XAPP909 (v1.3) June 5, 2007

Reference System: MCH OPB SDRAM with OPB Central DMA

Author: James Lucero

Abstract

This application note demonstrates the use of the Multi-CHannel (MCH) On-Chip Peripheral Bus (OPB) Synchronous DRAM (SDRAM) controller in a MicroBlaze™ processor system. The MCH ports of the MCH OPB SDRAM controller 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 SDRAM controller to handle cacheline access from MicroBlaze and OPB sequential address (burst) transactions on the OPB from the OPB Central DMA controller. The MCH OPB SDRAM memory controller can be implemented on Virtex™-II Pro, Spartan™-3, and Virtex-4 FPGAs with boards that support SDRAM.

The SDRAM memory space is split into a cacheable block of memory and a non-cacheable block of memory. The stand-alone software application provided with this reference system is run out of 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 Virtex-II Pro 2VP7-FG456 demo board.

Included Systems

Included with this application note is the reference system for the Memec Virtex-II Pro 2VP7-FG456 demo board. The reference system is available for download at:

- www.xilinx.com/bvdocs/appnotes/xapp909.zip

Introduction

Many software applications using FPGAs are executed from the main memory. Running applications from the main memory is slower compared to applications run from 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 going out to main memory. Caching instructions and data are stored on high speed memory on the FPGA which gives faster access time. Xilinx offers solutions like MCH OPB SDRAM, MCH OPB DDR, and MCH OPB EMC that use caching for the memory controllers. The reference system described in this application note uses MCH OPB SDRAM and is configured to use caching.

© 2005-2007 Xilinx, Inc. All rights reserved. All Xilinx trademarks, registered trademarks, patents, and further disclaimers are as listed at http://www.xilinx.com/legal.htm. PowerPC is a trademark of IBM Inc. All other trademarks and registered trademarks are the property of their respective owners. All specifications are subject to change without notice.

NOTICE OF DISCLAIMER: Xilinx is providing this design, code, or information "as is." By providing the design, code, or information as one possible implementation of this feature, application, or standard, Xilinx makes no representation that this implementation is free from any claims of infringement. You are responsible for obtaining any rights you may require for your implementation. Xilinx expressly disclaims any warranty whatsoever with respect to the adequacy of the implementation, including but not limited to any warranties or representations that this implementation is free from claims of infringement and any implied warranties of merchantability or fitness for a particular purpose.

Hardware and Software Requirements

The hardware and software requirements are:

- Memec Virtex-II Pro 2VP7-FG456 demo 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 Virtex-II Pro 2VP7-FG456 demo board. The system uses the MicroBlaze processor with 8 KB for both the instruction cache (I-cache), and the data cache (D-cache). By selecting **Enable Cache Link** inside BSB, enables the system to use MCH OPB SDRAM. The OPB Central DMA controller in the reference system generates the burst transactions to the MCH OPB SDRAM. The OPB UARTLite with interrupts and the OPB GPIO 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

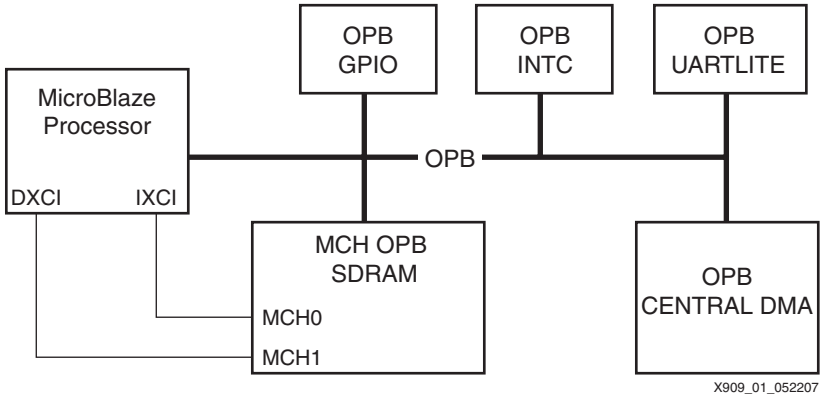

Figure 1: **Reference System Block Diagram**

Address Map

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

Table 1: **Reference System Address Map**

Peripheral	Instance	Base Address	High Address
opb_mdm	debug_module	0x41400000	0x4140FFFF
lmb_bram_if_cntlr	dlmb_cntlr	0x00000000	0x00003FFF
lmb_bram_if_cntlr	ilmb_cntlr	0x00000000	0x00003FFF
opb_uartlite	RS232	0x40600000	0x4060FFFF
opb_gpio	LEDs_4Bit	0x40000000	0x4000FFFF
mch_opb_sdram	SDRAM_8Mx32	0x22000000	0x23FFFFFF

Peripheral	Instance	Base Address	High Address
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 SDRAM is also configured to support the MicroBlaze processor MCH connections and OPB bursting. This illustrates the capacity of the MCH OPB SDRAM to handle OPB bursts and cacheline transactions concurrently.

The following sections explain the configuration of the MicroBlaze processor, the MCH OPB SDRAM, 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 0x22000000 and 0x2200FFFF. 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.

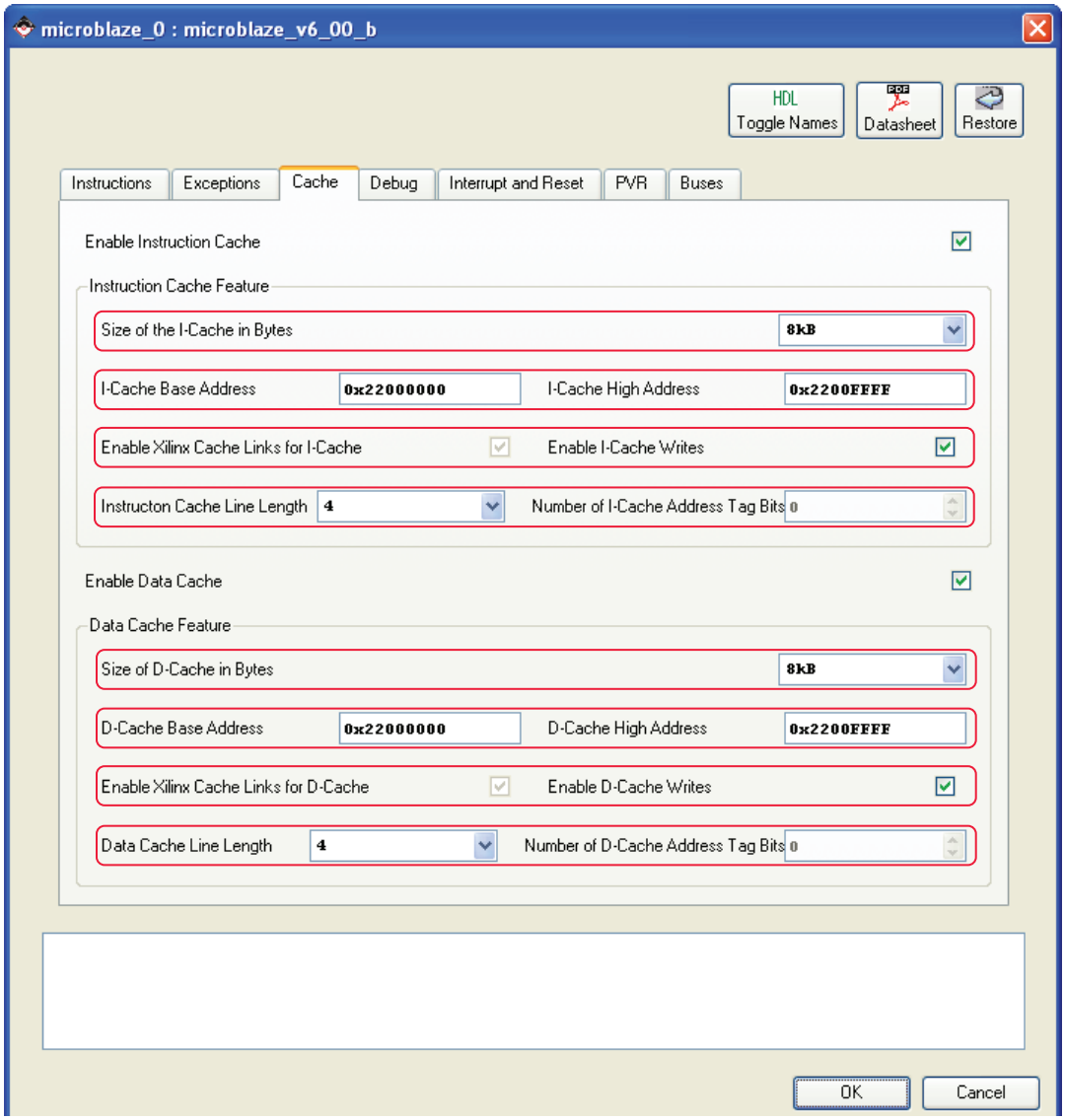

Figure 2: **Setting MicroBlaze Cache Parameters**

Setting the MCH OPB SDRAM parameters

The MCH OPB SDRAM 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 SDRAM needs to be configured to support OPB bursting to handle the transactions from the OPB Central DMA controller effectively.

The MCH OPB SDRAM is configured for 2 channels by configuring the parameter *Number of Channels* to 2. The MCH OPB SDRAM is configured to support burst transactions by setting the parameters *Include A Slave OPB Interface* to 1 and *Support OPB Burst* to 1 in the User tab under the MCH OPB SDRAM core as shown in Figure 3.

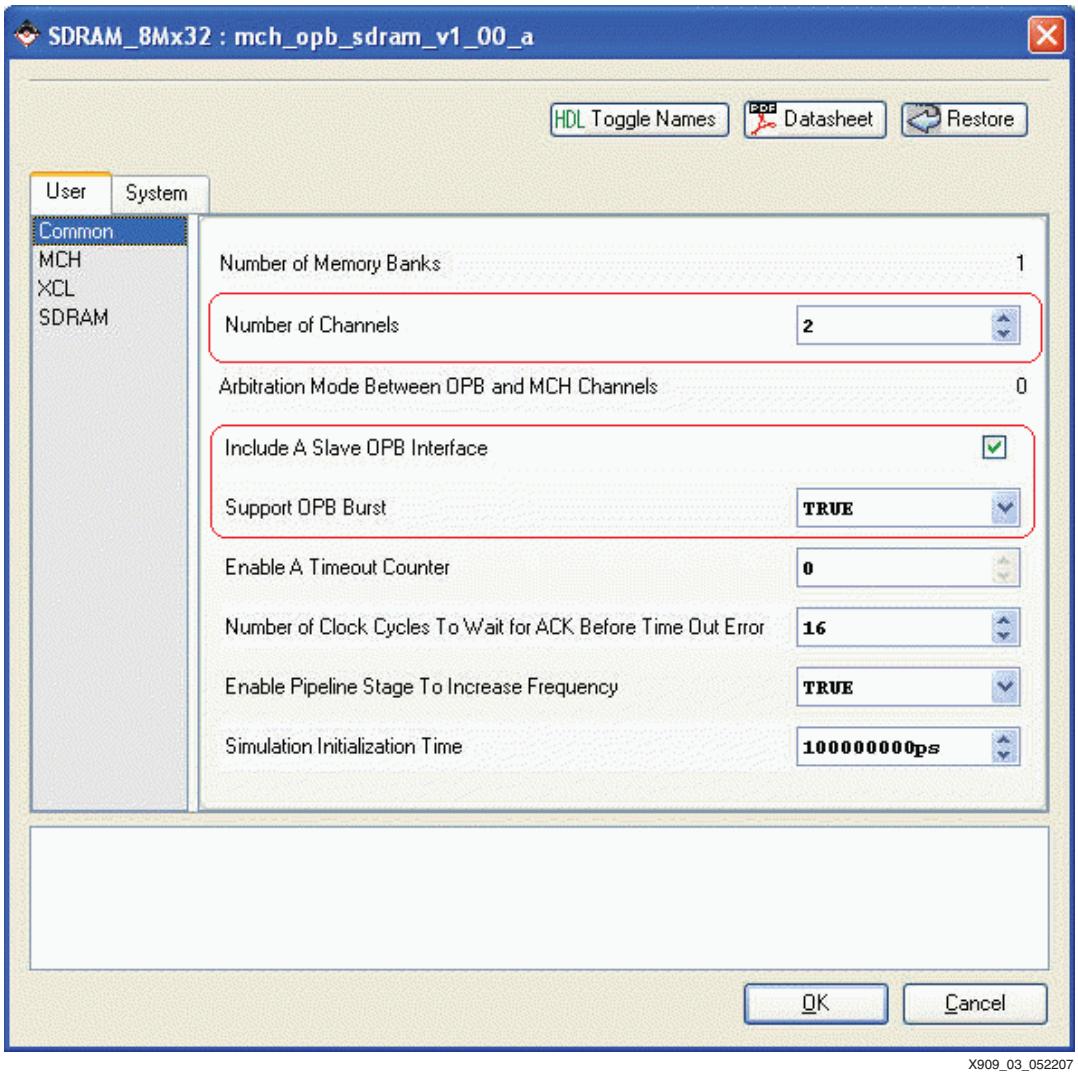

Figure 3: **Setting MCH OPB SDRAM Parameters**

Setting MCH Properties for the MCH OPB SDRAM

The MCH interface properties need to be set for the MCH OPB SDRAM. The MCH OPB SDRAM currently supports only the XCL protocol, therefore the parameter *Interface Protocol for Channels* are set to 0 which indicates the XCL protocol. The *Depth of Access Buffer for Channel* parameter is set to the default value 16 for all the channels. The parameter *Depth of Read Data Buffer for* the channels hat 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 SDRAM core as shown in Figure 4.

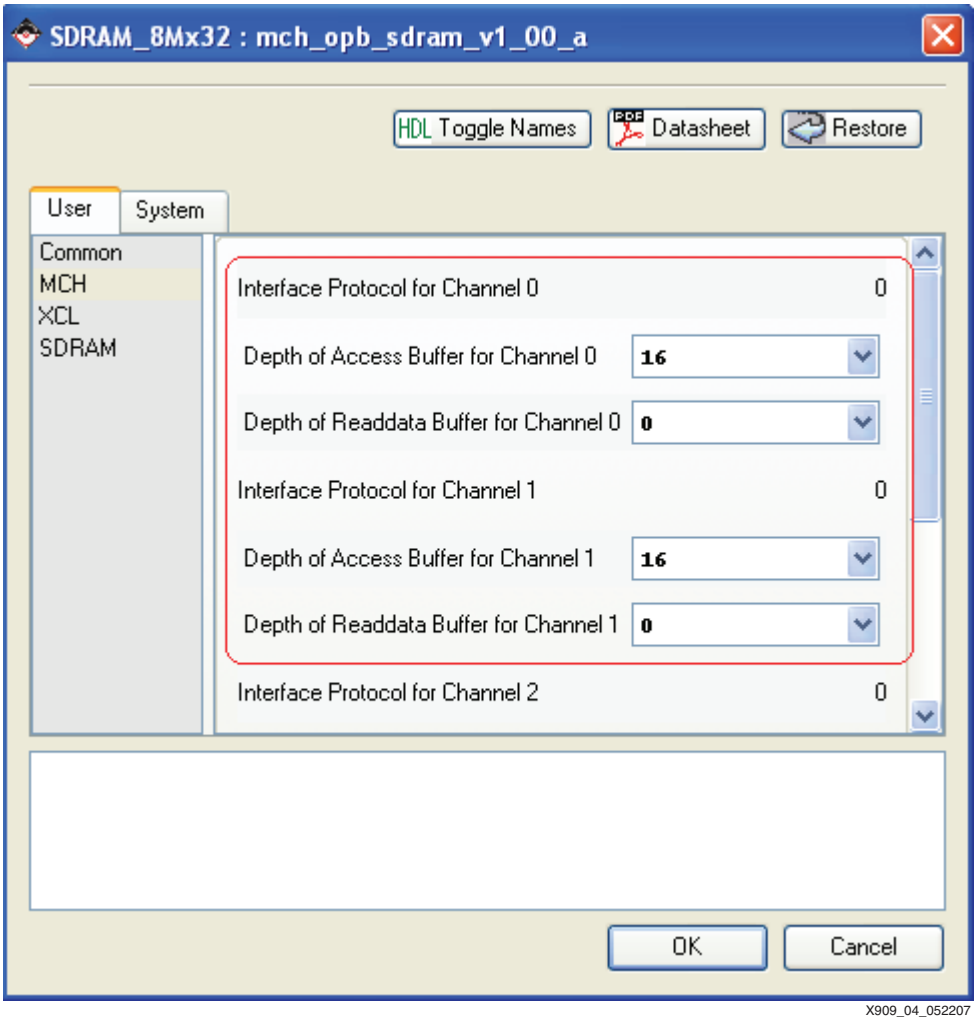

Figure 4: **Setting MCH Properties for MCH OPB SDRAM**

Setting XCL Properties for the MCH OPB SDRAM

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 are set to four words.

The I-cache of MicroBlaze processor will only do read accesses to memory, therefore the parameter *Type of Write Transaction for Channel 0 Configured as XCL* 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 Transaction for Channel 0 Configured as XCL* 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 Transaction for Channel 1 Configured as XCL* 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 SDRAM core as shown in Figure 5.

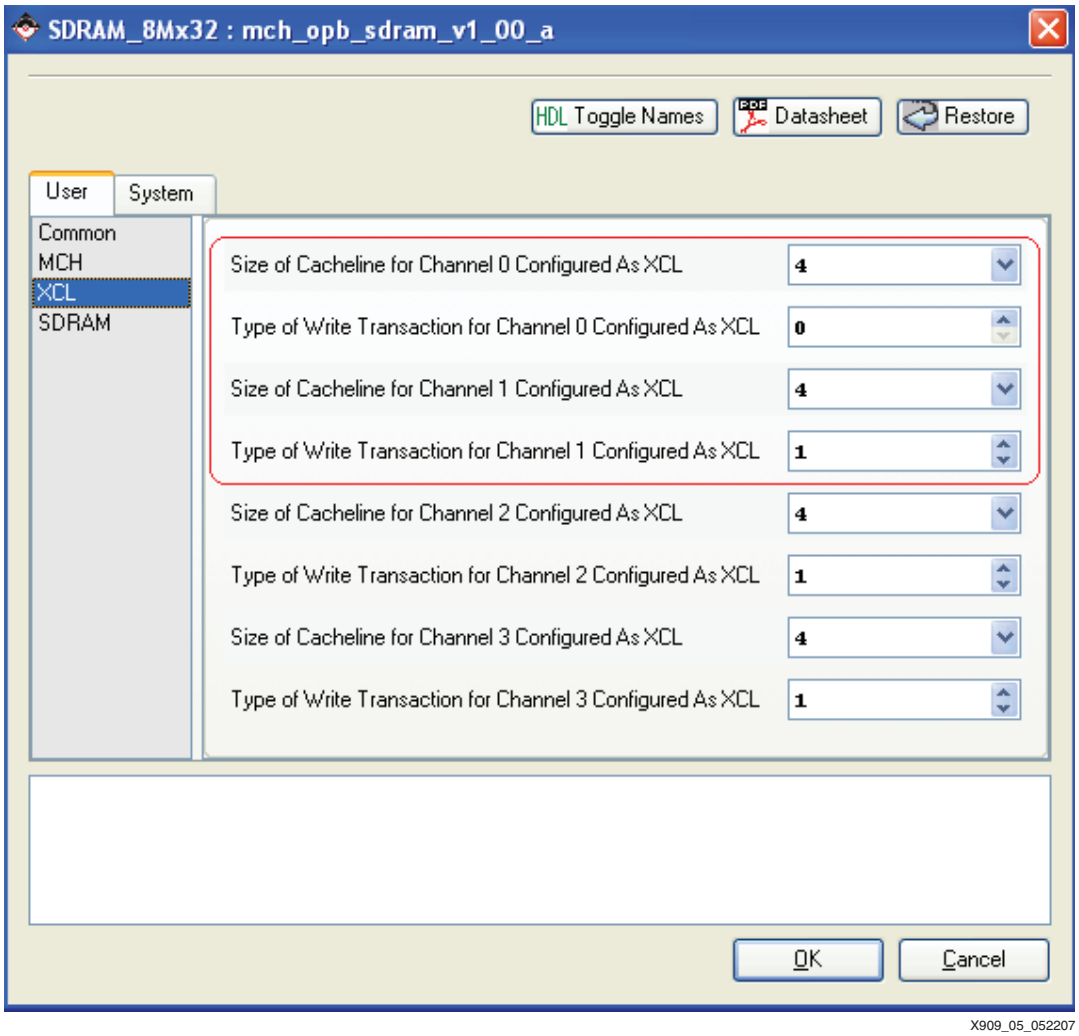

Figure 5: **Setting XCL Properties for MCH OPB SDRAM**

Connecting the MCH OPB SDRAM 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 SDRAM. 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 6.

```
56  BEGIN microblaze
57   PARAMETER INSTANCE = microblaze_0
58   PARAMETER HW_VER = 6.00.b
59   PARAMETER C_DEBUG_ENABLED = 1
60   PARAMETER C_NUMBER_OF_PC_BRK = 1
61   PARAMETER C_NUMBER_OF_RD_ADDR_BRK = 0
62   PARAMETER C_NUMBER_OF_WR_ADDR_BRK = 0
63   PARAMETER C_USE_ICACHE = 1
64   PARAMETER C_CACHE_BYTE_SIZE = 8192
65   PARAMETER C_USE_DCACHE = 1
66   PARAMETER C_DCACHE_BYTE_SIZE = 8192
67   PARAMETER C_ICACHE_USE_FSL = 1
68   PARAMETER C_DCACHE_USE_FSL = 1
69   PARAMETER C_ICACHE_BASEADDR = 0x22000000
70   PARAMETER C_ICACHE_HIGHADDR = 0x2200FFFF
71   PARAMETER C_ADDR_TAG_BITS = 0
72   PARAMETER C_DCACHE_BASEADDR = 0x22000000
73   PARAMETER C_DCACHE_HIGHADDR = 0x2200FFFF
74   PARAMETER C_DCACHE_ADDR_TAG = 0
75   BUS_INTERFACE DLMB = dlmb
76   BUS_INTERFACE ILMB = ilmb
77   BUS_INTERFACE DOPB = mb_opb
78   BUS_INTERFACE IOPB = mb_opb
79   BUS_INTERFACE IXCL = ixcl
80   BUS_INTERFACE DXCL = dxcl
81   PORT CLK = sys_clk_s
82   PORT RESET = microblaze_rst
83   PORT DBG_CAPTURE = DBG_CAPTURE_s
84   PORT DBG_CLK = DBG_CLK_s
85   PORT DBG_REG_EN = DBG_REG_EN_s
86   PORT DBG_TDI = DBG_TDI_s
87   PORT DBG_TDO = DBG_TDO_s
88   PORT DBG_UPDATE = DBG_UPDATE_s
89   PORT Interrupt = Interrupt
90  END
```

X909_06_052207

Figure 6: **MicroBlaze XCL Connections in the MHS File**

The MCH OPB SDRAM 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 SDRAM.

A portion of the MHS file with the bus interface connections of the MCH OPB SDRAM is shown in Figure 7.

```
BEGIN mch_opb_sdram
 PARAMETER INSTANCE = SDRAM_8Mx32
 PARAMETER HW_VER = 1.00.a
 PARAMETER C_INCLUDE_HIGHSPEED_PIPE = 1
 PARAMETER C_SDRAM_TCCD = 1
 PARAMETER C_SDRAM_TRAS = 48000
 PARAMETER C_SDRAM_TRC = 70000
 PARAMETER C_SDRAM_TRFC = 75000
 PARAMETER C_SDRAM_TRCD = 19000
 PARAMETER C_SDRAM_TRRD = 16000
 PARAMETER C_SDRAM_TRP = 19000
 PARAMETER C_SDRAM_TREF = 64
 PARAMETER C_SDRAM_CAS_LAT = 2
 PARAMETER C_SDRAM_COL_AWIDTH = 9
 PARAMETER C_SDRAM_BANK_AWIDTH = 2
 PARAMETER C_SDRAM_AWIDTH = 12
 PARAMETER C_SDRAM_DWIDTH = 32
 PARAMETER C_MEM0_BASEADDR = 0x22000000
 PARAMETER C_MEM0_HIGHADDR = 0x23ffffff
 PARAMETER C_INCLUDE_OPB_BURST_SUPPORT = 1
 PARAMETER C_MCH0_RDDATABUF_DEPTH = 0
 PARAMETER C_MCH1_RDDATABUF_DEPTH = 0
 PARAMETER C_MCH2_RDDATABUF_DEPTH = 0
 PARAMETER C_MCH3_RDDATABUF_DEPTH = 0
 PARAMETER C_XCL0_WRITEXFER = 0
 BUS_INTERFACE SOPB = mb_opb

 # MCH interfaces of the MCH OPB SDRAM connects to the Caches of MicroBlaze
 BUS_INTERFACE MCH0 = ixcl
 BUS_INTERFACE MCH1 = dxcl

 PORT SDRAM_CLK_in = sys_clk_s
 PORT SDRAM_DQ = fpga_0_SDRAM_8Mx32_SDRAM_DQ
 PORT SDRAM_Addr = fpga_0_SDRAM_8Mx32_SDRAM_Addr
 PORT SDRAM_DQM = fpga_0_SDRAM_8Mx32_SDRAM_DQM
 PORT SDRAM_WEn = fpga_0_SDRAM_8Mx32_SDRAM_WEn
 PORT SDRAM_CKE = fpga_0_SDRAM_8Mx32_SDRAM_CKE
 PORT SDRAM_CSn = fpga_0_SDRAM_8Mx32_SDRAM_CSn
 PORT SDRAM_CASn = fpga_0_SDRAM_8Mx32_SDRAM_CASn
 PORT SDRAM_RASn = fpga_0_SDRAM_8Mx32_SDRAM_RASn
 PORT SDRAM_Clk = fpga_0_SDRAM_8Mx32_SDRAM_Clk
 PORT SDRAM_BankAddr = fpga_0_SDRAM_8Mx32_SDRAM_BankAddr
END
```

X909_07_052207

Figure 7: **MCH OPB SDRAM 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 8.

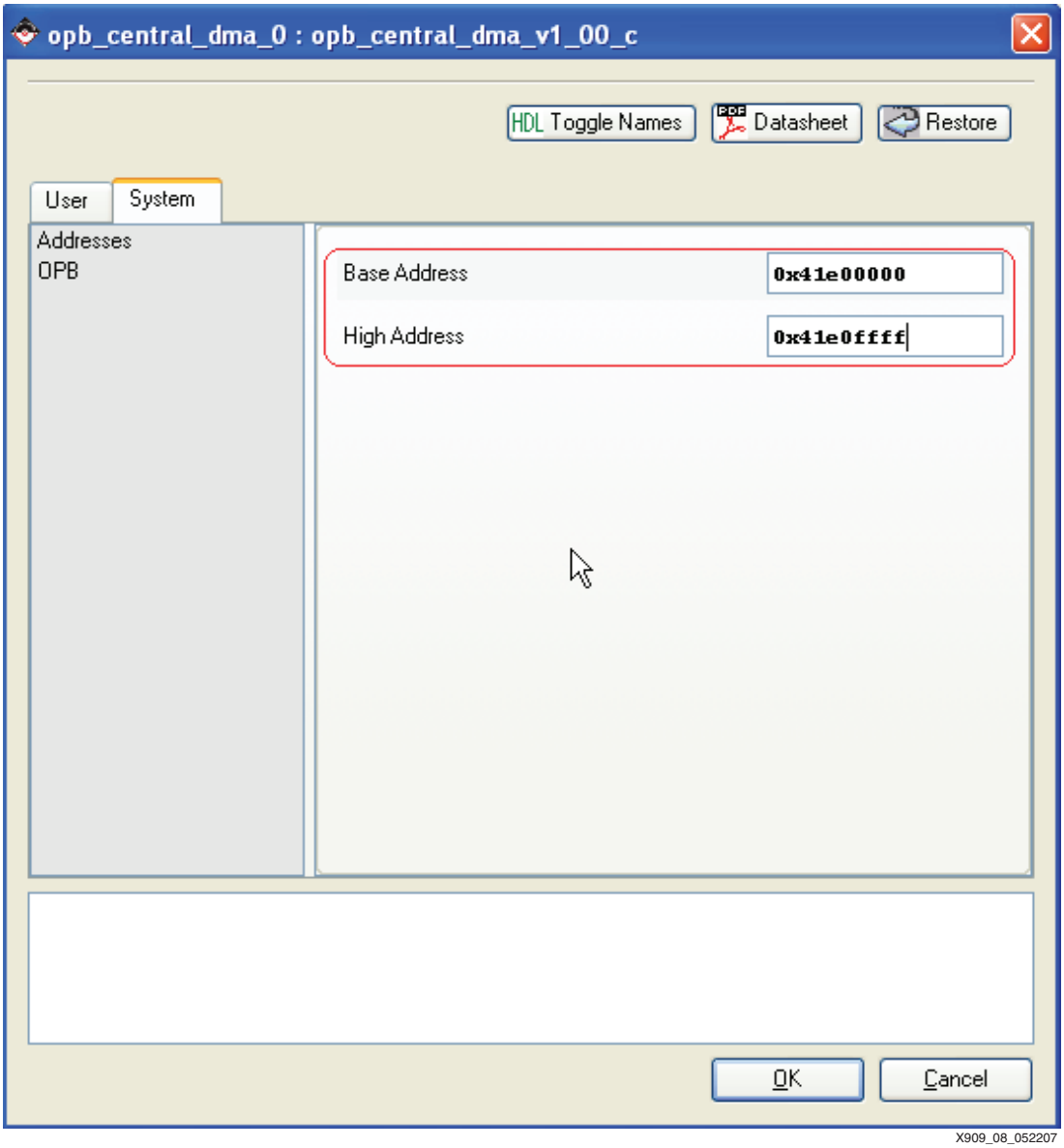

Figure 8: **Setting System Parameters for OPB Central DMA**

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 9. 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.

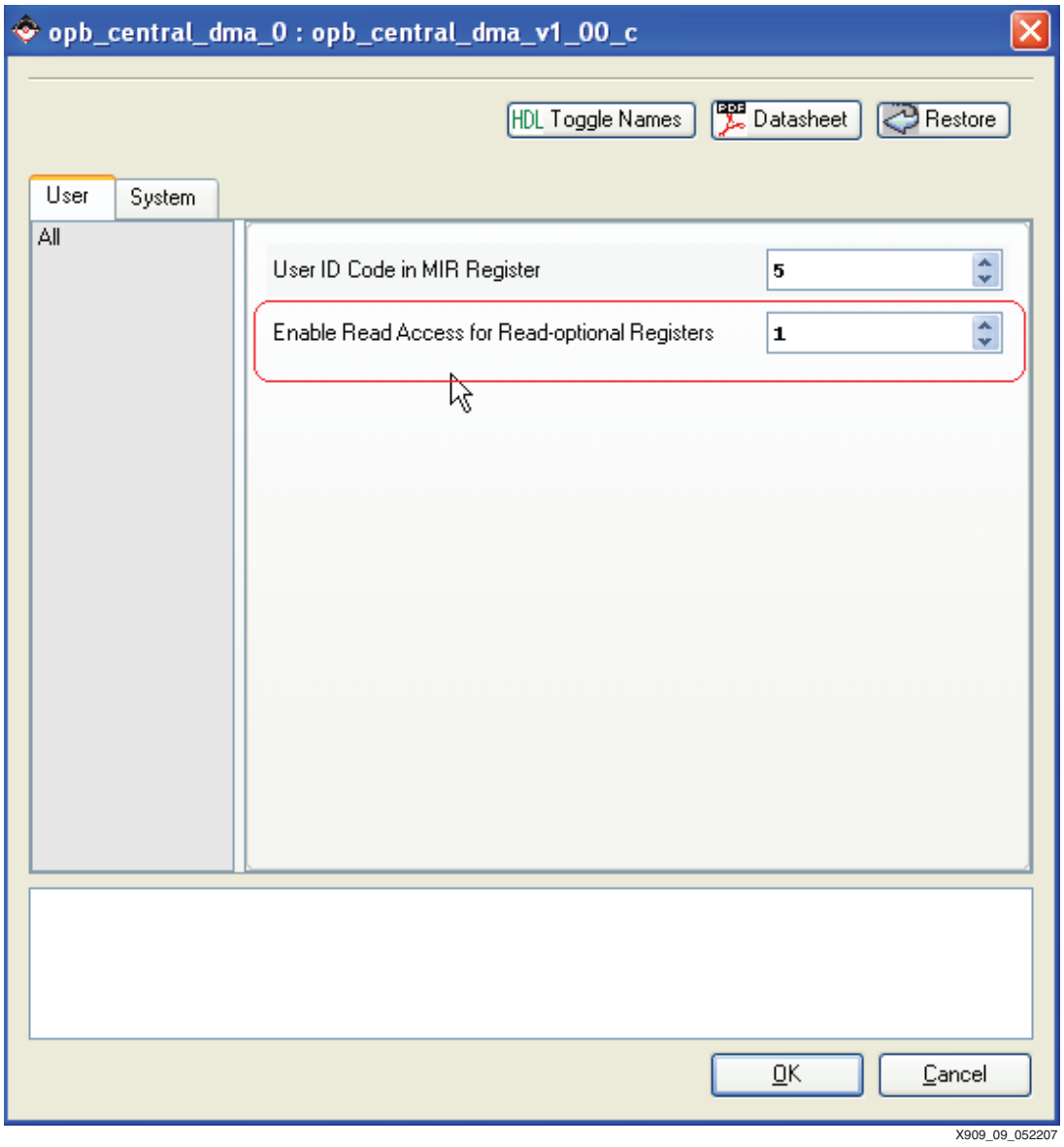

Figure 9: **Setting User Parameters for OPB Central DMA**

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 Bus Interface connections of the OPB Central DMA to the OPB Bus are shown in Figure 10

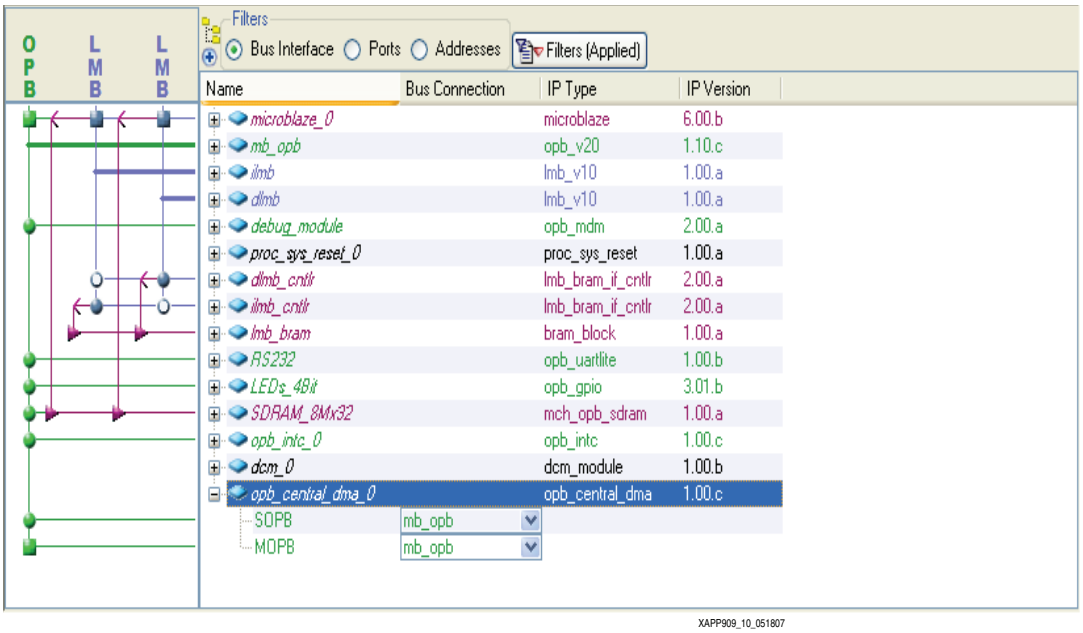

Figure 10: **Bus Interface Filter, Setting Master and Slave for OPB Central DMA**

The DMA interrupt signal is connected to the OPB Interrupt Controller (OPB INTC) in the reference system. Choose the **Ports Filter** and expand the opb_central_dma_0 tree node. Connect the output port **DMA_Interrupt** to **DMA_Irpt**. Obtain the list of interrupts by expanding the opb_intc_0 tree node and clicking on the last port under the **Net** for **Intr**. This will bring up the **Interrupt Connection Dialog** box. Add the **DMA_Irpt** output to Connected Interrupts as shown inThe interrupt connections are shown in Figure 11

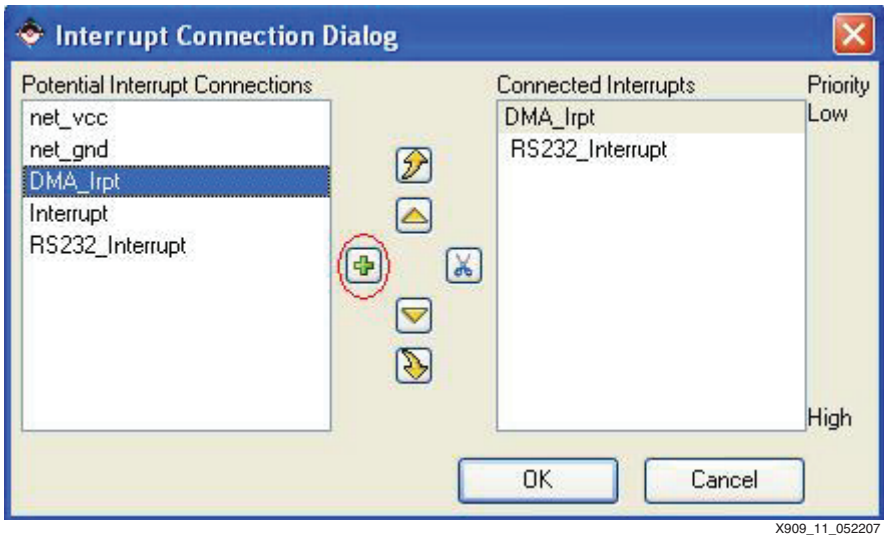

Figure 11: **Interrupt Connect Dialog Box, Adding OPB Central DMA Interrupt**

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.

The software application described above is run out of the on-board SDRAM memory.

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.

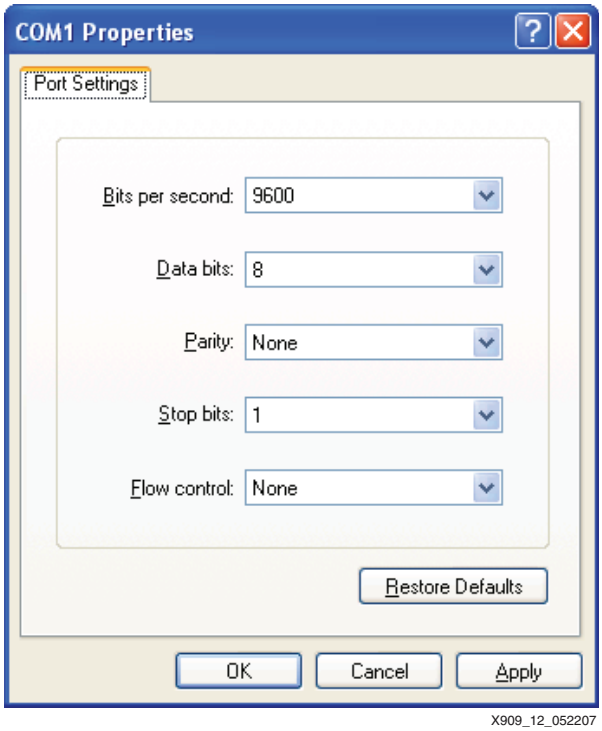

Figure 12: **HyperTerminal Settings**

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 xapp909.cmd
   ```

3. Invoke XMD and connect to the PowerPC® 405 processor by the following command:

   ```
   xmd -opt xapp909.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() --
```

Conclusion

This application note describes how to take advantage of the caching features in the MCH OPB SDRAM memory controller. The reference system (built for the Memec Virtex-II Pro 2VP7-FG456 demo board) includes a stand-alone software application that runs inside the cacheable block of main memory. This application tests DMA operation inside the non-cacheable block.

References

UG081, *MicroBlaze Processor Reference Design*

DS492, *MCH OPB Synchronous DRAM (SDRAM) Controller (v1.00a)* Product Specification

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
10/19/05	1.0	Initial Xilinx release.
10/26/05	1.1	Fixed www.xilinx.com/bvdocs/appnotes/xapp909.zip link.
9/21/06	1.2	Updated to the latest Iron version.
6/5/07	1.3	Updated for EDK 9.1.01i.