XAPP923 (v1.2) June 5, 2007

Reference System: MCH OPB EMC with OPB Central DMA

Author: Sundararajan Ananthakrishnan

Summary

This application note demonstrates the use of the Multi-CHannel (MCH) On Chip Peripheral Bus (OPB) External Memory Controller (EMC) in a MicroBlaze™ processor system. The MCH ports of the MCH OPB EMC connect to the cache ports of the MicroBlaze processor to allow efficient cacheline accesses by the processor. The OPB Central DMA controller is also included in this system to illustrate the capability of the MCH OPB EMC to handle cacheline access from MicroBlaze processor and OPB sequential address (burst) transactions on the OPB from the OPB Central DMA controller simultaneously.

The reference design uses Static RAM (SRAM) as external memory. The SRAM memory space is split into a cacheable block of memory and a non-cacheable block of memory. The standalone software application provided with this reference system is executed from the cacheable block of memory and the DMA transfers occur inside the non-cacheable block of memory.

This application note describes how to set up the parameters for the MicroBlaze processor, the MCH OPB EMC and the OPB Central DMA controller. This reference system is targeted for the Memec Spartan™-3 3S1500 board with the P160 communications Module Rev 2 board.

Included Systems

Included with this application note is the reference system for the Memec Spartan-3 3S1500 board with the P160 communications Module Rev 2 board.

- www.xilinx.com/bvdocs/appnotes/xapp923.zip

Introduction

Many software applications using FPGAs are executed from the main memory. Running applications from the main memory is slow compared to applications run from the on-chip memory based on physical limitations of the main memory. The advantage of main memory is its size. For example, operating systems heavily use instructions and data located in main memory. By using caching, some common instructions and data can be located inside the cache instead of the main memory. Caching instructions and data are stored in high-speed memory on the FPGA which gives faster access time. The MCH OPB EMC, MCH OPB SDRAM, and MCH OPB DDR provide efficient cacheline accesses to external memory for the MicroBlaze processor. The memory controller reference system described in this application note uses the MCH OPB EMC.

Hardware and Software Requirements

The hardware and software requirements are:

- Memec Spartan-3 3S1500 board with the P160 communications Module Rev 2 board
- Xilinx Platform USB cable or Parallel IV programming cable
- RS232 serial cable and serial communication utility (HyperTerminal)
- Xilinx Platform Studio 9.1.01i
- Xilinx Integrated Software Environment (ISE™) 9.1.03i

Reference System Specifics

This reference system is built using the Memec Spartan-3 3S1500 board with the P160 communications Module Rev 2 board. The system uses the MicroBlaze processor with 8 KB for both the instruction cache (I-cache), and the data cache (D-cache). The OPB Central DMA controller in the reference system generates the burst transactions to the MCH OPB EMC. The OPB UARTLite with interrupts and the OPB GPIO (to control four LEDs) are used in the reference system.

Refer to Figure 1 for the block diagram and Table 1 for the address map of the system.

Block Diagram

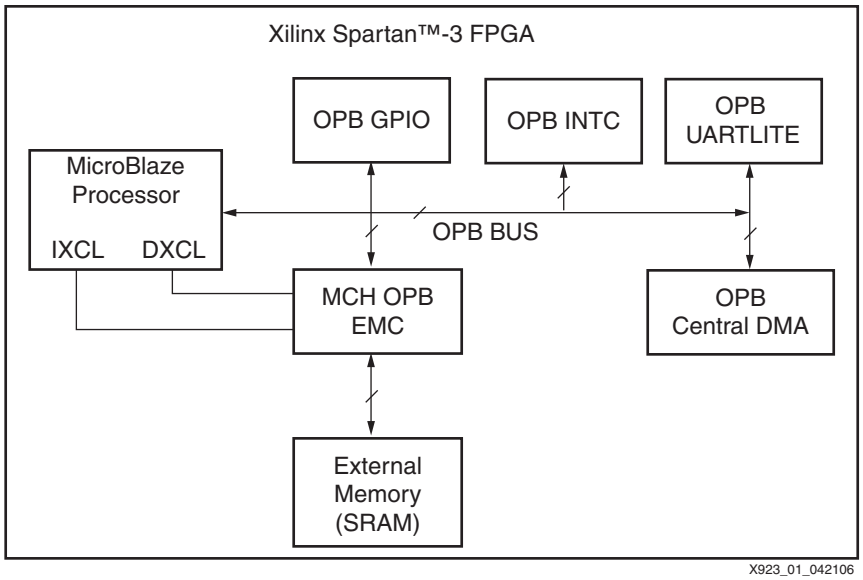

Figure 1: **Reference System Block Diagram**

Address Map

The address mapping for the IP cores in the reference system is given in Table 1.

Table 1: **Reference System Address Map**

Table 2:

Peripheral	Instance	Base Address	High Address
opb_mdm	debug_module	0x4140000	0x4140FFFF
lmb_v10	dlmb_cntlr	0x00000000	0x00003FFF
lmb_v10	ilmb_cntlr	0x00000000	0x00003FFF
opb_uart16550	RS232	0x40400000	0x4040FFFF
opb_gpio	LEDs_8Bit	0x40000000	0x4000FFFF
mch_opb_emc	SRAM_256Kx32	0x30000000	0x300FFFFF
opb_intc	opb_intc_0	0x41200000	0x4120FFFF
opb_central_dma	opb_central_dma_0	0x41E00000	0x41E0FFFF

System Configuration

This system requires that the MicroBlaze processor be configured to support Xilinx CacheLink (XCL) caching. The MCH OPB EMC is also configured to support the MicroBlaze processor MCH connections and OPB bursting. This illustrates the capacity of the MCH OPB EMC to handle OPB bursts and cacheline transactions concurrently. The OPB Central DMA controller is configured to allow access to its registers.

The following sections explain the configuration of the MicroBlaze processor, the MCH OPB EMC, and the OPB Central DMA controller.

Setting the MicroBlaze processor parameters

Under the Parameters Tab of the MicroBlaze processor, change the parameters as shown in Figure 2. These parameters set the cacheable block of main memory between 0x30000000 and `0x3000FFFF`. Both the instruction cache and the data cache are enabled. The size of the instruction cache and data cache are set to 8K respectively. The XCL interface is enabled for both the instruction cache and the data cache. This allows the MicroBlaze processor to use the dedicated XCL interface instead of the shared OPB bus for cacheline transfers.

Note: The I-cache writes have been enabled. This enables the WIC (Write Instruction Cache) instruction to write into the I-cache.

See the *MicroBlaze Processor Reference Guide* under the sections Instruction Cache and Data Cache for more details on the MicroBlaze processor caches.

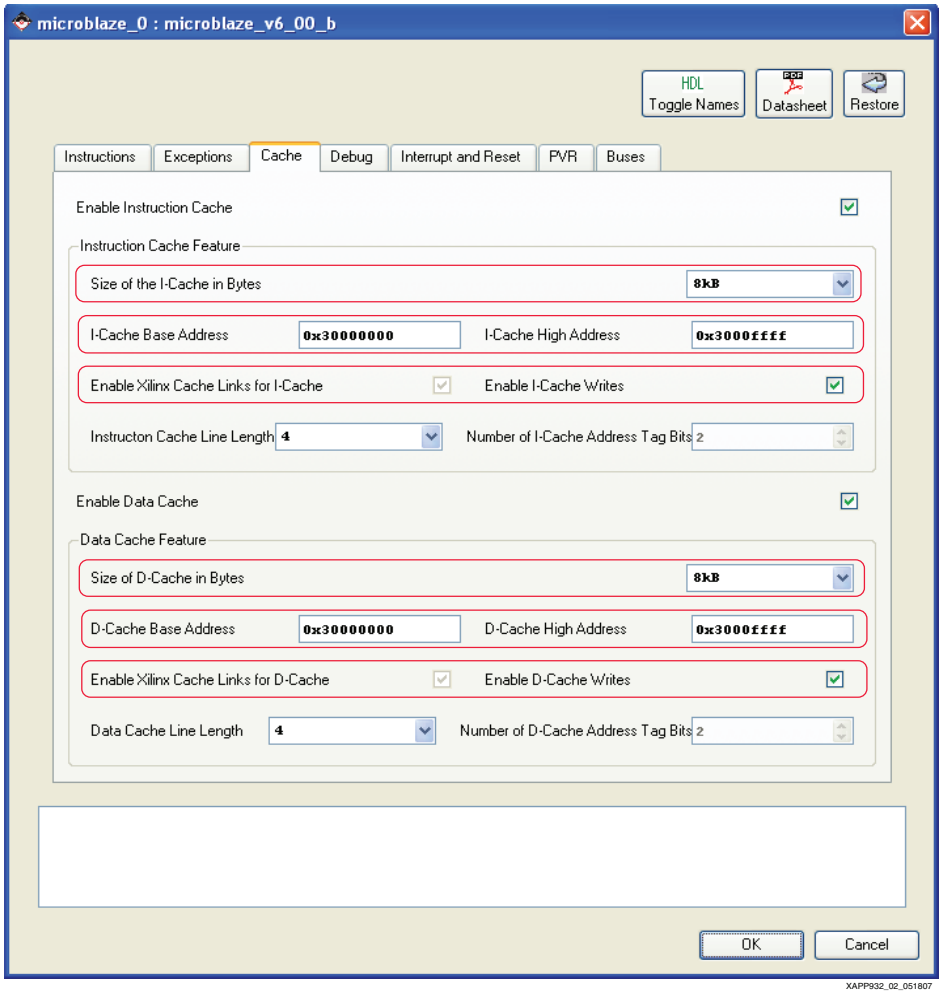

Figure 2: **Setting MicroBlaze Cache Parameters**

Note: Actual values of base address, high address, and cache size can vary from system to system and values must be adjusted based on the main memory base address.

Setting the MCH OPB EMC parameters

The MCH OPB EMC needs to be configured to support 2 MCH channels which connect to the MicroBlaze processor instruction and data caches. The XCL properties of these channels also need to be set appropriately. In addition, the MCH OPB EMC needs to be configured to support OPB bursting to handle the transactions from the OPB Central DMA controller effectively.

The MCH OPB EMC is configured for 2 channels by configuring the “Number of MCH Channels” (C_NUM_CHANNELS) to 2. The MCH OPB EMC is configured to support burst transactions by setting the parameters “Include OPB Slave Interface” (C_INCLUDE_OPB_IPIF) to 1 and “Support OPB Burst” (C_INCLUDE_OPB_BURST_SUPPORT) to 1 in the All tab under the MCH OPB EMC core as shown in Figure 3.

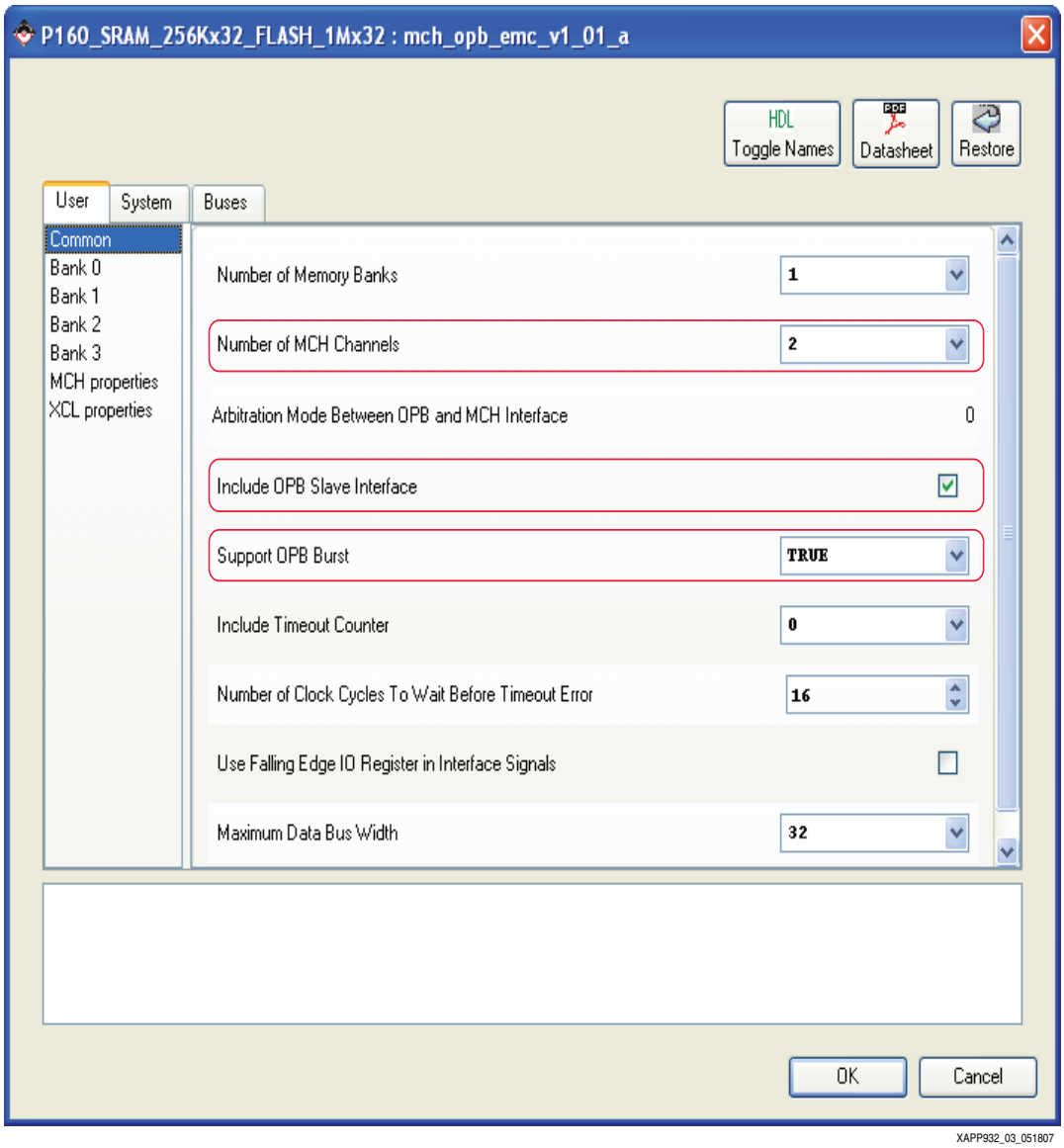

Figure 3: **Setting MCH OPB EMC Parameters**

Setting Memory Bank Parameters for the MCH OPB EMC

The MCH OPB EMC currently supports only one bank of memory. The width of memory bank can be 8, 16, or 32. The default width of the memory bank supported is 32. The memory bank can be asynchronous or synchronous. The asynchronous memory type is supported by configuring the parameter “Bank 0 is synchronous” (C_SYNC_MEM_0) to 0 (Asynchronous).

The memory bank timing parameters value must be set if the memory type in the bank is asynchronous. The timing values can be set by referring to the datasheet of the memory device. These values can be set in the Bank 0 section under the User tab in the MCH OPB EMC core.

The parameter settings for memory bank 0 are shown in Figure 4.

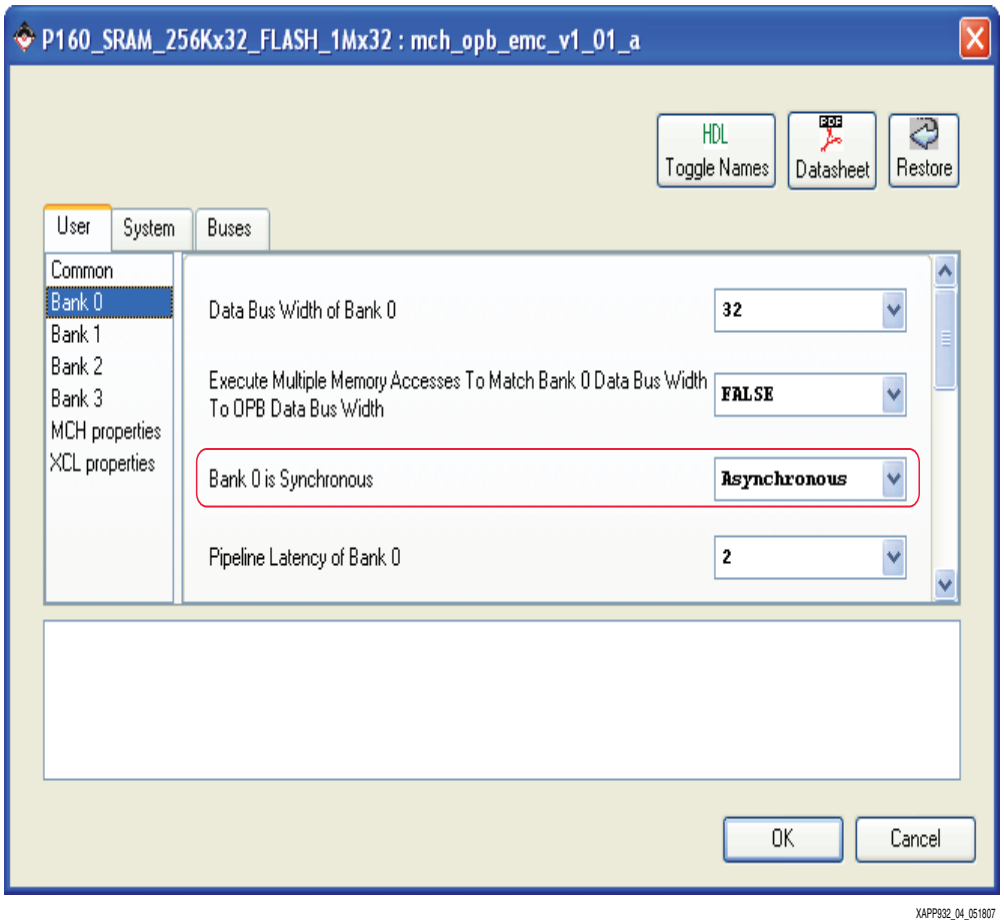

Figure 4: **Setting Memory Bank Parameters for MCH OPB EMC**

Setting MCH Properties for the MCH OPB EMC

The MCH interface properties need to be set for the MCH OPB EMC. The MCH OPB EMC currently supports only the XCL protocol, therefore the parameter “Interface Protocol of channels” (C_MCH0_PROTOCOL) is set to 0 which indicates the XCL protocol. The “Depth of Access Buffer” (C_MCHx_ACCESSBUF_DEPTH) parameter is set to the default value 16 for all the channels. The parameter “Depth of Read Data Buffer" (C_MCHx_RDDATABUF_DEPTH) for the channels that connect the I-cache and D-cache is set to 0, because the MicroBlaze processor can consume the data as soon as its available. Setting

this parameter eliminates the read data buffer and the latency that normally exists while reading the data from this buffer.

These parameters are set in the MCH properties section in the User tab of the MCH OPB EMC core as shown in Figure 5.

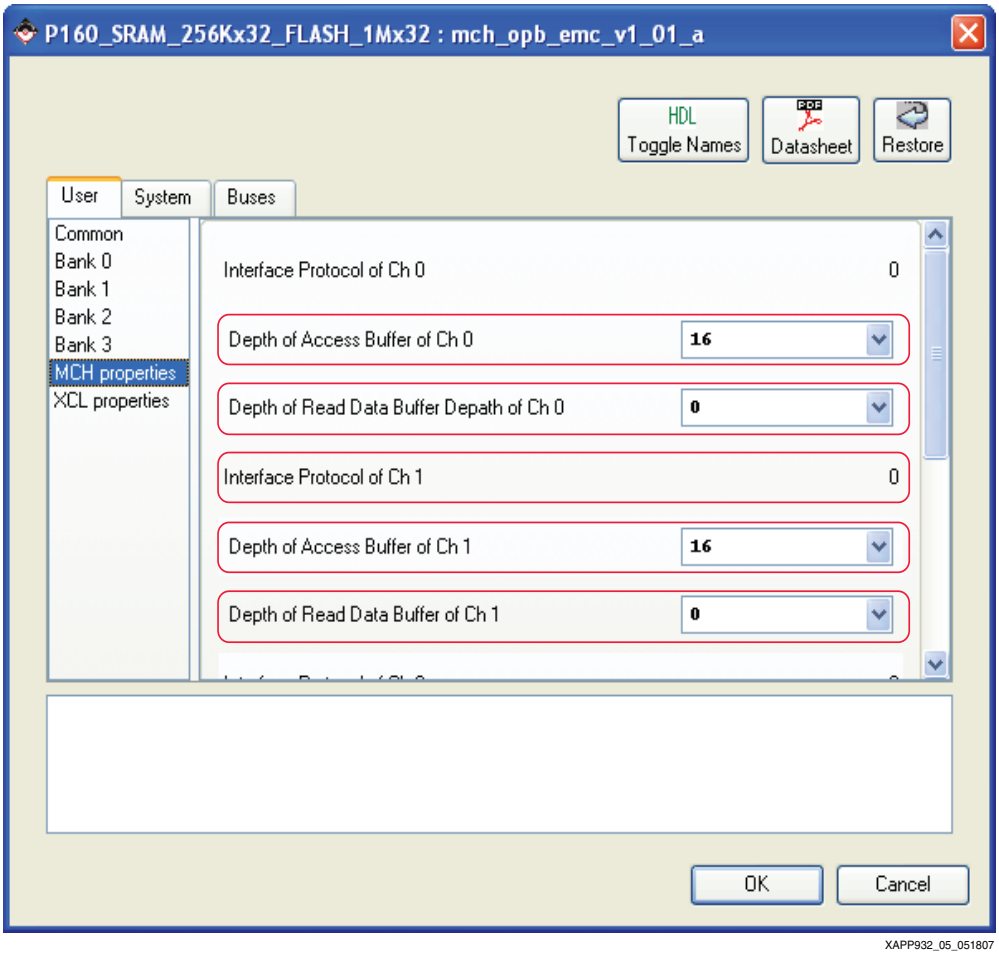

Figure 5: **Setting MCH Properties for MCH OPB EMC**

Setting XCL Properties for the MCH OPB EMC

The size of the cacheline, in number of 32-bit words, is set for all the channels that are configured as XCL channels. The MCH Channel 0 is connected to the I-cache of the MicroBlaze processor. The MCH Channel 1 is connected to the D-cache of the MicroBlaze processor. Since the cacheline size of the I-cache and D-cache for MicroBlaze processor is four words, the cachelines sizes for channels 0 and 1 (C_XCLx_LINESIZE) are set to four words.

The I-cache of MicroBlaze processor will only do read accesses to memory, therefore the parameter "Type of Write Transfers for MCH channel 0" (C_XCL0_WRITEXFER) is set to 0. This indicates that this channel will not perform any memory write transfers and reduces the logic implemented for this channel. In Figure 2, I-cache writes are enabled that enables the WIC instruction to write into the I-cache. Enabling the WIC instruction does not effect the setting of the "Type of Write Transfer" (C_XCL0_WRITEXFER) parameter of MCH channel 0.

The D-cache of the MicroBlaze processor will perform only single beat writes to memory, therefore the "Type of Write Transfers for MCH channel 1" (C_XCL1_WRITEXFER) is set to 1. This indicates that only single-beat writes are performed on this channel.

These parameters are set in the XCL properties section under the User tab of the MCH OPB EMC core as shown in Figure 6.

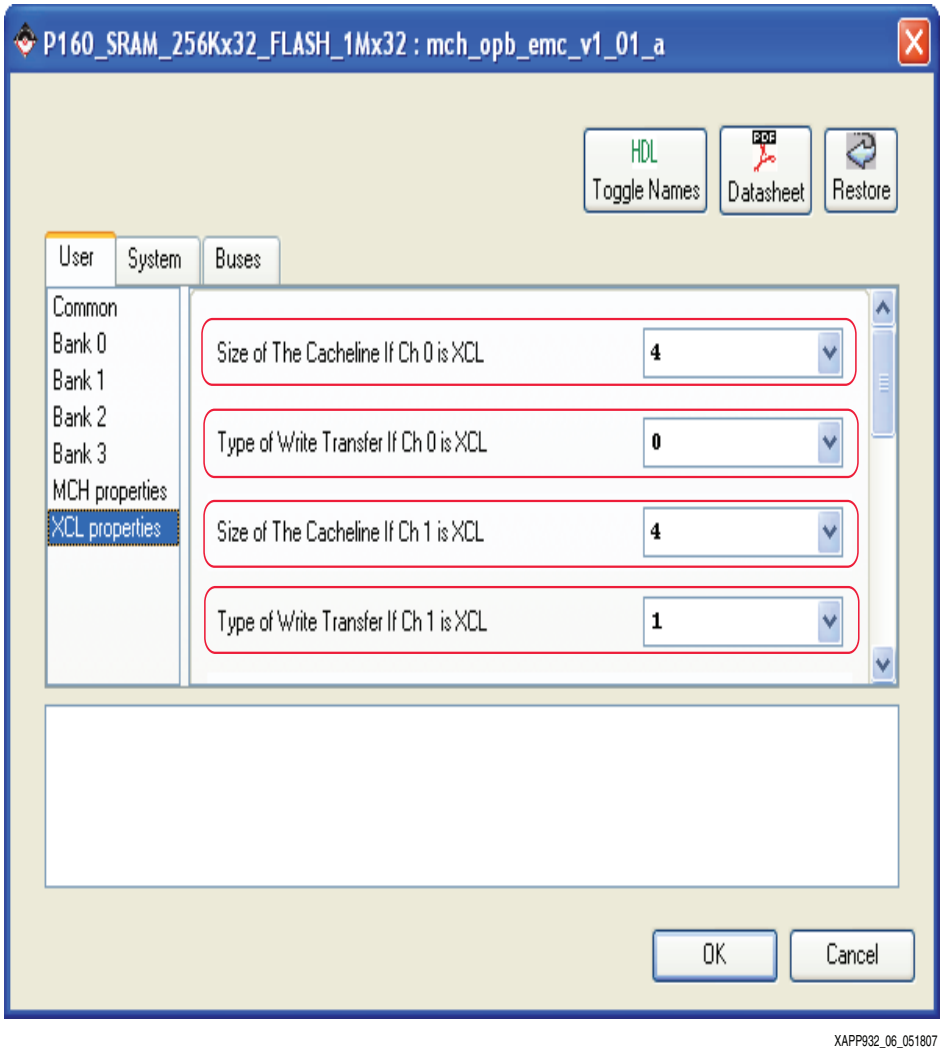

Figure 6: **Setting XCL Properties for MCH OPB EMC**

Connecting the MCH OPB EMC to the MicroBlaze Processor

To allow the MicroBlaze processor to cache over XCL, connections must be made between the MicroBlaze processor and the MCH OPB EMC. These connections can be viewed in the `system.mhs` file.

The MicroBlaze processor uses the following interface connections:

- BUS_INTERFACE IXCL = ixcl
- BUS_INTERFACE DXCL = dxcl

A portion of the MHS file showing the bus interface connections of the MicroBlaze processor is shown in Figure 7.

```
BEGIN microblaze
 PARAMETER INSTANCE = microblaze_0
 PARAMETER HW_VER = 6.00.b
 PARAMETER C_DEBUG_ENABLED = 1
 PARAMETER C_AREA_OPTIMIZED = 1
 PARAMETER C_NUMBER_OF_PC_BRK = 2
 PARAMETER C_NUMBER_OF_RD_ADDR_BRK = 1
 PARAMETER C_NUMBER_OF_WR_ADDR_BRK = 1
 PARAMETER C_USE_ICACHE = 1
 PARAMETER C_CACHE_BYTE_SIZE = 8192
 PARAMETER C_USE_DCACHE = 1
 PARAMETER C_DCACHE_BYTE_SIZE = 8192
 PARAMETER C_ICACHE_USE_FSL = 1
 PARAMETER C_DCACHE_USE_FSL = 1
 PARAMETER C_ICACHE_BASEADDR = 0x30000000
 PARAMETER C_ICACHE_HIGHADDR = 0x3000ffff
 PARAMETER C_ADDR_TAG_BITS = 2
 PARAMETER C_DCACHE_BASEADDR = 0x30000000
 PARAMETER C_DCACHE_HIGHADDR = 0x3000ffff
 PARAMETER C_DCACHE_ADDR_TAG = 2
 BUS_INTERFACE DLMB = dlmb
 BUS_INTERFACE ILMB = ilmb
 BUS_INTERFACE DOPB = mb_opb
 BUS_INTERFACE IOPB = mb_opb
# Microblaze caches connect to the MCH interfaces on the MCH OPE
 BUS_INTERFACE IXCL = ixcl
 BUS_INTERFACE DXCL = dxcl
 PORT CLK = sys_clk_s
 PORT RESET = microblaze_rst
 PORT DBG_CAPTURE = DBG_CAPTURE_s
 PORT DBG_CLK = DBG_CLK_s
 PORT DBG_REG_EN = DBG_REG_EN_s
 PORT DBG_TDI = DBG_TDI_s
 PORT DBG_TDO = DBG_TDO_s
 PORT DBG_UPDATE = DBG_UPDATE_s
 PORT Interrupt = Interrupt
END
```

XAPP932_07_0518

Figure 7: **MicroBlaze XCL Connections in the MHS File**

The MCH OPB EMC uses the following interface connections.

- BUS_INTERFACE MCH0 = ixcl
- BUS_INTERFACE MCH1 = dxcl

This connects the MicroBlaze I-cache to MCH channel 0 and the MicroBlaze D-cache to channel 1 of the MCH OPB EMC.

A portion of the MHS file with the bus interface connections of the MCH OPB EMC is shown in Figure 8.

```
BEGIN mch_opb_emc
 PARAMETER INSTANCE = P160_SRAM_256Kx32_FLASH_1Mx32
 PARAMETER HW_VER = 1.01.a
 PARAMETER C_MCH_OPB_CLK_PERIOD_PS = 15151
 PARAMETER C_NUM_BANKS_MEM = 1
 PARAMETER C_NUM_CHANNELS = 2
 PARAMETER C_INCLUDE_OPB_IPIF = 1
 PARAMETER C_INCLUDE_OPB_BURST_SUPPORT = 1
 PARAMETER C_INCLUDE_DATAWIDTH_MATCHING_0 = 0
 PARAMETER C_SYNCH_MEM_0 = 0
 PARAMETER C_MEM0_WIDTH = 32
 PARAMETER C_MAX_MEM_WIDTH = 32
 PARAMETER C_TCEDV_PS_MEM_0 = 60000
 PARAMETER C_TWC_PS_MEM_0 = 60000
 PARAMETER C_TAVDV_PS_MEM_0 = 60000
 PARAMETER C_TWP_PS_MEM_0 = 60000
 PARAMETER C_THZCE_PS_MEM_0 = 10000
 PARAMETER C_TLZWE_PS_MEM_0 = 10000
 PARAMETER C_MEM0_BASEADDR = 0x30000000
 PARAMETER C_MEM0_HIGHADDR = 0x300fffff
 PARAMETER C_MEM1_BASEADDR = 0x30100000
 PARAMETER C_MEM1_HIGHADDR = 0x301fffff
 PARAMETER C_MEM2_BASEADDR = 0x30200000
 PARAMETER C_MEM2_HIGHADDR = 0x302fffff
 PARAMETER C_MEM3_BASEADDR = 0x30300000
 PARAMETER C_MEM3_HIGHADDR = 0x303fffff
 PARAMETER C_SYNCH_PIPEDELAY_0 = 2
# ICACHE
 PARAMETER C_MCH0_ACCESSBUF_DEPTH = 16
 PARAMETER C_MCH0_RDDATABUF_DEPTH = 0
 PARAMETER C_XCL0_LINESIZE = 4
 PARAMETER C_XCL0_WRITEXFER = 0
# DCACHE
 PARAMETER C_MCH1_ACCESSBUF_DEPTH = 16
 PARAMETER C_MCH1_RDDATABUF_DEPTH = 0
 PARAMETER C_XCL1_LINESIZE = 4
 PARAMETER C_XCL1_WRITEXFER = 1
 BUS_INTERFACE SOPB = mb_opb
# MCH Interfaces connect to the Microblaze Caches
 BUS_INTERFACE MCH0 = ixcl
 BUS_INTERFACE MCH1 = dxcl
 PORT MCH_OPB_Clk = sys_clk_s
 PORT MCH_OPB_Rst = periph_rst_7
 PORT Mem_A = fpga_0_P160_SRAM_256Kx32_FLASH_1Mx32_Mem_A_split
 PORT Mem_DQ = fpga_0_P160_SRAM_256Kx32_FLASH_1Mx32_Mem_DQ
 PORT Mem_BEN = fpga_0_P160_SRAM_256Kx32_FLASH_1Mx32_Mem_BEN
 PORT Mem_WEN = fpga_0_P160_SRAM_256Kx32_FLASH_1Mx32_Mem_WEN
 PORT Mem_OEN = fpga_0_P160_SRAM_256Kx32_FLASH_1Mx32_Mem_OEN
 PORT Mem_CEN = fpga_0_P160_SRAM_256Kx32_FLASH_1Mx32_Mem_CEN
END
```

XAPP932_08_051807

Figure 8: **MCH OPB EMC Interface Connections in the MHS File**

Setting System Parameters for the OPB Central DMA

The addresses of the OPB Central DMA controller must be set to allow the MicroBlaze processor to access its registers.

In the System tab of the OPB Central DMA core, the Base Address is set as `0x41E00000` and the High Address is set to `0x41E0FFFF` as shown in the Figure 9.

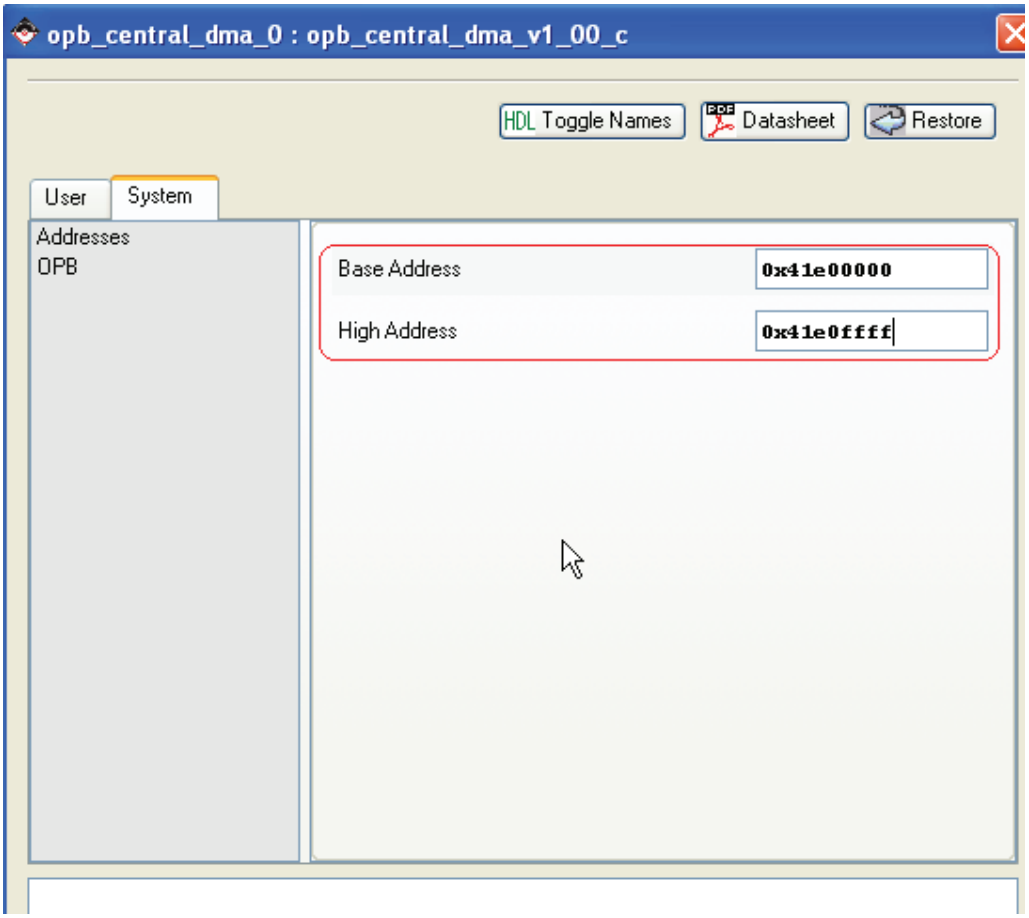

Figure 9: **Setting System Parameters for OPB Central DMA**

Setting User Parameters for the OPB Central DMA

In the User tab of the OPB Central DMA core, set the parameter Enable Read Access for Read -optional Registers to 1 as shown in Figure 10. This enables the software application to read the OPB Central DMA registers to check if the DMA interrupt is properly cleared after a DMA operation has completed.

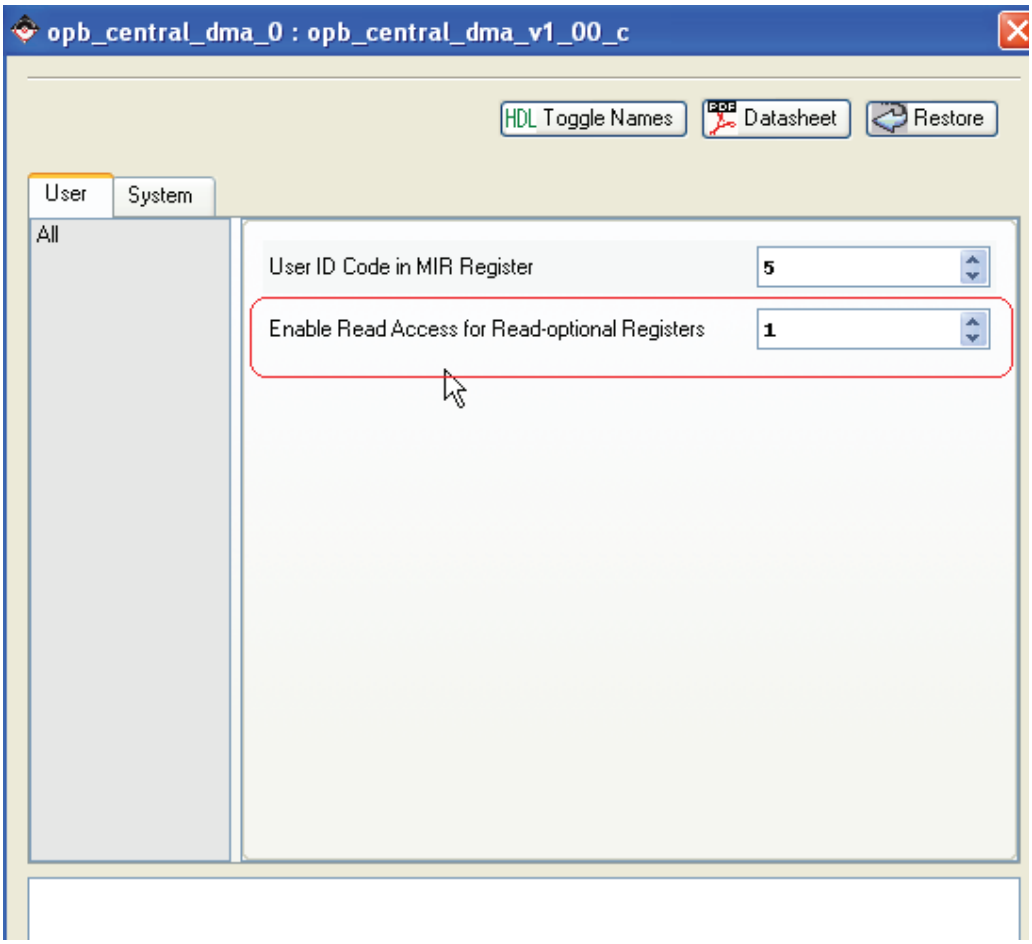

Figure 10: **Setting User Parameters for OPB Central DMA**

Connecting the OPB Central DMA Controller to the System

Connect the **opb_central_dma_0 sopb** as slave and the **opb_central_dma_0 mopb** as master to the OPB Bus connection by marking the connection to the OPB Bus in the BUS INTERFACE filter of the reference system. The DMA interrupt signal is connected to the OPB Interrupt Controller (OPB INTC) in the reference system.

The port connections of the OPB Central DMA to the OPB Bus are shown in Table 3 and are described in Figure 11.

Table 3: **OPB Central DMA port connections**

Port Name	Connection	I/O	Description
SOPB_CLK	sys_clk_s	I	This is the clock for the OPB Central DMA slave.
MOPB_CLK	sys_clk_s	I	This is the clock for the OPB Central DMA master.
DMA_Interrupt	DMA_Irpt	O	This signal goes to the OPB Interrupt Controller.

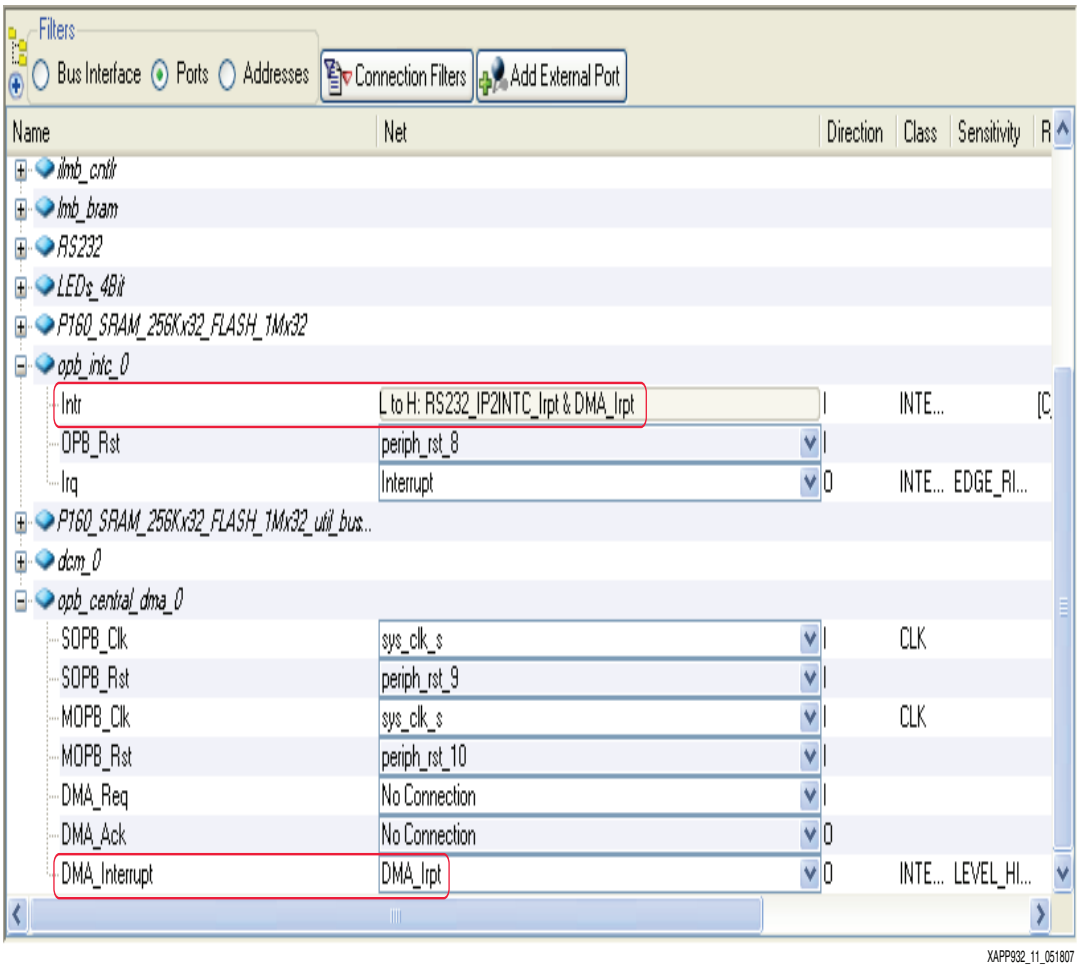

Figure 11: **OPB Central DMA port connections**

The Software Application

The software application, which is executed from the cacheable block of main memory, tests DMA operations out of the non-cacheable block of main memory.

At the start of the application, the memory block at the DMA source address and the DMA destination address are cleared. The data is written to the memory block at the source address. OPB Central DMA is initialized and set up to use interrupts. DMA operations start when the source base address and destination base address are written to the appropriate OPB Central DMA register. An interrupt occurs when the DMA transfer is complete.

When DMA operations are complete, the data at the source address are compared with the data at the destination address to ensure the correct data transfer. Also, the software application clears the interrupt generated by the DMA transfer.

In the software application, set all the linker script options to main memory and do not initialize the block RAMs. Add the source for the software application to the project and copy the source to the new reference system directory. The software application is found under the project root directory `Test_App/TestApp_MemoryCaching.c`.

Executing the Reference System

To execute the reference system, the bitstream needs to be generated and the software application needs to be compiled. The bitstream and the compiled software application for this system are available in `ready_for_download/` under the project root directory.

A HyperTerminal or similar program needs to be connected to the COM port and the board's UART needs to be connected to the COM port. Set the HyperTerminal to the Baud Rate of **9600**, Data Bits to **8**, Parity to **None** and Flow Control to **None**. See Figure 12 for the settings.

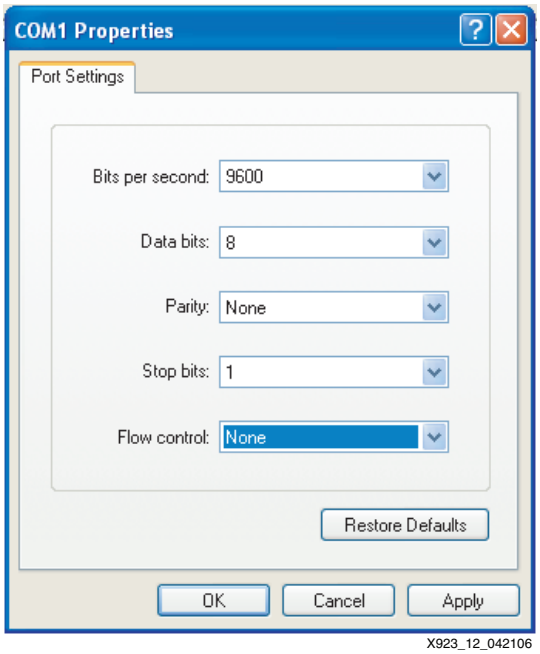

Figure 12: **HyperTerminal Settings**

Executing the Reference System using the Pre-Built Bitstream and the Compiled Software Applications

To execute the system using files inside the `ready_for_download/` in the project root directory, follow these steps:

1. Change directories to the `ready_for_download` directory.
2. Use iMPACT to download the bitstream by using the following:

   impact -batch xapp923.cmd
3. Invoke XMD and connect to the MicroBlaze processor by the following command:

   xmd -opt xapp923.opt
4. Download the executables by the following command:

   dow executable.elf

Executing the Reference System from EDK

To execute the system using EDK, follow these steps:

1. Open `system.xmp` inside EDK.
2. Use **Hardware→Generate Bitstream** to generate a bitstream for the system.
3. Use **Software→Build All User Applications** to build the software applications.
4. Download the bitstream to the board with **Device Configuration→Download Bitstream**.
5. Launch XMD with **Debug→Launch XMD...**
6. Download the executables by the following command:

   dow executable.elf

Running the Software Applications

To run the either of software applications, use the ***run*** command inside XMD. The status of the software application is displayed in the HyperTerminal data screen.

Running the OPB Central DMA Software Application

After downloading the bitstream, download the software application `executable.elf` to main memory using XMD. After downloading the software application, the program must be executed. The status of the software application is displayed to the HyperTerminal. Once the DMA operations are complete and verified, the LEDs blink several times and the output reads as follows:

```
-- Entering main() --

Starting Writing and Clearing Source and Destination Address.
Finished Writing and Clearing Source and Destination Address.
Starting DMA Transfer
Waiting.
DMA Transfer Complete, Verifying Destination Data
Destination Data is Correct
DMA Interrupt Cleared
Congratulations! DMA Operations Completed Successfully!

-- Exiting main() --
```

If the software application fails, the HyperTerminal output reads as follows:

```
-- Entering main() --

Starting Writing and Clearing Source and Destination Address.
Finished Writing and Clearing Source and Destination Address.
Starting DMA Transfer
Waiting.
DMA Transfer Complete, Verifying Destination Data
Read from DMA Destination Address Failed
Interrupt Didn't Clear
DMA Transactions Were Not Successful

-- Exiting main() --
```

Conclusion

This application note describes how to set up the MCH OPB EMC in a MicroBlaze processor system. The reference system is built for the Memec Spartan-3 3S1500 board with the P160 communications Module Rev 2 board. The system includes a software application that runs in the cacheable block of main memory and tests DMA operations out of the non-cacheable block of main memory.

References

UG081, *MicroBlaze Processor Reference Guide*

DS500, *Multi-CHannel OPB External Memory Controller Product Specification.*

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
5/23/06	1.0	Initial Xilinx release.
11/14/06	1.1	Updated for EDK 8.2.02i.
6/5/07	1.2	Updated for EDK 9.1.01i.