Multi-Port Memory Controller 2 (MPMC2) IP Configurator Interface

User Guide

UG245 (v2.0) April 6, 2007

XILINX®

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
02/14/06	1.0	Initial Xilinx release
04/03/06	1.0.1	Minor text edits
10/20/06	1.2	EDK/ISE 8.2 release
04/06/07	2.0	EDK/ISE 9.1i release

Table of Contents

About This Guide

The *Multi-Port Memory Controller 2 (MPMC2) IP Configurator Interface* shows how to install the IP Configurator, how to build a pre-configured pcore, and how to customize pcores to suit specific systems. The Multi-Port Memory Controller 2 is referred to in this document as MPMC2.

Additional Resources

To find additional documentation, see the Xilinx website at:

www.xilinx.com/support/library.htm.

To search the Answer Database of silicon, software, and IP questions and answers, or to create a technical support WebCase, see the Xilinx website at:

www.xilinx.com/support/mysupport.htm.

Conventions

This document uses the following conventions. An example illustrates each convention.

Typographical

The following typographical conventions are used in this document:

Convention	Meaning or Use	Example
`Courier font`	Messages, prompts, and program files that the system displays	`speed grade: - 100`
`Courier bold`	Literal commands that you enter in a syntactical statement	**`ngdbuild`** *design_name*
Helvetica bold	Commands that you select from a menu	**File →Open**
	Keyboard shortcuts	**Ctrl+C**

Convention	Meaning or Use	Example
Italic font	Variables in a syntax statement for which you must supply values	**ngdbuild** *design_name*
	References to other manuals	See the *Development System Reference Guide* for more information.
	Emphasis in text	If a wire is drawn so that it overlaps the pin of a symbol, the two nets are *not* connected.
Square brackets [ ]	An optional entry or parameter. However, in bus specifications, such as **bus[7:0]**, they are required.	**ngdbuild** [*option_name*] *design_name*
Braces { }	A list of items from which you must choose one or more	**lowpwr =**{**on**|**off**}
Vertical bar |	Separates items in a list of choices	**lowpwr =**{**on**|**off**}
Vertical ellipsis . . .	Repetitive material that has been omitted	IOB #1: Name = QOUT' IOB #2: Name = CLKIN' . . .
Horizontal ellipsis ...	Repetitive material that has been omitted	**allow block** *block_name loc1 loc2 ... locn;*

Online Document

The following conventions are used in this document:

Convention	Meaning or Use	Example
Blue text	Cross-reference link to a location in the current document	See the section "Additional Resources" for details. Refer to "Title Formats" in Chapter 1 for details.
Red text	Cross-reference link to a location in another document	See Figure 2-5 in the *Virtex-II Platform FPGA User Guide.*
Blue, underlined text	Hyperlink to a website (URL)	Go to http://www.xilinx.com for the latest speed files.

Multi-Port Memory Controller 2 (MPMC2) IP Configurator Interface

Introduction

This user guide describes the features of the MultiPort Memory Controller 2 (MPMC2) IP Configurator interface, covering the generation of both preconfigured and customized MPMC2 pcores for use in XILINX™ Platform Studio (XPS). Multiple XPS reference projects are included to help you get up and running within a few mouse clicks.

MPMC2 pcores can be configured to access one to eight memory ports and include a set of port interface modules for connecting to PowerPC™ 405 processor (PPC405), MicroBlaze™, and CoreConnect™ structures. The MPMC2 pcore also supports the Communication Direct Memory Access Controller (CDMAC) that provides full-duplex, high-bandwidth, 32-bit LocalLink interfaces into memory.

Technical Support

Check the MPMC2 website at www.xilinx.com/esp/wired/optical/xlnx_net/mpmc.htm for updated designs, documentation, and helpful hints.

To receive support on MPMC2, file a web case at www.xilinx.com/support/clearexpress/websupport.htm or contact your local Processor Specialist FAE.

Additional Resources

- *ML300 Reference Design*
  http://www.xilinx.com/products/boards/ml300/docs/ml300_ref_des_ug.pdf
- *ML40X EDK Processor Reference Design*
  http://www.xilinx.com/bvdocs/userguides/ug082.pdf
- *Multi-Port Memory Controller 2 (MPMC2) User Guide* :
  .<MPMC2 Install Directory>/docs/ug253.pdf
- *PowerPC 405 Processor Block Reference Manual*
  http://www.xilinx.com/bvdocs/userguides/ug018.pdf

Requirements

Windows

- Microsoft Windows® XP (Service Pack 2)

Linux

- Red Hat® Linux™ (kernel 2.4 or later)

Software

- EDK 9.1 (Service Pack 1 or later)
- ISE™ 9.1 (Service Pack 3 or later)

Note: For currently supported features, see the `release_notes_<release date>.txt` file located in the docs directory of the MPMC2 installation.

Installation

Before installing the MPMC2 IP Configurator, you must have installed ISE and EDK, along with the required service packs.

Windows XP:

1. Unzip MPMC2 into its own directory (for example, `C:\mpmc2`).
2. Ensure that the **`%XILINX_EDK%/cygwin/bin`** executable path is in the PATH environment variable. From a command prompt, type **`sed -V`** to verify.

Linux:

Unzip MPMC2 into its own directory.

Building a Preconfigured Pcore

This section explains how to build a preconfigured design using one of the provided reference cores. The step numbers correspond to the numbers in the figure callouts.

1. Open the MPMC2 IP Configurator, as shown in Figure 1.

   Windows XP:

   a. Open a windows browser and navigate to the mpmc2 install directory.

   a. Double-click `mpmc2_gui.exe`.

   Linux:

   From a command prompt, type:

   ```
   cd <mpmc2_install_directory>./mpmc2_gui
   ```

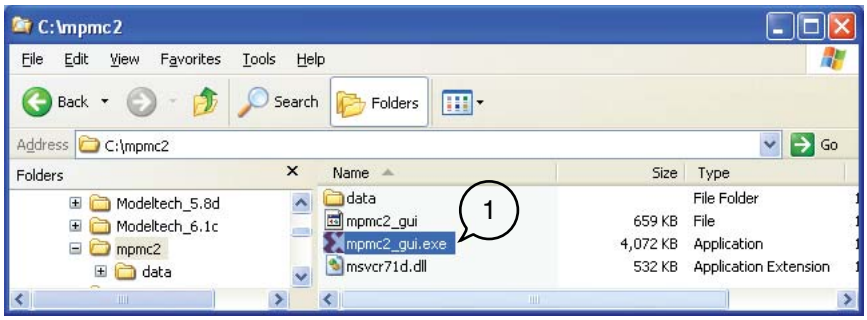
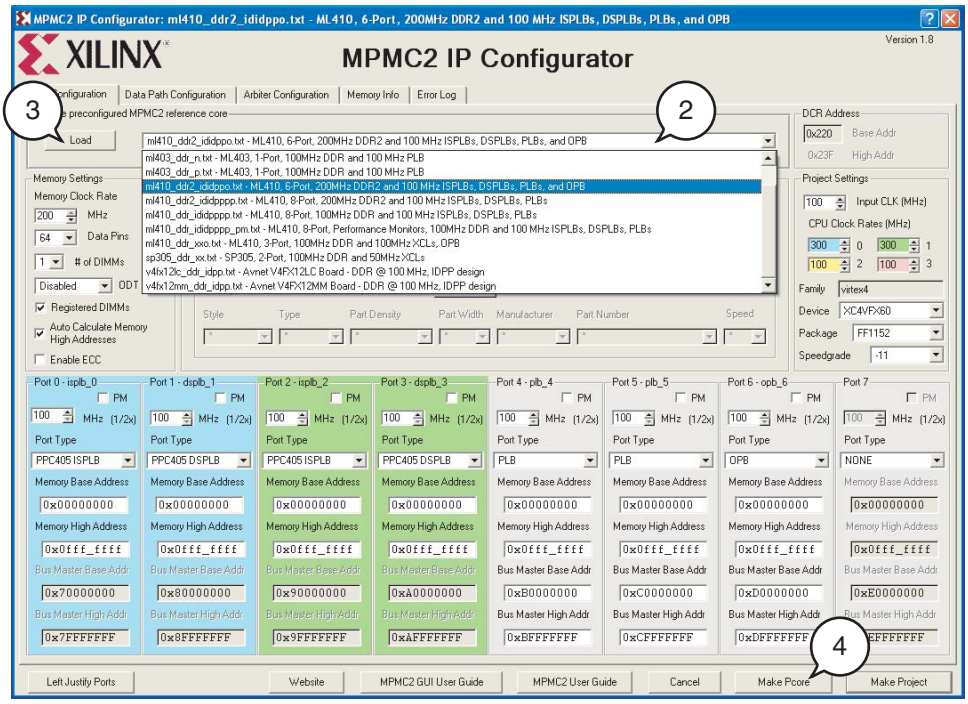


Figure 1: **Launching the MPMC2 IP Configurator**

2. Select a reference core from the drop-down list.
3. Click **Load** to load the chosen design.

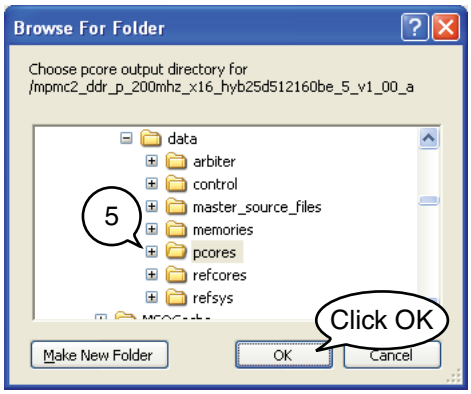
   Note: This version of the interface (v1.8) is not backward compatible with older interface-generated files. Do not use the version 1.8 interface to load designs created with older interfaces. To update a design from an older MPMC2 release, you must generate a new pcore by starting from an existing v1.8 reference core and modifying it in the interface to match the settings from the older interface design.
4. Click **Make Pcore**.



UG245_02_030907

Figure 2: **Select a Preconfigured Core**

5. Choose an output directory for the selected pcore and click **OK**. The default is `<install_directory>/data/pcores`.



UG245_03_020306

Figure 3: **Choose Output Directory**

After completing these steps, the MPMC2 pcore is ready to be integrated into an EDK project.

Building a Custom Pcore

Customizing an MPMC2 pcore can range from making a small frequency change to completely reconfiguring the port types and arbiter algorithms. This section outlines the options available to customize an MPMC2 pcore for different system needs.

The IP Configurator is divided into the following tabs:

- "Base Configuration Tab"
- "Data Path Configuration Tab"
- "Arbiter Configuration Tab"
- "Memory Info Tab"
- "Error Log Tab"

The following sections step through these tabs and define the settings and options.

Base Configuration Tab

The interface options that are available under the Base Configuration tab, shown in Figure 4, are the starting point for defining an MPMC2 core. To build a custom system, begin by loading a pre-configured reference core that closely resembles the system you require.

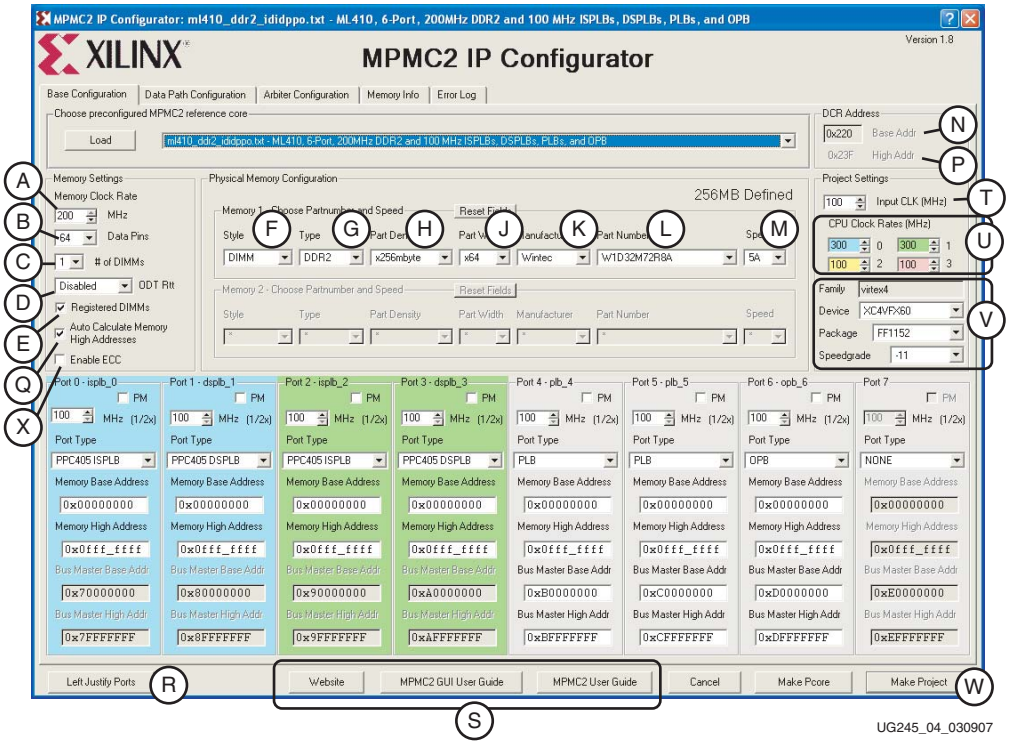


Figure 4: **MPMC2 Configuration Options**

System Memory Settings

The settings in this section apply to all of the memory in a configuration. The lettered sub-headings correspond to the letters in Figure 4, page 11.

A. Memory Clock Rate

The **Memory Clock Rate** is the speed at which the memory is clocked. For example, 200 MHz = DDR400, 100 MHz = DDR200.

B. Data Pins

The **Data Pins** setting specifies the number of actual data pins connected to the memory devices. This number does not include any additional bits used for Error Correction Code (ECC).

C. # of DIMMs

The **# of DIMMs** setting indicates to the pcore how many chip select pins are attached to the FPGA or how many DIMMs are installed.

D. ODT Rtt

For DDR2 memories, the **ODT Rtt** drop-down box specifies whether On-Die Termination (ODT) is enabled or disabled. If enabled, you can select the termination value.

E. Registered DIMMs

The **Registered DIMMs** checkbox lets you specify that the DIMM in use has on-board register devices.

X. Enable ECC

The **Enable ECC** checkbox lets you specify that Error Correction Code (ECC) be enabled.

Physical Memory Configuration

The fields in this section are for selecting a specific memory part and speed grade. The values in the Part Number field are filtered by values chosen in the **Style**, **Type**, **Part Density,** and **Part Width** fields. If a combination is not valid or does not exist, the **Part Number** field displays `NOT FOUND`. When you choose a speed grade the corresponding the memory settings for a specific device loaded into the MPMC2 configuration, and those settings display in the **Memory Info** tab.

F. Style

The **Style** drop-down list lets you select between discrete memory components placed on a board, or a plug-in DIMM memory module.

G. Type

The **Type** field lets you specify the fundamental memory type to use with the MPMC2 core.

H. Part Density

The **Part Density** setting denotes the number of bits of storage for any particular part in the memory configuration. For DIMMs, the part density describes the total size of the DIMM module.

J. Part Width

The **Part Width** is the per-part data width. In the case of a DIMM, part width is the entire data width of the DIMM module.

K. Manufacturer

Manufacturer is the manufacturer of the memory components or memory module.

L. Part Number

Part Number is the manufacturer device or DIMM part number.

M. Speed

Speed is the speed grade of the memory device.

DCR Address

MPMC2 uses the Device Control Register (DCR) bus of the PowerPC 405 processor to connect to the CDMAC Port Interface Modules (PIMs), Error Correction Code (ECC) registers, and the per-port Performance Monitors (PMs). Each instance of a CDMAC PIM and ECC support consumes 16 locations of DCR space. Additionally, if any per-port performance monitors are instantiated, an additional 16 locations of DCR space is consumed. Unlike the CDMAC PIMs, the performance monitors only consume 16 locations regardless of how many are used.

N. Base Addr

You can alter the DCR **Base Address** to a location of your choice. The DCR base address must be 16 location aligned (for example, the LSD must always be a zero).

P. High Addr

The DCR **High Address** is automatically set based on how the MPMC2 pcore is configured.

Other Selections

Q. Auto Calculate Memory High Addresses

When you check **Auto Calculate Memory High Address**, the MCMP2 IP configurator auto-calculates the memory high address for each port based on the memory size and the defined memory base address.

R. Left Justify Ports

You must define MPMC2 ports in order, from port 0 to port N. If a port type in the middle of the port list is changed to **`NONE`**, the ports must be left-justified. When you click the **Left Justify Ports** button, it automatically moves all ports to the left and aligns the defined ports in series. You cannot save an MPMC2 system unless you have left-justified the ports.

S. Documentation Buttons

When you click the **Documentation Buttons**, it opens the corresponding web page or documentation.

V. Family, Device, Package, and Speed Grade

This section is used for pcore and project generation, and for the **BRAM Management Report Window**. It specifies the target FPGA for the design.

Project Generation

The MPMC2 interface can create a template EDK project based on the configuration setting in the IP Configurator interface.

T. Input CLK

The **Input CLK** section is for project generation only, and allows you to specify the master system clock input frequency. This value guides the clock and DCM creation in the template project.

U. CPU Frequencies

The **CPU Frequencies** section is for project generation only, and allows you to specify the required CPU frequencies. The color of the boxes for CPU 0, 1, 2, and 3 correspond to the processor ports that you specify. These values guide the clock and DCM creation in the template project.

W. Make Project

Note: This Project Generation feature is being phased out. By default, the interface will no longer show the "Make Project" button. You can enable the Make Project button by setting `allow_pg` to **`1`** in the file `<mpmc2_install>/data/ver.txt`.

When you click the **Make Project** button, the MPMC2 IP Configurator interface prompts for a directory and project name, and when those names are specified it generates a project that can be loaded into XPS. MPMC2 project generation provides a template that includes an XPS project from which to start designing, with MPMC2, memory connections, processors, clocks, and some peripheral devices already configured. When using project generation, the following are the typical required steps:

1. Use XPS or edit the MHS file as needed to add IP to the system.
2. Edit the UCF file for your board pinout, and add any other IP ports that are required for your design.
3. Add software projects as appropriate; none are provided.
4. If you use CDMAC PIMs, you must connect the LocalLink interfaces either by using XPS or by editing the MHS file (or both) prior to running Platgen.
5. Verify that the clock, reset, and DCM connections are correct for your board layout, system, and clock frequencies. The clock, reset, and DCM connections are provided for illustrative purposes and might not be correct in all conditions. Verify all connections and settings before implementing the design in hardware.
6. Review the MHS file for informational messages and placeholder text that requires user input or modification.
7. By default, MPMC2 project generation connects PowerPC 405 processors to ISPLB and DSPLB ports, MicroBlaze to XCL ports, PLB TFT Controllers (v1.00.b and v1.00.c) to PLB Ports, and UART to the OPB ports. Edit these port and IP connections as necessary.
   - Processor connections include local processor memory (OCM and LMB BRAMs) and JTAG debug ports (JTAGPPC and MDM).
   - the PLB TFT Controller is a PLB master capable of burst read transactions to read and display video data.
     - v1.00.b performs eight-word cache line reads
     - v1.00.c performs 16 double-word burst reads

- Refer to the *ML300 Reference Design* document for more information about the PLB TFT Controller v1.00.b. "Additional Resources," page 7 contains a link to this reference document.
- Refer to the *ML40X EDK Processor Reference Design* for more information about the PLB TFT Controller v1.00.c. "Additional Resources," page 7 contains a link to this reference document.

Also refer to the following text files for more information:

```
<mpmc2_install>\data\proj_readme.txt
<mpmc2_install>\data\my_ip.txt
<mpmc2_install>\data\how_to_add_your_ip.txt
```

Note: You must review carefully and verify all system settings and connections. The system created by the MPMC2 project generation feature should be used as an approximate guide only for building MPMC2 systems, and it might not be correct under all conditions. MPMC2 project generation might not instantiate the latest version of IP cores.

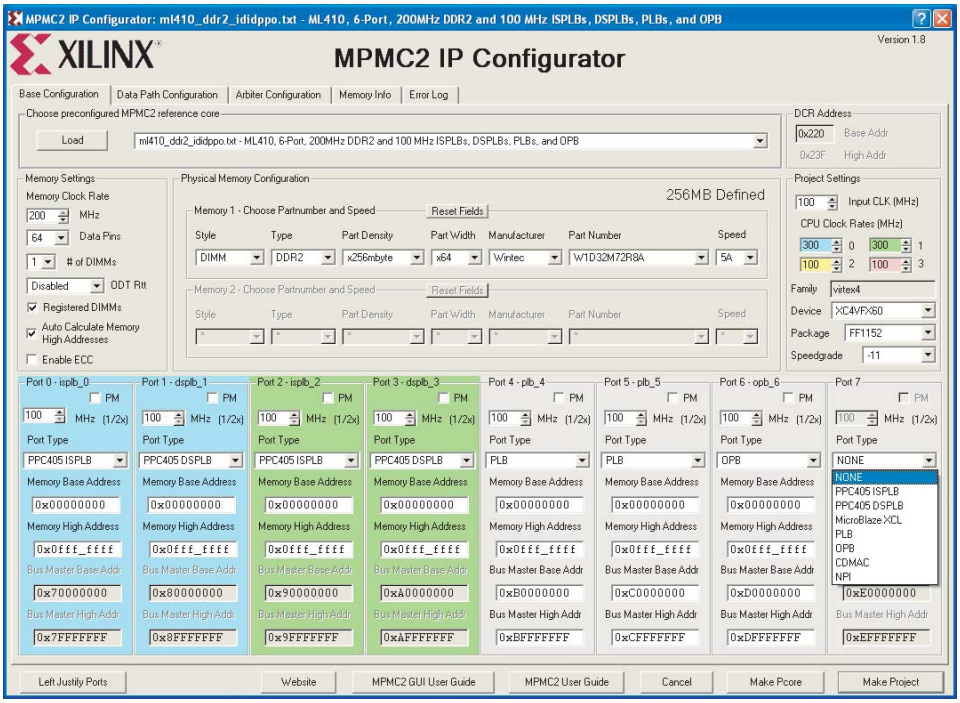
Selecting Port Interface Modules

Figure 5, page 17 shows the **Port Type** drop-down list. You can define each of the eight ports as any one of the seven Port Interface Modules (PIMs) described in Table 1, page 18. The port types are undefined by default, as indicated by **NONE**.

You can specify a memory base address and memory high address for each defined port, outlining the memory address space to which each port will have access.

The On-Chip Peripheral Bus (OPB) and Processor Local Bus (PLB) port types allow master ports only when you have defined data-side PLB (DSPLB) port types, thus activating the **Bus Master Base Addr** and **Bus Master High Addr** fields. The Bus Master fields are for specifying the address range on the OPB or PLB bus in which master requests from the DSPLB are accepted.

The PM checkbox in each port enables a performance monitor for each port. For details, refer to the *Multi-Port Memory Controller 2 (MPMC2) User Guide* and the release notes in `<install_directory>/docs`.



UG245_05_030907

Figure 5: **Selecting Port Types**

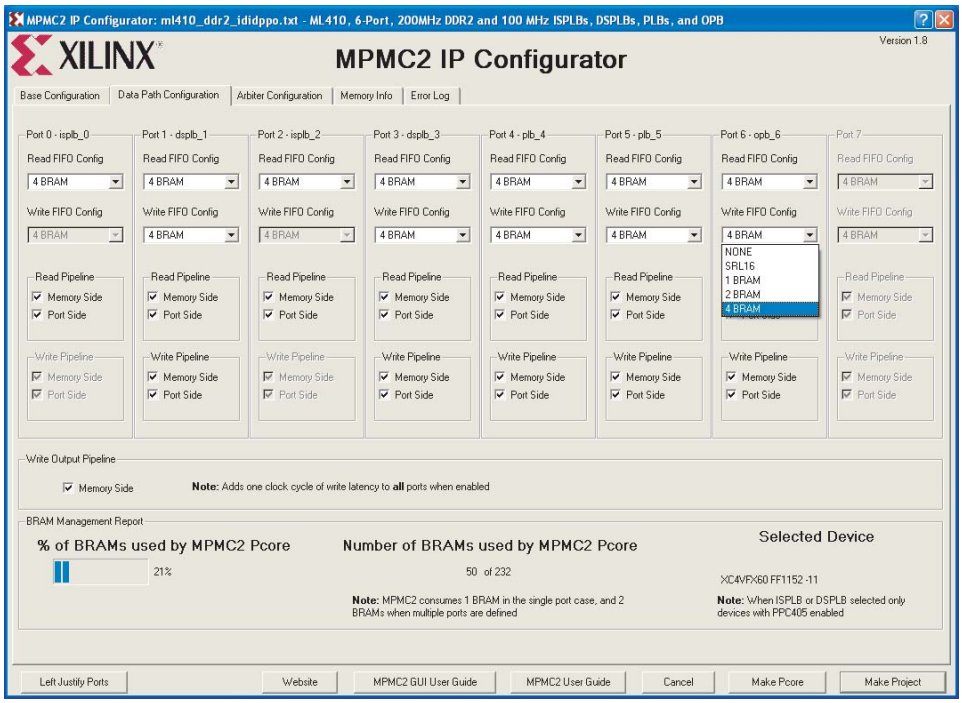
The "Additional Resources," page 7 contains links to the *PowerPC 405 Processor Block Reference Manual*, which is referenced in Table 1.

Table 1: **Port Types**

Port Interface Module Type	Description
NONE	NONE indicates an undefined port and requires that any ports to the right of it are left-justified. See "R. Left Justify Ports," page 14.
PowerPC 405 ISPLB	The Instruction-Side Port Logical Block (ISPLB) module connects the PowerPC 405 instruction-side PLB interface directly to the MPMC2. The ISPLB port is a dedicated MPMC2 port that avoids the cost of arbitration present with the typical PLB. The ISPLB module has higher performance because it is designed specifically for ISPLB behavior as defined in the *PowerPC 405 Processor Block Reference Manual*.
PowerPC405 DSPLB	The Double-Sided Port Logical Bus (DSPLB) module connects the PowerPC 405 DSPLB interface directly to the MPMC2. The DSPLB port is a dedicated MPMC2 port that avoids the cost of arbitration present with the typical PLB. The DSPLB module has higher performance than others because it is designed specifically for DSPLB behavior as defined in the *PowerPC 405 Processor Block Reference Manual*.
MicroBlaze XCL	The XCL module connects a MicroBlaze Xilinx Cache Link (XCL) interface to the MPMC2. This module contains the logic to respond to any slave-side XCL transaction the MicroBlaze soft processor is capable of emitting.
PLB	The Port Logical Bus (PLB) module connects PLB devices to the MPMC2 pcore. This module contains logic to respond to any slave-side transactions provided by external arbiters. In addition, the PLB module can initiate master transactions on the bus from the DSPLB modules, allowing CPUs to communicate directly to any PLB device.
OPB	The On-Chip Peripheral Bus (OPB) module connects OPB devices to the MPMC2 pcore. The OPB module contains logic to respond to any slave side transactions provided by external arbiters. In addition, the OPB module can initiate master transactions on the bus from the DSPLB modules, allowing for CPUs to communicate directly to any OPB device.
CDMAC	The Communications Direct Memory Access Controller (CDMAC) module connects a CDMAC to the MPMC2 pcore. Each CDMAC contains a full-duplex, 32-bit LocalLink interface. These LocalLink interfaces can be used for custom purposes, or to connect to the LL_xEMACs. The CDMAC module provides highly intelligent DMA between communication devices. It uses a DCR interface to set parameters, and reads all of its descriptors directly from memory with a minimum of local CPU intervention. The CDMAC module supports full scatter and gather operations and numerous other features useful to communication devices.
NPI	The Native Port Interface (NPI) module is the native port interface for use with custom port interface modules.

Data Path Configuration Tab

This section defines the options available under the Data Path Configuration tab shown in Figure 6.



UG245_06_030907

Figure 6: **Selecting FIFOs and Device**

Per-Port Data Path Configuration

You can configure each port to support different implementations of FIFOs. Two basic FIFO types are supported: Shift Register Lookup tables (SRL) and Block RAM (BRAM). Depending on memory width, FIFOs can have different sizes:

- Use two BRAMs with eight-, 16-, and 32-bit memory widths
- Use four BRAMs with 64-bit memory widths

Refer to the *Multi-Port Memory Controller 2 (MPMC2) User Guide* and the release notes in the `<installation_directory>/docs` directory to determine which FIFO configurations are valid for a given memory type.

Read Pipeline

The **Memory Side** checkbox adds one pipeline stage to the input (memory side) of the read data path FIFOs. The **Port Side** checkbox adds one pipeline stage to the output (port interface side) of the read data path FIFOs.

Write Pipeline

The **Memory Side** adds one pipeline stage to the output (memory side) of the write data path FIFOs. The **Port Side** checkbox adds one pipeline stage to the input (port interface side) of the write data path FIFOs.

Write Output Pipeline

The Write Output Pipeline adds one pipeline stage between the output of the write FIFO multiplexer and the memory data path IOBs. This setting is always enabled and cannot be user-disabled.

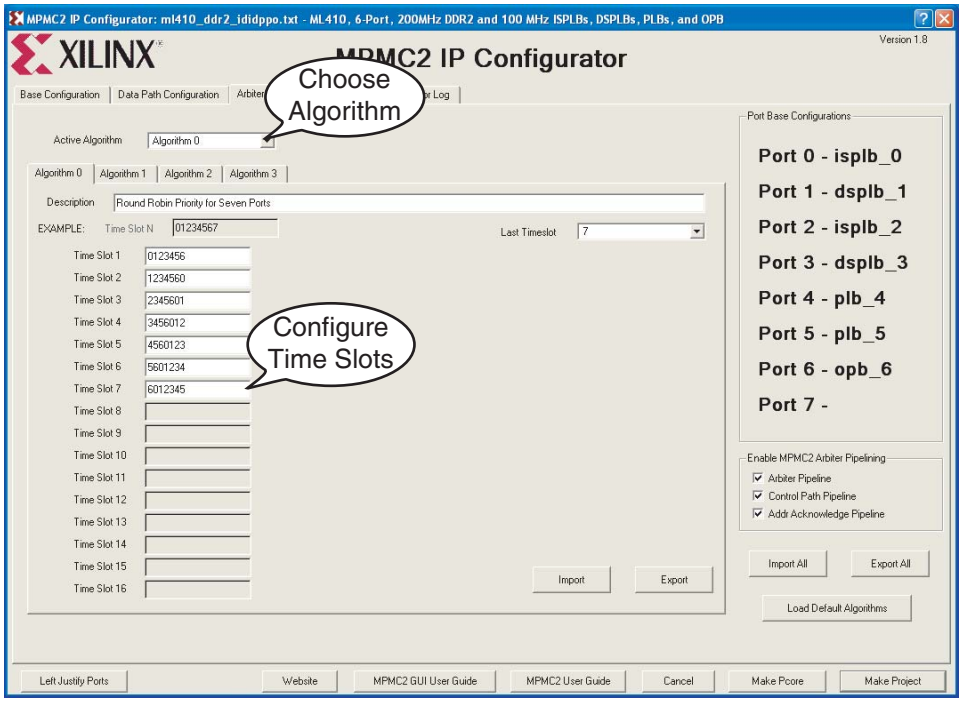
MPMC2 BRAM Management Report

The MPMC2 **BRAM Management Report** section shown in Figure 6, page 19 displays the MPMC2 block RAM usage for a particular device. This information is used to determine the percentage of BRAMs used in a particular device only.

The **Device** drop-down list, shown in Figure 4, page 11 (label *V*), allows you to select the device.

Arbiter Configuration Tab

Figure 7 displays the options available under the Arbiter Configuration tab. The MPMC2 control path arbiter allows for very complex arbitration. Each arbitration algorithm is stored in a BRAM. Several arbitration algorithms can be stored in the arbitration BRAM. The current implementation of MPMC2 allows for 16 algorithms, 4 of which are configurable via the Arbiter Configuration tab.



UG245_07_030907

Figure 7: **Configuring Arbiter**

Active Algorithm

Use the **Active Algorithm** drop-down list to specify the initial algorithm used in the MPMC2 Arbiter during operation.

Last Time Slot

Use the **Last Time Slot** drop-down list to specify how many time slots to use in a particular algorithm. Each time slot field must contain numbers corresponding to defined ports, and must contain the same number of digits as defined ports. For example, if ports 0, 1, and 2 are active, the following configuration would be valid:

```
012
120
201
```

If a configuration is incorrect, the displayed string turns red. If a port has not been referenced in an algorithm, a warning displays when you save and exit that the algorithm will skip over the missing port.

Port Base Configurations

The **Port Base** area displays the currently defined ports to help with algorithm configuration.

Enable MPMC2 Arbiter Pipelining

Arbiter Pipeline

Check the **Arbiter Pipeline** box to add one pipeline stage to the BRAM in the arbitration logic to improve timing.

Control Path Pipeline

Check **Control Path** box to add one pipeline stage in the control path (this improves timing significantly). This setting is always enabled and cannot be user-disabled.

Address Acknowledge Pipeline

Check the **Address Acknowledge** box to set the port interface address to acknowledge synchronous, rather than combinational communication.

Importing and Exporting

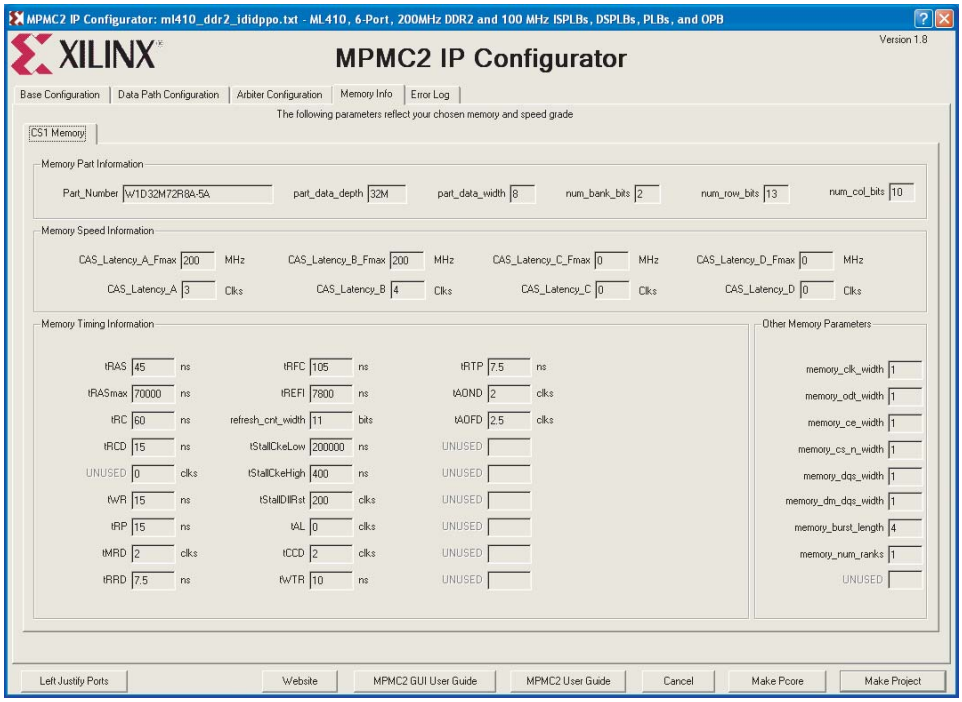
The MPMC2 IP Configurator allows the import and export of single arbiter algorithms and a set of four arbiter algorithms. To effect all four algorithms, use the **Import All** or **Export All** buttons.

Loading Default Algorithms

The Load Default Algorithm button detects how many ports are defined and loads the four default algorithms accordingly. Previous algorithm information is erased and replaced with Round Robin and Fixed Priority algorithms. Use the Load Defaults button to completely start over with a new, known working set of algorithms only. This is useful if you have changed the number of ports or you are working on a new design.

Memory Info Tab

The Memory Info tab shown in Figure 8 displays the parametric values from manufacturer data sheets, such as part number, size, speed, timing, and other memory device information. This information is compiled into a set of memory parameter files. The values for a specific device are loaded into the MPMC2 configuration when you choose a speed grade. (See "Base Configuration Tab," page 11.)



UG245_08_030907

Figure 8: **Memory Info**

Error Log Tab

The Error Log tab shown in Figure 9 captures error and warning information from the interface and its underlying tools. The error log should be reviewed after running **Make Pcore** or **Make Project** processes and can provide useful information for debugging purposes.

Note: The Error Log is a display in the interface and not a text file. To copy text from the error log using a PC, use **Ctrl-A** and **Ctrl-C** instead of highlighting the text using a mouse click or attempting to open `<mpmc2_install_directory>/logs/err.txt` in a text editor.

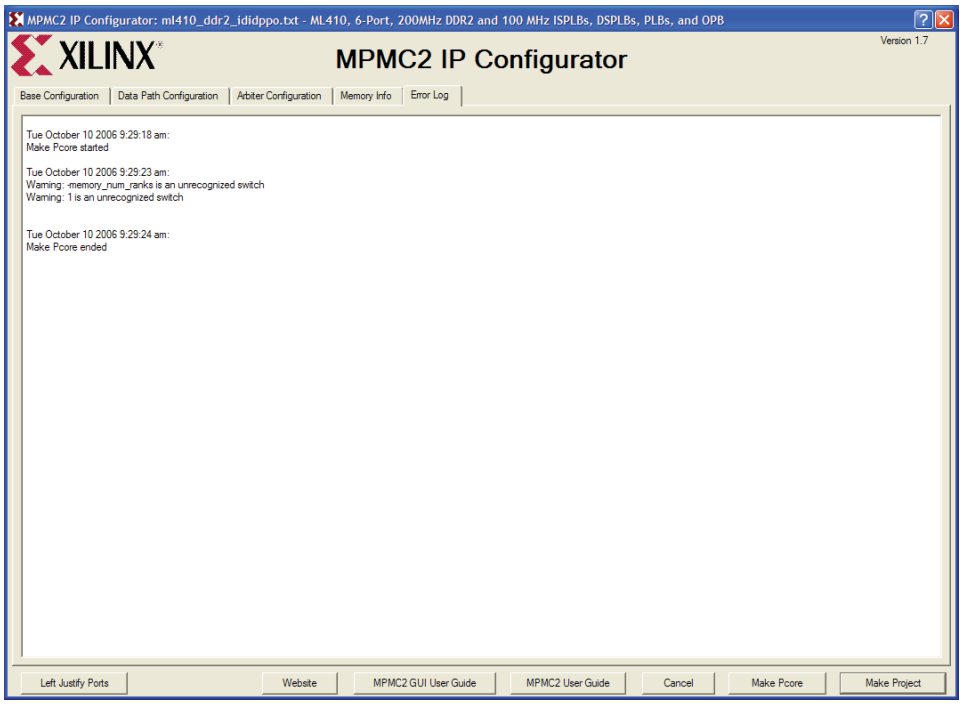


Figure 9: **Error Log**

MPMC2 Pcore Naming Convention

The MPMC2 IP Configurator interface uses a specific naming convention for the pcores it creates for XPS. This convention reduces or prevents name conflicts among various pcores. The convention is a set of fields separated by underscores, as shown in the following example:

```
mpmc2_ddr_idpocx_200mhz_x16_hyb25d512160be_5_0
```

where:

mpmc2_ = the memory controller type (MPMC2).

ddr_ = memory to which the MPMC2 interfaces.

idpocx_ = port types, in left to right order.

where:

i = ISPLB PIM

d = DSPLB PIM

p = PLB PIM

o = OPB PIM

c = CDMAC PIM

x = XCL PIM

n = Native Port Interface (NPI)

200mhz_ = clock frequency at which the memory is run.

Note: This frequency *MUST* be set properly as it sets up the memory control states.

x16_ = memory devic part width.

hyb25d512160be = the part number of the memory device.

5 = the speed grade of memory device.

0 = the instance number of the MPMC2 in MHS (only used for the PARAMETER INSTANCE field within the MHS file).

Note: Other parameters could be set differently, for example number of BRAMs per FIFO, *and those changes will not be reflected* in this naming convention.

Be very careful when building pcores with the same number (and order) of ports, same style of memory, same frequency of memory, and same memory device type as they will have the same name. Because you cannot rename the cores, you must edit and rename a number of different files. Use separate directories when you generate a pcore to prevent overwriting existing pcores with the same name.