RAM-Type Interface for Embedded User Flash Memory

June 2012 | Reference Design RD1126

Introduction

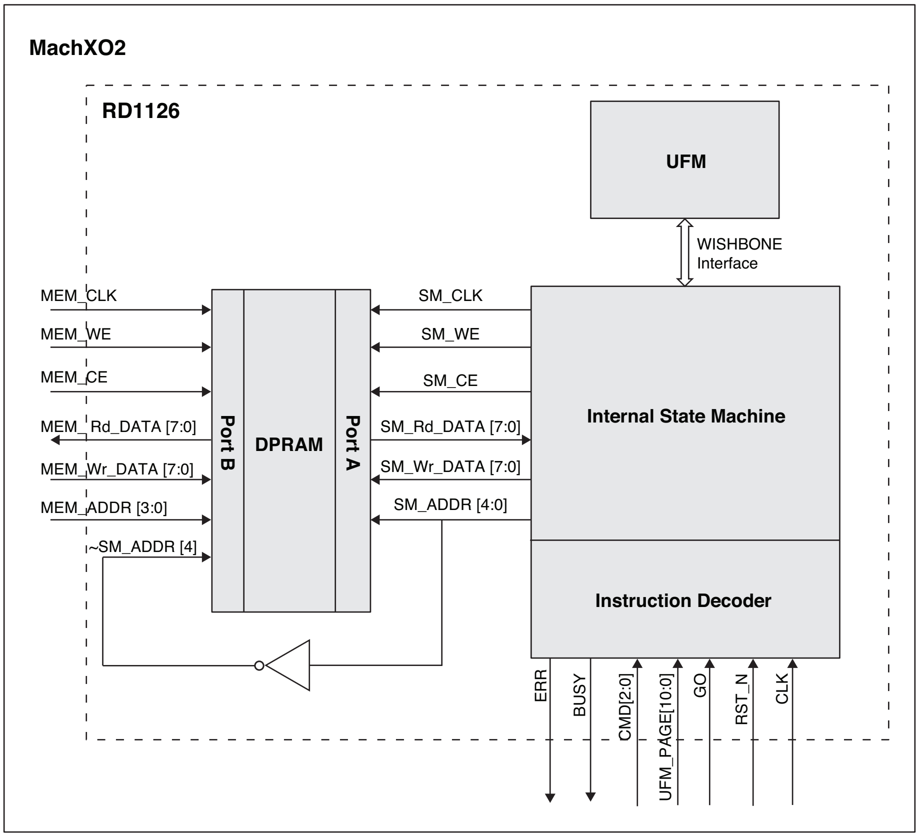
MachXO2™-640/U and higher density devices provide a User Flash Memory (UFM) block, which can be used for a variety of applications including PROM data storage, or general purpose non-volatile Flash memory. In some design contexts, the UFM may be used concurrently to store EFB initialization data, or in rare situations configuration data overflow, as specified by user settings.

The UFM is page addressable. Each page has 128 bits (16 bytes). The UFM block connects to the device core through the Embedded Function Block (EFB) WISHBONE interface. If desired, users can also access the UFM block independently through the hardened JTAG, I^{2}C and SPI interfaces of the device.

This reference design is intended to provide a familiar and intuitive extension to the MachXO2 Embedded Functional Block User Flash Memory (EFB UFM). This design facilitates users to access the MachXO2 EFB UFM module without the knowledge of WISHBONE protocol.

Architecture

Figure 1. MachXO2 UFM Simplified RAM-Type Interface Block Diagram



Port Descriptions

Table 1. Port Descriptions

Port	Direction	Width (Bits)	Description
CLK	Input	1	Clock input[1]
RST_N	Input	1	Active low reset signal
Command Interface			
GO	Input	1	High (‘1’) starts command process. GO is ignored if BUSY is asserted.
CMD	Input	3	Command operation
UFM_PAGE	Input	11	Target UFM Page
BUSY	Output	1	Status bit. Indicates operation in progress.
ERR	Output	1	Error bit. The last operation failed. Will be cleared on subsequent GO.
Data Interface			
MEM_CLK	Input	1	Clock for the DPRAM module
MEM_WE	Input	1	Write enable for the DPRAM module
MEM_CE	Input	1	Clock enable for the DPRAM module
MEM_ADDR	Input	4	DPRAM address
MEM_Wr_DATA	Input	8	DPRAM write data
MEM_Rd_DATA	Output	8	DPRAM read data

[1] Modify the design parameter READ_DELAY when CLK > 16.6 MHz. See discussion below.

Functional Description

This design facilitates user access to the MachXO2 EFB UFM module without the knowledge of WISHBONE protocol. The user has to provide the necessary command sequence with UFM page address. A GO signal will trigger the interface to perform the necessary WISHBONE transactions to read or write the UFM.

UFM

The User Flash Memory of the MachXO2 device features non-volatile storage in a single sector. Other features include:

- Non-volatile storage up to 256Kbits
- 100K write cycles
- Write access is performed page-wise; each page has 128 bits (16 bytes)
- Auto-increment addressing

Typical of Flash technology, the array must be cleared (erased) before data can be overwritten. The erased state of the UFM bits is ‘0’. The smallest erasable unit is the entire sector. Thus, any data which must be preserved (e.g. Config data overflow, or EBR init data) must be read out and stored in another available memory (e.g. EBR) prior to the Erase command. After the erase is complete, the data is written back to the UFM. The user is responsible for performing the storage operation. This requirement, along with limited erase cycles and long erase/program times, renders the UFM a poor solution for ‘scratch pad’ RAM (or similar) applications. MachXO2 Embedded Block RAM (EBR) is the recommended on-board memory type for high-churn, volatile data storage.

In the design’s default configuration, the internal UFM read operation has been optimized for lower CLK rates. For CLK rates exceeding 16.6 MHz, this design must be configured to insert additional Retrieval Delay into the state machine which generates the UFM read command. A parameter, READ_DELAY, is provided for this purpose. The source file “UFM_WB_top.v” can be modified directly or the parameter passed in the module instantiation. The minimum value for READ_DELAY can be calculated as follows:

READ_DELAY(min) = 240/PERIOD - 4

where PERIOD = CLK period in ns

Example, for CLK = 47MHz, PERIOD = 21.28ns thus READ_DELAY >= 8 (7.28 rounded up)

Refer to the “Reading Flash Pages” section of TN1204, MachXO2 Programming and Configuration Usage Guide for more information on the Read UFM command structure and timing requirements.

DPRAM

The design incorporates a 32-byte (2 UFM page), byte addressable True Dual port RAM. The internal state machine accesses one port of the DPRAM. During the UFM write operation the state machine reads data from the DPRAM and writes it to the UFM. During UFM read operations the data read from the UFM is stored in the DPRAM.

The second DPRAM port is accessed by the user to exchange data to or from the UFM. The DPRAM memory is divided into two pages of 16 bytes data each. The Most Significant Bit (MSB) of the DPRAM addresses are controlled by the internal state machine for page swapping. Thus, only one page (16 bytes) of the DPRAM is available through the user port at any given time. The available page is synchronized by the reference design and optimized for maximum throughput.

During a UFM write operation, the DPRAM page swap happens once a GO is recognized. For higher throughput in back-to-back writes, this allows the user to immediately start loading the next page to be transferred to the UFM. During a read operation, the DPRAM page swap happens at the end of a read UFM operation and the user can start reading the DPRAM data once BUSY signal is low. In the case of back-to-back Read operations, the data remains available in the DPRAM until the subsequent Read is complete, allowing for higher throughput operation.

Commands

Table 2. List of Commands

Command	Operation	BUSY Signal	Page Swap
000	Read 1 page	Clear when finished	End of transaction
001	Read next page	Clear when finished	End of transaction
010	Write 1 page	Clear upon UFM busy clear.	Beginning of transaction
011	Write next page	Clear upon UFM busy clear	Beginning of transaction
100	Enable UFM access	Clear upon UFM busy clear.	N/A
101	Disable UFM access	Clear when finished	N/A
110	(undefined)	–	–
111	Erase UFM	Clear upon UFM busy clear	N/A

Command Descriptions

- **Enable UFM access (100)** – Required to enable UFM read/write access. Exercising this command will temporarily disable certain features of the device, notably GSR, User SPI port and Power Controller. These features are restored when UFM access is disabled using Disable UFM access command (101). See TN1246, MachXO2 EFB User’s Guide, for more information on this behavior. BUSY is asserted until the devices internal Flash pumps are fully charged.
- **Write 1 page (010)** – The 16 bytes of data from DPRAM is written into the UFM page specified by UFM_PAGE. The internal UFM page pointer is auto incremented at the completion of the command. Make sure that 16 bytes of data that has to be written to the UFM is loaded to the DPRAM prior to issuing a GO. BUSY will be asserted as soon as a GO is recognized and will be de-asserted when programming is complete.
- **Write next page (011)** – The 16 bytes of data from DPRAM are written into the subsequent UFM page pointed by the internal UFM page pointer. The internal UFM page pointer is auto-incremented at the completion of the

command. Make sure that 16 bytes of data that has to be written to the UFM is loaded to the DPRAM prior to issuing a GO. BUSY will be asserted as soon as a GO is recognized and will be de-asserted when programming is complete.

- **Read 1 page (000)** – 16 bytes of data is read back from the UFM page specified by UFM_PAGE and stored in the DPRAM. The internal UFM page pointer is auto-incremented at the completion of the command. Following a Read command, the 16 bytes of UFM data can be read from the DPRAM after BUSY de-asserts.
- **Read next page (011)** – The subsequent UFM page pointed by the internal UFM page pointer is read and stored into the DPRAM. The internal UFM page pointer is auto-incremented at the completion of the command. Following a Read command, the 16 bytes of UFM data can be read from the DPRAM after BUSY de-asserts.
- **Disable UFM access (101)** – This command disables UFM interface for change access.
- **Erase UFM (111)** – This command is issued to completely erase (set to '0') the UFM. BUSY is asserted until erasure is complete.

Typical Command Sequences

Notes:

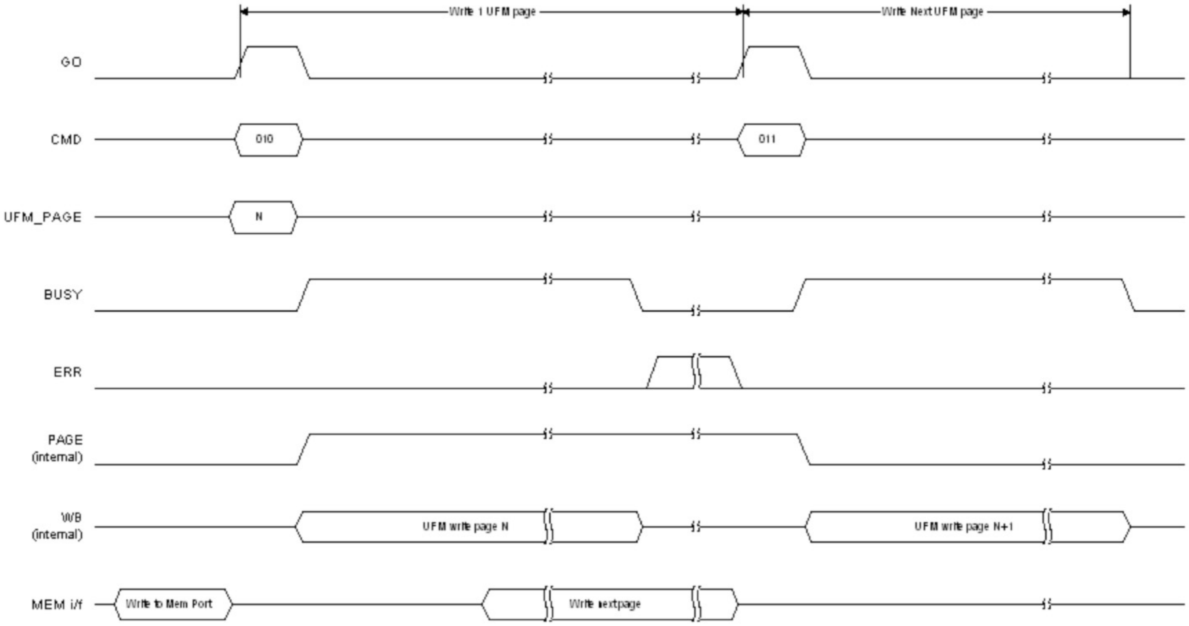
- *A GO will be ignored when BUSY is asserted.*
- *Any UFM read/write or erase command while UFM is disabled will assert the ERR signal.*

Write to UFM

The following sequence explains the necessary commands to be followed for a UFM write operation. The sequence assumes the target UFM row is in the erased state.

1. **Enable UFM access (100)** – Check for BUSY signal to be low and issue enable UFM command on the CMD bus and issue a GO. BUSY signal is asserted once a GO is recognized.
2. **Write 1 page (010)** – Load the DPRAM with one page (16 bytes) of data to be written to the UFM. Check for BUSY signal to be low, issue the Write 1 Page command on the CMD bus, UFM page address on the UFM_PAGE bus and issue a GO. The DPRAM page swap happens once a GO is recognized so that the user can load the next page to be written to the UFM into the DPRAM while the first page is being transferred to the UFM.
3. **Write next page (011)** – Wait for the previous page to be transferred to the UFM successfully and the BUSY signal goes low. Now issue a write next page command and issue GO. The internal UFM page pointer is auto-incremented.
4. **Repeat steps 2 and 3 until all desired pages are written**. Figure 2 illustrates steps 2 and 3.
5. **Disable UFM access (101)** – Check for BUSY signal to be low and issue disable UFM command on the CMD bus and issue a GO. BUSY signal is asserted once a GO is recognized.

Figure 2. Write to UFM Waveform Description



Read from UFM

The following sequence explains the necessary commands to be followed for a UFM read operation.

1. **Enable UFM access (100)** – Check for the BUSY signal to be low and issue an enable UFM command on the CMD bus and issue a GO. BUSY signal is asserted once a GO is recognized.

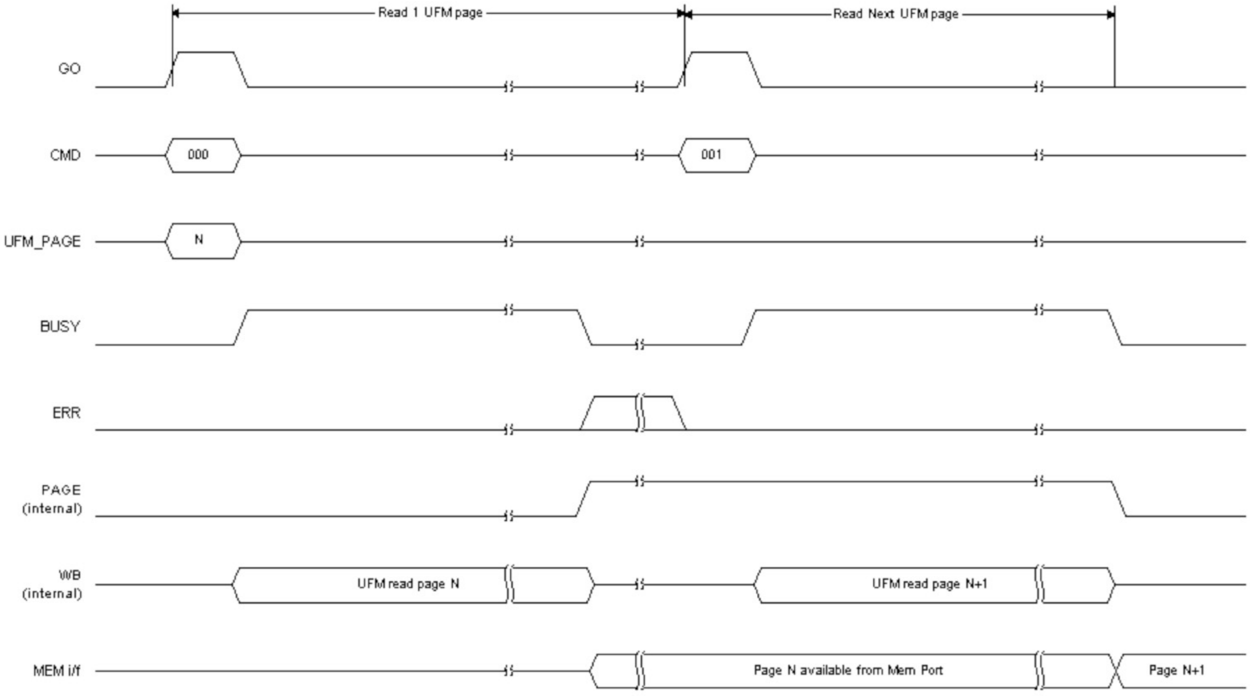
2. **Read 1 page (000)** – Check for BUSY signal to be low, issue the Read 1 Page command on the CMD bus, UFM page address on the UFM_PAGE bus and issue a GO. Data read from UFM will be loaded in the DPRAM and user can access the data once BUSY signal goes low.

3. **Read next page (011)** – Check for the BUSY signal to be low, issue the read next page command on the CMD bus and issue a GO. The internal UFM page pointer is auto. Data read from UFM will be loaded in the DPRAM and user can access the data once the BUSY signal goes low.

4. **Repeat steps 2 and 3 until all desired pages have been read**. Figure 3 illustrates steps 2 and 3.

5. **Disable UFM access (101)** – Check for the BUSY signal to be low and issue a disable UFM command on the CMD bus and issue a GO. The BUSY signal is asserted once a GO is recognized.

Figure 3. Read from UFM Waveform Description



Read Modify Write Back to UFM

The page read from UFM can be retained in the DPRAM and data modification may be performed directly on the DPRAM, followed by a Write 1 Page operation. The following sequence explains the necessary commands to be followed in order to modify the contents of UFM.

Note: The modified data cannot be arbitrary. Only erased bits/bytes (='0') of the UFM can be given new values. This sequence is provided to support applications where small amounts of data, or data unaligned to UFM page boundaries, is accumulated over time (e.g. a data logger application).

1. **Enable UFM access (100)** – Check for BUSY signal to be low and issue an enable UFM command on the CMD bus and issue a GO. BUSY signal is asserted once a GO is recognized.

2. **Read 1 page (000)** – Check for BUSY signal to be low, issue the Read 1 Page command on the CMD bus, UFM page address on the UFM_PAGE bus and issue a GO. Data read from the UFM will be loaded in the DPRAM and is available to read or edit once the BUSY signal goes low.

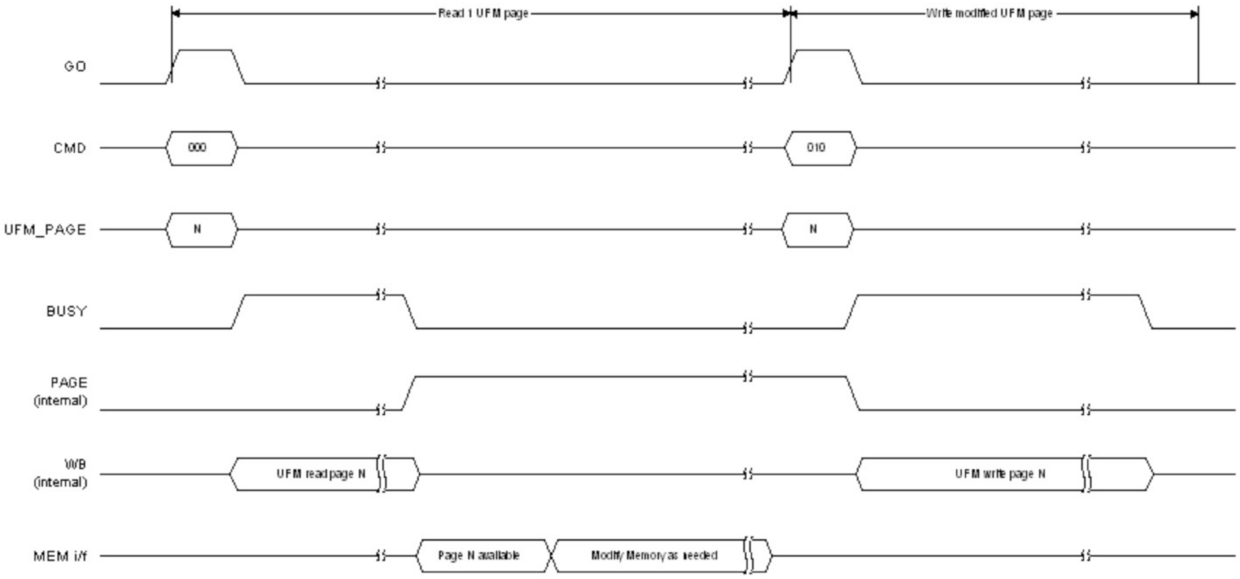
3. **Modify DPRAM data as needed**. The DPRAM data is byte addressable. The user may modify the intended byte(s) directly in the DPRAM. Alternatively, the data may be read from, and re-written to, the DPRAM.

4. **Write 1 page (010)** – Once the data in the DPRAM has been modified or re-written, check for the BUSY signal to be low, issue the Write 1 Page command on the CMD bus, set the UFM page address on the UFM_PAGE bus and issue a GO. Note that UFM page address must be specified for both read and write.

5. **Repeat steps 2 through 4 until all desired pages have been modified**. Figure 4 illustrates steps 2 through 4.

6. **Disable UFM access (101)** – Check for the BUSY signal to be low and issue a disable UFM command on the CMD bus and issue a GO. BUSY signal is asserted once a GO is recognized.

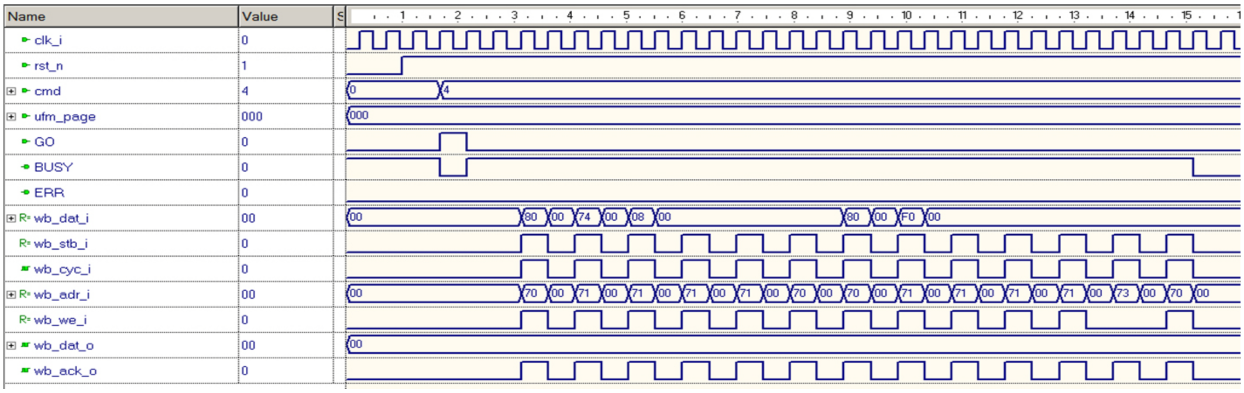
Figure 4. Read-Modify and Write Back to UFM Waveform Description



HDL Simulation and Verification

Enable Command

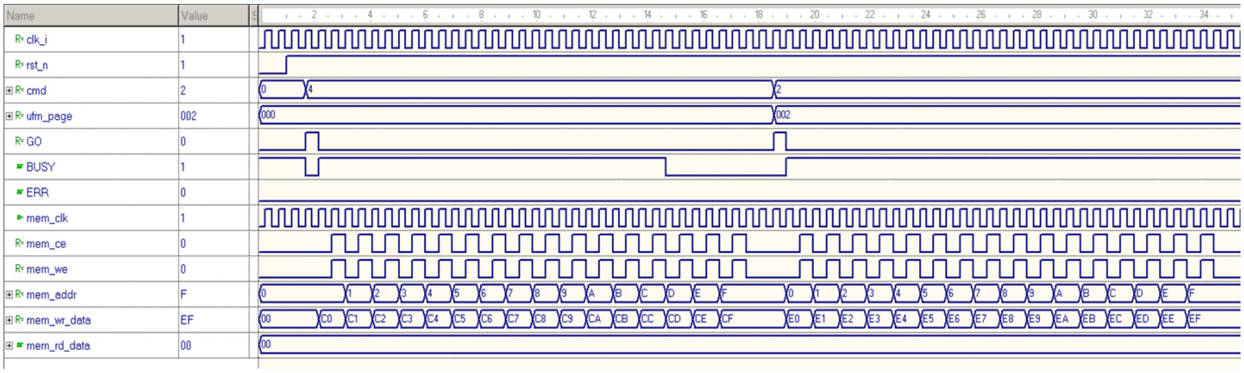
Figure 5. Enable Command HDL Simulation Waveform



Name	Value
clk_i	0
rst_n	1
cmd	4
ufm_page	000
GO	0
BUSY	0
ERR	0
wb_dat_i	00
wb_stb_i	0
wb_cyc_i	0
wb_adr_i	00
wb_we_i	0
wb_dat_o	00
wb_ack_o	0

Write Command

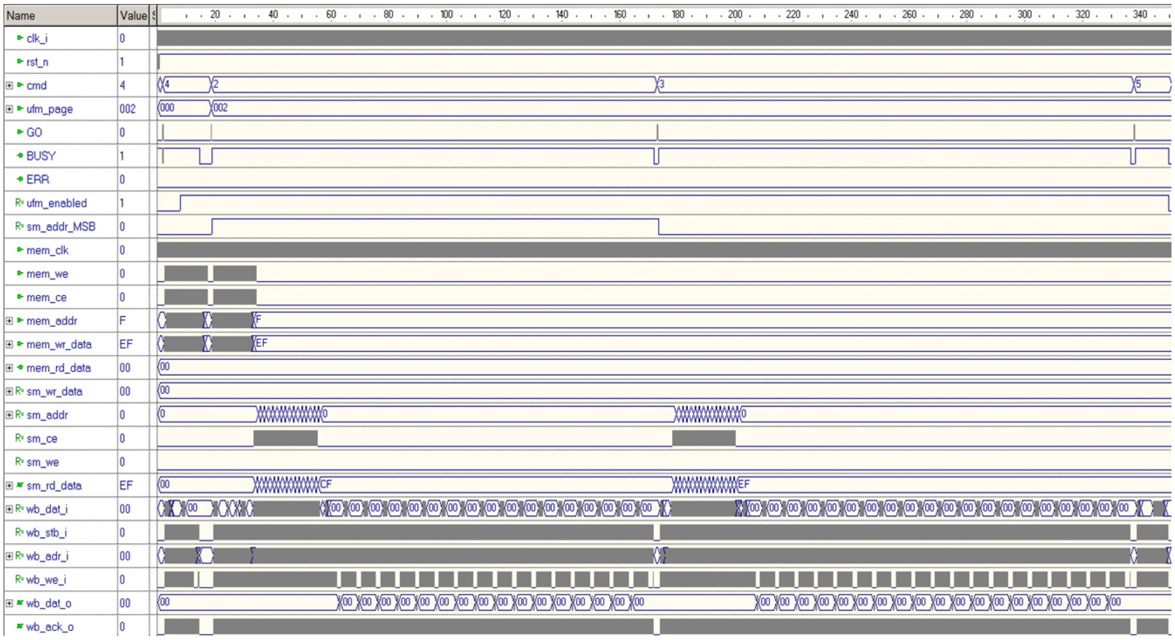
Figure 6. Loading of DPRAM HDL Simulation Waveform



Name	Value
clk_i	1
rst_n	1
cmd	2
ufm_page	002
GO	0
BUSY	1
ERR	0
mem_clk	1
mem_ce	0
mem_we	0
mem_addr	F
mem_wr_data	EF
mem_rd_data	00

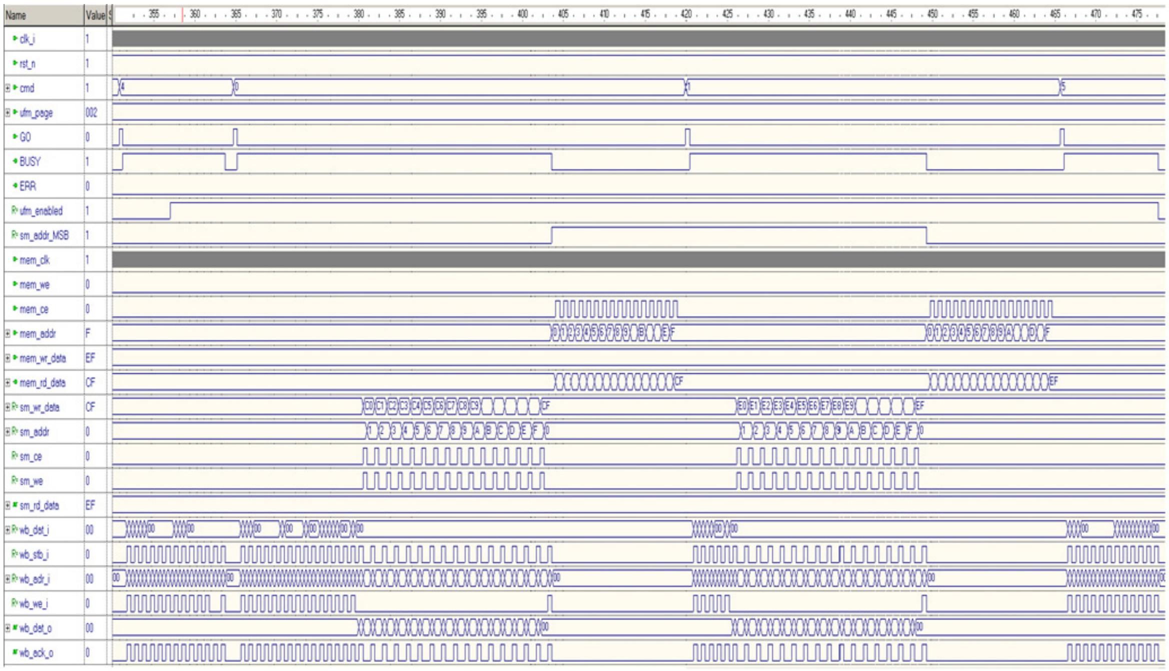
2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34
0 4 2
000 002
0 1 2 3 4 5 6 7 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F
00 C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 EA EB EC ED EE EF
00

Figure 7. Write to UFM Command Sequence HDL Simulation Waveform



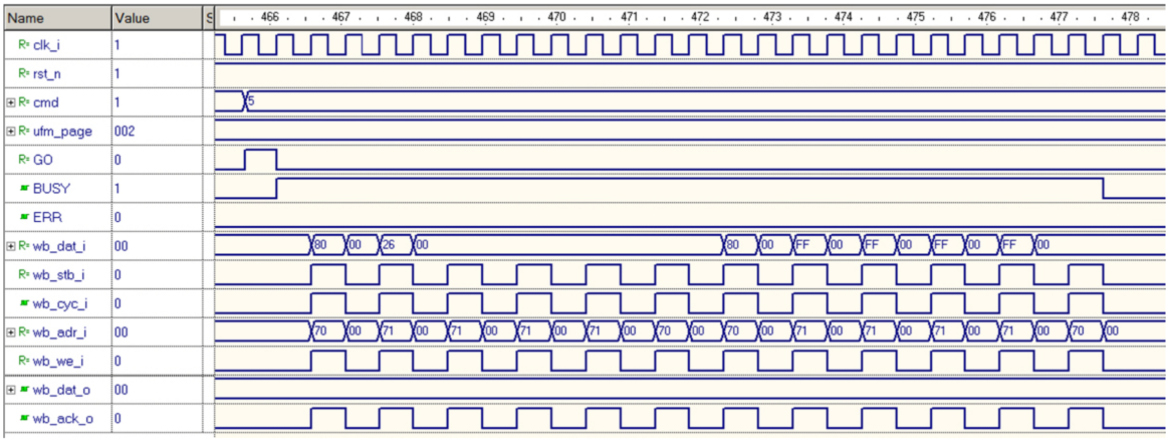
Read Command

Figure 8. Read from UFM Command Sequence HDL Simulation Waveform



Disable Command

Figure 9. Disable Command HDL Simulation Waveform



Implementation

Table 3. Performance and Resource Utilization

Family	Language	Speed Grade	Utilization	fmax (MHZ)	I/Os	Architecture Resources
MachXO2[1]	Verilog	-3	239 LUTs	>50	42	1- EFB1- EBR

[1] Performance and utilization characteristics are generated using LCMXO2-1200ZE-3TG100C with Lattice Diamond® 1.4 design software.

Technical Support Assistance

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

Revision History

Date	Version	Change Summary
April 2012	01.0	Initial release.
June 2012	01.1	Updated design and document to include READ_DELAY parameterization.

Appendix A. UFM Command Sequences

The following table explains the different UFM internal command structure involved for the various operations of this reference design.

Table 4. MachXO2 Embedded Function Block Commands

Operation	Command (Hex)	Operand (Hex)	Write Data	Read Data
Enable UFM Access (100)				
Enable UFM	74	08 00 00	—	—
Status check	F0	00 00 00	—	1 byte status data
Write One Page (010)				
Set Address	B4	00 00 00	40 00 aa aa[1]	—
Write UFM	C9	00 00 01	16 bytes data from DPRAM	—
Status check	F0	00 00 00	—	1 byte status data
Write Next Page (011)				
Write UFM	C9	00 00 01	16 bytes data from DPRAM	—
Status check	F0	00 00 00	—	1 byte status data
Read One Page (000)				
Set Address	B4	00 00 00	40 00 aa aa[1]	—
Read UFM	CA	10 00 01	—	16 bytes data from UFM
Read Next Page (011)				
Read UFM	CA	10 00 01	—	16 bytes data from UFM
Disable UFM Access (101)				
Disable UFM	26	00 00 00	—	—
Bypass	FF	FF FF FF	—	—
Erase UFM (111)				
Erase UFM	0E	08 00 00	—	—
Status check	F0	00 00 00	—	1 byte status data

1. The aa aa value is copied from the 11-bit UFM_PAGE module input.