FCRAM I IP Core

User's Guide

October 2005
ipug34_02.0

Introduction

Fast Cycle RAM (FCRAM) is a DRAM technology with a specialized memory core technology that achieves faster random access times and offers lower power consumption than traditional DRAMs. FCRAM is a trademark of Fujitsu Ltd., Japan. FCRAM technology is an attractive solution for applications that require DRAM densities, low power consumption and fast random cycle performance approaching SRAM speeds.

Lattice's FCRAM I controller core provides an ideal solution for interfacing a user's application to FCRAM I technology. The core provides a simple user command interface on one side and a FCRAM I memory interface on the other side. The core handles all timing and control to the FCRAM I. It also checks the user commands for access violations and prevents bank collisions and read-write turnaround time violations. The core also incorporates a user configurable refresh controller that handles refreshes to the FCRAM I. This core allows designers to focus on the application rather than the FCRAM I interface resulting in a faster time to market.

This user's guide describes the FCRAM I controller core and explains how it can be implemented in Lattice's programmable device technologies.

The FCRAM I controller core comes with the documents and files listed below:

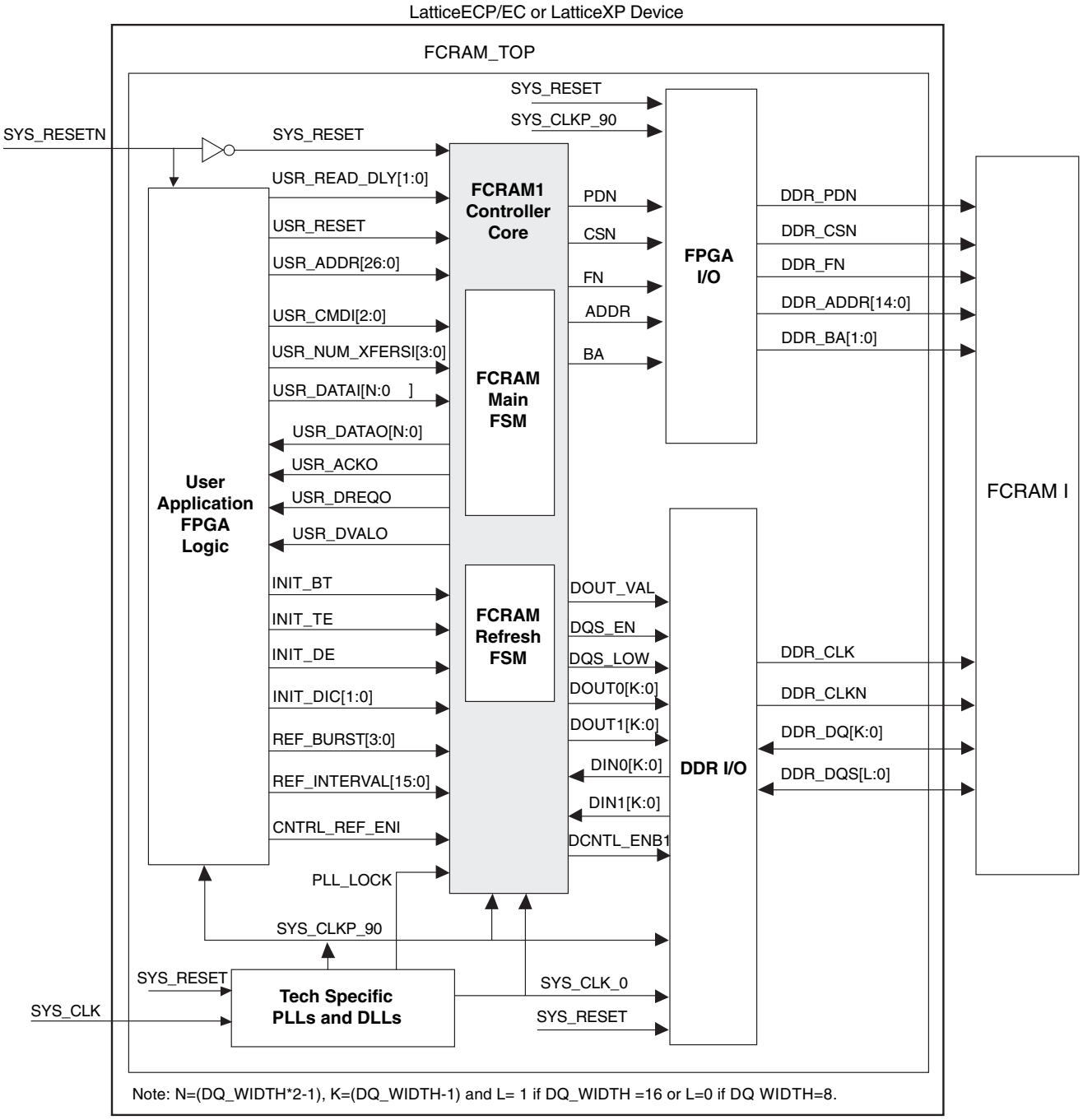
- Data sheet
- Gate level netlist
- RTL simulation model
- Core instantiation template

Features

- Supports 8- and 16-bit FCRAM I DDR interfaces for data
- Supports parameterizeable burst lengths of 2 or 4
- Supports parameterizeable CAS latencies of 3 or 4
- Supports controller initiated auto-refresh or user initiated auto/self-refresh
- Built-in initialization sequence automatically programs the FCRAM I with user's settings
- Supports bi-directional DDR interface between the controller and FCRAM I (DQ and DQS).
- Supports a target FCRAM I DDR interface frequency of 154 to167 MHz in a LatticeEC™, LatticeECP™, or LatticeXP™ device

Block Diagram

Figure 1. FCRAM I IP Core Block Diagram



General Description

As faster applications emerge in the networking and communications markets, there is a need for higher performance memories to meet the architectural requirements of these emerging applications. FCRAM memory technology addresses the needs of some of these emerging applications. For applications that require DRAM densities with random cycle performance approaching SRAM speeds, FCRAM provides a cost effective solution.

Lattice's FCRAM I controller core provides a simple and flexible solution to interfacing different FCRAM I memories to user application logic. On the memory side, the FCRAM I controller core interfaces to FCRAM I data bus widths with (configurable) 8- or 16-bit DDR memory interfaces. On the user side the core provides a simple command interface that accepts read, write and refresh commands. The core interfaces to 256 Mbit FCRAM I devices (speed grades -60/CAS latency 3 and 4 and -55/CAS latency 3).

Overview of FCRAM Operation

This section presents a brief overview of FCRAM I operation. Details on FCRAM I operation can be found in Toshiba or Fujitsu FCRAM I data sheets.

Currently there are specifications for two versions of FCRAM devices, sometimes denoted as FCRAM I and FCRAM II. FCRAM I are the first generation of the FCRAM devices that perform around 154MHz to 200MHz, have a Random access time (t_{RC}) between 25ns and 30ns, use a bi-directional data strobe signal and use SSTL2 I/O. FCRAM II are the second generation of the FCRAM devices that perform around 167MHz to 333MHz, have a Random access time (t_{RC}) between 20ns and 25ns, use a unidirectional data strobe signal and use SSTL18 or HSTL I/O. The controller design described in this document interfaces to FCRAM I devices.

FCRAM I is a Double Data Rate (DDR) based memory interface (i.e. data is transferred on both rising and falling edges of the clock, `DDR_CLK`). Most FCRAM devices have four banks and memory is addressed by row, column, and bank. Memory accesses commands (encoded in the FN and CSN signals) are presented on two consecutive cycles. On the first cycle the first command, row (upper address), and bank are presented. On the second cycle the second command and the column (lower address) are presented. Both read and write accesses to the FCRAM are performed in bursts of either two or four locations. Therefore, once a row, bank and column are selected the FCRAM accesses a consecutive number of columns with a given "burst length". The tables below show a summary of the commands and corresponding functions.

Table 1. First Command

Symbol	Function	~CS	FN	BA1~BA0	A14~A9	A8	A7	A6~A0
DESEL	Device Deselect	H	X	X	X	X	X	X
RDA	Read with Auto-close	L	H	BA	UA	UA	UA	UA
WRA	Write with Auto-close	L	L	BA	UA	UA	UA	UA

Table 2. Second Command

Symbol	Function	~CS	FN	BA1~BA0	A14~A13	A12~A11	A10~A9	A8	A7	A6~A0
LAL	Lower Address Latch (x16)	H	X	X	V	V	X	X	X	LA
LAL	Lower Address Latch (x8)	H	X	X	V	X	X	X	LA	LA
REF	Auto-Refresh	L	X	X	X	X	X	X	X	X
MRS	Mode Register Set	L	X	V	L	L	L	L	V	V

Notes:
1. L = logic Low, H = logic High, X = either L or H, V = Valid (specified value), BA = Bank address, UA = Upper address, LA = lower Address.
2. Refer to the State Diagram, Figure 2.

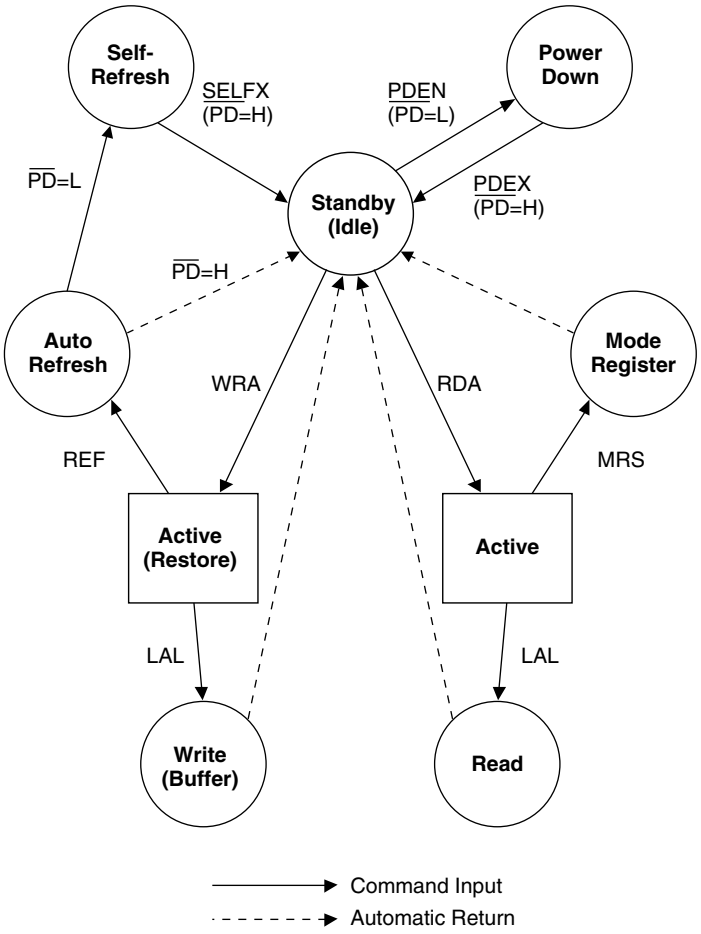
In Table 2, the x8 device A14 (VW0) and A13 (VW1) are data mask bits that control DQ0-DQ7. For the x16 device A14 and A13 are data mask bits that control DQ0-DQ7 and A12 and A11 are data mask bits that control DQ8-DQ15. The truth table for the mask bits is shown in Table 3.

Table 3. Mask Bits Truth Table

Symbol	Function	VW0	VW1
BL=2	Write All words	L	X
	Write First One Word	H	X
BL=4	Reserved	L	L
	Write All Words	H	L
	Write First Two Words	L	H
	Write First One Word	H	H

As shown in the tables above and the state diagram in Figure 2, the first cycle dictates if the access proceeds along the read command (RDA) or write command (WRA) branch of the state diagram. The second cycle either places the state machine in a LAL command, MRS command, or REF command. Command and address signals get latched by the FCRAM on a positive edge of the clock. FCRAM I uses a bi-directional data strobe signal (DQS) to capture data on reads and writes. During a read from FCRAM I the memory device provides the read data aligned with the rising edge of the DQS signal. The receiving circuit (FCRAM controller) needs to re-align the DQS to capture the read data. During a write to FCRAM I the transmitting circuit (FCRAM controller) needs to align the DQS edge to the center of the write data. For 32Mx8 FCRAM I there is a single DQS signal for the eight data bits. For 16Mx16 FCRAM I there are two DQS signals, one each for the upper and lower data bytes.

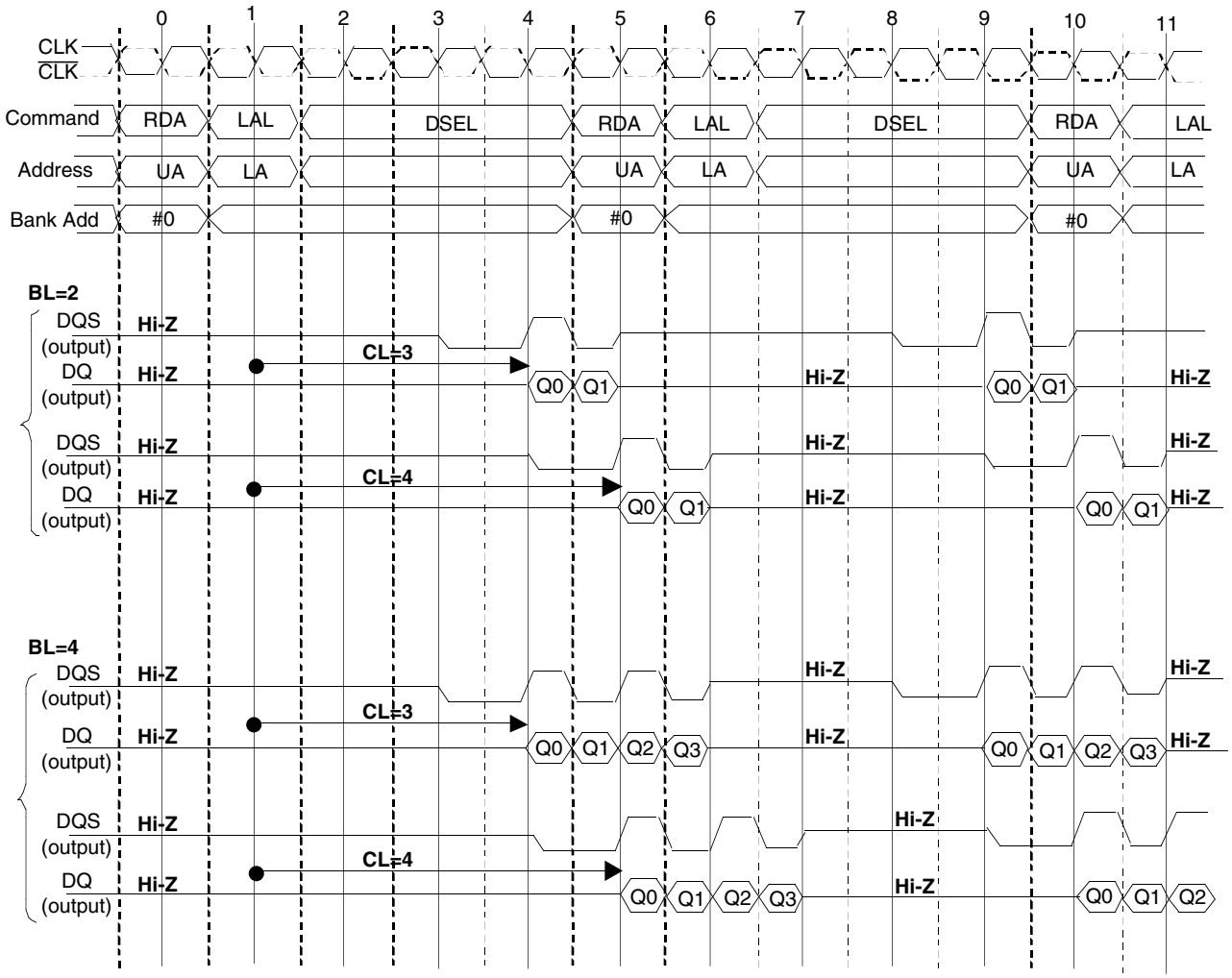
Figure 2. FCRAM State Diagram



Read Accesses

Figure 3 shows an example of read accesses to the same bank (bank 0), for burst lengths of two and four and CAS latencies of three and four. Note that the upper address (UA) and bank (bank 0) are active during the first command cycle (RDA) and the lower address (LA) is active during the second command cycle (LAL). Data becomes available to the receiving circuit CL cycles after the complete read command is issued.

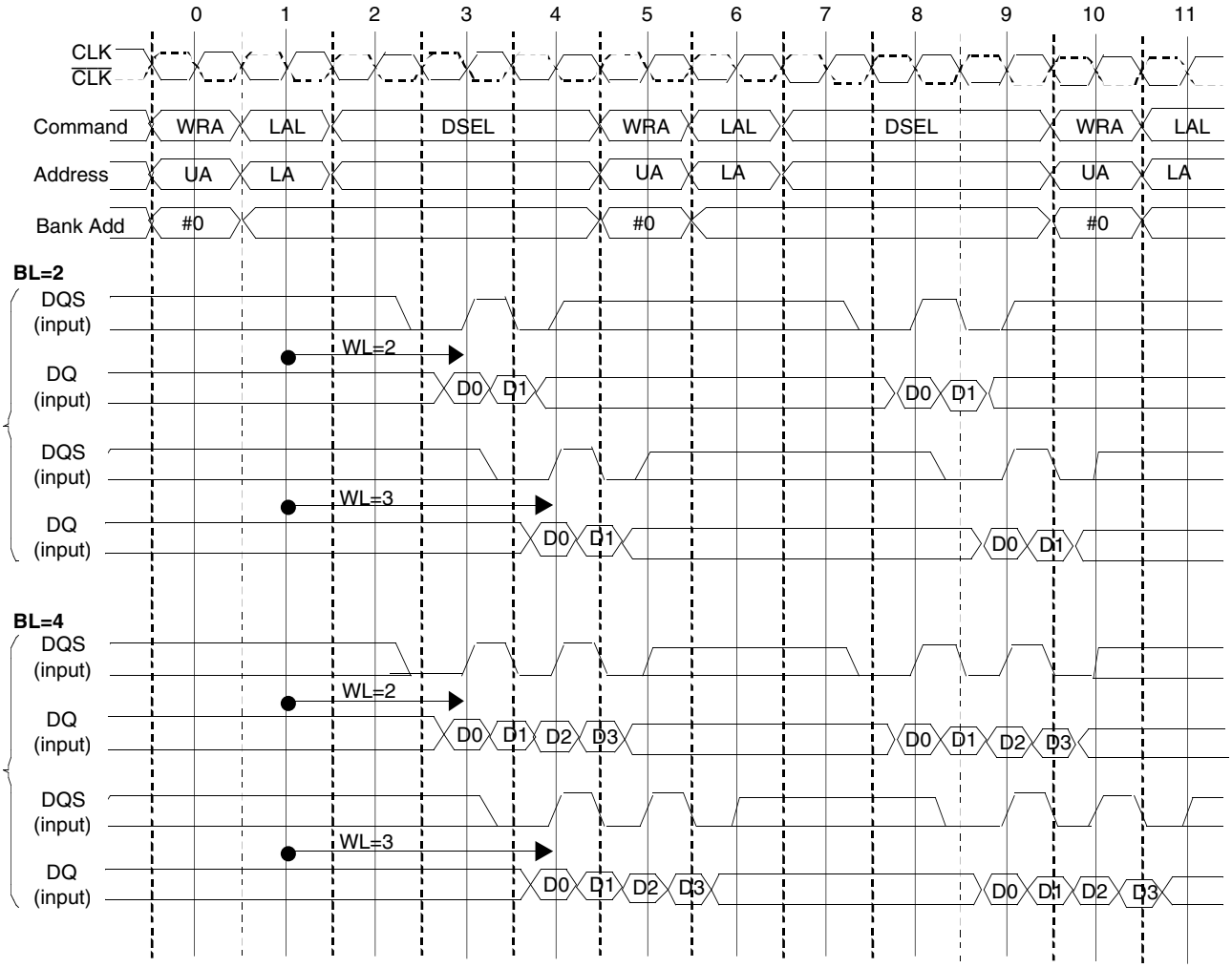
Figure 3. FCRAM I Read Timing



Write Accesses

Figure 4 shows an example of writes accesses to the same bank (bank 0), for burst lengths of two and four and CAS latencies of three and four. Note that the upper address (UA) and bank (bank 0) are active during the first command cycle (WRA) and the lower address (LA) is active during the second command cycle (LAL). Data must be stable for t_{DS} (Data input Setup Time from DQS) before the rising edge of the DQS signal. The first rising edge of DQS occurs WL (CL-1) cycles after the complete write command is issued.

Figure 4. FCRAM I Write Timing



Refresh

Since FCRAM is a DRAM based memory technology, it requires periodic refreshes to maintain the data written to any memory location. As shown in the state diagram and command tables, an auto refresh is accomplished by following the WRA command with a REF command. A self refresh is initiated if the PD_n pin is asserted low within the t_{FPDL} timing window during a REF command. The self refresh state is exited when PD_n is de-asserted. Note that the self refresh state is a special power down state in which the FCRAM input and output buffers are disabled (except for PD pin) and therefore, power dissipation in the device is lowered while in this state.

Mode Register Set (MRS)

FCRAM has mode specific parameters that are set by programming Mode Registers within the FCRAM. The MRS registers are set when RDA is issued as the first command and MRS as the second command. During the RDA cycle the address and bank are ignored, and during the second cycle (MRS command) the bank bits are used to determine which one of two mode registers is selected, the regular mode register (MRS) or the extended mode register (EMRS), while the address bits contain the mode register configuration data. Once the fields in the mode registers are set up, the register contents are maintained until the Mode register is set up again by another MRS command or power is lost. Table 4 describes the configuration fields in the two Mode registers:

Table 4. Title

Mode Register Field	Description
Regular Mode Register Fields (MRS)	
Burst Length (BL[2:0])	This is the number of columns that each read or write access will cycle through. BL can be set to two (0x2) or four (0x4).
Burst Type (BT)	This field sets the way the lower address bits sequence in a burst cycle. BT can be set to sequential (0) or interleaved (1).
CAS Latency (CL[2:0])	For a Read – Data becomes available to the receiving circuit CL cycles after the complete read command is issued. For a Write – The first rising edge of DQS occurs (CL-1) cycles after the complete write command is issued.
Test Mode (TE)	This field is reserved and should be set to zero.
Extended Mode Register Fields (EMRS)	
DLL Enable (DS)	The FCRAM DLL is enabled if this field is set to zero.
Output Driver Impedance Control (DIC[1:0])	This field sets the type of output driver on the FCRAM, to either a Normal (0x0), strong (0x1), weak (0x2) or weakest output driver (0x3).

Note: All FCRAM registers are set based on user configuration inputs during the INIT state, which occurs immediately after `sys_resetn` is asserted. Other than using `sys_resetn`, the user may not initiate a mode reg access.

Access Violations

In order for proper access to FCRAM, access timing specified in the vendor's FCRAM data sheet should be met. There are three major types of access violations that can occur:

1. **Bank collisions** – these access violations occur if the minimum wait times required before accessing the same bank are violated. According to the FCRAM specification, once a bank access occurs the user must wait TRC clock cycles before the user can access the same bank again, so any read, write or read write combination that accesses the same bank within an TRC clock cycles causes a bank conflict violation.

2. **Read-Write turnaround times** – these access violations occur if the minimum wait times required turning around from read to write or write to read accesses to different banks are violated. The write to read turnaround time TWRD is specified as one clock cycle so this violation should not occur, however, the read to write turnaround time TRWD needs to be met to avoid a read to write turnaround violation.

3. **Refresh time violations** – these violations occur if accesses to memory occur after the refresh counter has expired. According to the FCRAM specification the user must wait TREFI(min) before issuing another auto refresh and the maximum time between auto refreshes can not exceed TREFI(max).

Functional Description

The controller shown in the block diagram in Figure 1 provides a convenient user interface to the FCRAM I. The user provides a command (nop, read, write, or refresh) and an address (bank, row, and column) and the controller circuit handles all of the timing and control to the FCRAM I. The user application and user interface of the FCRAM I controller reside in the same IP_core clock domain (sys_clkp_90 clock). The controller handles all clock domain transfers necessary to communicate with the external FCRAM I. As shown in the block diagram the FCRAM I controller core has two main blocks, a main FSM and a refresh FSM. In addition the FCRAM top level wrapper has a place holder for the user's application, PLLs and DLLs for clock generation and timing, FPGA I/O buffers, and flip flops and DDR I/O cells to interface to the FCRAM memory. The user should substitute their own user application to access the external FCRAM I memory via the FCRAM I controller.

Table 5 refers to user configurable Parameters that can be used to custom configure the FCRAM I controller. These parameters need be set prior to synthesis. Table 6 lists the I/O used by the FCRAM I controller core. An overview of the FCRAM I controller is given below and the sections that follow describe the controller in more detail.

Controller

The controller first checks the user command for FCRAM I access violations such as bank collisions, read-write turnaround time violation, and checks the refresh counter. When the command can be executed, the controller generates the necessary signals to the FCRAM I. For a write access the controller transfers user write data from the sys_clkp_90 clock domain on the user side to the FCRAM I DDR data bus, and centers the data with the DQS strobe, which the FCRAM I uses to clock in the user data. The controller provides the user side with a data request signal (USR_DREQO) for the number of user data segments (clock cycles) the user is requesting by the USR_NUM_XFERSI input. When a read order is executed, the controller clocks in the read data from the FCRAM I by delaying the incoming DQS strobe to capture read data and then transfers the data back into the IP_core clock domain (sys_clkp_90) to present to the user along with a signal indicating valid data is available (USR_DVALO). When any command has been accepted by the controller, it provides an acknowledge signal (USR_ACKO) back to the user interface and the user can then supply the next command.

Initialization

On power-up the FCRAM I controller follows a pre-set initialization sequence whereby it programs the FCRAM I mode registers. On power-up the FCRAM I mode registers are undefined, and therefore must be programmed with the user's settings before read and write accesses to the FCRAM I can take place.

After power-up the user must hold the user reset signal to the controller (usr_reset) active for at least 200µs after the FPGA PLL locks. Once the user disables the usr_reset input, the controller will latch the user's initialization inputs, and write the MRS (mode register set) and EMRS (extended mode register set) with the user's parameters. The following initialization input signals must be set by the user prior to deasserting the usr_reset signal.

Table 5. User Initialization Parameters

Input Signal Name	Meaning	Values
init_bt	Burst Type	0 - Sequential, 1 - Interleaved
init_te	Test Mode	0 - Normal Operation
init_de	DLL Enable	0 - DLL Enabled, 1 - DLL Disabled
init_dic	Output Driver Impedance Control	0x0 - Normal Output Driver[1] 0x1 - Strong Output Driver 0x2 - Weaker Output Driver 0x3 - Weakest Output Driver
Ref_burst[3:0]	Number of Refreshes in a Burst	1 to 8
Ref_interval[15:0]	Refresh Interval to Wait in Clock Cycles	0 - 32767

1. Only option 0x0 (normal) is presently supported.

Read, Write Accesses

The user interface of the FCRAM I controller accepts a simple set of commands to perform reads, writes, and refreshes. The allowed commands are listed in Table 6.

Table 6. User Commands

Value of usr_cmdi[2:0]	Command
0xx	NOP
100	Write request
101	Self refresh
110	Read request
111	Auto refresh

Read Accesses: For reads from the FCRAM I, the user drives the usr_cmdi input requesting a read command, the usr_addri with the address to be read, and the usr_num_xfersi. The usr_cmdi[2:0] is the command

and needs to be set to 110 (Read Request). The usr_addri[26:0] is the user address and is broken down as follows:

usr_addri[26:25] = bank[1:0]
usr_addri[24:10] = row[14:0]
usr_addri[9:0] = col[9:0]

Note the x8 device needs 8 col bits while the x16 device only needs 7 col bits so, the unused upper col bits need to be zeroed out.

The usr_num_xfersi sets the number of 16-bit (32Mx8) or 32-bit (16Mx16) "user data segments" that need to be read. The controller checks for violations, adds the appropriate number of IDLE states if a violation would occur, and then performs the read when it is allowed.

When the read operation can be performed, the controller drives the read address to the FCRAM I, and then recovers the DDR read data from the FCRAM I using the DQS signal to clock data into the controller. The controller transfers the read data back to the IP_core clock domain to be read by the user. The controller will assert the usr_dvalo signal high for "usr_num_xferi" number of clock cycles when the read data is valid on the user side. Once the usr_acko signal is returned by the controller, the user can move on to the next command by changing the usr_cmdi and associated values.

When board delays approach multiples of a clock period the usr_read_dly[1:0] signal can be used to delay the latching of read data from the FCRAM by up to three clock cycles, note this is in addition to the CL latency. The usr_read_dly[1:0] is a binary encoded signal into the FCRAM I controller where "00" sets no additional delay, "01" sets one additional delay, "10" sets two additional clock delays and "11" sets three additional clock delays. Note, in most designs these input bits will be set to zero.

Write Accesses: For writes to the FCRAM I, the user drives the usr_datai, usr_addri, usr_num_xfersi, and usr_cmdi inputs to the controller. The usr_datai is the16-bit (32Mx8 FCRAM) or 32-bit (16Mx16 FCRAM) user write data. The format of the usr_addri is the same as stated above for the read accesses.

The usr_num_xfersi sets the number of 16-bit (32Mx8) or 32-bit (16Mx16) "user data segments" that need to be transferred. The usr_cmdi[2:0] is the command and needs to be set to 100 (Write Request). The controller checks for violations, and adds the appropriate number of IDLE states if a violation would occur, and then performs the write when it is allowed.

When the controller is ready to accept the user write data the controller will assert the usr_dreqo signal high, for "usr_num_xferi" number of clock cycles. The user must update the write data during the clock cycle following the usr_dreqo signal being asserted high (see Figure 1).

The controller transfers the user's data to the DDR clock domain, and drives the data onto the DDR outputs, along with the DQS strobe signal, which is used by the FCRAM I to clock data in. Once the usr_acko signal is returned, by the controller, the user can move on to the next command, by changing the usr_cmdi, and associated values.

Multiple User Data Transfers: After each read or write transaction the FCRAM automatically closes the row accessed and pre-charges the bank, so accesses to memory are always sized in single BURST_LENGTH chunks of either two or four data segments, where each data segment is the width of the FCRAM data bus (DDR_BUS_WIDTH). In order to support multiple data transfers within a single user command the controller supports a "number of transfers" port (USR_NUM_XFERSI) that lets the user specify up to fifteen "user data segments" per command. Each user data segment is two times the width of the FCRAM data bus (DDR_BUS_WIDTH*2).

For example when the BURST_LENGTH is set to two and USR_NUM_XFERSI = 1 (single user data segment), a single clock cycle on the user side will transfer or receive a single BURST_LENGTH chunk of two "memory data segments" on the memory side interface.

If the BURST_LENGTH is set to four, and USR_NUM_XFERSI = 2, two clock cycles on the user side will transfer or receive a single BURST_LENGTH chunk of four memory data segments on the memory side interface.

If the BURST_LENGTH is set to four, and USR_NUM_XFERSI = 3, three clock cycles on the user side will transfer or receive one and a half BURST_LENGTH chunks of four memory data segments on the memory side interface. Note however, in order to support odd number of USR_NUM_XFERSI (for example when USR_NUM_XFERSI =3) on a read access the controller will return data for two full BURST_LENGTH chunks of four memory data segments, but will "mask" out the final odd transfer with the usr_dvalo signal. On write accesses to odd numbers of USR_NUM_XFERSI (e.g. 3) the controller puts out two full BURST_LENGTH chunks of four memory data segments (i.e DQS signal continues through the end of the second burst length chunk) but sets data mask bits to "write all words" for the first write command during the LAL part of the write command, and sets the mask bits to "write first two words" during the LAL part of the second write command.

If the BURST_LENGTH is set to four, and USR_NUM_XFERSI = 4, four clock cycles on the user side will transfer or receive two BURST_LENGTH chunks of four memory data segments on the memory side interface.

Table 7 summarizes the above discussion.

Table 7. Title?

BURST_LENGTH	USR_NUM_XFERSI	Number of Clock Cycles on User Side (User Data Segments)	Number of Clock Cycles on Memory Side (Memory Side Data Segments)	Chunks Transferred	Mask Bits Used on Write Access
2	1	1	2	1	No
2	2	2	4	2	No
2	3	3	6	3	No
2	4	4	8	4	No
2	5	5	10	5	No
2	6	6	12	6	No
2	7	7	14	7	No
2	8	8	16	8	No
2	9	9	18	9	No
2	10	10	20	10	No
2	11	11	22	11	No
2	12	12	24	12	No
2	13	13	26	13	No
2	14	14	28	14	No
2	15	15	30	15	No
4	1	1	4 (first 2 clocks valid)	0.5	Yes
4	2	2	4	1	No
4	3	3	8 (first 6 clocks valid)	1.5	Yes
4	4	4	8	2	No
4	5	5	12 (first 10 clocks valid)	2.5	Yes
4	6	6	12	3	No
4	7	7	16 (first 14 clocks valid)	3.5	Yes
4	8	8	16	4	No
4	9	9	20 (first 18 clocks valid)	4.5	Yes
4	10	10	20	5	No
4	11	11	24 (first 22 clocks valid)	5.5	Yes
4	12	12	24	6	No
4	13	13	28 (first 26 clocks valid)	6.5	Yes
4	14	14	28	7	No
4	15	15	32 (first 30 clocks valid)	7.5	Yes

The controller also automatically increments the bank, column and row addresses during successive read or write accesses within a single command where USR_NUM_XFERSI is set greater than one. Note since each read or write access spans a "BURST_LENGTH" number of columns, when the bank address transitions from three to zero the controller's current address {bank, row, column} increments by the selected BURST_LENGTH (i.e. two or four). For multiple accesses within a single command the FCRAM memory is accessed first along banks, then across columns and finally down rows.

Refresh

There are two refresh modes available to the user. First is the user initiated refresh mode. In this mode, the user is responsible for generating refresh commands at the proper time to satisfy the FCRAM I. This is selected by setting the usr_ref_eni input to 0. In this mode, the user is responsible for initiating all refreshes via the user commands listed in Table 6. The user may enter the auto-refresh state by setting the usr_cmdi inputs to auto_refresh (111). Alternatively the user may enter the self-refresh state by setting the usr_cmdi input to the self-refresh command (110). The controller will stay in the self-refresh state until the self-refresh command is active. In order to comply with the FCRAM I self refresh exit timing specifications the user must issue at least one auto-refresh command immediately after exiting the self refresh state.

The second refresh mode is the controller initiated refresh mode. This is selected by setting the usr_ref_eni input to 1. In this mode the controller will automatically initiate an auto-refresh based on a fixed refresh interval determined by a timer (set by the ref_interval[15:0] inputs), which is programmed by the user. There is also a user-programmable refresh burst length parameter that sets the number of refreshes for a given burst (set by the ref_burst[3:0] inputs).

As mentioned earlier, according to the FCRAM specification the user (or controller) must wait T_{REFI}(min) before issuing another auto refresh and the maximum time between auto refreshes can not exceed T_{REFI}(max), however, if the refresh commands are presented consecutively (up to eight commands max) in a burst, then the refresh timing constraints can be distributed across a larger period. This allows for the user (or controller) to wait a longer period before performing another auto refresh, if burst of auto refresh commands are initially applied.

In the controller initiated refresh mode the FCRAM controller automatically calculates the required refresh interval and applies refresh commands within the appropriate time intervals. In the user-initiated refresh mode the user will need to provide auto refresh commands in the appropriate intervals. The equations below can be used to calculate the required refresh intervals:

$ref\_\ interval_{MIN} = T_{REFI}(min) \times (ref\_burst)/T_{CK}$

and

$ref\_\ interval_{MAX} = T_{REFI}(max) \times (ref\_burst)/T_{CK} - (user\_num\_xfersi + T_{RC})$

Note that since a read or write access could occur just before the end of the refresh interval the max refresh interval needs to be reduced by the time for the maximum number of transfers possible and any associated idle time after a memory access.

Examples

The figures below show examples of timing waveforms for write, read and refresh commands interfacing with a 32Mx8 FCRAM I. BURST_LENGTH = 4 and CAS_LATENCY = 4 in all the examples.

Figure 5 shows a write command to bank 1, row 0x0040 and column 0x00 (usr_addri = 0x2010000) and with USR_NUM_XFERSI = 2. The two 16-bit "user data segments" for the write are 0x9c1f and 0x2efc. Note since USR_NUM_XFERSI = 2, the memory DDR interface should transmit a single BURST_LENGTH chunk of four memory data segments. The command, address, number of transfers, and first write data segment are presented by the user at T1, at T2 the controller acknowledges the command and requests more data by asserting usr_dreqo high for two clock cycles. At T3 the user needs to provide the next data segment. Coincidently on the falling edge of cycle T3 the controller puts out a WRA command to the FCRAM by asserting both ddr_csn and ddr_fn low. Note the bank address is set to 0x1 and the ddr_addr is set to 0x0040. On the falling edge of cycle

T4 the controller puts out a LAL command, note that ddr_addr[14] = 1, and ddr_addr[13] = 0, i.e. the mask bit VW0 and VW1 are set to high and low respectively. This instructs the FCRAM to write all words. After a Write Latency of 3 (CL =4) from T5 to T8, the write data appears on the memory DDR interface as a single BURST_LENGTH chunk of four memory data segments, all of which are written into the FCRAM. The DQS strobes (output from controller) are centered aligned with the data. Note the data will be written to bank 0x1, row 0x0040 with 0x1f written to column 0x00, 0x9c written to column 0x01, 0xfc written to column 0x02 and 0x2e written to column 0x03.

Figure 5. Write Access Timing with Number of User Transfers Set to Two

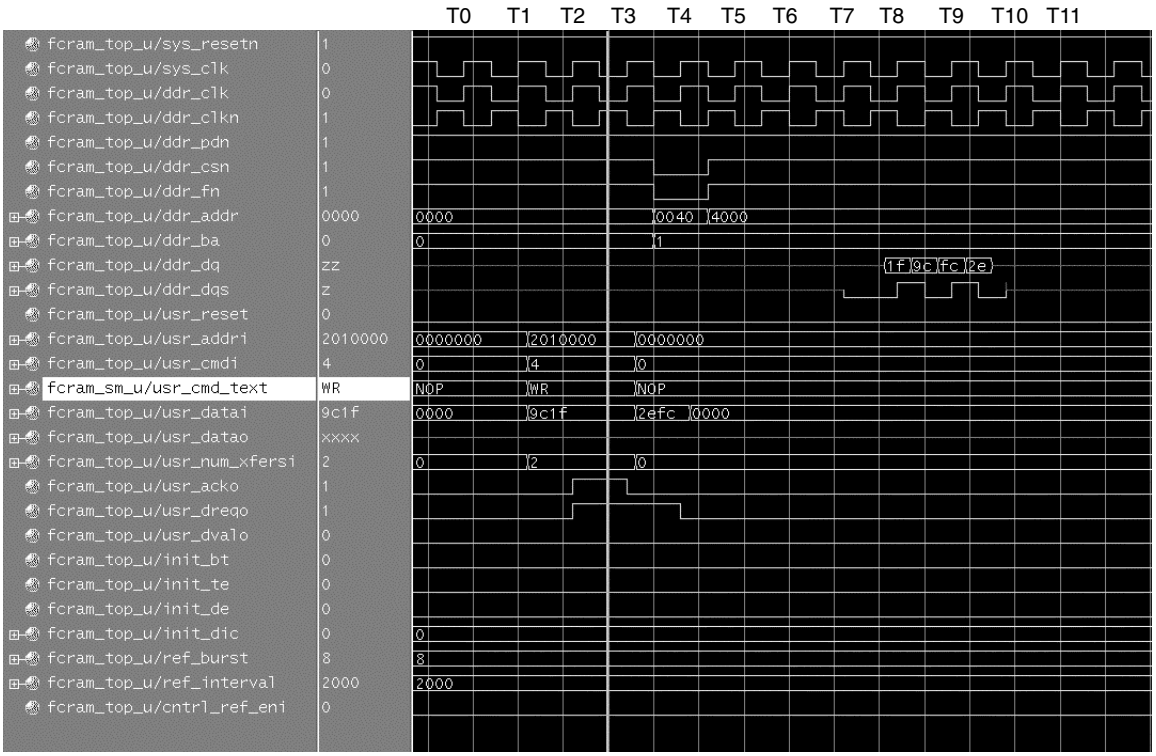


Figure 6 shows a read command to bank 1, row 0x0040 and column 0x00 (usr_addri = 0x2010000) and with USR_NUM_XFERSI = 2. Note this is the same address that was written to in the previous example and the two 16-bit "user data segments" read back should be 0x9c1f and 0x2efc and since USR_NUM_XFERSI = 2, the memory DDR interface will receive a single BURST_LENGTH chunk of four memory data segments. The command, address, and number of transfers are presented by the user at T0, at T1 the controller acknowledges the command. On the falling edge of cycle T2 the controller puts out a RDA command to the FCRAM by asserting `ddr_csn` low and keeping `ddr_fn` high. Note the bank address is set to 0x1 and the `ddr_addr` is set to 0x0040. On the falling edge of cycle T3 the controller puts out a LAL command, note that mask bits are not used during read commands. After a CAS Latency of 4 from T4 to T8, the read data appears on the memory DDR interface as a single BURST_LENGTH chunk of four memory data segments, with the DQS strobes (input to controller) edge aligned with the data. At T11 the read data is presented to the user side with the usr_dvalo signal demarking the two valid user data segments.

Figure 6. Read Access Timing with Number of User Transfers Set to Two

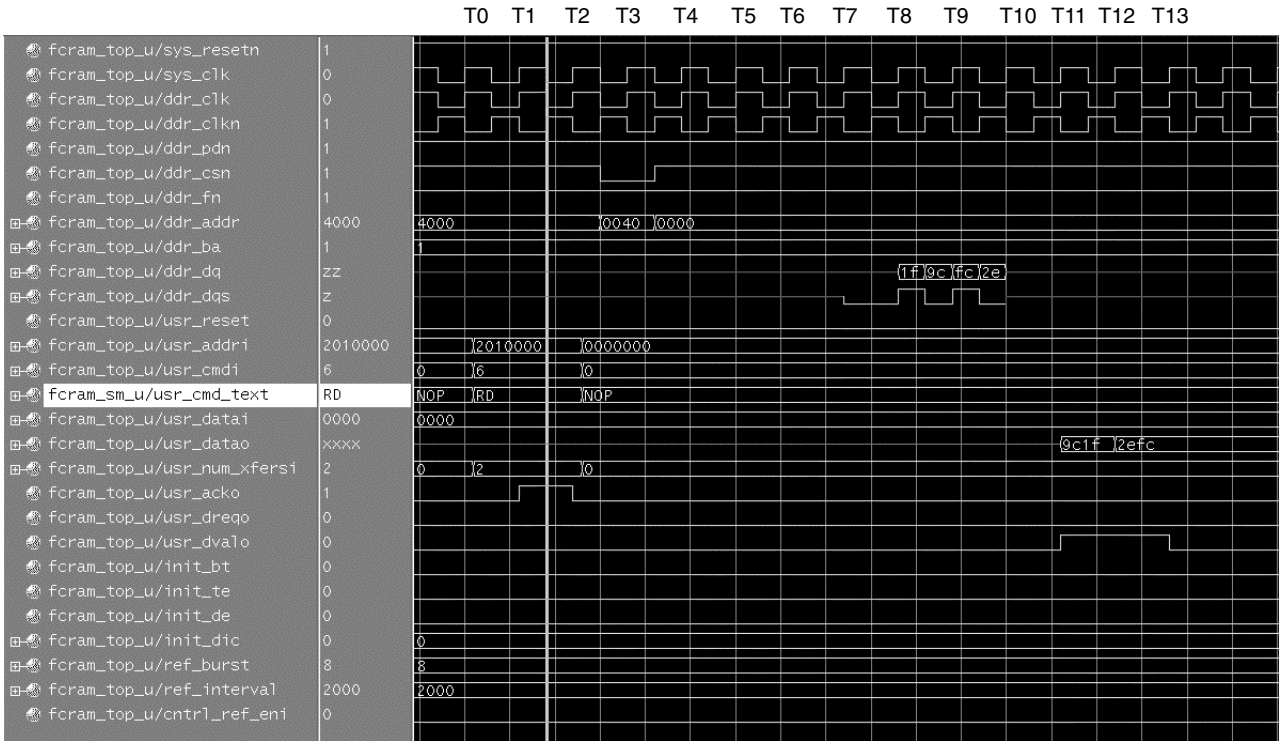


Figure 7 shows a write command to bank 2, row 0x0040 and column 0x00 (usr_addri = 0x4010000) and with USR_NUM_XFERSI = 3. The three 16-bit "user data segments" for the write are 0xb0e0, 0xa5a2 and 0x7110. Note since USR_NUM_XFERSI = 3, the memory DDR interface should transmit one and a half BURST_LENGTH chunks of four memory data segments. Note however, it is not possible to put out "half" BURST_LENGTH chunks. Instead, the controller does the following things:

- Puts out two full BURST_LENGTH chunks of four memory data segments (i.e., DQS signal continues through the end of the second burst length chunk)
- Sets data mask bits to "write all words" for the first write command during the LAL part of the write command
- Sets the mask bits to "write first two words" during the LAL part of the second write command.

The command, address, number of transfers, and first write data segment are presented by the user at T0, at T1 the controller acknowledges the command and requests more data by asserting `usr_dreqo` high for three clock cycles. At T2 and T3 the user needs to provide the next, respective, data segments. On the falling edge of cycle T2 the controller puts out a WRA command to the FCRAM by asserting both `ddr_csn` and `ddr_fn` low. Note the bank address is set to 0x2 and the ddr_addr is set to 0x0040. On the falling edge of cycle T3 the controller puts out a LAL command, note that ddr_addr[14] = 1, and ddr_addr[13] = 0, i.e. the mask bit VW0 and VW1 are set to high and low respectively. This instructs the FCRAM to write all words. On the falling edge of cycle T4 the controller puts out another WRA command to the FCRAM by asserting both `ddr_csn` and `ddr_fn` low. Note the bank address is set to 0x3 (since the first full BURST_LENGTH chunk incremented across four columns and the next bank needs to be targeted) and the `ddr_addr` is again set to 0x0040 (same row and column of next bank). On the falling edge of cycle T5 the controller puts out a LAL command, note that ddr_addr[14] = 0, and ddr_addr[13] = 1, i.e. the mask bit VW0 and VW1 are set to low and high respectively. This instructs the FCRAM to only write the first two words.

After a Write Latency of 3 (CL =4) from T4 to T7, the write data appears on the memory DDR interface as a two BURST_LENGTH chunks of four memory data segments, of which only six are written into the FCRAM. The DQS strobes (output from controller) are centered aligned with the data. Note data written to bank 0x2, row 0x0040 will be 0xe0 written to column 0x00, 0xb0 written to column 0x01, 0xa2 written to column 0x02 and 0xa5 written to col-

umn 0x03, and data written to bank 0x3, row 0x0040 will be 0x10 written to column 0x00, and 0x71 written to column 0x01.

Figure 7. Write Access Timing with Number of User Transfers Set to Three

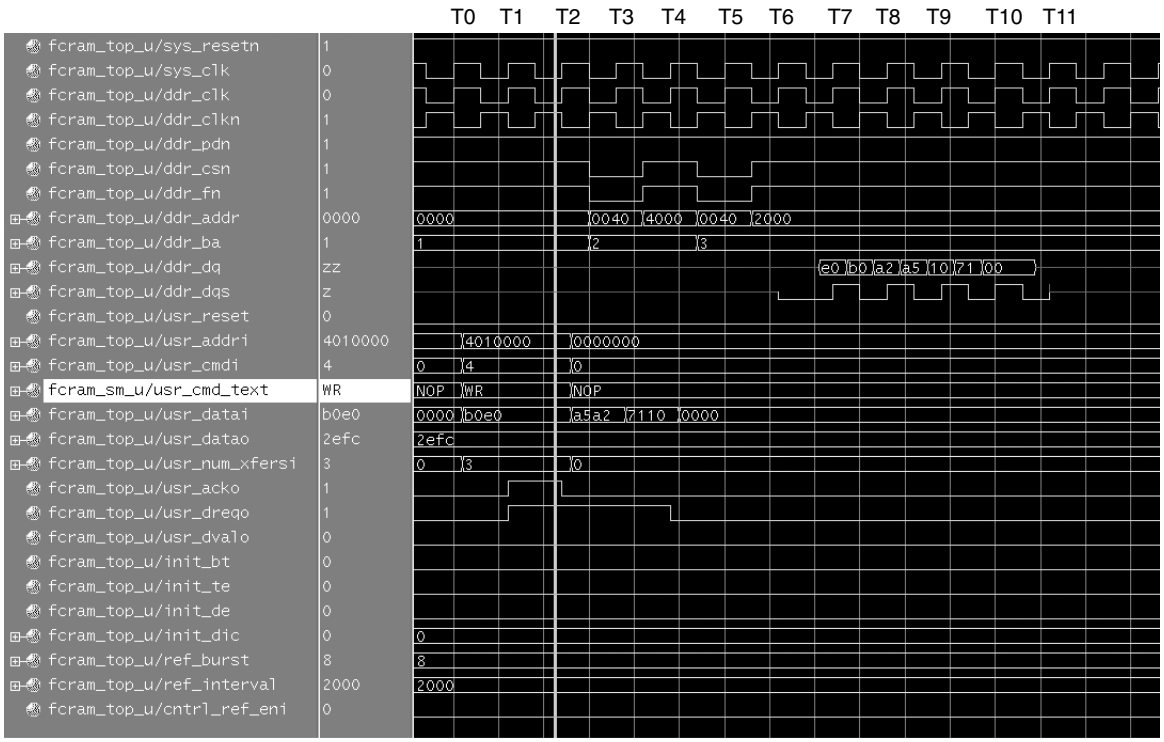


Figure 8 shows a read command to bank 2, row 0x0040 and column 0x00 (usr_addri = 0x4010000) and with USR_NUM_XFERSI = 3. Note this is the same address that was written to in the previous example and the three 16-bit "user data segments" read back should be 0xb0e0 and 0xa5a2 and 0x7110 and since USR_NUM_XFERSI = 3, the memory DDR interface will receive a two BURST_LENGTH chunks of four memory data segments each, of which only the first six segments will be valid. The command, address, and number of transfers are presented by the user at T0, at T1 the controller acknowledges the command. On the falling edge of cycle T2 the controller puts out a RDA command to the FCRAM by asserting `ddr_csn` low and keeping `ddr_fn` high. Note the bank address is set to 0x2 and the `ddr_addr` is set to 0x0040. On the falling edge of cycle T3 the controller puts out a LAL command, note that mask bits are not used during read commands. On the falling edge of cycle T4 the controller puts out another RDA command to the FCRAM by asserting `ddr_csn` low and keeping `ddr_fn` high. Note the bank address is set to 0x3 and the `ddr_addr` is set to 0x0040. On the falling edge of cycle T5 the controller puts out a LAL command, note again that mask bits are not used. After a CAS Latency of 4 from T4 to T8, the read data appears on the memory DDR interface as a two BURST_LENGTH chunks of four memory data segments each, with the DQS strobes (input to controller) edge aligned with the data. At T11 the read data is presented to the user side with the `usr_dvalo` signal demarking the three valid user data segments.

Figure 8. Read Access Timing with Number of User Transfers Set to Three

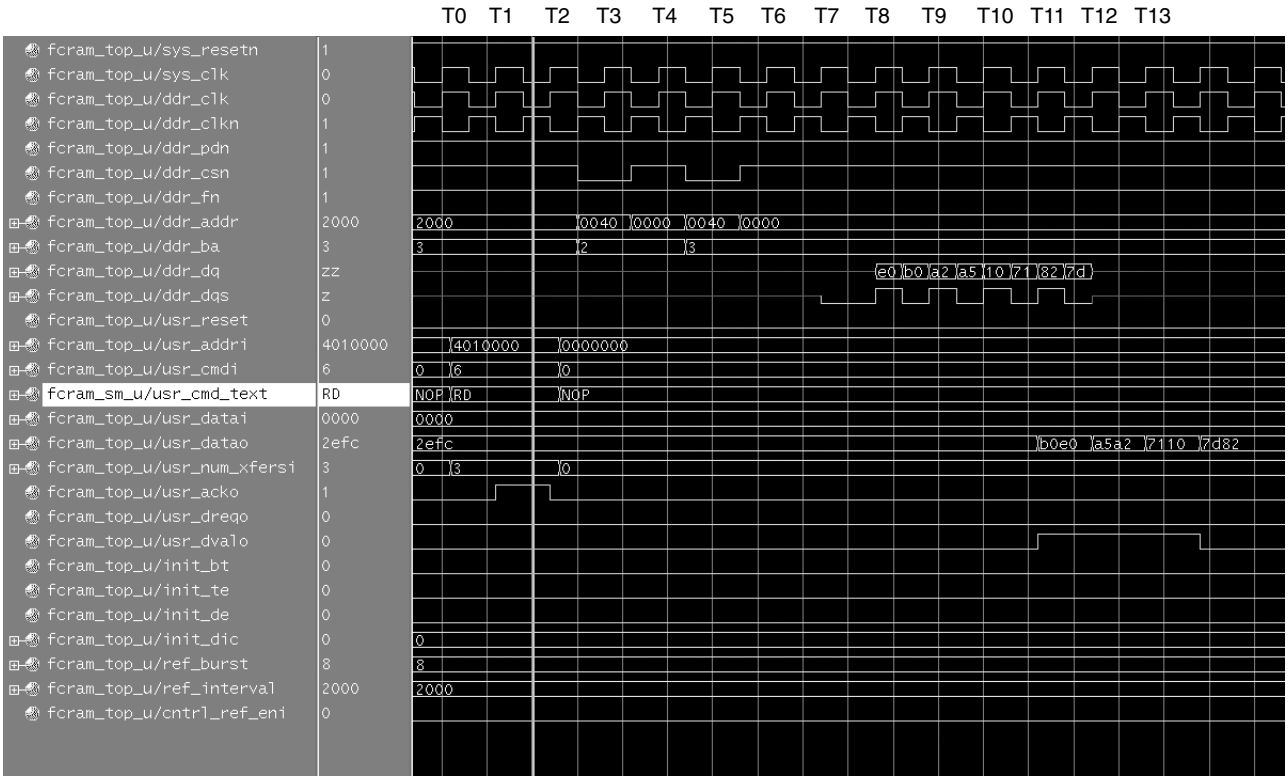


Figure 9 shows a self-refresh command initiated by the user (note `cntrl_ref_eni` is low), this is done by setting the user command to 0x5. The controller will acknowledge the command and put out a WRA command, followed by a REF command with `ddr_pdn` (power down pin to FCRAM) asserted low. The controller will stay in the self-refresh state until the self-refresh command is active. In order to comply with the FCRAM I self refresh exit timing specifications the user must issue at least one auto-refresh command immediately after exiting the self refresh state. Note that the self refresh state is a special power down state in which the FCRAM input and output buffers are disabled (except for PD pin) and therefore, power dissipation in the device is lowered while in this state.

Figure 9. Self-refresh Timing

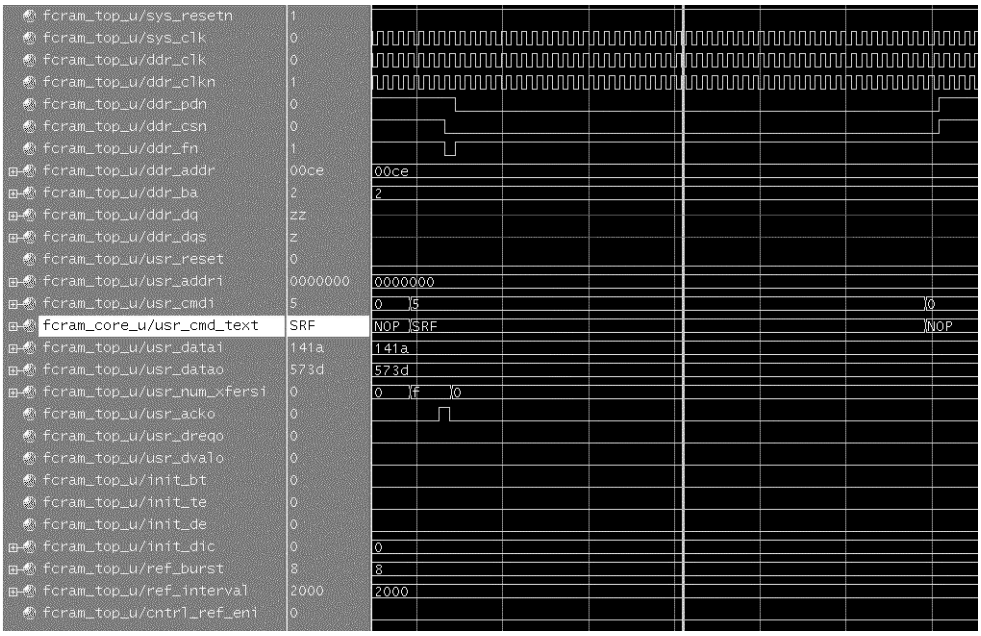
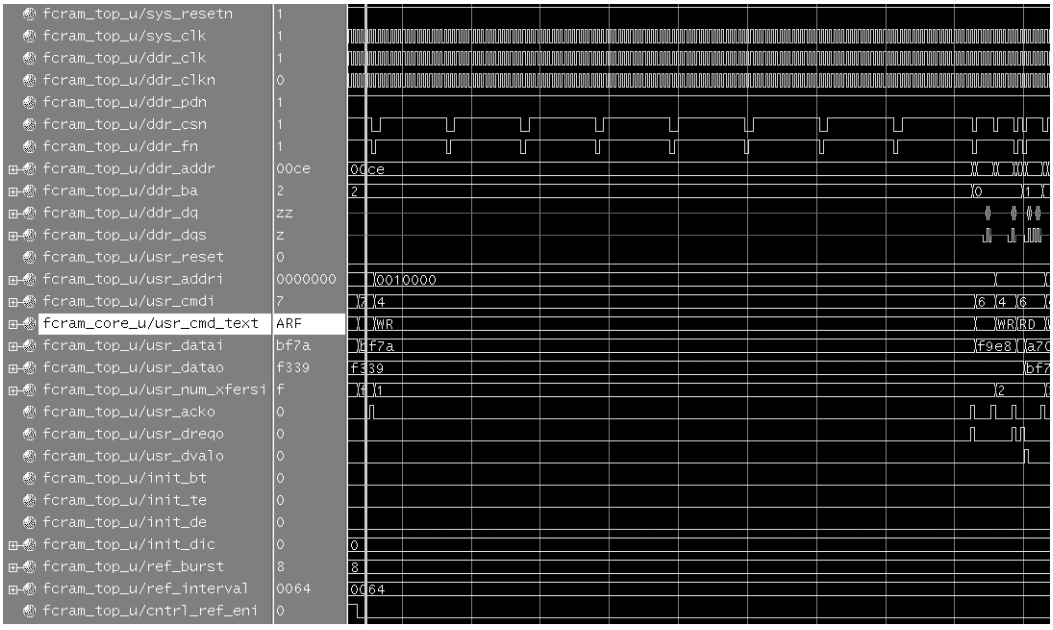


Figure 10 shows an auto-refresh command initiated by the user (note `cntrl_ref_eni` is low), this is done by setting the user command to 0x7. The controller will acknowledge the command and put out a WRA command, followed by a REF command with `ddr_pdn` (power down pin to FCRAM) set high. Note in the example shown the `ref_burst` is set to eight, therefore the controller puts out eight consecutive auto-refresh commands spaced IREFC clock cycles apart.

Figure 10. Auto-refresh Timing with Refresh Burst Set to Eight



Parameter Descriptions

The list of parameters used for configuring the FCRAM I controller core are listed below. The values of these parameters are to be set and must be done prior to synthesis or functional verification.

Table 8. User-configurable Parameters

Number	Parameter	Description	Choice	Default
1	DDR_BUS_WIDTH	Specifies the DDR data bus width of the FCRAM	8 bits wide 16 bits wide	8 bits wide
2	BURST_LENGTH	Specifies the FCRAM Burst Length	2,4	4
3	CAS_LATENCY	Specifies the FCRAM CAS latency	3,4	3
4	MEMORY_TYPE	Specifies the type of memory configuration of the FCRAM and IO type that the controller will interface to.	32Mx8 (FCRAM I, SSTL-2), 16Mx16 (FCRAM I, SSTL-2)	32Mx8 (FCRAM I, SSTL-2)

Signal Descriptions

Figure 9 defines all I/O interface ports available in this IP core.

Table 9. Signal definitions of FCRAM I Controller Core

Signal Name	Direction Input/Output	Width (Bits) [Each Signal]	Description
System Clock and Reset			
sys_clk	Input	1	System clock
sys_reset	Input	1	Core Reset – active high
sys_clkp_90	Input	1	System clock with phase 900 lagging sys_clk_0
sys_clk_0	Input	1	System clock reference out of PLL
pll_lock	Input	1	PLL Lock signal to controller core
FCRAM I Interface			
ddr_clk	Output	1	DDR clock to FCRAM I
ddr_clkn	Output	1	DDR clock (inverted) to FCRAM I
ddr_pdn	Output	1	DDR power down control
ddr_csn	Output	1	DDR chip select
ddr_fn	Output	1	DDR function control
ddr_addr	Output	15	DDR address
ddr_ba	Output	2	DDR bank address
ddr_dq	Biput	16, 8	DDR Data input/output for FCRAM I 16-bit interface DDR Data input/output for FCRAM I 8-bit interface
ddr_dqs	Biput	1	DDR write/read data strobe for FCRAM I
User Interface			
usr_reset	Input	1	User reset
usr_read_dly	Input	2	User Read Delay – Read data is latched an additional N clocks (after CL) after read command is issued (N=0:3)
usr_addri	Input	27	User address for read or write
usr_cmdi	Input	3	User command
usr_datao	Output	32, 16	User output data from FCRAM I controller
usr_datai	Input	32, 16	User input data to FCRAM I controller (x16 FCRAM) User input data to FCRAM I controller (x8 FCRAM)
usr_num_xfersi	Input	4	User selected number of transfers
usr_acko	Output	1	User acknowledge from controller
usr_dvalo	Output	1	User data out valid
init_bt	Input	1	Initial value of burst type
init_te	Input	1	Initial value of test mode
init_de	Input	1	Initial value of DLL disable
init_dic	Input	2	Initial value of output driver impedance control
ref_burst	Input	4	User refresh parameter - burst length
ref_interval	Input	16	User refresh parameter - refresh interval
cntrl_ref_eni	Input	1	Controller refresh enable
usr_dreqo	Output	1	User is requested for new write data

Timing Specifications

DDR Interface

AC Timing for the FCRAM I is shown below. Details of FCRAM I timing can be found in the FCRAM I vendor's data sheets.

Figure 11. AC Timing for FCRAM I Write

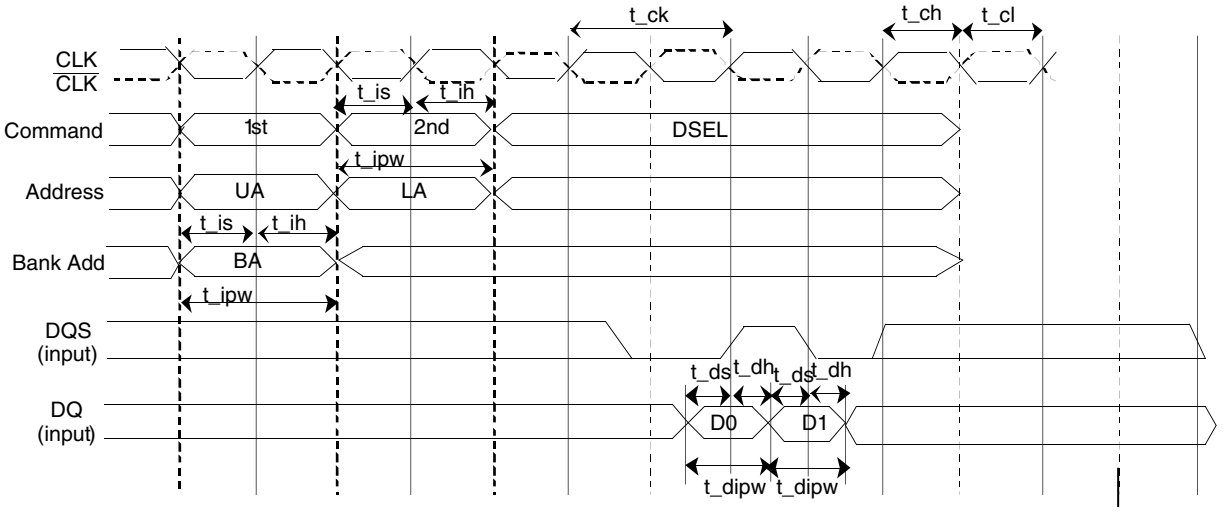


Figure 12. AC Timing for FCRAM I Read

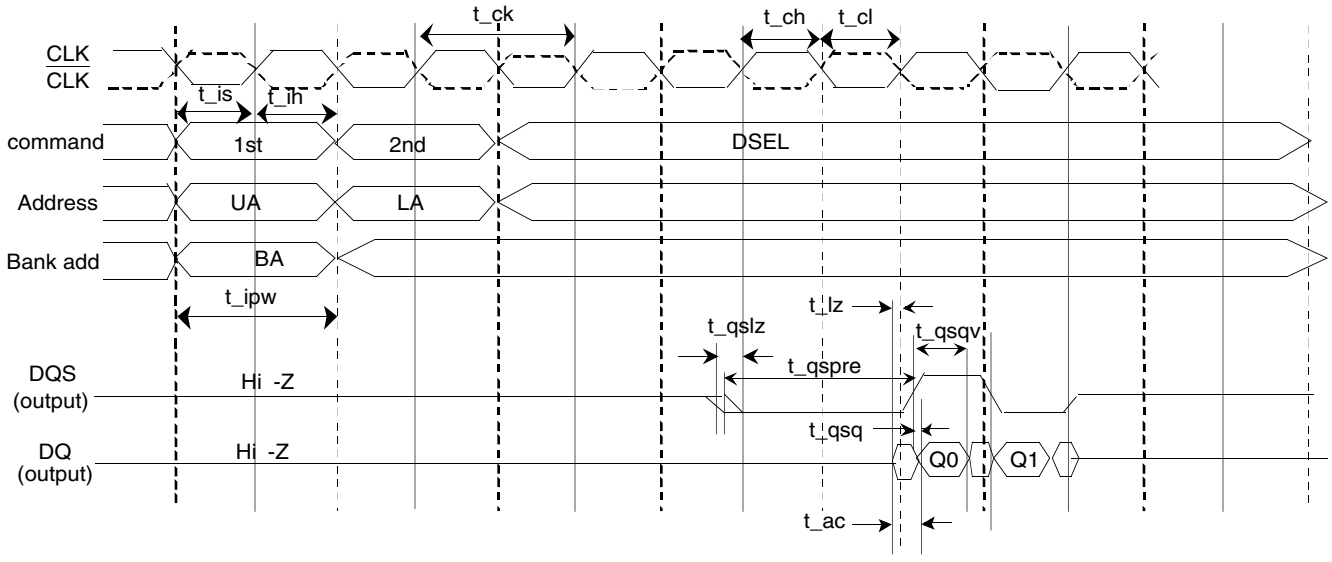


Table 10. FCRAM I Timing Characteristics

Symbol	Parameter		-50		-55		-60		Units
			Min.	Max.	Min.	Max.	Min.	Max.	
t_ck	Clock Cycle Time	CL=3	5.5	8.5	6	12	6.5	12	ns
		CL=4	5	8.5	5.5	12	6	12	ns
t_ch	Clock High Time		0.45xt_ck	—	0.45xt_ck	—	0.45xt_ck	—	ns
t_cl	Clock Low Time		0.45xt_ck	—	0.45xt_ck	—	0.45xt_ck	—	ns
t_qsq	Data output skew from DQS		—	0.4	—	0.45	—	0.5	ns
t_qspre	DQS (read) Preamble pulse width		0.9x t_ck -0.2	1.1x t_ck +0.2	0.9x t_ck -0.2	1.1x t_ck +0.2	0.9x t_ck -0.2	1.1x t_ck +0.2	ns
t_hp	CLK half period (minimum of actual t_ch, t_cl)		Min (t_ch, t_cl)	—	Min (t_ch, t_cl)	—	Min (t_ch, t_cl)	—	ns
t_qsqv	Data Output Valid time from DQS		t_hp-0.55	—	t_hp-0.6	—	t_hp-0.65	—	ns
t_ds	Data Input Setup Time from DQS		0.5	—	0.5	—	0.6	—	ns
t_dh	Data Input Hold Time from DQS		0.5	—	0.5	—	0.6	—	ns
t_dipw	Data Input Pulse width		1.5	—	1.5	—	1.9	—	ns
t_is	Command/address input Setup Time		0.9	—	0.9	—	1.0	—	ns
t_ih	Command/address input Hold Time		0.9	—	0.9	—	1.0	—	ns
t_ipw	Command/address Input Pulse width		2.0	—	2.0	—	2.2	—	ns
t_lz	Data-out Low Impedance time from CLK		-0.65	—	-0.75	—	-0.85	—	ns
t_hz	Data-out High Impedance time from CLK		—	0.65	—	0.75	—	0.85	ns
t_qslz	DQS-out Low Impedance time from CLK		-0.65	—	-0.75	—	-0.85	—	ns
t_qshz	DQS-out High Impedance time from CLK		-0.65	0.65	-0.75	0.75	-0.85	0.85	ns

Some of the DC electrical specifications for FCRAM I are as follows:

Table 11. FCRAM I DC Electrical Specifications

Symbol	Parameter	Min	Typ	Max	Units
V_{DD}	Power supply voltage	2.35	2.5	2.65	V
V_{DDQ}	Power supply voltage (for I/O buffer)	2.35	V_{DD}	V_{DD}	V
V_{REF}	Input reference voltage	V_{DDQ}/2 x 96%	V_{DDQ}/2	V_{DDQ}/2 x 104%	V
V_{IH} (DC)	Input DC high voltage	V_{REF} + 0.2	—	VREF + 0.2	V
V_{IL} (DC)	Input DC low voltage	-0.1	—	VREF - 0.2	V

See the vendor data sheets for details on DC electrical specifications

References

- Toshiba TC59LM814/06CFT-50,-55,-60 Data Sheet
- LatticeECP/EC Family Data Sheet, Lattice Semiconductor
- LatticeXP Family Data Sheet, Lattice Semiconductor
- ispLEVER® User Documentation, Lattice Semiconductor

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Appendix for LatticeECP/EC FPGAs

Table 12. Performance and Utilization[1]

Parameter File	SLICEs	LUTs	Registers	I/Os	sysMEM™ EBRs	f_{MAX} (MHz)
fcram_one_e2_1_001.lpc	469	588	494	129	0	166.66

1. Performance and utilization characteristics are generated using LFEC20E-5F672C in Lattice's ispLEVER 4.2 software.

Supplied Netlist Configurations

The Ordering Part Number (OPN) for FCRAM I IP on LatticeECP/EC devices is FCRAM-ONE-E2-N1 (for all configurations of the netlist package). Table 12 lists the evaluation netlists that can be downloaded from the Lattice web site at www.latticesemi.com.

You can use the IPexpress software tool to help generate new configurations of this IP core. IPexpress is the Lattice IP configuration utility, and is included as a standard feature of the ispLEVER design tools. Details regarding the usage of IPexpress can be found in the IPexpress and ispLEVER help system. For more information on the ispLEVER design tools, visit the Lattice web site at: www.latticesemi.com/software.

Appendix for LatticeXP FPGAs

Table 13. Performance and Utilization[1]

Parameter File	SLICEs	LUTs	Registers	I/Os	sysMEM™ EBRs	f_{MAX} (MHz)
fcram_one_xm_1_001.lpc	407	512	512	129	0	166.66

1. Performance and utilization characteristics are generated using LFXP10C-5F388C in Lattice ispLEVER® 5.0 software.

Supplied Netlist Configurations

The Ordering Part Number (OPN) for FCRAM I IP on LatticeXP devices is FCRAM-ONE-XM-N1 (for all configurations of the netlist package). Table 13 lists the evaluation netlists that can be downloaded from the Lattice web site at www.latticesemi.com.

You can use the IPexpress software tool to help generate new configurations of this IP core. IPexpress is the Lattice IP configuration utility, and is included as a standard feature of the ispLEVER design tools. Details regarding the usage of IPexpress can be found in the IPexpress and ispLEVER help system. For more information on the ispLEVER design tools, visit the Lattice web site at: www.latticesemi.com/software.

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for Development Software ***category:***

Click to view products by Lattice ***manufacturer:***

Other Similar products are found below :

RAPPID-560XBSW RAPPID-567XFSW DG-ACC-NET-CD SRP004001-01 SW006021-1NH SW163052 SYSWINEV21 Core429-SA SW500006-HPA CWP-BASIC-FL W128E13 CWP-PRO-FL SYSMACSE210L SYSMACSE203L AD-CCES-NODE-1 NT-ZJCAT1-EV4 CWA-BASIC-FL RAPPID-567XKSW CWA-STANDARD-R SW89CN0-ZCC CWA-LS-DVLPR-NL VDSP-21XX-PCFLOAT RAPPID-563XMSW IPS-EMBEDDED SWR-DRD-L-01 SDAWIR-4532-01 SYSMAC-SE201L MPROG-PRO535E AFLCF-08-LX-CE060-R21 WS02-CFSC1-EV3-UP SYSMAC-STUDIO-EIPCPLR LIB-PL-PC-N-1YR-DISKID SYSMACSE2XXL LS1043A-SWSP-PRM 1120270005 1120270006 MIKROBASIC PRO FOR FT90X (USB DONGLE) MIKROC PRO FOR AVR (USB DONGLE LICENSE) MIKROC PRO FOR FT90X (USB DONGLE) MIKROBASIC PRO FOR AVR (USB DONGLE LICEN MIKROBASIC PRO FOR FT90X MIKROC PRO FOR DSPIC30/33 (USB DONGLE LI MIKROC PRO FOR FT90X MIKROC PRO FOR PIC32 (USB DONGLE LICENSE 52202-588 MIKROPASCAL PRO FOR ARM (USB DONGLE LICE MIKROPASCAL PRO FOR FT90X MIKROPASCAL PRO FOR FT90X (USB DONGLE) MIKROPASCAL PRO FOR PIC32 (USB DONGLE LI SW006021-2H