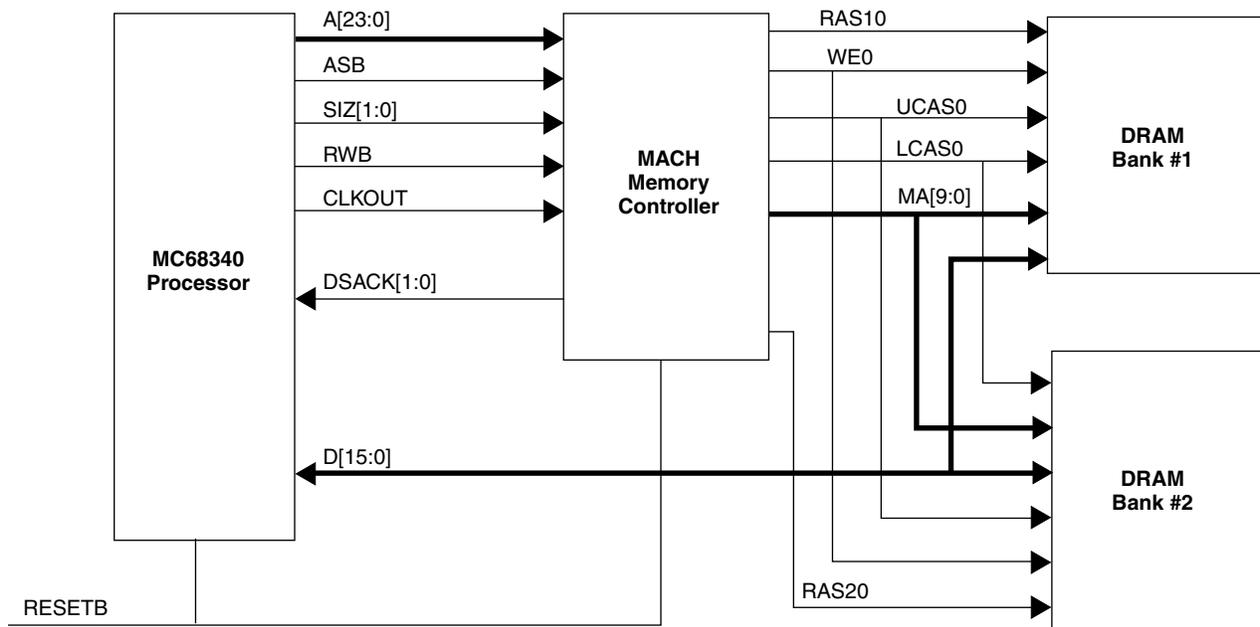


Introduction

Fast Page Mode DRAM (FPM DRAM) offers improved speed over standard DRAM since memory accesses performed within the same address row (page) require a precharge only for the first access. Subsequent accesses within the page can then be made without the time penalty of additional precharge cycles.

Although it has lost popularity in some areas to faster technologies such as double data rate (DDR) SDRAM, FPM DRAM is still a very cost-effective solution when the highest performance is not required. The Fast Page Mode DRAM controller described in this document interfaces to the MC68340, a popular embedded microprocessor. Due to the processor-dependent nature of memory control, the design will likely require modification if implemented for another processor family.

Figure 1. Fast Page Mode DRAM System



There are many kinds of design techniques for memory control such as EDC (Error Detection and Correction), parity generation, bank interleaving and various refresh methods, however, only the following essential issues are discussed in this document to describe the memory control functions with effectiveness:

- Address decoding and multiplexing
- Bank selection and data sizing
- RAS, CAS timing generation
- Interfacing to the processor
- Refresh request generation
- Random access timing
- Page mode access timing

In order to design a DRAM controller successfully, the timing and functionality of the following four major outputs are carefully considered:

- RAS (Row Address Strobe)
- CAS (Column Address Strobe)
- Data-ready
- Multiplexed address

Figure 1 illustrates the diagram of typical system level implementation using this DRAM controller design.

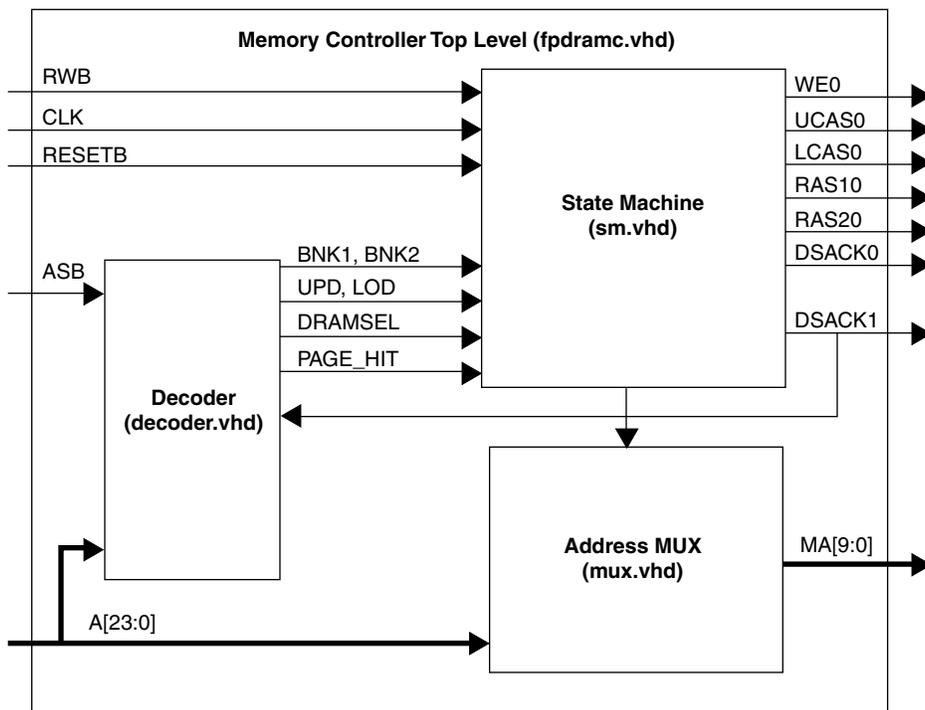
Table 1. Pin Descriptions

Pin Name	Type	Pin Description
A[23:0]	In	Address Bus – Defines the address of the data to be transferred during a bus cycle.
ASB	In	Address Strobe – Indicates that a valid address from the CPU is on the address bus.
SIZ[1:0]	In	Size – Indicates the number of bytes remaining to be transferred for the current cycle.
RWB	In	Read/Write# – Indicates the direction of data transfer on the bus.
CLKOUT	In	System Clock output from the CPU.
DSACK[1:0]	Out	Data and Size Acknowledge – During DRAM access cycle, the controller asserts DSACK[1:0] to indicate the data ready. These signals also indicate to the CPU the size of the DRAM port (16 bits).
RESETB	In	Active Low System Reset.
MA[9:0]	Out	Memory Address. These signals are the multiplexed outputs for the DRAM address lines from the CPU address and provide the row and column address for the access of the DRAM.
WEO	Out	Active Low DRAM Write Enable – Controls the write operation and provides the early write cycle timing to the DRAM.
RAS[1:2]O	Out	Active Low Row Address Strobe – Provides the strobe for a row address on the DRAM address lines. The RAS1O selects the memory bank1, while the RAS2O enables the memory bank2.
UCASO	Out	Active low Column Address Strobe – Provides the strobe for the upper column address on the DRAM address lines to select the upper byte of the 16-bit data lines of the DRAM.
LCASO	Out	Active low Column Address Strobe – Provides the strobe for the lower column address on the DRAM address lines to select the lower byte of the 16-bit data lines on the DRAM.

Functional Description

The functional block diagram of the FPM DRAM controller is shown in Figure 2. It consists of three modules: a decoder, an address multiplexer, and a state machine.

Figure 2. Block Diagram



Decoder Module

The decoder block decodes the following output signals:

DRAMSEL	DRAM Memory Area (0000000h ~ 07FFFFFFh)
BNK1	DRAM Bank1 Select (0000000h ~ 01FFFFFFh) normal access And (0400000h ~ 05FFFFFFh) page access
BNK2	DRAM Bank2 Select (0200000h ~ 03FFFFFFh) normal access And (0600000h ~ 07FFFFFFh) page access
UPD	Upper Byte Data Enable (used to generate UCASO)
LOD	Lower Byte Data Enable (used to generate LCASO)
PAGE_HIT	Active when A22 = '1' in a bus cycle. This corresponds to the memory regions 0400000h ~ 05FFFFFFh and 0600000h ~ 07FFFFFFh.

DRAM Address Multiplexer Module

The DRAMs targeted in this design latch 10 address bits at a time to generate the full 20-bit address. The first half of the address is latched when the RAS signal is asserted. The second half is latched when CAS is asserted. The address multiplexer block is used to convert the 20-bit address output from the CPU into two 10-bit addresses for input to the DRAMs.

To meet the DRAM timing requirement, the multiplexer select signal (DD_RASB) is delayed from the RAS signal by passing through two internal feedback paths of the device. This is accomplished by creating extra signals and forcing the software to keep them using source-code attributes. More levels of delay may be required, depending on the speed of the CPLD and DRAM, in order to meet the RAS address hold time requirement.

State Machine Module

The state machine implements the central control mechanism of the DRAM controller. It provides arbitration between memory accesses and refresh cycles, and controls the timing of the various DRAM interface signals to ensure correct operation. There are 12 states defined to perform this function.

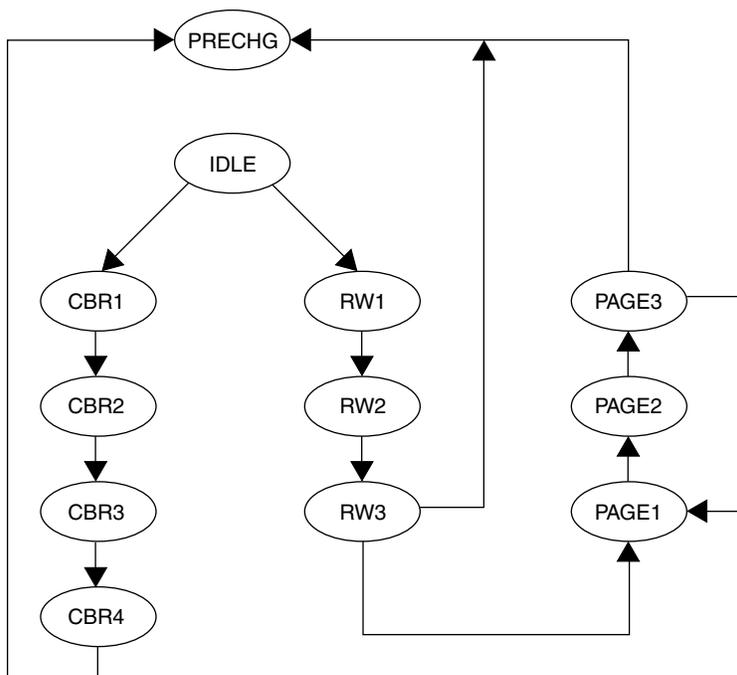
State	Description
IDLE	Idle State or no service
RW1	Initial or Random Access State 1
RW2	Initial or Random Access State 2
RW3	Initial or Random Access State 3
PAGE1	Page Access State 1
PAGE2	Page Access State 2
PAGE3	Page Access State 3
CBR1	CAS-Before-RAS Refresh State 1
CBR2	CAS-Before-RAS Refresh State 2
CBR3	CAS-Before-RAS Refresh State 3
CBR4	CAS-Before-RAS Refresh State 4
PRECHG	RAS Pre-Charge State

A random access cycle is initiated when the microprocessor asserts the address strobe signal (ASB) and the desired address is in the random-access region. The state machine starts from the IDLE State and moves through RW1, RW2, and RW3. During this time, The appropriate RAS line (RAS1O or RAS2O) is asserted along with DSACK. Depending on the transfer size UCASO and/or LCASO are then asserted. Additional cycles must pass through PRECHARGE and IDLE before additional accesses.

Page mode accesses occur when the address falls within the page-mode region. Like the random cycle, a page-mode cycle also first transitions from IDLE through RW1, RW2, and RW3. However, instead of moving to PRECHARGE, the state-machine instead moves through PAGE1, PAGE2, and PAGE3 for additional accesses. This can continue without precharge-time penalty until the microprocessor stops the transfer or an access is required within another page.

A CAS-before-RAS refresh cycle takes the CBR1, CBR2, CBR3, and CBR4 path when the refresh counter reaches a set value. During this time, a refresh acknowledge signal (REFACK) is asserted while CAS, and then RAS, are asserted to both memory banks in order to refresh all memory simultaneously.

Figure 3. State Diagram



The state machine module also contains a 9-bit synchronous refresh counter, clocked by the system clock, that generates a terminal-count signal when the count reaches a value of 186 HEX. Assuming a 25 MHz system clock, the terminal count occurs every 15.6 μ s and acts as a clock enable for the REFREQ signal. Once the state machine recognizes the REFREQ, it goes to the CBR1 state and asserts REFACK signal. The refresh counter is cleared by the REFACK. After being cleared, the refresh counter begins counting again to generate the next refresh cycle.

Design Considerations

The design of DRAM controllers for the Motorola 68000 family CPUs usually require one to three wait states for an access cycle depending on the memory performance and configuration. A page mode access can eliminate these wait states because no RAS pre-charge time is required, and the RAS access time is eliminated after the first access. As long as the CPU issues the addresses within the same DRAM page, the DRAM controller can accomplish the access cycle with minimum CPU clock cycles. In this design, a page mode access requires three CPU clock cycles. Non-page mode and initial accesses need 4 to 6 clock cycles, depending on how closely spaced the CPU accesses are.

Here is the brief memory controller specification implemented in this design:

- 4Mbytes dual bank 16 bits wide memory control
 - Bank0: 2 Mbytes (000000H - 1FFFFFFH)
 - Bank1: 2 Mbytes (200000H - 3FFFFFFH)
- Interface with MC68340-25MHz embedded processor
- Use two 60 ns 16 Mbits DRAM (1M x 16)
- Support random and page mode read/write access
- Refresh
 - Built-in refresh timing generator
 - CAS-before-RAS refresh mode

For this memory controller design, two 60ns fast-page-mode 1Mbit x 16 DRAMs are used. A mechanism is needed to distinguish between standard memory accesses and page mode accesses. This design uses address bit A22 to differentiate between these two types of accesses. A memory map for the memory system follows:

Address Range	Access type	Bank size
000000H - 1FFFFFFH	Normal access to Bank1	2 MB
200000H - 3FFFFFFH	Normal access to Bank2	2 MB
400000H - 5FFFFFFH	Page access to Bank1	2 MB
600000H - 7FFFFFFH	Page access to Bank2	2 MB

The page mode detection mechanism requires some overhead on the part of the software for the host system. Likely required will be custom routines to translate addresses to the page mode region for those accesses that can benefit from page mode acceleration. In addition, some steps may have to be taken (i.e. disabling interrupts temporarily) to ensure that accesses intended for another page are not inadvertently executed while a page mode access is in progress, as is commonly done in embedded systems. Even though there is some software overhead involved, significant performance enhancements can be achieved for certain types of operations. This type of page mode detection scheme has the benefit that a page mode access can be detected during the normal address decode process and it does not require lengthening the memory cycles of non-page mode accesses.

The page size for this DRAM is 1024 bits (1Kbyte). The bus connecting between the CPU and memory is 16 bits wide. Therefore the page size, as seen by software, for this memory controller is 2K bytes wide.

The MC68340 CPU has 32 address lines. The upper eight lines (A31-A24) can be configured as a parallel I/O port, as interrupt acknowledge lines, or as the upper byte of the address. This design assumes that the CPU was configured as either Port A[7:0] or IACK[7:0]. If the system requires larger memory map, the upper eight address lines should be decoded as well.

Transfer Modes

This design is implemented to access two separate memory banks. RAS1O is connected to the RAS signal of the bank1 DRAM, while RAS2O is connected to the bank2 RAS signal. Each DRAM bank has two 8-bit wide data ports, and the UCASO signal from the CPLD controls the upper side port, while the LCASO controls the lower port. The memory control interface for MC68340 supports the following transfer modes:

Even Byte-to-Word transfer	Valid data on the upper byte (D[15:8])
Odd Byte-to-Word transfer	Valid data on the lower byte (D[7:0])
Word-to-Word aligned transfer	Valid data on the both bytes (D[15:0])
Long Word-to-Word aligned transfer	Valid data on the both bytes (D[15:0])

When Even Byte-to-Word transfer occurs (A0 = 0), only the UCASO signal will be active to access the data on the upper 8-bit data bus. The lower byte on the data bus will be ignored. In case of Odd Byte-to-Word transfer mode (A0 = 1), the lower 8-bit data lines are used for the transfer by LCASO signal and the upper byte on the data bus is discarded. When the MC68340 performs Long Word-to-Word transfer, it holds the same address except A1 signal as current one and adjusts the data size from 32 bits to 16 bits on SIZ1 and SIZ0 on the next access, so that the 32 bit wide access can be completed. Table 2 shows the supported transfer modes and the relations among the signals.

Table 2. Supported Transfer Modes

Transfer Mode	SIZ1	SIZ0	A0	DSACK1	DSACK0	D[15:8]	D[7:0]
Even Byte-to-Word	0	1	0	0	X	Valid	Invalid
Odd Byte-to-Word	0	1	1	0	X	Invalid	Valid
Word-to-Word	1	0	0	0	X	Valid	Valid
Long Word-to-Word	0	0	0	0	X	Valid	Valid

Timing Diagrams

Figure 4. Random Mode Read and Write

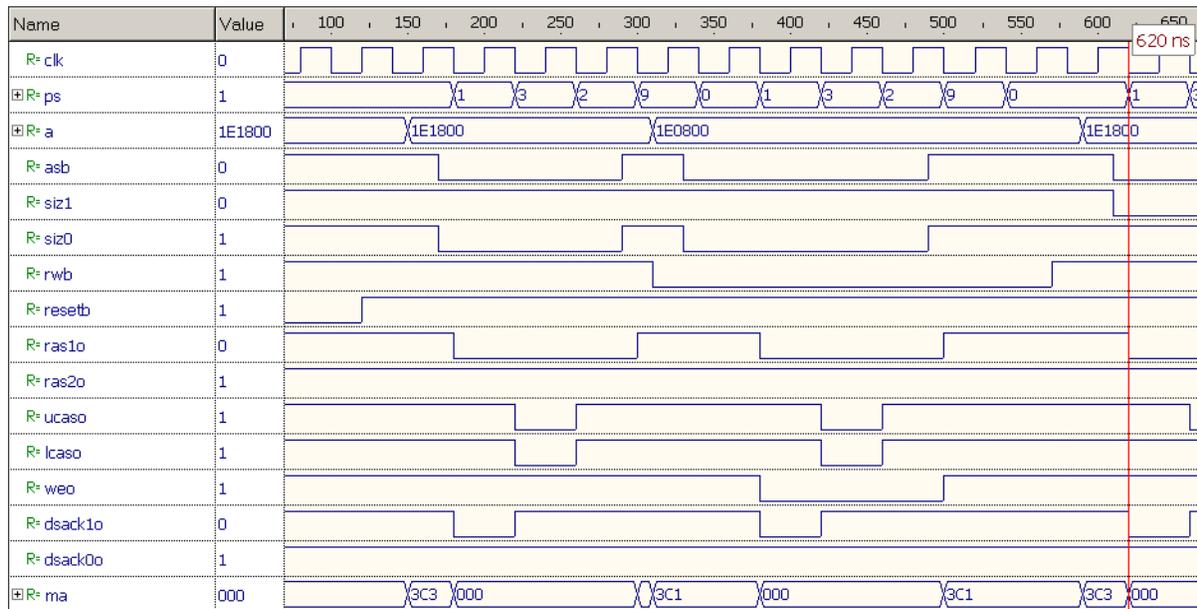


Figure 5. Page Mode Access

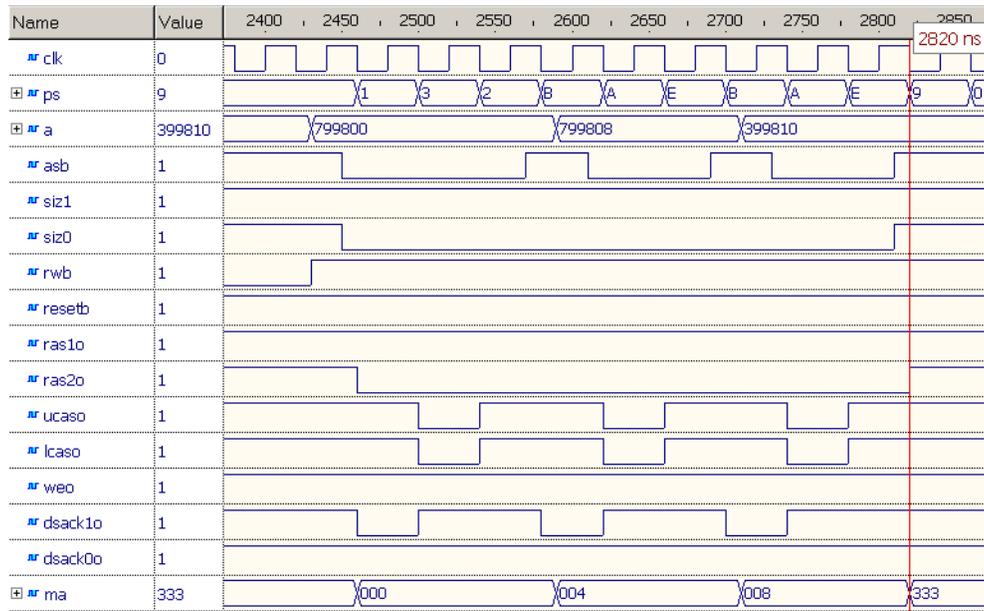
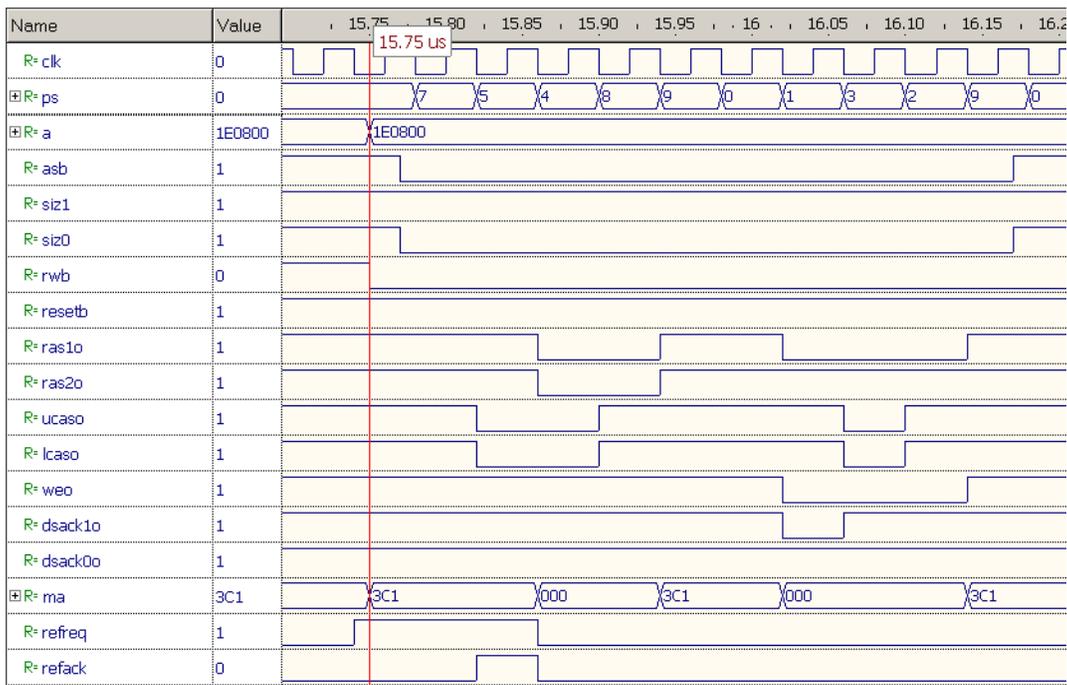


Figure 6. CAS Before RAS (CBR) Refresh



Implementation

This design is implemented in Verilog and VHDL. When using this design in a different device, density, speed, or grade, performance and utilization may vary. Default settings are used during the fitting of the design.

Table 3. Performance and Resource Utilization

Device Family	Language	Speed Grade	Utilization	f _{MAX} (MHz)	I/Os	Architecture Resources
MachXO2™ 1	Verilog	-6	56 LUTs	> 200	47	N/A
	VHDL	-6	55 LUTs	> 200	47	N/A
MachXO™ 2	Verilog	-5	55 LUTs	> 200	47	N/A
	VHDL	-5	55 LUTs	> 200	47	N/A
LatticeXP2™ 3	Verilog	-5	56 LUTs	> 200	47	N/A
	VHDL	-5	55 LUTs	> 200	47	N/A
ispMACH® 4000ZE ⁴	Verilog	-5 ns	42 Macrocells	> 165	47	N/A
	VHDL	-5 ns	43 Macrocells	> 165	47	N/A
ispMACH M4A3 ⁵	Verilog	-5.5 ns	42 Macrocells	> 90	47	N/A
	VHDL	-5.5 ns	43 Macrocells	> 90	47	N/A
ispLSI® 5000VE ⁶	Verilog	180 MHz	42 Macrocells	> 100	47	N/A
	VHDL	180 MHz	43 Macrocells	> 100	47	N/A

1. Performance and utilization characteristics are generated using LCMXO2-1200HC-6TG144CES, with Lattice ispLEVER® 8.1 SP1 and Lattice Diamond™ 1.1 software.
2. Performance and utilization characteristics are generated using LCMXO256E-5T100C, with Lattice ispLEVER 8.1 SP1 and Lattice Diamond 1.1 software.
3. Performance and utilization characteristics are generated using LFXP2-5E-5M132C, with Lattice ispLEVER 8.1 SP1 and Lattice Diamond 1.1 software.
4. Performance and utilization characteristics are generated using LC4256ZE-5TN100C with Lattice ispLEVER Classic 1.4 software.
5. Performance and utilization characteristics are generated using M4A3-128/64-55VC, with Lattice ispLEVER Classic 1.4 software.
6. Performance and utilization characteristics are generated using ispLSI5128VE-180LT128 with Lattice ispLEVER Classic 1.4 software

Source Code

fpdram.v/fpdram.vhd	Top-level source code
decoder.v/decoder.vhd	Address decoder
mux.v/mux.vhd	DRAM address multiplexer
sm.v/sm.vhd	State machine

Test Bench

fpdramtb.vhd VHDL test bench for the Page Mode DRAM Controller

Technical Support Assistance

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

Revision History

Date	Version	Change Summary
—	—	Previous Lattice releases.
February 2009	02.1	Updated Implementation section.
February 2010	02.2	Added support for LatticeXP2 device family. Added Verilog support for all families.
November 2010	02.3	Added support for the MachXO2 device family. Updated to support Lattice Diamond 1.1 design software. Updated to support ispLEVER 8.1 SP1 design software.