTR supply. As illustrated in Figure 7-18, the Power Mux selects the 1.8V Regulator after the VTR and the VTR_1.8 power supplies exceed preset voltage thresholds. The Power Mux selects the VBAT supply as soon as either the VTR or VTR_1.8 power supplies drop below these thresholds (see FIGURE 7-21: VTR Power-Down Timing on page 127).

Note that the Power Mux only switches 1.8 volts. To provide minimum VBAT power consumption for 3.3V VBAT powered outputs when VTR is fully powered, supply switching from VBAT to VTR must be done externally.

There is a separate power mux not shown in Figure 7-18 to ensure that the 32.768 KHz Silicon Oscillator is powered when VTR is powered for TYPE 1 Power Supply Configurations.

7.7.2 VCC POWER GOOD

VCC Power Good is defined by the VCC_PWRGD input pin (Table 7-4). VCC_PWRGD is also synchronized to the 20 MHz Oscillator and used for the functions shown in Table 7-29.

The VCC_PWRGD input must always be driven to a '1' or a '0,' even when VCC is 0 V. The minimum VCC_PWRGD pulse width (high and low) is shown below in Table 7-29.

TABLE 7-29: VCC_PWRGD INPUT TIMING

Parameter	Symbol	MIN	TYP	MAX	Units	Notes
VCC_PWRGD Pulse Width	t_{VPGPW}	31	–	–	ns	

TABLE 7-30: FUNCTIONS AFFECTED BY VCC Power Good

Name	Reference
Host Clock Domain	see Section 7.4.11.4, "Host Clock Domain," on page 120
LPC RESET	Section 7.7.3, "LPC RESET," on page 129
nSIO_RESET	"iRESET OUT," on page 133.
VCC_PWRGD_BUFF	Table 7-4, "Power, Clocks, and Resets Port List," on page 96.
VCC_PWRGD	"VCC PWRGD," on page 133

7.7.3 LPC RESET

LPC RESET (LPC_RST# in Table 7-4) is defined by VCC_PWRGD, LRESET# and VTRGD as illustrated in Table 7-31. LPC RESET only affects logic that is driven by PCI_CLK.

TABLE 7-31: LPC RESET DEFINITION

VCC_PWRGD (Note 7-42)	LRESET# (Note 7-43)	VTRGD (Note 7-44)	LPC RESET (Note 7-45)
0	X	X	0
1		0	Undefined
	0	1	0
	1		1

Note 7-42 this is the Table 7-4 VCC_PWRGD input.

Note 7-43 this is the Table 7-4 LRESET# input. The EC can determine the state of the LRESET# input using registers in Table 6-8, "LPC Bus Monitor Register," on page 90.

Note 7-44 see Section 7.6.4, "VTRGD," on page 126.

Note 7-45 LPC RESET is the Table 7-4 LPC_RST# output. The trailing edge of LPC_RST# is synchronized to the PCI_CLK in Table 7-4.

© 2012-2018 Microchip Technology Inc.

7.7.4 WATCH-DOG TIMER FORCED RESET

A Watch-Dog Timer Forced Reset (Figure 7-22) occurs when the WDT_ALRT input (Table 7-4) is asserted ('1'). As shown in Figure 7-22, VTRGD is de-asserted ('0') and nSYS_RST and nEC_RST are asserted ('0') when WDT_ALRT is asserted. The VTRGD reset time (t_{RST}) is determined by the 32.768 KHz Silicon Oscillator as shown in Table 7-32. Following the VTRGD reset time, the nSYS_RST Delay Time (t_{SYNC}) and the nEC_RST Delay Time ($t_{STRETCH}$) are determined using the 20 MHz Oscillator as described in Figure 7-20, "VTR Power-Up Timing" and Table 7-28, "VTR Power-Up Timing".

Note that analog reset signal functions are not shown in Figure 7-22 because it is assumed that the power supplies are fully powered and stable during a Watch-Dog Timer Forced Reset.

FIGURE 7-22: Watch-Dog Timer Forced Reset TIMING

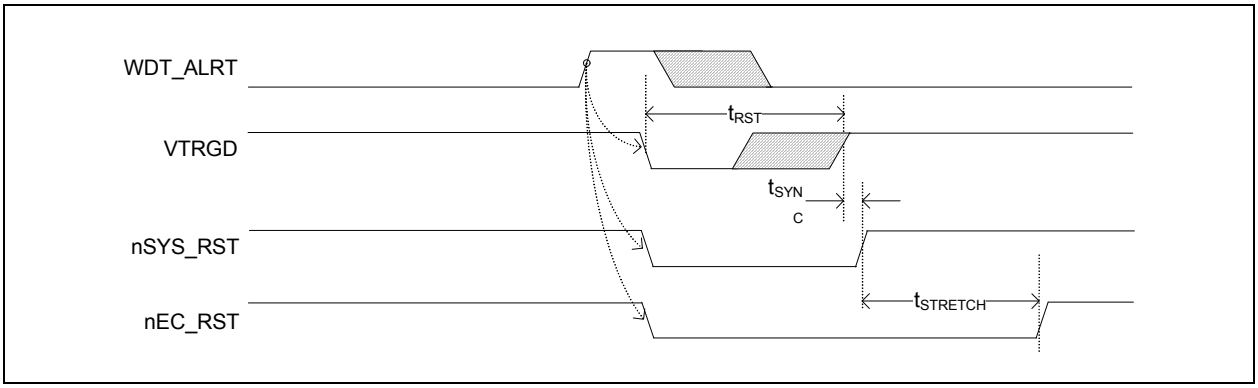


TABLE 7-32: Watch-Dog Timer Forced Reset TIMING

Parameter	Symbol	MIN	TYP	MAX	Units
VTRGD Reset Time	t_{RST}	1	–	2	32.768 KHz Silicon Oscillator Clock Cycles
nSYS_RST Delay Time	t_{SYNC}	(see FIGURE 7-20: VTR Power-Up Timing on page 127)			
nEC_RST Delay Time	$t_{STRETCH}$				

7.8 Registers Interface

The Power, Clocks, and Resets registers are located in two address ranges with two Base Address as indicated in Table 7-33. See Note 4-1 on page 60.

TABLE 7-33: POWER, CLOCKS AND RESETS INTERFACE BASE ADDRESS TABLE

Power, Clocks and Reset Blocks	LDN from (Table 4-2 on page 59)	AHB Base Address
Power, Clock & Reset (VTR PWR'ed)	32h	F0_C800h
Power, Clock & Reset (VBAT PWR'ed)	33h	F0_CC00h

Table 7-34 is a register summary for the Power, Clock & Reset (VTR PWR'ed) registers. Table 7-35 is a register summary for the Power, Clock & Reset (VBAT PWR'ed) registers

Each EC address is indicated as an SPB Offset from its AHB base address as indicated in Table 7-33.

The following tables summarize the registers allocated for each Instance. The offset field in the following table is the offset from the Embedded Controller (EC) Base Address.

© 2012-2018 Microchip Technology Inc.

TABLE 7-34: Power, Clocks, and Resets VTR-POWERED REGISTERS SUMMARY

Offset (HEX) (Note 7-46)	Register Name	Access	Page Reference
0	EC Clock Divider Register	R/W	132
4	MCHP Reserved	R/W	-
8	PCR Status and Control Register	R	133
C	Clock Control Register	R/W	134
10	LPC Blocks Sleep Enables Register	R/W	136
14	EC Blocks Sleep Enables Register 1	R/W	137
18	LPC Blocks Clock Required Status Register	R	141
1C	EC Blocks Clock Required Status Register 1	R	137
20	Reserved	R	–
24	MCHP Reserved	R/W	–
28	Reserved	R	–
2C	Reserved	R	–
30	EC Blocks Sleep Enables Register 2	R/W	138
34	EC Blocks Clock Required Status Register 2	R	143
38	Reserved	R	–
3C	Reserved	R	–
40	Clock Tree Control 0 Register	R/W	144
41	Clock Tree Control 2 Register	R/W	145
44	Clock Tree Control 1 Register	R/W	146
48	VREG Control Register	R/W	147
4C	Reserved	R	–
50	Reserved	R	–
54	MCHP Reserved	R/W	–

Note 7-46 all register addresses are naturally aligned on 32-bit boundaries. Offsets for registers that are smaller than 32 bits are reserved and must not be used for any other purpose.

TABLE 7-35: Power, Clocks, and Resets VBAT-POWERED REGISTERS SUMMARY

Offset (HEX) (Note 7-46)	Register Name	Access	Page Reference
0	Power-Fail and Reset Status Register	R/W	148
4	Clock Enable Register	R/W	149
8	MCHP Reserved	R/W	–
C	MCHP Reserved	R/W	–

7.8.1 EC CLOCK DIVIDER REGISTER

7.8.1.1 Overview

The Block Sleep Enable Registers (Table 7-36) contains the EC_CLK_DIV bits that are used to program the EC clock and the EC_AHB clock enable frequency as described in Table 7-37.

TABLE 7-36: EC Clock Divider Register

HOST ADDRESS	N/A				N/A		**HOST SIZE**	
EC OFFSET	00h				32-bit		**EC SIZE**	
POWER	VTR				01h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	**---**		**D10**	**D9**	**D8**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R	R	R	R	R/W	R/W	R/W	R/W
BIT NAME	Reserved				EC_CLK_DIV			

EC_CLK_DIV

The EC_CLK_DIV bits contain the binary encoded divider that determines the EC clock and EC_AHB clock enable frequency (Table 7-37). Writing '0' to the Block Sleep Enable Registers has no effect. The EC_CLK_DIV default is 01h.

TABLE 7-37: EC_CLK_DIV ENCODING

EC_CLK_DIV	**Nominal Frequency (MHz)**
0h	No Change
1h	20.27 (default)
2h	10.14
3h - Fh	Reserved

APPLICATION NOTE: To support EC traffic to the LPC Subsystem, the EC_AHB clock frequency must be equal to or less than the LPC_AHB clock frequency.

APPLICATION NOTE: the JTAG clock cannot be higher than 1/2 the EC clock as defined by the EC Clock Divider Register.

© 2012-2018 Microchip Technology Inc.

7.8.2 PCR STATUS AND CONTROL REGISTER

TABLE 7-38: PCR Status and Control Register

HOST ADDRESS	N/A			N/A			**HOST SIZE**	
EC OFFSET	08h			32-bit			**EC SIZE**	
POWER	VTR			N/A			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	**---**		**D10**	**D9**	**D8**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R	R	R	R/W	R	R	R	R
BIT NAME	Reserved		FREQ LOCK	iRESET OUT	Reserved	Reserved	VCC PWRGD	Reserved

VCC PWRGD

The VCC PWRGD bit reflects the state of the synchronized VCC_PWRGD input pin (see Section 7.7.2, "VCC Power Good," on page 129). The VCC PWRGD pin can generate an either-edge interrupt as described in the note associated with the VCC PWRGD signal in Section 24.6, "GPIO Indexing," on page 391.

iRESET OUT

The iRESET OUT bit is used by firmware to control the nSIO_RESET signal function (Table 7-4). Firmware can program the state of iRESET OUT except when the VCC PWRGD bit is not asserted ('0'), in which case iRESET OUT is 'don't care' and nSIO_RESET is asserted ('0') (Table 7-39). In all other cases, the nSIO_RESET signal function is always the inverse of the iRESET OUT bit.

APPLICATION NOTE: it should be noted that when the iRESET OUT bit is asserted ('1') the internal nSIO_RESET is asserted even if the nRESET_OUT pin is configured as an alternate function.

Note 7-47 When the iRESET OUT bit is set to '1', the falling edge of VCC PWRGD will cause the nRESET_OUT signal function to be asserted within 15ns, A subsequent rising edge of VCC PWRGD will cause the nRESET_OUT signal function to de-assert within 3 MCLKs.

Note 7-48 nSIO_RESET is also the source for the nRESET_OUT signal function.

TABLE 7-39: iRESET OUT **BIT BEHAVIOR**

VCC PWRGD	iRESET OUT	nSIO_RESET **& nRESET_OUT (See Note 7-48)**	**Description**
0	X	0 (ASSERTED)	The iRESET OUT bit does not affect the state of nSIO_RESET when VCC PWRGD is not asserted.
1	1	0 (ASSERTED)	The iRESET OUT bit can only be written by firmware when VCC PWRGD is asserted.
	0	1 (NOT ASSERTED)	

FREQ LOCK

FREQ LOCK is asserted ('1') when the accuracy of the 20 MHz Oscillator is within the tightest tolerance described in Table 7-6, "20 MHz Oscillator Timing Parameters," on page 101.

7.8.3 CLOCK CONTROL REGISTER

TABLE 7-40: Clock Control Register

HOST ADDRESS	N/A			N/A			**HOST SIZE**	
EC OFFSET	0Ch			32-bit			**EC SIZE**	
POWER	VTR			04h			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	**---**		**D10**	**D9**	**D8**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
BIT NAME	SLEEP CONFIG			WAIT FOR LOCK	32KHz OUTPUT	MCHP Reserved	SLEEP_FLAG	Reserved

SLEEP_FLAG

The SLEEP_FLAG affects the system power state as described in Section 7.4.10.2, "EC Controlled Dynamic Power States," on page 112. The SLEEP_FLAG is R/W. EC firmware asserts SLEEP_FLAG (‘1’), which is then typically de-asserted (‘0’) by hardware as described in Section 7.4.10.6, "Wake Interface," on page 118.

APPLICATION NOTE: Asserting the SLEEP_FLAG allows the device to be put into lower power states as described in Section 7.4.10.2, "EC Controlled Dynamic Power States," on page 112. If the main oscillator is shut down as a result of asserting the SLEEP_FLAG, then only the enabled wake capable interrupts can restart the oscillator and bring the part out of its sleep state. When the main oscillator is shut down, non-wake capable interrupts cannot occur. As part of the wake process, the SLEEP_FLAG is cleared by hardware, re-enabling all blocks by de-asserting all the block SLEEP_ENABLE signals as described in Section 7.4.10.5.4, "Block Sleep Enables".

The typical sequence to go to sleep is this:

```
disable_interrupts();
assert_all_sleep_enables();
CLOCK_CONTROL_REGISTER->SLEEP_FLAG = 1;
_SLEEP();
```

where the macro `_SLEEP()` both enables interrupts and issues the ARC sleep instruction.

However, if any interrupt, including non-Wake interrupts, occurs just before the SLEEP_FLAG is asserted, it will remain pending and will fire as soon as the ARC issues the sleep instruction. The SLEEP_FLAG will remain asserted, and therefore all blocks are kept in the low power state by their respective SLEEP_ENABLE signals. As a result, an interrupt service routine may not proceed properly because blocks that it must manipulate remain in their idle state.

If a system uses the sleep interface then every Interrupt Service Routine must manually clear the SLEEP_FLAG. This ensures that all blocks that are supposed to be active when the ISR is in process will be active.

© 2012-2018 Microchip Technology Inc.

32KHZ OUTPUT

The 32KHz OUTPUT bit controls the MEC1632 32KHZ_OUT signal function (Table 7-4).

0: The 32KHZ_OUTPUT clock output signal is enabled (default)

1: The 32KHZ_OUTPUT clock output signal is disabled. The 32KHZ_OUTPUT signal is driven low.

WAIT FOR LOCK

This bit controls the behavior of the system after a wake event occurs.

0: The system is clocked by the 10 MHz Ring Oscillator when a wake event occurs and switches to the 20 MHz Oscillatorwhen the FREQ LOCK bit is asserted. (default)

1: System clocking remains gated until the FREQ LOCK bit is asserted

SLEEP CONFIG

The SLEEP CONFIG field determines the latency following a wake event until the 20 MHz Oscillator is locked (FREQ LOCK) and clocking the system. This latency is determined as defined in Table 7-41.

TABLE 7-41: SLEEP CONFIG BIT ENCODING

SLEEP CONFIG			Wake Latency	Description (Note 7-49)
D7	D6	D5		
0	0	0	0 ns	Master Clock Trees, the 1.8V Regulator, the 20 MHz Oscillator, and the regulator for the 20 MHz Oscillator all remain powered and running during sleep cycles (SYSTEM HEAVY SLEEP 1) (DEFAULT)
0	0	1	200 ns	Master Clock Trees gated and the 1.8V Regulator is suspended during sleep cycles. (SYSTEM HEAVY SLEEP 2)
0	1	0	460 µs	Master Clock Trees gated, the 1.8V Regulator is suspended, and the 20 MHz Oscillator is stopped during sleep cycles. (SYSTEM HEAVY SLEEP 3)
0	1	1	4.6 ms	Master Clock Trees gated, the 1.8V Regulator is suspended, the 20 MHz Oscillator is stopped, and the regulator for the 20 MHz Oscillator is powered down during sleep cycles. (SYSTEM DEEPEST SLEEP)
1	X	X	–	Reserved

Note 7-49 Suspending the 1.8V Regulator also depends on the state of the VREG SUS bit in the VREG Control Register.

7.8.4 BLOCK SLEEP ENABLE REGISTERS

The Block Sleep Enables identified in the Block Sleep Enable Registers are described below in Section 7.8.4.1, "LPC Blocks Sleep Enables Register," on page 136 and Section 7.8.4.2, "EC Blocks Sleep Enables Registers," on page 137. The behavior of the Block Sleep Enables is described in "Block Sleep Enables," on page 117.

7.8.4.1 LPC Blocks Sleep Enables Register

TABLE 7-42: LPC Blocks Sleep Enables Register

HOST ADDRESS	N/A			N/A			**HOST SIZE**	
EC OFFSET	10h			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**. . .**			**D2**	**D9**	**D8**
HOST TYPE	–	–	**. . .**			–	–	–
EC TYPE	R							
BIT NAME	Reserved							
BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	RTC		EMI	FLASH SPI	MCHP Reserved	UART	LPC	Reserved

TABLE 7-43: LPC BLOCKS SLEEP ENABLES/CLOCK REQUIRED REGISTERS BIT NAMES

Bit Name	Block Name	Block Cross Reference
UART	16C550A UART	Section 13.8, "Sleep Enable/ Clock Request Power State Controls," on page 239
FLASH SPI	LPC GP-SPI	Section 33.0, "General Purpose Serial Peripheral Interface (GP-SPI)," on page 475
EMI	Embedded Memory Interface	Section 8.0, "Embedded Memory Interface," on page 151
RTC	Real-Time Clock	Section 23.0, "RTC With Date and DST Adjustment," on page 375
MCHP Reserved	–	MCHP Reserved Sleep bits must be set to '1' to put the device into Heavy and Deep Sleep States. See Table 7-14, "EC Controlled Dynamic Power States," on page 112

Some LPC accessible blocks have no Block Sleep Enable bit in the LPC Blocks Sleep Enables Register. Table 7-44 describes these.

TABLE 7-44: LPC BLOCKS NOT CONTROLLED BY LPC Blocks Sleep Enables Register

Block Name	Block Cross Reference
LPC Interface	Section 6.10.4, "EC Clock Control Register," on page 92
Legacy Port Functions	Legacy Support on page 197
ACPI EC Interface	Section 9.0, "ACPI Embedded Controller Interface (ACPI-ECI)," on page 170
8042 Emulated Keyboard Controller Interface	Section 10.0, "8042 Emulated Keyboard Controller," on page 190
ACPI PM1 Interface	Section 11.0, "ACPI PM1 Block Interface," on page 205
Mailbox Registers Interface	Section 12.0, "MailBox Register Interface," on page 213

© 2012-2018 Microchip Technology Inc.

7.8.4.2 EC Blocks Sleep Enables Registers

TABLE 7-45: EC BLOCKS SLEEP ENABLES REGISTER 1

HOST ADDRESS	N/A			N/A			**HOST SIZE**	
EC OFFSET	14h			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D28**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R	R	R	R	R/W	R/W	R/W	R/W
BIT NAME	Reserved				EEPROM	WDT	HIB1	HIB0
BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	FLASH	SPI_P	SMB1	SMB0	PWM3	PWM2	PWM1	PWM0
BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R/W	R/W	R/W	R	R	R	R	R/W
BIT NAME	PS2_2	PS2_1	PS2_0	Reserved				TACH2
BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
BIT NAME	TACH1	TACH0	TFDP	RC_ID	C/T3	C/T2	C/T1	C/T0

© 2012-2018 Microchip Technology Inc.

TABLE 7-46: EC BLOCKS SLEEP ENABLES REGISTER 2

HOST ADDRESS	N/A				N/A		**HOST SIZE**	
EC OFFSET	30h				32-bit		**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
	EC SPB							
BIT	**D31**	**D30**	**D28**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R	R	R	R	R/W	R/W	R/W	R/W
BIT NAME	Reserved				SMB3	Rsrvd	TACH5	TACH4
BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	PWM15	PWM14	PWM13	PWM12	PWM11	PWM10	PWM9	PWM8
BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	LED2	LED1	LED0	CEC	Rsrvd	KSC	PECI	ADC
BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	DMA	CCT	SMB2	TACH3	PWM7	PWM6	PWM5	PWM4

TABLE 7-47: EC BLOCKS SLEEP ENABLES/CLOCK REQUIRED REGISTERS BIT NAMES

Bit Name	Block Name (Note 7-50)	Block Cross Reference
C/T0	16-Bit Counter/Timer 0	Section 20.0, "16-Bit Timer Interface," on page 351
C/T1	16-Bit Counter/Timer 1	
C/T2	16-Bit Counter/Timer 2	
C/T3	16-Bit Counter/Timer 3	
RC_ID	RC ID Interface	Section 32.0, "RC Identification Detection (RC_ID)," on page 466
TFDP	TFDP Interface	Section 41.0, "Trace FIFO Debug Port (TFDP)," on page 562
TACH0	Tachometer 0	Section 30.0, "TACH Monitor," on page 451
TACH1	Tachometer 1	
TACH2	Tachometer 2	
TACH3	Tachometer 3	
TACH4	Tachometer 4	
TACH5	Tachometer 5	

© 2012-2018 Microchip Technology Inc.

TABLE 7-47: EC BLOCKS SLEEP ENABLES/CLOCK REQUIRED REGISTERS BIT NAMES

Bit Name	Block Name (Note 7-50)	Block Cross Reference
PS2_0	PS/2 Interface 0	Section 37.0, "PS/2 Device Interface," on page 530
PS2_1	PS/2 Interface 1	
PS2_2	PS/2 Interface 2	
PWM0	PWM 0	Section 31.0, "PWM Controller," on page 460l
PWM1	PWM 1	
PWM2	PWM 2	
PWM3	PWM 3	
PWM4	PWM 4	
PWM5	PWM 5	
PWM6	PWM 6	
PWM7	PWM 7	
PWM8	PWM8	
PWM9	PWM9	
PWM10	PWM10	
PWM11	PWM11	
PWM12	PWM12	
PWM13	PWM13	
PWM14	PWM14	
PWM15	PWM15	
SMB0	SMBus 0	Section 27.0, "SMB Device Interface," on page 432
SMB1	SMBus 1	
SMB2	SMBus 2	
SMB3	SMBus 3	
SPI_P	SPI Peripheral Interface	Section 33.0, "General Purpose Serial Peripheral Interface (GP-SPI)," on page 475
CCT	Capture Compare Timer	Section 25.0, "Input Capture and Compare Timer," on page 406
DMA	DMA	Section 26.0, "DMA Controller," on page 419
ADC	ADC	Section 29.0, "Analog to Digital Converter," on page 439
PECI	PECI	Section 28.0, "PECI Interface," on page 437
KSC	Key scan	Section 38.0, "Keyboard Matrix Scan Support," on page 539
HIB0	Hibernation Timer 0	Section 21.0, "Hibernation Timer," on page 366
HIB1	Hibernation Timer 1	
WDT	Watchdog Timer	Section 18.0, "Watchdog Timer Interface," on page 330
LED0	Blinking/Breathing LED 0	Section 36.0, "Blinking/Breathing PWM," on page 510
LED1	Blinking/Breathing LED 1	
LED2	Blinking/Breathing LED 2	
CEC	HDMI CEC Interface	Section 19.0, "HDMI-CEC Interface Controller," on page 337
EEPROM	2KB EEPROM	Section 15.0, "EEPROM," on page 266

TABLE 7-47: EC BLOCKS SLEEP ENABLES/CLOCK REQUIRED REGISTERS BIT NAMES

Bit Name	Block Name (Note 7-50)	Block Cross Reference
MCHP Reserved	–	– **Note:** MCHP Reserved Sleep Enable bits must be set to '1' to put the device into Heavy and Deep Sleep States. See Table 7-14, "EC Controlled Dynamic Power States," on page 112
Reserved	–	–

Note 7-50 Some EC accessible blocks have no Block Sleep Enable bit in the EC Blocks Sleep Enables Register 1. Table 7-48 describes these.

TABLE 7-48: EC BLOCKS NOT CONTROLLED BY EC Blocks Sleep Enables Registers

Block Name	Block Cross Reference
MCU Serial Debug Port	Section 41.0, "Trace FIFO Debug Port (TFDP)," on page 562
Master BC Link B	Section 39.0, "BC-Link Master," on page 547
Master BC Link A	
Master BC Link D	
Flash Interface	Section 14.0, "Embedded Flash Subsystem," on page 240
EC	Section 16.0, "ARC 625D Embedded Controller," on page 280
Interrupt Aggregater	Section 17.0, "EC Interrupt Aggregator," on page 287 (see also Section 7.4.9.2.2, "INTERRUPT AGGREGATOR CLOCK TREE," on page 109)
Week Alarm Timer	Section 22.0, "Week Alarm Interface," on page 370
GPIO	Section 24.0, "GPIO Interface," on page 388 (see also Section 7.4.9.2.3, "GPIO CLOCK TREE," on page 109)
VCI	Section 34.0, "VBAT-Powered Control Interface," on page 499
VBAT_RAM	Section 35.0, "VBAT Powered RAM," on page 508

7.8.5 CLOCK REQUIRED STATUS REGISTERS

The Clock Required Status Registers indicates the core clock status per block as defined in Section 7.4.8, "Generic Block Clocking Model," on page 105. Like the Block Sleep Enable Registers, there are two types of Clock Required Status Registers: the LPC Blocks Clock Required Status Register and the EC Blocks Clock Required Status Registers.

When a bit in the Clock Required Status Registers is asserted ('1'), the block is enabled and requires that the 20 MHz Oscillator remain running as defined in "EC Power State Controls," on page 115.

When a bit in the Clock Required Status Registers is not asserted ('0'), the block is either not enabled as defined in the Generic Block Clocking Model, or has been commanded to sleep and no longer requires the 20 MHz Oscillator.

© 2012-2018 Microchip Technology Inc.

7.8.5.1 LPC Blocks Clock Required Status Register

TABLE 7-49: LPC Blocks Clock Required Status Register

HOST ADDRESS	N/A				N/A		**HOST SIZE**	
EC OFFSET	18h				32-bit		**EC SIZE**	
POWER	VTR				N/A		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D28**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved						BDP1	BDP0
BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	RTC	MCHP Reserved	EMI	FLASH SPI	MCHP Reserved	UART	LPC	Reserved

7.8.5.2 EC Blocks Clock Required Status Registers

TABLE 7-50: EC BLOCKS CLOCK REQUIRED STATUS REGISTER 1

HOST ADDRESS	N/A			N/A			**HOST SIZE**	
EC OFFSET	1Ch			32-bit			**EC SIZE**	
POWER	VTR			N/A			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D28**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R	R	R	R	R/W	R/W	R/W	R/W
BIT NAME	Reserved				EEPROM	WDT	HIB1	HIB0
BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	FLASH Note 7-51	SPI_P	SMB1	SMB0	PWM3	PWM2	PWM1	PWM0
BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R/W	R/W	R/W	R	R	R	R	R/W
BIT NAME	PS2_2	PS2_1	PS2_0	Reserved				TACH2
BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
BIT NAME	TACH1	TACH0	TFDP	RC_ID	C/T3	C/T2	C/T1	C/T0

Note 7-51 The Flash Clock Required status bit FLASH is asserted whenever the ARC or a JTAG Debug master attempts to access the EC Blocks Clock Required Status Registers.

© 2012-2018 Microchip Technology Inc.

TABLE 7-51: EC BLOCKS CLOCK REQUIRED STATUS REGISTER 2

HOST ADDRESS	N/A				N/A		**HOST SIZE**	
EC OFFSET	34h				32-bit		**EC SIZE**	
POWER	VTR				N/A		nSYS_RST **DEFAULT**	
	EC SPB							
BIT	**D31**	**D30**	**D28**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R	R	R	R	R/W	R/W	R/W	R/W
BIT NAME	Reserved				SMB3	Rsrvd	TACH5	TACH4
BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	PWM15	PWM14	PWM13	PWM12	PWM11	PWM10	PWM9	PWM8
BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	LED2	LED1	LED0	CEC	Rsrvd	KSC	PECI	ADC
BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	DMA	CCT	SMB2	TACH3	PWM7	PWM6	PWM5	PWM4

TABLE 7-52: BLOCKS NOT CONTROLLED BY SLEEP ENABLE BITS WITH CLOCK REQUIRED STATUS OUTPUTS

Bit Name	Block Name	Block Cross Reference
LEGACY	Legacy Port Functions	Legacy Support on page 197
UART	16C550A UART	Section 13.8, "Sleep Enable/ Clock Request Power State Controls," on page 239
LPC	LPC Interface	Section 6.10.4, "EC Clock Control Register," on page 92
FLASH	Flash Interface	Section 14.0, "Embedded Flash Subsystem," on page 240
MBCLB	Master BC Link B	Section 39.0, "BC-Link Master," on page 547
MBCLA	Master BC Link A	
MBCLD	Master BC Link D	
BDP0	BIOS Debug Port 0	Section 40.0, "Port 80 BIOS Debug Port," on page 555
BDP1	BIOS Debug Port 1	

7.8.6 CLOCK TREE CONTROL 0 REGISTER

TABLE 7-53: Clock Tree Control 0 Register

HOST ADDRESS	N/A			N/A			**HOST SIZE**	
EC OFFSET	40h			32-bit			**EC SIZE**	
POWER	VTR			00h			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	**---**		**D10**	**D9**	**D8**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	FECT3 OFF	FECT3 ON	FECT2 OFF	FECT2 ON	FECT1 OFF	FECT1 ON	FECT0 OFF	FECT0 ON

FECT0 ON

Force EC Clock Tree 0 on.

FECT0 OFF

Force EC Clock Tree 0 off.

FECT1 ON

Force EC Clock Tree 1 on.

FECT1 OFF

Force EC Clock Tree 1 off.

FECT2 ON

Force EC Clock Tree 2 on.

FECT2 OFF

Force EC Clock Tree 2 off.

FECT3 ON

Force EC Clock Tree 3 on.

FECT3 OFF

Force EC Clock Tree 3 off.

© 2012-2018 Microchip Technology Inc.

7.8.7 CLOCK TREE CONTROL 2 REGISTER

TABLE 7-54: Clock Tree Control 2 Register

HOST ADDRESS	N/A				N/A		**HOST SIZE**	
EC OFFSET	41h				32-bit		**EC SIZE**	
POWER	VTR				00h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	**---**		**D10**	**D9**	**D8**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R	R	R	R	R/W	R/W	R/W	R/W
BIT NAME	RES	RES	RES	RES	FECT5 OFF	FECT5 ON	FECT4 OFF	FECT4 ON

FECT4 ON

Force EC Clock Tree 4 on.

FECT4 OFF

Force EC Clock Tree 4 off.

FECT5 ON

Force EC Clock Tree 5 on.

FECT5 OFF

Force EC Clock Tree 5 off.

7.8.8 CLOCK TREE CONTROL 1 REGISTER

TABLE 7-55: Clock Tree Control 1 Register

HOST ADDRESS	N/A			N/A			**HOST SIZE**	
EC OFFSET	44h			32-bit			**EC SIZE**	
POWER	VTR			00h			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	**---**		**D10**	**D9**	**D8**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	FGPT OFF	FGPT ON	FINTT OFF	FINTT ON	FHT1 OFF	FHT1 ON	FHT0 OFF	FHT0 ON

Note: Bits [7:4] have no affect when EC Clock Tree 0 is off.

FHT0 ON

Force Host Clock Tree 0 on.

FHT0 OFF

Force Host Clock Tree 0 off.

Note 7-52 Forcing Host Clock Tree 0 ‘off’ inhibits access to the Clock Tree Control 1 Register register. nSYS_RST is required to re-enable access to this register.

FHT1 ON

Force Host Clock Tree 1 on.

FHT1 OFF

Force Host Clock Tree 1 off.

FINTT ON

Force INTERRUPT AGGREGATOR CLOCK TREE on.

FINTT OFF

Force INTERRUPT AGGREGATOR CLOCK TREE off.

FGPT ON

Force GPIO CLOCK TREE on.

FGPT OFF

Force GPIO CLOCK TREE off.

© 2012-2018 Microchip Technology Inc.

7.8.9 VREG CONTROL REGISTER

TABLE 7-56: VREG Control Register

HOST ADDRESS	N/A				N/A		**HOST SIZE**	
EC OFFSET	48h				32-bit		**EC SIZE**	
POWER	VTR				00h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	**---**		**D10**	**D9**	**D8**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	VREG SUS	MCHP Reserved						

MCHP RESERVED

Writing to MCHP Reserved bits may cause undesirable results.

VREG SUS

TABLE 7-57: VREG SUS BIT ENCODING

VREG SUS	Description
0	1.8V Regulator is always powered.
1	Suspend 1.8V Regulator in SYSTEM HEAVY SLEEP 2, SYSTEM HEAVY SLEEP 3 and SYSTEM DEEPEST SLEEP state (see Table 7-14, “EC Controlled Dynamic Power States,” on page 112)

7.9 VBAT Powered Registers

7.9.1 POWER-FAIL AND RESET STATUS REGISTER

7.9.1.1 Overview

The Power-Fail and Reset Status Register collects and retains the VBAT RST, FLASH and WDT event status when VTR is unpowered. Asserted events can cause interrupts as described in Section 7.10, "Interrupt Interface," on page 150.

TABLE 7-58: Power-Fail and Reset Status Register

HOST ADDRESS	N/A			N/A			**HOST SIZE**	
EC OFFSET	00h			32-bit			**EC SIZE**	
POWER	VBAT			1XX000Xb			VBAT_POR **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	**---**		**D10**	**D9**	**D8**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R/WC	R/WC	R/WC	R	R	R	R	R
BIT NAME	VBAT RST	FLASH	WDT	Reserved				DET32K_IN

DET32K_IN

TABLE 7-59: DET32K_IN BIT ENCODING

DET32K_IN	Description
0	No clock detected on the 32KHZ_IN pin
1	Clock is detected on the 32KHZ_IN pin

WDT

The WDT bit is asserted (‘1’) following a Watch-Dog Timer Forced Reset as described in Section 7.7.4, "Watch-Dog Timer Forced Reset," on page 130. To clear the WDT bit EC firmware must write a ‘1’ to this bit; writing a ‘0’ to the WDT bit has no affect.

FLASH

The FLASH bit is set to ‘1’ by hardware when FLASH_PGM in Table 7-4, “Power, Clocks, and Resets Port List,” on page 96 is asserted. FLASH_PGM is asserted when the Embedded Flash is placed in Program Mode or Erase Mode. To clear the FLASH bit EC firmware must write a ‘1’ to this bit; writing a ‘0’ to the FLASH bit has no affect.

VBAT RST

The VBAT RST bit is set to ‘1’ by hardware when a VBAT_POR is detected. This is the register default value. To clear VBAT RST EC firmware must write a ‘1’ to this bit; writing a ‘0’ to VBAT RST has no affect.

The Clock Enable Register AD_EN, 32K_EN, XTL32K_EN, XTL_SEL, and EXT32K_VBAT bits should not toggle on the same write; i.e., only one bit should be updated at a time.

© 2012-2018 Microchip Technology Inc.

7.9.2 CLOCK ENABLE REGISTER

7.9.2.1 Overview

The Clock Enable register controls the 32KHz oscillators.

TABLE 7-60: Clock Enable Register

HOST ADDRESS	N/A				N/A		**HOST SIZE**	
EC OFFSET	04h				8-bit		**EC SIZE**	
POWER	VBAT				00h		VBAT_POR **DEFAULT**	
					N/A		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	**---**		**D10**	**D9**	**D8**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BIT	D7	D6	D5	D4	D3	D2	D1	**D0**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	R	R	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Reserved		XOSEL	XTL_SEL	XTL32K_EN	AD_EN	32K_EN	EXT32K_VBAT

EXT32K_VBAT

EXT32K_VBAT selects between a VTR powered, or VBAT powered external 32KHz clock input on the 32KHZ_IN pin:

0 - The external 32KHz clock input is VTR powered (default)

1 - The external 32KHz clock input is VBAT powered

32K_EN

The 32K_EN bit controls the 32.768 KHz Silicon Oscillator

0 - The internal 32KHz silicon oscillator is off (default)

Note 7-53 MCLK must not stop before 40 µs min after the 32K_EN bit is asserted.

AD_EN

The Activity Detector bit AD_EN controls 32 KHz Clock Domain Switching as described in Table 7-9.

XTL32K_EN

The XTL32K_EN bit controls the 32.768 KHz Crystal Oscillator

0 - The 32KHz crystal oscillator is off (default)

1 - The 32KHz crystal oscillator is on

XTL_SEL

The XTL_SEL bit selects between the 32.768 KHz Crystal Oscillator and the 32.768 KHz Silicon Oscillator:

0 - The clock source for the 32KHz clock domain comes from the internal silicon oscillator (default)

1 - The clock source for the 32KHz clock domain comes from the external crystal oscillator. Note that the external crystal must be operating before this option is selected.

XOSEL

The XOSEL bit selects whether the 32KHz crystal oscillator is driven by a single-ended input or is connected to a 32.768KHz parallel resonant crystal:

0 - A crystal resonator is connected between the XTAL1 and XTAL2 pins (default)

1 - The XTAL2 pin is used as an input for a single-ended 32 KHz clock; the XTAL1 pin must be tied to VSS.

7.10 Interrupt Interface

The Power, Clocks, and Resets Interrupt Interface inputs include the VBAT RST, FLASH and WDT status bits in the Power-Fail and Reset Status Register. The Interrupt Interface output is PCR_INT in Table 7-4, “Power, Clocks, and Resets Port List,” on page 96. The corresponding bit in the EC Interrupt Aggregator is bit PFR in GIRQ23 Source Register.

Whenever any Interrupt Interface input is asserted, PCR_INT is asserted; when all Interrupt Interface inputs are not asserted, PCR_INT is not asserted. PCR_INT may be masked as described in Section 17.0, "EC Interrupt Aggregator," on page 287.

© 2012-2018 Microchip Technology Inc.

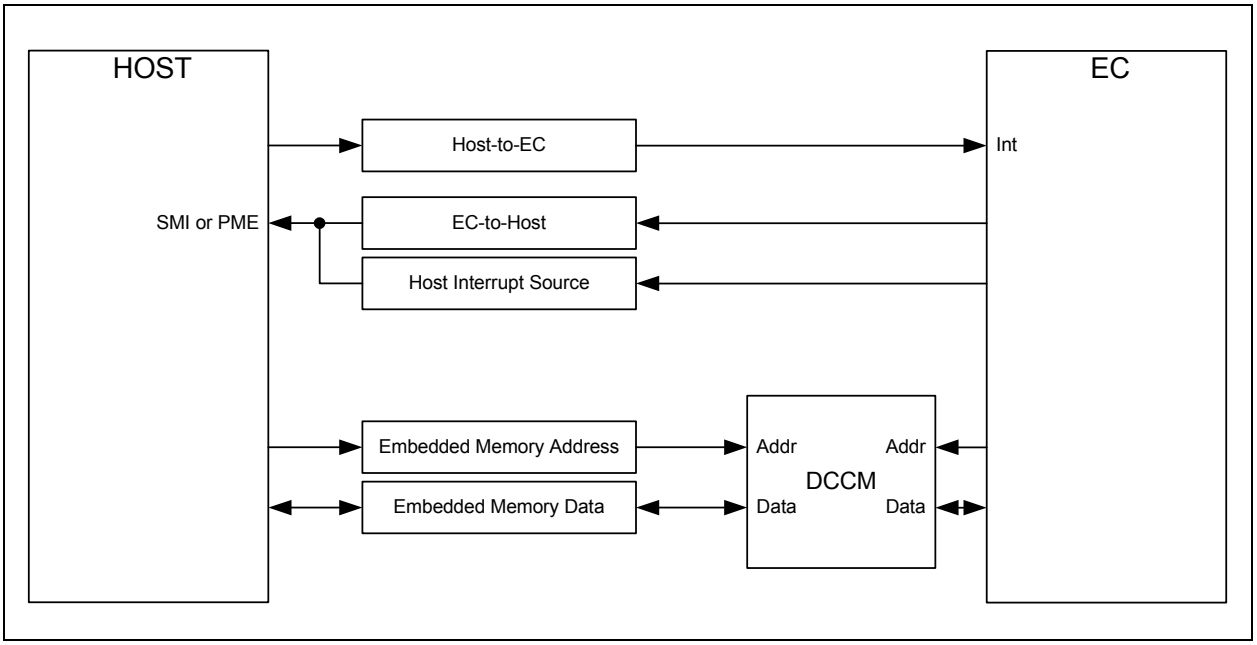
8.0 EMBEDDED MEMORY INTERFACE

8.1 General Description

The Embedded Memory Interface provides a standard run-time mechanism for the host to communicate with the Embedded Controller (EC) and other logical components in the MEC1632 (Figure 8-1). The Embedded Memory Interface includes 12 byte-addressable registers in the Host's I/O address space, as well as 20 bytes of registers that are accessible only by the EC. The Embedded Memory Interface can be used by the Host to read any byte in a region of EC closely-coupled memory, designated by the EC, without requiring any assistance from the EC. A portion of the memory can be written by the Host without any EC assistance as well.

8.1.1 BLOCK DIAGRAM

FIGURE 8-1: EMBEDDED MEMORY INTERFACE BLOCK DIAGRAM



8.2 Power, Clocks and Reset

8.2.1 POWER DOMAIN

This block is powered by the VTR power supply.

8.2.2 CLOCKS

This block has two clock inputs, the LPC Bus Clock and the EC clock.

The Embedded Memory Interface includes support for system-level clock gating. The clock required output is the inversion of the sleep enable input.

8.2.3 RESET

This block is reset when nSYS_RST is asserted.

8.3 Interrupts

Each instance of the Embedded Memory Interface can generate an interrupt event for the HOST-to-EC events. See HOST-to-EC Mailbox Register on page 158. The interrupt source for the EMI is routed onto the EM_MBX bit in the GIRQ15 Source Register and is edge-sensitive, active high.

When the EM_MBX interrupt status bit is cleared while there is data in the HOST-to-EC Mailbox Register, the interrupt remains not asserted until the host writes another byte. If the host writes a second byte to the HOST-to-EC Mailbox Register before the EC processes the EM_MBX interrupt resulting from the first byte, the data from the first byte will be overwritten by the second before the EC has a chance to process the first. This can be avoided with a proper handshaking protocol between the host and EC (for example, the EC should process a data in the HOST-to-EC Mailbox Register before signaling the host that the data register is free).

8.3.1 EMBEDDED MEMORY INTERFACE SIRQ ROUTING

The Embedded Memory Interface can generate a SIRQ event for the EC-to-HOST EC events. See HOST-to-EC Mailbox Register on page 158 and Interrupt Source Register on page 162. This interrupt is routed to the SIRQ block (see SERIRQ Configuration Registers on page 72). For this interrupt, the SELECT on page 72 is cleared to ‘0’ in the Interrupt Configuration Register for the selected SIRQ frame.

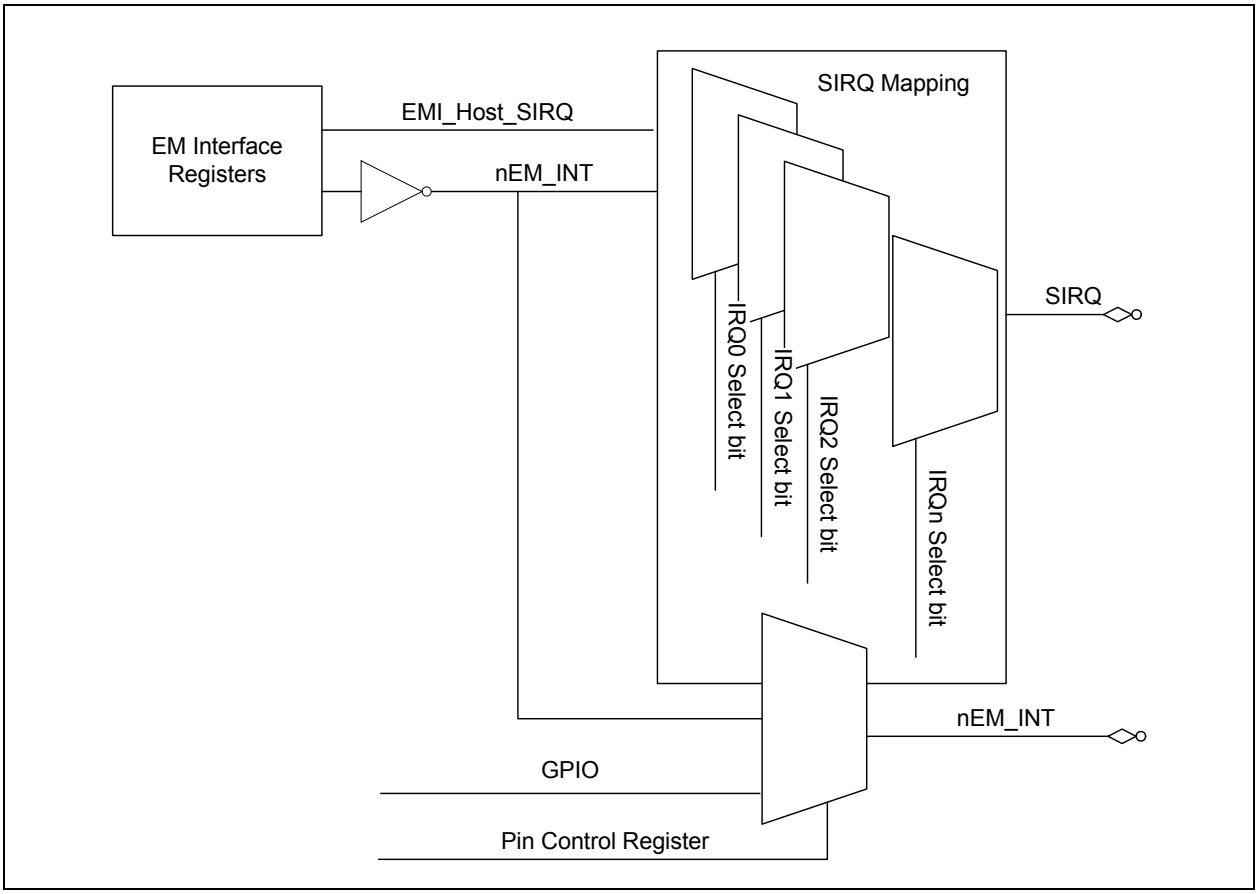
The Embedded Memory Interface can also generate an event on an external pin, nEM_INT. The pin can be routed to SMI or PME inputs, as required. The EC can cause the event to be generated by either writing the EC-to-Host Mailbox Register or by setting any of the enabled EC_SWI[14:0] bits in the Interrupt Source Register to 1. The event can be routed to any frame in the SIRQ stream or to the external pin. To enable routing to the SIRQ stream, the bit SELECT on page 72 is set to ‘1’ in the Interrupt Configuration Register for the selected SIRQ frame. The event can be routed to the pin by selecting the nEM_INT signal function in the associated Pin Control Register on page 396.

The event produces a standard active low on the serial IRQ stream and active low on the open drain nEM_INT pin. See FIGURE 8-2: Embedded Memory Interface SIRQ and nEM_INT routing on page 153.

See Section 5.7.1, "SERIRQ Configuration Registers," on page 72.

© 2012-2018 Microchip Technology Inc.

FIGURE 8-2: EMBEDDED MEMORY INTERFACE SIRQ AND NEM_INT ROUTING



8.4 Description

The Embedded Memory Interface contains a Mailbox that enables the Host to send an 8-bit message to the EC and the EC to send an 8-bit message to the Host. When written by the sender, the messages can generate an interrupt at the receiver.

In addition to the messages that can be exchanged, the Embedded Memory Interface permits the Host to read and write a portion of the EC's Data Closely Coupled Memory (DCCM). Host reads and writes take place without intervention or assistance from the EC.

The Embedded Memory Interface occupies 12 bytes in the Host I/O space. Two bytes constitute the Host-to-EC and EC-to-Host message links. Six bytes are used for the interface into the EC DCCM, two for address and four for data. The four data bytes are used for reads and writes to the EC DCCM DMI.

When the Host reads one of the four bytes in the Embedded Memory Interface data register, data from the DCCM at the address defined by the Embedded Memory Interface address register is returned to the Host. Writes to a byte write the corresponding byte in the DCCM. The Embedded Memory Interface can be configured so that, although Host I/O is always byte at a time, transfers between the Embedded Memory Interface data bytes and the DCCM can configured to occur as single bytes, 2-byte blocks or 4-byte blocks. This is done so that data that the EC treats as 16-bit or 32-bit will be consistent in the Host, even though one byte of the DCCM data may change between two or more 8-bit accesses by the Host.

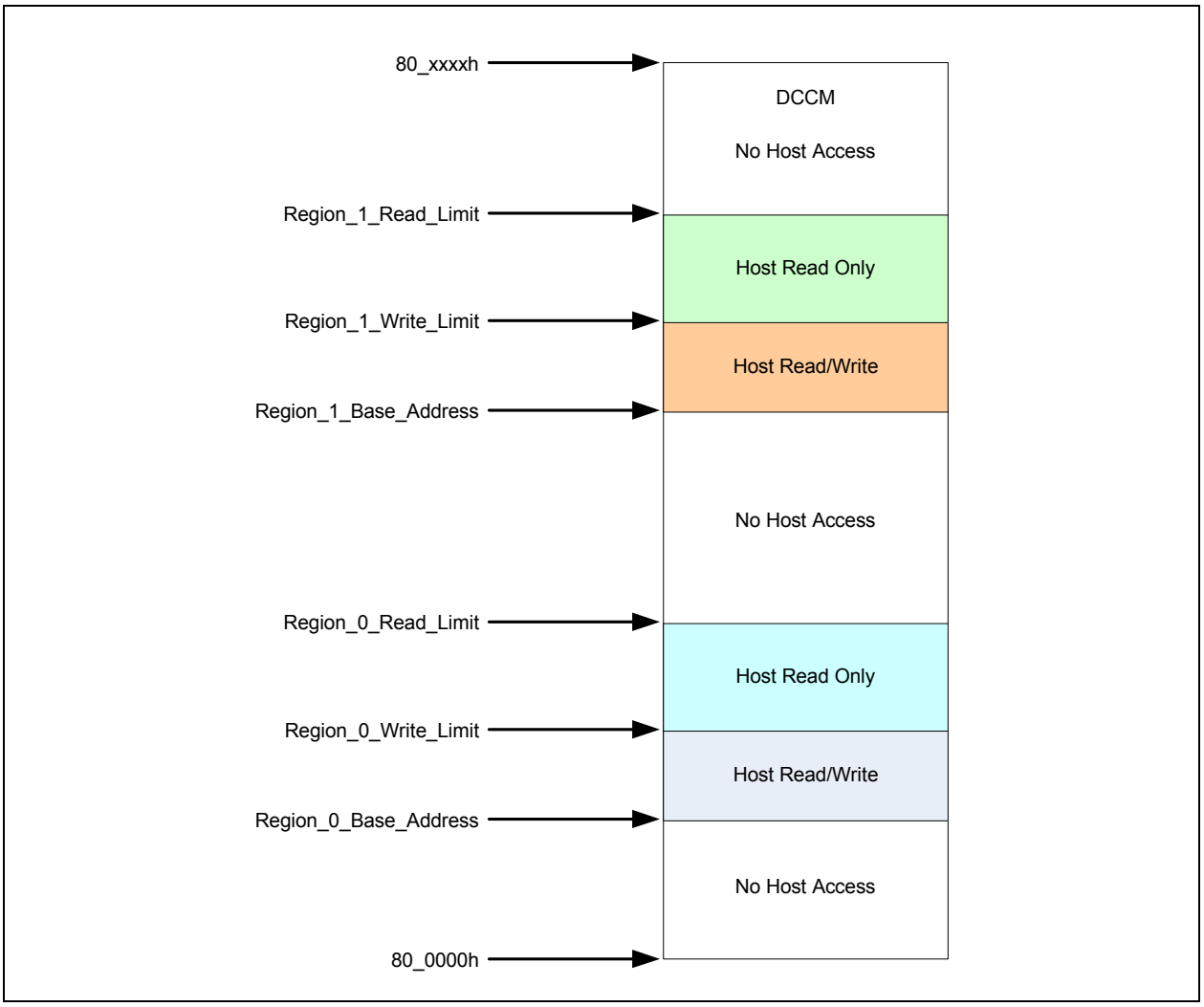
In addition, there is an auto-increment function for the Embedded Memory Interface address register. When enabled, the Host can read or write blocks of memory in the DCCM by repeatedly accessing the Embedded Memory Interface data register, without requiring Host updates to the Embedded Memory Interface address register.

APPLICATION NOTE: the RAM_Select bit in the AHB SRAM Configuration Register must be asserted (‘1’) to properly configure the DCCM for use by the Embedded Memory Interface.

8.4.1 EMBEDDED MEMORY MAP

Each Embedded Memory interface provides direct access for the Host into two windows in the EC DCCM SRAM. This mapping is shown in Figure 8-3, "Embedded Memory Addressing":

FIGURE 8-3: EMBEDDED MEMORY ADDRESSING

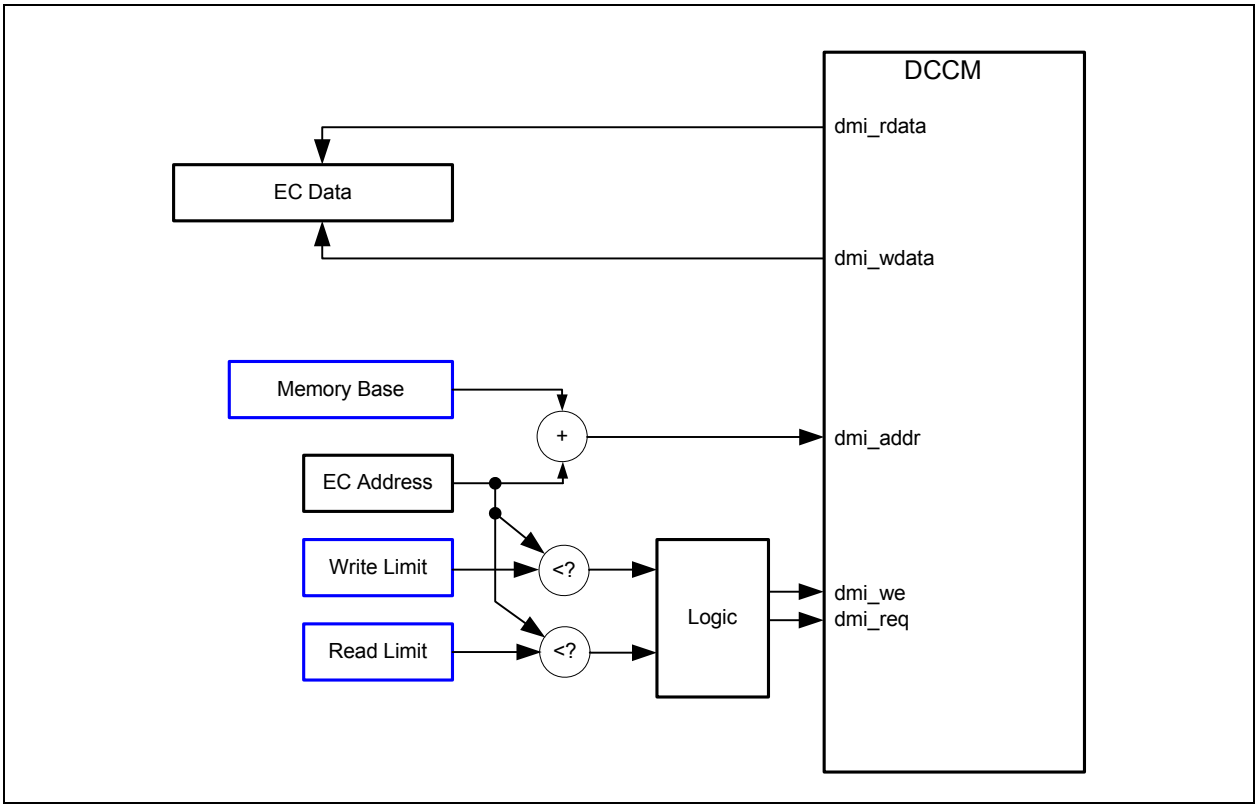


The Base addresses, the Read limits and the Write limits are defined by registers that are in the EC address space and cannot be accessed by the Host. In each region, the Read limit need not be greater than the Write limit. The regions can be contiguous or overlapping.For example, if the Region 0 Read limit is set to 0 and the Write limit is set to a positive number, then the Embedded Memory interface defines a region in the EC memory that the EC can read and write but is write-only for the host. This might be useful for storage of security data, which the Host might wish to send to the EC but should not be readable in the event a virus invades the Host.

© 2012-2018 Microchip Technology Inc.

Each window into the EC memory can be as large as 32K bytes. The Embedded Memory Interface uses the EC's DCCM Direct Memory Interface (DMI) in order to access the memory. Figure 8-4, "Embedded Memory Region Address Control" shows the relationship between one of the regions in the Embedded Memory Interface and the DCCM DMI:

FIGURE 8-4: EMBEDDED MEMORY REGION ADDRESS CONTROL



8.4.2 EMBEDDED MEMORY INTERFACE USAGE

The Embedded Memory Interface provides a generic facility for communication between the Host and the EC and can be used for many functions. Some examples are:

- Virtual registers. A block of read-only memory locations in the DCCM can be used to implement a set of virtual registers. The EC can update these locations with that the Host can later read.
- Program downloading. Because the Instruction Closely Coupled Memory is implemented in the same SRAM as the DCCM, the Embedded Memory Interface can be used by the Host to download new program segments for the EC. The Read/Write window would be configured by the Host to point to the beginning of the loadable program region, which could then be loaded by the Host.
- Data exchange. The Read/Write portion of the memory window can be used to contain a communication packet. The Host, by default, "owns" the packet, and can write it at any time. When the Host wishes to communicate with the EC, it sends the EC a command, through the Host-to-EC message facility, to read the packet and perform some operations as a result. When it is completed processing the packet, the EC can inform the Host, either through a message in the EC-to-Host channel or by triggering an event such as an SMI directly. If return results are required, the EC can write the results into the Read/Write region, which the Host can read directly when it is informed that the EC has completed processing. Depending on the command, the operations could entail update of virtual registers in the DCCM, reads of any register in the EC address space, or writes of any register in the EC address space. Because there are two regions that are defined by the base registers, the memory used for the communication packet does not have to be contiguous with a set of virtual registers.

Because there are two Embedded Memory Interface memory regions, the Embedded Memory Interface cannot be used for more than two of these functions at a time. The Host can request that the EC switch from one function to another through the use of the Host-to-EC mailbox register.

The Application ID Register is provided to help software applications track ownership of an Embedded Memory Interface. An application can write the Application ID Register with its Application ID, then immediately read it back. If the read value is not the same as the value written, then another application has ownership of the interface.

Note:

- The protocol used to pass commands back and forth through the Embedded Memory Interface Registers Interface is left to the System designer. Microchip can provide an application example of working code in which the host uses the Embedded Memory Interface registers to gain access to all of the EC registers.
- The EC must be awake (i.e., the EC must not be in sleep mode with its clocks gated) in order for the EMI to read or write data in the DCCM/ICCM. System software can insure that the EC is awake when the Host is accessing the EMI with an appropriate protocol using the Host-to-EC mailbox register. Before accessing the memory, the Host sends a “Do Not Sleep” command to the EC through the mailbox register. Writing the register generates an EC interrupt, which wakes the EC. The “Do Not Sleep” command sets a state bit that the EC can check before it issues the Sleep command. When the Host has completed accessing the memory, the Host sends a “Sleep Permitted” command to the EC.

8.5 Registers

The Embedded Memory Interface has its own Logical Device Number, and Base Address as indicated in Table 8-1. The Host LPC I/O addresses for the Embedded Memory Interface are selected via a Base Address Register (see Section 5.6.2, "Base Address Registers," on page 66). LPC access to configuration registers is through Host Access Configuration Port (see Section 5.5.1, "Host Access Port," on page 65.)

Table 8-2 is a register summary for the Embedded Memory Interface block.

TABLE 8-1: Embedded Memory Interface BASE ADDRESS TABLE

Embedded Memory Interface **Instance**	**LDN from (Table 4-2 on page 59)**	**AHB Base Address**
EM Interface	10h	FF_4000h

Note: The Host LPC I/O addresses for this instance is selected via a Base Address Register (see Section 5.6.2, "Base Address Registers," on page 66). LPC access to configuration registers is through the Host Access Configuration Port (see Section 5.5.1, "Host Access Port," on page 65.)

The Table 8-2 is a register summary for one instance of the Embedded Memory Interface. The LPC I/O address for each Run-Time Register is described below as an offset from its Base Address Register. Each EC address is indicated as an SPB Offset from its AHB base address. Each Configuration register is accessed through the Host Access Port is via its LDN indicated in Table 8-1 on page 156 and its Host Access Port index which is described as “Host Config Index” in the tables below.

TABLE 8-2: Embedded Memory Interface REGISTER SUMMARY

	Host I/O Access			**EC Interface**			
Register Name	**Host I/O Offset**	**SPB Offset**	**Host Type**	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
HOST-to-EC Mailbox Register	00h	00h	R/W	00h	0	R/W	Note 8-1
EC-to-Host Mailbox Register	01h	01h	R/WC	01h	1	R/WC	Note 8-2
EC Address Register	02h 03h	02h 03h	R/W	02h	2-3	R/W	

© 2012-2018 Microchip Technology Inc.

TABLE 8-2: Embedded Memory Interface REGISTER SUMMARY (CONTINUED)

	Host I/O Access			EC Interface			
Register Name	**Host I/O Offset**	**SPB Offset**	**Host Type**	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
EC Data Register	04h 05h 06h 07h	04h 05h 06h 07h	R/W	04h	0-3	R/W	
Interrupt Source Register	08h 09h	08h 09h	Table 8-9	08h	0-1	Table 8-9	
Interrupt Mask Register	0Ah 0Bh	0Ah 0Bh	R/W	0Ah	2-3	R/W	
Application ID Register	0Ch	0Ch	R/W	0Ch	0	R/W	
HOST-to-EC Mailbox Register	-	-	-	100h	0	R/WC	Note 8-1
EC-to-Host Mailbox Register	-	-	-	101h	1	R/W	Note 8-2
Memory Base Address 0 Register	-	-	-	104h	0-3	R/W	
Memory Read Limit 0 Register	-	-	-	108h	0-1	R/W	
Memory Write Limit 0 Register	-	-	-	10Ah	2-3	R/W	
Memory Base Address 1 Register	-	-	-	10Ch	0-3	R/W	
Memory Read Limit 1 Register	-	-	-	110h	0-1	R/W	
Memory Write Limit 1 Register	-	-	-	112h	2-3	R/W	
Interrupt Set Register	-	-	-	114h	0-1	R/W	
Host Clear Enable Register	-	-	-	116h	2-3	R/W	

Note 8-1 Interrupt is cleared when read by the EC.

Note 8-2 Interrupt is cleared when read by the host.

8.5.1 EMBEDDED MEMORY INTERFACE CONTROL REGISTERS

Mailbox Register, HOST-to-EC, and Mailbox Register, EC-to-HOST, are specifically designed to pass commands between the host and the EC (FIGURE 8-1: on page 151). If enabled, these registers can generate interrupts.

When the host performs a write of the HOST-to-EC mailbox register, an interrupt will be generated and seen by the EC if unmasked. When the EC writes the HOST-to-EC mailbox register using the EC-only offset address 100h, it can reset the register to 00h, providing a simple means for the EC to inform the host that an operation has been completed.

When the EC writes the EC-to-HOST mailbox register, an SIRQ event or an event such as SMI or PME may be generated and seen by the host if unmasked. The Host CPU can reset the EC-to-HOST mailbox register to 00h, providing a simple means for the host to inform that EC that an operation has been completed.

PROGRAMMER'S NOTE: The protocol used to pass commands back and forth through the Mailbox Registers Interface is left to the System designer. Microchip can provide an application example of working code in which the host uses the Mailbox registers to gain access to all of the EC registers.

8.6 Registers

8.6.1 HOST-TO-EC MAILBOX REGISTER

TABLE 8-3: HOST-TO-EC MAILBOX REGISTER

HOST OFFSET	00h			8-Bit			**HOST SIZE**	
EC OFFSET	00h			8-Bit			**EC SIZE**	
POWER	VTR			00h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	HOST_EC_MBOX[7:0]							

TABLE 8-4: HOST-TO-EC MAILBOX REGISTER

HOST OFFSET	-			-			**HOST SIZE**	
EC OFFSET	100h			8-Bit			**EC SIZE**	
POWER	VTR			00h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
BIT NAME	HOST_EC_MBOX[7:0]							

HOST_EC_MBOX[7:0]

If enabled, an interrupt to the EC marked by the EM_MBX bit in the GIRQ15 Source Register will be generated whenever the Host writes this register. The Host and the EC can read and write this register at offset 000h. The EC can also read this register at offset 100h.

Writes of a 1 to any bit in this register by the EC to this register at offset 100h will cause the bit to be cleared. Writes of a 0 to any bit have no effect.

© 2012-2018 Microchip Technology Inc.

8.6.2 EC-TO-HOST MAILBOX REGISTER

TABLE 8-5: EC-TO-HOST MAILBOX REGISTER

HOST OFFSET	01h					8-Bit	**HOST SIZE**	
EC OFFSET	01h					8-Bit	**EC SIZE**	
POWER	VTR					00h	nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
EC TYPE	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
BIT NAME	EC_HOST_MBOX[7:0]							

TABLE 8-6: EC-TO-HOST MAILBOX REGISTER

HOST OFFSET	-					-	**HOST SIZE**	
EC OFFSET	101h					8-Bit	**EC SIZE**	
POWER	VTR					00h	nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	EC_HOST_MBOX[7:0]							

EC_HOST_MBOX[7:0]

An EC write to this register will set bit EC_WR in the Interrupt Source Register to '1b'. If enabled, setting bit EC_WR to '1b' generates a Host SIRQ event as well was the external nEM_INT event. The EC can also read and write this register at offset 101h.

Writes of a 1 to any bit in this register at offset 01h, by the Host or by the EC, will cause the bit to be cleared. Writes of a 0 to any bit have no effect.

8.6.3 EC ADDRESS REGISTER

TABLE 8-7: EC ADDRESS REGISTER

HOST OFFSET	Byte 0: 02h Byte 1: 03h			8-bit			**HOST SIZE**	
EC OFFSET	02h			16-bit			**EC SIZE**	
POWER	VTR			0000h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Region	EC_Address[14:8]						
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	EC_Address[7:2]						Access_Type	

ACCESS_TYPE

This field defines the type of access that occurs when the EC Data Register is read or written.

00: 8-bit access. Any byte read of Byte 0 through Byte 3 in the EC Data Register causes the corresponding byte within the 32-bit double word addressed by EC_Address to be loaded into the byte of EC Data Register and returned by the read. Any byte write to Byte 0 through Byte 3 in the EC Data Register writes the corresponding byte within the 32-bit double word addressed by EC_Address, as well as the byte of the EC Data Register.

01: 16-bit access. A read of Byte 0 in the EC Data Register causes the 16 bits in the DCCM at an offset of EC_Address to be loaded into Byte 0 and Byte 1 of the EC Data Register. The read then returns the contents of Byte 0. A read of Byte 2 in the EC Data Register causes the 16 bits in the DCCM at an offset of EC_Address+2 to be loaded into Byte 2 and Byte 3 of the EC Data Register. The read then returns the contents of Byte 2. A read of Byte 1 or Byte 3 in the EC Data Register return the contents of the register, without any update from the DCCM.

A write of Byte 1 in the EC Data Register causes Bytes 1 and 0 of the EC Data Register to be written into the 16 bits in the DCCM at an offset of EC_Address. A write of Byte 3 in the EC Data Register causes Bytes 3 and 2 of the EC Data Register to be written into the 16 bits in the DCCM at an offset of EC_Address+2. A write of Byte 0 or Byte 2 in the EC Data Register updates the contents of the register, without any change to the DCCM.

10: 32-bit access. A read of Byte 0 in the EC Data Register causes the 32 bits in the DCCM at an offset of EC_Address to be loaded into the entire EC Data Register. The read then returns the contents of Byte 0. A read of Byte 1, Byte 2 or Byte 3 in the EC Data Register returns the contents of the register, without any update from the DCCM.

A write of Byte 3 in the EC Data Register causes the EC Data Register to be written into the 32 bits in the DCCM at an offset of EC_Address. A write of Byte 0, Byte 1 or Byte 2 in the EC Data Register updates the contents of the register, without any change to the DCCM.

11: Auto-increment 32-bit access. This defines a 32-bit access, as in the 10 case. In addition, any read or write of Byte 3 in the EC Data Register causes the EC Address Register to be incremented by 1. That is, the EC_Address field will point to the next 32-bit double word in the DCCM.

EC_ADDRESS[14:2]

This field defines the location in memory that can be read and/or written with the EC Data Register. The address is an offset from the base of the Host-accessible region in the EC DCCM SRAM. The base of the Host-accessible region.

© 2012-2018 Microchip Technology Inc.

REGION

When this bit is 0, the address defined by EC_Address[14:2] is relative to the base address specified by the Memory Base Address 0 Register. When this bit is 1, the address defined by EC_Address[14:2] is relative to the base address specified by the Memory Base Address 1 Register.

8.6.4 EC DATA REGISTER

TABLE 8-8: EC DATA REGISTER

HOST OFFSET	Byte 0: 04h Byte 1: 05h Byte 2: 06h Byte 3: 07h			8-bit			**HOST SIZE**	
EC OFFSET	04h			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			**VTR POR DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D31**	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Data3[7:0]							
BYTE2 BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Data2[7:0]							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Data1[7:0]							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Data0[7:0]							

DATA

This is a 32-bit register which returns data to the Host from the EC DCCM at the address specified by EC_Address[14:2]. The description of bits Access_Type in the EC Address Register defines which reads and writes from the Host trigger transfers of data between this register and the DCCM.

A write to the EC Data Register when the EC Address Register is in a read-only or a no-access region, as defined by the Memory Base and Limit registers, will update the EC Data Register but memory will not be modified. A read to the EC Data Register when the EC Address Register is in a no-access region, as defined by the Memory Base and Limit registers, will not trigger a memory read and will not modify the EC Data Register. In auto-increment mode (Access_Type=11b), reads of Byte 3 of the EC Data Register will still trigger increments of the EC Address Register when the address is out of bounds, while writes of Byte 3 will not.

8.6.5 INTERRUPT SOURCE REGISTER

TABLE 8-9: INTERRUPT SOURCE REGISTER

HOST OFFSET	Byte 0: 08h Byte 1: 09h			8-Bit			**HOST SIZE**	
EC OFFSET	08h			16-Bit			**EC SIZE**	
POWER	VTR			0000h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
EC TYPE	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
BIT NAME	EC_SWI[14:7]							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R
EC TYPE	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R
BIT NAME	EC_SWI[6:0]							EC_WR

EC_WR

This bit is set autonomously when the EC-to-Host Mailbox Register has been written by the EC at offset 101h. An SIRQ event or an external nEM_INT event to the Host is generated when any bit in this register (EC_WR or any bit in EC_SWI[14:0]) is ‘1b’ and the corresponding bit in the Interrupt Mask Register register is ‘1b’.

This bit is automatically cleared by a read of the EC-to-Host Mailbox Register at offset 01h.

EC_SWI[14:0]

Each bit in this field is cleared when written with a ‘1b’. The ability to clear the bit can be disabled if the corresponding bit in the Host Clear Enable Register is set to ‘0b’.

The EC can generate an interrupt to the Host by setting any bit in this field to ‘1b’. The EC can set bits to ‘1b’ by writing the corresponding bit in the Interrupt Set Register to ‘1b’.

© 2012-2018 Microchip Technology Inc.

8.6.6 INTERRUPT MASK REGISTER

TABLE 8-10: INTERRUPT MASK REGISTER

HOST OFFSET	Byte 0: 0Ah Byte 1: 0Bh			8-Bit			**HOST SIZE**	
EC OFFSET	0Ah			16-Bit			**EC SIZE**	
POWER	VTR			0000h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	EC_SWI_EN[14:7]							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	EC_SWI_EN[6:0]							EC_WR_EN

EC_WR_EN

If this bit is ‘1b’, the interrupt generated by bit EC_WR in the Interrupt Source Register is enabled.

EC_SWI_EN[14:0]

Each bit that is set to ‘1b’ in this field enables the generation of and interrupt by the corresponding bit in the EC_SWI[14:0] field in the Interrupt Source Register.

8.6.7 APPLICATION ID REGISTER

TABLE 8-11: APPLICATION ID REGISTER

HOST OFFSET	0Ch			8-Bit			**HOST SIZE**	
EC OFFSET	0Ch			8-Bit			**EC SIZE**	
POWER	VTR			00h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Application_ID[7:0]							

APPLICATION_ID

When this field is 00h it can be written with any value. When set to a non-zero value, writing that value will clear this register to 00h. When set to a non-zero value, writing any value other than the current contents will have no effect.

8.6.8 MEMORY BASE ADDRESS 0 REGISTER

This register is accessible to the EC only.

TABLE 8-12: MEMORY BASE ADDRESS 0 REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC OFFSET	104h			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			**VTR POR DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D31**	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE2 BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Memory_Base_Address_0[23:16]							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Memory_Base_Address_0[15:8]							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R	R
BIT NAME	Memory_Base_Address_0[7:2]						Reserved	

MEMORY_BASE_ADDRESS_0[23:2]

This register defines the beginning of a region in the Embedded Controller's Data Closely Coupled Memory that is shared between the Host and the EC. The region defined by this base register, Region 0, is used when bit 15 of the EC Address Register is 0. The access will be to a memory location at an offset defined by the contents of the EC Address Register, relative to the beginning of the region defined by this register. Therefore, a read or write to the memory that is triggered by the EC Data Register will occur at DCCM_Base_Address + Memory_Base_Address_0[23:2] + EC_Address[14:2].

For example, if Region = 0, the Memory_Base_Address_0[23:2] = 1000h and the EC_Address[14:2] = 20h, then the AHB address of the access will be 80_1020h.

© 2012-2018 Microchip Technology Inc.

8.6.9 MEMORY READ LIMIT 0 REGISTER

This register is accessible to the EC only.

TABLE 8-13: MEMORY READ LIMIT 0 REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC OFFSET	108h			16-bit			**EC SIZE**	
POWER	VTR			0000h			**VTR POR DEFAULT**	
BUS	LPC SPB							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Reserved	Memory_Read_Limit_0[14:8]						
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R	R
BIT NAME	Memory_Read_Limit_0[7:2]						Reserved	

MEMORY_READ_LIMIT_0[14:2]

Whenever a read of any byte in EC Data Register is attempted, and bit 15 of EC_Address is 0, the field EC_Address[14:2] in the EC Address Register is compared to this field. As long as EC_Address[14:2] is less than Memory_Read_Limit_0[14:2] the EC Data Register will be loaded from the DCCM.

8.6.10 MEMORY WRITE LIMIT 0 REGISTER

This register is accessible to the EC only.

TABLE 8-14: MEMORY WRITE LIMIT 0 REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC OFFSET	10Ah			16-bit			**EC SIZE**	
POWER	VTR			0000h			**VTR POR DEFAULT**	
BUS	LPC SPB							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Reserved	Memory_Write_Limit_0[14:8]						
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R	R
BIT NAME	Memory_Write_Limit_0[7:2]						Reserved	

© 2012-2018 Microchip Technology Inc.

MEMORY_WRITE_LIMIT_0[14:2]

Whenever a write of any byte in EC Data Register is attempted and bit 15 of EC_Address is 0, the field EC_Address[14:2] in the EC Address Register is compared to this field. As long as EC_Address[14:2] is less than Memory_Write_Limit_0[14:2] the addressed bytes in the EC Data Register will be written into the DCCM. If EC_Address[14:2] is greater than or equal to Memory_Write_Limit_0[14:2] no writes will take place.

8.6.11 MEMORY BASE ADDRESS 1 REGISTER

This register is accessible to the EC only.

TABLE 8-15: MEMORY BASE ADDRESS 1 REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC OFFSET	10Ch			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			**VTR POR DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D31**	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE2 BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Memory_Base_Address_1[23:16]							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Memory_Base_Address_1[15:8]							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R	R
BIT NAME	Memory_Base_Address_1[7:2]						Reserved	

MEMORY_BASE_ADDRESS_1[23:2]

This register defines the beginning of a region in the Embedded Controller's Data Closely Coupled Memory that is shared between the Host and the EC. The region defined by this base register, Region 1, is used when bit 15 of the EC Address Register is 1. The access will be to a memory location at an offset defined by the contents of the EC Address Register, relative to the beginning of the region defined by this register. Therefore, a read or write to the memory that is triggered by the EC Data Register will occur at DCCM_Base_Address + Memory_Base_Address_1[23:2] + EC_Address[14:2].

For example, if Region = 1, the Memory_Base_Address_1[23:2] = 1000h and the EC_Address[14:2] = 20h, then the AHB address of the access will be 80_1020h.

© 2012-2018 Microchip Technology Inc.

8.6.12 MEMORY READ LIMIT 1 REGISTER

This register is accessible to the EC only.

TABLE 8-16: MEMORY READ LIMIT 1 REGISTER

HOST OFFSET	N/A				N/A		**HOST SIZE**	
EC OFFSET	110h				16-bit		**EC SIZE**	
POWER	VTR				0000h		**VTR POR DEFAULT**	
BUS	LPC SPB							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Reserved	Memory_Read_Limit_1[14:8]						
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R	R
BIT NAME	Memory_Read_Limit_1[7:2]						Reserved	

MEMORY_READ_LIMIT_1[14:2]

Whenever a read of any byte in EC Data Register is attempted, and bit 15 of EC_Address is 1, the field EC_Address[14:2] in the EC Address Register is compared to this field. As long as EC_Address[14:2] is less than Memory_Read_Limit_1[14:2] the EC Data Register will be loaded from the DCCM.

8.6.13 MEMORY WRITE LIMIT 1 REGISTER

This register is accessible to the EC only.

TABLE 8-17: MEMORY WRITE LIMIT 1 REGISTER

HOST OFFSET	N/A				N/A		**HOST SIZE**	
EC OFFSET	112h				16-bit		**EC SIZE**	
POWER	VTR				0000h		**VTR POR DEFAULT**	
BUS	LPC SPB							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Reserved	Memory_Write_Limit_1[14:8]						
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R	R
BIT NAME	Memory_Write_Limit_1[7:2]						Reserved	

MEMORY_WRITE_LIMIT_1[14:2]

Whenever a write of any byte in EC Data Register is attempted and bit 15 of EC_Address is 1, the field EC_Address[14:2] in the EC Address Register is compared to this field. As long as EC_Address[14:2] is less than Memory_Write_Limit_1[14:2] the addressed bytes in the EC Data Register will be written into the DCCM. If EC_Address[14:2] is greater than or equal to Memory_Write_Limit_1[14:2] no writes will take place.

8.6.14 INTERRUPT SET REGISTER

This register is accessible to the EC only.

TABLE 8-18: INTERRUPT SET REGISTER

HOST OFFSET	N/A				N/A		**HOST SIZE**	
EC OFFSET	114h				16-Bit		**EC SIZE**	
POWER	VTR				0000h		nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/WS	R/WS	R/WS	R/WS	R/WS	R/WS	R/WS	R/WS
BIT NAME	EC_SWI_Set[14:7]							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/WS	R/WS	R/WS	R/WS	R/WS	R/WS	R/WS	R
BIT NAME	EC_SWI_Set[6:0]							Reserved

EC_SWI_SET[14:0]

This register provides the EC with a means of updating the Interrupt Source Register. Writing a bit in this field with a ‘1b’ sets the corresponding bit in the Interrupt Source Register to ‘1b’. Writing a bit in this field with a ‘0b’ has no effect. Reading this field returns the current contents of the Interrupt Source Register.

© 2012-2018 Microchip Technology Inc.

8.6.15 HOST CLEAR ENABLE REGISTER

This register is accessible to the EC only.

TABLE 8-19: HOST CLEAR ENABLE REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC OFFSET	116h			16-Bit			**EC SIZE**	
POWER	VTR			0000h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Host_Clr_Enable[14:7]							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
BIT NAME	Host_Clr_Enable[6:0]							Reserved

HOST_CLR_ENABLE[14:0]

When a bit in this field is ‘0b’, the corresponding bit in the Interrupt Source Register cannot be cleared by writes to the Interrupt Source Register. When a bit in this field is ‘1b’, the corresponding bit in the Interrupt Source Register can be cleared when that register bit is written with a ‘1b’.

These bits allow the EC to control whether the status bits in the Interrupt Source Register are based on an edge or level event.

9.0 ACPI EMBEDDED CONTROLLER INTERFACE (ACPI-ECI)

9.1 Introduction

The ACPI Embedded Controller Interface (ACPI-ECI) is a Host/EC Message Interface. The ACPI specification defines the standard hardware and software communications interface between the OS and an embedded controller. This interface allows the OS to support a standard driver that can directly communicate with the embedded controller, allowing other drivers within the system to communicate with and use the EC resources; for example, Smart Battery and AML code.

The ACPI Embedded Controller Interface (ACPI-ECI) provides a four byte full duplex data interface which is a superset of the standard ACPI Embedded Controller Interface (ACPI-ECI) one byte data interface. The ACPI Embedded Controller Interface (ACPI-ECI) defaults to the standard one byte interface.

Note 9-1 The EC host in Table 9-8 & Table 9-10 corresponds to the EC in the ACPI specification. This interface is referred to elsewhere in this chapter as ACPI_EC.

Note 9-2 The LPC host in Table 9-8 & Table 9-10 corresponds to the “System Host Interface to OS” in the ACPI specification. This interface is referred to elsewhere in this chapter as ACPI_OS.

9.2 References

1. Advanced Configuration and Power Interface Specification, Revision 4.0 June 16, 2009, Hewlett-Packard Corporation Intel Corporation Microsoft Corporation Phoenix Technologies Ltd. Toshiba Corporation

9.3 Terminology

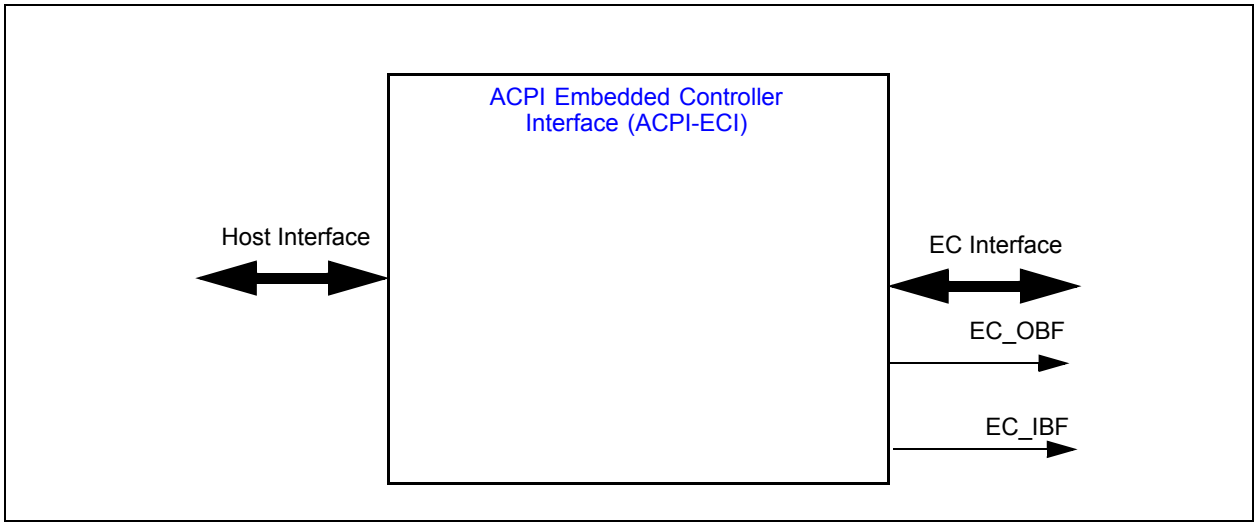
TABLE 9-1: TERMINOLOGY

Term	Definition
ACPI_EC	The EC host corresponding to the ACPI specification interface to the EC.
ACPI_OS	The LPC host corresponding to the ACPI specification interface to the “System Host Interface to OS”. ACPI_OS terminology is not meant to distinguish the ACPI System Management from Operating System but merely the hardware path upstream towards the CPU.

9.4 Interface

This block is designed to be accessed externally and internally via a register interface.

FIGURE 9-1: ACPI EMBEDDED CONTROLLER INTERFACE (ACPI-ECI) I/O DIAGRAM



© 2012-2018 Microchip Technology Inc.

9.4.1 SIGNAL DESCRIPTION

TABLE 9-2: SIGNAL DESCRIPTION TABLE

Name	Direction	Description
EC_IBF	Output	Output follows IBF in STATUS OS-Register on page 179
EC_OBF	Output	Output follows inversion of OBF in STATUS OS-Register on page 179

9.4.2 HOST INTERFACE

The registers defined for the ACPI Embedded Controller Interface (ACPI-ECI) are accessible by the System Host and the Embedded Controller as indicated in Section 9.11, "EC-Only Registers" and Section 9.10, "Runtime Registers".

9.5 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

9.5.1 POWER DOMAINS

TABLE 9-3: POWER SOURCES

Name	Description
VTR	The logic and registers implemented in this block reside on this single power well.

9.5.2 CLOCK INPUTS

This block only requires the Host interface clocks to synchronize registers access.

9.5.3 RESETS

TABLE 9-4: RESET SIGNALS

Name	Description
nSYS_RST	nSYS_RST resets all the logic and registers in ACPI Embedded Controller Interface (ACPI-ECI).

9.6 Interrupts

This section defines the Interrupt Sources generated from this block.

TABLE 9-5: INTERRUPTS

Source	Description
EC_OBF	EC_OBF interrupt is asserted when the OBF in the STATUS EC-Register is cleared to ‘0’.
EC_IBF	EC_IBF interrupt is asserted when the IBF in the STATUS EC-Register is set to ‘1’.

Note: The usage model from the ACPI specification requires both SMI’s and SCI’s. The ACPI_OS SMI & SCI interrupts are not implemented in the ACPI Embedded Controller Interface (ACPI-ECI). The SMI_EVT and SCI_EVT bits in the Section 9.10.6, "STATUS OS-Register," on page 179 are software flags and this block does not initiate SMI or SCI events.

9.7 Low Power Modes

The ACPI Embedded Controller Interface (ACPI-ECI) automatically enters low power mode when no transaction targets it.

9.8 Description

The ACPI Embedded Controller Interface (ACPI-ECI) provides an APCI-EC interface that adheres to the ACPI specification. The ACPI Embedded Controller Interface (ACPI-ECI) includes two modes of operation: Legacy Mode and Four-byte Mode.

The ACPI Embedded Controller Interface (ACPI-ECI) defaults to Legacy Mode which provides single byte Full Duplex operation. Legacy Mode corresponds to the ACPI specification functionality as illustrated in FIGURE 9-2: on page 173. The EC interrupts in FIGURE 9-2: on page 173 are implemented as EC_OBF & EC_IBF. See Section 9.6, "Interrupts," on page 171.

In Four-byte Mode, the ACPI Embedded Controller Interface (ACPI-ECI) provides four byte Full Duplex operation. Four-byte Mode is a superset of the ACPI specification functionality as illustrated in FIGURE 9-2: on page 173.

Both Legacy Mode & Four-byte Mode provide Full Duplex Communication which allows data/command transfers in one direction while maintaining data from the other direction; communications can flow both ways simultaneously.

In Legacy Mode, the ACPI Embedded Controller Interface (ACPI-ECI) contains three registers: ACPI OS COMMAND Register, STATUS OS-Register, and OS2EC Data EC-Register Byte 0. These standard registers occupy two addresses in the ACPI_OS address space (see Table 9-9).

The OS2EC Data EC-Register Byte 0 and ACPI OS COMMAND Register registers appear as a single 8-bit data register in the ACPI_EC address space. The CMD bit in the STATUS OS-Register is used by the ACPI_EC to discriminate commands from data written by the ACPI_OS to the ACPI_EC. The CMD bit is controlled by hardware: ACPI_OS writes to the OS2EC Data EC-Register Byte 0 register clear the CMD bit; ACPI_OS writes to the ACPI OS COMMAND Register set the CMD bit.

© 2012-2018 Microchip Technology Inc.

FIGURE 9-2: BLOCK DIAGRAM CORRESPONDING TO THE ACPI SPECIFICATION

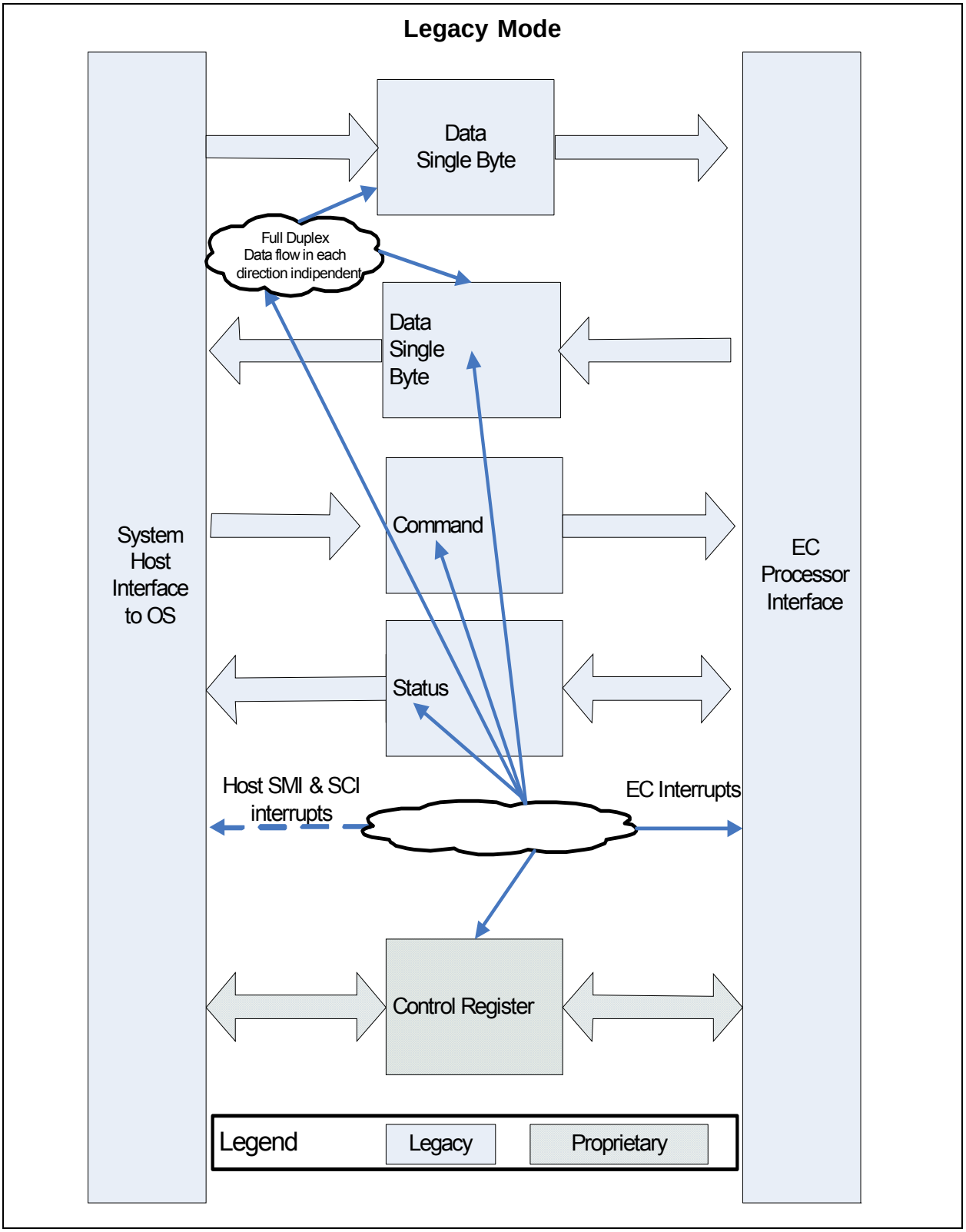
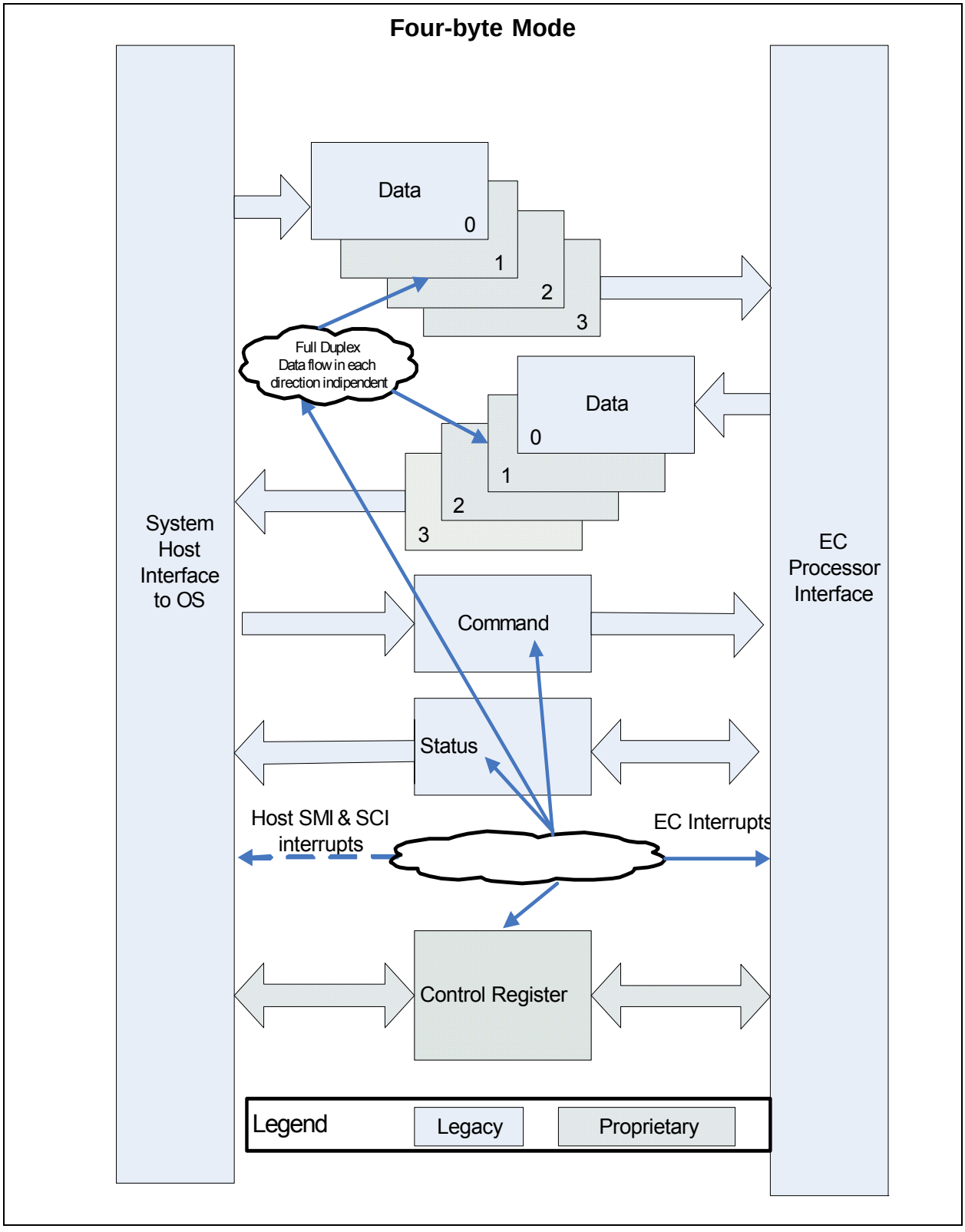


FIGURE 9-2: BLOCK DIAGRAM CORRESPONDING TO THE ACPI SPECIFICATION



© 2012-2018 Microchip Technology Inc.

9.9 Register Aliasing between Runtime and EC-Only Registers

Table 9-6, "Runtime Registers Aliasing into EC-Only Registers" indicates the aliasing from Runtime Registers to EC-Only Registers. The “Aliased Access” column distinguishes the aliasing based on access type. See individual register descriptions for more details.

TABLE 9-6: Runtime Registers ALIASING INTO EC-ONLY REGISTERS

Runtime Registers				Map of Aliasing into EC-Only Registers		
LPC Offset	**EC Offset**	Runtime Registers **Register Name (Mnemonic)**	**Aliased Access**	**EC Offset**	EC-Only Registers **Register Name (Mnemonic)**	**Aliased Access**
00h	00h	ACPI OS Data Register Byte 0	W	108h	OS2EC Data EC-Register Byte 0	R
00h	00h	ACPI OS Data Register Byte 0	R	100h	EC2OS Data EC-Register Byte 0	W
01h	01h	ACPI OS Data Register Byte 1	W	109h	OS2EC Data EC-Register Byte 1	R
01h	01h	ACPI OS Data Register Byte 1	R	101h	EC2OS Data EC-Register Byte 1	W
02h	02h	ACPI OS Data Register Byte 2	W	10Ah	OS2EC Data EC-Register Byte 2	R
02h	02h	ACPI OS Data Register Byte 2	R	102h	EC2OS Data EC-Register Byte 2	W
03h	03h	ACPI OS Data Register Byte 3	W	10Bh	OS2EC Data EC-Register Byte 3	R
03h	03h	ACPI OS Data Register Byte 3	R	103h	EC2OS Data EC-Register Byte 3	W
04h	04h	ACPI OS COMMAND Register	W	108h	OS2EC Data EC-Register Byte 0	R
04h	04h	STATUS OS-Register	R	104h	STATUS EC-Register	W
05h	05h	Byte Control OS-Register	R	105h	Byte Control EC-Register	R/W
06h	06h	Reserved		106h	Reserved	
07h	07h	Reserved		107h	Reserved	

Table 9-7, "EC-Only Registers Summary" indicates the aliasing from EC-Only Registers to Runtime Registers. The “Aliased Access” column distinguishes the aliasing based on access type. See individual register descriptions for more details.

TABLE 9-7: EC-Only Registers SUMMARY

EC-Only Registers			Map of Aliasing into Runtime Registers			
EC Offset	EC-Only Registers **Register Name (Mnemonic)**	**Aliased Access**	**LPC Offset**	**EC Offset**	Runtime Registers **Register Name (Mnemonic)**	**Aliased Access**
108h	OS2EC Data EC-Register Byte 0	R	00h	00h	ACPI OS Data Register Byte 0	W
108h	OS2EC Data EC-Register Byte 0	R	04h	04h	ACPI OS COMMAND Register	W
109h	OS2EC Data EC-Register Byte 1	R	01h	01h	ACPI OS Data Register Byte 1	W
10Ah	OS2EC Data EC-Register Byte 2	R	02h	02h	ACPI OS Data Register Byte 2	W
10Bh	OS2EC Data EC-Register Byte 3	R	03h	03h	ACPI OS Data Register Byte 3	W
104h	STATUS EC-Register	W	04h	04h	STATUS OS-Register	W
105h	Byte Control EC-Register	R/W	05h	05h	Byte Control OS-Register	R
106h	Reserved	R			Reserved	R
107h	Reserved	R			Reserved	R
100h	EC2OS Data EC-Register Byte 0	W	00h	00h	ACPI OS Data Register Byte 0	R
101h	EC2OS Data EC-Register Byte 1	W	01h	01h	ACPI OS Data Register Byte 1	R
102h	EC2OS Data EC-Register Byte 2	W	02h	02h	ACPI OS Data Register Byte 2	R
103h	EC2OS Data EC-Register Byte 3	W	03h	03h	ACPI OS Data Register Byte 3	R

© 2012-2018 Microchip Technology Inc.

9.10 Runtime Registers

The registers listed in the Runtime Registers Register Summary table are for a single instance of the ACPI Embedded Controller Interface (ACPI-ECI). The addresses of each register listed in this table are defined as a relative offset to the host “Base Address” defined in Runtime Registers Address Range Table.

TABLE 9-8: Runtime Registers ADDRESS RANGE TABLE

Block Instance	Instance Number	Host	Address Space	Base Address
ACPI-EC	0	EC Note 9-1	24-bit internal address space	FF_0800h +100h
ACPI-EC	0	LPC Note 9-2	I/O	Programmed BAR
ACPI-EC	1	EC Note 9-1	24-bit internal address space	FF_0C00h +100h
ACPI-EC	1	LPC Note 9-2	I/O	Programmed BAR
ACPI-EC	2	EC Note 9-1	24-bit internal address space	FF_1000h +100h
ACPI-EC	2	LPC Note 9-2	I/O	Programmed BAR
ACPI-EC	3	EC Note 9-1	24-bit internal address space	FF_1400h +100h
ACPI-EC	3	LPC Note 9-2	I/O	Programmed BAR

Note 9-3 The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

TABLE 9-9: Runtime Registers REGISTER SUMMARY

Offset	Register Name (Mnemonic)
00h	ACPI OS Data Register Byte 0
01h	ACPI OS Data Register Byte 1
02h	ACPI OS Data Register Byte 2
03h	ACPI OS Data Register Byte 3
04h	ACPI OS COMMAND Register
04h	STATUS OS-Register
05h	Byte Control OS-Register
06h	Reserved
07h	Reserved

© 2012-2018 Microchip Technology Inc.

9.10.1 ACPI OS DATA REGISTER BYTE 0

OS_DATA_B0
; ALIAS=(OS2EC_DATA_ECREG_B0)
; ALIAS=EC2OS_DATA_ECREG_B0

This register is aliased; see ACPI-OS DATA BYTES[3:0] on page 177, OS2EC DATA BYTES[3:0] on page 184, and EC2OS DATA BYTES[3:0] on page 186 for detailed description of access rules.

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	ACPI-OS DATA BYTE 0 **OS_DAT_B0; ALIASED** This is byte 0 of the 32-bit ACPI-OS DATA BYTES[3:0].	R/W	0h	nSYS_RST

ACPI-OS DATA BYTES[3:0]

Writes by the ACPI_OS to the ACPI-OS DATA BYTES[3:0] are aliased to the OS2EC DATA BYTES[3:0] registers. Reads by the ACPI_OS from the ACPI-OS DATA BYTES[3:0] are aliased to the EC2OS DATA BYTES[3:0] registers.

All access to the ACPI-OS DATA BYTES[3:0] registers should be orderly: Least Significant Byte to Most Significant Byte when byte access is used.

Writes to any of the four ACPI-OS DATA BYTES[3:0] registers clear the CMD bit in the STATUS OS-Register (the state of the Four Byte Access bit in the Byte Control OS-Register has no impact.)

When the Four Byte Access bit in the Byte Control OS-Register is cleared to '0', the following access rules apply:

1. Writes to the ACPI OS Data Register Byte 0 register set the IBF bit in the STATUS OS-Register.
2. Reads from the ACPI OS Data Register Byte 0 register clear the OBF bit in the STATUS OS-Register.
3. All writes to ACPI-OS DATA BYTES[3:1] registers complete without error but the data are not registered.
4. All reads from ACPI-OS DATA BYTES[3:1] registers return 00h without error.
5. Access to ACPI-OS DATA BYTES[3:1] has no effect on the IBF & OBF bits in the STATUS OS-Register.

When the Four Byte Access bit in the Byte Control OS-Register is set to '1', the following access rules apply:

1. Writes to the ACPI OS Data Register Byte 3 register sets the IBF bit in the STATUS OS-Register.
2. Reads from the ACPI OS Data Register Byte 3 registers clears the OBF bit in the STATUS OS-Register.

9.10.2 ACPI OS DATA REGISTER BYTE 1

OS_DATA_B1
; ALIAS=OS2EC_DATA_ECREG_B1
; ALIAS=EC2OS_DATA_ECREG_B1

This register is aliased; see ACPI-OS DATA BYTES[3:0] on page 177, OS2EC DATA BYTES[3:0] on page 184, and EC2OS DATA BYTES[3:0] on page 186 for a detailed description of access rules.

Offset	01h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	ACPI-OS DATA BYTE 1 **OS_DAT_B1; ALIASED** This is byte 1 of the 32-bit ACPI-OS DATA BYTES[3:0].	R/W	0h	nSYS_RST

9.10.3 ACPI OS DATA REGISTER BYTE 2

OS_DATA_B2
; ALIAS=OS2EC_DATA_ECREG_B2
; ALIAS=EC2OS_DATA_ECREG_B2

This register is aliased; see ACPI-OS DATA BYTES[3:0] on page 177, OS2EC DATA BYTES[3:0] on page 184, and EC2OS DATA BYTES[3:0] on page 186 for a detailed description of access rules.

Offset	02h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	ACPI-OS DATA BYTE 2 **(OS_DAT_B2; ALIASED)** This is byte 2 of the 32-bit ACPI-OS DATA BYTES[3:0].	R/W	0h	nSYS_RST

9.10.4 ACPI OS DATA REGISTER BYTE 3

OS_DATA_B3
; ALIAS=OS2EC_DATA_ECREG_B3
; ALIAS=EC2OS_DATA_ECREG_B3

This register is aliased; see ACPI-OS DATA BYTES[3:0] on page 177, OS2EC DATA BYTES[3:0] on page 184, and EC2OS DATA BYTES[3:0] on page 186 for a detailed description of access rules.

Offset	03h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	ACPI-OS DATA BYTE 3 **(OS_DAT_B3; ALIASED)** This is byte 3 of the 32-bit ACPI-OS DATA BYTES[3:0].	R/W	0h	nSYS_RST

9.10.5 ACPI OS COMMAND REGISTER

COMMAND_OSREG
; ALIAS=(OS2EC_DATA_ECREG_B0)

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	ACPI-OS COMMAND **(OS_CMD; ALIASED)** Writes to the this register are aliased in the OS2EC Data EC-Register Byte 0. Writes to the this register also set the CMD and IBF bits in the STATUS OS-Register	W	0h	nSYS_RST

© 2012-2018 Microchip Technology Inc.

9.10.6 STATUS OS-REGISTER

STATUS_OSREG
; ALIAS=(STATUS_ECREG)

This read-only register is aliased to the STATUS EC-Register on page 188. the STATUS EC-Register on page 188 has read/write access.

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	User Defined **(UD-Bit7_OS_OS; ALIASED)**	R	0b	nSYS_RST
6	SMI_EVT **(SMI_EVT_OS; ALIASED)** This bit is set when an SMI event is pending; i.e., the ACPI_EC is requesting an SMI query; This bit is cleared when no SMI events are pending. This bit is an ACPI_EC-maintained software flag that is set when the ACPI_EC has detected an internal event that requires system management interrupt handler attention. The ACPI_EC sets this bit before generating an SMI. **Note:** The usage model from the ACPI specification requires both SMI's and SCI's. The ACPI_OS SMI & SCI interrupts are not implemented in the ACPI Embedded Controller Interface (ACPI-ECI). The SMI_EVT and SCI_EVT bits in the Section 9.10.6, "STATUS OS-Register," on page 179 are software flags and this block does not initiate SMI or SCI events.	R	0b	nSYS_RST
5	SCI_EVT **(SCI_EVT_OS; ALIASED)** This bit is set by software when an SCI event is pending; i.e., the ACPI_EC is requesting an SCI query; SCI Event flag is clear when no SCI events are pending. This bit is an ACPI_EC-maintained software flag that is set when the embedded controller has detected an internal event that requires operating system attention. The ACPI_EC sets this bit before generating an SCI to the OS. **Note:** The usage model from the ACPI specification requires both SMI's and SCI's. The ACPI_OS SMI & SCI interrupts are not implemented in the ACPI Embedded Controller Interface (ACPI-ECI). The SMI_EVT and SCI_EVT bits in the Section 9.10.6, "STATUS OS-Register," on page 179 are software flags and this block does not initiate SMI or SCI events.	R	0b	nSYS_RST
4	BURST **(BURST_OS; ALIASED)** The BURST bit is set when the ACPI_EC is in Burst Mode for polled command processing; the BURST bit is cleared when the ACPI_EC is in Normal mode for interrupt-driven command processing. The BURST bit is an ACPI_EC-maintained software flag that indicates the embedded controller has received the Burst Enable command from the host, has halted normal processing, and is waiting for a series of commands to be sent from the host. Burst Mode allows the OS or system management handler to quickly read and write several bytes of data at a time without the overhead of SCIs between commands. The BURST bit is maintained by ACPI_EC software, only.	R	0b	nSYS_RST

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
3	CMD **(CMD_OS; ALIASED)** This bit is set when the OS2EC Data EC-Register Byte 0 contains a command byte written into ACPI OS COMMAND Register; this bit is cleared when the OS2EC DATA BYTES[3:0] contains a data byte written into the ACPI-OS DATA BYTES[3:0]. This bit is hardware controlled: • ACPI_OS writes to any of the four ACPI-OS DATA BYTES[3:0] bytes clears this bit • ACPI_OS writes to the ACPI OS COMMAND Register sets this bit. **Note:** This bit allows the embedded controller to differentiate the start of a command sequence from a data byte write operation.	R	0b	nSYS_RST
2	User Defined **(UD-Bit2_OS; ALIASED)**	R	0b	nSYS_RST

© 2012-2018 Microchip Technology Inc.

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
1	IBF **(IBF_OS; ALIASED)** The Input Buffer Full bit is set to indicate that a the ACPI_OS has written a command or data to the ACPI_EC and that data is ready. This bit is automatically cleared when data has been read by the ACPI_EC. **Note:** The setting and clearing of this IBF varies depending on the setting of the following bits: CMD bit in this register and Four Byte Access bit in the Byte Control OS-Register. Three scenarios follow: 1. The IBF is set when the ACPI_OS writes to the ACPI OS COMMAND Register. This same write autonomously sets the CMD bit in this register. The IBF is cleared if the CMD bit in this register is set and the ACPI_EC reads from OS2EC Data EC-Register Byte 0. **Note:** When the CMD bit in this register is set the Four Byte Access bit in the Byte Control OS-Register has no impact on the IBF bit behavior. 2. A write by the ACPI_OS to the ACPI OS Data Register Byte 0 sets the IBF bit if the Four Byte Access bit in the Byte Control OS-Register is in the cleared to '0' state prior to this write. This same write autonomously clears the CMD bit in this register. A read of the OS2EC Data EC-Register Byte 0 clears the IBF bit if the Four Byte Access bit in the Byte Control OS-Register is in the cleared to '0' state prior to this read. 3. A write by the ACPI_OS to the ACPI OS Data Register Byte 3 sets the IBF bit if the Four Byte Access bit in the Byte Control OS-Register is in the set to '1' state prior to this write. This same write autonomously clears the CMD bit in this register. A read of the OS2EC Data EC-Register Byte 3 clears the IBF bit if the Four Byte Access bit in the Byte Control OS-Register is in the set to '1' state prior to this read. An EC_IBF interrupt signals the ACPI_EC that there is data available. The ACPI Specification usage model is as follows: 1. The ACPI_EC reads the STATUS EC-Register and sees the IBF flag set, 2. The ACPI_EC reads all the data available in the OS2EC DATA BYTES[3:0]. This causes the IBF bit to be automatically cleared by hardware. 3. The ACPI_EC must then generate a software interrupt (See Note: on page 171) to alert the ACPI_OS that the data has been read and that the host is free to write more data to the ACPI_EC as needed.	R	0h	nSYS_RST

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
0	OBF **(OBF_OS; ALIASED)** The Output Buffer Full bit is set to indicate that the ACPI_EC has written data to the ACPI_OS and that data is ready. This bit is automatically cleared when all data have been read by the ACPI_OS. **Note:** The setting and clearing of this OBF varies depending on the setting Four Byte Access bit in the Byte Control OS-Register. Two scenarios follow: 1. The OBF bit is set if the Four Byte Access bit in the Byte Control OS-Register is '0' when the ACPI_EC writes to the EC2OS Data EC-Register Byte 0. The OBF is cleared if the Four Byte Access bit in the Byte Control OS-Register is cleared to '0' when the ACPI_OS reads from the ACPI OS Data Register Byte 0. 2. The OBF is set if the Four Byte Access bit in the Byte Control OS-Register is set to '1' when the ACPI_EC writes to the EC2OS Data EC-Register Byte 3. The OBF is cleared if the Four Byte Access bit in the Byte Control OS-Register is set to '1' when the ACPI_OS reads from the ACPI OS Data Register Byte 3. The ACPI Specification usage model is as follows: 1. The ACPI_EC must generate a software interrupt (See Note: on page 171) to alert the ACPI_OS that the data is available. 2. The ACPI_OS reads the STATUS OS-Register and sees the OBF flag set, the ACPI_OS reads all the data available in the ACPI-OS DATA BYTES[3:0]. 3. The ACPI_OS reads all the data available in the ACPI-OS DATA BYTES[3:0]. This causes the OBF bit to be automatically cleared by hardware and the associated EC_OBF interrupt to be asserted.	R	0h	nSYS_RST

© 2012-2018 Microchip Technology Inc.

9.10.7 BYTE CONTROL OS-REGISTER

BYTE_CNL_OSREG
; ALIAS=BYTE_CNL_ECREG

This register is aliased to the Byte Control EC-Register on page 189. No behavioral differences occur due to address aliasing.

Offset	05			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:1	RESERVED	RES	-	-
0	Four Byte Access **(4BYTE_ACCESS_EN_OS;ALIASED)** When this bit is set to '1', the ACPI Embedded Controller Interface (ACPI-ECI) accesses four bytes through the ACPI-OS DATA BYTES[3:0]. When this bit is cleared to '0', the ACPI Embedded Controller Interface (ACPI-ECI) accesses one byte through the ACPI OS Data Register Byte 0. The corresponds to Legacy Mode described in Section 9.8, "Description," on page 172. **Note:** This bit effects the behavior of the IBF & OBF bits in the STATUS OS-Register. **Note:** See ACPI-OS DATA BYTES[3:0] on page 177, OS2EC DATA BYTES[3:0] on page 184, and EC2OS DATA BYTES[3:0] on page 186 for detailed description of access rules.	R	0b	nSYS_RST

APPLICATION NOTE: The ACPI_OS access Base Address Register (BAR) should be configured to match the access width selected by the Four Byte Access bit in the Byte Control OS-Register. This BAR in not described in this chapter.

9.11 EC-Only Registers

The registers listed in the EC-Only Registers Register Summary table are for a single instance of the ACPI Embedded Controller Interface (ACPI-ECI). The addresses of each register listed in this table are defined as a relative offset to the host "Base Address" defined in EC-Only Registers Address Range Table.

TABLE 9-10: EC-Only Registers ADDRESS RANGE TABLE

Block Instance	Instance Number	Host	Address Space	Base Address
ACPI-EC	0	EC Note 9-1	24-bit internal address space	ACPI EC Channel 0 LDN 2h, Base address FF_0800h
ACPI-EC	1	EC Note 9-1	24-bit internal address space	ACPI EC Channel 1 LDN 3h, Base address FF_0C00h
ACPI-EC	2	EC Note 9-1	24-bit internal address space	ACPI EC Channel 2 LDN 4h, Base address FF_1000h
ACPI-EC	3	EC Note 9-1	24-bit internal address space	ACPI EC Channel 3 LDN 5h, Base address FF_1400h

TABLE 9-11: EC-Only Registers REGISTER SUMMARY

Offset	Register Name (Mnemonic)
100h	EC2OS Data EC-Register Byte 0
101h	EC2OS Data EC-Register Byte 1
102h	EC2OS Data EC-Register Byte 2
103h	EC2OS Data EC-Register Byte 3
104h	STATUS EC-Register
105h	Byte Control EC-Register
106h	Reserved
107h	Reserved
108h	OS2EC Data EC-Register Byte 0
109h	OS2EC Data EC-Register Byte 1
10Ah	OS2EC Data EC-Register Byte 2
10Bh	OS2EC Data EC-Register Byte 3

9.11.1 OS2EC DATA EC-REGISTER BYTE 0

(OS2EC_DATA_ECREG_B0)
; ALIAS=OS_DATA_B0
; ALIAS=COMMAND_OSREG

This register is aliased; see ACPI-OS DATA BYTES[3:0] on page 177, OS2EC DATA BYTES[3:0] on page 184, and EC2OS DATA BYTES[3:0] on page 186 for detailed description of access rules.

Offset	108h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	OS TO EC DATA BYTE 0 **(OS2EC_DAT_B0)** This is byte 0 of the 32-bit OS2EC DATA BYTES[3:0].	R/W	0h	nSYS_RST

OS2EC DATA BYTES[3:0]

When the CMD bit in the STATUS OS-Register is cleared to '0', reads by the ACPI_EC from the OS2EC DATA BYTES[3:0] are aliased to the ACPI-OS DATA BYTES[3:0].

All access to the OS2EC DATA BYTES[3:0] registers should be orderly: Least Significant Byte to Most Significant Byte when byte access is used.

When the Four Byte Access bit in the Byte Control OS-Register is cleared to '0', the following access rules apply:

1. Writes to the OS2EC DATA BYTES[3:0] have no effect on the OBF bit in the STATUS OS-Register.
2. Reads from the OS2EC Data EC-Register Byte 0 clears the IBF bit in the STATUS OS-Register.
3. All reads from OS2EC DATA BYTES[3:1] return 00h without error.
4. Access to OS2EC DATA BYTES[3:1 has no effect on the IBF & OBF bits in the STATUS OS-Register.

When the Four Byte Access bit in the Byte Control OS-Register is set to '1', the following access rules apply:

1. Writes to the OS2EC DATA BYTES[3:0] have no effect on the OBF bit in the STATUS OS-Register.
2. Reads from the OS2EC Data EC-Register Byte 3 clears the IBF bit in the STATUS OS-Register.

© 2012-2018 Microchip Technology Inc.

9.11.2 OS2EC DATA EC-REGISTER BYTE 1

OS2EC_DATA_ECREG_B1
; ALIAS=OS_DATA_B1

This register is aliased; see ACPI-OS DATA BYTES[3:0] on page 177, OS2EC DATA BYTES[3:0] on page 184, and EC2OS DATA BYTES[3:0] on page 186 for detailed description of access rules.

Offset	109h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	OS2EC Data BYTE 1 **(OS2EC_DAT_B1)** This is byte 1 of the 32-bit OS2EC DATA BYTES[3:0].	R/W	0h	nSYS_RST

9.11.3 OS2EC DATA EC-REGISTER BYTE 2

OS2EC_DATA_ECREG_B2
; ALIAS=OS_DATA_B2

This register is aliased; see ACPI-OS DATA BYTES[3:0] on page 177, OS2EC DATA BYTES[3:0] on page 184, and EC2OS DATA BYTES[3:0] on page 186 for detailed description of access rules.

Offset	10Ah			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	OS2EC Data BYTE 2 **(OS2EC_DAT_B2)** This is byte 2 of the 32-bit OS2EC DATA BYTES[3:0].	R/W	0h	nSYS_RST

9.11.4 OS2EC DATA EC-REGISTER BYTE 3

OS2EC_DATA_ECREG_B3
; ALIAS=OS_DATA_B3

This register is aliased; see ACPI-OS DATA BYTES[3:0] on page 177, OS2EC DATA BYTES[3:0] on page 184, and EC2OS DATA BYTES[3:0] on page 186 for detailed description of access rules.

Offset	10Bh			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	OS2EC Data BYTE 3 **(OS2EC_DAT_B3)** This is byte 3 of the 32-bit OS2EC DATA BYTES[3:0].	R/W	0h	nSYS_RST

9.11.5 EC2OS DATA EC-REGISTER BYTE 0

EC2OS_DATA_ECREG_B0
; ALIAS=OS_DATA_B0

This register is aliased; see ACPI-OS DATA BYTES[3:0] on page 177, OS2EC DATA BYTES[3:0] on page 184, and EC2OS DATA BYTES[3:0] on page 186 for detailed description of access rules.

Offset	100h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	EC2OS DATA BYTE 0 **(EC2OS_DAT_B0)** This is byte 0 of the 32-bit EC2OS DATA BYTES[3:0].	R/W	0h	nSYS_RST

EC2OS DATA BYTES[3:0]

Writes by the ACPI_EC to the EC2OS DATA BYTES[3:0] are aliased to the ACPI-OS DATA BYTES[3:0]

All access to the EC2OS DATA BYTES[3:0] registers should be orderly: Least Significant Byte to Most Significant Byte when byte access is used.

When the Four Byte Access bit in the Byte Control OS-Register is cleared to '0', the following access rules apply:

1. Writes to the EC2OS Data EC-Register Byte 0 set the OBF bit in the STATUS OS-Register.
2. Reads from the EC2OS DATA BYTES[3:0] have no effect on the IBF bit in the STATUS OS-Register.
3. All reads from EC2OS DATA BYTES[3:1] return 00h without error.
4. All writes to EC2OS DATA BYTES[3:1] complete without error but the data are not registered.
5. Access to EC2OS DATA BYTES[3:1] have no effect on the IBF & OBF bits in the STATUS OS-Register.

When the Four Byte Access bit in the Byte Control OS-Register is set to '1', the following access rules apply:

1. Writes to the EC2OS Data EC-Register Byte 3 set the OBF bit in the STATUS OS-Register.
2. Reads from the EC2OS DATA BYTES[3:0] have no effect on the IBF bit in the STATUS OS-Register.

9.11.6 EC2OS DATA EC-REGISTER BYTE 1

EC2OS_DATA_ECREG_B1
; ALIAS=OS_DATA_B1

This register is aliased; see ACPI-OS DATA BYTES[3:0] on page 177, OS2EC DATA BYTES[3:0] on page 184, and EC2OS DATA BYTES[3:0] on page 186 for detailed description of access rules.

Offset	101h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	EC2OS DATA BYTE 1 **(EC2OS_DAT_B1)** This is byte 1 of the 32-bit EC2OS DATA BYTES[3:0].	R/W	0h	nSYS_RST

© 2012-2018 Microchip Technology Inc.

9.11.7 EC2OS DATA EC-REGISTER BYTE 2

EC2OS_DATA_ECREG_B2
; ALIAS=OS_DATA_B2

This register is aliased; see ACPI-OS DATA BYTES[3:0] on page 177, OS2EC DATA BYTES[3:0] on page 184, and EC2OS DATA BYTES[3:0] on page 186 for detailed description of access rules.

Offset	102h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	EC2OS DATA BYTE 2 **(EC2OS_DAT_B2)** This is byte 2 of the 32-bit EC2OS DATA BYTES[3:0].	R/W	0h	nSYS_RST

9.11.8 EC2OS DATA EC-REGISTER BYTE 3

EC2OS_DATA_ECREG_B3
; ALIAS=OS_DATA_B3

This register is aliased; see ACPI-OS DATA BYTES[3:0] on page 177, OS2EC DATA BYTES[3:0] on page 184, and EC2OS DATA BYTES[3:0] on page 186 for detailed description of access rules.

Offset	103h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	EC2OS DATA BYTE 3 **(EC2OS_DAT_B0)** This is byte 3 of the 32-bit EC2OS DATA BYTES[3:0].	R/W	0h	nSYS_RST

9.11.9 STATUS EC-REGISTER

(STATUS_ECREG)
; ALIAS=STATUS_OSREG

This register is aliased to the STATUS OS-Register on page 179. The STATUS OS-Register is a read only version of this register.

Offset	104h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	User Defined **(UD-Bit7_EC; ALIASED)**	R/W	0b	nSYS_RST
6	SMI_EVT **(SMI_EVT_EC; ALIASED)** See SMI_EVT bit in STATUS OS-Register on page 179 for bit description.	R/W	0b	nSYS_RST
5	SCI_EVT **(SCI_EVT_EC; ALIASED)** See SCI_EVT bit in STATUS OS-Register on page 179 for bit description.	R/W	0b	nSYS_RST
4	BURST **(BURST_EC; ALIASED)** See BURST bit in STATUS OS-Register on page 179 for bit description.	R/W	0b	nSYS_RST
3	CMD **(CMD_EC; ALIASED)** See CMD bit in STATUS OS-Register on page 179 for bit description.	R	0b	nSYS_RST
2	User Defined **(UD-Bit2_EC; ALIASED)**	R/W	0b	nSYS_RST
1	IBF **(IBF_EC;ALIASED)** See IBF bit in STATUS OS-Register on page 179 for bit description.	R	0h	nSYS_RST
0	OBF **(OBF_EC; ALIASED)** See OBF bit in STATUS OS-Register on page 179 for bit description.	R	0h	nSYS_RST

APPLICATION NOTE: The IBF and OBFbits are not cleared (‘0’) by hardware when VCC_PWRGD is asserted or when the LPC interface powers down; for example, following system changes state S3->S0, S5->S0, G3-> S0. For further information on how these bits are cleared, refer to IBF and OBF bit descriptions in STATUS OS-Register on page 179.

© 2012-2018 Microchip Technology Inc.

9.11.10 BYTE CONTROL EC-REGISTER

BYTE_CNL_ECREG
; ALIAS=BYTE_CNL_OSREG

This register is aliased to the Byte Control OS-Register on page 183. The Byte Control OS-Register is a read only version of this register.

Offset	105h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:1	RESERVED	RES	-	-
0	Four Byte Access **(4BYTE_ACCESS_EN_OS;ALIASED)** See Four Byte Access bit in Byte Control OS-Register on page 183 for bit description. **Note:**	R/W	0b	nSYS_RST

10.0 8042 EMULATED KEYBOARD CONTROLLER

10.1 General Description

The MEC1632 keyboard controller uses the EC to produce a superset of the features provided by the industry-standard 8042 keyboard controller. The 8042 Emulated Keyboard Controller is a Host/EC Message Interface with hardware assists to emulate 8042 behavior and provide Legacy GATEA20 support.

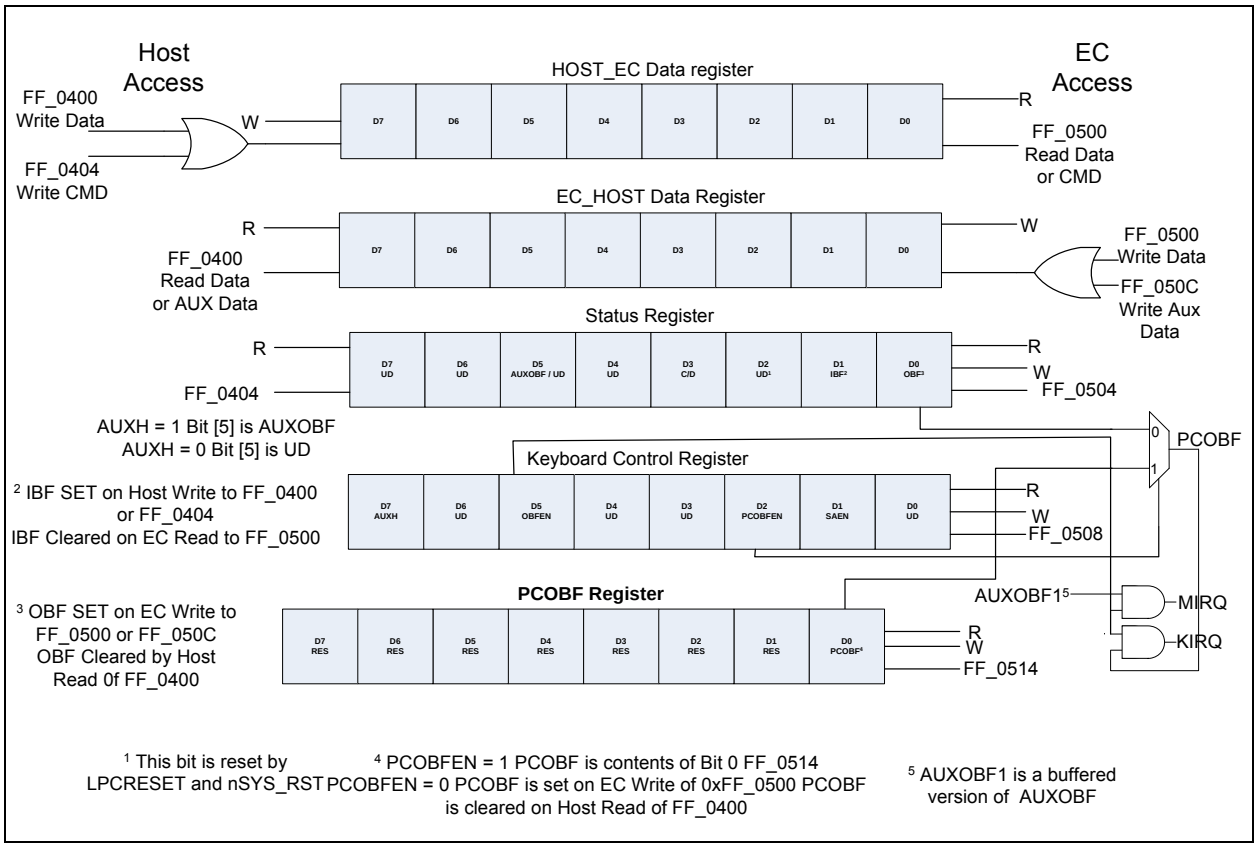
Note:	there is no VCC emulation in hardware for this interface.

10.1.1 FEATURES

- Legacy Keyboard Support
- Emulated 8042 Operation
- Port 92 Legacy A20M Support

10.1.2 BLOCK DIAGRAM

FIGURE 10-1: BLOCK DIAGRAM OF 8042 Emulated Keyboard Controller



© 2012-2018 Microchip Technology Inc.

10.1.3 BLOCK DIAGRAM SIGNAL LIST

TABLE 10-1: 8042 Emulated Keyboard Controller SIGNAL LIST

Signal Name	Direction	Description
EC_IBF	Output	Interrupt generated by the host writing either data or command to the data register
EC_OBF	Output	Interrupt generated by the host reading either data or aux data from the data register
KIRQ	Output	Routed to the Host SIRQ
MIRQ	Output	Routed to the Host SIRQ
KBRST	Output	Routed to Pin Function

10.2 Power, Clocks and Reset

10.2.1 POWER DOMAIN

This block is powered by VTR.

See Section 7.1.1, "Power Configuration," on page 95 for details on power domains.

10.2.2 CLOCKS

This block has four clock inputs: LPC Bus Clock, 1MHZ (MCLK_DIV20_EN_HST) clock source. LPC Bus Clock is used to the accessible and clock the registers in this block. The 1MHz Host Clock Domain is used to clock the counter in the CPU_RESET circuitry.

See Section 7.4, "Clock Generator," on page 98 for details on clocks.

10.3 Power On Reset

This block is reset on a nSYS_RST and nSIO_RESET. On a reset, all Register are reset to 00h and the state machines are set to idle.

See Section 7.6, "Reset Interface," on page 124 for details on reset.

10.4 Interrupts

The 8042 Emulated Keyboard Controller can generate a KBD_OBF interrupt when the OBF bit in the Keyboard Status Read Register is cleared (falling edge sensitive) and a KBD_IBF interrupt when the IBF bit in the Keyboard Status Read Register is set (active high, level sensitive). These interrupt sources are routed to the GIRQ19 Source Register.

10.4.1 8042 EMULATED KEYBOARD CONTROLLER (LDN 1H) SIRQ ROUTING

The 8042 Emulated Keyboard Controller can generate two SIRQ events for the EC-to-HOST EC events:KIRQ & MIRQ. For the KIRQ interrupt the Interrupt Configuration Register, SELECT on page 72 is cleared to ‘0’ and for MIRQ the Interrupt Configuration Register, SELECT is set to’1’.

See Logical Device Configuration, Section 5.7, "SERIRQ Interrupts," on page 71.

10.5 Instance Description

There are two blocks defined in this chapter: 8042 MSG Interface and the Port92-Legacy. The MEC1632 has one instance of each block.

10.6 Registers

The 8042 Emulated Keyboard Controller has a 8042 MSG Interface block which has its own Logical Device Number, and Base Address as indicated in Table 10-2. The Host LPC I/O addresses for the 8042 MSG Interface is selected via Base Address Register (see Section 5.6.2, "Base Address Registers," on page 66). LPC access to configuration registers is through Host Access Configuration Port (see Section 5.5.1, "Host Access Port," on page 65).

The 8042 Emulated Keyboard Controller also has a Port92-Legacy block which has a separate Logical Device Number and Base Address Register as indicated in Table 10-2. The Base Address Register for the Port92-Legacy has only one writable bit, the Valid Bit, since the only I/O accessible Register has a fixed address.

Table 10-3 is a register summary for the 8042 MSG Interface block and Table 10-14 on page 198 is a register summary for the Port92-Legacy block.

TABLE 10-2: 8042 Emulated Keyboard Controller BASE ADDRESS TABLE

8042 Emulated Keyboard Controller **Blocks**	**LDN from** (Table 4-2 on page 59)	**AHB Base Address**
8042 MSG Interface	1h	FF_0400h
Port92-Legacy	8h	FF_2000h

Note 10-1 The Host LPC I/O addresses for this instance is selected via a Base Address Register (see Section 5.6.2, "Base Address Registers," on page 66). LPC access to configuration registers is through the Host Access Configuration Port (see Section 5.5.1, "Host Access Port," on page 65).

Table 10-3 is a register summary for the 8042 MSG Interface block. The LPC I/O address for each Run-Time Register is described below as an offset from its Base Address Register. Each EC address is indicated as an SPB Offset from its AHB base address. Each Configuration register access through the Host Access Port is via its LDN indicated in Table 10-2 on page 192 and its Host Access Port index which is described as “Host Config Index” in the tables below.

TABLE 10-3: 8042 MSG Interface REGISTER SUMMARY

	Host I/O Access			**CMD** Note 10-2	**EC Interface**			
Register Name	**Host I/O Index**	**SPB Offset**	**Host Type**		**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
HOST_EC Data/CMD Register	00h	00h	W	0	100h	0	R	
	04h	04h		1		0	R	
EC_HOST Data/AUX Data Register	00h	00h	R	0	100h	0	W	
				0	10Ch	0		
Keyboard Status Read Register	04h	04h	R	0	104h	3	R/W	
PCOBF Register	-	-	-	0	114h	0	R/W	
Keyboard Control Register	-	-	-	0	108h	0	R/W	
	Host Access				**EC Interface**			
Register Name	**Host Config. Index**	**SPB Offset**	**Host Type**	**N/A**	**EC Offset**	**Byte Lane**	**EC Type**	
Activate Register	30h	330h	R/W		330h	0	R/W	

Note 10-2 CMD is bit D3 in the Keyboard Status Read Register.

Note 10-3 All Registers listed in Table 10-3 are powered by VTR and reset by nSYS_RST.

© 2012-2018 Microchip Technology Inc.

10.7 8042 MSG Interface Configuration Registers

10.7.1 ACTIVATE REGISTER

TABLE 10-4: 8042 MSG Interface ACTIVATE REGISTER

HOST CONFIG INDEX	330h					8-bit	**HOST SIZE**	
EC OFFSET	NA					8-bit	**EC SIZE**	
POWER	VTR					00h	nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R							R/W
EC TYPE	-	-	-	-	-	-	-	-
BIT NAME	Reserved							Activate

ACTIVATE

0=1MHz Clock to the CPU_REST block is disabled, ONLY if the PGEN is not active

1=1MHz Clock to the CPU_REST block is enabled

10.8 8042 MSG Interface Runtime Registers

10.8.1 HOST_EC DATA / CMD REGISTER

This is an 8-bit HOST write-only register. When written with Data, the C/D status bit of the status register is cleared to zero and the IBF bit is set.

When written with a command, the C/D status bit of the status register is set to one and the IBF bit is set to a 1.

TABLE 10-5: HOST_EC DATA/CMD REGISTER

HOST I/O INDEX	00h Data 04h Command					8-bit	**HOST SIZE**	
EC OFFSET	100h					8-bit	**EC SIZE**	
POWER	VTR					00h	nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	W	W	W	W	W	W	W	W
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Data							

10.8.2 EC_HOST DATA / AUX DATA REGISTER

This is an 8-bit read-only register. When read, by the HOST, the PCOBF and/or AUXOBF interrupts are cleared and the OBF flag in the status register is cleared.

TABLE 10-6: EC_HOST DATA/AUX DATA REGISTER

HOST I/O INDEX	00h			8-bit			**HOST SIZE**	
EC OFFSET	100h Data or 10Ch Aux Data			8-bit			**EC SIZE**	
POWER	VTR			00h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R	R	R	R	R	R
EC TYPE	W	W	W	W	W	W	W	W
BIT NAME	Data							

10.8.3 KEYBOARD STATUS READ

This is an 8 bit read only register. Refer to the description of the Status Register (EC OFFSET 104h) for more information.

TABLE 10-7: KEYBOARD STATUS READ REGISTER

HOST I/O INDEX	4h			8-bit			**HOST SIZE**	
EC OFFSET	104h			8-bit			**EC SIZE**	
POWER	VTR			00h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R	R	R	R	R	R
EC TYPE	R/W	R/W	R/W	R/W	R	R/W	R	R
BIT NAME	UD	UD	AUXOBF	UD	C/D	UD[1]	IBF	OBF

This register is read-only for the Host and read/write by the EC. The EC cannot write to bits 0, 1, or 3 of the Status register.

APPLICATION NOTE: The IBF and OBFbits are not cleared (‘0’) by hardware when VCC_PWRGD is asserted or when the LPC interface powers down; for example, following system changes state S3->S0, S5->S0, G3-> S0. To clear the IBF bit in firmware, read offset 100h; to clear the OBF bit in firmware, read offset 000h.

UD[1]

Read/Write by EC. These bits are user-definable. This bit is reset to ‘0’ when LRESET# pin signal function is asserted.

UD

Read/Write by EC. These bits are user-definable.

C/D

Command Data - This bit specifies whether the input data register contains data or a command (“0” = data, “1” = command). During a host command write operation, this bit is set to “1”, during a host data write operation, this bit is set to “0”.

© 2012-2018 Microchip Technology Inc.

IBF

Input Buffer Full - This flag is set to “1” whenever the host system writes data or a command into the HOST_EC Data/CMD Register. Setting this flag activates the EC's EC_IBF interrupt if enabled. When the EC reads the HOST_EC Data / CMD Register, this bit is automatically reset and the interrupt is cleared.

OBF

The Output Buffer Full (OBF) bit is set when the EC writes a byte of Data or AUX Data into the EC_HOST Data / AUX Data Register. When the host reads the data, the OBF bit is automatically cleared by hardware and a EC_OBF interrupt is generated.

AUXOBF

Auxiliary Output Buffer Full - This flag is set to “1” whenever the EC writes AUX Data into EC_HOST Data / AUX Data Register. This flag is reset to “0” whenever the EC writes into the Data into EC_HOST Data / AUX Data Register.

TABLE 10-8: PCOBF REGISTER

HOST OFFSET	NA			8-bit			**HOST SIZE**	
EC OFFSET	114h			8-bit			**EC SIZE**	
POWER	VTR			00h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R							R/W
BIT NAME	Reserved							PCOBF

Note 10-4 Refer to the PCOBF Description for information on this register. This is a “1” bit register (bits 1-7=0 on read).

10.9 EC-to-Host Keyboard Communication

The EC can write to the EC_HOST Data/AUX Data Register via EC OFFSET 100h or EC OFFSET 10Ch (Aux Host Data) respectively. A write to either of these addresses automatically sets bit 0 (OBF) in the Status register. A write to EC OFFSET 100h also sets PCOBF. A write to EC OFFSET 10Ch also sets AUXOBF. See Table 10-9.

TABLE 10-9: HOST-INTERFACE FLAGS

EC Address	Flag
EC OFFSET 100h (R/W)	PCOBF (KIRQ) output signal goes high
EC OFFSET 10Ch (W)	AUXOBF (MIRQ) output signal goes high

The HOST_EC Data / CMD Register and EC_HOST Data / AUX Data Register are each 8 bits wide. A write to this 8 bit register by the EC will load the EC_HOST Data / AUX Data Register, set the OBF flag and set the PCOBF output if enabled.

Note 10-5 Refer to the PCOBF and Keyboard Status Read Register descriptions for more information.

10.9.1 PCOBF DESCRIPTION

(The following description assumes that OBFEN = 1 in Keyboard Control Register on page 197); PCOBF is gated onto KIRQ. The KIRQ signal is a system interrupt which signifies that the EC has written to the KBD Data Read register via EC OFFSET100h. On power-up, PCOBF is reset to 0. PCOBF will normally reflect the status of writes to EC OFFSET 100h, if PCOBFEN (bit 2 of Configuration register “0”) = “0”. (KIRQ is normally selected as IRQ1 for keyboard support). PCOBF is cleared by hardware on a HOST read of the EC_HOST Data / AUX Data Register.

© 2012-2018 Microchip Technology Inc.

Additional flexibility has been added which allows firmware to directly control the PCOBF output signal, independent of data transfers to the host-interface data output register. This feature allows the MEC1632 to be operated via the host “polled” mode. This firmware control is active when PCOBFEN = 1 and firmware can then bring PCOBF high by writing a “1” to the LSB of the 1 bit data register, PCOBF, at EC OFFSET 114h. The firmware must also clear this bit by writing a “0” to the LSB of the 1 bit data register at EC OFFSET 114h.

The PCOBF register is also readable; bits 1-7 will return a “0” on the read back. The value read back on bit 0 of the register always reflects the present value of the PCOBF output. If PCOBFEN = 1, then this value reflects the output of the firmware latch at EC OFFSET 114h. If PCOBFEN = 0, then the value read back reflects the in-process status of write cycles to EC OFFSET 100h (i.e., if the value read back is high, the host interface output data register has just been written to). If OBFEN=0, then KIRQ is driven inactive (low).

10.9.2 AUXOBF1 DESCRIPTION

(The following description assumes that OBFEN = 1 in Keyboard Control Register on page 197); This bit is multiplexed onto MIRQ. The AUXOBF1/MIRQ signal is a system interrupt which signifies that the EC has written to the EC_HOST Data / AUX Data Register.

On power-up, after nSYS_RST, AUXOBF1 is reset to 0. AUXOBF1 will normally reflects the status of writes to EC OFFSET 10Ch. (MIRQ is normally selected as IRQ12 for mouse support). AUXOBF1 is cleared by hardware on a read of the Host Data Register. If OBFEN=0, then KIRQ is driven inactive (low).

TABLE 10-10: STATUS AND INTERRUPT BEHAVIOR OF WRITING TO OUTPUT DATA REGISTER

	Host I/F Status Register Bits			
Write to Register	AUXOBF (D5)	OBF (D0)	OBFEN=0	OBFEN=1
EC OFFSET 100h	0	1	KIRQ=0	KIRQ=1
EC OFFSET 10Ch	1	1	MIRQ=0	MIRQ=1

TABLE 10-11: OBFEN AND PCOBFEN EFFECTS ON KIRQ

OBFEN	PCOBFEN	
0	X	KIRQ is inactive and driven low
1	0	KIRQ = PCOBF@EC OFFSET 100h
1	1	KIRQ = PCOBF@EC OFFSET 114h

TABLE 10-12: OBFEN AND AUXH EFFECTS ON MIRQ

OBFEN	AUXH	
0	X	MIRQ is inactive and driven low
1	0	MIRQ = PCOBF@EC OFFSET 10Ch; Status Register D5 = User Defined
1	1	MIRQ = PCOBF@EC OFFSET 10Ch; Status Register D5 = Hardware Controlled

© 2012-2018 Microchip Technology Inc.

10.9.2.1 Keyboard Control

TABLE 10-13: KEYBOARD CONTROL REGISTER

HOST OFFSET	NA			8-bit			**HOST SIZE**	
EC OFFSET	108h			8-bit			**EC SIZE**	
POWER	VTR			00h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	AUXH	UD	OBFEN	UD	UD	PCOB-FEN	SAEN	UD

UD

User defined bit

AUXH

Aux in Hardware; when high, AUXOBF of the status register is set in hardware by a write to EC OFFSET 10Ch. When low, AUXOBF of the status register is a user defined bit (UD) and R/W.

OBFEN

When set,PCOBF is gated onto KIRQ and AUXOBF1 is gated onto MIRQ. When low, KIRQ and MIRQ are driven low. Software should not change this bit when OBF of the status register is equal to 1.

PCOBFEN

When high, PCOBF reflects whatever value was written to the PCOBF firmware latch assigned to 114h. When low, PCOBF reflects the status of writes to EC OFFSET 100h (the output data register).

SAEN

Software-assist enable. When set to “1,” SAEN allows control of the GATEA20 signal via firmware. If SAEN is reset to ‘0’, GATEA20 corresponds to either the last host-initiated control of GATEA20 or the firmware write to EC OFFSETs 108h or 10Ch.

10.10 Legacy Support

10.11 Port92-Legacy Registers

The Table 10-14 is a register summary for the Port92-Legacy block. The LPC I/O address of the PORT92 Register is fixed. Each EC address is indicated as an SPB Offset from its AHB base address listed for the Port92-Legacy block in Table 10-2, “8042 Emulated Keyboard Controller Base Address Table,” on page 192. Each Configuration register access through the Host Access Port address via its LDN indicated in Table 10-2 on page 192 and its Host Access Port index which is described as “Host Config Index” in the tables below.

TABLE 10-14: Port92-Legacy SUPPORT REGISTER SUMMARY

	Host I/O Access			EC Interface			
Register Name	**Host I/O Address**	**SPB Offset**	**Host Type**	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
GATEA20 Control Register	-	-	-	100h	0	R/W	
SETGA20L Register	-	-	-	108h	0	W	
RSTGA20L Register	-	-	-	10Ch	0	W	
PORT92 Register	92h	000h	R/W	000h	0	R/W	
	Host Access			**EC Interface**			
Register Name	**Host Config. Index**	**SPB Offset**	**Host Type**	**SPB Offset**	**Byte Lane**	**EC Type**	
PORT92 Enable Register	30h	3F0h	R/W	330h	0	R/W	

Note 10-6 CMD is bit D3 in the Keyboard Status Read Register.

All Registers listed in Table 10-14 are powered by VTR and reset by nSYS_RST except the PORT92 Register which is powered by VTR and reset by nSIO_RESET (See Section , "iRESET OUT," on page 133).

10.12 Configuration Registers

10.12.1 PORT 92 ENABLE

The MEC1632 supports LPC I/O writes to port 92h as a quick alternate mechanism for generating a CPU_RESET pulse or controlling the state of GATEA20.

TABLE 10-15: PORT92 ENABLE REGISTER

HOST CONFIG INDEX	30h					8-bit	**HOST SIZE**	
EC OFFSET	330h					8-bit	**EC SIZE**	
POWER	VTR					00h	nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R							R/W
EC TYPE	-	-	-	-	-	-	-	-
BIT NAME	Reserved							P92_EN

10.13 Runtime Registers

10.13.1 PORT 92

The MEC1632 supports LPC I/O writes to port HOST I/O address 92h as a quick alternate mechanism for generating a CPU_RESET pulse or controlling the state of GATEA20.

The PORT92 Register resides at HOST I/O address 92h and is used to support the alternate reset (ALT_RST#) and alternate GATEA20 (ALT_A20) functions. This register defaults to 00h on assertion of nSIO_RESET (See Section , "iRESET OUT," on page 133).

© 2012-2018 Microchip Technology Inc.

Setting the Port92 Enable bit (PORT92 Enable Register) enables the Port92h Register. When Port92 is disabled, by clearing the Port92 Enable bit, then access to this register is completely disabled (I/O writes to host 92h are ignored and I/O reads float the system data bus SD[7:0]).

TABLE 10-16: PORT92 REGISTER

HOST I/O ADDRESS	0092h					8-bit	**HOST SIZE**	
EC OFFSET	00h					8-bit	**EC SIZE**	
POWER	VTR					00h	nSIO_RESET **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R						R/W	R/W
EC TYPE	-	-	-	-	-	-	-	-
BIT NAME	Reserved						ALT_GATEA20	ALT_CPU_RESET

ALT_CPU_RESET

This bit provides an alternate means to generate a CPU_RESET pulse. The CPU_RESET output provides a means to reset the system CPU to effect a mode switch from Protected Virtual Address Mode to the Real Address Mode. This provides a faster means of reset than is provided through the EC keyboard controller. Writing a “1” to this bit will cause the ALT_RST# internal signal to pulse (active low) for a minimum of 6µs after a delay of 14µs. Before another ALT_RST# pulse can be generated, this bit must be written back to “0”.

ALT_GATEA20

This bit provides an alternate means for system control of the MEC1632 GATEA20 pin.

0=ALT_A20 is driven low

1=ALT_A20 is driven high

When Port 92 is enabled, writing a 0 to bit 1 of the PORT92 Register forces ALT_A20 low. ALT_A20 low drives GATEA20 low, if A20 from the keyboard controller is also low. When Port 92 is enabled, writing a 1 to bit 1 of the PORT92 Register forces ALT_A20 high. ALT_A20 high drives GATEA20 high regardless of the state of A20 from the keyboard controller.

10.13.2 GATE A20

The MEC1632 contains on-chip logic support for the GATEA20 hardware speed-up feature. GATEA20 is part of the control required to mask address line A20 to emulate 8086 addressing.

In addition to the ability for the host to control the GATEA20 output signal directly, a configuration bit called “SAEN” (Software Assist Enable, bit 1 of Keyboard Control Register is provided; when set, SAEN allows firmware to control the GATEA20 output.

When SAEN is set, a 1 bit register (GATEA20 Control Register) controls the GATEA20 output. The register bit allocation is shown in Table 10-13.

Note 10-7 Refer to the GATEA20 Control description for information on this register. This is a one bit register (Bits 1-7=0 on read).

10.13.3 GATEA20 CONTROL

TABLE 10-17: GATEA20 CONTROL REGISTER

HOST OFFSET	NA			8-bit			**HOST SIZE**	
EC OFFSET	100h			8-bit			**EC SIZE**	
POWER	VTR			00h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R							R/W
BIT NAME	Reserved							GATEA20

GATEA20

Writing a “0” into bid GATEA20 causes the GATEA20 output to go low, and vice versa.

Host control and firmware control of GATEA20 affect two separate register elements. Read back of GATEA20 through the use of EC OFFSET 100h reflects the present state of the GATEA20 output signal: if SAEN is set, the value read back corresponds to the last firmware-initiated control of GATEA20; if SAEN is reset, the value read back corresponds to the last host-initiated control of GATEA20.

Host control of the GATEA20 output is provided by the hardware interpretation of the “GATEA20 sequence” (see Table 10-18). The foregoing description assumes that the SAEN configuration bit is reset.

When the MEC1632 receives a “D1” command followed by data (via the host interface), the on-chip hardware copies the value of data bit 1 in the received data field to the GATEA20 host latch. At no time during this host-interface transaction will PCOBF or the IBF flag (bit 1) in the Keyboard Status Read Register be activated; for example, this host control of GATEA20 is transparent to firmware, with no consequent degradation of overall system performance. Table 10-18 details the possible GATEA20 sequences and the MEC1632 responses.

On VCC_POR, GATEA20 will be set.

An additional level of control flexibility is offered via a memory-mapped synchronous set and reset capability. Any data written to EC OFFSET 108h causes the GATEA20 host latch to be set; any data written to EC OFFSET 10Ch causes it to be reset. This control mechanism should be used with caution. It was added to augment the “normal” control flow as described above, not to replace it. Since the host and the firmware have asynchronous control capability of the host latch via this mechanism, a potential conflict could arise. Therefore, after using the EC OFFSET 108h and EC OFFSET 10Ch addresses, firmware should read back the GATEA20 status via EC OFFSET 100h (with SAEN = 0) to confirm the actual GATEA20 response.

TABLE 10-18: GATEA20 COMMAND/DATA SEQUENCE EXAMPLES

Data Byte	R/W	D[0:7]	IBF FLAG	GATEA20	Comments
1 0 1	W W W	D1 DF FF	0 0 0	Q 1 1	GATEA20 Turn-on Sequence
1 0 1	W W W	D1 DD FF	0 0 0	Q 0 0	GATEA20 Turn-off Sequence
1 1 0 1	W W W W	D1 D1 DF FF	0 0 0 0	Q Q 1 1	GATEA20 Turn-on Sequence(*)

© 2012-2018 Microchip Technology Inc.

TABLE 10-18: GATEA20 COMMAND/DATA SEQUENCE EXAMPLES (CONTINUED)

Data Byte	R/W	D[0:7]	IBF FLAG	GATEA20	Comments
1 1 0 1	W W W W	D1 D1 DD FF	0 0 0 0	Q Q 0 0	GATEA20 Turn-off Sequence(*)
1 1 1	W W W	D1 XX** FF	0 1 1	Q Q Q	Invalid Sequence

Note 10-8

- All examples assume that the SAEN configuration bit is 0.
- “Q” indicates the bit remains set at the previous state.
- *Not a standard sequence.
- **XX = Anything except D1.
- If multiple data bytes, set IBF and wait at state 0. Let the software know something unusual happened.
- For data bytes, only D[1] is used; all other bits are don't care.

TABLE 10-19: SETGA20L REGISTER

HOST OFFSET	NA					8-bit	**HOST SIZE**	
EC OFFSET	108h					8-bit	**EC SIZE**	
POWER	VTR					00h	nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	W							
BIT NAME								

Note 10-9 Refer to the GATEA20 Hardware Speed-up description for information on this register. A write to this register sets GATEA20.

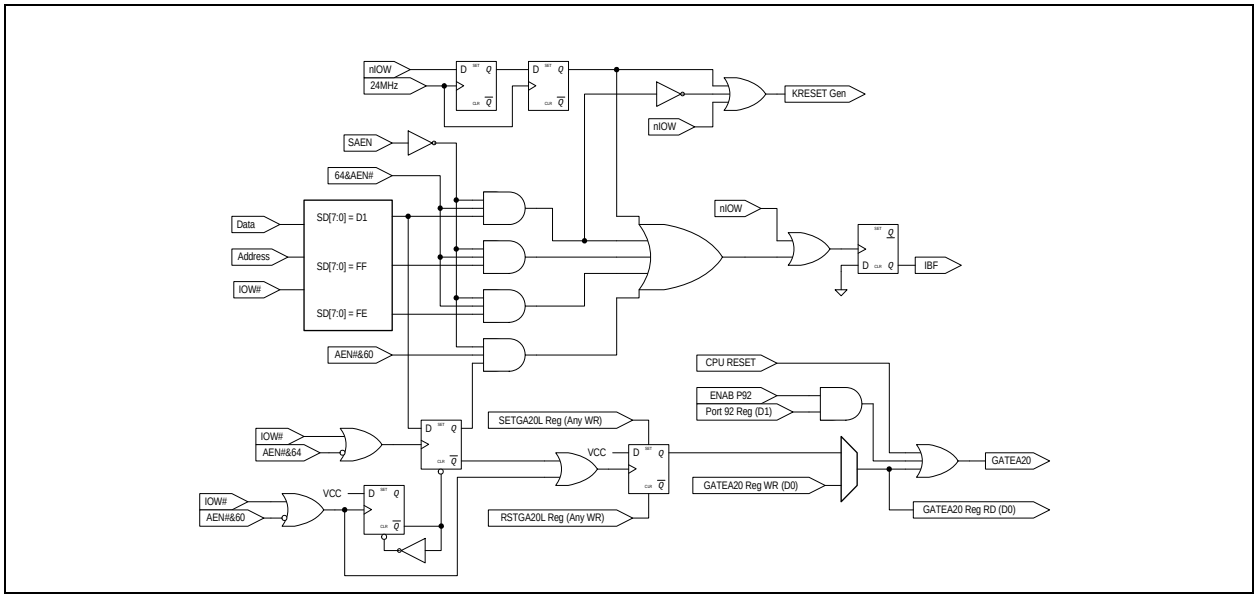
TABLE 10-20: RSTGA20L REGISTER

HOST OFFSET	NA					8-bit	**HOST SIZE**	
EC OFFSET	10Ch					8-bit	**EC SIZE**	
POWER	VTR					00h	nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	W							
BIT NAME								

Note 10-10 Refer to the GATEA20 Hardware Speed-up description for information on this register. A write to this register re-sets GATEA20.

© 2012-2018 Microchip Technology Inc.

FIGURE 10-2: GATEA20 IMPLEMENTATION DIAGRAM

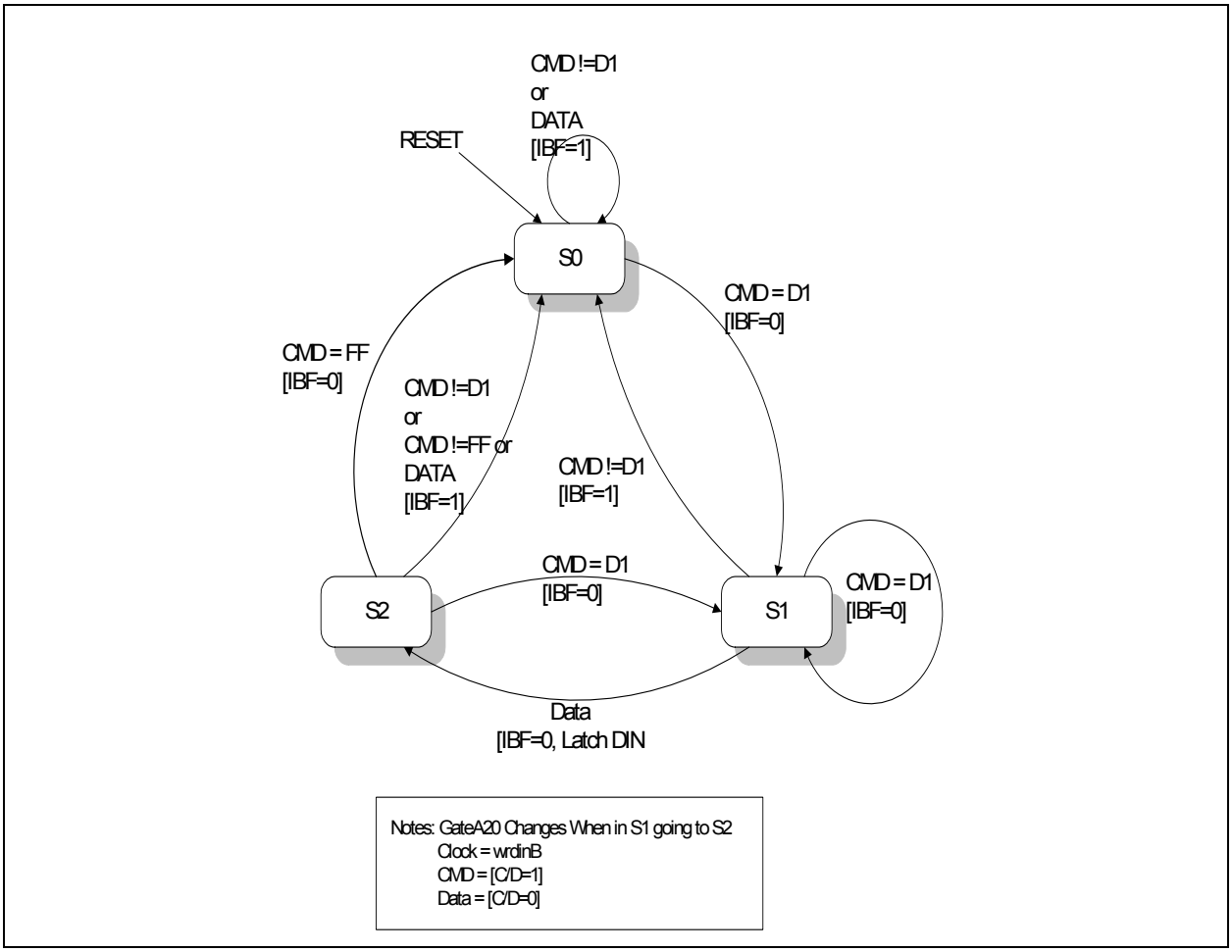


Note 10-11 Host Commands (FF, FE, & D1) do not cause IBF. The method of blocking IBF in Figure 10-2 is the nIOW not being asserted when FF, FE, & D1 Host commands are written".

The hardware GATEA20 state machine returns to state S1 from state S2 when CMD = D1 (Figure 10-3).

© 2012-2018 Microchip Technology Inc.

FIGURE 10-3: GATEA20 STATE MACHINE



10.14 CPU_RESET Hardware Speed-Up

The ALT_CPU_RESET bit generates, under program control, the ALT_RST# signal, which provides an alternate, means to drive the MEC1632 CPU_RESET pin which in turn is used to reset the Host CPU. The ALT_RST# signal is internally NANDed together with the KBDRESET# pulse from the KRESET Speed up logic to provide an alternate software means of resetting the host CPU.

Note 10-12 Before another ALT_RST# pulse can be generated, ALT_CPU_RESET must be cleared to “0” either by an nSIO_RESET (See Section , "iRESET OUT," on page 133) or by a write to the PORT92 Register with bit 0 = “0”. A ALT_RST# pulse is not generated in the event that the ALT_CPU_RESET bit is cleared and set before the prior ALT_RESET# pulse has completed.

Note 10-13 this function is qualified by the SLP_EN signal or the Activate bit. If either of these signals goes to ‘0’, then the 1MHz clocks source is disabled if the PGEN is not currently active. See EC Blocks Sleep Enables/Clock Required Registers Bit Names on page 138 & Section 7.4.10.5.4, "Block Sleep Enables," on page 117.

FIGURE 10-4: CPU_RESET IMPLEMENTATION DIAGRAM

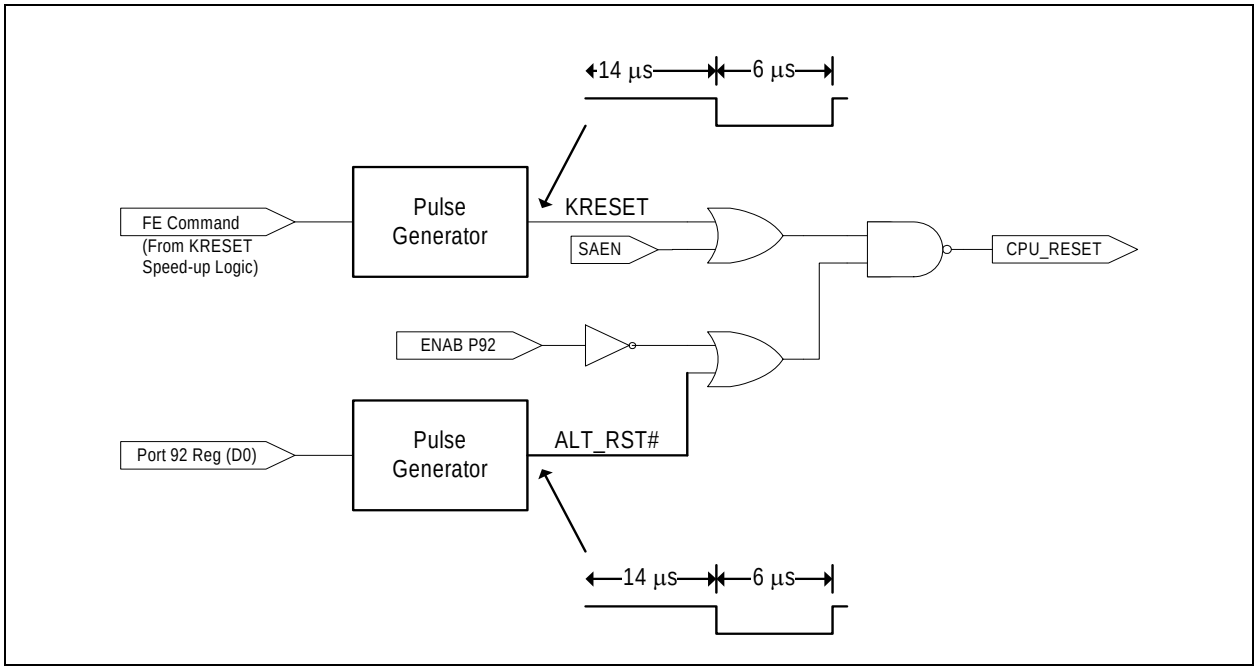


FIGURE 10-5: CPU_RESET TIMING

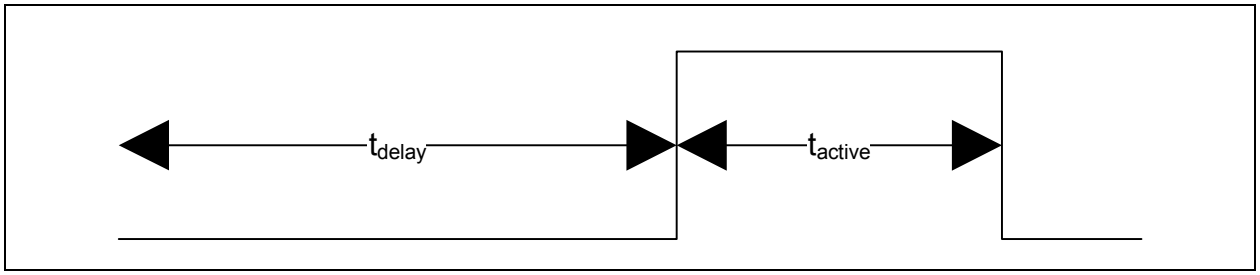


TABLE 10-21: CPU_RESET TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t_{delay}	Delay prior to active pulse	14	15	15.5	µs
t_{active}	Active pulse width	6	8	8.5	µs

Note 10-14 Figure 10-5 & Table 10-21 refers to Figure 10-4 in which CPU_RESET is the inverse of ALT_RST# & KRESET.

Note 10-15 The KBRST pin function is the output of CPU_RESET described in Section 10.14, "CPU_RESET Hardware Speed-Up," on page 203.

© 2012-2018 Microchip Technology Inc.

11.0 ACPI PM1 BLOCK INTERFACE

11.1 General Description

The MEC1632 supports ACPI as described in this section. These features comply with the ACPI Specification, Revision 1.0, through a combination of hardware and EC software.

The MEC1632 implements the ACPI fixed registers but includes only those bits that apply to the power button sleep button and RTC alarm events. The ACPI WAK_ SLP_TYPx, and SLP_ bits are also supported.

11.2 Power, Clocks and Reset

11.2.1 POWER DOMAIN

This block is powered by VTR

See Section 7.1.1, "Power Configuration," on page 95 for details on power domains.

11.2.2 CLOCKS

This block has one clock input, the LPC Bus Clock.

See Section 7.4, "Clock Generator," on page 98 for details on clocks.

11.2.3 POWER ON RESET

This block is reset on a nSYS_RST. After nSYS_RST is asserted, all registers are set to ‘0’.

See Section 7.6, "Reset Interface," on page 124 for details on reset.

11.3 Interrupts

11.3.1 SCI INTERRUPTS TO THE HOST

The functions described in the following sub-sections can generate a SCI event on the EC_SCI# pin. In the MEC1632, an SCI event is considered the same as an ACPI wakeup or runtime event. The EC can also generate a SCI on the EC_SCI# pin by setting the EC_SCI_ bit in the EC_SCI# Pin Interface on page 211.

11.3.2 INTERRUPTS TO THE EC

An Interrupt is generated to the EC on PM1_CTL2 bit of GIRQ13 Source Register by the Host writing to Power Management 1 Control Register 2 (PM1_CNTRL 2).

An Interrupt is generated to the EC on PM1_EN2 bit of GIRQ13 Source Register by the Host writing to Power Management 1 Enable Register 2 (PM1_EN 2).

An Interrupt is generated to the EC on PM1_STS2 bit of GIRQ13 Source Register by the Host writing to Power Management 1 Status Register 2 (PM1_STS 2).

11.3.3 ACPI PM1 BLOCK SCI EVENT-GENERATING FUNCTIONS

11.3.3.1 Power Button with Override

The power button has a status and an enable bit in the PM1_BLK of registers to provide an SCI upon the button press. The status bit is software Read/Writable by the EC; the enable bit is Read-only by the EC. It also has a status and enable bit in the PM1_BLK of registers to indicate and control the power button override (fail-safe) event. These bits are not required by ACPI. The power button override event status bit is software Read/Writable by the EC; the enable bit is software read-only by the EC. The enable bit for the override event is located at bit 1 in the Power Management 1 Control Register 2 (PM1_CNTRL 2).

The PWRBTN_ bit is set by the Host to enable the generation of an SCI due to the power button event. The status bit is set by the EC when it generates a power button event and is cleared by the Host writing a ‘1’ to this bit (writing a ‘0’ has no effect); it can also be cleared by the EC. If the enable bit is set, the EC will generate an SCI power management event.

© 2012-2018 Microchip Technology Inc.

11.3.3.2 Sleep Button

The sleep button has a status and an enable bit in the PM1_BLK of registers to provide an SCI upon the button press. The status bit is software Read/Writable by the EC; the enable bit is Read-only by the EC.

The SLPBTN_ bit is set by the Host to enable the generation of an SCI due to the sleep button event. The status bit is set by the EC when it generates a sleep button event and is cleared by the Host writing a '1' to this bit (writing a '0' has no effect); it can also be cleared by the EC. If the enable bit is set, the EC will generate an SCI power management event.

11.3.4 RTC ALARM

The ACPI specification requires that the RTC alarm generate a hardware wake-up event from the sleeping state. The RTC alarm can be enabled as an SCI event and its status can be determined through bits in the PM1_BLK of registers. The status bit is software Read/Writable by the EC; the enable bit is Read-only by the EC.

The RTC_ bit is set by the Host to enable the generation of an SCI due to the RTC alarm event. The status bit is set by the EC when the RTC generates an alarm event and is cleared by the Host writing a '1' to this bit (writing a '0' has no effect); it can also be cleared by the EC. If the enable bit is set, the EC will generate an SCI power management event.

11.4 Registers

Each instance of the ACPI PM1 Block Interface has its own Logical Device Number, and Base Address as indicated in Table 11-1.

TABLE 11-1: ACPI PM1 Block Interface BASE ADDRESS TABLE

ACPI PM1 Block Interface **Instance**	Table 4-1 on page 59 **LDN**	**AHB Base Address**
ACPI PM1 Block Interface	6h	FF_1800h

The ACPI register model consists of a number of fixed register blocks that perform designated functions. A register block consists of a number of registers that perform Status, Enable and Control functions. The ACPI specification deals with events (which have an associated interrupt status and enable bits, and sometimes an associated control function) and control features. The status registers illustrate what defined function is requesting ACPI interrupt services (SCI). Any status bit in the ACPI specification has the following attributes:

Status bits are only set through some defined hardware or EC event.

Unless otherwise noted, status bits are cleared by the system writing a "1" to that bit position, and upon nSYS_RST. Writing a '0' has no effect.

Status bits only generate interrupts while their associated bit in the enable register is set.

Function bit positions in the status register have the same bit position in the enable register (there are exceptions to this rule, special status bits have no enables).

Note that this implies that if the respective enable bit is reset and the hardware event occurs, the respective status bit is set; however no interrupt is generated until the enable bit is set. This allows software to test the state of the event (by examining the status bit) without necessarily generating an interrupt. There are a special class of status bits that have no respective enable bit, these are called out specifically, and the respective enable bit in the enable register is marked as reserved for these special cases.

The enable registers allow the setting of the status bit to generate an interrupt (under EC control). As a general rule, there is an enable bit in the enable register for every status bit in the status register. The control register provides special controls for the associated event, or special control features that are not associated with an interrupt event. The order of a register block is the status registers, followed by enable registers, followed by control registers.

The registers in the MEC1632 ACPI PM1 Block Interface occupy eight addresses in the host I/O space and are specified as offsets from the ACPI PM1 Block base address (Table 11-1).

© 2012-2018 Microchip Technology Inc.

TABLE 11-2: ACPI PM1 Block Interface REGISTER SUMMARY

	Host Access			EC Access			
Register Name	**Host I/O Index**	**SPB Offset**	**Host Type**	**SPB Offset**	**Byte lane**	**EC Type**	**Notes**
Power Management 1 Status Register 1 (PM1_STS) 1	0h	00h	R	100h	0	R	
Power Management 1 Status Register 2 (PM1_STS 2)	1h	01h	R/WC	101h	1	R/W	Table 11-4 Note 11-4
Power Management 1 Enable Register 1 (PM1_EN 1)	2h	02h	R	102h	2	R	
Power Management 1 Enable Register 2 (PM1_EN 2)	3h	03h	R/W	103h	3	R	Table 11-6 Note 11-5
Power Management 1 Control Register 1 (PM1_CNTRL 1)	4h	04h	R	104h	0	R	
Power Management 1 Control Register 2 (PM1_CNTRL 2)	5h	05h	R/W	105h	1	R	Table 11-8 Note 11-6
Power Management 2 Control Register 1 (PM2_CNTRL 1)	6h	06h	R	106h	2	R	Note 11-2
Power Management 2 Control Register 2 (PM2_CNTRL 2)	7h	07h	R	107h	3	R	Note 11-2
EC_PM_STS Register	-	-	-	110h	0	R/W	

Note 11-1 Byte 0 of this register is reserved.

Note 11-2 These registers return ‘0’ when read, writes have no effect.

11.4.1 POWER MANAGEMENT 1 STATUS 1 (PM1_STS 1)

TABLE 11-3: POWER MANAGEMENT 1 STATUS REGISTER 1 (PM1_STS) 1

HOST OFFSET	0h			8-bit			**HOST SIZE**
EC OFFSET	100h						**EC SIZE**
POWER	VTR			00h			nSYS_RST **DEFAULT**
BUS	LPC SPB						
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R	R	R	R	R	R
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							

RESERVED

Reserved bits return ‘0’ when read.

11.4.2 POWER MANAGEMENT 1 STATUS 2 (PM1_STS 2)

TABLE 11-4: POWER MANAGEMENT 1 STATUS REGISTER 2 (PM1_STS 2)

HOST OFFSET	1h			8-bit			**HOST SIZE**	
EC OFFSET	101h						**EC SIZE**	
POWER	VTR			00h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/WC	R	R	R	R/WC	R/WC	R/WC	R/WC
EC TYPE	R/W	R	R	R	R/W	R/W	R/W	R/W
BIT NAME	WAK_ STS	Reserved			PWRBT-NOR_ STS	RTC_ STS	SLPBT-N_STS	PWRBTN _STS

Note 11-3 These bits are set/cleared by the EC directly i.e., writing '1' sets the bit and writing '0' clears it. These bits can also be cleared by the Host software writing a one to this bit position and by nSYS_RST. Writing a 0 by the Host has no effect.

Note 11-4 An interrupt (PM1_CTL2) is generated to the EC when the Host writes to this register.

PWRBTN_STS

This bit can be set or cleared by the EC to simulate a Power button status if the power is controlled by the EC. The Host writing a one to this bit can also clear this bit. The EC must generate the associated SCI interrupt under software control.

SLPBTN_STS

This bit can be set or cleared by the EC to simulate a Sleep button status if the sleep state is controlled by the EC. The Host writing a one to this bit can also clear this bit. The EC must generate the associated SCI interrupt under software control.

RTC_STS

This bit can be set or cleared by the EC to simulate a RTC status. The Host writing a one to this bit can also clear this bit. The EC must generate the associated SCI interrupt under software control.

PWRBTNOR_STS

This bit can be set or cleared by the EC to simulate a Power button override event status if the power is controlled by the EC. The Host writing a one to this bit can also clear this bit. The EC must generate the associated hardware event under software control.

WAK_STS

This bit can be set or cleared by the EC. The Host writing a one to this bit can also clear this bit.

© 2012-2018 Microchip Technology Inc.

11.4.3 POWER MANAGEMENT 1 ENABLE 1 (PM1_EN 1)

TABLE 11-5: POWER MANAGEMENT 1 ENABLE REGISTER 1 (PM1_EN 1)

HOST OFFSET	02h					8-bit	**HOST SIZE**	
EC OFFSET	102h						**EC SIZE**	
POWER	VTR					00h	nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R	R	R	R	R	R
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							

RESERVED

Reserved bits return '0' when read.

11.4.4 POWER MANAGEMENT 1 ENABLE 2 (PM1_EN 2)

TABLE 11-6: POWER MANAGEMENT 1 ENABLE REGISTER 2 (PM1_EN 2)

HOST OFFSET	03h					8-bit	**HOST SIZE**	
EC OFFSET	103h						**EC SIZE**	
POWER	VTR					00h	nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R	R	R	R/W	R/W	R/W
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved					RTC_EN	SLPBTN_EN	PWRBTN_EN

Note 11-5 An interrupt (PM1_EN2) is generated to the EC when the Host writes to this register.

PWRBTN_EN

This bit can be read or written by the Host. It can be read by the EC.

SLPBTN_EN

This bit can be read or written by the Host. It can be read by the EC.

RTC_EN

This bit can be read or written by the Host. It can be read by the EC.

11.4.5 POWER MANAGEMENT 1 CONTROL 1 (PM1_CNTRL 1)

TABLE 11-7: POWER MANAGEMENT 1 CONTROL REGISTER 1 (PM1_CNTRL 1)

HOST OFFSET	04h						8-bit	**HOST SIZE**
EC OFFSET	104h							**EC SIZE**
POWER	VTR						00h	nSYS_RST **DEFAULT**
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R	R	R	R	R	R
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							

RESERVED

Reserved bits return '0' when read.

11.4.6 POWER MANAGEMENT 1 CONTROL 2 (PM1_CNTRL 2)

TABLE 11-8: POWER MANAGEMENT 1 CONTROL REGISTER 2 (PM1_CNTRL 2)

HOST OFFSET	5h						8-bit	**HOST SIZE**
EC OFFSET	105h							**EC SIZE**
POWER	VTR						00h	nSYS_RST **DEFAULT**
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	W	R/W			R/W	R/W
EC TYPE	R	R	R/WC	R			R	R
BIT NAME	Reserved		SLP_EN	SLP_TYPx			PWRBTNOR_EN	Reserved

Note 11-6 An interrupt (PM1_CTL2) is generated to the EC when the Host writes to this register.

PWRBTNOR_EN

This bit can be set or cleared by the Host, read by the EC.

SLP_TYPx

These bits can be set or cleared by the Host, read by the EC.

SLP_EN

Refer to Table 11-9, "SLP_EN Definition"

TABLE 11-9: SLP_EN DEFINITION

Host / EC	R/W	Description
Host	Read	Always reads 0
	Write	Writing a 0 has no effect, Writing a 1 sets this bit
EC	Read	Reads the value of the bit
	Write	Writing a 0 has no effect, Writing a 1 clears this bit

© 2012-2018 Microchip Technology Inc.

11.4.7 POWER MANAGEMENT 2 CONTROL 1 (PM2_CNTRL 1)

TABLE 11-10: POWER MANAGEMENT 2 CONTROL REGISTER 1 (PM2_CNTRL 1)

HOST OFFSET	06h					8-bit	**HOST SIZE**	
EC OFFSET	106h						**EC SIZE**	
POWER	VTR					00h	nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R	R	R	R	R	R
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							

RESERVED

Reserved bits return ‘0’ when read.

11.4.8 POWER MANAGEMENT 2 CONTROL 2 (PM2_CNTRL 2)

TABLE 11-11: POWER MANAGEMENT 2 CONTROL REGISTER 2 (PM2_CNTRL 2)

HOST OFFSET	07h					8-bit	**HOST SIZE**	
EC OFFSET	107h						**EC SIZE**	
POWER	VTR					00h	nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R	R	R	R	R	R
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							

RESERVED

Reserved bits return ‘0’ when read.

11.5 EC_SCI# Pin Interface

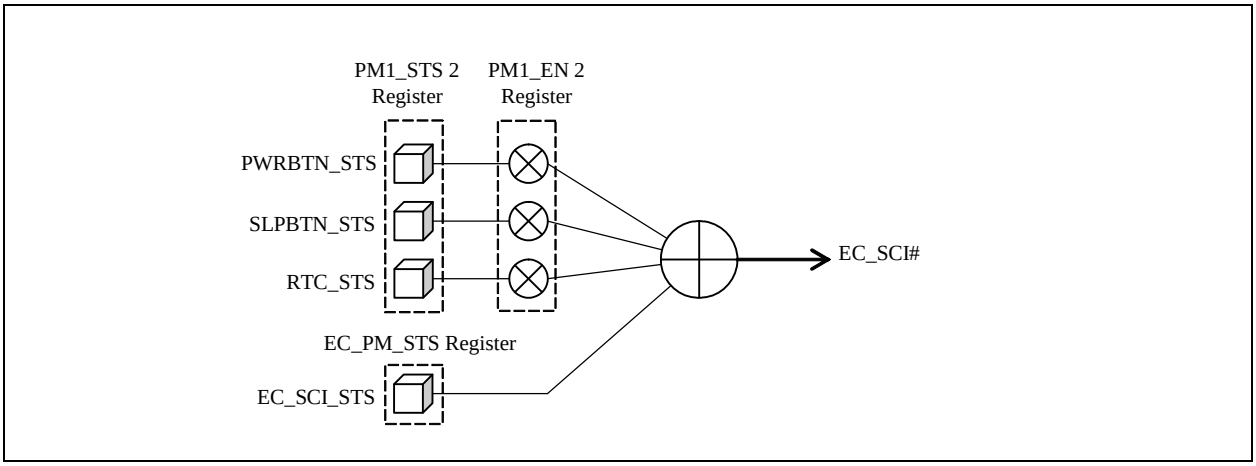
The EC_SCI# pin logic hardware is shown below in Figure 11-1.

Any or all of the PWRBTN, SLPBTN_STS, and RTC_ bits in the Power Management 1 Status Register 2 (PM1_STS 2) can assert the EC_SCI# pin if enabled by the PWRBTN_, SLPBTN_, and RTC_ bits in the PM1_EN 2 register.

The EC_SCI_ bit can assert the EC_SCI# pin at any time, without being enabled. The EC_SCI_ bit is located in the EC_PM_STS Register.

The EC_SCI_ bit is in the MEC1632 and is read/write by the EC. If the EC_SCI_ bit is “1”, an interrupt is generated on the EC_SCI# pin.

FIGURE 11-1: HARDWARE EC_SCI# INTERFACE



11.5.1 EC_PM_STS REGISTER

TABLE 11-12: EC_PM_STS REGISTER

HOST OFFSET							**HOST SIZE**	
EC OFFSET	110h					8-bit	**EC SIZE**	
POWER	VTR					00h	nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	UD[6:0							EC_SCI_ STS

EC_SCI_STS

If the EC_SCI_ bit is “1”, an interrupt is generated on the EC_SCI# pin.

UD[6:0]

User-defined bits. This bits do not generate an interrupt.

© 2012-2018 Microchip Technology Inc.

12.0 MAILBOX REGISTER INTERFACE

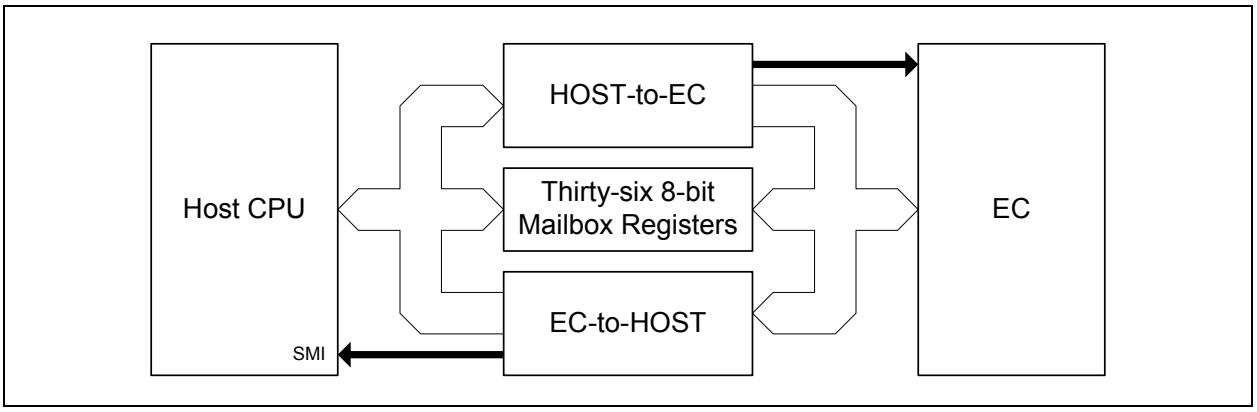
12.1 General Description

The MailBox Register Interface provides a standard run-time mechanism for the host to communicate with the Embedded Controller (EC) and other logical components in the MEC1632 (Figure 12-1). The Mailbox Registers Interface includes a total of 36 index-addressable 8-bit registers and a Mailbox Registers Interface Host Access Port. Thirty-two of 36 index-addressable 8-bit registers are EC Mailbox registers. The Mailbox Registers Interface Host Access Port consists of two 8-bit run-time registers that occupy two addresses in the HOST I/O space. The Mailbox Registers Interface Host Access Port is used by the host to access the 36 index-addressable 8-bit registers.

Note:	In this specification, host access to registers in the Mailbox Registers Interface through the host access port are identified by the prefix MBX in front of a hexadecimal index address.

12.1.1 BLOCK DIAGRAM

FIGURE 12-1: MAILBOX BLOCK DIAGRAM



12.2 Power, Clocks and Reset

12.2.1 POWER DOMAIN

This block is powered by the VTR power supply.

See Section 7.1.1, "Power Configuration," on page 95 for details on power domains.

12.2.2 CLOCKS

This block has one clock input, the LPC Bus Clock.

See Section 7.4, "Clock Generator," on page 98 for details on clocks.

12.2.3 RESET

This block is reset when nSYS_RST is asserted.

In addition the MBX_INDEX Register & MBX_DATA Register are reset when VCC_PWRGD Signal Pin function is de-asserted.

See Section 7.6, "Reset Interface," on page 124 for details on reset.

12.3 Interrupts

The MailBox Register Interface can generate an interrupt event for the HOST-to-EC events. See HOST-to-EC Mailbox Register on page 218. The interrupt source is routed onto the MBX bit in the GIRQ15 Source Register and is a level, active high signal.

12.3.1 MAILBOX REGISTER INTERFACE (LDN 0H) SIRQ ROUTING

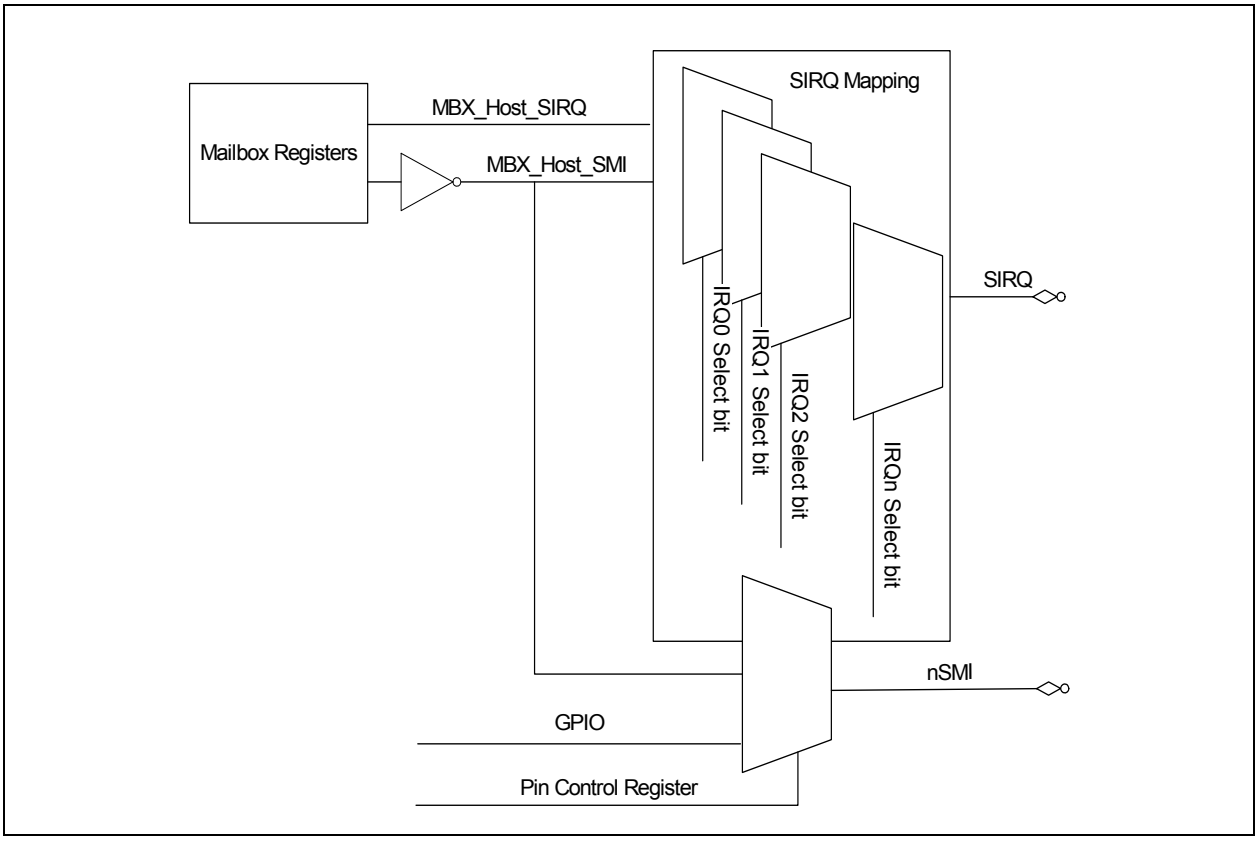
The MailBox Register Interface can generate a SIRQ event for the EC-to-HOST EC events. See HOST-to-EC Mailbox Register on page 218. This interrupt is routed to the SIRQ block (see Section 5.7.1, "SERIRQ Configuration Registers," on page 72). For this interrupt, the SELECT on page 72 is cleared to '0' in the Interrupt Configuration Register for the selected SIRQ frame.

The MailBox Register Interface can generate a SMI event from the SMI Interrupt Source Register on page 219. The SMI event can be routed to any frame in the SIRQ stream and to the nSMI pin. To enable SMI routing to the SIRQ stream, the bit SELECT on page 72 is set to '1' in the Interrupt Configuration Register for the selected SIRQ frame. The SMI event can be routed to nSMI pin by selecting the nSMI signal function in the associated Pin Control Register on page 396.

The SMI event produces a standard active low on the serial IRQ stream and active low on the open drain nSMI pin. See FIGURE 12-2: Mailbox SIRQ and SMI Routing on page 214.

See Section 5.7.1, "SERIRQ Configuration Registers," on page 72.

FIGURE 12-2: MAILBOX SIRQ AND SMI ROUTING



© 2012-2018 Microchip Technology Inc.

12.4 Registers Summary

The MailBox Register Interface has its own Logical Device Number, and Base Address as indicated in Table 12-1. The Host LPC I/O addresses for the MailBox Register Interface are selected via a Base Address Register (see Section 5.6.2, "Base Address Registers," on page 66). LPC access to configuration registers is through Host Access Configuration Port (see Section 5.5.1, "Host Access Port," on page 65.)

Table 12-2 is a register summary for the MailBox Register Interface block.

TABLE 12-1: MailBox Register Interface BASE ADDRESS TABLE

MailBox Register Interface Instance	LDN from (Table 4-2 on page 59)	AHB Base Address
Mailbox Interface	0h	FF_0000h

Note: The Host LPC I/O addresses for this instance is selected via a Base Address Register (see Section 5.6.2, "Base Address Registers," on page 66). LPC access to configuration registers is through the Host Access Configuration Port (see Section 5.5.1, "Host Access Port," on page 65).

The Table 12-2 is a register summary for one instance of the MailBox Register Interface. The LPC I/O address for each Run-Time Register is described below as an offset from its Base Address Register. Each EC address is indicated as an SPB Offset from its AHB base address. Each Configuration register is accessed through the Host Access Port is via its LDN indicated in Table 12-1 on page 215 and its Host Access Port index which is described as “Host Config Index” in the tables below.

TABLE 12-2: MailBox Register Interface REGISTER SUMMARY

		Host I/O Access			EC Interface			
	Register Name	**Host I/O Index**	**SPB Offset**	**Host Type**	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
	MBX_INDEX Register	00h	00h	R/W	-	-	-	
	MBX_DATA Register	01h	01h	R/W	-	-	-	
		Mailbox Index						
1.	HOST-to-EC Mailbox Register	MBX 00h	-	R/W	100h	0	R/WC	Note 12-1
2.	EC-to-Host Mailbox Register	MBX 01h	-	R/WC	104h	0	R/W	Note 12-2
3.	SMI Interrupt Source Register	MBX 02h	-	Table 12-7	108h	0	Table 12-7	
4.	SMI Interrupt Mask Register	MBX 03H	-	R/W	10Ch	0	R/W	
5.	**Mailbox register [0]**	MBX10h	-	R/W	110h	0	R/W	
6.	**Mailbox register [1]**	MBX11h	-			1		
7.	**Mailbox register [2]**	MBX12h	-			2		
8.	**Mailbox register [3]**	MBX13h	-			3		
9.	**Mailbox register [4]**	MBX14h	-	R/W	114h	0	R/W	
10.	**Mailbox register [5]**	MBX15h	-			1		
11.	**Mailbox register [6]**	MBX16h	-			2		
12.	**Mailbox register [7]**	MBX17h	-			3		

TABLE 12-2: MailBox Register Interface REGISTER SUMMARY (CONTINUED)

		Host I/O Access			EC Interface			
	Register Name	**Host I/O Index**	**SPB Offset**	**Host Type**	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
13.	**Mailbox register [8]**	MBX18h	-	R/W	118h	0	R/W	
14.	**Mailbox register [9]**	MBX19h	-			1		
15.	**Mailbox register (A]**	MBX1Ah	-			2		
16.	**Mailbox register [B]**	MBX1Bh	-			3		
17.	**Mailbox register [C]**	MBX1Ch	-	R/W	11Ch	0	R/W	
18.	**Mailbox register [D]**	MBX1Dh	-			1		
19.	**Mailbox register [E]**	MBX1Eh	-			2		
20.	**Mailbox register [F]**	MBX1Fh	-			3		
21.	**Mailbox register [10]**	MBX20h	-	R/W	120h	0	R/W	
22.	**Mailbox register [11])**	MBX21h	-			1		
23.	**Mailbox register [12]**	MBX22h	-			2		
24.	**Mailbox register [13]**	MBX23h	-			3		
25.	**Mailbox register [14]**	MBX24h	-	R/W	124h	0	R/W	
26.	**Mailbox register [15]**	MBX25h	-			1		
27.	**Mailbox register [16]**	MBX26h	-			2		
28.	**Mailbox register [17]**	MBX27h	-			3		
29.	**Mailbox register [18]**	MBX28h	-	R/W	128h	0	R/W	
30.	**Mailbox register [19]**	MBX29h	-			1		
31.	**Mailbox register [1A]**	MBX2Ah	-			2		
32.	**Mailbox register [1B]**	MBX2Bh	-			3		
33.	**Mailbox register [1C]**	MBX2Ch	-	R/W	12Ch	0	R/W	
34.	**Mailbox register [1D]**	MBX2Dh	-			1		
35.	**Mailbox register [1E]**	MBX2Eh	-			2		
36.	**Mailbox register [1F]**	MBX2Fh	-			3		

Note 12-1 Interrupt is cleared when read by the EC.

Note 12-2 Interrupt is cleared when read by the host.

12.4.1 MAILBOX REGISTERS INTERFACE HOST ACCESS PORT

The Mailbox registers access port is two runtime registers that occupy two addresses in the Host I/O space: MBX_INDEX Register & MBX_DATA Register.

To access a Mailbox register once the Mailbox Registers Interface Base Address has been initialized, write the Mailbox register index address to the MBX Index port and read or write the Mailbox register data from the MBX data port.

See Table 12-2, “MailBox Register Interface Register Summary,” on page 215.

12.4.2 MAILBOX CONTROL REGISTERS

Mailbox Register, HOST-to-EC, and Mailbox Register, EC-to-HOST, are specifically designed to pass commands between the host and the EC (FIGURE 12-1: on page 213). If enabled, these registers can generate interrupts.

Mailbox Register and Mailbox Register are not dual-ported, so the HOST BIOS and Keyboard BIOS must be designed to properly share these registers. When the host performs a write of the HOST-to-EC mailbox register, an interrupt will be generated and seen by the EC if unmasked. When the EC writes FF to the HOST-to-EC mailbox register, resets the register to 00h, providing a simple means for the EC to inform the host that an operation has been completed.

When the EC writes the EC-to-HOST mailbox register, an SMI may be generated and seen by the host if unmasked. When the Host CPU writes FFh to the EC-to-HOST mailbox register, the EC-to-HOST register resets to 00h, providing a simple means for the host to inform that EC that an operation has been completed.

© 2012-2018 Microchip Technology Inc.

PROGRAMMER'S NOTE: The protocol used to pass commands back and forth through the Mailbox Registers Interface is left to the System designer. Microchip can provide an application example of working code in which the host uses the Mailbox registers to gain access to all of the EC registers.

12.5 Register Details

12.5.1 MAILBOX INDEX REGISTER

TABLE 12-3: MBX_INDEX REGISTER

HOST OFFSET	00h				8-Bit	**HOST SIZE**		
EC OFFSET	NA					**EC SIZE**		
POWER	VTR				00h 00h	nSYS_RST **DEFAULT** **&** VCC_PWRGD **DE-ASSERTION**		
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	-	-	-	-	-	-	-	-
BIT NAME	INDEX[7:0]							

12.5.2 MAILBOX DATA REGISTER

TABLE 12-4: MBX_DATA REGISTER

HOST OFFSET	01				8-Bit	**HOST SIZE**		
EC OFFSET	NA					**EC SIZE**		
POWER	VTR				00h 00h	nSYS_RST **DEFAULT** **&** VCC_PWRGD **DE-ASSERTION**		
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	-	-	-	-	-	-	-	-
BIT NAME	DATA[7:0]							

12.5.3 HOST-TO-EC MAILBOX REGISTER

TABLE 12-5: HOST-TO-EC MAILBOX REGISTER

HOST OFFSET	MBX_00h			8-Bit				**HOST SIZE**
EC OFFSET	100h			8-Bit				**EC SIZE**
POWER	VTR			00h				nSYS_RST **DEFAULT**
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
BIT NAME	HOST_EC_MBOX[7:0]							

HOST_EC_MBOX[7:0]

If enabled, an interrupt to the EC marked by the MBX bit in the GIRQ15 Source Register will be generated whenever the Host writes this register.

This register is cleared when written with FFh.

12.5.4 EC-TO-HOST MAILBOX REGISTER

TABLE 12-6: EC-TO-HOST MAILBOX REGISTER

HOST OFFSET	MBX_01h			8-Bit				**HOST SIZE**
EC OFFSET	104h			8-Bit				**EC SIZE**
POWER	VTR			00h				nSYS_RST **DEFAULT**
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	EC_HOST_MBOX[7:0]							

EC_HOST_MBOX[7:0]

An EC write to this register will set bit EC_WR in the SMI Interrupt Source Register to ‘1b’. If enabled, setting bit EC_WR to ‘1b’ generates a Host SMI.

This register is cleared when written with FFh.

© 2012-2018 Microchip Technology Inc.

12.5.5 SMI INTERRUPT SOURCE REGISTER

TABLE 12-7: SMI INTERRUPT SOURCE REGISTER

HOST OFFSET	MBX_02h					8-Bit	**HOST SIZE**	
EC OFFSET	108h					8-Bit	**EC SIZE**	
POWER	VTR					00h	nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	-
BIT NAME	EC_SWI[6:0]							EC_WR

EC_WR

This bit is set autonomously when the EC-to-Host Mailbox Register has been written. An SMI to the Host is generated when any bit in this register (EC_WR or any bit in EC_SWI[6:0]) is '1b' and the corresponding bit in the SMI Interrupt Mask Register register is '1b'.

This bit is automatically cleared by a read of the EC-to-Host Mailbox Register. The bit is also cleared when written with a '1b', by either the Host or the EC.

EC_SWI[6:0]

The EC can generate an SMI to the Host by writing any non-zero value to this field.

Each bit in this field is cleared when written with a '1b'.

12.5.6 SMI INTERRUPT MASK REGISTER

TABLE 12-8: SMI INTERRUPT MASK REGISTER

HOST OFFSET	MBX_03h					8-Bit	**HOST SIZE**	
EC OFFSET	10Ch					8-Bit	**EC SIZE**	
POWER	VTR					00h	nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	EC_SWI_EN[6:0]							EC_WR_EN

EC_WR_EN

If this bit is '1b', bit EC_WR in the SMI Interrupt Source Register is enabled.

EC_SWI_EN[6:0]

Each bit that is set to '1b' in this field enables the corresponding bit in the EC_SWI[6:0] field n the SMI Interrupt Source Register.

13.0 TWO PIN SERIAL PORT (UART)

13.1 General Description

The MEC1632 incorporates one full function UART. The UART is compatible with the 16450, the 16450 ACE registers and the 16C550A. The UART performs serial-to-parallel conversion on received characters and parallel-to-serial conversion on transmit characters. Two sets of baud rates are provided. When the 1.8432 MHz source clock is selected, standard baud rates from 50 to 115.2K are available. When the source clock is 20.27 MHz, baud rates from 79.18 K to 1.267 K are available. The character options are programmable for 1 start; 1, 1.5 or 2 stop bits; even, odd, sticky or no parity; and prioritized interrupts. The UART contains a programmable baud rate generator that is capable of dividing the input clock or crystal by a number from 1 to 65535. The UART is also capable of supporting the MIDI data rate. Refer to the Configuration Registers for information on disabling, powerdown and changing the base address of the UART. The interrupt from a UART is enabled by programming OUT2 of the UART to a logic “1”. OUT2 being a logic “0” disables that UART's interrupt. The UART is accessible by both the Host and the EC.

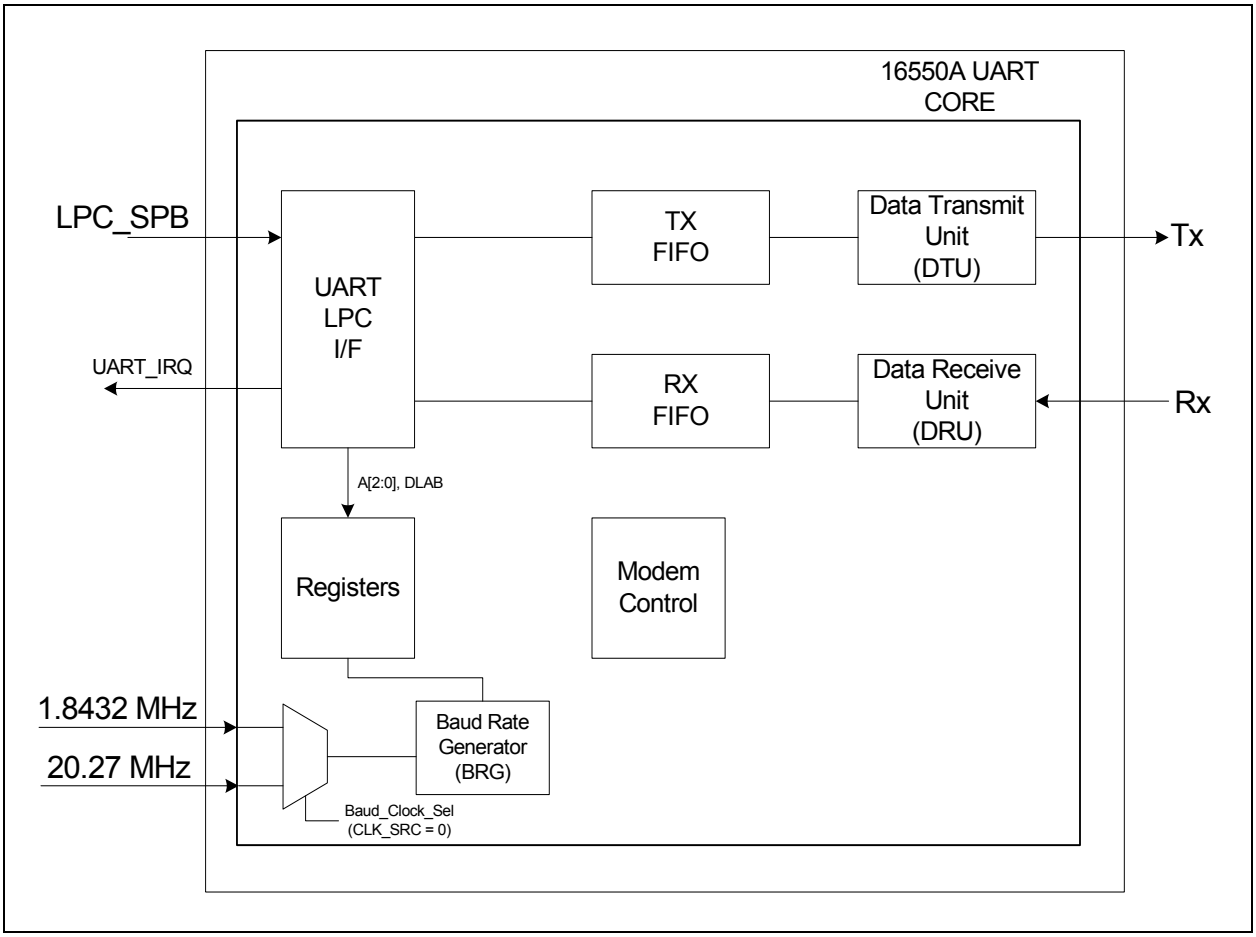
13.1.1 FEATURES

- Programmable word length, stop bits and parity
- Programmable baud rate generator
- Interrupt generator
- Loop-back mode
- Interface registers
- 16-byte Transmit FIFO
- 16-byte Receive FIFO
- Multiple clock sources
- VTR & VCC operation
- Pin Polarity control
- Low power sleep mode

© 2012-2018 Microchip Technology Inc.

13.1.2 BLOCK DIAGRAM

FIGURE 13-1: SERIAL PORT (UART) BLOCK DIAGRAM



13.1.3 BLOCK DIAGRAM SIGNAL LIST

TABLE 13-1: SERIAL PORT (UART) REGISTER INTERFACE PORT LIST

Signal Name	Direction	Description
UART_INT	Output	Host Interrupt routed to SERIRQ
EC IF	I/O Bus	Bus used for register access
MCLK	Input	Block operating clock
UART_RX	Input	UART Receive data pin
UART_TX	Output	UART Transmit data pin
UART_CLK	Input	UART Alternate clock pin (1.8432MHz)
nSYS_RST	input	VTR POR reset
nSIO_RESET	input	VCC POR reset

13.2 Power, Clocks and Reset

13.2.1 POWER DOMAIN

This block is powered by the VTR Power Supply.

See Section 7.1.1, "Power Configuration," on page 95 for details on power domains.

13.2.2 CLOCKS

Registers in this block are clocked at the LPC Bus Clock rate which is derived by the MCLK. Baud rates are derived from 1.8432MHz. The 1.8432MHz. is itself derived from either MCLK or sourced from UART_CLK Signal Pin Function.

See Section 7.4, "Clock Generator," on page 98 for details on clocks.

In order to maintain communicating with acceptable error, an accurate baud clock is required.

Note 13-1 The FREQ LOCK bit in the PCR Status and Control Register on page 133 must be set in order to insure an accurate baud clock when the CLK_SRC bit is '0' in the Configuration Select Register on page 238, the baud clock is internally sourced.

Note 13-2 When the CLK_SRC bit is '1' in the Configuration Select Register on page 238, the baud clock is externally sourced from the UART_CLK pin. The UART_CLK requires a frequency of 1.8432 MHz ± 2%.

13.2.3 RESET

Table 13-2 details the effect of nSYS_RST or nSIO_RESET on each of the runtime registers of the Serial Port.

TABLE 13-2: RESET FUNCTION TABLE

Register Signal	Reset Control	Reset State
Interrupt Enable Register	RESET	All bits low
Interrupt Identification Reg.		Bit 0 is high; Bits 1 - 7 low
FIFO Control		All bits low
Line Control Reg.		
MODEM Control Reg.		
Line Status Reg.		All bits low except 5, 6 high
MODEM Status Reg.		Bits 0 - 3 low; Bits 4 - 7 input
TXD1, TXD2		High
INTRPT (RCVR errs)	RESET/Read LSR	Low
INTRPT (RCVR Data Ready)	RESET/Read RBR	
INTRPT (THRE)	RESET/Read IIR/Write THR	
OUT2B	RESET	High
RTSB		
DTRB		
OUT1B		
RCVR FIFO	RESET/ FCR1*FCR0/_FCR0	All Bits Low
XMIT FIFO	RESET/ FCR1*FCR0/_FCR0	

The Runtime register can be configured to be reset on either nSYS_RST or nSIO_RESET. The POWER bit in the Configuration Select Register controls which reset effects the runtime registers. The Refer to Table 13-2 for effected registers and Section 7.0, "Power, Clocks, and Resets" for definitions of nSYS_RST on page 128 or nSIO_RESET on page 97.

See Section 7.6, "Reset Interface," on page 124 for details on reset.

© 2012-2018 Microchip Technology Inc.

13.3 Interrupts

13.3.1 EC INTERRUPT

The Two Pin Serial Port (UART) can generate an EC interrupt event. The interrupt source is routed onto the UART_RX bit in the GIRQ15 Source Register, and is a level sensitive, active high signal.

13.3.2 HOST INTERRUPT

The Two Pin Serial Port (UART) can generate a SIRQ event to the Host. See the Interrupt Enable Register (IER) on page 227 and the Interrupt Identification Register (IIR) on page 229. This interrupt is routed to the SIRQ block (see SERIRQ Configuration Registers on page 72).

13.4 Registers

The Two Pin Serial Port (UART) registers are located on the Host SPB.

Each instance of the Two Pin Serial Port (UART) has its own Logical Device Number, and Base Address as indicated in Table 13-3.

TABLE 13-3: Two Pin Serial Port (UART) BASE ADDRESS TABLE

Two Pin Serial Port (UART) Instance	LDN from (Table 4-2 on page 59)	AHB Base Address
UART	7h	FF_1C00h

Note 13-3 The Host LPC I/O addresses for each instance is selected via a Base Address Register (see Section 5.6.2, "Base Address Registers," on page 66). LPC access to configuration registers is through the Host Access Configuration Port (see Section 5.5.1, "Host Access Port," on page 65).

Table 13-4 is a register summary for one instance of the Two Pin Serial Port (UART). The LPC I/O address for each Run-Time Register is described below as an offset from its Base Address Register. Each EC address is indicated as an SPB Offset from its AHB base address. Each Configuration register access through the Host Access Port is via its LDN indicated in Table 13-3 on page 223 and its Host Access Port index which is described as “Host Config Index” in the tables below.

The following table summarizes the registers allocated for the Controller. The offset field in the following table is the offset from the Embedded Controller's (EC) Base Address.

TABLE 13-4: Two Pin Serial Port (UART) REGISTER SUMMARY

Register Name	Host I/O Access			DLAB (Note 13-4)	EC Interface			Notes
	Host I/O Index	SPB Off-set	Host Type		SPB Off-set	Byte Lane	EC Type	
Receive Buffer Register (RB)	00h	00h	R	0	00h	0	R	Note 13-5
Transmit Buffer Register (TB)	00h	00h	W	0	00h	0	W	Note 13-5
Programmable Baud Rate Generator (and Divisor) **(LSByte)**	00h	00h	R/W	1	00h	0	R/W	Note 13-5
Programmable Baud Rate Generator (and Divisor) **(MSByte)**	01h	01h	R/W	1	01h	1	R/W	Note 13-5
Interrupt Enable Register (IER)	01h	01h	R/W	0	01h	1	R/W	Note 13-5
FIFO Control Register (FCR)	02h	02h	W	X	02h	2	W	Note 13-5
Interrupt Identification Register (IIR)	02h	02h	R	X	02h	2	R	Note 13-5
Line Control Register (LCR)	03h	03h	R/W	X	03h	3	R/W	Note 13-5
Modem Control Register (MCR)	04h	04h	R/W	X	04h	0	R/W	Note 13-5
Line Status Register (LSR)	05h	05h	R	X	05h	1	R	Note 13-5
Modem Status Register (MSR)	06h	06h	R	X	06h	2	R	Note 13-5
Scratchpad Register (SCR)	07h	07h	R/W	X	07h	3	R/W	Note 13-5
Register Name	**Host Access**			**N/A**	**EC Interface**			
	Host Config. Index	**SPB Off-set**	**Host Type**		**EC Offset**	**Byte Lane**	**EC Type**	
Activate	30h	330h	R/W		330h	0	R/W	Note 13-5
Configuration Select Register	F0h	3F0h	R/W		3F0h	0	R/W	Note 13-5

Note 13-4 DLAB is Bit 7 of the Line Control Register

Note 13-5 Access to this register should be limited to 8-bit loads and stores. 16-bit or 32-bit stores will be blocked and 16-bit or 32-bit loads have unexpected results. JTAG Debugger access should indirect using peek_poke_arc macros described in IEEE Std 1149.1.

© 2012-2018 Microchip Technology Inc.

13.5 Register Summary

TABLE 13-5: REGISTER SUMMARY

Address (Note 13-6)	R/W	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADDR = 0 DLAB = 0	R	Receive Buffer r	Data Bit 7	Data Bit 6	Data Bit 5	Data Bit 4	Data Bit 3	Data Bit 2	Data Bit 1	Data Bit 0 (Note 13-7)
ADDR = 0 DLAB = 0	W	Transmitter Holding r	Data Bit 7	Data Bit 6	Data Bit 5	Data Bit 4	Data Bit 3	Data Bit 2	Data Bit 1	Data Bit 0
ADDR = 1 DLAB = 0	R/W	Interrupt Enable r	Reserved				Enable Modem Status Interrupt (EMSI)	Enable Receiver Line Sta-tus Inter-rupt (ELSI)	Enable Trans-mit-ter Holding Register Empty Interrupt (ETHREI)	Enable Received Data Avail-able Inter-rupt (ERDAI)
ADDR = 2	R	Interrupt Ident. r	FIFOs Enabled (Note 13-11)	FIFOs Enabled (Note 13-11)	Reserved		Interrupt ID Bit (Note 13-11)	Interrupt ID Bit	Interrupt ID Bit	"0" if Inter-rupt Pend-ing
ADDR = 2	W	FIFO Control r	RCVR Trig-ger MSB	RCVR Trig-ger LSB	Reserved		DMA Mode Select (Note 13-12)	XMIT FIFO Reset	RCVR FIFO Reset	FIFO Enable
ADDR = 3	R/W	Line Control r	Divisor Latch Access Bit (DLAB)	Set Break	Stick Parity	Even Parity Select (EPS)	Parity Enable (PEN)	Number of Stop Bits (STB)	Word Length Select Bit 1 (WLS1)	Word Length Select Bit 0 (WLS0)
ADDR = 4	R/W	MODEM Control r	**Reserved**			**Loop**	**OUT2 (Note 13-9)**	**OUT1 (Note 13-9)**	**Request to Send (RTS)**	**Data Ter-minal Ready (DTR)**
ADDR = 5	R/W	Line Status r	**Error in RCVR FIFO (Note 13-11)**	**Transmit-ter Empty (TEMT) (Note 13-8)**	**Transmit-ter Hold-ing Regis-ter (THRE)**	**Break Interrupt (BI)**	**Framing Error (FE)**	**Parity Error (PE)**	**Overrun Error (OE)**	**Data Ready (DR)**
ADDR = 6	R/W	MODEM Status r	Data Car-rier Detect (DCD)	Ring Indica-tor (RI)	Data Set Ready (DSR)	Clear to Send (CTS)	Delta Data Carrier Detect (DDCD)	Trailing Edge Ring Indicator (TERI)	Delta Data Set Ready (DDSR)	Delta Clear to Send (DCTS)
ADDR = 7	R/W	Scratch r (Note 13-10)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADDR = 0 DLAB = 1	R/W	Divisor Latch (LS)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADDR = 1 DLAB = 1	R/W	Divisor Latch (MS)	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

UART Register Summary Notes:

Note 13-6 DLAB is Bit 7 of the Line Control Register (ADDR = 3).

Note 13-7 Bit 0 is the least significant bit. It is the first bit serially transmitted or received.

Note 13-8 When operating in the XT mode, this bit will be set any time that the transmitter shift register is empty.

Note 13-9 This bit no longer has a pin associated with it.

Note 13-10 When operating in the XT mode, this register is not available.

Note 13-11 These bits are always zero in the non-FIFO mode.

Note 13-12 Writing a one to this bit has no effect. DMA modes are not supported in this chip.

© 2012-2018 Microchip Technology Inc.

13.6 Detailed Description of Accessible Runtime Registers

13.6.1 RECEIVE BUFFER REGISTER (RB)

TABLE 13-6: RECEIVE BUFFER (RB)

HOST OFFSET	0h (DLAB=0)					8-bit	**HOST SIZE**	
EC OFFSET	0h (DLAB=0)					8-bit	**EC SIZE**	
POWER	VCC or VTR					00h	nSYS_RST or nSIO_RESET **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R	R	R	R	R	R
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Received Data byte [7:0]							

RECEIVED DATA BYTE

This register holds the received incoming data byte. Bit 0 is the least significant bit, which is transmitted and received first. Received data is double buffered; this uses an additional shift register to receive the serial data stream and convert it to a parallel 8 bit word which is transferred to the Receive Buffer register. The shift register is not accessible.

13.6.2 TRANSMIT BUFFER REGISTER (TB)

TABLE 13-7: TRANSMIT BUFFER (TB)

HOST OFFSET	0h (DLAB=0)					8-bit	**HOST SIZE**	
EC OFFSET	0h (DLAB=0)					8-bit	**EC SIZE**	
POWER	VCC or VTR					00h	nSYS_RST or VCC Power Good **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	W	W	W	W	W	/W	W	W
EC TYPE		W	W	W	W	W	W	W
BIT NAME	Transmit data byte [7:0]							

TRANSMIT DATA BYTE

This register contains the data byte to be transmitted. The transmit buffer is double buffered, utilizing an additional shift register (not accessible) to convert the 8 bit data word to a serial format. This shift register is loaded from the Transmit Buffer when the transmission of the previous byte is complete.

© 2012-2018 Microchip Technology Inc.

13.6.3 INTERRUPT ENABLE REGISTER (IER)

TABLE 13-8: INTERRUPT ENABLE (IER)

HOST OFFSET	1h (DLAB=0)					8-bit	**HOST SIZE**	
EC OFFSET	1h (DLAB=0)					8-bit	**EC SIZE**	
POWER	VCC or VTR					00h	nSYS_RST or nSIO_RESET **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Reserved				EMSI	ELSI	ETHREI	ERDAI

The lower four bits of this register control the enables of the five interrupt sources of the Serial Port interrupt. It is possible to totally disable the interrupt system by resetting bits 0 through 3 of this register. Similarly, setting the appropriate bits of this register to a high, selected interrupts can be enabled. Disabling the interrupt system inhibits the Interrupt Identification Register and disables any Serial Port interrupt out of the MEC1632. All other system functions operate in their normal manner, including the Line Status and MODEM Status Registers. The contents of the Interrupt Enable Register are described below.

ERDAI

This bit enables the Received Data Available Interrupt (and timeout interrupts in the FIFO mode) when set to logic "1".

ETHREI

This bit enables the Transmitter Holding Register Empty Interrupt when set to logic "1".

ELSI

This bit enables the Received Line Status Interrupt when set to logic "1". The error sources causing the interrupt are Overrun, Parity, Framing and Break. The Line Status Register must be read to determine the source.

EMSI

This bit enables the MODEM Status Interrupt when set to logic "1". This is caused when one of the Modem Status Register bits changes state.

13.6.4 FIFO CONTROL REGISTER (FCR)

TABLE 13-9: FIFO CONTROL (FCR)

HOST OFFSET	02h					8-bit	**HOST SIZE**	
EC OFFSET	02h					8-bit	**EC SIZE**	
POWER	VCC or VTR					00h	nSYS_RST or VCC Power Good **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	W	W	W	W	W	W	W	W
EC TYPE	W	W	W	W	W	W	W	W
BIT NAME	RECV FIFO Trigger Level		Reserved			Clear XMIT FIFO	Clear RECV FIFO	EXRF

This is a write only register at the same location as the IIR.

Note 13-13 DMA is not supported.

EXRF

Enable XMIT and RECV FIFO. Setting this bit to a logic "1" enables both the XMIT and RCVR FIFOs. Clearing this bit to a logic "0" disables both the XMIT and RCVR FIFOs and clears all bytes from both FIFOs. When changing from FIFO Mode to non-FIFO (16450) mode, data is automatically cleared from the FIFOs. This bit must be a 1 when other bits in this register are written to or they will not be properly programmed.

CLEAR RECV FIFO

Setting this bit to a logic "1" clears all bytes in the RCVR FIFO and resets its counter logic to “0”. The shift register is not cleared. This bit is self-clearing.

CLEAR XMIT FIFO

Setting this bit to a logic "1" clears all bytes in the XMIT FIFO and resets its counter logic to “0”. The shift register is not cleared. This bit is self-clearing.

RECV FIFO TRIGGER LEVEL

These bits are used to set the trigger level for the RCVR FIFO interrupt.

TABLE 13-10: RECV FIFO TRIGGER LEVEL

Bit 7	Bit 6	RECV FIFO Trigger Level (Bytes)
0	0	1
	1	4
1	0	8
	1	14

© 2012-2018 Microchip Technology Inc.

13.6.5 INTERRUPT IDENTIFICATION REGISTER (IIR)

TABLE 13-11: INTERRUPT IDENTIFICATION (IIR)

HOST OFFSET	02h					8-bit	**HOST SIZE**	
EC OFFSET	02h					8-bit	**EC SIZE**	
POWER	VCC or VTR					01h	nSYS_RST or nSIO_RESET **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R	R	R	R	R	R
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	FIFO_En		Reserved		IntID			IPEND

By accessing this register, the host CPU can determine the highest priority interrupt and its source. Four levels of priority interrupt exist. They are in descending order of priority:

1. Receiver Line Status (highest priority)
2. Received Data Ready
3. Transmitter Holding Register Empty
4. MODEM Status (lowest priority)

Information indicating that a prioritized interrupt is pending and the source of that interrupt is stored in the Interrupt Identification Register (refer to Table 13-12). When the CPU accesses the IIR, the Serial Port freezes all interrupts and indicates the highest priority pending interrupt to the CPU. During this CPU access, even if the Serial Port records new interrupts, the current indication does not change until access is completed. The contents of the IIR are described below.

IPEND

This bit can be used in either a hardwired prioritized or polled environment to indicate whether an interrupt is pending. When bit 0 is a logic "0", an interrupt is pending and the contents of the IIR may be used as a pointer to the appropriate internal service routine. When bit 0 is a logic "1", no interrupt is pending.

INTID

These three bits of the IIR are used to identify the highest priority interrupt pending as indicated by Table 13-12. In non-FIFO mode, Bit[3] is a logic “0”. In FIFO mode Bit[3] is set along with Bit[2] when a timeout interrupt is pending.

TABLE 13-12: INTERRUPT CONTROL TABLE

FIFO Mode Only	Interrupt Identification Register			Interrupt Set and Reset Functions			
Bit 3	**Bit 2**	**Bit 1**	**Bit 0**	**Priority Level**	**Interrupt Type**	**Interrupt Source**	**Interrupt Reset Control**
0	0	0	1	-	None	None	-
	1	1	0	Highest	Receiver Line Status	Overrun Error, Parity Error, Framing Error or Break Interrupt	Reading the Line Status Register
		0		Second	Received Data Available	Receiver Data Available	Read Receiver Buffer or the FIFO drops below the trigger level.
1					Character Timeout Indication	No Characters Have Been Removed From or Input to the RCVR FIFO during the last 4 Char times and there is at least 1 char in it during this time	Reading the Receiver Buffer Register
0	0	1		Third	Transmitter Holding Register Empty	Transmitter Holding Register Empty	Reading the IIR Register (if Source of Interrupt) or Writing the Transmitter Holding Register
	0	0		Fourth	MODEM Status	Clear to Send or Data Set Ready or Ring Indicator or Data Carrier Detect	Reading the MODEM Status Register

FIFO_EN

These two bits are set when the FIFO CONTROL Register bit 0 equals 1.

13.6.6 LINE CONTROL REGISTER (LCR)

TABLE 13-13: LINE CONTROL (LCR)

HOST OFFSET	03h			8-bit			**HOST SIZE**	
EC OFFSET	03h			8-bit			**EC SIZE**	
POWER	VCC or VTR			00h			nSYS_RST or nSIO_RESET **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	DLAB	Break Control	Stick Parity	Parity Select	Enable Parity	Stop Bits	Word Length	

© 2012-2018 Microchip Technology Inc.

This register contains the format information of the serial line. The bit definitions are:

WORD LENGTH

These two bits specify the number of bits in each transmitted or received serial character. The encoding of bits 0 and 1 is as follows:

STOP BITS

This bit specifies the number of stop bits in each transmitted or received serial character. Table 13-14 summarizes the information.

TABLE 13-14: STOP BITS

Bit 2	Word Length	Number of Stop Bits
0	--	1
1	5 bits	1.5
	6 bits	2
	7 bits	
	8 bits	

Note 13-14 The receiver will ignore all stop bits beyond the first, regardless of the number used in transmitting.

TABLE 13-15: SERIAL CHARACTER

Bit 1	Bit 0	Word Length
0	0	5 Bits
0	1	6 Bits
1	0	7 Bits
1	1	8 Bits

The Start, Stop and Parity bits are not included in the word length.

ENABLE PARITY

Parity Enable bit. When bit 3 is a logic "1", a parity bit is generated (transmit data) or checked (receive data) between the last data word bit and the first stop bit of the serial data. (The parity bit is used to generate an even or odd number of 1s when the data word bits and the parity bit are summed).

PARITY SELECT

Even Parity Select bit. When bit 3 is a logic "1" and bit 4 is a logic "0", an odd number of logic "1"'s is transmitted or checked in the data word bits and the parity bit. When bit 3 is a logic "1" and bit 4 is a logic "1" an even number of bits is transmitted and checked.

STICK PARITY

Stick Parity bit. When parity is enabled it is used in conjunction with bit 4 to select Mark or Space Parity. When LCR bits 3, 4 and 5 are 1 the Parity bit is transmitted and checked as a 0 (Space Parity). If bits 3 and 5 are 1 and bit 4 is a 0, then the Parity bit is transmitted and checked as 1 (Mark Parity). If bit 5 is 0 Stick Parity is disabled.

Bit 3 is a logic "1" and bit 5 is a logic "1", the parity bit is transmitted and then detected by the receiver in the opposite state indicated by bit 4.

BREAK CONTROL

Set Break Control bit. When bit 6 is a logic "1", the transmit data output (TXD) is forced to the Spacing or logic "0" state and remains there (until reset by a low level bit 6) regardless of other transmitter activity. This feature enables the Serial Port to alert a terminal in a communications system.

DLAB

Divisor Latch Access Bit (DLAB). It must be set high (logic "1") to access the Divisor Latches of the Baud Rate Generator during read or write operations. It must be set low (logic "0") to access the Receiver Buffer Register, the Transmitter Holding Register, or the Interrupt Enable Register.

© 2012-2018 Microchip Technology Inc.

13.6.7 MODEM CONTROL REGISTER (MCR)

TABLE 13-16: MODEM CONTROL (MCR)

HOST OFFSET	04h			8-bit			**HOST SIZE**	
EC OFFSET	04h			8-bit			**EC SIZE**	
POWER	VCC or VTR			00h			nSYS_RST or nSIO_RESET **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Reserved			LOOP-BACK	OUT2	OUT1	RTS	DTR

This 8-bit register controls the interface with the MODEM or data set (or device emulating a MODEM). The contents of the MODEM control register are described below.

DTR

This bit controls the Data Terminal Ready (nDTR) output. When bit 0 is set to a logic "1", the nDTR output is forced to a logic "0". When bit 0 is a logic "0", the nDTR output is forced to a logic "1".

RTS

This bit controls the Request To Send (nRTS) output. Bit 1 affects the nRTS output in a manner identical to that described above for bit 0.

OUT1

This bit controls the Output 1 (OUT1) bit. This bit does not have an output pin and can only be read or written by the CPU.

OUT2

Output 2 (OUT2). This bit is used to enable an UART interrupt. When OUT2 is a logic "0", the serial port interrupt output is forced to a high impedance state - disabled. When OUT2 is a logic "1", the serial port interrupt outputs are enabled.

LOOPBACK

This bit provides the loopback feature for diagnostic testing of the Serial Port. When bit 4 is set to logic "1", the following occur:

1. The TXD is set to the Marking State (logic "1").
2. The receiver Serial Input (RXD) is disconnected.
3. The output of the Transmitter Shift Register is "looped back" into the Receiver Shift Register input.
4. All MODEM Control inputs (nCTS, nDSR, nRI and nDCD) are disconnected.
5. The four MODEM Control outputs (nDTR, nRTS, OUT1 and OUT2) are internally connected to the four MODEM Control inputs (nDSR, nCTS, RI, DCD).
6. The Modem Control output pins are forced inactive high.
7. Data that is transmitted is immediately received.

This feature allows the processor to verify the transmit and receive data paths of the Serial Port. In the diagnostic mode, the receiver and the transmitter interrupts are fully operational. The MODEM Control Interrupts are also operational but the interrupts' sources are now the lower four bits of the MODEM Control Register instead of the MODEM Control inputs. The interrupts are still controlled by the Interrupt Enable Register.

© 2012-2018 Microchip Technology Inc.

13.6.8 LINE STATUS REGISTER (LSR)

TABLE 13-17: LINE STATUS (LSR)

HOST OFFSET	05h						8-bit	**HOST SIZE**
EC OFFSET	05h						8-bit	**EC SIZE**
POWER	VCC or VTR						60h	nSYS_RST or nSIO_RESET **DEFAULT**
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R	R	R	R	R	R
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	FIFO Error	Trans-mit Error	Trans-mit Empty	Break Interrupt	Frame Error	Parity Error	Overrun Error	Data Ready

DATA READY

Data Ready (DR). It is set to a logic "1" whenever a complete incoming character has been received and transferred into the Receiver Buffer Register or the FIFO. Bit 0 is reset to a logic "0" by reading all of the data in the Receive Buffer Register or the FIFO.

OVERRUN ERROR

Overrun Error (OE). Bit 1 indicates that data in the Receiver Buffer Register was not read before the next character was transferred into the register, thereby destroying the previous character. In FIFO mode, an overrun error will occur only when the FIFO is full and the next character has been completely received in the shift register, the character in the shift register is overwritten but not transferred to the FIFO. The OE indicator is set to a logic "1" immediately upon detection of an overrun condition, and reset whenever the Line Status Register is read.

PARITY ERROR

Parity Error (PE). Bit 2 indicates that the received data character does not have the correct even or odd parity, as selected by the even parity select bit. The PE is set to a logic "1" upon detection of a parity error and is reset to a logic "0" whenever the Line Status Register is read. In the FIFO mode this error is associated with the particular character in the FIFO it applies to. This error is indicated when the associated character is at the top of the FIFO.

FRAME ERROR

Framing Error (FE). Bit 3 indicates that the received character did not have a valid stop bit. Bit 3 is set to a logic "1" whenever the stop bit following the last data bit or parity bit is detected as a zero bit (Spacing level). The FE is reset to a logic "0" whenever the Line Status Register is read. In the FIFO mode this error is associated with the particular character in the FIFO it applies to. This error is indicated when the associated character is at the top of the FIFO. The Serial Port will try to resynchronize after a framing error. To do this, it assumes that the framing error was due to the next start bit, so it samples this 'start' bit twice and then takes in the 'data'.

BREAK INTERRUPT

Break Interrupt (BI). Bit 4 is set to a logic “1” whenever the received data input is held in the Spacing state (logic “0”) for longer than a full word transmission time (that is, the total time of the start bit + data bits + parity bits + stop bits). The BI is reset after the CPU reads the contents of the Line Status Register. In the FIFO mode this error is associated with the particular character in the FIFO it applies to. This error is indicated when the associated character is at the top of the FIFO. When break occurs only one zero character is loaded into the FIFO. Restarting after a break is received, requires the serial data (RXD) to be logic “1” for at least 1/2 bit time.

Bits 1 through 4 are the error conditions that produce a Receiver Line Status Interrupt BIT 3.

Note 13-15 whenever any of the corresponding conditions are detected and the interrupt is enabled.

© 2012-2018 Microchip Technology Inc.

TRANSMIT EMPTY

Transmitter Holding Register Empty (THRE). Bit 5 indicates that the Serial Port is ready to accept a new character for transmission. In addition, this bit causes the Serial Port to issue an interrupt when the Transmitter Holding Register interrupt enable is set high. The THRE bit is set to a logic "1" when a character is transferred from the Transmitter Holding Register into the Transmitter Shift Register. The bit is reset to logic "0" whenever the CPU loads the Transmitter Holding Register. In the FIFO mode this bit is set when the XMIT FIFO is empty, it is cleared when at least 1 byte is written to the XMIT FIFO. Bit 5 is a read only bit.

TRANSMIT ERROR

Transmitter Empty (TEMT). Bit 6 is set to a logic "1" whenever the Transmitter Holding Register (THR) and Transmitter Shift Register (TSR) are both empty. It is reset to logic "0" whenever either the THR or TSR contains a data character. Bit 6 is a read only bit. In the FIFO mode this bit is set whenever the THR and TSR are both empty,

FIFO ERROR

This bit is permanently set to logic "0" in the 450 mode. In the FIFO mode, this bit is set to a logic "1" when there is at least one parity error, framing error or break indication in the FIFO. This bit is cleared when the LSR is read if there are no subsequent errors in the FIFO.

13.6.9 MODEM STATUS REGISTER (MSR)

TABLE 13-18: MODEM STATUS (MSR)

HOST ADDRESS	06h			8-bit			**HOST SIZE**	
EC OFFSET	06h			8-bit			**EC SIZE**	
POWER	VCC or VTR			xxxx0000b			nSYS_RST or nSIO_RESET **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R	R	R	R	R	R
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	DCD#	RI#	DSR	CTS	DCD	RI	DSR	CTS

This 8 bit register provides the current state of the control lines from the MODEM (or peripheral device). In addition to this current state information, four bits of the MODEM Status Register (MSR) provide change information.

These bits are set to logic "1" whenever a control input from the MODEM changes state. They are reset to logic "0" whenever the MODEM Status Register is read.

CTS

Delta Clear To Send (DCTS). Bit 0 indicates that the nCTS input to the chip has changed state since the last time the MSR was read.

DSR

Delta Data Set Ready (DDSR). Bit 1 indicates that the nDSR input has changed state since the last time the MSR was read.

RI

Trailing Edge of Ring Indicator (TERI). Bit 2 indicates that the nRI input has changed from logic "0" to logic "1".

DCD

Delta Data Carrier Detect (DDCD). Bit 3 indicates that the nDCD input to the chip has changed state.

Note 13-16 Whenever bit 0, 1, 2, or 3 is set to a logic "1", a MODEM Status Interrupt is generated.

© 2012-2018 Microchip Technology Inc.

CTS

This bit is the complement of the Clear To Send (nCTS) input. If bit 4 of the MCR is set to logic "1", this bit is equivalent to nRTS in the MCR.

DSR

This bit is the complement of the Data Set Ready (nDSR) input. If bit 4 of the MCR is set to logic "1", this bit is equivalent to DTR in the MCR.

RI#

This bit is the complement of the Ring Indicator (nRI) input. If bit 4 of the MCR is set to logic "1", this bit is equivalent to OUT1 in the MCR.

DCD

This bit is the complement of the Data Carrier Detect (nDCD) input. If bit 4 of the MCR is set to logic "1", this bit is equivalent to OUT2 in the MCR.

APPLICATION NOTE: The Modem Status Register (MSR) only provides the current state of the UART MODEM control lines in Loopback Mode. The MEC1632 does not support external connections for the MODEM Control inputs (nCTS, nDSR, nRI and nDCD) or for the four MODEM Control outputs (nDTR, nRTS, OUT1 and OUT2).

13.6.10 SCRATCHPAD REGISTER (SCR)

TABLE 13-19: SCRATCH PAD (SCR)

HOST OFFSET	07h			8-bit			**HOST SIZE**	
EC OFFSET	07h			8-bit			**EC SIZE**	
POWER	VCC or VTR			00h			nSYS_RST or nSIO_RESET **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Scratch							

SCRATCH

This 8 bit read/write register has no effect on the operation of the Serial Port. It is intended as a scratchpad register to be used by the programmer to hold data temporarily.

13.6.11 PROGRAMMABLE BAUD RATE GENERATOR (AND DIVISOR)

TABLE 13-20: PROGRAMMABLE BAUD RATE GENERATOR (AND DIVISOR)

HOST OFFSET	BYTE1: 01h (DLAB = 1) BYTE0: 00h (DLAB = 1)			8-bit			**HOST SIZE**	
EC OFFSET	BYTE1: 01h (DLAB = 1) BYTE0: 00h (DLAB = 1)			8-bit			**EC SIZE**	
POWER	VCC or VTR			0000h			nSYS_RST or nSIO_RESET **DEFAULT**	
BUS	LPC SPB							
BYTE1 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Baud_Clock_Sel	Baud_Rate_Divisor						
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Baud_Rate_Divisor[7:0]							

BAUD_CLOCK_SEL

If the CLK_SRC bit is '0' and the Baud_Clock_Sel bit is '0,' the 1.8432MHz clock is used to generate the baud clock. Table 13-21 shows some baud rates that can be generated with this clock. The CLK_SRC bit is D0 in the UART Logical Device configuration register offset 0xF0.

If the CLK_SRC bit is '0' and the Baud_Clock_Sel bit is '1,' MCLK is used to generate the baud clock. Table 13-22 shows some baud rates that can be generated with this clock.

If the CLK_SRC bit is '1,' the Baud_Clock_Sel bit as no effect.

BAUD_RATE_DIVISOR

The Serial Port contains a programmable Baud Rate Generator that is capable of dividing the internal clock source by any divisor from 1 to 65535. The clock source is either a 1.8432MHz clock derived from MCLK or MCLK directly. The output frequency of the Baud Rate Generator is 16x the Baud rate. Two eight bit latches store the divisor in 16 bit binary format. These Divisor Latches must be loaded during initialization in order to insure desired operation of the Baud Rate Generator. Upon loading either of the Divisor Latches, a 16 bit Baud counter is immediately loaded. This prevents long counts on initial load. If a 0 is loaded into the BRG registers, the output divides the clock by the number 3. If a 1 is loaded, the output is the inverse of the input oscillator. If a two is loaded, the output is a divide by 2 signal with a 50% duty cycle. If a 3 or greater is loaded, the output is low for 2 bits and high for the remainder of the count.

Table 13-21 and Table 13-22 shows the baud rates possible.

TABLE 13-21: UART BAUD RATES (1.8432MHZ SOURCE)

Desired Baud Rate	Divisor Used to Generate 16X Clock
50	2304
75	1536
110	1047
134.5	857
150	768
300	384
600	192

© 2012-2018 Microchip Technology Inc.

TABLE 13-21: UART BAUD RATES (1.8432MHZ SOURCE) (CONTINUED)

Desired Baud Rate	Divisor Used to Generate 16X Clock
1200	96
1800	64
2000	58
2400	48
3600	32
4800	24
7200	16
9600	12
19200	6
38400	3
57600	2
115200	1

TABLE 13-22: UART BAUD RATES (MCLK SOURCE)

Desired Baud Rate	BAUD_CLOCK_SEL	Divisor Used to Generate 16X Clock
79180	1	16
105573	1	12
115170	1	11
126688	1	10
140764	1	9
158359	1	8
180982	1	7
211146	1	6
253375	1	5
316719	1	4
422292	1	3
633438	1	2
1266875	1	1

13.7 Detailed Description of Configuration Registers

13.7.1 ACTIVATE

TABLE 13-23: ACTIVATE REGISTER

HOST OFFSET	30h					8-bit	**HOST SIZE**	
EC OFFSET	330h					32-bit	**EC SIZE**	
POWER	VTR					00b	nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R	R	R	R	R	R/W
EC TYPE	R	R	R	R	R	R	R	R/W
BIT NAME	Reserved							Activate

ACTIVATE

When this bit is 1, the UART logical device is powered and functional. When this bit is 0, the UART logical device is powered down and inactive.

13.7.2 CONFIGURATION

TABLE 13-24: CONFIGURATION SELECT REGISTER

HOST OFFSET	F0h				8-bit		**HOST SIZE**	
EC OFFSET	3F0h				8-bit		**EC SIZE**	
POWER	VTR				00b		nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R	R	R	R	R	R/W
EC TYPE	R	R	R	R	R	R/W	R/W	R/W
BIT NAME	Reserved					Polarity	Power	CLK_SRC

CLK_SRC

When this bit is 0, the UART clock is derived from the internal 20 MHz Oscillator. When this bit is 1, the UART clock is derived from an external clock source.

POWER

When this bit is 1, the UART Runtime Registers (the registers at offsets 0h through 7h from the base of the UART Logical Device) are controlled by VCC. They are set to their POR defaults on a nSIO_RESET. In addition, pins associated with the UART are powered down and place in a High-Z state on nSIO_RESET.

When this bit is 0, the UART Runtime Registers are controlled by VTR. They are set to their POR defaults on an nSYS-_RST. In addition, the state of the UART pins is controlled by VTR.

POLARITY

When the Polarity bit is asserted (‘1’), the UART_TX and UART_RX pins functions are inverted. When the Polarity bit is not asserted (default), the UART_TX and UART_RX pins functions are not inverted.

© 2012-2018 Microchip Technology Inc.

13.8 Sleep Enable/ Clock Request Power State Controls

TABLE 13-25: UART BLOCK CLOCK GATING BEHAVIOR

Activate	External Sleep Input	Block Idle Status (Note 13-17)	Clock Required Status Output	State	Description
0	X	X	0	DISABLED	Two Pin Serial Port (UART) is disabled by firmware and the core clock is not needed. **Note:** It is up to the host to ensure that the block is not in use before the Activate bit is de-asserted.
1	0	NOT IDLE	1	NORMAL OPERATION	The block is neither disabled by firmware nor commanded to sleep. The UART enters a low power state while monitoring for a start bit on Rx.
	1	NOT IDLE	1	PREPARING TO SLEEP	A sleep command has been asserted but the core clock is still required because the block is not idle.
		IDLE	0	SLEEPING	A sleep command has been asserted, the block is idle and the core clock can be stopped.

Note 13-17 the Two Pin Serial Port (UART) 'idle' status is defined in Table 13-26.

TABLE 13-26: UART IDLE STATUS

Transmitter Active?	Receiver Active?	Character Time-out Active	Status
NO	NO	NO	Idle
YES	X	YES	Not Idle
X	YES	YES	
X	X	YES	

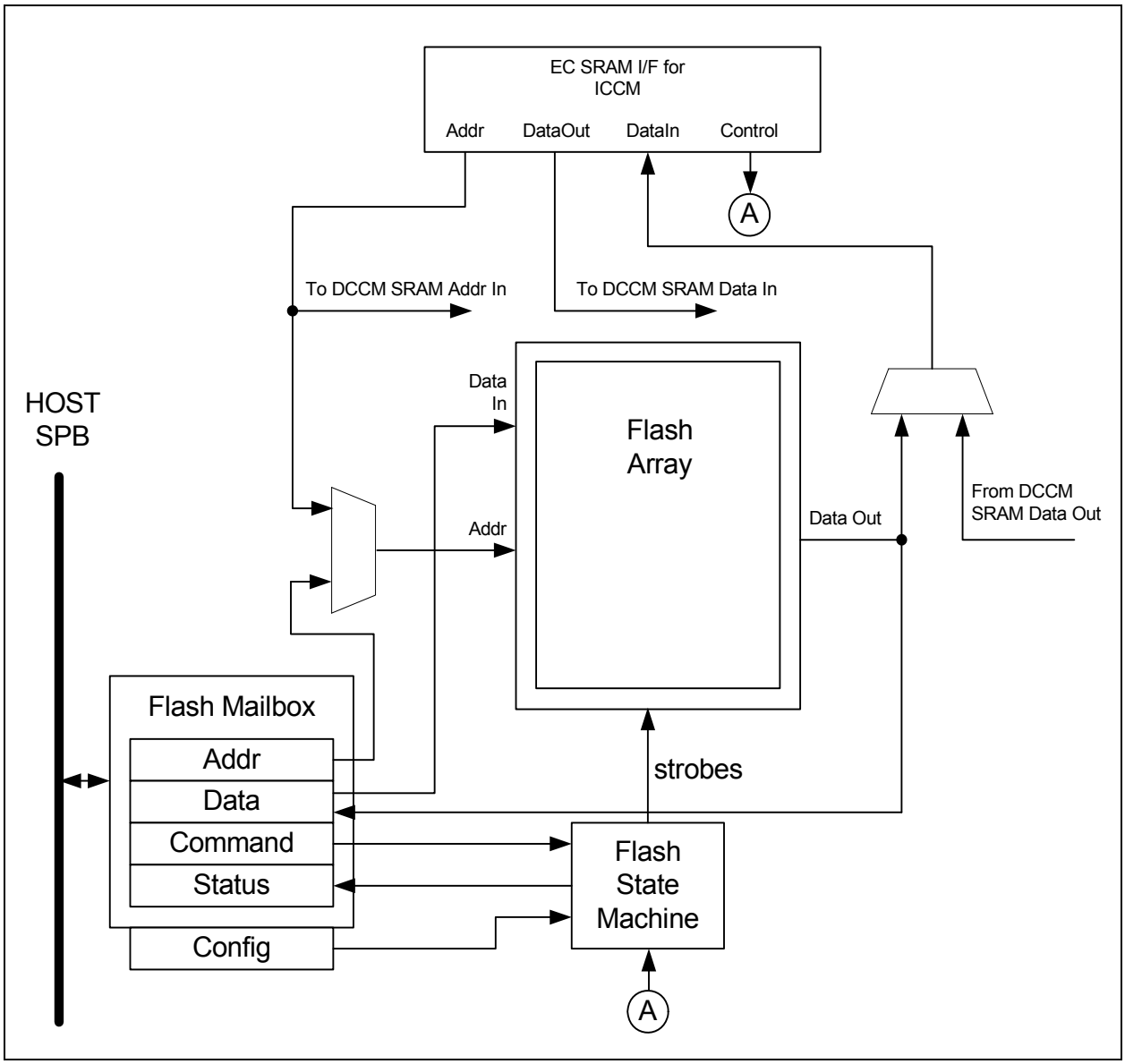
14.0 EMBEDDED FLASH SUBSYSTEM

14.1 General Description

The MEC1632 Embedded Flash Subsystem includes a 192KB embedded Flash memory. The memory appears in the system AHB address space and can store both instructions and data. The Flash memory can be programmed by the Embedded Controller, by the Host via LPC, through the JTAG interface, and by using the Gang Programmer Interface.

14.2 Block Diagram

FIGURE 14-1: EMBEDDED FLASH BLOCK DIAGRAM



© 2012-2018 Microchip Technology Inc.

14.3 Port List

TABLE 14-1: EMBEDDED FLASH PORT LIST

Signal Name	Direction	Description
ARC_CLK_DISABLE.	INPUT	Indicates the ARC is sleeping
FLASH_SLP_EN	INPUT	Indicates that the system is trying to shut down the ring oscillator
FLASH_CLK_REQ	OUTPUT	Indicates when the flash is ready to sleep (have clocks removed)

14.4 Power, Clocks and Reset

14.4.1 POWER DOMAIN

This block is powered by the VTR Power Supply.

See Section 7.0, "Power, Clocks, and Resets," on page 95 for details on power domains.

14.4.2 CLOCKS

This block uses the 20.27 MHz MCLK. All Flash signal timing is derived from the MCLK.

See Section 7.4, "Clock Generator," on page 98 for details on clocks.

14.4.2.1 Clock Idle

The Embedded Flash controller will keep the internal oscillator operating as long as the controller is not in the Standby state. This permits the controller to complete any program or erase operation even though the Embedded Controller may be in its sleep state.

The ARC, Section 7.0, "Power, Clocks, and Resets," on page 95, and the Embedded Flash Subsystem interact to determine when the MEC1632 can stop the oscillator.

FIGURE 14-2: FLASH_CLK_REQ INTERFACE

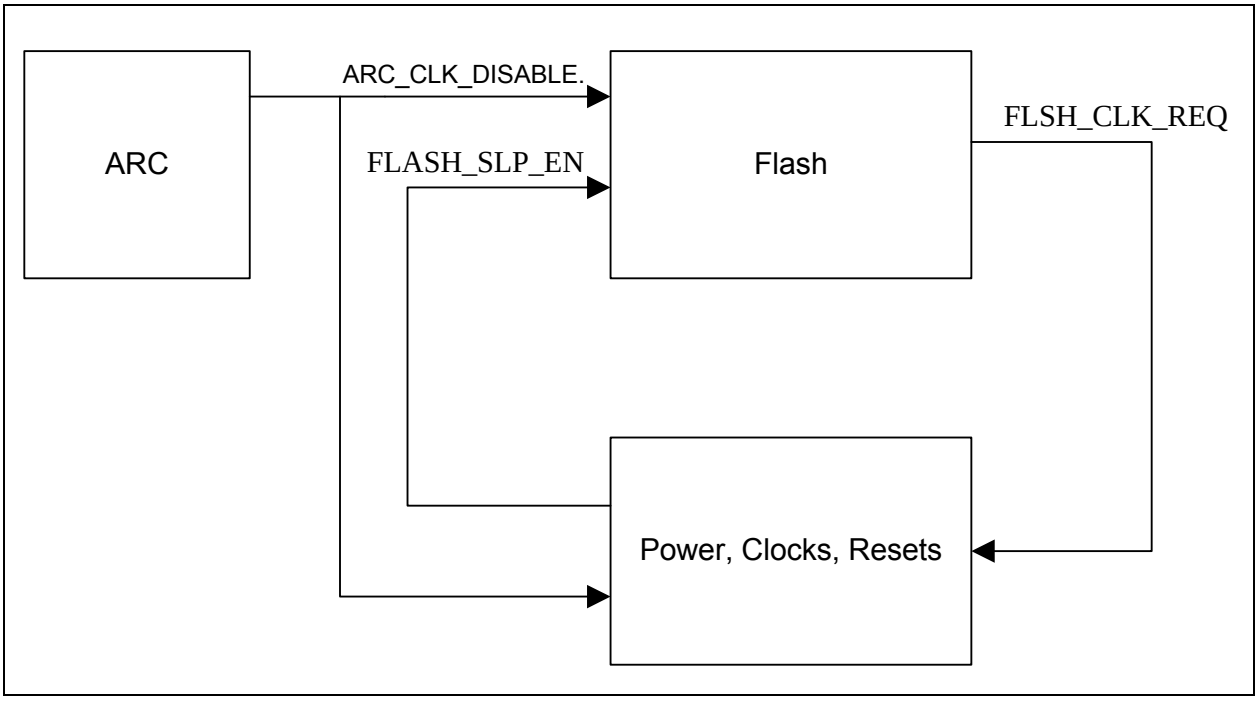


TABLE 14-2: Embedded Flash Subsystem POWER MANAGEMENT

ARC Sleeping ARC_CLK_DISABLE. on page 241	Flash Mode Section 14.9	FLASH_SLP_EN	FLASH_CLK_REQ	Power	Description
0	X	0	1	High power state	ARC not sleeping clocks required and block is not commanded to sleep.
X	Not Standby Mode	X	1	High power state	Force clocks for Host or JTAG access (i.e. while ARC is sleeping)
1		1	1	High power state	ARC is sleeping but the Flash clock is still required because the block is not idle.
1	Standby Mode	1	0	Low power state	A sleep command has been asserted, the block is idle and the core clocks are stopped.

14.4.3 RESET

This block is reset by nSYS_RST. Following a reset, all registers are set to their default values, and the internal state machines are reset to the standby state.

See Section 7.6, "Reset Interface," on page 124 for details on reset.

The JTAG interface registers referenced in this chapter have an asynchronous reset. See Note 14-1 on page 243.

14.4.4 TRACKING FLASH PROGRAM OR ERASE ACTIVITY

When the Embedded Flash Command Register is placed in Program Mode or Erase Mode, the FLASH bit is asserted in the Power-Fail and Reset Status Register.

APPLICATION NOTE: The purpose of the FLASH bit in the Power-Fail and Reset Status Register is to provide the EC with status. Once this bit is asserted by hardware, EC firmware should clear the FLASH bit as soon as possible after every Program Mode or Erase Mode operation. EC software can detect unexpected events which may indicate flash corruption. For example, VTR POR, VCC POR, LRESET#, and VCC_PWRGD transitions, as well as, completion of host flash erase or programming are example events when firmware should examining & clear the FLASH bit in the Power-Fail and Reset Status Register can be useful.

If a reset occurs, the software reset handler can examine the FLASH bit to determine if the Flash memory might be corrupted. Corruption can occur if Flash programming or erasure is interrupted by a reset.

If VCC power is removed, or if LRESET# is asserted, the LPC interface becomes inactive and the Embedded Flash controller must ensure that the flash subsystem is in a consistent state. When the LPC bus is inoperative the EC should have full access to the Embedded Flash. Table 14-3, "VCC PWRGD and LRESET# Behavior" describes the different possible states and the response.

© 2012-2018 Microchip Technology Inc.

TABLE 14-3: VCC PWRGD AND LRESET# BEHAVIOR

Embedded Flash Configuration Register	Embedded Flash Command Register	Action on LRESET# Asserted or VCC PWRGD De-Asserted
Host_Ctl	Flash_Mode	
0	X	No action (see Note 14-1)
1	Standby Mode	• Reg_Ctl_En, Reg_Ctl and Host_Ctl are set to '0b'. • EC_Int in Embedded Flash Command Register is set to '1b'. • The Embedded Flash is set to the Instruction Memory Interface and a wakeup interrupt is sent to the EC.
1	Read Mode	• Any read of the Embedded Flash array in progress is completed. • All strobes to the Embedded Flash array are set to '0b' • Reg_Ctl_En, Reg_Ctl and Host_Ctl are set to '0b'. The Program Mode is set to Standby. • EC_Int in Embedded Flash Command Register is set to '1b'. • The Embedded Flash is set to the Instruction Memory Interface and a wakeup interrupt is sent to the EC.
1	Program Mode	• Any word currently being programmed to the Embedded Flash array in progress is completed. • The Program Mode epilogue sequence is issued. • All strobes to the Embedded Flash array are set to '0b' • Reg_Ctl_En, Reg_Ctl and Host_Ctl are set to '0b'. The Program Mode is set to Standby. • EC_Int in Embedded Flash Command Register is set to '1b'. • The Embedded Flash is set to the Instruction Memory Interface and a wakeup interrupt is sent to the EC.
1	Erase Mode	• The erase sequence is completed. • All strobes to the Embedded Flash array are set to '0b' • Reg_Ctl_En, Reg_Ctl and Host_Ctl are set to '0b'. • EC_Int in Embedded Flash Command Register is set to '1b'. The Program Mode is set to Standby. • The Embedded Flash is set to the Instruction Memory Interface and a wakeup interrupt is sent to the EC.

Note 14-1 The assertion of LRESET# or the de-assertion of VCC PWRGD has no effect in this state; the EC can set or clear Reg_Ctl_En, Reg_Ctl and Host_Ctl and EC_Int is not set.

APPLICATION NOTE: Firmware should examine FLASH bit in the Power-Fail and Reset Status Register on page 148 after either VCC POR or LRESET# assertion.

14.5 Interrupts

The Embedded Flash Subsystem can generate interrupts to the EC for four events, three of which are reported in the three error bits in the Embedded Flash Status Register: Protect_Err, CMD_Err, & Busy_Err. The error bits are routed onto the FLASH_CMD_ERR, FLASH_PROTECT_ERR, FLASH_BUSY_ERR bits of the GIRQ15 Source Register on page 313.

In addition, asserting EC_Int in the Embedded Flash Command Register can be used to generate an interrupt to the EC. This bit are routed onto the FLASH_EC_INT, bit of the GIRQ15 Source Register on page 313.

© 2012-2018 Microchip Technology Inc.

EC interrupts generated by the LRESET# and VCC PWRGD pin signals are utilized as part of algorithms described in Section 14.11, "Programming the Embedded Flash Array," on page 253. The LRESET# pin signal sets the LRESET# bit in the GIRQ15 Source Register on page 313. The VCC PWRGD pin signal sets the VCC_PWRGD_INT bit in the GIRQ14 Source Register on page 311.

14.6 Flash Memory Array

The Flash Memory Array is composed of a Main Memory array as defined in Table 14-4.

An erase operation in the Flash Memory Array sets the affected memory array bits to one, while program operations write zeros. To reprogram any '0' bit in a page to '1,' the page must be erased.

The Flash Memory Array erases and programs with a 3.3V power supply; its IO interface operates at 1.8V. To modify the contents of the Flash Memory Array, VTR must be >3V before program or erase operations may begin. A summary of the MEC1632 Flash Memory Array features is shown below in Table 14-4.

TABLE 14-4: MEC1632 EMBEDDED FLASH FEATURE SUMMARY

Feature		Description
PROG/ERASE VOLTAGE		3.3V ± 10% (T_J = 0×C to 125×C)
READ VOLTAGE		1.8V ± 10% (T_J = 0×C to 125×C)
BUS WIDTH		32-bit
READ ACCESS/CYCLE TIME		30 ns max.
MAIN MEMORY BLOCK		48k x 32 in 96 pages of 2048 bytes
BOOT BLOCK	SIZE	4096 bytes
	LOCATION	Bottom
ERASE	TYPES	Page/Mass (2048 bytes/page)
	Program/ Erase Cycles	100,000 cycles (main array)
	Erase time, page	5 ms max
	Erase time, mass erase	40 ms max
PROGRAMMING		Per 32-bit word
WORD PROGRAM TIME		12 µS max
INTERFACE		All Program and Erase Operations are Enabled via a Command Sequence Interface using the Embedded Flash Register Interface.

14.6.1 FLASH ADDRESS MAPPING

The Main Block of the Flash Memory Array is located at 00_0000h in the EC address space. Address greater than 192KB will return FFFF_FFFFh. All locations can be used for both program and data by the EC.

The control registers used for programming the Flash are part of the Embedded Flash Logical Device. They are located in AHB address space starting at location FF_3800h. The contents of the Main Block can also be read using the control registers.

14.7 Instruction Memory Interface

When Reg_Ctl of the Embedded Flash Command Register is '0b', access to the Embedded Flash memory is through the Instruction Closely Coupled Memory interface of the EC. The registers in the Register Interface may be read or written, but no operations will be initiated on the Embedded Array. The Flash memory can be read with no wait states at the peak EC clock rate.

The minimum flash read access time through the EC's Instruction Closely Coupled Memory interface is shown in Table 14-5.

© 2012-2018 Microchip Technology Inc.

TABLE 14-5: FLASH READ ACCESS TIME

	Parameter	Symbol	Value			Units	Notes
			MIN	TYP	MAX		
1.	Flash Read Access Time	t_{Flsh_Rd}	2	–	–	MCLK	

14.8 Register Interface

The Embedded Flash Subsystem has it own Logical Device Number and Base Address as indicated in Table 14-6.

TABLE 14-6: Embedded Flash Subsystem BASE ADDRESS TABLE

Embedded Flash Interface Instance	LDN	AHB Base Address
Embedded Flash Interface	Eh	FF_3800h

Table 14-7, “Embedded Flash Subsystem Register Summary,” on page 246 summarizes the registers allocated for the Embedded Flash Subsystem.

The Table 14-7 is a register summary for one instance of the Embedded Flash Subsystem. The LPC I/O address for each Run-Time Register is described below as an offset from its Base Address Register. Each EC address is indicated as an SPB Offset from its AHB base address.

Both the Host and the Embedded Controller can communicate with the Embedded Flash Subsystem through a set of registers located on the LPC SPB bus. The Host uses the two byte Flash Mailbox interface, while the EC can access the register interface described in Section 14.8.2, "EC Register Interface".

14.8.1 FLASH MAILBOX INTERFACE

The Flash Mailbox interface provides the Host with a Message Interface similar to the MailBox Register Interface (see Chapter 12.0). The Flash Mailbox interface port occupies two addresses in the Host I/O space: FL_MBX_INDEX Register & FL_MBX_DATA Register.

To access a Flash Mailbox register once the Flash Mailbox BAR has been initialized, write the Host Offset of the desired 8-bit register into the Flash Mailbox INDX register. The register can then be accessed by reading or writing the Flash Mailbox DATA register.

It consists of 16 index-addressable 8-bit registers. These registers correspond to the four 32-bit registers that are shared between the Host and the EC.

To access a Flash Mailbox register once the Flash Mailbox BAR has been initialized, write the Host Offset of the desired 8-bit register into the Flash Mailbox INDX register. The register can then be accessed by reading or writing the Flash Mailbox DATA register.

Note 14-2 In this specification, Host access to registers in the Flash Mailbox through the Flash Mailbox Access Port are identified by the prefix INDX in OFFSET fields in the register tables in Section 14.12, "Detailed Description of Accessible Registers".

14.8.2 EC REGISTER INTERFACE

The following table summarizes the registers allocated for the Embedded Flash Subsystem. The offset field in the following table is the offset from the EC Base Address.

TABLE 14-7: EMBEDDED FLASH SUBSYSTEM REGISTER SUMMARY

	Host I/O Access			EC Interface		
Flash Mailbox interface **Register Name**	**Host I/O Index**	**SPB Offset**	**Host Type**	**SPB Offset**	**EC Type**	nSYS_RST
FL_MBX_INDEX Register	00h	00h	R/W	00h	R/W	00h
FL_MBX_DATA Register	04h	04h	R/W	04h	R/W	00h
Register Name	**Flash Mailbox Index**		**Host Type**	**SPB Offset**	**EC Type**	**VTR POR**
Embedded Flash Data Register	INDX 00h, 01h, 02,h, 03h	-	R/W	100h	R/W	0000_0000h
Embedded Flash Address Register	INDX 04h, 05h, 06h,	-	R/W	104h	R/W	0000_0000h
Embedded Flash Command Register	INDX 08h, 09h	-	R/W	108h	R/W	0000_0000h
Embedded Flash Status Register	INDX 0Ch, 0Dh	-	R/WC	10Ch	R/WC	0000_0000h
Embedded Flash Configuration Register	N/A	-	-	110h	R/W	0000_0000h
Embedded Flash Initialization Register	N/A	-	-	114h	R/W	0000_0000h

14.8.3 FLASH ADDRESS AND DATA REGISTERS

The Embedded Flash Data Register and the Embedded Flash Address Register are each implemented as a two entry FIFO, as illustrated in Figure 14-3, "Embedded Flash Controller Address and Data registers". The figure is suggestive and does not represent the precise implementation. Writes are always directed to the Tail register of the FIFO and reads are always sourced from the Head register. Each of the two registers has an associated Valid bit (**V** in the figure). The Tail Valid bit is set when the full Tail register is written and cleared when the Tail register is copied into the Head register. The Head Valid bit of the Embedded Flash Data Register is cleared whenever the full Head register is read by the Host or EC, or when the Embedded Flash Controller completes a Program transaction.The Head Valid bit of the Embedded Flash Address Register is cleared whenever the Embedded Flash controller completes a Read, Program or Erase function. Reading the Embedded Flash Address Register by a Host or EC register read does not affect the Head Valid bit. If the Head Valid bit is cleared and the Tail Valid bit is set, the Tail register is copied into the Head register, the Tail Valid bit is cleared and the Head Valid bit is set. All Valid bits are cleared when Flash_Mode of the Embedded Flash Command Register is set to Standby Mode.

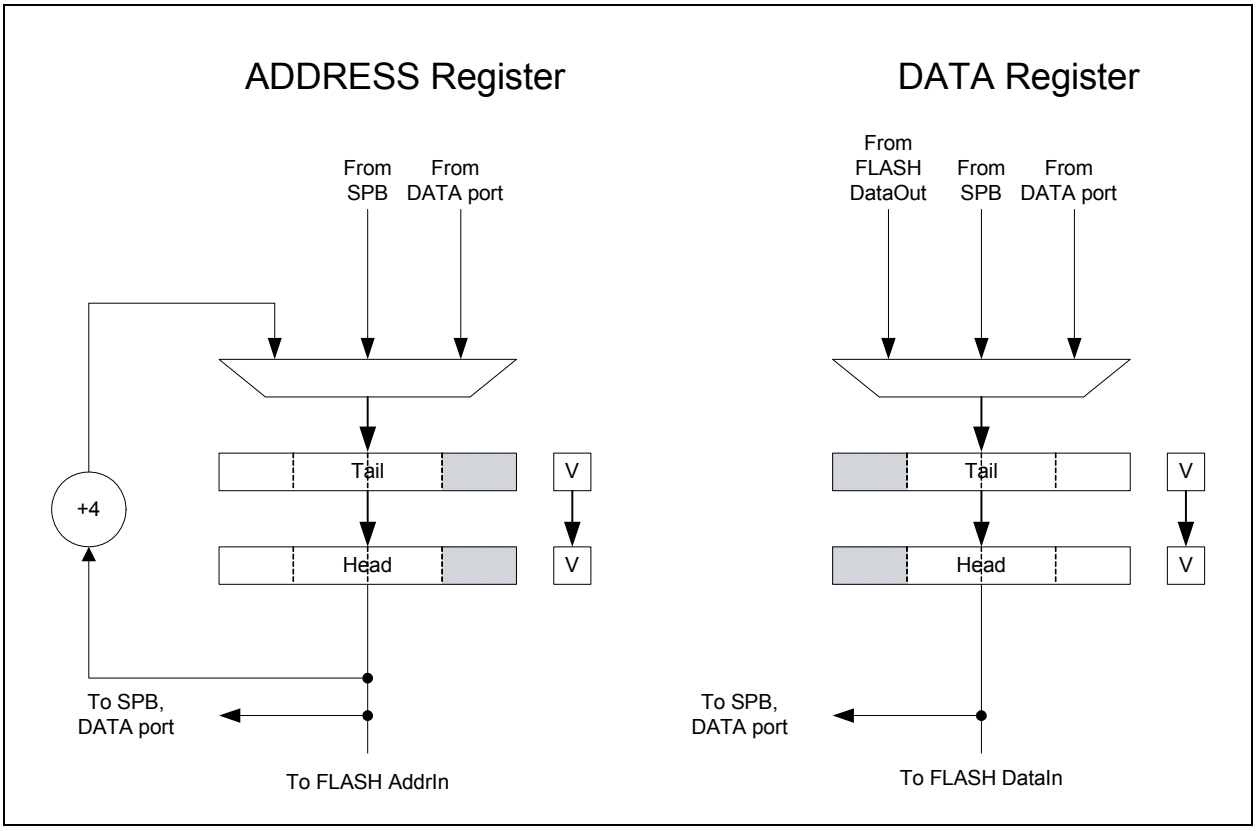
The Embedded Flash controller uses byte 0 of the Embedded Flash Address Register and byte 3 of the Embedded Flash Data Register to determine when the full register has been read or written. Writing byte 0 of the Embedded Flash Address Register sets the Valid bit on the Tail of the Address FIFO, while writing byte 3 of the Embedded Flash Data Register sets the Valid bit for the Tail of the Data FIFO. Reading byte 3 of the Embedded Flash Data Register clears the Valid bit for the Head of the Data FIFO. As stated above, reading any byte of the Embedded Flash Address Register does not cause a change in the Head Valid bit.

When the Valid bits of both the Head and Tail of the Embedded Flash Data Register are set, Data_Full in the Embedded Flash Status Register is set. If either bit is cleared, Data_Full is cleared. When the Valid bits of both the Head and Tail of the Embedded Flash Address Register are set, Address_Full in the Embedded Flash Status Register is set. If either bit is cleared, Address_Full is cleared.

The requesting Master will be stalled while attempting to read the Embedded Flash Data Register if the Embedded Flash controller is Busy and in the process of reading data into the Embedded Flash Data Register. This will result in at most one EC wait state. Because the Host cannot read the Flash registers faster than the controller can read from the Flash array, a Host Read access over the LPC bus will not have any added Wait SYNC cycles because the Flash controller is busy. In other cases, a Host or EC read of either the Embedded Flash Data Register or the Embedded Flash Address Register always returns the value in the Head register, whether the Valid bit is set or not.

© 2012-2018 Microchip Technology Inc.

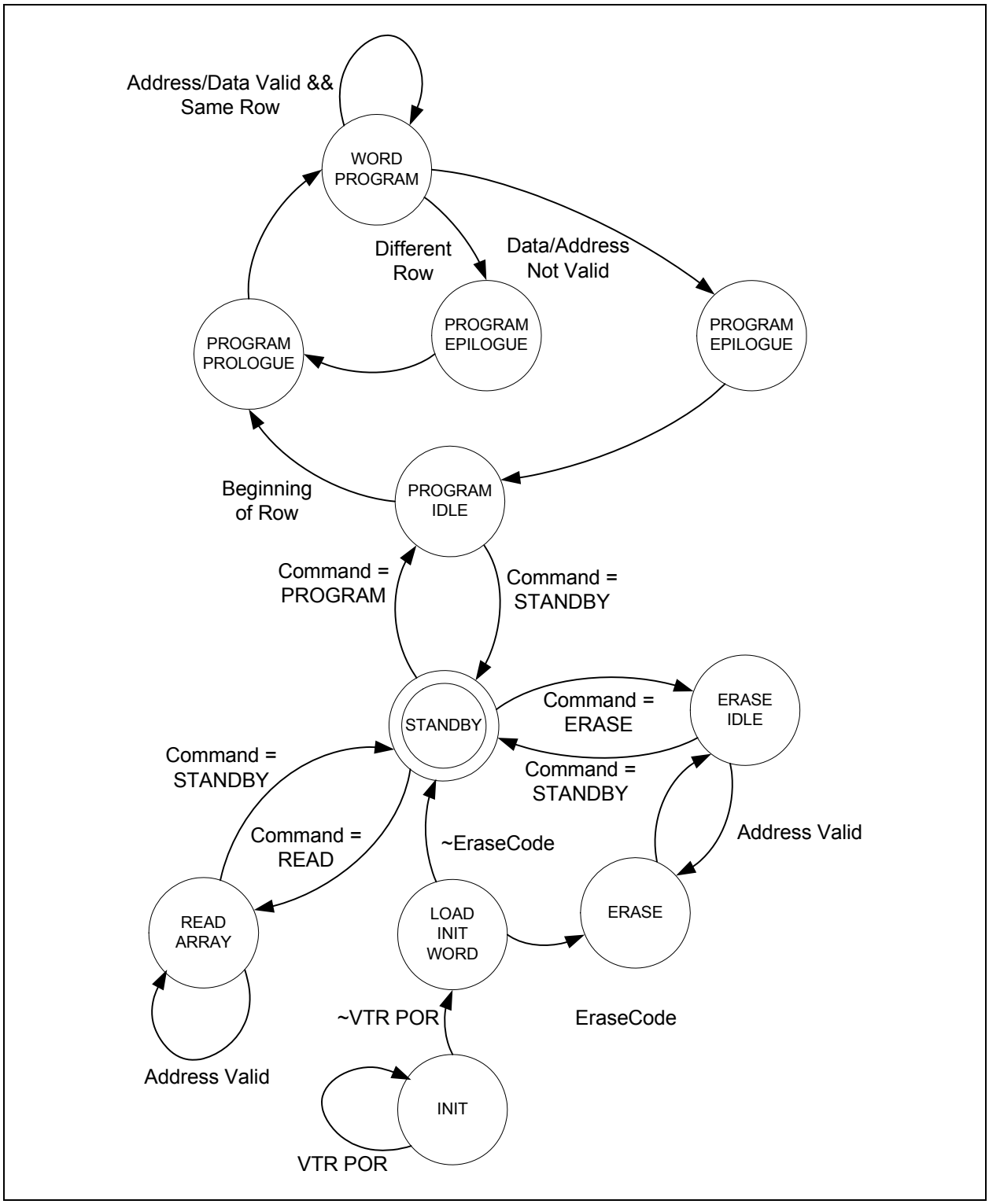
FIGURE 14-3: EMBEDDED FLASH CONTROLLER ADDRESS AND DATA REGISTERS



14.9 Embedded Flash Controller State Sequencing

The Embedded Flash controller proper timing on all strobe signals of the Embedded Flash array. Figure 14-4, "Embedded Flash Controller State Diagram" illustrates the primary state transitions for the controller.

FIGURE 14-4: EMBEDDED FLASH CONTROLLER STATE DIAGRAM



© 2012-2018 Microchip Technology Inc.

During VTR Reset, the Embedded Flash controller is kept in an initial state (INIT). In normal operation, when VTR POR (nSYS_RST) is de-asserted, the Embedded Flash controller reads the initialization value into the Embedded Flash Initialization Register, as described in Section 14.10.4, "Flash Data Initialization," on page 252, after which it remains in Standby Mode. If the ME (Mass Erase) field in the JTAG TEST REGISTER 4/Reset Register (Dh) is '1b' when VTR POR (nSYS_RST) is de-asserted the Embedded Flash controller, after transitioning to the Flash Data Initialization state will transition to Erase Mode. See Section 14.10.5, "Emergency Mass Erase," on page 252 for detailed sequence and description.

After initialization during normal operation, the Embedded Flash controller is placed into one of four modes (Standby Mode, Read Mode, Program Mode or Erase Mode) by setting Flash_Mode of the Embedded Flash Command Register to the appropriate values (0, 1, 2 or 3, respectively). Any write to the Embedded Flash Command Register while the controller is busy (when Busy of the Embedded Flash Status Register is '1b') will not modify the state and will set Busy_Err in the Embedded Flash Status Register.

14.9.1 STANDBY MODE

When in this mode all strobes to the Flash Memory Array are de-asserted and the Flash Memory Array is placed in its lowest power state. Both the Embedded Flash Data Register FIFO and the Embedded Flash Address Register FIFO are flushed. The registers can be read and written by software without error. Data returned on a read of a flushed FIFO is undefined. On a transition from Standby Mode to Program Mode, Read Mode or Erase Mode the FIFOs will be invalid and no action dependent on valid data in a FIFO will take place until new data is written into the registers.

The Embedded Flash controller must return to Standby Mode after any other state. An attempt to set the controller into a state other than Standby Mode when the controller is in Program Mode, Read Mode or Erase Mode will put the controller in Standby Mode and set CMD_Err of the Embedded Flash Status Register. Flash_Mode of the Embedded Flash Command Register will also be left in Standby Mode. If the controller is busy, writes to the Embedded Flash Command Register are ignored and Busy_Err in the Embedded Flash Status Register is set.

14.9.2 READ MODE

When the Embedded Flash controller is in Read Mode the addressing strobes are kept de-asserted as long as no Read is in progress. When the strobes are de-asserted, the Embedded Flash array is maintained in its lowest power state (equivalent to Standby Mode). When the Valid bit associated with the Head register in the Embedded Flash Address Register is '1b' and the Data FIFO is not full, the data in the Embedded Flash array that corresponds to the Embedded Flash Address Register is loaded into the Tail register of the Embedded Flash Data Register. The Data FIFO can hold two read results.

The Address FIFO is advanced when the Embedded Flash controller completes a read from the Embedded Flash array and stores the result in the Data FIFO. When byte 3 (the most significant byte) of the Embedded Flash Data Register is read, the Data FIFO is advanced. A read of the will return the value of the Head register but will not advance the FIFO. A read of the Embedded Flash Data Register when the FIFO is empty but the controller is busy reading from the array will stall the read until the controller has completed the lookup and the data can be returned to the requester.

14.9.2.1 Read Mode Timing Parameters

The minimum flash read access time through the EC Register Interface is one MCLKs.

14.9.2.2 Burst Read Mode

When Burst in the Embedded Flash Command Register is set to '1b', the Read function automatically increments the Embedded Flash Address Register in order to minimize the time necessary to read a block of memory from the Embedded Flash. As in Read Mode, a Flash read is initiated when the Head of the Address FIFO is Valid and the Data FIFO is empty. Whenever the Tail register in the Address FIFO is not Valid and the Head register in the FIFO is Valid, the Head register is incremented by 4 and written into the Tail. The Address FIFO will thus contain a sequence of consecutive word addresses as long as the Embedded Flash controller remains in Burst Read Mode.

Reading the Embedded Flash Data Register from either the EC or the JTAG interface always reads 32 bits at a time, so the Data Register FIFO is always advanced when the EC or JTAG does a data read. In Burst Read Mode the entire Flash memory can be read by repeatedly reading the Embedded Flash Data Register without the need for reading or writing any other register in the Embedded Flash Subsystem.

The Flash Mailbox interface has an additional mechanism. When in Burst Read Mode, the INDX portal has special behavior when it has the value 0h through 3h (that is, when INDX is set to point to the Embedded Flash Data Register). Every time the DATA portal is read by the Host, INDX will be automatically incremented by 1 when its value is 0h through 2h. If DATA is read by the Host when INDX is 3h, INDX is set to 0h and the Valid bit of the Head register of the Data

FIFO is cleared. If there is a valid address in the Address FIFO, the sequence to read the next data value, described above, will be initiated. When in Burst Read Mode, the entire contents of the Flash memory can be read by the Host with a sequence of reads to DATA, without any intervening writes to INDX.

APPLICATION NOTE: In Burst Read Mode the Embedded Flash Data Register will always contain the data from the next two locations after the last data location that is read. Software should ensure that the last word read does not cause an unintended read into a protected region. For example, a Burst Read that is intended to read the last word before the Protected Data Region will attempt to read into the protected region, which may cause an unintended protection error. To read the last data before the protected region, software should turn off Burst Read Mode and return to Standby Mode two words before the end of the page before the protected region. It should then re-enter Read Mode and read the last two words by explicitly writing the addresses into the Embedded Flash Address Register.

14.9.3 PROGRAM MODE

When the Embedded Flash controller is in Standby Mode, setting the Embedded Flash Command Register to Program Mode will set up the Embedded Flash array for programming. The Embedded Flash Data Register and the Embedded Flash Address Register FIFOs are used for Program Mode in a manner similar to their use in Read Mode. When the Head registers in both the Address and Data FIFO are Valid, the Flash strobes are sequenced in order to write the contents of the Data register into the Flash Memory at the address specified in the Address register. At the end of the Word Program sequence the two FIFOs are advanced.

Note: In Program Mode there should always be an erase before writing the same DWORD twice.

14.9.3.1 Program Mode Timing Parameters

The following three sequences are used in programming the Embedded Flash: Program Prologue, Word Program, Program Epilogue.

During all three sequences, the controller asserts Busy while the sequence is in progress. Flash programming time is shown in Table 14-8.

TABLE 14-8: FLASH PROGRAMMING TIME

Parameter	Symbol	Value			Units	Notes
		MIN	TYP	MAX		
Single DWORD Programming Time	–	19	–	–	µs	
Burst DWORD Programming Time (single row)	–	13.5	–	–	µs	

The Program Prologue sequence is issued before the first time any word in a row can be programmed. The Program Epilogue sequences is issued after the last time any word of a row is programmed. The Embedded Flash controller automatically issues the Program Prologue and Program Epilogue sequences as required. After the controller is placed in Program Mode, the Program Prologue is issued as soon as the Head registers in both the Embedded Flash Data Register and the Embedded Flash Address Register are Valid. Once the Program Prologue is completed, the controller will immediately issue the Word Program sequence.

As long as the Head registers in both the Embedded Flash Data Register and the Embedded Flash Address Register are Valid the controller will continue to issue Word Program sequences. If either FIFO is invalid at the end of the Word Program sequence, the controller will issue the Program Epilogue sequence and the controller will be left in the initial Program Mode state. If new data arrives in the FIFOs, then the Program Prologue followed by the Word Program sequence will be issued.

14.9.3.2 Burst Program Mode

Burst Program Mode is enabled whenever Burst in the Embedded Flash Command Register is set and the controller is in Program Mode. The behavior is similar to Burst Read Mode. When Burst is enabled, the Embedded Flash Address Register FIFO is always kept filled automatically with incrementing addresses. Whenever the Tail register in the Address FIFO is not Valid and the Head register is valid, the Head register is incremented by 4 and stored in the Tail. The controller otherwise behaves as in Program Mode. When both the Head register of the Address FIFO and the Head register of the Data FIFO are Valid, a Word Program sequence is initiated (with the possible addition of Program Prologue and

© 2012-2018 Microchip Technology Inc.

Program Epilogue, as described above). When the programming sequence is complete, the Head registers of both FIFOs are marked not Valid. With this mechanism the entire Flash array can be programmed with a sequence of writes to the Embedded Flash Data Register without any writes to the Embedded Flash Address Register or Embedded Flash Command Register.

The Flash Mailbox interface has an additional mechanism, as in Burst Read Mode. The INDX portal has special behavior when it has the value 0h through 3h (that is, when INDX is set to point to the Embedded Flash Data Register). Every time the DATA portal is written by the Host, INDX will be automatically incremented by 1 when its value is 0h through 2h. If DATA is written by the Host when INDX is 3h, INDX is set to 0h and the Tail register of the Data FIFO is advanced to the Head and the Tail is marked not Valid. If there is a valid address in the Head of the Address FIFO, the Word Program sequence will be initiated. In Burst Program Mode, the entire contents of the Flash memory can be programmed by the Host with a sequence of writes to DATA, without any intervening writes to INDX.

14.9.4 ERASE MODE

When the Embedded Flash Command Register is set to Erase Mode, the state machine will wait until a valid address is written into the Embedded Flash Address Register. Once the register is valid, Busy in the Embedded Flash Status Register is set to ‘1b and the Embedded Flash controller sequences the Embedded Flash array strobes to erase part or all of the Embedded Flash. All bits in the erased area are set to ‘1b’. Busy remains set during the operation. When the erase operation is complete Busy is set to ‘0b’. The controller remains in Erase Mode and can accept additional addresses in the Embedded Flash Address Register.

Bits[23:19] of the Embedded Flash Address Register selects between page and mass erase of the entire array selected. Some of the Bits[17:0] of the Embedded Flash Address Register select which page is erased and the other bits are ignored. The Embedded Flash Address Register should be configured with the proper page address before setting the Embedded Flash Command Register to Erase Mode.

Table 14-9, "Erase Mode Functions" provides the required decoding of the memory to be erased.

TABLE 14-9: ERASE MODE FUNCTIONS

Address Register Bits[23:19]	Area Erased
xxxx0b	2KB Page in Embedded Flash Main Array specified by Embedded Flash Address Register bits[17:11] & bits[10:0] are ignored
11111b	Mass erase of entire Embedded Flash Main Array

14.10 Flash Lock Controls

14.10.1 FLASH WRITE PROTECT

When the Boot_Lock in the Embedded Flash Configuration Register is asserted, the bottom 4K bytes (0000h – 0FFFh) of the Flash Main Memory Array (the Boot Block) are write protected and cannot be changed by any programming method including Program Mode or Erase Mode. The Mass Erase function is therefore disabled, since it would erase the Boot Block. The rest of the Embedded Flash array, is not affected. If Boot_Lock is not asserted, all programming and erase functions, including Erase Mode, are enabled and the bottom 4K bytes are not protected.

14.10.2 FLASH BOOT PROTECT

The Embedded Flash Boot Block can be protected from all reads and writes, from either the Host or the EC. If Boot_Protect_En in the Embedded Flash Configuration Register is ‘1b’ the first time there is a address reference to the Embedded Flash Subsystem that is outside the Boot Block (that is, the first time the EC does either a program fetch to an address that is larger than 00_0FFFH or a data reference to an address that is larger than 00_0FFFh and less than 80_0000h), Boot_Block and Boot_Lock in the Embedded Flash Status Register are set. Boot_Lock prevents any programming or erase operations on the Boot Block, as described in Section 14.10.1, "Flash Write Protect". Mass Erase is also disabled. In addition, any attempt to read the Boot Block, from either the Instruction Memory Interface or the Register Interface will return FFFF_FFFFh. Once set, only a VTR POR (nSYS_RST) can clear Boot_Block.

Typically, Flash Boot Protect will be the last function performed as part of boot. The EC boot code should configure the Interrupt Vector Table to be outside the Boot Block. It should then set the Boot_Protect_En bit, then jump to a location outside of the Boot Block. From that point on the Boot Block will be protected for both read and write access from either the Host or the EC.

© 2012-2018 Microchip Technology Inc.

14.10.2.1 JTAG Disable to protect the flash Boot Block

If Boot_JTAG_Block in the Embedded Flash Initialization Register is set to ‘0b’, the JTAG Debug Data Registers interface will not be accessible to the MEC1632 JTAG pins as long as Boot_Block in the Embedded Flash Status Register is ‘0b’. This insures that an external device cannot read or write the Boot Block while the EC is executing within it. If Boot_JTAG_Block is in its default state of ‘1b’, an external debugger could potentially halt the EC and then read or reprogram any word in the Boot Block.

As described in Section 14.10.4, "Flash Data Initialization", data in the Embedded Flash Initialization Register are configured by programming the initialization vector in the Embedded Flash array.

14.10.3 FLASH DATA PROTECT

The Flash Data Protect Region is the last 4KB in the Embedded Flash main array. If Data_Protect in the Embedded Flash Configuration Register is ‘1b’ the Flash Data Protect Region is protected from any read or write accesses by the Host through the Mailbox interface. The EC can access the Data Protect Region through both the Instruction Memory Interface and the Register Interface, as long as the JTAG port is not enabled. Once JTAG is enabled, by de-asserting the JTAG Reset pin (JTAG_RST#), Data_Block in the Embedded Flash Status Register is set to ‘1b’ and the entire Data Protect Region is blocked from all accesses, by the Host, by the EC or by JTAG. Any reads return FFFF_FFFFh, and any attempt to program data in the region will be blocked. Any page erase operation in the Data Protect Region is blocked and Mass Erase is also disabled. Once Data_Block is set, it can only be cleared by a VTR POR (nSYS_RST). Clearing Data_Protect in the Embedded Flash Configuration Register will not change Data_Block, so the only way to access the protected region once JTAG is enabled is to power cycle the device.

14.10.4 FLASH DATA INITIALIZATION

Before the EC come out of reset, the Embedded Flash Controller:

- reads one 32-bit word from offset FFCh of the Flash main memory block and writing it to the Embedded Flash Initialization Register on page 263.

The read-only Embedded Flash Initialization Register can be used for configuration and initialization data. Once written, this data will stay constant until the upper page of the Boot Block is erased and the VTR power is cycled.

Boot_JTAG_Block bit in the Embedded Flash Initialization Register is used to control JTAG access during the boot period. If this bit is programmed to ‘0b’, JTAG cannot access any locations within the MEC1632, other than the JTAG test registers, while the Boot Block is readable. If Boot_Block in the Embedded Flash Status Register is set, JTAG accesses are enabled, although any access to the Boot Block is blocked by Boot_Block. If Boot_JTAG_Block is left in the default ‘1b’ state, a JTAG master could halt the EC just after VTR POR (nSYS_RST) and manipulate all locations in the MEC1632, including any data in the Boot Block.

14.10.5 EMERGENCY MASS ERASE

If the Boot_JTAG_Block bit located in the Embedded Flash Initialization Register is asserted but the EC boot code is faulty, it is possible that the EC could become stuck in the boot sequence without ever enabling JTAG access or LPC access. In that state, there would be no way to erase the Embedded Flash and reprogram the EC, since both the JTAG port and LPC access would be inaccessible. For this reason, a fail-safe mechanism is included in the Embedded Flash controller.

If the ME (Mass Erase) field is ‘1b’ when VTR POR (nSYS_RST) is de-asserted the Embedded Flash controller, after transitioning to the Flash Data Initialization state will transition instead to Erase Mode. After the Mass Erase is completed the Embedded Flash is held in an idle state and the Embedded Controller will remain idle. JTAG and LPC will be blocked as well.

An additional power cycle of the MEC1632 will be required to reset the Embedded Flash Initialization Register and re-enable the Embedded Flash and the Embedded Controller. After the additional power cycle JTAG can be used to reprogram the Embedded Flash.

In order to trigger an Emergency Mass Erase, the JTAG interface should be used to perform the following sequence:

1. Place the JTAG_RST# pin in the asserted (low) state and apply VTR Power.
2. De-assert the JTAG_RST# pin (high) to take the JTAG interface out of the reset state (see Section 46.11, "JTAG Interface Timing," on page 628).
3. The POR EN field in the JTAG RESET TEST Register (2h) should be set to ‘1b’ in order to enable JTAG control of the VTR POR circuitry.
4. The VTR POR field in the JTAG RESET TEST Register (2h) should be set to ‘1b’ in order to generate a VTR POR in the MEC1632.

© 2012-2018 Microchip Technology Inc.

5. The ME field in the JTAG TEST REGISTER 4/Reset Register (Dh) should be set to ‘1b’.
6. The VTR POR field in the JTAG RESET TEST Register (2h) should be set to ‘0b’ which terminates the VTR POR.
7. Delay long enough to ensure a Mass Erase completes and the VTR POR (nSYS_RST timing) should be set to ‘1b’ in order to generate a VTR POR in the MEC1632.
8. The VTR POR field in the JTAG RESET TEST Register (2h) should be set to ‘0b’ which terminates the VTR POR.

14.11 Programming the Embedded Flash Array

In normal operation, the EC fetches all of its instructions from the Embedded Flash array. While it is executing out of Flash, the program interface using the registers described in the next section cannot be used. When the Register Interface is used, the EC cannot fetch either instructions or data out of the Embedded Flash. The EC can either sleep, or run entirely out of the EC Data Memory. When running out of EC Data Memory, EC firmware should insure than all interrupts are disabled, unless the interrupt vector table is relocated to the EC Data Memory along with any interrupt handler that is enabled.

There are three methods by which the Embedded Flash array can be programmed:

- The Flash may be programmed through the JTAG interface. Through the JTAG interface, the EC can be halted or directed to run entirely out of the EC Data Memory. Once the EC no longer requires the Flash, the registers in the next section can be used to program the array, by first setting Reg_Ctl_En in the Embedded Flash Configuration Register (enabling register control of the Flash), then using the other registers to program the array. Once programming is completed, Reg_Ctl_En in the Embedded Flash Configuration Register is cleared (enabling EC instruction access) and then setting the EC into the RUN state. See Section 44.0, "JTAG and XNOR," on page 580 for a description of JTAG operation.
- The Flash may be programmed by the Host. After co-ordination between the Host and the EC, the EC configures the Embedded Flash Configuration Register to enable the register programming mode and to give the Host write access to the registers (by setting both Reg_Ctl_En and Host_Ctl to ‘1b’). The EC would then disable all interrupts except the EC_Int Flash Mailbox interrupt from the Host. The EC Interrupts for LRESET# and VCC PWRGD pin signals should also be enabled (see Section 14.5, "Interrupts," on page 243), in order to restore control to the EC in the event the Host loses power and does not return control to the EC properly. The EC then signals the Host that it can proceed. At this point the EC should verify that the Host can take control, by checking the state of LRESET# and VCC PWRGD. If LRESET# is asserted or VCC PWRGD is de-asserted, the Host will not respond to the EC signal to take control, and the EC should therefore cancel the hand off to the Host, clear Reg_Ctl_En and Host_Ctl, re-enable interrupts and continue. If the Host is capable of responding to the EC signal, the EC finally either puts itself to sleep or runs entirely out of the. When the Host finishes programming the Embedded Flash array, it sets EC_Int in the Embedded Flash Command Register to ‘1b’, which wakes the EC.

The Host may also program the Flash through the use of the MailBox Register Interface. The Host communicates the address to be modified and the data to program through a pre-arranged set of mailbox registers, then sends an interrupt to the EC. The EC would then program the Flash while running out of the on-chip SRAM, as described in the following bullet item.

- The EC can itself program the Flash array by loading a program to do so into the on-chip 4KB SRAM. Once the EC is running out of SRAM, it can disable all interrupts and configure the Flash array to be programmable through the registers (by setting Reg_Ctl_En in the Embedded Flash Configuration Register). After programming is completed, the EC can reset the Reg_Ctl bit to make the Flash accessible to instruction fetch.

In all cases, the Embedded Flash Data Register can be updated to set up the next write while the Flash is busy programming the current write.

14.11.1 TRANSFERRING CONTROL TO THE HOST

In order to transfer control of the Embedded Flash to the Host, the EC has to enable the Host access to the Register interface, inform the Host that it has access, then either sleep or execute code entirely out of the EC Data Memory. The EC must sleep or execute in RAM because it will no longer be able to use the ICCM instruction memory interface. The following sequence is an example of how this transfer can be accomplished:

```
// This code is located in the Flash
// Command_Address is 0xFF3908
// Config_Address is 0xFF3910
//
*Config_Address |= REG_CTL_EN | HOST_CTL; // Permit the Host to control the Flash
registers
Disable_all_interrupts();
Enable_Interrupt(EC_Int);         // Enable transition from Host back to EC
Enable_Interrupt(LRESET#);
Enable_Interrupt(VCC_PWRGD);
Host_Message(FLASH_READY);        // Tell Host it can take control of the Flash
if( assserted(LRESET) || deasserted(VCCPWRGD) ) {
    cancel_flash_transfer();      // Host is not powered: annul transfer
                                  // restore Flash configuration/command registers
    restore_interupts();
    return();                     // return to regular EC code
} else {
    sleep();                      // Wait for EC_Int
}
```

The next code sequence would be executed by the Host once it receives the message from the EC that it may take control of the Flash:

```
// INDX is the address of INDX in the Host I/O space
// DATA is the address of DATA in the Host I/O space
//
INDX = 9; // index of Command register byte where Reg_Ctl bit is located
DATA = REG_CTL; Take control of the Flash Address and Data regsiters
```

As an alternative to the Flash-based EC code described above, the transfer function can run partially in Flash and partially in SRAM. In this mode it is not necessary to enable LRESET# or VCC PWRGD interrupts, since SRAM-based EC code can check the state of those signals through polling: Normal termination of Flash programming occurs as before when the Host sets the EC_Int bit in the Command register.

```
// This code is located in the Flash
// Command_Address is 0xFF3908
// Config_Address is 0xFF3910
//
*Config_Address |= REG_CTL_EN;    // Permit control of Flash via register interface
Disable_all_interrupts();
Enable_Interrupt(EC_Int);         // Enable transition back to EC
goto SRAM_Flash_Code;
...
SRAM_Flash_Code:                  // start of code in SRAM
*Command_Address |= REG_CTL;
*Config_Address |= HOST_CTL;
Host_Message(FLASH_READY);
while(true) {                     // commence polling on LRESET/VCC PWRGD
    if( asserted(LRESET) || deasserted(VCCPWRGD) ) {
          cancel_flash_transfer();// Host is not powered: annul transfer
                                  // restore Flash configuration/command registers
          restore_interrupts();
          return();               // return to EC code in Flash
    }
}
```


© 2012-2018 Microchip Technology Inc.

14.11.2 READING THE EMBEDDED FLASH

The Embedded Flash memory, in both the main array can most conveniently be read with Burst mode set. An initial Flash address should be configured, and then the rest of the Flash memory can be read with just a sequence of reads to the data register. The following pseudocode illustrates how reading the Flash from the Host might proceed:

```
// Byte_array[] is a data structure to receive the data. It is a sequence of bytes
// Flashbase is the first address in the flash memory to be read
// Limit is the total number of bytes to read
// INDX is the address of INDX in the Host I/O space
// DATA is the address of DATA in the Host I/O space
//
// this code works as long as the read does not wrap around the end of the Flash array
//
INDX = 8; // index of Command register
DATA = READ | BURST;
INDX = 6; // bits 24:16 of Flash Address register
DATA = Flashbase.byte2;
INDX = 5; // bits 15:8 of Flash Address register
DATA = Flashbase.byte1;
INDX = 4; // bits 7:0 of Flash Address register
DATA = Flashbase.byte0;      // the Flash address advances in the Address FIFO
                             // Read from the Flash is started
INDX = 0; // bits 7:0 of Flash Data register
for( i = 0; i < Limit; i++)
{
    Byte_array[i] = DATA;
}
INDX = 8; // index of Command register
DATA = STANDBY; // Exit READ mode and flush Address/Data FIFOs
```

Reading from the EC is similar, except that data can be read four bytes at a time:

```
// Word_array[] is a data structure to receive the data.
//   It is a sequence of 32-bit words
// Flashbase is the first address in the flash memory to be read
// Limit is the total number of 32-bit words to read
// Flash_Address is 0xFF3904
// Data_Address is 0xFF3900
// Command_Address is 0xFF3908
//
// this code works as long as the read does not wrap around the end of the Flash array
//
*Command_Address = READ | BURST;
*Flash_Address = Flashbase;      // the Flash address advances in the Address FIFO
for( i = 0; i < Limit; i++)
{
    Word_array[i] = *Flash_Data;
}
*Command_Address = STANDBY; // Exit READ mode and flush Address/Data FIFOs
```

14.11.3 WRITING THE EMBEDDED FLASH

The Embedded Flash memory can most conveniently be programmed with Burst mode set. An initial Flash address should be configured, and then the rest of the Flash memory can be written with just a sequence of writes to the data register. The following pseudocode illustrates how reading the Flash from the Host might proceed:

```
// Byte_array[] is a data structure that sources the data. It is a sequence of bytes
// Flashbase is the first address in the flash memory to be programmed
// Limit is the total number of bytes to write. It is assumed to be a multiple of
//    the 256-byte row size for simplicity
// INDX is the address of INDX in the Host I/O space
// DATA is the address of DATA in the Host I/O space
//
// this code works as long as the write block does not wrap around
// the end of the Flash array.
//
INDX = 8; // index of Command register
DATA = PROGRAM | BURST;
INDX = 6; // bits 24:16 of Flash Address register
DATA = Flashbase.byte2;
INDX = 5; // bits 15:8 of Flash Address register
DATA = Flashbase.byte1;
INDX = 4; // bits 7:0 of Flash Address register
DATA = Flashbase.byte0;      // the Flash address advances in the Address FIFO
for( i = 0; i < Limit; )
{
    INDX = 0xC; // index of Status register, to check space in Data FIFO
    while( DATA & DATA_FULL ); // stall while Data FIFO has no space
    INDX = 0; // bits 7:0 of Flash Data register
    // Write one word (four bytes) into the Flash array
    DATA = Byte_array[i++];
    DATA = Byte_array[i++];
    DATA = Byte_array[i++];
    DATA = Byte_array[i++];
}
INDX = 8; // index of Command register
DATA = STANDBY; // Exit PROGRAM mode and flush Address/Data FIFOs
```

Writing from the EC is similar, except that data can be written four bytes at a time:

```
// Word_array[] is a data structure to receive the data.
//   It is a sequence of 32-bit words
// Flashbase is the first address in the flash memory to be written
// Limit is the total number of 32-bit words to write
// Flash_Address is 0xFF3904
// Data_Address is 0xFF3900
// Command_Address is 0xFF3908
// Status_Address is 0xFF390C
//
// this code works as long as the program block does not wrap around
// the end of the Flash array
//
*Command_Address = PROGRAM | BURST;
*Flash_Address = Flashbase;      // the Flash address advances in the Address FIFO
for( i = 0; i < Limit; i++)
{
    while( *Status_Address & DATA_FULL ); // stall while Data FIFO has no space
    *Flash_Data = Word_array[i];
}
*Command_Address = STANDBY; // Exit PROGRAM mode and flush Address/Data FIFOs
```


© 2012-2018 Microchip Technology Inc.

14.12 Detailed Description of Accessible Registers

See Section 14.8, "Register Interface," on page 245.

14.12.1 EMBEDDED FLASH DATA REGISTER

The Embedded Flash Data Register can only be written when the Embedded Flash Controller is in Program Mode. In Standby Mode, Read Mode and Program Mode the register is read-only. Writes will complete but have no effect.

TABLE 14-10: EMBEDDED FLASH DATA REGISTER

HOST INDEX	BYTE3: INDX 03h BYTE2: INDX 02h BYTE1: INDX 01h BYTE0: INDX 00h			8-bit			**HOST SIZE**	
EC OFFSET	100h			32/16/8-bit			**EC SIZE**	
POWER	VTR			0000_0000h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D31**	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Flash_Data[31:24]							
BYTE2 BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Flash_Data[23:16]							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Flash_Data[15:8]							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Flash_Data[7:0]							

FLASH_DATA[31:0]

This 32-bit register holds the data to be written into the Flash memory array during a program cycle, as well as the data returned from a Flash memory read during a Read From SPB cycle. It should be set up before the Embedded Flash Address Register is configured.

14.12.2 EMBEDDED FLASH ADDRESS REGISTER

TABLE 14-11: EMBEDDED FLASH ADDRESS REGISTER

HOST INDEX	BYTE3: INDX 07h BYTE2: INDX 06h BYTE1: INDX 05h BYTE0: INDX 04h					8-bit	**HOST SIZE**	
EC OFFSET	104h					32/16/8-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D31**	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	R	R	R	R	R	R	R	R
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE2 BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Flash_Address[23:16]							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Flash_Address[15:8]							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R	R
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R	R
BIT NAME	Flash_Address[7:2]						Reserved	

FLASH_ADDRESS[23:0]

This register represents a byte address for the Embedded Flash array. Since all read and write operations are to 32-bit quantities, the low-order to bits (Flash_Address[1:0] must always be 0, so these bits are reserved. If the Flash state machine is in Read Mode, writing Byte0 of the Embedded Flash Address Register initiates the Read sequence. If the Flash state machine is in Program Mode, the Program sequence is initiated when there is both a valid Data value in the Embedded Flash Data Register and the Embedded Flash Address Register has been updated by writing Byte0. Bit18 is not used in the MEC1632.

Bits[23:19] are only examined in Erase Mode. See Section 14.9.4, "Erase Mode," on page 251 for required decoding during Erase Mode.

The following provides the required decoding during Read Mode and Program Mode.

© 2012-2018 Microchip Technology Inc.

14.12.3 EMBEDDED FLASH COMMAND REGISTER

This register is Read-Only while Busy in the Embedded Flash Status Register is ‘1b’. An attempt to write this register while Busy is asserted will not modify the register and will set Busy_Err in the Embedded Flash Status Register.

A write to this register causes the Embedded Flash controller to transition to the selected state.

TABLE 14-12: EMBEDDED FLASH COMMAND REGISTER

HOST INDEX	BYTE0: INDX 8h BYTE1: INDX 9h			8-bit		**HOST SIZE**		
EC OFFSET	108h			32/16/8-bit		**EC SIZE**		
POWER	VTR			00h		nSYS_RST **DEFAULT**		
BUS	LPC SPB							
BYTE[3:2] BIT	**D31**	**D30**	**D29**	**...**		**D18**	**D17**	**D16**
HOST TYPE	R	R	R	R	R	R	R	R
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	R	R	R	R	R	R	R	R/W
EC TYPE	R	R	R	R	R	R	R	R/W
BIT NAME	Reserved							Reg_ Ctl
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R	R/W	R/W	R/W	R/W	R/W
EC TYPE	R	R	R	R/W	R/W	R/W	R/W	R/W
BIT NAME	Reserved				EC_ Int	Burst	Flash_Mode[1:0]	

FLASH_MODE

The field determines which Master may write the registers in the Embedded Flash controller:

0 The Embedded Flash controller is placed in the Standby mode.

1 The Embedded Flash controller is placed in Read mode.

2 The Embedded Flash controller is placed in Program mode.

3 The Embedded Flash controller is placed in Erase mode.

BURST

If the Embedded Flash controller is in Read Mode or Program Mode and this bit is ‘1b’, the contents of the Head register in the Embedded Flash Address Register FiFO will be incremented by 4 and written into the Tail register whenever the Head register is Valid. When this bit is ‘0b’, the Embedded Flash Address Register and the INDEX register are not incremented automatically.

See Section 14.9.2.2, "Burst Read Mode," on page 249 and Section 14.9.3.2, "Burst Program Mode," on page 250 for information about the use of Burst.

© 2012-2018 Microchip Technology Inc.

EC_INT

Setting this bit to ‘1b’ generates an EC interrupt and simultaneously sets Reg_Ctl_En and Host_Ctl in the Embedded Flash Configuration Register as well as Reg_Ctl_En in this register to ‘0b’. By setting these bits to ‘0b’ the Embedded Flash array is configured for EC instruction access, so that when the EC awakes and responds to the interrupt request triggered by EC_Int, it can fetch the interrupt vector from its usual location in the Embedded Flash array. See Section 14.5, "Interrupts," on page 243.

The response to setting this bit is the same as the response to VCC PWRGD or PCIRESET signals, as detailed in Table 14-3, "VCC PWRGD and LRESET# Behavior". If the Embedded Flash Controller is in either Erase Mode or Program Mode the programming sequence is terminated, the Embedded Flash controller is set to Standby Mode, Reg_Ctl, Reg_Ctl_En and Host_Ctl are set to ‘0b’ and the wakeup interrupt is sent to the EC.

REG_CTL

When this bit is set, the address input and the control strobes of the Flash Memory Array are sourced from the registers in the Embedded Flash Subsystem. Software on either the Host or the EC can read and write the Flash Memory Array by reading and writing the registers in the subsystem. The EC cannot execute instructions out of the Flash Memory Array.

When this bit is cleared (which is the default), the address input and the control strobes of the Flash Memory Array are sourced from the Instruction Closely Coupled Memory interface. The EC can execute instructions directly from the Flash Memory Array. The other registers in the Embedded Flash Subsystem can be read or written, but do not affect the Flash Memory Array.

Note:	A 32-bit write from either the EC or JTAG with Reg_Ctl_En set to ‘1b’ can simultaneously write the Byte 0 control bits in the Embedded Flash Command Register.

This bit can only be set if Reg_Ctl_En in the Embedded Flash Configuration Register is ‘1b’.

14.12.4 EMBEDDED FLASH STATUS REGISTER

TABLE 14-13: EMBEDDED FLASH STATUS REGISTER

HOST INDEX	BYTE 1: INDX Dh BYTE 0: INDX Ch			8-bit			**HOST SIZE**	
EC OFFSET	10Ch			32/16/8-bit			**EC SIZE**	
POWER	VTR			0000_0000h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE[3:2] BIT	**D31**	**D30**	**D29**	**...**		**D18**	**D17**	**D16**
HOST TYPE	R	R	R	R	R	R	R	R
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	R	R	R	R	R	R/WC	R/WC	R/WC
EC TYPE	R	R	R	R	R	R/WC	R/WC	R/WC
BIT NAME	Reserved					Protect_Err	CMD_Err	Busy_Err
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R	R	R	R	R	R
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved	Data_Block	Boot_Block	Reserved	Boot_Lock	Address_Full	Data_Full	Busy

© 2012-2018 Microchip Technology Inc.

BUSY

This bit reflects the state of the Embedded Flash controller. This bit is set while the controller is processing a flash control sequence. While this bit is set the Embedded Flash Command Register can not be written. If a write is attempted on this register, the write fails and Busy_Err in this register is set.

This bit is read-only.

DATA_FULL

This bit reflects the value of the Valid bit of the Tail register of the Embedded Flash Data Register. When it is set the FIFO is full and any additional writes will set Busy_Err.

This bit is read-only.

ADDRESS_FULL

This bit reflects the value of the Valid bit of the Tail register of the Embedded Flash Address Register. When it is set the FIFO is full and any additional writes will set Busy_Err.

This bit is read-only.

BOOT_LOCK

This bit is set whenever the Boot Block is write-protected. The Boot Block will be write-protected when Boot_Lock in the Embedded Flash Configuration Register is set. This bit is also set when Boot_Block is set, so that whenever the Boot Block is read-protected it is also write-protected.

A copy is maintained in this register so that the Host can access it, since the Host has no access to the Embedded Flash Configuration Register. See Section 14.10.1, "Flash Write Protect," on page 251 for a description of write-protecting the boot block.

This bit is read-only.

BOOT_BLOCK

When this bit is ‘1b’, the Boot Block is protected from all access. See Section 14.10.2, "Flash Boot Protect," on page 251. Once set, it will only be cleared by a VTR Power On Reset.

This bit is read-only.

DATA_BLOCK

When this bit is ‘1b’, the Protected Data Block is protected from all access. See Section 14.10.4, "Flash Data Initialization," on page 252. Once set, it will only be cleared by a VTR Power On Reset.

This bit is read-only.

BUSY_ERR

This bit is set if

- A write to the Embedded Flash Command Register occurs while Busy is set.
- A write to the Embedded Flash Address Register occurs while Address_Full is set.
- A write to the Embedded Flash Data Register occurs while Data_Full is set

This bit is sticky: once set, it remains set until cleared by a write with the value ‘1b’ to this bit.

CMD_ERR

If the Embedded Flash controller is in Read Mode, Program Mode or Erase Mode, this bit is set if the Embedded Flash Command Register is set to any value other than Standby Mode.

This bit is sticky: once set, it remains set until cleared by a write with the value ‘1b’ to this bit.

© 2012-2018 Microchip Technology Inc.

PROTECT_ERR

If Boot_Lock or Boot_Block in the Embedded Flash Status Register is set and a program operation to an address within the range 0000h - 0FFFh, or a page erase operation to page 0, or a mass erase operation to the main Embedded Flash array occurs, this bit is set. In addition, if Boot_Block is set and any read access to the Embedded Flash in the range 0000h through 0FFFh is attempted, either through the Instruction Memory Interface or the Register Interface, this bit will be set.

This bit is sticky: once set, it remains set until cleared by a write with the value ‘1b’ to this bit.

14.12.5 EMBEDDED FLASH CONFIGURATION REGISTER

TABLE 14-14: EMBEDDED FLASH CONFIGURATION REGISTER

HOST INDEX	N/A			N/A			**HOST SIZE**	
EC OFFSET	110h			32/16/8-bit			**EC SIZE**	
POWER	VTR			00h			nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE[3:2] BIT	**D31**	**D30**	**D29**	**...**		**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Reserved		INHIBIT_ JTAG	Data_ Protect	Boot_ Protect_E n	Boot_ Lock	Host_ Ctl	Reg_ Ctl_En

Note: This register is accessible only by the EC.

REG_CTL_EN

When this bit is set, Reg_Ctl of the Embedded Flash Command Register can be set to 1. When this bit is clear (the default), Reg_Ctl is forced to 0 and cannot be set.

Because this bit overrides any writes to Reg_Ctl, EC firmware can clear this bit to prevent the Host from getting access to the Embedded Flash register interface.

HOST_CTL

When this bit is set, the Embedded Flash Address Register, the Embedded Flash Data Register and the Embedded Flash Command Register can be read and written by the Host via the Flash Memory Mailbox registers in the Flash Mailbox interface. The EC is inhibited from reading or writing these registers. Writes have no effect and all reads return all 0’s.

© 2012-2018 Microchip Technology Inc.

When this bit is clear (the default), these registers can be read and written by the EC. The Host is inhibited from reading or writing these registers. Writes have no effect and all reads return all 0's.

The Embedded Flash Status Register is always readable by both the Host and the EC, independent of the state of Host_Ctl.

BOOT_LOCK

The Boot_Lock bit permits the EC to lock the Flash boot block. When Boot_Lock is '1b', the Flash boot block is locked. Any attempt to write data in the address range 00000h through 00FFFh in the Embedded Flash address space will fail and set Busy in the Embedded Flash Status Register. When Boot_Lock is '0b' (the default), the Flash boot block is unlocked.

BOOT_PROTECT_EN

When this bit is set, the Boot Protect function is enabled. The first time the EC does either a program fetch to an address that is larger than 00_0FFFH or a data reference to an address that is larger than 00_0FFFh and less than 80_0000h causes Boot_Block in the Embedded Flash Status Register to be set and all further access to the Boot Block will be prohibited. See Section 14.10.2, "Flash Boot Protect," on page 251.

DATA_PROTECT

If this bit is '1b', the top two 2KB pages in the Embedded Flash array are not readable or writable through the Host access mailbox. The two pages are accessible by the EC through both the Instruction Memory Interface and the Register Interface, as long as the JTAG port is not enabled. Once JTAG is enabled on the pins, Data_Block in the Embedded Flash Status Register is set and the two pages become inaccessible to all interfaces.

If this bit is '0b' (the default), the top two pages can be read and written normally.

INHIBIT_JTAG

When this bit is '1b', the JTAG interface is blocked from any access to the EC or the internal buses in the MEC1632. Only the registers in the JTAG interface are accessible. If this bit is '0b', accesses by the JTAG interface is blocked if is Boot_JTAG_Block n the Embedded Flash Initialization Register is '0' and Boot_Block in the Embedded Flash Status Register is '0'.

14.12.6 EMBEDDED FLASH INITIALIZATION REGISTER

The Embedded Flash Initialization Register is a read-only register that is loaded from address 0_0FFCh in the Embedded Flash Array when VTR Power On Reset is de-asserted but while the EC is still held in Reset. The address is the last 32-bit word in the Boot Block.

This register is reset by hardware to be all 0's. The Boot_JTAG_Block bit will be 0, which blocks JTAG access to the ARC address space. If the read of the Flash is successful, the value of this register will be FFFF_FFFFh when the Flash is fully erased. In this state JTAG is not blocked.

TABLE 14-15: EMBEDDED FLASH INITIALIZATION REGISTER

HOST INDEX	N/A				N/A		**HOST SIZE**	
EC OFFSET	114h				32/16/8-bit		**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
BUS	LPC SPB							
BYTE3 BIT	**D31**	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Initial_Data[31:24]							
BYTE2 BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Initial_Data[23:16]							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Initial_Data[15:8]							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Initial_Data[7:1							Boot_ JTAG_ Block

Note: This register is accessible only by the EC.

BOOT_JTAG_BLOCK

If this bit is '0b' the JTAG interface is blocked from any access to the MEC1632 as long as Boot_Block in the Embedded Flash Status Register is '0b'. This means that as long as the Boot Block is accessible by the EC an external JTAG function cannot read or write any address inside the device, including the Boot Block. Once boot code renders the Boot Block inaccessible by causing Boot_Block to be set, JTAG functionality is enabled. JTAG cannot read or write any data in the Embedded Flash Boot Block, including the data used to load the Embedded Flash Initialization Register.

If this bit is '1b' (the value that will be loaded if the Boot Block is erased), JTAG access will be enabled as soon as VTR POR is de-asserted.

BITS[31:1] INITIAL_DATA

The data in this field are loaded after VTR POR is de-asserted, along with Boot_JTAG_Block. The data can be used for device identification or configuration.

© 2012-2018 Microchip Technology Inc.

14.12.7 FLASH MAILBOX INDEX REGISTER

TABLE 14-16: FL_MBX_INDEX REGISTER

HOST OFFSET	00h					8-Bit	**HOST SIZE**	
EC OFFSET	00h					8-Bit	**EC SIZE**	
POWER	VTR					00h	**VCC POR DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	-	-	-	-	-	-	-	-
BIT NAME	INDEX7	INDEX6	INDEX5	INDEX4	INDEX3	INDEX2	INDEX1	INDEX0

14.12.8 FLASH MAILBOX DATA REGISTER

TABLE 14-17: FL_MBX_DATA REGISTER

HOST OFFSET	04h					8-Bit	**HOST SIZE**	
EC OFFSET	04h					8-bit	**EC SIZE**	
POWER	VTR					00h	**VCC POR DEFAULT**	
BUS	LPC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	-	-	-	-	-	-	-	-
BIT NAME	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0

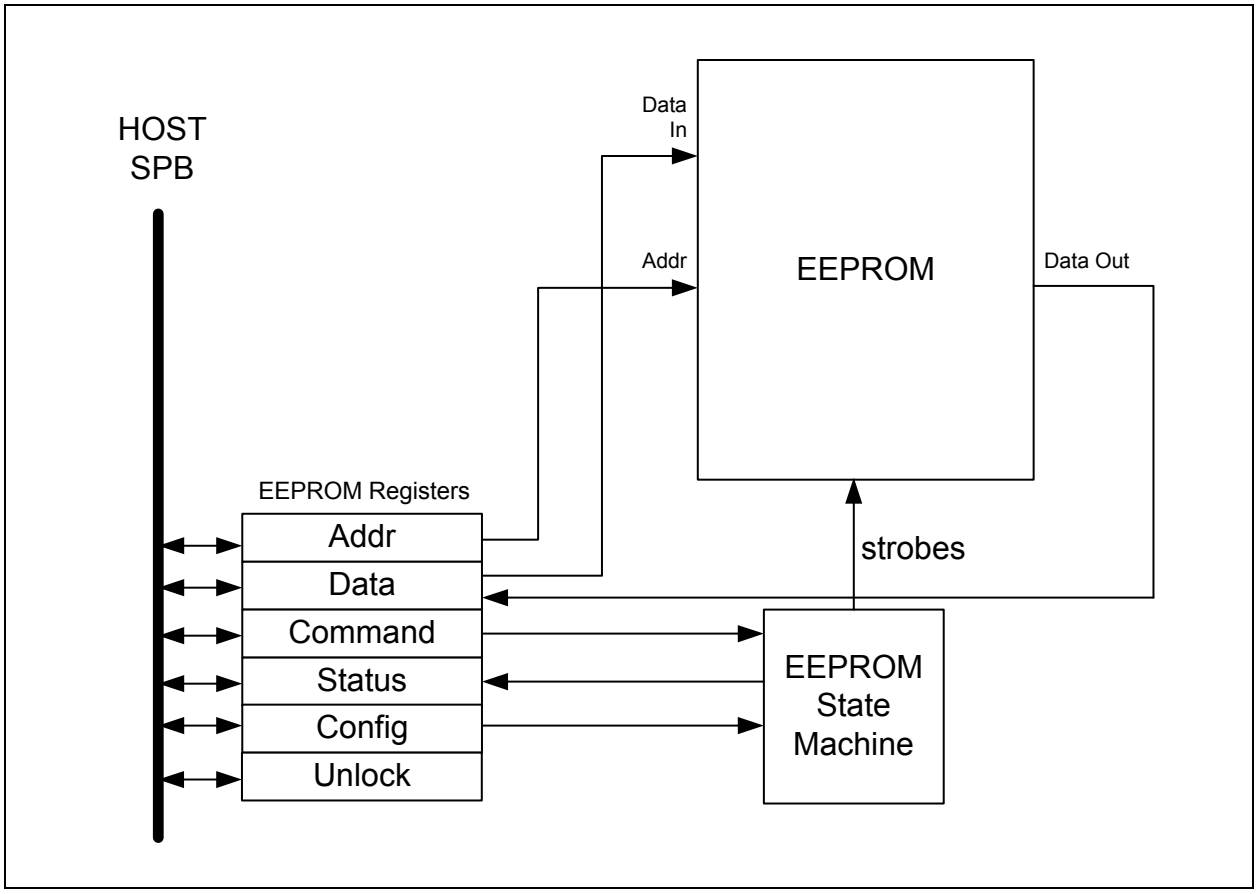
15.0 EEPROM

15.1 General Description

The MEC1632 includes a 2K x 8bit EEPROM (Electrically Erasable Programmable Read Only Memory).

15.2 Block Diagram

FIGURE 15-1: EEPROM BLOCK DIAGRAM



15.3 Port List

TABLE 15-1: EEPROM PORT LIST

Signal Name	Direction	Description
EEPROM_SLP_EN	INPUT	Indicates that the system is trying to shut down the ring oscillator. This bit is implemented in the EC Blocks Sleep Enables Register 1. Please refer to Section 7.8.4.2, "EC Blocks Sleep Enables Registers," on page 137 for details.
EEPROM_CLK_REQ	OUTPUT	Indicates when the EEPROM is ready to sleep (have clocks removed)

© 2012-2018 Microchip Technology Inc.

15.4 Power, Clocks and Reset

15.4.1 POWER DOMAIN

This block is powered by the VTR Power Supply.

See Section 7.0, "Power, Clocks, and Resets," on page 95 for details on power domains.

15.4.2 CLOCKS

This block uses the 20.27 MHz MCLK. All EEPROM signal timing is derived from the MCLK.

See Section 7.4, "Clock Generator," on page 98 for details on clocks.

15.4.2.1 Clock Idle

The EEPROM controller will keep the internal oscillator operating as long as the controller is not in the Standby state. This permits the controller to complete any program or erase operation even though the Embedded Controller may be in its sleep state.

The Section 7.0, "Power, Clocks, and Resets," on page 95 and the EEPROM interact to determine when the MEC1632 can stop the oscillator.

FIGURE 15-2: EEPROM_CLK_REQ INTERFACE

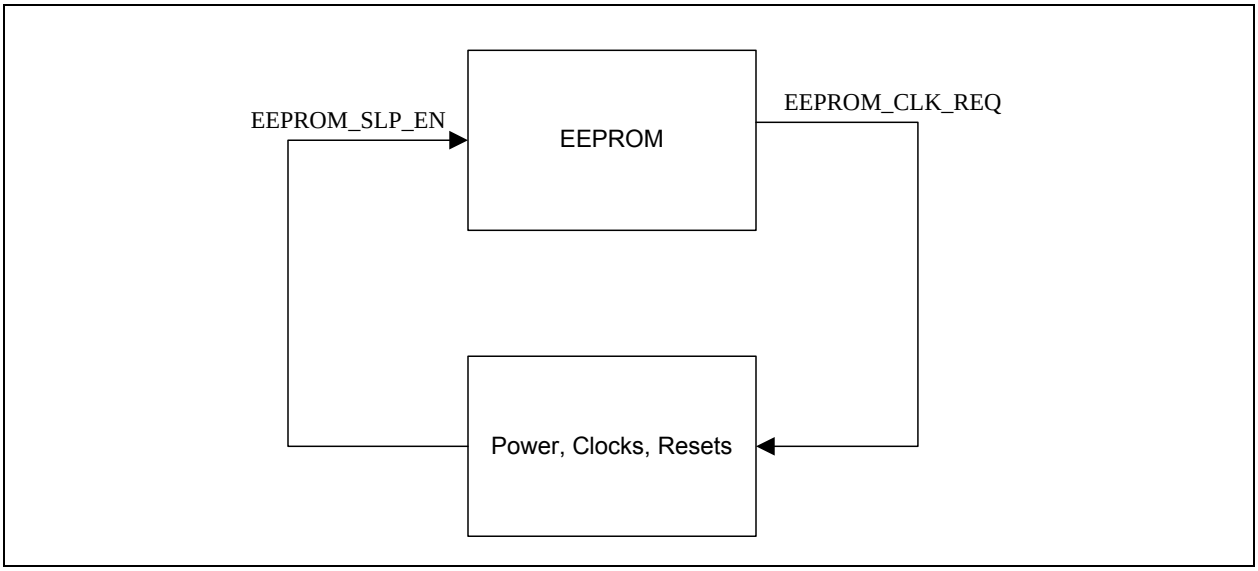


TABLE 15-2: EEPROM POWER MANAGEMENT

EEPROM_SLP_EN	EEPROM Mode	EEPROM_CLK_REQ	Power	Description
0	x	1	High power state	EEPROM clock is required because ARC clocks are not disabled and the block has not been commanded to sleep.
1	Not Standby Mode	1	High power state	EEPROM clock is required because the block is not idle.
1	Standby Mode	0	Low power state	A sleep command has been asserted, the block is idle and the core clocks are stopped.

15.4.3 RESET

This block is reset by nSYS_RST. Following a reset, all registers are set to their default values, and the internal state machines are reset to the standby state.

See Section 7.6, "Reset Interface," on page 124 for details on reset.

The JTAG interface registers referenced in this chapter have an asynchronous reset. See 15.5 on page 268.

15.5 Interrupts

The EEPROM can generate interrupts to the EC for two events which are reported in the two error bits in the EEPROM Status Register: CMD_Err, and Busy_Err. The error bits are routed onto the EEPROM_CMD_ERR and EEPROM_BUSY_ERR bits of the GIRQ14 Source Register.

15.6 EEPROM Operation

The EEPROM consists of a 2K array of bytes, organized into 8-byte pages. An erase operation in the EEPROM sets the affected memory array bits to one, while program operations write zeros. To reprogram any '0' bit in a page to '1,' the page must be erased. Individual bytes can be read and programmed; erasure occurs only on an 8-byte page.

The EEPROM erases and programs with a 3.3V power supply; its IO interface operates at 1.8V. To modify the contents of the EEPROM, VTR must be >3V before program or erase operations may begin. A summary of the MEC1632 EEPROM features is shown below in Table 15-3.

TABLE 15-3: MEC1632 2KB EEPROM FEATURE SUMMARY

Feature		Description
PROG/ERASE VOLTAGE		3.3V ± 10% (T_J = 0×C to 125×C)
READ VOLTAGE		1.8V ± 10% (T_J = 0×C to 125×C)
BUS WIDTH		8-bit
READ ACCESS/CYCLE TIME		40 ns max
MAIN MEMORY BLOCK		2K x 8 bits
ERASE	TYPES	Page/Mass (8 bytes/page)
	Program/Erase Cycles	250,000 min
	Erase time, 8-byte page	5 ms max
	Erase time, mass erase	40 ms max
PROGRAMMING		Per 8-bit byte
INTERFACE		All Program and Erase Operations are Enabled via a Command Sequence Interface using the EEPROM EC Register Interface.

15.6.1 EEPROM OPERATION

15.6.1.1 EEPROM Initialization

The EEPROM controller reads from offsets 3FCh to 3FFh from the EEPROM Memory Block and loads the least significant 31 bits of the data into an internal register. If the low 31 bits are not 7FFF_FFFFh, and bit 31 is a '1,' the EEPROM_BLOCK bit in the EEPROM Status Register is set to '1' and the EEPROM Operation Memory Block becomes inaccessible.

15.6.1.2 Password Use

Offset 3FCh is the last 32-bit word in the EEPROM memory block. If bit 31 is a '0' in that word, then the location can be treated as EEPROM data that stores only positive integers in the range 0 to 2 billion (or so). If bit 31 is a '1' and the rest of the word is not all 1's (that is, if the word is not in its initial, erased, state), then the word contains a password that can be used to unlock the EEPROM memory block.

© 2012-2018 Microchip Technology Inc.

To keep the EEPROM memory block secure, and to insure that the EC firmware is authorized to read its contents, the EC firmware can use this password. The following code can be executed as part of the 4KB Boot Block portion of the main flash array:

```
int password;
#define SECRET_PASSWORD 0xXXXXXXXX
#define Unlock_register 0xF02C20 // address of key register

password = eeprom_read(0x3FC);
if( password == 0xFFFFFFFF )
{
    // EEPROM is unlocked and the key not yet installed
    // establish key and force a reboot
    eeprom_write(0x3FF, SECRET_PASSWORD);
    force_reset();
} else {
    // unlock the EEPROM
    *Unlock_register = SECRET_PASSWORD;
}
```

The subroutines force_reset() forces a system reset, using, for example, the watchdog timer. If the EC firmware knows the right password, it can unblock the EEPROM Operation memory block; if it does not, then the block will remain inaccessible. The code can run in the Boot Block so that the SECRET_PASSWORD (a constant in the code) cannot be read via JTAG or over the LPC bus, and thus can only be known to a valid EC firmware block.

15.6.1.3 Flash Emergency Mass Erase

The Emergency Mass Erase function, when invoked, erases the EEPROM as well as the Flash Memory Array.

15.6.2 REGISTER INTERFACE

The EEPROM has its own Logical Device Number and Base Address as indicated in Table 15-4.

TABLE 15-4: EEPROM BASE ADDRESS TABLE

EEPROM Instance	LDN	AHB Base Address
EEPROM	Bh	F0_2C00h

Table 15-5, “EEPROM Register Summary,” on page 269 summarizes the registers allocated for the EEPROM.

15.6.3 EC REGISTER INTERFACE

The following table summarizes the registers allocated for the EEPROM. The offset field in the following table is the offset from the EC Base Address.

TABLE 15-5: EEPROM REGISTER SUMMARY

	EC Interface		
Register Name	**SPB Offset**	**EC Type**	nSYS_RST
EEPROM Data Register	00h	R/W	0000_0000h
EEPROM Address Register	04h	R/W	0000_0000h
EEPROM Command Register	08h	R/W	0000_0000h
EEPROM Status Register	0Ch	R/WC	0000_0000h
EEPROM Configuration Register	10h	R/W	0000_0000h
EEPROM Unlock Register	20h	W	–

15.6.4 EEPROM ADDRESS AND DATA REGISTERS

The EEPROM Data Register and the EEPROM Address Register are each implemented as a two entry FIFO, as illustrated in Figure 15-3, "EEPROM Controller Address and Data registers". The figure is suggestive and does not represent the precise implementation. Writes are always directed to the Tail register of the FIFO and reads are always sourced from the Head register. Each of the two registers has an associated Valid bit (**V** in the figure). The Tail Valid bit

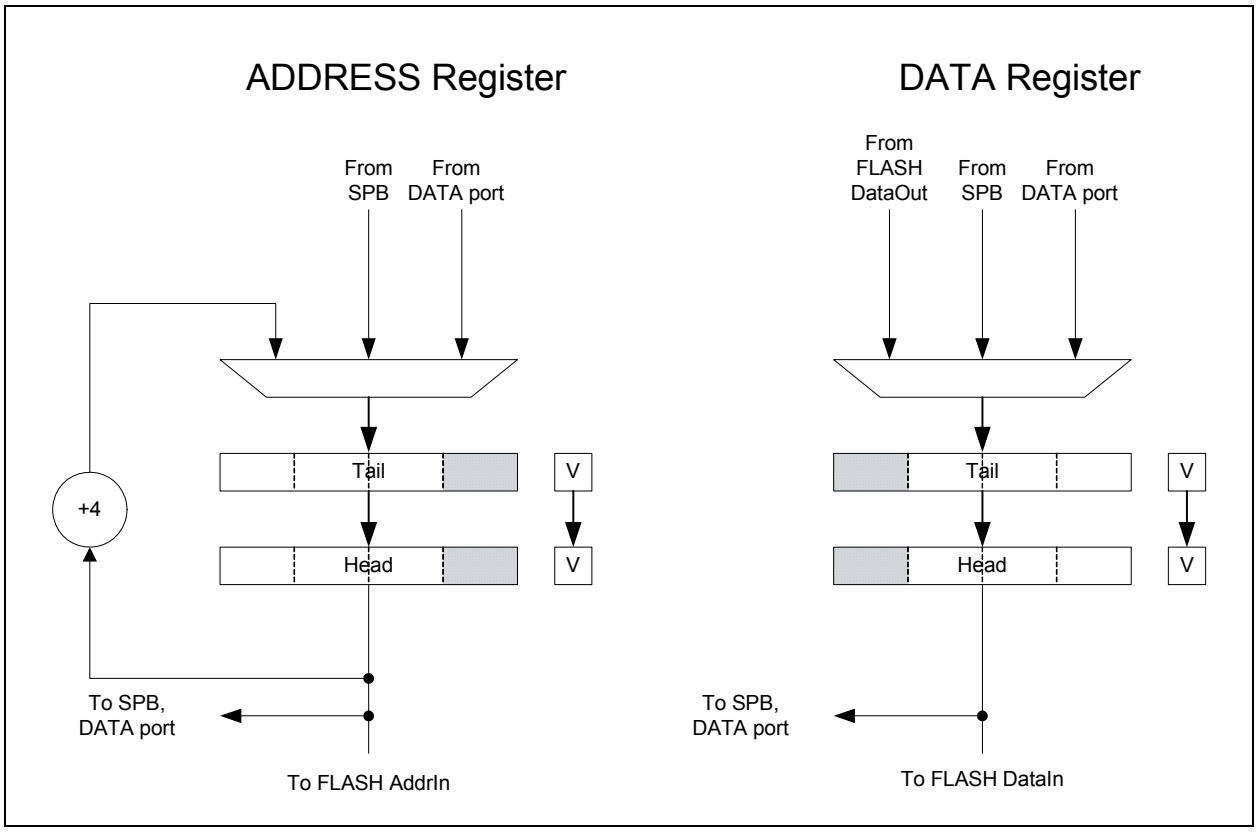
is set when the full Tail register is written and cleared when the Tail register is copied into the Head register. The Head Valid bit of the EEPROM Data Register is cleared whenever the full Head register is read by the EC, or when the EEPROM Controller completes a Program transaction.The Head Valid bit of the EEPROM Address Register is cleared whenever the EEPROM controller completes a Read, Program or Erase function. Reading the EEPROM Address Register by an EC register read does not affect the Head Valid bit. If the Head Valid bit is cleared and the Tail Valid bit is set, the Tail register is copied into the Head register, the Tail Valid bit is cleared and the Head Valid bit is set. All Valid bits are cleared when EEPROM_MODE of the EEPROM Command Register is set to Standby Mode.

Writing the EEPROM Address Register sets the Valid bit on the Tail of the Address FIFO, while writing the EEPROM Data Register sets the Valid bit for the Tail of the Data FIFO. Reading the EEPROM Data Register clears the Valid bit for the Head of the Data FIFO. As stated above, reading any byte of the EEPROM Address Register does not cause a change in the Head Valid bit.

When the Valid bits of both the Head and Tail of the EEPROM Data Register are set, Data_Full in the EEPROM Status Register is set. If either bit is cleared, Data_Full is cleared. When the Valid bits of both the Head and Tail of the EEPROM Address Register are set, Address_Full in the EEPROM Status Register is set. If either bit is cleared, Address_Full is cleared.

The requesting Master will be stalled while attempting to read the EEPROM Data Register if the EEPROM controller is Busy and in the process of reading data into the EEPROM Data Register. This will result in at most one EC wait state.

FIGURE 15-3: EEPROM CONTROLLER ADDRESS AND DATA REGISTERS



After initialization during normal operation, the EEPROM controller is placed into one of four modes (Standby Mode, Read Mode, Program Mode or Erase Mode) by setting EEPROM_MODE of the EEPROM Command Register to the appropriate values (0, 1, 2 or 3, respectively). Any write to the EEPROM Command Register while the controller is busy (when Busy of the EEPROM Status Register is '1b') will not modify the state and will set Busy_Err in the EEPROM Status Register.

© 2012-2018 Microchip Technology Inc.

15.6.5 STANDBY MODE

When in this mode all strobes to the EEPROM Memory Array are de-asserted and the EEPROM Memory Array is placed in its lowest power state. Both the EEPROM Data Register FIFO and the EEPROM Address Register FIFO are flushed. The registers can be read and written by software without error. Data returned on a read of a flushed FIFO is undefined. On a transition from Standby Mode to Program Mode, Read Mode or Erase Mode the FIFOs will be invalid and no action dependent on valid data in a FIFO will take place until new data is written into the registers.

The EEPROM controller must return to Standby Mode after any other state. An attempt to set the controller into a state other than Standby Mode when the controller is in Program Mode, Read Mode or Erase Mode will put the controller in Standby Mode and set CMD_Err of the EEPROM Status Register. EEPROM_MODE of the EEPROM Command Register will also be left in Standby Mode. If the controller is busy, writes to the EEPROM Command Register are ignored and Busy_Err in the EEPROM Status Register is set.

15.6.6 READ MODE

When the EEPROM controller is in Read Mode the addressing strobes are kept de-asserted as long as no Read is in progress. When the strobes are de-asserted, the EEPROM array is maintained in its lowest power state (equivalent to Standby Mode). When the Valid bit associated with the Head register in the EEPROM Address Register is '1b' and the Data FIFO is not full, the data in the EEPROM array that corresponds to the EEPROM Address Register is loaded into the Tail register of the EEPROM Data Register. The Data FIFO can hold two read results.

The Address FIFO is advanced when the EEPROM controller completes a read from the EEPROM array and stores the result in the Data FIFO. When the EEPROM Data Register is read, the Data FIFO is advanced. A read of the will return the value of the Head register but will not advance the FIFO. A read of the EEPROM Data Register when the FIFO is empty but the controller is busy reading from the array will stall the read until the controller has completed the lookup and the data can be returned to the requester.

Reads of data from the EEPROM take a single cycle.

15.6.6.1 Burst Read Mode

When Burst in the EEPROM Command Register is set to '1b', the Read function automatically increments the EEPROM Address Register in order to minimize the time necessary to read a block of memory from the EEPROM. As in Read Mode, an EEPROM read is initiated when the Head of the Address FIFO is Valid and the Data FIFO is empty. Whenever the Tail register in the Address FIFO is not Valid and the Head register in the FIFO is Valid, the Head register is incremented by 1 and written into the Tail. The Address FIFO will thus contain a sequence of consecutive byte addresses as long as the EEPROM controller remains in Burst Read Mode.

Reading the EEPROM Data Register from either the EC or the JTAG interface always reads 8 bits at a time, so the Data Register FIFO is always advanced when the EC or JTAG does a data read. In Burst Read Mode the entire EEPROM memory can be read by repeatedly reading the EEPROM Data Register without the need for reading or writing any other register in the EEPROM Subsystem.

15.6.7 PROGRAM MODE

When the EEPROM controller is in Standby Mode, setting the EEPROM Command Register to Program Mode will set up the EEPROM array for programming. The EEPROM Data Register and the EEPROM Address Register FIFOs are used for Program Mode in a manner similar to their use in Read Mode. When the Head registers in both the Address and Data FIFO are Valid, the EEPROM strobes are sequenced in order to write the contents of the Data register into the EEPROM Memory at the address specified in the Address register. At the end of the Program sequence the two FIFOs are advanced.

Note:	In Program Mode there should always be an erase before writing the same byte twice.

15.6.7.1 Program Mode Timing Parameters

The following three sequences are used in programming the Embedded Flash: Program Prologue, Word Program, Program Epilogue.

During all three sequences, the controller asserts Busy while the sequence is in progress. Flash programming time is shown in Table 15-6.

TABLE 15-6: EEPROM PROGRAMMING TIME

Parameter	Symbol	Value			Units	Notes
		MIN	TYP	MAX		
Single Byte Programming Time	–	19	–	–	µs	
Burst Byte Programming Time (single page)	–	13.5	–	–	µs	

The Program Prologue sequence is issued before the first time any byte in a page can be programmed. The Program Epilogue sequences is issued after the last time any byte of a page is programmed. The EEPROM Flash controller automatically issues the Program Prologue and Program Epilogue sequences as required.

15.6.7.2 Burst Program Mode

Burst Program Mode is enabled whenever Burst in the EEPROM Command Register is set and the controller is in Program Mode. The behavior is similar to Burst Read Mode. When Burst is enabled, the EEPROM Address Register FIFO is always kept filled automatically with incrementing addresses. Whenever the Tail register in the Address FIFO is not Valid and the Head register is valid, the Head register is incremented by 1 and stored in the Tail. The controller otherwise behaves as in Program Mode. When both the Head register of the Address FIFO and the Head register of the Data FIFO are Valid, a Program sequence is initiated (with the possible addition of Program Prologue and Program Epilogue, as described above). When the programming sequence is complete, the Head registers of both FIFOs are marked not Valid. With this mechanism the entire EEPROM array can be programmed with a sequence of writes to the EEPROM Data Register without any writes to the EEPROM Address Register or EEPROM Command Register.

15.6.8 ERASE MODE

When the EEPROM Command Register is set to Erase Mode, the state machine will wait until a valid address is written into the EEPROM Address Register. A valid address consists of the address of an 8-byte page; the low 3 bits of the address register should be 0. The remaining 8 bits of the Address specify one of the 256 pages in the EEPROM array. Once the register is valid, Busy in the EEPROM Status Register is set to ‘1b and the EEPROM controller sequences the EEPROM array strobes to erase part or all of the EEPROM. All bits in the erased area are set to ‘1b’. Busy remains set during the operation. When the erase operation is complete Busy is set to ‘0b’. The controller remains in Erase Mode and can accept additional addresses in the EEPROM Address Register.

The Mass_Erase[4:0] field of the EEPROM Address Register selects between the 8-byte page erase and mass erase of the entire array. When the field is all 1’s (1Fh), any erase operation erases the entire EEPROM array, rather than just an 8-byte page.

15.6.9 EMERGENCY ERASE FEATURE

If the EC Boot Code becomes corrupted, an Emergency Mass Erase operation may be performed on the Embedded Flash. See Section 14.10.5, "Emergency Mass Erase," on page 252. When the embedded flash is erased, the EEPROM is also erased.

© 2012-2018 Microchip Technology Inc.

15.7 Programming the EEPROM

Unlike the main Flash array, the EC can access the EEPROM when executing out of Flash or SRAM.

15.7.1 READING THE EEPROM

The EEPROM memory can be read with Burst mode set. An initial EEPROM address should be configured, and then the rest of the EEPROM memory can be read with just a sequence of reads to the data register. The following pseudo-code illustrates how reading the EEPROM might proceed:

```
// Byte_array[] is a data structure to receive the data. It is a sequence of bytes
// EEPROMbase is the first address in the EEPROM memory to be read
// Limit is the total number of bytes to read


// Array[] is a data structure to receive the data.
// EEPROM_Base is the first address in the EEPROM memory to be read
// Limit is the total number of 8-bit bytes to read
// EEPROM_Address is the AHB address of the EEPROM Address Register
// EEPROM_Data is the AHB Address of the EEPROM Data Register
//
// this code works as long as the read does not wrap around the end of the EEPROM
array
//
*Command_Address = READ | BURST;
*EEPROM_Address = EEPROM_Base;
for( i = 0; i < Limit; i++)
{
    Array[i] = *EEPROM_Data;
}
*Command_Address = STANDBY; // Exit READ mode and flush Address/Data FIFOs
```

15.7.2 WRITING THE EEPROM

The EEPROM memory can be programmed with Burst mode set. An initial EEPROM address should be configured, and then the rest of the EEPROM memory can be written with just a sequence of writes to the data register. The following pseudocode illustrates how reading the EEPROM might proceed:

```
// Array[] is a data structure of bytes that sources the data.
// EEPROM_Base is the first address in the EEPROM memory to be programmed
// Limit is the total number of 8-bit bytes to write.
// EEPROM_Address is the AHB address of the EEPROM Address Register
// EEPROM_Data is the AHB Address of the EEPROM Data Register
//Status_Address is the AHB address of the EEPROM Status Register
//
*Command_Address = PROGRAM | BURST;
*EEPROM_Address = EEPROM_Base;      // the EEPROM address advances in the Address FIFO
for( i = 0; i < Limit; i++)
{
    while( *Status_Address & DATA_FULL ); // stall while Data FIFO has no space
    *EEPROM_Data = Array[i];
}
*Command_Address = STANDBY; // Exit PROGRAM mode and flush Address/Data FIFOs
```

15.8 Detailed Description of Accessible Registers

15.8.1 EEPROM DATA REGISTER

The EEPROM Data Register can only be written when the EEPROM Controller is in Program Mode. In Standby Mode, Read Mode and Program Mode the register is read-only. Writes will complete but have no effect.

TABLE 15-7: EEPROM DATA REGISTER

HOST INDEX	N/A				N/A		**HOST SIZE**	
EC OFFSET	00h				8-bit		**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE3 BIT	**D31**	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE2 BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	EEPROM_Data[7:0]							

EEPROM_DATA[31:0]

This 8-bit register holds the data to be written into the EEPROM memory array during a program cycle, as well as the data returned from an EEPROM read during a read cycle. It should be set up before the EEPROM Address Register is configured.

© 2012-2018 Microchip Technology Inc.

15.8.2 EEPROM ADDRESS REGISTER

TABLE 15-8: EEPROM ADDRESS REGISTER

HOST INDEX	N/A				N/A		**HOST SIZE**	
EC OFFSET	04h				32/16/8-bit		**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE3 BIT	**D31**	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE2 BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Mass_Erase[4:0]					EEPROM_Address[10:8]		
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	EEPROM_Address[7:0]							

EEPROM_ADDRESS[10:0]

This register represents a byte address in the EEPROM. If the EEPROM state machine is in Read Mode, writing Byte0 of the EEPROM Address Register initiates the Read sequence. If the EEPROM state machine is in Program Mode, the Program sequence is initiated when there is both a valid Data value in the EEPROM Data Register and the EEPROM Address Register has been updated by writing Byte0.

The following provides the required decoding during Read Mode and Program Mode:

MASS_ERASE[4:0]

When this field is 1Fh, an erase operation erases the entire EEPROM array.

15.8.3 EEPROM COMMAND REGISTER

This register is Read-Only while Busy in the EEPROM Status Register is ‘1b’. An attempt to write this register while Busy is asserted will not modify the register and will set Busy_Err in the EEPROM Status Register.

A write to this register causes the EEPROM controller to transition to the selected state.

TABLE 15-9: EEPROM COMMAND REGISTER

HOST INDEX	N/A			N/A			**HOST SIZE**	
EC OFFSET	08h			32/16/8-bit			**EC SIZE**	
POWER	VTR			00h			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE[3:2] BIT	**D31**	**D30**	**D29**	**...**		**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R	R	R	R/W	R/W	R/W
EC TYPE	R	R	R	R	R	R/W	R/W	R/W
BIT NAME	Reserved					Burst	EEPROM_Mode	

EEPROM_MODE

The field determines which Master may write the registers in the EEPROM controller:

0 The EEPROM controller is placed in the Standby mode.

1 The EEPROM controller is placed in Read mode.

2 The EEPROM controller is placed in Program mode.

3 The EEPROM controller is placed in Erase mode.

BURST

If the EEPROM controller is in Read Mode or Program Mode and this bit is ‘1b’, the contents of the Head register in the EEPROM Address Register FiFO will be incremented by 1 and written into the Tail register whenever the Head register is Valid. When this bit is ‘0b’, the EEPROM Address Register and the INDEX register are not incremented automatically.

See Section 15.6.6.1, "Burst Read Mode," on page 271 and Section 15.6.7.2, "Burst Program Mode," on page 272 for information about the use of Burst.

© 2012-2018 Microchip Technology Inc.

15.8.4 EEPROM STATUS REGISTER

TABLE 15-10: EEPROM STATUS REGISTER

HOST INDEX	N/A				N/A		**HOST SIZE**	
EC OFFSET	0Ch				32/16/8-bit		**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE[3:2] BIT	**D31**	**D30**	**D29**	**...**		**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R/WC	R/WC
BIT NAME	Reserved						CMD_Err	Busy_Err
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	R	R	R	R	R	R	R	R
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	EEPROM_BLOCK	Reserved				Address_Full	Data_Full	Busy

BUSY

This bit reflects the state of the EEPROM controller. This bit is set while the controller is processing an EEPROM control sequence. While this bit is set the EEPROM Command Register can not be written. If a write is attempted on this register, the write fails and Busy_Err in this register is set.

This bit is read-only.

DATA_FULL

This bit reflects the value of the Valid bit of the Tail register of the EEPROM Data Register. When it is set the FIFO is full and any additional writes will set Busy_Err.

This bit is read-only.

ADDRESS_FULL

This bit reflects the value of the Valid bit of the Tail register of the EEPROM Address Register. When it is set the FIFO is full and any additional writes will set Busy_Err.

This bit is read-only.

EEPROM_BLOCK

When EEPROM_BLOCK is '1,' the EEPROM Operation is protected from all access. Once set, EEPROM_BLOCK will only be cleared by a VTR POR.

The EEPROM_BLOCK bit is read only.

BUSY_ERR

This bit is set if

- A write to the EEPROM Command Register occurs while Busy is set.
- A write to the EEPROM Address Register occurs while Address_Full is set.
- A write to the EEPROM Data Register occurs while Data_Full is set

This bit is sticky: once set, it remains set until cleared by a write with the value ‘1b’ to this bit.

CMD_ERR

If the EEPROM controller is in Read Mode, Program Mode or Erase Mode, this bit is set if the EEPROM Command Register is set to any value other than Standby Mode.

This bit is sticky: once set, it remains set until cleared by a write with the value ‘1b’ to this bit.

15.8.5 EEPROM CONFIGURATION REGISTER

TABLE 15-11: EEPROM CONFIGURATION REGISTER

HOST INDEX	N/A			N/A			**HOST SIZE**	
EC OFFSET	10h			32/16/8-bit			**EC SIZE**	
POWER	VTR			00h			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE[3:2] BIT	**D31**	**D30**	**D29**	**...**		**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R/W	R/W
BIT NAME	Reserved						EEPROM_FORCE_BLOCK	EEPROM_PROTECT

EEPROM_PROTECT

If EEPROM_PROTECT is ‘1,’ the EEPROM Operation is accessible through the register interface as long as JTAG is not enabled. Once JTAG is enabled on the pins, the EEPROM_BLOCK bit in the EEPROM Status Register is set to ‘1’ and the entire EEPROM Operation becomes both read and write protected.

EEPROM_FORCE_BLOCK

Writing the EEPROM_FORCE_BLOCK bit to ‘1’ sets the EEPROM_BLOCK in the EEPROM Status Register to ‘1’ locking down the EEPROM Operation. The EEPROM_FORCE_BLOCK is write-only.

© 2012-2018 Microchip Technology Inc.

15.8.6 EEPROM UNLOCK REGISTER

TABLE 15-12: EEPROM UNLOCK REGISTER

HOST INDEX	N/A			N/A			**HOST SIZE**	
EC OFFSET	20h			32/16/8-bit			**EC SIZE**	
POWER	VTR			00h			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE[3:1] BIT	**D31**	**D30**	**D29**	**...**		**D10**	**D9**	**D0**
HOST TYPE	–	–	–	–	–	–	–	–
EC TYPE	W	W	W	W	W	W	W	W
BIT NAME	EEPROM_UNLOCK							

EEPROM_UNLOCK

When the 31-bit EEPROM_UNLOCK register is written, the least significant 31 bits of the write are compared to the internal register that stores the key. If all bits match, the EEPROM_BLOCK bit in the EEPROM Status Register cleared, and the EEPROM Operation array can be read or written.

The EEPROM_UNLOCK register is write only.

16.0 ARC 625D EMBEDDED CONTROLLER

16.1 General Description

This chapter contains a description of the Embedded Controller used in the MEC1632.

The Embedded Controller on the MEC1632 is an ARC 625D Processor, Revision 4.6, by ARC International. The ARC625D is a full-featured 32-bit embedded processor. Its features include:

- 5-stage instruction pipeline with single-cycle instruction execution
- Static branch prediction
- 32-bit data, instruction and address buses
- 16- and 32-bit instructions, with no overhead for switching between 16- and 32-bits
- 32 32-bit general purpose registers
- Scoreboarded data memory pipeline to reduce data stalls
- Debug features
  - Debug host can access all registers and CPU memory, with a JTAG interface to host tools
  - Multiple action points for real-time instruction and data breakpoints
- Industry standard AHB system interface
- Power saving features
  - Sleep mode via software instruction
  - Clock gating

Two highly configurable action points for debugging

SYNC instruction (added in ARC 625D Processor, Revision 4.6)

Delay Register, an extension register to the ARC

The ARC625D is highly configurable. The configuration used in the MEC1632 incorporates:

- 4K Boot ROM to download an (in-factory environment) SRAM-based EC application
- 192KB single cycle Embedded Flash Closely Coupled instruction memory
- 16-KB Single Cycle 32-bit wide dual-ported SRAM, accessible as both Closely Coupled Data Memory and Instruction Memory
- Interrupt controller with 32 interrupts
- Normalize instruction, which can find leading ones and zeros in a word
- Multiply instruction, which completes a 32x32 multiply in 3 cycles
- Divide Assist instruction
- Two full-featured Actionpoints (Dedicated Breakpoint Blocks), which can trigger breakpoints on both instruction accesses and data access

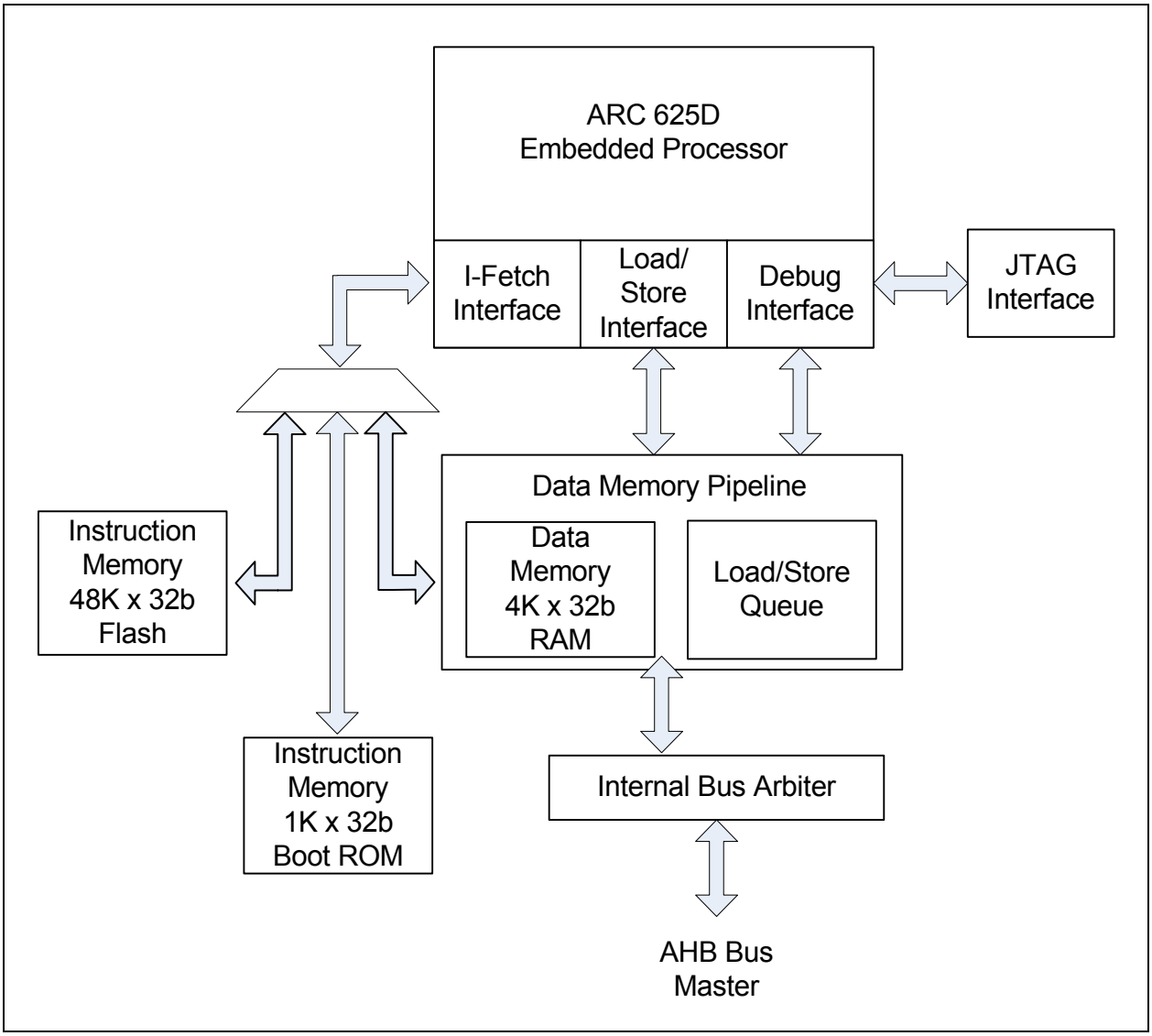
For details on the architecture of the ARC625D processor, see ARCompact™ Instruction Set Architecture Programmer's Reference, ARC International, April 2009.

Note:	See Section 1.1, "Tool Requirements" for tool requirements for the ARC 625D Processor.

© 2012-2018 Microchip Technology Inc.

16.2 Block Diagram

FIGURE 16-1: ARC BLOCK DIAGRAM



16.2.1 POWER ON RESET

This block is reset on a nEC_RST. See Section 7.6, "Reset Interface," on page 124 for details on reset.

The default location of the interrupt vector table for the ARC processor is now 010_0000h to access the Boot ROM. The Boot ROM downloads an SRAM-based EC application using the MEC1632 built-in serial UART. If no communication is established within a configurable timeout period, the Delay Register restores the UART to its default state and starts loading code from the first physical address of the embedded Flash.

16.3 EC Clocking

The ARC 625D Embedded Controller can be configured to run at various clock rates as described in Section 7.8.4, "Block Sleep Enable Registers," on page 136 (see Section 7.0, "Power, Clocks, and Resets," on page 95).

16.4 EC Memory Map

The ARC processor executes code out of the EC Instruction Memory. This instruction memory is an ARC ICCM (Instruction Closely-Coupled Memory) and each 32-bit word can be accessed in one processor cycle. Data references can come from either the EC Data Memory or from addresses located in the AHB Address Space. The EC Data Memory is an ARC DCCM (Data Closely-Coupled Memory), so each 32-bit word can read or written in one cycle.

See ARC Address Space on page 57 for further details on the ARC address space.

16.4.1 EC DATA MEMORY

The EC has a 16KB Closely Coupled Data memory, implemented with static RAM and organized 4K x 32 bits. Loads and stores to this memory are completed in one cycle. The base address of the memory is 80_0000h in the EC address space and extends to location 80_3FFFh. The EC cannot execute instructions from this address range.

The 16KB Data memory also appears in the instruction space in the address range 6_0000h through 6_3FFF as described in Section 16.4.2, "EC Instruction Memory".

Any data load/store to non-existent memory will return FFFF_FFFFh.

16.4.2 EC INSTRUCTION MEMORY

The primary instruction memory for the EC is a 48K x 32 bit Embedded Flash memory, located at locations 00_0000h through 02_FFFFh in the EC address space. The address range 10_0000h through 10_0FFFh is mapped to the Boot ROM. (See Section 16.2.1, "Power On Reset," on page 281)

Instruction fetches to these two blocks complete in one cycle. The ARC can access locations in the Embedded Flash Memory or the Boot ROM through load and store instructions.

The 16KB Data memory also appears in the instruction space in the address range 6_0000h through 6_3FFF. The memory is dual-ported, so instruction fetches from this space can occur in parallel with data loads and stores, without wait states for either instruction fetch or data reference. If the Embedded Flash memory is configured to be only accessible via the register interface (see Section 16.9, "EC Registers," on page 285), the EC can execute instructions out of the SRAM. For example, the EC could run code that programs the Flash while the Embedded Flash memory is set for the register interface.

RAM_Select in the AHB SRAM Configuration Register can be used to disable instruction access to the SRAM. If instruction access to the SRAM is not needed, disabling it saves power.

Instruction fetches in the range of 00_0000h through 7F_FFFFh do not incur bus errors. Any instruction fetch to non-existent memory will return FFFF_FFFFh.

16.5 ARC Pipelining

The ARC625D processor is pipelined with five pipe stages. Loads and stores are further pipelined through the Load/Store Queue as shown in Figure 17.1, “ARC Block Diagram”, so loads and stores will take additional cycles to complete. The AHB bus is also pipelined. Because of the different pipelines, it is difficult to determine exactly how long a load or store to a register will take if the register is located on either the LPC SPB bus or the EC SPB bus.

Because the ARC processor issues all instructions in order and resolves data hazards within the pipeline, software will typically not have to consider pipeline effects. Stores will complete in the order issued and no load instruction will return data until all stores issued previously have completed. However, there may be some situations in which it is necessary to make sure that pipelines have flushed and all stores have completed before further code execution. The following three instruction sequence ensures this outcome:

1. STORE to a memory location
2. LOAD from the same memory location to a processor register
3. Issue any instruction that uses the register in step 2) as one of its sources

The following assembly code is an example of the sequence:

```
; R0 = a value to be written to an AHB memory location
; R1 = the AHB address of the location to be written
;
ST R0, [R1,0]            ; store
LD R0, [R1,0]             ; load from same location
ADD R0, R0, 0 ; dummy instruction dependent on R0
```


© 2012-2018 Microchip Technology Inc.

16.6 ARC Extensions

Microchip provides the following extension registers to the ARC 625D Embedded Controller:

16.6.1 DELAY REGISTER

The Delay Register is an ARC Extension register in the Auxiliary registers Address Space. Writing a DELAY_VALUE to the Delay Register halts the ARC pipeline for approximately 1 µs to 32 µs. The delay is determined by the formula "*(*DELAY_VALUE*+1) × 21 × CLK_PERIOD*", where *CLK_PERIOD* is the cycle time of MCLK, or 49.33ns ±2%. A 1 µs delay is ensured to be at least 1 µs even when the master clock is running at its maximum speed. Valid DELAY_VALUE arguments are 0 - 31. Only the least significant five bits of the Delay Register are examined; all bits above bit 4 are ignored.

DELAY_VALUE arguments are written to the Delay Register with the ARC Auxiliary Register store instruction (`sr`). Loads of the Delay Register, with the ARC Auxiliary Register load instruction (`lr`) will return the last stored DELAY_VALUE. There is no delay associated with Delay Register loads.

While the instruction is holding up the pipeline, no new instructions can enter (for example, any interrupt processing is held for the duration of the auxiliary store instruction). Writes to the Delay Register produce no other side effects.

Address	FFFF_FFF9h in the Auxiliary registers Address Space.			
Bits	**Description**	**Type (Note 16-1)**	**Default**	**Reset Event**
31:5	RESERVED	RES	0	–
4:0	DELAY_VALUE The delay time in microseconds is the DELAY_VALUE + 1. Valid DELAY_VALUE arguments are 0 - 31.	R/W	0	nEC_RST

Note 16-1 reads of the Delay Register return the last written DELAY_VALUE.

16.6.2 EXAMPLE

An example subroutine that can delay an arbitrary number of microseconds is shown below in Figure 16-2.

FIGURE 16-2: DELAY REGISTER CODE EXAMPLE

```
#define DELAY_REGISTER 0xFFFFFFF9

void forced_delay( unsigned delay_in_microseconds )
{
    while( delay_in_microseconds > 32 )
     {
     _sr(31, DELAY_REGISTER);
     delay_in_microseconds -= 32;
     }
    _sr(--delay_in_microseconds, DELAY_REGISTER);
}
```

16.7 EC AHB Bus Interface

The ARC Embedded Controller has a single AHB Bus Master interface; see Section 4.3.2, "AHB Address Space," on page 59. The ARC can have at most one access pending on the AHB at one time. The ARC can perform 8-bit, 16-bit and 32-bit loads and stores on the AHB. Instruction fetches over the AHB can take the form of a 32-bit word load.

Possible AHB bus errors are described in Section 4.4.3, "AHB Bus Errors," on page 62. The ARC processor responds to a bus error with Memory Error exception. The first address that caused a memory error is recorded in the AHB Error Address Register. Because ARC exceptions are imprecise, and since several bus errors can occur between the time a bus error address is recorded and the time the ARC processes the exception, it is not always possible to determine which instruction caused the bus error.

16.8 Actionpoints (Dedicated Breakpoint Blocks)

Actionpoints are defined in the ARC 600 Ancillary Components Manual, Chapter 4. They are dedicated hardware blocks that provide an alternative source of breakpoints when the debugger cannot write to memory (e.g., the code being debugged is in ROM). They also provide the ability to break on memory or Aux register accesses.

The primary justification for including Actionpoints in the design is to provide breakpointing for code in ROM, while code is running at full speed (as opposed to being single-stepped). The debugger by default prefers to write Breakpoint instructions (BRK_S: 7FFFh) into memory in order to perform breakpoints at specified PC values. It will instead use actionpoints if:

- The memory area is declared as ROM,

  - or -

- The flag “-off=prefer_soft_bp“ is given to the debugger.

Actionpoints are controlled by a dedicated set of Aux Registers, in the range 220h - 237h, organized as 3 registers per actionpoint. These are:

- AMV: A 32-bit value (Address or sometimes Data). This register supplies the initial trigger value, as masked by the AMM register. Upon triggering, it is over-written by hardware with the exact value seen.
- AMM: A 32-bit mask applied to the AMV register, making any desired bits don’t-cares.
- AC: Control register, selecting modes

The status of all actionpoints is visible in the Aux Register DEBUG, at 5h.

The MEC1632 incorporates two full-featured ARC Actionpoints, Actionpoint 0 and Actionpoint 1.

16.8.1 ACTIONPOINT CONFIGURATIONS

Actionpoints may configured (at processor HW build) to be one of two configurations, Minimal or Full.

Minimal actionpoints may be configured to do the following:

- Trigger on an access by address and access type (Instruction, Bus access, or Aux Register)
- Instruction breakpoints trigger on execution at the address, not at the fetch itself
- Act by either Halt (debugger acts) or SW Interrupt (target SW acts)
- Qualify between Reads and Writes (or both)
- Qualify by masking bits of the address
- Invert Condition (Trigger if No Match)
- Gang actionpoints in pairs or quads: both/all must match

Full actionpoints add the following capabilities:

- Match on opcode for instruction fetches
- Match on data value for data read/write in Aux registers or memory
- Two 34-bit inputs from arbitrary sources (per actionpoint)

16.8.2 SIGNIFICANT LIMITATIONS

Address ranges may degrade performance. Because address matching is bit-masked, it may take multiple actionpoints to refine an address range. Even then, the final range is liable to be too big. The debugger allows a range to be too big, and continues from the breakpoint if the resulting trigger was not in the desired range. Note that this means that the

© 2012-2018 Microchip Technology Inc.

program was being halted at undesired / unexpected times, and so is not running at full speed. A reliable way to avoid this is to specify a range only as a power of 2 in size, aligned on a boundary that is also a power of 2, the same or larger than the range.

There is no way to trigger on both the value and the address of a bus read (Memory, I/O), because the data and the address are not present simultaneously. An Aux register access, however, can trigger on data when either read or written. Do not try to enable Read and Write in the same actionpoint, because that will select only the Write data bus to monitor.

16.8.3 DEBUGGER SUPPORT

As of version 8.7 of the Metaware toolset, the debugger supports:

1. Break on Instruction fetch by address
   - Actionpoint is used if ROM detected or “-off=prefer_soft_bp” argument is specified
2. Break on Memory Space data accesses
   - Read/Write or Both
   - Address and Mask
   - Range, if size is power of two and target aligned to a power of two (requires 2 Minimal actionpoints, paired)
   - Value and Mask (requires 2 Minimal actionpoints, paired)
   - Value and Range (requires 4 Full actionpoints, quadded)
3. Aux Register accesses
   - Read/Write or Both

16.9 EC Registers

TABLE 16-1: AHB SRAM CONFIGURATION REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC ADDRESS	F0_FC00h			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE[3-1] BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R/W
BIT NAME	Reserved							RAM_ Select

RAM_SELECT

When this bit is clear (the default case), the 16KB on-chip SRAM that is part of the EC can only be accessed by loads and stores starting at address 80_0000h. The EC can read and write data in the SRAM at addresses starting at 80_0000h but cannot directly execute instructions.

When this bit is set, the 16KB SRAM is configured to be simultaneously accessible in the address range address 6_0000h through 6_3FFFh. The EC can execute directly out of the SRAM. The EC can still read and write data in the SRAM, with no time penalty per load or store.

TABLE 16-2: AHB ERROR ADDRESS REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC ADDRESS	F0_FC04h			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE3 BIT	**D31**	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE2 BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
BIT NAME	EC_Addr[23:16]							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
BIT NAME	EC_Addr[15:8]							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
BIT NAME	EC_Addr[7:0]							

EC_ADDR[23:0]

If an AHB bus error occurs as the result of an EC AHB bus access, the address that caused the error is held. Once an address is held, additional bus errors are ignored, so this register records the first AHB address that caused an AHB bus error. Any write to this register re-enables capturing AHB bus addresses.

© 2012-2018 Microchip Technology Inc.

17.0 EC INTERRUPT AGGREGATOR

17.1 General Description

The EC Interrupt Aggregator works in conjunction with the ARC625D processor's interrupt interface to handle hardware interrupts and exceptions.

Exceptions are synchronous to instructions, are not maskable, and have higher priority than interrupts. All three exceptions - reset, memory error, and instruction error - are hardwired directly to the processor.

Interrupts are typically asynchronous and are maskable. As shown in Figure 17-1, certain interrupts are connected to the processor, but the majority are connected to the EC Interrupt Aggregator. The latter latches, arbitrates, and forwards the highest-level active interrupt to the processor's IRQ3 interrupt input. It also generates a jump vector associated with the selected interrupt. This vector is made available in one of the processor core's registers and is used to address a location in the Interrupt Vector Table (in memory) that contains the address of the interrupt handler.

The aggregator provides four priority levels for incoming interrupts. The processor provides three: mid- and low priorities for interrupts and high priority for exceptions.

Interrupts classified as wake events can be recognized without a running clock, e.g., while the MEC1632 is in sleep state.

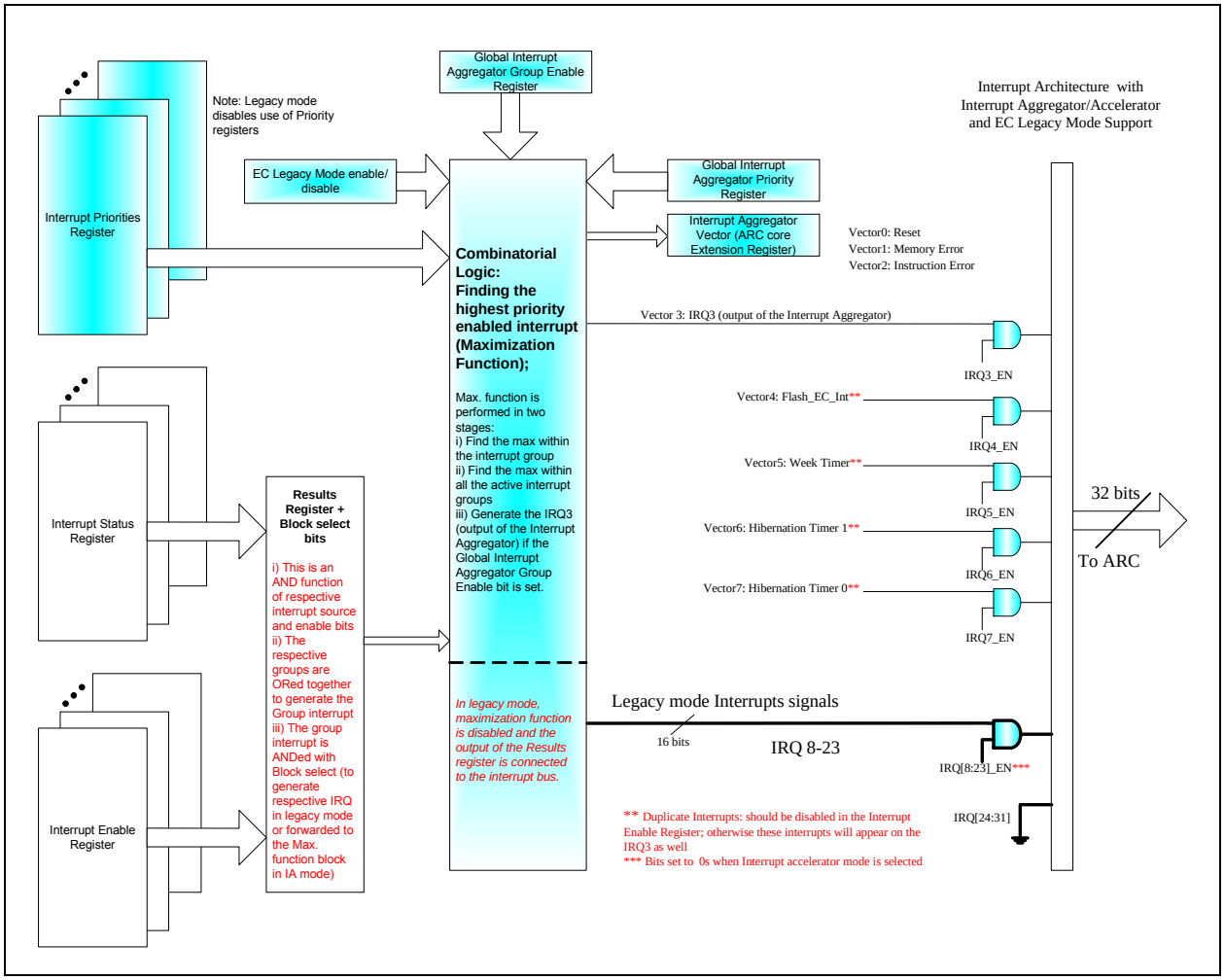
The EC Interrupt Aggregator can also operates in legacy mode to maintain compatibility with previous generation. In this mode it forwards up to 16 output interrupts to the processor's IRQ[8:23] but does not generate jump vectors.

This chapter focuses on the EC Interrupt Aggregator. Please refer to ARC International's *ARCompact™ Instruction Set Architecture Programmer's Reference*, March 2005 for more information on interrupt and exception handling by the ARC625D core.

Features of the EC Interrupt Aggregator:

- Edge-triggered inputs
- 4 priority levels
- 16 x 31 input interrupts (31 interrupts per group; 16 groups)
- Wake interrupts recognized while clock is stopped
- Programmable base address of the Vector Table
- Assist fast interrupt handling by software
  - Provides interrupt's jump vector in processor's extension register for fast access to Interrupt Vector Table
  - Support for NORM instruction to quickly locate the active highest-level interrupt

FIGURE 17-1: ARCHITECTURAL VIEW OF INTERRUPT AGGREGATOR



17.2 Interrupt Summary

Table 17-1, "EC Interrupt Structure" summarizes the ARC interrupts, priorities and vector locations.

- Link registers (ILINK1 / ILINK2) are the processor's registers that hold the value of the next PC when an interrupt occurs.
- Inside the processor, exceptions have HIGH priority and interrupts have MID or LOW priority. Within a priority level, a higher numbered interrupt has higher priority. For example, the Flash_EC_Int Interrupt with relative priority L26 has higher priority than the Week Timer's Interrupt with a relative priority L25. An exception is interrupt #7, which always has highest priority within its level.
- Byte offset: The ARC processor implements a table of jumps rather than interrupt vectors. When an interrupt occurs, the processor jumps to fixed addresses in memory, which contain a jump instruction to the interrupt handler. Byte offsets are vector offsets to the jump table.

Details on processor handling of interrupts can be found in the ARCompact™ Instruction Set Architecture Programmer's Reference, ARC International, April 2009.

© 2012-2018 Microchip Technology Inc.

TABLE 17-1: EC INTERRUPT STRUCTURE

Vector	Name	Link Register	Priority (Default)	Relative Priority	Byte Offset
0	Reset	-	High	H1	0x00
1	Memory Error	ILINK2	High	H2	0x08
2	Instruction Error	ILINK2	High	H3	0x10
3	Interrupt Aggregator	ILINK1	level 1 (low)	L27	0x18
4	Flash_EC_Int	ILINK1	level 1 (low)	L26	0x20
5	Week Timer	ILINK1	level 1 (low)	L25	0x28
6	Hibernation Timer 1	ILINK2	level 2 (mid)	M2	0x30
7	Hibernation Timer 0	ILINK2	level 2 (mid)	M1	0x38
8	IRQ8	ILINK1	level 1 (low)	L24	0x40
9	IRQ9	ILINK1	level 1 (low)	L23	0x48
10	IRQ10	ILINK1	level 1 (low)	L22	0x50
11	IRQ11	ILINK1	level 1 (low)	L21	0x58
12	IRQ12	ILINK1	level 1 (low)	L20	0x60
13	IRQ13	ILINK1	level 1 (low)	L19	0x68
14	IRQ14	ILINK1	level 1 (low)	L18	0x70
15	IRQ15	ILINK1	level 1 (low)	L17	0x78
16	IRQ16	ILINK1	level 1 (low)	L16	0x80
17	IRQ17	ILINK1	level 1 (low)	L15	0x88
18	IRQ18	ILINK1	level 1 (low)	L14	0x90
19	IRQ19	ILINK1	level 1 (low)	L13	0x98
20	IRQ20	ILINK1	level 1 (low)	L12	0xA0
21	IRQ21	ILINK1	level 1 (low)	L11	0xA8
22	IRQ22	ILINK1	level 1 (low)	L10	0xB0
23	IRQ23	ILINK1	level 1 (low)	L9	0xB8
24	IRQ24 (tied low)	ILINK1	level 1 (low)	L8	0xC0
25	IRQ25 (tied low)	ILINK1	level 1 (low)	L7	0xC8
26	IRQ26 (tied low)	ILINK1	level 1 (low)	L6	0xD0
27	IRQ27 (tied low)	ILINK1	level 1 (low)	L5	0xD8
28	IRQ28 (tied low)	ILINK1	level 1 (low)	L4	0xE0
29	IRQ29 (tied low)	ILINK1	level 1 (low)	L3	0xE8
30	IRQ30 (tied low)	ILINK1	level 1 (low)	L2	0xF0
31	IRQ31 (tied low)	ILINK1	level 1 (low)	L1	0xF8

© 2012-2018 Microchip Technology Inc.

17.3 Operation

17.3.1 REGISTER CONTROL OF INPUT INTERRUPTS

Associated with each interrupt are

- a Source bit that is set to indicate when an interrupt is active
- an Interrupt Enable bit to allow interrupt generation
- a Result bit to indicate when an enabled interrupt is active
- a priority level determined by its 2 Priority Level bits

Input interrupts are organized into groups; each group comprises 31 interrupts. Associated with each group is a set of 32-bit registers (Source, Enable, Result, Priority) described above. In addition, incoming interrupts can also be controlled on group basis, i.e., all interrupts in a group can be enabled / disabled by a bit in the Group Select register. This is summarized in Table 17-2.

TABLE 17-2: INTERRUPT AGGREGATOR GROUP REGISTERS

Register Name	Width (Bits)	Purpose
Interrupt Source	31	Latches asynchronous signals from on-chip devices
Interrupt Enable	31	Enables each interrupt
Interrupt Priority	62	A 2 bit priority for each of the 31 possible interrupt sources
Interrupt Result	31	Each bit is 1 if the corresponding Interrupt Source bit is 1, the interrupt is enabled, and the priority of the interrupt is equal to or greater than the current priority. The content of this register changes continuously, i.e., combinatorially, based on outputs from the other three registers.

The 31st bit (i.e. the most significant bit) of the Result register is to control the use of the NORM instruction. should set to 0. Setting this bit to '0' enables the use of the ARC NORM (normalize) instruction as a Find-First-One instruction (that is, NORM will return the bit number of the highest numbered bit that is a 1).

17.3.2 REGISTER CONTROL OF OUTPUT INTERRUPTS AND GLOBAL REGISTERS

Output interrupts to the processor, IRQ[23:03], are individually enabled. The lowest three LSBs, [2:0], are not used due the three exceptions (reset, memory error, instruction error) being directly connected to the processor.

Two global registers, IA Priority and IA Vector, are implemented as core extension registers. This means they can be used in any ARC instruction that can reference the full 6-bit core register number.

TABLE 17-3: INTERRUPT AGGREGATOR GLOBAL REGISTERS

Register Name	Width (Bits)	Purpose
IA Priority	3	The current priority level for the interrupt. A setting of 4 or higher blocks all interrupts from the Interrupt Aggregator. Interrupts at this priority level or higher will be allowed to be propagated to IRQ3. This is implemented as Core Extension register R55.
IA Group Enable	16	Enables individual Interrupt Group within IA
IA Vector	24	The address of a vector in memory. The vector is the address of an interrupt handler. for the interrupt selected by the Interrupt Aggregator. This is an ARC Core Extension Register R56.
IA IRQ Enable	20	Enable individual IRQ lines going into ARC

17.3.3 GENERATION OF INTERRUPT OUTPUTS AND JUMP VECTOR

1. The Interrupt Status register and the Interrupt Enable register respective bits are ANDed together.
2. All the bits in the same group are ORed
3. The resultant ORed bit is ANDed with the respective Block Select Register bit to generate the IRQi in **legacy mode**.
4. In the **accelerated mode**, the results of step 3 along with the priorities set in the GIRQx Interrupt Priority Registers are fed into the Maximization Function block to generate the highest priority interrupt that is propagated to IRQ3. The address of the respective interrupt handler is subsequently loaded in the IA Vector register.
5. Prior to sending the interrupts to the ARC, IRQi_EN is ANDed with the respective IRQi.

© 2012-2018 Microchip Technology Inc.

17.3.3.1 Priority Levels

The Interrupt Aggregator adds 4 levels of interrupt priority to the 2 levels the ARC provides. Each of the potential chip interrupts (16 interrupt source registers times up to 31 sources per register) can independently be assigned one of 4 priority levels. The Interrupt Priority Register in each group has a 2-bit priority field for each of the 31 possible interrupt sources. The 3-bit IA Priority Register sets a current priority level for all groups.

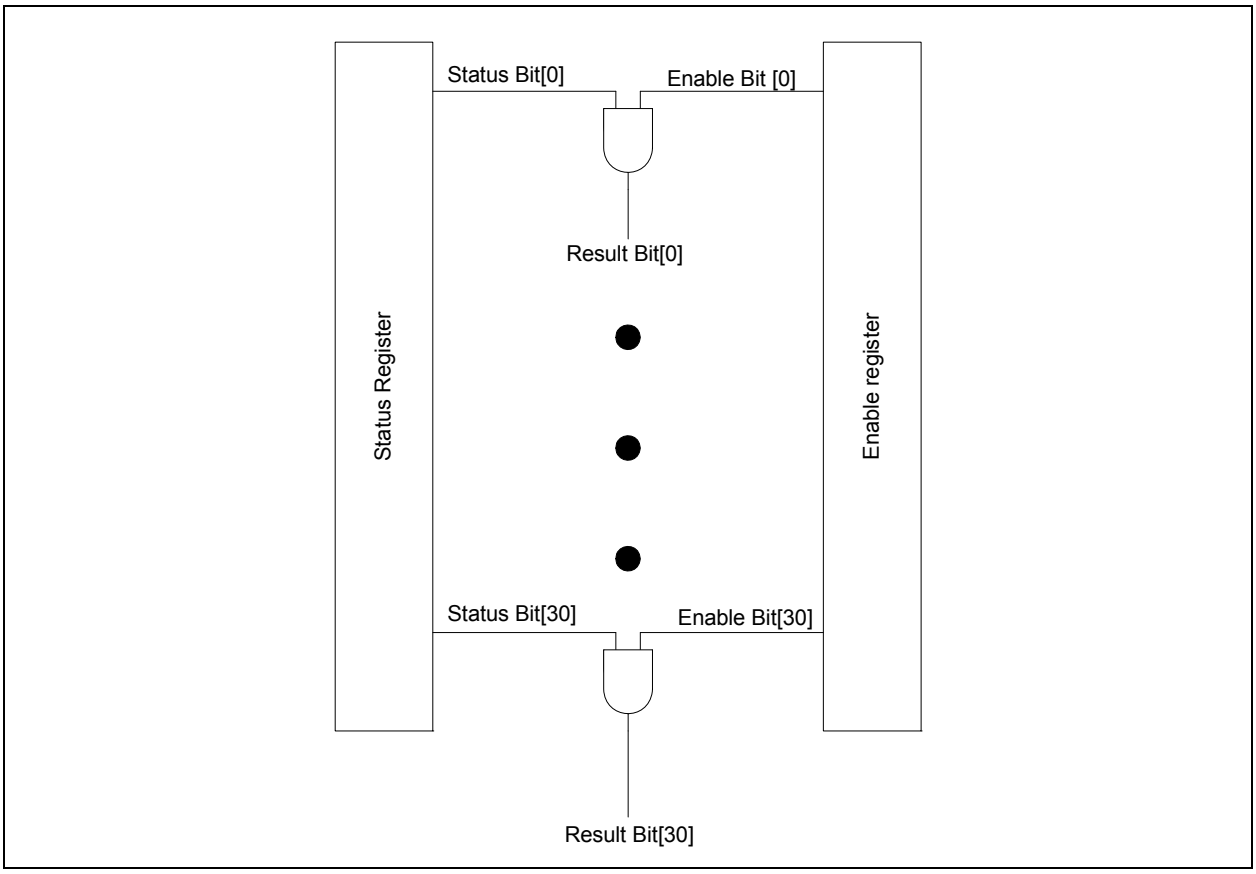
An individual interrupt *i* is enabled if bit *i* in the Interrupt Source register is 1 (asserted), bit *i* in the Interrupt Enable register is 1 (enabled), and the 2 bits for interrupt *i* in the Interrupt Priority Register represent a number that is greater than or equal to the IA Priority Register as well as greater or equal to the priority level of any other interrupt that is currently enabled.

The MEC1632 Interrupt Aggregator selects the interrupt with the highest priority among all active interrupts. Interrupts at priorities below the current are blocked. Within a group the interrupt with the higher bit number has priority over interrupts with lower bit number (assuming that these bits are at the same priority level).

17.3.3.2 Interrupt 'Result'

As shown in Figure 17-1 the Source and Enable bits are latched, but the Result bits are not. The latter change combinationally with inputs Source and Enable inputs.

FIGURE 17-2: EXAMPLE OF THE RESULTS REGISTER



17.3.3.3 Group Interrupt Request

There are 16 Group Interrupt Request signals, one for each bank in the Interrupt Aggregator. An interrupt is propagated within the interrupt chain (that results in the IRQ3) if the priority assigned to the source bit in the Group Priority Register is greater than or equal to the contents of the IA Priority Register (Current Priority bits). For example, assigning a priority of 3 to a source bit means maximum priority for that source, which will always be enabled if the corresponding enable

© 2012-2018 Microchip Technology Inc.

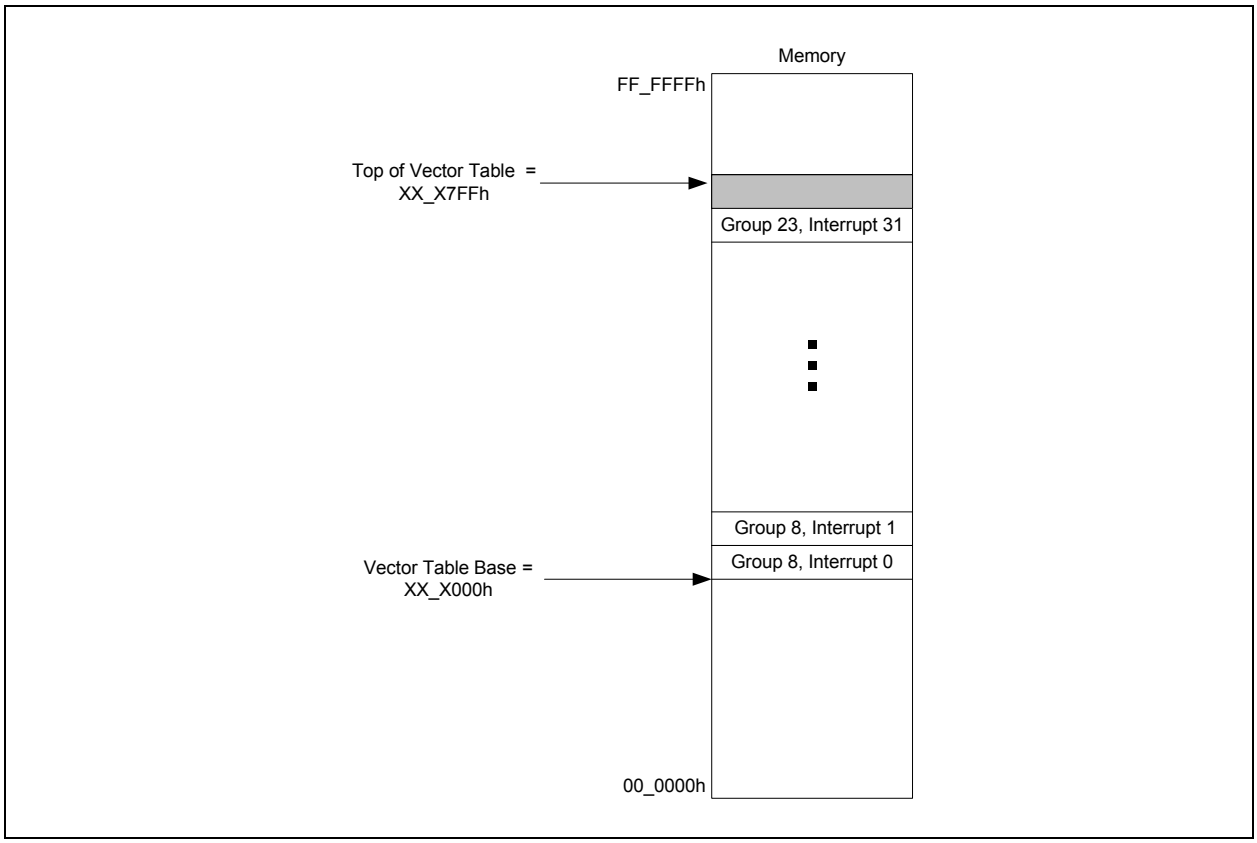
bit is set and the IA Priority Register is 3 or less. Setting the IA Priority Register to 4 or greater disables all interrupts in the Interrupt Aggregator. Software must maintain the IA Priority Register, and stack the value in memory if nested interrupts are required.

17.3.3.4 Interrupt Vector Generation

The Interrupt Aggregator continually selects from among all of its active inputs to generate one IRQ that is connected to the ARC interrupt controller on ARC input IRQ3. At the same time, the Aggregator generates an index into an Interrupt Vector table which addresses a pointer to the handler for the interrupt that is to be serviced.

The Interrupt Aggregator Interrupt Vector Table is a table of 4-byte addresses that is 2KB in length (16 Groups times 31 interrupts per group times 4 bytes per address). The FIGURE 17-3: Interrupt Aggregator Vector Table on page 292 illustrates the Interrupt Vector Table:

FIGURE 17-3: INTERRUPT AGGREGATOR VECTOR TABLE



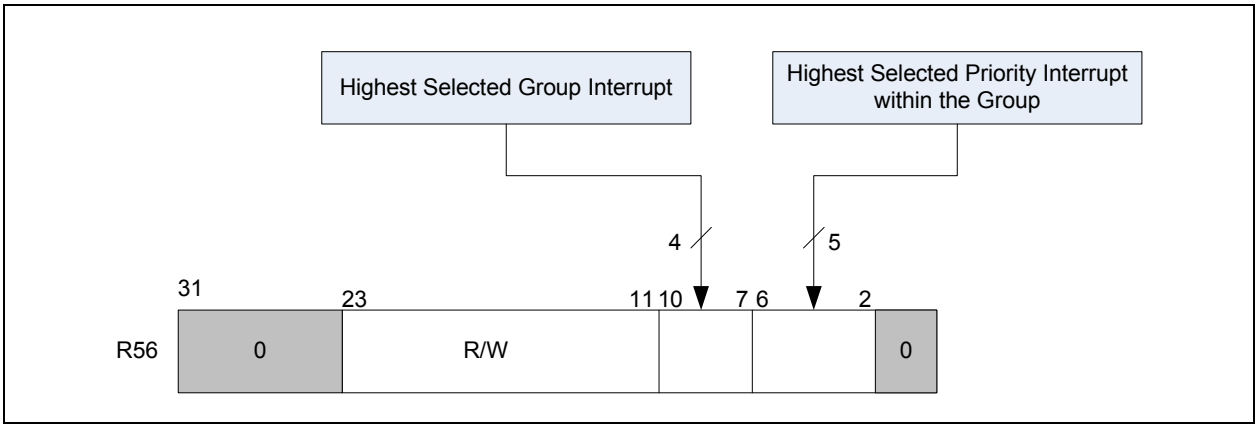
The table must begin on a 2KB address boundary. Since there are only 31 possible interrupt sources per Group Interrupt Request, the 32nd vector entry is always null. The table will typically reside in the program Flash memory.

A new extension register is added to the ARC register set to support interrupt vectoring. Register R56 is a pointer into the Interrupt Vector table and may be used in any ARC instruction that can reference one of the general purpose registers. Bits 23 through 11 of R56 are readable and writable; setting this range establishes the base address of the table. Bits 10 through 0 are read-only, which is the reason the vector table must start on a 2KB address boundary. Since ARC addresses are 16 bits, bits 31 through 24 of R56 are reserved and always read as 0.

The FIGURE 17-4: Interrupt Vector Generation on page 293 illustrates the mechanism by which bits 10 through 2 of R56 are set:

© 2012-2018 Microchip Technology Inc.

FIGURE 17-4: INTERRUPT VECTOR GENERATION



A Highest Group Interrupt block selects the highest numbered GIRQ currently active and enabled. It should be noted that even though the group numbering ranges from GIRQ8-GIRQ22, there are four bits allocated to represent these numbers. For example, if GIRQ8 is selected then value 0x0 will be assigned to bits [10:7] of the R56. Likewise, if GIRQ22 is selected, value 0xE will be assigned to bits [10:7] of R56. Table 17-4, "GIRQi Mapping to Bits [10:7] of R56" shows the mapping of the GIRQi to the bits [10:7] bits of R56.

TABLE 17-4: GIRQI MAPPING TO BITS [10:7] OF R56

$GIRQ_i$	Mapped Bit Values in the Group Select Field
GIRQ8	0000
GIRQ9	0001
GIRQ10	0010
GIRQ11	0011
GIRQ12	0100
GIRQ13	0101
GIRQ14	0110
GIRQ15	0111
GIRQ16	1000
GIRQ17	1001
GIRQ18	1010
GIRQ19	1011
GIRQ20	1100
GIRQ21	1101
GIRQ22	1110

The “Highest Priority Interrupt within the Group” block selects the highest numbered interrupt within the group that is active and enabled. The five bit result becomes bits 6 through 2 of R56. The low two bits of R56 are always 0, since the register always points to a word-aligned address.

A two instruction sequence is sufficient to vector to one of the Interrupt Aggregator interrupts. The following two-instruction sequence fits into the IRQ3 8-byte vector slot used by the ARC:

```
LDRx,[R56]              ;fetch the address for the active interrupt
J[Rx]                   ;Jump to handler
```

The register Rx should be reserved for exclusive use by the interrupt handler using the compiler option “-Hreg_reserve=x”, so that the register is never live when an interrupt is serviced.

An alternate ARC handler would be

```
JCommon_Int_Handler        ;Jump to handler
```

where `Common_Int_Handler` is a routine that handles interrupt state, performs any interrupt stack maintenance, and sets up whatever is required for the actual interrupt handler. The jump instruction is 8 bytes long and can reach any address in the ARC address space. When the common code is finished, the same two instruction sequence listed above can be used to jump to the correct interrupt handler.

The minimal ARC vector would consist of one 8-byte jump instruction. The time penalty for adding the load from R56 is two cycles, one for the standard ARC load delay and an additional cycle the ARC requires when fetching data from instruction space.

17.3.4 NON MASKABLE INTERRUPTS

The ARC does not have a non maskable interrupt input. It is straightforward to assign one of the Hibernation or Week timers, which can generate ARC interrupts on ARC IRQ5 through IRQ7, to ARC priority level 2, and all other interrupts to priority level 1. Since no other interrupt handler should ever disable interrupts on Priority 2, the timer interrupt will always be enabled.

It may not be possible to use a non maskable interrupt when the flash memory is being programmed by the Host. The Flash interrupt is asserted when a Host has completed Flash programming and the flash array is again available to the ARC. While the flash is being programmed, the ARC has no program space (which is normally in the flash), and so must sleep. Since it has no vector table, it cannot respond to interrupts while the flash is busy. All interrupts except Flash_EC_Int must be disabled; the Flash interrupt can be enabled since it is only asserted when the flash is again available to the ARC. One simple way to ensure this is to assign the Flash_EC_Int interrupt Priority Level 2, assign all other ARC IRQs to Priority Level 1, and disable Priority Level 1 before putting the ARC to sleep. An alternative would be to remap the ARC Interrupt Vector Base Register to the SRAM and define an interrupt vector table, along with its handlers, in the SRAM.

17.3.5 WAKE CAPABLE INTERRUPTS

The EC Interrupt Aggregator routes logic from WAKE Event Sources to the WAKE input of the Power, Clocks, and Resets Power Management Interface to wake the system. This logic requires no clocks.

The interrupt sources AND’ed with the corresponding Enable bit will be OR’ed to produce a wake event.

The wake up sources are identified with a “Y” in the “WAKE” column of the Bit definitions table for each IRQ’s Source Register.

Note:

- GPIO wake events require the Interrupt Detection field of the GPIO Pin Control Register to be set to Rising Edge Triggered, Falling Edge Triggered, or Either Edge Triggered. If the Interrupt Detection field is set to any other value, a GPIO input will not trigger a wake interrupt.
- Wake capable alternate functions implemented on GPIO pins require the Interrupt Detection field of the corresponding GPIO Pin Control Register to be set to Rising Edge Triggered, Falling Edge Triggered, or Either Edge Triggered. If the Interrupt Detection field is set to any other value, the alternate function input is not guaranteed to trigger a wake interrupt.

APPLICATION NOTE: Neither LPC accesses nor JTAG debug accesses are wake capable. In order to enable LPC transactions to MEC1632 Logical Devices while the MEC1632 is in a Sleep mode in which the main oscillator is shut off, just before entering sleep EC firmware must enable an interrupt on the falling edge of the GPIO associated with the LFRAME# input. When responding to this LFRAME/GPIO interrupt EC firmware should disable the LFRAME/GPIO interrupt until firmware determines that it is again appropriate to enter a Deep Sleep mode. Similarly, EC firmware must enable an interrupt on the falling edge of the GPIO of the GPIO associated with JTAG_CLK if JTAG debug accesses are required while the MEC1632 is in a sleep mode in which the main clock is turned off.

© 2012-2018 Microchip Technology Inc.

17.3.6 NON-WAKE CAPABLE INTERRUPTS

These interrupts require a running clock in their source block to be recognized and presented to the interrupt aggregator. Please consult the WAKE column of the Bit definitions table for each IRQ's Source Register.

17.3.7 INTERRUPTS DIRECTLY CONNECTED TO ARC PROCESSOR

Interrupts from the two Hibernation Timers, the Week Timer and the Embedded Flash interface are routed to the EC Interrupt lines IRQ7 through IRQ4, respectively.

PROGRAMMER'S NOTE: In non-legacy mode the Hibernation Timers, the Week Timer and the Embedded Flash interface are directly connected to the interrupt line and should be disabled in the Interrupt Enable Register. Otherwise these interrupts will be duplicated on the IRQ3 as well as their respective interrupt lines.

17.3.8 DISABLING INTERRUPTS

Because of pipeline latency, delay through the Load/Store queue and traffic on the AHB bus, writes to SPB registers can potentially take many processor cycles to complete. Because of this latency, the IA Group Select Register register and the IRQ Enable Register should not be used for disabling interrupts for software operations like critical sections. Several interrupts could potentially fire between the STORE instruction to the interrupt registers and the instruction after the STORE.

The ARC FLAG instruction is used to modify the E1 and E2 interrupt enable bits in the STATUS32 register. If the FLAG instruction is used, software can insure that no unexpected interrupts will be processed in the middle of a critical section. The following example illustrates how the FLAG instruction might be used to implement a critical section:

```
FLAG; 0                 ; disable all interrupts
SYNC                    ; Complete I/O ops
;
; <critical section code here>
;
FLAG; 6;                ; enable all interrupts
```

17.4 Power, Clocks and Reset

17.4.1 POWER DOMAIN

This block is powered by VTR for wake up capability.

See Section 7.5, "Power Configuration," on page 121 for details on power domains.

17.4.2 CLOCKS

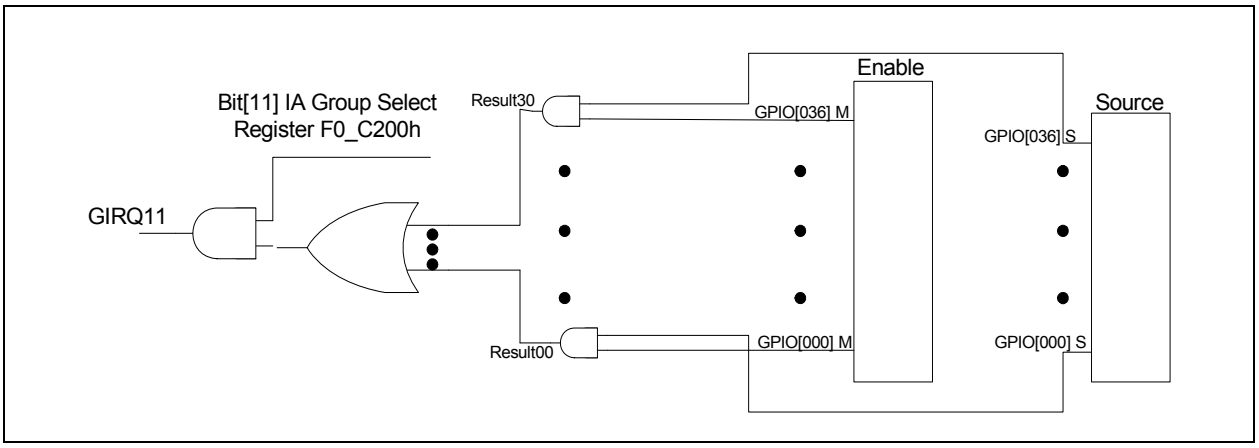
Use MCLK

17.4.3 RESET

This block is reset by nSYS_RST. Following a reset, Interrupt Source, Enable and Result registers default to '0' and all interrupts are enabled.

17.4.4 INTERRUPT ROUTING

FIGURE 17-5: GPIO INTERRUPT STRUCTURE EXAMPLE



17.5 Registers Overview

17.5.1 ADDRESSING

The EC Interrupt Aggregator has its own Logical Device Number, and Base Address as indicated in Table 17-5.

TABLE 17-5: EC Interrupt Aggregator BASE ADDRESS TABLE

Block	LDN	AHB Base Address
EC Interrupt Aggregator	30h	F0_C000h

17.5.2 REGISTERS SUMMARY

Table 17-6 is a register summary for the EC Interrupt Aggregator block. Each EC address is indicated as an SPB Offset from its AHB base address.

TABLE 17-6: EC Interrupts REGISTER SUMMARY

	EC Interface			Notes	
Register Name	**SPB Offset**	**Byte Lane**	**EC Type**	**Bit Definitions**	**Logical Devices**
GIRQ8 Source Register	00h	3-0	R/WC	Table 17-9	GPIO140-GPIO176
GIRQ8 Enable Register	04h	3-0	R/W		
GIRQ8 Result Register	08h	3-0	R		
GRIQ8a Priority Register	0Ch	3-0	R/W	Table 17-7	
GRIQ8b Priority Register	10h	3-0	R/W	Table 17-8	
GIRQ9 Source Register	14h	3-0	R/WC	Table 17-13	GPIO100-GPIO136
GIRQ9 Enable Register	18h	3-0	R/W		
GIRQ9 Result Register	1Ch	3-0	R		
GRIQ9a Priority Register	20h	3-0	R/W	Table 17-7	
GRIQ9b Priority Register	24h	3-0	R/W	Table 17-8	
GIRQ10 Source Register	28h	3-0	R/WC	Table 17-17	GPIO040-GPIO076
GIRQ10 Enable Register	2Ch	3-0	R/W		

© 2012-2018 Microchip Technology Inc.

TABLE 17-6: EC Interrupts REGISTER SUMMARY (CONTINUED)

	EC Interface			Notes	
Register Name	**SPB Offset**	**Byte Lane**	**EC Type**	**Bit Definitions**	**Logical Devices**
GIRQ10 Result Register	30h	3-0	R		
GRIQ10a Priority Register	34h	3-0	R/W	Table 17-7	
GRIQ10b Priority Register	38h	3-0	R/W	Table 17-8	
GIRQ11 Source Register	3Ch	3-0	R/WC	Table 17-22	GPIO000-GPIO036
GIRQ11 Enable Register	40h	3-0	R/W		
GIRQ11 Result Register	44h	3-0	R		
GRIQ11a Priority Register	48h	3-0	R/W	Table 17-7	
GRIQ11b Priority Register	4Ch	3-0	R/W	Table 17-8	
GIRQ12 Source Register	50h	3-0	R/WC	Table 17-26	SMBus
GIRQ12 Enable Register	54h	3-0	R/W		
GIRQ12 Result Register	58h	3-0	R		
GRIQ12a Priority Register	5Ch	3-0	R/W	Table 17-7	
GRIQ12b Priority Register	60h	3-0	R/W	Table 17-8	
GIRQ13 Source Register	64h	3-0	R/WC	Table 17-30	ACPI PM1
GIRQ13 Enable Register	68h	3-0	R/W		
GIRQ13 Result Register	6Ch	3-0	R		
GRIQ13a Priority Register	70h	3-0	R/W	Table 17-7	
GRIQ13b Priority Register	74h	3-0	R/W	Table 17-8	
GIRQ14 Source Register	78h	3-0	R/WC	Table 17-34	LPC Interface, EC GP-SPI, Embedded Flash Interface
GIRQ14 Enable Register	7Ch	3-0	R/W		
GIRQ14 Result Register	80h	3-0	R		
GRIQ14a Priority Register	84h	3-0	R/W	Table 17-7	
GRIQ14b Priority Register	88h	3-0	R/W	Table 17-8	
GIRQ15 Source Register	8Ch	3-0	R/WC	Table 17-38	Mailbox Interface, UART
GIRQ15 Enable Register	90h	3-0	R/W		
GIRQ15 Result Register	94h	3-0	R		
GRIQ15a Priority Register	98h	3-0	R/W	Table 17-7	
GRIQ15b Priority Register	9Ch	3-0	R/W	Table 17-8	
GIRQ16 Source Register	A0h	3-0	R/WC	Table 17-42	RC ID, ADC, PECI
GIRQ16 Enable Register	A4h	3-0	R/W		
GIRQ16 Result Register	A8h	3-0	R		
GRIQ16a Priority Register	ACh	3-0	R/W	Table 17-7	
GRIQ16b Priority Register	B0h	3-0	R/W	Table 17-8	
GIRQ17 Source Register	B4h	3-0	R/WC	Table 17-46	TACH
GIRQ17 Enable Register	B8h	3-0	R/W		
GIRQ17 Result Register	BCh	3-0	R		
GRIQ17a Priority Register	C0h	3-0	R/W	Table 17-7	

TABLE 17-6: EC Interrupts REGISTER SUMMARY (CONTINUED)

	EC Interface			Notes	
Register Name	**SPB Offset**	**Byte Lane**	**EC Type**	**Bit Definitions**	**Logical Devices**
GRIQ17b Priority Register	C4h	3-0	R/W	Table 17-8	
GIRQ18 Source Register	C8h	3-0	R/WC	Table 17-50	BC Bus Master
GIRQ18 Enable Register	CCh	3-0	R/W		
GIRQ18 Result Register	D0h	3-0	R		
GRIQ18a Priority Register	D4h	3-0	R/W	Table 17-7	
GRIQ18b Priority Register	D8h	3-0	R/W	Table 17-8	
GIRQ19 Source Register	DCh	3-0	R/WC	Table 17-53	Keyboard Controller (8042), PS/2
GIRQ19 Enable Register	E0h	3-0	R/W		
GIRQ19 Result Register	E4h	3-0	R		
GRIQ19a Priority Register	E8h	3-0	R/W	Table 17-7	
GRIQ19b Priority Register	ECh	3-0	R/W	Table 17-8	
GIRQ20 Source Register	F0h	3-0	R/WC	Table 17-57	ACPI EC MSG
GIRQ20 Enable Register	F4h	3-0	R/W		
GIRQ20 Result Register	F8h	3-0	R		
GRIQ20a Priority Register	FCh	3-0	R/W	Table 17-7	
GRIQ20b Priority Register	100h	3-0	R/W	Table 17-8	
GIRQ21 Source Register	104h	3-0	R/WC	Table 17-61	Reserved
GIRQ21 Enable Register	108h	3-0	R/W		
GIRQ21 Result Register	10Ch	3-0	R		
GRIQ21a Priority Register	110h	3-0	R/W	Table 17-7	
GRIQ21b Priority Register	114h	3-0	R/W	Table 17-8	
GIRQ22 Source Register	118h	3-0	R/WC	Table 17-65	GPIO[217:200]
GIRQ22 Enable Register	11Ch	3-0	R/W		
GIRQ22 Result Register	120h	3-0	R		
GRIQ22a Priority Register	124h	3-0	R/W	Table 17-7	
GRIQ22b Priority Register	128h	3-0	R/W	Table 17-8	
GIRQ23 Source Register	12Ch	3-0	R/WC	Table 17-69	Week Alarm Timer, 16-bit Timer Hibernation Timer Input Capture and Compare Timer
GIRQ23 Enable Register	130h	3-0	R/W		
GIRQ23 Result Register	134h	3-0	R		
GRIQ23a Priority Register	138h	3-0	R/W	Table 17-7	
GRIQ23b Priority Register	13Ch	3-0	R/W	Table 17-8	
IA Group Select Register	200h	3-0	R/W	Table 17-72	
IRQ Enable Register	204h	3-0	R/W	Table 17-73	

© 2012-2018 Microchip Technology Inc.

17.6 Register Descriptions

Input interrupts are grouped into groups of up to 31 interrupts each. Associated with each group is a set of Source, Enable, Priority, and Result registers. Registers are 32-bit. There are two Priority registers for each group since an interrupt's priority level is encoded by 2 bits.

The aforementioned four register 'types' are first described in generic terms. Subsequent sections describe register bits in terms of specific interrupts they are associated with.

17.6.1 GIRQX SOURCE REGISTERS

There are 16 Group Source Enable registers, one per Interrupt Group.

Int_Source[30:0]:

A bit in this field is set when the corresponding interrupt input is active. The Interrupt Aggregator recognizes level-triggered, active-high inputs. Other input types are to be captured and relayed to the Aggregator. For example, the GPIO interface can register external edge-triggered interrupts and forward them to the Aggregator as active-high, level interrupts.

17.6.2 GIRQX ENABLE REGISTERS

There are 16 Group Interrupt Enable registers, one per Interrupt Group.

Int_Enable[30:0]:

Each bit in this field enables an interrupt from the like-numbered bit in the associated Interrupt Source register. Interrupt *i* is disabled if Int_Enable[*i*] is 0 and enabled if Int_Enable[*i*] is 1 and the Int_Priority[*i*] is greater than or equal to the current priority level. See Table 17-6, "EC Interrupts Register Summary" for EC Offset addresses for the 16 GIRQx Enable registers.

17.6.3 GIRQX A PRIORITY REGISTERS

There are 16 Group Interrupt A Priority registers, one per Interrupt Group. Group Interrupt A Priority registers, combined with the Group Interrupt B Priority registers, determine the 2-bit priority level for all interrupts in an Interrupt Group. The format of all A Priority registers is the same, described in Table 17-7, "GIRQx A Priority Register": *It should be noted that at times not all corresponding source register bits are used, hence the respective Int_priority bits should not be populated.*

TABLE 17-7: GIRQX A PRIORITY REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC OFFSET	xxh			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			**VTR POR DEFAULT**	
BYTE3 BIT	**D31**	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Int_Priority15		Int_Priority14		Int_Priority13		Int_Priority12	
BYTE2 BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Int_Priority11		Int_Priority10		Int_Priority9		Int_Priority8	
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Int_Priority7		Int_Priority6		Int_Priority5		Int_Priority4	
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Int_Priority3		Int_Priority2		Int_Priority1		Int_Priority0	

INT_PRIORITY[15:0]

Each of the 16 2-bit fields in this register sets the priority level for interrupts assigned to Interrupt Source register bit 15 through bit 0, in the same Interrupt Group. See Table 17-6, "EC Interrupts Register Summary" for EC Offset addresses for the sixteen GIRQx A Priority registers.

© 2012-2018 Microchip Technology Inc.

17.6.4 GIRQX B PRIORITY REGISTERS

There are 16 Group Interrupt B Priority registers, one per Interrupt Group. Group Interrupt B Priority registers, combined with the Group Interrupt A Priority registers, determine the 2-bit priority level for all interrupts in an Interrupt Group. The format of all B Priority registers is the same, described in Table 17-8, "GIRQx B Priority Register":*It should be noted that at times not all corresponding source register bits are used, hence the respective Int_priority bits should not be populated.*

TABLE 17-8: GIRQX B PRIORITY REGISTER

HOST OFFSET	N/A			N/A		**HOST SIZE**		
EC OFFSET	xxh			32-bit		**EC SIZE**		
POWER	VTR			0000_0000h		**VTR POR DEFAULT**		
BYTE3 BIT	**D31**	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Reserved		Int_Priority30		Int_Priority29		Int_Priority28	
BYTE2 BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Int_Priority27		Int_Priority26		Int_Priority25		Int_Priority24	
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Int_Priority23		Int_Priority22		Int_Priority21		Int_Priority20	
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Int_Priority19		Int_Priority18		Int_Priority17		Int_Priority16	

INT_PRIORITY[30:16]

Each of the 16 2-bit fields in this register sets the priority level for interrupts assigned to Interrupt Source register bit 30 through bit 16, in the same Interrupt Group. See Table 17-6, "EC Interrupts Register Summary" for EC Offset addresses for the sixteen GIRQx B Priority registers.

17.6.5 GIRQX RESULT REGISTERS

There are 16 Group Interrupt Result registers, one per Interrupt Group.

INT_RESULT[30:0]

Each bit in this field is 1 if an interrupt from the like-numbered bit in the associated Interrupt Source register is active. Interrupt *i* in each Interrupt Group is active if and only if Int_Source[*i*] is 1, Int_Enable[*i*] is 1 and Int_Priority[*i*] is greater than or equal to IA Priority register as well as any enabled interrupt in all Interrupt Groups. The GIRQx Result Register is not latched but is a function of Int_Source, Int_Enable and Int_Priority. See Figure 17-2 for an explanation of how Result is generated.

17.6.6 GIRQ8 SOURCE REGISTER

TABLE 17-9: GIRQ8 SOURCE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	00h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	RWC except for reserved bits See Table 17-10 & Note 17-1 on page 302						
BIT NAME	Reserved	GPIO[140:176]						

TABLE 17-10: BIT DEFINITIONS GIRQ8 SOURCE REGISTER

Bit	EC Type	Signal Name	Wake	Description
[7:0]	R/WC	GPIO[147:140]	Y	GPIO Interrupt Signal. This bit is set when a interrupt source is triggered. This bit is sticky; once set, it remains set until cleared by a write with the value '1' to this bit. Write to this bit with a value of '0' have no effect. **Note:** GIRQ8 is allocated for GPIOs 140 - 176. See pinout for GPIOs implemented.
[15:8]	R/WC	GPIO[157:150]	Y	
[23:16]	R/WC	GPIO[167:160]	Y	
[30:24]	R/WC	GPIO[176:170]	Y	
31	R	Reserved	N	This bit is always reserved

Note 17-1 Reserved Bits shown in the Source register are also reserved read only in the Corresponding Enable & Result register.

© 2012-2018 Microchip Technology Inc.

17.6.7 GIRQ8 ENABLE REGISTER

TABLE 17-11: GIRQ8 ENABLE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	04h					32-bit	**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	RW except for reserved bits See Table 17-10 & Note 17-1 on page 302						
BIT NAME	Reserved	GPIO[140:176]						

17.6.8 GIRQ8 RESULT REGISTER

TABLE 17-12: GIRQ8 RESULT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	08h					32-bit	**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	GPIO[140:176]						

NORM

This bit is a read write bit used by the NORM instruction.

17.6.9 GIRQ9 SOURCE REGISTER

TABLE 17-13: GIRQ9 SOURCE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	14h					32-bit	**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	RWC except for reserved bits See Table 17-14 & Note 17-1 on page 302						
BIT NAME	Reserved	GPIO[100:136]						

TABLE 17-14: BIT DEFINITIONS GIRQ9 SOURCE REGISTER

Bit	EC Type	Signal Name	Wake	Description
[7:0]	R/WC	GPIO[107:100]	Y	GPIO Interrupt Signal. This bit is set when a interrupt source is triggered. This bit is sticky; once set, it remains set until cleared by a write with the value '1' to this bit. Write to this bit with a value of '0' have no effect. **Note:** GIRQ9 is allocated for GPIOs 100 - 136. See pinout for GPIOs implemented.
[15:8]	R/WC	GPIO[117:110]	Y	
[23:16]	R/WC	GPIO[127:120]	Y	
[30:24]	R/WC	GPIO[136:130]	Y	
31	R	Reserved	N	This bit is always reserved

17.6.10 GIRQ9 ENABLE REGISTER

TABLE 17-15: GIRQ9 ENABLE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	18h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	RW except for reserved bits See Table 17-14 & Note 17-1 on page 302						
BIT NAME	Reserved	GPIO[100:136]						

17.6.11 GIRQ9 RESULT REGISTER

TABLE 17-16: GIRQ9 RESULT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	1Ch					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	GPIO[100:136]						

NORM

This bit is a read write bit used by the NORM instruction.

© 2012-2018 Microchip Technology Inc.

17.6.12 GIRQ10 SOURCE REGISTER

TABLE 17-17: GIRQ10 SOURCE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	28h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	RWC except for reserved bits See Table 17-18 & Note 17-1 on page 302						
BIT NAME	Reserved	GPIO[040:076]						

TABLE 17-18: BIT DEFINITIONS GIRQ10 SOURCE REGISTER

Bit	EC Type	Signal Name	Wake	Description
[7:0]	R/WC	GPIO[047:040]	Y	GPIO Interrupt Signal. This bit is set when a interrupt source is triggered. This bit is sticky; once set, it remains set until cleared by a write with the value '1' to this bit. Write to this bit with a value of '0' have no effect. **Note:** GIRQ10 is allocated for GPIOs 040 - 076. See pinout for GPIOs implemented.
[15:8]	R/WC	GPIO[057:050]	Y	
[23:16]	R/WC	GPIO[067:060]	Y	
[30:24]	R/WC	GPIO[076:070]	Y	
31	R	Reserved	N	This bit is always reserved

17.6.13 GIRQ10 ENABLE REGISTER

TABLE 17-19: G IRQ10 ENABLE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	2Ch					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	RW except for reserved bits See Table 17-18 & Note 17-1 on page 302						
BIT NAME	Reserved	GPIO[040:076]						

17.6.14 GIRQ10 RESULT REGISTER

TABLE 17-20: GIRQ10 RESULT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	30h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	GPIO[040:076]						

NORM

This bit is a read write bit used by the NORM instruction.

17.6.15 GIRQ11 SOURCE REGISTER

TABLE 17-21: GIRQ11 SOURCE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	3Ch					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	RWC except for reserved bits See Table 17-22 & Note 17-1 on page 302						
BIT NAME	Reserved	GPIO[000:036]						

TABLE 17-22: BIT DEFINITIONS GIRQ11 SOURCE REGISTER

Bit	EC Type	Signal Name	Wake	Description
[7:0]	R/WC	GPIO[007:000]	Y	GPIO Interrupt Signal. This bit is set when a interrupt source is triggered. This bit is sticky; once set, it remains set until cleared by a write with the value '1' to this bit. Write to this bit with a value of '0' have no effect. **Note:** GIRQ11 is allocated for GPIOs 000 - 036. See pinout for GPIOs implemented.
[15:8]	R/WC	GPIO[017:010]	Y	
[23:16]	R/WC	GPIO[027:020]	Y	
[30:24]	R/WC	GPIO[036:030]	Y	
31	R	Reserved	N	This bit is always reserved

© 2012-2018 Microchip Technology Inc.

17.6.16 GIRQ11 ENABLE REGISTER

TABLE 17-23: GIRQ11 ENABLE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	40h					32-bit	**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	RW except for reserved bits See Table 17-22 & Note 17-1 on page 302						
BIT NAME	Reserved	GPIO[000:036]						

17.6.17 GIRQ11 RESULT REGISTER

TABLE 17-24: GIRQ11 RESULT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	44h					32-bit	**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	GPIO[000:036]						

NORM

This bit is a read write bit used by the NORM instruction.

17.6.18 GIRQ12 SOURCE REGISTER

TABLE 17-25: GIRQ12 SOURCE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	50h					32-bit	**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	RWC except for reserved bits See Table 17-26 & Note 17-1 on page 302						
BIT NAME	Reserved	Refer To Table 17-26, “Bit Definitions GIRQ12 Source Register,” on page 308						

TABLE 17-26: BIT DEFINITIONS GIRQ12 SOURCE REGISTER

Bit	Signal Name	Wake	Description
0	SMB0	N	I2C/SMBus controller 0 interrupt. This interrupt is signaled when the I2C/SMBus controller 0 asserts its interrupt request
1	SMB1	N	I2C/SMBus controller 1 interrupt. This interrupt is signaled when the I2C/SMBus controller 1 asserts its interrupt request
2	SMB2	N	I2C/SMBus controller 2 interrupt. This interrupt is signaled when the I2C/SMBus controller 2 asserts its interrupt request
3	SMB00 WK	Y	I2C/SMBus Port 00 Wake interrupt. This interrupt is asserted when an edge is detected on the SDAT pin of port SMB00.
4	SMB01 WK	Y	I2C/SMBus Port 01 Wake interrupt. This interrupt is asserted when an edge is detected on the SDAT pin of port SMB01.
5	SMB02 WK	Y	I2C/SMBus Port 02 Wake interrupt. This interrupt is asserted when an edge is detected on the SDAT pin of port SMB02.
6	SMB03 WK	Y	I2C/SMBus Port 03 Wake interrupt. This interrupt is asserted when an edge is detected on the SDAT pin of port SMB03.
7	SMB04 WK	Y	I2C/SMBus Port 04 Wake interrupt. This interrupt is asserted when an edge is detected on the SDAT pin of port SMB04.
8	SMB05 WK	Y	I2C/SMBus Port 05 Wake interrupt. This interrupt is asserted when an edge is detected on the SDAT pin of port SMB05.
9	SMB06 WK	Y	I2C/SMBus Port 06 Wake interrupt. This interrupt is asserted when an edge is detected on the SDAT pin of port SMB06.
10	SMB07 WK	Y	I2C/SMBus Port 07 Wake interrupt. This interrupt is asserted when an edge is detected on the SDAT pin of port SMB07.
11	SMB08 WK	Y	I2C/SMBus Port 10 Wake interrupt. This interrupt is asserted when an edge is detected on the SDAT pin of port SMB10.
12	SMB09 WK	Y	I2C/SMBus Port 09 Wake interrupt. This interrupt is asserted when an edge is detected on the SDAT pin of port SMB09.
13	SMB10 WK	Y	I2C/SMBus Port 10 Wake interrupt. This interrupt is asserted when an edge is detected on the SDAT pin of port SMB10.
16:14	Reserved	N	Reserved
17	SB_TSI	Y	SB-TSI (Port 11) wake interrupt.
18	SMB3	N	I2C/SMBus controller 3 interrupt. This interrupt is signaled when the I2C/SMBus controller 3 asserts its interrupt request
19	RTC	Y	Real Time Clock Interrupt. This interrupt is signaled when the RTC asserts any of its enabled interrupt requests.
20	RTC_ALRM	Y	Real Time Clock Alarm Interrupt. This interrupt is signaled when the RTC asserts its Alarm interrupt.
30-21	Reserved	N	Reserved
31	Reserved	N	This bit is always reserved

Note 17-2 Source bits have corresponding Enable and Result bits.

© 2012-2018 Microchip Technology Inc.

17.6.19 GIRQ12 ENABLE REGISTER

TABLE 17-27: GIRQ12 ENABLE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	54h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/W except for reserved bits See Table 17-26 & Note 17-1 on page 302						
BIT NAME	Reserved	Refer To Table 17-26, “Bit Definitions GIRQ12 Source Register,” on page 308						

17.6.20 GIRQ12 RESULT REGISTER

TABLE 17-28: GIRQ12 RESULT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	58h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	Refer to Table 17-26, “Bit Definitions GIRQ12 Source Register,” on page 308						

NORM

This bit is a read write bit used by the NORM instruction.

17.6.21 GIRQ13 SOURCE REGISTER

TABLE 17-29: GIRQ13 SOURCE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	64h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	RWC except for reserved bits See Table 17-30 & Note 17-1 on page 302						
BIT NAME	Reserved	Refer To Table 17-30, “Bit Definitions GIRQ13 Source Register,” on page 310						

TABLE 17-30: BIT DEFINITIONS GIRQ13 SOURCE REGISTER

Bit	Signal Name	Wake	Description
0	PM1_CTL2	N	PM1_CTL2 written by Host
1	PM1_EN2	N	PM1_EN2 written by Host
2	PM1_STS2	N	PM1_STS2 written by Host
15:3	Reserved	N	Reserved
16	DMA_0	N	DMA Channel 0
17	DMA_1	N	DMA Channel 1
18	DMA_2	N	DMA Channel 2
19	DMA_3	N	DMA Channel 3
20	DMA_4	N	DMA Channel 4
21	DMA_5	N	DMA Channel 5
22	DMA_6	N	DMA Channel 6
23	DMA_7	N	DMA Channel 7
24	DMA_8	N	DMA Channel 8
25	DMA_9	N	DMA Channel 9
30-26	Reserved	N	Reserved
31	Reserved	N	This bit is always reserved

Note 17-3 Source bits have corresponding Enable and Result bits.

17.6.22 GIRQ13 ENABLE REGISTER

TABLE 17-31: GIRQ13 ENABLE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	68h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W except for reserved bits See Table 17-30 & Note 17-1 on page 302						
BIT NAME	Reserved	Refer To Table 17-30, “Bit Definitions GIRQ13 Source Register,” on page 310						

© 2012-2018 Microchip Technology Inc.

17.6.23 GIRQ13 RESULT REGISTER

TABLE 17-32: GIRQ13 RESULT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	6Ch					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	Refer to Table 17-30, “Bit Definitions GIRQ13 Source Register,” on page 310						

NORM

This bit is a read write bit used by the NORM instruction.

17.6.24 GIRQ14 SOURCE REGISTER

TABLE 17-33: GIRQ14 SOURCE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	78h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	RWC except for reserved bits See Table 17-34 & Note 17-1 on page 302						
BIT NAME	Reserved	Refer To Table 17-34, “Bit Definition GIRQ14 source Register,” on page 311						

TABLE 17-34: BIT DEFINITION GIRQ14 SOURCE REGISTER

Bit	Signal Name	Wake	Description
0	LPCPD#	Y	LPC Power Down pin state
1	LRESET#	Y	LRESET Interrupt. This interrupt is signaled when LRESET is asserted.
2	LPC_AHB_ERR	N	Either a LPC BAR conflict or an AHB bus error occurred as a result of an LPC access.
3	SPI_TXBE_GP	N	Tx buffer empty from the GP_SPI block on EC AHB
4	SPI_RXBF_GP	N	Rx buffer bfull from the GP_SPI block on EC AHB
5	FLASH_BUSY_ERR	N	Embedded Flash Busy Error
6	FLASH_CMD_ERR	N	Embedded Flash Command Error
7	FLASH_PROTECT_ERR	N	Embedded Flash Protect Error
8	FLASH_EC_INT	N	Host-to-EC Interrupt that transfers control of the Flash to the EC
9	VCC_PWRGD_INT	Y	VCC_PWRGD (from GPIO 57)
15:10	Reserved	-	Reserved
16	GP_SPI_TXBE	N	Tx buffer empty from the GP_SPI block on LPC AHB

TABLE 17-34: BIT DEFINITION GIRQ14 SOURCE REGISTER (CONTINUED)

Bit	Signal Name	Wake	Description
17	GP_SPI_RXBF	N	Rx buffer bfull from the GP_SPI block on LPC AHB
18	ASIF_INT	N	Interrupt from the ASIF block's EC logical interface
22-19	Reserved	-	Reserved
23	EEPROM_BUSY_ERR	N	Embedded EEPROM Busy Error
24	EEPROM_CMD_ERR	N	Embedded EEPROM Command Error
30-25	Reserved	-	Reserved
31	Reserved	-	This bit is always reserved

Note 17-4 Source bits have corresponding Enable and Result bits.

17.6.25 GIRQ14 ENABLE REGISTER

TABLE 17-35: GIRQ14 ENABLE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	7Ch					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/W except for reserved bits See Table 17-34 & Note 17-1 on page 302						
BIT NAME	Reserved	Refer To Table 17-34, "Bit Definition GIRQ14 source Register," on page 311						

17.6.26 GIRQ14 RESULT REGISTER

TABLE 17-36: GIRQ14 RESULT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	80h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	Refer to Table 17-34, "Bit Definition GIRQ14 source Register," on page 311						

NORM

This bit is a read write bit used by the NORM instruction.

© 2012-2018 Microchip Technology Inc.

17.6.27 GIRQ15 SOURCE REGISTER

TABLE 17-37: GIRQ15 SOURCE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	8Ch					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	RWC except for reserved bits See Table 17-38 & Note 17-1 on page 302						
BIT NAME	Reserved	Refer To Table 17-38, “Bit Definition GIRQ15 Source Register,” on page 313						

TABLE 17-38: BIT DEFINITION GIRQ15 SOURCE REGISTER

Bit	Signal Name	Wake	Description
0	UART_RX	N	UART Interrupt
1	MBX	N	Mailbox Register Interface EC Interrupt
2	EM_MBX	N	Embedded Memory Interface Host-to-EC Mailbox Interrupt
7-3	Reserved	N	Reserved
8	BDP0_INT	N	Port 80 BIOS Debug Ports
9	BDP1_INT	N	
10	LED0	N	Blinking/Breathing PWM
11	LED1	N	
12	LED2	N	
30-13	Reserved	N	Reserved
31	Reserved	N	This bit is always reserved

Note 17-5 Source bits have corresponding Enable and Result bits.

17.6.28 GIRQ15 ENABLE REGISTER

TABLE 17-39: GIRQ15 ENABLE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	90h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/W except for reserved bits See Table 17-38 & Note 17-1 on page 302						
BIT NAME	Reserved	Refer To Table 17-38, “Bit Definition GIRQ15 Source Register,” on page 313						

17.6.29 GIRQ15 RESULT REGISTER

TABLE 17-40: GIRQ15 RESULT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	94h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	Refer to Table 17-38, “Bit Definition GIRQ15 Source Register,” on page 313						

NORM

This bit is a read write bit used by the NORM instruction.

17.6.30 GIRQ16 SOURCE REGISTER

TABLE 17-41: GIRQ16 SOURCE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	A0h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	RWC except for reserved bits See Table 17-42 & Note 17-1 on page 302						
BIT NAME	Reserved	Refer To Table 17-42, “Bit Definition IRQ16 Source Register,” on page 314						

TABLE 17-42: BIT DEFINITION IRQ16 SOURCE REGISTER

Bit	Signal Name	Wake	Description
0	RCID	N	0-to-1 transition of RC_ID done flag
1	ADC_ONESTAT	N	ADC’s one-shot conversion completion interrupt
2	ADC_RTPSTAT	N	ADC’s repeated conversion interrupt
3	PECI_INT	N	PECI interrupt
4	CEC_INT	N	HDMI-CEC Interface Controller
7-5	Reserved	N	Reserved
8	Reserved	N	Reserved
9	Reserved	N	Reserved
30-10	Reserved	N	Reserved
31	Reserved	N	This bit is always reserved

Note 17-6 Source bits have corresponding Enable and Result bits.

© 2012-2018 Microchip Technology Inc.

17.6.31 GIRQ16 ENABLE REGISTER

TABLE 17-43: GIRQ16 ENABLE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	A4h					32-bit	**EC SIZE**	
POWER	VTR					0000_00000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/W except for reserved bits See Table 17-42 & Note 17-1 on page 302						
BIT NAME	Reserved	Refer To Table 17-42, “Bit Definition IRQ16 Source Register,” on page 314						

17.6.32 GIRQ16 RESULT REGISTER

TABLE 17-44: GIRQ16 RESULT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	A8h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	Refer to Table 17-42, “Bit Definition IRQ16 Source Register,” on page 314						

NORM

This bit is a read write bit used by the NORM instruction.

17.6.33 GIRQ17 SOURCE REGISTER

TABLE 17-45: GIRQ17 SOURCE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	B4h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	RWC except for reserved bits See Table 17-46 & Note 17-1 on page 302						
BIT NAME	Reserved	Refer To Table 17-46, “Bit Definition GIRQ17 Source Register,” on page 316						

TABLE 17-46: BIT DEFINITION GIRQ17 SOURCE REGISTER

Bit	Signal Name	Wake	Description
0	TACH0	N	Fan TACH 0 Interrupt.
1	TACH1	N	Fan TACH 1 Interrupt.
2	TACH2	N	Fan TACH 2 Interrupt.
3	TACH3	N	Fan TACH 3 Interrupt.
4	TACH4	N	Fan TACH 4 Interrupt.
5	TACH5	N	Fan TACH 5 Interrupt.
30-6	Reserved	N	Reserved
31	Reserved	N	This bit is always reserved

Note 17-7 Source bits have corresponding Enable and Result bits.

17.6.34 GIRQ17 ENABLE REGISTER

TABLE 17-47: GIRQ17 ENABLE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	B8h						32-bit	**EC SIZE**
POWER	VTR						0000_0000h	nSYS_RST **DEFAULT**
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/W except for reserved bits See Table 17-46 & Note 17-1 on page 302						
BIT NAME	Reserved	Refer To Table 17-46, “Bit Definition GIRQ17 Source Register,” on page 316						

17.6.35 GIRQ17 RESULT REGISTER

TABLE 17-48: GIRQ17 RESULT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	98h						32-bit	**EC SIZE**
POWER	VTR						0000_0000h	nSYS_RST **DEFAULT**
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	Refer to Table 17-46, “Bit Definition GIRQ17 Source Register,” on page 316						

NORM

This bit is a read write bit used by the NORM instruction.

© 2012-2018 Microchip Technology Inc.

17.6.36 GIRQ18 SOURCE REGISTER

TABLE 17-49: GIRQ18 SOURCE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	C8h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	RWC except for reserved bits See Table 17-50 & Note 17-1 on page 302						
BIT NAME	Reserved	Refer To Table 17-50, “Bit Definition IRQ18 Source Register,” on page 317						

TABLE 17-50: BIT DEFINITION IRQ18 SOURCE REGISTER

Bit	Signal Name	Wake	Description
0	BCM_BUSY_CLR[A]	N	BC-LINK Busy Clear Flag Interrupt
1	BCM_ERR[A]	N	BC_LINK Error Flag Interrupt
2	BCM_INT#[A]	Y	Interrupt from the BC_LINK Companion
3	BCM_BUSY_CLR[B]	N	BC-LINK Busy Clear Flag Interrupt
4	BCM_ERR[B]	N	BC_LINK Error Flag Interrupt
5	BCM_INT#[B]	Y	Interrupt from the BC_LINK Companion
8-6	Reserved	N	Reserved
9	BCM_BUSY_CLR[D]	N	BC-LINK Busy Clear Flag Interrupt
10	BCM_ERR[D]	N	BC_LINK Error Flag Interrupt
11	BCM_INT#[D]	Y	Interrupt from the BC_LINK Companion
15-12	Reserved	N	Reserved
16	KEYSCAN	Y	Interrupt from keyscan block
30-17	Reserved	N	Reserved
31	Reserved	N	This bit is always reserved

Note 17-8 Source bits have corresponding Enable and Result bits.

17.6.37 GIRQ18 ENABLE REGISTER

TABLE 1: GIRQ18 ENABLE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	CCh					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/W except for reserved bits See Table 17-50 & Note 17-1 on page 302						
BIT NAME	Reserved	Refer To Table 17-50, “Bit Definition IRQ18 Source Register,” on page 317						

© 2012-2018 Microchip Technology Inc.

17.6.38 GIRQ18 RESULT REGISTER

TABLE 17-51: GIRQ18 RESULT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	D0h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	Refer to Table 17-50, “Bit Definition IRQ18 Source Register,” on page 317						

NORM

This bit is a read write bit used by the NORM instruction.

17.6.39 GIRQ19 SOURCE REGISTER

TABLE 17-52: GIRQ19 SOURCE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	DCh					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	RWC except for reserved bits See Table 17-53 & Note 17-1 on page 302						
BIT NAME	Reserved	Refer To Table 17-53, “Bit Definition GIRQ19 Source Register,” on page 318						

TABLE 17-53: BIT DEFINITION GIRQ19 SOURCE REGISTER

Bit	Signal Name	Wake	Description
0	KBD_OBF	N	Keyboard Controller OBF Interrupt. This interrupt is signaled when the OBF bit in the KBD status register has been clear.
1	KBD_IBF	N	Keyboard Controller IBF Interrupt. This interrupt is signaled when the host writes to the KBD Command or Data port.
12-2	Reserved	N	Reserved
13	PS2_ACT_0	N	PS2_0 Activity Interrupt form PS/2 Block
14	PS2_ACT_1	N	PS2_1 Activity Interrupt form PS/2 Block
15	PS2_ACT_2	N	PS2_2 Activity Interrupt form PS/2 Block
16	Reserved	N	Reserved
17	PS2_WK_0A	Y	PS2_0A Start Detectform pin signal
18	PS2_WK_0B	Y	PS2_0B Start Detectform pin signal
19	PS2_WK_1A	Y	PS2_1A Start Detectform pin signal
20	PS2_WK_1B	Y	PS2_1B Start Detectform pin signal

© 2012-2018 Microchip Technology Inc.

TABLE 17-53: BIT DEFINITION GIRQ19 SOURCE REGISTER (CONTINUED)

Bit	Signal Name	Wake	Description
21	PS2_WK_2	Y	PS2_2 Start Detectform pin signal
30-22	Reserved	N	Reserved
31	Reserved	N	This bit is always reserved

Note 17-9 Source bits have corresponding Enable and Result bits.

17.6.40 GIRQ19 ENABLE REGISTER

TABLE 17-54: GIRQ19 ENABLE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	E0h					32-bit	**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/W except for reserved bits See Table 17-53 & Note 17-1 on page 302						
BIT NAME	Reserved	Refer To Table 17-53, “Bit Definition GIRQ19 Source Register,” on page 318						

17.6.41 GIRQ19 RESULT REGISTER

TABLE 17-55: GIRQ19 RESULT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	E4h					32-bit	**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	Refer to Table 17-53, “Bit Definition GIRQ19 Source Register,” on page 318						

NORM

This bit is a read write bit used by the NORM instruction.

17.6.42 GIRQ20 SOURCE REGISTER

TABLE 17-56: GIRQ20 SOURCE REGISTER

HOST ADDRESS	N/A							**HOST SIZE**
EC OFFSET	F0h						32-bit	**EC SIZE**
POWER	VTR						0000_0000h	nSYS_RST **DEFAULT**
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	RWC except for reserved bits See Table 17-57 & Note 17-1 on page 302						
BIT NAME	Reserved	Refer To Table 17-57, “Bit Definition IRQ20 Source Register,” on page 320						

TABLE 17-57: BIT DEFINITION IRQ20 SOURCE REGISTER

Bit	Signal Name	Wake	Description
0	EC2_OBF	N	Embedded Controller 2 OBF Interrupt. This interrupt is signaled when the OBF bit in the EC2 status register has been clear.
1	EC2_IBF	N	Embedded Controller 2 IBF Interrupt. This interrupt is signaled when the host writes to the EC2 Command or Data port.
2	EC1_OBF	N	Embedded Controller 1 OBF Interrupt. This interrupt is signaled when the OBF bit in the EC1 status register has been clear.
3	EC1_IBF	N	Embedded Controller 1 IBF Interrupt. This interrupt is signaled when the host writes to the EC1 Command or Data port.
4	EC0_OBF	N	Embedded Controller 0 OBF Interrupt. This interrupt is signaled when the OBF bit in the EC0 status register has been clear.
5	EC0_IBF	N	Embedded Controller 0 IBF Interrupt. This interrupt is signaled when the host writes to the EC0 Command or Data port.
6	EC3_OBF	N	Embedded Controller 3OBF Interrupt. This interrupt is signaled when the OBF bit in the EC3 status register has been clear.
7	EC3_IBF	N	Embedded Controller 3 IBF Interrupt. This interrupt is signaled when the host writes to the EC3 Command or Data port.
30-8	Reserved	N	Reserved
31	Reserved	N	This bit is always reserved

Note 17-10 Source bits have corresponding Enable and Result bits.

© 2012-2018 Microchip Technology Inc.

17.6.43 GIRQ20 ENABLE REGISTER

TABLE 17-58: GIRQ20 ENABLE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	F4h					32-bit	**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Reserved	Refer to Table 17-57, “Bit Definition IRQ20 Source Register,” on page 320						

17.6.44 GIRQ20 RESULT REGISTER

TABLE 17-59: GIRQ20 RESULT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	C8h					32-bit	**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	Refer to Table 17-57, “Bit Definition IRQ20 Source Register,” on page 320						

NORM

This bit is a read write bit used by the NORM instruction.

17.6.45 GIRQ21 SOURCE REGISTER

TABLE 17-60: GIRQ21 SOURCE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	104h					32-bit	**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	RWC except for reserved bits See Table 17-61 & Note 17-1 on page 302						
BIT NAME	Reserved	Refer To Table 17-61, “Bit Definition IRQ21 Source Register,” on page 322						

TABLE 17-61: BIT DEFINITION IRQ21 SOURCE REGISTER

Bit	Signal Name	Wake	Description
30:0	Reserved	N	Reserved
31	Reserved	N	This bit is always reserved

Note 17-11 Source bits have corresponding Enable and Result bits.

17.6.46 GIRQ21 ENABLE REGISTER

TABLE 17-62: GIRQ21 ENABLE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	108h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	▪	▪	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/W except for reserved bits See Table 17-61 & Note 17-1 on page 302						
BIT NAME	Reserved	Refer To Table 17-61, “Bit Definition IRQ21 Source Register,” on page 322						

17.6.47 GIRQ21 RESULT REGISTER

TABLE 17-63: GIRQ21 RESULT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	10Ch					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	▪	▪	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	Refer To Table 17-61, “Bit Definition IRQ21 Source Register,” on page 322						

NORM

This bit is a read write bit used by the NORM instruction.

© 2012-2018 Microchip Technology Inc.

17.6.48 GIRQ22 SOURCE REGISTER

TABLE 17-64: GIRQ22 SOURCE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	118h					32-bit	**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
BIT NAME	Reserved	Refer To Table 17-65, “Bit Definitions GIRQ22 Source Register,” on page 323						

TABLE 17-65: BIT DEFINITIONS GIRQ22 SOURCE REGISTER

BIT	EC TYPE	SIGNAL NAME	WAKE	DESCRIPTION
[7:0]	R/WC	GPIO[207:200]	Y	GPIO Interrupt Signal. This bit is set when a interrupt source is triggered. This bit is sticky; once set, it remains set until cleared by a write with the value ‘1’ to this bit. Write to this bit with a value of ‘0’ have no effect. **Note:** GIRQ22 is allocated for GPIOs 200 - 236. See pinout for GPIOs implemented.
[15:8]	R/WC	GPIO[217:210]	Y	
[23:16]	R/WC	GPIO[227:220]	Y	
[30:24]	R/WC	GPIO[236:230]	Y	
31	R	Reserved	N	This bit is always reserved

17.6.49 GIRQ22 ENABLE REGISTER

TABLE 17-66: GIRQ22 ENABLE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	11Ch					32-bit	**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Reserved	Refer To Table 17-65, “Bit Definitions GIRQ22 Source Register,” on page 323						

17.6.50 GIRQ22 RESULT REGISTER

TABLE 17-67: GIRQ22 RESULT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	120h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	Refer To Table 17-65, “Bit Definitions GIRQ22 Source Register,” on page 323						

NORM

This bit is a read write bit used by the NORM instruction.

17.6.51 GIRQ23 SOURCE REGISTER

TABLE 17-68: GIRQ23 SOURCE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	12Ch					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
BIT NAME	Reserved	Refer To Table 17-69, “Bit Definitions GIRQ23 Source Register,” on page 324						

TABLE 17-69: BIT DEFINITIONS GIRQ23 SOURCE REGISTER

Bit	Signal Name	Wake	Description
0	TIMER0	N	16-bit Timer Interrupt
1	TIMER1	N	16-bit Timer Interrupt
2	TIMER2	N	16-bit Timer Interrupt
3	TIMER3	N	16-bit Timer Interrupt
4	Reserved	N	Reserved
5	PFR	N	Power Fail Register Interrupt
6	Reserved	N	Reserved
7	WEEK_ALR	Y	Week Alarm Interrupt. Week Timer has reached it’s terminal count.
8	VCI_OVRD_IN	Y	Pin input of VCI_OVRD_IN
9	VCI_IN0	Y	input of VCI_IN0# pins
10	VCI_IN1	Y	input of VCI_IN1# pins
11	VCI_IN2	Y	input of VCI_IN2# pins
12	VCI_IN3	Y	input of VCI_IN3# pins

© 2012-2018 Microchip Technology Inc.

TABLE 17-69: BIT DEFINITIONS GIRQ23 SOURCE REGISTER (CONTINUED)

Bit	Signal Name	Wake	Description
13	HTIMER0	Y	Hibernation Timer Interrupt. This interrupt is signaled when the hibernation timer has counter down to zero.
14	HTIMER1	Y	Hibernation Timer Interrupt. This interrupt is signaled when the hibernation timer has counter down to zero.
15	Reserved	N	Reserved
16	CAPTURE TIMER	N	The Free Running timer in the Capture/Compare Time transitioned from FFFF_FFFFh to 0000_0000h
17	CAPTURE 0	N	Capture register 0 in the Capture/Compare Timer unit has acquired a new value
18	CAPTURE 1	N	Capture register 1in the Capture/Compare Timer unit has acquired a new value
19	CAPTURE 2	N	Capture register 2 in the Capture/Compare Timer unit has acquired a new value
20	CAPTURE 3	N	Capture register 3 in the Capture/Compare Timer unit has acquired a new value
21	CAPTURE 4	N	Capture register 4 in the Capture/Compare Timer unit has acquired a new value
22	CAPTURE 5	N	Capture register 5 in the Capture/Compare Timer unit has acquired a new value
23	COMPARE 0	N	Compare register 0 in the Capture/Compare Time unit tripped
24	COMPARE 1	N	Compare register 1 in the Capture/Compare Time unit tripped
25	VCI_IN4	Y	input of VCI_IN4# pin
26	VCI_IN5	Y	input of VCI_IN5# pin
[30:27]	Reserved	N	Reserved
31	Reserved	N	This bit is always reserved

17.6.52 GIRQ23 ENABLE REGISTER

TABLE 17-70: GIRQ23 ENABLE REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	130h					32-bit	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	■	■	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Reserved	Refer To Table 17-69, “Bit Definitions GIRQ23 Source Register,” on page 324						

17.6.53 GIRQ23 RESULT REGISTER

TABLE 17-71: GIRQ23 RESULT REGISTER

HOST ADDRESS							HOST SIZE	
EC OFFSET	134h					32-bit	EC SIZE	
POWER	VTR				0000_0000h		nSYS_RST DEFAULT	
BUS	EC SPB							
BIT	D31	D30	D29	■	■	D2	D1	D0
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W-	R	R	R	R	R	R	R
BIT NAME	NORM	Refer To Table 17-69, “Bit Definitions GIRQ23 Source Register,” on page 324						

Bit 31 NORM

This bit is a read write bit used by the NORM instruction.

17.6.54 IA GROUP SELECT REGISTER

TABLE 17-72: IA GROUP SELECT REGISTER

HOST ADDRESS							HOST SIZE	
EC OFFSET	200h					32-bit	EC SIZE	
POWER	VTR				00FF_FF00h		VTR POR DEFAULT	
BIT	D31	D30	D29	D28	D27	D26	D25	D24
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BIT	D23	D22	D21	D20	D19	D18	D17	D16
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	GIRQ_Enable[23:16]							
BIT	D15	D14	D13	D12	D11	D10	D9	D8
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	GIRQ_Enable[15:8]							
BIT	D7	D6	D5	D4	D3	D2	D1	D0
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							

© 2012-2018 Microchip Technology Inc.

GIRQ_ENABLE[23:8]

Group IRQ Enable. Enable or disable all interrupts in a group.

0= All Interrupts in the associated GIRQ will be disabled.

1= All Interrupts in the associated GIRQ will be enabled.

17.6.55 IRQ ENABLE REGISTER

TABLE 17-73: IRQ ENABLE REGISTER

HOST ADDRESS							**HOST SIZE**	
EC OFFSET	204h			32-bit			**EC SIZE**	
POWER	VTR			00FF_FF00h			**VTR POR DEFAULT**	
BIT	**D31**	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	IRQ Enable[23:16]							
BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	IRQ Enable[15:8]							
BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R	R	R
BIT NAME	IRQ Enable[7:3]					Reserved		

IRQ ENABLE[31:0]

IRQ Enable. Enable or disable all IRQs going to the ARC.

0= Respective IA IRQi will be disabled.

1= Respective IA IRQi will be enabled.

17.7 Extension Core Registers

17.7.1 EXTENSION CORE REGISTER R56

Register R56 is the Interrupt Aggregator Vector register. The Interrupt Aggregator automatically generates an address of a 4-byte location in the EC address space. The location contains a 4-byte address of an interrupt handler for the highest priority interrupt selected by the Interrupt Aggregator.

This register can be used in any ARC instruction with a 6-bit register address field.

TABLE 17-74: IA VECTOR REGISTER R56

BIT	D31	D30	D29	D28	D27	D26	D25	D24
TYPE	R	R	R	R	R	R	R	R
FIELD	0	0	0	0	0	0	0	0
BIT	D23	D22	D21	D20	D19	D18	D17	D16
TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
FIELD	VT_Base[12:5]							
BIT	D15	D14	D13	D12	D11	D10	D9	D8
TYPE	R/W	R/W	R/W	R/W	R/W	R	R	R
FIELD	VT_Base[4:0]					Group_Select[3:1]		
BIT	D7	D6	D5	D4	D3	D2	D1	D0
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	Group_ Select[0]	Bit_Select[4:0]					0	0

D[6:2]: BIT_SELECT[4:0]

This field is the index of the interrupt with the highest priority within the current group that is asserted, enabled, and set to a priority level that is equal to or greater than the current IA priority level. The interrupt with the highest bit number has the highest priority within a group.

This field is set by hardware and is read-only.

D[10:7]: GROUP_SELECT[3:0]

This field is the index of the GIRQ with the highest priority in the Interrupt Aggregator among GIRQs in which at least one interrupt is asserted, enabled and set to a priority level that is equal to or greater than the current IA priority level. The GIRQ with the highest index has the highest priority among the GIRQs.

This field is set by hardware and is read-only.

VT_BASE[11:0]

This field corresponds to bits 23:11 of the Interrupt Vector Table in the AHB address space. The IQ Vector Register is an address in the 24-bit AHB address space which contains the address of an interrupt handler. The AHB address space includes the Boot Memory, the Flash Memory and the Data/Instruction SRAM.

This field can be read and written by firmware. It should be initialized to the base address of the Interrupt Vector Table before interrupts are enabled.

© 2012-2018 Microchip Technology Inc.

17.7.2 EXTENSION CORE REGISTER R55

Extension Core Register R55 is a 6-bit register that sets the current priority level for the Interrupt Aggregator. Bits 31:6 are reserved and always return 0 on reads. An interrupt priority level of 4 or higher disables all interrupts.

This register can be used in any ARC instruction with a 6-bit register address field.

TABLE 17-75: IA PRIORITY REGISTER R55

BIT	D31	D30	D29	D28	D27	D26	D25	D24
TYPE	R	R	R	R	R	R	R	R
FIELD	Reserved							
BIT	D23	D22	D21	D20	D19	D18	D17	D16
TYPE	R	R	R	R	R	R	R	R
FIELD	Reserved							
BIT	D15	D14	D13	D12	D11	D10	D9	D8
TYPE	R	R	R	R	R	R	R	R
FIELD	Reserved							
BIT	D7	D6	D5	D4	D3	D2	D1	D0
EC TYPE	R	R	R	R	R	R/W	R/W	R/W
BIT NAME	Reserved		Hi_Pri			Current_Priority		

CURRENT_PRIORITY[2:0]

This is a read/write field that determines the current priority level for the Interrupt Aggregator. Only interrupts that are configured to be at this priority level or higher will be used to generate the interrupt vector. A Current_Priority value of 4 or higher will block all interrupts in the Interrupt Aggregator.

HI_PRI

This three-bit field is set whenever Extension Core Register R56 (IA Vector Register) is read. The value is the **priority level + 1** assigned to the interrupt that corresponds to the vector read in R56. It is the hardware that adds the 1 to the current priority level of the generated interrupt. For example, if the generated interrupt had a priority 2 then hardware will add 1 to it and write a value of 3 to the Hi_Pri bits.

18.0 WATCHDOG TIMER INTERFACE

18.1 General Description

The function of the Watchdog Timer is to provide a mechanism to detect if the embedded controller has failed.

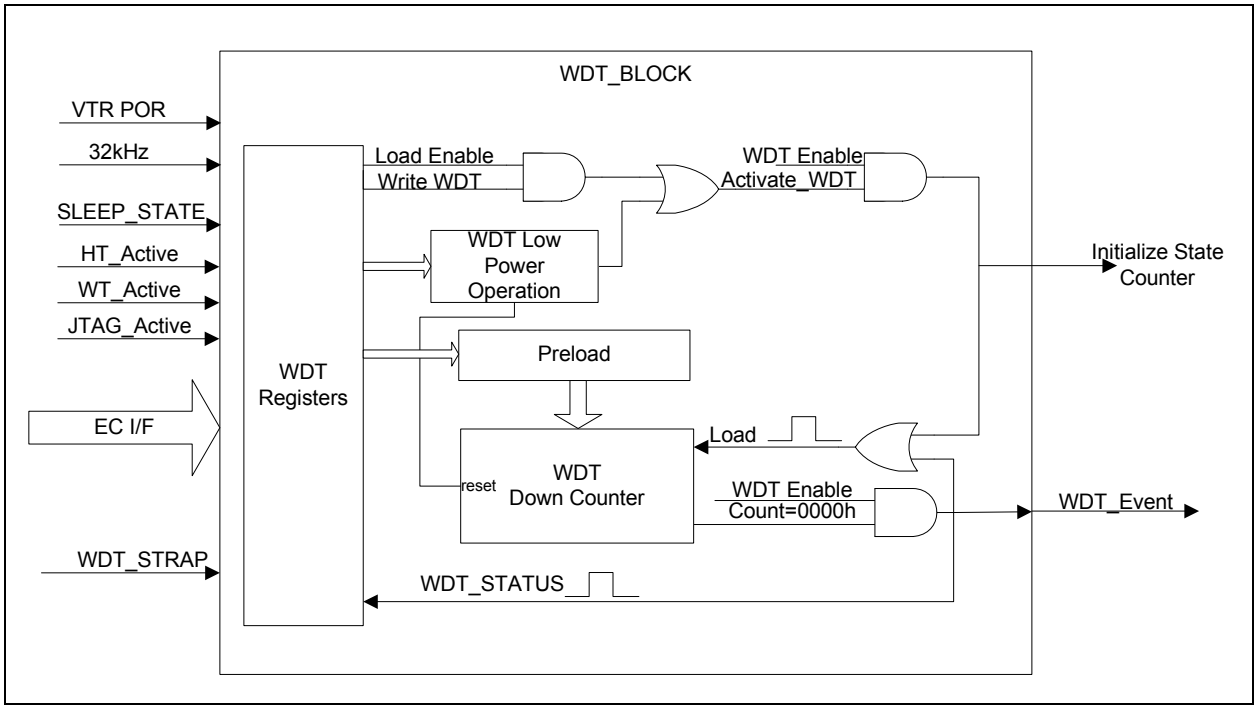
When enabled, the WATCHDOG Timer (WDT) circuit will generate a WDT Event if the user program fails to reload the WDT within a specified length of time known as the WDT Interval.

This timer can be held inactive via the WDT Stall feature if the Hibernation timer, Week Timer, or the JTAG interface are enabled and active. This featured if enabled can be used to avoid unintended system resets.

Some operations can be carried out without any delay, e.g., registers can be read at any time and disabling the WDT takes effect immediately. On the other hand, 'kicking' the WDT may have a latency of up to 1 32-kHz cycle (~ 30 us). Similarly, when the load register is altered, the WDT cannot be enabled for up to 1 32-kHz cycle. Note that the ring oscillator must not be stopped within one 32-kHz clock following register write events.

18.2 Block Diagram

FIGURE 18-1: Watchdog Timer Interface BLOCK DIAGRAM



© 2012-2018 Microchip Technology Inc.

18.3 Watchdog Timer Interface Signal List

TABLE 18-1: Watchdog Timer Interface SIGNAL LIST

Signal Name	Direction	Description
nSYS_RST	INPUT	Power on Reset to the block
32kHz	INPUT	X32K_CLK, Clock source for WDT logic
HT_Active	INPUT	Signal indicating the Hibernation Timer is active and counting. See Section 21.0, "Hibernation Timer," on page 366.
WT_Active	INPUT	Signal indicating the Week Timer is active and counting. See Section 22.0, "Week Alarm Interface," on page 370.
JTAG_Active	INPUT	Signal indicating the JTAG interface is active. See Section 44.0, "JTAG and XNOR," on page 580
SPB Interface	I/O Bus	Bus used by microprocessor to access the registers in this block.
WDT Event	OUTPUT	Pulse generated when WDT expires (Note 18-1)

Note 18-1 In the MEC1632, the WDT Event output is routed to the WDT_ALRT input (see Table 7-4, “Power, Clocks, and Resets Port List,” on page 96). Asserting the WDT Event output causes a Watch-Dog Timer Forced Reset (see Section 7.7.4, "Watch-Dog Timer Forced Reset," on page 130).

18.4 Power, Clocks and Reset

18.4.1 POWER DOMAIN

This block is powered by the VTR Power Supply.

See Section 7.1.1, "Power Configuration," on page 95 for details on power domains.

18.4.2 CLOCKS

This block has two clock inputs, the EC Bus Clock and X32K_CLK. The EC Bus Clock is used in the interface to the embedded controller accessible registers. The 32.768KHz X32K_CLK is the clock source for the Watchdog Timer functional logic, including the counter.

See Section 7.4, "Clock Generator," on page 98 for details on clocks.

18.4.3 RESET

This block is reset on a nSYS_RST.

See Section 7.6, "Reset Interface," on page 124 for details on reset.

18.4.4 POWER MANAGEMENT

In all cases, X32K_CLK does not affect Power Management; i.e., the Watchdog Timer Interface operates normally when the MCLK is stopped. The sleep enable inputs have no affect on the Watchdog Timer Interface and the clock required outputs are only asserted during register read/write cycles for as long as necessary to propagate updates to the block core (Table 18-2).

TABLE 18-2: WDT POWER MANAGEMENT

SLEEP_EN	Bus Access Cycle?	CLK_REQ	Description
X	Yes	1	CLK_REQ is only asserted for as long as necessary to propagate updates to the block core
	No	0	CLK_REQ is not asserted when the EC is not accessing the register interface. (Note that this block *cannot* prevent the chip from entering the system deepest sleep states.)

18.5 WDT Event output routing

The WDT Event (output) causes the Section 7.7.4, "Watch-Dog Timer Forced Reset," on page 130.

The WDT Event state is also retained through a Watch-Dog Timer Forced Reset in the WDT bit of the Power-Fail and Reset Status Register on page 148. The Power-Fail and Reset Status Register can generate a interrupt via the PFR interrupt GIRQ23 Source Register on page 324.

The WDT Event output is not directly connected to an EC interrupt.

18.6 WDT Operation

18.6.1 WDT ACTIVATION MECHANISM

The WDT is activated by the following sequence of operations during normal operation. Note that the WDT Load Register can be programmed only when WDT is disabled (WDT Enable = '1')

1. Load the WDT Load Register with the count value. '0' is an invalid load value.
2. Set the WDT Enable bit in the WDT Control Register.

The WDT Activation Mechanism starts the WDT decrementing counter.

18.6.2 WDT DEACTIVATION MECHANISM

The WDT is deactivated by the clearing the WDT Enable bit in the WDT Control Register. The WDT Deactivation Mechanism places the WDT in a low power state in which clock are gated and the counter stops decrementing.

18.6.3 WDT RELOAD MECHANISM

The WDT must be reloaded within periods that are shorter than the programmed watchdog interval; otherwise the WDT will underflow and a WDT Event will be generated and the WDT Status bit will be set in the WDT Control Register. It is the responsibility of the user program to continually execute sections of code which reload the watchdog timer (WDT) causing the counter to be reloaded

There are two methods of reloading the WDT: a write to the WDT Kick Register or the WDT Activation Mechanism.

18.6.4 WDT INTERVAL

The WDT Interval is the time it takes for the WDT to decrements from the WDT Load Register value to 0000h. The WDT Count Register value takes 1.007ms to decrement by 1 count.

18.6.5 WDT STALL OPERATION

The WDT has several events that can cause the WDT STALL. When a WDT STALL event is asserted, the WDT stops decrementing, and the WDT enters a low power state. When a WDT STALL event is de-asserted, the counter resumes decrementing from the count at which it stopped.

The WDT STALL feature has been implemented for convenience. If the system designer chooses not to utilize the WDT STALL feature, the WDT defaults with the WDT STALL feature disabled.

There are three Stall inputs to the WDT: HT_Active, WT_Active, and JTAG_Active, corresponding to the Hibernation Timer, the Week Alarm Timer, & the J-TAG interface being active. The Stall inputs have individual enable bits: HT STALL_EN, WT STALL_EN, JTAG STALL_EN bits in the WDT Control Register on page 334.

Note 18-2 Only a single instance of the Hibernation Timer (Hibernation Timer.0 on page 367) is routed to the HT_Active stall input of the Watchdog Timer Interface.

© 2012-2018 Microchip Technology Inc.

TABLE 18-3: WDT STALL EVENT BEHAVIOR

WDT Stall Input (Activity Indicator)	WDT Control Register on page 334		WDT Behavior	WDT Event Output
	STALL_EN Bit	WDT Enable Bit		
X	X	0	Counter is reset and not active. Clock source to counter is gated to save power.	0
X	0	1	Count is active. If counter > 0000h	0
0	1	1		
X	X	1	Count is decremented to 0000h	1
1	1	1	Counter is not active. Clock source to counter is gated to save power.	0

Note 18-3 When the counter reaches 0000h it wraps to the preload value and starts counting down again. This creates a pulse on the WDT Event output.

18.7 Instance Description

There is one instance of the Watchdog Timer Interface block implemented in the MEC1632.

Each instance of the Watchdog Timer Interface has its own Logical Device Number, and Base Address as indicated in Table 18-4.

TABLE 18-4: Watchdog Timer Interface BASE ADDRESS TABLE

Watchdog Timer Instance	LDN from (Table 4-2 on page 59)	AHB Base Address
Watchdog Timer	1h	F0_0400h

The Table 18-5 is a register summary for one instance of the Watchdog Timer Interface. Each EC address is indicated as an SPB Offset from its AHB address.

TABLE 18-5: Watchdog Timer Interface REGISTER SUMMARY

	EC Interface			
Register Name	SPB Offset	Byte Lane	EC Type	Notes
WDT Load Register	00h	0	R/W	
		1		
WDT Control Register	04h	0	R/W	
WDT Kick Register	08h	0	W	
WDT Count Register	0Ch	0	R	
		1		

Note: All Registers listed in Table 18-5 are powered by VTR and reset by nSYS_RST.

Note 18-4 All register are clocked by the EC Bus Clock.

18.8 Detailed Register Descriptions

18.8.1 WDT LOAD REGISTER

TABLE 18-6: WDT LOAD REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	00h			16-bit			**EC SIZE**	
POWER	VTR			FFFFh			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D15**	**D14**	**D13**	■	■	**D2**	**D1**	**D0**
HOST TYPE	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	WDT load[15:0]							

WDT LOAD[15:0]

Writing this field reloads the Watch Dog Timer counter.

WDT Load Register can be programmed only when WDT Enable = '0'.

'0' is not a valid load value.

To verify that load has taken place, it is recommended that software polls the WDT Count Register until its value reflects that of the new load value.

18.8.2 WDT CONTROL REGISTER

TABLE 18-7: WDT CONTROL REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	04h			8-bit			**EC SIZE**	
POWER	VTR			00h (Note 18-5)			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/WC	R/W
BIT NAME	Reserved			JTAG STALL_EN	WT STALL_EN	HT STALL_EN	WDT Status	WDT Enable

Note 18-5 The default for the WDT Control Register changes depending on the state of the WDT Status bit.

WDT ENABLE

Note:	The default of the WDT is inactive.

In WDT Operation, the WDT is activated by the sequence of operations defined in Section 18.6.1, "WDT Activation Mechanism" and deactivated by the sequence of operations defined in Section 18.6.2, "WDT Deactivation Mechanism". In WDT STALL Operation, hardware may be enabled to automatically activate and deactivate the WDT.

© 2012-2018 Microchip Technology Inc.

WDT STATUS

WDT Status is set by hardware if the last reset of MEC1632 was caused by an underflow of the WDT. See Section 18.6.3, "WDT Reload Mechanism," on page 332 for more information.

This bit must be cleared by the EC firmware writing a '1' to this bit. Writing a '0' to this bit has no effect.

JTAG STALL_EN

This bit is used to enable the JTAG_Active (JTAG_RST# pin not asserted) WDT STALL Operation on page 332.

0= JTAG_Active WDT STALL Operation not enabled

1= JTAG_Active WDT STALL Operation enabled

WT STALL_EN

This bit is used to enable the WT_Active (Week Timer) WDT STALL Operation on page 332.

0= WT_Active WDT STALL Operation events not enabled

1= WT_Active WDT STALL Operation events enabled.

HT STALL_EN

This bit is used to enable the HT_Active (Hibernation Timer) WDT STALL Operation on page 332.

0= HT_Active WDT STALL Operation events not enabled

1= HT_Active WDT STALL Operation events enabled.

18.8.3 WDT KICK REGISTER

TABLE 18-8: WDT KICK REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	08h			8-bit			**EC SIZE**	
POWER	VTR			n/a			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
EC TYPE	W	W	W	W	W	W	W	W
BIT NAME	Kick							

KICK

The WDT Kick Register is a strobe. Reads of the WDT Kick Register return 0.

Writes to the WDT Kick Register cause the WDT to reload the WDT Load Register value and start decrementing when the WDT Enable bit in the WDT Control Register is set to '1'. When the WDT Enable bit in the WDT Control Register is cleared to '0', writes to the WDT Kick Register have no effect.

18.8.4 WDT COUNT REGISTER

TABLE 18-9: WDT COUNT REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	0Ch			16-bit			**EC SIZE**	
POWER	VTR			FFFFh			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D15**	**D14**	**D13**	▪	▪	**D2**	**D1**	**D0**
HOST TYPE	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	WDT COUNT[15:0]							

WDT COUNT[15:0]

This read-only register provide the current WDT count.

© 2012-2018 Microchip Technology Inc.

19.0 HDMI-CEC INTERFACE CONTROLLER

19.1 Overview

This chapter describes an implementation for an HDMI Consumer Electronics Control (CEC) Interface as defined in References [1].

The HDMI-CEC Interface Controller described in this document handles in hardware the transfer of CEC data across the physical interface; including, all initiator/follower data/framing bit timing, logical address decoding, contention detection/Lost Arbitration, Line Error Handling and message block acknowledgment.

Firmware is required to configure physical addressing and the CEC high level protocol, which is beyond the scope of this document.

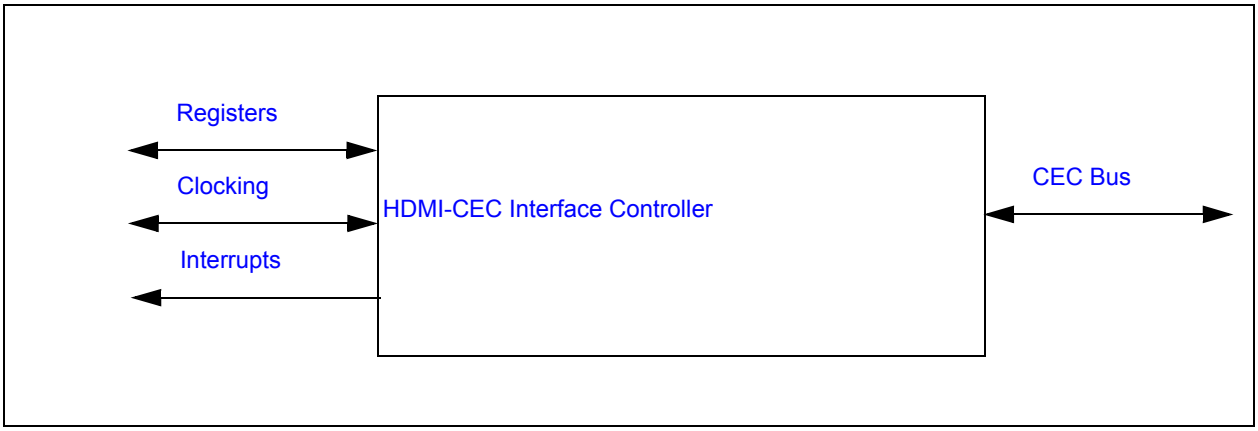
19.2 References

1. High-Definition Multimedia Interface Specification Version 1.3a, Supplement 1 CEC, November 10, 2006.

Note:	Italicized text in this chapter typically refers to content defined in References [1].

19.3 Block Diagram

FIGURE 19-1: HDMI-CEC Interface Controller Block Diagram



19.4 CEC Bus

19.4.1 OVERVIEW

The CEC Bus is a single wire bus that provides high-level control functions between all of the various audiovisual products in a user's environment (see *3, "Overview"* in References [1]). The MEC1632 CEC Bus interface includes CEC_IN and CEC_OUT pins as defined in MEC1632 Pin Configuration.

19.4.2 EXTERNAL INTERFACE

Figure 19-2 illustrates an example of an external CEC Bus interface network. CEC_IN and CEC_OUT pin characteristics, including polarity, direction and buffer type, can be programmed using the GPIO Pin Control Registers (Table 19-1).

FIGURE 19-2: CEC Bus APPLICATION EXAMPLE

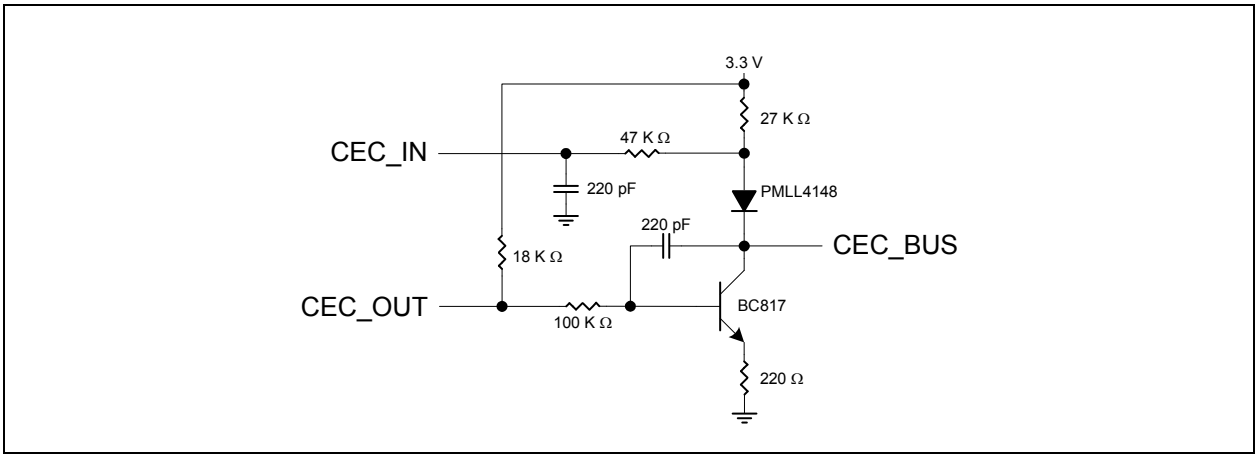


TABLE 19-1: EXTERNAL INTERFACE PIN CONFIGURATION

Signal Function	Direction	Output Buffer Type	Polarity
CEC_IN	INPUT	X	NON-INVERTED
CEC_OUT	OUTPUT	OPEN DRAIN	INVERTED

19.4.3 FILTERING

When enabled, the HDMI-CEC Interface Controller filters the CEC_IN pin as defined in Table 19-2. Filtering is enabled using the FILTEN bit in the CEC Control Register.

TABLE 19-2: HDMI-CEC INTERFACE INPUT FILTER

	Parameter	Symbol	Value			Units	Notes
			MIN	TYP	MAX		
1.	Filtered Pulse Width	t_{FPW}	100	–	–	ns	

19.4.4 PAD

Table 19-3 is taken from *CEC 4, "Electrical Specification"* in References [1]. These requirements are to be satisfied using an external electrical network.

TABLE 19-3: ELECTRICAL SPECIFICATIONS

	Parameter	Symbol	Value			Units	Notes
			MIN	TYP	MAX		
1.	Leakage Current in Powered-off State	I_{IL}	–	–	1.8	µA	Note 19-1
2.	Output Voltage Logic '0'	V_{OL}	0.0	–	0.6	V	Note 19-3
3.	Output Voltage Logic '1'	V_{OH}	2.5	–	3.63	V	–
4.	Output Current Logic '0'	I_{OL}	–	–	20	mA	–
5.	High to Low Input Voltage Threshold Logic '0'	V_{IL}	–	–	0.8	V	–
6.	Low to High Input Voltage Threshold Logic '1'	V_{IH}	–	–	2.0	V	–
7.	Input Hysteresis	V_{HYS}	–	400	–	mV	Note 19-2

© 2012-2018 Microchip Technology Inc.

TABLE 19-3: ELECTRICAL SPECIFICATIONS (CONTINUED)

	Parameter	Symbol	Value			Units	Notes
			MIN	TYP	MAX		
8.	Rise Time (10% to 90%)	t_R	–	–	250	µs	–
9.	Fall Time (10% to 90%)	t_F	–	–	50	µs	–
10.	Internal Device Pull-up	R_{PU}	23.4	26	28.6	K Ohms	–
11.	Bus Capacitance	C_{LOAD}	–	–	7200	pF	Note 19-4

Note 19-1 this effectively requires that the internal pull-up circuit shall be disconnected from the CEC line when the device is off. For example, this can be implemented by connecting an isolating diode between the CEC input pin and the internal pull-up circuit, such that diode is reverse-biased in the off state with an external device pulling-up the CEC line.

Note 19-2 input hysteresis is handled by the pad.

Note 19-3 during transition from Logic ‘1’ to Logic ‘0’ a negative overshoot with maximum 300mV and up to 150µs duration is allowed.

Note 19-4 the device shall remain within specification under the full-range of load conditions.

19.5 Interrupts

If any of the Interrupt Interface (Table 19-4) status bits are asserted when the corresponding enable bit is asserted, an external interrupt is asserted.

Note: CEC_INT in the GIRQ16 Source Register is the HDMI-CEC Interface Controller interrupt source bit in the MEC1632 (see Section 17.0, "EC Interrupt Aggregator").

TABLE 19-4: INTERRUPT INTERFACE

	Status Bits		Enable Bits	
	Name	Register	Name	Register
1.	IFDONE	Initiator Status Register	IFDONE_EN	Initiator Control Register
2.	IFE		IFE_EN	
3.	FFDONE	Follower Status Register	FFDONE_EN	Follower Control Register
4.	FDR		FDR_EN	
5.	FFF		FFF_EN	

19.6 Power

19.6.1 INTERFACE

The HDMI-CEC Interface Controller requires a single power plane (VTR). Power Management can be controlled by the Clocking interface as described in Section 19.6.2.

19.6.2 POWER MANAGEMENT

Note: The CEC bit in the EC Blocks Sleep Enables Register 2 and the EC Blocks Clock Required Status Register 2 are the sleep enable and clock required status bits for the HDMI-CEC Interface Controller in the MEC1632 (see Section 7.0, "Power, Clocks, and Resets," on page 95).

TABLE 19-5: HDMI-CEC Interface Controller Power Management

ACTIVATE(Note 19-5)	External SLEEP_EN Input (Note 19-6)	Internal Idle (Note 19-7)	Core Clock Required Status Output (Note 19-6)	Mode	Power	Description
0	X	X	0	DISABLED	MINIMUM	The HDMI-CEC Interface Controller is disabled by firmware and the core clock is gated 'off' internally. **Note:** It is up to the host to ensure that the block is not in use before the internal enable bit is asserted.
1	0	NOT IDLE	1	FULL POWER	MAXIMUM	FULL POWER mode identifies the normal operation mode where the block is neither disabled by firmware nor commanded to sleep by the Clocking interface.
		IDLE				
	1	NOT IDLE		PREPARING TO SLEEP		A sleep command has been asserted but the core clock is still required because the block is not idle.
		IDLE	0	SLEEPING	MINIMUM	A sleep command has been asserted, the block is idle and the core clocks are stopped. In the SLEEPING mode, normal operation cannot be resumed until the external sleep command is not asserted.

Note 19-5 the ACTIVATE bit is in the CEC Control Register.

Note 19-6 the external sleep enable input and clock required status output are part of the Clocking interface.

Note 19-7 the INTERNAL IDLE state in part depends upon the state of the IDLE bit in the Initiator Status Register.

19.7 Clocking

TABLE 19-6: Clocking SIGNAL INTERFACE

Signal Name	Description	Reference
MCLK_DIV203_EN	Core Clock	Table 19-7, "Core Clock Timing"
CEC_SLEEP_ENABLE	Sleep Enable Command Input	Section 19.6.2, "Power Management," on page 339
CEC_CLOCK_REQUIRED	Clock Required Status Output	

TABLE 19-7: CORE CLOCK TIMING

Parameter	Symbol	MIN	TYP	MAX	Units
Core Clock Frequency	f_{CLOCK}	90	100	110	KHz

© 2012-2018 Microchip Technology Inc.

19.8 Functional Description

19.8.1 OVERVIEW

The HDMI-CEC Interface Controller includes an Initiator Interface and a Follower Interface that function independently within the constraints defined in the subsections that follow. Data transfer buffering in each interface include a 9-bit wide x 16 deep FIFO and a shift register. Each interface can generate interrupts as described in Section 19.5, "Interrupts," on page 339 and the Initiator Interface hardware senses the Bus Idle Condition to initiate message frame transfers.

19.8.2 INITIATOR INTERFACE

19.8.2.1 Overview

The Initiator Interface describes data transfers in the HDMI-CEC Interface Controller as an initiator when the ACTIVATE bit is asserted. The Initiator Interface is disabled when the ACTIVATE bit is not asserted.

Specific accesses to the Initiator Data Register, Initiator Control Register and Initiator Status Register are required for successful Message Initiation. The Initiator Interface hardware is also capable of detecting Lost Arbitration.

Note that enforcing CEC message frame size constraints in the HDMI-CEC Interface Controller is the responsibility of firmware because the Initiator Interface can send any number of message blocks in a frame.

19.8.2.2 Message Initiation

Initiator Interface begins a message frame transaction when the START bit in the Initiator Control Register is asserted, the IFE bit is not asserted, and hardware detects a Bus Idle Condition as described in Section 19.8.4 on page 342.

APPLICATION NOTE: Recommended Message Initiation procedure:
- Assert the IFLUSH bit to clear the Initiator Interface FIFO,
- Write all message data to the Initiator Interface FIFO using the Initiator Data Register,
- Assert the START bit.

The message frame transaction is complete, or terminated because of an error (see Section 19.8.2.4, "Error Handling," on page 341), when the IFDONE bit in the Initiator Status Register is asserted.

To transfer message frames larger than 16 blocks, additional message data can be written to the Initiator Data Register whenever the IFE interrupt is asserted. In this case, data must be written within 20 milliseconds to avoid an underrun error.

19.8.2.3 Lost Arbitration

Lost Arbitration occurs when the Initiator Interface detects bus contention during CEC line arbitration, in which case the LAB bit in the Initiator Status Register is asserted. As a result of Lost Arbitration the transfer is aborted by the Initiator Interface and the task of decoding the rest of the message header is passed to the Follower Interface. If the destination logical address matches one of the addresses configured in the Claimed Logical Addresses Register, the Follower Interface continues decoding the rest of the message frame.

APPLICATION NOTE: Following Lost Arbitration, the Initiator Interface firmware is responsible for flushing the Initiator Interface FIFO, and re-initiating the transaction (see also Section 19.8.2.4, "Error Handling," on page 341).

19.8.2.4 Error Handling

If at any time during an Initiator Interface message transfer the UNDRN, ACKERR, LAB, or CE bits in the Initiator Status Register are asserted, the transfer is aborted by the initiator and the IFDONE bit is asserted. Note that in all cases the transfer is terminated, except during Lost Arbitration as described in Section 19.8.2.3.

Following message frame errors as defined above, the IFLUSH bit in the Initiator Control Register may need to be asserted to delete residual data in the Initiator Interface FIFO.

19.8.3 FOLLOWER INTERFACE

19.8.3.1 Overview

The Follower Interface in the HDMI-CEC Interface Controller responds to data transfers when the ACTIVATE bit is asserted and the Claimed Logical Addresses Register is greater than zero. The Follower Interface is disabled when the ACTIVATE bit is not asserted; if the ACTIVATE bit is asserted, no message frames will be decoded when the Claimed Logical Addresses Register is zero.

Specific accesses to the Follower Data Register, Follower Control Register and Follower Status Register are required for successful Follower Message Response. The Follower Interface hardware is also capable of Line Error Handling as described in Section 19.8.3.3. The Follower Interface can respond to frames that contain any number of message blocks.

19.8.3.2 Follower Message Response

19.8.3.2.1 Overview

When the HDMI-CEC Interface Controller is activated as described in Section 19.8.3.1, the Follower Interface responds to all signaling on the CEC Bus when the Initiator Interface is not active, or following Lost Arbitration. The Follower Interface supports Directed Messages and Broadcast Messages.

When the FFDONE bit in the Follower Status Register is asserted, the Follower Interface has successfully decoded a single message frame and placed it in the Follower Interface FIFO. It is the responsibility of firmware to read the message data using the Follower Data Register before another message arrives. If the Follower Interface tries to write received message data to the FIFO when the FFF bit is asserted, an overrun error occurs (see Section 19.9.7.1, "OVRN," on page 348), the current data block is negatively acknowledged, and the message is terminated.

In addition to detecting overrun errors, the Follower Interface implements Line Error Handling for message blocks that cannot be decoded successfully, as described in Section 19.8.3.3 on page 342.

19.8.3.2.2 Directed Messages

When the Follower Interface decodes a message header addressed to one of the logical device addresses configured in the Claimed Logical Addresses Register that are less than 15, the message data blocks are decoded until the EOM bit is detected. These are Directed Messages.

Acknowledge bit handling for Directed Messages is different than for Broadcast Messages (see Section 19.9.6.4, "ACKERR," on page 347).

19.8.3.2.3 Broadcast Messages

When the Follower Interface decodes a message header addressed to logical device address 15, which like directed message addresses is configured in the Claimed Logical Addresses Register, the message data blocks are decoded until the EOM bit is detected. These are Broadcast Messages.

Acknowledge bit handling for Broadcast Messages is different than for Directed Messages (see Section 19.9.6.4, "ACKERR," on page 347).

19.8.3.3 Line Error Handling

Whenever the Follower Interface detects spurious pulses on the CEC Bus while decoding a message frame, the BERR bit in the Follower Status Register is asserted and the initiator is notified that a potential error has occurred. Spurious pulses and initiator notification are defined in *CEC 7.4, "CEC Line Error Handling"* in References [1].

Line Error Handling in the HDMI-CEC Interface Controller also includes bus time-out detection, which is indicated by the BTO bit in the Follower Status Register. In all cases, the Follower Interface response to Line Error Handling includes terminating message frame reception.

19.8.4 BUS IDLE CONDITION

The Bus Idle Condition is uniquely determined for each device on the CEC Bus and depends upon the state of the CEC Bus itself and the device system state. In the HDMI-CEC Interface Controller, the Bus Idle Condition is considered 'asserted' when the IDLE bit in the Initiator Status Register is '1,' which occurs according to the signal free time as defined in *CEC 9, "CEC Arbitration"* in References [1].

The start of a message frame by the HDMI-CEC Interface Controller as an initiator (see Section 19.8.2, "Initiator Interface," on page 341) requires that the Bus Idle Condition be asserted, which is determined by device hardware as described in Table 19-8.

© 2012-2018 Microchip Technology Inc.

TABLE 19-8: Bus Idle Condition ASSERTION

	SFT (Note 19-8)	Factor
1.	3	The previous message was send by the internal initiator and was not successful.
2.	5	RESET (including a hardware reset), or the previous message was send by an external initiator.
3.	7	The previous message was send by the internal initiator and was successful.

Note 19-8 the Signal Free Time (SFT) is determined by the specified number, or greater, of nominal data bit periods where the CEC Bus remains high following the end of the last message.

For example, if the HDMI-CEC Interface Controller sends a successful message frame (A) it will not be eligible to send a subsequent frame for 16.8 ms (2.4 ms x 7) nominal following the end of frame A. In this same scenario, if an external initiator begins a message frame (B) 12 ms (2.4 ms x 5) after the end of frame A, the internal HDMI-CEC Interface Controller initiator will then be eligible to send a subsequent frame 12 ms after the end of frame B.

In all cases, the IDLE bit is de-asserted in the same bit period as a valid (or invalid) start bit. The IDLE bit does not generate an interrupt.

Immediately following the end of an Initiator Interface message transfer, successful or otherwise, the Bus Idle Condition can be temporarily overridden using the SFT5 bit in the CEC Control Register (see Section 19.9.2, "CEC Control Register," on page 344).

19.9 Registers

19.9.1 SUMMARY

TABLE 19-9: HDMI-CEC Interface Controller REGISTER SUMMARY

Register Name	Mnemonic	EC Interface			Notes
		SPB Offset	Byte Lane	EC Type	
CEC Control Register	CCR	00h	0	R/W	–
Claimed Logical Addresses Register	CLAR	04h	1-0	R/W	–
Initiator Data Register	IDR	08h	1-0	R/W	–
Follower Data Register	FDR	0Ch	1-0	R	–
Initiator Status Register	ISTR	10h	0	R/WC	Note 19-9
Follower Status Register	FSTR	12h	2	R/WC	Note 19-9
Initiator Control Register	ICR	18h	0	R/W	–
Follower Control Register	FCR	1Ch	0	R/W	–
MCHP Reserved Register	MCHPRSRV	20h	3-0	R/W	–

Note 19-9 R/WC type bits must be written with a ‘1’ to clear the bit.

19.9.2 CEC CONTROL REGISTER

TABLE 19-10: CEC Control Register

HOST ADDRESS	n/a			n/a				**HOST SIZE**
EC OFFSET	00h			8-bit				**EC SIZE**
POWER	VTR			00h				**DEFAULT**
BUS	EC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R/W	R/W	R/W	R/W
BIT NAME	Reserved				SFT5	FILTEN	RESET	ACTIVATE

19.9.2.1 ACTIVATE

When the ACTIVATE bit is asserted (‘1’), the HDMI-CEC Interface Controller is enabled for normal operation, subject to the constraints of the Power Management interface as defined in Section 19.6.2 on page 339.

When the ACTIVATE bit is not asserted (default), the HDMI-CEC Interface Controller is disabled and in the lowest power consumption state. The ACTIVATE bit does not affect the Registers.

19.9.2.2 RESET

When the RESET bit is asserted, the hardware state machines and Registers, except for the ACTIVATE bit, are reset to their default state. The RESET bit is cleared by hardware within one register access cycle.

APPLICATION NOTE: If there is an error condition that requires an HDMI-CEC Interface Controller RESET, the reset operation should completed within the minimum bus idle time to avoid unnecessary message NACKs.

19.9.2.3 FILTEN

When the FILTEN bit is asserted (‘1’), the Filtering on the CEC Bus is enabled. When FILTEN is not asserted (default), Filtering is disabled. See Section 19.4.3, "Filtering," on page 338.

19.9.2.4 SFT5

When the SFT5 bit is asserted (‘1’), the Signal Free Time (SFT) is changed to ‘5’ once only (see Table 19-8, “Bus Idle Condition Assertion,” on page 343).

The SFT5 bit is cleared, and normal SFT behavior resumes when the Bus Idle Condition is asserted.

19.9.2.5 Reserved

Reserved bits cannot be written and return ‘0’ when read.

© 2012-2018 Microchip Technology Inc.

19.9.3 CLAIMED LOGICAL ADDRESSES REGISTER

TABLE 19-11: Claimed Logical Addresses Register

HOST ADDRESS	n/a					n/a	**HOST SIZE**	
EC OFFSET	04h					16-bit	**EC SIZE**	
POWER	VTR					0000h	**DEFAULT**	
BUS	EC SPB							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	ADDR15	ADDR14	ADDR13	ADDR12	ADDR11	ADDR10	ADDR9	ADDR8
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	ADDR7	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0

When any bit in the Claimed Logical Addresses Register is asserted (‘1’), Directed Messages and Broadcast Messages addressed to that logical address are claimed by the Follower Interface. The Claimed Logical Addresses Register default effectively disables the Follower Interface.

19.9.4 INITIATOR DATA REGISTER

19.9.4.1 Description

The Initiator Data Register is two bytes wide and is used to write the Initiator Interface FIFO for CEC header and data block values. For 8-bit access cycles, the most significant byte of the Initiator Data Register must be written first. See also *CEC 6.1, “Header/Data Block Description”* in References [1].

Reads of the Initiator Data Register return zero.

TABLE 19-12: Initiator Data Register

HOST ADDRESS	n/a					n/a	**HOST SIZE**	
EC OFFSET	08h					16-bit	**EC SIZE**	
POWER	VTR					0000h	**DEFAULT**	
BUS	EC SPB							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	W
BIT NAME	Reserved							EOM
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	W	W	W	W	W	W	W	W
BIT NAME	INITIATOR_DATA							

19.9.4.2 EOM

The EOM bit is used for the End of Message bit in header/data blocks in CEC message frames. When the EOM bit is '0,' more data blocks follow; a '1' specifies that the message is complete.

19.9.4.3 INITIATOR_DATA

The INITIATOR_DATA register is used for the Information bits in header/data blocks in CEC message frames.

19.9.4.4 Reserved

Reserved bits cannot be written and return '0' when read.

19.9.5 FOLLOWER DATA REGISTER

19.9.5.1 Description

The Follower Data Register is two bytes wide and is used to read the Follower Interface FIFO for CEC header and data block values. For 8-bit access cycles, the most significant byte of the Follower Data Register must be read first. See also *CEC 6.1, "Header/Data Block Description"* in References [1].

Reads of the Follower Data Register when the FFNE bit in the Follower Status Register is not asserted ('0') return undefined data.

TABLE 19-13: Follower Data Register

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	0Ch			16-bit			**EC SIZE**	
POWER	VTR			0000h			**DEFAULT**	
BUS	EC SPB							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							EOM
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	FOLLOWER_DATA							

19.9.5.2 EOM

The EOM bit is the received End of Message bit from header/data blocks in CEC message frames. When the EOM bit is '0,' more data blocks follow; a '1' specifies that the message is complete.

19.9.5.3 FOLLOWER_DATA

The FOLLOWER_DATA register is used for the received Information bits from header/data blocks in CEC message frames.

19.9.5.4 Reserved

Reserved bits cannot be written and return '0' when read.

© 2012-2018 Microchip Technology Inc.

19.9.6 INITIATOR STATUS REGISTER

TABLE 19-14: Initiator Status Register

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	10h			8-bit			**EC SIZE**	
POWER	VTR			21h			**DEFAULT**	
BUS	EC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/WC	R	R/WC	R/WC	R/WC	R/WC	R
BIT NAME	Reserved	IFDONE	IFE	CE	ACKERR	UNDRN	LAB	IDLE

19.9.6.1 IDLE

When the IDLE bit is asserted (‘1’), the CEC Bus is idle (Bus Idle Condition). When the IDLE bit is not asserted (‘0’), the CEC Bus is busy.

See Section 19.8.4, "Bus Idle Condition," on page 342 for details regarding the behavior of the IDLE bit.

19.9.6.2 LAB

The LAB (Lost Arbitration) bit is asserted when bus contention is detected by the Initiator Interface during CEC line arbitration (see *CEC 9, “CEC Arbitration”* in References [1]).

See also Section 19.9.6.7, "IFDONE," on page 347.

19.9.6.3 UNDRN

The UNDRN (Initiator Underrun) bit is asserted (‘1’) when the initiator shift register requires data from the Initiator Interface FIFO and the IFE bit is asserted. See also Section 19.9.6.7, "IFDONE," on page 347.

19.9.6.4 ACKERR

The ACKERR (Acknowledge Error) bit is asserted (‘1’) when the follower NACKs an acknowledge bit, both for Directed Messages and Broadcast Messages.

For example, in Directed Messages if the follower acknowledge bit response is ‘1,’ the ACKERR bit is ‘1;’ in Broadcast Messages if a follower acknowledge bit response is ‘0,’ the ACKERR bit is ‘1.’

See also Section 19.9.6.7, "IFDONE," on page 347.

19.9.6.5 CE

The CE (Contention Error) bit is asserted (‘1’) when bus contention is detected by the Initiator Interface at any point following CEC line arbitration. This is likely to be a consequence of CEC Line Error Handling (see *CEC 7.4, “CEC Line Error Handling”* in References [1]).

See also Section 19.9.6.7, "IFDONE," on page 347.

19.9.6.6 IFE

The IFE (Initiator FIFO Empty) bit is asserted (‘1’) when the Initiator Interface retrieves the last data entry from the FIFO. The IFE bit can generate a level sensitive interrupt as described in Section 19.5, "Interrupts," on page 339.

19.9.6.7 IFDONE

The IFDONE (Initiator Frame Done) bit is asserted (‘1’) when a message block with the EOM bit asserted (‘1’) has been transferred by the Initiator Interface, or the message frame has been terminated because an error occurred as defined by the LAB, UNDRN, ACKERR, and CE bits.

The IFDONE bit can generate a level sensitive interrupt as described in Section 19.5, "Interrupts," on page 339.

19.9.6.8 Reserved

Reserved bits cannot be written and return ‘0’ when read.

19.9.7 FOLLOWER STATUS REGISTER

TABLE 19-15: Follower Status Register

HOST ADDRESS	n/a				n/a		**HOST SIZE**	
EC OFFSET	12h				8-bit		**EC SIZE**	
POWER	VTR				00h		**DEFAULT**	
BUS	EC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/WC	R/WC	R	R	R/WC	R/WC	R/WC
BIT NAME	Reserved	FFDONE	FDR	FFF	FFNE	BTO	BERR	OVRN

19.9.7.1 OVRN

The OVRN (Follower Overrun) bit is asserted (‘1’) when data from the follower shift register is transferred to the Follower Interface FIFO when the FFF bit is asserted. See also Section 19.8.3, "Follower Interface," on page 342.

See also Section 19.9.7.7, "FFDONE," on page 348.

19.9.7.2 BERR

The BERR (Bus Error Detected) bit is asserted (‘1’) when the follower detects spurious pulses on the CEC Bus as defined in Section 19.8.3.3, "Line Error Handling," on page 342.

See also Section 19.9.7.7, "FFDONE," on page 348.

19.9.7.3 BTO

The BTO (Bus Time Out Detected) bit is asserted (‘1’) when the follower detects that the CEC Bus is held high too long (see Section 19.8.3.3, "Line Error Handling," on page 342).

See also Section 19.9.7.7, "FFDONE," on page 348.

19.9.7.4 FFNE

The FFNE (Follower FIFO Not Empty) bit is asserted (‘1’) when there is data in the Follower Interface FIFO. The FFNE bit is not asserted (‘0’) when the Follower Interface FIFO is empty.

19.9.7.5 FFF

The FFF (Follower FIFO Full) bit is asserted (‘1’) when there is no room in the Follower Interface FIFO for more data. The FFF bit can generate a level sensitive interrupt as described in Section 19.5, "Interrupts," on page 339.

19.9.7.6 FDR

The FDR (Follower Data Ready) bit is asserted (‘1’) whenever received data is written to the Follower Interface FIFO. The FDR bit can generate a level sensitive interrupt as described in Section 19.5, "Interrupts," on page 339.

19.9.7.7 FFDONE

The FFDONE (Follower Frame Done) bit is asserted (‘1’) when a message block with the EOM bit asserted (‘1’) has been received by the Follower Interface, or the message frame has been terminated because an error occurred as defined by the OVRN, BERR, and BTO bits.

The FFDONE bit can generate a level sensitive interrupt as described in Section 19.5, "Interrupts," on page 339.

19.9.7.8 Reserved

Reserved bits cannot be written and return ‘0’ when read.

© 2012-2018 Microchip Technology Inc.

19.9.8 INITIATOR CONTROL REGISTER

TABLE 19-16: Initiator Control Register

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	18h			8-bit			**EC SIZE**	
POWER	VTR			00h			**DEFAULT**	
BUS	EC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/W	R/W	R	R	R	R/W	R/W
BIT NAME	Reserved	IFDONE_EN	IFE_EN	Reserved			START	IFLUSH

19.9.8.1 IFLUSH

When the IFLUSH bit is asserted (‘1’), data in the Initiator Interface FIFO is cleared. When set, the IFLUSH bit is automatically cleared by hardware within one register access cycle. See also Section 19.8.2.4, "Error Handling," on page 341.

The IFLUSH bit can be used to terminate an Initiator Interface message frame transfer.

19.9.8.2 START

When the START bit is asserted (‘1’), the Initiator Interface begins a message frame transfer if the IFE bit is not asserted. If the Initiator Interface FIFO is empty when the START bit is asserted, the message frame transfer begins as soon as a write to the Initiator Data Register occurs. In both cases, the START bit is automatically cleared by hardware within one register access cycle.

The START bit only needs to be asserted once per message frame.

19.9.8.3 IFE_EN

Enable bit for the IFE interrupt (see Section 19.9.6.6, "IFE," on page 347).

19.9.8.4 IFDONE_EN

Enable bit for the IFDONE interrupt (see Section 19.9.6.7, "IFDONE," on page 347).

19.9.8.5 Reserved

Reserved bits cannot be written and return ‘0’ when read.

19.9.9 FOLLOWER CONTROL REGISTER

TABLE 19-17: Follower Control Register

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	1Ch			8-bit			**EC SIZE**	
POWER	VTR			00h			**DEFAULT**	
BUS	EC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/W	R/W	R/W	R	R	R	R/W
BIT NAME	Reserved	FFDONE_EN	FDR_EN	FFF_EN	Reserved			FFLUSH

19.9.9.1 FFLUSH

When the FFLUSH bit is asserted (‘1’), data in the Follower Interface FIFO is cleared. When set, the FFLUSH bit is automatically cleared by hardware within one register access cycle.

19.9.9.2 FFF_EN

Enable bit for the FFF interrupt (see Section 19.9.7.5, "FFF," on page 348).

19.9.9.3 FDR_EN

Enable bit for the FDR interrupt (see Section 19.9.7.6, "FDR," on page 348).

19.9.9.4 FFDONE_EN

Enable bit for the FFDONE interrupt (see Section 19.9.7.7, "FFDONE," on page 348).

19.9.9.5 Reserved

Reserved bits cannot be written and return ‘0’ when read.

19.9.10 MCHP RESERVED REGISTER

This register is reserved for use by Microchip. Writes to this register may produce undesirable results.

© 2012-2018 Microchip Technology Inc.

20.0 16-BIT TIMER INTERFACE

20.1 General Description

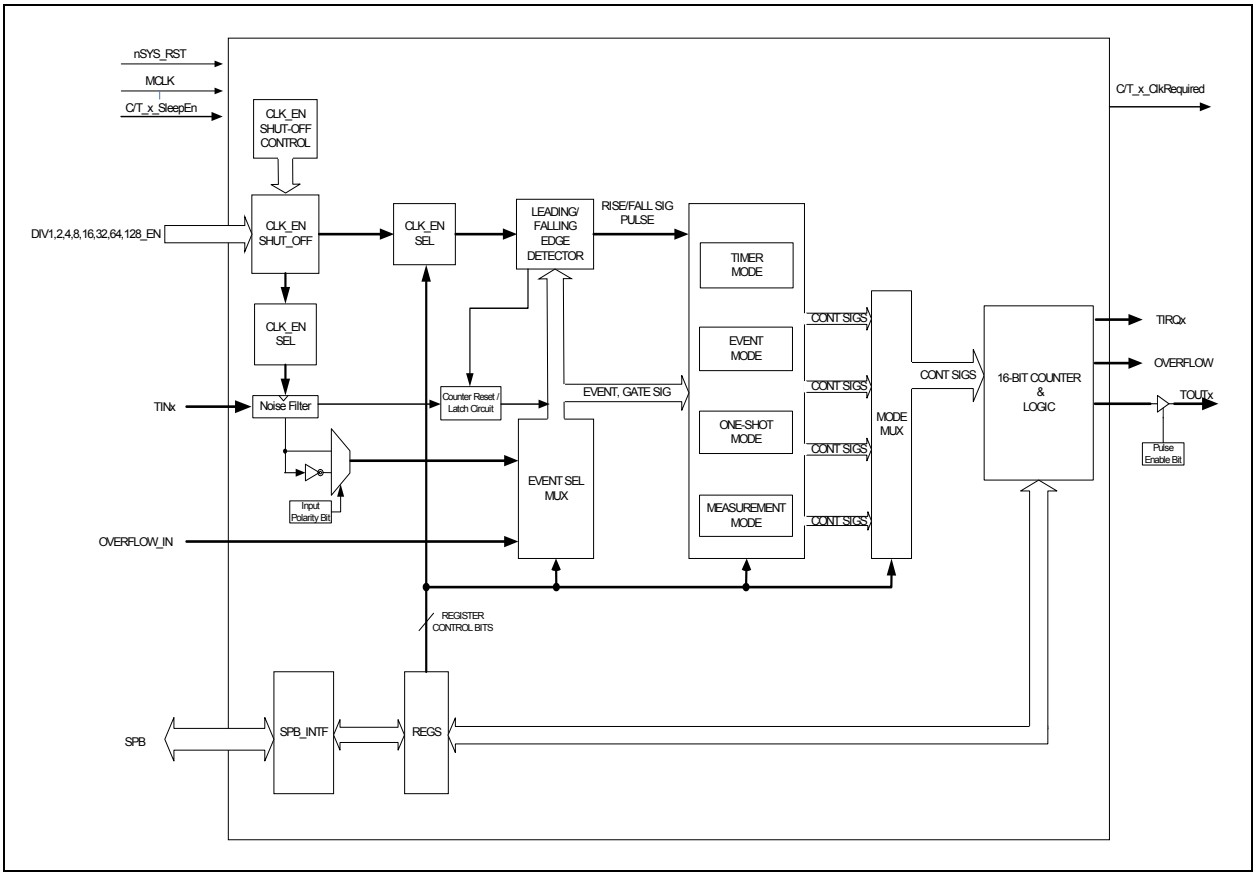
The MEC1632 16-Bit Timer Interface implements four 16-bit auto-reloading timer/counters. The clock for each timer/counter is derived from the system clock and can be divided down by a prescaler. Input-Only and Input/Output timers can also use an external input pin to clock or gate the counter. To aid operation in noisy environments the external input pin also has a selectable noise filter. If large counts are required, the output of each timer/counter can be internally connected to the next timer/counter.

The following section defines terms used in this chapter.

Term	Definition
Overflow	When the timer counter transitions from FFFFh to 0000h
Underflow	When the timer counter transitions from 0000h to FFFFh.
Timer Tick Rate	This is the rate at which the timer is incremented or decremented.

20.2 Block Diagram

FIGURE 20-1: BLOCK DIAGRAM FOR TIMER X



20.3 Signal List for Block Diagram

TABLE 20-1: BLOCK DIAGRAM SIGNAL LIST DESCRIPTION

Signal Name	Direction	Description
VTR POR	INPUT	nSYS_RST
MCLK	INPUT	20.27 MHz clock source to block.
DIV1,2,4,8,16,32,64,128_EN	INPUT	Clock Enables for supporting Filter and Timer frequencies.
TINx	INPUT	Timer x Input signal
TIRQx	OUTPUT	Timer x Interrupt Request
C/T_x_SleepEn, x=0-3	INPUT	Sleep Enable signals to counters 1-4
C/T_x_ClkRequired, x=0-3	INPUT	Clock required signals from counters 1-4
TOUTx	OUTPUT	Timer x Output signal
SPB_IF	I/O Bus	Bus used by microprocessor to access the registers in this block.

20.4 Timer Connections

For external inputs/outputs (TINx/TOUTx) to/from timers, please see Pin Configuration chapter for a description of the 16-bit Counter/Timer Interface.

TABLE 20-2: TIMER CASCADING DESCRIPTION

Timer Name	Timer Type	Over-Flow/ Under-flow Input's Connection
Timer 0	General Purpose	from Timer 3
Timer 1	General Purpose	from Timer 0
Timer 2	General Purpose	from Timer 1
Timer 3	General Purpose	from Timer 2

Note: The cascading connections are independent of the TINx/TOUTx connections.

20.5 Power, Clocks and Reset

20.5.1 POWER DOMAIN

This block is powered by the VTR power supply.

20.5.2 CLOCKS

There is a clock enable input for each of the supported frequencies listed in Table 20-12, "Timer Clock Frequencies," on page 364. Any of these enables may be selected for the Timer Clock Frequency. Independently, any of these clock frequencies may be selected for the filter clock via the FCLK[3:0] bits located in Section 20.11.2, "Timer x Clock and Event Control Register," on page 363.

The Event input is synchronized to FCLK and (if enabled) filtered by a three stage filter. The resulting recreated clock is used to clock the timer in Event mode. In Bypass Mode (Sync Only), the pulse width of the external signal must be at least 2x the pulse width of the FCLK source. If the Event input not in Bypass Mode (Sync and Filter), the pulse width of the external signal must be at least 4x the pulse width of the sync and filter clock.

20.5.3 RESET

This block is reset on a nSYS_RST. On nSYS_RST all timers are reset to their default values. The timers are also reset by the RESET bit in each Timer x Control Register.

See Section 7.6, "Reset Interface," on page 124 for details on reset.

© 2012-2018 Microchip Technology Inc.

20.6 Interrupts

The timers in the MEC1632 can be used to generate interrupts when the timer overflows or underflows. The timer interrupts are routed to the TIMER0, TIMER1, TIMER2, and TIMER3 bits in GIRQ15 Source Register.

Note: No interrupts are generated while the ENABLE bit is cleared.

20.7 Low Power Modes

This block is designed to conserve power when it is either sleeping or a clock source is not required.

During normal operation, if the timer is disabled via the PD bit the TIMERx_CLK_REQ signal is de-asserted. This indicates to the clock generator logic that this timer does not require the MCLK clock source.

During Sleep modes the clock input is gated, the TIMERx_CLK_REQ signal is asserted, and the interrupt output goes to the inactive state. When the block returns from sleep, if enabled, it will be restarted from the preload value.

Note: The timer is terminated one TCLK after the SLEEP ENABLE is asserted.

The following table illustrates the low power mode options.

TABLE 20-3: BLOCK CLOCK GATING IN LOW POWER MODES

Power Down (PD) Bit	SLEEP ENABLE	Block Idle Status	TIMERX_CLK_REQ	State	Description
1	X	NOT IDLE	1	PREPARING to SLEEP	The core clock is still required for up to one Timer Clock period.
		IDLE	0	SLEEPING	The block is idle and the core clock can be stopped.
0	0	X	1	NORMAL OPERATION	The block in neither disabled by firmware nor commanded to SLEEP
	1	NOT IDLE	1	PREPARING to SLEEP	The core clock is still required for up to one Timer Clock period.
		IDLE	0	SLEEPING	The block is idle and the core clock can be stopped.

20.8 Noise Filter

The noise filter uses Filter Clock (FCLK) to filter the signal on the TINx pins. The external pin must remain in the same state for three FCLK ticks before the internal state changes. The Filter Bypass: bit in the Timer x Control Register is used to bypass the noise filter.

- The signal TIN may be optionally only synchronized, or synchronized and filtered depending on the filter bypass bit
- The minimum FCLK period must be at least 2X the duration of the TIN signal so that signal can be reliably captured in the bypass mode
- The minimum FCLK period must be at least 4X the duration of the TIN signal so that signal can be reliably captured in the non-bypass mode
- In One-Shot mode, the TIN duration could be smaller than a TCLK period. The filtered signal is latched until the signal is seen in the TCLK domain. This also applies in the filter bypass mode

© 2012-2018 Microchip Technology Inc.

20.9 Operating Modes

20.9.1 STARTING AND STOPPING

The MEC1632 timers can be started and stopped by setting and clearing the Timer Enable bit in the Timer Control Register in all modes, except one-shot.

20.9.2 TIMER MODE

The Timer mode of the MEC1632 is used to generate periodic interrupts to the EC. When operating in this mode the timer always counts down based on one of the internally generated clock sources. The Timer mode is selected by setting the Timer Mode Select bits in the Timer Control Register. See Section 20.11.1, "Timer x Control Register," on page 361.

The period between timer interrupts and the width of the output pulse is determined by the speed of the clock source, the clock divide ratio and the value programmed into the Timer Reload Register. The timer clock source and clock rate are selected using the Clock Source Select bits (TCLK) in the Timer x Clock and Event Control Register. See Section 20.11.2, "Timer x Clock and Event Control Register," on page 363.

TABLE 20-4: TIMER MODE OPERATIONAL SUMMARY

Item	Description
Timer Clock Frequencies	This mode supports all the programmable frequencies listed in Table 20-12, "Timer Clock Frequencies"
Filter Clock Frequencies	This mode supports all the programmable frequencies listed in Table 20-12, "Timer Clock Frequencies"
Count Operation	Down Counter
Reload Operation	When the timer underflows: RLOAD = 1, timer reloads from Timer Reload Reg RLOAD = 0, timer rolls over to FFFFh.
Count Start Condition	UPDN = 0 (timer only mode): ENABLE = 1 UPDN = 1 (timer gate mode): ENABLE = 1 & TIN = 1;
Count Stop Condition	UPDN = 0: ENABLE = 0; UPDN = 1: (ENABLE= 0 | TIN = 0)
Interrupt Request Generation Timing	When timer underflows from 0000h to reload value (as determined by RLOAD) an interrupt is generated.
TINx Pin Function	Provides timer gate function
TOUTx Pin Function	TOUT toggles each time the timer underflows (if enabled).
Read From Timer	Current count value can be read by reading the Timer Count Register
Write to Preload Register	After the firmware writes to the Timer Reload Register asserting the RESET loads the timer with the new value programmed in the Timer Reload Register. Note: If the firmware does not assert RESET, the timer will automatically load the Timer Reload Register value when the timer underflows. When the timer is running, values written to the Timer Reload Register are written to the timer counter when the timer underflows. The assertion of Reset also copies the Timer Reload Register into the timer counter.
Selectable Functions	• Reload timer on underflow with programmed Preload value (Basic Timer) • Reload timer with FFFFh in Free Running Mode (Free-running Timer) • Timer can be started and stopped by the TINx input pin (Gate Function) • The TOUTx pin changes polarity each time the timer underflows (Pulse Output Function)

© 2012-2018 Microchip Technology Inc.

20.9.2.1 Timer Mode Underflow

The MEC1632 timers operating in Timer mode can underflow in two different ways. One method, the Reload mode shown in Figure 20-2, is to reload the value programmed into the Reload register and continue counting from this value. The second method, Free Running mode Figure 20-3, is to set the timer to FFFFh and continue counting from this value. The underflow behavior is controlled by the RLOAD bit in the Timer Control Register.

FIGURE 20-2: RELOAD MODE BEHAVIOR

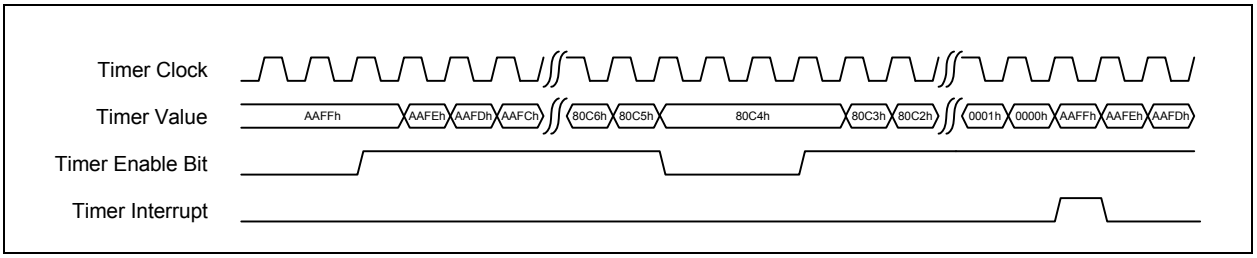
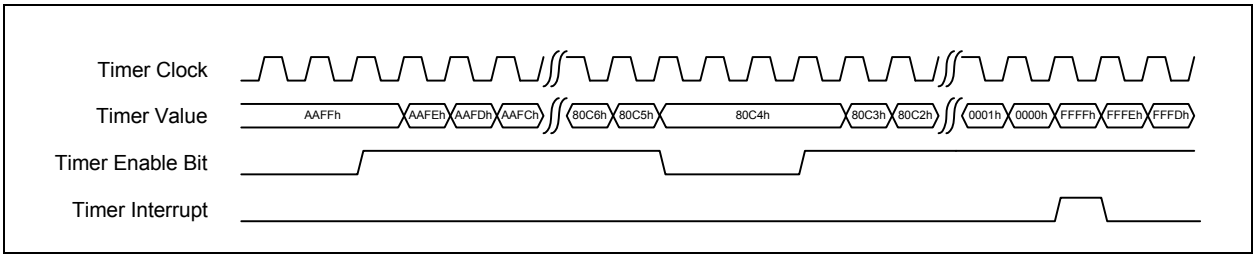


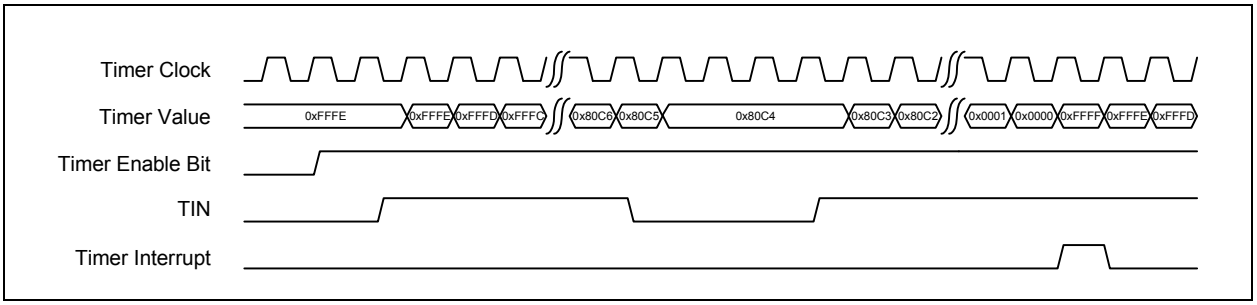
FIGURE 20-3: FREE RUNNING MODE BEHAVIOR



20.9.2.2 Timer Gate Function

The TIN pin on each timer can be used to pause the timer's operation when the timer is running. The timer will stop counting when the TIN pin is deasserted and count when the TIN pin is asserted. Figure 20-4 shows the timer behavior when the TIN pin is used to gate the timer function. The UPDN bit is used to enable and disable the Timer Gate function when in the Timer mode.

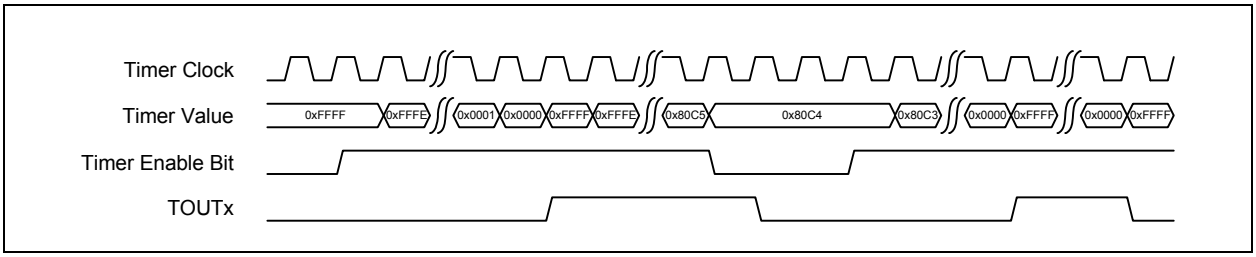
FIGURE 20-4: TIMER GATE OPERATION



20.9.2.3 Timer Mode Pulse Output

The four Timers can be used to generate a periodic output pulse. The output pulse changes state each time the timer underflows. The output is also cleared when the EN bit is cleared. Figure 20-5 shows the behavior of the TOUTx pin when it is used as a pulse output pin.

FIGURE 20-5: TIMER PULSE OUTPUT



20.9.3 EVENT MODE

Event mode is used to count events that occur external to the timer. The timer can be programmed to count the overflow output from the previous timer or an edge on the TIN pin. The direction the timer counts in Event mode is controlled by the UPDN bit in the Timer Control Register. When the timer is in Event mode, the TOUTx signal can be used to generate a periodic output pulse when the timer overflows or underflows. Figure 20-5 illustrates the pulse output behavior of the TOUTx pin in event mode when the timer underflows.

The timer can be programmed using the Clock and Event Control register to respond to the following events using the EVENT bits and the EDGE bits: rising edge of TINx, falling edge of TINx, rising and falling edge of TINx, rising edge of overflow input, falling edge of the overflow input, and the rising and falling edges of the overflow input.

TABLE 20-5: EVENT MODE OPERATIONAL SUMMARY

Item	Description
Count Source	• External signal input to TINx pin (effective edge can be selected by software) • Timer x-1 overflow
Timer Clock Frequencies	This mode supports all the programmable frequencies listed in Table 20-12, "Timer Clock Frequencies"
Filter Clock Frequencies	This mode supports all the programmable frequencies listed in Table 20-12, "Timer Clock Frequencies"
Count Operation	Up/Down Counter
Reload Operation	• When the timer underflows: RLOAD = 1, timer reloads from Timer Reload Reg RLOAD = 0, timer rolls over to FFFFh. • When the timer overflows: RLOAD = 1, timer reloads from Timer Reload Reg RLOAD = 0, timer rolls over to 0000h.
Count Start Condition	Timer Enable is set (ENABLE = 1)
Count Stop Condition	Timer Enable is cleared (ENABLE = 0)
Interrupt Request Generation Timing	When timer overflows or underflows
TINx Pin Function	Event Generation
TOUTx Pin Function	TOUT toggles each time the timer underflows/overflows (if enabled).
Read From Timer	Current count value can be read by reading the Timer Count Register

© 2012-2018 Microchip Technology Inc.

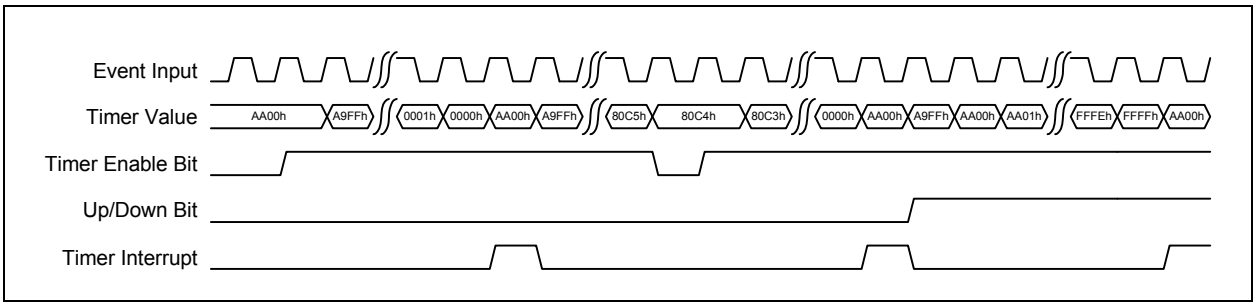
TABLE 20-5: EVENT MODE OPERATIONAL SUMMARY (CONTINUED)

Item	Description
Write to Preload Register	After the firmware writes to the Timer Reload Register, asserting the RESET loads the timer with the new value programmed in the Timer Reload Register. Note: If the firmware does not assert RESET, the timer will automatically load the Timer Reload Register value when the timer underflows.
Selectable Functions	• The direction of the counter is selectable via the UPDN bit. • Reload timer on underflow/overflow with programmed Preload value (Basic Timer) • Reload timer with FFFFh in Free Running Mode (Free-running Timer) • Pulse Output Function The TOUTx pin changes polarity each time the timer underflows or overflows.

20.9.3.1 Event Mode Operation

The timer starts counting events when the ENABLE bit in the Timer Control Register is set and continues to count until the ENABLE bit is cleared. When the ENABLE bit is set, the timer continues counting from the current value in the timer except after a reset event. After a reset event, the timer always starts counting from the value programmed in the Reload Register if counting down or from 0000h if counting up. Figure 20-6 shows an example of timer operation in Event mode. The RLOAD bit controls the behavior of the timer when it underflows or overflows.

FIGURE 20-6: EVENT MODE OPERATION



20.9.4 ONE-SHOT MODE

The One-Shot mode of the timer is used to generate a single interrupt to the EC after a specified amount of time. The timer can be configured to start using the ENABLE bit (Figure 20-7) or on a timer overflow event from the previous timer. See Section 20.11.2, "Timer x Clock and Event Control Register," on page 363 for configuration details. The ENABLE bit must be set for an event to start the timer. The ENABLE bit is cleared one clock after the timer starts. The timer always starts from the value in the Reload Register and counts down in One-Shot mode.

TABLE 20-6: ONE SHOT MODE OPERATIONAL SUMMARY

Item	Description
Timer Clock Frequencies	This mode supports all the programmable frequencies listed in Table 20-12, "Timer Clock Frequencies"
Filter Clock Frequencies	This mode supports all the programmable frequencies listed in Table 20-12, "Timer Clock Frequencies"
Count Operation	Down Counter
Reload Operation	When the timer underflows the timer will stop. When the timer is enabled timer starts counting from value programmed in Timer Reload Register. (RLOAD has no effect in this mode)
Count Start Condition	Setting the ENABLE bit to 1 starts One-Shot mode. The timer clock automatically clears the enable bit one timer tick later. **Note:** One-Shot mode may be enabled in Event Mode. In Event mode an overflow from the previous timer is used for timer tick rate.

TABLE 20-6: ONE SHOT MODE OPERATIONAL SUMMARY (CONTINUED)

Item	Description
Count Stop Condition	• Timer is reset (RESET = 1) • Timer underflows
Interrupt Request Generation Timing	When an underflow occurs.
TINx Pin Function	One Shot External input
TOUTx Pin Function	The TOUTx pin is asserted when the timer starts and de-asserted when the timer stops
Read From Timer	Current count value can be read by reading the Timer Count Register
Write to Preload Register	After the firmware writes to the Timer Reload Register, asserting the RESET loads the timer with the new value programmed in the Timer Reload Register. Note: If the firmware does not assert RESET, the timer will automatically load the Timer Reload Register value when the timer underflows.
Selectable Functions	• Pulse Output Function The TOUTx pin is asserted when the timer starts and de-asserted when the timer stops.

FIGURE 20-7: TIMER START BASED ON ENABLE BIT

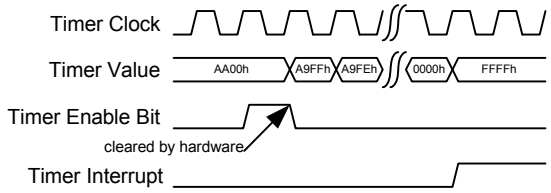
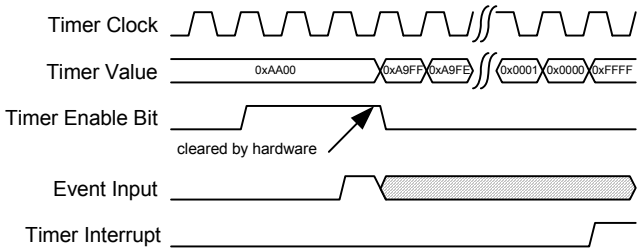
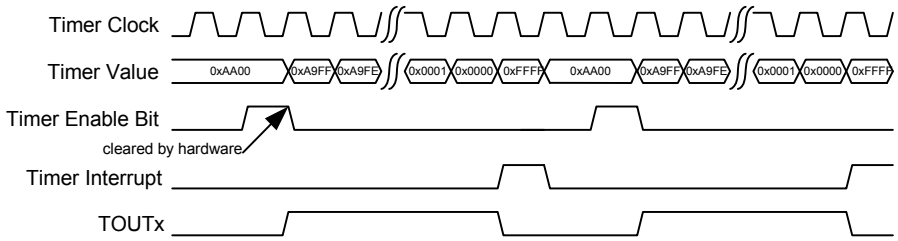


FIGURE 20-8: TIMER START BASED ON EXTERNAL EVENT



© 2012-2018 Microchip Technology Inc.

FIGURE 20-9: ONE SHOT TIMER WITH PULSE OUTPUT



20.9.5 MEASUREMENT MODE

The Measurement mode is used to measure the pulse width or period of an external signal. An interrupt to the EC is generated after each measurement or if the timer overflows and no measurement occurred. The timer measures the pulse width or period by counting the number of clock between edges on the TINx pin. The timer always stars counting at zero and counts up to 0xFFFF. The accuracy of the measurement depends on the speed of the clock being used. The speed of the clock also determines the maximum pulse width or period that can be detected.

TABLE 20-7: MEASUREMENT MODE OPERATIONAL SUMMARY

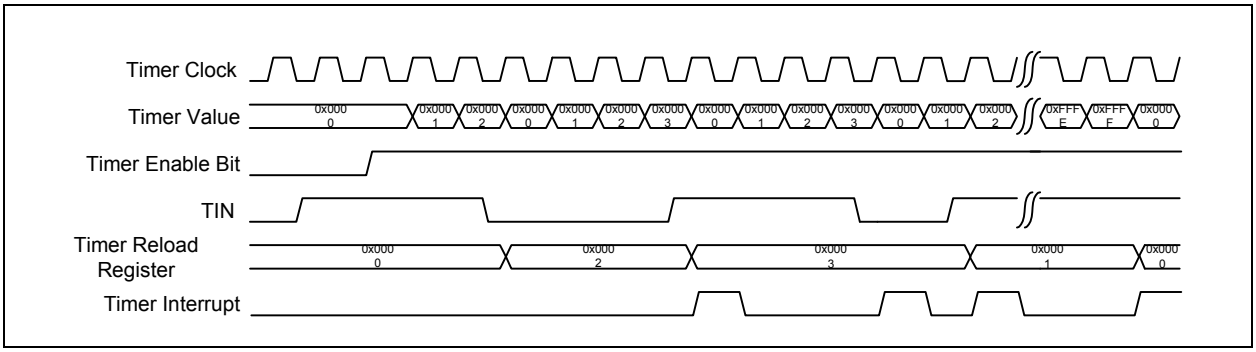
Item	Description
Timer Clock Frequencies	This mode supports all the programmable frequencies listed in Table 20-12, "Timer Clock Frequencies"
Filter Clock Frequencies	This mode supports all the programmable frequencies listed in Table 20-12, "Timer Clock Frequencies"
Count Operation	• Up Count • At measurement pulse's effective edge, the count value is transferred to the Timer Reload Register and the timer is loaded with 0000h and continues counting.
Count Start Condition	• Timer enable is set (ENABLE = 1)
Count Stop Condition	• Timer is reset (RESET = 1) • Timer overflows • Timer enable is cleared (ENABLE = 0)
Interrupt Request Generation Timing	• When timer overflows • When a measurement pulse’s effective edge is input. (An interrupt is not generated on the first effective edge after the timer is started.)
TINx Pin Function	Programmable Input port or Measurement input
Read From Timer	When the Timer x Reload Register is read it indicates the measurement result from the last measurement made. The Timer x Reload Register reads 0000h if the timer overflows before a measurement is made.
Write to Timer	Timer x Reload Register is Read-Only in Measurement mode

20.9.5.1 Pulse Width Measurements

The timers measure pulse width by counting the number of timer clocks since the last rising or falling edge of the TINx input. To measure the pulse width of a signal on the TINx pin, the EDGE bits in the Clock and Event Control Register, must be set to start counting on rising and falling edges. The timer starts measuring on the next edge (rising or falling) on the TINx pin after the ENABLE bit is set. The Reload register stores the result of the last measurement taken. If the timer overflows, 0x0000 is written to the Reload register and the ENABLE bit is cleared stopping the timer. Figure 20-10 shows the timer behavior when measuring pulse widths.

The timer will not assert an interrupt in Pulse Measurement mode until the timer detects both a rising and a falling edge.

FIGURE 20-10: PULSE WIDTH MEASUREMENT

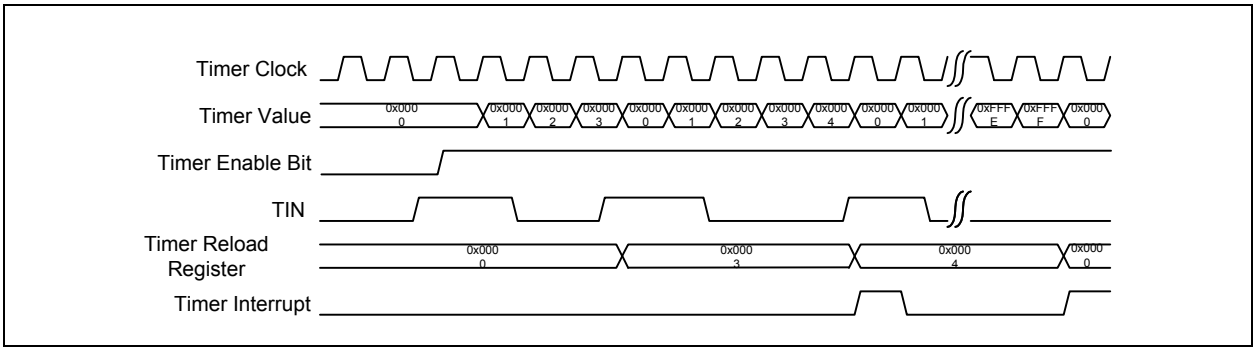


20.9.5.2 Period Measurements

The timers in the MEC1632 measure the period of a signal by counting the number of timer clocks between either rising or falling edges of the TINx input. The measurement edge is determined by the EDGE bits in the Clock and Event Control Register. The timer starts measuring on the next edge (rising or falling) on the TINx pin after the ENABLE bit is set. The reload register stores the result of the last measurement taken. If the timer overflows, 0x0000 is written to the reload register. Figure 20-11 shows the timer behavior when measuring the period of a signal.

The timer will not signal an interrupt in period measurement mode until the timer detects either two rising edges or two falling edges.

FIGURE 20-11: PULSE PERIOD MEASUREMENT



20.10 16-Bit Counter/Timer Interface Register Summary

There are four instances of the 16-Bit Timer Interface block implemented in the MEC1632 enumerated as [0:3] with an overflow/underflow interface. Each instance of the 16-Bit Timer Interface has its Base Address as indicated in Table 20-8.

TABLE 20-8: 16-BIT COUNTER/TIMER INTERFACE BASE ADDRESS TABLE

16-Bit Timer Interface **Instance**	**LDN from** (Table 4-2 on page 59)	**AHB Base Address**
16-bit Timer.**0**	3h	F0_0C00h
16-bit Timer.**1**		F0_0C80h = F0_0C00h + 80h
16-bit Timer.**2**		F0_0D00h = F0_0C00h + 100h
16-bit Timer.**3**		F0_0D80h = F0_0C00h + 180h

Table 20-9 is a register summary for one instance of the 16-Bit Timer Interface.

© 2012-2018 Microchip Technology Inc.

TABLE 20-9: 16-BIT COUNTER/TIMER INTERFACE REGISTER SUMMARY

	EC Interface		
Register Name	**SPB Offset**	**EC Type**	**Notes**
Timer x Control Register	00h	R/W	
Timer x Clock and Event Control Register	04h	R/W	
Timer x Reload Register	08h	R/W	
Timer x Count Register	0Ch	R	

20.11 Detailed Register Descriptions

20.11.1 TIMER X CONTROL REGISTER

TABLE 20-10: TIMER X CONTROL REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	00h			16-bit			**EC SIZE**	
POWER	VTR			0200h			**VTR POR DEFAULT**	
	EC SPB							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R/W	R/W	R/W
BIT NAME	Reserved			TIMERx_CLK_REQ	SLEEP_ENABLE	TOUT Polarity	PD	Filter Bypass
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	RLOAD	TOUT_EN	UPDN	INPOL	MODE		RESET	ENABLE

ENABLE

Timer Enable - This bit is used to start and stop the timer. This bit does not reset the timer count but does reset the timer pulse output. This bit will be cleared when the timer starts counting in One-Shot mode.

0=Timer is disabled

1=Timer is enabled

Note:	The ENABLE bit is cleared after the RESET cycle is done. Firmware must poll the RESET bit.

RESET

Timer Reset - This bit stops the timer and resets the internal counter to the value in the Timer Reload Register. This bit also clears the Timer Enable bit if it is set. This bit is self clearing after the timer is reset. Firmware must poll this RESET bit.

0=Normal timer operation

1=Timer reset

APPLICATION NOTE: When the RESET takes effect interrupts are blocked. Interrupts are not blocked until RESET takes effect and the ENABLE bit is cleared. If interrupts are not desired, firmware must mask interrupt in the interrupt block.

MODE

Timer Mode Select - These bits control the timer mode.

00=Timer Mode

01=Event Mode

10=One Shot Mode

11=Measurement Mode

INPOL

Timer Input Polarity. This bit selects the polarity of the TINx input

0=TINx input is active low (inverted)

1=TINx input is active high (non-inverted)

UPDN

Up/Down. In Event mode this bit selects the timer count direction.

Event Mode:

0=The timer counts down

1=The timer counts up

Timer Mode:

0=TINx pin has no effect on the timer

1=TINx pin pauses the timer when deasserted

TOUT_EN

TOUT Enable

0=TOUT pin is pin in the inactive state (driven low)

1=TOUT function is enabled

RLOAD

Reload Control:

This bit controls how the timer is reloaded on overflow or underflow in Event and Timer modes, it has no effect in One Shot mode.

0=Roll timer over to FFFFh and continue counting when counting down and rolls over to 0000h and continues counting when counting up.

1=Reload timer from Timer Reload Register and continue counting.

Filter Bypass:

Filter Bypass permits TINx to bypass the noise filter and go directly into the timer

0=Filter enabled on TINx (default)

1=Filter bypassed on TINx

PD

Power Down:

0=The timer is in a running state.

1=The timer is powered down and all clocks are gated (default).

© 2012-2018 Microchip Technology Inc.

TOUT POLARITY

This bit determines the polarity of the TOUT signal. In timer modes that toggle the TOUT signal, this polarity bit will not have a perceivable difference, except to determine the inactive state. in One-Shot mode this determines if the pulsed output is active high or active low.

0=Active high (default)

1=Active low

SLEEP ENABLE

This bit is a read-only bit that reflects the state of the SLEEP ENABLE signal. This signal stops the timer and resets the internal counter to the value in the Timer Reload Register. Once the timer is disabled, the TIMERX_CLK_REQ bits will be deasserted. This signal does not clear the Timer Enable bit if it is set. If the timer is enabled, the counter will resume operation when the SLEEP ENABLE signal is deasserted. The timer is held in reset as long as the input signal is asserted.

0=Normal timer operation. In Normal Mode, the timer operates as configured. When returning from a sleep mode, if enabled, the counter will be restarted from the preload value.

1=Sleep Mode Requested. In Sleep Mode, the timer is reset, the counter is disabled, and the TIMERX_CLK_REQ outputs are deasserted.

TIMERX_CLK_REQ

The TIMERX_CLK_REQ bit is a read-only bit that reflects the state of the TIMERX_CLK_REQ output signal.

0=Indicates the MCLK clock domain can be turned 'off' when appropriate

1=Indicates the MCLK clock domain is required to be 'on.'

20.11.2 TIMER X CLOCK AND EVENT CONTROL REGISTER

TABLE 20-11: TIMER X CLOCK AND EVENT CONTROL REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	04h			16-bit			**EC SIZE**	
POWER	VTR			0000h			**VTR POR DEFAULT**	
	EC SPB							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R/W	R/W	R/W	R/W
BIT NAME	Reserved				FCLK			
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
BIT NAME	EVENT	EDGE		Reserved	TCLK			

TCLK

This field is the Timer Clock Select, used to determine the clock source to the 16-bit timer. Available frequencies are shown in Table 20-12:

TABLE 20-12: TIMER CLOCK FREQUENCIES

Timer Clock Select	Frequency Divide Select	Frequency Selected
0000	Divide by 1	20.27MHz
0001	Divide by 2	10.13Mhz
0010	Divide by 4	5.07MHz
0011	Divide by 8	2.53Mhz
0100	Divide by 16	1.27MHz
0101	Divide by 32	633KHz
0110	Divide by 64	317KHz
0111	Divide by 128	158KHz
1xxx	Reserved	Reserved

EDGE

Edge Type Select. These bits are used to select the edge type that the timer counts. In One-Shot mode these bits select which edge starts the timer.

Event Mode:

00=Counts falling edges

01=Counts rising edges

10=Counts rising and falling edges

11=No event selected

One-Shot Mode:

00=Starts counting on a falling edge

01=Starts counting on a rising edge

10=Starts counting on a rising or falling edge

11=Start counting when the Enable bit is set

Measurement Mode:

00=Measures the time between falling edges

01=Measures the time between rising edges

10=Measures the time between rising edges and falling edges and the time between falling edges and rising edges

11=No event selected

EVENT

Event Select:

This bit is used to select the count source when the timer is operating in event mode.

0=Timer x-1 overflow is count source

1=TINx is count source

FCLK

This field is the Filter Clock Select, used to determine the clock source for the TINx noise filter. Available frequencies are the same as the Timer clock and are shown in Table 20-12, "Timer Clock Frequencies".

© 2012-2018 Microchip Technology Inc.

20.11.3 TIMER X RELOAD REGISTER

TABLE 20-13: TIMER X RELOAD REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	08h			16-bit			**EC SIZE**	
POWER	VTR			FFFFh			**VTR POR DEFAULT**	
	EC SPB							
BIT	**D15**	**D14**	**D13**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Timer Reload [15:0]							

TIMER RELOAD

The Timer Reload register is used in Timer and One-Shot modes to set the lower limit of the timer. In Event mode the Timer Reload register sets either the upper or lower limit of the timer depending on if the timer is counting up or down. Valid Timer Reload values are 0001h - FFFFh. If the timer is running, the reload value will not be updated until the timer overflows or underflows.

Note:	Programming a 0000h as a preload value is not a valid count value.

20.11.4 TIMER X COUNT REGISTER

TABLE 20-14: TIMER X COUNT REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	0Ch			16-bit			**EC SIZE**	
POWER	VTR			FFFFh			**VTR POR DEFAULT**	
	EC SPB							
BIT	**D15**	**D14**	**D13**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Timer Count [15:0]							

TIMER COUNT

The Timer Count register returns the current value of the timer in all modes.

21.0 HIBERNATION TIMER

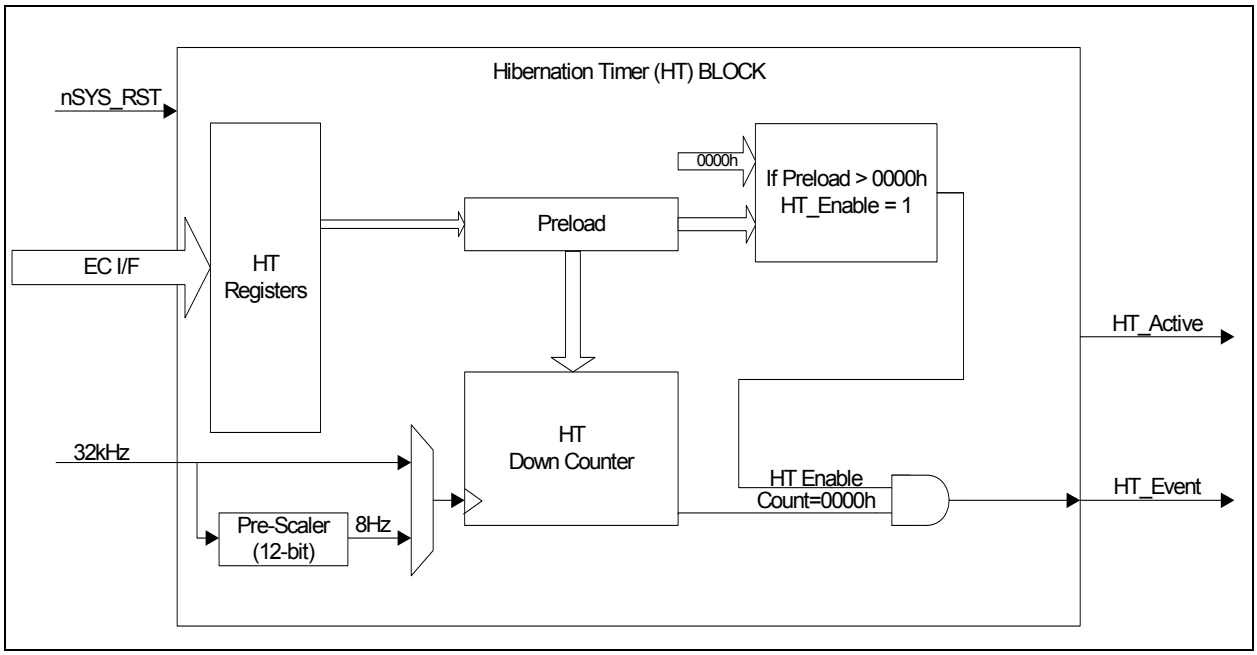
21.1 General Description

The Hibernation Timer can generate a wake event to the Embedded Controller (EC) when it is in a hibernation mode. This block supports wake events up to 2 hours in duration. The timer is a 16-bit binary count-down timer that can be programmed in 30.5us and 0.125 second increments for period ranges of 30.5us to 2s or 0.125s to 136.5 minutes, respectively. Writing a non-zero value to this register starts the counter from that value. A wake-up interrupt is generated when the count reaches zero.

See GIRQ23 Source Register on page 324 for details on enabling the Hibernation Timer wake-up event.

21.2 Block Diagram

FIGURE 21-1: HIBERNATION TIMER BLOCK DIAGRAM



21.3 Block Diagram Signal List

TABLE 21-1: BLOCK DIAGRAM SIGNAL LIST DESCRIPTION

Signal Name	Direction	Description
nSYS_RST	INPUT	VTR Power on Reset.
X32K_CLK	INPUT	32Khz, Clock Source for Hibernation Timer
HT_Active	OUTPUT	Signal indicating that the timer is enabled and actively counting
HT_Event	OUTPUT	Signal indicating that the timer is enabled and has expired. This signal is used to generate an Hibernation Timer interrupt event.
E/C IF	I/O Bus	Bus used by microprocessor to access the registers in this block.
SLEEP_EN	INPUT	Sleep Enable input from the Block Sleep Enable Registers. See also Section 21.7, "Sleep Interface," on page 369.
CLOCK_REQ	OUTPUT	Clock Required output to the Clock Required Status Registers. See also Section 21.7, "Sleep Interface," on page 369.

© 2012-2018 Microchip Technology Inc.

21.4 Power, Clocks and Reset

21.4.1 POWER DOMAIN

This block is powered by the VTR Power Supply.

See Section 7.5, "Power Configuration," on page 121 for details on power domains.

21.4.2 CLOCKS

This block has two clock inputs, the EC Bus Clock, and X32K_CLK. The EC Bus Clock is used in the interface to the embedded controller accessible registers. The 32.768KHz X32K_CLK is the clock source for the Hibernation Timer functional logic, including the counters.

See Section 7.4, "Clock Generator," on page 98 for details on clocks.

21.4.3 RESET

This block is reset on a nSYS_RST.

See Section 7.6, "Reset Interface," on page 124 for details on reset.

21.4.4 POWER MANAGEMENT

In all cases, X32K_CLK does not affect Power Management; i.e., the Hibernation Timer operates normally when MCLK is stopped. The sleep enable inputs have no affect on the Hibernation Timer and the clock required outputs are only asserted during register read/write cycles for as long as necessary to propagate updates to the block core (Table 21-2).

TABLE 21-2: HIBERNATION TIMER POWER MANAGEMENT

SLEEP_EN	Bus Access Cycle?	CLK_REQ	Description
X	Yes	1	CLK_REQ is only asserted for as long as necessary to propagate updates to the block core
	No	0	CLK_REQ is not asserted when the EC is not accessing the register interface. (Note that this block *cannot* prevent the chip from entering the system deepest sleep states.)

21.5 Interrupts

Each instance of the Hibernation Timer in the MEC1632 can be used to generate interrupts and wake-up events when the timer value transitions from ‘1’ to ‘0’. The Hibernation Timer interrupts are routed to the HTIMER1 & HTIMER0 bits in GIRQ23 Source Register on page 324.

21.6 Registers

There are two instances of Hibernation Timer block implemented in the MEC1632 enumerated as.Hibernation Timer.0 & Hibernation Timer.1. Each instance of the Hibernation Timer has its own Logical Device Number, and Base Address as indicated in Table 21-3.

TABLE 21-3: Hibernation Timer BASE ADDRESS TABLE

Hibernation Timer **Instance**	**LDN from (Table 4-2 on page 59)**	**AHB Base Address**
Hibernation Timer**.0**	0h	F0_0000h
Hibernation Timer**.1**		F0_0000h + 80h

The Table 21-4 is a register summary for one instance of the Hibernation Timer. Each EC address is indicated as an SPB Offset from its AHB base address.

TABLE 21-4: Hibernation Timer REGISTER SUMMARY

Register Name	EC Interface			
	SPB Offset	Byte Lane	EC Type	Notes
HTimer x Preload Register	00h	1-0	R/W	
Hibernation Timer x Control Register	04h	0	R/W	
Hibernation Timer x Count Register	08h	1-0	R	

21.6.1 HTIMER X PRELOAD REGISTER

TABLE 21-5: HTIMER X PRELOAD REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	00h			16-bit			**EC SIZE**	
POWER	VTR			0000h			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D15**	**D14**	**D13**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	HT Preload[15:0]							

HT PRELOAD[15:0]

This register is used to set the Hibernation Timer Preload value. Writing this register to a non-zero value resets the down counter to start counting down from this programmed value. Writing this register to 0000h disables the hibernation counter. The resolution of this timer is determined by the CTRL bit in the HTimer x Control Register.

Writes to the HTimer x Control Register are completed with an EC SPB bus cycle.

21.6.2 HTIMER X CONTROL REGISTER

TABLE 21-6: HIBERNATION TIMER X CONTROL REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	04h			16-bit			**EC SIZE**	
POWER	VTR			0000h			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D15**	**D14**	**D13**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R							R/W
BIT NAME	Reserved							CTRL

CTRL

0= The Hibernation Timer has a resolution of 30.5us per LSB, which yields a maximum time of ~2seconds.

1= The Hibernation Timer has a resolution of 0.125s per LSB, which yields a maximum time in excess of 2 hours.

© 2012-2018 Microchip Technology Inc.

21.6.3 HTIMER X COUNT REGISTER

TABLE 21-7: HIBERNATION TIMER X COUNT REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	08h			16-bit			**EC SIZE**	
POWER	VTR			0000h			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BIT	**D15**	**D14**	**D13**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R							
BIT NAME	Count[15:0]							

COUNT[15:0]

The current state of the Hibernation Timer.

21.7 Sleep Interface

The Hibernation Timer CLOCK_REQ output is only asserted during a register read/write cycle. The Hibernation Timer 32kHz clock requirement does not influence the CLOCK_REQ output.

22.0 WEEK ALARM INTERFACE

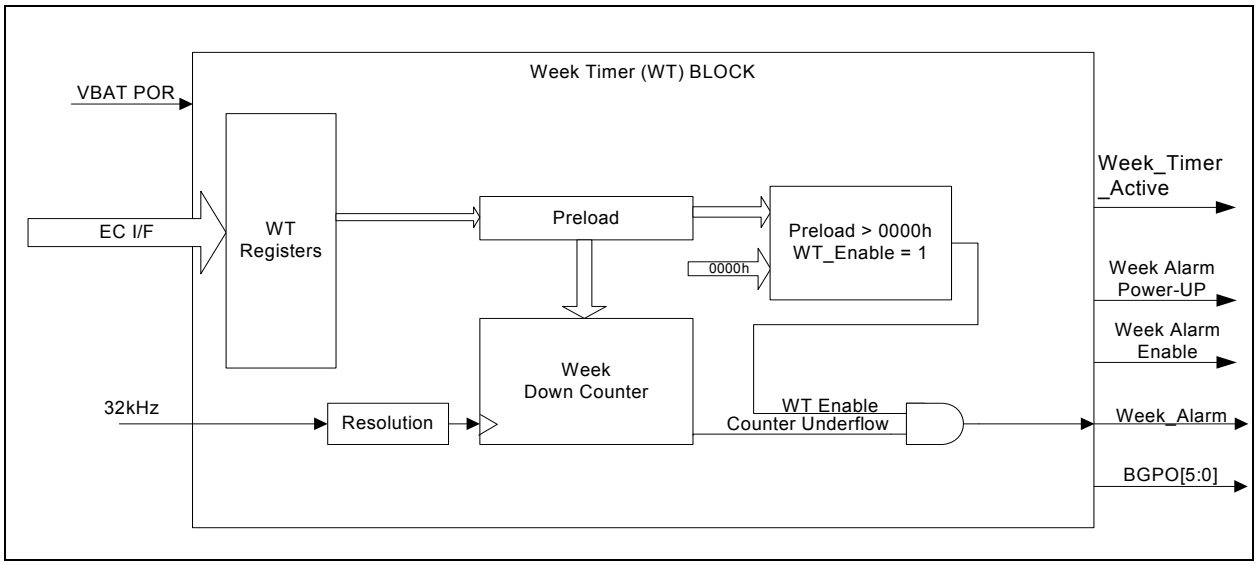
22.1 General Description

The Week Alarm Interface provides a 16-bit, battery-powered, Week Timer that supports 1 ms, 1 second and 1 minute resolution and auto reloads following a counter underflow. In addition, this block interfaces directly with the VBAT-Powered Control Interface and includes a VBAT-backed general-purpose output pin (BGPO[5:0]).

It takes up to two X32K_CLK clock period for registers to get updated after register writes. The 20 MHz Oscillator must not be stopped for at least one X32K_CLK clock period following a register write.

22.2 Block Diagram

FIGURE 22-1: WEEK TIMER BLOCK DIAGRAM



22.3 Signal List for Block Diagram

TABLE 22-1: WEEK ALARM INTERFACE SIGNAL LIST

Signal Name	Direction	Description
VBAT_POR	Input	VTR Power on Reset.
EC Bus Clock	Input	Bus Clock (part of the EC Interface)
X32K_CLK	Input	Core logic clock
Week Alarm Power-Up Output	Output	Week Timer wake up event signal
Week_alrm_en	Output	Output to control the function of the Week Timer Output
Week_alarm	Output	Week Timer Interrupt indicating that the timer has expired.
Week_Timer_Active	Output	The Week_Timer_Active output is asserted when the counter is enabled and counting. It is cleared when the Week_Timer is disabled or not counting.
EC Interface	I/O Bus	Bus used by microprocessor to access the registers in this block.
BGPO[5:0]	Output	VBAT-powered General Purpose Outputs

© 2012-2018 Microchip Technology Inc.

22.4 Power, Clocks and Reset

22.4.1 POWER DOMAIN

This block is powered by the VBAT power supply.

See Section 7.1.1, "Power Configuration," on page 95 for details on power domains.

22.4.2 CLOCKS

This block has two clock inputs: EC Bus Clock and the X32K_CLK. EC Bus Clock is used by the EC Data Memory Bus to interface to the embedded controller accessible registers. The 32.768KHz X32K_CLK clock source is the clock source for the week alarm logic.

See Section 7.4, "Clock Generator," on page 98 for details on clocks.

22.4.3 POWER ON RESET

This block is reset on VBAT_POR.

See Section 7.6, "Reset Interface," on page 124 for details on reset.

22.5 Interrupts

The Week Alarm Interface generates an interrupt and wakeup event following a Week Timer underflow. The Interrupt is routed WEEK_ALR in GIRQ23 Source Register. The Week Timer interrupt is asserted when VTR is powered. In addition, the WEEK_ALR wake and interrupt event can be asserted when VBAT is powered and VTR is unpowered. The WEEK_ALR wake event and Interrupt event is retained during VBAT and is detected after the next VTR POR power sequence.

22.6 Week Timer

The Week_Timer_Active output is asserted when the counter is enabled and counting. It is cleared when the Week_-Timer is disabled or not counting.

22.7 Week Alarm Power-Up Output

The internal Week Alarm Power-Up Output signal drives an input to the VBAT-Powered Control Interface as described in Section 34.0, "VBAT-Powered Control Interface," on page 499. The Week Alarm Power-Up Output signal is driven even when the VTR supply is unpowered.

The WEEK_ALRM_EN bit in the Week Timer Control Register enables the Week Alarm Power-Up Output function in the VBAT-Powered Control Interface (see FIGURE 34-1: on page 499). Once the internal Week Alarm Power-Up Output signal drives the input to the VBAT-Powered Control Interface as a result of an interrupt, the WEEK_ALRM_EN bit must be cleared to reset the Week Alarm Interface.

APPLICATION NOTE: The WEEK_ALRM_EN bit defaults to '0,' disabling the ability to power the system up by the Week Alarm Interface. This is necessary to avoid an uninitialized Week Alarm Interface from causing unintended power-up events.

22.8 Registers

The Week Alarm Interface has its own Logical Device Number, and Base Address as indicated in Table 22-2. Table 22-3 is a register summary for the Week Alarm Interface block. See Note 4-1 on page 60.

TABLE 22-2: Week Alarm Interface BASE ADDRESS TABLE

Week Alarm Interface Blocks	LDN from (Table 4-2 on page 59)	AHB Base Address
Week Alarm Timer	33h	F0_CC80h

Table 22-3 is a register summary for the Week Alarm Interface block. Each EC address is indicated as an SPB Offset from its AHB base address.

TABLE 22-3: Week Alarm Interface REGISTER SUMMARY

	EC Interface			
Register Name	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
Week Timer Control Register	00h	0	R/W	
Week Timer Reload Register	04h	0:1	R/W	
Week Timer Data Register	08h	0:1	R	

22.8.1 WEEK TIMER CONTROL REGISTER

The Week Timer Control Register is used to configure the Week Timer.

TABLE 22-4: WEEK TIMER CONTROL REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	00h			8-bit			**EC SIZE**	
POWER	VBAT			01h			VBAT_POR **DEFAULT**	
BUS	EC SPB							
BYTE2 BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R/W				
BIT NAME	Reserved			VTR_CTRL[5:1]				
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R/W				
BIT NAME	Reserved			BGPO[5:1]				
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/W	R/W	R	R	R/W (Note 22-2)		R/W
BIT NAME	Reserved	WEEK_ALRM_EN	BGPO0 (Note 22-1)	Reserved		RESOLUTION		WT_ENABLE

WT_ENABLE

Week Timer Enable - This bit is used to start and stop the Week Timer. The Week timer is held when the timer is disabled and starts counting from the value in the Week Timer Reload register when enabled.

0 - Week Timer is disabled.

1 - Week Timer is enabled.

(See Note 22-2 on page 373.)

RESOLUTION

Week Timer Resolution - These bits are used to control the resolution of the Week Timer counter.

00 - 1 minute resolution.

01 - 1 Second resolution

10 - 1 Millisecond resolution.

© 2012-2018 Microchip Technology Inc.

BGPO[5:0]

VBAT-powered General Purpose Output Control that is used as part of the VBAT-Powered Control Interface. The following options apply to each bit independently:

0 - output low (default)

1 - output high

Note 22-1 BGPO0 is always VBAT powered, hence has no associated VTR_CTRL bit.

WEEK_ALRM_EN

VCI Week Alarm Enable- This bit controls the routing of the Week Alarm Power-Up Output output to VBAT-Powered Control Interface to assert the VCI signal. Once the internal Week_alrm_en signal drives the input to the VBAT-Powered Control Interface as a result of Interrupts, this bit must be cleared to reset the Week Alarm Interface.

0 - Disable routing (Default)

1 - Enable routing

Note:	the Week Timer Enable bit [D0] must be cleared (‘0’) when changing the Week Timer Resolution bits. For example to change the resolution of the Week Timer two writes to the Week Timer Control Register are required: the first write de-asserts the WT_ENABLE bit, the second write modifies the RESOLUTION bits and asserts the WT_ENABLE bit.

VTR_CTRL[5:1]

The VTR_CTRL[5:1] bits define whether the VBAT, or VTR supply is powering each of the BGPO pins according to the following table:

TABLE 22-5: VTR_CTRL[N] BIT DEFINITION

ANY VTR_CTRL Bit (VTR_CTRL[5:1])	VTR	Pin Power	Description
1	‘ON’/’OFF’	VBAT	Output driven according to the BGPO[5:1] bits.
0 (Default)	‘ON’	VTR	Output driven according to GPIO output control register.
	‘OFF’		Output pin is tristated.

22.8.2 WEEK TIMER RELOAD REGISTER

TABLE 22-6: WEEK TIMER RELOAD REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	04h			16-bit			**EC SIZE**	
POWER	VBAT			2760h			VBAT_POR **DEFAULT**	
BUS	EC SPB							
BIT	**D15**	**D14**	**D13**	**…**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W Note 22-2							
BIT NAME	WEEK TIMER RELOAD[15:0]							

WEEK TIMER RELOAD[15:0]

This register contains the value that is used to reload the Week Timer Data Register when the latter underflows. A Reload value of 0000h is equivalent to a reload value of FFFFh. In both cases the Week Timer will count 2^{16} times before triggering an interrupt.

Note 22-2 The EC must clear the Enable bit in the Week Timer Control Register to perform a write access to the Week Timer Reload Register.

Note 22-3 A write to the Week Timer Reload Register of 0000h will be treated as a full count (FFFFh +1) and start downcounting when the WT_ENABLE bit in the Week Timer Control Register is set to '1'.

22.8.3 WEEK TIMER DATA REGISTER

TABLE 22-7: WEEK TIMER DATA REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	08h			16-bit			**EC SIZE**	
POWER	VBAT			2760h			VBAT_POR **DEFAULT**	
BUS	EC SPB							
BIT	**D15**	**D14**	**D13**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R							
BIT NAME	WEEK TIMER Counter[15:0]							

WEEK TIMER COUNTER[15:0]

The current state of the Week Timer.

© 2012-2018 Microchip Technology Inc.

23.0 RTC WITH DATE AND DST ADJUSTMENT

23.1 Introduction

This block provides the capabilities of an industry-standard 146818B Real-Time Clock module, without CMOS RAM. Enhancements to this architecture include:

- Industry standard Day of Month Alarm field, allowing for monthly alarms
- Configurable, automatic Daylight Savings adjustment
- Week Alarm for periodic interrupts and wakes based on Day of Week
- System Wake capability on interrupts, including Wake from Deep Sleep by the Alarm.

23.2 References

Motorola 146818B Data Sheet, available on-line

Intel Lynx Point PCH EDS specification

23.3 Terminology

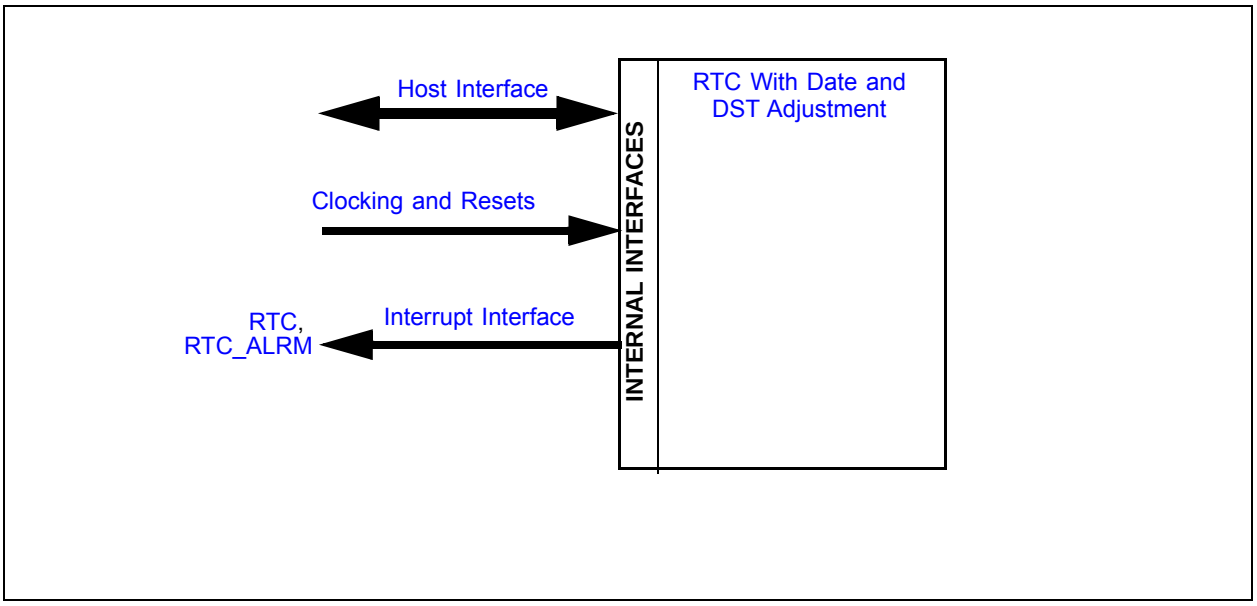
Time and Date Registers

This is the set of registers that are automatically counted by hardware every 1 second while the block is enabled to run and to update. These registers are: **Seconds**, **Minutes**, **Hours**, **Day of Week**, **Day of Month**, **Month**, and **Year**.

23.4 Interface

This block's connections are entirely internal to the chip.

FIGURE 23-1: RTC WITH DATE AND DST ADJUSTMENT INTERFACE DIAGRAM



23.4.1 HOST INTERFACE

The registers defined for the RTC With Date and DST Adjustment are accessible by the host as defined.

23.4.2 CLOCKING AND RESETS

This IP block has the following clocks and reset ports. For a complete list of all the clocks and resets associated with this block see Section 23.5, "Power, Clocks and Resets," on page 376.

TABLE 23-1: SIGNAL DESCRIPTION TABLE

Name	Direction	Description
VBAT_POR	Input	Reset asserted when VCC0 / VBAT power is cycled.
nSYS_RST	Input	Reset signal used to indicate when the chip's main internal rail is cycled.
X32K_CLK	Input	Clock source to the block. All internal block clocking is derived from this source.
LPC Bus Clock	Input	Clock source to the block, used only for bus communication while the chip's primary rail VTR is up.

23.4.3 INTERRUPT INTERFACE

This section defines the interrupt Interface signals routed to the chip's interrupt aggregator.

TABLE 23-2: SIGNAL DESCRIPTION TABLE

Name	Direction	Description
RTC	Output	This signal is the OR'd result of the interrupt signals defined in Section 23.6, "Interrupt Generation," on page 377. It is used to notify the interrupt controller that an event has occurred.
RTC_ALRM	Output	This signal is from the Alarm section only, presented after interrupt masking. It is used at the chip level to notify VCI logic that an Alarm event has occurred.

23.5 Power, Clocks and Resets

This section defines the Power, Clock, and Reset parameters of the block.

23.5.1 POWER DOMAINS

TABLE 23-3: POWER SOURCES

Name	Description
VBAT	This power well sources all of the internal registers and logic in this block.
VTR	This power well sources only bus communication. The block continues to operate internally while this rail is down.

23.5.2 CLOCKS

TABLE 23-4: CLOCKS

Name	Description
X32K_CLK	This 32KHz clock input drives all internal logic, and will be present at all times that the VBAT well is powered.
LPC Bus Clock	This clock input drives bus communication only.

© 2012-2018 Microchip Technology Inc.

23.5.3 RESETS

TABLE 23-5: RESET SIGNALS

Name	Description
VBAT_POR	This reset signal resets all of the registers and logic in this block.
RTC_RST	This reset signal resets all of the registers and logic in this block. It is triggered by VBAT_POR, but can also be triggered by the Soft Reset bit in the RTC Control Register.
nSYS_RST	This reset signal is used to inhibit the bus communication logic, and isolates this block from VTR powered circuitry on-chip. Otherwise it has no effect on the internal state.

23.6 Interrupt Generation

TABLE 23-6: GENERATED INTERRUPTS

Source	Description
Update Complete	This source triggers, at 1-second intervals, when the Time register updates have completed.
Alarm	This source triggers when the alarm value matches the current time (and date, if used).
Periodic	This source triggers at the chosen programmable rate.

23.7 Low Power Modes

The RTC With Date and DST Adjustment has no low-power modes. It runs continuously while the VBAT well is powered.

Note: The RTC_SLEEP_EN signal has no effect on the block. The RTC_CLK_REQ signal will go active for register accesses, however, this is a short-lived event.

23.8 Description

This block provides the capabilities of an industry-standard 146818B Real-Time Clock module, excluding the CMOS RAM and the SQW output. See the following registers, which represent enhancements to this architecture. These enhancements are listed below.

See the Date Alarm field of Register D for a Day of Month qualifier for alarms.

See the Week Alarm Register for a Day of Week qualifier for alarms.

See the registers Daylight Savings Forward Register and Daylight Savings Backward Register for setting up hands-off Daylight Savings adjustments.

See the RTC Control Register for enhanced control over the block's operations.

23.9 RTC Register Set

The registers listed in Table 23-8, "RTC Register Set Summary" are for a single instance of the RTC With Date and DST Adjustment block. The addresses of each register listed in this table are defined as a relative offset to the host "Base Address" defined in Table 23-7, "RTC Register Set Address Range Table".

TABLE 23-7: RTC REGISTER SET ADDRESS RANGE TABLE

Instance Name	Instance Number	Host	Address Space	Base Address
RTC	0	EC	EC System Address Space	FF_4400h

Note 23-1 The Base Address indicates where the first register can be accessed in a particular address space for a block instance. Add the register's Offset to this value to obtain the direct address of the register.

TABLE 23-8: RTC REGISTER SET SUMMARY

Offset	Register Name (Mnemonic)
00h	Seconds Register
01h	Seconds Alarm Register
02h	Minutes Register
03h	Minutes Alarm Register
04h	Hours Register
05h	Hours Alarm Register
06h	Day of Week Register
07h	Day of Month Register
08h	Month Register
09h	Year Register
0Ah	Register A
0Bh	Register B
0Ch	Register C
0Dh	Register D
0Eh	(reserved)
0Fh	(reserved)
10h	RTC Control Register
14h	Week Alarm Register
18h	Daylight Savings Forward Register
1Ch	Daylight Savings Backward Register
20h	

Note 23-2 Note that this extended register set occupies offsets that have historically been used as CMOS RAM. Code ported to use this block should be examined to ensure that it does not assume that RAM exists in this block.

23.9.1 SECONDS REGISTER

(Seconds)

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	**Seconds** Displays the number of seconds past the current minute, in the range 0--59. Presentation may be selected as binary or BCD, depending on the DM bit in Register B. Values written must also use the format defined by the current setting of the DM bit.	R/W	00h	RTC_RST

© 2012-2018 Microchip Technology Inc.

23.9.2 SECONDS ALARM REGISTER

(Seconds_Alarm)

Offset	01h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	**Seconds Alarm** Holds a match value, compared against the Seconds Register to trigger the Alarm event. Values written to this register must use the format defined by the current setting of the DM bit in Register B. A value of 11xxxxxxb written to this register makes it don't-care (always matching).	R/W	00h	RTC_RST

23.9.3 MINUTES REGISTER

(Minutes)

Offset	02h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	**Minutes** Displays the number of minutes past the current hour, in the range 0--59. Presentation may be selected as binary or BCD, depending on the DM bit in Register B. Values written must also use the format defined by the current setting of the DM bit.	R/W	00h	RTC_RST

23.9.4 MINUTES ALARM REGISTER

(Minutes_Alarm)

Offset	03h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	**Minutes Alarm** Holds a match value, compared against the Minutes Register to trigger the Alarm event. Values written to this register must use the format defined by the current setting of the DM bit in Register B. A value of 11xxxxxxb written to this register makes it don't-care (always matching).	R/W	00h	RTC_RST

23.9.5 HOURS REGISTER

(Hours)

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	**Hours AM/PM** In 12-hour mode (see bit “24/12” in register B), this bit indicates AM (0) or PM (1).	R/W	0b	RTC_RST
6:0	**Hours** Displays the number of the hour, in the range 1--12 for 12-hour mode (see bit “24/12” in register B), or in the range 0--23 for 24-hour mode. Presentation may be selected as binary or BCD, depending on the DM bit in Register B. Values written must also use the format defined by the current setting of the DM bit.	R/W	00h	RTC_RST

23.9.6 HOURS ALARM REGISTER

(Hours_Alarm)

Offset	05h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	**Hours Alarm** Holds a match value, compared against the Hours Register to trigger the Alarm event. Values written to this register must use the format defined by the current settings of the DM bit and the 24/12 bit in Register B. A value of 11xxxxxxb written to this register makes it don’t-care (always matching).	R/W	00h	RTC_RST

23.9.7 DAY OF WEEK REGISTER

(Day_of_Week)

Offset	06h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	**Day of Week** Displays the day of the week, in the range 1 (Sunday) through 7 (Saturday). Numbers in this range are identical in both binary and BCD notation, so this register’s format is unaffected by the DM bit.	R/W	00h	RTC_RST

© 2012-2018 Microchip Technology Inc.

23.9.8 DAY OF MONTH REGISTER

(Day_of_Month)

Offset	07h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	**Day of Month** Displays the day of the current month, in the range 1--31. Presentation may be selected as binary or BCD, depending on the DM bit in Register B. Values written must also use the format defined by the current setting of the DM bit.	R/W	00h	RTC_RST

23.9.9 MONTH REGISTER

(Month)

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	**Month** Displays the month, in the range 1--12. Presentation may be selected as binary or BCD, depending on the DM bit in Register B. Values written must also use the format defined by the current setting of the DM bit.	R/W	00h	RTC_RST

23.9.10 YEAR REGISTER

(Year)

Offset	09h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	**Year** Displays the number of the year in the current century, in the range 0 (year 2000) through 99 (year 2099). Presentation may be selected as binary or BCD, depending on the DM bit in Register B. Values written must also use the format defined by the current setting of the DM bit.	R/W	00h	RTC_RST

23.9.11 REGISTER A

(Register_A)

Offset	0Ah			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	Update In Progress (status_update_in_progress) '0' indicates that the Time and Date registers are stable and will not be altered by hardware soon. '1' indicates that a hardware update of the Time and Date registers may be in progress, and those registers should not be accessed by the host program. This bit is set to '1' at a point 488us (16 cycles of the 32K clock) before the update occurs, and is cleared immediately after the update. See also the Update-Ended Interrupt, which provides more useful status.	RO	0b	RTC_RST
6:4	Division Chain Select (division_chain_select) This field provides general control for the Time and Date register updating logic. **000b (default)** is a reserved value. It should be initialized to another value before Enabling the block in the RTC Control Register (See uarch for actual behavior) **010b** is the required setting for normal operation. It is also necessary to set the Block Enable bit in the RTC Control Register to '1' for counting to begin. **11xb** written to this field will halt counting. The next time that 010b is written, updates will begin 500ms later. **Other values are reserved.**	R/W	000b	RTC_RST
3:0	Rate Select (RS) (period_divide) This field selects the rate of the Periodic Interrupt source.	R/W	0h	RTC_RST

TABLE 23-9: REGISTER A FIELD RS: PERIODIC INTERRUPT SETTINGS

RS (HEX)	Interrupt Period
0	Never Triggered
1	3.90625 ms
2	7.8125 ms
3	122.070 us
4	244.141 us
5	488.281 us
6	976.5625 us
7	1.953125 ms
8	3.90625 ms
9	7.8125 ms
A	15.625 ms
B	31.25 ms
C	62.5 ms
D	125 ms
E	250 ms
F	500 ms

© 2012-2018 Microchip Technology Inc.

23.9.12 REGISTER B

(Register_B)

Offset	0Bh			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	Update Cycle Inhibit (SET) (halt) In its default state '0', this bit allows hardware updates to the Time and Date registers, which occur at 1-second intervals. A '1' written to this field inhibits updates, allowing these registers to be cleanly written to different values. Writing '0' to this bit allows updates to continue.	R/W	0b	RTC_RST
6	Periodic Interrupt Enable (PIE) (int_en_periodic) '1' allows the Periodic Interrupt events to be propagated as interrupts.	R/W	0b	RTC_RST
5	Alarm Interrupt Enable (AIE) (int_en_alarm) '1' allows the Alarm Interrupt events to be propagated as interrupts.	R/W	0b	RTC_RST
4	Update-Ended Interrupt Enable (UIE) (int_en_update_complete) '1' allows the Update Ended Interrupt events to be propagated as interrupts.	R/W	0b	RTC_RST
3	RESERVED	RES	-	-
2	Data Mode (DM) (data_mode) 0 = BCD Mode for Dates and Times 1 = Binary Mode for Dates and Times	R/W	0b	RTC_RST
1	Hour Format (24/12) (mode_24_hour) 0 = 12-Hour Format for Hours and Hours Alarm registers. 1 = 24-Hour Format for Hours and Hours Alarm registers. 12-Hour format has an AM/PM bit, and value range 1--12. 24-Hour format keeps the AM/PM bit off, with value range 0--23.	R/W	0b	RTC_RST
0	Daylight Savings Enable (DSE) (daylightsaving_enable) '1' enables automatic hardware updating of the hour, using the registers Daylight Savings Forward and Daylight Savings Backward to select the yearly date and hour for each update.	R/W	0b	RTC_RST

Note: The **Data Mode** and **Hour Format** bits affect only how values are presented as they are being read and how they are interpreted as they are being written. They do not affect the internal contents or interpretations of registers that have already been written, nor do they affect how those registers are represented or counted internally. This mode bits may be set and cleared dynamically, for whatever I/O data representation is desired by the host program.

23.9.13 REGISTER C

(Register_C)

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	Interrupt Request Flag (IRQF) (intr) A '1' in this bit position indicates that any of bits[6:4] below is active after masking by their respective Enable bits in Register B. This bit is automatically cleared by every Read access to this register.	RC	0b	RTC_RST
6	Periodic Interrupt Flag (PF) (status_periodic) A '1' in this bit position means that a Periodic Interrupt event has occurred since the last time this register was read. This bit displays status regardless of the Periodic Interrupt Enable bit in Register B. This bit is automatically cleared by every Read access to this register.	RC	0b	RTC_RST
5	Alarm Flag (AF) (status_alarm) A '1' in this bit position means that an Alarm event has occurred since the last time this register was read. This bit displays status regardless of the Alarm Interrupt Enable bit in Register B. This bit is automatically cleared by every Read access to this register.	RC	0b	RTC_RST
4	Update-Ended Interrupt Flag (UF) (status_update_complete) A '1' in this bit position means that a Time and Date update has completed since the last time this register was read. This bit displays status regardless of the Update-Ended Interrupt Enable bit in Register B. Presentation of this status indicates that the Time and Date registers will be valid and stable for over 999ms. This bit is automatically cleared by every Read access to this register.	RC	0b	RTC_RST
3:0	RESERVED	RES	-	-

23.9.14 REGISTER D

(Register_D)

Offset	0Dh			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:6	RESERVED	RES	-	-
5:0	Date Alarm (alarm_date) This field, if set to a non-zero value, will inhibit the Alarm interrupt unless this field matches the contents of the Month register also. If this field contains 00h (default), it represents a don't-care, allowing more frequent alarms.	R/W	00h	RTC_RST

© 2012-2018 Microchip Technology Inc.

23.9.15 RTC CONTROL REGISTER

(RTC_Control)

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:4	RESERVED	RES	-	-
3	Alarm Enable (alarm_enable) '1' enables the Alarm features.	R/W	0b	RTC_RST
2	VCI Enable (vci_enable) '1' allows Alarm events to activate the RTC_ALRM signal to chip-level VCI circuitry (if present).	R/W	0b	RTC_RST
2	RESERVED: RAM bit with no effect.	R/W	0b	RTC_RST
1	Soft Reset (soft_reset) A '1' written to this bit position will trigger the RTC_RST reset, resetting the block and all registers except this one and the Test Register. This bit is self-clearing at the end of the reset, one cycle of the Bus Clock later, and so requires no explicit programming to clear it or to wait for it to complete.	R/W	0b	VBAT_POR
0	Block Enable (enable) This bit must be '1' in order for the block to function internally. Registers may be initialized first, before setting this bit to '1' to start operation.	R/W	0b	RTC_RST

23.9.16 WEEK ALARM REGISTER

(Week_Alarm)

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	Alarm Day of Week (alarm_weekdays) This register, if written to a value in the range 1--7, will inhibit the Alarm interrupt unless this field matches the contents of the Day of Week Register also. If this field is written to any value 11xxxxxxb (like the default FFh), it represents a don't-care, allowing more frequent alarms, and will read back as FFh until another value is written.	R/W	FFh	RTC_RST

23.9.17 DAYLIGHT SAVINGS FORWARD REGISTER

(DST_Forward)

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	DST Forward AM/PM (dls_fwd_am_pm) This bit selects AM vs. PM, to match bit[7] of the Hours register if 12-Hour mode is selected in Register B at the time of writing.	R/W	0b	RTC_RST
30:24	DST Forward Hour (dls_fwd_hour) This field holds the matching value for bits[6:0] of the Hours register. The written value will be interpreted according to the 24/12 Hour mode and DM mode settings at the time of writing.	R/W	00h	RTC_RST
23:19	RESERVED	RES	-	-
18:16	DST Forward Week (dls_fwd_week) This value matches an internally-maintained week number within the current month. Valid values for this field are: 1 = First week of month 2 = Second week of month 3 = Third week of month 4 = Fourth week of month 5 = Last week of month	R/W	0h	RTC_RST
15:11	RESERVED	RES	-	-
10:8	DST Forward Day of Week (dls_fwd_weekday) This field matches the Day of Week Register bits[2:0].	R/W	0h	RTC_RST
7:0	DST Forward Month (dls_fwd_month) This field matches the Month Register.	R/W	00h	RTC_RST

This is a 32-bit register, accessible also as individual bytes. When writing as individual bytes, ensure that the DSE bit (in Register B) is off first, or that the block disabled or stopped (SET bit), to prevent a time update while this register may have incompletely-updated contents.

When enabled by the DSE bit in Register B, this register defines an hour and day of the year at which the Hours register will be automatically incremented by 1 additional hour.

There are no don't-care fields recognized. All fields must be already initialized to valid settings whenever the DSE bit is '1'.

Fields other than Week and Day of Week use the current setting of the DM bit (binary vs. BCD) to interpret the information as it is written to them. Their values, as held internally, are not changed by later changes to the DM bit, without subsequently writing to this register as well.

Note: An Alarm that is set inside the hour after the time specified in this register will not be triggered, because that one-hour period is skipped. This period includes the exact time (0 minutes : 0 seconds) given by this register, through the 59 minutes : 59 seconds point afterward.

© 2012-2018 Microchip Technology Inc.

23.9.18 DAYLIGHT SAVINGS BACKWARD REGISTER

(DST_Backward)

Offset	1Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31	DST Backward AM/PM (dls_bkwd_am_pm) This bit selects AM vs. PM, to match bit[7] of the Hours register if 12-Hour mode is selected in Register B at the time of writing.	R/W	0b	RTC_RST
30:24	DST Backward Hour (dls_bkwd_hour) This field holds the matching value for bits[6:0] of the Hours register. The written value will be interpreted according to the 24/12 Hour mode and DM mode settings at the time of writing.	R/W	00h	RTC_RST
23:19	RESERVED	RES	-	-
18:16	DST Backward Week (dls_bkwd_week) This value matches an internally-maintained week number within the current month. Valid values for this field are: 1 = First week of month 2 = Second week of month 3 = Third week of month 4 = Fourth week of month 5 = Last week of month	R/W	0h	RTC_RST
15:11	RESERVED	RES	-	-
10:8	DST Backward Day of Week (dls_bkwd_weekday) This field matches the Day of Week Register bits[2:0].	R/W	0h	RTC_RST
7:0	DST Backward Month (dls_bkwd_month) This field matches the Month Register.	R/W	00h	RTC_RST

This is a 32-bit register, accessible also as individual bytes. When writing as individual bytes, ensure that the DSE bit (in Register B) is off first, or that the block disabled or stopped (SET bit), to prevent a time update while this register may have incompletely-updated contents.

When enabled by the DSE bit in Register B, this register defines an hour and day of the year at which the Hours register increment will be inhibited from occurring. After triggering, this feature is automatically disabled for long enough to ensure that it will not re-trigger the second time this Hours value appears, and then this feature is re-enabled automatically.

There are no don’t-care fields recognized. All fields must be already initialized to valid settings whenever the DSE bit is ‘1’.

Fields other than Week and Day of Week use the current setting of the DM bit (binary vs. BCD) to interpret the information as it is written to them. Their values, as held internally, are not changed by later changes to the DM bit, without subsequently writing to this register as well.

Note: An Alarm that is set inside the hour before the time specified in this register will be triggered twice, because that one-hour period is repeated. This period will include the exact time (0 minutes : 0 seconds) given by this register, through the 59 minutes : 59 seconds point afterward.

24.0 GPIO INTERFACE

24.1 General Description

The MEC1632 GPIO Interface provides general purpose input monitoring and output control, as well as managing many aspects of pin functionality; including, multi-function Pin Multiplexing Control, GPIO Direction control, PU/PD (PU_PD) resistors, asynchronous wakeup and synchronous Interrupt Detection (int_det), GPIO Direction, and Polarity control.

Features of the GPIO Interface include:

- Inputs:
  - Asynchronous rising and falling edge wakeup detection
  - Interrupt High or Low Level
- Outputs:
  - Push Pull or Open Drain output
  - Programmable power well emulation
- Pull up or pull down resistor control
- Interrupt and wake capability available for all GPIOs
- 8 GPIO Pass-Through Ports
- Group- or individual control of GPIO data. See Section 24.5 and Section 24.10
- Multi-function pin multiplexing is controlled by the GPIO Interface
- Power consumption in the GPIO Interface is reduced by automatically disabling pull-up resistors when GPIO outputs are driven ‘low,’ and disabling pull-down resistors when GPIO outputs are driven ‘high’.

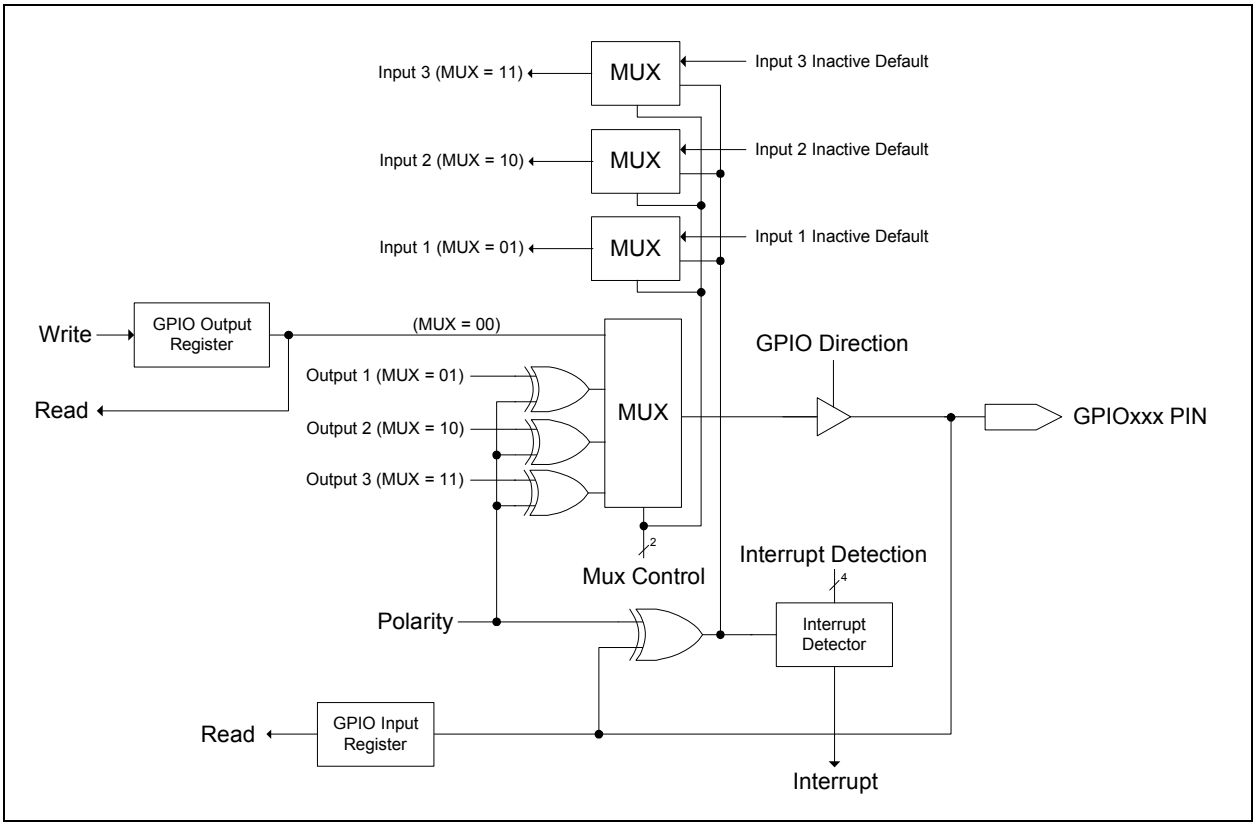
24.2 Block Diagram

The GPIO Interface Block Diagram shown in Figure 24-1 illustrates the functionality of a single MEC1632 GPIO Interface pin. The source for the Pin Multiplexing Control, Interrupt Detection (int_det), GPIO Direction, and Polarity controls in Figure 24-1 is a Pin Control Register that is associated with each pin (see Section 24.9.1, "Pin Control Register," on page 396).

The MEC1632 supports up to four independent signal functions per pin including the GPIO signal function itself, which is always positioned at Mux Control = ‘00.’ The GPIO Input Registers and the GPIO Output Registers provide the GPIO Interface ‘Read’ and ‘Write’ functionality illustrated in Figure 24-1 (see Section 24.9.3, "GPIO Input Registers," on page 402 and Section 24.9.2, "GPIO Output Registers," on page 399).

© 2012-2018 Microchip Technology Inc.

FIGURE 24-1: GPIO Interface BLOCK DIAGRAM



24.3 Power, Clocks and Reset

24.3.1 POWER DOMAIN

This block is powered by the VTR Power Supply.

24.3.2 CLOCKS

This block uses the EC Bus Clock and MCLK. EC Bus Clock is used for access to registers. The MCLK is used for synchronizing the GPIO inputs.

24.3.3 RESET

This block is reset on a nSYS_RST. On reset, all Registers are reset to their default values.

24.4 Interrupts

Each pin in the GPIO Interface has both an interrupt and/or a Wake-up event (Table 24-1). The interrupt source is routed onto the GPIO Status Bits corresponding to the specific GPIO identified in the GIRQ8 Source Register - GIRQ11 Source Register, and the GIRQ22 Source Register. The GPIO Interface can generate an interrupt on a high level, low level, rising edge and falling edge, as configured by the Interrupt Detection (int_det) bits in the Pin Control Register associated with the GPIO signal function.

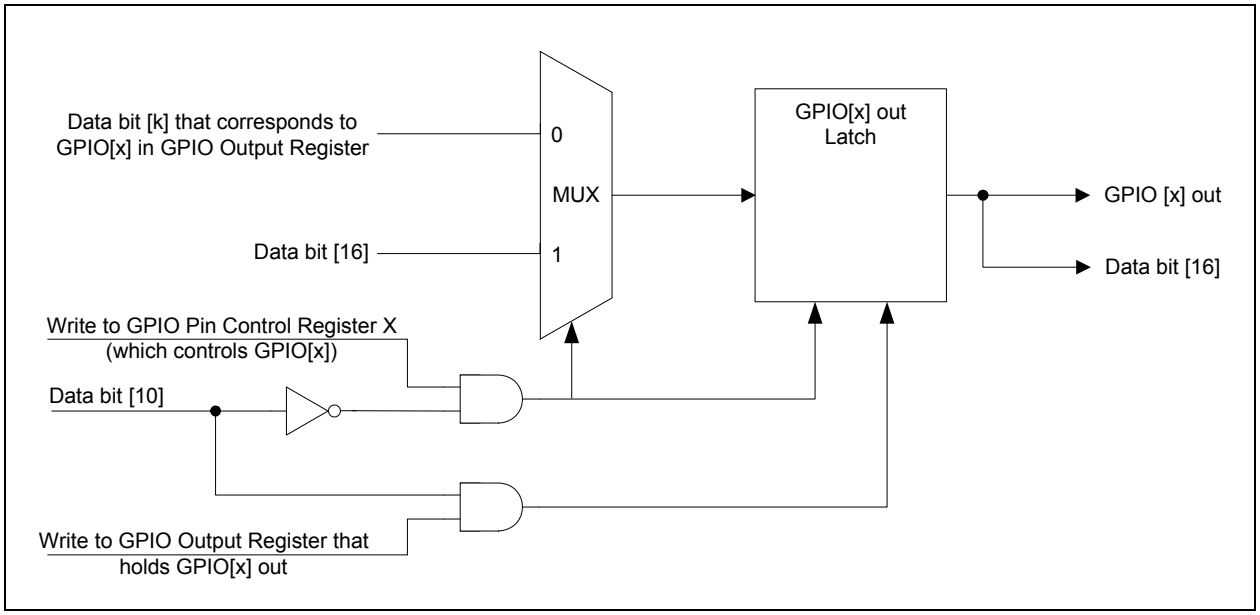
TABLE 24-1: GPIO INTERRUPT/WAKE EVENT TIMING PARAMETERS

Symbol	Parameter	Limits			Units	Comments
		MIN	NOM	MAX		
t_{WAKE}	Wake Event Time	5	–	–	ns	Minimum pulse required on a GPIO pin to wake the system.
$t_{INTERRUPT}$	Interrupt Status Assertion Time	300	–	–	ns	Minimum pulse required on a GPIO pin to assert an IRQ Status bit.

24.5 Accessing GPIOs

There are two ways to specify GPIO input and output port data. In the legacy approach that maintains compatibility with earlier generation devices, outputs to individual GPIO ports are grouped into four 32-bit GPIO Output Registers (see Table 24-6.) It is incumbent on firmware to modify particular bit(s) while not disturbing the others. The MEC1632 supports an alternative approach in which each port's output is individually specified, i.e., Bit [16] in the port Pin Control Register is used for output data. Bit [10] Output GPIO Write Enable is used to enable this alternative write to the GPIO on a per-bit basis. Figure 24-2 illustrates the concept. On reads, Bit [16] returns the programmed value while Bit [24] reflects the state of GPIO input from the pad regardless of setting of Bit [10].

FIGURE 24-2: OUTPUT DATA TO GPIO



© 2012-2018 Microchip Technology Inc.

24.6 GPIO Indexing

Each GPIO signal function name consists of a 4-character prefix (“GPIO”) followed by a 3-digit octal-encoded index number. GPIO Indexing is done sequentially starting from ‘GPIO000’. There is a unique index number for each GPIO pin function. Index numbers for the NORM Exception Bits are skipped; e.g., there is no GPIO037, GPIO077, GPIO137 or GPIO177 (see Section 24.9.3, "GPIO Input Registers," on page 402).

24.7 Pin Multiplexing Control

As described above in Section 24.2, "Block Diagram", pin multiplexing depends upon the Mux Control bits in the Pin Control Register. There is a Pin Control Register for each GPIO signal function.

The MEC1632 Pin Control Register address offsets shown in the following tables depends on the GPIO Index number. Pin Control Register defaults are also shown in these tables.

GPIO signal function names in parentheses in the Pin Control Register tables represent interrupt/wake-only GPIO signal functions. The Mux Control bits in the Pin Control Register for these pins should not be programmed ‘00.’

© 2012-2018 Microchip Technology Inc.

TABLE 24-2: PIN CONTROL MUXING TABLE 1

Pin Ref. Number	GPIO Name (Octal)	Pin Control Reg. Offset (Hex)	Pin Control Reg. POR Value (Hex)	POR Default Signal Function	Mux Control = 00	Mux Control = 01	Mux Control = 10	Mux Control = 11
15	GPIO000	0000	00001000	VCI_IN3#	GPIO000	VCI_IN3#	Reserved	Reserved
78	GPIO001	0004	00000000	GPIO001	GPIO001	PWM4	Reserved	Reserved
79	GPIO002	0008	00000000	GPIO002	GPIO002	Reserved	Reserved	Reserved
147	GPIO003	000C	00000000	GPIO003	GPIO003	SMB00_DATA	Reserved	Reserved
146	GPIO004	0010	00000000	GPIO004	GPIO004	SMB00_CLK	Reserved	Reserved
145	GPIO005	0014	00000000	GPIO005	GPIO005	SMB01_DATA	Reserved	Reserved
144	GPIO006	0018	00000000	GPIO006	GPIO006	SMB01_CLK	Reserved	Reserved
140	GPIO007	001C	00000000	GPIO007	GPIO007	SMB03_DATA	Reserved	Reserved
141	GPIO010	0020	00000000	GPIO010	GPIO010	SMB03_CLK	Reserved	Reserved
62	GPIO011	0024	00000000	GPIO011	GPIO011	nSMI	Reserved	Reserved
88	GPIO012	0028	00000000	GPIO012	GPIO012	SMB07_DATA	Reserved	Reserved
89	GPIO013	002C	00000000	GPIO013	GPIO013	SMB07_CLK	Reserved	Reserved
80	GPIO014	0030	00000000	GPIO014	GPIO014	Reserved	Reserved	Reserved
81	GPIO015	0034	00000000	GPIO015	GPIO015	Reserved	Reserved	Reserved
70	GPIO016	0038	00000000	GPIO016	GPIO016	Reserved	Reserved	Reserved
84	GPIO017	003C	00000000	GPIO017	GPIO017	Reserved	Reserved	Reserved
161	GPIO020	0040	00000000	GPIO020	GPIO020	Reserved	Reserved	Reserved
26	GPIO021	0044	00000000	GPIO021	GPIO021	Reserved	Reserved	Reserved
148	GPIO022	0048	00000000	GPIO022	GPIO022	BCM_B_CLK	Reserved	Reserved
149	GPIO023	004C	00000000	GPIO023	GPIO023	BCM_B_DAT	Reserved	Reserved
150	GPIO024	0050	00000000	GPIO024	GPIO024	BCM_B_INT#	Reserved	Reserved
95	GPIO025	0054	00000000	GPIO025	GPIO025	Reserved	Reserved	Reserved
96	GPIO026	0058	00000000	GPIO026	GPIO026	Reserved	TIN1	Reserved
97	GPIO027	005C	00000000	GPIO027	GPIO027	Reserved	Reserved	Reserved
98	GPIO030	0060	00000000	GPIO030	GPIO030	Reserved	Reserved	Reserved
87	GPIO031	0064	00000000	GPIO031	GPIO031	Reserved	Reserved	Reserved
86	GPIO032	0068	00000000	GPIO032	GPIO032	Reserved	Reserved	Reserved
25	GPIO033	006C	00000000	GPIO033	GPIO033	RC_ID	Reserved	Reserved
152	GPIO034	0070	00000000	GPIO034	GPIO034	Reserved	Reserved	Reserved
120	GPIO035	0074	00000000	GPIO035	GPIO035	Reserved	Reserved	Reserved
153	GPIO036	0078	00000000	GPIO036	GPIO036	Reserved	Reserved	Reserved
85	GPIO040	0080	00000000	GPIO040	GPIO040	Reserved	Reserved	Reserved
117	GPIO041	0084	00000000	GPIO041	GPIO041	SYS_SHDN#	Reserved	Reserved
126	GPIO042	0088	00002000	PECI_DAT	GPIO042	Reserved	PECI_DAT	Reserved
127	GPIO043	008C	00000000	GPIO043	GPIO043	Reserved	Reserved	Reserved
128	GPIO044	0090	00000000	GPIO044	GPIO044	VREF_VTT	Reserved	Reserved
154	GPIO045	0094	00000000	GPIO045	GPIO045	Reserved	Reserved	Reserved
155	GPIO046	0098	00000000	GPIO046	GPIO046	Reserved	Reserved	Reserved
156	GPIO047	009C	00000000	GPIO047	GPIO047	Reserved	Reserved	Reserved
64	GPIO050	00A0	00000000	GPIO050	GPIO050	FAN_TACH0	Reserved	Reserved
66	GPIO051	00A4	00000000	GPIO051	GPIO051	Reserved	Reserved	Reserved
68	GPIO052	00A8	00000000	GPIO052	GPIO052	Reserved	Reserved	Reserved
72	GPIO053	00AC	00000000	GPIO053	GPIO053	PWM0	Reserved	Reserved
74	GPIO054	00B0	00000000	GPIO054	GPIO054	Reserved	Reserved	Reserved
76	GPIO055	00B4	00000000	GPIO055	GPIO055	Reserved	Reserved	Reserved
77	GPIO056	00B8	00000000	GPIO056	GPIO056	Reserved	Reserved	Reserved
20	GPIO057	00BC	00001000	VCC_PWRGD	GPIO057	VCC_PWRGD	Reserved	Reserved
29	GPIO060	00C0	00000000	GPIO060	GPIO060	Reserved	Reserved	Reserved
63	GPIO061	00C4	00000000	GPIO061	GPIO061	LPCPD#	Reserved	Reserved
19	GPIO062	00C8	00000200	GPIO062	GPIO062	Reserved	Reserved	Reserved
54	GPIO063	00CC	00001000	SER_IRQ	GPIO063	SER_IRQ	Reserved	Reserved
51	GPIO064	00D0	00001000	LRESET#	GPIO064	LRESET#	Reserved	Reserved
55	GPIO065	00D4	00001000	PCI_CLK	GPIO065	PCI_CLK	Reserved	Reserved
53	GPIO066	00D8	00001000	LFRAME#	GPIO066	LFRAME#	Reserved	Reserved
52	GPIO067	00DC	00001000	CLKRUN#	GPIO067	CLKRUN#	Reserved	Reserved
111	GPIO070	00E0	00000000	GPIO070	GPIO070	Reserved	Reserved	Reserved
112	GPIO071	00E4	00000000	GPIO071	GPIO071	Reserved	Reserved	Reserved
113	GPIO072	00E8	00000000	GPIO072	GPIO072	Reserved	Reserved	Reserved
114	GPIO073	00EC	00000000	GPIO073	GPIO073	Reserved	Reserved	Reserved
115	GPIO074	00F0	00000000	GPIO074	GPIO074	Reserved	Reserved	Reserved
116	GPIO075	00F4	00000000	GPIO075	GPIO075	Reserved	Reserved	Reserved
118	GPIO076	00F8	00000000	GPIO076	GPIO076	Reserved	Reserved	Reserved
61	GPIO100	0100	00001000	nEC_SCI	GPIO100	nEC_SCI	Reserved	Reserved
5	GPIO101	0104	00000000	GPIO101	GPIO101	Reserved	Reserved	Reserved

© 2012-2018 Microchip Technology Inc.

TABLE 24-3: PIN CONTROL MUXING TABLE 2

Pin Ref. Number	GPIO Name (Octal)	Pin Control Reg. Offset (Hex)	Pin Control Reg. POR Value (Hex)	POR Default Signal Function	Mux Control = 00	Mux Control = 01	Mux Control = 10	Mux Control = 11
6	GPIO102	0108	00000000	GPIO102	GPIO102	Reserved	Reserved	Reserved
134	GPIO104	0110	00001000	UART_TX	GPIO104	UART_TX	Reserved	Reserved
135	GPIO105	0114	00001000	UART_RX	GPIO105	UART_RX	Reserved	Reserved
21	GPIO106	0118	00001000	nRESET_OUT	GPIO106	nRESET_OUT	Reserved	Reserved
99	GPIO107	011C	00000000	GPIO107	GPIO107	Reserved	Reserved	Reserved
103	GPIO110	0120	00000000	GPIO110	GPIO110	Reserved	Reserved	Reserved
104	GPIO111	0124	00000000	GPIO111	GPIO111	Reserved	Reserved	Reserved
105	GPIO112	0128	00000000	GPIO112	GPIO112	Reserved	Reserved	Reserved
106	GPIO113	012C	00000000	GPIO113	GPIO113	Reserved	Reserved	Reserved
107	GPIO114	0130	00000000	GPIO114	GPIO114	Reserved	Reserved	Reserved
108	GPIO115	0134	00000000	GPIO115	GPIO115	Reserved	Reserved	Reserved
100	GPIO120	0140	00000000	GPIO120	GPIO120	KSO7	Reserved	Reserved
157	GPIO121	0144	00000000	GPIO121	GPIO121	Reserved	Reserved	Reserved
158	GPIO122	0148	00000000	GPIO122	GPIO122	Reserved	Reserved	Reserved
159	GPIO123	014C	00000000	GPIO123	GPIO123	Reserved	Reserved	Reserved
101	GPIO124	0150	00000000	GPIO124	GPIO124	Reserved	Reserved	Reserved
102	GPIO125	0154	00000000	GPIO125	GPIO125	Reserved	Reserved	Reserved
160	GPIO126	0158	00000000	GPIO126	GPIO126	KSO13	Reserved	Reserved
151	GPIO127	015C	00001000	A20M	GPIO127	A20M	Reserved	Reserved
90	GPIO130	0160	00000000	GPIO130	GPIO130	Reserved	Reserved	Reserved
91	GPIO131	0164	00000000	GPIO131	GPIO131	Reserved	Reserved	Reserved
92	GPIO132	0168	00000000	GPIO132	GPIO132	Reserved	Reserved	Reserved
123	GPIO133	016C	00000000	GPIO133	GPIO133	Reserved	Reserved	Reserved
124	GPIO134	0170	00000000	GPIO134	GPIO134	Reserved	Reserved	Reserved
125	GPIO135	0174	00000000	GPIO135	GPIO135	PWM11	Reserved	Reserved
93	GPIO140	0180	00000000	GPIO140	GPIO140	Reserved	Reserved	Reserved
136	GPIO141	0184	00000000	GPIO141	GPIO141	SMB05_DATA	Reserved	Reserved
137	GPIO142	0188	00000000	GPIO142	GPIO142	SMB05_CLK	Reserved	Reserved
138	GPIO143	018C	00000000	GPIO143	GPIO143	SMB04_DATA	Reserved	Reserved
139	GPIO144	0190	00000000	GPIO144	GPIO144	SMB04_CLK	Reserved	Reserved
129	GPIO145	0194	00000000	GPIO145	GPIO145	Reserved	JTAG_TDI	Reserved
130	GPIO146	0198	00000000	GPIO146	GPIO146	Reserved	JTAG_TDO	Reserved
131	GPIO147	019C	00000000	GPIO147	GPIO147	Reserved	Reserved	JTAG_CLK
132	GPIO150	01A0	00000000	GPIO150	GPIO150	Reserved	Reserved	JTAG_TMS
82	GPIO151	01A4	00000000	GPIO151	GPIO151	Reserved	Reserved	Reserved
83	GPIO152	01A8	00000000	GPIO152	GPIO152	Reserved	Reserved	Reserved
164	GPIO153	01AC	00000000	GPIO153	GPIO153	LED2	Reserved	Reserved
142	GPIO154	01B0	00000000	GPIO154	GPIO154	Reserved	PS2_CLK1B	Reserved
143	GPIO155	01B4	00000000	GPIO155	GPIO155	Reserved	PS2_DAT1B	Reserved
162	GPIO156	01B8	00000000	GPIO156	GPIO156	LED0	Reserved	Reserved
163	GPIO157	01BC	00000000	GPIO157	GPIO157	LED1	Reserved	Reserved
11	GPIO161	01C4	00001000	VCI_IN2#	GPIO161	VCI_IN2#	Reserved	Reserved
12	GPIO162	01C8	00001000	VCI_IN1#	GPIO162	VCI_IN1#	Reserved	Reserved
13	GPIO163	01CC	00001000	VCI_IN0#	GPIO163	VCI_IN0#	Reserved	Reserved
14	GPIO164	01D0	00001000	VCI_OVRD_IN	GPIO164	VCI_OVRD_IN	Reserved	Reserved
1	GPIO165	01D4	00000000	GPIO165	GPIO165	32KHZ_IN	Reserved	Reserved
30	GPIO166	01D8	00000000	GPIO166	GPIO166	Reserved	Reserved	Reserved
121	GPIO170	01E0	00001000	MSCLK	GPIO170	MSCLK	Reserved	Reserved
122	GPIO171	01E4	00001001	MSDATA	GPIO171	MSDATA	Reserved	Reserved
7	GPIO172	01E8	00000000	GPIO172	GPIO172	Reserved	Reserved	Reserved
8	GPIO173	01EC	00000000	GPIO173	GPIO173	Reserved	Reserved	Reserved
9	GPIO174	01F0	00000000	GPIO174	GPIO174	Reserved	Reserved	Reserved
165	GPIO175	01F4	00000000	GPIO175	GPIO175	32KHZ_OUT	Reserved	Reserved
32	GPIO200	0200	00001000	ADC0	GPIO200	ADC0	Reserved	Reserved
34	GPIO201	0204	00001000	ADC1	GPIO201	ADC1	Reserved	Reserved
37	GPIO202	0208	00001000	ADC2	GPIO202	ADC2	Reserved	Reserved
39	GPIO203	020C	00001000	ADC3	GPIO203	ADC3	Reserved	Reserved
41	GPIO204	0210	00001000	ADC4	GPIO204	ADC4	Reserved	Reserved
43	GPIO205	0214	00001000	ADC5	GPIO205	ADC5	Reserved	Reserved
45	GPIO206	0218	00001000	ADC6	GPIO206	ADC6	Reserved	Reserved
47	GPIO207	021C	00001000	ADC7	GPIO207	ADC7	Reserved	Reserved
33	GPIO210	0220	00001000	ADC8	GPIO210	ADC8	Reserved	Reserved
35	GPIO211	0224	00001000	ADC9	GPIO211	ADC9	Reserved	Reserved
38	GPIO212	0228	00001000	ADC10	GPIO212	ADC10	Reserved	Reserved

TABLE 24-4: PIN CONTROL MUXING TABLE 3

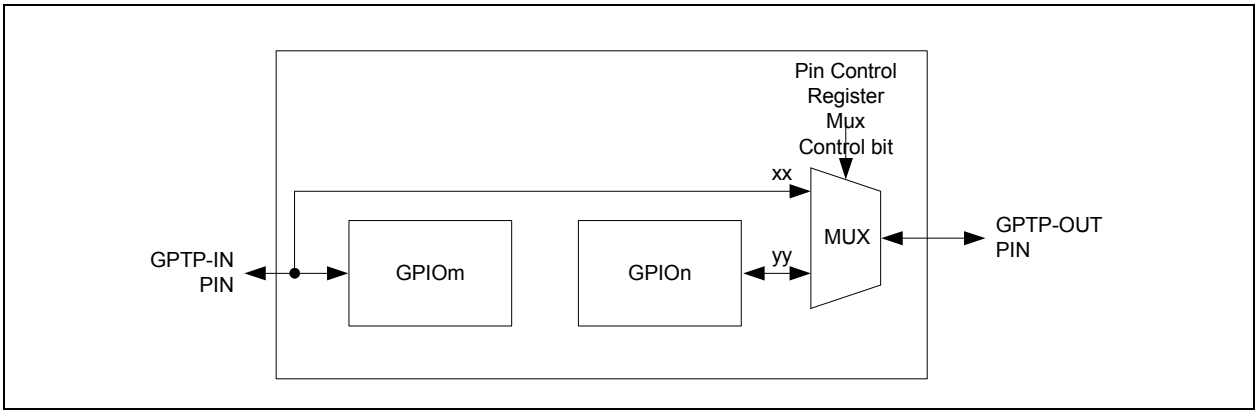
Pin Ref. Number	GPIO Name (Octal)	Pin Control Reg. Offset (Hex)	Pin Control Reg. POR Value (Hex)	POR Default Signal Function	Mux Control = 00	Mux Control = 01	Mux Control = 10	Mux Control = 11
40	GPIO213	022C	00001000	ADC11	GPIO213	ADC11	Reserved	Reserved
42	GPIO214	0230	00001000	ADC12	GPIO214	ADC12	Reserved	Reserved
44	GPIO215	0234	00001000	ADC13	GPIO215	ADC13	Reserved	Reserved
46	GPIO216	0238	00001000	ADC14	GPIO216	ADC14	Reserved	Reserved
48	GPIO217	023C	00001000	ADC15	GPIO217	ADC15	Reserved	Reserved
119	GPIO220	0240	00000000	GPIO220	GPIO220	VIN	Reserved	Reserved
65	GPIO222	0248	00000000	GPIO222	GPIO222	Reserved	Reserved	Reserved
67	GPIO223	024C	00000000	GPIO223	GPIO223	Reserved	Reserved	Reserved
69	GPIO224	0250	00000000	GPIO224	GPIO224	Reserved	Reserved	Reserved
71	GPIO230	0260	00000000	GPIO230	GPIO230	ECGP_SCLK	Reserved	Reserved
73	GPIO231	0264	00000000	GPIO231	GPIO231	ECGP_SOUT	Reserved	Reserved
75	GPIO233	026C	00000000	GPIO233	GPIO233	ECGP_SIN	Reserved	Reserved
16	GPIO234	0270	00001000	VCI_IN4#	GPIO234	VCI_IN4#	Reserved	Reserved
17	GPIO235	0274	00001000	VCI_IN5#	GPIO235	VCI_IN5#	Reserved	Reserved

24.8 GPIO Pass-Through Ports

GPIO Pass-Through Ports (GPTP) can multiplex two general purpose I/O pins as shown in Figure 24-3. GPIO Pass-Through Ports connect the GPTP-IN pin to the GPTP-OUT pin. The GPTP are sequentially assigned values 0:7. The GPTP port assignment have no relation to the GPIO Indexing assignments. The GPTP ports are controlled by the Mux Control bits in the Pin Control Register associated with the GPTP-OUT signal function.

In order to enable the GPTP Pass-Through Mode, the GPTP-IN (GPIOm in Figure 24-3) Pin Control Register must assign Mux Control = 00 (GPIO) and the GPIO Direction bit = 0 (input); the GPTP-OUT (GPIOn in Figure 24-3) Pin Control Register must assign Mux Control = the GPTP_OUT signal function and GPIO Direction bit = 1 (output). The Mux Control = GPTP-OUT signal function can differ from pin to pin See Section 24.6, "GPIO Indexing," on page 391.

FIGURE 24-3: GPIO PASS-THROUGH PORT EXAMPLE



Note:

- The Pin Control Register Mux Control fields shown in Figure 24-3 are illustrated as ‘xx’ and ‘yy’ because this figure is an example, it does not represent the actual GPIO multiplexing configuration. The GPIO Multiplexing tables in this chapter must be used to determine the correct values to use to select between a GPIO and the pass-through.
- When Pass-Through Mode is enabled, the GPIOn output is disconnected from the GPIOn pin and the GPIOm pin signal appears on GPIOn pin. Note that in this case the GPIOm input register still reflects the state of the GPIOm pin.

© 2012-2018 Microchip Technology Inc.

24.9 Registers

The GPIO Interface Registers include GPIO Input Registers, GPIO Output Registers and a Pin Control Register for each signal function. The GPIO Interface has its own Logical Device Number and Base Address as indicated in Table 24-5.

Table 24-6 is a register summary for the GPIO Interface. Each EC address is indicated as an SPB Offset from its AHB base address.

TABLE 24-5: GPIO Interface BASE ADDRESS TABLE

GPIOs Blocks	LDN from (Table 4-2 on page 59)	AHB Base Address
GPIOs	31h	F0_C400h

TABLE 24-6: GPIO Interface REGISTER SUMMARY

	EC Interface			
Register Name	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
Pin Control Register	000h - 200h	0-3	R/W	
GPIO[000:036] Output Register	280h	0-3	R/W	
GPIO[040:076] Output Register	284h	0-3	R/W	
GPIO[100:136] Output Register	288h	0-3	R/W	
GPIO[140:176] Output Register	28Ch	0-3	R/W	
GPIO[200:236] Output Register	290h	0-3	R/W	
GPIO[000:036] Input Register	300h	0-3	R/W	
GPIO[040:076] Input Register	304h	0-3	R/W	
GPIO[100:136] Input Register	308h	0-3	R/W	
GPIO[140:176] Input Register	30Ch	0-3	R/W	
GPIO[200:236] Input Register	310h	0-3	R/W	

APPLICATION NOTE: Bit31 in the five GPIO input registers (GPIO[200:236] Input Register, GPIO[140:176] Input Register, GPIO[100:136] Input Register, GPIO[040:076] Input Register, GPIO[000:036] Input Register) is a single bit register that can be set or cleared by software. It is provided to enable the use of the NORM instruction in the ARC instruction set in order to quickly find the first set or cleared bit in the register. Setting Bit31 to '0b' makes the NORM instruction find the first set bit; setting Bit31 to '1b' makes the NORM instruction find the first cleared bit. For example, if Bit31 of GPIO[000:036] Input Register is '0b' and Bit17 is the highest numbered bit position for which a GPIO signal is set high, then the instruction sequence:

LDR0,[GPIO[000:036] Input Register]

NORMR0,R0

will return the value 17 in ARC register R0.

24.9.1 PIN CONTROL REGISTER

TABLE 24-7: PIN CONTROL REGISTER

Offset	See Note 24-1			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:25	RESERVED	RES	-	-
24	GPIO input from pad On reads, Bit [24] reflects the state of GPIO input from the pad regardless of setting of Bit [10].	R	Note 24-1	nSYS_RST
23:17	RESERVED	RES	-	-
16	Alternative GPIO Data The Alternative GPIO Data bit can be used to determine the level on the GPIO pin when configured as an output depending on the state of the Output GPIO Write Enable (see also Table 24-8). If Output GPIO Write Enable = 1, the GPIO[x] output is unaffected by this bit. If Output GPIO Write Enable = 0: 0 = GPIO[x] out = '0' 1 = GPIO[x] out = '1' Reads of the Alternative GPIO Data bit return the value on the pin.	R/W	Note 24-1	nSYS_RST
15:14	RESERVED	RES	-	-
13:12	Mux Control The Mux Control field determines the active signal function for a pin. 00 = GPIO Function Selected 01 = Signal Function 1 Selected 10 = Signal Function 2 Selected 11 = Signal Function 3 Selected **Note:** The output buffer should be configured for push-pull operation via the Output Buffer Type bit of this register when the alternate function is enabled, unless otherwise noted.	R/W	Note 24-1	nSYS_RST
11	Polarity 0 = Non-inverted 1 = Inverted When the Polarity bit is set to '1' and the Mux Control bits are greater than '00,' the selected signal function outputs are inverted and Interrupt Detection (int_det) sense defined in Table 24-9, "Edge Enable and Interrupt Detection Bits Definition" is inverted. When the Mux Control field selects the GPIO signal function (Mux = '00'), the Polarity bit does not effect the output. Regardless of the state of the Mux Control field and the Polarity bit, the state of the pin is always reported without inversion in the GPIO input register. See FIGURE 24-1: GPIO Interface Block Diagram on page 389	R/W	Note 24-1	nSYS_RST

© 2012-2018 Microchip Technology Inc.

TABLE 24-7: PIN CONTROL REGISTER (CONTINUED)

Offset	See Note 24-1			
Bits	**Description**	**Type**	**Default**	**Reset Event**
10	Output GPIO Write Enable Every GPIO has two mechanisms to set a GPIO data output: Output GPIO Bit and Alternative GPIO Data bit. **Note:** See Section 24.9.2, "GPIO Output Registers," on page 399 for a description of the GPIO Output registers. The Output GPIO Write Enable bit is used to determine the source of the GPIO output (Table 24-8). 0 = Alternative GPIO Data write enabled When this bit is zero the Alternative GPIO Data write is enabled and the Output GPIO is disabled. 1 = Output GPIO enable When this bit is one the Alternative GPIO Data write is disabled and the Output GPIO is enabled. **Note:** See description in Section 24.10, "Programmer's Notes," on page 404.	R/W	Note 24-1	nSYS_RST
9	GPIO Direction 0 = Input 1 = Output The GPIO Direction bit controls the buffer direction only when the Mux Control field is '00' selecting the pin signal function to be GPIO. When the Mux Control field is greater than '00' (i.e., a non-GPIO signal function is selected) the GPIO Direction bit has no affect and the selected signal function logic directly controls the pin direction.	R/W	Note 24-1	nSYS_RST
8	Output Buffer Type 0 = Push-Pull 1 = Open Drain **Note:** Unless explicitly stated otherwise, pins with (I/O/OD) or (O/OD) in their buffer type column in the tables in are compliant with the following Programmable OD/PP Multiplexing Design Rule: Each compliant pin has a programmable open drain/push-pull buffer controlled by the Output Buffer Type bit in the associated Pin Control Register. The state of this bit controls the mode of the interface buffer for all selected functions, including the GPIO function.	R/W	Note 24-1	nSYS_RST
7	Edge Enable (edge_en) 0 = Edge detection disabled 1 = Edge detection enabled **Note:** See Table 24-9, "Edge Enable and Interrupt Detection Bits Definition" and Table 24-10, "GPIO Interrupt Programming Procedure". **Note:** In order to put the pin in its lowest power state, the Edge Enable bit should be set to '0', and the Interrupt Detection (int_det) field set to 100b. This combination ensures that no interrupt will be generated and that no wakeup function will be enabled.	R/W	Note 24-1	nSYS_RST

TABLE 24-7: PIN CONTROL REGISTER (CONTINUED)

Offset	See Note 24-1			
Bits	**Description**	**Type**	**Default**	**Reset Event**
6:4	Interrupt Detection (int_det) The interrupt detection bits determine the event that generates a GPIO event. **Note:** See Table 24-9, "Edge Enable and Interrupt Detection Bits Definition" and Table 24-10, "GPIO Interrupt Programming Procedure".	R/W	Note 24-1	nSYS_RST
3:2	Power Gating Signals The GPIO pin will be tristated when the selected power well is off. 00 = VTR 01 = VCC 10 = RESERVED 11 = RESERVED	R/W	Note 24-1	nSYS_RST
1:0	PU/PD (PU_PD) These bits are used to enable an internal pull-up. 00 = None 01 = Pull Up Enabled 10 = Pull Down Enabled 11 = None	R/w	Note 24-1	nSYS_RST

Note 24-1 see Section 24.7, "Pin Multiplexing Control," on page 391 for the offset and default values for each Pin Control Register.

TABLE 24-8: GPIO OUTPUT SOURCE SELECTION

Output GPIO Write Enable	GPIO Output Register (Note 24-2)	Alternative GPIO Data	Description
0	READ-ONLY	READ/WRITE	State of the Alternative GPIO Data bit determines the state of the GPIO output pin.
1	READ/WRITE	READ-ONLY	State of the appropriate bit in the GPIO Output Registers determines the state of the GPIO output pin.

Note 24-2 see Section 24.9.2, "GPIO Output Registers," on page 399.

TABLE 24-9: EDGE ENABLE AND INTERRUPT DETECTION BITS DEFINITION

D7	D6	D5	D4	Selected Function
0	0	0	0	Low Level Sensitive
0	0	0	1	High Level Sensitive
0	0	1	0	Reserved
0	0	1	1	Reserved
0	1	0	0	Interrupt events are disabled
0	1	0	1	Reserved
0	1	1	0	Reserved
0	1	1	1	Reserved
1	1	0	1	Rising Edge Triggered
1	1	1	0	Falling Edge Triggered
1	1	1	1	Either edge triggered

© 2012-2018 Microchip Technology Inc.

Note: Only edge triggered interrupts can wake up the 20 MHz Oscillator. The GPIO must be configured for edge-triggered interrupts (Interrupt Detection set to 101b - 111b), edge-triggered interrupts must be enabled (Edge Enable set to 1b) and the GPIO interrupt must be enabled in the interrupt aggregator to generate a wake event.

APPLICATION NOTE: to prevent any adverse affects from spurious interrupts when changing the Edge Enable (edge_en) and Interrupt Detection (int_det) bits, follow the steps defined in Table 24-10, "GPIO Interrupt Programming Procedure".

TABLE 24-10: GPIO INTERRUPT PROGRAMMING PROCEDURE

	Step	Description
1.	Disable	Disable the ARC interrupts to prevent false ISR calls
2.	Configure	Configure the appropriate GPIO interrupts using the Edge Enable (edge_en) and Interrupt Detection (int_det) bits
3.	Sync	Execute an ARC Sync instruction to allow I/O operations to complete before executing subsequent instructions
4.	Clear	Clear the appropriate GPIO interrupt source bits in the EC Interrupt Aggregator
5.	Enable	Enable the ARC interrupts

24.9.2 GPIO OUTPUT REGISTERS

The following sections show the register positions for accessing all possible GPIOs, whether the GPIO is implemented or not. See Section 24.7, "Pin Multiplexing Control" in order to determine which GPIOs are present in the MEC1632.

See also Table 24-8, “GPIO Output Source Selection,” on page 398.

24.9.2.1 Output GPIO[000:036]

TABLE 24-11: GPIO[000:036] OUTPUT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	See Table 24-5 on page 395 and Table 24-6 on page 395					32	**EC SIZE**	
POWER	VTR					0000_0000h	nSYS_RST **DEFAULT**	
	EC SPB							
BIT	**D31**	**D30**	**D29**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	See Table 24-12 on page 399							
BIT NAME	GPIO[000:036] Output							

TABLE 24-12: BIT DEFINITIONS GPIO[000:036] OUTPUT REGISTER

Bit	EC Type	Signal Name	Description
[7:0]	R/W	GPIO[007:000]	Each bit monitors the corresponding pins status.
[15:8]	R/W	GPIO[017:010]	
[23:16]	R/W	GPIO[027:020]	
[30:24]	R/W	GPIO[036:030]	
31	R	Reserved	This bit is always reserved

24.9.2.2 Output GPIO[040:076]

TABLE 24-13: GPIO[040:076] OUTPUT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	See Table 24-5 on page 395 - Table 24-6 on page 395			32			**EC SIZE**	
POWER	VTR			0000_000h			nSYS_RST **DEFAULT**	
	EC SPB							
BIT	**D31**	**D30**	**D29**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	See Table 24-14 on page 400							
BIT NAME	GPIO[040:076] Output							

TABLE 24-14: BIT DEFINITIONS GPIO[040:076] OUTPUT REGISTER

Bit	EC Type	Signal Name	Description
[7:0]	R/W	GPIO[047:040]	Each bit monitors the corresponding pins status
[15:8]	R/W	GPIO[057:050]	
[23:16]	R/W	GPIO[067:060]	
[30:24]	R/W	GPIO[076:070]	
31	R	Reserved	This bit is always reserved

24.9.2.3 Output GPIO[100:136]

TABLE 24-15: GPIO[100:136] OUTPUT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	See Table 24-5 on page 395 - Table 24-6 on page 395			32			**EC SIZE**	
POWER	VTR			0000_0000h			nSYS_RST **DEFAULT**	
	EC SPB							
BIT	**D31**	**D30**	**D29**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	See Table 24-16 on page 400							
BIT NAME	GPIO[100:136] Output							

TABLE 24-16: BIT DEFINITIONS GPIO[100:136] OUTPUT REGISTER

Bit	EC Type	Signal Name	Description
[7:0]	R/W	GPIO[107:100]	Each bit monitors the corresponding pins status.
[15:8]	R/W	GPIO[117:110]	
[23:16]	R/W	GPIO[127:120]	
[30:24]	R/W	GPIO[136:130]	
31	R	Reserved	This bit is always reserved

© 2012-2018 Microchip Technology Inc.

24.9.2.4 Output GPIO[140:176]

TABLE 24-17: GPIO[140:176] OUTPUT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	See Table 24-5 on page 395 - Table 24-6 on page 395				32		**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
	EC SPB							
BIT	**D31**	**D30**	**D29**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	See Table 24-18 on page 401							
BIT NAME	GPIO[140:176] Output							

TABLE 24-18: GPIO[140:176] OUTPUT REGISTER

Bit	EC Type	Signal Name	Description
[7:0]	R/W	GPIO[147:140]	Each bit monitors the corresponding pins status.
[15:8]	R/W	GPIO[157:150]	
[23:16]	R/W	GPIO[167:160]	
[30:24]	R	GPIO[176:170]	
31	R	Reserved	This bit is always reserved

24.9.2.5 Output GPIO[200:236]

TABLE 24-19: GPIO[200:236] OUTPUT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	See Table 24-5 on page 395 - Table 24-6 on page 395				32		**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
	EC SPB							
BIT	**D31**	**D30**	**D29**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	See Table 24-20 on page 401							
BIT NAME	GPIO[200:236] Output							

TABLE 24-20: GPIO[200:236] OUTPUT REGISTER

Bit	EC Type	Signal Name	Description
[7:0]	R/W	GPIO[207:200]	Each bit monitors the corresponding pins status.
[15:8]	R/W	GPIO[217:210]	
[23:16]	R/W	GPIO[227:220]	
[30:24]	R/W	GPIO[236:230]	
31	R	Reserved	This bit is always reserved

24.9.3 GPIO INPUT REGISTERS

24.9.3.1 Overview

The MEC1632 GPIO Interface includes four GPIO Input Registers which are always active as illustrated in FIGURE 24-1: GPIO Interface Block Diagram on page 389. The GPIO Input Registers can always be used to read the state of a pin (excluding the NORM Exception Bits), even when the pin is in an output mode and/or when a signal function other than the GPIO signal function is selected; i.e., the Pin Control Register Mux Control bits are not equal to '00.'

24.9.3.2 NORM Exception Bits

There can be up to 31 GPIO signal function names in each of the GPIO Input Registers (Bits D0 - D30). Bit D31 in each of these registers, the NORM Exception Bits, are single bit registers that can be set or cleared by software. The NORM Exception Bits are provided to enable the use of the NORM instruction in the ARC instruction set in order to quickly find the first set or cleared bit in the register. Setting Bit D31 to '0b' makes the NORM instruction find the first set bit; setting Bit D31 to '1b' makes the NORM instruction find the first cleared bit. For example, if Bit D31 of GPIO[000:036] Input Register is '0b' and Bit17 is the highest numbered bit position for which a GPIO signal is set high, then the following instruction sequence will return the value 17 in ARC register R0.

```
LD    R0,[GPIO[000:036] Input Register]
NORM  R0,R0
```

24.9.3.3 Input GPIO[000:036]

TABLE 24-21: GPIO[000:036] INPUT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	See Table 24-5 on page 395 - Table 24-6 on page 395				32		**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
	EC SPB							
BIT	**D31**	**D30**	**D29**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	GPIO[000:036] INPUT						

24.9.3.4 Input GPIO[040:076]

TABLE 24-22: GPIO[040:076] INPUT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	See Table 24-5 on page 395 - Table 24-6 on page 395				32		**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
	EC SPB							
BIT	**D31**	**D30**	**D29**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	GPIO[040:076] INPUT						

© 2012-2018 Microchip Technology Inc.

24.9.3.5 Input GPIO[100:136]

TABLE 24-23: GPIO[100:136] INPUT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	See Table 24-5 on page 395 - Table 24-6 on page 395				32		**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
	EC SPB							
BIT	**D31**	**D30**	**D29**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	GPIO[100:136] INPUT						

24.9.3.6 Input GPIO[140:176]

TABLE 24-24: GPIO[140:176] INPUT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	See Table 24-5 on page 395 - Table 24-6 on page 395				32		**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
	EC SPB							
BIT	**D31**	**D30**	**D29**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	GPIO[140:176] INPUT						

24.9.3.7 Input GPIO[200:236]

TABLE 24-25: GPIO[200:236] INPUT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	See Table 24-5 on page 395 - Table 24-6 on page 395				32		**EC SIZE**	
POWER	VTR				0000_0000h		nSYS_RST **DEFAULT**	
	EC SPB							
BIT	**D31**	**D30**	**D29**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	NORM	GPIO[200:236] INPUT						

© 2012-2018 Microchip Technology Inc.

24.10 Programmer's Notes

As mentioned in Section 24.1 there are two ways to access GPIO input and output pin signals. This note aims to describe these access mechanisms and their relative merits. The two schemes can be employed concurrently.

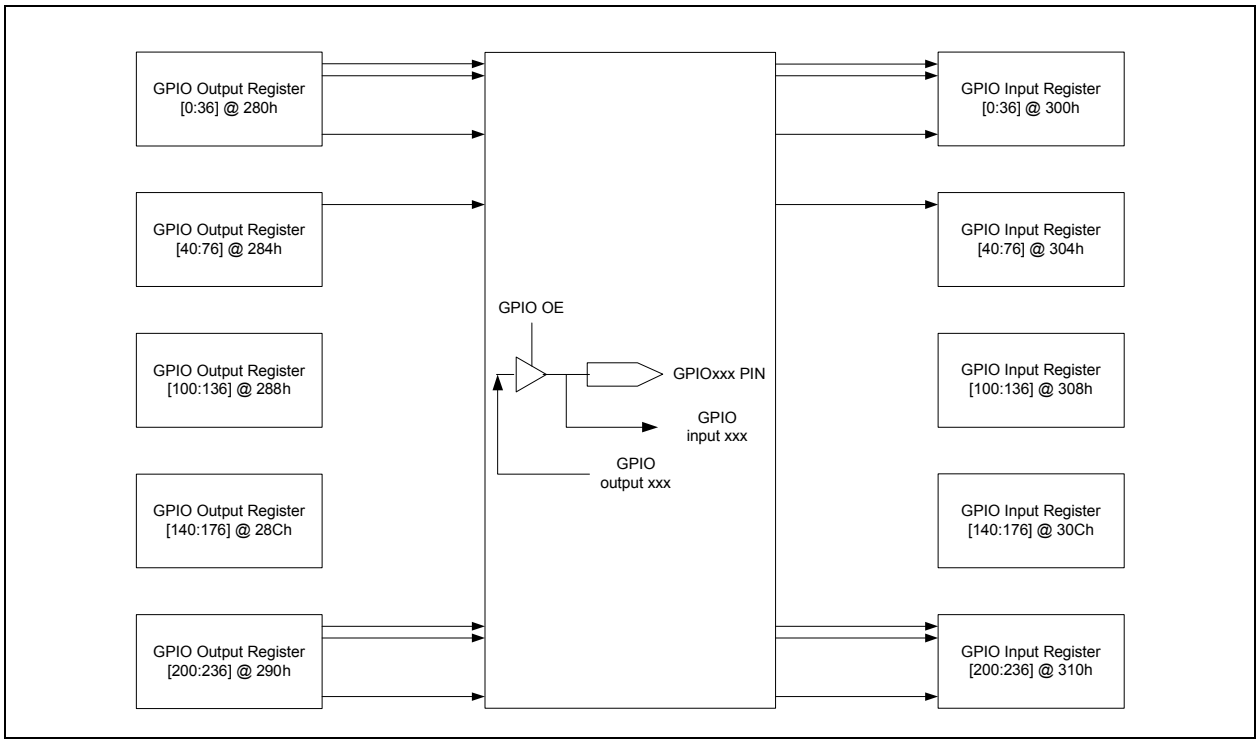
In the following description, the terms pins, ports, lines, signals are used interchangeably.

24.10.1 ACCESSING GPIO - MECHANISM 1

24.10.1.1 Overview

In the legacy approach, referred to hereafter as Mechanism 1, GPIO lines are grouped into 5 logical groups of 32 each. Associated with each group is an output register that controls the value to be driven out to the GPIO lines in the group and an input register whose value reflects the value seen on the group's GPIO lines' input side. Each non-reserved bit in the input/output register corresponds to a GPIO line. Since the number of GPIO lines that are implemented is less than that indicated by the logical labeling (0 to 236 in octal notation), some register bits are reserved. Figure 24-4 illustrates the arrangements.

FIGURE 24-4: ACCESSING GPIOS - SCHEME 1 (LEGACY SCHEME)



24.10.1.2 Access Mechanism

To change the output value of a particular GPIO line, the software first reads the GPIO Output Register associated with the group the line belongs to. The output register's value is logically (bitwise) ORed with a 32-bit value formed by setting the bit corresponding the GPIO line to the desired value while keeping all other bits zero. The result is then written to the same output register. Note that the value on GPIO output is as specified by its register bit value and is not affected by setting of Polarity in the Pin Control Register, which affects only the polarity of Alternate Function signal multiplexed on the GPIO pin. GPIO lines belonging to the same group can be changed together in one step.

To monitor the value on the input side of GPIO buffer, read the GPIO Input Register associated with the group the GPIO line belongs to. For example, GPIO102 corresponds to Bit 2 in GPIO Input Register at offset 308h. Bit value can be extracted by applying appropriate mask and/or shift. In the current example, the mask would be 0x4; the shift, >>2.

Bit 31 in each GPIO Input Register, the NORM Exception Bits, is not associated with a GPIO line, rather it is to be used in conjunction with the ARC's NORM instruction to quickly locate the first bit in the register that is set or cleared. See Section 24.9.3.2 for details.

© 2012-2018 Microchip Technology Inc.

24.10.1.3 Applicability

Scheme 1 is simple and allows updates to multiple GPIO lines with one register update. However, when there is more than one software process accessing GPIO lines that share the same register, there is the potential for conflicts. In such cases the test-and-set sequence described in Section 24.10.1.2 must be an atomic operation. Since there is no support for this, software must implement a synchronization mechanism, e.g., mutex, semaphore, or spin lock, to serialize accesses.

24.10.2 ACCESSING GPIO - MECHANISM 2

24.10.2.1 Overview

In this mechanism each GPIO port is individually - as opposed to in groups of 32 as in Mechanism 1 -accessed via its Pin Control Register. The mechanism is enabled on a per-port basis. Once enabled, it overrides Mechanism 1, i.e., a write to a GPIO Output Register will not affect ports for which Mechanism 2 have been enabled. The mechanism can be dynamically enabled and disabled.

24.10.2.2 Access Mechanism

Figure 24-2 shows how relevant bits in the Pin Control Register are used to effect GPIO accesses in Mechanism 2.

To change a GPIO output port, the Output GPIO Write Enable bit its Pin Control Register is set to enable Mechanism 2, and the Alternative GPIO Data bit is set to the port's desired output value. Both bits can be updated with a single register write. As long as Output GPIO Write Enable is cleared ('0'), Mechanism 2 is in effect. This bit must be set ('1') to switch to Mechanism 1.

The state of GPIO input port is reflected by the GPIO input from pad bit in the Pin Control Register. This holds true regardless of whether Mechanism 2 is enabled.

The port's input, output, and programmed output values are available in the same Pin Control Register.

24.10.2.3 Applicability

Mechanism 2 provides accesses at the port level and hence greatly reduces the unnecessary sharing of GPIO ports among different processes or threads. Even when a port is shared, the per-port control eliminates the need for atomic test-and-set.

Unless the applications call for the ability to update more than one GPIO port at the same time, it is recommended that Mechanism 2 be used. Alternatively, the two mechanisms can be together, with Mechanism 1 be applied to ports that do not need exclusive accesses.

© 2012-2018 Microchip Technology Inc.

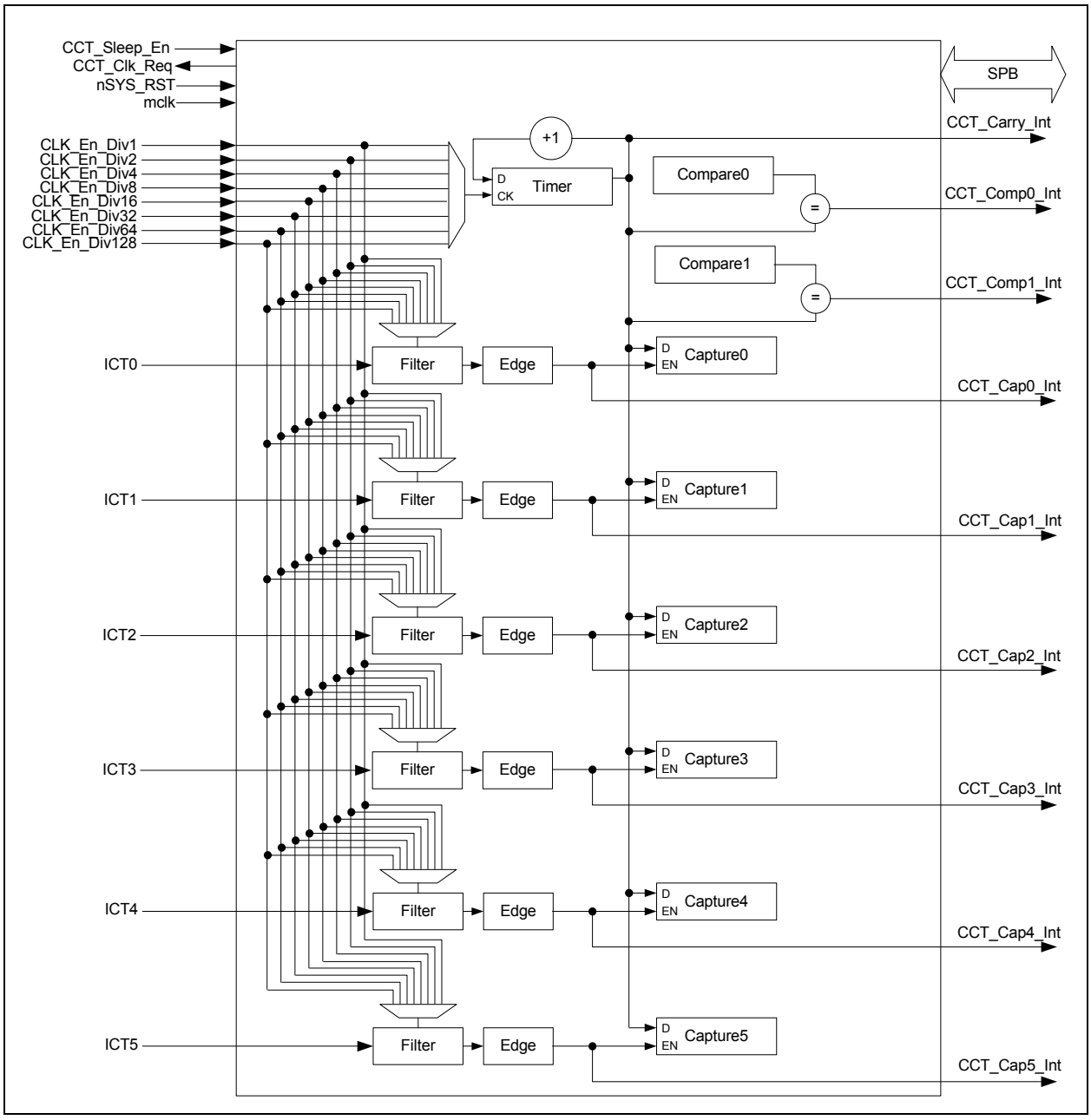
25.0 INPUT CAPTURE AND COMPARE TIMER

25.1 General Description

The Input Capture and Compare Timers block contains a 32-bit timer running at the main system clock frequency. The timer is free-running and is associated with six 32-bit capture registers and two compare registers. Each capture register can record the value of the free-running timer based on a programmable edge of its associated input pin. An interrupt can be generated for each capture register each time it acquires a new timer value. The timer can also generate an interrupt when it automatically resets and can additionally generate two more interrupts when the timer matches the value in either of two 32-bit compare registers.

25.2 Capture and Compare Timer Block Diagram

FIGURE 25-1: CAPTURE AND COMPARE TIMER BLOCK DIAGRAM



© 2012-2018 Microchip Technology Inc.

25.3 Block Diagram Signal List

TABLE 25-1: Input Capture and Compare Timer **SIGNAL LIST**

Signal Name	Direction	Description
SPB	I/O Bus	MEC1632 peripheral bus
MCLK	INPUT	Master MEC1632 clock
nSYS_RST	INPUT	Block reset signal
CLK_EN_DIV1	INPUT	MCLK clock enable
CLK_EN_DIV2	INPUT	MCLK/2 clock enable
CLK_EN_DIV4	INPUT	MCLK/4 clock enable
CLK_EN_DIV8	INPUT	MCLK/8 clock enable
CLK_EN_DIV16	INPUT	MCLK/16 clock enable
CLK_EN_DIV32	INPUT	MCLK/32 clock enable
CLK_EN_DIV64	INPUT	MCLK/64 clock enable
CLK_EN_DIV128	INPUT	MCLK/128 clock enable
ICT[5:0]	INPUT	External capture trigger signals (Note 25-1)
CCT_CARRY_INT	OUTPUT	Free-running timer wraparound interrupt
CCT_COMP[1:0]_INT	OUTPUT	Timer compare interrupts
CCT_CAP[5:0]_INT	OUTPUT	Timer capture interrupts
CCT_SLEEP_EN	INPUT	External enable/disable signal used to put the block in the lowest power consumption state. 0=No Sleep Requested. The block should operate as configured. 1=Sleep Requested. The block enters sleep mode. See Low Power Mode on page 408.
CCT_CLK_REQ	OUTPUT	This output indicates when this block requires this clock input. 0= MCLK can be turned 'off' when appropriate 1= MCLK is required to be 'on.'

Note 25-1 External capture trigger signal inputs ICT0 - ICT5 are identical to FAN_TACH0 - FAN_TACH5.

25.4 Power, Clocks and Reset

25.4.1 POWER DOMAIN

This block is powered by the VTR power supply.

25.4.2 CLOCKS

The timer in this unit is driven by mclk, the main system clock.

25.4.3 RESET

This block is reset on a nSYS_RST. On nSYS_RST the timer and all Capture and Compare registers are reset to their default values. The timer is also reset by the Free_Reset bit in the Capture and Compare Timer Control Register.

25.5 Interrupts

Interrupts from the Input Capture and Compare Timer block are routed to GIRQ23 Source Register of the Interrupt Aggregator. There are a total of nine interrupts from this block: one each for the six capture, one each for the two compare registers and one when the Free Running Timer wraps around. The interrupt signals are always generated by this block and can be queried or enabled through the Source and Enable registers in the Interrupt Aggregator.

© 2012-2018 Microchip Technology Inc.

25.6 Low Power Mode

This block is designed to conserve power when it is either sleeping or disabled. There are two ways to put the Capture and Compare block into a low power mode: Disable the Activate Bits or assert the CCT_SLEEP_EN signal to the Capture and Compare Timer block. The following table summarizes the Capture and Compare Timer behavior for each of these low power modes.

TABLE 25-2: BLOCK CLOCK GATING IN LOW POWER MODES

Activate Bits	CCT_SLEEP_En	Block Idle Status	CCT_CLK_REQ	State	Description
All 0	X	NOT IDLE	1	PREPARING to SLEEP	The core clock is required until the current transaction is completed and the Block is IDLE.
		IDLE	0	SLEEPING	The block is idle and the core clock can be stopped.
Any 1	0	X	1	NORMAL OPERATION	The block in neither disabled by firmware nor commanded to SLEEP
	1	NOT IDLE	1	PREPARING to SLEEP	The core clock is required until the current transaction is completed and the Block is IDLE.
		IDLE	0	SLEEPING	The block is idle and the core clock can be stopped.

25.7 Noise Filter

The noise filter uses Filter Clock (FCLK) to filter the signal on the ICTx pins. An external ICTx pin must remain in the same state for three FCLK ticks before the internal state changes. The Filter Bypass bit is used to bypass the noise filter. Each ICT input capture register can individually bypass the filter, but all ICT input capture registers that use the filter use the same Filter Clock.

- The signal ICT may be optionally only synchronized, or synchronized and filtered depending on the filter bypass bit.
- The minimum FCLK period must be at least 2X the duration of the ICT signal so that signal can be reliably captured in the bypass mode.
- The minimum FCLK period must be at least 4X the duration of the ICT signal so that signal can be reliably captured in the non-bypass mode.

25.8 Operation

25.8.1 INPUT CAPTURE

The Input Capture block consists of a free-running 32-bit timer and 6 capture registers. Each of the capture registers is associated with an input pin as well as an interrupt source bit in the Interrupt Aggregator: Table 25-3, "Pin Capture Interrupt Assignments" shows the assignment of pins to the Capture registers:

TABLE 25-3: PIN CAPTURE INTERRUPT ASSIGNMENTS

Capture Register	Capture Pin	Interrupt
Capture 0 Register	ICT0	CAPTURE 0, GIRQ23 Source Register
Capture 1 Register	ICT1	CAPTURE 1, GIRQ23 Source Register
Capture 2 Register	ICT2	CAPTURE 2, GIRQ23 Source Register
Capture 3 Register	ICT3	CAPTURE 3, GIRQ23 Source Register
Capture 4 Register	ICT4	CAPTURE 4, GIRQ23 Source Register
Capture 5 Register	ICT5	CAPTURE 5, GIRQ23 Source Register

© 2012-2018 Microchip Technology Inc.

The Capture registers store the current value of the Free Running timer whenever the associated input signal changes, according to the programmed edge detection. An interrupt is also generated to the EC. The Capture registers are read-only. The registers are updated every time an edge is detected. If software does not read the register before the next edge, the value is lost.

25.8.2 COMPARE INTERRUPT GENERATION

There are two 32-bit Compare registers. Each of these registers can independently generate an interrupt to the EC when the 32-bit free running Capture timer matches the contents of the Compare register.

25.9 Input Capture and Compare Timers Register Summary

There is one instance of the Input Capture and Compare Timer block implemented in the MEC1632.

TABLE 25-4: Input Capture and Compare Timer BASE ADDRESS TABLE

Input Capture and Compare Timer **Instance**	**LDN**	**AHB Base Address**
Input Capture and Compare Timer	2h	F0_0800h

TABLE 25-5: Input Capture and Compare Timer REGISTER SUMMARY

	EC Interface			
Register Name	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
Capture and Compare Timer Control Register	00h	3-0	R/W	
Capture Control 0 Register	04h	3-0	R/W	
Capture Control 1 Register	08h	3-0	R/W	
Free Running Timer Register	0Ch	3-0	R/W	
Capture 0 Register	10h	3-0	R	
Capture 1 Register	14h	3-0	R	
Capture 2 Register	18h	3-0	R	
Capture 3 Register	1Ch	3-0	R	
Capture 4 Register	20h	3-0	R	
Capture 5 Register	24h	3-0	R	
Compare 0 Register	28h	3-0	R/W	
Compare 1 Register	2Ch	3-0	R/W	

Note: The registers in this block with 32-bit values (Free Running Timer Register, Capture 0 Register, Capture 1 Register, Capture 2 Register, Capture 3 Register, Capture 4 Register, Capture 5 Register, Compare 0 Register, Compare 1 Register) should be read with 32-bit accesses. If these registers are read with multiple 8-bit or 16-bit accesses, the register values could change between the multiple accesses to the registers.

25.10 Detailed Register Descriptions

25.10.1 CAPTURE AND COMPARE TIMER CONTROL REGISTER

TABLE 25-6: CAPTURE AND COMPARE TIMER CONTROL REGISTER

HOST ADDRESS	n/a				n/a		**HOST SIZE**	
EC OFFSET	00h				32-bit		**EC SIZE**	
POWER	VTR				0000h		**VTR POR DEFAULT**	
BUS	EC SPB							
BYTE[3-2] BIT	**D31**	**D30**	**D29**	**...**		**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R/W	R/W
BIT NAME	Reserved						Compare Enable1	Compare Enable0
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/W	R/W	R/W	R	R/W	R/W	R/W
BIT NAME	Reserved	TCLK			Reserved	Free_ Reset	Free_ Enable	Activate

ACTIVATE

0=The timer block is powered down and all clocks are gated. (default).

1=The timer block is in a running state

FREE_ENABLE

Free-Running Timer Enable. This bit is used to start and stop the free running timer. This bit does not reset the timer count. The timer starts counting at 0000_0000h on reset and wraps around back to 0000_0000h after it reaches FFFF_FFFFh.

0=Timer is disabled. The Free Running Timer Register is writable.

1=Timer is enabled. The Free Running Timer Register is read-only.

Note: The Free_Enable bit is cleared after the RESET cycle is done. Firmware must poll the Free_Reset bit to determine when it is safe to re-enable the timer.

FREE_RESET

Free Running Timer Reset. This bit stops the timer and resets the internal counter to 0000_0000h. This bit does not affect the Free_Enable bit. This bit is self clearing after the timer is reset.

0=Normal timer operation

1=Timer reset

© 2012-2018 Microchip Technology Inc.

TCLK

This 3-bit field sets the clock source for the Free Running Counter (see Section 25.10.4, "Free Running Timer Register," on page 415). The available frequencies are shown in Table 25-7:

TABLE 25-7: FREE RUNNING TIMER CLOCK FREQUENCIES

Timer Clock Select	Frequency Selected
000	MCLK
001	MCLK/2
010	MCLK/4
011	MCLK/8
100	MCLK/16
101	MCLK/32
110	MCLK/64
111	MCLK/128

COMPARE ENABLE0

Compare Enable for Compare 0 Register. If this bit is 1, a match between the Compare 0 Register and the Free Running Timer Register will cause an interrupt to be generated. If this bit is 0, no interrupt will be generated.

COMPARE ENABLE1

Compare Enable for Compare 1 Register. If this bit is 1, a match between the Compare 1 Register and the Free Running Timer Register will cause an interrupt to be generated. If this bit is 0, no interrupt will be generated.

25.10.2 CAPTURE CONTROL 0 REGISTER

TABLE 25-8: CAPTURE CONTROL 0 REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	04h			32-bit			**EC SIZE**	
POWER	VTR			0000h			**VTR POR DEFAULT**	
BUS	EC SPB							
BYTE3 BIT	**D31**	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R	R	R/W	R/W	R/W
BIT NAME	FCLK_SEL3			Reserved		Filter Byp3	Capture_Edge3	
BYTE2 BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R	R	R/W	R/W	R/W
BIT NAME	FCLK_SEL2			Reserved		Filter Byp2	Capture_Edge2	
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R	R	R/W	R/W	R/W
BIT NAME	FCLK_SEL1			Reserved		Filter Byp1	Capture_Edge1	
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R	R	R/W	R/W	R/W
BIT NAME	FCLK_SEL0			Reserved		Filter Byp0	Capture_Edge0	

CAPTURE_EDGE0

Capture Timer value Edge Type Select. This field selects the edge type that triggers the capture of the free-running timer into Capture 0 Register. See Table 25-9.

TABLE 25-9: CAPTURE EDGE SELECTION

Capture Edge Select	Edge that Triggers Capture
00	Falling edges
01	Rising edges
10	Both rising and falling edges
11	Capture event disabled

FILTER BYP0

Filter Bypass permits ICT0 to bypass the noise filter and go directly into the timer. A 0 enables the filter and a 1 bypasses the filter.

FCLK_SEL0

Filter Clock Select, used to determine the clock source to the input filter. Available frequencies are shown in Table 25-10:

© 2012-2018 Microchip Technology Inc.

TABLE 25-10: FILTER CLOCK FREQUENCIES

Timer Clock Select	Frequency Selected
000	MCLK
001	MCLK/2
010	MCLK/4
011	MCLK/8
100	MCLK/16
101	MCLK/32
110	MCLK/64
111	MCLK/128

CAPTURE_EDGE1

Capture Timer value Edge Type Select. This field selects the edge type that triggers the capture of the free-running timer into Capture 1 Register. See Table 25-9.

FILTER BYP1

Filter Bypass permits ICT1 to bypass the noise filter and go directly into the timer. A 0 enables the filter and a 1 bypasses the filter.

FCLK_SEL1

Filter Clock Select, used to determine the clock source to the input filter. Available frequencies are shown in Table 25-10.

CAPTURE_EDGE2

Capture Timer value Edge Type Select. This field selects the edge type that triggers the capture of the free-running timer into Capture 2 Register. See Table 25-9.

FILTER BYP2

Filter Bypass permits ICT2 to bypass the noise filter and go directly into the timer. A 0 enables the filter and a 1 bypasses the filter.

FCLK_SEL2

Filter Clock Select, used to determine the clock source to the input filter. Available frequencies are shown in Table 25-10.

CAPTURE_EDGE3

Capture Timer value Edge Type Select. This field selects the edge type that triggers the capture of the free-running timer into Capture 3 Register. See Table 25-9.

FILTER BYP3

Filter Bypass permits ICT3 to bypass the noise filter and go directly into the timer. A 0 enables the filter and a 1 bypasses the filter.

FCLK_SEL3

Filter Clock Select, used to determine the clock source to the input filter. Available frequencies are shown in Table 25-10.

© 2012-2018 Microchip Technology Inc.

25.10.3 CAPTURE CONTROL 1 REGISTER

TABLE 25-11: CAPTURE CONTROL 1 REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	08h			32-bit			**EC SIZE**	
POWER	VTR			0000h			**VTR POR DEFAULT**	
BUS	EC SPB							
BYTE[3-2] BIT	**D31**	**D30**	**D29**	**...**		**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R	R	R/W	R/W	R/W
BIT NAME	FCLK_SEL5			Reserved		Filter Byp5	Capture_Edge5	
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R	R	R/W	R/W	R/W
BIT NAME	FCLK_SEL4			Reserved		Filter Byp4	Capture_Edge4	

CAPTURE_EDGE4

Capture Timer value Edge Type Select. This field selects the edge type that triggers the capture of the free-running timer into Capture 4 Register. See Table 25-9.

FILTER BYP4

Filter Bypass permits ICT4 to bypass the noise filter and go directly into the timer. A 0 enables the filter and a 1 bypasses the filter.

FCLK_SEL4

Filter Clock Select, used to determine the clock source to the input filter. Available frequencies are shown in Table 25-10.

CAPTURE_EDGE5

Capture Timer value Edge Type Select. This field selects the edge type that triggers the capture of the free-running timer into Capture 5 Register. See Table 25-9.

FILTER BYP5

Filter Bypass permits ICT5 to bypass the noise filter and go directly into the timer. A 0 enables the filter and a 1 bypasses the filter.

FCLK_SEL5

Filter Clock Select, used to determine the clock source to the input filter. Available frequencies are shown in Table 25-10.

© 2012-2018 Microchip Technology Inc.

25.10.4 FREE RUNNING TIMER REGISTER

TABLE 25-12: FREE RUNNING TIMER REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	0Ch			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			**VTR POR DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Free Running Timer[31:0]							

FREE RUNNING TIMER[31:0]

This register contains the current value of the Free Running Timer. A Capture Timer interrupt is signaled to the Interrupt Aggregator when this register transitions from FFFF_FFFFh to 0000_0000h.

When Free_Enable in Capture and Compare Timer Control Register is 1, this register is read-only. When Free_Enable is 0, this register may be written.

25.10.5 CAPTURE 0 REGISTER

TABLE 25-13: CAPTURE 0 REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	10h			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			**VTR POR DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Capture 0[31:0]							

CAPTURE 0[31:0]

This register saves the value copied from the Free Running timer on a programmed edge of ICT0.

25.10.6 CAPTURE 1 REGISTER

TABLE 25-14: CAPTURE 1 REGISTER

HOST ADDRESS	n/a				n/a		**HOST SIZE**	
EC OFFSET	14h				32-bit		**EC SIZE**	
POWER	VTR				0000_0000h		**VTR POR DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Capture 1[31:0]							

CAPTURE 1[31:0]

This register saves the value copied from the Free Running timer on a programmed edge of ICT1.

25.10.7 CAPTURE 2 REGISTER

TABLE 25-15: CAPTURE 2 REGISTER

HOST ADDRESS	n/a				n/a		**HOST SIZE**	
EC OFFSET	18h				32-bit		**EC SIZE**	
POWER	VTR				0000_0000h		**VTR POR DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Capture 2[31:0]							

CAPTURE 2[31:0]

This register saves the value copied from the Free Running timer on a programmed edge of ICT2.

25.10.8 CAPTURE 3 REGISTER

TABLE 25-16: CAPTURE 3 REGISTER

HOST ADDRESS	n/a				n/a		**HOST SIZE**	
EC OFFSET	1Ch				32-bit		**EC SIZE**	
POWER	VTR				0000_0000h		**VTR POR DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Capture 3[31:0]							

© 2012-2018 Microchip Technology Inc.

CAPTURE 3[31:0]

This register saves the value copied from the Free Running timer on a programmed edge of ICT3.

25.10.9 CAPTURE 4 REGISTER

TABLE 25-17: CAPTURE 4 REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	20h			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			**VTR POR DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Capture 4[31:0]							

CAPTURE 4[31:0]

This register saves the value copied from the Free Running timer on a programmed edge of ICT4.

25.10.10 CAPTURE 5 REGISTER

TABLE 25-18: CAPTURE 5 REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	24h			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			**VTR POR DEFAULT**	
BUS	EC SPB							
BIT	**D31**	**D30**	**D29**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Capture 5[31:0]							

CAPTURE 5[31:0]

This register saves the value copied from the Free Running timer on a programmed edge of ICT5.

25.10.11 COMPARE 0 REGISTER

TABLE 25-19: COMPARE 0 REGISTER

HOST ADDRESS	n/a					n/a		HOST SIZE
BUS	EC SPB							
POWER	VTR					0000_0000h		VTR POR DEFAULT
BIT	D31	D30	D29	…		D2	D1	D0
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Compare 0[31:0]							

COMPARE 0[31:0]

A Compare 0 interrupt is generated when this register matches the value in the Free Running Timer.

25.10.12 COMPARE 1 REGISTER

TABLE 25-20: COMPARE 1 REGISTER

HOST ADDRESS	n/a					n/a		HOST SIZE
EC OFFSET	2Ch					32-bit		EC SIZE
POWER	VTR					0000_0000h		VTR POR DEFAULT
BUS	EC SPB							
BIT	D31	D30	D29	…		D2	D1	D0
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Compare 1[31:0]							

COMPARE 1[31:0]

A Compare 1 interrupt is generated when this register matches the value in the Free Running Timer.

© 2012-2018 Microchip Technology Inc.

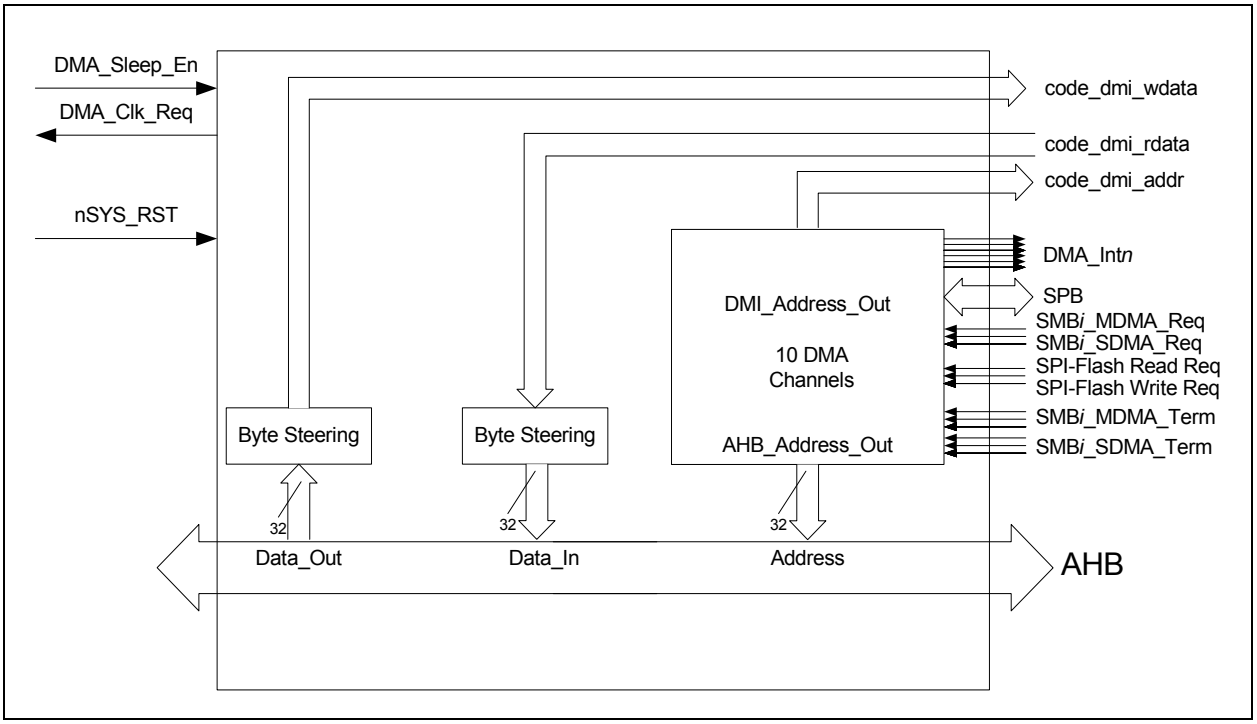
26.0 DMA CONTROLLER

26.1 General Description

This block describes the MEC1632 DMA controller. The DMA controller is designed to move data between the SMBus and SPI Flash controllers and the EC closely-coupled SRAM memory. There are ten independent channels that each move byte-wide data between the SMBus and SPI Flash controllers and the SRAM in either direction.

26.2 DMA Block Diagram

FIGURE 26-1: DMA BLOCK DIAGRAM



26.3 Block Diagram Signal List

TABLE 26-1: DMA Controller SIGNAL LIST

Signal Name	Direction	Description
AHB	I/O Bus	MEC1632 system bus
SPB	I/O Bus	MEC1632 peripheral bus
code_dmi_wdata, code_dmi_rdata, code_dmi_addr	I/O Bus	Direct Memory Interface (DMI) to ARC ICCM memory
MCLK	INPUT	Master MEC1632 clock
nSYS_RST	INPUT	Block reset signal
DMA_Int[9:0]	OUTPUT	DMA Interrupt signals
SMB[3:0]_MDMA_Req	INPUT	DMA request control from SMBus Master channel.
SMB[3:0]_SDMA_Req	INPUT	DMA request control from SMBus Slave channel.
SMB[3:0]_MDMA_Term	INPUT	DMA termination control from SMBus Master channel.
SMB[3:0]_SDMA_Term	INPUT	DMA termination control from SMBus Slave channel.

TABLE 26-1: DMA Controller SIGNAL LIST (CONTINUED)

Signal Name	Direction	Description
SPIFLASH_Read_Req	INPUT	DMA request control from SPI Flash receive Buffer register
SPIFLASH_Write_Req	INPUT	DMA request control from SPI Flash transmit Buffer register
DMA_SLEEP_EN	INPUT	External enable/disable signal used to put the block in the lowest power consumption state. 0=No Sleep Requested. The block should operate as configured. 1=Sleep Requested. The block enters sleep mode. See Low Power Mode on page 420.
DMA_CLK_REQ	OUTPUT	This output indicates when this block requires this clock input. 0= MCLK can be turned 'off' when appropriate 1= MCLK is required to be 'on.'

26.4 Power, Clocks and Reset

26.4.1 POWER DOMAIN

This block is powered by the VTR power supply with a separate Analog supply (AVDD).

26.4.2 CLOCKS

This block has three clock inputs: MCLK, the AHB bus clock enable, used for AHB transfers, and the EC clock enable, used for DMI transfers.

APPLICATION NOTE: The DMA Controller requires the EC clock in order to write into the EC SRAM. The EC, therefore, must not be in sleep mode at any time during a DMA transfer.

26.4.3 RESET

This block is reset on a nSYS_RST.

26.5 DMA Interrupts

Each channel of the DMA controller generates an interrupt event to the EC which indicate a DMA transfer is complete.

26.6 Low Power Mode

This block is designed to conserve power when it is either sleeping or disabled. There are two ways to put the DMA Controller into a low power mode: Disable all DMA channels via the Activate Bits or assert the DMA_SLEEP_EN signal to the DMA Controller. The following table summarizes the DMA Controller behavior for each of these low power modes.

TABLE 26-2: BLOCK CLOCK GATING IN LOW POWER MODES

Activate	DMA_SLEEP_EN	Busy	DMA_CLK_REQ	State	Description
All 0	X	X	0	INACTIVE	All channels are disabled
Any 1	0	X	1	NORMAL OPERATION	The block in neither disabled by firmware nor commanded to SLEEP
Any 1	1	1	1	PREPARING to SLEEP	The core clock is required until the current transaction is completed and the Block is IDLE.
Any 1	1	0	0	SLEEPING	The block is idle and the core clock can be stopped.

APPLICATION NOTE: The DMA controller will keep its Clock_Required status output asserted as long as the Run bit is on in any of the controller's channels. This means that the part will not enter a heavy sleep mode, and the main oscillator will stay on, as long as any controller channel is in the Run state. Before setting the SLEEP_FLAG in order to enter the heavy sleep states, firmware must make sure that the Run bits are clear in all channels. If a DMA channel is

© 2012-2018 Microchip Technology Inc.

required for any activity that is initiated by a Wake interrupt, the Wake interrupt Interrupt Service Routine should assert the Run bit in that channel.

For example, if the system uses SMB Controller 1 as a Slave SMBus port, and DMA Controller channel 7 as the SMB Slave DMA channel, then in order to enter a heavy sleep state, firmware must clear the Run bit in Channel 7 and enable Wake interrupts on the Clock and Data pins for SMB Controller 1 before asserting SLEEP_FLAG. The Interrupt Service Routines for Wake interrupts for the Clock and Data pins for SMB Controller 1 must assert the Run bit for DMA Channel 7, so that the SMB network layer controller can pull data out of the pin interface without additional processor intervention.

26.7 Operation

The MEC1632 features a ten channel DMA controller. The DMA controller can autonomously move data from I/O devices to and from EC local memory without EC intervention.

The DMA has the following characteristics:

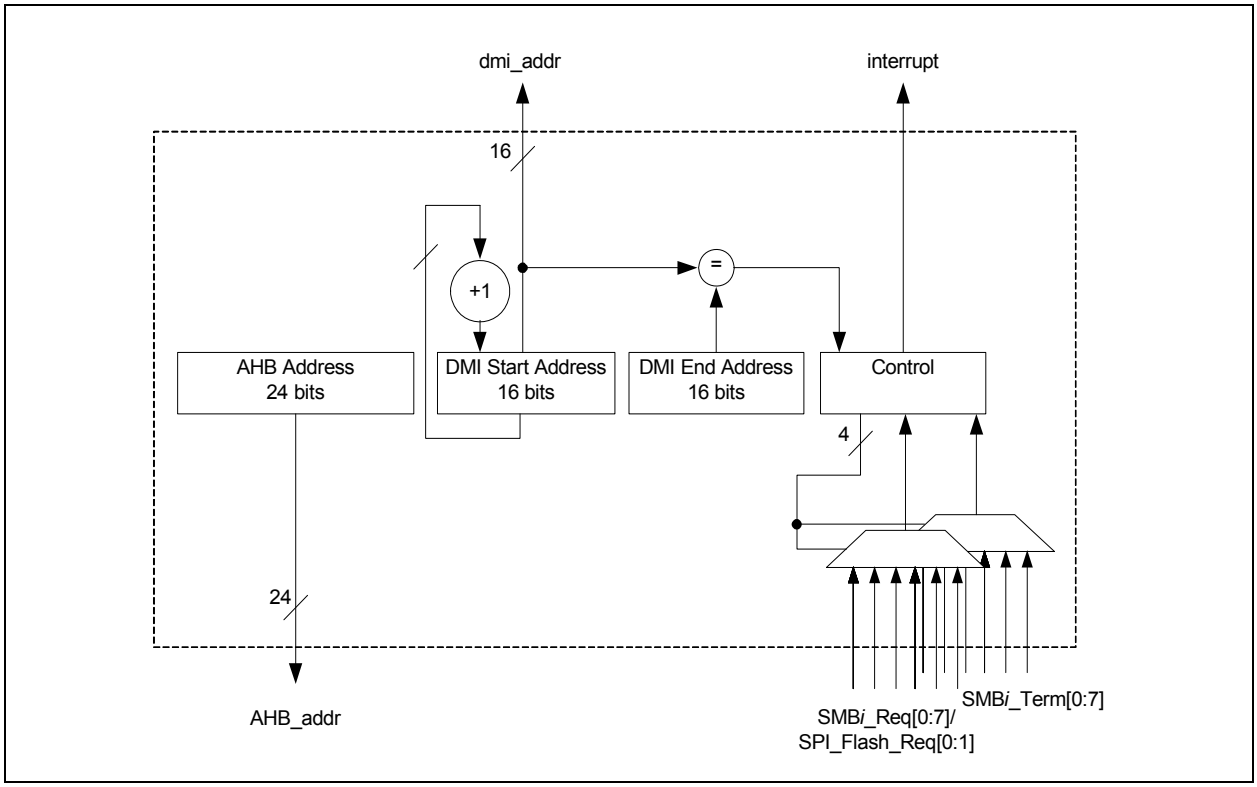
- Data is only moved 1 byte (8 bits) at a time
- Data only moves between devices on the AHB bus, or devices connected to an AHB bus bridge, and the EC SRAM. Since the SRAM is dual-ported to be both ICCM and DCCM, the DMA can be interpreted as moving data into and out of the DCCM as well as the ICCM.
- The number of DMA channels can be less than the number of I/O devices that can use the DMA for data transfers, so the DMA channels are shareable and can be assigned to any device.

The DMA controller is not designed to communicate with I/O devices with more than an 8-bit interface. The controller will accesses SRAM buffers only with incrementing addresses (that is, it cannot start at the top of a buffer, nor does it handle circular buffers automatically). The controller does not handle chaining (that is, automatically starting a new DMA transfer when one finishes).

26.7.1 DMA CHANNELS

Each DMA channel is capable of bi-directional data movement between an logical device and the EC closely-coupled memory. A single DMA channel is illustrated in Figure 26-2, "DMA Channel":

FIGURE 26-2: DMA CHANNEL



There are 10 possible logical devices that may connect to a DMA Controller channel. There are four SMBus controllers, each of which has a separate read request and write request. In addition, the SPI Flash controller has a read request and a write request. The SMBus controllers also provide a termination signal for each direction of each device. The SPI Flash device does not provide termination signals, so the two termination inputs corresponding to the SP Flash Read request and the SPI Flash Write request are always held to 0.

Based on the DIR in the channel's DMA Control Register, bytes are copied from a device's Receive Buffer to the ICCM/DCCM, or from the ICCM/DCCM to a device's Transmit Buffer. The AHB Address Register is programmed to be the address of the buffer required for the transfer. Software is responsible for insuring that this address is associated with the correct SRAM buffer defined by the DMI Start Address Register and by the correct SMBus as selected by the DEVICE field in the Control register.

The DMI Start Address is an offset from the base of the SRAM. It is an offset from both the base of the DCCM and the base of the ICCM, since the two memories are different aliases of a single dual-ported SRAM. This register is loaded into a 16-bit counter, which increments by 1 under state machine control in order to generate the current DMI address.

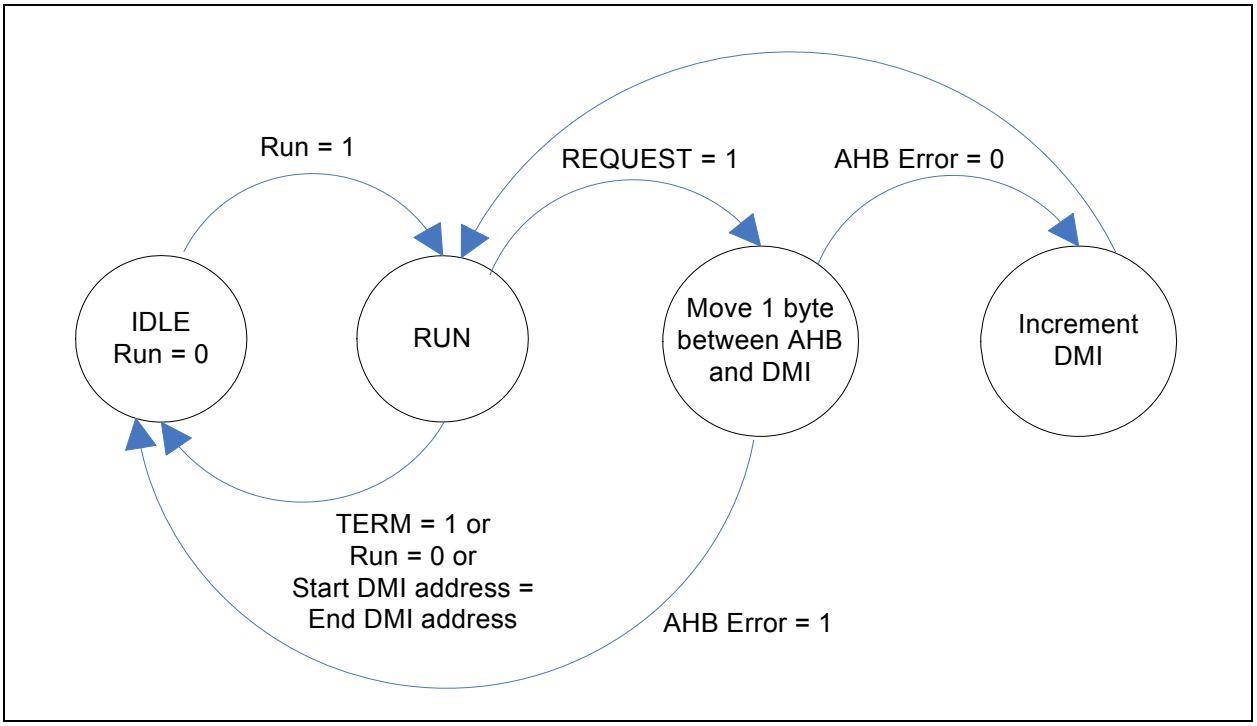
The End Address register contains the address one greater than the last byte to transfer. The DMA transfer terminates when the current DMI address equals the End Address. If the DMA channel is configured with the Start Address and End Address registers set to the same address, no bytes will be transferred. If the End Address is configured with an address that is less than the Start Address, the DMI Start Address register will wrap around from FFFFh to 0000h. The DMA Controller always sends 16 address bits to the SRAM. If fewer than 16 bits are required to address the ICCM/DCCM, then the SRAM will ignore the upper bits of the Start Address and End Address registers. For example, if the ICCM/DCCM is 16KB, bits 14 to 15 in the Start and End Address registers will be ignored by the SRAM.

The state machine that runs each DMA channel is illustrated in Figure 26-3, "Channel State Machine". When software sets the bit to 1, the state machine enters the RUN state and waits for an assertion of the selected REQUEST input. As long as REQUEST is asserted when the state machine is in the RUN state the DMA controller starts a 1-byte wide AHB bus transaction, in the direction defined by DIR. After the AHB transaction completes, the current DMI address counter is incremented by 1 and compared to the End Address register. If the current address is not equal to the End Address, the state machine returns to the RUN state. If the current address is equal to the End Address, or if the TERM input is

© 2012-2018 Microchip Technology Inc.

asserted, the DMA transaction is terminated and the state machine returns to the IDLE state. If there is an AHB bus error on the transfer between the AHB and the DMI, the DMI address counter increment is inhibited. The state machine returns to the IDLE state and the Status is set to AHB Bus Error.

FIGURE 26-3: CHANNEL STATE MACHINE



26.7.2 I/O DEVICES

The DMA Controller is configured to work with any of the four SMBus controllers and the SPI Flash controller in the MEC1632. Table 26-3, "DMA Device Selection" shows the mapping between the AHB Address and DEVICE and DIR fields for each I/O device.

TABLE 26-3: DMA DEVICE SELECTION

DEVICE	DIR	Device Name	Data AHB Address	Request Signals
0	0	SMBUS Controller 0	F0_184Ch (SMBus Slave Receive Buffer)	SMB0_SDMA_Req
0	1		F0_1848h (SMBus Slave Transmit Buffer)	
1	0		F0_1854h (SMBus Master Receive Buffer)	SMB0_MDMA_Req
1	1		F0_1850h (SMBus Master Transmit Buffer)	
2	0	SMBUS Controller 1	F0_18CCh (SMBus Slave Receive Buffer)	SMB1_SDMA_Req
2	1		F0_18C8h (SMBus Slave Transmit Buffer)	
3	0		F0_18D4h (SMBus Master Receive Buffer)	SMB1_MDMA_Req
3	1		F0_18D0h (SMBus Master Transmit Buffer)	

© 2012-2018 Microchip Technology Inc.

TABLE 26-3: DMA DEVICE SELECTION (CONTINUED)

DEVICE	DIR	Device Name	Data AHB Address	Request Signals
4	0	SMBUS Controller 2	F0_194Ch (SMBus Slave Receive Buffer)	SMB2_SDMA_Req
4	1		F0_1948h (SMBus Slave Transmit Buffer)	
5	0		F0_1954h (SMBus Master Receive Buffer)	SMB2_MDMA_Req
5	1		F0_1950h (SMBus Master Transmit Buffer)	
6	0	SPI Flash Controller	FF_3C04h (SPI RX_Data Register)	SPIFLASH_Read_Req
6	1		FF_3C03h (SPI TX_Data Register)	SPIFLASH_Write_Req
7	0/1	Reserved	-	–
8	0	SMBUS Controller 3	F0_19CCh (SMBus Slave Receive Buffer)	SMB3_SDMA_Req
8	1		F0_19C8h (SMBus Slave Transmit Buffer)	
9	0		F0_19D4h (SMBus Master Receive Buffer)	SMB3_MDMA_Req
9	1		F0_19D0h (SMBus Master Transmit Buffer)	

Note 26-1 See the DEVICE field in the DMA Control Register on page 428.

Note 26-2 See the DIR bit in the DMA Control Register on page on page 428.

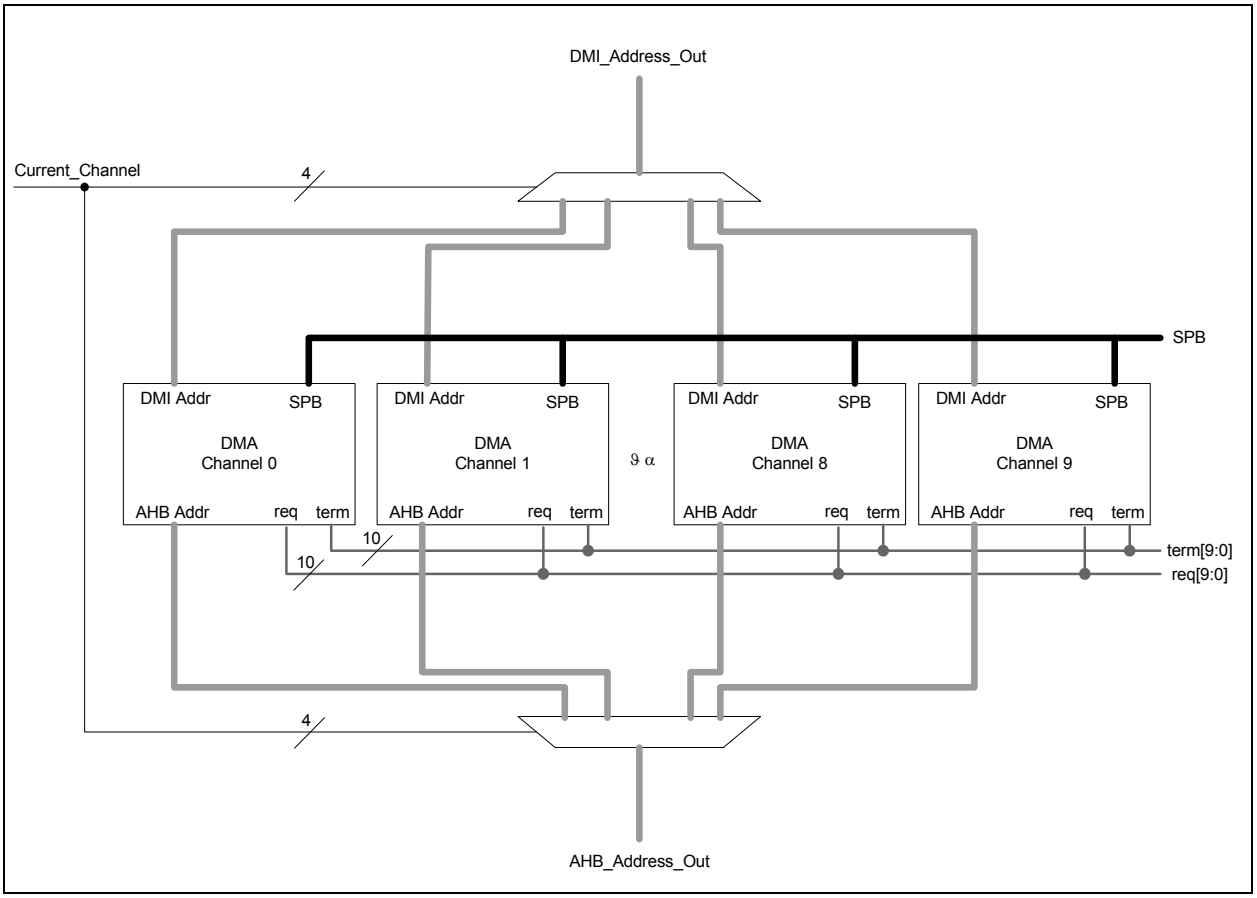
The Request signals from the devices are the Data register status signals. In the read (0, from the device to SRAM) direction, a DMA request is asserted as long as the Receive Data register is not empty. In the transmit (1, from the SRAM to the device) direction, a DMA request is asserted as long as the Transmit Data register is not full. For the SMBus controllers, the Terminate signal is asserted if the SMBus controller detects an error condition during an SMBus transaction, or if software shuts down the SMBus controller. There is no Terminate signal for the SPI Flash controller.

© 2012-2018 Microchip Technology Inc.

26.7.3 DMA CHANNEL ARBITRATION AND MULTIPLEXING

The ten DMA channels share the DMI interface and the AHB bus. Figure 26-4, "DMA Channel Multiplexing" illustrates the multiplexing of the channels onto the busses.

FIGURE 26-4: DMA CHANNEL MULTIPLEXING



A DMA Channel is ready to run as long as it is in the Run state and its selected Request input is asserted. The DMA controller services DMA Channels on a first-come first-served basis. If two channels become ready to run simultaneously, they are served in numerical order (channel 0 before channel 1, etc.). If multiple channels are continuously ready (that is, their respective Request inputs are always asserted), then they will be served in round-robin order.

26.8 DMA Registers

The base address for the DMA Controller block in the AHB address space is listed in Table 26-4, "DMA Controller Base Address Table".

TABLE 26-4: DMA Controller BASE ADDRESS TABLE

DMA Controller Instance	LDN from (Table 4-2 on page 59)	AHB Base Address
DMA Channel0	9h	F0_2400h + 000h
DMA Channel1		F0_2400h + 020h
DMA Channel2		F0_2400h + 040h
DMA Channel3		F0_2400h + 060h
DMA Channel4		F0_2400h + 080h
DMA Channel5		F0_2400h + 0A0h
DMA Channel6		F0_2400h + 0C0h
DMA Channel7		F0_2400h + 0E0h
DMA Channel8		F0_2400h + 100h
DMA Channel9		F0_2400h + 120h

The following table summarizes the registers allocated for the DMA Controller. The offset field in the following table is the offset from the AHB Base Address defined in Table 26-4 on page 426.

TABLE 26-5: DMA Controller REGISTER SUMMARY

	EC Interface			
Register Name	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
DMA Control Register	0h	1-0	R/W	
DMI End Address Register	4h	1-0	R/W	
DMI Start Address Register	8h	1-0	R/W	
AHB Address Register	Ch	3-0	R/W	
DMA Activate Register	10h	0	R/W	

© 2012-2018 Microchip Technology Inc.

26.8.1 DMA CONTROL REGISTER

The DMA Control Register is used to control the behavior of the DMA controller.

TABLE 26-6: DMA CONTROL REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC OFFSET	00h			16-bit			**EC SIZE**	
POWER	VTR			0000h			**VTR POR DEFAULT**	
BUS	EC SPB							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R/W	R/W	R/W	R/W	R/W
BIT NAME	Reserved			Device				DIR
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R/W
BIT NAME	Reserved		Busy	Status		Done	Request	Run

RUN

When this bit is 1, the state machine is active and the channel continually tries to move a byte between the I/O device at AHB Address and SRAM at DMA Address. When this bit is 1, software cannot modify any of the other registers in the channel, or any other bits in this register besides, in order to insure that no AHB transaction is modified while it is in progress.

Setting this bit to 0 will halt the DMA function. If there is an AHB transfer in progress when this bit is set to 0 the transfer will complete before the DMA state machine returns to the IDLE state. Firmware should query the Busy bit after setting Run to 0 in order to determine when the DMA transaction has terminated.

The DMA_Int signal is only asserted when is 1.

REQUEST

Read-only. This bit is always 0 when Run is 0, and is set when the DMA request input is 1. The DMA request input is selected by DEVICE and DIR.

DONE

Read-only. This bit is always 0 when is Run is 0, and is 1 when the DMA Controller state machine returns to the IDLE state. The DMA Controller state machine will transition back to IDLE when DMI Start Address Register equals the DMI End Address Register, when the DMA Termination input is 1 or if the AHB transaction is terminated with a bus error. The DMA Termination input is selected by DEVICE and DIR. This bit is routed to the interrupt controller.

STATUS

Read-only. This field is updated whenever Done goes from 0 to 1 or when Run goes from 1 to 0, and indicates why a DMA transfer completed. Status values are:

00:Run is set to 0. This field is always 0 when Run is 0.

01:Start Address matched End Address

10:DMA_Term input asserted

11:An AHB bus error occurred on the transfer

Status values are shown in order of priority. If more than one condition caused a return to the IDLE state, the condition with the lowest Status value is reported.

BUSY

Read-only. This bit is 1 when the DMA State Machine is not in the IDLE state and 0 when the DMA State Machine is in the IDLE state.

DIR

DMA transfer direction. 0 for reads from the AHB device to DCCM memory, 1 for writes from DCCM memory to the AHB device. When combined with DEVICE, determines which DMA_request input is used to start a DMA transfer

DEVICE

The DEVICE field selects which I/O device is assigned to this DMA channel. See Table 26-3, “DMA Device Selection,” on page 423 for the DEVICE field to Device Name mapping.

26.8.2 DMI END ADDRESS REGISTER

This address defines the DMA stops transferring. When the incrementer that was loaded from the DMI Start Address Register is equal to this register, the DMA completes.

TABLE 26-7: DMI END ADDRESS REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC OFFSET	04h			16-bit			**EC SIZE**	
POWER	VTR			00h			**VTR POR DEFAULT**	
BUS	EC SPB							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	DMI_End_Address[15:8]							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	DMI_End_Address[7:0]							

DMI_END_ADDRESS

This field contains the address one past the last byte to be transferred for the DMA channel. The DMA transfer stops when the current DMI address is equal to this register. If the End Address register is equal to the Start Address register when is set to 1, no data are transferred and the DMA terminates immediately.

© 2012-2018 Microchip Technology Inc.

26.8.3 DMI START ADDRESS REGISTER

Note: This register is 16-bit only. It does not support 8-bit accesses.

TABLE 26-8: DMI START ADDRESS REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC ADDRESS	08h			16bit			**EC SIZE**	
POWER	VTR			0000h			**VTR POR DEFAULT**	
BUS	EC SPB							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	DMI_Start_Address[15:8]							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	DMI_Start_Address[7:0]							

DMI_START_ADDRESS[15:0]

This field defines an offset from the base of the SRAM, which is an offset from both the base of the DCCM and the base of the ICCM, since the two memories are different aliases of a single dual-ported SRAM. This register is loaded into a 16-bit counter, which increments by 1 under state machine control, when is set to 1. This register defines the initial byte address for bytes to be transferred on the associated DMA channel.

When first written by software, this register contains the start address in the DCCM for the DMA transfer. While a DMA transfer is in progress, this register contains the address of the next byte to be transferred. When the DMA transfer completes, this register is one greater than the address of the last byte transferred. Software can determine how many bytes were transferred overall by subtracting the value it used to configure this register initially from the value of this register when the transfer completes.

26.8.4 AHB ADDRESS REGISTER

The AHB Address Register is the address of the I/O device that is the source or sink of the DMA transfer.

TABLE 26-9: AHB ADDRESS REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC ADDRESS	0Ch			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			**VTR POR DEFAULT**	
BUS	EC SPB							
BYTE3 BIT	**D31**	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE2 BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	AHB_Address[23:16]							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	AHB_Address[15:8]							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	AHB_Address[7:0]							

AHB_ADDRESS[23:0]

This is the address of the I/O port in the AHB address space. Software is responsible for insuring that this address is the correct address for the I/O device assigned to the channel.

© 2012-2018 Microchip Technology Inc.

26.8.5 DMA ACTIVATE REGISTER

The DMA Activate Register is used to gate clocks to a DMA channel, in order to conserve power. Software must set the Activate bit to '1b' in order for a channel to operate.

TABLE 26-10: DMA ACTIVATE REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC OFFSET	10h			8-bit			**EC SIZE**	
POWER	VTR			00h			**VTR POR DEFAULT**	
BUS	EC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R/W
BIT NAME	Reserved							Activate

ACTIVATE

When this bit is 0, the MCLK is gated to this channel, so the channel will not operate. When this bit is 1, the channel is provided with the system clock and the channel can operate.

27.0 SMB DEVICE INTERFACE

27.1 General Description

The MEC1632 SMB Device Interface includes four instances of an SMBus controller core: SMBus[3:0]. This chapter describes aspects of the SMB Device Interface that are unique to the MEC1632 instantiations of this core; including, Power Domain, Resets, Clocks, Interrupts, Registers and the Physical Interface. For a *General Description*, *Features*, *Block Diagram*, *Functional Description, Registers Interface* and other core-specific details, see Ref [1] (note: in this chapter, *italicized text* typically refers to SMBus controller core interface elements as described in Ref [1]).

27.1.1 REFERENCES

1. SMBus Controller Core Interface, Revision 2.0 (10 MHz), v3.31, Core-Level Architecture Specification, SMSC, 10/25/13

27.1.2 SMB PIN SIGNAL INTERFACE DESCRIPTION

The pin signals are defined in the Pin Configuration chapter.

27.2 Power, Clocks and Reset

27.2.1 POWER DOMAIN

This block is powered by the VTR Power Supply.

See Section 7.5, "Power Configuration," on page 121 for details on power domains. For more detail about the SMBus controller core Power Domain, see *Section 3.1, "Power Configuration"* in Ref [1].

27.2.2 CLOCKS

SMB Device Interface Clocking is described below in Table 27-1. Use this table when programming the SMBus controller core bus clock and timing values as specified in Ref [1].

TABLE 27-1: SMB Device Interface CLOCKING

Clock Source (Note 27-1)	SMBus Controller Core Clock (Note 27-2)	Frequency	Description
MCLK	CORE_CLK	20.27 MHz	–
MCLK_DIV2_EN	BAUD_CLK_EN	10.14 MHz	Use this frequency when programming the *Bus Clock Register*, *Data Timing Register* and the *Time-Out Scaling Register* described in Ref [1].
EC_BUS_CLK_EN	SPB_CLK_EN	Programmable	EC Bus Clock.

Note 27-1 See Section 7.4.11, "MCLK Sourced Clocking," on page 120.

Note 27-2 For more detail about SMBus controller core Clocking see *Chapter 2, "Hardware Interface"* and *Section 3.3,* "*Clocking*" in Ref [1].

27.2.3 RESETS

Each of the SMBus controller core instances in the MEC1632 SMB Device Interface are reset by nSYS_RST. See Section 7.6, "Reset Interface," on page 124 for details on resets in the MEC1632. For more detail about SMBus controller core Resets, see *Section 3.2,* "*Reset Interface*" in Ref [1].

© 2012-2018 Microchip Technology Inc.

27.3 Interrupts

Each EC SMB Controller has both an activity interrupt event and a START Bit detection Wake-up event. The SMB activity interrupt events are routed to the SMB0, SMB1, SMB2 and SMB3 bits in the GIRQ12 Source Register. The START Bit detection Wake-up events are routed to the SMB00 WK, SMB01 WK, SMB02 WK, SMB03 WK, SMB04 WK, SMB05 WK, SMB06 WK, SMB07 WK, SMB08 WK, SMB09 WK, SMB10 WK and SB_TSI bits in the GIRQ12 Source Register. The edge detection of the interrupt and wake events are controlled by their associated pin control registers in the Section 24.0, "GPIO Interface," on page 388.

APPLICATION NOTE: The pin control registers for GPIOs that are associated with the SDAT pins for ports supporting wake events should be programmed to Input, Falling Edge Triggered, non-inverted polarity detection.

27.4 DMA

Each EC SMB Controller can utilize two DMA Controller channels as defined in Ref [1]. DMA Channel configuration is defined in Table 26-3, “DMA Device Selection,” on page 423.

27.5 Registers

Each SMBus controller core instance in the MEC1632 SMB Device Interface has unique Register Interface Addressing, defined by a base address as indicated in Table 27-2. For more detail about SMBus controller core registers, see *Chapter 5, “Registers Interface”* in Ref [1].

TABLE 27-2: SMB Device Interface BASE ADDRESS TABLE

SMB Device Interface **Instance**	**LDN from (Table 4-3 on page 60)**	**AHB Base Address**
SMBus**.0**	6h	F0_1800h
SMBus**.1**		F0_1880h = F0_1800h + 80h
SMBus**.2**		F0_1900h = F0_1800h + 100h
SMBus**.3**		F0_1980h = F0_1800h + 180h

27.6 Physical Interface

27.6.1 OVERVIEW

The Physical Interface for the SMB controller core is configurable for up to 15 ports as defined below in Section 27.6.2, "SMBus Port Selection".

Each of the 4 SMBus controllers can be connected to any of the ports defined in the table. The *PORT SEL [3:0]* bits in each controller will appear the same (Table 27-3). The default for each field is 15, Reserved (which means that the SMB controller is not connected to a port).

An SMB port should be connected to a single controller. An attempt to configure the *PORT SEL [3:0]* bits in one controller to a value already assigned to another controller may result in unexpected results.

The port signal-function names and pin numbers are defined in the Pin Configuration chapter. The SMB Device Interface port selection is made using the *PORT SEL [3:0]* bits in the *Configuration Register* as described in Ref [1] and in the subsections that follow.

For SMB Device Interface port signal functions that are alternate functions of GPIO pins, the buffer type for these pins must be configured as open-drain outputs when the port is selected as defined in Section 27.6.2, "SMBus Port Selection". For more information regarding the SMBus controller core Physical Interface, see *Section 2.2,* “*Physical Interface*” in Ref [1].

27.6.2 SMBUS PORT SELECTION

SMBus Port Selection is defined below in Table 27-3.

TABLE 27-3: SMBUS PORT SELECTION

Port SEL [3:0]				Port (Note 27-3)
3	2	1	0	
0	0	0	0	SMB00
0	0	0	1	SMB01
0	0	1	0	SMB02
0	0	1	1	SMB03
0	1	0	0	SMB04
0	1	0	1	SMB05
0	1	1	0	SMB06
0	1	1	1	SMB07
1	0	0	0	SMB08
1	0	0	1	SMB09
1	0	1	0	SMB10
1	0	1	1	SB-TSI
1100b - 1111b				Reserved

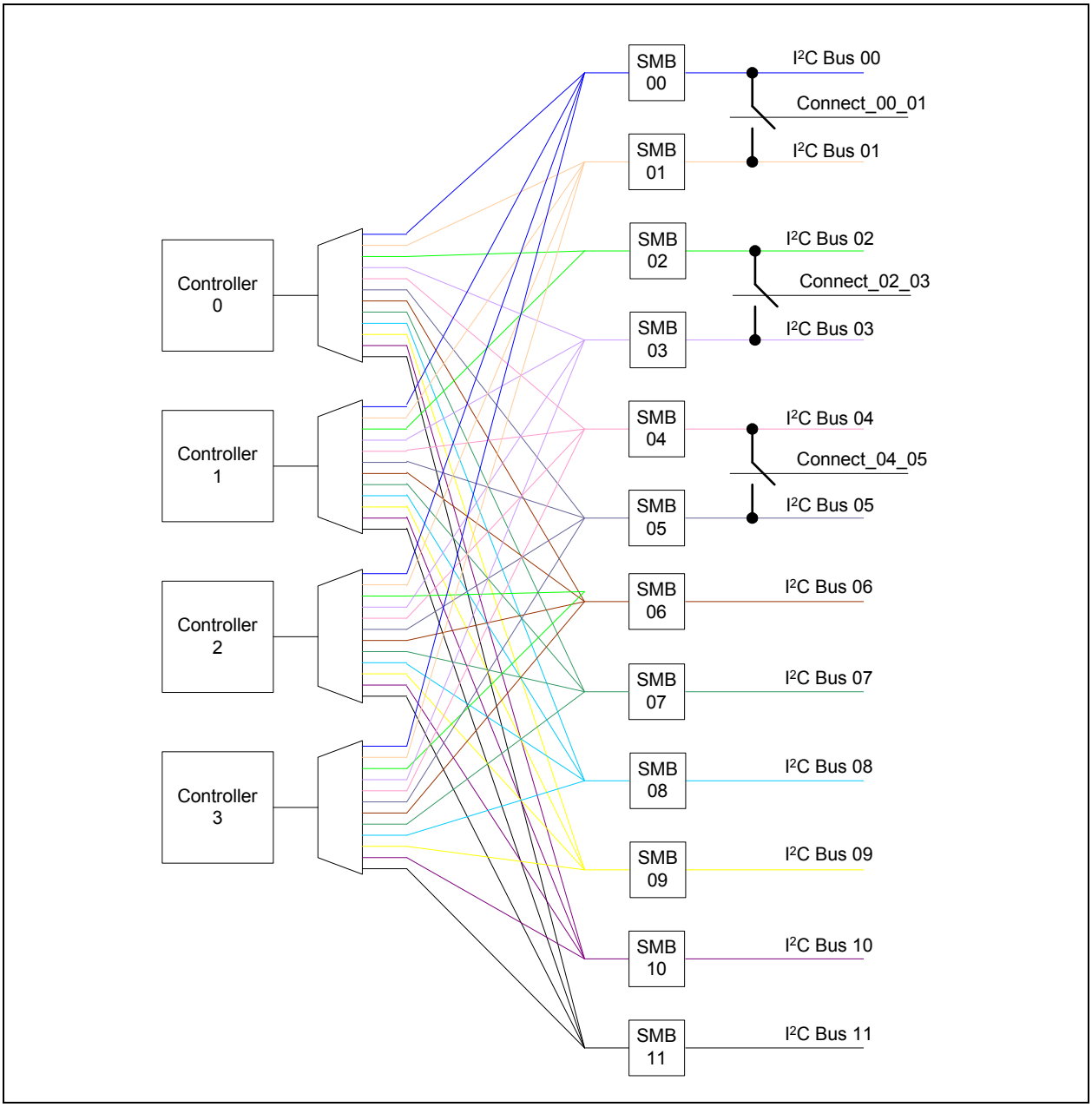
Note 27-3 see Pin Configuration chapter for a description of the SMB Device Interface pin configuration.

27.7 Port Isolation

- The SMBus Port Isolation feature in the MEC1632 enables electrical connectivity, or isolation between pairs of I^2C/SMBus ports.
- SMBus Port Isolation is only present between three pairs of ports as shown in TABLE 27-4: SMBus Port Connectivity Example on page 435. SMBus ports SMB06 through SMB10, as well as the SB-TSI port, are not affected by the SMBus Port Isolation feature.
- The Port Isolation feature is controlled by the LinkPort bits in the Port Isolation Register (see Section 27.7.1, "Port Isolation Register," on page 436).

© 2012-2018 Microchip Technology Inc.

TABLE 27-4: SMBUS PORT CONNECTIVITY EXAMPLE



27.7.1 PORT ISOLATION REGISTER

The Port Isolation Register is located in LDN 3Fh (F0_FC00h) in the System Registers block.

Offset	40h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:3	Reserved	RES	0	–
2	LinkPort4-5 (LP45) When this bit is asserted (‘1’), the SMBus04 port is electrically connected the SMBus05 port. When LP45 is not asserted (‘0’) (default), the SMBus04 port is electrically isolated from the SMBus05 port.	R/W	0	VTR POR
1	LinkPort2-3 (LP23) When this bit is asserted (‘1’), the SMBus02 port is electrically connected the SMBus03 port. When LP23 is not asserted (‘0’) (default), the SMBus02 port is electrically isolated from the SMBus03 port.			
0	LinkPort0-1 (LP01) When this bit is asserted (‘1’), the SMBus00 port is electrically connected the SMBus01 port. When LP01 is not asserted (‘0’) (default), the SMBus00 port is electrically isolated from the SMBus01 port.			

© 2012-2018 Microchip Technology Inc.

28.0 PECI INTERFACE

28.1 Overview

The MEC1632 includes a PECI Interface to allow the EC to retrieve temperature readings from PECI-compliant devices. The PECI Interface implements the PHY and Link Layer of a PECI host controller as defined in References[1] and includes hardware support for the PECI 3.0 command set.

The block has a 32 Byte FIFO for PECI 3.0 compliance.

This chapter focuses on MEC1632 specific PECI Interface configuration information such as Register Addressing, Power Domain, Resets, Physical Interface, Interrupts and Clocking. For a functional description of the MEC1632 PECI Interface refer to References [1].

28.2 References

1. PECI Interface Core, Rev. 1.31, Core-Level Architecture Specification, SMSC Confidential, 4/15/11.

28.3 Register Addressing

The PECI Interface module is attached to EC SPB. It is assigned EC LDN 19h with base address F0_6400h; register addresses are aligned on 4-byte boundaries. The PECI Interface registers are summarized in Table 28-1. For register details see References [1].

TABLE 28-1: PECI Interface REGISTERS SUMMARY

Address Offset	Mnemonic	Register Description	NL Access (Note 28-1)
0x00	SSTWRBUF	Write Data Register	RW
0x04	SSTRDBUF	Read Data Register	RW
0x08	SSTSCTL	Control Register	RW
0x0C	SSTSTA1	Status Register 1	RWC
0x10	SSTSTA2	Status Register 2	RWC
0x14	SSTERR	Error Register	RWC
0x18	SSTINTEN1	Interrupt Enable 1 Register	RW
0x1C	SSTINTEN2	Interrupt Enable 2 Register	RW
0x20	SSTOBT1	Optimal Bit Time Register (Low Byte)	RW
0x24	SSTOBT2	Optimal Bit Time Register (High Byte)	RW
0x28	SSTRTR1	Request Timer Register (Low Byte)	RW
0x2C	SSTRTR2	Request Timer Register (High Byte)	RW
0x30-0x3C	–	Reserved	R
0x40	SSTBLKID	Block ID Register	R
0x44	SSTREV	Revision Register	R
0x48 - 0x7C	Reserved and Test Registers	MCHP Reserved. MCHP Reserved registers are reserved for use by Microchip, only. Reading and Writing MCHP Reserved registers may cause undesirable results.	RW

Note 28-1 “R” means the register is read-only, writes have no affect; “RW” means the register can written and read; “RWC” means the register can written and read but that a ‘1’ must be written to a register bit to clear it.

28.4 Block Interface Parameters

28.4.1 SIGNAL LIST

TABLE 28-2: PECI Interface Signal List

Signal Name	Type	Description
VREF_VTT	INPUT	PECI Voltage Reference pin
PECI_DAT	INPUT/OUTPUT	PECI Data signal pin (VREF_VTT)
EC SPB	I/O Bus	EC MEC1632 peripheral bus
MCLK	INPUT	Master Clock
SPB_CLK_EN	INPUT	MEC1632 clock enable signal for Host interface clock
nSYS_RST	INPUT	Synchronous block reset signal
PECI_INT	OUTPUT	Interrupt signal from PECI controller to EC
SLEEP_EN	INPUT	External sleep enable control
CLOCK_REQ	OUTPUT	Clock required status
VTR	POWER	Digital logic voltage supply
GND		Ground

28.4.2 POWER DOMAIN

The PECI Interface core logic is powered by VTR; the Physical Interface Power Domain is VREF_VTT.

28.4.3 RESETS

The PECI Interface is reset on a nSYS_RST. The PECI Interface core also includes soft reset capabilities which reset control logic and part of registers. See References [1] for details.

28.4.4 CLOCKING

The PECI Interface Clocking requirement is defined below in Table 28-3.

TABLE 28-3: Clocking

Domain	Clock	Type	Frequency
CORE	MCLK	Fixed Clock	20.27 MHz
HOST	MCLK	Fixed Clock	20.27 MHz
	SPB_CLK_EN	Variable Clock Enable	20.27 MHz and slower.

28.4.5 INTERRUPTS

The interrupt from the PECI Interface module is routed to the PECI_INT bit of GIRQ16 Source Register.

28.4.6 PHYSICAL INTERFACE

The pin configuration for the MEC1632 PECI Physical Interface is defined in the PECI Interface section located in the Pin Configuration chapter.

28.4.7 SLEEP ENABLE/CLOCK REQUIRED POWER STATE CONTROLS

For a description of the PECI Interface Sleep Enable/Clock Required Power State Controls see the PECI Interface Core, Rev. 1.31, Core-Level Architecture Specification, SMSC Confidential, 4/15/11.

© 2012-2018 Microchip Technology Inc.

29.0 ANALOG TO DIGITAL CONVERTER

29.1 General Description

This block is designed to convert external analog voltage readings into digital values. It consists of a single successive-approximation Analog-Digital Converter that can support up to sixteen channels (refer to Pin Configuration to see which ADC Channels have been implemented).

Note: The characteristics of this interface are shown in Table 29-2.

APPLICATION NOTE: Transitions on ADC GPIOs are not permitted when Analog to Digital Converter readings are being taken.

A 0.1µF capacitor is recommended for the ADC inputs. If the ADC source impedance is greater than 10KΩ, then the 0.1µF capacitor is required.

29.2 Terminology

TABLE 29-1: TERMINOLOGY

Term	Definition
ADC	Analog Digital Converter
Conversion Cycle	The Conversion Cycle refers to the round-robin loop used to acquire the enabled ADC conversion readings.
Conversion Time Per Channel	Acquisition time for a single ADC channel during a conversion cycle.
Inter-Conversion Cycle Time	Time between Conversion Cycles
Power-up Conversion Latency	The Power-up Conversion Latency is the delay incurred before the first reading of the first conversion cycle after a power down event.

29.3 ADC Characteristics

TABLE 29-2: ADC CHARACTERISTICS

Parameter	MIN	TYP	MAX	Unit
Resolution	–	–	10	Bits
Total Inputs (Programmable)	–	–	16 (Note 29-1)	Channels
Conversion Time per channel	–	–	12	ADC_CLKs
	–	10	–	µs/channel **Note:** 10 us/channel is ideal value based on 1.2 MHz clock.
Inter-Conversion Cycle Time			2.5	ADC_CLKs
Power-up Conversion Latency			12	ADC_CLK
Absolute Accuracy	–	2	4	LSB
Integral Non-Linearity	-0.5	–	+0.5	LSB
Differential Non-Linearity	-0.5	–	+0.5	LSB
Input Impedance	7	10	–	MOhms

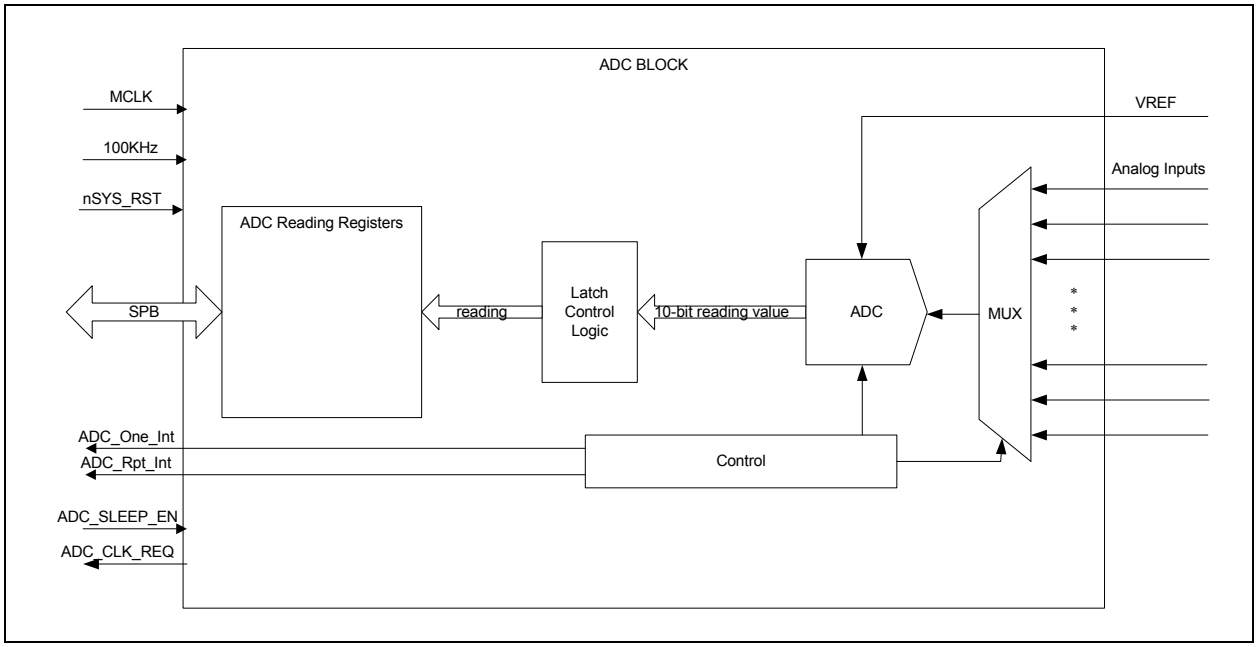
TABLE 29-2: ADC CHARACTERISTICS (CONTINUED)

Parameter	MIN	TYP	MAX	Unit
Analog Input Range	0	–	VREF_ADC	Volts
VREF_ADC	2.97	–	AVTR_ADC	Volts
VREF_ADC Impedance	14 K	16 K	–	Ohms
AVTR_ADC	2.97	3.3	3.63	Volts

Note 29-1 This is a 16-channel ADC. The maximum number of channels is limited in hardware by the pinout and in software by the channel enable bits located in the ADC One Shot Register and ADC One Shot Register registers.

29.4 ADC Block Diagram

FIGURE 29-1: ADC BLOCK DIAGRAM



Note: Unused Analog Inputs are terminated internally.

29.5 Block Diagram Signal List

TABLE 29-3: Analog to Digital Converter SIGNAL LIST

Signal Name	Direction	Description
VREF_ADC	INPUT	Analog Voltage Reference
AVTR_ADC	POWER	Analog Supply
VSS_ADC	POWER	Analog Ground
SPB	I/O Bus	EC MEC1632 peripheral bus
MCLK	INPUT	Master MEC1632 clock
nSYS_RST	INPUT	Block reset signal

© 2012-2018 Microchip Technology Inc.

TABLE 29-3: Analog to Digital Converter SIGNAL LIST (CONTINUED)

Signal Name	Direction	Description
ADC_One_Int	OUTPUT	Interrupt signal from ADC controller to EC for One-shot ADC conversion
ADC_Rpt_Int	OUTPUT	Interrupt signal from ADC controller to EC for Repeated ADC conversion
Analog Inputs	INPUT	Supports up to 16 analog voltage inputs from pins. See Section 2.5.7, "Analog Data Acquisition Interface," on page 55 for ADC pins implemented.
ADC_SLEEP_EN	INPUT	External enable/disable signal used to put the block in the lowest power consumption state. 0=No Sleep Requested. The block should operate as configured. 1=Sleep Requested. The block enters sleep mode. See Low Power States on page 442.
ADC_CLK_REQ	OUTPUT	This output indicates when this block requires this clock input. 0= MCLK can be turned 'off' when appropriate 1= MCLK is required to be 'on.'

29.6 Power, Clocks and Reset

29.6.1 POWER DOMAIN

This block is powered by the VTR power supply with a separate Analog supply (AVDD).

29.6.2 CLOCKS

TABLE 29-4: ADC CLOCK SOURCES

Name	Description
MCLK	This block is a synchronous design clocked by MCLK. All other clocks internal to this block are derived from this source.
ADC_CLK	The 1.2MHz clock source (ADC_CLK), which is generated from MCLK, is used to derive the 10 us period used for delay generation in the block.

29.6.3 RESET

This block is reset on a nSYS_RST.

29.7 ADC Interrupts

The ADC generates an interrupt/wake-up events to the EC which indicate an ADC conversion cycle is complete. The ADC_OneStat bit and the ADC_RptStat bit in the ADC Control Register are set when conversion cycles complete. The two status bits are routed to the ADC bits in the GIRQ16 Source Register.

29.8 Low Power States

The Analog to Digital Converter is designed to conserve power when the block is in the IDLE state. The block may enter the IDLE state as shown in the table when the Activate is 0 or the ADC_SLEEP_EN bit is asserted if the ADC is not BUSY.

Note:

- The block is considered to be “NOT IDLE” when the ADC is activated and ADC_SLEEP_EN = 0 or when ADC_SLEEP_EN = 1 and the ADC is performing a conversion. The ADC will always complete the current conversion when the ADC_SLEEP_EN is asserted before it will remove the clock required signal.
- The block will never enter the IDLE state when it is configured for continuous mode (i.e., Start_Repeat = 1 & one or more ADC channels enabled) if the delay time is 0 sec. However, if a delay greater than 0 sec is programmed, the block will enter the IDLE state after the conversion cycle completes.
- The block will only enter the IDLE state in one-shot mode (i.e., Start_Once = 1) after the conversion cycle completes.
- Clearing the Activate bit or setting the ADC_SLEEP_EN bit will not terminate a conversion cycle. A conversion cycle can only be terminated by clearing the Start_Repeat and Start_Once bits and completing the current conversion.
- There is no way to immediately terminate a conversion. However, firmware can minimize the conversion cycle time by clearing the enable bits per channel (worst case is conversion time for one channel)
- Clearing the Active bit or setting the ADC_SLEEP_EN bit are non-destructive operations. All registers will maintain their programmed state. There is no requirement to reconfigure the block following sleep.

TABLE 29-5: BLOCK CLOCK GATING IN LOW POWER STATES

Activate Bit	ADC_SLEEP_EN	Block Idle Status	ADC_CLK_REQ	State	Description
0	X	NOT IDLE	1	PREPARING to DISABLE	The core clock is required until the current transaction is completed and the Block is IDLE.
		IDLE	0	DISABLED	The block is idle and the core clock can be stopped.
1	0	X	1	NORMAL OPERATION	The block in neither disabled by firmware nor commanded to SLEEP
	1	NOT IDLE	1	PREPARING to SLEEP	The core clock is required until the current transaction is completed and the Block is IDLE.
		IDLE	0	SLEEPING	The block is idle and the core clock can be stopped.

29.9 Operation

The MEC1632 features a sixteen channel successive approximation Analog to Digital Converter. The ADC architecture features excellent linearity and converts analog signals to 10 bit words.

Conversion takes 10 microseconds per 10-bit word. The sixteen channels are implemented with a single high speed ADC fed by a sixteen input analog multiplexor. The multiplexor cycles through the sixteen voltage channels, starting with the lowest-numbered channel and proceeding to the highest-number channel, selecting only those channels that are programmed to be active. The total loop time is comprised of the time to convert the enabled channels, scan the disabled (or unused) channels, and the Inter-Conversion Cycle Time.

29.9.1 TOTAL LOOP TIME:

[Time for Conversions] + [Time for Scanning Disabled Channels] + [Inter-Conversion Cycle Time]

[(#CHANNELS_ENABLED) * 12 ADC_CLKs] + [(16 channels - #CHANNELS_ENABLED) * 1 MCLK] + [2.5 ADC_CLKs]

Note: The first conversion cycle following a power down event incurs an additional delay referred to as Power-up Conversion Latency.

© 2012-2018 Microchip Technology Inc.

The input range on the voltage channels spans from 0V to the external voltage reference. With a voltage reference of 3.3V, this provides resolutions of 3.2mV. The range can easily be extended with the aid of resistor dividers. The accuracy of any voltage reading depends on the accuracy and stability of the voltage reference input.

The ADC conversion cycle starts either when the Start_Once bit in the ADC Control Register to set to 1 or when the ADC Repeat Timer counts down to 0. When the Start_Once is set to 1 the conversion cycle converts channels enabled by configuration bits in the ADC One Shot Register. When the Repeat Timer counts down to 0 the conversion cycle converts channels enabled by configuration bits in the ADC Repeat Register. When both the Start_Once bit and the Repeat Timer request conversions the Start_Once conversion is completed first.

Note:	If software repeatedly sets Start_Once to 1 at a rate faster than the Repeat Timer count down interval, the conversion cycle defined by the ADC Repeat Register will not be executed.

29.9.2 INPUT TERMINATION

See APPLICATION NOTE: on page 439.

29.9.3 ADC REGISTERS

The base address for the ADC block in the AHB address space is listed in Table 29-6, "Analog to Digital Converter Base Address Table".

TABLE 29-6: Analog to Digital Converter BASE ADDRESS TABLE

ADC **Instance**	**LDN from (Table 4-2 on page 59)**	**AHB Base Address**
ADC	1Ah	F0_6800h

The following table summarizes the registers allocated for the ADC. The offset field in the following table is the offset from the Embedded Controller (EC) AHB Base Address.

- All register accesses require a single 16, or 32 bit read of the ADC Channel Reading Registers, because the data holding registers have been removed; i.e., two 8 bit reads cannot guarantee data coherency.

TABLE 29-7: Analog to Digital Converter REGISTER SUMMARY

	EC Interface			
Register Name	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
ADC Control Register	0h	3-0	R/W	
ADC Delay Register	4h	3-0	R/W	
ADC Status Register	8h	3-0	R/W	
ADC One Shot Register	Ch	3-0	R/W	
ADC Repeat Register	10h	3-0	R/W	

TABLE 29-7: Analog to Digital Converter REGISTER SUMMARY (CONTINUED)

	EC Interface			
Register Name	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
ADC Channel 0 Reading Registers	14h	3-0	R	Table 29-13 Note 29-2
ADC Channel 1 Reading Registers	18h	3-0	R	
ADC Channel 2 Reading Registers	1Ch	3-0	R	
ADC Channel 3 Reading Registers	20h	3-0	R	
ADC Channel 4 Reading Registers	24h	3-0	R	
ADC Channel 5 Reading Registers	28h	3-0	R	
ADC Channel 6 Reading Registers	2Ch	3-0	R	
ADC Channel 7 Reading Registers	30h	3-0	R	
ADC Channel 8 Reading Registers	34h	3-0	R	
ADC Channel 9 Reading Registers	38h	3-0	R	
ADC Channel 10 Reading Register	3Ch	3-0	R	
ADC Channel 11 Reading Register	40h	3-0	R	
ADC Channel 12 Reading Register	44h	3-0	R	
ADC Channel 13 Reading Register	48h	3-0	R	
ADC Channel 14 Reading Register	4Ch	3-0	R	
ADC Channel 15 Reading Register	50h	3-0	R	

29.9.4 ADC CONTROL REGISTER

The ADC Control Register is used to control the behavior of the Analog to Digital Converter.

TABLE 29-8: ADC CONTROL REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC OFFSET	00h			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			**VTR POR DEFAULT**	
BUS	EC SPB							
BYTE[3-1] BIT	**D31**	**D30**	**D29**	**...**		**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/WC	R/WC	R	R	R/W	R/W	R/W	R/W
BIT NAME	ADC_ OneStat	ADC_ RptStat.	Reserved		Power_S aver	Start_ Repeat	Start_ Once	Activate

ACTIVATE

0: The ADC is disabled and placed in a low power state. Any conversion cycle in process will complete before the block is shut down, so that the reading registers will contain valid data but no new conversion cycles will begin.

1: Start_Once or Start_Repeat can begin data conversions by the ADC. A one cycle reset pulse is sent to the ADC core when this bit changes from 0 to 1.

© 2012-2018 Microchip Technology Inc.

START_ONCE

Writing this bit with a 1 will start a single conversion cycle of all ADC channels enabled by bits Single_En[15:0] in the ADC One Shot Register. The conversion cycle will begin without a delay. Every channel that is enabled will be converted in 10 μs. After all channels enabled by Single_En[15:0] are complete, ADC_OneStat will be set to 1. When the conversion cycle begins the bit is cleared.

If Start_Once is written with a 1 while a conversion cycle is in progress because Start_Repeat is set, the conversion cycle will complete, followed immediately by a conversion cycle using Single_En[15:0] to control the channel conversions.

Writing this bit with a 0 has no effect.

START_REPEAT

Writing this bit with a 1 will start a conversion cycle of all ADC channels enabled by bits Rpt_En[15:0] in the ADC Repeat Register. The conversion cycle will begin after a delay determined by Start_Delay[15:0] in the ADC Delay Register. The Start_Delay[15:0] value is loaded into an internal Repeat Timer register and the conversion cycle begins when the Repeat Timer counts down to 0. Every channel that is enabled will be converted in 10 μs. After all channels enabled by Rpt_En[15:0] are complete, ADC_RptStat will be set to 1. As long as Start_Repeat is 1 when the Repeat Timer counts down to 0, the Repeat Timer will be reloaded with Repeat_Delay[15:0], so that the ADC will repeatedly begin conversion cycles with a period defined by Repeat_Delay[15:0]. If the delay period expires and a conversion cycle is already in progress because Start_Once was written with a 1, the cycle in progress will complete, followed immediately by a conversion cycle using Rpt_En[15:0] to control the channel conversions.

Setting this bit to 0 will not terminate any conversion cycle in process, but will clear the Repeat Timer and inhibit any further periodic conversions.

POWER_SAVER

The Analog to Digital Converter includes a optional power saving feature to keep the ADC analog section off for as long as possible. To do this, the controller powers down the ADC between conversions sequences.

The Power_Saver feature is not optimized for power cycling. For example, if a one-shot conversion sequence follows immediately after a repeat conversion sequence (or vice versa), the ADC controller will still disable the ADC after the first conversion sequence, and re-enable it before starting the second conversion sequence.

ADC_ONESTAT

This bit is cleared whenever an ADC conversion cycle is begun when Start_Once is written with a 1 and is set to 1 when the conversion cycle started by writing Start_Once completes.

This bit is also cleared when it is written with a 1. Writing a 0 to this bit has no effect.

This bit can be used to generate an EC interrupt.

ADC_RPTSTAT

This bit is cleared whenever an ADC conversion cycle is begun when Start_Repeat is 1 and is set to 1 when a repeating conversion cycle completes.

This bit is also cleared when it is written with a 1. Writing a 0 to this bit has no effect.

This bit can be used to generate an EC interrupt.

© 2012-2018 Microchip Technology Inc.

29.9.5 ADC DELAY REGISTER

The ADC Delay register determines the delay from setting Start_Repeat in the ADC Control Register and the start of a conversion cycle. This register also controls the interval between conversion cycles in repeat mode.

TABLE 29-9: ADC DELAY REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC OFFSET	04h			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			**VTR POR DEFAULT**	
BUS	EC SPB							
BYTE3 BIT	**D31**	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Repeat_Delay[15:8]							
BYTE2 BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Repeat_Delay[7:0]							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Start_Delay[15:9]							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Start_Delay[7:0]							

START_DELAY[15:0]

This field determines the starting delay before a conversion cycle is begun when Start_Repeat is written with a 1. The delay is in units of 40µs. A value of 0 means no delay before the start of a conversion cycle, and a value of 0xFF means a delay of 2.6 seconds.

This field has no effect when Start_Once is written with a 1.

REPEAT_DELAY[15:0]

This field determines the interval after one conversion cycle completes and the next cycle begins when Start_Repeat is 1. The delay is in units of 40µs. A value of 0 means no delay between conversion cycles, and a value of 0xFF means a delay of 2.6 seconds.

This field has no effect when Start_Once is written with a 1.

Note:	If the Repeat Timer counts down to 0 more than once while a conversion cycle is in progress, only one periodic conversion cycle will be requested.

© 2012-2018 Microchip Technology Inc.

29.9.6 ADC STATUS REGISTER

The ADC Status Register indicates whether the ADC has completed a conversion cycle.

TABLE 29-10: ADC STATUS REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC ADDRESS	08h			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			**VTR POR DEFAULT**	
BUS	EC SPB							
BYTE[3:2] BIT	**D31**	**D30**	**D29**	**...**		**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
BIT NAME	ADC_Ch_Status15	ADC_Ch_Status14	ADC_Ch_Status13	ADC_Ch_Status12	ADC_Ch_Status11	ADC_Ch_Status10	ADC_Ch_Status9	ADC_Ch_Status8
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
BIT NAME	ADC_Ch_Status7	ADC_Ch_Status6	ADC_Ch_Status5	ADC_Ch_Status4	ADC_Ch_Status3	ADC_Ch_Status2	ADC_Ch_Status1	ADC_Ch_Status0

ADC_CH_STATUS[15:0]

Each bit in this field reports the conversion status of the corresponding ADC channel. All bits are cleared either by being written with a '1,' or following a system reset (nSYS_RST). Each bit is set when the conversion on the corresponding channel is complete. When ADC_CH_Status[15:0] matches Single_En[15:0] after a conversion cycle initiated by a write to the Start_Once bit in the ADC Control Register), bit ADC_OneStat in the ADC Control Register is set and an interrupt to the EC will occur (if the interrupt is enabled). When ADC_CH_Status[15:0] matches Rpt_En[15:0] after a conversion cycle initiated by a value of 1 in bit Start_Repeat in the ADC Control Register), bit ADC_RptStat in the ADC Control Register is set and an interrupt to the EC will occur (if the interrupt is enabled).

Conversions always start with the lowest-numbered enabled channel and proceed to the highest-numbered enabled channel.

29.9.7 ADC ONE SHOT REGISTER

The ADC One Shot Register is used to control which ADC channels are captured during a one-shot conversion cycle initiated by the Start_Once bit in the ADC Control Register.

TABLE 29-11: ADC ONE SHOT REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC ADDRESS	0Ch			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			**VTR POR DEFAULT**	
BUS	EC SPB							
BYTE[3:2] BIT	**D31**	**D30**	**D29**	**...**		**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Single_ En15	Single_ En14	Single_ En13	Single_ En12	Single_ En11	Single_ En10	Single_ En9	Single_ En8
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Single_ En7	Single_ En6	Single_ En5	Single_ En4	Single_ En3	Single_ En2	Single_ En1	Single_ En0

SINGLE_EN[15:0]

Each bit in this field enables the corresponding ADC channel when a single cycle of conversions is started when the Start_Once bit in the ADC Control Register is written with a 1. If a Single_En[*i*] bit is 1, the channel is enabled. If a Single_En[*i*] bit is 0, the channel is disabled. At least one channel must be enabled before a conversion cycle can be initiated. Conversions start with the lowest-numbered channel that is enabled and proceed to the highest-numbered enabled channel. If this register is changed while a conversion cycle is in progress the conversion cycle will use the new values for channels that have not yet been examined, but will not rescan channels that have already been checked.

© 2012-2018 Microchip Technology Inc.

29.9.8 ADC REPEAT REGISTER

The ADC Repeat Register is used to control which ADC channels are captured during a one-shot conversion cycle initiated by the Start_Repeat bit in the ADC Control Register.

TABLE 29-12: ADC REPEAT REGISTER

HOST OFFSET	N/A				N/A		**HOST SIZE**	
EC ADDRESS	10h				32-bit		**EC SIZE**	
POWER	VTR				0000_0000h		**VTR POR DEFAULT**	
BUS	EC SPB							
BYTE[3:2] BIT	**D31**	**D30**	**D29**	**...**		**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Rpt_ En15	Rpt_ En14	Rpt_ En13	Rpt_ En12	Rpt_ En11	Rpt_ En10	Rpt_ En9	Rpt_ En8
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Rpt_ En7	Rpt_ En6	Rpt_ En5	Rpt_ En4	Rpt_ En3	Rpt_ En2	Rpt_ En1	Rpt_ En0

RPT_EN[15:0]

Each bit in this field enables the corresponding ADC channel for each pass of the Repeated ADC Conversion that is controlled by bit Start_Repeat in the ADC Control Register. If a Rpt_En[*i*] bit is 1, the channel is enabled. If a Rpt_En[*i*] bit is 0, the channel is disabled. At least one channel must be enabled before a conversion cycle can be initiated. Conversions start with the lowest-numbered channel that is enabled and proceed to the highest-numbered enabled channel. If this register is changed while a conversion cycle is in progress the conversion cycle will use the new values for channels that have not yet been examined, but will not rescan channels that have already been checked.

29.9.9 ADC CHANNEL READING REGISTERS

All 16 ADC channels return their results into a 32-bit reading register. In each case the low 10 bits of the reading register return the result of the Analog to Digital conversion and the upper 22 bits return 0. Table 29-13 shows the format of all the reading registers. Table 29-7, “Analog to Digital Converter Register Summary,” on page 443 shows the addresses of all the reading registers.

Note 29-2 The ADC Channel Reading Registers access require single 16, or 32 bit reads; i.e., two 8 bit reads cannot guarantee data coherency.

TABLE 29-13: ADC CHANNEL X READING REGISTER

HOST OFFSET	N/A			N/A			**HOST SIZE**	
EC ADDRESS	xxh			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			**VTR POR DEFAULT**	
BUS	EC SPB							
BYTE[3:2] BIT	**D31**	**D30**	**D29**	**...**		**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved						ADCx_[9:8]	
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	ADCx_[7:0]							

ADCX_[9:0]

This read-only field reports the 10-bit output reading of ADCx.

© 2012-2018 Microchip Technology Inc.

30.0 TACH MONITOR

30.1 General Description

This block is designed to monitor tach output signals or locked rotor signals from various types of fans to determine their speed. One mode returns the value in number of CLOCK_LOW pulses. Another mode returns the value in pulses per programmed amount of time. This second mode can use the raw tach input. Each Tach is associated with a pair of limit registers that define maximum and minimum acceptable Tach counter values. If the readings on a Tach is outside these limits an interrupt to the EC can be generated.

In typical systems the fans are powered by the main power supply. Firmware may disable this block when it detects the main power rail has been turned off.

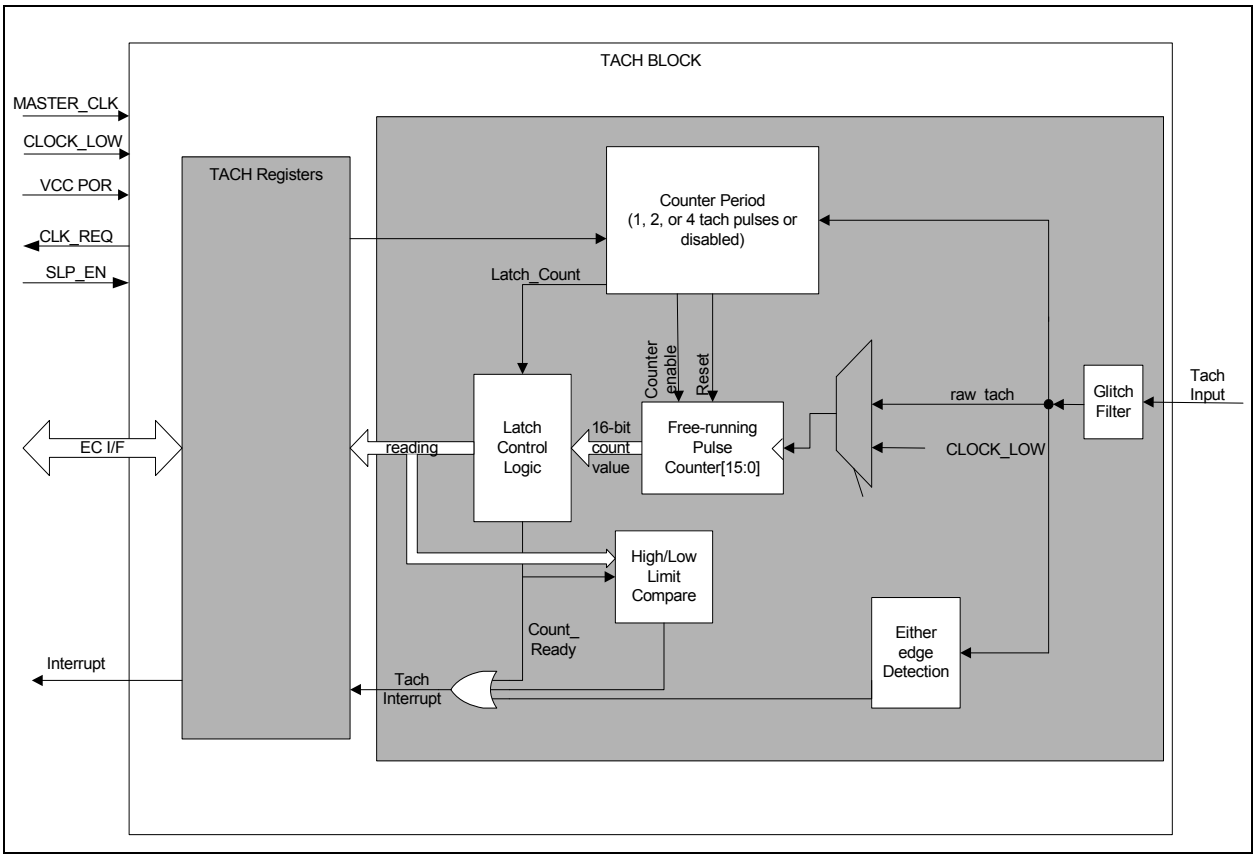
APPLICATION NOTE: This block can be utilized with Fans running at the following speed range:100 to 30K RPM.

The TACH Monitor performs the following functions:

- Count the number of pulses detected on the raw tach input.
- Count the number of clocks for a programmed number of pulses.
- Generate an interrupt when the count value is latched into the reading register.
- Generate a programmable either-edge triggered interrupt for detecting when the tach input changes state. This may be used for Locked Rotor detection.
- Generate an interrupt when the count value latched into the reading register is greater than the high limit or less than the low limit.

30.2 TACH Monitor Block Diagram

FIGURE 30-1: BLOCK DIAGRAM OF TACH MONITOR



Note:	Once the counter is enabled it is a 16-bit free-running counter. Latch count value on a read or when number of tach pulses is detected (if enabled) for 1, 2, or 4 pulses. Counter is reset to 0000h if the count value is latched by a programmed number of tach pulses and on a VCC POR. Counter enable is software controlled signal.

30.3 Block Diagram Signal List

TABLE 30-1: TACH PORT LIST

Signal Name	Direction	Description
VTR POR	INPUT	nSYS_RST
Master_Clock	INPUT	MCLK
CLOCK_LOW	INPUT	100KHz MCLK_DIV203_EN
EC I/F	I/O Bus	EC-side SPB bus
Tach Input	INPUT	Tachometer signal from TACHx Pin
Interrupts	OUTPUT	Interrupt used to indicate that either Tach Input has changed state or the TACH reading has been updated. One per TACH
SLP_EN	INPUT	Sleep Enable input from MEC1632 Clock Generator Power Management Interface.
CLK_REQ	OUTPUT	Clock Required output to MEC1632 Clock Generator Power Management Interface.

30.4 Power, Clocks and Reset

30.4.1 POWER DOMAIN

This block is powered by the VTR Power Supply.

See Section 7.5, "Power Configuration," on page 121 for details on power domains.

30.4.2 CLOCKS

This block uses the EC Bus Clock and the 100KHz MCLK_DIV203_EN. EC Bus Clock is used when reading and writing the TACH Monitor control registers. The individual TACH counters are driven by Clock_Low, the MCLK_DIV203_EN.

The TACH Monitor clock required output (CLK_REQ) is the inversion of the sleep enable input (SLP_EN). The CLK_REQ output is not asserted when the TACH Monitor is disabled.

See also Section 7.4, "Clock Generator," on page 98 for details on clocks.

30.4.2.1 Clock Idle

When the internal ring oscillator is disabled or when the TACH block is disabled, the internal TACH counters are reset. The reading register is not affected. This insures that inaccurate readings are not generated if the master clock halts in the middle of a TACH reading or when the TACH starts up.

Note:	Each Tach pin should be pulled up via an external resistor to the main power supply.

30.4.3 RESET

This block is reset on a nSYS_RST.

See Section 7.6, "Reset Interface," on page 124 for details on reset.

30.5 TACH Interrupts

Each TACH Monitor in the MEC1632 can be used to generate one interrupt event. Each TACH Monitor interrupt source is a level, active high signal. The TACH Monitor interrupts are routed to the TACH5, TACH4, TACH3, TACH2, TACH1, & TACH0 bits in GIRQ17 Source Register on page 315. The TACH Monitor interrupts generate interrupt events.

© 2012-2018 Microchip Technology Inc.

30.6 TACH Circuitry

The TACH Circuitry is implemented as a pulse counter. There are two types of toggling signals that can be used to increment the counter: the raw tach input or CLOCK_LOW. See FIGURE 30-1: Block Diagram of TACH Monitor on page 451. The two modes for incrementing the counter are controlled by Tach Reading Mode Select in the TACHx Control Register.

If the raw tach is used to increment the counter, the circuitry can be configured as a free-running counter that increments when a pulse from the tach is detected (i.e., input signal transitions from low-to-high).The counter is latched into the reading register (Tachx Counter in the TACHx Control Register) every time it is incremented. If this mode is selected, firmware will monitor the number of pulses detected over a period of time to determine the speed of the attached fan.

If CLOCK_LOW is used to increment the counter, the raw tach input will be used to determine when to latch the current count value into the reading register and reset the counter to 0000h. The counter is latched after a programmed number of tach pulses is detected. The programmed period can be configured to be 1, 2, or 4 tach pulses in duration.

Each Tach counter has comparison logic to compare the counter value with the high limit and low limit registers.

30.6.1 TACH INTERRUPT SOURCES

There are three interrupt source events: notify EC when reading is updated, notify EC when TACH input toggles, or notify EC when the TACH reading exceeds a programmed limit. The corresponding interrupt status bits are Count Ready Status Toggle Status Out-of-Limit Status Bits[3,1,0] in TACHx Status Register on page 457.

30.6.1.1 Count Reading Ready Status

This status bit is asserted when the counter value is latched. The bit is implemented in Bit D3 of the TACHx Status Register.

30.6.1.2 Tach Input Toggle Status

This status bit is asserted when the Tach input changes state. The bit is implemented in Bit D2 of the TACHx Status Register.

30.6.1.3 TACH Out-of_Limit STATUS

To generate a TACH out-of-limit status event, the high and low limits may be programmed in the TACHx High Limit Register and TACHx Low Limit Register. An out-of-limit event is triggered when the reading register (Tachx Counter in the TACHx Control Register) is set to a value less than the TACHx Low Limit Register or to a value greater than the TACHx High Limit Register. If the value in the Tachx Counter violates the programmed limits the TACH limit registers a status event will be generated, indicating the out-of-limit event. This status bit is implemented in Bit D0 of the TACHx Status Register. This signal may be used to interrupt the Embedded Controller, if enabled via Bit D0 of the TACHx Control Register.

Note:	If the TACHx Low Limit Register is set to 0000h, no out-of-limit event will be triggered by a Tachx Counter value that is below the limit. If the TACHx High Limit Register is set to FFFFh, no out-of-limit event will be triggered by a Tachx Counter value that is above the limit.

APPLICATION NOTE: Out-of-Limit checks are typically only used when the tach counter is incremented in Mode 1 (in which the counter counts the number of CLOCK_LOW until a programmed number of pulses occur on the raw tach input).

30.7 Registers

There are six block instances defined in this chapter: TACH[5:0].

Each instance of the TACH Monitor has its own Logical Device Number, and Base Address as indicated in Table 30-2.

TABLE 30-2: TACH Monitor BASE ADDRESS TABLE

TACH Monitor Instances	LDN from (Table 4-3 on page 60)	AHB Base Address
TACH0	18h	F0_6000h
TACH1		F0_6080h
TACH2		F0_6100h
TACH3		F0_6180h
TACH4		F0_6200h
TACH5		F0_6280h

Table 30-3 is a register summary for one instance of the TACH Monitor.

TABLE 30-3: TACHX REGISTER SUMMARY

	EC Interface			
Register Name	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
TACHx Control Register	00h	3-0	R/W	
TACHx Status Register	04h	0	R/W	
TACHx High Limit Register	08h	1-0	R/W	
TACHx Low Limit Register	0Ch	1-0	R/W	

© 2012-2018 Microchip Technology Inc.

30.7.1 DETAILED DESCRIPTION OF TACHOMETER REGISTER VALUES

This section describes the parameters that must be stored in hardware registers that will used by the TACH logic.

30.7.1.1 TACHx Control Register

TABLE 30-4: TACHX CONTROL REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	00h			32-bit			**EC SIZE**	
POWER	VTR			0000_0000h			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE3 BIT	**D31**	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	TACHx Counter [15:8] Register							
BYTE2 BIT	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	TACHx Counter [7:0] Register							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Tach Input INT_EN	Count Ready INT_EN	Reserved	Tach Edges		Tach Reading Mode Select	Reserved	Filter Enable
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R/W	R/W
BIT NAME	Reserved						TACH Enable	Tach Out-of-Limit Enable

TACH OUT-OF-LIMIT ENABLE

The TACH Out-of_Limit Enable is used to enable Bit[0] TACH Out-of_Limit Status bit to generate an interrupt event.

0=disable interrupt output from tach block (default)

1=enable interrupt output from tach block

TACH ENABLE

This bit enables the TACH logic.

0=TACH Idle (default)

This mode gates the clocks to the TACH block. The TACHx pin is tristate in the idle mode.

1=TACH Monitoring enabled.

APPLICATION NOTE: This bit gates the clocks into the block. When re-enabled, the internal counters will continue from the last known state and stale status events may still be pending. Firmware should discard any status or reading values until the reading value has been updated at least one time after the enable bit is set.

FILTER ENABLE

The TACH glitch filter rejects input pulses that are less than three CLOCK_LOW periods wide.

0=Filter disabled (default)

1=Filter enabled

APPLICATION NOTE: It is recommended that the tach input filter always be enabled.'

TACH READING MODE SELECT

0 = Counter is incremented when Tach Input transitions from low-to-high state (default)

1 = Counter is incremented on the rising edge of the CLOCK_LOW input. The counter is latched into Tachx Counter and reset when the programmed number of edges is detected.

TACH EDGES

A tach signal is a square wave with a 50% duty cycle. Typically, two tach periods represents one revolution of the fan. A tach period consists of three tach edges.

This programmed value represents the number of tach edges that will be used to determine the interval for which the number of CLOCK_LOW pulses will be counted

00 = 2 Tach edges (1/2 tach period)

01 = 3 Tach edges (1 tach period)

10 = 5 Tach edges (2 tach periods)

01 = 9 Tach edges (4 tach periods)

COUNT READY INT_EN

0=disable interrupt output from tach block (default)

1=enable interrupt output from tach block

TACH INPUT INT_EN

0=disable interrupt output from tach block (default)

1=enable interrupt output from tach block

TACHX COUNTER

This 16-bit field contains the latched value of the tach counter, which may be configured to operate as a free-running counter or to be gated by the tach input signal.

If the counter is free-running (Mode 0), it increments (if enabled) at the rate determined by the raw tach signal and latched into this field every time it is incremented. The act of reading this field will not reset the counter, which rolls over to 0000h after FFFFh. The firmware will compute the delta between the current count reading and the previous count reading, to determine the number of pulses detected over a programmed period.

If the counter is being gated by the tach input and clocked by the CLOCK_LOW (Mode 1), the counter will be latched into the reading register when the programmed number of edges is detected or when the counter reaches FFFFh and the counter will be reset to zero.

APPLICATION NOTE: In Mode 1, a counter rate of FFFFh means that the tach did not detect the programmed number of edges in 655ms. A stuck fan can be detected by setting the TACHx High Limit Register to a number less than FFFFh. If the counter then reaches FFFFh, the reading register will be set to FFFFh and an out-of-limit interrupt can be sent to the EC.

© 2012-2018 Microchip Technology Inc.

30.7.1.2 TACHx Status Register

TABLE 30-5: TACHX STATUS REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	04h			32-bit			**EC SIZE**	
POWER	VTR			0000h			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE3-1 BIT	**D31**	**D32**	**D31**	**...**		**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R/WC	R/WC	R	R/WC
BIT NAME	Reserved				Count Ready Status	Toggle Status	TACH Pin Status	Tach Out-of-Limit Status

TACH OUT-OF-LIMIT STATUS

This bit is set when the Tach Count value is greater than the high limit or less than the low limit. It is cleared when written with a 1. To disable this status event set the limits to their extreme values. If enabled via TACH Out-of-Limit Enable in the TACHx Control Register, this status bit will assert the Interrupt signal, which may be enabled to cause an interrupt event to the embedded controller.

0=Within Limits (TACH count value is less than or equal to the high limit or greater than or equal to the low limit).

1=Out of Limits (TACH count value is greater than the high limit or less than the low limit).

TACH PIN STATUS

This bit reflects the state of Tach Input. This bit is a read only bit that may be polled by the embedded controller.

0=Tach Input is low

1=Tach Input is high

TOGGLE STATUS

This bit is set when Tach Input changes state. It is cleared when written with a 1. If enabled via Tach Input INT_EN in the TACHx Control Register, this status bit will assert the Interrupt signal, which may be enabled to cause an interrupt event to the embedded controller.

0=Tach stable (default)

1=Tach Input changed state (this bit is set on a low-to-high or high-to-low transition)

APPLICATION NOTE: Some fans offer a Locked Rotor output pin that generates a level event if a locked rotor is detected. This bit may be used in combination with the tach pin status bit to detect a locked rotor signal event from a fan.

APPLICATION NOTE: Tach Input may come up as active for Locked Rotor events. This would not cause an interrupt event because the pin would not toggle. Firmware must read the status events as part of the initialization process, if polling is not implemented.

© 2012-2018 Microchip Technology Inc.

COUNT READY STATUS

The Count Ready Status bit remains cleared to '0' when the Tach Reading Mode Select bit in the TACHx Control Register is clear to '0'.

When the Tach Reading Mode Select bit in the TACHx Control Register is set to '1', The Count Ready Status bit is set when the counter value is latched by the hardware. It is cleared when written with a 1. If enabled via the Count Ready INT_EN bit in the TACHx Control Register, this status bit will assert the Interrupt signal, which may be enabled to cause an interrupt event to the embedded controller.

0=Reading not ready

1=Reading ready

30.7.1.3 TACHx High Limit Register

TABLE 30-6: TACHX HIGH LIMIT REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSETS	08h			32-bit			**EC SIZE**	
POWER	VTR			FFFFh			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE[3:2] BIT	**D31**	**D30**	**D29**	**...**		**D18**	**D17**	**D16**
HOST TYPE	R	R	R	R	R	R	R	R
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BYTE1-0 BIT	**D15**	**D14**	**D13**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	TACHx High Limit [15:0] Register							

The TACHx High Limit [15:0] value is compared with the value in the TACHx Control Register. If the value in the TACHx Control Register is greater than the value programmed in the TACHx High Limit Register the TACH Out-of_Limit STATUS bit will be set. The TACH Out-of-Limit status event may be enabled to generate an interrupt to the embedded controller via Bit[0] of the TACHx Control Register.

Note: To disable this event program FFFFh into this register.

© 2012-2018 Microchip Technology Inc.

30.7.1.4 TACHx Low Limit Register

TABLE 30-7: TACHX LOW LIMIT REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSETS	0Ch			32-bit			**EC SIZE**	
POWER	VTR			0000h			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE3-2 BIT	**D31**	**D29**	**D28**	**...**		**D18**	**D17**	**D16**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BIT	**D15**	**D14**	**D13**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	TACHx Low Limit [15:0] Register							

The TACHx Low Limit [15:0] value is compared with the value in the Tachx Counter Field of the TACHx Control Register. If the value in the Tachx Counter Field is less than the value programmed in the TACHx Low Limit Register the TACH Out-of_Limit STATUS bit will be set. The TACH Out-of-Limit status event may be enabled to generate an interrupt to the embedded controller via Bit[0] of the TACHx Control Register.

To disable this event program 0000h into this register.

31.0 PWM CONTROLLER

31.1 General Description

The function of this block is to generate a PWM output that may be used to control 4-wire fans, blink LEDs, etc. Each PWM can generate an arbitrary duty cycle output at frequencies from 0.095 Hz to 10 MHz.

The PWMx Counter ON Time registers and PWMx Counter OFF Time registers determine the operation of the PWM_OUTPUT signal. See Section 31.3.1, "PWMx Counter ON/OFF Time Registers," on page 464 for a description of the PWM_OUTPUT signal.

31.1.1 PWM_OUTPUT

The PWM_OUTPUT signal is used to generate a duty cycle at a frequency. This block has been designed such that the PWM signal may be programmed to hold PWM_OUTPUT high, to hold PWM_OUTPUT low, or to toggle PWM_OUTPUT. If the PWM is configured to toggle, then PWM_OUTPUT will alternate high and low for the programmed duration in the PWMx Counter ON/OFF Time Registers registers as defined in the register description. The PWM equations are described in Figure 31-2.

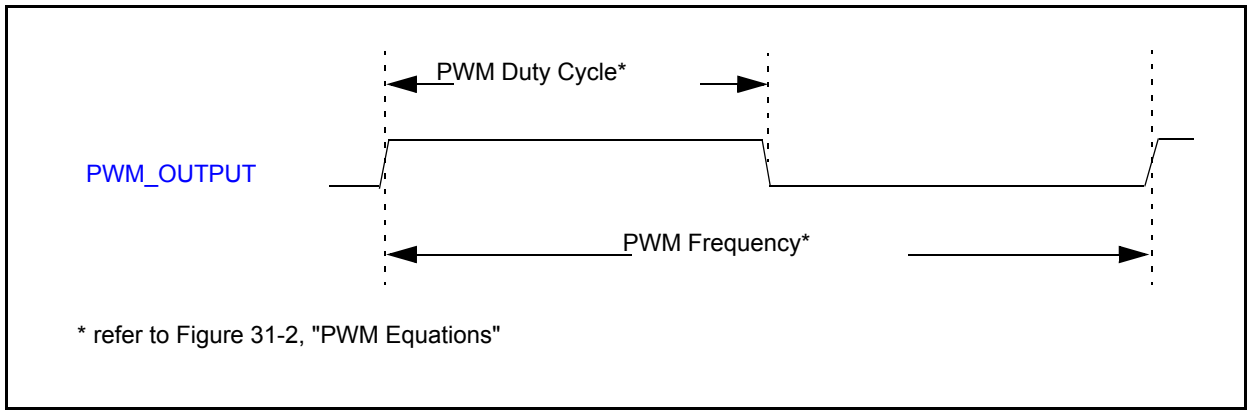
The PWM pin signal functions are routed pins described in the Fan PWM & Tachometer Interface section located in the Pin Configuration chapter.

31.1.2 PWM FEATURES

APPLICATION NOTE: Each PWM pin signal functions is muxed with a GPIO pin signal function. The pin's default signal function is GPIO input as controlled by the associated Pin Control Register. (See Section 24.0, "GPIO Interface," on page 388). At VTR POR or when a WDT event occurs (see Section 18.0, "Watchdog Timer Interface," on page 330,) the pin will tristate. For fan applications, an external resistor termination can provide the pin state to force the external fans to the full on state, thereby protecting the system from overheating.

31.1.3 PWM CONTROLLER BLOCK DIAGRAM

FIGURE 31-1: PWM FUNCTIONAL DIAGRAM



© 2012-2018 Microchip Technology Inc.

FIGURE 31-2: PWM EQUATIONS

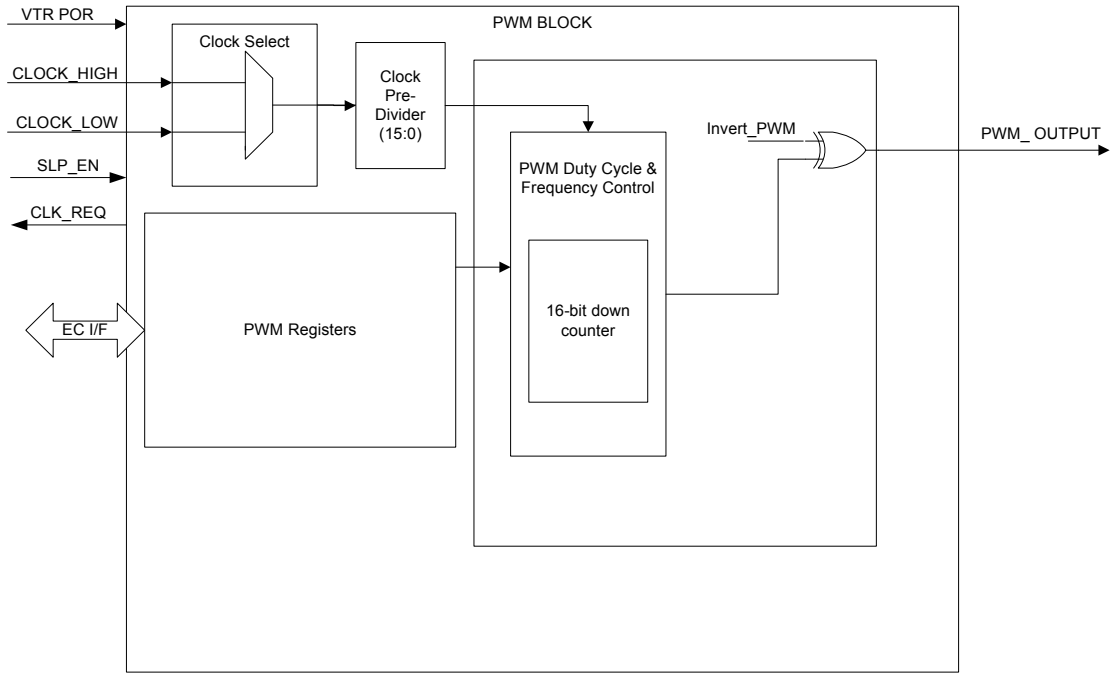
$$\text{PWM Frequency} = \frac{1}{(\text{PreDivider} + 1)} X \frac{(\overline{\text{Clock Select}} * 20.27\ \text{MHz} \ \&\ \text{Clock Select} * (20.27\ \text{MHz}/203))}{(\text{PWM Counter OFF Time}[15:0] + 1) + (\text{PWM Counter ON Time}[15:0] + 1)}$$

$$\text{PWM Duty Cycle} = \frac{(\text{PWM Counter ON Time}[15:0] + 1)}{(\text{PWM Counter OFF Time}[15:0] + 1) + (\text{PWM Counter ON Time}[15:0] + 1)}$$

Legend:
* = a Boolean AND operator
& = a Boolean OR operator

Notes:
Clock Select, PreDivider, PWM Counter OFF Time[15:0], and PWM Counter ON Time[15:0] are register values defined below.

FIGURE 31-3: BLOCK DIAGRAM OF PWM CONTROLLER



31.1.3.1 Block Diagram Signal List

TABLE 31-1: BLOCK DIAGRAM SIGNAL LIST DESCRIPTION

Signal Name	Direction	Description
VTR POR	INPUT	VTR Power on Reset.
CLOCK_HIGH	INPUT	20.27MHz MCLK.
CLOCK_LOW	INPUT	100KHz MCLK_DIV203_EN.
PWM_OUTPUT	OUTPUT	Pulse Width Modulated signal to PWMx pin.
E/C IF	I/O Bus	EC-side SPB bus.
SLP_EN	INPUT	Sleep Enable input.
CLK_REQ	OUTPUT	Clock Required output.

31.2 Power, Clocks and Reset

31.2.1 POWER DOMAIN

This block is powered by the VTR Power Supply.

See Section 7.5, "Power Configuration," on page 121 for details on power domains.

31.2.2 CLOCKS

This block uses the EC Bus Clock, the 20.27MHz MCLK and the 100KHz MCLK_DIV203_EN. The EC Bus Clock is used when reading and writing the PWM Controller control registers. The individual PWM counters can be driven either by MCLK or MCLK_DIV203_EN.

See Section 7.4, "Clock Generator," on page 98 for details on clocks.

31.2.2.1 Pre-Divider

The clock source to the PWM Down Counter used to generate a duty cycle and frequency on the PWM_OUTPUT may be pre-divided via bits D6:D3 in the PWMx Configuration Register. This results in a wide range of frequencies for the pwm output. Table 31-2 shows examples of frequencies supported.

TABLE 31-2: EXAMPLE OF PWM FREQUENCIES

Clock	Clock Select	Clock Re-Divider	High Count	Low Count	PWM Output Frequency (Hz)	PWM Output Duty Cycle
20.27 MHz	0	0	32767	32767	309	50%
20.27 MHz	0	0	192	192	52,500	50%
20.27 MHz	0	0	382	2	52,500	99%
20.27 MHz	0	0	960	960	10,500	50%
20.27 MHz	0	0	32767	32767	309	50%
100 KHz	1	1	32767	32767	0.76	50%
100 KHz	1	11	32767	32767	0.13	50%

31.2.2.2 Sleep Enable

The Embedded Controller can put each PWM into a sleep state. When a PWM is in the sleep state the internal counters are reset to 0 and the internal state of the PWM and thus the PWM_OUTPUT signal is set to the OFF state.

The PWM Controller clock required output (CLK_REQ) is the inversion of the sleep enable input (SLP_EN). The CLK_REQ output is not asserted when the PWM Controller is disabled.

The PWM participation in the sleep state is controlled by the PWMx bits in the EC Blocks Sleep Enables Register 1 and EC Blocks Sleep Enables Register 2.

© 2012-2018 Microchip Technology Inc.

31.2.3 RESET

This block is reset by nSYS_RST. After the assertion of nSYS_RST, PWM_OUTPUT is held in the OFF state and the hardware resets the pwm counter registers to their default value.

See Section 7.6, "Reset Interface," on page 124 for details on reset.

31.3 Registers

There are 16 instances of the PWM Controller block implemented in the MEC1632 enumerated [15:0]. Each instance of the PWM Controller has its Base Address as indicated in Table 31-3, "PWMx Controller Base Address Table":

TABLE 31-3: PWMX CONTROLLER BASE ADDRESS TABLE

PWM Controller **Instance**	**LDN from (Table 4-3 on page 60)**	**AHB Base Address**
PWM(7-0)**.0**	16h	F0_5800h
PWM(7-0)**.1**		F0_5880h
PWM(7-0)**.2**		F0_5900h
PWM(7-0)**.3**		F0_5980h
PWM(7-0)**.4**		F0_5A00h
PWM(7-0)**.5**		F0_5A80h
PWM(7-0)**.6**		F0_5B00h
PWM(7-0)**.7**		F0_5B80h
PWM(15-8)**.8**	17h	F0_5C00h
PWM(15-8)**.9**		F0_5C80h
PWM(15-8)**.10**		F0_5D00h
PWM(15-8)**.11**		F0_5D80h
PWM(15-8)**.12**		F0_5E00h
PWM(15-8)**.13**		F0_5E80h
PWM(15-8)**.14**		F0_5F00h
PWM(15-8)**.15**		F0_5F80h

Table 31-4 summarizes the registers allocated for each Instance. The offset field in the following table is the offset from the Embedded Controller (EC) Base Address.

TABLE 31-4: PWMX REGISTER SUMMARY

	EC Interface			
Register Name	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
PWMx Counter ON Time Register	00h	1-0	R/W	
PWMx Counter OFF Time Register	04h	1-0	R/W	
PWMx Configuration Register	08h	1-0	R/W	

TABLE 31-5: PWMX EC ACCESSIBLE REGISTERS

Offset	**Register Name**	**VTR POR (Suspend)**
0h	PWMx Counter ON Time Register	0000h
4h	PWMx Counter OFF Time Register	FFFFh
8h	PWMx Configuration Register	0000h

© 2012-2018 Microchip Technology Inc.

31.3.1 PWMX COUNTER ON/OFF TIME REGISTERS

TABLE 31-6: PWMX COUNTER ON TIME REGISTER

HOST ADDRESS	n/a				n/a			**HOST SIZE**
EC OFFSET	00h				16-bit			**EC SIZE**
POWER	VTR				0000h			nSYS_RST **DEFAULT**
BUS	EC SPB							
BIT	**D15**	**D14**	**D13**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	PWMx Counter ON Time[15:0]							

TABLE 31-7: PWMX COUNTER OFF TIME REGISTER

HOST ADDRESS	n/a				n/a			**HOST SIZE**
EC OFFSET	04h				16-bit			**EC SIZE**
POWER	VTR				FFFFh			nSYS_RST **DEFAULT**
BUS	EC SPB							
BIT	**D15**	**D14**	**D13**	**...**		**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	PWMx Counter OFF Time[15:0]							

The PWMx Counter ON/OFF Time registers determine both the duty cycle and frequency of the signal generated on PWM_OUTPUT. See FIGURE 31-2: PWM Equations on page 461.

If the PWMx Counter OFF Time[15:0] is set to zero, PWM_OUTPUT is held high (Full On). If the PWMx Counter ON Time is set to zero and the PWMx Counter OFF Time[15:0] is not set to zero, PWM_OUTPUT is held low (Full Off). Note that the default case is full off. Otherwise, both the high and low count registers will contain a value that will be used to determine the length of time PWM_OUTPUT will be held high and low. See Table 31-8, "PWM_OUTPUT State".

TABLE 31-8: PWM_OUTPUT STATE

PWM Count ON Time	PWM Count OFF Time	State of PWM_OUTPUT
Don't Care	0000h	Full On
0000h	Non-Zero Value	Full Off
Non-Zero Value	Non-Zero Value	Toggling On and Off

The counter values preload a 16-bit down-counter that is clocked by either the high frequency clock source or the low frequency clock source (see bit[1] CLK_Select of PWMx Configuration Register). The firmware will program the on and off count values that correspond to the PWM Current Duty Cycle and PWM Frequency. When PWM_OUTPUT is OFF and the internal counter is zero, the PWMx Counter ON Time is loaded into the counter. The PWM_OUTPUT signal will transition to the ON state and the internal counter will count down to zero at the programmed frequency for the duration of the programmed on time. Similarly, when the PWM_OUTPUT is in the ON state and the internal counter is zero, the PWMx Counter OFF Time is loaded into the counter. The PWM_OUTPUT signal will transition OFF and the internal counter will count down to zero at the programmed frequency for the duration of the programmed off time.

The PWMx Counter ON/OFF Time Registers may be updated at any time. Values written into the two registers are kept in holding registers. The holding registers are transferred into the PWMx Counter ON/OFF Time Registers when all four bytes have been written with new values and the internal counter completes the OFF time count. If the PWM is in the

© 2012-2018 Microchip Technology Inc.

Full On state then the PWMx Counter ON/OFF Time Registers are updated from the holding registers as soon as all four bytes have been written. Once the two registers have been updated the holding registers are marked empty. and all four bytes must again be written before the holding registers will be reloaded into the PWMx Counter ON/OFF Time Registers. Reads of the PWMx Counter ON/OFF Time Registers return the current contents of the registers that are used to load the counter and not the holding registers.

31.3.2 PWMX CONFIGURATION REGISTER

TABLE 31-9: PWMX CONFIGURATION REGISTER

HOST ADDRESS	n/a				n/a		**HOST SIZE**	
EC OFFSET	08h				16-bit		**EC SIZE**	
POWER	VTR				0000h		nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R							
BIT NAME	Reserved							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Reserved	Clock Pre-Divider				Invert	Clock Select	PWM Enable

PWM ENABLE

0= disabled (gates clocks to save power) (default)

1= enabled

Note:	When the PWM enable bit is set to 0 the internal counters are reset and the internal state machine is set to the OFF state. In addition, the PWM_OUTPUT signal is set to the inactive state as determined by the Invert bit. The PWMx Counter ON/OFF Time Registers are not affected by the PWM enable bit and may be read and written while the PWM enable bit is 0.

CLOCK SELECT

The Clk_Select bit determines the clock source used by the PWM duty cycle and frequency control logic.

0= 20.27MHz MCLK (default)

1= 100KHz MCLK_DIV203_EN

INVERT

0= PWM_OUTPUT ON State is active high

1= PWM_OUTPUT ON State is active low

CLOCK PRE-DIVIDER

The Clock source for the 16-bit down counter (see PWMx Counter ON/OFF Time Registers) is determined by bit D1 of this register. The Clock source is then divided by the value of Pre-Divider+1 and the resulting signal determines the rate at which the down counter will be decremented. For example, a Pre-Divider value of 1 divides the input clock by 2 and a value of 2 divides the input clock by 3. A Pre-Divider of 0 will disable the Pre-Divider option.

© 2012-2018 Microchip Technology Inc.

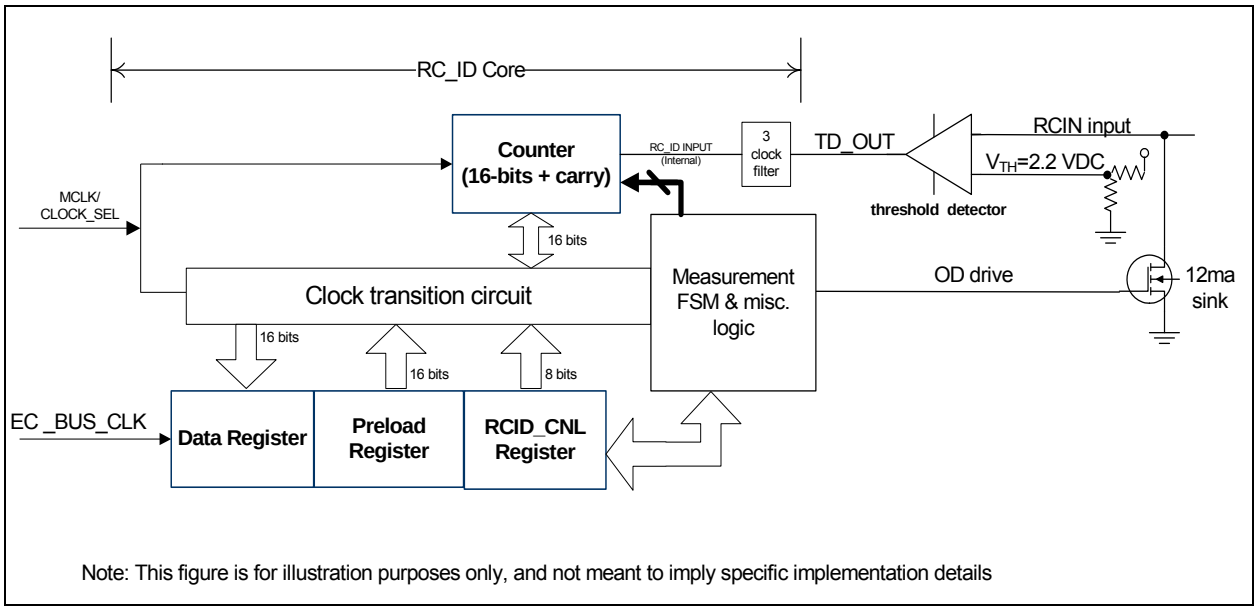
32.0 RC IDENTIFICATION DETECTION (RC_ID)

32.1 General Description

The Resistor/Capacitor Identification Detection (RC_ID) interface provides a single pin interface which can discriminate a number of quantized RC constants. The judicious selection of RC values can provide a low cost means for system element configuration identification. The RC_ID I/O pin measures the charge/discharge time for an RC circuit connected to the pin as shown in Figure 32-1.

32.2 Block Diagram

FIGURE 32-1: BLOCK DIAGRAM OF RC Identification Detection (RC_ID)



32.3 Power, Clocks and Reset

32.3.1 POWER DOMAIN

This block is powered by the VTR Power Supply.

See Section 7.1.1, "Power Configuration," on page 95 for details on power domains.

32.3.2 CLOCKS

This block uses the EC Bus Clock and MCLK. The EC Bus Clock is used to access the Registers described in this block. MCLK is divided down to provide a sampling clock.

See Section 7.1.2, "Clock Generator," on page 95 for details on clocks.

32.3.3 POWER ON RESET

This block is reset on a nSYS_RST. Following a system reset, all Register are reset to 00h and the state machines are set to Reset.

See Section 7.6, "Reset Interface," on page 124 for details on reset.

32.4 Interrupts

The RC Identification Detection (RC_ID) can generate an RCID_DONE interrupt when the DONE bit in the RCID_CTL Register is set. The interrupt source is routed onto the RCID bit in GIRQ16 Source Register on page 314 and is a level, active high signal.

© 2012-2018 Microchip Technology Inc.

32.5 Time Constants

This section lists a set of R and C values which can be connected to the RC_ID pin. Note that risetime generally follow RC time Tau. Firmware should use the Max and Min Limits to create quantized states.

32.5.1 SPECIFIC TIME CONSTANTS

TABLE 32-1: SAMPLE RC VALUES (C=2200 PF. R VARIED)

LIMITS				External Circuit Components		
MIN count	**MAX count**	**Range**	**Band Gap**	**C (pF)**	**R (K)**	
Avg-20%	Avg+10%	max-min	between RC values	**10%**	**5%**	
38	53	15		2200	1	
75	105	30	22	2200	2	
162	224	62	57	2200	4.3	
309	426	117	85	2200	8.2	
1202	1654	452	776	2200	33	
2244	3086	842	590	2200	62	
4538	6241	1703	1452	2200	130	
7999	11000	3001	1758	2200	240	
MIN	**MAX**	**Range**		**C (pF)**	**R (K)**	**Tau =RxC**
Risetime (usec)	**Risetime (usec)**	max-min		**10%**	**5%**	**(usec)**
1.88	3.00	1.12		2200	1	2.20
3.74	5.14	1.40		2200	2	4.40
8.01	11.02	3.01		2200	4.3	9.46
15.28	21.02	5.73		2200	8.2	18.04
59.31	81.56	22.24		2200	33	72.60
110.72	152.24	41.52		2200	62	136.40
223.89	307.84	83.96		2200	130	286.00
394.67	542.67	148.00		2200	240	528.00

TABLE 32-2: SAMPLE RC VALUES (C=3000 PF. R VARIED)

LIMITS				External Circuit Components		
MIN count	**MAX count**	**Range**	**Band Gap**	**C (pF)**	**R (K)**	
Avg-20%	Avg+10%	max-min	between RC values	**10%**	**5%**	
51	72	21		3000	1	
102	142	40	30	3000	2	
222	306	84	80	3000	4.3	
423	583	160	117	3000	8.2	
1682	2314	632	1099	3000	33	
3155	4339	1184	841	3000	62	
6344	8724	2380	2005	3000	130	
11156	15340	4184	2432	3000	240	
MIN	**MAX**	**Range**		**C (pF)**	**R (K)**	**Tau =RxC**
Risetime (usec)	**Risetime (usec)**	max-min		**10%**	**5%**	**(usec)**
2.55	4.00	1.45		3000	1	3.00
5.08	6.99	1.91		3000	2	6.00
10.96	15.07	4.11		3000	4.3	12.90
20.91	28.75	7.84		3000	8.2	24.60
83.02	114.16	31.13		3000	33	99.00
155.65	214.02	58.37		3000	62	186.00
313.01	430.38	117.38		3000	130	390.00
550.38	756.78	206.39		3000	240	720.00

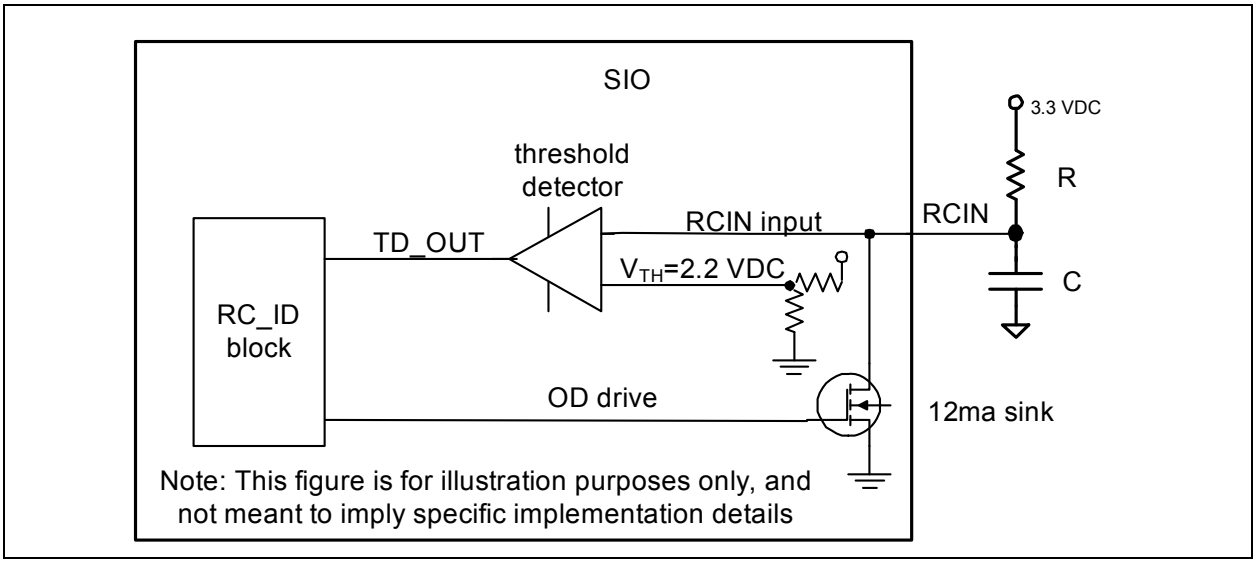
© 2012-2018 Microchip Technology Inc.

TABLE 32-3: SAMPLE RC VALUES (C=4700 PF. R VARIED)

LIMITS				External Circuit Components		
MIN count	**MAX count**	**Range**	**Band Gap**	**C (pF)**	**R (K)**	
Avg-20%	Avg+10%	max-min	between RC values	**10%**	**5%**	
80	112	32		4700	1	
160	222	62	48	4700	2	
344	474	130	122	4700	4.3	
659	907	248	185	4700	8.2	
2617	3600	983	1710	4700	33	
4862	6686	1824	1262	4700	62	
9866	13567	3701	3180	4700	130	
17384	23904	6520	3817	4700	240	
MIN Risetime (usec)	**MAX Risetime (usec)**	**Range**		**C (pF)**	**R (K)**	**Tau =RxC**
		max-min		**10%**	**5%**	**(usec)**
3.99	5.60	1.61		4700	1	4.70
7.94	10.95	3.01		4700	2	9.40
17.00	23.65	6.65		4700	4.3	20.21
32.53	44.88	12.35		4700	8.2	38.54
129.14	176.55	47.41		4700	33	155.10
239.86	328.79	88.93		4700	62	291.40
486.74	648.23	161.49		4700	130	611.00
857.64	1140.70	283.06		4700	240	1128.00

32.6 Block Diagram

FIGURE 32-2: RCID CIRCUIT.INTERFACING TO THE RC_ID BLOCK



The RC_ID block initiates the discharging followed by the charging of the external RC circuit (see Figure 32-1. At the same time, the RCID input goes through a threshold detector set at 68% of 3.3 VDC. The TD_OUT gates the 16 bit Counter clocked by MCLK divided down by the Clock Select Field in the RCID_CTL Register on page 472. The input has an input glitch rejection filter. Any change in input less than 3 MCLK-wide is ignored.

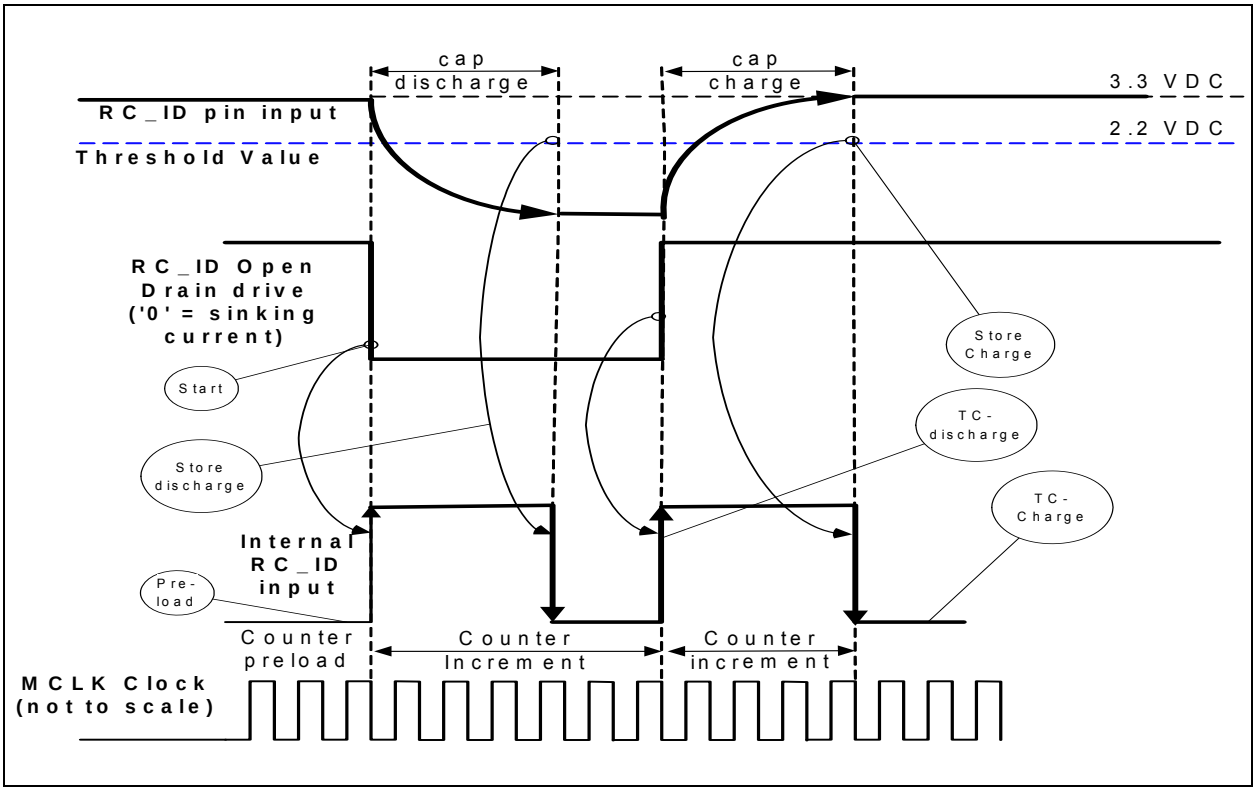
APPLICATION NOTE: After completion of a measurement cycle with the DONE bit set to '1' in the RCID_CTL Register, the programmer must place the RC_ID back into the reset state to before starting a new measurement.

TABLE 32-4: RC_ID MEASUREMENT STATES

State	Description
Reset	The RCID_CTL Register ENABLE bit is cleared to '0' by nSYS_RST or a write. The pin OD output driver is tristated, the RCID blocked is placed in low power mode. The DONE, CY_ER, TC bits in the RCID_CTL Register are autonomously cleared to '0'. **Note:** Clearing the ENABLE bit with a value other than 00h to the RCID_CTL Register is not defined and may create unpredictable results.
Enabled	Setting the ENABLE bit to '1' in the RCID_CTL Register places the RC_ID interface active high power state. **Note:** A 300us delay is required between the Enabled state and the Start state.
Start	The Start state is initiated by a write to the RCID_CTL Register setting the START bit to '1'. The counter is initiated to the preload value of 0000h and starts incrementing. The pin OD driver begins to sink current and external capacitor starts discharging. The DONE, CY_ER, TC bits in the RCID_CTL Register are autonomously cleared to '0'.
Detect Discharge	The pin voltage decays as the external capacitor discharges and the counter continues to increment until the terminal count is reached. The pin voltage is monitored to detect the discharge voltage reaches below the threshold voltage before the counter reaches terminal count.
TC-Discharged	The incrementing counter reaches the terminal count value of FFFFh. The TC bit in the RCID_CTL Register is autonomously set to '1'. If the pin voltage fails to discharge below the threshold voltage before the counter reaches terminal count during Detect Discharge, then CY_ER bit in the RCID_CTL Register is autonomously set to '1'; otherwise, then CY_ER bit remains clear. The pin OD output driver is tristated and the counter starts incrementing from 0000h.
Detect Charge	The pin voltage rises to the threshold voltage, counter stops counting, and the present value of the counter is stored in the RC_ID Data Register. The DONE bit in the RCID_CTL Register is autonomously set to '1'.
TC-Charged	The incrementing counter reaches the terminal count value of FFFFh and the pin voltage is below the threshold value. The CY_ER bit and DONE bits in the RCID_CTL Register are autonomously set to '1'. **Note:** The Detect Charge and the TC-Discharged STATES are mutually exclusive.

© 2012-2018 Microchip Technology Inc.

FIGURE 32-3: RC_ID TIMING DIAGRAMRC_ID OPERATION AND TIMINGS



All registers are VTR powered and are placed in reset when nSYS_RST (internal signal) is ‘0’.

32.7 Registers

Each instance of the RC Identification Detection (RC_ID) has its own Logical Device Number, and Base Address as indicated in Table 32-5.

TABLE 32-5: RC Identification Detection (RC_ID) BASE ADDRESS TABLE

RC ID Instance	LDN from (Table 4-2 on page 59)	AHB Base Address
RC ID	4h	F0_1000h

Table 32-6 is a register summary for this instance of the RC Identification Detection (RC_ID).

TABLE 32-6: RC Identification Detection (RC_ID) REGISTER SUMMARY

	EC Interface			
Register Name	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
RCID_CTL Register	00h	0	R/W	
RC_ID Data Register	04h	0	R	
		1		

32.7.1 RC_ID CONTROL REGISTER

TABLE 32-7: RCID_CTL REGISTER

HOST OFFSET	N/A						**HOST SIZE**	
EC OFFSET	00h					16-Bit	**EC SIZE**	
POWER	VTR					00h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R/W	R/W
BIT NAME	Reserved						Clock_Sel	
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R	R	R	R	R	R
BIT NAME	ENABLE	START	Reserved			CY_ER	TC	DONE

DONE

This read only status is set when the RCID completes a measurement and enters the “Detect Charge” or “TC-Charged” Measurement States described in Table 32-4, “RC_ID Measurement States,” on page 470.

This bit is cleared when RCID enters the “Reset” or “Start” Measurement States described in Table 32-4, “RC_ID Measurement States,” on page 470.

TC

This read only status bit is set when the RCID enters the “TC-Discharged” or “TC-Charged” Measurement State described in Table 32-4, “RC_ID Measurement States,” on page 470.

This bit is cleared when RCID enters the “Reset” or “Start” Measurement States described in Table 32-4, “RC_ID Measurement States,” on page 470.

CY_ER

This bit is a read only status bit and indicates when set to ‘1’ that the counter reached terminal count during the Capacitive Discharge or Charge phases without crossing the voltage threshold. This is an error condition.

START

Setting this bit to ‘1’ causes the RCID to enter the “Start” RCID Measurement State described in Table 32-4, “RC_ID Measurement States,” on page 470. A 300us delay is required between the Enabled state and the Start state.

All writes to this register during other RC_ID states should clear this bit to 00h. See Note 32-1.

ENABLE

Clearing this bit to ‘0’ causes the RCID to enter the “Reset” RCID Measurement State described in Table 32-4, “RC_ID Measurement States,” on page 470.

Setting this bit to ‘1’ starts the clock input to the RCID and arms the counter.

APPLICATION NOTE: The ENABLE bit should remain set during the entire measurement; therefore all writes to the RCID_CTL Register during a measurement should set this bit.

Note 32-1 When writing to the RCID_CTL Register to clear the ENABLE bit, the START bit should be cleared to ‘0’. Clearing the ENABLE bit with a value other than 00h is not defined and may create unpredictable results.

© 2012-2018 Microchip Technology Inc.

CLOCK_SEL

This field selects the frequency of the Counter circuit clock. Table 32-8, "Clock Select Field" shows the clock frequencies that can be selected:

TABLE 32-8: CLOCK SELECT FIELD

Clock_Sel	Counter Clock		Resolution per Bit	Full Count Duration
0	DIVIDE BY 1	20.27 MHz	49.33 ns	3.23 ms
1	DIVIDE BY 2	10.14 MHz	98.67 ns	6.47 ms
2	DIVIDE BY 4	5.07 MHz	197.34 ns	12.93 ms
3	DIVIDE BY 8	2.53 MHz	394.67 ns	25.86 ms

The values in the Clock Select Field should only be changed when the ENABLE bit in the RCID_CTL Register is cleared to '0'.

32.7.2 RC_ID DATA REGISTER

The RC_ID Data Register provides a 16 bit counter value with a 1/MCLK Resolution per bit.

Reads from this register in the Detect Charge Measurement States described in Table 32-4, "RC_ID Measurement States," on page 470 provides the bytes of the measured result for the Charge time.

TABLE 32-9: RC_ID DATA REGISTER

HOST ADDRESS							**HOST SIZE**	
EC OFFSET	04h					16-bit	**EC SIZE**	
POWER	VTR					0000h	EC SPB **DEFAULT**	
BUS	EC SPB							
BYTE1 BIT	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Data[15:8]							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Data[7:0]							

32.8 Low Power Mode

The RC Identification Detection (RC_ID) interface is designed to conserve power when sleeping or disabled. Table 32-10 summarizes the RC Identification Detection (RC_ID) interface Low Power Mode behavior.

TABLE 32-10: BLOCK CLOCK GATING IN LOW POWER MODES

ENABLE Bit	DONE Bit	RCID_SLEEP_EN	Block Idle Status (Note 32-2)	RCID_CLOCK_REQ	State	Description
0	X	X	X	0	SLEEPING	The block is disabled and the clock can be stopped.
1	0	0	NOT IDLE	1	NORMAL OPERATION	The block is not idle and neither disabled by firmware nor commanded to sleep.
1	0	1	NOT IDLE	1	PREPARING TO SLEEP	The block is commanded to sleep, but the clock is required until the Block is idle.
1	1	1	IDLE	0	SLEEPING	The block is commanded to sleep and idle. The clock can be stopped.

Note 32-2 The DONE bit indicates the RC Identification Detection (RC_ID) interface 'idle' state.

© 2012-2018 Microchip Technology Inc.

33.0 GENERAL PURPOSE SERIAL PERIPHERAL INTERFACE (GP-SPI)

33.1 General Description

The SPI interfaces may be used to communicate with various peripheral devices, e.g., EEPROMS, DACs, ADCs, that use a standard Serial Peripheral Interface. There are two instances of GP-SPI controller, one located on the EC SPB bus and the other on the LPC SPB bus. The latter is intended for flash access by the host and EC; it can optionally work in conjunction with the DMA Controller to move data to and from the closely coupled SRAM with minimal software overhead.

Characteristics of the GP-SPI Controller include:

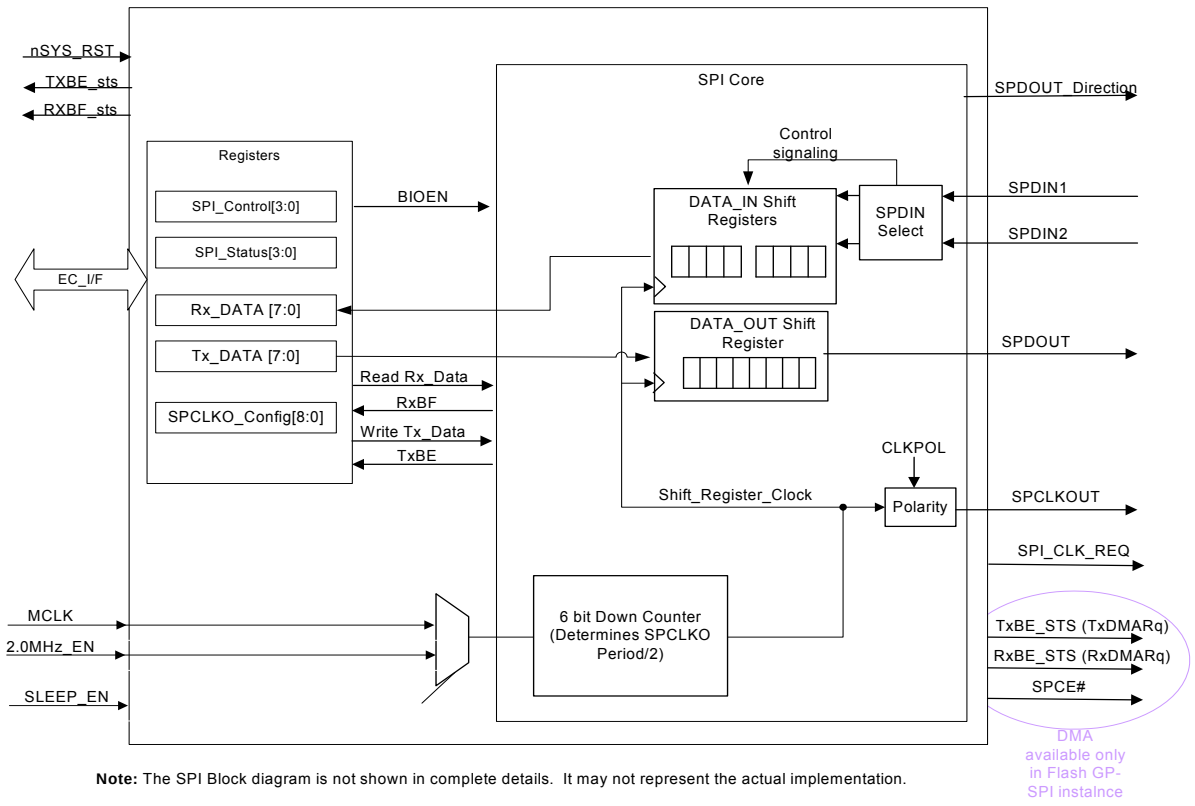
- 8-bit serial data transmitted and received simultaneously over two data pins in Full Duplex mode with options to transmit and receive data serially on one data pin in Half Duplex (Bidirectional) mode.
- An internal programmable clock generator and clock polarity and phase controls allowing communication with various SPI peripherals with specific clocking requirements.
- SPI cycle completion that can be determined by status polling or interrupts.
- The ability to read data in on both SPDIN and SPDOUT in parallel. This allows this SPI Interface to support dual data rate read accesses for emerging double rate SPI flashes
- Support of back-to-back reads and writes without clock stretching, provided the host can read and write the data registers within one byte transaction time.
- Hardware hooks to DMA Engine (available only in the Flash GP-SPI on the LPC SPB bus).

The MEC1632 SPI is a master only device and does not support multiple-master SPI configurations.

The GP-SPI controller on LPC SPB bus has its IO signals (pins) multiplexed with those from of the EC AHB SPI Flash Read Controller whose Master Bridge Enable register bit controls the multiplexer. See SPI Controllers Interface section located in the Pin Configuration chapter for pins listing.

33.2 SPI Block Diagram

FIGURE 33-1: SPI BLOCK DIAGRAM



Note: The SPI Block diagram is not shown in complete details. It may not represent the actual implementation.

33.3 SPI Block Diagram Signal Description

TABLE 33-1: SPI BLOCK SIGNALS

Signal	Direction	Description
nSYS_RST	INPUT	VTR Power on Reset.
TXBE_STS	OUTPUT	Interrupt output to EC driven by TXBE status bit
RXBF_STS	OUTPUT	Interrupt output to EC driven by RXBF status bit
MCLK	INPUT	Clock input to SPI Interface logic
2.0MHz_EN	INPUT	Clock Enable input to SPI Interface logic
SPDOUT	OUTPUT	Serial Data Out to the SPDOUT pin
SPDIN1	INPUT	Serial Data In 1 from SPPDIN pin. Input in full-duplex mode and Dual Read mode
SPDIN2	INPUT	Serial Data In 2 from SPDOUT pin. Input in bi-directional mode and Dual Read mode.
SPI_CLK	OUTPUT	SPI Clock output used to drive the SPCLK pin.

© 2012-2018 Microchip Technology Inc.

TABLE 33-1: SPI BLOCK SIGNALS (CONTINUED)

Signal	Direction	Description
SPDOUT_Direction	OUTPUT	The SPDOUT pin may be used as an output or an input. This signal is used to determine the direction of the SPDOUT buffer. 0=output (SPDOUT pin is controlled by the SPDOUT signal) 1=input (SPDOUT pin is an input driving the SPDIN2 signal) The SPDOUT pin has I/O capability. The I/O capability is implemented to support the Half-Duplex mode of operation (also referred to as bi-directional mode) and the Dual Read mode.
SLEEP_EN	INPUT	External enable/disable signal used to put the block in the lowest power consumption state. 0=No Sleep Requested. The block should operate as configured. 1=Sleep Requested. The block enters sleep mode. See Low Power Mode on page 480.
SPI_CLK_REQ	OUTPUT	This output indicates when this block requires this clock input. 0= MCLK can be turned 'off' when appropriate 1= MCLK is required to be 'on.'
LPC/EC SPB Bus IF	I/O Bus	Bus used by microprocessor to access the registers in this block.
SPI_CS#	OUTPUT	SPI chip select (only for the Flash GP-SPI on LPC SPB bus)
TxBE_STS	OUTPUT	Tx DMA request (only for the Flash GP-SPI on LPC SPB bus)
RxBE_STS	OUTPUT	Rx DMA request (only for the Flash GP-SPI on LPC SPB bus)

33.4 SPI Interface Signals

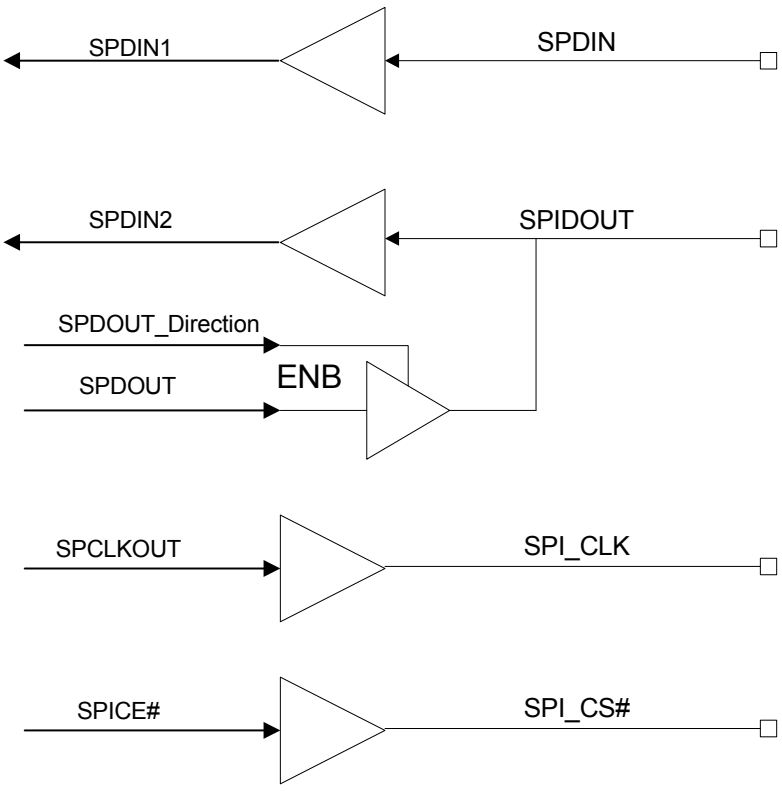
The following subsections describe the SPI Block Signals that are routed to the SPI pins. This chapter utilizes generic signal nomenclature for the Pin Signal Functions. Table 33-2 is a specific lookup table for pin signal function names used in this chapter and used elsewhere. Figure 33-2 show typical routing of the block interface (illustrated in Figure 33-1 to the pins. The Miscellaneous Functions table located in the Pin Configuration chapter includes the pin description for the SPI interface.

Since there is only one instance of SPI port in the MEC1632, there is a one to one correspondence.

TABLE 33-2: PIN SIGNAL FUNCTION NOMENCLATURE LOOKUP TABLE

Routing Figure Generic Pin Signal Name	Pin Signal Function Name Used in Other Contexts	Pin Function Signal Description
SPCLK	ECGP_SCLK	General Purpose SPI Clock
SPDOUT	ECGP_SOUT	General Purpose SPI Output
SPDIN	ECGP_SIN	General Purpose SPI Input

FIGURE 33-2: TYPICAL BLOCK/PIN INTERFACE



Note: This figure is for illustration purposes only, and not meant to imply specific implementation details

33.4.1 SPDOUT PIN - SERIAL PERIPHERAL DATA OUT

In Full Duplex Mode, this is the serial data output to the SPI interface. In half-duplex mode (also referred to as bi-directional mode) this is the serial data I/O port for the SPI interface.

For special SPI Flash devices that support Dual Read Modes the SPDOUT operates as in input in parallel with the SPDIN during the data portion of the Fast Dual Read command.

Note:	In the Bi-directional mode, some slave devices may tristate the last few bits to signal a turn-around; therefore, an external weak pull-up may be required on the pin.

33.4.2 SPDIN PIN - SERIAL PERIPHERAL DATA IN

In Full Duplex Mode, this is the serial data input from the SPI interface. In half-duplex mode (also referred to as bi-directional mode) this pin is unused.

Note:	Some slave device may tristate the SPDIN pin during command phase; therefore, an external weak pull-up or pull-down may be required on the pin.

© 2012-2018 Microchip Technology Inc.

33.4.3 SPCLK PIN - SERIAL PERIPHERAL CLOCK

This is the serial clock driven by the MEC1632 SPI (master) and connected to all SPI slaves. All data (input and output) is sampled/shifted on SPCLK according to the clock controls CLKPH and CLKPOL (See TCLKPH and CLKPOL in Section 33.11.6, "SPICC - SPI Clock Control Register," on page 491).

Note 33-1 In the MEC1632, the General Purpose Serial Peripheral Interface (GP-SPI) pins are 8 mA buffers. The maximum SPCLK pin clock frequency is MCLK/2 for all modes. Limited functionality is available when SPCLK pin clock frequency is MCLK although performance is not guaranteed. See TABLE 33-14: on page 493 and Section 33.9.5.5, "Limits of SPI configurations," on page 487.

33.5 Power, Clocks and Reset

33.5.1 POWER DOMAIN

This block is powered by the VTR Power Supply.

See Section 7.1.1, "Power Configuration," on page 95 for details on power domains.

33.5.2 CLOCKS

This block uses the EC Bus Clock / LPC Bus Clock, MCLK and the 2MHz (MCLK_DIV10_EN). EC Bus Clock / LPC Bus Clock is used when reading and writing the General Purpose Serial Peripheral Interface (GP-SPI) registers. The 6-bit down counter may use either MCLK or MCLK_DIV10_EN to directly decrement the counter, which is the SPI_CLK source.

See Section 7.4, "Clock Generator," on page 98 for definition and details on clocks of: MCLK on page 120, MCLK_DIV10_EN on page 120, and EC Bus Clock on page 120.

33.6 Reset

This block is reset on a nSYS_RST. On reset, the General Purpose SPI interface defaults to disabled, The block can also be reset by software, by setting the Soft Reset bit located in the SPICR - SPI Control Register. Setting this bit reinitializes the SPI Control block back to its nSYS_RST state.

See Section 7.6, "Reset Interface," on page 124 for definition and details on reset: nSYS_RST on page 128.

33.7 SPI Interrupts / DMA Requests

The General Purpose Serial Peripheral Interface (GP-SPI) can generate an interrupt events to the Embedded Controller (EC) to indicate that the block requires servicing. The SPI TXBE status and RXBF status bits in the SPISR - SPI Status Register are routed onto the SPI_TXBE_GP & SPI_RXBF_GP bits of the GIRQ14 Source Register. In the Flash GP-SPI instance, these status bits are also connected respectively to the DMA Controller's SPI Flash Write and Read requests signals.

© 2012-2018 Microchip Technology Inc.

33.8 Low Power Mode

This block is designed to conserve power when it is either sleeping or disabled. There are two ways to put the SPI interface into a low power mode: Disabled the SPI Interface via the Enable Bit or Assert the SLEEP_EN signal to the SPI Interface. The following table summarizes the SPI behavior for each of these low power modes.

TABLE 33-3: BLOCK CLOCK GATING IN LOW POWER MODES

Enable Bit	SLEEP_EN	Block Idle Status	SPI_CLK_REQ	State	Description
0	X	NOT IDLE	1	PREPARING to SLEEP	The core clock is required until the current transaction is completed and the Block is IDLE.
		IDLE	0	SLEEPING	The block is idle and the core clock can be stopped.
1	0	X	1	NORMAL OPERATION	The block in neither disabled by firmware nor commanded to SLEEP
	1	NOT IDLE	1	PREPARING to SLEEP	The core clock is required until the current transaction is completed and the Block is IDLE.
		IDLE	0	SLEEPING	The block is idle and the core clock can be stopped.

33.8.1 DISABLING THE SPI INTERFACE BLOCK VIA THE ENABLE BIT

The enable bit is located in Section 33.11.1, "SPIAR - SPI Enable Register," on page 488. When this bit is cleared the SPI interface is in its lowest power state. The MCLK clock input is gated and the SPDOUT and SPI_CLK pins are set to their inactive state as determined by the configuration bits.

Note: The SPI Interface is required to finish the current transaction and enter the Idle state before deasserting the SPI_CLK_REQ signal and gating its internal clock source.

33.8.2 ASSERTING THE SLEEP_EN SIGNAL TO THE SPI INTERFACE BLOCK

When the SLEEP_EN signal is asserted the SPI interface completes the current transaction and then enters the low power state. In the low power state the MCLK clock input is gated, the SPDOUT and SPI_CLK pins are set to their inactive state as determined by the configuration bits, and the SPI_CLK_REQ signal is de-asserted.

33.9 Operation

The Serial Peripheral Interface (SPI) block is a master SPI block used to communicate with external SPI devices. The SPI master is responsible for generating the SPI clock and is designed to operate in Full Duplex, Half Duplex, and Dual modes of operation. The clock source may be programmed to operated at various clock speeds. The data is transmitted serially via 8-bit transmit and receive shift registers. Communication with SPI peripherals that require transactions of varying lengths can be achieved with multiple 8-bit cycles.

This block has many configuration options: The data may be transmitted and received either MSbit or LSbit first; The SPI Clock Polarity may be either active high or active low; Data may be sampled or presented on either the rising of falling edge of the clock (referred to as the transmit clock phase); and the SPI_CLK SPDOUT frequency may be programmed to a range of values as illustrated in Table 33-14, “SPI_CLK Frequencies,” on page 493. In addition to these many programmable options, this feature has several status bits that may be enabled to notify the host that data is being transmitted or received.

33.9.1 INITIATING AN SPI TRANSACTION

All SPI transactions are initiated by a write to the TX_DATA register. No read or write operations can be initiated until the Transmit Buffer is Empty, which is indicated by a one in the TXBE status bit.

© 2012-2018 Microchip Technology Inc.

If the transaction is a write operation, the host writes the TX_DATA register with the value to be transmitted. Writing the TX_DATA register causes the TXBE status bit to be cleared, indicating that the value has been registered. If empty, the SPI Core loads this TX_DATA value into an 8-bit transmit shift register and begins shifting the data out. Loading the value into the shift register causes the TXBE status bit to be asserted, indicating to software that the next byte can be written to the TX_DATA register.

If the transaction is a read operation, the host initiates a write to the TX_DATA register in the same manner as the write operation. Unlike the transmit command, the host must clear the RXBF status bit by reading the RX_DATA register before writing the TX_DATA register. This time, the host will be required to poll the RXBF status bit to determine when the value in the RX_DATA register is valid.

Note:

- If the SPI interface is configured for Half Duplex mode, the host must still write a dummy byte to receive data.
- Since RX and TX transactions are executed by the same sequence of transactions, data is always shifted into the RX_DATA register. Therefore, every write operation causes data to be latched into the RX_DATA register and the RXBF bit is set. This status bit should be cleared before initiating subsequent transactions. The host utilizing this SPI core to transmit SPI Data must discard the unwanted receive bytes.
- The length and order of data sent to and received from a SPI peripheral varies between peripheral devices. The SPI must be properly configured and software-controlled to communicate with each device and determine whether SPIRD data is valid slave data.

The following diagrams show sample single byte and multi-byte SPI Transactions.

FIGURE 33-3: SINGLE BYTE SPI TX/RX TRANSACTIONS (FULL DUPLEX MODE)

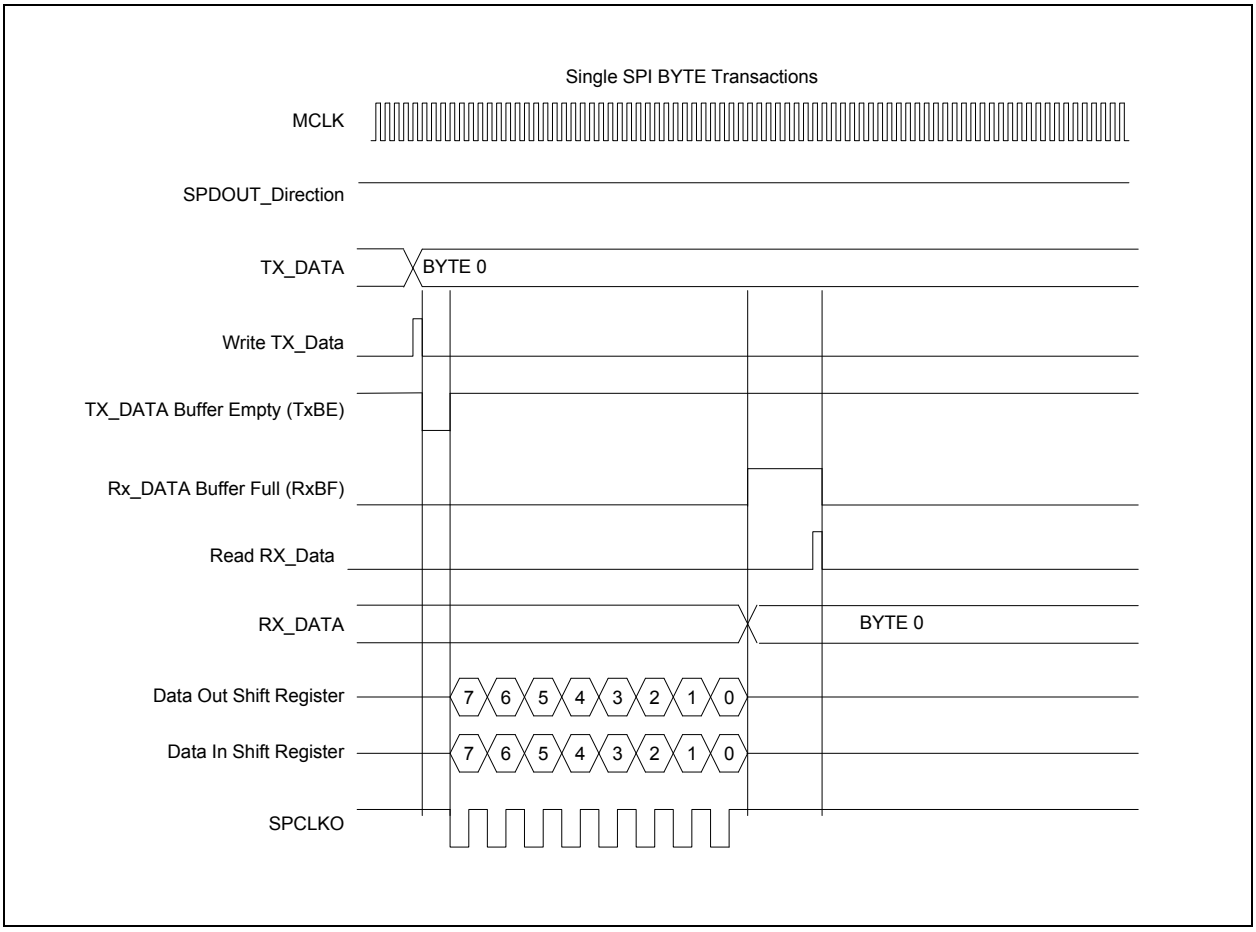
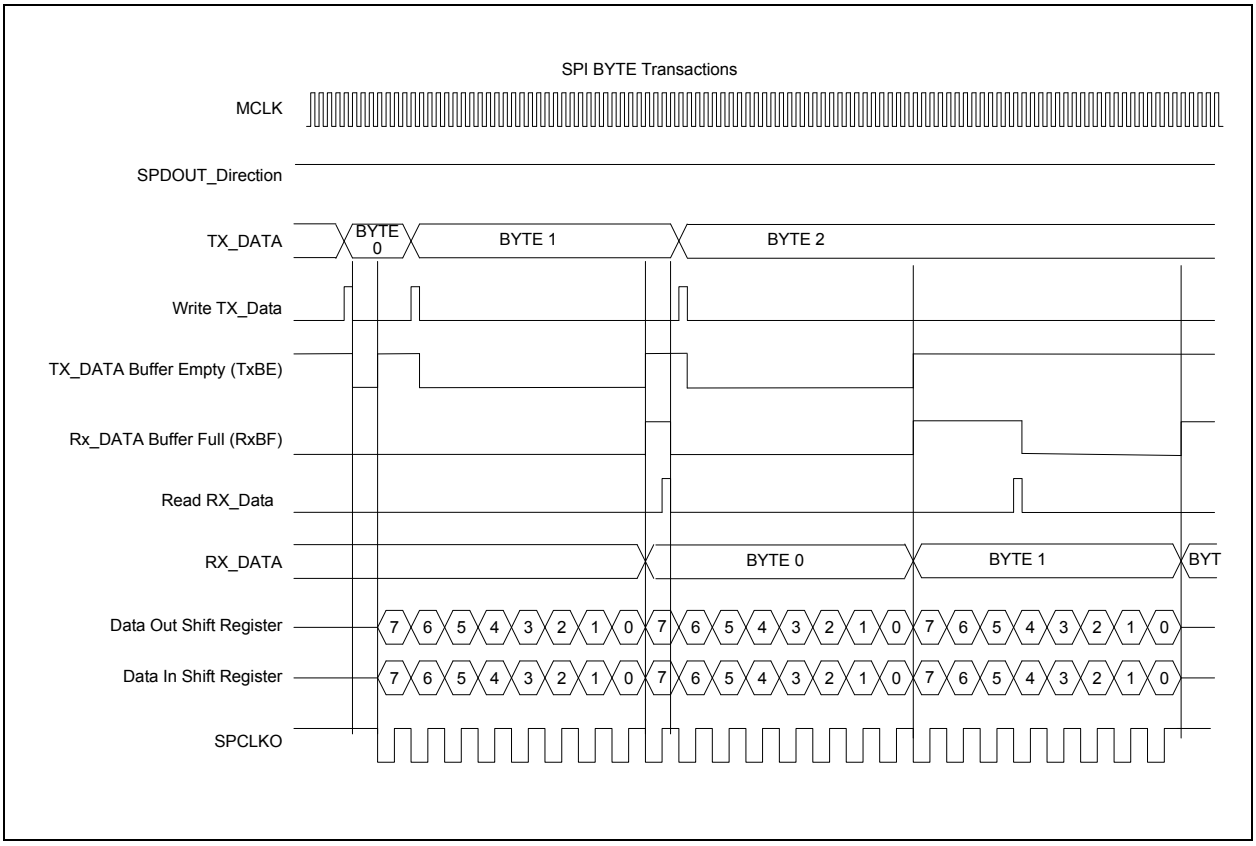


FIGURE 33-4: MULTI-BYTE SPI TX/RX TRANSACTIONS (FULL DUPLEX MODE)



The data may be configured to be transmitted MSB or LSB first. This is configured by the LSBF bit in the Section 33.11.2, "SPICR - SPI Control Register," on page 488. The transmit data is shifted out on the edge as selected by the TCLKPH bit in the SPICC register. See Section 33.11.6, "SPICC - SPI Clock Control Register," on page 491. All received data can be sampled on a rising or falling SPI_CLK edge using RCLKPH (see RCLKPH in Section 33.11.6, "SPICC - SPI Clock Control Register," on page 491 for clock controls). This clock setting must be identical to the clocking requirements of the current SPI slave.

Note:	Common peripheral devices require a chip select signal to be asserted during a transaction. Chip selects for SPI devices may be controlled by MEC1632 GPIO pins.

There are three types of transactions that can be implemented for transmitting and receiving the SPI data. They are Full Duplex, Half Duplex, and Dual Mode. These modes are define in Section 33.9.3, "Types of SPI Transactions," on page 483.

33.9.2 DMA MODE (FLASH GP-SPI ONLY)

Transmit and receive operations can use a DMA channel. Note that only one DMA channel may be enabled at a time. Setting up the DMA Controller involves specifying the device (Flash GP-SPI), direction (transmit/receive), and the start and end addresses of the DMA buffers in the closely couple memory. Please refer to the DMA Controller chapter for register programming information.

SPI transmit / DMA write: the GP-SPI block's transmit empty (TxBE) status signal is used as a write request to the DMA controller, which then fetches a byte from the DMA transmit buffer and writes it to the GP-SPI's SPI TX Data Register (SPITD). As content of the latter is transferred to the internal Tx shift register from which data is shifted out onto the SPI

© 2012-2018 Microchip Technology Inc.

bus bit by bit, the Tx Empty signal is again asserted, triggering the DMA fetch-and-write cycle. The process continues until the end of the DMA buffer is reached - the DMA controller stops responding to an active Tx Empty until the buffer's address registers are reprogrammed.

SPI receive / DMA read: the AUTO_READ bit in the SPI Control Register must be set. The driver first writes (dummy data) to the SPI TX Data Register (SPITD) to initiate the toggling of the SPI clock, enabling data to be shifted in. After one byte is received, the Rx Full (RxBF) status signal, used as a read request to the DMA controller, is asserted. The DMA controller then reads the received byte from the GP-SPI's SPI RX Data Register (SPIRD) and stores it in the DMA receive buffer. With AUTO_READ set, this read clears both the RxBF and TxBE. Clearing TxBE causes (dummy) data from the SPI TX Data Register (SPITD) to be transferred to the internal shift register, mimicking the effect of the afore-mentioned write to the SPI TX Data Register (SPITD) by the driver. SPI clock is toggled again to shift in the second read byte. This process continues until the end of the DMA buffer is reached - the DMA controller stops responding to an active Tx Empty until the buffer's address registers are reprogrammed.

33.9.3 TYPES OF SPI TRANSACTIONS

The MEC1632 SPI can be configured to operate in three modes: Full Duplex, Half Duplex, and Dual Mode.

33.9.3.1 Full Duplex

In Full Duplex Mode, serial data is transmitted and received simultaneously by the SPI master over the SPDOUT and SPDIN pins. To enable Full Duplex Mode clear SPDIN Select.

When a transaction is completed in the full-duplex mode, the RX_DATA shift register always contains received data (valid or not) from the last transaction.

33.9.3.2 Half Duplex

In Half Duplex Mode, serial data is transmitted and received sequentially over a single data line (referred to as the SPD-OUT pin). To enable Half Duplex Mode set SPDIN Select to 01b. The direction of the SPDOUT signal is determined by the BIOEN bit (See Section , "BIOEN," on page 489).

- To transmit data in half duplex mode set the BIOEN bit before writing the TX_DATA register.
- To receive data in half duplex mode clear the BIOEN bit before writing the TX_DATA register with a dummy byte.

Note:	The Software driver must properly drive the BIOEN bit and store received data depending on the transaction format of the specific slave device.

33.9.3.3 Dual Mode of Operation

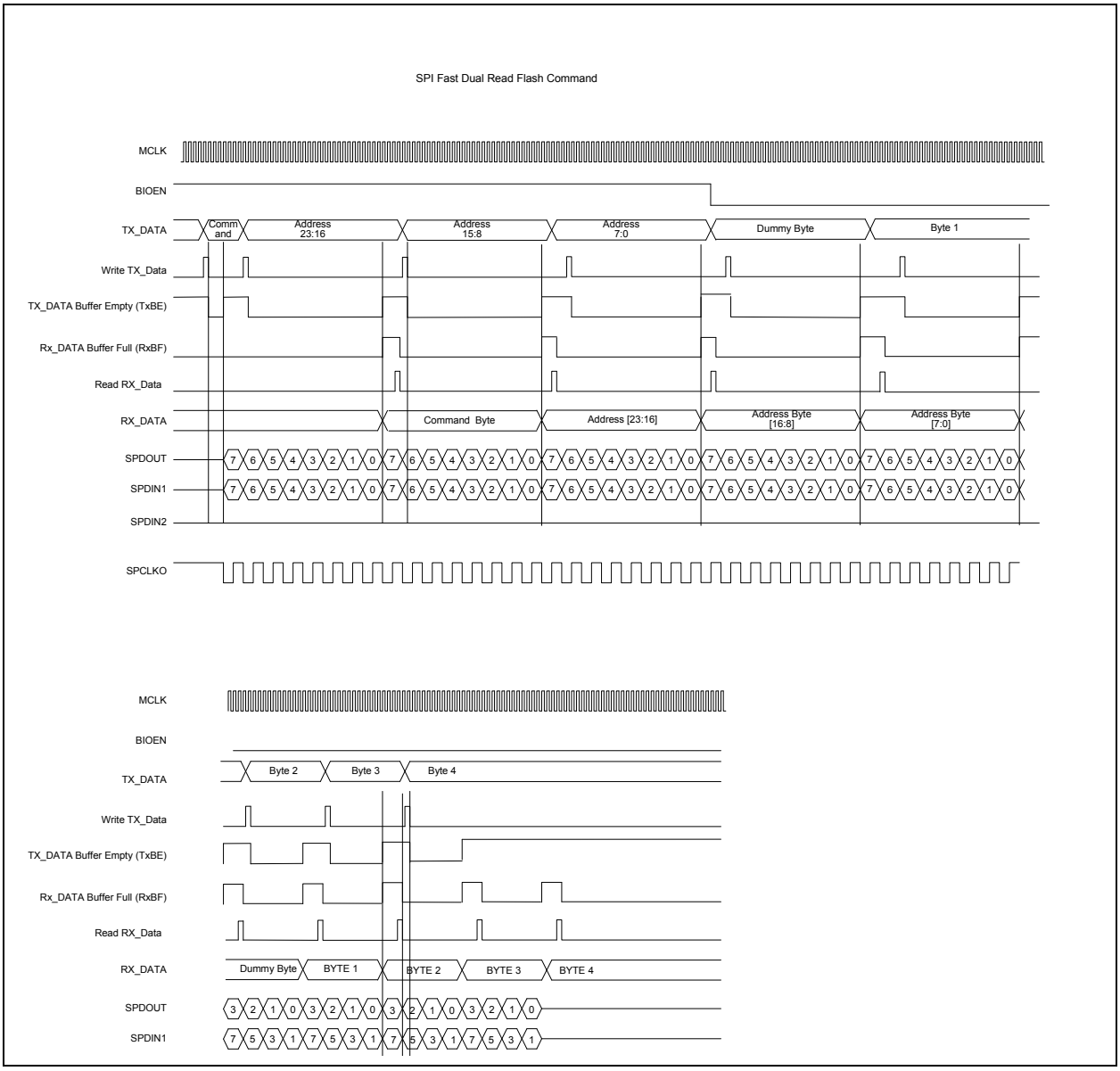
Note:	The Dual Mode of Operation has been implemented to support selected SPI Flash devices that support the Fast Dual Mode command.

In Dual Mode, serial data is transmitted sequentially from the SPDOUT pin and received in by the SPI master from the SPDOUT and SPDIN pins. This essentially doubles the received data rate. To enable Dual Mode of operation the SPI core must be configured to receive data in path on the SPDIN1 and SPDIN2 inputs via SPDIN Select. The BIOEN bit determines if the SPI core is transmitting or receiving. The setting of this bit determines the direction of the SPDOUT signal. The SPDIN Select bits are configuration bits that remain static for the duration of a dual read command. The BIOEN bit must be toggled to indicate when the SPI core is transmitting and receiving. For a description of the BIOEN bit see BIOEN on page 489.

- To transmit data in dual mode set the BIOEN bit before writing the TX_DATA register.
- To receive data in dual mode clear the BIOEN bit before writing the TX_DATA register with a dummy byte. The even bits (0,2,4,and 6) are received on the SPDOUT pin and the odd bits (1,3,5,and 7) are received on the SPDIN pin. The hardware assembles these received bits into a single byte and loads them into the RX_DATA register accordingly.

The following diagram illustrates a Dual Fast Read Command that is supported by some SPI Flash devices.

FIGURE 33-5: DUAL FAST READ FLASH COMMAND



Note:	When the SPI core is used for flash commands, like the Dual Read command, the host discards the bytes received during the command, address, and dummy byte portions of the transaction.

© 2012-2018 Microchip Technology Inc.

33.9.4 HOW BIOEN BIT CONTROLS DIRECTION OF SPDOUT BUFFER

When the SPI is configured for Half Duplex mode or Dual Mode the SPDOUT pin operates as a bi-directional signal. The BIOEN bit is used to determine the direction of the SPDOUT buffer when a byte is transmitted. Internally, the BIOEN bit is sampled to control the direction of the SPDOUT buffer when the TX_DATA value is loaded into the transmit shift register. The direction of the buffer is never changed while a byte is being transmitted.

Since the TX_DATA register may be written while a byte is being shifted out on the SPDOUT pin, the BIOEN bit does not directly control the direction of the SPDOUT buffer. An internal DIRECTION bit, which is a latched version of the BIOEN bit determines the direction of the SPDOUT buffer. The following list summarizes when the BIOEN bit is sampled.

- The DIRECTION bit is equal to the BIOEN bit when data is not being shifted out (i.e., SPI interface is idle).
- The hardware samples the BIOEN bit when it is shifting out the last bit of a byte to determine if the buffer needs to be turned around for the next byte.
- The BIOEN bit is also sampled any time the value in the TX_DATA register is loaded into the shift register to be transmitted.

APPLICATION NOTE: If a TAR (Turn-around time) is required between transmitting and receiving bytes on the SPDOUT signal, software should allow all the bytes to be transmitted before changing the buffer to an input and then load the TX_DATA register to begin receiving bytes. This allows the SPI block to operate the same as legacy SPI devices.

33.9.5 CONFIGURING THE CLOCK GENERATOR FOR AN SPI TRANSACTION

The SPI Core generates the SPI_CLK signal to the external SPI device. This clock may be configured for a range of frequencies as illustrated in Table 33-14, “SPI_CLK Frequencies,” on page 493. The clock phase and polarity are configurable as well. The following sections define how to program these features.

USER’S NOTE: The clock source configuration should not be changed during an SPI transaction.

33.9.5.1 Configuring the Frequency of the SPI Clock

The frequency of the SPI_CLK signal is determined by the clock source enabled to the clock generator and the preload value of the clock generator down counter. The clock generator toggles the SPI_CLK output every time the counter underflows, while data is being transmitted. If the preload value is set to 0 the MCLK clock source bypasses the down counter to directly create the clock generator output.

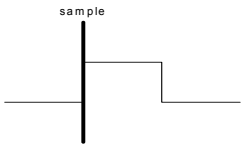
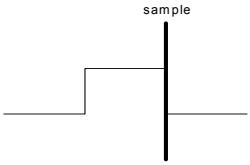
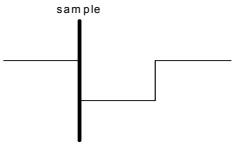
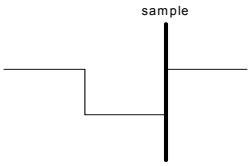
Note: When the SPI interface is in the idle state and data is not being transmitted, the SPI_CLK signal stops in the inactive state as determined by the configuration bits.

The clock source to the down counter is determined by Bit CLKSRC. Either the MCLK clock or the 2.0 MHz clock enable (2.0MHz_EN) can be used to decrement the down counter in the clock generator logic.

33.9.5.2 Configuring the SPI Mode

In practice, there are four modes of operation that define when data should be latched. These four modes are the combinations of the SPI_CLK polarity (CLKPOL) and phase (RCLKPH and TCLKPH). Phase is programmable independently for the receive and transmit phases. CLKPOL, RCLKPH and TCLKPH bits are programmable as defined in Section 33.9.5.3 and Section 33.9.5.4 below.

TABLE 33-4: SPI MODES

SPI Mode	CLKPOL	CLKPH	Definition	Diagram
0	0	0	data sampled on rising edge of clock	
1	0	1	data sampled on falling edge of clock	
2	1	0	data sampled on falling edge of clock	
3	1	1	data sampled on rising edge of clock	

33.9.5.3 Configuring the Polarity of the SPI Clock

The output of the clock generator may be inverted to create an active high or active low clock pulse. This is used to determine the inactive state of the SPI_CLK signal and is used for determining the first edge for shifting the data. The polarity is selected by Bit CLKPOL in the SPI Clock Control Register (SPICC).

33.9.5.4 Configuring the Phase of the SPI Clock

The SPI devices need to know when to sample the data, which may be either on the first edge of the clock or on the second edge of the clock. The phase of the clock is selected independently for receiving data and transmitting data. The receive phase is determine by Bit RCLKPH and the transmit phase is determine by TCLKPH in the SPI Clock Control Register (SPICC).

© 2012-2018 Microchip Technology Inc.

33.9.5.5 Limits of SPI configurations

The following limits Modes, clock frequency, & board layout (see also Note 33-1).

Master (MEC1632)				Slave (not MEC1632)	
Max SPICLK Frequency	Dual Mode of Operation	**Output Data Transitions**	**Input Data Sampled**	**Output Data Transitions**	**Input Data Sampled**
MCLK	Not supported	Pos edge of clk	Pos edge of clk	Pos edge of clk	Either edge of clk
MCLK/2 or less	Supported	All combinations are valid			

33.10 Instance Description

There are two instances of General Purpose Serial Peripheral Interface (GP-SPI) block implemented in the MEC1632.

Each instance of the General Purpose Serial Peripheral Interface (GP-SPI) has its own Logical Device Number, and Base Address as indicated in Table 33-5.

TABLE 33-5: General Purpose Serial Peripheral Interface (GP-SPI) BASE ADDRESS TABLE

General Purpose Serial Peripheral Interface (GP-SPI) **Instance**	**LDN from (Table 4-2 on page 59)**	**AHB Base Address**
EC GP-SPI	7h	F0_1C00h
Flash SPI	Fh	FF_3C00h (Note 33-2)

The Table 33-6 is a register summary for one instance of the General Purpose Serial Peripheral Interface (GP-SPI). Each EC address is indicated as an SPB Offset from its AHB base address.

Note 33-2 All of the addresses in the Detailed Register Descriptions below refer to the EC GP-SPI instance register addresses which are 32-bit aligned and must be divided by four to correctly represent the addressing for the Flash SPI instance which is accessible to the LPC host.

TABLE 33-6: General Purpose Serial Peripheral Interface (GP-SPI) REGISTER SUMMARY

	EC Interface			
Register Name	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
SPIAR - SPI Enable Register	00h	3-0	R/W	
SPICR - SPI Control Register	04h	3-0	R/W	
SPISR - SPI Status Register	08h	3-0	R	
SPITD - SPI TX_Data Register	0Ch	3-0	R/W	
SPIRD - SPI RX_Data Register	10h	3-0	R	
SPICC - SPI Clock Control Register	14h	3-0	R/W	
SPICG - SPI Clock Generator Register	18h	3-0	R/W	
SPIAR - SPI Enable Register	00h	3-0	R/W	

33.11 Detailed Register Descriptions

APPLICATION NOTE: In the SPI registers some configuration bits are assumed to be static, while others may be updated dynamically by software. The BIOEN and ENABLE bits are considered dynamic bits that can be modified by software at anytime, regardless if a transaction is active or not. These values are latched in hardware, so as to not affect the current operation being performed. All other bits are considered static and cannot be changed by Software while an SPI Transaction is in process.

33.11.1 SPIAR - SPI ENABLE REGISTER

TABLE 33-7: SPI ENABLE REGISTER

HOST ADDRESS	n/a			n/a			**HOST SIZE**	
EC OFFSET	00h			8-bit			**EC SIZE**	
POWER	VTR			00h			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R/W
BIT NAME	Reserved							Enable

ENABLE

0=Disabled. Clocks are gated to conserve power and the SPDOUT and SPI_CLK signals are set to their inactive state

1=Enabled. The device is fully operational.

33.11.2 SPICR - SPI CONTROL REGISTER

TABLE 33-8: SPI CONTROL REGISTER

HOST ADDRESS	n/a			8-bit			**HOST SIZE**	
EC OFFSET	04h			8-bit			**EC SIZE**	
POWER	VTR			02h			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R/W	R/W	R/W	R/W	R/W
BIT NAME	Reserved	CE	AUTO_READ	Soft Reset	SPDIN Select		BIOEN	LSBF

LSBF

Least Significant Bit First control.

0= The data is transferred in MSB-first order. (default)

1= The data is transferred in LSB-first order.

© 2012-2018 Microchip Technology Inc.

BIOEN

Bidirectional Output Enable control. When the SPI is configured for Half Duplex mode or Dual Mode the SPDOUT pin operates as a bi-directional signal. The BIOEN bit is used by the internal DIRECTION bit to control the direction of the SPDOUT buffers. The direction of the buffer is never changed while a byte is being transmitted.

0=The SPDOUT_Direction signal configures the SPDOUT signal as an input.

1=The SPDOUT_Direction signal configures the SPDOUT signal as an output. (default)

Note:	If the SPIMODE bit is configured for Full Duplex mode the BIOEN bit must be set to '1' to configure the SPDOUT signal as an output.

APPLICATION NOTE: Although the design supports back-to-back transmissions even when the direction of the buffer is changed, it is the software's responsibility to avoid collisions on the SPDOUT signal. The designed has been implemented to support a 0 second (max) turn-around (TAR) time. If TAR greater than zero is required, the software must wait for the transmission in one direction to complete before writing the TX_DATA register to start sending/receiving in the opposite direction.

SPDIN SELECT

The SPDIN Select which SPI input signals are enabled when the BIOEN bit is configured as an input.

00= SPDIN1 only //Select this option for Full Duplex (default)

01=SPDIN2 only //Select this option for Half Duplex

1x=SPDIN1 and SPDIN2 //Select this option for Dual Mode

SOFT RESET

Soft Reset is a self-clearing bit. Writing zero to this bit has no effect. Writing a one to this bit resets the entire SPI Interface, including all counters and registers back to their initial state (i.e., the same as a nSYS_RST).

AUTO_READ

When this bit is 1, a read of the SPI RX_DATA Register will both clear the RXBF status bit, and in addition it will clear the TXBE status bit. Clearing the TXBE status bit will cause the contents of the TX_DATA register to be copied into the 8-bit transmit shift register, which then begins shifting data out. Because shifting data out is accompanied by clocking the SPI Clock, a read of RX_DATA will cause the next byte from the SPI device to be shifted in to RX_DATA. The contents of the TX_DATA register is not significant, since when reading from RX_DATA it is just used to generate the SPI clock. When this bit is 0 (default), a read of the SPI RX_DATA Register will clear the RXBF bit, but not the TXBE bit.

CE

1= SPCE# output signal is asserted, i.e., driven to logic '0'

0= SPCE# output signal is asserted, i.e., driven to logic '1'

33.11.3 SPISR - SPI STATUS REGISTER

TABLE 33-9: SPI STATUS REGISTER

HOST ADDRESS	n/a					8-bit	**HOST SIZE**	
EC OFFSET	08h					8-bit	**EC SIZE**	
POWER	VTR					01h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved					ACTIVE	RxBF	TxBE

TXBE

Transmit Data Buffer Empty status. This bit is a read-only bit used to indicate that the Tx_Data buffer is empty. Writing the Tx_DATA Buffer clears this bit. This signal may be used to generate an interrupt to the EC, if not masked.

RXBF

Receive Data Buffer Full status. This bit is a read-only bit used to indicate when the Rx_Data buffer is full. Reading the Rx_DATA Buffer clears this bit. This signal may be used to generate an interrupt to the EC, if not masked.

ACTIVE

The ACTIVE bit indicates that a transaction controlled by the General Purpose Serial Peripheral Interface (GP-SPI) is in progress. The ACTIVE bit is asserted (‘1’) when the GP-SPI controller transfers data from the SPITD - SPI TX_Data Register into the DATA_OUT Shift Register.

ACTIVE is de-asserted (‘0’) follow a system reset (nSYS_RST) or when data from the DATA_IN Shift Register is transferred to the SPIRD - SPI RX_Data Register. Note that data is only transferred from the DATA_IN Shift Register to the SPIRD - SPI RX_Data Register when RxBF is not asserted (‘0’); i.e., the SPIRD - SPI RX_Data Register is not overwritten by incoming data when RxBF is ‘1.’

33.11.4 SPITD - SPI TX_DATA REGISTER

TABLE 33-10: SPI TX DATA REGISTER (SPITD)

HOST ADDRESS	n/a					8-bit	**HOST SIZE**	
EC OFFSET	0Ch					8-bit	**EC SIZE**	
POWER	VTR					00h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	TX_Data[7:0]							

© 2012-2018 Microchip Technology Inc.

TX_DATA[7:0]

A write to this register with the TxBE bit asserted '1' initiates an SPI transaction. If the Transmit Shift Register is empty the byte written to this register will be loaded into the shift register and the TxBE flag will be asserted. This indicates that the next byte can be written into the TX_DATA register. This byte will remain in the TX_DATA register until the SPI core has finished shifting out the previous byte. Once the shift register is empty, the hardware will load the pending byte into the shift register and once again assert the TxBE bit.

Note:	The TX_DATA register must not be written when the TxBE bit is zero. Writing this register may overwrite the transmit data before it is loaded into the shift register.

Reading the TX_DATA register will return the last value written to this register.

33.11.5 SPIRD - SPI RX_DATA REGISTER

TABLE 33-11: SPI RX DATA REGISTER (SPIRD)

HOST ADDRESS	n/a					8-bit	**HOST SIZE**	
EC OFFSET	10h					8-bit	**EC SIZE**	
POWER	VTR					00h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	RX_Data[7:0]							

RX_DATA[7:0]

This register is used to read the value returned by the external SPI device. At the end of a byte transfer the RX_DATA register contains serial input data (valid or not) from the last transaction and the RXBF bit is set to one. This status bit indicates that the RX_DATA register has been loaded with a the serial input data. The RX_DATA register should not be read before the RXBF bit is set.

Note:	The RX_DATA register must be read, clearing the RXBF status bit before writing the TX_DATA register. The data in the receive shift register is only loaded into the RX_DATA register when this bit is cleared. If a data byte is pending in the receive shift register the value will be loaded immediately into the RX_DATA register and the RXBF status flag will be asserted. Software should read the RX_DATA register twice before starting a new transaction to make sure the RX_DATA buffer and shift register are both empty.

33.11.6 SPICC - SPI CLOCK CONTROL REGISTER

TABLE 33-12: SPI CLOCK CONTROL REGISTER (SPICC)

HOST ADDRESS	n/a					8-bit	**HOST SIZE**	
EC OFFSET	14h					8-bit	**EC SIZE**	
POWER	VTR					02h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R/W	R	R/W	R/W	R/W
BIT NAME	Reserved			CLKSRC	Reserved	CLKPOL	RCLKPH	TCLKPH

APPLICATION NOTE: the default CLKPOL, RCLKPH and TCLKPH values are appropriate for transactions with typical MODE 0 SPI Flash devices.

TCLKPH

The TCLKPH bit determines the Transmit Clock Phase, the SPCLK edge on which the master will clock data out.

0= Valid data is clocked out on the SPDOUT signal prior to the first SPI_CLK edge. The slave device should sample this data on the first and following odd SPI_CLK edges (i.e., sample data on rising edge). (default)

1= Valid data is clocked out on the first SPI_CLK edge on SPDOUT signal. The slave device should sample this data on the second and following even SPI_CLK edges (i.e., sample data on falling edge).

Note: This functionality is independent of the polarity of SPCLK. See Section 46.10, "Serial Peripheral Interface (SPI) Timings," on page 623 for timing diagrams.

RCLKPH

The RCLKPH bit determines the Receive Clock Phase, the SPI_CLK edge on which the master will sample data.

0= Valid data is expected on the SPDIN signal on the first SPI_CLK edge. This data is sampled on the first and following odd SPI_CLK edges (i.e., sample data on rising edge).

1= Valid data on SPDIN signal is expected after the first SPI_CLK edge. This data is sampled on the second and following even SPI_CLK edges (i.e., sample data on falling edge). (default).

Note: This functionality is independent of the polarity of SPI_CLK. See Section 46.10, "Serial Peripheral Interface (SPI) Timings," on page 623 for timing diagrams.

CLKPOL

This bit controls the polarity of the SPI clock.

0= The SPI_CLK is low when the interface is idle and the first clock edge is a rising edge. (default)

1= The SPI_CLK signal is high when the interface is idle and the first clock edge is a falling edge

CLKSRC

This bit controls the clock source to SPI Clock Generator

0=The clock source to the SPI Clock Generator 6-bit down counter is MCLK (default)

1=The clock source to the SPI Clock Generator 6-bit down counter is 2MHz clock enable

Note: The CLKSRC bit should not be changed during a SPI transaction.

33.11.7 SPICG - SPI CLOCK GENERATOR REGISTER

TABLE 33-13: SPI CLOCK GENERATOR REGISTER (SPICG)

HOST ADDRESS	n/a			8-bit			**HOST SIZE**	
EC OFFSET	18h			8-bit			**EC SIZE**	
POWER	VTR			02h			nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Reserved		Preload[5:0]					

© 2012-2018 Microchip Technology Inc.

PRELOAD[5:0]

The SPI Clock Generator Preload value. The SPI_CLK signal is a clock output with a 50% duty cycle. The signal is generated from a 6-bit down counter that toggles the SPI_CLK pin every time it reaches zero and reloads the counter. This counter is decremented at the rate of the input clock enable, which is either MCLK or the 2MHz clock enable.

$$\text{SPCLKO_FREQ} = \left(\left(\frac{1}{2} \times \text{CLOCK_ENABLE_FREQ} \right) / \text{PRELOAD} \right)$$

The PRELOAD field contains the preload value for the counter that determines the resulting frequency. The following table outlines the ranges of frequencies possible. Notice that a preload value of zero effectively bypasses the counter and maps the MCLK clock onto the SPI_CLK signal.

TABLE 33-14: SPI_CLK FREQUENCIES

Clock Enable Frequency	Preload	SPI_CLK Frequency	Notes
Don't Care	0	MCLK	Max Frequency (Note 33-3)
MCLK	1	MCLK/2	
MCLK	2	MCLK/4	Default Frequency
MCLK	3	MCLK/6	
MCLK	63	MCLK/126	
2.0MHz_EN	0	2.0MHz_EN	
2.0MHz_EN	1	2.0MHz_EN/2	
2.0MHz_EN	2	2.0MHz_EN/4	
2.0MHz_EN	3	2.0MHz_EN/6	
2.0MHz_EN	63	2.0MHz_EN/126	Min Frequency

Note 33-3 When the Preload value is programmed to zero the MCLK Clock Source is directly mapped to the SPI_CLK signal. Since the MCLK signal is a 50% duty cycle it may be directly used as an SPI_CLK frequency.

33.12 SPI Examples

33.12.1 FULL DUPLEX MODE TRANSFER EXAMPLES

33.12.1.1 Read Only

The slave device used in this example is a MAXIM MAX1080 10 bit, 8 channel ADC:

- The SPI block is activated by setting the enable bit in SPIAR - SPI Enable Register
- The SPIMODE bit is de-asserted '0' to enable the SPI interface in Full Duplex mode.
- The CLKPOL and TCLKPH bits are de-asserted '0', and RCLKPH is asserted '1' to match the clocking requirements of the slave device.
- The LSBF bit is de-asserted '0' to indicate that the slave expects data in MSB-first order.
- Assert #CS using a GPIO pin.
- Write a valid command word (as specified by the slave device) to the SPITD - SPI TX_Data Register with TXFE asserted '1'. The SPI master automatically clears the TXFE bit indicating the byte has been put in the TX buffer. If the shift register is empty the TX_DATA byte is loaded into the shift register and the SPI master reasserts the TXFE bit. Once the data is in the shift register the SPI master begins shifting the data value onto the SPDOUT pin and drives the SPCLK pin. Data on the SPDIN pin is also sampled on each clock.
- Once the TXFE bit is asserted the SPI Master is ready to receive its next byte. Before writing the next TX_DATA value, software must clear the RXBF status bit by reading the SPIRD - SPI RX_Data Register.
- A dummy 8 bit data value (any value) is written to the TX_DATA register. The SPI master automatically clears the TXFE bit, but does not begin shifting the dummy data value onto the SPDOUT pin. This byte will remain in the TX_DATA register until the TX shift register is empty.

- After 8 SPI_CLK pulses from the first transmit bytes:
  - The first SPI cycle is complete, RXBF bit is asserted '1', and the SPINT interrupt is asserted, if enabled. (See Section 33.7, "SPI Interrupts / DMA Requests"). The data now contained in SPIRD - SPI RX_Data Register is invalid since the last cycle was initiated solely to transmit command data to the slave. This particular slave device drives '0' on the SPDIN pin to the master while it is accepting command data. This SPIRD data is ignored.
  - Once the first SPI cycle is completed, the SPI master takes the pending data in the TX_DATA register and loads it into the TX shift register. Loading the shift register automatically asserts the TXFE bit, begins shifting the dummy data value onto the SPDOUT pin, and drives the SPCLK pin. Data on the SPDIN pin is also sampled on each clock.
- Once the TXFE bit is asserted the SPI Master is ready to receive its next byte. Before writing the next TX_DATA value, software must clear the RXBF status bit by reading the SPIRD - SPI RX_Data Register.
- The final SPI cycle is initiated when another dummy 8 bit data value (any value) is written to the TX_DATA register. Note that this value may be another dummy value or it can be a new 8 bit command to be sent to the ADC. The new command will be transmitted while the final data from the last command is received simultaneously. This overlap allows ADC data to be read every 16 SPCLK cycles after the initial 24 clock cycle.The SPI master automatically clears the TXFE bit, but does not begin shifting the dummy data value onto the SPDOUT pin. This byte will remain in the TX_DATA register until the TX shift register is empty.
- After 8 SPI_CLK pulses, the second SPI cycle is complete:
  - The first SPI cycle is complete, RXBF bit is asserted '1', and the SPINT interrupt is asserted, if enabled. (See Section 33.7, "SPI Interrupts / DMA Requests"). The data now contained in SPIRD - SPI RX_Data Register is the first half of a valid 16 bit ADC value. SPIRD is read and stored.
  - Once the second SPI cycle is completed, the SPI master takes the pending data in the TX_DATA register and loads it into the TX shift register. Loading the shift register automatically asserts the TXFE bit, begins shifting the data value onto the SPDOUT pin, and drives the SPCLK pin. Data on the SPDIN pin is also sampled on each clock.
- After 8 SPI_CLK pulses, the final SPI cycle is complete, TXBF is asserted '1', and the SPINT interrupt is asserted (if enabled). The data now contained in SPIRD - SPI RX_Data Register is the second half of a valid 16 bit ADC value. SPIRD is read and stored.
- If a command was overlapped with the received data in the final cycle, #CS should remain asserted and the SPI master will initiate another SPI cycle. If no new command was sent, #CS is released and the SPI is idle.

33.12.1.2 Read/Write

The slave device used in this example is a Fairchild NS25C640 FM25C640 64K Bit Serial EEPROM. The following subsections describe the read and write sequences.

33.12.1.2.1 Read

- The SPI block is activated by setting the enable bit in SPIAR - SPI Enable Register
- The SPIMODE bit is de-asserted '0' to enable the SPI interface in Full Duplex mode.
- The CLKPOL, TCLKPH and RCLKPH bits are de-asserted '0' to match the clocking requirements of the slave device.
- The LSBF bit is de-asserted '0' to indicate that the slave expects data in MSB-first order.
- Assert CS# low using a GPIO pin.
- Write a valid command word (as specified by the slave device) to the SPITD - SPI TX_Data Register with TXFE asserted '1'. The SPI master automatically clears the TXFE bit indicating the byte has been put in the TX buffer. If the shift register is empty the TX_DATA byte is loaded into the shift register and the SPI master reasserts the TXFE bit. Once the data is in the shift register the SPI master begins shifting the data value onto the SPDOUT pin and drives the SPI_CLK pin. Data on the SPDIN pin is also sampled on each clock.
- Once the TXFE bit is asserted the SPI Master is ready to receive its next byte. Before writing the next TX_DATA value, software must clear the RXBF status bit by reading the SPIRD - SPI RX_Data Register.
- Next, EEPROM address A15-A8 is written to the TX_DATA register. The SPI master automatically clears the TXFE bit, but does not begin shifting the dummy data value onto the SPDOUT pin. This byte will remain in the TX_DATA register until the TX shift register is empty.
- After 8 SPI_CLK pulses from the first transmit byte (Command Byte transmitted):
  - The first SPI cycle is complete, RXBF bit is asserted '1', and the SPINT interrupt is asserted, if enabled. (See Section 33.7, "SPI Interrupts / DMA Requests"). The data now contained in SPIRD - SPI RX_Data Register is

© 2012-2018 Microchip Technology Inc.

invalid since the last cycle was initiated solely to transmit command data to the slave. This particular slave device tri-states the SPDIN pin to the master while it is accepting command data. This SPIRD data is ignored.

USER'S NOTE:	External pull-up or pull-down is required on the SPDIN pin if it is tri-stated by the slave device. Once the first SPI cycle is completed, the SPI master takes the pending data in the TX_DATA register (EEPROM address A15-A8) and loads it into the TX shift register. Loading the shift register automatically asserts the TXFE bit, begins shifting the dummy data value onto the SPDOUT pin, and drives the SPI_CLK pin. Data on the SPDIN pin is also sampled on each clock. Note: The particular slave device ignores address A15-A13.

- Once the TXFE bit is asserted the SPI Master is ready to receive its next byte. Before writing the next TX_DATA value, software must clear the RXBF status bit by reading the SPIRD - SPI RX_Data Register.
- Next, EEPROM address A7-A0 is written to the TX_DATA register. The SPI master automatically clears the TXFE bit, but does not begin shifting this data value onto the SPDOUT pin. This byte will remain in the TX_DATA register until the TX shift register is empty.
- After 8 SPI_CLK pulses from the second transmit byte (Address Byte (MSB) transmitted):
  - EEPROM address A15-A8 has been transmitted to the slave completing the second SPI cycle. Once again, the RXBF bit is asserted '1' and the SPINT interrupt is asserted, if enabled. (See Section 33.7, "SPI Interrupts / DMA Requests"). The data now contained in SPIRD - SPI RX_Data Register is invalid since the last cycle was initiated solely to transmit address data to the slave.
  - Once the second SPI cycle is completed, the SPI master takes the pending data in the TX_DATA register (EEPROM address A7-A0) and loads it into the TX shift register. Loading the shift register automatically asserts the TXFE bit, begins shifting the dummy data value onto the SPDOUT pin, and drives the SPCLK pin. Data on the SPDIN pin is also sampled on each clock.
- Once the TXFE bit is asserted the SPI Master is ready to receive its next byte. Before writing the next TX_DATA value, software must clear the RXBF status bit by reading the SPIRD - SPI RX_Data Register.
- Next, a dummy byte is written to the TX_DATA register. The SPI master automatically clears the TXFE bit, but does not begin shifting this data value onto the SPDOUT pin. This byte will remain in the TX_DATA register until the TX shift register is empty.
- After 8 SPI_CLK pulses, the third SPI cycle is complete (Address Byte (LSB) transmitted):
  - EEPROM address A7-A0 has been transmitted to the slave completing the third SPI cycle. Once again, the RXBF bit is asserted '1' and the SPINT interrupt is asserted, if enabled. (See Section 33.7, "SPI Interrupts / DMA Requests"). The data now contained in SPIRD - SPI RX_Data Register is invalid since the last cycle was initiated solely to transmit address data to the slave.
  - Once the third SPI cycle is completed, the SPI master takes the pending data in the TX_DATA register (dummy byte) and loads it into the TX shift register. Loading the shift register automatically asserts the TXFE bit, begins shifting the dummy data value onto the SPDOUT pin, and drives the SPCLK pin. Data on the SPDIN pin is also sampled on each clock.
- Once the TXFE bit is asserted the SPI Master is ready to receive its next byte. Before writing the next TX_DATA value, software must clear the RXBF status bit by reading the SPIRD - SPI RX_Data Register.
- If only one receive byte is required, the host would not write any more value to the TX_DATA register until this transaction completes. If more than one byte of data is to be received, another dummy byte would be written to the TX_DATA register (one dummy byte per receive byte is required). The SPI master automatically clears the TXFE bit when the TX_DATA register is written, but does not begin shifting this data value onto the SPDOUT pin. This byte will remain in the TX_DATA register until the TX shift register is empty.
- After 8 SPI_CLK pulses, the fourth SPI cycle is complete (First Data Byte received):
  - The dummy byte has been transmitted to the slave completing the fourth SPI cycle. Once again, the RXBF bit is asserted '1' and the SPINT interrupt is asserted, if enabled. (See Section 33.7, "SPI Interrupts / DMA Requests"). Unlike the command and address phases, the data now contained in SPIRD - SPI RX_Data Register is the 8-bit EEPROM data since the last cycle was initiated to receive data from the slave.
  - Once the fourth SPI cycle is completed, the SPI master takes the pending data in the TX_DATA register (if any) and loads it into the TX shift register. This process will be repeated until all the desired data is received.
- The host software will read and store the EEPROM data value in SPIRD - SPI RX_Data Register.
- If no more data needs to be received by the master, CS# is released and the SPI is idle. Otherwise, master continues reading the data by writing a dummy value to the TX_DATA register after every 8 SPI_CLK cycles.

© 2012-2018 Microchip Technology Inc.

33.12.1.2.2 Write

- The SPI block is activated by setting the enable bit in SPIAR - SPI Enable Register
- The SPIMODE bit is de-asserted '0' to enable the SPI interface in Full Duplex mode.
- The CLKPOL, TCLKPH and RCLKPH bits are de-asserted '0' to match the clocking requirements of the slave device.
- The LSBF bit is de-asserted '0' to indicate that the slave expects data in MSB-first order.
- Assert WR# high using a GPIO pin.
- Assert CS# low using a GPIO pin.
- Write a valid command word (as specified by the slave device) to the SPITD - SPI TX_Data Register with TXFE asserted '1'. The SPI master automatically clears the TXFE bit indicating the byte has been put in the TX buffer. If the shift register is empty the TX_DATA byte is loaded into the shift register and the SPI master reasserts the TXFE bit. Once the data is in the shift register the SPI master begins shifting the data value onto the SPDOUT pin and drives the SPI_CLK pin. Data on the SPDIN pin is also sampled on each clock.
- Once the TXFE bit is asserted the SPI Master is ready to receive its next byte. Before writing the next TX_DATA value, software must clear the RXBF status bit by reading the SPIRD - SPI RX_Data Register.
- Next, EEPROM address A15-A8 is written to the TX_DATA register. The SPI master automatically clears the TXFE bit, but does not begin shifting the dummy data value onto the SPDOUT pin. This byte will remain in the TX_DATA register until the TX shift register is empty.
- After 8 SPI_CLK pulses from the first transmit byte (Command Byte transmitted):
  - The first SPI cycle is complete, RXBF bit is asserted '1', and the SPINT interrupt is asserted, if enabled. (See Section 33.7, "SPI Interrupts / DMA Requests"). The data now contained in SPIRD - SPI RX_Data Register is invalid since the last cycle was initiated solely to transmit command data to the slave. This particular slave device tri-states the SPDIN pin to the master while it is accepting command data. This SPIRD data is ignored.

USER'S NOTE:	External pull-up or pull-down is required on the SPDIN pin if it is tri-stated by the slave device. Once the first SPI cycle is completed, the SPI master takes the pending data in the TX_DATA register (EEPROM address A15-A8) and loads it into the TX shift register. Loading the shift register automatically asserts the TXFE bit, begins shifting the dummy data value onto the SPDOUT pin, and drives the SPI_CLK pin. Data on the SPDIN pin is also sampled on each clock. Note: The particular slave device ignores address A15-A13.

- Once the TXFE bit is asserted the SPI Master is ready to receive its next byte. Before writing the next TX_DATA value, software must clear the RXBF status bit by reading the SPIRD - SPI RX_Data Register.
- Next, EEPROM address A7-A0 is written to the TX_DATA register. The SPI master automatically clears the TXFE bit, but does not begin shifting this data value onto the SPDOUT pin. This byte will remain in the TX_DATA register until the TX shift register is empty.
- After 8 SPI_CLK pulses from the second transmit byte (Address Byte (MSB) transmitted):
  - EEPROM address A15-A8 has been transmitted to the slave completing the second SPI cycle. Once again, the RXBF bit is asserted '1' and the SPINT interrupt is asserted, if enabled. (See Section 33.7, "SPI Interrupts / DMA Requests"). The data now contained in SPIRD - SPI RX_Data Register is invalid since the last cycle was initiated solely to transmit address data to the slave.
  - Once the second SPI cycle is completed, the SPI master takes the pending data in the TX_DATA register (EEPROM address A7-A0) and loads it into the TX shift register. Loading the shift register automatically asserts the TXFE bit, begins shifting the dummy data value onto the SPDOUT pin, and drives the SPCLK pin. Data on the SPDIN pin is also sampled on each clock.
- Once the TXFE bit is asserted the SPI Master is ready to receive its next byte. Before writing the next TX_DATA value, software must clear the RXBF status bit by reading the SPIRD - SPI RX_Data Register.
- Next, a data byte (D7:D0) is written to the TX_DATA register. The SPI master automatically clears the TXFE bit, but does not begin shifting this data value onto the SPDOUT pin. This byte will remain in the TX_DATA register until the TX shift register is empty.
- After 8 SPI_CLK pulses, the third SPI cycle is complete (Address Byte (LSB) transmitted):
  - EEPROM address A7-A0 has been transmitted to the slave completing the third SPI cycle. Once again, the RXBF bit is asserted '1' and the SPINT interrupt is asserted, if enabled. (See Section 33.7, "SPI Interrupts / DMA Requests"). The data now contained in SPIRD - SPI RX_Data Register is invalid since the last cycle was initiated solely to transmit address data to the slave.

© 2012-2018 Microchip Technology Inc.

- Once the third SPI cycle is completed, the SPI master takes the pending data in the TX_DATA register (data byte D7:D0) and loads it into the TX shift register. Loading the shift register automatically asserts the TXFE bit, begins shifting the dummy data value onto the SPDOUT pin, and drives the SPCLK pin. Data on the SPDIN pin is also sampled on each clock.

- Once the TXFE bit is asserted the SPI Master is ready to receive its next byte. Before writing the next TX_DATA value, software must clear the RXBF status bit by reading the SPIRD - SPI RX_Data Register.
- If only one data byte is to be written, the host would not write any more values to the TX_DATA register until this transaction completes. If more than one byte of data is to be written, another data byte would be written to the TX_DATA register. The SPI master automatically clears the TXFE bit when the TX_DATA register is written, but does not begin shifting this data value onto the SPDOUT pin. This byte will remain in the TX_DATA register until the TX shift register is empty.
- After 8 SPI_CLK pulses, the fourth SPI cycle is complete (First Data Byte transmitted):
  - The data byte has been transmitted to the slave completing the fourth SPI cycle. Once again, the RXBF bit is asserted '1' and the SPINT interrupt is asserted, if enabled. (See Section 33.7, "SPI Interrupts / DMA Requests"). Like the command and address phases, the data now contained in SPIRD - SPI RX_Data Register is invalid since the last cycle was initiated to transmit data to the slave.
  - Once the fourth SPI cycle is completed, the SPI master takes the pending data in the TX_DATA register (if any) and loads it into the TX shift register. This process will be repeated until all the desired data is transmitted.
- If no more data needs to be transmitted by the master, CS# and WR# are released and the SPI is idle.

33.12.2 HALF DUPLEX (BIDIRECTIONAL MODE) TRANSFER EXAMPLE

The slave device used in this example is a National LM74 12 bit (plus sign) temperature sensor.

- The SPI block is activated by setting the enable bit in SPIAR - SPI Enable Register
- The SPIMODE bit is asserted '1' to enable the SPI interface in Half Duplex mode.
- The CLKPOL, TCLKPH and RCLKPH bits are de-asserted '0' to match the clocking requirements of the slave device.
- The LSBF bit is de-asserted '0' to indicate that the slave expects data in MSB-first order.
- BIOEN is asserted '0' to indicate that the first data in the transaction is to be received from the slave.
- Assert #CS using a GPIO pin.

//Receive 16-bit Temperature Reading

- Write a dummy command byte (as specified by the slave device) to the SPITD - SPI TX_Data Register with TXFE asserted '1'. The SPI master automatically clears the TXFE bit indicating the byte has been put in the TX buffer. If the shift register is empty the TX_DATA byte is loaded into the shift register and the SPI master reasserts the TXFE bit. Once the data is in the shift register the SPI master begins shifting the data value onto the SPDOUT pin and drives the SPI_CLK pin. This data is lost because the output buffer is disabled. Data on the SPDIN pin is sampled on each clock.
- Once the TXFE bit is asserted the SPI Master is ready to receive its next byte. Before writing the next TX_DATA value, software must clear the RXBF status bit by reading the SPIRD - SPI RX_Data Register.
- Next, another dummy byte is written to the TX_DATA register. The SPI master automatically clears the TXFE bit, but does not begin shifting the dummy data value onto the SPDOUT pin. This byte will remain in the TX_DATA register until the TX shift register is empty.
- After 8 SPI_CLK pulses from the first receive byte
  - The first SPI cycle is complete, RXBF bit is asserted '1', and the SPINT interrupt is asserted, if enabled. (See Section 33.7, "SPI Interrupts / DMA Requests"). The data now contained in SPIRD - SPI RX_Data Register is the first half of the 16 bit word containing the temperature data.
  - Once the first SPI cycle is completed, the SPI master takes the pending data in the TX_DATA register (dummy byte 2) and loads it into the TX shift register. Loading the shift register automatically asserts the TXFE bit, begins shifting the dummy data value onto the SPDOUT pin, and drives the SPI_CLK pin. Data on the SPDIN pin is also sampled on each clock.
- Once the TXFE bit is asserted the SPI Master is ready to receive its next byte. Before writing the next TX_DATA value, software must clear the RXBF status bit by reading the SPIRD - SPI RX_Data Register.

//Transmit Next Reading Command

- BIOEN is asserted '1' to indicate that data will now be driven by the master.
- Next, a command byte is written to the TX_DATA register. This value is the first half of a 16 bit command to be sent to temperature sensor peripheral. The SPI master automatically clears the TXFE bit, but does not begin shifting the command data value onto the SPDOUT pin. This byte will remain in the TX_DATA register until the TX shift register is empty. This data will be transmitted because the output buffer is enabled. Data on the SPDIN pin is sampled on each clock.
- After 8 SPI_CLK pulses from the second receive byte:
  - The second SPI cycle is complete, RXBF bit is asserted '1', and the SPINT interrupt is asserted, if enabled. (See Section 33.7, "SPI Interrupts / DMA Requests"). The data now contained in SPIRD - SPI RX_Data Register is the second half of the 16 bit word containing the temperature data.
  - Once the first SPI cycle is completed, the SPI master takes the pending data in the TX_DATA register (command byte 1) and loads it into the TX shift register. Loading the shift register automatically asserts the TXFE bit, begins shifting the dummy data value onto the SPDOUT pin, and drives the SPI_CLK pin. Data on the SPDIN pin is also sampled on each clock.
- Once the TXFE bit is asserted the SPI Master is ready to receive its next byte. Before writing the next TX_DATA value, software must clear the RXBF status bit by reading the SPIRD - SPI RX_Data Register.
- Next, the second command byte is written to the TX_DATA register. The SPI master automatically clears the TXFE bit, but does not begin shifting the command data value onto the SPDOUT pin. This byte will remain in the TX_DATA register until the TX shift register is empty.
- After 8 SPI_CLK pulses from the first transmit byte:
  - The third SPI cycle is complete, RXBF bit is asserted '1', and the SPINT interrupt is asserted, if enabled. (See Section 33.7, "SPI Interrupts / DMA Requests"). The data now contained in SPIRD - SPI RX_Data Register is invalid, since this command was used to transmit the first command byte to the SPI slave.
  - Once the first SPI cycle is completed, the SPI master takes the pending data in the TX_DATA register (command byte 2) and loads it into the TX shift register. Loading the shift register automatically asserts the TXFE bit, begins shifting the dummy data value onto the SPDOUT pin, and drives the SPI_CLK pin. Data on the SPDIN pin is also sampled on each clock.
- Once the TXFE bit is asserted the SPI Master is ready to transmit or receive its next byte. Before writing the next TX_DATA value, software must clear the RXBF status bit by reading the SPIRD - SPI RX_Data Register.
- Since no more data needs to be transmitted, the host software will wait for the RXBF status bit to be asserted indicating the second command byte was transmitted successfully.
- #CS is de-asserted.

© 2012-2018 Microchip Technology Inc.

34.0 VBAT-POWERED CONTROL INTERFACE

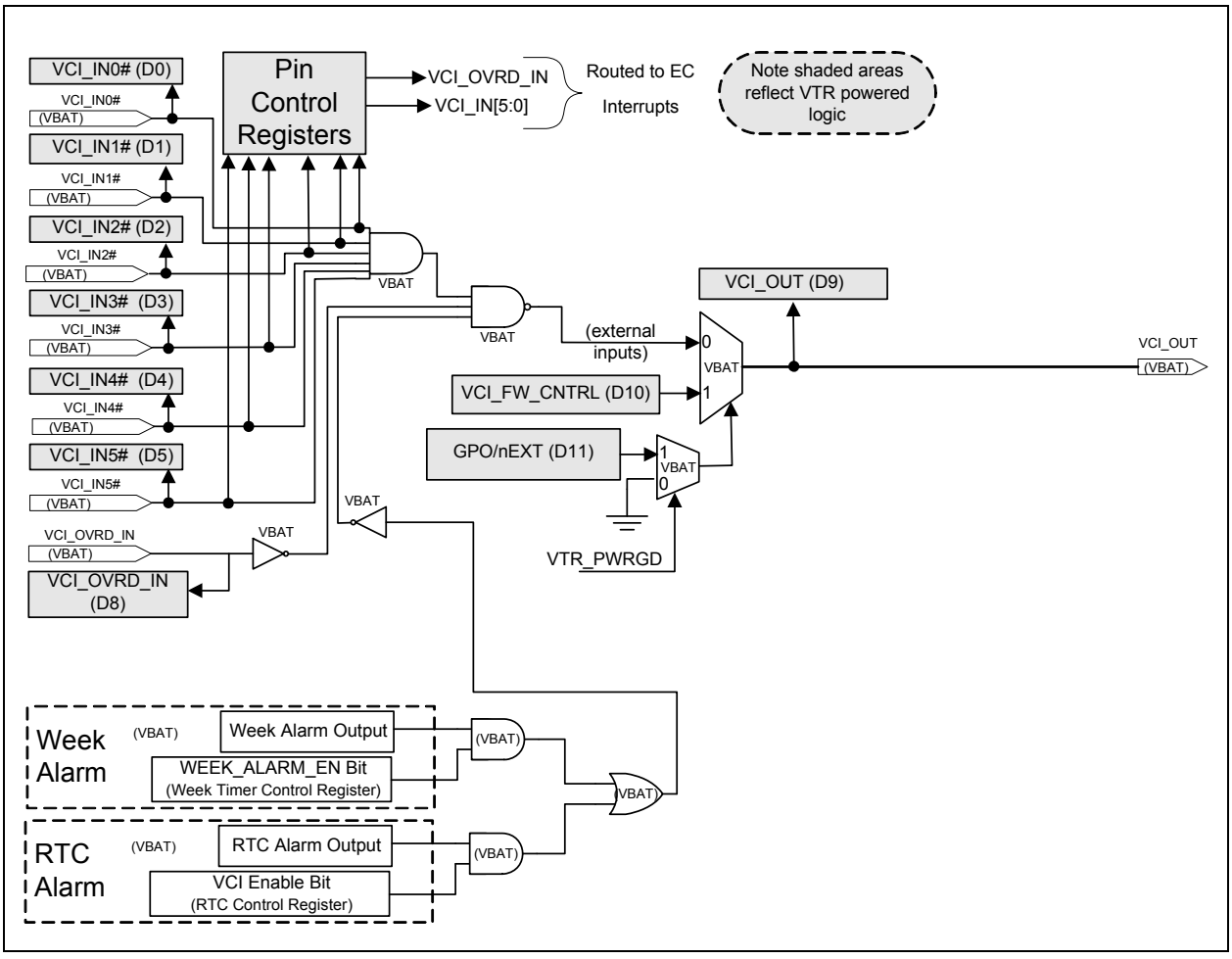
34.1 General Description

The VBAT-Powered Control Interface has VBAT powered combinational logic and input and output signal pins. The VBAT-Powered Control Interface has one VTR powered register (see Section 34.11.1, "VCI Register," on page 505). The VBAT-Powered Control Interface block interfaces with the VBAT-powered Week Alarm Interface on page 370 and the VTR-powered Week Timer Control Register on page 372.

34.2 Features

34.3 Block Diagram

FIGURE 34-1: VBAT-POWERED CONTROL INTERFACE BLOCK DIAGRAM



Note: Shaded bits are located in this block in the VCI Register, other logic is in the Week Alarm Interface.

34.4 Block Diagram Signal List

TABLE 34-1: VBAT-POWERED CONTROL INTERFACE SIGNAL LIST

Signal Name	Direction	Description
VCI_OUT	Output	VBAT-Powered Control Interface Pin Status Bit
VCI_IN[5:0]#	Input	Active low control inputs (see Section 34.10, "Registers," on page 504.)
VCI_OVRD_IN	Input	Mux control in the VBAT control logic (Note 34-3). See also the Week Alarm Interface chapter.
Week Alarm Output	Input	Asserted signal enables Week Alarm terminal count to assert VCI_OUT
RTC Alarm Output	Input	Asserted signal enables the Real Time Clock Alarm to assert VCI_OUT
Interrupts	Output	See Section 34.6, "Interrupts," on page 500
EC Interface	I/O Bus	Bus used by microprocessor to access the registers in this block.

This block is powered by the VBAT power supply.

34.5 Power, Clocks and Reset

34.5.1 POWER DOMAIN

This block is in the VTR power domain for EC interaction and uses the VBAT power domain for memory retention.

34.5.2 CLOCKS

The VBAT-Powered Control Interface has one clock input, which is used to interface to the embedded controller accessible register.

34.5.3 POWER ON RESET

VBAT-Powered Control Interface register bits that have default values are reset by hardware on VBAT_POR.

34.6 Interrupts

The VBAT-Powered Control Interface generates interrupts and wakeup events for the following pin inputs:VCI_IN[5:0]# & VCI_OVRD_IN. The Interrupts are routed to the VCI_OVRD_IN, VCI_IN0, VCI_IN1, VCI_IN2, VCI_IN3, VCI_IN4, and VCI_IN5 bits in GIRQ23 Source Register. The VBAT-Powered Control Interface Interrupt and wake events can only be asserted when VBAT and VTR are both applied; edge detection is disabled when VTR is unpowered. The edge detection for the interrupt and wake events are controlled for each pin by GPIO Pin Control Registers (see Section 24.9.1, "Pin Control Register," on page 396).

34.7 General Description

The MEC1632 VBAT-Powered Control Interface (VCI) is illustrated in Figure 34-1. This block contains the VCI Register, which monitors the status of the VCI_IN[5:0]#, VCI_OVRD_IN and VCI_OUT pins and also provides firmware control of VCI_OUT when VTR is present.

The state of the VCI_OUT pin can be determined using the following table. Signals in the table are described in Section TABLE 34-1:, "VBAT-Powered Control Interface Signal List," on page 500 and in Section 34.10, "Registers," on page 504.

© 2012-2018 Microchip Technology Inc.

TABLE 34-2: VCI OUTPUT TRUTH TABLE

Inputs														Output	Description
VCI_OVRD_IN Pin	VCI_IN0# Pin	VCI_IN1# Pin	VCI_IN2# Pin	VCI_IN3# Pin	VCI_IN4# Pin	VCI_IN5# Pin	Week Alarm Output	WEEK_ALARM_EN	VCI_FW_CNTRL Bit	RTC Alarm Output	VCI Enabled Bit (RTC)	GPO/nEXT Bit	VTRGD	VCI_OUT Pin	
X	0	X	X	X	X	X	X	X	X	X	X	X	0	1	VTR = OFF External inputs can drive VCI_OUT
X	X	0	X	X	X	X	X	X	X	X	X	X	0	1	
X	X	X	0	X	X	X	X	X	X	X	X	X	0	1	
X	X	X	X	0	X	X	X	X	X	X	X	X	0	1	
X	X	X	X	X	0	X	X	X	X	X	X	X	0	1	
X	X	X	X	X	X	0	X	X	X	X	X	X	0	1	
1	X	X	X	X	X	X	X	X	X	X	X	X	0	1	
X	X	X	X	X	X	X	1	1	X	1	1	X	0	1	
0	1	1	1	1	1	1	0	0	X	0	0	X	0	0	
X	0	X	X	X	X	X	X	X	X	X	X	0	1	1	VTR = ON External inputs can drive VCI_OUT (GPO/nEXT = ‘0’)
X	X	0	X	X	X	X	X	X	X	X	X	0	1	1	
X	X	X	0	X	X	X	X	X	X	X	X	0	1	1	
X	X	X	X	0	X	X	X	X	X	X	X	0	1	1	
X	X	X	X	X	0	X	1	1	X	1	1	0	1	1	
1	X	X	X	X	X	X	X	X	X	X	X	0	1	1	
0	1	1	1	1	1	1	0	0	X	0	0	0	1	0	
X	X	X	X	X	X	X	X	X	0	X	X	1	1	0	VTR = ON EC drives VCI_OUT (GPO/nEXT = ‘1’)
X	X	X	X	X	X	X	X	X	1	X	X	1	1	1	

Note: When VTRGD is not asserted ‘0,’ VTR is unpowered and both the EC and the VCI Register are unavailable. When VTRGD is asserted ‘1,’ the VTR well is powered, the VCI Register is powered, accessible to the EC, and contributes to the resultant logic driving the VCI_OUT pin.

APPLICATION NOTE: The VBAT-Powered Control Interface can be used in a system as follows:

1. The initial condition is the VBAT battery is installed causing a VBAT_POR, no AC power applied and no power button event. Therefore no power is applied to VTR power well. The VCI_OUT pin is de-asserted, the EC is not running, the Week Alarm has not been initialized or enabled.
2. Either applying AC charger power and asserting the VCI_OVRD_IN input or depressing a power button asserting a one or more of the VCI_IN[5:0]# input pins causes the VCI_OUT pin to be asserted (maybe a pulse for seconds). This powers up the VTR power well. This starts the EC and allows access to the VCI Register.
3. If the VCI Register is not written then the VBAT-Powered Control Interface glue logic behaves the same way as when VTR is empowered; the VCI_OUT pin is asserted when any one of the an external input (FIGURE 34-1: on page 499) are asserted: the VCI_OVRD_IN, or VCI_IN[5:0]#, or the Week Alarm Power-up signal are asserted.
4. The EC can read the status of the VCI_OVRD_IN and VCI_IN[5:0]# in the VCI Register (similar to a GP input.) The EC can enable interrupts from these pins.
5. The EC can take programmable control of the VCI_OUT pin output by setting both the VCI_FW_CNTRL and the GPO/nEXT bits in the VCI Register.

© 2012-2018 Microchip Technology Inc.

Note: BIOS should set VCI_FW_CNTRL bit to 1 prior to setting the GPO/nEXT bit to 1 to provide a glitch free VCI_OUT pin output.

6. Clearing the VCI_FW_CNTRL bit and setting the GPO/nEXT bit in the VCI Register causes the VCI_OUT pin to be de-asserted. The VCI_OUT pin remains de-asserted until VTR is empowered. When VTR=0, the VCI_OVRD_IN and VCI_IN[5:0]# inputs pins control the VCI_OUT pin.

Note:

- The VCI_IN[5:0]# pins have no direct effect on the VCI_OUT pin when VCI_OVRD_IN is asserted. However, when VTR is on and EC is running, the VCI_IN[5:0]# pins can cause interrupts and the EC can de-assert the VCI_OUT pin by programming the VCI_FW_CNTRL and the GPO/nEXT bits in the VCI Register. The VCI_OVRD_IN and the VCI_IN[5:0]# pins can also cause EC wake events.
- When VTRGD is not asserted '0,' VTR is unpowered and both the EC and the VCI Register are unavailable. When VTRGD is asserted '1,' the VTR well is powered, the VCI Register is powered, accessible to the EC, and contributes to the resultant logic driving the VCI_OUT pin

34.7.1 VCI PINS

- There is a latching option on all VCI_IN[5:0]# inputs.The latches are set to '1' on VBAT POR and when the corresponding bit in the Latch Enable Register is '1.' They are reset to '0' when the corresponding VCI_IN# pin is '0.'
- The VCI_IN[3:0]# pins are unlatched on power-on; the VCI_IN[5:4]# pins come up by default with the latches enabled (see Section 34.11.2, "Latch Enable Register," on page 506).
- Each of the VCI_IN[5:0]# pinsThe VCI_IN pin has an optional 140 ns analog glitch filter. See also the Filters Bypass bit 12 in the VCI Register.
- When the VTR power rail is fully powered, the MEC1632 GPIO functions in the VBAT-Powered Control Interface are full function GPIOs. When the VTR rail is not powered, The GPIO output drivers are tri-stated.Note that power for all VBAT-Powered Control Interface output drivers is sourced from the VBAT power pin.

34.7.2 INPUT POLARITY

The VCI_IN# pins have an optional polarity inversion. The inversion takes place after any input filtering and before the VCI_IN signals are latched in the VCI_IN# status bits in the VCI Register. Edge detection occurs before the polarity inversion. The inversion is controlled by a battery-backed configuration bits in the VCI Polarity Register.

34.7.3 EDGE EVENT STATUS

Each VCI_IN# input pin is associated with two register bits used to record edge transitions on the pins. The edge detection takes place after any input filtering, before polarity control and occurs even if the VCI_IN# input is not enabled as part of the VCI_OUT logic. One bit is set whenever there is a high-to-low transition on the VCI_IN# pin. The other bit is set whenever there is a low-to-high transition on the VCI_IN# pin. Edge detection occurs after any input filtering.

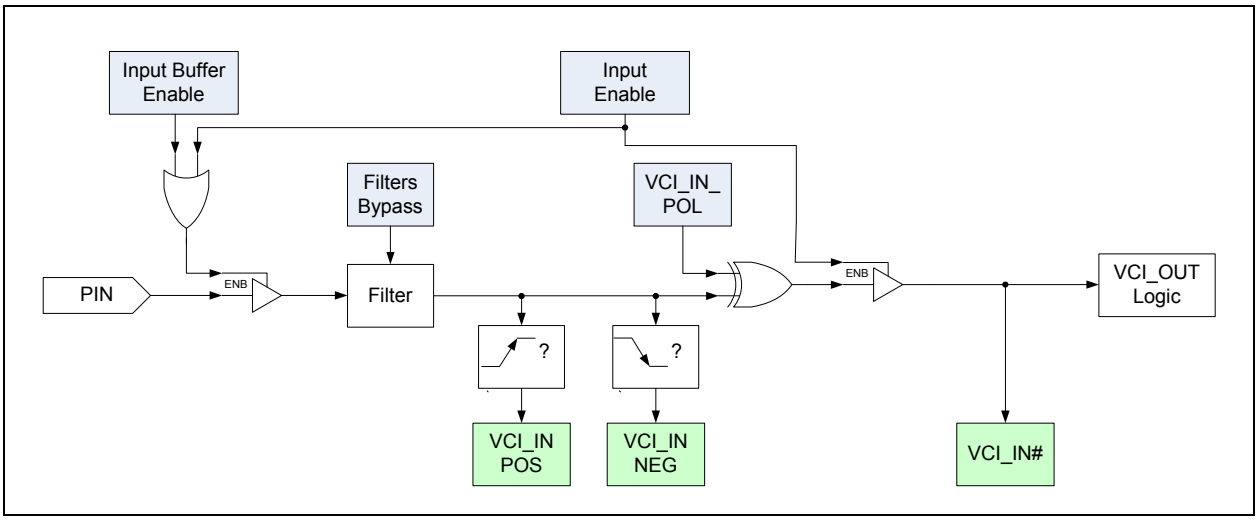
The two edge detection registers (VCI Posedge Detect Register and VCI Negedge Detect Register) are active even if the corresponding control bit in the VCI Input Enable Register is 0 (VCI function not enabled) or the corresponding bit in the Latch Enable Register is 0 (no latching).

In order to minimize power drain on the VBAT circuit, the edge detection logic operates only when the input buffer for a VCI_IN# pin is enabled. The input buffer is enabled either when the VCI_IN# pin is configured to determine the VCI_OUT pin, as controlled by the VCI_IN[5:0]# field of the VCI Register, or when the input buffer is explicitly enabled in the VCI Input Enable Register. When the pins are not enabled transitions on the pins are ignored.

© 2012-2018 Microchip Technology Inc.

The relationship among the edge event latches, the polarity control and enables is shown in the following figure:

FIGURE 34-2: VCI INPUT PIN CONFIGURATION



34.8 Input Timing

For pins where latching is not enabled, when VTRGD is not asserted, or VTRGD is asserted and the GPO/nEXT bit is '0,' transition times on the VCI_IN[5:0]# and VCI_OVRD_IN input pins must not be greater than t_R, t_F as shown in Figure 34-3 and Table 34-3.

FIGURE 34-3: VBAT-Powered Control Interface Input Timing

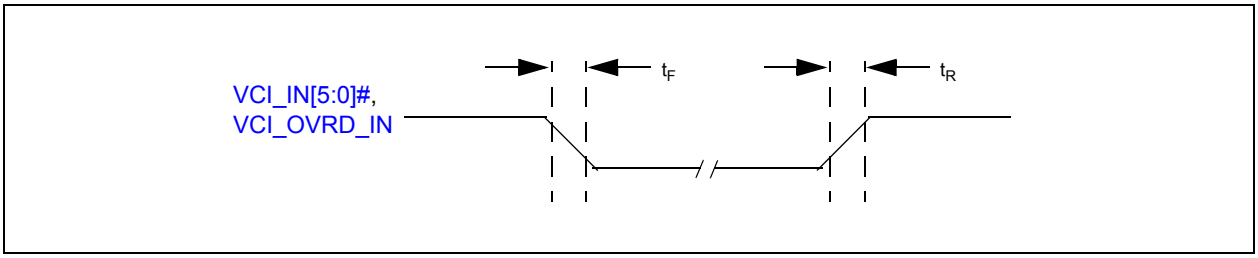


TABLE 34-3: INPUT TIMING PARAMETERS

Parameter	Symbol	MIN	TYP	MAX	Units	Conditions
VCI_IN[5:0]# and VCI_OVRD_IN Rise and Fall Time	t_R, t_F	–	–	1	µs	VTRGD = '0,' or VTRGD = '1' and GPO/nEXT = '0.'

34.9 Input Filtering

Input Filtering suppresses spikes on the VBAT-Powered Control Interface inputs as defined in Table 34-4. Input Filtering is controlled by the Filters Bypass bit in the VCI Register.

TABLE 34-4: VBAT-Powered Control Interface Input Filtering

Name	Description	MIN	TYP	MAX	Units
t_{SP}	Pulse width of spikes suppressed by Input Filtering.	50	–	140	ns

34.10 Registers

The VBAT-Powered Control Interface has its own Logical Device Number, and Base Address as indicated in Table 34-5.

Table 34-6 is a register summary for the VBAT-Powered Control Interface block.

Each instance of the VBAT-Powered Control Interface Base Address as indicated in Table 34-5.

TABLE 34-5: VBAT-Powered Control Interface BASE ADDRESS TABLE

VBAT-Powered Control Interface Instance	LDN from (Table 4-3 on page 60)	AHB Base Address
VBAT-Powered Control Interface	34h	F0_D000h

Table 34-6 is a register summary for the VBAT-Powered Control Interface block. Each EC address is indicated as an SPB Offset from its AHB base address.

TABLE 34-6: VBAT-POWERED CONTROL INTERFACE REGISTER SUMMARY

Register Name	EC Offset
VCI Register	00h
Latch Enable Register	04h
Latch Resets Register	08h
VCI Input Enable Register	0Ch
VCI Polarity Register	14h
VCI Posedge Detect Register	18h
VCI Negedge Detect Register	1Ch
VCI Buffer Enable Register	20h

34.11 Detailed Description of Accessible Registers

TABLE 34-7: REGISTER BIT ACCESS TYPES

Register Bit Type	Description
R	**Read**: A register or bit with this attribute can be read.
W	**Write**: A register or bit with this attribute can be written.
RO	**Read Only**: A register or bit with this attribute is read only, writes have no effect.
RS	**Read to Set**: A register or bit with this attribute is set on read.
RC	**Read to Clear**: A register or bit with this attribute is cleared after the read, writes have no effect.
WO	**Write Only**: A register or bit with this attribute is write only, reads return zero.
WC	**Write One to Clear**: Writing a one to a bit with this attribute clears (‘0’) the value, writing a zero has no effect.
WS	**Write One to Set**: Writing a one to a bit with this attribute sets the value to ‘1’, writing a zero has no effect.

© 2012-2018 Microchip Technology Inc.

TABLE 34-7: REGISTER BIT ACCESS TYPES (CONTINUED)

Register Bit Type	Description
WZS	**Write Zero to Set**: Writing a zero to a bit with this attribute sets the value to '1', writing a one has no effect.
RES	**Reserved**: Reads of a register or bit with this attribute return zero, writes are ignored.

34.11.1 VCI REGISTER

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:13	Reserved	RES	0	–
12	Filters Bypass The Filters Bypass bit is used to enable and disable the input filters on the VCI_IN[5:0]# pins. When the Filters Bypass bit is '0,' the input filters are enabled (default). When the Filters Bypass bit is '1,' the input filters are disabled. See also Section 34.9, "Input Filtering," on page 504.	RW	0	VBAT_POR
11	GPO/nEXT The GPO/nEXT bit controls selecting between the external VBAT-Powered Control Interface inputs, or the VCI_FW_CNTRL bit output to control the VCI_OUT pin. When GPO/nEXT is set ('1'), VCI_OUT follows the VCI_FW_CNTRL bit setting. When GPO/nEXT is clear ('0'), VCI_OUT follows the external inputs.	RW	0	nSYS_RST
10	VCI_FW_CNTRL This bit can allow EC firmware to control the state of the VCI_OUT pin. For example, when VTR_PWRGD is asserted and the GPO/nEXT bit is '1', clearing the VCI_FW_CNTRL bit de-asserts the active high VCI_OUT pin. APPLICATION NOTE: BIOS should set the VCI_FW_CNTRL bit to 1 prior to setting the GPO/nEXT bit to 1 on power up.	RW	0	nSYS_RST
9	VCI_OUT This bit provides the current status of the VCI_OUT pin.	R	Note 34-1	nSYS_RST
8	VCI_OVRD_IN This bit provides the current status of the VCI_OVRD_IN pin.	R	Note 34-1	nSYS_RST
7:6	Reserved	RES	0	–
5:0	VCI_IN[5:0]# These bits provide the latched state of the associated VCI_IN# pin, if latching is enabled (Note 34-2); or, the current state of the pin if latching is not enabled.	R	Note 34-1	nSYS_RST

Note 34-1 The VCI_IN[5:0]#, VCI_OVRD_IN, and VCI_OUT pin bits default to the state of their respective input pins.

Note 34-2 latched bits are cleared by VBAT_POR.

34.11.2 LATCH ENABLE REGISTER

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:6	Reserved	RES	0	–
5:0	Latching Enables (LE) When a Latching Enable bit is asserted (‘1’), the latching function for the corresponding VCI_IN# pin is enabled (default for VCI_IN#[5:4]). When a LE bit is not asserted (‘0’), latching for the corresponding VCI_IN# pin is not enabled (default for VCI_IN#[3:0]).	RW	30h	VBAT_POR

34.11.3 LATCH RESETS REGISTER

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:6	Reserved	RES	0	–
5:0	Latch Resets (LS) When a Latch Resets bit is asserted (‘1’), the corresponding VCI_IN# latch is de-asserted (‘1’). **Note:** Reads of the Latch Resets Register are undefined.	WS	–	–

34.11.4 VCI INPUT ENABLE REGISTER

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:6	Reserved	RES	0	–
5:0	Input Enables (IE) When an Input Enables bit is asserted (‘1’) (default for VCI_IN[3:0]#) the corresponding input function is enabled. When an Input Enables bit is not asserted (‘0’) (default for VCI_IN[5:4]#) the corresponding VCI_IN# function is not enabled (Note 34-3). If a VCI input is not enabled, it does not affect the input status or the VCI_OUT pin, even if the input is ‘0.’ Latches are not asserted, even if VCI_IN# pin is low, during a VBAT power transition.	RW	Fh	VBAT POR

Note 34-3 The VCI_OVRD_IN pin is not affected by the VCI Input Enable Register and cannot be disabled.

© 2012-2018 Microchip Technology Inc.

34.11.5 VCI POLARITY REGISTER

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:6	Reserved	Reserved	0	–
5:0	VCI_IN_POL These bits determine the polarity of the VCI_IN input signals: 0: Active Low (default) 1: Active High. The value on the pins is inverted before use	RW	0	VBAT POR

34.11.6 VCI POSEDGE DETECT REGISTER

Offset	18h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:6	Reserved	Reserved	0	–
5:0	VCI_IN_POS] These bits record a low to high transition on the VCI_IN# pins. A “1” indicates a transition occurred. 0: No edge detected 1: Positive Edge Detected	RWC	0	VBAT POR

34.11.7 VCI NEGEDGE DETECT REGISTER

Offset	1Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:6	Reserved	Reserved	0	–
5:0	VCI_IN_NEG These bits record a high to low transition on the VCI_IN# pins. A “1” indicates a transition occurred. 0: No edge detected 1: Negative Edge Detected	RWC	0	VBAT POR

34.11.8 VCI BUFFER ENABLE REGISTER

Offset	20h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:6	Reserved	Reserved	0	–
5:0	VCI_BUFFER_EN These bits enable edge detection on the VCI_IN pins. 0: Edge detection enabled by the VCI_IN# field (default) 1: Edge detection enabled independent of the VCI_IN[# field	RW	0	VBAT POR

35.0 VBAT POWERED RAM

35.1 Abstract

35.1.1 CHIP LEVEL INTERFACE CLOCK DOMAIN/POWER DOMAIN CROSSINGS

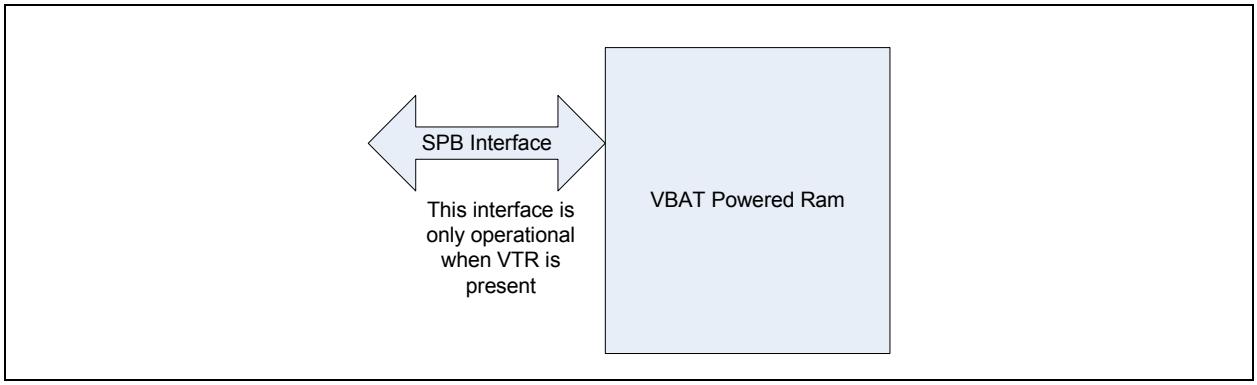
Powered by VBAT but only clocked or accessed when VTR power is available.

35.2 General Description

The VBAT Powered RAM provides a 64 Byte Random Accessed Memory that is operational while VTR is powered, and will retain its values while powered by VBAT powered and VTR is unpowered. The RAM is organized as a 16 “words” x 32-bit wide for a total of 64 bytes.

35.3 Block Diagram

FIGURE 35-1: BLOCK DIAGRAM OF VBAT POWERED RAM



35.4 Power, Clocks and Reset

35.4.1 POWER DOMAIN

This block is in the VTR power domain for EC interaction and uses the VBAT power domain for memory retention.

See Section 7.1.1, "Power Configuration," on page 95 for details on power domains.

35.4.2 CLOCKS

The VBAT Powered RAM has one clock input., the EC Bus Clock.

See Section 7.4, "Clock Generator," on page 98 for details on clocks.

35.4.3 POWER ON RESET

The VBAT Powered RAM is reset on a VBAT_POR.

See Section 7.6, "Reset Interface," on page 124 for details on reset.

35.5 Interrupts

The VBAT Powered RAM has no interrupts

© 2012-2018 Microchip Technology Inc.

35.6 Registers

The VBAT Powered RAM has its own Logical Device Number, and Base Address as indicated in Table 35-1. See Note 4-1 on page 60.

TABLE 35-1: VBAT Powered RAM BASE ADDRESS TABLE

VBAT Powered RAM Blocks	LDN from (Table 4-2 on page 59)	AHB Base Address
VBAT Backed Memory	33h	F0_CD00h

Each 32-bit RAM location is an SPB Offset from the AHB base address.

36.0 BLINKING/BREATHING PWM

36.1 Introduction

LEDs are used in computer applications to communicate internal state information to a user through a minimal interface. Typical applications will cause an LED to blink at different rates to convey different state information. For example, an LED could be full on, full off, blinking at a rate of once a second, or blinking at a rate of once every four seconds, in order to communicate four different states.

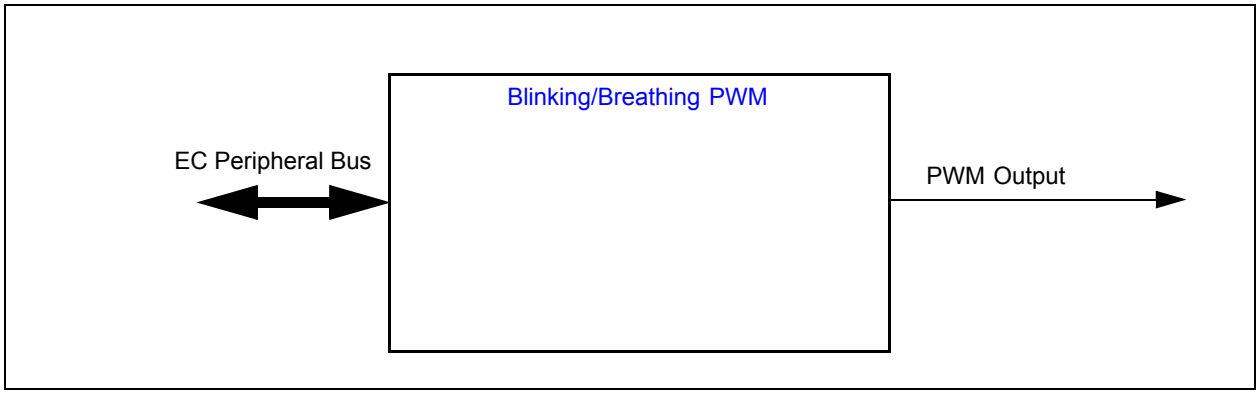
As an alternative to blinking, an LED can “breathe”, that is, oscillate between a bright state and a dim state in a continuous, or apparently continuous manner. The rate of breathing, or the level of brightness at the extremes of the oscillation period, can be used to convey state information to the user that may be more informative, or at least more novel, than traditional blinking.

The blinking/breathing hardware is implemented using a PWM. The PWM can be driven either by the 20.27 MHz clock or by a 32.768 KHz clock input. When driven by the 20.27 MHz clock, the PWM can be used as a standard 8-bit PWM in order to control a fan. When used to drive blinking or breathing LEDs, the 32.768 KHz clock source is used.

36.2 Interface

This block is designed to drive a pin on the pin interface and to be accessed internally via a registered host interface.

FIGURE 36-1: BLOCK DIAGRAM



36.2.1 SIGNAL DESCRIPTION

TABLE 36-1: SIGNAL DESCRIPTION TABLE

Name	Direction	Description
PWM Output	Output	Output of PWM
EC Peripheral Bus	Input/Output	EC Peripheral Bus
Sync_OUT	Output	Synchronize multiple instances utilizing the Synchronize (SYNC) bit in the LED CONFIGURATION (LED_CFG) register.
Sync_IN	Input	Synchronize multiple instances utilizing the Synchronize (SYNC) bit in the LED CONFIGURATION (LED_CFG) register.

36.2.2 HOST INTERFACE

The blinking/breathing PWM block is accessed by a controller over the standard register interface.

© 2012-2018 Microchip Technology Inc.

36.3 Power, Clocks and Reset

36.3.1 POWER DOMAINS

TABLE 36-2: POWER SOURCES

Name	Description
VTR	Main power. The source of main power for the device is system dependent.

36.3.2 CLOCK INPUTS

TABLE 36-3: CLOCK INPUTS

Name	Description
MCLK_5HZ_EN	5 Hz clock
X32K_CLK	32.768 KHz clock
MCLK	20.27 MHz clock

36.3.3 RESETS

TABLE 36-4: RESET SIGNALS

Name	Description
nSYS_RST	System reset

36.3.4 POWER MANAGEMENT

In all cases, X32K_CLK does not affect Power Management; i.e., the Blinking/Breathing PWM operates normally when MCLK is stopped, except as a General Purpose PWM. The sleep enable input has no affect on the Blinking/Breathing PWM and the clock required outputs are only asserted during register read/write cycles for as long as necessary to propagate updates to the block core (Table 36-5).

See also Section 36.5, "Low Power Mode," on page 512.

TABLE 36-5: BLINKING/BREATHING PWM POWER MANAGEMENT

SLEEP_EN	PWM Mode?	Bus Access Cycle?	CLK_REQ	Description
X	No	Yes	1	CLK_REQ is only asserted for as long as necessary to propagate updates to the block core
		No	0	CLK_REQ is not asserted when the EC is not accessing the register interface. (Note that this block *cannot* prevent the chip from entering the system deepest sleep states.)
	Yes	X	1	In PWM Mode, CLK_REQ is asserted whenever the PWM is enabled.

36.4 Interrupts

Each PWM can generate an interrupt. The interrupt is asserted for one 20.27 MHz clock period whenever the PWM WDT times out. The PWM WDT is described in Section 36.6.3.1, "PWM WDT," on page 515.

Note: LED0, LED1, and LED2 in the GIRQ15 Source Register are the interrupt source bits for the three instances of the Blinking/Breathing PWM in the MEC1632 (see Section 17.0, "EC Interrupt Aggregator," on page 287).

TABLE 36-6: INTERRUPTS

Source	Description
PWM_WDT	PWM watchdog time out

36.5 Low Power Mode

The Blinking/Breathing PWM may be put into a Low Power Mode by the chip-level power, clocks, and reset (PCR) circuitry (Table 36-7). Low Power Mode is only prevented in the General Purpose PWM mode. When the 32.768 KHz clock is selected, the Blinking/Breathing PWM function continues to operate, even when the 20.27 MHz clock is stopped.

Note: The LED[2:0] bits in EC Blocks Sleep Enables Register 2 and EC Blocks Clock Required Status Register 2 are the sleep enable and clock required status bits for the three instances of the Blinking/Breathing PWM in the MEC1632 (see Section 7.0, "Power, Clocks, and Resets," on page 95).

TABLE 36-7: LOW POWER MODE BEHAVIOR

Clock Source (Note 36-2	Control (Note 36-1)	Mode	Low Power Mode	Description
X	‘00’b	PWM ‘OFF’	Yes	32.768 KHz clock is required.
X	‘01’b	Breathing	Yes	
1	‘10’b	General Purpose PWM	No	20.27 MHz clock is required, even when a sleep command to the block is asserted.
0	‘10’b	Blinking	Yes	32.768 KHz clock is required.
X	‘11’b	PWM ‘ON’	Yes	

Note 36-1 The CONTROL field is Bits[1:0] in the LED CONFIGURATION (LED_CFG) register.

Note 36-2 CLOCK SOURCE is Bit2 in the LED CONFIGURATION (LED_CFG) register.

36.6 Description

36.6.1 BREATHING

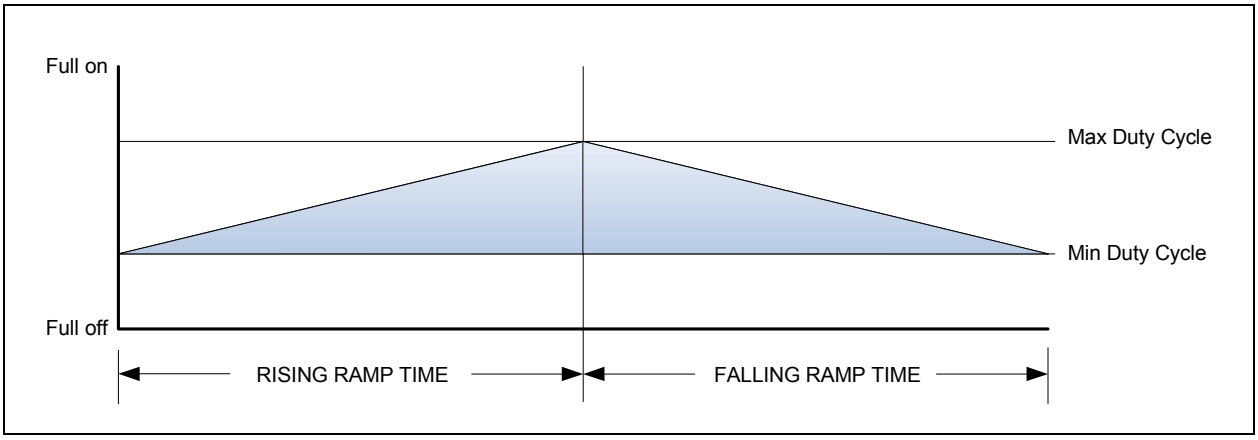
If an LED blinks rapidly enough, the eye will interpret the light as reduced brightness, rather than a blinking pattern. Therefore, if the blinking period is short enough, modifying the duty cycle will set the apparent brightness, rather than a blinking rate. At a blinking rate of 128Hz or greater, almost all people will perceive a continuous light source rather than an intermittent pattern.

Because making an LED appear to breathe is an aesthetic effect, the breathing mechanism must be adjustable or customers may find the breathing effect unattractive. There are several variables that can affect breathing appearance, as described below.

© 2012-2018 Microchip Technology Inc.

The following figure illustrates some of the variables in breathing:

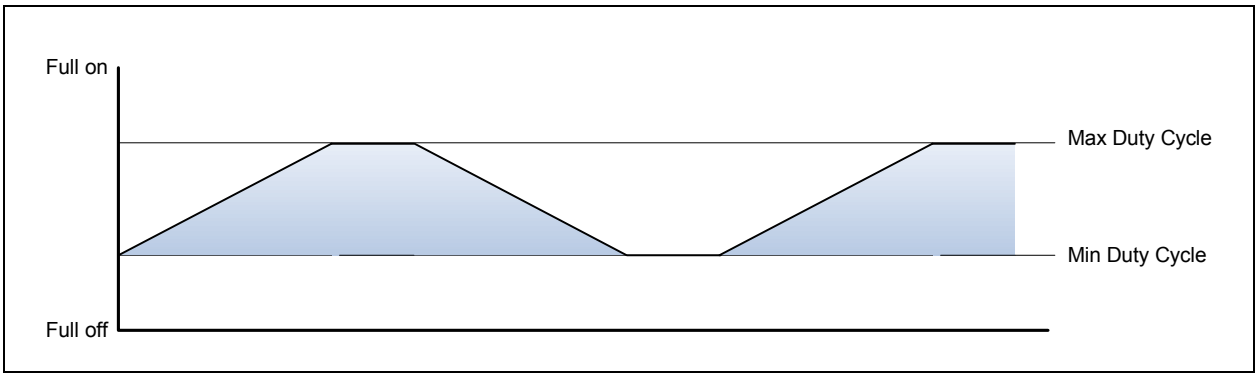
FIGURE 36-2: BREATHING LED EXAMPLE



The breathing range of and LED can range between full on and full off, or in a range that falls within the full-on/full-off range, as shown in this figure. The ramp time can be different in different applications. For example, if the ramp time was 1 second, the LED would appear to breathe quickly. A time of 2 seconds would make the LED appear to breathe more leisurely.

The breathing pattern can be clipped, as shown in the following figure, so that the breathing effect appears to pause at its maximum and minimum brightnesses:

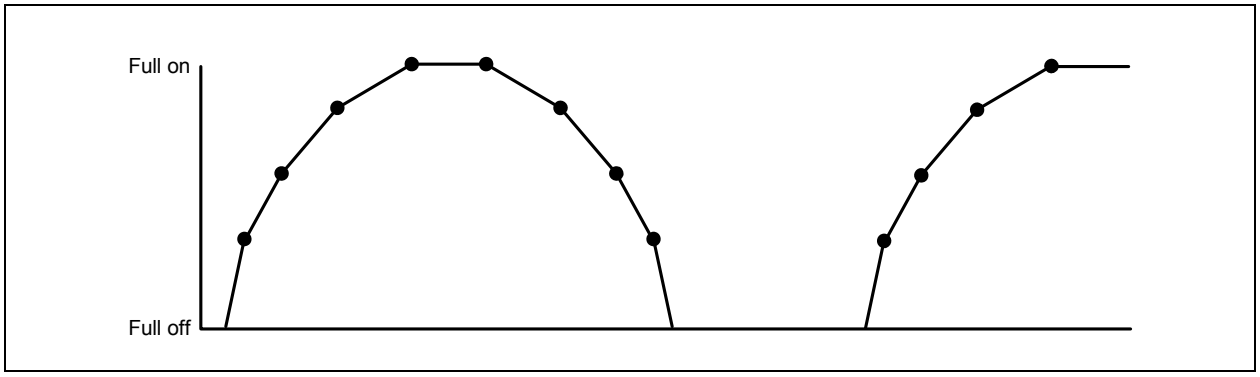
FIGURE 36-3: CLIPPING EXAMPLE



The clipping periods at the two extremes can be adjusted independently, so that for example an LED can appear to breathe (with a short delay at maximum brightness) followed by a longer “resting” period (with a long delay at minimum brightness).

The brightness can also be changed in a non-linear fashion, as shown in the following figure:

FIGURE 36-4: EXAMPLE OF A SEGMENTED CURVE



In this figure, the rise and fall curves are implemented in 4 linear segments and are the rise and fall periods are symmetric.

The breathing mode uses the 32.768 KHz clock for its time base.

36.6.2 BLINKING

When configured for blinking, a subset of the hardware used in breathing is used to implement the blinking function. The PWM (an 8-bit accumulator plus an 8-bit duty cycle register) drives the LED directly. The Duty Cycle register is programmed directly by the user, and not modified further. The PWM accumulator is configured as a simple 8-bit up counter. The counter uses the 32.768 KHz clock, and is pre-scaled by the Delay counter, to slow the PWM down from the 128Hz provided by directly running the PWM on the 32.768 KHz clock.

With the pre-scaler, the blink rate of the LED could be as fast as 128Hz (which, because it is blinking faster than the eye can distinguish, would appear as a continuous level) to 0.03125Hz (that is, with a period of 7.8ms to 32 seconds). Any duty cycle from 0% (0h) to 100% (FFh) can be configured, with an 8-bit precision. An LED with a duty cycle value of 0h will be fully off, while an LED with a duty cycle value of FFh will be fully on.

In Blinking mode the PWM counter is always in 8-bit mode.

Table 36-8, "LED Blink Configuration Examples" shows some example blinking configurations:

TABLE 36-8: LED BLINK CONFIGURATION EXAMPLES

Prescale	Duty Cycle	Blink Frequency	Blink	
			'ON' Time	'OFF' Time
000h	00h	128 Hz	–	FULL 'OFF'
000h	FFh	128 Hz	FULL 'ON'	–
001h	40h	64 Hz	3.9 msec	11.7 msec
003h	80h	32 Hz	15.6 msec	15.6 msec
07Fh	20h	1 Hz	125 msec	875 msec
0BFh	16h	0.66 Hz	129 msec	1.371 sec
0FFh	10h	0.5 Hz	125 msec	1.875 sec
180h	0Bh	0.33 Hz	129 msec	2.878 sec
1FFh	40h	0.25 Hz	1 sec	3 sec

The Blinking and General Purpose PWM modes share the hardware used in the breathing mode. The Prescale value is derived from the LD field of the LED_DELAY register and the Duty Cycle is derived from the MIN field of the LED LIMITS register.

© 2012-2018 Microchip Technology Inc.

TABLE 36-9: BLINKING MODE CALCULATIONS

Parameter	Unit	Equation (Note 36-3)
FREQUENCY	Hz	X32K_CLK/(PRESCALE + 1)/256
PERIOD	Seconds	1/FREQUENCY
'H' Width	Seconds	PERIOD x (DUTY CYCLE/256)
'L' Width	Seconds	PERIOD x (1 - DUTY CYCLE/256)

Note 36-3 see also Table 36-8 for programming examples.

36.6.3 GENERAL PURPOSE PWM

When used in the Blinking configuration with the MCLK, the LED module can be used as a general-purpose programmable Pulse-Width Modulator with an 8-bit programmable pulse width. It can be used for fan speed control, sound volume, etc. With the MCLK source, the PWM frequency can be configured in the range shown in Table 36-10.

TABLE 36-10: PWM CONFIGURATION EXAMPLES

Prescale	PWM Frequency
000h	79.2 KHz
001h	39.6 KHz
003h	19.8 KHz
006h	11.3 KHz
00Bh	6.6 KHz
07Fh	619 Hz
1FFh	155 Hz
FFFh	19.3 Hz

TABLE 36-11: GENERAL PURPOSE PWM MODE CALCULATIONS

Parameter	Unit	Equation (Note 36-4)
FREQUENCY	Hz	MCLK/(PRESCALE + 1)/256
PERIOD	Seconds	1/FREQUENCY
'H' Width	Seconds	PERIOD x (DUTY CYCLE/256)
'L' Width	Seconds	PERIOD x (1 - DUTY CYCLE/256)

Note 36-4 When the Duty Cycle is 00h, the General Purpose PWM is fully 'OFF'; when the Duty Cycle is FFh, the General Purpose PWM is fully 'ON'.'

36.6.3.1 PWM WDT

When the PWM is configured as a general-purpose PWM (in the Blinking configuration with the 20.27 MHz clock), the PWM includes a Watch Dog Timer (WDT). The WDT consists of an internal 8-bit counter and an 8-bit reload value (the field WDTLD in LED CONFIGURATION (LED_CFG) register). The internal counter is loaded with the reset value of WDTLD (14h, or 4 seconds) on system nSYS_RST and loaded with the contents of WDTLD whenever either the LED CONFIGURATION (LED_CFG) register is written or the MIN byte in the LED LIMITS (LED_LIMIT) register is written (the MIN byte controls the duty cycle of the PWM).

Whenever the internal counter is non-zero, it is decremented by 1 for every tick of the 5 Hz clock. If the counter decrements from 1 to 0, a WDT Terminal Count causes an interrupt to be generated and reset sets the Control (CNTRL) bit in the LED CONFIGURATION (LED_CFG) to 3h, which forces the PWM to be full on. No other PWM registers or fields are affected.

If the 5 Hz clock halts, the watchdog timer stops decrementing but retains its value, provided the device continues to be powered. When the 5 Hz clock restarts, the watchdog counter will continue decrementing where it left off.

© 2012-2018 Microchip Technology Inc.

Setting the WDTLD bits to 0 disables the PWM WDT. Other sample values for WDTLD are:

01h = 200 ms

02h = 400 ms

03h = 600 ms

04h = 800 ms

. . .

14h = 4seconds

FFh = 51 seconds

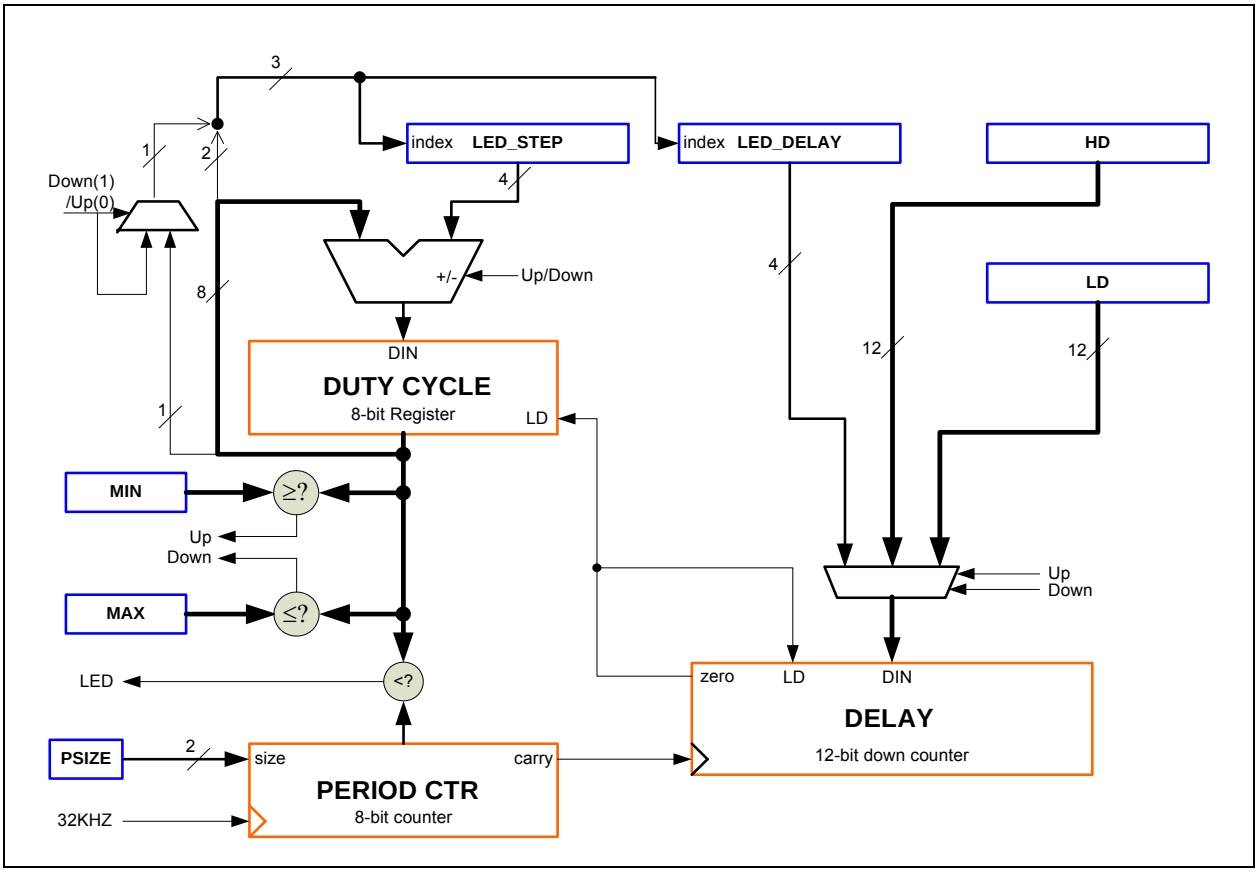
36.7 Implementation

In addition to the registers described in Section 36.8, "EC-Only Registers", the PWM is implemented using a number of components that are interconnected differently when configured for breathing operation and when configured for blinking/PWM operation.

36.7.1 BREATHING CONFIGURATION

When configured for breathing, components of the blinking/breathing hardware are interconnected as shown in the following figure:

FIGURE 36-5: BREATHING LED HARDWARE CONFIGURATION



In this figure, the boxes outlined in blue refer to registers, or fields in the registers, and boxes outlined in orange are internal counters and registers that are not directly accessible to firmware.

© 2012-2018 Microchip Technology Inc.

The **PSIZE** parameter can configure the PWM to one of three modes: 8-bit, 7-bit and 6-bit. The **PERIOD CTR** counts ticks of its input clock. In 8-bit mode, it counts from 0 to 255 (that is, 256 steps), then repeats continuously. In this mode, a full cycle takes 7.8ms (128Hz). In 7-bit mode it counts from 0 to 127 (128 steps), and a full cycle takes 3.9ms (256Hz). In 6-bit mode it counts from 0 to 63 (64 steps) and a full cycle takes 1.95ms (512Hz).

The output of the LED circuit is asserted whenever the **PERIOD CTR** is less than the contents of the **DUTY CYCLE** register. The appearance of breathing is created by modifying the contents of the **DUTY CYCLE** register in a continuous manner. When the LED control is off the internal counters and registers are all reset to 0 (i.e. after a write setting the RESET bit in the LED CONFIGURATION (LED_CFG) Register.) Once enabled, the **DUTY CYCLE** register is increased by an amount determined by the LED_STEP register and at a rate determined by the **DELAY** counter. Once the duty cycle reaches its maximum value (determined by the field MAX), the duty cycle is held constant for a period determined by the field HD. Once the hold time is complete, the **DUTY CYCLE** register is decreased, again by an amount determined by the LED_STEP register and at a rate determined by the **DELAY** counter. When the duty cycle then falls at or below the minimum value (determined by the field MIN), the duty cycle is held constant for a period determined by the field LD. Once the hold time is complete, the cycle repeats, with the duty cycle oscillating between MIN and MAX.

The rising and falling ramp times as shown in FIGURE 36-2: Breathing LED Example on page 513 can be either symmetric or asymmetric depending on the setting of the Symmetry (SYM) bit in the LED CONFIGURATION (LED_CFG) Register. In Symmetric mode the rising and falling ramp rates have mirror symmetry; both rising and falling ramp rates use the same (all) 8 segments fields in each of the following registers (see Table 36-12): the LED UPDATE STEPSIZE (LED_STEP) register and the LED UPDATE INTERVAL (LED_INT) register. In Asymmetric mode the rising ramp rate uses 4 of the 8 segments fields and the falling ramp rate uses the remaining 4 of the 8 segments fields (see Table 36-12).

The parameters MIN, MAX, HD, LD and the 8 fields in LED_STEP and LED_INT determine the brightness range of the LED and the rate at which its brightness changes. See the descriptions of the fields in Section 36.8, "EC-Only Registers", as well as the examples in Section 36.7.3, "Breathing Examples" for information on how to set these fields.

TABLE 36-12: SYMMETRIC BREATHING MODE REGISTER USAGE

Rising/ Falling Ramp Times in FIGURE 36-2: on page 513	Duty Cycle	Segment Index	Symmetric Mode Register Fields Utilized	
X	000xxxxxb	000b	STEP[0]/INT[0]	Bits[3:0]
X	001xxxxxb	001b	STEP[1]/INT[1]	Bits[7:4]
X	010xxxxxb	010b	STEP[2]/INT[2]	Bits[11:8]
X	011xxxxxb	011b	STEP[3]/INT[3]	Bits[15:12]
X	100xxxxxb	100b	STEP[4]/INT[4]	Bits[19:16]
X	101xxxxxb	101b	STEP[5]/INT[5]	Bits[23:20]
X	110xxxxxb	110b	STEP[6]/INT[6]	Bits[27:24]
X	111xxxxxb	111b	STEP[7]/INT[7]	Bits[31:28]
Note: In Symmetric Mode the Segment_Index[2:0] = Duty Cycle Bits[7:5]				

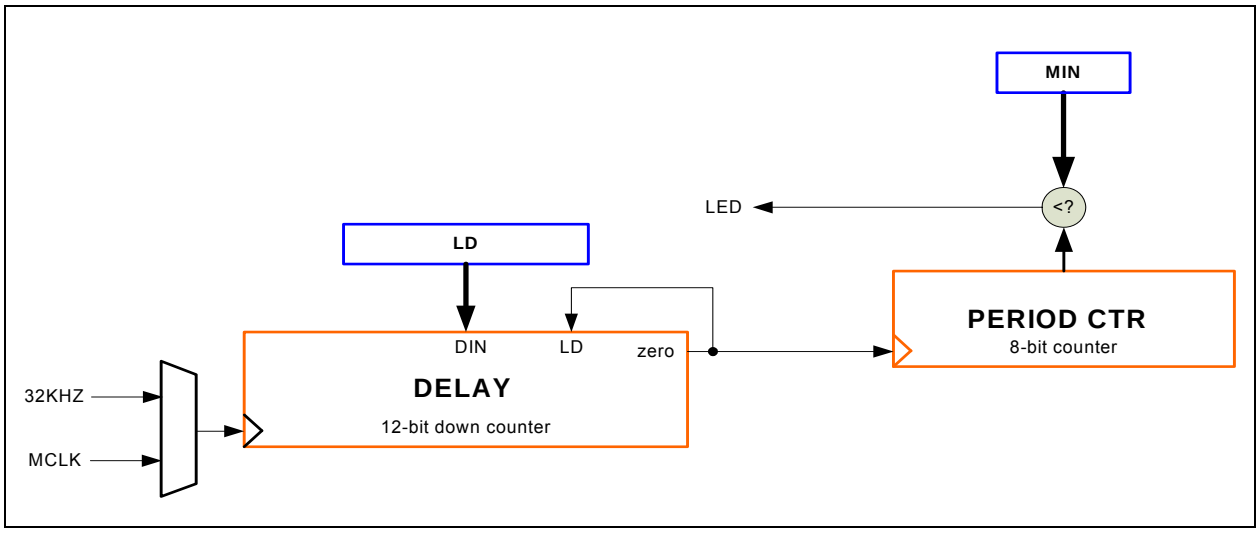
TABLE 36-13: ASYMMETRIC BREATHING MODE REGISTER USAGE

Rising/ Falling Ramp Times in FIGURE 36-2: on page 513	Duty Cycle	Segment Index	Asymmetric Mode Register Fields Utilized	
Rising	00xxxxxxb	000b	STEP[0]/INT[0]	Bits[3:0]
Rising	01xxxxxxb	001b	STEP[1]/INT[1]	Bits[7:4]
Rising	10xxxxxxb	010b	STEP[2]/INT[2]	Bits[11:8]
Rising	11xxxxxxb	011b	STEP[3]/INT[3]	Bits[15:12]
falling	00xxxxxxb	100b	STEP[4]/INT[4]	Bits[19:16]
falling	01xxxxxxb	101b	STEP[5]/INT[5]	Bits[23:20]
falling	10xxxxxxb	110b	STEP[6]/INT[6]	Bits[27:24]
falling	11xxxxxxb	111b	STEP[7]/INT[7]	Bits[31:28]
Note: In Asymmetric Mode the Segment_Index[2:0] is the bit concatenation of following: Segment_Index[2] = (FALLING RAMP TIME in FIGURE 36-2: on page 513) and Segment_Index[1:0] = Duty Cycle Bits[7:6].				

36.7.2 BLINKING CONFIGURATION

When configured for blinking or standard PWM operation, components of the blinking/breathing hardware are interconnected as shown in the following figure:

FIGURE 36-6: BREATHING LED HARDWARE CONFIGURATION



The Delay counter and the PWM counter are the same as in the breathing configuration, except in this configuration they are connected differently. The Delay counter is clocked on either the 32.768 KHz clock or the 20.27 MHz clock, rather than the output of the PWM. The PWM counter is clocked by the zero output of the Delay counter, which functions as a prescalar for the input clocks to the PWM. The Delay counter is reloaded from the LD field of the LED_DELAY register. When the LD field is 0 the input clock is passed directly to the PWM counter without prescaling. In Blinking/PWM mode the PWM counter is always 8-bit, and the PSIZE parameter has no effect.

© 2012-2018 Microchip Technology Inc.

The frequency of the PWM pulse waveform is determined by the formula:

$$f_{PWM} = \frac{f_{clock}}{(256 \times (LD + 1))}$$

where f_{PWM} is the frequency of the PWM, f_{clock} is the frequency of the input clock (32.768 KHz clock or 20.27 MHz clock) and LD is the contents of the LD field.

Note:	When the Duty Cycle is set to 00h, the PWM is fully off. When the Duty Cycle is set to FFh, the PWM is fully on. This is equivalent to setting the Configuration Register CONTROL field to Always Off and Always On.

The other registers in the block do not affect the PWM or the LED output in Blinking/PWM mode.

36.7.3 BREATHING EXAMPLES

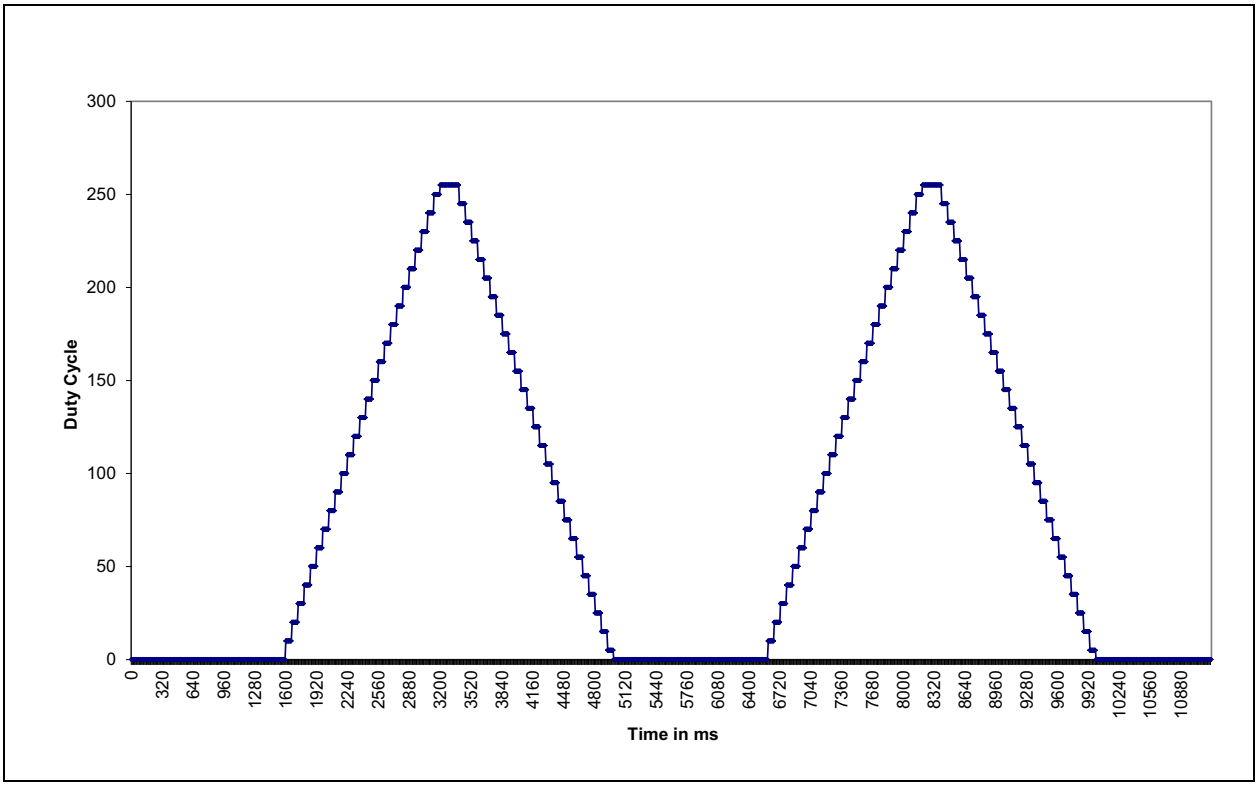
36.7.3.1 Linear LED brightness change

In this example, the brightness of the LED increases and diminishes in a linear fashion. The entire cycle takes 5 seconds. The rise time and fall time are 1.6 seconds, with a hold time at maximum brightness of 200ms and a hold time at minimum brightness of 1.6 seconds. The LED brightness varies between full off and full on. The PWM size is set to 8-bit, so the time unit for adjusting the PWM is approximately 8ms. The registers are configured as follows:

TABLE 36-14: LINEAR EXAMPLE CONFIGURATION

FIELD	**VALUE**							
PSIZE	8-bit							
MAX	255							
MIN	0							
HD	25 ticks (200ms)							
LD	200 ticks (1.6s)							
Duty cycle most significant bits	000b	001b	010b	011b	100b	101b	110b	1110
LED_INT	8	8	8	8	8	8	8	8
LED_STEP	10	10	10	10	10	10	10	10

FIGURE 36-7: LINEAR BRIGHTNESS CURVE EXAMPLE



36.7.3.2 Non-linear LED Brightness Change

In this example, the brightness of the LED increases and diminishes in a non-linear fashion. The brightness forms a curve that is approximated by four piece wise-linear line segments. The entire cycle takes about 2.8 seconds. The rise time and fall time are about 1 second, with a hold time at maximum brightness of 320ms and a hold time at minimum brightness of 400ms. The LED brightness varies between full off and full on. The PWM size is set to 7-bit, so the time unit for adjusting the PWM is approximately 4ms. The registers are configured as follows:

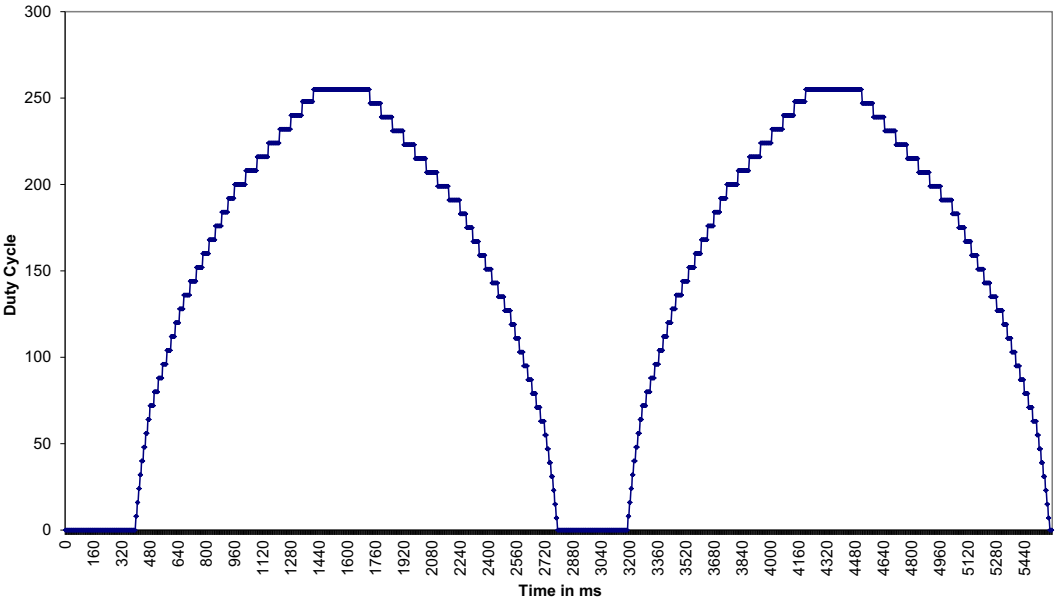
TABLE 36-15: NON-LINEAR EXAMPLE CONFIGURATION

FIELD	**VALUE**							
PSIZE	7-bit							
MAX	255 (effectively 127)							
MIN	0							
HD	80 ticks (320ms)							
LD	100 ticks (400ms)							
Duty cycle most significant bits	000b	001b	010b	011b	100b	101b	110b	1110
LED_INT	2	3	6	6	9	9	16	16
LED_STEP	4	4	4	4	4	4	4	4

© 2012-2018 Microchip Technology Inc.

The resulting curve is shown in the following figure:

FIGURE 36-8: NON-LINEAR BRIGHTNESS CURVE EXAMPLE



36.7.4 BLINKING/BREATHING PWM ACTIVE LEVEL

The Blinking/Breathing PWM is active high. When the PWM is configured to be fully on, the pin is driving high. When the PWM is configured to be fully off, which is the default state, the pin is low.

If firmware requires the Blinking/Breathing PWM to be active low, the Polarity bit in the GPIO Configuration Register associated with the PWM can be set to 1, which inverts the output polarity.

36.8 EC-Only Registers

The registers listed in the EC-Only Registers Address Range Table table are for a single instance of the Blinking/Breathing PWM. The addresses of each register listed in this table are defined as a relative offset to the host “Base Address” defined in EC-Only Registers Address Range Table.

TABLE 36-16: EC-Only Registers ADDRESS RANGE TABLE

Block Instance	Instance Number	Host	Address Space	Base Address
LED	0	EC	24-bit internal address space	F0_8400h
LED	1	EC	24-bit internal address space	F0_8400h + 80h
LED	2	EC	24-bit internal address space	F0_8400h + 100h

TABLE 36-17: EC-ONLY REGISTER SUMMARY

Offset	Regiser Name (Mnemonic)
00h	LED CONFIGURATION (LED_CFG)
04h	LED LIMITS (LED_LIMIT)
08h	LED DELAY (LED_DELAY)
0Ch	LED UPDATE STEPSIZE (LED_STEP)
10h	LED UPDATE INTERVAL (LED_INT)
14h	LED OUTPUT DELAY (LED_ODLY)

In the following register definitions, a “PWM period” is defined by time the PWM counter goes from 000h to its maximum value (FFh in 8-bit mode, FEh in 7-bit mode and FCh in 6-bit mode, as defined by the PSCALE field in register LED_CFG). The end of a PWM period occurs when the PWM counter wraps from its maximum value to 0.

The registers in this block can be written 32-bits, 16-bits or 8-bits at a time. Writes to LED CONFIGURATION (LED_CFG) take effect immediately. Writes to LED LIMITS (LED_LIMIT) are held in a holding register and only take effect only at the end of a PWM period. The update takes place at the end of every period, even if only one byte of the register was updated. This means that in blink/PWM mode, software can change the duty cycle with a single 8-bit write to the MIN field in the LED_LIMIT register. Writes to LED DELAY (LED_DELAY), LED UPDATE STEPSIZE (LED_STEP) and LED UPDATE INTERVAL (LED_INT) also go initially into a holding register. The holding registers are copied to the operating registers at the end of a PWM period only if the Enable Update bit in the LED CONFIGURATION (LED_CFG) is set to 1. If LED_CFG is 0, data in the holding registers is retained but not copied to the operating registers when the PWM period expires. To change an LED breathing configuration, software should write these three registers with the desired values and then set LED_CFG to 1. This mechanism ensures that all parameters affecting LED breathing will be updated consistently, even if the registers are only written 8 bits at a time.

© 2012-2018 Microchip Technology Inc.

36.8.1 LED CONFIGURATION (LED_CFG)

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	RESERVED	RES	-	-
16	Symmetry (SYM) When this bit is a 0, the rising and falling ramp times (as shown in FIGURE 36-2: Breathing LED Example on page 513) are in Symmetric mode. Table 36-12 defines the Segment Index that are utilized by LED UPDATE STEPSIZE (LED_STEP) register and the LED UPDATE INTERVAL (LED_INT) register. When this bit is a 1, the rising and falling ramp times (as shown in FIGURE 36-2: Breathing LED Example on page 513) are in Asymmetric mode. Table 36-13 defines the Signet Index that are utilized by LED UPDATE STEPSIZE (LED_STEP) register and the LED UPDATE INTERVAL (LED_INT) register.	R/W	0b	nSYS-_RST
15:8	WDT Reload (WDTLD) The PWM Watchdog Timer counter reload value. On system reset, it defaults to 14h, which corresponds to a 4 second Watchdog timeout value.	R/W	14h	nSYS-_RST
7	RESET Writes a '1' to this bit resets the Blinking/Breathing PWM Registers their default values. This bit is self clearing. Writes a '0' to this bit has no effect.	WO	0b	nSYS-_RST
6	Enable Update (ENUP; HW_CLR) This bit is set to 1 when written with a 1. Writes of 0 have no effect. Hardware clears this bit to 0 when the breathing configuration registers are updated at the end of a PWM period. The current state of the bit is readable any time. This bit is used to enable consistent configuration of LED_DELAY, LED_STEP and LED_INT. As long as this bit is 0, data written to those three registers is retained in a holding register. When this bit is 1, data in the holding register are copied to the operating registers at the end of a PWM period. When the copy completes, hardware clears this bit to 0.	R/WS	0b	nSYS-_RST
5:4	PWM Size (PSIZE) This bit controls the behavior of PWM: 0 = PWM is configured as an 8-bit PWM 1 = PWM is configured as a 7-bit PWM 2 = PWM is configured as a 6-bit PWM 3 = Reserved	R/W	0b	nSYS-_RST
3	Synchronize (SYNC) When this bit is 1, all counters for all LEDs are reset to their initial values. When this bit is 0 in the LED CONFIGURATION (LED_CFG) Register for all LEDs, then all counters for LEDs that are configured to blink or breathe will increment or decrement, as required. To synchronize blinking or breathing, the SYNC bit should be set for at least one LED, the control registers for each LED should be set to their required values, then the SYNC bits should all be cleared. If the all LEDs are set for the same blink period, they will all be synchronized.	R/W	0b	nSYS-_RST

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
2	Clock Source (CLKSRC) This bit controls the base clock for the PWM. It is only valid when CNTRL is set to blink (2). 0 = Clock source is the 32.768 KHz clock 1 = Clock source is the 20.27 MHz clock	R/W	0b	nSYS-_RST
1:0	Control (CNTRL) This bit controls the behavior of PWM: 0 = PWM is always off. All internal registers and counters are reset to 0. Clocks are gated 1 = LED breathing configuration 2 = LED blinking (standard PWM) 3 = PWM is always on	R/W	00b 11b	nSYS-_RST WDT TC

36.8.2 LED LIMITS (LED_LIMIT)

This register may be written at any time. Values written into the register are held in an holding register, which is transferred into the actual register at the end of a PWM period. The two byte fields may be written independently. Reads of this register return the current contents and not the value of the holding register.

Offset	04h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:16	RESERVED	RES	-	-
15:8	Maximum (MAX) In breathing mode, when the current duty cycle is greater than or equal to this value the breathing apparatus holds the current duty cycle for the period specified by the field HD in register LED_DELAY, then starts decrementing the current duty cycle	R/W	0h	nSYS-_RST
7:0	Minimum (MIN) In breathing mode, when the current duty cycle is less than or equal to this value the breathing apparatus holds the current duty cycle for the period specified by the field LD in register LED_DELAY, then starts incrementing the current duty cycle In blinking mode, this field defines the duty cycle of the blink function.	R/W	0h	nSYS-_RST

© 2012-2018 Microchip Technology Inc.

36.8.3 LED DELAY (LED_DELAY)

This register may be written at any time. Values written into the register are held in an holding register, which is transferred into the actual register at the end of a PWM period if the Enable Update bit in the LED Configuration register is set to 1. Reads of this register return the current contents and not the value of the holding register.

Offset	08h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:24	RESERVED	RES	-	-
23:12	High Delay (HD) In breathing mode, the number of PWM periods to wait before updating the current duty cycle when the current duty cycle is greater than or equal to the value MAX in register LED_LIMIT. 0 = The delay counter is bypassed and the current duty cycle is decremented after one PWM period 1 = The delay counter is bypassed and the current duty cycle is decremented after two PWM period … 4095: The current duty cycle is decremented after 4096 PWM periods	R/W	000h	nSYS_RST
11:0	Low Delay (LD) The number of PWM periods to wait before updating the current duty cycle when the current duty cycle is greater than or equal to the value MIN in register LED_LIMIT. 0 = The delay counter is bypassed and the current duty cycle is incremented after one PWM period 1 = The delay counter is bypassed and the current duty cycle is incremented after two PWM period … 4095: The current duty cycle is incremented after 4096 PWM periods In blinking mode, this field defines the prescalar for the PWM clock	R/W	000h	nSYS_RST

36.8.4 LED UPDATE STEPSIZE (LED_STEP)

This register has eight segment fields which provide the amount the current duty cycle is adjusted at the end of every PWM period. Segment field selection is decoded based on the segment index. The segment index equation utilized depends on the Symmetry (SYM) bit in the LED CONFIGURATION (LED_CFG) Register (See Table 36-12 on page 517 and Table 36-13 on page 518 for detailed description of segment selection)

- In Symmetric Mode the Segment_Index[2:0] = Duty Cycle Bits[7:5]
- In Asymmetric Mode the Segment_Index[2:0] is the bit concatenation of following: Segment_Index[2] = (FALLING RAMP TIME in FIGURE 36-2: on page 513) and Segment_Index[1:0] = Duty Cycle Bits[7:6].

This register may be written at any time. Values written into the register are held in an holding register, which is transferred into the actual register at the end of a PWM period if the Enable Update bit in the LED Configuration register is set to 1. Reads of this register return the current contents and not the value of the holding register.

In 8-bit mode, each 4-bit STEPSIZE field represents 16 possible duty cycle modifications, from 1 to 16 as the duty cycle is modified between 0 and 255:

0: Modify the duty cycle by 1

1: Modify the duty cycle by 2

...

15: Modify the duty cycle by 16

In 7-bit mode, the least significant bit of the 4-bit field is ignored, so each field represents 8 possible duty cycle modifications, from 1 to 8, as the duty cycle is modified between 0 and 127:

0, 1: Modify the duty cycle by 1

2, 3: Modify the duty cycle by 2

...

14, 15: Modify the duty cycle by 8

In 6-bit mode, the two least significant bits of the 4-bit field is ignored, so each field represents 4 possible duty cycle modifications, from 1 to 4 as the duty cycle is modified between 0 and 63:

0, 1, 2, 3: Modify the duty cycle by 1

4, 5, 6, 7: Modify the duty cycle by 2

8, 9, 10, 11: Modify the duty cycle by 3

12, 13, 14, 15: Modify the duty cycle by 4

Offset	0Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:28	Update Step7 (STEP7) Amount the current duty cycle is adjusted at the end of every PWM period when the segment index is equal to 111.	R/W	0h	nSYS_RST
27:24	Update Step6 (STEP6) Amount the current duty cycle is adjusted at the end of every PWM period when the segment index is equal to 110.	R/W	0h	nSYS_RST
23:20	Update Step5 (STEP5) Amount the current duty cycle is adjusted at the end of every PWM period when the segment index is equal to 101	R/W	0h	nSYS_RST
19:16	Update Step4 (STEP4) Amount the current duty cycle is adjusted at the end of every PWM period when the segment index is equal to 100.	R/W	0h	nSYS_RST
15:12	Update Step3 (STEP3) Amount the current duty cycle is adjusted at the end of every PWM period when the segment index is equal to 011.	R/W	0h	nSYS_RST
11:8	Update Step2 (STEP2) Amount the current duty cycle is adjusted at the end of every PWM period when the segment index is equal to 010.	R/W	0h	nSYS_RST
7:4	Update Step1 (STEP1) Amount the current duty cycle is adjusted at the end of every PWM period when the segment index is equal to 001.	R/W	0h	nSYS_RST
3:0	Update Step0 (STEP0) Amount the current duty cycle is adjusted at the end of every PWM period when the segment index is equal to 000.	R/W	0h	nSYS_RST

36.8.5 LED UPDATE INTERVAL (LED_INT)

This register has eight segment fields which provide the number of PWM periods between updates to current duty cycle. Segment field selection is decoded based on the segment index. The segment index equation utilized depends on the Symmetry (SYM) bit in the LED CONFIGURATION (LED_CFG) Register (See Table 36-12 on page 517 and Table 36-13 on page 518 for detailed description of segment selection)

- In Symmetric Mode the Segment_Index[2:0] = Duty Cycle Bits[7:5]
- In Asymmetric Mode the Segment_Index[2:0] is the bit concatenation of following: Segment_Index[2] = (FALLING RAMP TIME in FIGURE 36-2: on page 513) and Segment_Index[1:0] = Duty Cycle Bits[7:6].

© 2012-2018 Microchip Technology Inc.

This register may be written at any time. Values written into the register are held in an holding register, which is transferred into the actual register at the end of a PWM period if the Enable Update bit in the LED Configuration register is set to 1. Reads of this register return the current contents and not the value of the holding register.

Offset	10h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:28	Update Interval7 (INT7) The number of PWM periods between updates to current duty cycle when the segment index is equal to 111. 0 = Wait 1 PWM period … 15 = Wait 16 PWM periods	R/W	0h	nSYS_RST
27:24	Update Interval6 (INT6) The number of PWM periods between updates to current duty cycle when the segment index is equal to 110. 0 = Wait 1 PWM period … 15 = Wait 16 PWM periods	R/W	0h	nSYS_RST
23:20	Update Interval5 (INT5) The number of PWM periods between updates to current duty cycle when the segment index is equal to 101. 0 = Wait 1 PWM period … 15 = Wait 16 PWM periods	R/W	0h	nSYS_RST
19:16	Update Interval4 (INT4) The number of PWM periods between updates to current duty cycle when the segment index is equal to 100. 0 = Wait 1 PWM period … 15 = Wait 16 PWM periods	R/W	0h	nSYS_RST
15:12	Update Interval3 (INT3) The number of PWM periods between updates to current duty cycle when the segment index is equal to 011. 0 = Wait 1 PWM period … 15 = Wait 16 PWM periods	R/W	0h	nSYS_RST
11:8	Update Interval2 (INT2) The number of PWM periods between updates to current duty cycle when the segment index is equal to 010. 0 = Wait 1 PWM period … 15 = Wait 16 PWM periods	R/W	0h	nSYS_RST
7:4	Update Interval1 (INT1) The number of PWM periods between updates to current duty cycle when the segment index is equal to 001. 0 = Wait 1 PWM period … 15 = Wait 16 PWM periods	R/W	0h	nSYS_RST
3:0	Update Interval0 (INT0) The number of PWM periods between updates to current duty cycle when the segment index is equal to 000. 0 = Wait 1 PWM period … 15 = Wait 16 PWM periods	R/W	0h	nSYS_RST

© 2012-2018 Microchip Technology Inc.

36.8.6 LED OUTPUT DELAY (LED_ODLY)

This register permits the transitions for multiple blinking/breathing LED outputs to be skewed, so as not to present too great a current load. The register defines a count for the number of clocks the circuitry waits before turning on the output, either on initial enable, after a resume from Sleep, or when multiple outputs are synchronized through the Sync control in the LED CONFIGURATION (LED_CFG) register.

When more than one LED outputs are used simultaneously, the LED OUTPUT DELAY fields of each should be configured with different values so that the outputs are skewed. When used with the 32KHz clock domain as a clock source, the differences can be as small as 1.

Offset	14h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	RESERVED	RES	-	-
7:0	Output Delay (ODLY) The delay, in counts of the clock defined in Clock Source (CLKSRC), in which output transitions are delayed. When this field is 0, there is no added transition delay. When the LED is programmed to be Always On or Always Off, the Output Delay field has no effect.	R/W	0h	nSYS_RST

37.0 PS/2 DEVICE INTERFACE

37.1 General Description

There are three PS/2 Ports in the MEC1632 independent EC PS/2 serial ports implemented in hardware which are directly controlled by the EC (see FIGURE 37-1: on page 530). The hardware implementation eliminates the need to bit bang I/O ports to generate PS/2 traffic, however bit banging is available via the associated GPIO pins.

The EC PS/2 serial channels use a synchronous serial protocol to communicate with the auxiliary device. A PS/2 channel has Clock and Data signal lines. The signal lines are bi-directional and employ open drain outputs capable of sinking 16mA. A pull-up resistor, typically 10K, is connected to both lines. This allows either the MEC1632 EC PS/2 logic or the auxiliary device to drive the lines. Regardless of the drive source, the auxiliary device always provides the clock for transmit and receive operations. The serial packet is made up of eleven bits, listed in the order they appear on the data line: start bit, eight data bits (least significant bit first), odd parity, and stop bit. Each bit cell is from 60µS to 100µS long.

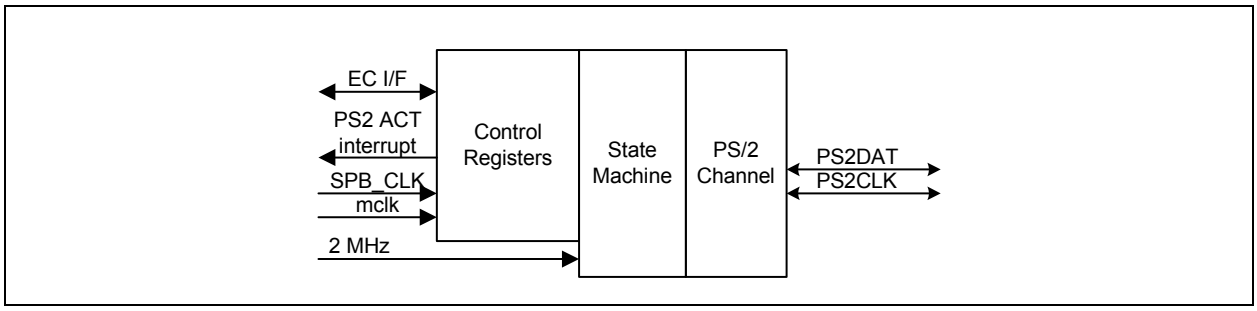
All PS/2 Serial Channel signals (CLK and DAT) are driven by open drain drivers which can be pulled to VTR or VCC (+3.3V nominal) through 10K-ohm resistors.

The MEC1632 supports a PS/2 Wake Interface that can wake the EC from the IDLE or SLEEP states. The PS/2 Wake Interface is powered by VTR and can generate wake interrupts without a clock.

APPLICATION NOTE: The PS/2 Wake Interface is only active when the KBC/Mouse and external pull-up resisters are powered by the VCC1supply. The external pull-up resister must always be powered by the same source as the KBC/Mouse.

37.2 Block Diagram

FIGURE 37-1: PORT PS/2 BLOCK DIAGRAM



37.3 PS/2 Port Physical Layer Byte Transmission Protocol

The PS/2 physical layer transfers a byte of data via an eleven bit serial stream as shown in Table 37-1. A logic 1 is sent at an active high level. Data sent from a Keyboard or mouse device to the host is read on the falling edge of the clock signal. The Keyboard or mouse device always generates the signal. The Host may inhibit communication by pulling the Clock line low. The Clock line must be continuously high for at least 50 microseconds before the Keyboard or mouse device can begin to transmit its data. See Table 37-2, "PS/2 Port Physical Layer Bus States".

© 2012-2018 Microchip Technology Inc.

TABLE 37-1: PS/2 PORT PHYSICAL LAYER BYTE TRANSMISSION PROTOCOL

Bit	Function
1	Start bit (always 0)
2	Data bit 0 (least significant bit)
3	Data bit 1
4	Data bit 2
5	Data bit 3
6	Data bit 4
7	Data bit 5
8	Data bit 6
9	Data bit 7 (most significant bit)
10	Parity bit (odd parity)
11	Stop Bit (always 1)

FIGURE 37-2: PS/2 PORT PHYSICAL LAYER BYTE TRANSMISSION PROTOCOL

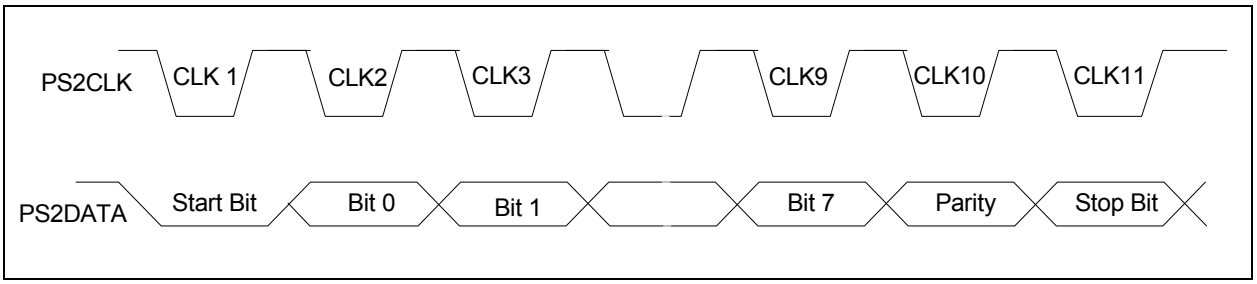


TABLE 37-2: PS/2 PORT PHYSICAL LAYER BUS STATES

Data	Clock	State
high	high	Idle
high	low	Communication Inhibited
low	low	Request to Send

37.4 Block Diagram SIgnal List

TABLE 37-3: PS/2 PORT LIST

Signal Name	Direction	Description
PS2DAT	INPUT/OUTPUT	Data from the PS/2 device
PS2CLK	INPUT/OUTPUT	Clock from the PS/2 device
nSYS_RST	INPUT	Chip Power on Reset (i.e., Suspend Well)
SPB_CLK	INPUT	Clock Source to EC micro controller. Used for reading/writing registers on the EC memory i/f.
mclk	INPUT	MCLK
2 MHz	INPUT	MCLK_DIV10_EN, State machine clock

TABLE 37-3: PS/2 PORT LIST (CONTINUED)

Signal Name	Direction	Description
SLEEP_EN	INPUT	External Power Management enable/disable input signal used to put the block in the lowest power consumption state. 0=No Sleep Requested. The block should operate as configured. 1=Sleep Requested. The block enters sleep mode.
CLK_REQ	OUTPUT	Power Management output indicates when this block requires MCLK input. 0= Clock can be turned 'off' when appropriate 1= Clock is required to be 'on.'
EC Interface	I/O BUS	EC-side SPB bus
PS2 ACT	OUTPUT	Asynchronous Interrupt

37.5 Power, Clocks and Reset

37.5.1 POWER DOMAIN

This block is powered by the VTR Power Supply.

See Section 7.1.1, "Power Configuration," on page 95 for details on power domains.

37.5.2 CLOCKS

This block uses the EC Bus Clock, MCLK and the 2MHz MCLK_DIV10_EN. EC Bus Clock is used when reading and writing the PS/2 Device Interface registers. The state machine is clocked with MCLK_DIV10_EN.

See Section 7.4, "Clock Generator," on page 98 for details on clocks.

37.6 Reset

This block is reset on a nSYS_RST.

See Section 7.6, "Reset Interface," on page 124 for details on reset.

37.7 Instance Description

There are three block instances defined in this chapter: PS/2[0,1,2]. The pin signals are defined in the PS/2 Interface table defined in the Pin Configuration chapter. PS/2 ports ending with signal functions ending with “A” or “B” are muxed to a single controller. Only one Port set of clock and data are intended to used at a time (either “A” or “B” not both.) The unused port segment should have its associated Pin Control Register’ s, Mux Control Field programmed away from the PS2 controller. See Section 24.0, "GPIO Interface," on page 388.

37.8 Interrupts

Each EC PS/2 Channel has two interrupts: a PS/2 activity interrupt event and a START Bit detection Wake-up event. A PS/2 Channel activity interrupt event is generated by changes in status bits in this block. The PS/2 Channel activity interrupt event are routed to the PS2_ACT_0, PS2_ACT_1, PS2_ACT_2 bits in the GIRQ19 Source Register on page 318. The START Bit detection Wake-up event is a PS/2 Channel/segment (see Section 37.9) Data pin signal edge detection interrupt and wake event. Each PS/2 Channel/segment START Bit detection Wake-up event is controlled by their associated Data pin’s Pin Control Register. (See Section 24.0, "GPIO Interface," on page 388. The START Bit detection Wake-up events routed to the PS2_WK_0A, PS2_WK_0B, PS2_WK_1A, PS2_WK_1B, PS2_WK_2 bits in the GIRQ19 Source Register

APPLICATION NOTE: The pin control registers for a PS2 wakeup event should be programmed to Input, Falling Edge Triggered, non-inverted polarity detection.

© 2012-2018 Microchip Technology Inc.

37.9 Registers

There are three block instances defined in this chapter: PS/2[0,1,2]. The pin signals are defined in the Pin Configuration chapter. PS/2 ports ending with signal functions ending with “A” or “B” are muxed to a single controller. Only one Port set of clock and data are intended to be used at a time (either “A” or “B” not both.) The unused port segment should have its associated Pin Control Register’s Mux Control Field programmed away from the PS2 controller. (See Section 24.0, "GPIO Interface," on page 388).

Each instance of the PS/2 Device Interface has its Base Address as indicated in Table 37-4.

TABLE 37-4: PS/2 Device Interface BASE ADDRESS TABLE

Instance Name	Instance Number	LDN from (Table 4-2 on page 59)	AHB Base Address
PS/2	**0**	22h	F0_8800h
PS/2	**1**	22h	F0_8880h
PS/2	**2**	22h	F0_8900h

Table 37-5 is a register summary for each instance of the PS/2 Device Interface.

TABLE 37-5: PS/2 Device Interface REGISTER SUMMARY

	EC Interface			
Register Name	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
PS/2 Transmit Buffer Register	00h	0	W	
PS/2 Receive Buffer Register	00h	0	R	
PS/2 Control Register	04h	0	R/W	
PS/2 Status Register	08h	0	R/WC	

37.10 Detailed Description of Accessible Registers

37.10.1 PS/2 TRANSMIT BUFFER

TABLE 37-6: PS/2 TRANSMIT BUFFER REGISTER

HOST ADDRESS	NA						**HOST SIZE**	
EC OFFSET	00h					8-bit	**EC SIZE**	
POWER	VTR					00h	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	--	-	-	-	-
EC TYPE	W	W	W	W	W	W	W	W
BIT NAME	Transmit Data[7:0]							

TRANSMIT DATA

The byte written to this register, when PS2_T/R and PS2_EN in the PS/2 Control Register and XMIT_IDLE in the PS/2 Status Register are set, is transmitted automatically by the PS/2 channel control logic. If any of these three bits (PS2_T/R, PS2_EN, and XMIT_IDLE) are not set, then writes to this register are ignored. On successful completion of this transmission or upon a Transmit Time-out condition, the PS2_T/R bit is automatically cleared and the XMIT_IDLE bit is automatically set. The PS2_T/R bit must be written to a ‘1’ before initiating another transmission to the remote device.

Note:

- Even if PS2_T/R, PS2_EN, and XMIT_IDLE are all set, writing the Transmit Register will not start a transmission if RDATA_RDY in the PS/2 Status Register is set. The automatic PS/2 logic forces data to be read from the Receive Register before allowing a transmission.
- An interrupt is generated on the low to high transition of XMIT_IDLE.
- All bits of this register are write only.

37.10.2 PS/2 RECEIVE BUFFER

TABLE 37-7: PS/2 RECEIVE BUFFER REGISTER

HOST ADDRESS	NA						**HOST SIZE**	
EC OFFSET	00h					8-bit	**EC SIZE**	
POWER	VTR					FFh	nSYS_RST **DEFAULT**	
BUS	EC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	--	-	-	-	-
EC TYPE	R							
BIT NAME	Receive Data[7:0]							

RECEIVE DATA

When PS2_EN=1 and PS2_T/R=0 in the PS/2 Control Register, the PS2 Channel is configured to automatically receive data on that channel (both the CLK and DATA lines will float waiting for the peripheral to initiate a reception by sending a start bit followed by the data bits). After a successful reception, data is placed in this register and the RDATA_RDY bit in the PS/2 Status Register is set and the CLK line is forced low by the PS2 channel logic. RDATA_RDY is cleared and the CLK line is released to high-z following a read of this register. This automatically holds off further receive transfers until the EC has had a chance to get the data.

Note:

- The Receive Register is initialized to FFh after a read or after a Time-out has occurred.
- The channel can be enabled to automatically transmit data (PS2_EN=1) by setting PS2_T/R while RDATA_RDY is set, however a transmission can not be kicked off until the data has been read from the Receive Register.
- An interrupt is generated on the low to high transition of RDATA_RDY.
- If a receive time-out (REC_TIMEOUT=1 in the PS/2 Control Register) or a transmit time-out (XMIT_TIMEOUT=1 in the PS/2 Control Register) occurs the channel is busied (CLK held low) for 300us (Hold Time) to ensure that the peripheral aborts. Writing to the Transmit Register will be allowed, however the data written will not be transmitted until the Hold Time expires.
- All bits in this register are read only

Note 37-1 In receive mode the RX_BUSY bit for a particular channel is set in the PS/2 Status Register.

© 2012-2018 Microchip Technology Inc.

37.10.3 PS/2 CONTROL

There are three PS/2 Control Registers, one for each channel.

TABLE 37-8: PS/2 CONTROL REGISTER

HOST ADDRESS	NA						**HOST SIZE**	
EC OFFSET	04h						8-bit	**EC SIZE**
POWER	VTR						00h	nSYS_RST **DEFAULT**
BUS	EC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Reserved		STOP		PARITY		PS2_EN	PS2_T/R

PS2_T/R

PS/2 Channel Transmit/Receive (default = 0). Configures the PS2 logic for automatic transmission when set or reception when cleared. This bit is only valid when PS2_EN is set.

When set the PS/2 channel is enabled to transmit data. To properly initiate a transmit operation, this bit must be set prior to writing to the Transmit Register. Writes to the Transmit Register are blocked when this bit is cleared. Upon setting the PS2_T/R bit, the channel will drive its CLK line low and then float the DATA line and hold this state until a write occurs to the Transmit Register or until the PS2_T/R bit is cleared. Writing to the Transmit Register initiates the transmit operation. MEC1632 drives the data line low and, within 80ns, floats the clock line (externally pulled high by the pull-up resistor) to signal to the external PS/2 device that data is now available. The PS2_T/R bit is cleared on the 11th clock edge of the transmission or if a Transmit Time-out error condition occurs.

Note: If the PS2_T/R bit is set while the channel is actively receiving data prior to the leading edge of the 10th (parity bit) clock edge, the receive data is discarded. If this bit is not set prior to the 10th clock signal, then the receive data is saved in the Receive Register.

When the PS2_T/R bit is cleared, the PS/2 channel is enabled to receive data. Upon clearing this bit, if RDATA_RDY is also cleared, the channel's CLK and DATA will float waiting for the external PS/2 device to signal the start of a transmission. If the PS2_T/R bit is set while RDATA_RDY is set, then the channel's DATA line will float but its CLK line will be held low, holding off the peripheral, until the Receive Register is read.

PS2_EN

PS2 Channel ENable (default = 0). When PS2_EN is set, the PS/2 State machine is enabled allowing the channel to perform automatic reception or transmission depending on the bit value of PS2_T/R. When PS2_EN is cleared, the channel's automatic PS/2 state machine is disabled and the PS/2 channel's CLK pin driven low and DATA pin not driven.

Note: If the PS2_EN bit is cleared prior to the leading edge (falling edge) of the 10th (parity bit) clock edge the receive data is discarded (RDATA_RDY remains low). If the PS2_EN bit is cleared following the leading edge of the 10th clock signal, then the receive data is saved in the Receive Register (RDATA_RDY goes high) assuming no parity error.

PARITY

These bits are used to set the parity expected by the PS/2 channel state machine. These bits are therefore only valid when PS2_EN is set.

00=Receiver expects Odd Parity (default).

01=Receiver expects Even Parity.

10=Receiver ignores level of the parity bit (10th bit is not interpreted as a parity bit).

11=Reserved

STOP

These bits are used to set the level of the stop bit expected by the PS/2 channel state machine. These bits are therefore only valid when PS2_EN is set.

00=Receiver expects an active high stop bit.

01=Receiver expects an active low stop bit.

10=Receiver ignores the level of the Stop bit (11th bit is not interpreted as a stop bit).

11=Reserved.

APPLICATION NOTE: Changing values in the control register at a rate faster than 2 MHz, may result in unpredictable behavior.

This register should be read to determine the status of PS2_T/R and PS2_EN prior to clearing by writing a 1 to that bit.

37.10.4 PS/2 STATUS

TABLE 37-9: PS/2 STATUS REGISTER

HOST ADDRESS	NA						**HOST SIZE**	
EC OFFSET	08h						8-bit	**EC SIZE**
POWER	VTR						10h	nSYS_RST **DEFAULT**
BUS	EC SPB							
BYTE0 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/WC	R	R/WC	R	R/WC	R/WC	R/WC	R
BIT NAME	XMIT_ START_ TIMEOUT	RX_ BUSY	XMIT_ TIME-OUT	XMIT_ IDLE	FE	PE	REC_ TIME-OUT	RDATA_R DY

APPLICATION NOTE: This register should be read to determine the status of R/WC bits prior to clearing by writing a 1 to that bit.

RDATA_RDY

Receive Data Ready: Under normal operating conditions, this bit is set following the falling edge of the 11th clock given successful reception of a data byte from the PS/2 peripheral (i.e., no parity, framing, or receive time-out errors) and indicates that the received data byte is available to be read from the Receive Register. This bit may also be set in the event that the PS2_EN bit is cleared following the 10th CLK edge. Reading the Receive Register clears this bit.

Note: An Interrupt is generated on the low-to-high transition of the RDATA_RDY bit.

REC_TIMEOUT

Following assertion of the REC_TIMEOUT bit, the channel's CLK line is automatically pulled low for a minimum of 300us until the PS/2 status register is read. Under PS2 automatic operation, PS2_EN is set, this bit is set on one of three receive error conditions:

- When the receiver bit time (time between falling edges) exceeds 300us.
- If the time from the first bit (start) to the 10th bit (parity) exceeds 2ms.
- On a receive parity error along with the Parity Error (PE) bit.
- On a receive framing error due to an incorrect STOP bit along with the framing error (FE) bit.
- The REC_TIMEOUT bit is cleared when the Status Register is read.

Note: An Interrupt is generated on the low-to-high transition of the REC_TIMEOUT bit.

© 2012-2018 Microchip Technology Inc.

PE

Parity Error: When receiving data, the parity bit is clocked in on the falling edge of the 10th CLK edge. If the channel is configured to expect either even or odd parity and the 10th bit is contrary to the expected parity, then the PE and REC_-TIMEOUT bits are set following the falling edge of the 10th CLK edge and an interrupt is generated.

FE

Framing Error: When receiving data, the stop bit is clocked in on the falling edge of the 11th CLK edge. If the channel is configured to expect either a high or low stop bit and the 11th bit is contrary to the expected stop polarity, then the FE and REC_TIMEOUT bits are set following the falling edge of the 11th CLK edge and an interrupt is generated.

XMIT_IDLE

Transmitter Idle: When low, the XMIT_IDLE bit is a status bit indicating that the PS/2 channel is actively transmitting data to the PS2 peripheral device. Writing to the Transmit Register when the channel is ready to transmit will cause the XMIT_IDLE bit to clear and remain clear until one of the following conditions occur: the falling edge of the 11th CLK, XMIT_TIMEOUT is set; the PS2_T/R bit is cleared or the PS2_EN bit is cleared.

Note 37-2 An interrupt is generated on the low-to-high transition of XMIT_IDLE.

XMIT_TIMEOUT

When the XMIT_TIMEOUT bit is set, the PS2_T/R bit is held clear, the PS/2 channel's CLK line is pulled low for a minimum of 300us until the PS/2 Status register is read. The XMIT_TIMEOUT bit is set on one of three transmit conditions: when the transmitter bit time (time between falling edges) exceeds 300us, when the transmitter start bit is not received within 25ms from signaling a transmit start event or if the time from the first bit (start) to the 10th bit (parity) exceeds 2ms.

RX_BUSY

When a RX_BUSY bit is set, the associated channel is actively receiving PS/2 data; when a RX_BUSY bit is clear, the channel is idle. See Note 37-1 on page 534.

XMIT_START_TIMEOUT

When the XMIT_START_TIMEOUT bit is set, a start bit was not received within 25 ms following the transmit start event. Writing a '1' to the bit clears the XMIT_START_TIMEOUT bit. The XMIT_START_TIMEOUT bit is a 'sticky' bit and is intended to uniquely indicate the status of the transmit start bit time-out condition. These bit affect no other logic. Note that the transmit start bit time-out condition is also indicated by the XMIT_TIMEOUT bit.

PROGRAMMER'S NOTE: Always check that an EC PS/2 channel is idle, i.e. the RX_BUSY bit is clear, before attempting to transmit on that channel. Receive data may be lost by setting an EC PS/2 channel to transmit while the RX_BUSY bit is set depending where in the message frame the transmit mode change occurs.

37.11 Power Management

TABLE 37-10: PS/2 Device Interface Power Management

PS2_EN Bit	SLEEP_EN	Block Idle Status (Note 37-3)	CLK_REQ	State	Description
0	X	X	0	SLEEPING	The block is disabled and the clock can be stopped.
1	0	NOT IDLE	1	NORMAL OPERATION	The block is not idle and neither disabled by firmware nor commanded to sleep.
	1	NOT IDLE	1	PREPARING TO SLEEP	The block is commanded to sleep, but the clock is required until the Block is idle.
	1	IDLE	0	SLEEPING	The block is commanded to sleep and idle. The clock can be stopped.

© 2012-2018 Microchip Technology Inc.

Note 37-3 The PS/2 Device Interface ‘idle’ state is determined using the RX_BUSYand XMIT_IDLE bits as shown in Table 37-11.

TABLE 37-11: PS/2 IDLE STATUS

RX_BUSY	XMIT_IDLE	Status
0	1	IDLE
1	X	NOT IDLE
X	0	

© 2012-2018 Microchip Technology Inc.

38.0 KEYBOARD MATRIX SCAN SUPPORT

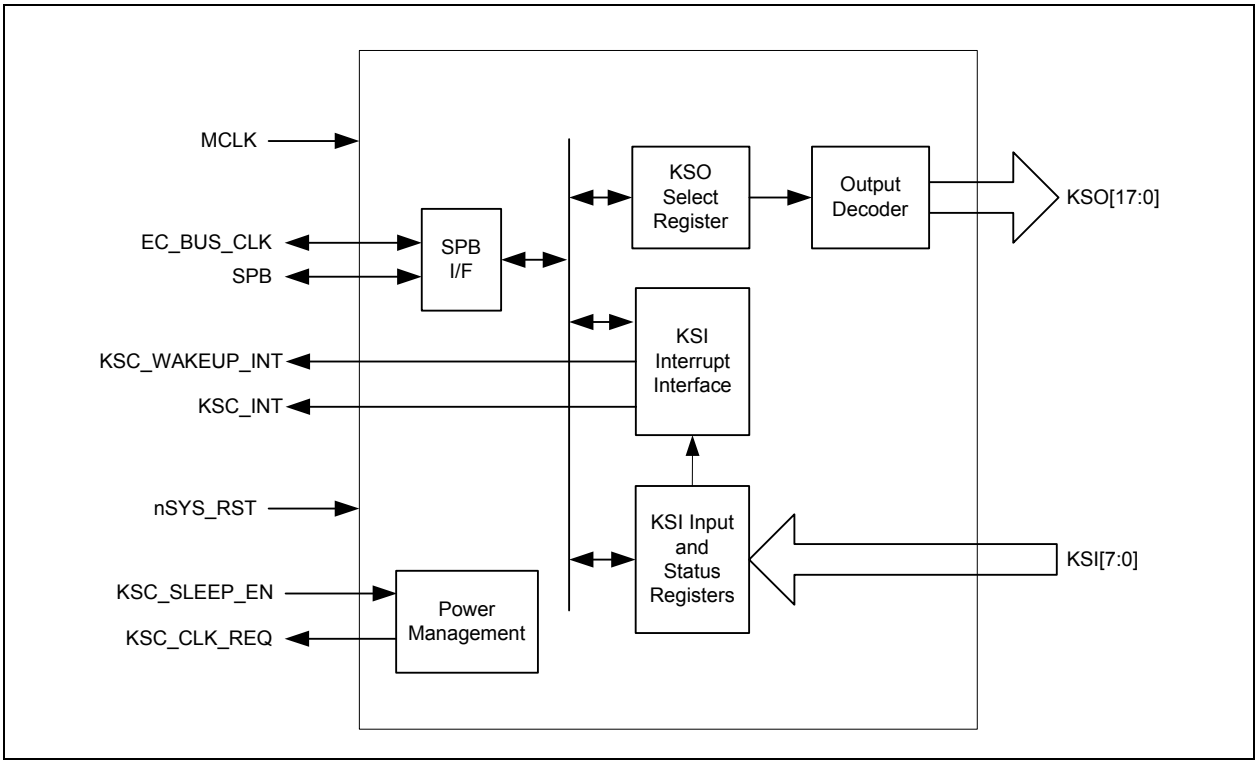
38.1 Overview

The Keyboard Scan Interface block provides a register interface to the EC to directly scan an external keyboard matrix of size up to 18x8. This block is attached to the EC SPB bus as EC Logical Device 8h.

APPLICATION NOTE: 18x8 is the maximum configuration. For smaller matrix size, firmware does not configure GPIO mux control of unused KSOs and masks out unused KSIs and associated interrupts.

38.2 Block Diagram

FIGURE 38-1: Keyboard Matrix Scan Support Block Diagram



38.3 Port List

TABLE 38-1: KEYBOARD SCAN INTERFACE SIGNAL LIST

Signal Name	Direction	Description
SPB	I/O Bus	EC MEC1632 peripheral bus
MCLK	INPUT	Master MEC1632 clock
EC_BUS_CLK_EN	INPUT	SPB Bus Clock
nSYS_RST	INPUT	Block reset signal
KSI[7:0]	INPUT	Column inputs from external keyboard matrix. See Section 38.6
KSO[17:0]	OUTPUT	Row outputs to external keyboard matrix. See Section 38.6
KSC_INT	OUTPUT	Interrupt request to the Interrupt Aggregator's interrupt interface
KSC_WAKEUP_INT	OUTPUT	Wake-up request to the Interrupt Aggregator's wake-up interface
KSC_SLEEP_EN	INPUT	Sleep enable. See Section 38.5.
KSC_CLK_REQ	OUTPUT	Clock request. See Section 38.5.

38.4 Power, Clocks, and Resets

38.4.1 POWER DOMAIN

This block is powered by the VTR power supply.

38.4.2 CLOCKS

The block uses the EC_BUS_CLK_EN and the MCLK

38.4.3 RESET

The block is reset on assertion of nSYS_RST.

38.5 Power Management

Keyboard Scan Interface power management operation is shown in Table 38-2.

TABLE 38-2: KEY SCAN INTERFACE POWER MANAGEMENT

KSEN (Note 38-1)	External SLEEP_EN Input (Note 38-2)	Scan Active (Note 38-3)	Core Clock Required Status Output (Note 38-2)	Mode	Power	Description
1	X	X	0	DISABLED	MINIMUM	The Keyboard Scan Interface is disabled by the EC and the core clock is gated ‘off’ internally.
0	X	YES	1	FULL POWER	MAXIMUM	FULL POWER mode identifies that the Keyboard Scan Interface is enabled and clocks are required to complete a scan related operation.
		NO	0	SLEEPING	MINIMUM	In SLEEPING mode, there will be no clocks to the Keyboard Scan Interface except during access to the registers.

Note 38-1 the KSEN bit is in the KSO Select Register in LPC Configuration Space. When the KSEN bit is ‘1’, there will be no clocks active in the Keyboard Scan Interface except during access to the registers.

Note 38-2 the external sleep enable input and clock required status output are part of the chip-level clocking interface. Note that the external SLEEP_EN input has no affect on the clock required status output.

Note 38-3 the SCAN ACTIVE mode in Table 38-2 depends upon access to the Keyboard Scan Interface registers and scan-related operations involving transitions on the KSO pins.

© 2012-2018 Microchip Technology Inc.

38.6 Pins and I/O Buffers

26 pins are connected to the keyboard matrix. All are multiplexed with GPIOs and are configured using GPIO pin control registers.

Row outputs: KSO[17:0] - 18 tri-state open-drain outputs; output well powered by VTR. When the block is disabled (KSEN = 1), KSO output buffers are tri-stated.

Column inputs: KSI[7:0] - 8 Schmitt trigger inputs with internal pull up; input well powered by VTR.

38.6.1 CONFIGURATION OF IO BUFFERS (INFORMATIONAL)

For KSO[17:0],

GPIO mux control bits are set to select KSO function.

Buffer type = open drain (KSOs from the Key Scan block can be used as output buffer enables; buffers are disabled if the block is disabled, i.e., when KSEN = 1).

Pull-up (if no external pull ups) or none (if external pull ups)

For KSI[7:0],

IO buffers can be configured either as signal function (KSI) or GPIO inputs. When the latter is selected, GPIO interrupts can be used in place of the block's interrupts.

38.6.2 PRE DRIVE MODE

There is an optional Pre Drive Mode that can be enabled to actively drive the KSO pins high before switching to open-drain operation. The active Pre Drive Mode timing is shown in Table 38-3.

TABLE 38-3: ACTIVE PRE DRIVE MODE TIMING

	Parameter	Symbol	Value			Units	Notes
			MIN	TYP	MAX		
1.	Active Pre Drive Mode	$t_{PREDRIVE}$	96.2	98.7	101.1	ns	

The PREDRIVE ENABLE bit in the Keyscan Extended Control Register is used to support the PREDRIVE option.

There are specific Key Scan Interface Pin Configuration Requirements to support the PREDRIVE option.

38.6.2.1 Pin Configuration Requirements

When the PREDRIVE ENABLE bit in the Keyscan Extended Control Register is not asserted ('0' default), the KSO pins must be configured as open-drain drivers using the GPIO Pin Control Registers.

When the PREDRIVE ENABLE bit in the Keyscan Extended Control Register is asserted ('1'), the KSO pins must be configured as push-pull drivers using the GPIO Pin Control Registers.

38.6.2.2 Predrive Mode Programming

The following precautions should be taken to prevent output pad damage during Predrive Mode Programming.

38.6.2.3 Asserting PREDRIVE ENABLE

1. Disable Key Scan Interface (KSEN = '1')
2. Enable Predrive function (PREDRIVE ENABLE = '1')
3. Program buffer type for all KSO pins to "push-pull"
4. Enable Keyscan Interface (KSEN ='0')

38.6.2.4 De-asserting PREDRIVE ENABLE

1. Disable Key Scan Interface (KSEN = '1')
2. Program buffer type for all KSO pins to "open-drain"
3. Disable Predrive function (PREDRIVE ENABLE = '0')
4. Enable Keyscan Interface (KSEN ='0')

© 2012-2018 Microchip Technology Inc.

38.7 Operation (Informational)

During scanning the firmware sequentially drives low one of the rows (KSO[17:0]) and then reads the column data line (KSI[7:0]). A key press is detected as a zero in the corresponding position in the matrix. Keys that are pressed are debounced by firmware. Once confirmed, the corresponding keycode is loaded into host data read buffer in the 8042 Host Interface module. Firmware may need to buffer keycodes in memory in case this interface is stalled or the host requests a Resend.

38.7.1 INTERRUPT GENERATION

To support interrupt-based processing, interrupt can optionally be generated on the high-to-low transition on any of the KSI inputs. Interrupts are to be registered without a running clock.

38.7.1.1 Runtime interrupt

KSC_INT output port is the block's runtime active-high level interrupt. It is connected to the interrupt interface of the Interrupt Aggregator, which then relays interrupts to the EC.

Associated with each KSI input are a status register bit and an interrupt enable register bit. A status bit is set when the associated KSI input goes from high to low. If the interrupt enable bit for that input is set, an interrupt is generated. Interrupt is de-asserted when the status bit and/or interrupt enable bit is clear. A status bit cleared by writing '1' to it.

Interrupts from individual KSIs are logically ORed together to drive the KSC_INT output port. Once asserted, interrupt is not asserted again until either all KSI[7:0] have returned high or the KSO Driver Select[4:0] has changed.

38.7.1.2 Wake-up interrupt

KSC_WAKEUP_INT is the block's wakeup interrupt. It is routed to of the Interrupt Aggregator, KEYSCAN bit of the GIRQ18 Source Register.

During sleep mode, i.e., when the bus clock is stopped, a high-to-low transition on any KSI whose interrupt enable bit is set causes the KSC_WAKEUP_INT to be asserted. Also set is the associated status bit in the EC Blocks Clock Required Status Register 2. KSC_WAKEUP_INT remains active until the bus clock is started.

The aforementioned transition on KSI also sets corresponding status bit in the KSI Status Register. If enabled, a runtime interrupt is also asserted on KSC_INT when the bus clock resumes running.

38.7.2 WAKE PROGRAMMING

Using the Keyboard Scan Interface to 'wake' the MEC1632 can be accomplished using either the Keyboard Scan Interface wake interrupt, or using the wake capabilities of the GPIO Interface pins that are multiplexed with the Keyboard Scan Interface pins.

Using the GPIO Interface for this purpose is the most complicated of the two methods because a minimum of seven pin control registers, and an interrupt enable register must be programmed for 'wake' functionality before the system can sleep. Enabling the Keyboard Scan Interface wake interrupt requires only a single interrupt enable access and is recommended over using the GPIO Interface for this purpose.

© 2012-2018 Microchip Technology Inc.

38.8 Registers

38.8.1 REGISTERS SUMMARY

The Keyboard Scan Interface is assigned EC Logical Device Number 8h, which has AHB base address F0_2000h.

TABLE 38-4: KSC REGISTER SUMMARY

Base Address: F0_2000h	EC Interface			
Register Name	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
Reserved	0h	3-0	R	
KSO Select Register	4h	0	R/W	
KSI Input Register	8h	0	R	
KSI Status Register	Ch	0	R/WC	
KSI Interrupt Enable Register	10h	0	R/W	
Keyscan Extended Control Register	14h	0	R/W	

38.8.2 KSO SELECT REGISTER

TABLE 38-5: KSO SELECT REGISTER

HOST OFFSET	**N/A**			N/A			**HOST SIZE**	
EC OFFSET	**4h (R/W)**				32-bit		**EC SIZE**	
POWER	VTR			0...040h			**VTR POR DEFAULT**	
	EC SPB							
BIT	**D31**	**D30**	**D29**	**.....**		**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	-	-	-	-	-	-	-	-
BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	KSO INVERT	KSEN	KSO ALL	KSO Driver Select[4:0]				

KSO INVERT

KSO INVERT = 1 inverts KSO[17:0]. When KSO INVERT = 0 KSO[17:0] operate normally See Table 38-7, “Keyboard Scan Out Control Summary,” on page 544.

KSEN

KSEN = 1 disables keyboard drives. See first row in Table 38-7. KSEN = 0 enables keyboard scan

KSO ALL

KSO ALL = 1, drives all KSO lines according to KSO INVERT bit. See Table 38-7, “Keyboard Scan Out Control Summary,” on page 544.

KSO DRIVER SELECT[4:0]

KSO Driver Select[4:0] controls the corresponding KSO line (00000b = KSO[0] etc.) according to KSO INVERT. See Table 38-6, "KSO Select Decode"

TABLE 38-6: KSO SELECT DECODE

KSO Select [4:0]	KSO Selected
00h	KSO00
01h	KSO01
02h	KSO02
03h	KSO03
04h	KSO04
05h	KSO05
06h	KSO06
07h	KSO07
08h	KSO08
09h	KSO09
0Ah	KSO10
0Bh	KSO11
0Ch	KSO12
0Dh	KSO13
0Eh	KSO14
0Fh	KSO15
10h	KSO16
11h	KSO17

TABLE 38-7: KEYBOARD SCAN OUT CONTROL SUMMARY

D7 KSO Invert	D6 KSEN	D5 KSO All	D[4:0] KSO Drivers Address	Description
X	1	x	x	Keyboard Scan disabled. KSO[17:0] output buffers disabled.
0	0	0	10001b-00000b	KSO[Drive Selected] driven low. All others driven high
1	0	0	10001b-00000b	KSO[Drive Selected] driven high. All others driven low
0	0	0	11111b-10010b	ALL KSO's driven high
1	0	0	11111b-10010b	All KSO's driven low
0	0	1	x	KSO[17:0] driven low
1	0	1	x	KSO[17:0] driven high

© 2012-2018 Microchip Technology Inc.

38.8.3 KSI INPUT REGISTER

TABLE 38-8: KSI INPUT REGISTER

BUS OFFSET	**N/A**				N/A		**HOST SIZE**	
EC OFFSET	**8h (R)**				32-bit		**EC SIZE**	
POWER	VTR				0...00h		**VTR POR DEFAULT**	
	EC SPB							
BIT	**D31**	**D30**	**D29**	**.....**		**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	-	-	-	-	-	-	-	-
BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	KS7	KS6	KS5	KS4	KS3	KS2	KS1	KS0

38.8.4 KSI STATUS REGISTER

TABLE 38-9: KSI STATUS REGISTER

HOST BUS OFFSET	**N/A**				N/A		**HOST SIZE**	
EC OFFSET	**Ch (R/WC)**				32-bit		**HOST SIZE**	
POWER	VTR				0...00h		**VTR POR DEFAULT**	
	EC SPB							
BIT	**D31**	**D30**	**D29**	**.....**		**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	-	-	-	-	-	-	-	-
BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
BIT NAME	KSI[7] STAT	KSI[6] STAT	KSI[5] STAT	KSI[4] STAT	KSI[3] STAT	KSI[2] STAT	KSI[1] STAT	KSI[0] STAT

38.8.4.1 Bit Definition

KSI[x] STAT bit is set on the falling edge of the corresponding KSI input. Writing a 1 to a bit will clear it.

A KSI interrupt is generated when its corresponding status bit and interrupt enable bit are both set. KSI interrupts are logically ORed together to produce KSC_INT and KSC_WAKEUP_INT.

38.8.5 KSI INTERRUPT ENABLE REGISTER

TABLE 38-10: KSI INTERRUPT ENABLE REGISTER

HOST BUS OFFSET	**N/A**				N/A		**HOST SIZE**	
EC OFFSET	**10h (R/W)**				32-bit		**HOST SIZE**	
POWER	VTR				0...00h		**VTR POR DEFAULT**	
	EC SPB							
BIT	**D31**	**D30**	**D29**	**.....**		**D10**	**D9**	**D8**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R	R	R	R	R	R	R	R
BIT NAME	-	-	-	-	-	-	-	-
BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	KSI[7] IntEn	KSI[6] IntEn	KSI[5] IntEn	KSI[4] IntEn	KSI[3] IntEn	KSI[2] IntEn	KSI[1] IntEn	KSI[0] IntEn

38.8.5.1 Bit Definition

KSI[x] IntEn enables interrupt generation due to high-to-low transition on KSI[x] input. An interrupt is generated only when both KSI[x] IntEn and KSI[x] STAT bits are set.

38.8.6 KEYSCAN EXTENDED CONTROL REGISTER

Offset	**14h**			
Bits	**Description**	**Type**	**Default**	**Reset Event**
32:1	RESERVED	RES	0	–
0	PREDRIVE ENABLE (KSO_PREDRIVE_EN)	RW	0	nSYS_RST

38.8.6.1 Bit Definition

KSO_PREDRIVE_EN bit enables the PREDRIVE mode to actively drive the KSO pins high before switching to open-drain operation (see Table 38-3, “Active Pre Drive Mode Timing,” on page 541). 0= disable, 1=enable.

© 2012-2018 Microchip Technology Inc.

39.0 BC-LINK MASTER

39.1 General Description

The function of this block is to.provide BC-Link to a slave device. The BC-Link protocol includes a start bit to signal the beginning of a message and a turnaround (TAR) period for bus transfer between the Master and Companion devices.

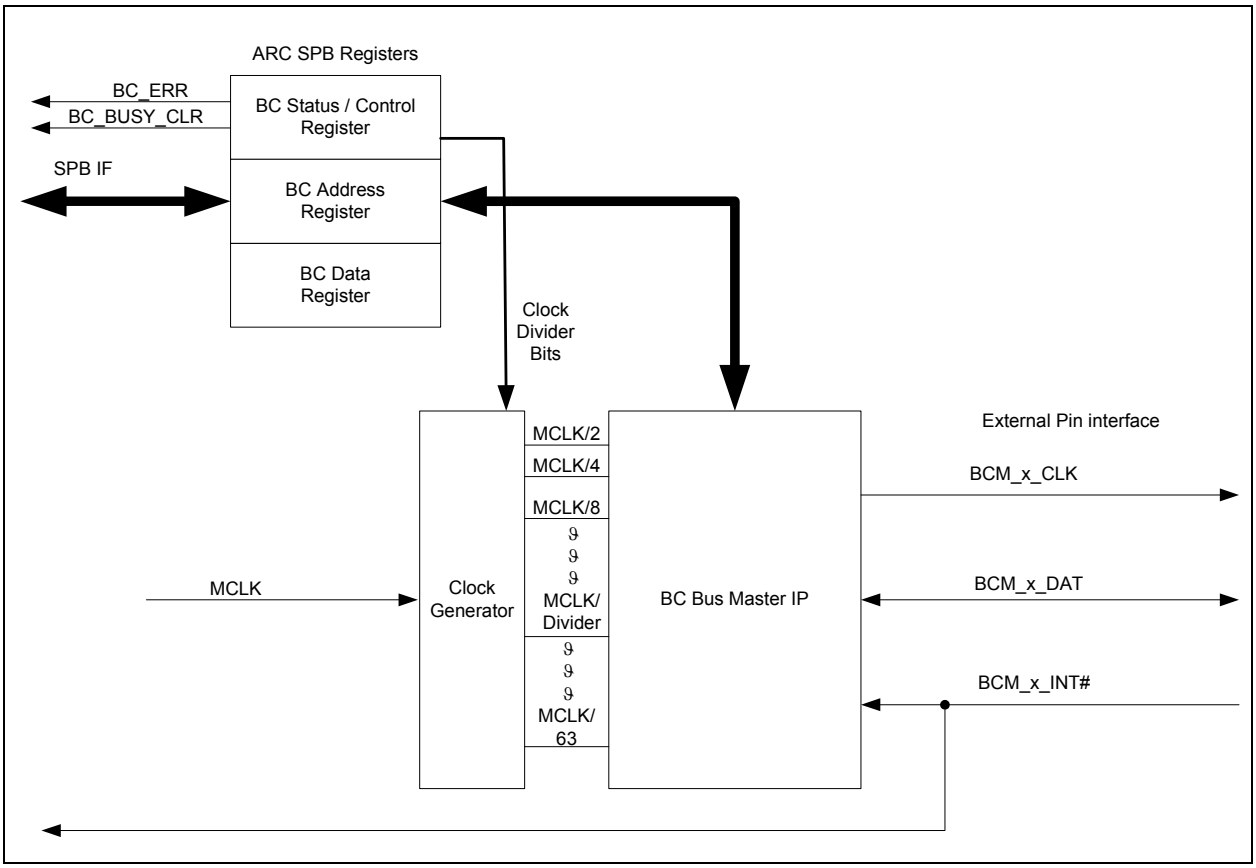
Note:	A weak pull-up resistor is recommended on the data line (100KΩ).

There are three instances of the BC-Link Master interface in the MEC1632. Instance A and B are high speed BC-links with high speed buffers attached. Instance D is a low speed BC-Link which uses 8 mA buffer. Instance D consists of signals LSBCM_D_INT#, LSBCM_D_DAT, LSBCM_D_CLK.

The maximum usable clock frequencies are described in Note 39-1 on page 549.

39.2 Block Diagram

FIGURE 39-1: BC-LINK MASTER BLOCK DIAGRAM



39.3 Signal List

TABLE 39-1: BC-LINK SIGNAL LIST

Signal Name	Direction	Description
BCM_x_CLK	OUTPUT	20.27MHz - 79.18KHz output clock, where x is A, B or D.
BCM_x_DAT	INPUT/OUTPUT	Bidirectional data line, where x is A, B or D.
BCM_x_INT#	INPUT	Input from the companion device, where x is A, B or D.
BC_x_ERR	OUTPUT	BC-Link master error interrupt
BC_x_BUSY_CLR	OUTPUT	BC-Link master Busy Clear interrupt
MCLK	INPUT	20.27MHz Master clock
SPB	I/O Bus	Bus used for register access
CLK_REQ	OUTPUT	Power Management clock required output.

Note: The Low Speed BC-Link Master Block has an identical organization, but the external signals are LSBCM_D_CLK, LSBCM_D_DAT and LSBCM_D_INT#.

39.4 Power, Clocks and Reset

39.4.1 POWER DOMAIN

This block is powered by the VTR Power Supply.

See Section 7.5, "Power Configuration," on page 121 for details on power domains.

39.4.2 CLOCKS

This block uses the EC Bus Clock and the MCLK. The EC Bus Clock is used to access the Registers described in this block. MCLK is divided down to generate the external bus clock.

See Section 7.4, "Clock Generator," on page 98 for details on clocks.

39.4.3 POWER ON RESET

This block is reset on a nSYS_RST. On reset, the BC-Link state machine transitions to the Idle state and waits for the address and data registers to be written.

See Section 7.6, "Reset Interface," on page 124 for details on reset.

39.5 Interrupts

Each BC-Link Master instance has three interrupts events: BC_BUSY_CLR, BC_ERR, & BC_INT#. The BC-Link Master BC_BUSY_CLR and BC_ERR interrupts are generated by changes in the BC-Link Status Register. The BC_INT# is an active low level interrupt generated by input pin signal function. The edge detection of the interrupt and wake events are controlled by their associated pin control registers in the Section 24.0, "GPIO Interface," on page 388

The BC-Link Master block Instance A are routed to the BCM_BUSY_CLR[A], BCM_ERR[A], & BCM_INT#[A] bits in the GIRQ21 Source Register on page 321. The BC-Link Master block Instance B are routed to the BCM_BUSY_CLR[B], BCM_ERR[B], & BCM_INT#[B] bits in the GIRQ21 Source Register on page 321. The BC-Link Master block Instance D are routed to the BCM_BUSY_CLR[D], BCM_ERR[D], & BCM_INT#[D] bits in the GIRQ21 Source Register on page 321.

39.5.1 INSTANCE BUFFERS VS CLOCK

There are three instances of the BC-Link Master block implemented in the MEC1632 enumerated as A, B and D. The BC-Link Interface pins are defined in the Pin Configuration chapter for all instances. The base addresses for all instances are listed in Table 39-3, "BC-Link Master Base Address Table".

Instance A and B are high speed BC-links with higher current buffers driving the external signals. Instance D is a low speed BC-Link, which is implemented lower current buffers. Signal names for the high speed interfaces are BCM[A,B]_CLK, BCM[A,B]_DAT and BCM[A,B]_INT#. Signal names for the low speed instance are LSBCM_D_CLK, LSBCM_D_DAT and LSBCM_D_INT#.

© 2012-2018 Microchip Technology Inc.

Note 39-1 For ribbon cable applications, the Low Speed BC-Link Master maximum clock frequency and the High Speed BC-Link Master maximum clock frequency are shown in Table 39-2. The Clock frequency is set with the BC-Link Clock Select Register. See also Note: on page 622.

TABLE 39-2: BC-Link Master PIN INTERFACE

Pin	Description	Buffer	Max Freq	Min Value in BC-Link Clock Select Register
BCM[A,B]_CLK	High Speed Clock supplied by the MEC1632 Device	O16	20.27 Mhz	0
BCM[A,B]_DAT	High Speed Data Line	IO16 Note 39-2		
BCM[A,B]_INT#	Interrupt signal	I		
LSBCM_D_CLK	Low Speed Clock supplied by the MEC1632 Device	O8	2.9 MHz	6
LSBCM_D_DAT	Data Line	IO8 Note 39-2		
LSBCM_D_INT	Interrupt signal	I		

Note 39-2 BCM[A,B]_DAT & LSBCM_D_DAT pins requires a weak pull up (100K).

39.6 Registers

TABLE 39-3: BC-Link Master BASE ADDRESS TABLE

BC-Link InstanceS	LDN from (Table 4-2 on page 59)	AHB Base Address
BC-LINK.A	5h	F0_1400h
BC-LINK.B		F0_1480h = F0_1400h + 80h
BC-Link.D		F0_1580h = F0_1400h + 180h

TABLE 39-4: BC-Link Master REGISTER SUMMARY

	EC Interface			
Register Name	**SPB Offset**	**Byte Lane**	**EC Type**	**Notes**
BC-Link Status Register	00h	0	R/W	
BC-Link Address Register	04h	0	R/W	
BC-Link Data Register	08h	0	R/W	
BC-Link Clock Select Register	0Ch	0	R/W	

39.6.1 BC-LINK STATUS

TABLE 39-5: BC-LINK STATUS REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	00h						8-bit	**EC SIZE**
POWER	VTR						81h	nSYS_RST **DEFAULT**
BUS	EC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-‘	-	-	-	-
EC TYPE	R/W	R/WC	R/W	R/W	R	R	R	R
BIT NAME	RESET	BC_ ERR	BC_ ERR_ INT_EN	BC_ Busy_ CLR_ INT_EN	Reserved			BUSY

BUSY

This bit is asserted to ‘1’ when the BC interface is transferring data and on reset. Otherwise it is cleared to ‘0’. When BUSY bit is cleared by hardware, an interrupt is generated if the BC_Busy_CLR_INT_EN bit is set to ‘1’.

BC_BUSY_CLR_INT_EN

This bit is an enable for generating an interrupt when the BUSY bit is cleared by hardware. When the BC_Busy_CLR_INT_EN bit is set to ‘1’, the interrupt signal is enabled. When the BC_Busy_CLR_INT_EN bit is cleared to ‘0’, the interrupt is disabled. When enabled this interrupt occurs after a BC Bus read or write. Figure 39-2 shows when the interrupt is generated.

BC_ERR_INT_EN

This bit is an enable for generating an interrupt when the BC_ERR bit is set by hardware. When the BC_ERR_INT_EN bit is ‘1’, the interrupt signal is enabled. When the BC_ERR_INT_EN bit is ‘0’,the interrupt is disabled.

BC_ERR

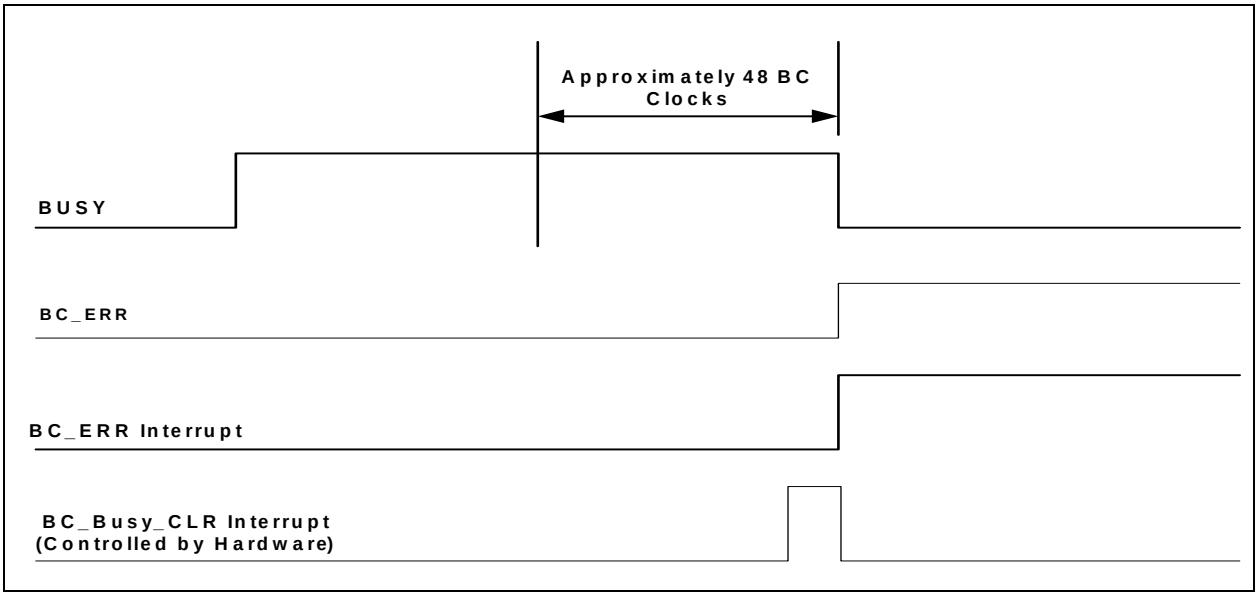
This bit indicates that a BC Bus Error has occurred. If an error occurs the BC_ERR bit is set by hardware when the BUSY bit is cleared. See Figure 39-2 for relative timing. When BC_ERR bit is set by hardware, an interrupt is generated if the BC_ERR_INT_EN bit is set to ‘1’.

This BC_ERR bit is cleared to ‘0’, by software writing a ‘1’ to this bit.

Errors that cause this interrupt are: Bad Data received by the BASE (CRC Error) or a time-out caused by the COMPANION not responding. All COMPANION errors cause the COMPANION to abort the operation and cause the BASE to time-out. Figure 39-2 shows the timing of this interrupt.

© 2012-2018 Microchip Technology Inc.

FIGURE 39-2: BC BUS BC_ERR INTERRUPT TIMING



RESET

When set to '1',the BC-Link Master Interface will be placed in reset and be held in reset until this bit is de-asserted. Reset causes the BUSY bit to be set and will not be cleared until the reset operation of the BC Interface is completed (approximately 48 BC clocks).

PROGRAMMER'S NOTE: The de-assertion of the BUSY bit on reset will not generate an interrupt, even if the BC_Busy_CLR_INT_EN is asserted 1. The BUSY bit must be polled.

39.6.2 BC-LINK ADDRESS

TABLE 39-6: BC-LINK ADDRESS REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	04h						8-bit	**EC SIZE**
POWER	VTR						00h	nSYS_RST **DEFAULT**
BUS	EC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Address[7:0]							

39.6.3 BC-LINK DATA

TABLE 39-7: BC-LINK DATA REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	08h						8-bit	**EC SIZE**
POWER	VTR						00h	nSYS_RST **DEFAULT**
BUS	EC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Data[7:0]							

39.6.4 BC CLOCK SELECT

TABLE 39-8: BC-LINK CLOCK SELECT REGISTER

HOST ADDRESS	N/A						**HOST SIZE**	
EC OFFSET	0Ch						8-bit	**EC SIZE**
POWER	VTR						4	nSYS_RST **DEFAULT**
BUS	EC SPB							
BYTE3 BIT	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
HOST TYPE	-	-	-	-	-	-	-	-
EC TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Divider[7:0]							

DIVIDER

The BC Clock is set to the Master Clock divided by this field, or MCLK/ (Divider[7:0] +1), Where Divider[7:0] is 0 to 255. The clock divider bits can only can be changed when the BC Bus is in soft RESET (when either the Reset bit is set by software or when the BUSY Bit is set by the interface).

Example settings are shown in Table 39-9, "Example Frequency Settings":

TABLE 39-9: EXAMPLE FREQUENCY SETTINGS

Divider	Frequency
0	20.27MHz
1	10.13MHz
2	6.756MHz
3	5.067MHz
4	4.054MHz
15	1.266MHz
21	0.921MHz
2A	0.471MHz
63	0.316MHz

© 2012-2018 Microchip Technology Inc.

39.7 BC-Link Master Operations

Descriptions of the BC-Link read and write operations follows:

39.7.1 READ

The BC-Link Read protocol requires two reads of the BC-Link Data Register. The two reads drive a two state-state machine: the two states are Read#1 and Read#2. The Read#1 of the BC-Link Data Register starts the read protocol on the BC-Link pins and sets the Busy bit in the BC-Link Status Register. The contents of the data read during Read#1 by the EC is stale and is not to be used. After the Busy bit in the BC-Link Status Register autonomously clears to '0', the Read#2 of the BC-Link Data Register transfers the data read from the peripheral/BC-Link companion chip to the EC.

1. Software starts by checking the status of the BUSY bit in the BC-Link Status Register on page 550. If the Busy bit is 0, proceed, if Busy is a 1 Wait.
2. Software writes the address of the register to be read into the BC-Link Address Register on page 551.
3. Software then reads the BC-Link Data Register on page 552. This read returns random data. The read activates the BC-Link Master to transmit the read request packet to the BC-Link companion. When the transfer initiates, the hardware sets the BUSY bit to a 1.
4. The BC-Link companion reads the selected register and transmits the read response packet to theBC-Link Master.

Note 39-3 The companion will ignore the read request if there is a CRC error, this will cause the base to time-out and issue a BC_ERR Interrupt.

5. The BC-Link Master loads the BC-Link Data Register on page 552 issues a BUSY Bit Clear interrupt and clears the busy bit to '0'.
6. Check the BC_ERR bit in the BC-Link Status Register on page 550.
7. Software can now read the BC-Link Data Register on page 552 which contains the valid data if there was no BC Bus error.
8. If a Bus Error occurs issue a soft reset by setting the Reset bit in the BC-Link Status Register on page 550.
9. The read can re-tried once BUSY is cleared.

PROGRAMMER'S NOTE: Steps 3 thorough 7 should be completed as a contiguous sequence. If not the LSBC interface could be presenting incorrect data when software thinks it is accessing a valid register read.

39.7.2 WRITE

1. Software starts by checking the status of the BUSY bit in the BC-Link Status Register on page 550. If the BUSY bit is 0, proceed, if BUSY is a 1 Wait.
2. Software writes the address of the register to be written into the BC-Link Address Register on page 551. Then writes the data to be written into the addressed register in to the BC-Link Data Register on page 552.
3. The write to the BC-Link Data Register on page 552 starts the BC_Link write operation. The BC-Link Master sets the BUSY bit in the BC-Link Status Register on page 550.
4. The BC-Link Master Interface transmits the write request packet.
5. When the write request packet is received by the BC-Link companion, the CRC is checked and data is written to the addressed companion register.
6. The companion sends an ACK if the write is completed.

Note 39-4 A time-out will occur approximately 16 BC-Link clocks after the packet is sent by the BC-Link Master. The BC-Link Master will issue a BC_ERR bit in the BC-Link Status Register on page 550 approximately 48 clocks later. and clear the BUSY bit.

7. The BC-Link Master issues the Busy bit Clear interrupt and clears the BUSY bit after receiving the ACK from the companion
8. If a Bus Error occurs issue a soft reset by setting the Reset bit in the BC-Link Status Register on page 550.
9. The write can re-tried once BUSY is cleared.

39.8 Power Management

BC-Link Master Power Management is illustrated in Table 39-10. Note that the BC-Link Master does not include a sleep enable input.

TABLE 39-10: BC-Link Master Power Management

Block Enable (Note 39-5)	Block Idle Status (Note 39-6)	Clock Required Status Output (CLK_REQ)	State	Description
DISABLED	X	0	DISABLED	Block is disabled by firmware and the core clock is not needed.
ENABLED	NOT IDLE	1	NORMAL OPERATION	The block is enabled and performing a transaction.
	IDLE	0	SLEEPING	The block is enabled and not performing a transaction. The clock may be stopped.

Note 39-5 The DISABLED state in Table 39-10 is defined as the Reset Bit being set to '1' and the & BUSY bit being cleared to'0' in the BC-Link Status Register.

Note 39-6 The IDLE state in Table 39-10 is defined as both the Reset & BUSY Bit being cleared to '0' in the BC-Link Status Register.

© 2012-2018 Microchip Technology Inc.

40.0 PORT 80 BIOS DEBUG PORT

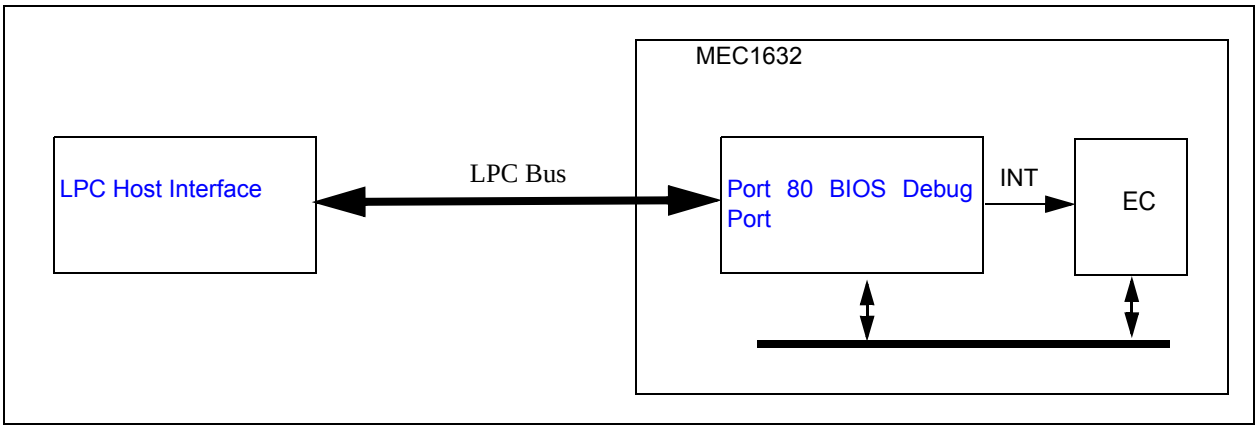
40.1 Overview

The Port 80 BIOS Debug Port emulates the functionality of a “Port 80” ISA plug-in card. In addition, a timestamp for the debug data can be optionally added.

Diagnostic data is written by the LPC Host Interface to the Port 80 BIOS Debug Port, which is located in LPC I/O address space (Figure 40-1). The Port 80 BIOS Debug Port generates Interrupts to the EC when host data is available. The EC reads this data along with the timestamp, if enabled.

The MEC1632 includes two instances of the Port 80 BIOS Debug Port.

FIGURE 40-1: BIOS DEBUG PORT SYSTEM VIEW



40.2 References

No references have been cited for this chapter.

40.3 Terminology

TABLE 40-1: TERMINOLOGY

Term	Definition
–	–

40.4 Interface

40.4.1 HOST INTERFACE

The host communicates to the Port 80 BIOS Debug Port using the LPC Host Interface (see Section 40.6, "LPC Host Interface," on page 557).

40.4.2 EC INTERFACE

The EC communicates to the Port 80 BIOS Debug Port using an internal bus interface and includes Interrupts as defined in Section 40.7, "Interrupts," on page 557.

FIGURE 40-2: POWER, CLOCKS, AND RESETS

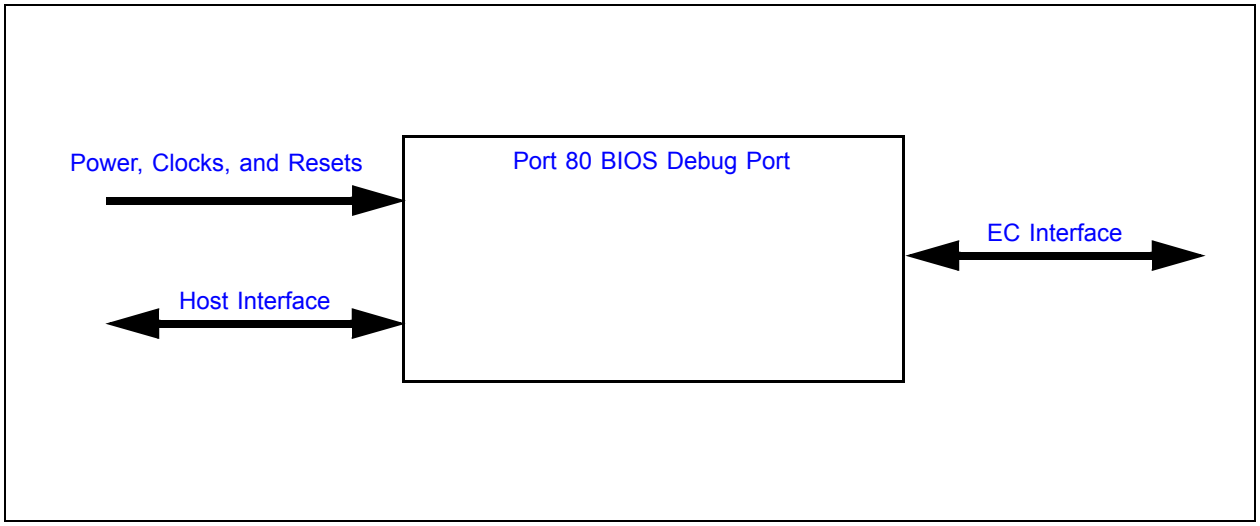


TABLE 40-2: POWER PLANES

Name	Description
VTR	The Port 80 BIOS Debug Port requires only a single power plane.

TABLE 40-3: CLOCKS

Name	Description
MCLK	Port 80 BIOS Debug Port core clock.

TABLE 40-4: RESETS

Name	Description
nSYS_RST	Port 80 BIOS Debug Port hardware reset.

40.5 Functional Description

The Port 80 BIOS Debug Port consists of a 32-bit wide x 16 deep FIFO, 24-bit free running timer, and Power Management Interface (Figure 40-3). Host and EC access to the Port 80 BIOS Debug Port is through Registers as defined in Section 40.9, "Registers," on page 558.

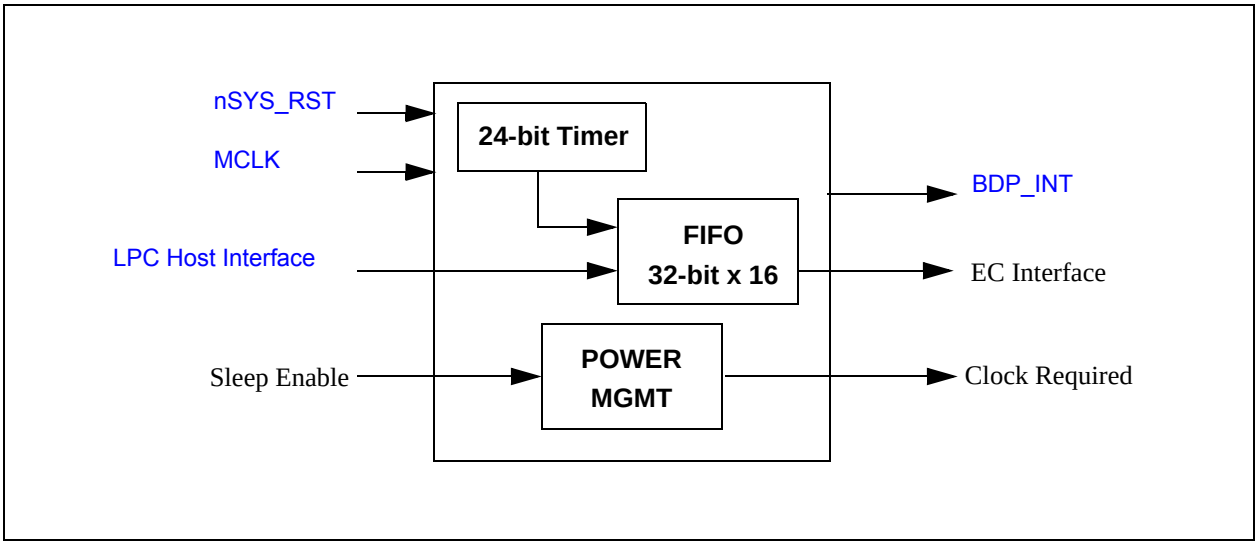
The Port 80 BIOS Debug Port Configuration Register and Count Register are used to control the FIFO and the 24-bit timer, as defined in Section 40.9.4, "Configuration Register," on page 560 and Section 40.9.6, "Count Register," on page 561.

Writes to the Port 80 BIOS Debug Port Host Data Register are concatenated with the 24-bit timestamp and written to the FIFO. Reads of the Host Data Register return zero. If writes to the Host Data Register overrun the FIFO, the oldest data are discarded and the OVERRUN status bit in the Status Register is asserted.

Only the EC can read data from the FIFO, using the EC Data Register. The use of this data is determined by EC Firmware alone.

© 2012-2018 Microchip Technology Inc.

FIGURE 40-3: Port 80 BIOS Debug Port BLOCK DIAGRAM



40.6 LPC Host Interface

Each Port 80 BIOS Debug Port instance is an LPC logical device and occupies one byte in the LPC I/O Address Space (Host Data Register). The LPC Host Interface must initialize each Port 80 BIOS Debug Port base address using the ISA Plug and Play Base Address Registers (BAR), and enable each instance using the Activate Register. The Port 80 BIOS Debug Port can be mapped to any 16-bit LPC I/O address value.

40.7 Interrupts

Note: BDP0_INT and BDP1_INTin the GIRQ15 Source Register are the interrupt source bits for the two instances of the Port 80 BIOS Debug Port in the MEC1632 (see Section 17.0, "EC Interrupt Aggregator," on page 287).

TABLE 40-5: INTERRUPTS

Name	Description
BDP_INT	The Port 80 BIOS Debug Port generates an edge-sensitive interrupt when the amount of data in the FIFO equals or exceeds the FIFO THRESHOLD (see Section 40.9.4, "Configuration Register," on page 560). Note that there is no enable bit for this interrupt in the Port 80 BIOS Debug Port Registers.

40.8 Power Management Interface

Note: BDP0 and BDP1 in the LPC Blocks Clock Required Status Register are the clock required status bits for the two instances of the Port 80 BIOS Debug Port in the MEC1632 (see Section 7.0, "Power, Clocks, and Resets," on page 95).

TABLE 40-6: Port 80 BIOS Debug Port Power Management Interface

Activate (Note 40-1)	External SLEEP_EN Input (Note 40-2)	Timer Active (Note 40-3)	Core Clock Required Status Output (Note 40-2)	Mode	Power	Description
0	X	X	0	DISABLED	MINIMUM	The Port 80 BIOS Debug Port is disabled by the LPC Host and the core clock is gated 'off' internally.
1	X	YES	1	FULL POWER	MAXIMUM	FULL POWER mode identifies that the Port 80 BIOS Debug Port is enabled by the LPC Host and the 24-bit Timer is active.
		NO	0	SLEEPING	MINIMUM	In SLEEPING mode, there will be no clocks to the Port 80 BIOS Debug Port except during access to the Registers.

Note 40-1 the ACTIVATE bit is in the Activate Register in LPC Configuration Space. When the ACTIVATE bit is '0', there will be no clocks active in the Port 80 BIOS Debug Port except during access to the Registers.

Note 40-2 the external sleep enable input and clock required status output are part of the chip-level clocking interface. Note that the external SLEEP_EN input has no affect on the clock required status output.

Note 40-3 the TIMER ACTIVE mode in Table 40-6 depends upon the state of the TIMER ENABLE bit in the Configuration Register.

40.9 Registers

40.9.1 SUMMARY

Each Port 80 BIOS Debug Port instance includes an ACTIVATE bit and Base Address Register (BAR), accessible by the LPC Host Interface in LPC Configuration space.

When the ACTIVATE bit is asserted '1', the Port 80 BIOS Debug Port is enabled. When the ACTIVATE bit is '0', writes by the LPC Host Interface to the Host Data Register are not claimed, the FIFO is flushed, the 24-bit Timer is reset, and the timer clock is stopped (Note 40-5). See also Section 40.8, "Power Management Interface," on page 558.

TABLE 40-7: Port 80 BIOS Debug Port **BASE ADDRESS TABLE**

Port 80 BIOS Debug Port **Instances**	**LDN** (Table 4-2, "Host Logical Devices in the MEC1632," on page 59)	**AHB Base Address**
BIOS Debug Port 0	20h	FF_8000h
BIOS Debug Port 1	21h	FF_8400h

© 2012-2018 Microchip Technology Inc.

TABLE 40-8: Port 80 BIOS Debug Port Registers Summary

Host Offset	EC Interface SPB OFFSET	Register Name	Host Access	EC Access
00h	00h	Host Data Register	YES	YES
–	100h	EC Data Register	NO	YES
–	104h	Configuration Register	NO	YES
–	108h	Status Register	NO	YES
–	10Ch	Count Register	NO	YES

TABLE 40-9: REGISTER BIT ACCESS TYPES

Register Bit Type	Description
R	**Read**: A register or bit with this attribute can be read.
W	**Write**: A register or bit with this attribute can be written.
RO	**Read Only**: A register or bit with this attribute is read only, writes have no effect.
RS	**Read to Set**: A register or bit with this attribute is set on read.
RC	**Read to Clear**: A register or bit with this attribute is cleared after the read, writes have no effect.
WO	**Write Only**: A register or bit with this attribute is write only, reads return zero.
WC	**Write One to Clear**: Writing a one to a bit with this attribute clears (‘0’) the value, writing a zero has no effect.
WS	**Write One to Set**: Writing a one to a bit with this attribute sets the value to ‘1’, writing a zero has no effect.
WZS	**Write Zero to Set**: Writing a zero to a bit with this attribute sets the value to ‘1’, writing a one has no effect.
RES	**Reserved**: Reads of a register or bit with this attribute return zero, writes are ignored.

40.9.2 HOST DATA REGISTER

The LPC Host Interface and the EC writes to the Port 80 BIOS Debug Port FIFO using the Host Data Register (Note 40-4).

Offset	00h			
Bits	**Description**	**Type (Note 40-4)**	**Default**	**Reset Event**
31:8	RESERVED	RES	0	–
7:0	HOST_DATA	WO	0	nSYS_RST

Note 40-4 reads of the Host Data Register return 00h.

40.9.3 EC DATA REGISTER

The EC reads the Port 80 BIOS Debug Port FIFO using the EC Data Register.

Offset	100h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	TIME_STAMP	RO	0	nSYS_RST
7:0	EC_DATA	RO	0	nSYS_RST

40.9.4 CONFIGURATION REGISTER

The EC configures and controls the Port 80 BIOS Debug Port using the Configuration Register.

Offset	104h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:8	RESERVED	RES	0	–
7:6	FIFO THRESHOLD The FIFO THRESHOLD bits determine the threshold for the Port 80 BIOS Debug Port Interrupts, as defined in Table 40-10.	RW	0	nSYS_RST
5	TIMER ENABLE (RUN_TSTAMP) When the TIMER ENABLE bit is '1', the 24-bit Timer is actively counting at a rate determined by the TIMEBASE SELECT bits. When the TIMER ENABLE bit is '0', counting is stopped (Note 40-5).	RW	0	nSYS_RST
4:3	TIMEBASE SELECT (TSTAMP_CLK) The TIMEBASE SELECT bits determine the clock for the 24-bit Timer as defined in Table 40-11 (Note 40-5).	RW	0	nSYS_RST
2	RESET TIMESTAMP (RST_TSTAMP) When the RESET TIMESTAMP bit is '1', the 24-bit Timer is reset to '0'. The RESET TIMESTAMP bit is self-clearing. Writing zero to the Count Register has the same effect.	WO	–	nSYS_RST
1	FLUSH (FLUSH_FIFO) When the FLUSH bit is '1', the FIFO is flushed. The FLUSH bit is self-clearing.	WO	–	nSYS_RST
0	RESERVED	RES	0	–

Note 40-5 the ACTIVATE bit does not affect the TIMER ENABLE bit or the TIMEBASE SELECT bits in the Configuration Register.

TABLE 40-10: FIFO THRESHOLD BITS

Bit 7	Bit 6	FIFO Depth Interrupt Threshold
0	0	1 (default)
0	1	4
1	0	8
1	1	14

TABLE 40-11: TIMEBASE SELECT BITS

Bit 4	Bit 3	24-bit Timer Clock
0	0	MCLK/8
0	1	MCLK/16
1	0	MCLK/32
1	1	MCLK/64

© 2012-2018 Microchip Technology Inc.

40.9.5 STATUS REGISTER

Offset	108h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
31:2	RESERVED	RES	0	–
1	OVERRUN (FIFO_OVERRUN) The OVERRUN bit is '1' when the host writes the Host Data Register when the FIFO is full.	WC	0	nSYS_RST
0	NOT EMPTY (FIFO_NEMPTY) The NOT EMPTY bit is '1' when there is data in the FIFO. The NOT EMPTY bit is '0' when the FIFO is empty.	RO	0	nSYS_RST

40.9.6 COUNT REGISTER

Offset	10Ch			
Bits	**Description**	**Type**	**Default**	**Reset Event**
32:8	COUNT Writes load data into the 24-bit Timer. Reads return the 24-bit Timer current value.	RW	–	–
7:0	RESERVED	RES	0	–

41.0 TRACE FIFO DEBUG PORT (TFDP)

41.1 Introduction

The TFDP serially transmits Embedded Controller (EC)-originated diagnostic vectors to an external debug trace system.

41.2 References

No references have been cited for this chapter.

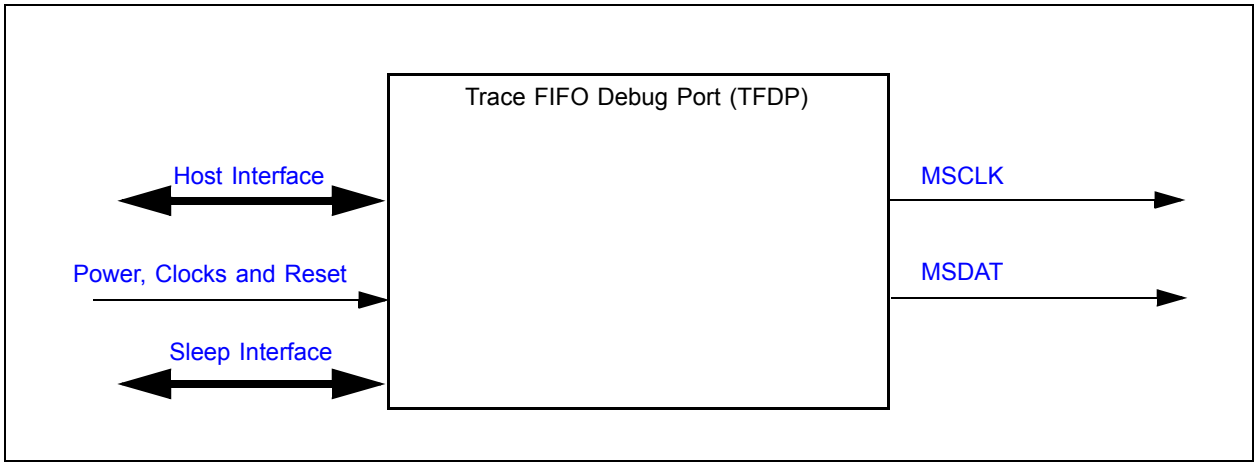
41.3 Terminology

There is no terminology defined for this chapter.

41.4 Interface

This block is designed to be accessed externally via the pin interface and internally via a registered host interface.

FIGURE 41-1: Trace FIFO Debug Port (TFDP) I/O DIAGRAM



41.4.1 SIGNAL DESCRIPTION

The Signal Description Table lists the signals that are typically routed to the pin interface.

TABLE 41-1: SIGNAL DESCRIPTION TABLE

Name	Direction	Description
MSCLK	Output	Derived from EC Bus Clock.
MSDAT	Output	Serialized data shifted out by MSCLK.

41.4.2 HOST INTERFACE

The registers defined for the Trace FIFO Debug Port (TFDP) are accessible by the various hosts as indicated in Section 41.9, "EC-Only Registers".

© 2012-2018 Microchip Technology Inc.

41.4.3 SLEEP INTERFACE

This section defines the Sleep Interface signals that are controlled at the chip-level.

TABLE 41-2: SIGNAL DESCRIPTION TABLE

Name	Direction	Description
SleepEnable	Input	Sleep Enable This input is connected to the TFDP Sleep Enable bit in the Power, Clocks, and Resets block.
ClockRequired	Output	Clock Required by block. This output is connected to the TFDP Clock Required bit in the Power, Clocks, and Resets block.

41.5 Power, Clocks and Reset

This section defines the Power, Clock, and Reset parameters of the block.

41.5.1 POWER DOMAINS

TABLE 41-3: POWER SOURCES

Name	Description
VTR	This power well sources all of the registers and logic in this block.

41.5.2 CLOCK INPUTS

TABLE 41-4: CLOCK INPUTS

Name	Description
EC Bus Clock	This clock input is used to derive the MSCLK.

41.5.3 RESETS

TABLE 41-5: RESET SIGNALS

Name	Description
nSYS_RST	This reset signal resets all of the registers and logic in this block.

41.6 Interrupts

There are no interrupts generated from this block.

41.7 Low Power Modes

Whenever the block is disabled (when its EN bit is set to 0) or when it is in its idle state (not involved in a data transfer) the block deasserts its ClockRequired signal, permitting the system to enter a system sleep state.

Setting the SleepEnable signal to ‘1’ will cause the block to complete its current data transfer (if any) and enter the idle stay. It will stay in the idle state, ignoring any data requests, as long as the SleepEnable is ‘1’.

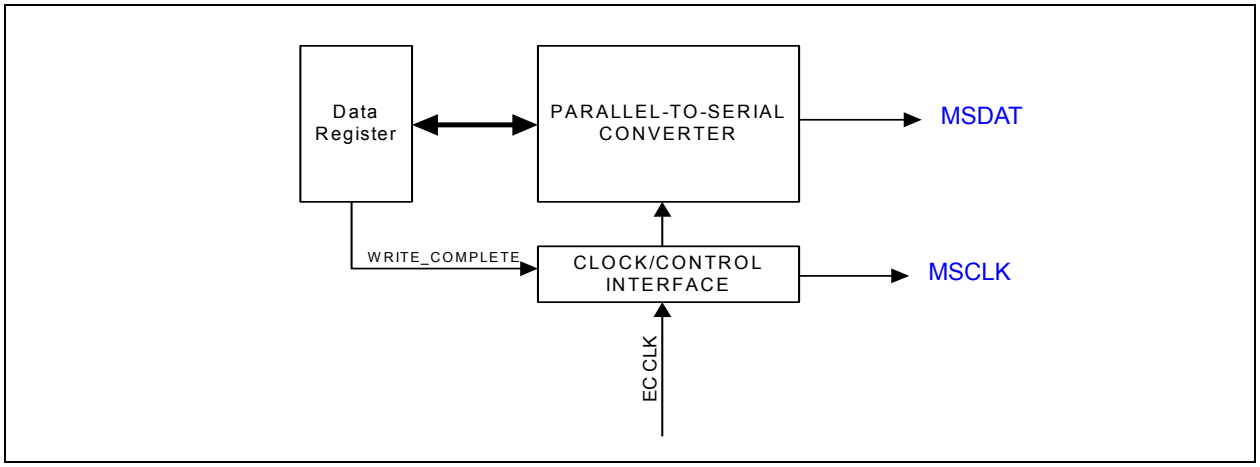
© 2012-2018 Microchip Technology Inc.

41.8 Description

The TFDP is a unidirectional (from processor to external world) two-wire serial, byte-oriented debug interface for use by processor firmware to transmit diagnostic information.

The TFDP consists of the Debug Data Register, Debug Control Register, a Parallel-to-Serial Converter, a Clock/Control Interface and a two-pin external interface (MSCLK, MSDAT). See Figure 41-2.

FIGURE 41-2: BLOCK DIAGRAM OF TFDP DEBUG PORT

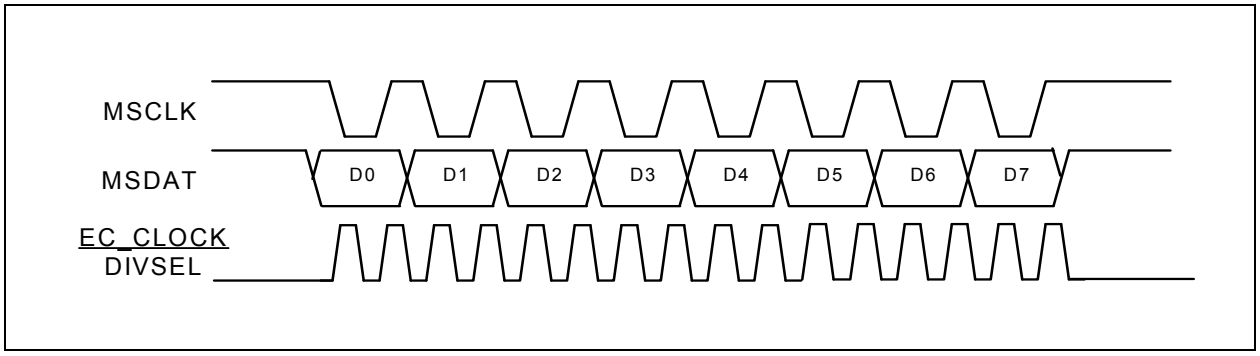


The firmware executing on the embedded controller writes to the Debug Data Register to initiate a transfer cycle (Figure 41-3). At first, data from the Debug Data Register is shifted into the LSB. Afterwards, it is transmitted at the rate of one byte per transfer cycle.

Data is transferred in one direction only from the Debug Data Register to the external interface. The data is shifted out at the clock edge. The clock edge is selected by the EDGE_SEL bit in the Debug Control Register. After being shifted out, valid data is provided at the opposite edge of the MSCLK. For example, when the EDGE_SEL bit is '0' (default), valid data is provided at the falling edge of MSCLK. The Setup Time (to the falling edge of MSCLK) is 10 ns, minimum. The Hold Time is 1 ns, minimum.

When the Serial Debug Port is inactive, the MSCLK and MSDAT outputs are '1.' The EC Bus Clock clock input is the transfer clock.

FIGURE 41-3: DATA TRANSFER



© 2012-2018 Microchip Technology Inc.

41.9 EC-Only Registers

The registers listed in the EC-Only Register Summary table are for a single instance of the Trace FIFO Debug Port (TFDP). The addresses of each register listed in this table are defined as a relative offset to the host “Base Address” defined in the EC-Only Register Address Range Table.

TABLE 41-6: EC-ONLY REGISTER ADDRESS RANGE TABLE

Instance Name	LDN from Table 4-3 on page 60	AHB Base Address
TFDP Debug Port	23h	F0_8C00h

Note 41-1 The Base Address indicates where the first register can be accessed in a particular address space for a block instance.

TABLE 41-7: EC-ONLY REGISTER SUMMARY

Offset	Register Name (Mnemonic)
00h	Debug Data Register
04h	Debug Control Register

41.9.1 DEBUG DATA REGISTER

The Debut Data Register is Read/Write. It always returns the last data written by the TFDP or the power-on default ‘00h’.

Offset	00h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7:0	Data Debug data to be shifted out on the TFDP Debug port. While data is being shifted out, the Host Interface will ‘hold-off’ additional writes to the data register until the transfer is complete.	R/W	00h	nSYS_RST

41.9.2 DEBUG CONTROL REGISTER

Offset	02h			
Bits	**Description**	**Type**	**Default**	**Reset Event**
7	RESERVED	R	0b	-
6:4	IP_Delay Inter-packet delay in terms of TFDP Debug output clocks. A value of 0 provides a 1 clock inter-packet period, while a value of 7 provides 8 clocks between packets:	R/W	000b	nSYS-_RST
3:2	DIVSEL The TFDP Debug output clock is determined by this field, according to Table 41-8, "TFDP Debug Clocking":	R/W	00b	nSYS-_RST
1	EDGE_SEL 0= Data is shifted out on the rising edge of the debug clock (Default) 1= Data is shifted out on the falling edge of the debug clock	R/W	0b	nSYS-_RST
0	EN 0= Clock is disabled (Default) 1= Clock enabled	R/W	0b	nSYS-_RST

TABLE 41-8: TFDP DEBUG CLOCKING

DIVSEL	TFDP Debug Clock
00	EC Bus Clock divided by 2
01	EC Bus Clock divided by 4
10	EC Bus Clock divided by 8
11	Reserved. Implemented as divide by 2.

© 2012-2018 Microchip Technology Inc.

42.0 BOOT ROM

42.1 Overview

The Boot ROM code can be used to program the internal Flash using the 16C550A UART (see Section 42.2, "Description"). The Boot ROM also includes a these bits apply to both the Flash Memory Array at address 00_0FFCh, and the Embedded Flash Initialization Register. to simplify access to the MEC1632 EEPROM in the internal Flash Memory Array.

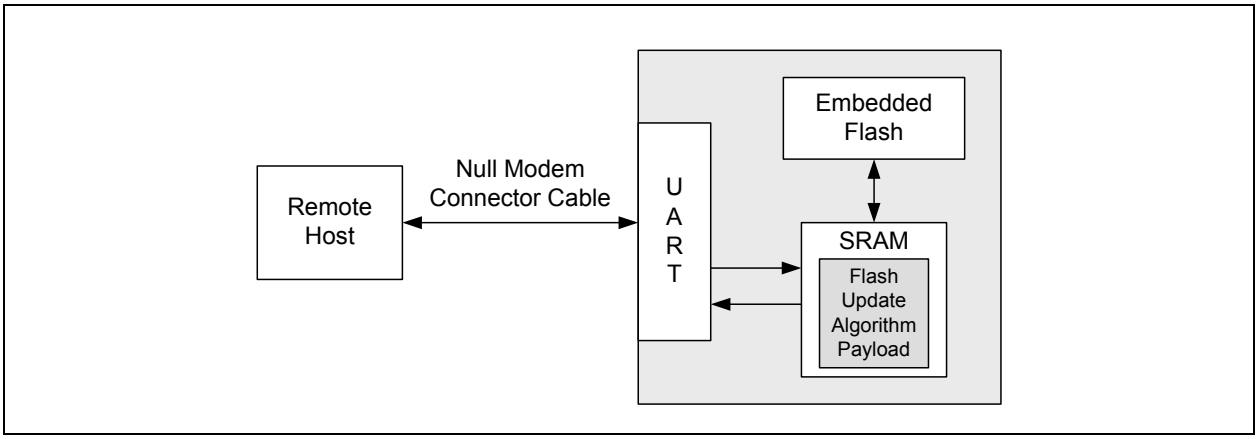
42.2 Description

The MEC1632 includes a 4 KB Boot ROM to download, in a factory environment, an SRAM-based EC application using the MEC1632 built-in serial UART. The SRAM application can then update the embedded Flash image. This behavior depends upon the state of Control Flags, as described in Section 42.3, on page 567.

Following a VTR power-on-reset, the Boot ROM configures the on-board 2-wire UART and attempts to establish communication with an external Host. If no communication is established within a configurable timeout period, the Boot ROM restores the UART to its default state and starts loading code from the first physical address of the embedded Flash. If proper Host communication is established, the Boot ROM downloads the application code into SRAM that performs the actual embedded Flash update. After the Flash application download is complete and the application payload is verified, the Boot ROM leaves the UART configured and transfers execution to the first physical address of SRAM.

Figure 42-1 illustrates a system-level view of the connection of the remote host to the MEC1632.

FIGURE 42-1: SYSTEM-LEVEL VIEW



42.3 Control Flags

The DWORD at address 00_0FFCh in the Flash Memory Array contains Boot ROM Control Flags, which are loaded by hardware into the read-only Embedded Flash Initialization Register following a VTR POR. The Boot ROM reads the state of the Control Flags using the Embedded Flash Initialization Register. The Control Flags are defined in Table 42-1.

Note:	If the Flash is changed at address 00_0FFCh, a VTR POR is required before the Boot ROM will see any change in the Control Flags. If the Flash is erased, the Boot ROM implements an infinite timeout waiting for the Host ACK following a VTR POR.

TABLE 42-1: Boot ROM Control Flags

Bits (Note 42-1)	Value	Description
0	0	JTAG accesses blocked except for mass erase.
	1	All JTAG accesses allowed.
1	0	Bypass Boot ROM and jump to 00_0000h.
	1	Enable Boot ROM host communication using UART.
3:2	00	20 ms timeout waiting for Host ACK.
	01	100 ms timeout waiting for Host ACK.
	10	200 ms timeout waiting for Host ACK.
	11	No timeout waiting for Host ACK (i.e., wait forever).
4	0	Use external UART clock (GPIO025 = 1.8432 MHz).
	1	Use internal UART clock (MCLK/11).

Note 42-1 these bits apply to both the Flash Memory Array at address 00_0FFCh, and the Embedded Flash Initialization Register.

42.4 UART Boot Strap Option

The Boot ROM samples the UART_BOOT_STRAP before evaluating the Boot ROM Control Flags. Based on the value of the strap, the boot sequence proceeds as follows:

TABLE 42-2: ACTION ON UART_BOOT_STRAP

UART Boot Strap Value	Bypass Bit in Flash	Action on POR
1	1	Initialize UART and wait for Host; Host response timeout is based on the Boot ROM Control Flags; UART baud clock based on Boot ROM Control Flags;
1	0	Jump to location 0 in the Flash; Do not activate UART
0	X	Initialize UART and wait for Host; No timeout for Host response; Use standard 1.8432MHz clock for UART; Boot ROM Control Flags are ignored

© 2012-2018 Microchip Technology Inc.

43.0 GANG PROGRAMMER INTERFACE

43.1 Introduction

The Gang Programmer Interface enables off-chip driven programming of the on-chip Flash memory after device packaging, and using equipment other than an ATE.

The Gang Programmer Interface accesses the on-chip Flash using the existing on-chip Flash Controller interface, and along with the on-chip JTAG interface provides the functions shown in Table 43-1. Activation of the Gang Programmer Interface requires the JTAG interface.

TABLE 43-1: GANG PROGRAMMER INTERFACE FUNCTIONS

	Function	Description
1.	Mass Erase	The Mass Erase function is *not* performed using the Gang Programmer Interface and only affects the Flash Main Memory Array and EEPROM. The Mass Erase function is implemented using the on-chip JTAG Interface and the Mass Erase (ME) bit.
2.	Mass Programming	The Mass Programming function is performed using the Gang Programmer Interface and can program 192KB Kbytes in ~1 sec.
3.	Mass Verify	The Mass Verify function is performed using the Gang Programmer Interface and can validate 192KB Kbytes in 0.5 sec., or less.

43.2 References

No references have been cited for this chapter

43.3 Terminology

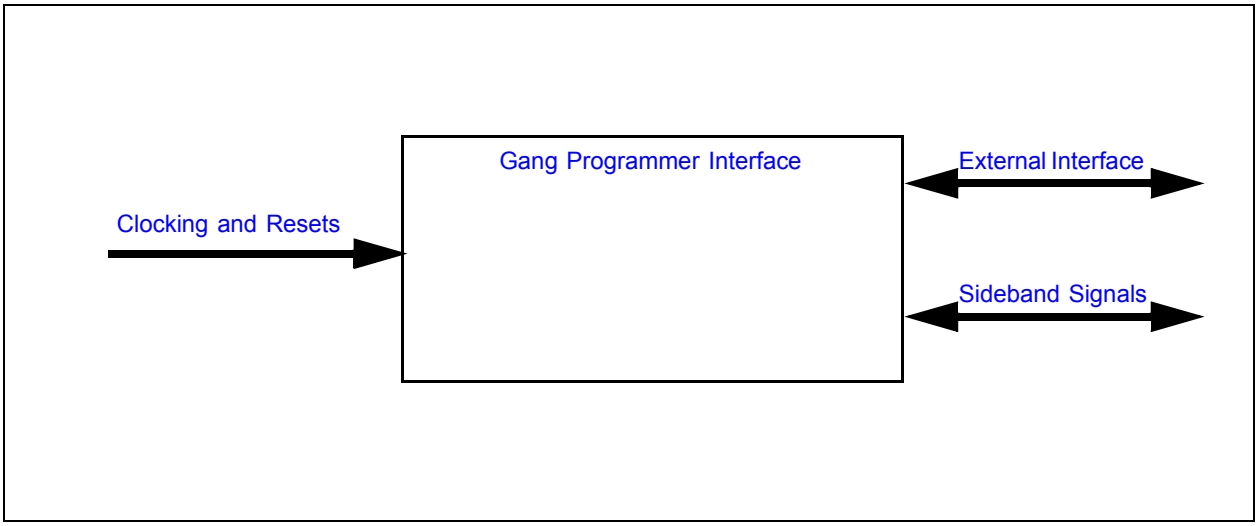
TABLE 43-2: TERMINOLOGY

Term	Definition
–	–

43.4 Interface

This block is an IP block designed to be incorporated into a chip. The following diagram illustrates the block interface.

FIGURE 43-1: BLOCK DIAGRAM



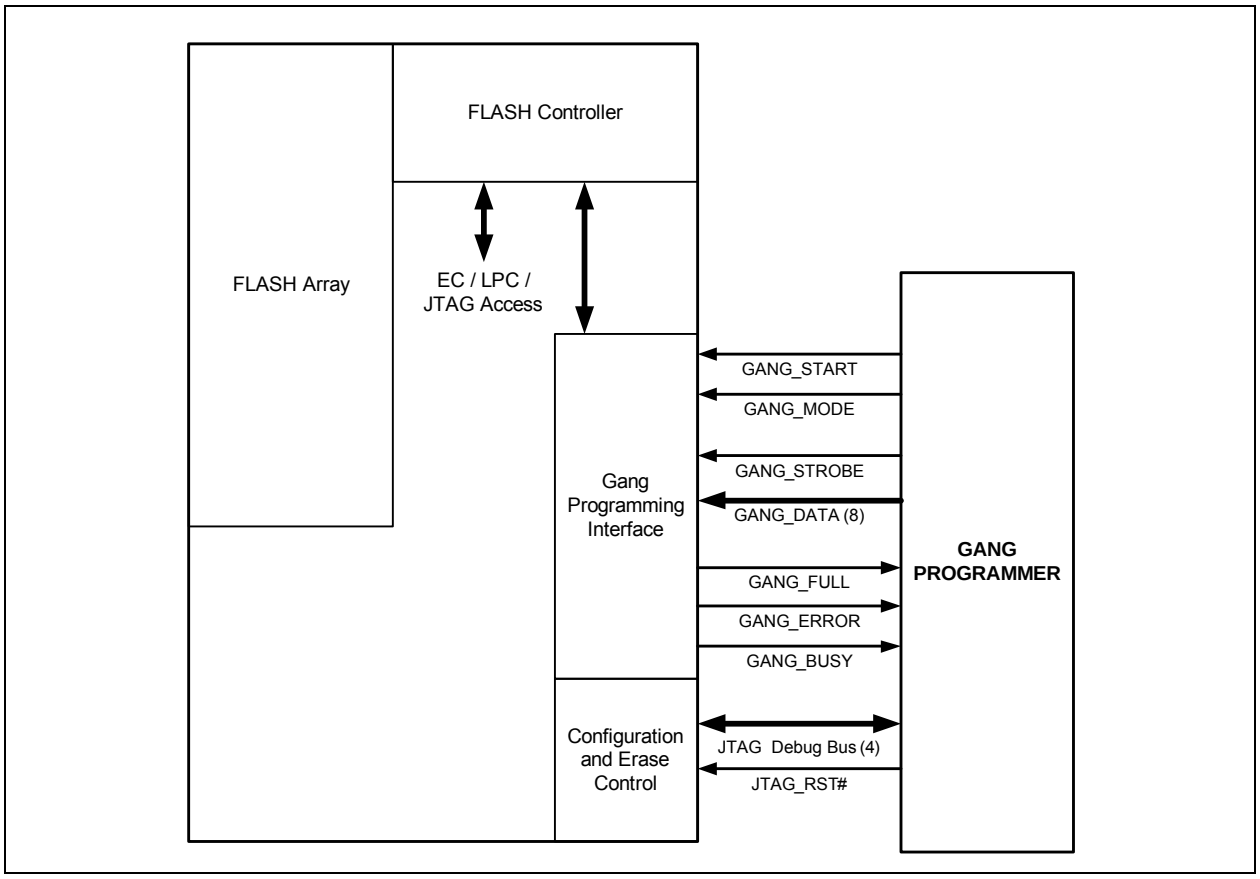
43.4.1 CLOCKING AND RESETS

This IP block has the following clocks and reset ports. For a complete list of all the clocks and resets associated with this block see Section 43.5, "Power, Clocks and Resets," on page 573.

TABLE 43-3: CLOCKING AND RESETS SIGNAL DESCRIPTION TABLE

Name	Direction	Description
nSYS_RST	Input	Reset asserted when power is cycled to the block
EC_BUS_CLK_EN	Input	Clock source to the block. All block clocks are derived from this source.

43.4.2 EXTERNAL INTERFACE SIGNALS



© 2012-2018 Microchip Technology Inc.

TABLE 43-4: EXTERNAL INTERFACE SIGNALS DESCRIPTION TABLE

Name	DiR (Note 43-1)	Size	Type	Description
GANG_START	Input	1	Command	A low-to-high transition of the GANG_START input starts a Gang Programmer Interface operation. On the rising edge of the GANG_START, the GANG_MODE signal is latched internally and the internal state machines are reset to their standby state. GANG_MODE determines what operation is to be performed.
GANG_MODE	Input	1		GANG_MODE is latched on the rising edge of the GANG_START. GANG_MODE = 1 means Mass Programming, GANG_MODE = 0 means Mass Verify.
GANG_STROBE	Input	1		GANG_STROBE pulses high to latch each byte of data. The GANG_STROBE pulse should be a minimum of 2 microseconds. GANG_START has priority over GANG_STROBE.

© 2012-2018 Microchip Technology Inc.

TABLE 43-4: EXTERNAL INTERFACE SIGNALS DESCRIPTION TABLE (CONTINUED)

Name	DiR (Note 43-1)	Size	Type	Description
GANG_DATA	Input	8	Data Transfer	GANG_DATA is the 8-bit Flash program data presented least-significant byte first from the lowest word address to the highest.
GANG_FULL	Output	1		GANG_FULL indicates whether there is space in the Flash Controller for the next byte of data. GANG_FULL = '1' means that there is no available space and the Gang Programmer must wait until GANG_FULL = '0' to continue.
GANG_ERROR	Output	1		Mass Programming: GANG_ERROR is set low when GANG_START is asserted. GANG_ERROR remains low unless the external system presents data while GANG_FULL = '1.' GANG_ERROR is asserted '1' when data has been lost as a result of this overrun. GANG_ERROR will retain its state until the next low-to-high transition of GANG_START. Mass Verify: GANG_ERROR is set high when GANG_START is asserted. GANG_ERROR remains high as long as GANG_START remains high. When GANG_START is de-asserted (transitions high-to-low), GANG_ERROR will be set low if no errors occurred during verification, and all presented data matched the contents of the Flash. GANG_ERRORR will be asserted '1' if any data byte presented by the Gang Programmer did not match the corresponding byte in the Flash, or the Gang Programmer did not present an integral number of 2048-byte sectors to the Gang Programmer Interface.
GANG_BUSY	Output	1		GANG_BUSY is held low initially and also while GANG_START is low. On the rising edge of GANG_START, GANG_BUSY goes high and remains high until either the entire array has been written or verified, or the GANG_START signal has again gone low, in which case the logic is reset. The Gang Programmer Interface de-asserts GANG_BUSY when the Mass Programming or Mass Verify operation has completed and the Flash block is in standby state. If GANG_START is driven low during a Mass Programming operation, GANG_BUSY is not de-asserted until the interface is again able to accept a new command. The external system must keep GANG_START low until GANG_BUSY is de-asserted. When the GANG_BUSY output goes back low to signal that the operation is complete the GANG_ERROR output indicates the summary status of the overall command, as appropriate to that command. This summary status remains presented until the GANG_START signal next goes high.

Note 43-1 all of the outputs in Table 43-4 require push-pull drivers; i.e., status output signals from multiple devices are not bussed in the Gang Programmer Interface.

© 2012-2018 Microchip Technology Inc.

43.4.3 SIDEBAND SIGNALS

TABLE 43-5: SIDEBAND SIGNALS DESCRIPTION TABLE

Name	Direction	Description
HoldOff	Output	This output is used to hold the EC in reset until Gang Programmer Interface operations have been completed (Figure 43-2).
nStart	Input	The nStart input is used to hold off the Gang Programmer Interface until the system is fully operational.
GangEnable	Input	When the GangEnable input is asserted '1,' the Gang Programmer Interface has been enabled by the JTAG interface. GangEnable represents the state of the Gang Enable bit in the JTAG Test register. See also Section 43.8.3.7, "Operational Flow," on page 579 for a description of the Gang Programmer Interface Operating Conditions.
MassErase	Input	The MassErase input is used to qualify the GangEnable input as described in Section 43.8.3.7, "Operational Flow," on page 579. MassErase represents the state of the Mass Erase (ME) bit in the JTAG Test register.

43.5 Power, Clocks and Resets

43.5.1 POWER DOMAINS

TABLE 43-6: POWER SOURCES

Name	Description
VTR	The Gang Programmer Interface runs in a single power domain.

43.5.2 CLOCKS

TABLE 43-7: CLOCKS

Name	Description
EC_BUS_CLK_EN	EC_BUS_CLK_EN provides all clocking for the Gang Programmer Interface.

43.5.3 RESETS

TABLE 43-8: RESET SIGNALS

Name	Description
nSYS_RST	nSYS_RST is the system reset (SysRst in Figure 43-2).

43.6 Interrupt Generation

None

43.7 Low Power Modes

The Gang Programmer Interface only consumes power when reading Flash Info block data and programming AHB registers immediately after a VTR POR (see Section 43.8.3, "Functional Description," on page 574). At all other times, clocking to the Gang Programmer Interface is gated 'off.'

43.8 Description

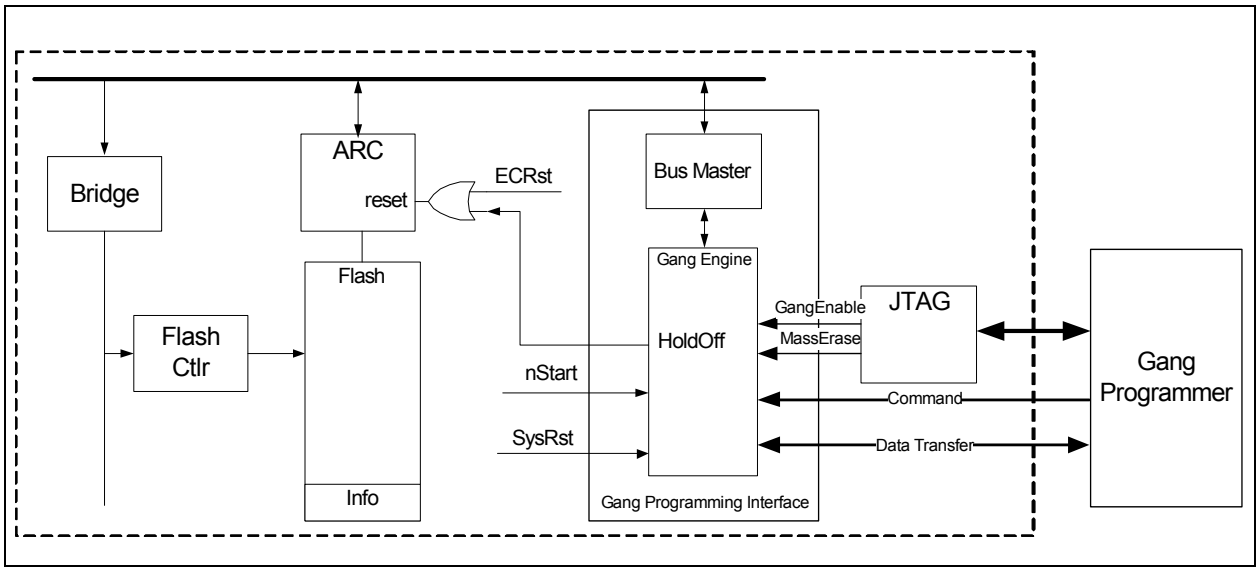
43.8.1 OVERVIEW

The Gang Programmer Interface consists of a state machine and a Bus Master Interface (Figure 43-2), and permits the Flash Array to be programmed in close to the minimum time permitted by the internal Flash hardware. The external Gang Programmer can program the devices using deterministic timing, or it can choose to optimize the programming time through the use of flow control signals that are output from the Gang Programmer Interface.

The Gang Programmer Interface is used to program only the main internal Flash Array. The emulated EEPROM region of the embedded Flash memory may be erased as part of the Mass Erase function but cannot be programmed, or read by the Gang Programmer Interface.

The Command and Data Transfer signals in Figure 43-2 refer to Table 43-4, “External Interface Signals Description Table,” on page 571.

FIGURE 43-2: MEC1632 INTERFACE SYSTEM VIEW



43.8.2 FLASH ARRAY

- 4 Byte Units (min.) for Programming
- 2048 Byte Units (min.) for Verification
- ~20 microseconds per Word Programming Time
- 64 Words per Row: 21 microseconds maintenance overhead per row = 20.33 microseconds/word total

43.8.3 FUNCTIONAL DESCRIPTION

43.8.3.1 Overview

The typical sequence for a gang programmer involves 3 steps:

1. Erase the main body of the Flash, using the Mass Erase function
2. Program the main body of the Flash, using the Mass Programming function
3. Verify that the program operation was successful using the Mass Verify function.

All of these functions are described in detail in the subsections that follow; including Gang Programmer Interface Activation, a description of Gang Programmer Data Transfers independent of the operating mode, and an illustration of the Gang Programmer Operational Flow.

© 2012-2018 Microchip Technology Inc.

43.8.3.2 Activation

Activation of the Gang Programmer Interface, and initiating the Mass Erase function requires the on-chip JTAG interface. A description of the JTAG Interface is beyond the scope of this document.

Activation of the Gang Programmer Interface occurs following nSYS_RST while GangEnable is asserted, and after the nStart input is asserted ('0') (Table 43-9). Multiplexing for the External Interface Signals (Table 43-4) is enabled upon Activation of the Gang Programmer Interface.

Once the Gang Programmer Interface is operational, the EC is held in reset and the Gang Programmer Interface retains control of the system until the next nSYS_RST event. Following Activation, the Gang Programmer Interface waits for a low-to-high transition on the GANG_START input to execute Mass Programming and Mass Verify commands (see Section 43.8.3.7, "Operational Flow").

TABLE 43-9: OPERATING CONDITIONS

	nStart	GangEnable	MassErase	Operational	Description (Note 43-3)
1.	X	0	X	No	Gang Programmer Interface disabled.
2.	1	1	X	No	Gang Programmer Interface enabled, but not operating until the on-chip system is fully ready.
3.	0	1	0	Yes (Note 43-2)	Gang Programmer Interface is enabled and operating as shown in Figure 43-6.
4.	X	1	1	No	Gang Programmer Interface is disabled when Mass Erase and Gang Enable are asserted.

Note 43-2 Once the Gang Programmer Interface is operational, the EC is held in reset and the Gang Programmer Interface retains control of the system until the next nSYS_RST event.

Note 43-3 GangEnable and MassErase are sampled on nSYS_RST.

43.8.3.3 Mass Erase

The Mass Erase function is triggered via the JTAG Interface using the ME bit. After a Mass Erase the JTAG interface is used to force an internal system reset. This resets the internal state and sets up the device so that a JTAG write can set the Gang Enable bit.

43.8.3.4 Mass Programming

Following Activation of the Gang Programmer Interface, Mass Programming is initiated using the GANG_START and GANG_MODE inputs as described in Table 43-4, "External Interface Signals Description Table," on page 571. The Gang Programmer Interface acknowledges Mass Programming by asserting GANG_BUSY, and de-asserting GANG_ERROR, and waiting for Data Transfers to begin. See Section 43.8.3.6 for a description of Data Transfers.

Once triggered, Mass Programming continues until the external Gang Programmer terminates it by de-asserting GANG_START (see FIGURE 43-3: Mass Programming Timing Diagram on page 576.) When GANG_START is brought low, further writing to the Flash is halted, even if fewer than 192KB Kbytes of data have been written. GANG_BUSY remains asserted until the Gang Programmer Interface completes the programming operation and the Flash controller is idle.

Mass Programming always starts at beginning of the Flash Array. Each 32-bit Word is provided as 4 one-byte transfers. The Gang Programmer is unaware of base address(es) of the data in the Flash Array, and only aware of the length of the array.

While the bytes are presented by the Gang Programmer during Mass Programming Data Transfers, the GANG_ERROR signal will indicate whether there have been any data Overrun errors. After the last Byte has been written to the FLASH, and the Gang Programmer de-asserts the GANG_START input, the GANG_BUSY output signal is de-asserted to signal the end of the operation. Note that the Gang Programmer must keep GANG_START low until GANG_BUSY is de-asserted. Once GANG_START is de-asserted, the GANG_ERROR error status will be held at its final value until GANG_START is next asserted.

Programming more than the 192KB KBytes of the internal Flash may cause errors, but the Gang Programmer Interface does not check if the Flash size was exceeded.

FIGURE 43-3: MASS PROGRAMMING TIMING DIAGRAM

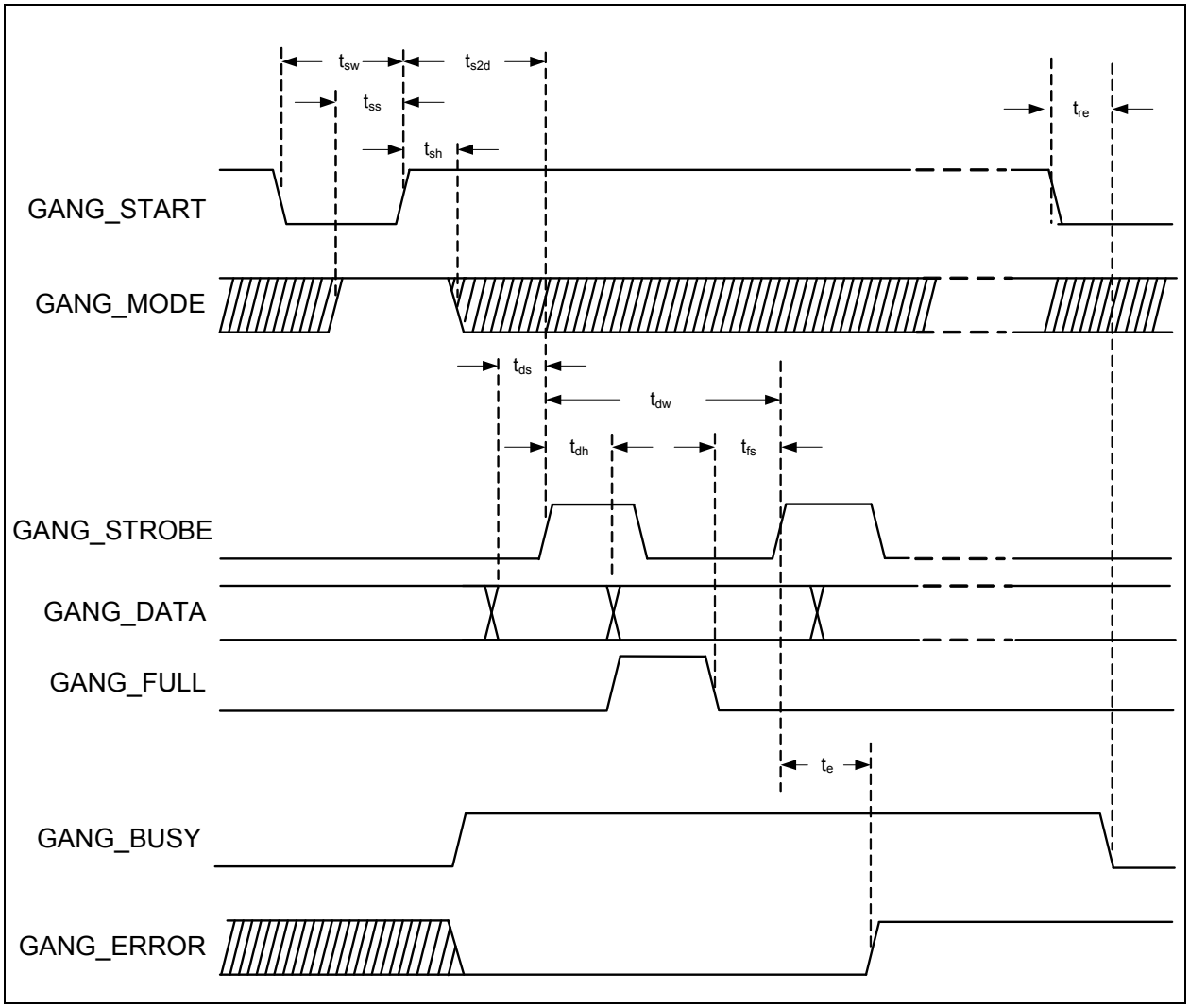


TABLE 43-10: MASS PROGRAM TIMING PARAMETERS

Name	Description	MIN	MAX	Units
t_{ss}	Gang_Mode setup time to rising edge of Gang_Start	50		nsec
t_{sw}	Gang_Start pulse width	200		nsec
t_{sh}	Gang_Mode hold time from rising edge of Gang_Start	200		nsec
t_{ds}	Gang_Data setup time to rising edge of Gang_Strobe	50		nsec
t_{dw}	Gang_Strobe pulse width	1		µsec
	Gang_Strobe pulse width, slow enough to prevent Gang_Full from going high	5		µsec
t_{dh}	Gang_Data hold time from rising edge of Gang_Strobe	400		nsec
t_f	Gang_Full low setup time before rising edge of Gang_Strobe	200		nsec
t_{re}	Gang_Start falling edge to Gang_Busy falling edge		20	µsec
t_e	Gang_Strobe rising edge to Gang_Error rising edge		20	µsec

© 2012-2018 Microchip Technology Inc.

43.8.3.5 Mass Verify

Following Activation of the Gang Programmer Interface, Mass Verify is initiated using the GANG_START and GANG_MODE inputs as described in Table 43-4, “External Interface Signals Description Table,” on page 571. The Gang Programmer Interface acknowledges Mass Verify by asserting GANG_BUSY, and asserting GANG_ERROR, and waiting for Data Transfers to begin. See Section 43.8.3.6 for a description of Data Transfers.

Mass Verify always starts at beginning of the Flash Array. Each 32-bit Word is provided as 4 one-byte transfers. The Gang Programmer must perform Mass Verify operations on sector boundaries (i.e., 2048 byte segments), or errors will occur. The Gang Programmer has no access to the actual Flash data.

Once triggered, Mass Verify continues until the external Gang Programmer terminates it by de-asserting GANG_START (see FIGURE 43-4: Mass Verify Timing Diagram on page 577.) When GANG_START is brought low, further comparison is halted, even if fewer than 192KB Kbytes of data have been verified. GANG_BUSY remains asserted until the Gang Programmer Interface is idle.

The Mass Verify function compares the contents of the Flash Array against a pattern provided by the Gang Programmer during Data Transfers. As each byte is input, the Gang Programmer Interface compare it against the corresponding Flash Array byte, and keeps a running 1-bit indication of whether any mismatches have occurred.

The end of the Mass Verify operation occurs when the Gang Programmer de-asserts GANG_START. When the GANG_BUSY output goes low after the Mass Verify function terminates, the GANG_ERROR output reflects the state of the 1-bit mismatch indicator; i.e., there is no early indication of a mismatch, or error. GANG_START will be high when GANG_BUSY de-asserts if a mismatch occurred, or the number of GANG_STROBE assertions was not an exact multiple of 2048. GANG_ERROR retains its state until GANG_START is next asserted.

FIGURE 43-4: MASS VERIFY TIMING DIAGRAM

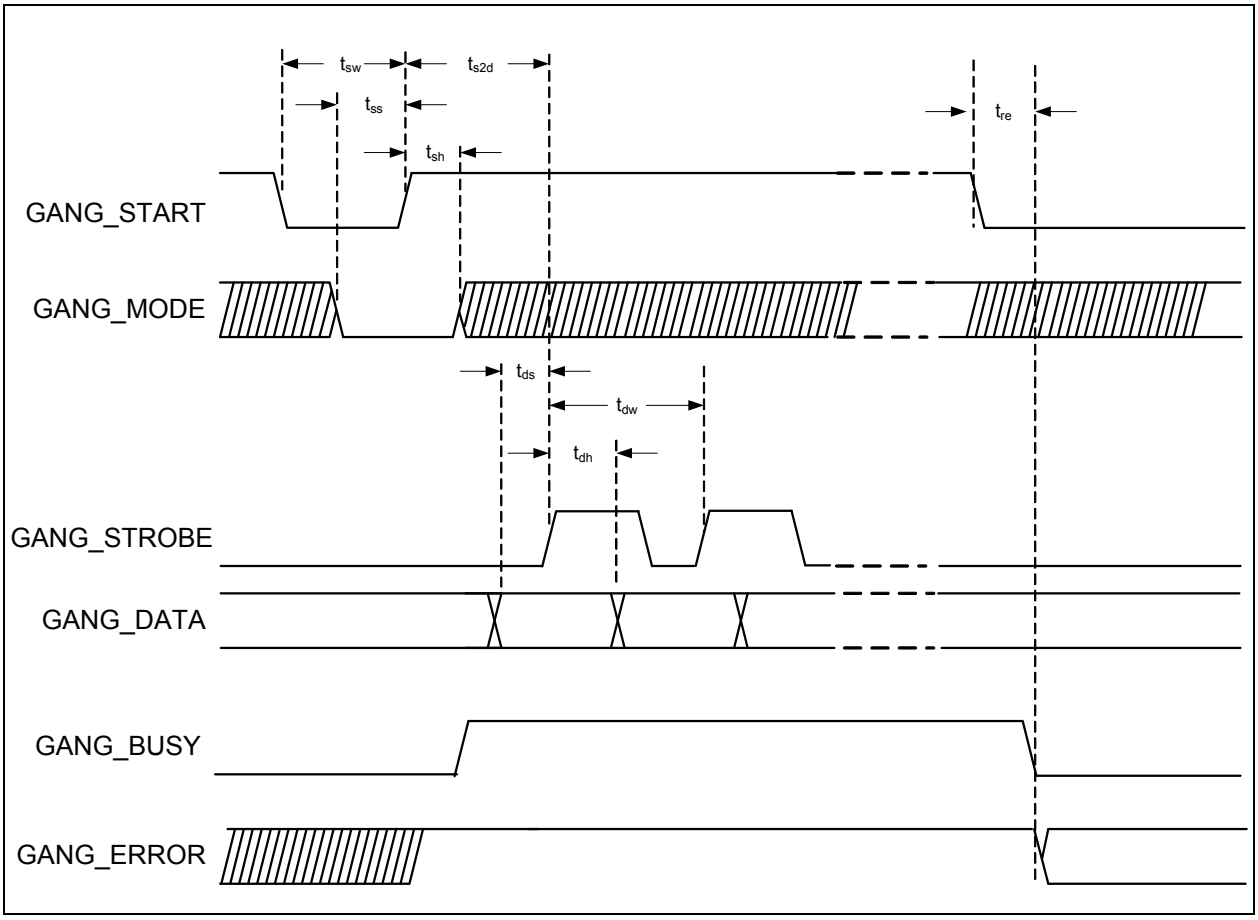


TABLE 43-11: MASS VERIFY TIMING PARAMETERS

Name	Description	MIN	MAX	Units
t_{ss}	Gang_Mode setup time to rising edge of Gang_Start	50		nsec
t_{sw}	Gang_Start pusle width	200		nsec
t_{sh}	Gang_Mode hold time from rising edge of Gang_Start	200		nsec
t_{ds}	Gang_Data setup time to rising edge of Gang_Strobe	50		nsec
t_{dw}	Gang_Strobe pulse width	750		nsec
t_{dh}	Gang_Data hold time from rising edge of Gang_Strobe	500		nsec
t_{re}	Gang_Start fall to Gang_Busy and Gang_Error response		200	nsec

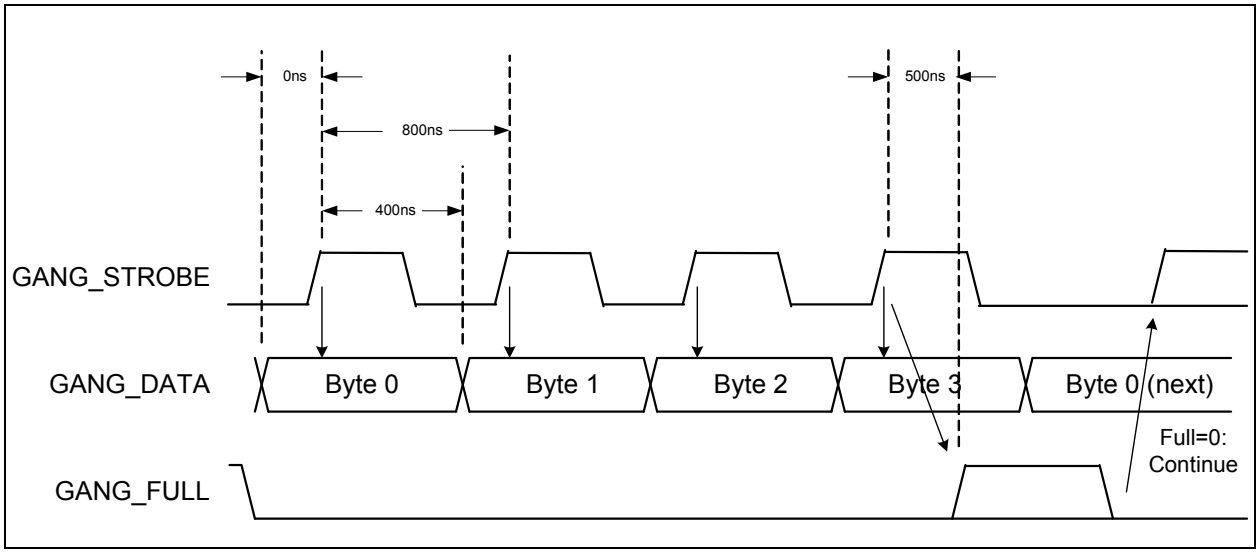
43.8.3.6 Data Transfers

The GANG_FULL output always indicates when space is available for new data. Once GANG_FULL is not asserted ('0'), it will not be asserted except on an external write of more data, as shown in Figure 43-5.

A low-to-high transition on GANG_STROBE signals the availability of a new byte of data on GANG_DATA pins. As shown in Figure 43-5, no setup time is required. The Gang Programmer must hold GANG_DATA valid for at least 400 ns after the low-to-high transition of GANG_STROBE. The period between low-to-high transitions on GANG_STROBE must be at least 800 ns. GANG_FULL will be asserted, if required, a maximum of 500 ns after the low-to-high transition of GANG_STROBE.

Because the on-chip Flash program time is approximately 5.2 microseconds/byte, if the Gang Programmer asserts GANG_STROBE with a minimum of 6 microseconds of delay between bytes, the programmer can ignore the GANG_-FULL signal.

FIGURE 43-5: DATA TRANSFER TIMING



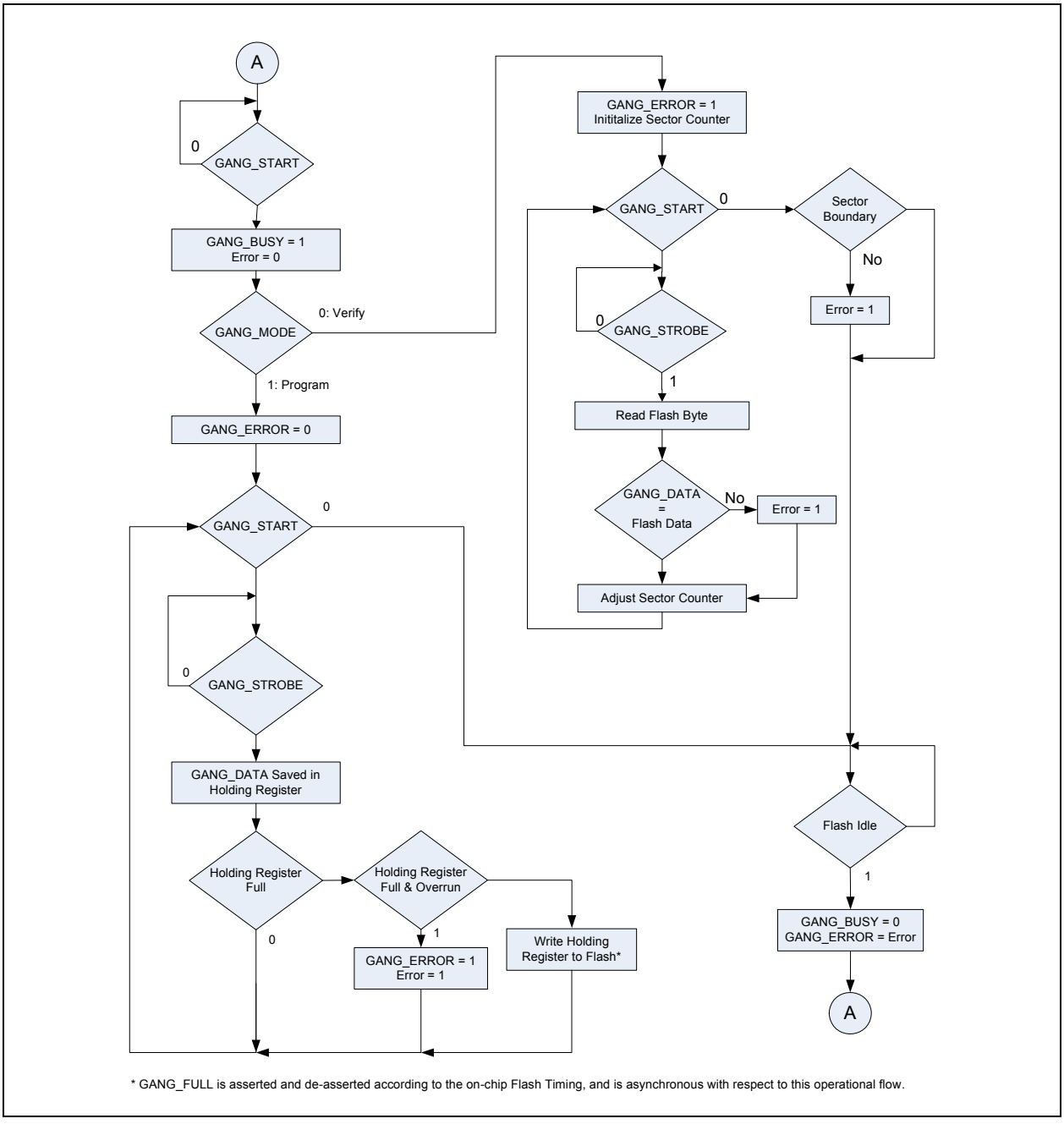
© 2012-2018 Microchip Technology Inc.

43.8.3.7 Operational Flow

Figure 43-6 illustrates the Operational Flow for the Mass Programming and Mass Verify commands as described in the sections above. Point 'A' in this figure is entered following Gang Programmer Interface Activation as described in Section 43.8.3.2, "Activation," on page 575, and again following command termination.

Note that GANG_FULL is asserted and de-asserted according to the on-chip Flash Timing, and is asynchronous with respect to the operational flow shown in Figure 43-6.

FIGURE 43-6: GANG PROGRAMMER OPERATIONAL FLOW



44.0 JTAG AND XNOR

44.1 General Description

The MEC1632 includes a JTAG Slave for Debugging ARC Firmware, a Boundary Scan slave, and a JTAG Master. All of these function share the MEC1632 JTAG Interface defined in the Pin Configuration chapter.

The JTAG slaves are asynchronously reset and deactivated when the JTAG_RST# input pin is asserted ('0'). When JTAG_RST# is not asserted, only one slave is enabled. JTAG Slave Selection depends on the state of the TAP Controller Select Strap Option. The JTAG Master is independent of the slave TAP controllers and cannot be used when the JTAG_RST# pin is not asserted.

44.2 Slave Selection

44.2.1 TAP CONTROLLER SELECT STRAP OPTION

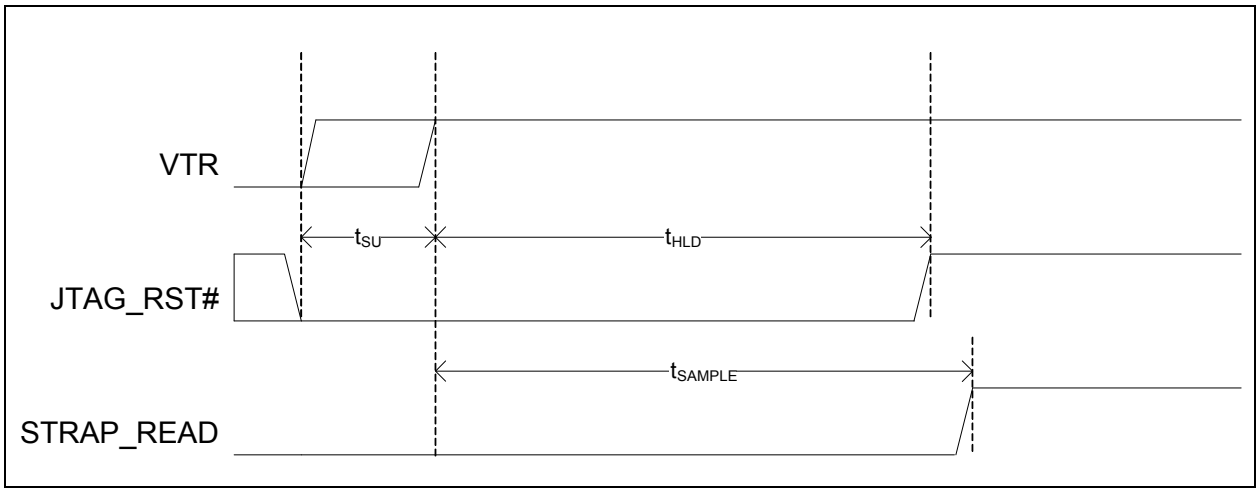
The TAP Controller Select Strap Option determines the JTAG slave that is selected when JTAG_RST# is not asserted. The state of the TAP Controller Select Strap Option pin, defined in the Pin Configuration chapter, is sampled by hardware at VTR POR according to the Slave Select Timing as defined in Section 44.2.2 and is registered internally to select between the debug and boundary scan TAP controllers.

If the TAP Controller Select Strap Option is sampled low, the debug TAP controller is selected; if the strap is sampled high, the boundary scan slave is selected. An internal pull-up resistor is enabled by default on the TAP Controller Select Strap Option pin and can be disabled by firmware, if necessary.

44.2.2 SLAVE SELECT TIMING

The JTAG_RST# input pin must be asserted at VTR power-up and follow the timing as defined in this section. The relationship between the TAP Controller Select Strap Option sample timing and the JTAG_RST# pin de-assertion timing is illustrated below in Figure 44-1 and Table 44-1.

FIGURE 44-1: ASYNCHRONOUS JTAG RESET AND TAP Controller Select Strap Option POWER-UP TIMING



© 2012-2018 Microchip Technology Inc.

TABLE 44-1: Slave Select Timing PARAMETERS

Parameter	Symbol	MIN	Units	Notes
JTAG_RST# Setup Time	t_{SU}	see Section 46.11, "JTAG Interface Timing," on page 628		
JTAG_RST# Hold Time	t_{HLD}			
TAP Controller Select Strap Option Sample Time	t_{SAMPLE}	ARC reset time	–	t_{SAMPLE} is the same as $t_{STRETCH}$ in FIGURE 7-20: VTR Power-Up Timing on page 127.

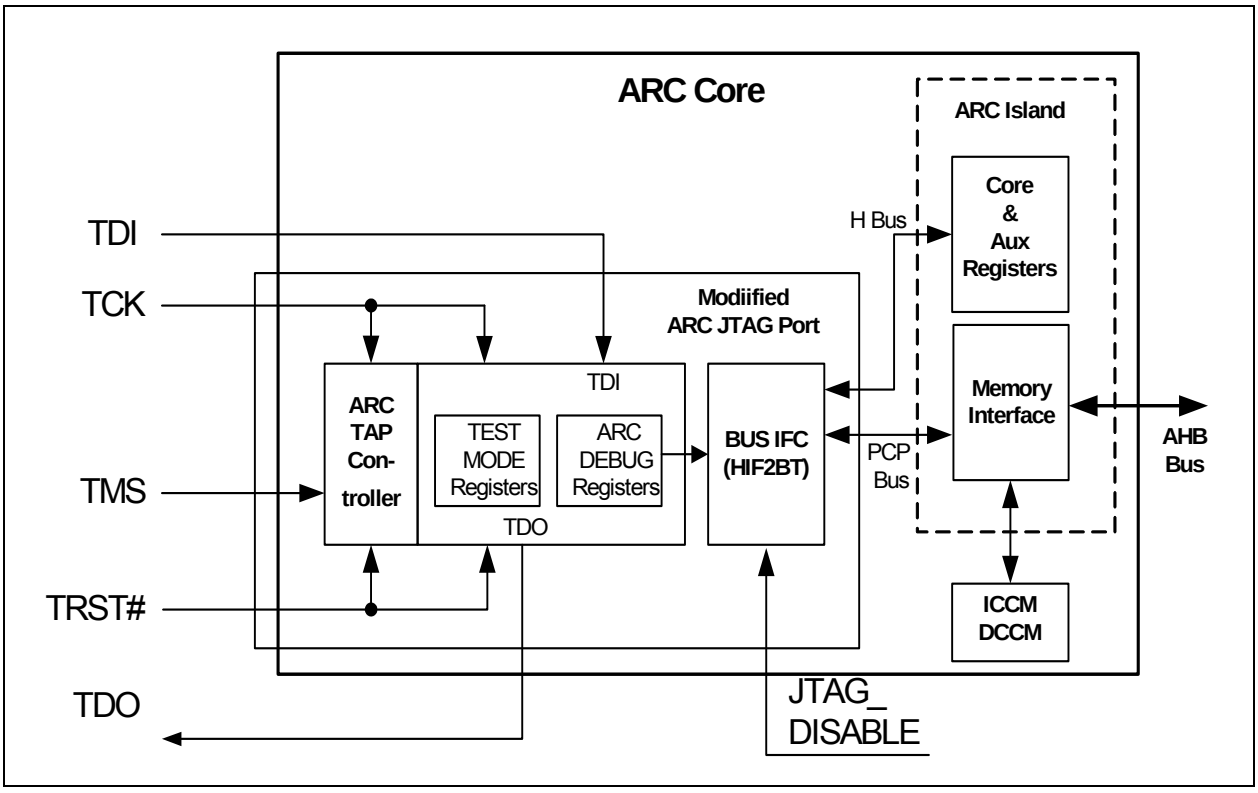
44.3 JTAG Slave for Debugging ARC Firmware

- The ARC JTAG Port is defined in the ARC 600 External Interfaces Reference Manual, Chapter 2. The modifications are described in this chapter. The ARC JTAG Port has been modified to provide additional Data Registers (see Note 44-3.) The TEST MODE Register provide additional on-chip support specific to the MEC1632.

44.3.1 ARC JTAG CAPABILITIES

- Fully compliant with IEEE1149.1 standard
- 4-bit Instruction Register
- Standard 1-bit BYPASS register
- Standard 32-bit IDCODE register
- Four JTAG registers give access to on-chip memory and register resources
- Can read or write a 32-bit quantity from or to any ARC Core Register, Aux Register or 32-bit aligned memory location. No other interfaces are provided or needed
- Accesses to Aux Registers and Memory do not require the ARC processor to be halted
- Memory accesses are always performed in units of 32 bits

FIGURE 44-2: BLOCK DIAGRAM OF ARC JTAG SYSTEM



As shown in Figure 44-2, the ARC JTAG Port sits within the ARC Core, between the JTAG signals and the ARC Island block, which contains the Core and Aux register sets and the Memory Interface to the Closely-Coupled Memories (ICCM / DCCM) and the AHB Bus (FLASH memory and Memory-Mapped I/O registers). It is not associated with any form of Boundary Scan, but instead is used by an external JTAG host to access memory and register resources on behalf of a debugger program. There are no other connections between the ARC JTAG Port and the rest of the ARC Core: register access is enough to halt and restart the processor, as well as to detect that the processor has halted (a continuous poll on the STATUS32 Aux Register).

Internally, the ARC JTAG port consists of the following sub-blocks:

- The ARC Test Access Port (TAP) Controller. This block is driven by the JTAG_CLK, TMS and JTAG_RST# inputs, and provides the control signals for the JTAG data transfers. It is a single state machine consisting of 16 states, whose transitions are controlled strictly by the state of TMS on each rising edge of JTAG_CLK. The low-active JTAG_RST# input provides an asynchronous reset, though JTAG_CLK and TMS together can also bring the TAP Controller to the reset state (5 consecutive JTAG_CLK rising edges with TMS=1).
- The ARC DEBUG & TEST MODE Registers blocks. This blocks handles the data transfers as directed by the TAP, and contains the registers and shift registers internal to the JTAG Port. The holding registers in this block consist of a 4-bit Instruction register, a 1-bit Bypass register, and a set of Data registers (see Note 44-3) of various lengths. There are three sets of Data Registers: JTAG Standard Data Registers, JTAG Debug Data Registers, & JTAG Test Mode Data Registers. The JTAG Test Mode Data Registers provide additional on-chip support specific to the MEC1632.
- The Bus Interface block. This block accepts values for the JTAG Debug Data Registers: ADDRESS, DATA, STATUS and TRANSACTION COMMAND, and uses them to request data transfers from the ARC's inner core ("Island") sub-block. \
- As part of the Boot block protection the JTAG Port's JTAG Debug Data Registers can be disabled to prevent an external debugger from potentially halting the EC and then reading or re programing any word in the Boot Block.

44.4 Boundary Scan

Boundary Scan includes registers and functionality as defined in IEEE 1149.1 and the MEC1632 BSDL file. Functionality implemented beyond the standard definition is summarized in Table 44-3. The MEC1632 Boundary Scan JTAG ID is shown in Table 44-2.

TABLE 44-2: Boundary Scan JTAG ID

Part Number	JTAG ID
MEC1632	02092445n

TABLE 44-3: EXTENDED Boundary Scan FUNCTIONALITY

Bits	Function	Description
12, 14	TAP Controller Select Strap Option Override	When the TAP Controller Select Strap Option Override is '1,' the TAP Controller Select Strap Option is overridden to select the debug TAP Controller until the next time that the TAP Controller Select Strap Option is sampled (Note 44-1).

Note 44-1 To set Strap Override Function, write 0X1FFFFD to the TAP controller instruction register, then write 0x5000 to the TAP controller data register. Note that the instruction register is 18 bits long; the data register is 16 bits long.

© 2012-2018 Microchip Technology Inc.

44.5 JTAG Master

44.5.1 OVERVIEW

The JTAG Master controller in the MEC1632 enables the embedded controller to perform full IEEE 1149.1 test functions as the master controller for test operations at assembly time or in the field.

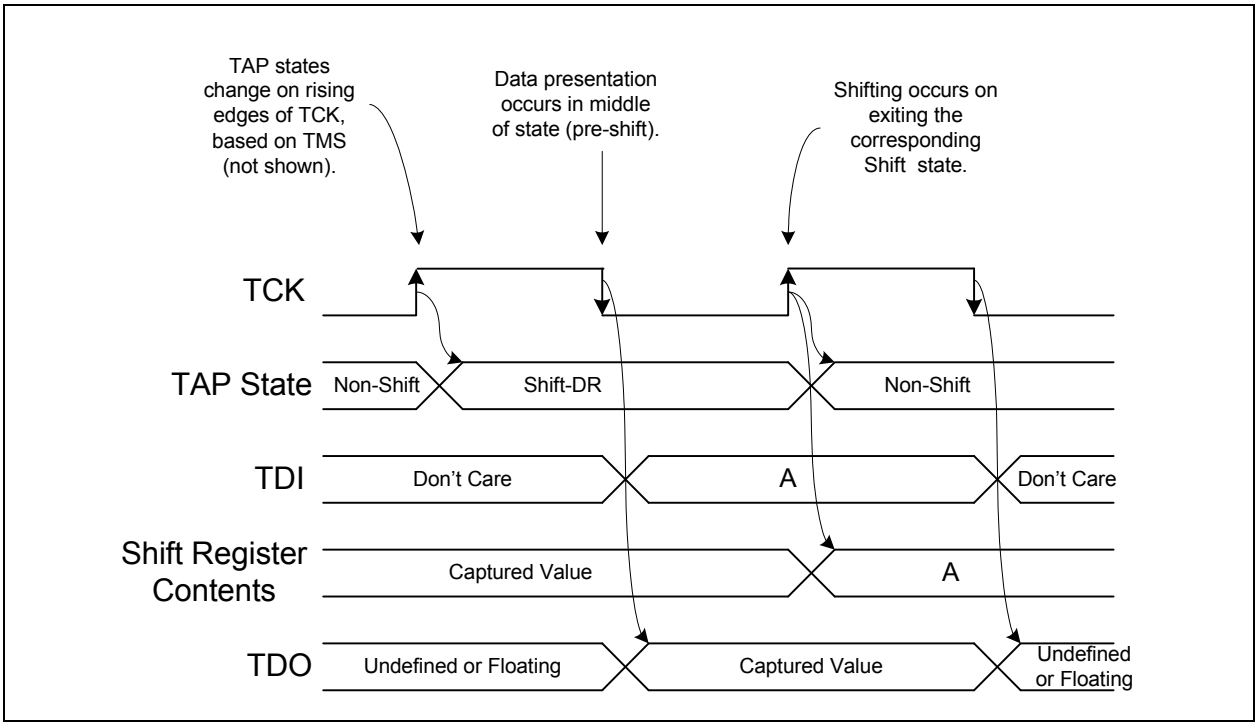
The JTAG Master interface shares the JTAG pin interface with the Boundary Scan and Debug TAP controllers; including, JTAG_CLK, JTAG_TDI, JTAG_TDO and JTAG_TMS. When the MEC1632 JTAG interface is configured as master, it is the responsibility of the master firmware to satisfy all requirements regarding JTAG port multiplexing. It is also it is the responsibility of the JTAG Master firmware to satisfy all requirements for external JTAG slave devices that require an external asynchronous reset (TRST#) input.

44.5.2 DESCRIPTION

When JTAG slave functions are not required and the JTAG Master is enabled, the JTAG Interface pins are turned around so that the pins JTAG_CLK, JTAG_TMS and JTAG_TDI become outputs and the JTAG_TDO becomes an input.

Figure 44-3, "JTAG Signal Clocking" shows the clocking behavior of JTAG in the TAP controller in a JTAG Slave device. The rows “TAP State” and “Shift Reg. Contents” refer to the state of the JTAG Slave device and are provided for reference. When configured as a Master, the JTAG interface drives JTAG_CLK and will shift out data onto JTAG_TMS and JTAG_TDI in parallel, updating the pins on the falling edge of JTAG_CLK. The Master will sample data on JTAG_TDO on the rising edge of JTAG_CLK.

FIGURE 44-3: JTAG SIGNAL CLOCKING



44.5.3 JTAG MASTER REGISTER INTERFACE

The JTAG Master interface uses the ARC Auxiliary Register interface; all JTAG Maser registers are Auxiliary registers, accessed by the ARC `lr` (load auxiliary register) and `sr` (store auxiliary register) instructions. Table 44-4, "JTAG Auxiliary Registers" lists these registers:

TABLE 44-4: JTAG AUXILIARY REGISTERS

Register Number	Auxiliary Name	LR/SR R/W	VTR Default	Description
FFFF_FFFFh	JTAG_COM	W	0000_000h	JTAG Master Command Register
FFFF_FFFEh	JTAG_TMS	R/W	0000_000h	JTAG Master register source for TMS pin
FFFF_FFFDh	JTAG_TDI	R/W	0000_000h	JTAG Master register source for TDI pin
FFFF_FFFCh	JTAG_TDO	R/W	0000_000h	JTAG Master register destination for TDO pin
FFFF_FFFBh	JTAG_STATUS	R	0000_000h	JTAG Master Status register
FFFF_FFFAh	JTAG_CONFIG	R/W	0000_000h	JTAG Master Configuration register

44.5.3.1 JTAG Master Command, JTAG_COM

FIGURE 44-4: JTAG_COM

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											Count[4:0]			

COUNT

If M in JTAG_CONFIG is 1, writing this field starts, clocking and shifting on the JTAG port. The JTAG Master port will shift count+1 times, so writing a 0 will shift 1 bit and writing 31 will shift 32 bits. The signal JTAG_CLK will cycle count+1 times. JTAG_TMS and JTAG_TDI will be shifted out on the falling edge of JTAG_CLK and JTAG_TDO will get shifted in on the rising edge of JTAG_CLK.

When M in JTAG_CONFIG is 0 the JTAG port is configured as a Slave and writing this field has no effect.

44.5.3.2 JTAG Master TMS, JTAG_TMS

FIGURE 44-5: JTAG_TMS

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMS[31:0]																															

TMS

When JTAG_COM is written, from 1 to 32 bits are shifted out of TMS, starting with bit 0, onto the JTAG_TMS pin. Shifting is at the rate determined by CLK in JTAG_CONFIG.

44.5.3.3 JTAG Master TDI, JTAG_TDI

FIGURE 44-6: JTAG_TDI

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDI[31:0]																															

TDI

When JTAG_COM is written, from 1 to 32 bits are shifted out of TDI, starting with bit 0, onto the JTAG_TDI pin. Shifting is at the rate determined by CLK in JTAG_CONFIG.

44.5.3.4 JTAG Master TDO, JTAG_TDO

FIGURE 44-7: JTAG_TDO

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDO[31:0]																															

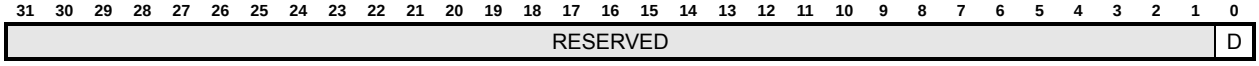
© 2012-2018 Microchip Technology Inc.

TDO

When JTAG_COM is written, from 1 to 32 bits are shifted into of TDO, starting with bit 0, onto the JTAG_TDO pin. Shifting is at the rate determined by CLK in JTAG_CONFIG.

44.5.3.5 JTAG Master Status, JTAG_STATUS

FIGURE 44-8: JTAG_STATUS



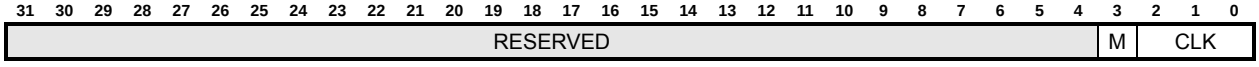
D

This bit is read-only.

This bit is set to 1 when JTAG_COM is written. It becomes 0 when the shifting has completed. Software can poll this bit to determine when a command has completed and it is therefore safe to retrieve the data in JTAG_TDO and to load new data into JTAG_TMS and JTAG_TDI.

44.5.3.6 JTAG Master Configuration, JTAG_CONFIG

FIGURE 44-9: JTAG_CONFIG



CLK

This field determines the clock rate of the JTAG_CLK signal. Options are shown in Table 44-5, "JTAG Clock Options":

TABLE 44-5: JTAG CLOCK OPTIONS

CLK Value	JTAG_CLK Clock Rate
0h	Reserved
1h	MCLK/2
2h	MCLK/4
3h	MCLK/8
4h	MCLK/16
5h	MCLK/32
6h	MCLK/64
7h	MCLK/128

APPLICATION NOTE: The ARC clock must be configured to be equal to or faster than the JTAG_CLK.

M

This bit controls Master/Slave JTAG multiplexing. When this bit is 0 (default), the JTAG port is configured as a slave. When this bit is 1, the JTAG port is configured as a Master.

44.6 JTAG Port Signal Interface Description

The signal pins are defined in the JTAG Interface section located in the Pin Configuration chapter.

The JTAG_CLK input is the clock that drives the JTAG interface. It is asynchronous to other clocks on-chip.

The TMS input is sampled on each rising edge of JTAG_CLK, and governs the transitions among the 16 states of the state machine (TAP) that controls the transfer of data.

The TDI input is the serial data input, shifted in during the Shift-IR and Shift-DR states of the TAP. It is sampled on rising edges of JTAG_CLK.

The TDO output is the serial data output. It is presented on falling edges of JTAG_CLK, 1/2 clock before each input shift, to provide setup and hold time to the next JTAG controller in the chain. The final TDO output pin, after all on-chip chaining (Figure 44-2) is held in high-impedance mode (floating) except when valid data is being presented. The enabled/disabled state of the pin is also changed on falling edges of JTAG_CLK.

The JTAG_RST# input provides the Async JTAG RESET. Note that the reset state of the JTAG port is only local to the JTAG port: its effect is to keep the JTAG port in an idle state and to disengage it from the rest of the system, so that it does not affect other on-chip logic in this state.

44.7 Power, Clocks and Reset

See Section 46.11, "JTAG Interface Timing," on page 628 power on sequence and reset timing.

44.7.1 POWER DOMAINS

The JTAG block is powered by VTR.

44.7.2 CLOCKS

The JTAG port runs internally from the externally-provided JTAG_CLK clock pulses only. There is no requirement for JTAG_CLK to be constantly running.

The following JTAG Registers interface to the ARC Island block (as illustrated in Figure 44-2) for access to registers and memory: STATUS Register (8h), TRANSACTION COMMAND Register (9h), & ADDRESS Register (Ah), DATA Register (Bh). There is a clock relationship required between JTAG_CLK and the ARC Core clock frequency. JTAG_CLK may be asynchronous, but it must be slower than 1/2 the frequency of the ARC Core clock. In practical terms, then, JTAG_CLK should be selected to be nominally 1/4 of the minimum Core clock frequency. See APPLICATION NOTE: on page 132.

APPLICATION NOTE: The Ashling JTAG interface box is not documented to operate any slower than JTAG_CLK = 1MHz, therefore the Core must be running at 4MHz in order to run the ARC debugger through the JTAG interface using the Ashling interface.

Stopping the Core clock disables the JTAG port for debugging purposes. It does not affect IDCODE or BYPASS operation.

See Section 46.11, "JTAG Interface Timing," on page 628 for the maximum frequency f_{clk} on the JTAG_CLK pin to access a JTAG Registers other than STATUS Register (8h), TRANSACTION COMMAND Register (9h), & ADDRESS Register (Ah), DATA Register (Bh).

44.7.3 RESET

The ARC JTAG block has two resets: Async JTAG RESET by its JTAG_RST# input and Sync JTAG RESET by JTAG protocol.

44.7.3.1 Async JTAG RESET

The JTAG_RST# pin provides the Async JTAG RESET to the JTAG Registers. The JTAG_RST# pin has an active low, asynchronous assertion and a synchronous de-assertion. The JTAG Registers will be reset asynchronously (and immediately) upon the active low JTAG_RST# assertion. Once the JTAG_RST# pin has been de-asserted, a delay of three JTAG_CLKs is required in order to access the JTAG Registers; i.e., the JTAG Registers will remain in reset for three clocks following the synchronous JTAG_RST# pin de-assertion. See Section 46.11, "JTAG Interface Timing," on page 628.

APPLICATION NOTE: After asserting and de-asserted the JTAG_RST# pin, a Sync JTAG RESET can be applied before starting to access the JTAG Registers (to meet the JTAG_RST# synchronous de-assertion requirement.

Note 44-2 JTAG registers, in particular the JTAG Test Mode Data Registers, are set to their initial values by the assertion of the JTAG_RST# pin, not the VTR Power On Reset. JTAG_RST# must be held low while the MEC1632 is powering up so the registers can be set to their proper default values. If JTAG_RST# is high during power up, the JTAG Test Mode Data Registers may be set to unpredictable values, which may trigger unwanted test modes.

© 2012-2018 Microchip Technology Inc.

APPLICATION NOTE: Care should be taken during VTR power up to insure that JTAG_RST# is asserted for a longer time then the VTR rise time due to capacitive loading. See Section 44.2.2, "Slave Select Timing," on page 580 for timing requirement.

Figure 44-10 illustrates an example of the Async JTAG RESET function. Refer to the “Async JTAG RESET Functional Description” table below the figure for details.

FIGURE 44-10: JTAG_RST# FUNCTIONAL EXAMPLE

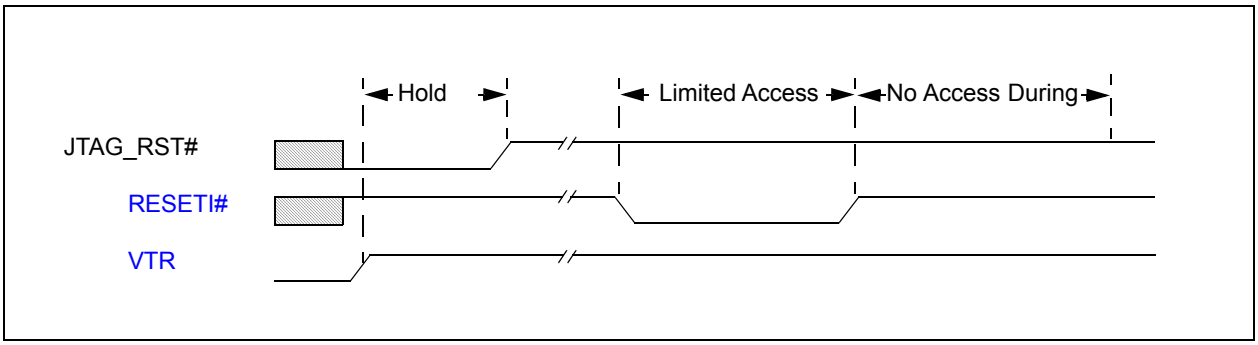


TABLE 44-6: Async JTAG RESET FUNCTIONAL DESCRIPTION

RESETI#	VTR	JTAG_RST#	Description
X	0	X	System unpowered
	0->1	1	Undefined
0		0	System powered, device held in reset
1->0	1	1	Limited JTAG access; e.g., JTAG register access only, JTAG Master held in reset
1	0->1	0 (Hold)->1	JTAG_RST# hold time is required as defined in Section 46.11, "JTAG Interface Timing," on page 628
0->1	1	1	No JTAG access allowed for the equivalent of the JTAG_RST# hold time (Section 46.11).
1		0	Normal Operation, JTAG disabled
		1	Normal Operation, JTAG enabled

44.7.3.2 Sync JTAG RESET

It can also be reset synchronously by a JTAG_CLK / TMS sequence, in accordance with the JTAG standard. A series of 5 successive JTAG_CLK rising edges, with TMS held high throughout, will accomplish this from any state.

The ARC JTAG port, upon entering its Reset state, will be prepared to accept an Instruction or Data transfer. It will also be disengaged from external circuitry, allowing it to operate normally.

The initial contents of the Instruction Register are the IDCODE command (Ch). If a Data transfer is performed first after Reset, without an preceding Instruction transfer, then the IDCODE value will be loaded into its 32-bit shift register and presented serially, after which will appear the bits shifted in from TDI.

The initial contents of the Data registers are as listed in Table 44-7 on page 592.

44.8 Interrupts

There are no interrupts assigned to the ARC JTAG block. Control of the processor is performed by monitoring, setting and clearing the H bit (Halt) in the Aux register STATUS32, and by register manipulations while the processor is halted.

However, any interrupt or other event that can be triggered by accessing registers (Core, Aux or Memory-Mapped) can be triggered through the JTAG port.

© 2012-2018 Microchip Technology Inc.

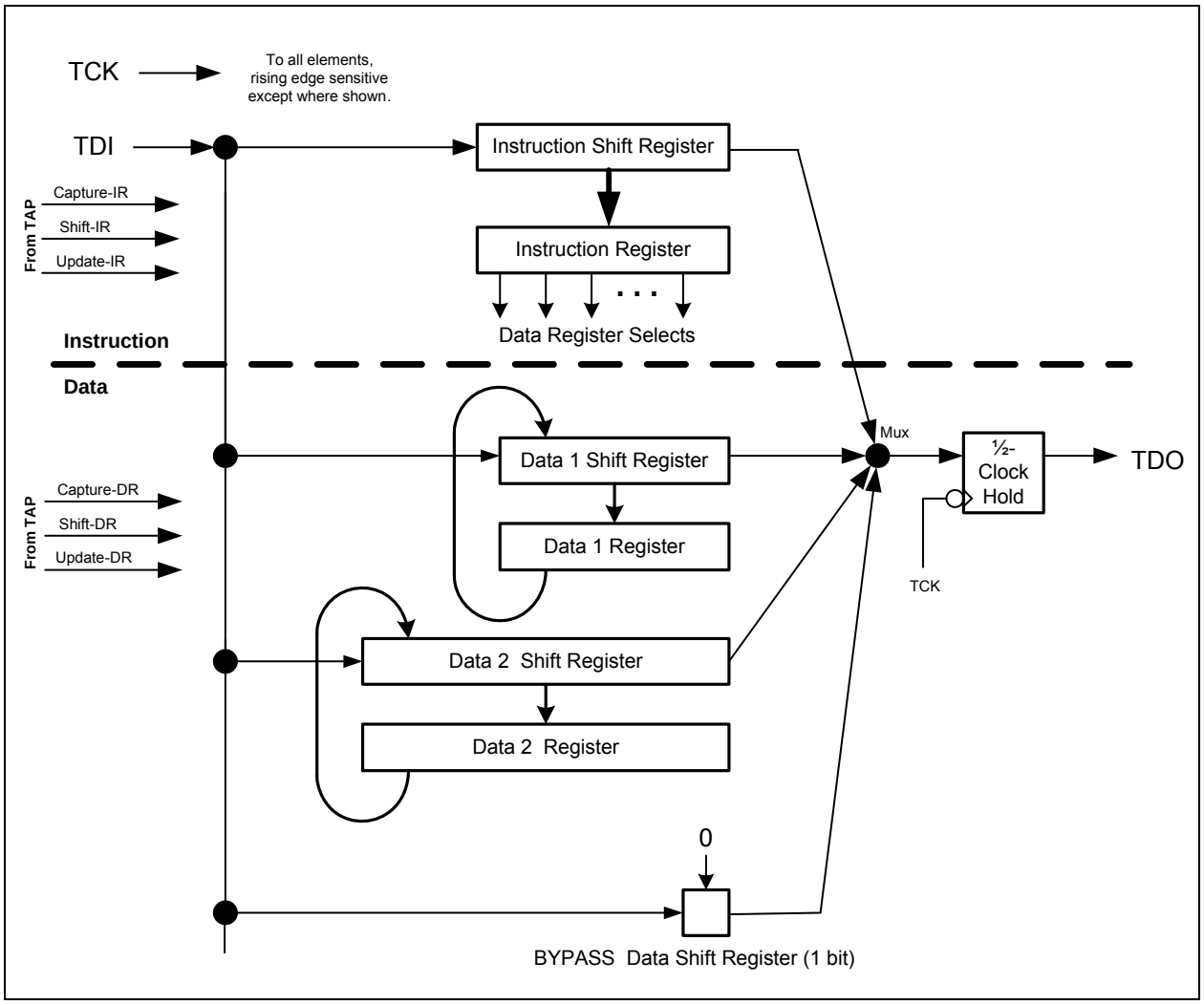
44.9 JTAG Background

The following is a simplified description, intended to provide background for the ARC JTAG port. For full details, see the JTAG specification (IEEE Standards 1149.1 and 1149.1b).

44.9.1 INTERNAL STRUCTURE

A JTAG port operates by transferring information serially into and out of an Instruction register and one or more Data registers. These registers are connected in parallel with each other, and can be of arbitrary length. See Figure 44-11.

FIGURE 44-11: STRUCTURE OF A JTAG PORT (SIMPLIFIED)



The protocol for shifting information makes a distinction between an Instruction transfer (to/from a single Instruction register) and a Data transfer (to/from one of several Data registers). The Instruction register is handled separately because it selects which specific Data register is accessed by subsequent Data transfers.

In daisy-chained JTAG controllers, the Instruction registers form one chain, and the currently-selected set of Data registers in each JTAG controller combine to form a second chain. To shorten the Data chain when not all JTAG controllers are of interest, a mandatory one-bit Data register called BYPASS is provided. There is no bypassing for the Instruction chain, so its full length must be shifted as each new instruction is transferred anywhere. Selecting the BYPASS Data register is the equivalent of a No-Operation instruction for a JTAG controller, and this instruction is always defined as a '1' in all Instruction register bits.

© 2012-2018 Microchip Technology Inc.

Each entity called a “Register” actually consists of two parts: the Register itself, and an associated Shift Register which connects to TDI and TDO. The Register may load from, and/or source information in parallel to, the Shift Register. These two parts are the same length, meaning that (for example) a 5-bit Register will be associated with a 5-bit Shift Register.

The Instruction register and the Data registers respond to decoded state signals from the TAP Controller sub-block (Section 44.9.2), which represent sub-steps of a transfer. The sub-steps they perform are **Capture**, which loads the shift register in parallel, **Shift**, which shifts information in from TDI and out on TDO, and **Update**, which writes information from the Shift Register in parallel. The **Capture-IR**, **Shift-IR** and **Update-IR** controls affect only the Instruction register. The **Capture-DR**, **Shift-DR** and **Update-DR** controls affect only the Data register that is currently selected by the contents of the Instruction register.

44.9.2 TAP CONTROLLER AND PROTOCOL

The JTAG protocol is driven by the level of the TMS (Test Mode Select) input pin at each rising edge of the JTAG_CLK clock. This is the responsibility of the TAP Controller section of the JTAG controller, which performs state transitions as illustrated in the state diagram in Figure 44-12. States whose names end with “IR” affect the Instruction register (the rightmost column of states in Figure 44-12), and those ending with “DR” affect a Data register (the middle column in Figure 44-12). Note that the TMS signal goes in parallel to all JTAG ports in a chain, so they are always in the same protocol state. The sequence of accessing any register is as follows:

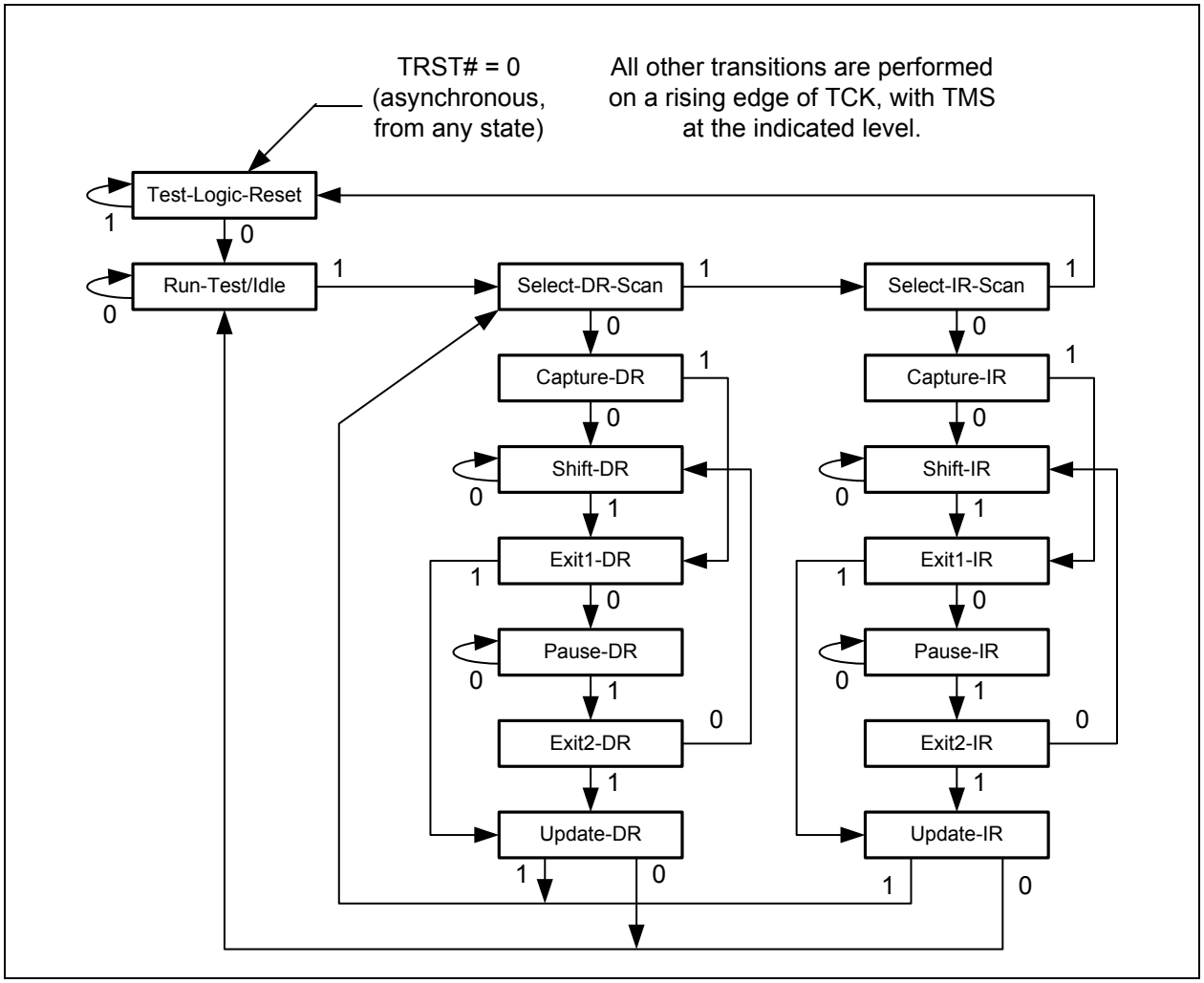
- Capture (IR or DR), which loads a shift register from its source in preparation for shifting it out. In the case of the Instruction register, this is a fixed value, and not the previous contents of the Instruction register. In the case of the BYPASS Data register, this is a fixed ‘0’ value. The Capture state is transitory, being present for only one JTAG_-CLK cycle, once per transfer.
- Shift (IR or DR), which shifts the Captured information in the Shift Register out on the TDO pin while also shifting information in from the TDI pin. The registers (by convention) shift from left to right, so the least-significant bit of a value is transferred first. This state may be held arbitrarily (holding TMS=0) to shift as many bits as desired.
- Update (IR or DR), which loads a Register from its Shift Register after the shifting has completed. The Update state is transitory, being present for only one JTAG_CLK cycle, once per transfer.

There is also a Pause state (IR or DR) which may be used to exit and re-enter the Shift state without terminating the transfer in progress. This state may be held (TMS=0) in order to delay for any desired number of JTAG_CLK cycles.

Outside of Instruction or Data transfers, there are two states which may be entered and held. These are shown in the leftmost column in Figure 44-12.

- The Test-Logic-Reset state holds the JTAG logic in its reset state. This re-initializes the registers that are internal to the JTAG logic. This state is entered asynchronously by assertion of JTAG_RST# low, and it can be seen in Figure 44-12 that, from any other state, this state will be entered by 5 successive JTAG_CLK cycles with TMS held to ‘1’.
- Run-Test/Idle holds JTAG logic idle, but not reset, between transfers.

FIGURE 44-12: TAP CONTROLLER STATE DIAGRAM



44.9.3 INTERFACE TIMING EXAMPLE

Figure 44-13 illustrates the timing relationship between data shifting and the TAP Controller's Shift states, using a 1-bit Data register as an example. (This is in fact the exact situation when the BYPASS Data register is selected: refer to FIGURE 44-11: on page 588.)

The TAP Controller changes states on each rising edge of JTAG_CLK, traversing the state table in Figure 44-12 as directed by the TMS input signal from the external interface.

Previous to the waveform in Figure 44-13, the TAP Controller has already passed through a Capture-DR state, so the 1-bit Shift Register has been pre-loaded with a "Capture Value", either from its associated parallel Register or from another source. (For the BYPASS register, this would be a fixed '0'.)

At the first rising edge of JTAG_CLK in Figure 44-13, the Shift-DR state is being entered. As yet, no valid data needs to be present on TDI or TDO.

At the first falling edge of JTAG_CLK, while the Shift-DR state is active, the TDO pin begins presenting the least-significant bit of the Shift Register (the only bit, in this example), which is holding the Captured Value. At about this time also, the external interface will drive TDI to the desired new state for this Data register.

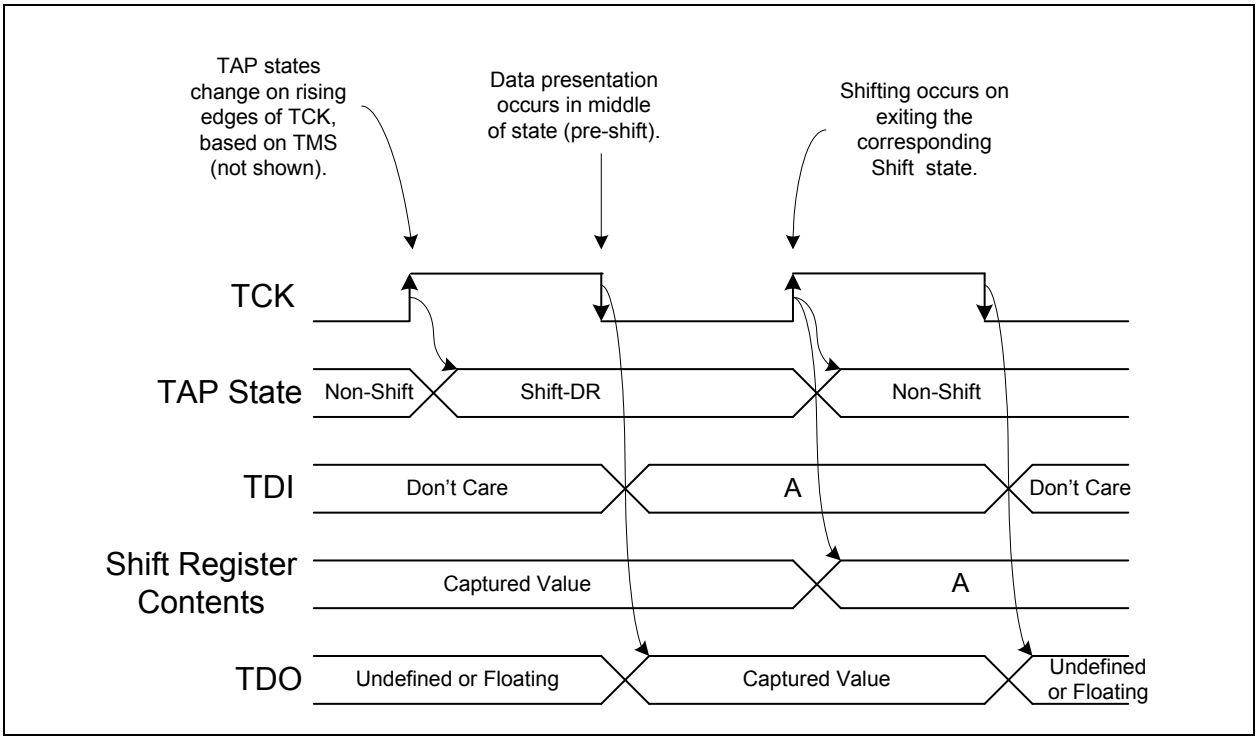
© 2012-2018 Microchip Technology Inc.

At the next rising edge of JTAG_CLK, the Shift-DR state is exited, and that same clock edge is used to actually perform the commanded shift. The TDI value “A” is shifted into the Shift Register. This same rising edge of JTAG_CLK is used by the external interface to shift in the Captured Value from TDO. The TDO output does not change yet, because it is held by a 1/2 clock delay stage (see FIGURE 44-11: on page 588), providing hold time for the external interface.

On the next falling edge, the TDO output changes. Since the Shift state is no longer present, TDO is not required at this time to present valid data, and in fact for an off-chip connection it is required to float at this time.

After this timing diagram completes, the TAP machine will continue to an Update-DR state, at which time the value A, now present in the Shift Register, will be written to its destination. (In the specific case of the BYPASS register, there is no destination, and that step will do nothing.)

FIGURE 44-13: TIMING ILLUSTRATION: 1-BIT DATA REGISTER



44.10 Registers

The ARC JTAG Port is defined in the ARC 600 External Interfaces Reference Manual, Chapter 2.

There are no JTAG registers accessible in any EC or AHB memory space. JTAG registers are accessible only through the JTAG pins themselves (Section 44.10.1, "Instruction Register," on page 591).

44.10.1 INSTRUCTION REGISTER

The Instruction Register is four bits wide. It selects among the implemented Data Registers as listed in Table 44-7. When the Tap Controller is placed into the Test-Logic-Reset state, the Instruction register is initialized to Ch, selecting the IDCODE Data Register.

Registers marked as MCHP reserved must not be modified. Modifications may lead to unpredictable and unwanted behavior.

TABLE 44-7: ARC JTAG INSTRUCTION REGISTER ENCODINGS

Instruction Register Contents	Data Register Selected	Function of Data Register	Width (Bits)	State on JTAG Reset (Hex)
0h	(Reserved: EXTEST)	Not implemented, but reserved as required by JTAG standard.	32	0000_0000h
1h	(Reserved: SAMPLE/PRELOAD)	Not implemented, but reserved as required by JTAG standard.	32	0000_0000h
2h	RESET TEST	RESET TEST Register (2h)	32	0000_0000h
3h	TEST - MCHP Reserved		32	0000_0000h
4h	MCHP Reserved	Reserved for future use.)	32	0000_0000h
5h	MCHP Reserved	MCHP Reserved	32	0000_0000h
6h	(Reserved)	(Reserved for future use.)	32	0000_0000h
7h	(Reserved)	(Reserved for future use.)	32	0000_0000h
8h	STATUS	STATUS Register (8h) Status of Current Debugger Transaction (Read-Only)	4	undefined (based on bus status)
9h	TRANSACTION COMMAND	TRANSACTION COMMAND Register (9h) Initiates / Specifies a Debugger Transaction	4	3
Ah	ADDRESS	ADDRESS Register (Ah) Address of a Debugger Transaction	32	0000_0000h
Bh	DATA	DATA Register (Bh) Data In / Data Out for Debugger Transactions	32	Out = 0000_0000h In = undefined
Ch	IDCODE	IDCODE Register (Ch) JTAG Standard IDCODE Register (Capture = Read-Only fixed value)	32	2000_24B1h
Dh	MCHP Reserved		32	0000_0000h
Eh	(Reserved)	Reserved for future use.	32	0000_0000h
Fh	BYPASS	BYPASS Register (Fh) JTAG Standard BYPASS Register (Capture = Read-Only ‘0’)	1	0

44.10.2 JTAG DEBUG DATA REGISTERS

Note 44-3 Unfortunately, ARC names one of its JTAG Debug Data registers “DATA”. To avoid confusion, while maintaining the terminology in both ARC and JTAG documentation, the term “Data register” will refer to any of the JTAG Data registers, and the term “DATA register” (all upper-case) will refer to the specific JTAG Data register that is selected by Instruction Register = B. See Section 44.10.2.2, "DATA Register (Bh)," on page 593.

The Debug Data Register set of the ARC JTAG Port provide the means for an external JTAG-connected debugger system to monitor and control the execution of a program. Using the JTAG Data registers ADDRESS, DATA, TRANSACTION COMMAND and STATUS, the debugger can perform “transactions” to read or write:

- Any Aux Register, giving it the ability to start, halt or step a program, and alter the PC and/or program status
- Any addressable memory or I/O location, as an aligned 32-bit value
- Any Core Register, if the processor is in a halted state

To write to a specific register or a memory location, the debugger will place the desired register number or memory address into the ADDRESS register, place the value to be written into the DATA register, and then trigger the transfer by placing the direction and addressing space (Core register / Aux register / Memory) into the TRANSACTION COMMAND register. It will then read the STATUS register until it indicates that the transaction is finished.

To read from a specific register or memory location, the debugger will place the desired register number or memory address into the ADDRESS register, and trigger the transfer by placing the direction and addressing space (Core register / Aux register / Memory) into the TRANSACTION COMMAND register. It will then read the STATUS register until it indicates that the transaction is finished, and read the DATA register to access the value.

© 2012-2018 Microchip Technology Inc.

Optimizations are possible in repeated accesses, because of the actions of the ADDRESS and DATA registers, as described in Section 44.10.2.1 and Section 44.10.2.2.

44.10.2.1 ADDRESS Register (Ah)

The ADDRESS register is a 32-bit register which receives from the debugger either a Core Register number, an Aux Register number or an address in the Memory space (memory or I/O).

Note:	As a memory address, the low-order 2 bits of the ADDRESS register are ignored (assumed by hardware to be 00), and a full 32-bit value is referenced at that location. There is no way for the debugger to specify a smaller width of data, and so a write to a single byte (for example) is performed using a read transaction followed by a write transaction, preserving the values of the unaffected bytes.

After use, the ADDRESS register automatically increments, by 1 if a register was accessed, and by 4 if a memory location was accessed. Therefore, as long as the JTAG TAP Controller is not brought to the Test-Logic-Reset state between accesses, it is not necessary to provide a new ADDRESS register value between transactions involving successive registers or memory locations. (The Test-Logic-Reset state must be avoided because it resets the value of the ADDRESS register.)

TABLE 44-8: ADDRESS REGISTER

INSTRUCTION REGISTER CONTENTS	Ah					32 bits	**REGISTER SIZE**	
POWER	VTR					0000_0000h	Async JTAG RESET **OR** Sync JTAG RESET **DEFAULT**	
BIT	**BIT31**	**BIT30**	**BIT29**	**...**		**BIT2**	**BIT1**	**BIT0**
JTAG TYPE	-	-	-	-	-	-	-	-
BIT NAME	Address[31:0]							

44.10.2.2 DATA Register (Bh)

The DATA register is a 32-bit register which is the ARC JTAG Port's portal for data values that are being read or written by a transaction. When writing to a register or memory, the DATA register will be set up by the debugger before the transaction is triggered. When reading from a register or memory, the DATA register will be read by the debugger as the last step of the transaction. See Note 44-2 on page 586.

The DATA register is not affected at the end of a write transaction, so (for example) to fill successive locations with the same value it is not necessary to provide it again, as long as the Test-Logic-Reset of the JTAG TAP Controller is not entered (which would clear it).

TABLE 44-9: DATA REGISTER

INSTRUCTION REGISTER CONTENTS	Bh					32 bits	**REGISTER SIZE**	
POWER	VTR					0000_0000h	Async JTAG RESET **OR** Sync JTAG RESET **DEFAULT**	
BIT	**BIT31**	**BIT30**	**BIT29**	**...**		**BIT2**	**BIT1**	**BIT0**
JTAG TYPE	-	-	-	-	-	-	-	-
BIT NAME	Data[31:0]							

44.10.2.3 TRANSACTION COMMAND Register (9h)

The TRANSACTION COMMAND register is written by the debugger to trigger a transaction. It is a 4-bit register, which is written with one of the values in Table 44-11 to specify the direction and addressing space of the transaction.

TABLE 44-10: DATA REGISTER

INSTRUCTION REGISTER CONTENTS	9h		4 bits	**REGISTER SIZE**
POWER	VTR		0h	Async JTAG RESET **OR** Sync JTAG RESET **DEFAULT**
BIT	**BIT3**	**BIT2**	**BIT1**	**BIT0**
JTAG TYPE	-	-	-	-
BIT NAME	Command[3:0]			

TABLE 44-11: TRANSACTION COMMAND REGISTER ENCODINGS

Encoding (Binary)	Transaction Type
0000	Write to Memory space
0001	Write to a Core register
0010	Write to an Aux register
0011	No Operation
0100	Read from Memory space
0101	Read from a Core register
0110	Read from an Aux register
0111	(obsolete Write form)
1000	(obsolete Read form)
(other)	Reserved

© 2012-2018 Microchip Technology Inc.

44.10.2.4 STATUS Register (8h)

The STATUS register is a 4-bit read-only register. It is read by the debugger to determine when a transaction has completed internally, and when the next transaction may be started. It also provides additional status information useful to the debugger.

TABLE 44-12: STATUS REGISTER

INSTRUCTION REGISTER CONTENTS	8h		4 bits	**REGISTER SIZE**
POWER	VTR		0h	Async JTAG RESET **OR** Sync JTAG RESET **DEFAULT**
BIT	**BIT3**	**BIT2**	**BIT1**	**BIT0**
JTAG TYPE	R	R	R	R
BIT NAME	-PC	-RD	FL-	ST-

(ST): STALLED

1 = The current transaction is stalled (busy)

0 = The current transaction is not stalled (not busy)

(FL): FAILURE

1 = The transaction has failed

0 = The transaction has not failed

A transaction will fail if it attempts to access a Core register while the processor is running. Bus errors should also set this bit.

(RD): READY

1 = The transaction is finished (ready)

0 = The transaction is not finished

(PC): PC_SEL

This bit has no direct hardware effect. It displays the state of the PC_SEL signal, which is bit 0 of the write-only Aux register PCPORT (Aux Register #24h). This bit is initialized to ‘1’ on a processor reset, and is used internally by the debugger system as a means to communicate configuration information.

44.10.3 JTAG STANDARD DATA REGISTERS

44.10.3.1 IDCODE Register (Ch)

This is a 32-bit read-only register containing the ID value that serves to identify the ARC JTAG Port as belonging to an ARC600 core in a component containing one processor (Table 44-13).

IDCODE registers are required to conform to the JTAG standard and they include an 11-bit Manufacturer ID number.

TABLE 44-13: IDCODE REGISTER

INSTRUCTION REGISTER CONTENTS	Ch			32 bits			**REGISTER SIZE**	
POWER	VTR			2000_24B1h			Async JTAG RESET **OR** Sync JTAG RESET **DEFAULT**	
BIT	**BIT31**	**BIT30**	**BIT29**	**...**		**BIT2**	**BIT1**	**BIT0**
JTAG TYPE	R	R	R	R	R	R	R	R
BIT NAME	IDCODE[31:0]							

44.10.3.2 BYPASS Register (Fh)

The BYPASS register consists only of a 1-bit shift register cell. The Capture-DR state clears it to '0' when selected. The Update-DR state does nothing.

The function of this register is to provide the minimum amount of delay (one bit of '0') when other JTAG ports on the chain are being exercised.

TABLE 44-14: BYPASS REGISTER

INSTRUCTION REGISTER CONTENTS	Fh	1 bit	**REGISTER SIZE**
POWER	VTR	1000_24B1h	Async JTAG RESET **OR** Sync JTAG RESET **DEFAULT**
BIT	**BIT0**		
JTAG TYPE			
BIT NAME	BYPASS		

© 2012-2018 Microchip Technology Inc.

44.10.4 JTAG TEST MODE DATA REGISTERS

JTAG Test Registers are 32-bit read/write registers that are used for test functions. These registers are always available to the JTAG port, even if access to the other JTAG registers is blocked.

44.10.4.1 RESET TEST Register (2h)

The RESET TEST Register is a 32-bit register used to explicitly control reset functions inside the MEC1632. The default for this register is 0000_0000h.

TABLE 44-15: RESET TEST REGISTER

INSTRUCTION REGISTER CONTENTS	2h			32 bits			**REGISTER SIZE**	
POWER	VTR			0000_0000h			Async JTAG RESET **DEFAULT**	
BIT	**Bits 31**	**Bit 30**	**Bit 29**	**Bit 28**	**Bit 27**	**Bit 26**	**Bit 25**	**Bit 24**
JTAG TYPE	R/W	R	R	R	R	R	R	R
BIT NAME	GANG_EN	Test	Reserved					
BIT	**Bits 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
JTAG TYPE	R	R	R	R	R	R	R	R
BIT NAME	Reserved							
BIT	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
JTAG TYPE	R	R/W	R	R	R	R	R/W	R/W
BIT NAME	Reserved	Test	Test	Test	Test	Test	Test	Test
BIT	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
JTAG TYPE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
BIT NAME	Test	Test	Test	Test	POR EN	VTR POR	VCC POR	ME

ME

Mass Erase. If this bit is '1' when the internal VTR Power On Reset signal transitions from '0' to '1', the Embedded Flash Subsystem will enter the Emergency Mass Erase mode, which will erase the entire Flash whether or not the Boot_-JTAG_Block bit in the Embedded Flash Initialization Register is set.

VCC POR

Asserts VCC Power On Reset: When the VCC POR active low bit is asserted '0' while the field POR EN in this register is '1', forces a VCC Power On Reset. When the VCC POR active low bit de-asserted '1', the VCC POR circuitry returns to its normal state.

VTR POR

Asserts VTR Power On Reset: When the VTR POR active low bit is asserted '0' while the field POR EN in this register is '1', forces a VTR Power On Reset. When the VTR POR active low bit de-asserted '1', the VCC POR circuitry returns to its normal state.

POR EN

Power On Reset Enable. When '1', the reset functions controlled by VCC POR and VTR POR are enabled. When '0', the VCC POR and VTR POR fields in this register have no effect on the POR circuitry.

TEST

All TEST bits should be set to ‘0’ when writing this register.

GANG_EN

When the GANG_EN bit is asserted '1' and the ME bit (also in the RESET TEST Register) is '0' when the internal Power On Reset transitions from '0' to '1', the Gang Programmer Interface is enabled. In this mode, the Embedded Flash Subsystem is permitted to complete its initialization sequence, and the EC is held in Reset. See Section 43.0, "Gang Programmer Interface," on page 569 for details regarding the operation of, and the multiplexing of pins in the Gang Programmer Interface.

44.10.4.2 TEST REGISTER 4/Reset Register (Dh)

The RESET TEST Register is a 32-bit register used to explicitly control reset functions inside the MEC1632. The default for this register is 0000_0000h.

TABLE 44-16: TEST REGISTER 4/RESET REGISTER

INSTRUCTION REGISTER CONTENTS	Dh			32 bits			**REGISTER SIZE**	
POWER	VTR			0000_0000h			Async JTAG RESET **DEFAULT**	
BIT	**BIT31**	**BIT30**	**BIT29**	**BIT28**	**BIT27**	**BIT26**	**BIT25**	**BIT24**
JTAG TYPE	R/W	R/W	R	R	R	R/W	R/W	R
BIT NAME	Test		Reserved			Test	Test	Test
BIT	**BIT23**	**BIT22**	**BIT21**	**BIT20**	**BIT19**	**BIT18**	**BIT17**	**BIT16**
JTAG TYPE	R	R	R/W	R/W	R/W	R/W	R/W	R
BIT NAME	Test	Test	Test	Test	Test	Test	Test	Rsrvd
BIT	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
JTAG TYPE	R/W	R	R	R	R/W	R/W	R/W	R/W
BIT NAME	Test	Reserved		Test	Test	Test		
BIT	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
JTAG TYPE	R	R	R	R/W	R/W	R/W	R/W	R/W
BIT NAME	Test	Test	Test	Test	Test	Test	Test_ XNOR_ En	ARC_ Fast_ Reset

ARC_FAST_RESET

If this bit is ‘1b’, the reset going to the ARC processor and select peripherals is reduced from its nominal 20ms duration. If this bit is ‘0b’, the ARC reset is stretched by the nominal delay.

TEST_XNOR_EN

If this bit is ‘1b’, the Device-Under-Test XNOR chain test mode is enabled. If this bit is ‘0b’, the XNOR mode is disabled. See Section 44.11, "XNOR Chain," on page 599.

Note:	Once the XNOR chain is enabled, a power cycle is required to re-establish JTAG operation.

TEST

All TEST bits should be set to ‘0; when writing this register.

© 2012-2018 Microchip Technology Inc.

44.10.5 JTAG STANDARD PORT DISCOVERY

This section provides information that is not unique to ARC, but is part of the JTAG standard, and is provided for information.

The Discovery process will identify each JTAG controller that has an IDCODE register. Part of what needs to be derived is the length of the Instruction register in each of the JTAG ports. If this cannot be derived from the IDCODE values, or if some JTAG ports do not have an IDCODE register, then the missing lengths must be provided by other means.

In the Test-Logic-Reset state, a JTAG port is required to initialize its Instruction register to select the IDCODE Data register if present, or if it is not present, then to select the BYPASS Data register.

The IDCODE Data register:

- Must be exactly 32 bits in length
- Must have '1' in its first (least-significant) bit
- Must not have the pattern 000011111111 (FFh) in its first (least-significant) 12 bits.
- Will contain a completely definitive port identification, because 11 bits of it are a Manufacturer ID number assigned by the JEDEC standards organization.

A BYPASS Data register access will initialize its 1-bit shift register to '0' at the Capture-DR state, effectively making the BYPASS register appear to be 1-bit read-only '0'.

Discovery, therefore, consists of the external JTAG host doing the following:

- Place the chain of JTAG controllers into the Test-Logic-Reset state.
- Do a Data register access, without an Instruction register access first.
  - This Data access will shift in 8 bits of ones, followed by all zeroes for the duration of the discovery phase.
- While shifting, examine the data appearing on TDO for IDCODE values.
  - A '0' indicates a JTAG port that has no IDCODE register. Collect only this bit, and note that the JTAG port exists. Start looking for an IDCODE value at the next bit.
  - A '1' indicates that an IDCODE register is coming. Collect this bit and the next 31 bits to identify the JTAG port. If, however, the value seen is 00h0000FF, then this is ensured to be the value provided originally on TDI, and indicates the end of the chain.
    XXXXXXX

44.11 XNOR Chain

44.11.1 OVERVIEW

The XNOR Chain test mode allows users to confirm that all MEC1632 pins are in contact with the motherboard during assembly and test operations. The XNOR Chain test mode is enabled and disabled through the JTAG interface, using bit Test_XNOR_En in JTAG TEST REGISTER 4/Reset Register (Dh).

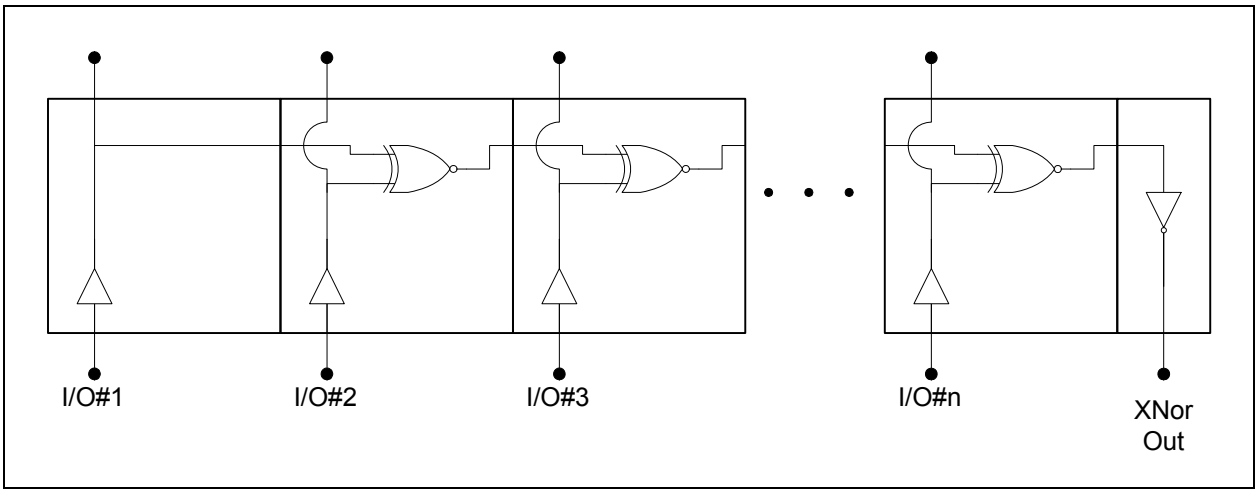
An example of an XNOR Chain test structure is illustrated below in Figure 44-14. When the XNOR Chain test mode is enabled all pins except for the Excluded Pins shown in Section 44.11.2 are disconnected from their internal functions and forced as inputs to the XNOR Chain. This allows a single input pin to toggle the XNOR Chain output if all other input pins are held high or low. The XNOR Chain output is the nRESET_OUT pin.

The tests that are performed when the XNOR Chain test mode is enabled require the board-level test hardware to control the device pins and observe the results at the XNOR Chain output pin; e.g., as described in Section 44.11.3, "Test Procedure," on page 600.

44.11.2 EXCLUDED PINS

The following pins are XNOR Chain Excluded Pins: POWER PLANE pins, VR_CAP, nRESET_OUT, and JTAG_RST#.

FIGURE 44-14: XNOR CHAIN TEST STRUCTURE



44.11.3 TEST PROCEDURE

44.11.3.1 Setup

1. Connect the VSS and AGND pins to ground.
2. Connect the VBAT and VTR pins to an unpowered 3.3V power source.
3. Connect an oscilloscope or voltmeter to the nRESET_OUT pin.
4. All other pins should be tied to ground.

Warning: Ensure power supply is off during Setup.

44.11.3.2 Testing

1. Turn on the 3.3V power source.
2. Enable the XNOR Chain through the JTAG interface (Test_XNOR_En in JTAG TEST REGISTER 4/Reset Register (Dh)). Note that at this point all inputs to the XNOR Chain are low and the output on the nRESET_OUT pin is high (refer to the Initial Configuration row in Table 44-17, "Toggling Inputs in Descending Pin Order").
3. Bring the highest numbered pin (N) high, where N is the number of pins to be tested as described in Note 44-4. The output on the nRESET_OUT pin should toggle (refer to Step 1 in Table 44-17.
4. In descending pin order successively bring each input high. As shown in Table 44-17 the nRESET_OUT pin toggles after each step. Continue until all inputs are high. The output on the nRESET_OUT pin is high (refer to the Final Configuration in Table 44-17).
5. The current state of the chip is now represented by the Initial Configuration row in Table 44-18, "Toggling Inputs in Ascending Pin Order".
6. Each input should now be brought low, starting at pin one (Step N+1) and continuing in ascending pin order until all inputs are low. The output on the nRESET_OUT pin is high (refer to the Final Configuration in Table 44-18.
7. Exit the XNOR Chain Test Mode by cycling VTR power.

© 2012-2018 Microchip Technology Inc.

TABLE 44-17: TOGGLING INPUTS IN DESCENDING PIN ORDER

	Pin Number (Note 44-4)							nRESET_OUT
	N	N - 1	N - 2	N - 3	N - 4	...	1	
Initial Configuration	L	L	L	L	L	L	L	H
Step 1	H	L	L	L	L	L	L	L
Step 2	H	H	L	L	L	L	L	H
Step 3	H	H	H	L	L	L	L	L
Step 4	H	H	H	H	L	L	L	H
Step 5	H	H	H	H	H	L	L	L
...	H	H	H	H	H	…	L	…
Step N-1	H	H	H	H	H	H	L	L
Final Configuration	H	H	H	H	H	H	H	H

TABLE 44-18: TOGGLING INPUTS IN ASCENDING PIN ORDER

	Pin Number (Note 44-4)							nRESET_OUT
	1	2	3	4	5	...	N	
Initial Configuration	H	H	H	H	H	H	H	H
Step N+1	L	H	H	H	H	H	H	L
Step N+2	L	L	H	H	H	H	H	H
Step N+3	L	L	L	H	H	H	H	L
Step N+4	L	L	L	L	H	H	H	H
Step N+5	L	L	L	L	L	H	H	L
...	L	L	L	L	L	…	H	…
Step N+(N-1)	L	L	L	L	L	L	H	L
Final Configuration	L	L	L	L	L	L	L	H

Note 44-4 pin numbers in these tables represent the number of pins to be tested and do not include the pins listed in Section 44.11.2, "Excluded Pins," on page 600.

45.0 ELECTRICAL SPECIFICATIONS

45.1 Maximum Ratings*

*Stresses exceeding those listed could cause permanent damage to the device. This is a stress rating only and functional operation of the device at any other condition above those indicated in the operation sections of this specification is not implied.

Note:	When powering this device from laboratory or system power supplies, it is important that the Maximum Ratings not be exceeded or device failure can result. Some power supplies exhibit voltage spikes on their outputs when the AC power is switched on or off. In addition, voltage transients on the AC power line may appear on the DC output. If this possibility exists, it is suggested that a clamp circuit be used.

45.1.1 MAXIMUM THERMAL RATINGS

TABLE 45-1: MAXIMUM THERMAL RATINGS

Parameter	Maximum Limits
Operating Temperature Range (Commercial)	0^{o} C to $+70^{o}$ C
Storage Temperature Range	-55^{o} C to $+150^{o}$ C
Lead Temperature Range	Refer to JEDEC Spec J-STD-020B

45.1.2 MAXIMUM SUPPLY VOLTAGE RATINGS

TABLE 45-2: POWER SUPPLY RATINGS

Symbol	Parameter	Maximum Limits
VBAT	Battery Backup Power Supply	4V
VTR	VTR Power Supply	4V
AVTR_ADC	Analog VTR Supply	4V
VREF_ADC	ADC Voltage Reference Pin	4V

45.1.3 MAXIMUM I/O VOLTAGE RATINGS

Parameter	Maximum Limits
Voltage with respect to ground on any signal pin without backdrive protection	-Determined by Power Supply I/O Buffer (Note 45-1)
Voltage with respect to ground on any signal pin without 5.0V Tolerance	-0.3 to 3.63V
Voltage with respect to ground on any signal pin with 5.0V Tolerance	-0.3 to 5.5V

Note 45-1 On any signal pin without backdrive protection, the voltage level on the pin must never exceed the voltage level of the power supply used to power the buffer.

© 2012-2018 Microchip Technology Inc.

45.2 Operational Specifications

45.2.1 POWER SUPPLY OPERATIONAL CHARACTERISTICS

TABLE 45-3: POWER SUPPLY OPERATING CONDITIONS

Symbol	Parameter	MIN	TYP	MAX	Units
VBAT	Battery Backup Power Supply	2.0	3.0	3.6	V
VTR	VTR Power Supply	2.97	3.3	3.63	V
AVTR_ADC	Analog VTR Supply	2.97	3.3	3.63	V
VREF_ADC	ADC Voltage Reference Pin	2.97	3.3	3.63	V

45.2.2 CAPACITIVE LOADING SPECIFICATIONS

The following table defines the maximum capacitive load validated for the buffer characteristics listed in Table 45-4, “DC Electrical Characteristics,” on page 604

CAPACITANCE T_A = 25°C; fc = 1MHz; V_{cc} = 3.3 VDC

Note: All output pins, except pin under test, tied to AC ground.

Parameter	Symbol	Limits			Units	Notes
		MIN	TYP	MAX		
Input Capacitance of PCI_I and PCI_IO pins	C_{IN}			Note 45-2	pF	
Input Capacitance of PCI_CLK pin	C_{IN}			Note 45-2	pF	
Output Load Capacitance supported by PCI_IO, PCI_O, and PCI_OD	C_{OUT}			Note 45-2	pF	
Input Capacitance of PECI_IO	C_{IN}			10	pF	
Output Load Capacitance supported by PECI_IO	C_{OUT}			10	pF	
Input Capacitance (all other input pins)	C_{IN}			10	pF	Note 45-3
Output Capacitance (all other output pins)	C_{OUT}			20	pF	Note 45-4

Note 45-2 The PCI buffers are designed to meet the defined PCI Local Bus Specification, Rev. 3.0, electrical requirements.

Note 45-3 All input buffers can be characterized by this capacitance unless otherwise specified.

Note 45-4 All output buffers can be characterized by this capacitance unless otherwise specified.

© 2012-2018 Microchip Technology Inc.

45.2.3 DC ELECTRICAL CHARACTERISTICS FOR I/O BUFFERS

TABLE 45-4: DC ELECTRICAL CHARACTERISTICS

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
I_AN	R_{IN}			75	m Ohm	Maximum DC current carrying capability = 84mA
I Type Input Buffer						TTL Levels
Low Input Level	V_{ILI}			0.8	V	
High Input Level	V_{IH}	2.0			V	
IS Type Input Buffer						
Low Input Level	V_{ILIS}			0.8	V	Schmitt Trigger
High Input Level	V_{IHIS}	2.2			V	Schmitt Trigger
Schmitt Trigger Hysteresis	V_{HYS}		250		mV	
O-4 mA Type Buffer						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 4 mA
High Output Level	V_{OH}	2.4			V	I_{OH} = -4 mA
IO-4 mA Type Buffer	_	_	_	_	_	Same characteristics as an I and an O-4mA.
OD-4 mA Type Buffer						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 4 mA
IOD-4 mA Type Buffer	_	_	_	_	_	Same characteristics as an I and an OD-4mA.
O-8 mA Type Buffer						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 8 mA
High Output Level	V_{OH}	2.4			V	I_{OH} = -8 mA
IO-8 mA Type Buffer	_	_	_	_	_	Same characteristics as an I and an O-8mA.
OD-8 mA Type Buffer						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 8 mA
IOD-8 mA Type Buffer	_	_	_	_	_	Same characteristics as an I and an OD-8mA.
O-8T mA Type Buffer						
Low Output Level	V_{OL}			0.4	V	I_{OL} = 9.63 mA
High Output Level	V_{OH}	2.4			V	I_{OH} = -9.29 mA See Section 45.2.3.1, "I-V Characteristic Curves for 8T mA Output Drivers," on page 607 for single stage turn-on characteristics.
IO-8T mA Type Buffer	_	_	_	_	_	Same characteristics as an I and an O-8T mA.

© 2012-2018 Microchip Technology Inc.

TABLE 45-4: DC ELECTRICAL CHARACTERISTICS (CONTINUED)

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
OD-8T mA Type Buffer Low Output Level	V_{OL}			0.4	V	I_{OL} = 9.63 mA See Section 45.2.3.1, "I-V Characteristic Curves for 8T mA Output Drivers," on page 607 for single stage turn-on characteristics.
IOD-8T mA Type Buffer	_	_	_	_	_	Same characteristics as an I and an OD-8T mA.
O-12 mA Type Buffer Low Output Level High Output Level	V_{OL} V_{OH}	 2.4		0.4	V V	I_{OL} = 12mA I_{OH} = -12mA
IO-12 mA Type Buffer	_	_	_	_	_	Same characteristics as an I and an O-12mA.
OD-12 mA Type Buffer Low Output Level	V_{OL}			0.4	V	I_{OL} = 12mA
IOD-12 mA Type Buffer	_	_	_	_	_	Same characteristics as an I and an OD-12mA.
I_AN-OD12 ma Type Buffer	_	_	_	_	_	Same characteristics as an I_AN and an OD-12mA.
O-16 mA Type Buffer Low Output Level High Output Level	V_{OL} V_{OH}	 2.4		0.4	V V	I_{OL} = 16mA I_{OH} = -16mA
IO-16 mA Type Buffer	_	_	_	_	_	Same characteristics as an I and an O-16mA.
OD-16 mA Type Buffer Low Output Level	V_{OL}			0.4	V	I_{OL} = 16mA
IOD-16 mA Type Buffer	_	_	_	_	_	Same characteristics as an I and an OD-16mA.
O-20 mA Type Buffer Low Output Level High Output Level	V_{OL} V_{OH}	 2.4		0.4	V V	I_{OL} = 20mA I_{OH} = -20mA
IO-20 mA Type Buffer	_	_	_	_	_	Same characteristics as an I and an O-20mA.
OD-20 mA Type Buffer Low Output Level	V_{OL}			0.4	V	I_{OL} = 20mA
IOD-20 mA Type Buffer	_	_	_	_	_	Same characteristics as an I and an OD-20mA.

TABLE 45-4: DC ELECTRICAL CHARACTERISTICS (CONTINUED)

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
PCI Buffers (PCI_ICLK, PCI_IO, PCI_I, PCI_O, PCI_OD)	V_{IH}	$0.5V_{TR}$		V_{TR} + 0.5	V	See *PCI Local Bus Specification Rev. 2.2*
	V_{IL}	-0.5		$0.3V_{TR}$	V	
	V_{TR}	3.0		3.6	V	LPC Supply Voltage
	I_{IL}	-10		+10	µA	$0 < V_{IN} < V_{CC}$
	V_{OH}	$0.9V_{TR}$			V	I_{OUT} = -500 µA
	V_{OL}			$0.1V_{TR}$	V	I_{OUT} = 1500 µA
	C_{IN}			10	pF	
V_{REF} Buffer						
PECI Bus Voltage	V_{BUS}	0.90		1.26	V	
SB-TSI Bus Voltage	V_{BUS}	1.425		1.9	V	
Input current	IDC			100	µA	
Input Low Current	I_{IL}	-10		+10	µA	
IO-PECI Input voltage range	VIn	-0.3		V_{REF} + 0.3	V	All input and output voltages are a function of VREF_VTT (V_{REF}) buffer input.
Hysteresis	VHYS	0.1 × V_{REF}	0.2× V_{REF}		V	See PECI Specification.
Low Input VLevel	VIL			0.275× V_{REF}	V	
High Input Level	VIH	0.725× V_{REF}			V	
Low Output Level	VOL			0.25× V_{REF}	V	0.5mA < IOL < 1mA
High Output Level	VOH	0.75 × V_{REF}			V	IOH = -6mA
SB-TSI (IOD type buffer) Input voltage range	VIn	-0.3		V_{REF} + 0.3	V	All input and output voltages are a function of VREF_VTT (V_{REF}) buffer input.
Hysteresis	VHYS	0.1 × V_{REF}	0.2× V_{REF}		V	
Low Input VLevel	VIL			0.275× V_{REF}	V	
High Input Level	VIH	0.725× V_{REF}			V	
Low Output Level	VOL			0.25× V_{REF}	V	0.5mA < IOL < 1mA
Pull-Down Impedance	PD	65	91	136	K Ohms	
Pull-Up Impedance	PU	53	74	110	K Ohms	

Note 45-5 All 5V Tolerant I-type & I/O-type input buffers can be pulled to 5 volts.

Note 45-6 All 5V Tolerant OD-type output buffers can be pulled to 5 volts.

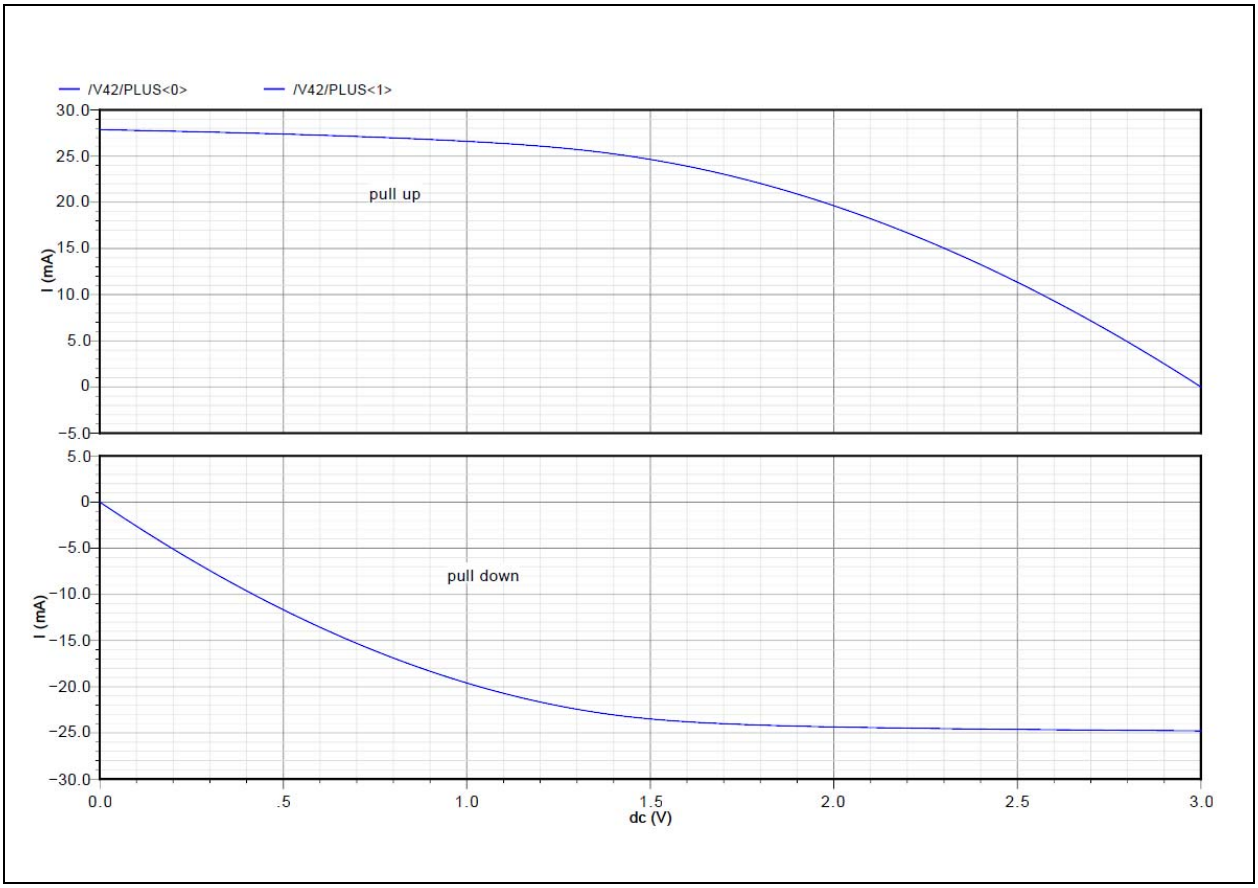
Note 45-7 All 5V Tolerant O-type and I/O-type output buffers will only drive to 3.3 volts, even if pulled-up externally to 5 volts.

© 2012-2018 Microchip Technology Inc.

45.2.3.1 I-V Characteristic Curves for 8T mA Output Drivers

IV curve simulation conditions for a single stage turn-on:

- SS VDDIO=3.0V Temp=85° C
- I=9.29mA with Delta V=0.4v for pull up
- I=9.63mA with Delta V=0.4V for pull down



45.2.3.2 Pin Leakage

Leakage characteristics for all pins is shown in the following table:

TABLE 45-5: Pin Leakage

(T_A = 0° C – 70° C Commercial)

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
Three-State leakage current	I_{IL}			+1 -5 -1	µA	V_{IN} > 2V 1V ≤ V_{IN} ≤ 2V V_{IN} < 1V

45.2.3.3 Backdrive Protection

All MEC1632 signal pins are Backdrive Protected except those listed in the Pin Configuration chapter as non-backdrive protected.

TABLE 45-6: Backdrive Protection

(T_A = 0° C – 70° C Commercial)

Parameter	Symbol	MIN	TYP	MAX	Units	Comments
Input Leakage	I_{IL}	-10		+10	µA	V_{IN} = 5.0 V (5.5 V) @ VTR = 0 V (Note 45-8)

Note 45-8 for VTR powered pins, VTR = 0 V, VBAT = 0 V.

45.3 Power Consumption

TABLE 45-7: I_{VTR} CURRENT CONSUMPTION

VCC	System "S" State	OSC State	ARC State	ARC CLK Freq	CLK Tree	Supply Current (mA)		Comments
						3.3V	3.6V	
						25^0C	70^0C	
3.3V	S0–S2	Ring @20 MHz	Run	20 MHz	All ON	13.75	17.0	FULL POWER (Note 45-9),
				10 MHz	All ON	11.0	14.0	
			Sleep	Off	All ON	7.0	8.5	EC SLEEP (Note 45-9)
					EC0	5.75	7.5	SYSTEM LIGHT SLEEP (Note 45-10)
					EC1	5.75	7.5	
					EC2	5.75	7.5	
					EC3	5.75	7.5	
					EC4	5.75	7.5	
					EC5	5.75	7.5	
					Host0	5.75	7.5	
					Host1	5.75	7.5	
		Ring @20 MHz	Sleep	Off	Master Clock Tree Only	3.75	4.25	SYSTEM HEAVY SLEEP 1 (LPC On)
						3.0	3.5	SYSTEM HEAVY SLEEP 1 (LPC Off)
					Master Clock Gated	1.25	2.0	SYSTEM HEAVY SLEEP 2 (LPC On)
						0.75	1.0	SYSTEM HEAVY SLEEP 2 (LPC Off)
		Off	Sleep	Off	Off	1.0	1.5	SYSTEM HEAVY SLEEP 3 (LPC On)
						0.25	0.75	SYSTEM HEAVY SLEEP 3 (LPC Off)

© 2012-2018 Microchip Technology Inc.

TABLE 45-7: I_{VTR} CURRENT CONSUMPTION (CONTINUED)

VCC	System "S" State	OSC State	ARC State	ARC CLK Freq	CLK Tree	Supply Current (mA)		Comments
						3.3V	3.6V	
						25^0C	70^0C	
0V	S3	Ring @20 MHz	Run	20 MHz	All ON	12.0	14.25	FULL POWER (Note 45-11),
				10 MHz	All ON	9.0	11.25	
				2 MHz	All ON	6.25	8.75	
			Sleep	Off	EC0	4.0	5.0	SYSTEM LIGHT SLEEP (Note 45-12)
					EC1	4.0	5.0	
					EC2	4.0	5.0	
					EC3	4.0	5.0	
					EC4	4.0	5.0	
					EC5	4.0	5.0	
		Ring @20 MHz	Sleep	Off	Master Clock Tree Only	3.0	4.0	SYSTEM HEAVY SLEEP 1
					Master Clock Gated	0.65	1.0	SYSTEM HEAVY SLEEP 2
		Off	Sleep	Off	Off	0.25	0.75	SYSTEM HEAVY SLEEP 3
						0.25	0.70	SYSTEM DEEPEST SLEEP
0V	S5/G3	Off	Sleep	Off	Off	See Table 45-8, "IVBAT Current Consumption"		VTR off

Note 45-9 The following devices are running in this power mode: one GPIO, 16-bit Timer 3, ADC, PWM0, PECI. LPC, EMI

Note 45-10 One device from the following list is running in this power mode: one GPIO, 16-bit Timer 3, ADC, PWM0, PECI. LPC, EMI

Note 45-11 The following devices are running in this power mode: one GPIO, 16-bit Timer 3, ADC, PWM0, PECI

Note 45-12 One device from the following list is running in this power mode: one GPIO, 16-bit Timer 3, ADC, PWM0, PECI

Note:

- On AC power, the System can enter the S3-S5 states when the EC is in sleep mode.
- All inputs not being tested are pulled up to power rails; all outputs are floating.
- The VREF_VTT pin is grounded during these measurements. This is necessary for lowest power even if the PECI or SB-TSI are not selected.
- The VREG SUS bit is asserted in the VREG Control Register during these measurements.

TABLE 45-8: I_{VBAT} CURRENT CONSUMPTION

VTR	VBAT=3.0V		VBAT=3.3V		Comments
	25^0C	70^0C	25^0C	70^0C	
0V	9.0µA	17.0µA	9.0µA	17.0µA	Internal 32KHz oscillator
	4.0µA	10.0µA	6.0µA	10.0µA	32KHz crystal oscillator
	4.0µA	7.0µA	4.0µA	8.0µA	External single-ended 32KHz clock source
	2.0µA	7.0µA	4.0µA	8.0µA	32KHz clock domain disabled

© 2012-2018 Microchip Technology Inc.

46.0 TIMING DIAGRAMS

46.1 VTR/VBAT Power-up and Power-down Timing

FIGURE 46-1: VTR/VBAT POWER-UP

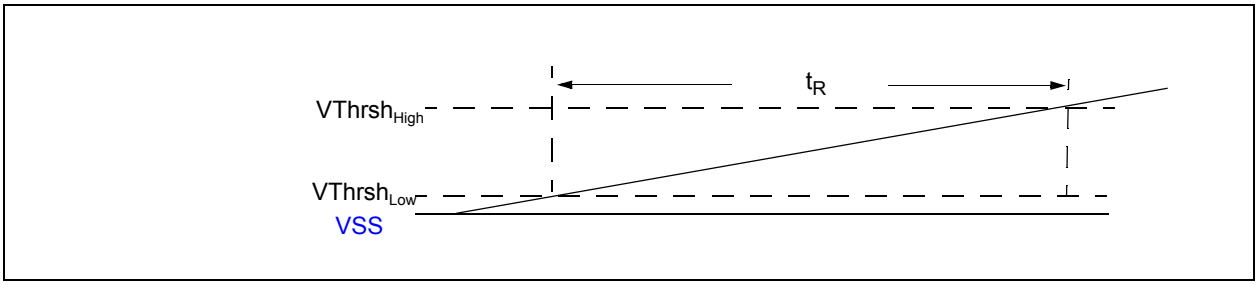


FIGURE 46-2: RESET AND POWER-DOWN

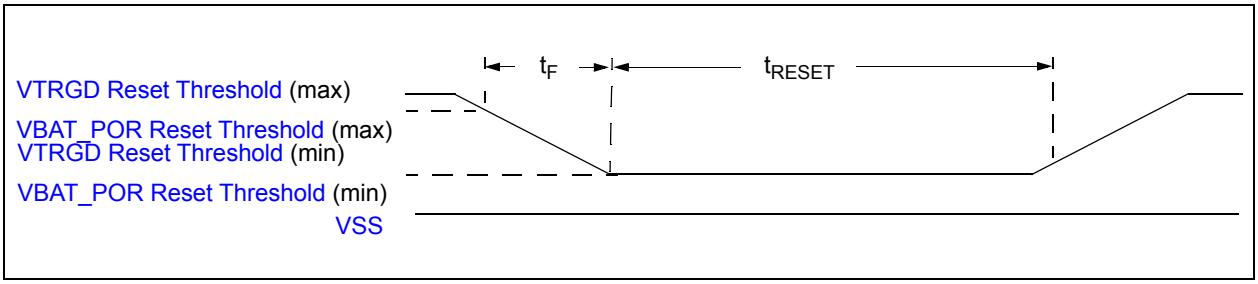


TABLE 46-1: VTR/VBAT POWER TIMING PARAMETERS

Symbol	Parameters	Limits		Units	Comments
		MIN	MAX		
t_F	VTR Fall time	30		µs	
	VBAT Fall time	30		µs	
t_R	VTR Rise time	0.050	30	ms	
	VBAT Rise time	0.025	30	ms	
t_{RESET}	Minimum Reset Time	1		µs	
$VThrsh_{Low}$	VTR Low Voltage Threshold	0.1 × VTR		V	
	VBAT Low Voltage Threshold	0.1 × VBAT		V	
$VThrsh_{High}$	VTR High Voltage Threshold		0.9 × VTR	V	
	VBAT High Voltage Threshold		0.9 × VBAT	V	

46.2 LPC Clock and Reset Timing

FIGURE 46-3: PCI CLOCK TIMING

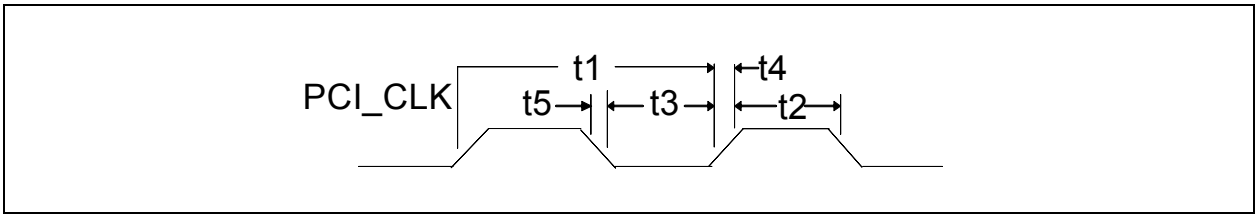


TABLE 46-2: PCI CLOCK TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t1	Period	30		45.5 (Note 46-1)	ns
t2	High Time	11			
t3	Low Time				
t4	Rise Time			3	
t5	Fall Time				

Note 46-1 The standard clock frequency supported is 33MHz (max 33.3ns period). Setting the Hand shake bit in the Host Interface allows the LPC to support a 24MHz PCI clock rate.

FIGURE 46-4: RESET TIMING

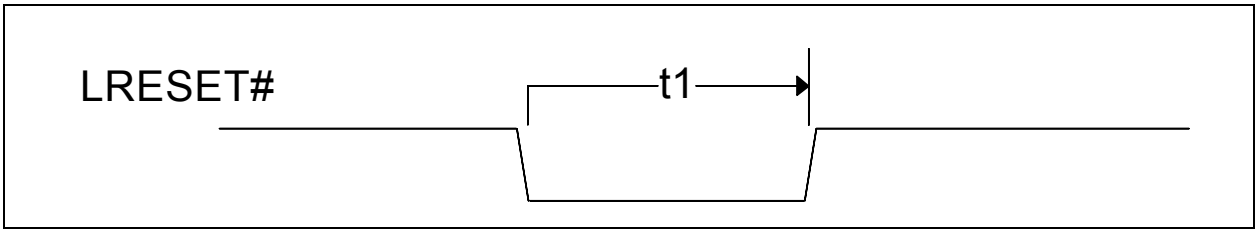


TABLE 2: RESET TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t1	LRESET# width	1			ms

© 2012-2018 Microchip Technology Inc.

46.3 LPC Timing

FIGURE 46-5: OUTPUT TIMING MEASUREMENT CONDITIONS, LPC SIGNALS

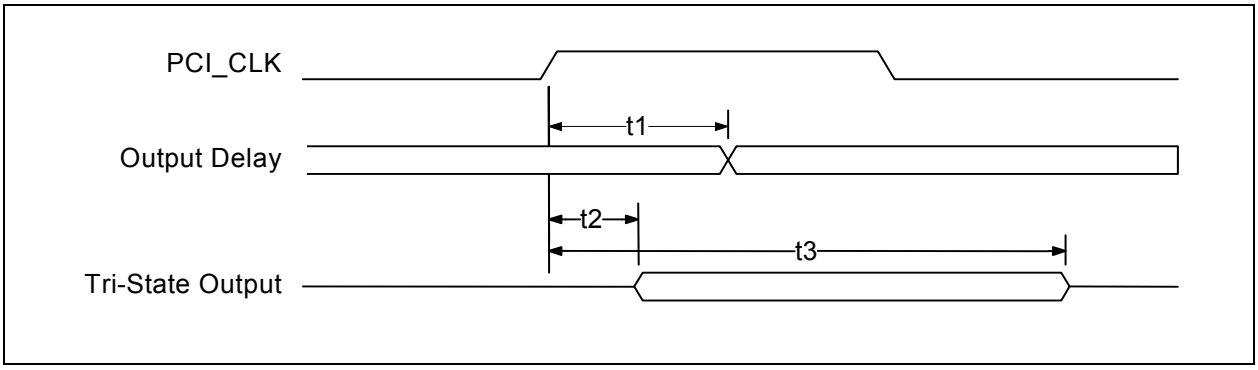


TABLE 46-3: OUTPUT TIMING MEASUREMENT CONDITIONS, LPC SIGNALS PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t1	PCI_CLK to Signal Valid Delay – Bused Signals	2		11	ns
t2	Float to Active Delay				
t3	Active to Float Delay			28	

FIGURE 46-6: INPUT TIMING MEASUREMENT CONDITIONS, LPC SIGNALS

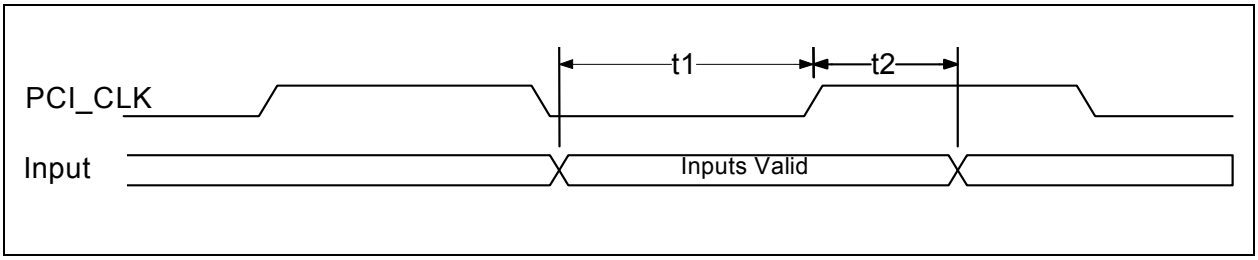
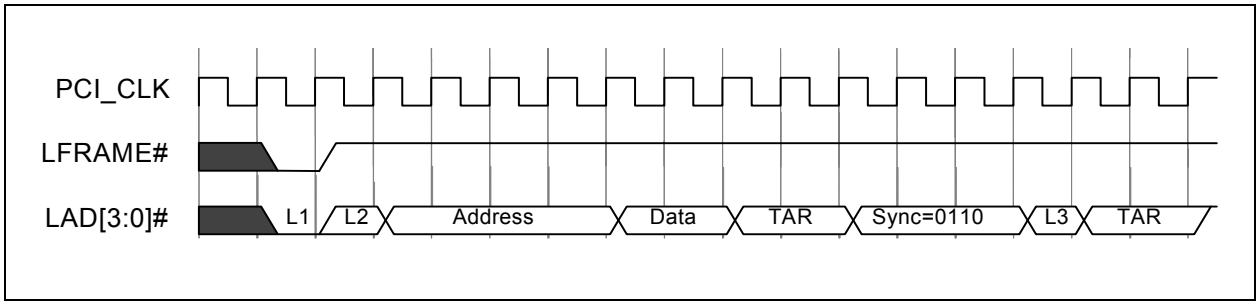


TABLE 46-4: INPUT TIMING MEASUREMENT CONDITIONS, LPC SIGNALS PARAMETERS

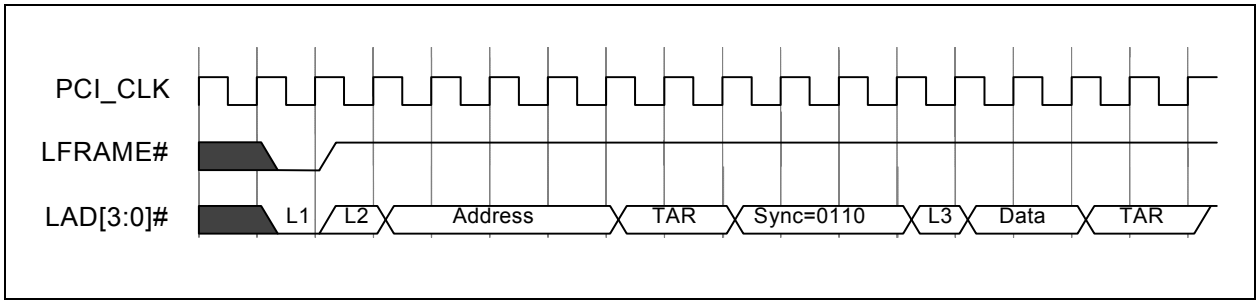
Name	Description	MIN	TYP	MAX	Units
t1	Input Set Up Time to PCI_CLK – Bused Signals	7			ns
t2	Input Hold Time from PCI_CLK	0			

FIGURE 46-7: I/O WRITE



Note: L1=Start; L2=CYCTYP+DIR; L3=Sync of 0000

FIGURE 46-8: I/O READ



Note: L1=Start; L2=CYCTYP+DIR; L3=Sync of 0000

46.4 Serial IRQ Timing

FIGURE 46-9: SETUP AND HOLD TIME

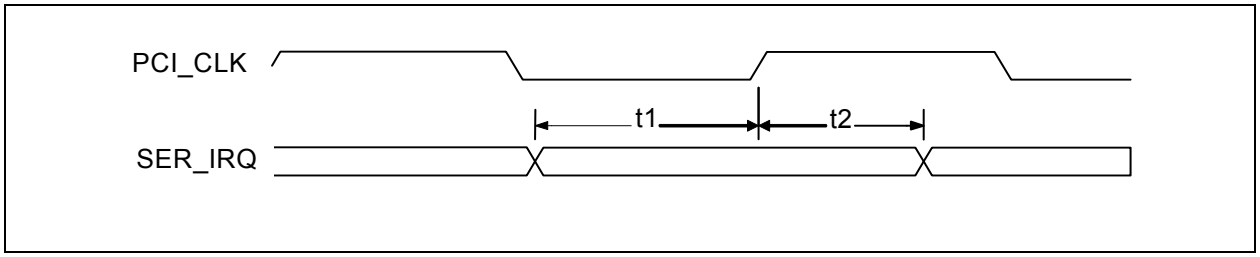


TABLE 46-5: SETUP AND HOLD TIME

Name	Description	MIN	TYP	MAX	Units
t1	SER_IRQ Setup Time to PCI_CLK Rising	7			ns
t2	SER_IRQ Hold Time to PCI_CLK Rising	0			

© 2012-2018 Microchip Technology Inc.

46.5 Serial Port Data Timing

FIGURE 46-10: SERIAL PORT DATA

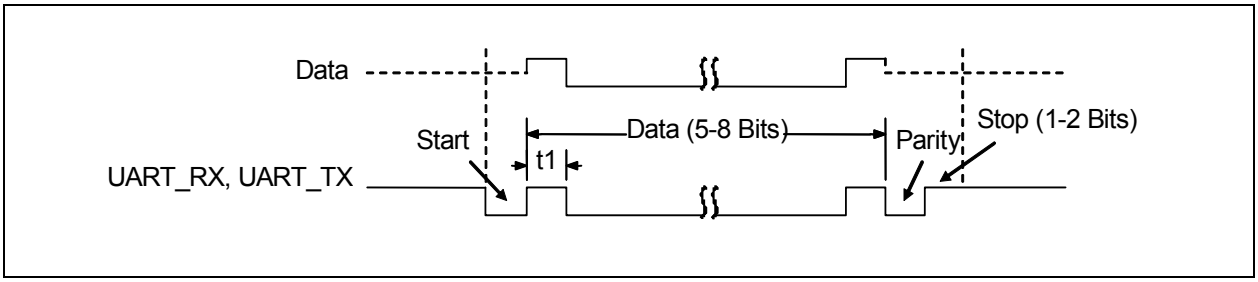


TABLE 46-6: SERIAL PORT DATA PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t1	Serial Port Data Bit Time		t_{BR} (Note 46-2)		ns

Note 46-2 t_{BR} is 1/Baud Rate. The Baud Rate is programmed through the divisor latch registers. Baud Rates have percentage errors indicated in Table 13-21, "UART Baud Rates (1.8432MHz source)," on page 236.

FIGURE 46-11: UART_CLK EXTERNAL CLOCK TIMING

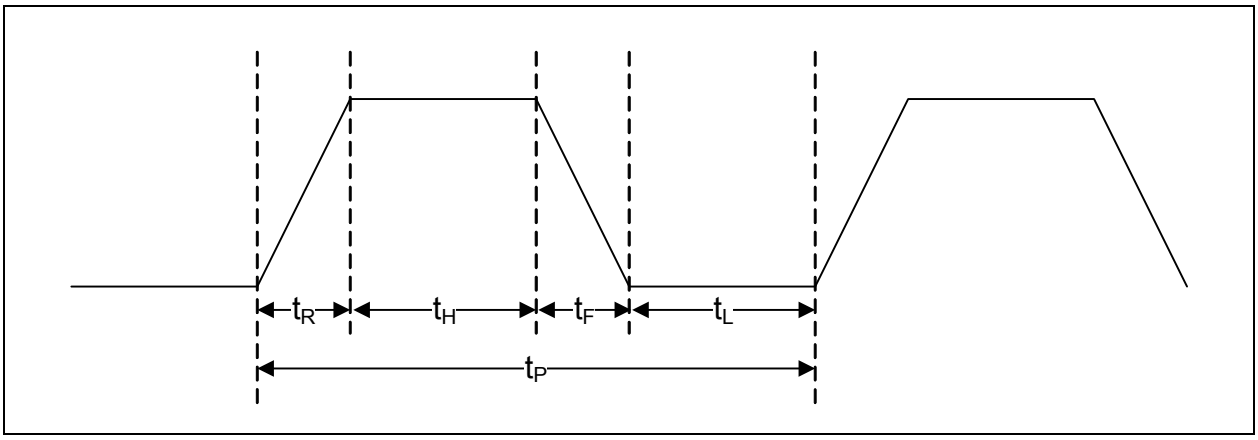


TABLE 46-7: UART_CLK EXTERNAL CLOCK TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t_P	Period	553.6	542.5	553.6	nsec
t_H	High Time	200			
t_L	Low Time				
t_R	Rise Time			10	
t_F	Fall Time				

46.6 I2C/SMBus Timing

Note that the following timing applies to all of the MEC1632 I2C/SMBus and SB-TSI functions.

FIGURE 46-12: I2C/SMBUS TIMING

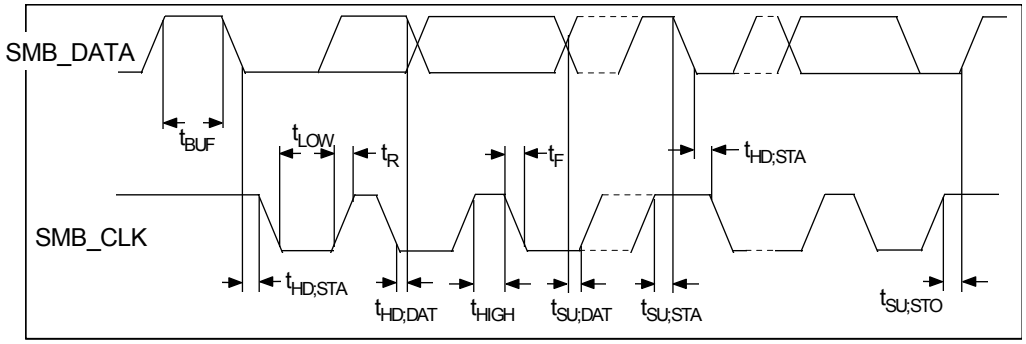


TABLE 46-8: I2C/SMBUS TIMING PARAMETERS

Symbol	Parameter	Standard-Mode		Fast Mode		Unit
		MIN	MAX	MIN	MAX	
f_{SCL}	SMB Clock Frequency		100		400	KHz
t_{BUF}	Bus Free Time	4.7		1.3		µs
$t_{SU;STA}$	START Condition Set-Up Time	4.7		0.6		µs
$t_{HD;STA}$	START Condition Hold Time	4.0		0.6		µs
t_{LOW}	SMB_CLK LOW Time	4.7		1.3		µs
t_{HIGH}	SMB_CLK HIGH Time	4.0		0.6		µs
t_R	SMB_CLK and SMB_DATA Rise Time		1.0		0.3	µs
t_F	SMB_CLK and SMB_DATA Fall Time		0.3		0.3	µs
$t_{SU;DAT}$	Data Set-Up Time	0.25		0.1		µs
$t_{HD;DAT}$	Data Hold Time	0		0		µs
$t_{SU;STO}$	STOP Condition Set-Up Time	4.0		0.6		µs

© 2012-2018 Microchip Technology Inc.

46.7 Fan Tachometer Timing

FIGURE 46-13: FAN TACHOMETER INPUT TIMING

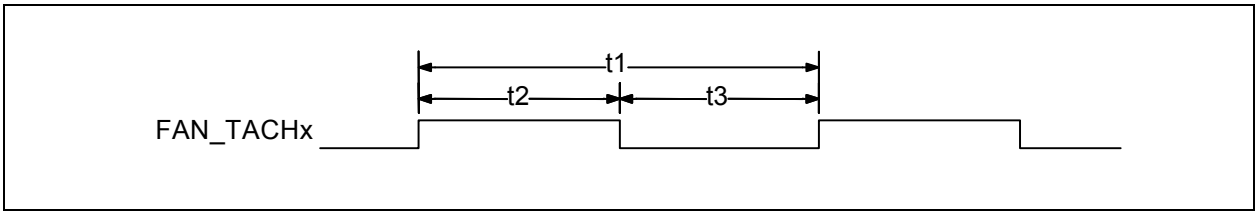


TABLE 46-9: FAN TACHOMETER INPUT TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t1	Pulse Time	100			µsec
t2	Pulse High Time				
t3	Pulse Low Time	10			

Note 46-3 t_{TACH} is the clock used for the tachometer counter. It is 30.52 * prescaler, where the prescaler is programmed in the Fan Tachometer Timebase Prescaler register.

46.8 PS/2 Timing

FIGURE 46-14: PS/2 TRANSMIT TIMING

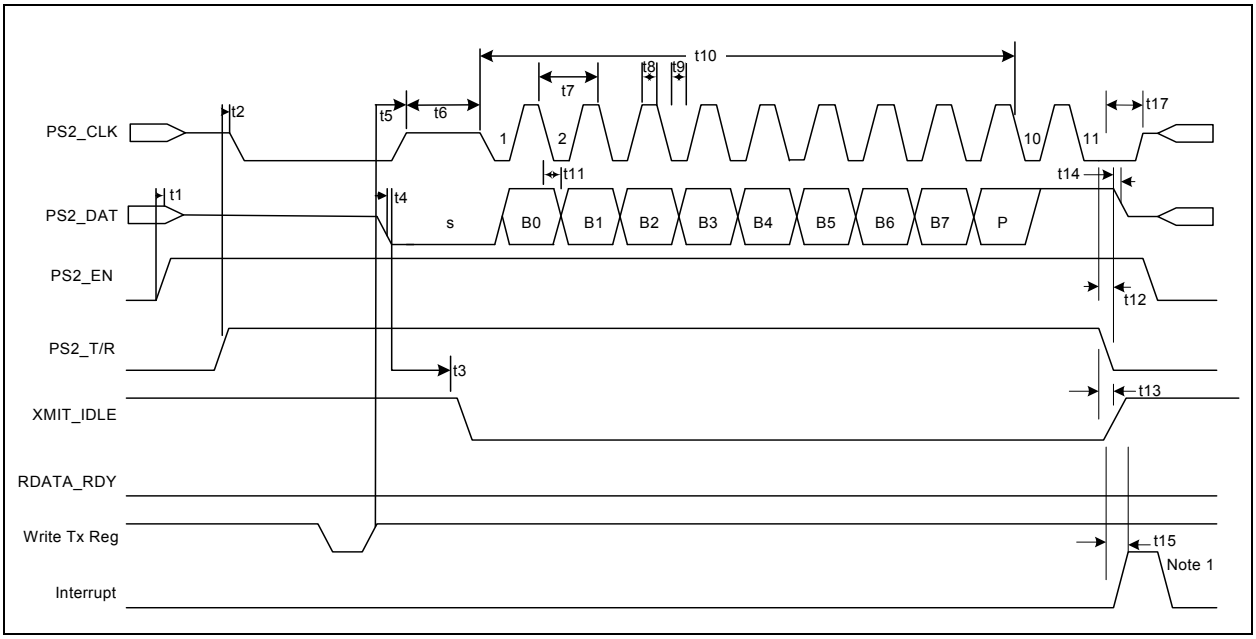


TABLE 46-10: PS/2 CHANNEL TRANSMISSION TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t1	The PS/2 Channel's CLK and DATA lines are floated following PS2_EN=1 and PS2_T/R=0.			1000	ns
t2	PS2_T/R bit set to CLK driven low preparing the PS/2 Channel for data transmission.				
t3	CLK line floated to XMIT_IDLE bit de-asserted.			1.7	
t4	Trailing edge of WR to Transmit Register to DATA line driven low.	45		90	
t5	Trailing edge of EC WR of Transmit Register to CLK line floated.	90		130	ns
t6	Initiation of Start of Transmit cycle by the PS/2 channel controller to the auxiliary peripheral's responding by latching the Start bit and driving the CLK line low.	0.002		25.003	ms
t7	Period of CLK	60		302	µs
t8	Duration of CLK high (active)	30		151	
t9	Duration of CLK low (inactive)				
t10	Duration of Data Frame. Falling edge of Start bit CLK (1st clk) to falling edge of Parity bit CLK (10th clk).			2.002	ms
t11	DATA output by MEC1632 following the falling edge of CLK. The auxiliary peripheral device samples DATA following the rising edge of CLK.			1.0	µs
t12	Rising edge following the 11th falling clock edge to PS_T/R bit driven low.	3.5		7.1	µs

© 2012-2018 Microchip Technology Inc.

TABLE 46-10: PS/2 CHANNEL TRANSMISSION TIMING PARAMETERS (CONTINUED)

Name	Description	MIN	TYP	MAX	Units
t13	Trailing edge of PS_T/R to XMIT_IDLE bit asserted.			500	ns
t14	DATA released to high-Z following the PS2_T/R bit going low.				
t15	XMIT_IDLE bit driven high to interrupt generated. Note1- Interrupt is cleared by writing a 1 to the status bit in Table 17-52, “GIRQ19 Source Register,” on page 318.				
t17	Trailing edge of CLK is held low prior to going high-Z				

FIGURE 46-15: PS/2 RECEIVE TIMING

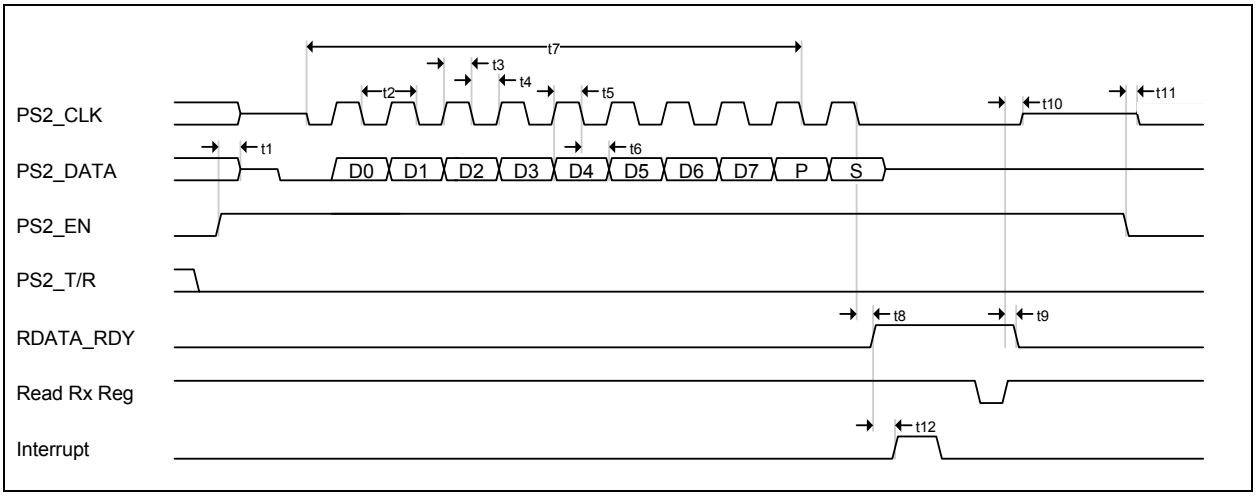


TABLE 46-11: PS/2 CHANNEL RECEIVE TIMING DIAGRAM PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t1	The PS/2 Channel’s CLK and DATA lines are floated following PS2_EN=1 and PS2_T/R=0.			1000	ns
t2	Period of CLK	60		302	µs
t3	Duration of CLK high (active)	30		151	
t4	Duration of CLK low (inactive)				
t5	DATA setup time to falling edge of CLK. MEC1632 samples the data line on the falling CLK edge.	1			
t6	DATA hold time from falling edge of CLK. MEC1632 samples the data line on the falling CLK edge.	2			
t7	Duration of Data Frame. Falling edge of Start bit CLK (1st clk) to falling edge of Parity bit CLK (10th clk).			2.002	ms
t8	Falling edge of 11th CLK to RDATA_RDY asserted.			1.6	µs

© 2012-2018 Microchip Technology Inc.

TABLE 46-11: PS/2 CHANNEL RECEIVE TIMING DIAGRAM PARAMETERS (CONTINUED)

Name	Description	MIN	TYP	MAX	Units
t9	Trailing edge of the EC's RD signal of the Receive Register to RDATA_RDY bit de-asserted.			500	ns
t10	Trailing edge of the EC's RD signal of the Receive Register to the CLK line released to high-Z.				
t11	PS2_CLK is "Low" and PS2_DATA is "Hi-Z" when PS2_EN is de-asserted.				
t12	RDATA_RDY asserted an interrupt is generated.				

© 2012-2018 Microchip Technology Inc.

46.9 BC-Link Master Timing

FIGURE 46-16: BC-LINK READ TIMING

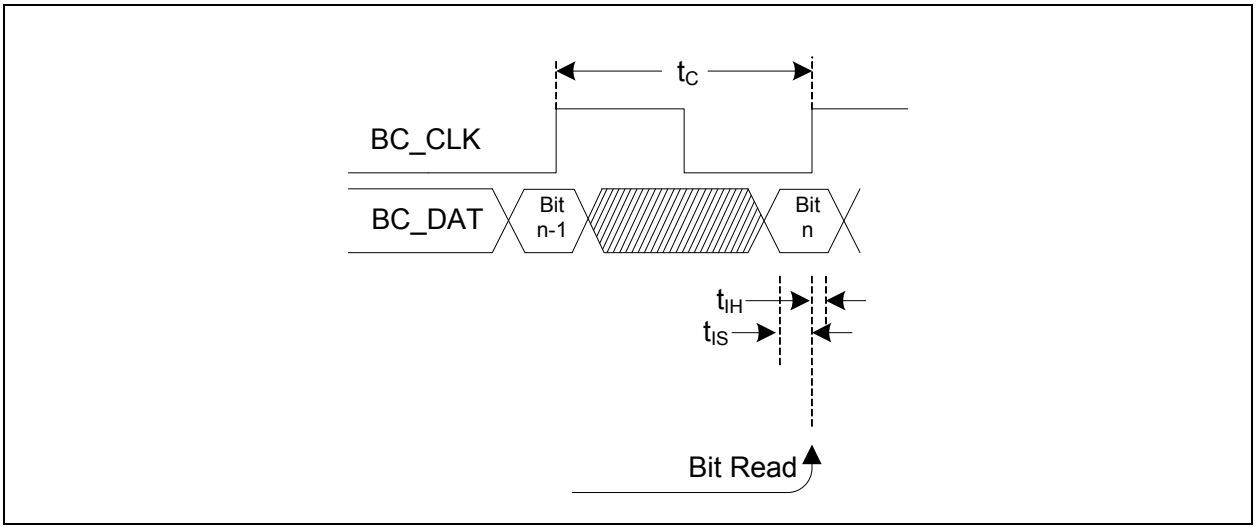


FIGURE 46-17: BC-LINK WRITE TIMING

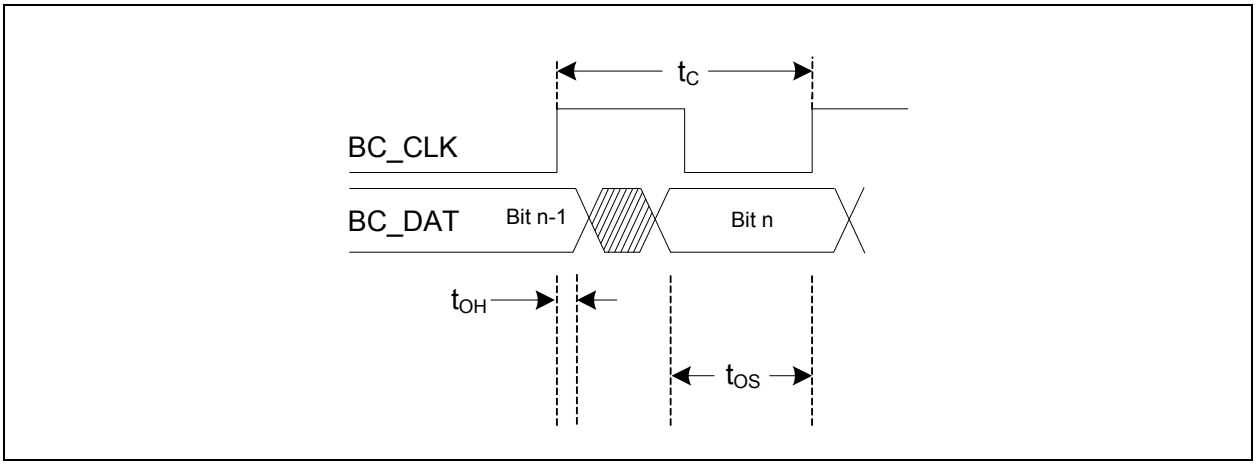


TABLE 46-12: BC-LINK MASTER TIMING DIAGRAM PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t_c(High Speed)	High Spec BC Clock Period	46.9			ns
t_c(Low Speed)	High Spec BC Clock Period	329			ns
t_F(High Speed)	High Spec BC Clock Frequency			21.3	MHz
t_F(Low Speed)	High Spec BC Clock Frequency			3.04	MHz
t_{OS}	Master Data output setup time before rising edge of CLK.			t_c-t_{OH}-MAX	ns
t_{OH}	Master Data data output hold time after rising edge of CLK			10	ns
t_{IS}	Master DATA input setup time before rising edge of CLK.	15			ns
t_{IH}	Master DATA input hold time after rising edge of CLK.	0			ns

Note: The BC-Link Clock frequency is limited by the application usage model (see Note 39-1 on page 549 & Table 39-2 on page 549.) The BC-Link Clock frequency is controlled by the BC-Link Clock Select Register. The tc(High Speed) parameter implies both BC-link master and companion devices are located on the same circuit board and a high speed clock setting is possible. The tc(Low Speed) parameter implies the BC-link master and companion devices are located on separate circuit boards connected by 12 inch ribbon cable and a low speed clock setting is required.

© 2012-2018 Microchip Technology Inc.

46.10 Serial Peripheral Interface (SPI) Timings

Note that the following timing applies to all of the MEC1632 Serial Peripheral Interface functions.

46.10.1 SPI CLOCK TIMING

FIGURE 46-18: SPI CLOCK TIMING

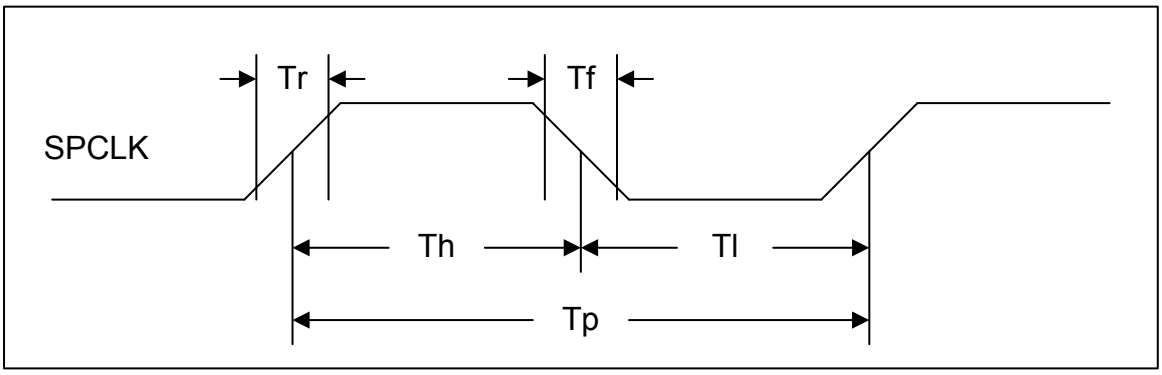


TABLE 46-13: SPI CLOCK TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
Tr	SPI Clock Rise Time. Measured from 10% to 90%.			10% of SPCLK Period	ns
Tf	SPI Clock Fall Time. Measured from 90% to 10%.			10% of SPCLK Period	ns
Th/Tl	SPI Clock High Time/SPI Clock Low Time	40% of SPCLK Period	50% of SPCLK Period (46.10.2)	60% of SPCLK Period	ns
Tp	SPI Clock Period – As selected by.SPICG - SPI Clock Generator Register	15.50		62492.25	ns

46.10.2 SPI SETUP AND HOLD TIMES

FIGURE 46-19: SPI SETUP AND HOLD TIMES

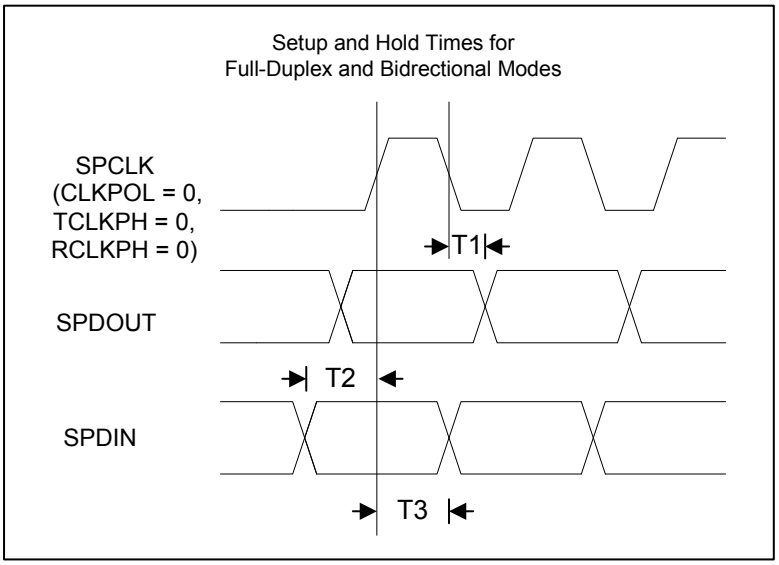


TABLE 46-14: SPI SETUP AND HOLD TIMES PARAMETERS

Name	Description	MIN	TYP	MAX	Units
T1	Data Output Delay			20	ns
T2	Data IN Setup Time	20			ns
T3	Data IN Hold Time	0			ns

© 2012-2018 Microchip Technology Inc.

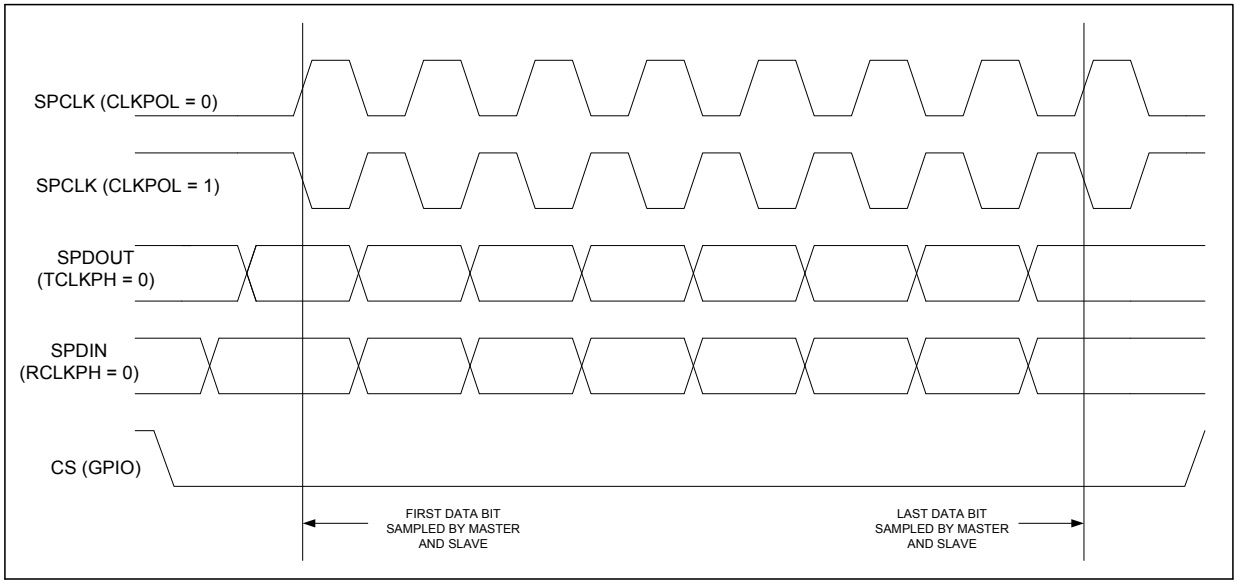
46.10.3 SPI INTERFACE TIMINGS

The following timing diagrams represent a single-byte transfer over the SPI interface using different SPCLK phase settings. Data bits are transmitted in bit order starting with the MSB (LSBF='0') or the LSB (LSBF='1'). See the SPICR - SPI Control Register for information on the LSBF bit. The CS signal in each diagram is a generic bit-controlled chip select signal required by most peripheral devices. This signal and additional chip selects can be GPIO controlled. Note that these timings for Full Duplex Mode are also applicable to Half Duplex (or Bi-directional) mode.

46.10.3.1 SPI Interface Timing – Full Duplex Mode (TCLKPH = 0, RCLKPH = 0)

In this mode, data is available immediately when a device is selected and is sampled on the first and following odd SPCLK edges by the master and slave.

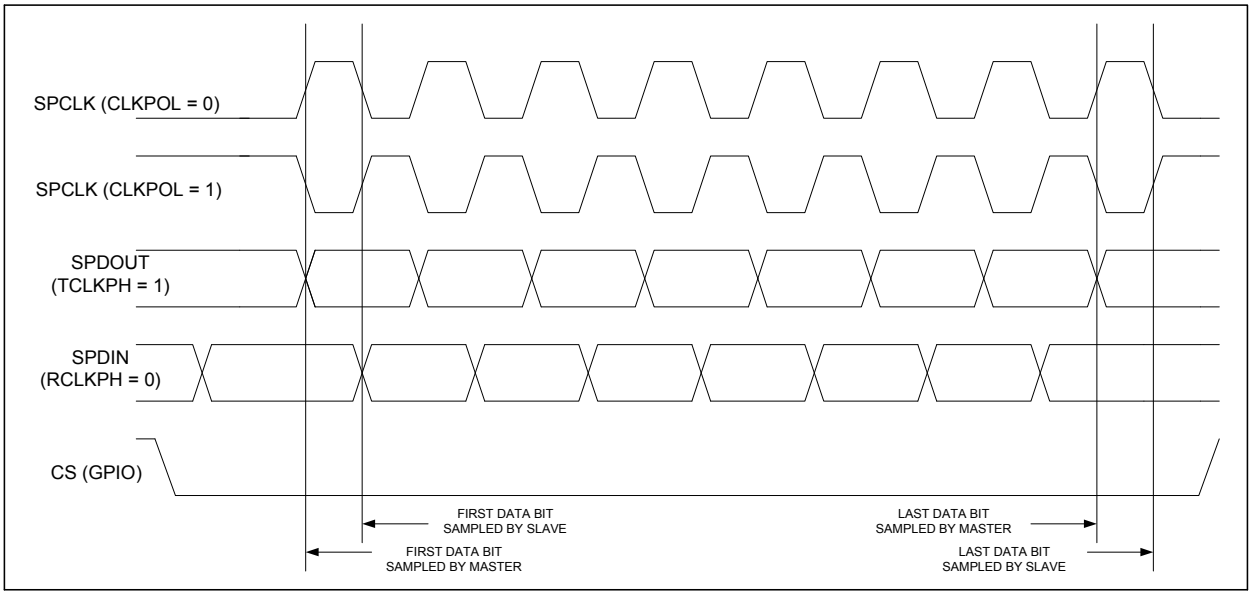
FIGURE 46-20: INTERFACE TIMING, FULL DUPLEX MODE (TCLKPH = 0, RCLKPH = 0)



46.10.3.2 SPI Interface Timing - Full Duplex Mode (TCLKPH = 1, RCLKPH = 0)

In this mode, the master requires an initial SPCLK edge before data is available. The data from slave is available immediately when the slave device is selected. The.data is sampled on the first and following odd edges by the master. The data is sampled on the second and following even SPCLK edges by the slave.

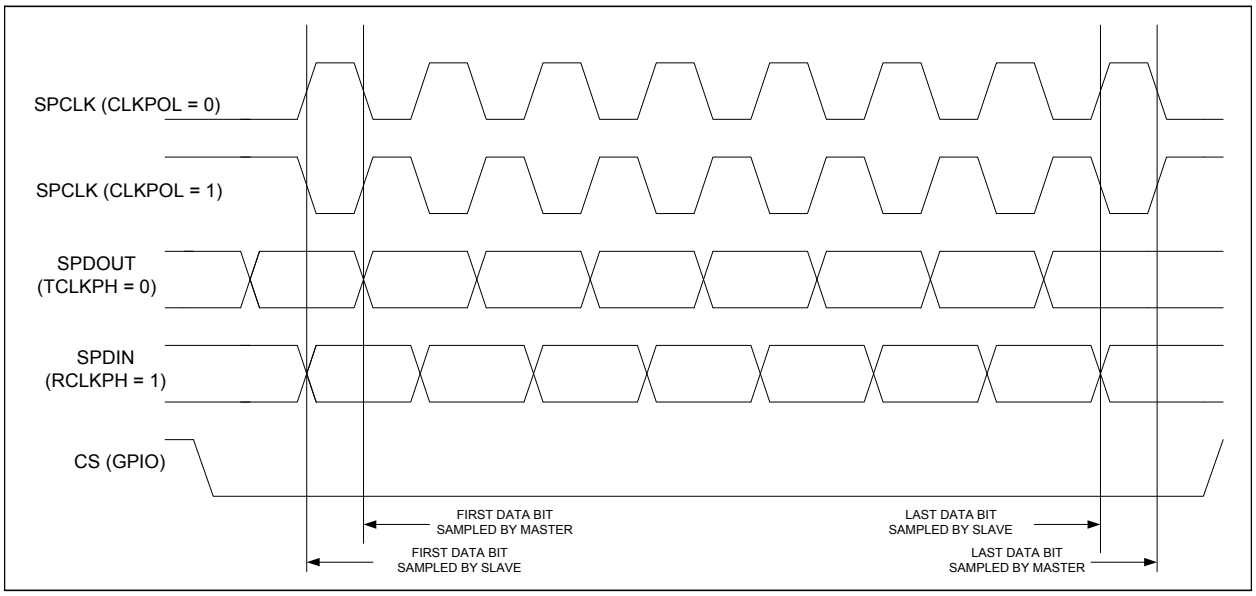
FIGURE 46-21: SPI INTERFACE TIMING, FULL DUPLEX MODE (TCLKPH = 1, RCLKPH = 0)



46.10.3.3 SPI Interface Timing - Full Duplex Mode (TCLKPH = 0, RCLKPH = 1)

In this mode, the data from slave is available immediately when the slave device is selected. The slave device requires an initial SPCLK edge before data is available. The data is sampled on the second and following even SPCLK edges by the master. The data is sampled on the first and following odd edges by the slave.

FIGURE 46-22: SPI INTERFACE TIMING, FULL DUPLEX MODE (TCLKPH = 0, RCLKPH = 1)

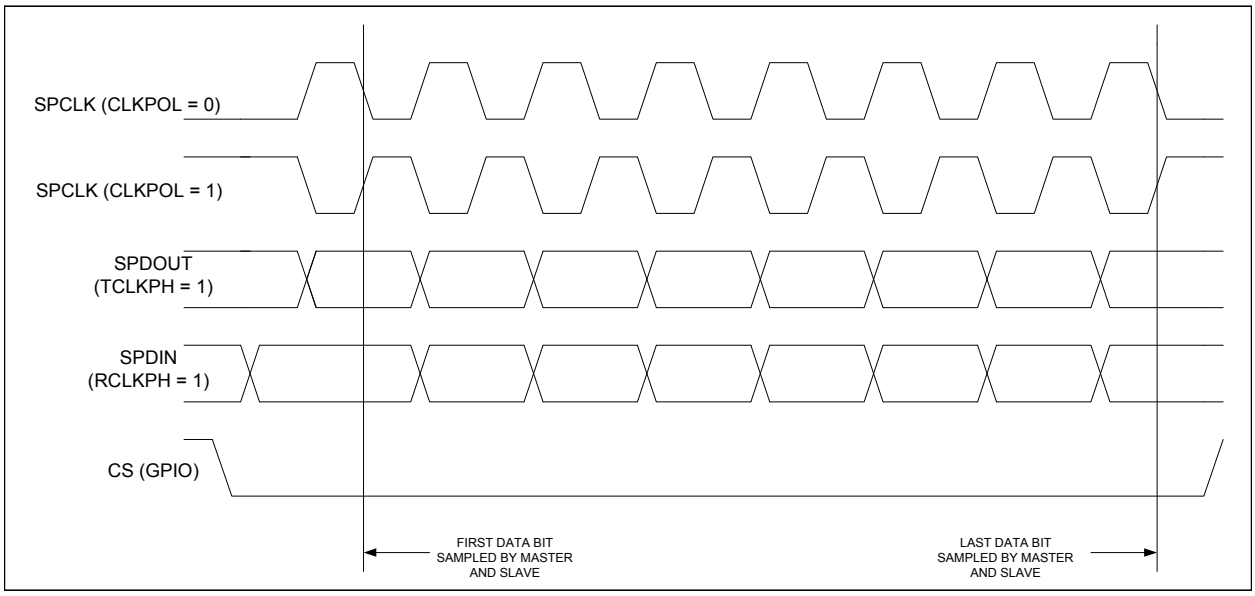


© 2012-2018 Microchip Technology Inc.

46.10.3.4 SPI Interface Timing - Full Duplex Mode (TCLKPH = 1, RCLKPH = 1)

In this mode, the master and slave require an initial SPCLK edge before data is available. Data is sampled on the second and following even SPCLK edges by the master and slave.

FIGURE 46-23: SPI INTERFACE TIMING - FULL DUPLEX MODE (TCLKPH = 1, RCLKPH = 1)



46.11 JTAG Interface Timing

FIGURE 46-24: JTAG POWER-UP & ASYNCHRONOUS RESET TIMING

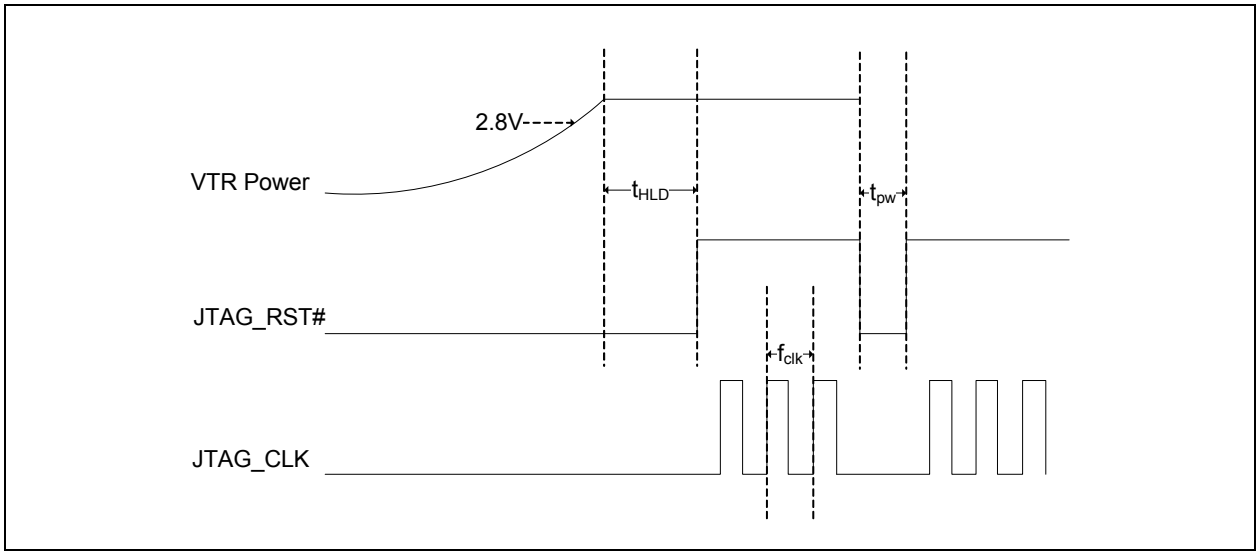
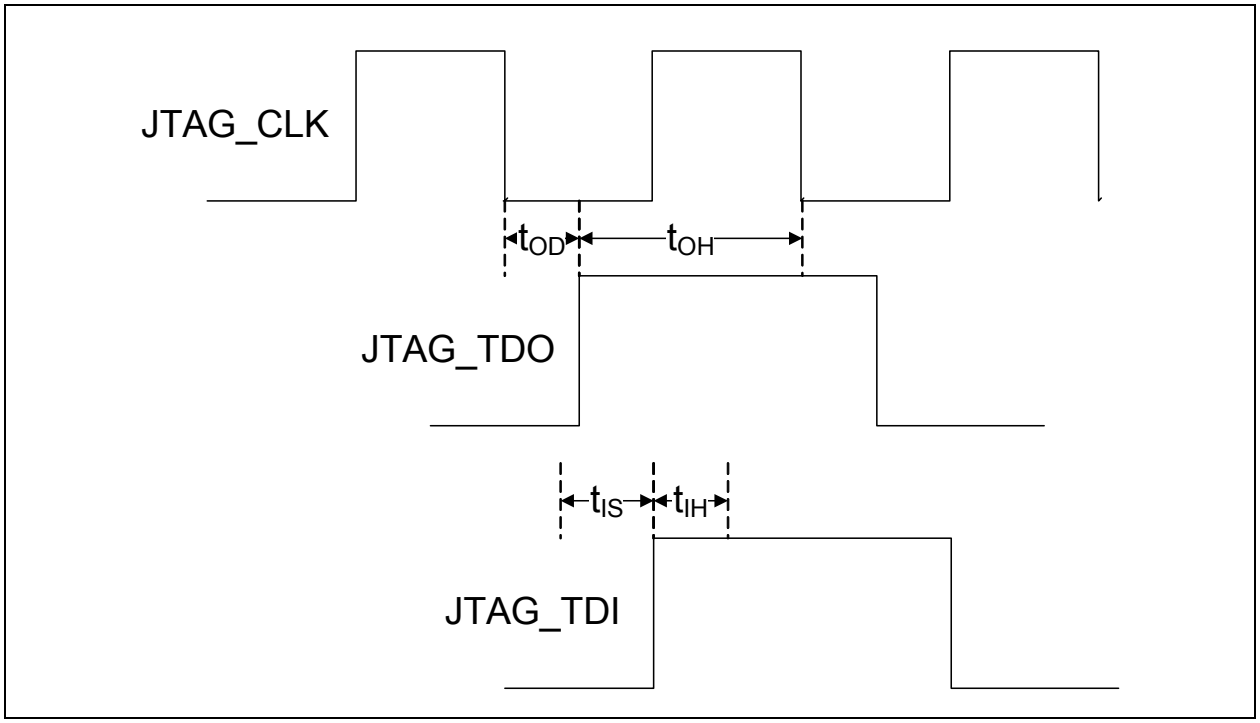


FIGURE 46-25: JTAG SETUP & HOLD PARAMETERS



© 2012-2018 Microchip Technology Inc.

TABLE 46-15: JTAG INTERFACE TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
t_{HLD}	JTAG_RST# de-assertion after VTR power is applied	5			ms
t_{pw}	JTAG_RST# assertion pulse width	500			ns
f_{clk}	JTAG_CLK frequency (see note)			8	MHz
t_{OD}	TDO output delay after falling edge of TCLK.	5		18	ns
t_{OH}	TDO hold time after falling edge of TCLK	1 TCLK - t_{OD}			ns
t_{IS}	TDI setup time before rising edge of TCLK.	15			ns
t_{IH}	TDI hold time after rising edge of TCLK.	7			ns

Note 46-4 f_{clk} is the maximum frequency to access a JTAG Register. Additional JTAG_CLK frequency constraints are described in Section 44.7.2, "Clocks", Section 44.7.2, "Clocks," on page 586.

46.12 Serial Debug Port Timing (TFDP)

FIGURE 46-26: SERIAL DEBUG PORT TIMING PARAMETERS

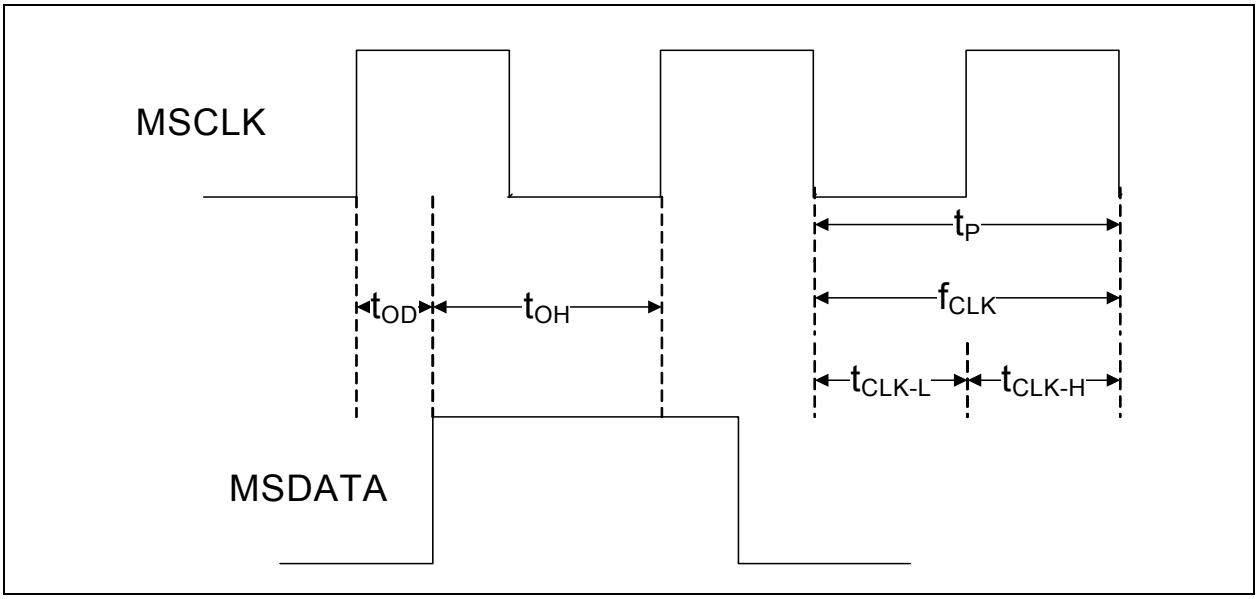


TABLE 46-16: SERIAL DEBUG PORT INTERFACE TIMING PARAMETERS

Name	Description	MIN	TYP	MAX	Units
f_{clk}	MSCLK frequency			10.15	MHz
t_{OD}	MSDATA output delay after falling edge of MSCLK.			5	ns
t_{OH}	MSDATA hold time after falling edge of TCLK	t_P - t_{OD}			ns
t_P	MSCLK Period	1/fclk			ns
t_{CLK-L}	MSCLK Low Time		t_P - t_{CLK-H}		sec
t_{CLK-H}	MSCLK High Time		t_P/2		sec

46.13 CLKRUN# Timing

(See Section 6.7.2, "Using CLKRUN#," on page 84.)

46.14 RESET Input Timing

(See Section 7.6.3, "RESET Pin Interface," on page 126.)

46.15 VCI Input Timing

(See Section 34.9, "Input Filtering," on page 504.)

46.16 GPIO Pulse Width Timing

(See Section 24.4, "Interrupts," on page 390.)

46.17 CPU Reset Timing

(See Section 10.14, "CPU_RESET Hardware Speed-Up," on page 203.)

46.18 Keyboard Scan Matrix Timing

(See Section 38.6.2, "Pre Drive Mode," on page 541.)

46.19 HDMI-CEC Timing

(See HDMI-CEC Interface Controller References[1].)

46.20 Gang Programmer Timing

(See Gang Programmer Interface Section 43.8.3, "Functional Description," on page 574.)

© 2012-2018 Microchip Technology Inc.

47.0 REFERENCE DOCUMENTS

This document was created using the following parent documents:

1. Intel Low Pin Count Specification, Revision 1.0, September 29, 1997
2. PCI Local Bus Specification, Revision 2.2, December 18, 1998
3. Advanced Configuration and Power Interface Specification, Revision 1.0b, February 2, 1999
4. System Management Bus Specification, Revision 1.1, December 11, 1998.
5. Plug and Play ISA Specification, Version 1.0a, Intel Corp. and Microsoft Corp., May 5, 1994
6. I2C-BUS Specification, Version 2.1, January 2000.
7. SMBus Controller Core Interface, Revision 2.0 (10 MHz), v3.31, Core-Level Architecture Specification, SMSC, 10/25/13
8. ECE1077 MEC-04 Keyboard Scan Extension, Product Architecture Specification, Rev 0.23, January 12, 2006, Confidential
9. Intel® 82801DBM I/O Controller Hub 4 Mobile (ICH4-M), Data Sheet, Order Number: 252337-001, Intel Corp., January 2003
10. BC-Link Specification, Revision 1.02, dated September 05, 2007
11. IEEE Std 1149.1
12. PECI Interface Core, Rev. 1.31, Core-Level Architecture Specification, SMSC Confidential, 4/15/11.
13. PCI Mobile Design Guide, Version 1.1, PCI-SIG, December 18, 1998.
14. *ARCompact™ Instruction Set Architecture Programmer's Reference*, ARC International, April 2009

APPENDIX A: DATA SHEET REVISION HISTORY

TABLE A-1: REVISION HISTORY

Revision	Section/Figure/Entry	Correction
DS00001592B (01-16-18)	Public release; document has been converted to the Microchip template.	

© 2012-2018 Microchip Technology Inc.

THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://www.microchip.com/support

© 2012-2018 Microchip Technology Inc.

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.[1] - XXX[2] - [X][3]	Example:
Device / Package / Tape and Reel Option	MEC1632-AUE = 169-pin LFBGA

Device:	MEC1632[1]	
Package:	AUE	= 169-pin LFBGA [2]
Tape and Reel Option:	Blank	= Tray packaging
	TR	= Tape and Reel [3]

Note 1: These products meet the halogen maximum concentration values per IEC61249-2-21.

2: All package options are RoHS compliant. For RoHS compliance and environmental information, please visit http://www.microchip.com/pagehandler/en-us/aboutus/ehs.html .

3: Tape and Reel identifier only appears in the catalog part number description. This identifier is used for ordering purposes and is not printed on the device package. Check with your Microchip Sales Office for package availability with the Tape and Reel option.

© 2012-2018 Microchip Technology Inc.

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE**.** Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, AnyRate, AVR, AVR logo, AVR Freaks, BeaconThings, BitCloud, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, Heldo, JukeBlox, KEELOQ, KEELOQ logo, Kleer, LANCheck, LINK MD, maXStylus, maXTouch, MediaLB, megaAVR, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, Prochip Designer, QTouch, RightTouch, SAM-BA, SpyNIC, SST, SST Logo, SuperFlash, tinyAVR, UNI/O, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

ClockWorks, The Embedded Control Solutions Company, EtherSynch, Hyper Speed Control, HyperLight Load, IntelliMOS, mTouch, Precision Edge, and Quiet-Wire are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BodyCom, chipKIT, chipKIT logo, CodeGuard, CryptoAuthentication, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, Inter-Chip Connectivity, JitterBlocker, KleerNet, KleerNet logo, Mindi, MiWi, motorBench, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PureSilicon, QMatrix, RightTouch logo, REAL ICE, Ripple Blocker, SAM-ICE, Serial Quad I/O, SMART-I.S., SQI, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2012-2018, Microchip Technology Incorporated, All Rights Reserved.

ISBN: 9781522425038

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
═ ISO/TS 16949 ═

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453
Tel: 317-536-2380

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608
Tel: 951-273-7800

Raleigh, NC
Tel: 919-844-7510

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110
Tel: 408-436-4270

Canada - Toronto
Tel: 905-695-1980
Fax: 905-695-2078

ASIA/PACIFIC

Australia - Sydney
Tel: 61-2-9868-6733

China - Beijing
Tel: 86-10-8569-7000

China - Chengdu
Tel: 86-28-8665-5511

China - Chongqing
Tel: 86-23-8980-9588

China - Dongguan
Tel: 86-769-8702-9880

China - Guangzhou
Tel: 86-20-8755-8029

China - Hangzhou
Tel: 86-571-8792-8115

China - Hong Kong SAR
Tel: 852-2943-5100

China - Nanjing
Tel: 86-25-8473-2460

China - Qingdao
Tel: 86-532-8502-7355

China - Shanghai
Tel: 86-21-3326-8000

China - Shenyang
Tel: 86-24-2334-2829

China - Shenzhen
Tel: 86-755-8864-2200

China - Suzhou
Tel: 86-186-6233-1526

China - Wuhan
Tel: 86-27-5980-5300

China - Xian
Tel: 86-29-8833-7252

China - Xiamen
Tel: 86-592-2388138

China - Zhuhai
Tel: 86-756-3210040

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444

India - New Delhi
Tel: 91-11-4160-8631

India - Pune
Tel: 91-20-4121-0141

Japan - Osaka
Tel: 81-6-6152-7160

Japan - Tokyo
Tel: 81-3-6880- 3770

Korea - Daegu
Tel: 82-53-744-4301

Korea - Seoul
Tel: 82-2-554-7200

Malaysia - Kuala Lumpur
Tel: 60-3-7651-7906

Malaysia - Penang
Tel: 60-4-227-8870

Philippines - Manila
Tel: 63-2-634-9065

Singapore
Tel: 65-6334-8870

Taiwan - Hsin Chu
Tel: 886-3-577-8366

Taiwan - Kaohsiung
Tel: 886-7-213-7830

Taiwan - Taipei
Tel: 886-2-2508-8600

Thailand - Bangkok
Tel: 66-2-694-1351

Vietnam - Ho Chi Minh
Tel: 84-28-5448-2100

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

Finland - Espoo
Tel: 358-9-4520-820

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Garching
Tel: 49-8931-9700

Germany - Haan
Tel: 49-2129-3766400

Germany - Heilbronn
Tel: 49-7131-67-3636

Germany - Karlsruhe
Tel: 49-721-625370

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Rosenheim
Tel: 49-8031-354-560

Israel - Ra'anana
Tel: 972-9-744-7705

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Padova
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Norway - Trondheim
Tel: 47-7289-7561

Poland - Warsaw
Tel: 48-22-3325737

Romania - Bucharest
Tel: 40-21-407-87-50

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Gothenberg
Tel: 46-31-704-60-40

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820

© 2012-2018 Microchip Technology Inc.

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for Interface - Specialised ***category:***

Click to view products by Microchip ***manufacturer:***

Other Similar products are found below :

CY7C910-51LMB MC33689DPEWR2 MC33975ATEKR2 MEC1632-AUE NVT4555UKZ RKSAS4 HMC677G32 LPC47N207-JV FTP-637DSL633R SM712GX04LF04-BA MC33689DPEW PCA9704PWJ MCW1001A-I/SS HOA6241-001 SC74HC4066ADTR2G AS3935-BQFT NCN5120MNTWG NCN5150DR2G NCN8025MTTBG C100N50Z4A DG407AK/883B SRT2-ATT01 TDA8035HN/C1/S1J LTC1694CS5#TRMPBF TLE9221SXXUMA2 DS90UB947TRGCRQ1 NCS2300MUTAG HMC677LP5E HMC677LP5ETR LTC1756EGN#PBF LTC1955EUH#PBF LT3669EUFD-2#PBF MXL1543BCAI MAX3170CAI+ XL1192D TLE9221SX CP82C59AZ KTU1109EFAA-TR CH368L CH7307C-DEF LTC1694CS5#TRPBF LTC1694IS5#TRM 73S8024RN-20IMF 73S8024RN-IL/F 78P2352-IGT/F DS2406+ DS2413P+ DS2413P+T&R DS28E17Q+ DS8113-RNG+