MPMC2 Release Notes: 6 April, 2007 (Version 1.8)

MPMC2 is unlike most IP Cores that are in use today because it merges the concepts of memory controller, bridges, and system topologies. MPMC2 supports a wide variety of different port types and numbers of ports, therefore many different system topologies are possible. It is beyond the scope of this document to describe how to build systems with differing topologies. It is very important for the user to be aware of what kind of topology they wish to build, and to properly set parameters in the MHS file.

Please read the following release notes carefully. Please also read through the MPMC2 User Guide and GUI documentation carefully before implementing your own design. You should also check the Xilinx Support website Answer Browser (http://www.xilinx.com/xlnx/xil_ans_browser.jsp) to see if your question has already been answered.   If you can not find an answer to your question, please go to http://www.xilinx.com/support/clearexpress/websupport.htm to file a WebCase on MPMC2. Please also check the MPMC2 web site (http://www.xilinx.com/mpmc2) for information and updates.

Table of Contents:

Requirements

EDK 9.1SP1
ISE 9.1SP3
PC: Windows XP Service Pack 2
Linux: Redhat Enterprise Linux with kernel versions of 2.4 or later.

MPMC2 V1.8 Supported Features

- Monolithic MPMC2 with PIMs "inside"
- MPMC2 Graphical User Interface IP Configurator
- Parameterizable
- Port Interface Modules (PIM)
- User Configurable Port Arbitration
- Performance Monitors
- XPS Project Generation
- 200 MHz Memory and ISPLB / DSPLB in Virtex-4 -12 Devices
    - Only synchronous systems supported
- Supported Devices
    - Virtex-4 and Virtex-5 Family Support
    - Virtex2 Pro and Spartan3/3e/3a Family Support to be released in the future as a patch (see Answer Record #25052)

MPMC2 V1.8 Changes from V1.7

New Supported Features

- Added new Reference Designs (ML501, ML505, ML561, ECC designs)
- Virtex-5 Support
- Added ECC Support for Virtex-4 and Virtex-5
- Uses MIG based Physical Layer for Virtex-4. Virtex-5, and Spartan3. More information about MIG is available at: http://www.xilinx.com/products/design_resources/mem_corner/
- Improved timing on XCL PIM. Added 8 word cacheline read support to XCL PIM.
- Modifications to NPI Interface to support ECC and to better support memory coherency.
- C_MPMC2_0_INIT0_STALL no longer exists. To skip initialization wait time, set C_SKIP_INIT_DELAY to 1. Do not set this parameter when running in hardware.
- Improved IDELAY handling: Support for IDELAY chaining and added parameter for user to specify the number of IDELAYCTRL elements to instantiate inside MPMC2
- NPI interface updates: added signals to the NPI interface to indicate write fifo empty, initialization done, read fifo latency, and to control if read-modify-write is needed (For ECC only).
- Differential clock drivers instantiated inside MPMC2.
- For DDR2, parameter added to specify if differential or single ended DQS is used. If differential DQS is used, differential drivers are instantiated inside MPMC2 and DDR2 mode register is set accordingly.

Issues Resolved

- Better support for burst transactions in the OPB PIM. (Previous OPB PIM handled bursts as a series of single word transactions).
- Fixed problem with XCL PIMs and 64 bit DDR/DDR2 memories
- Fixed control path timing calculation error for DDR2 memories when memory clock is less than 133 MHz.
- C_PLB_0_PLB_NUM_MASTERS and C_PLB_0_PLB_MID_WIDTH are now automatically handled by XPS. Do not set NUM_MASTERS or MID_WIDTH in the MHS file (will cause errors if manually set).
- Clock and Reset connections are inferred from PIM bus connections.. Mem_clk, clk2x, cal_cal no longer needed
- PLB PIMs hold PLB busy signal high until writes complete to memory. Permits proper handling of memory coherency across PLB PIMs

Xilinx Platform Studio Projects Provided

This release includes the following XPS projects:

ml410_ddr2_ididpppp_200mhz_100mhz

- Target Platform: ML410P,
- Target Memory: DDR2 @ 64-bits, clocked at 200 MHz
- This design has two PPC405s and four PLB ports. Each bus clocks at 100 MHz
- Each PPC405 is clocked at 300 MHz, while ISPLB/DSPLB is clocked at 100 MHz
- Each of the four PLBs has a PLB_TFT controller to test bus mastership
- The first two PLBs have rev_b PLB_TFT which only does cache line 8 transactions
- The last two PLBs have rev_c PLB_TFT which only does burst 32 transactions

ml410_ddr2_ididppo_200mhz_100mhz

- Target Platform: ML410P,
- Target Memory: DDR2 @ 64-bits, clocked at 200 MHz
- This design has two PPC405s and two PLB ports. Each bus clocks at 100 MHz
- Each PPC405 is clocked at 300 MHz, while ISPLB/DSPLB is clocked at 100 MHz
- Each of the two PLBs has a PLB_TFT controller to test bus mastership
- The first PLB has two rev_b PLB_TFT which only does cache line 8 transactions
- The last PLB has two rev_c PLB_TFT which only does burst 32 transactions
- The OPB has all OPB peripherals.

ml410_ddr2_idido_200mhz_100mhz_ecc

- Target Platform: ML410P,
- Target Memory: DDR2 @ 32-bits + ECC support enabled , clocked at 200 MHz
- This design has two PPC405s. Each bus clocks at 100 MHz
- Each PPC405 is clocked at 300 MHz, while ISPLB/DSPLB is clocked at 100 MHz
- The OPB has all OPB peripherals.

ml410_ddr2_ididpp_200mhz_sl72p8m128m8m_a37fyu

- Target Platform: Hypothetical ML410 to demonstrate MPMC2 system set up to use ODT with support for a dual rank DIMM. Note: Actual ML410 board does not have support for dual rank DIMMs. This design uses non-connected FPGA pins for assigning the second rank's control signals.
- Target Memory: DDR2 @ 64-bits, clocked at 200 MHz
- MPMC2 core is configured to use ODT and be a dual rank DIMM.
- This design has two PPC405s and four PLB ports. Each bus clocks at 100 MHz
- Each PPC405 is clocked at 300 MHz, while ISPLB/DSPLB is clocked at 100 MHz
- Each of the two PLBs has a PLB_TFT controller to test bus mastership
- The first PLB has two rev_b PLB_TFT which only does cache line 8 transactions
- The second PLB has two rev_c PLB_TFT which only does burst 32 transactions

ml410_ddr_ididpppp_100mhz_pm

- Target Platform: ML410P,
- Target Memory: DDR @ 32-bits, clocked at 100 MHz
- This design has two PPC405s and four PLB ports. Each bus clocks at 100 MHz
- Each port has a performance monitor attached.
- Each PPC405 is clocked at 300 MHz, while ISPLB/DSPLB is clocked at 100 MHz
- Each of the four PLBs has a PLB_TFT controller to test bus mastership
- The first two PLBs have rev_b PLB_TFT which only does cache line 8 transactions
- The last two PLBs have rev_c PLB_TFT which only does burst 32 transactions
- DDR designs may not function on ML410 Rev B boards, ML410 Rev C or later boards are required

ml405_ddr_ddp_100m_srl_fifos

- Target Platform: ML405
- Target Memory: DDR @ 32-bits, clocked at 100 MHz
- This design has a PPC405 and a single PLB port.
- The PPC405 is clocked at 300 MHz, while its ISPLB/DSPLB are clocked at 100 MHz
- The PLBs has a PLB_TFT rev_c controller to test bus mastership, (Burst32)
- FIFOs are implemented with SRLs leaving more BRAMs available for the user design (with the tradeoff of higher LUT utilization).

ml405_ddr_ddp_200mhz_200mhz_200mhz_100mhz_speed12

- Target Platform: Hypothetical ML405 with -12 speed grade silicon
  - (Note: ML405 ships with -10 speed grade silicon. This design illustrates the clock rates possible with faster speed grades)
- Target Memory: DDR @ 32-bits, clocked at 200 MHz
- This design has a PPC405 and a single PLB port.
- The PPC405 is clocked at 400 MHz, while its ISPLB/DSPLB are clocked at 200 MHz
- The PLBs has a PLB_TFT rev_c controller to test bus mastership, (Burst32)

ml403_ddr_idpp_100mhz

- Target Platform: ML403,
- Target Memory: DDR @ 32-bits, clocked at 100 MHz
- This design has a PPC405 and two PLB ports.
- The PPC405 is clocked at 300 MHz, while its ISPLB/DSPLB are clocked at 100 MHz
- The first PLB has a PLB_TFT rev_c controller to test bus mastership, (Burst32)
- The second PLB has PLB2OPB bridge and all OPB peripherals

ml403_ddr_p_100mhz

- Target Platform: ML403,
- Target Memory: DDR @ 32-bits, clocked at 100 MHz
- This design has a single PPC405 and a single PLB port.
- The PPC405 is clocked at 300 MHz, while its ISPLB/DSPLB are clocked at 100 MHz
- The PLB has a PLB_TFT rev_c controller to test bus mastership, (Burst32)

ml403_ddr_idpoc_100mhz

- Target Platform: ML403,
- Target Memory: DDR @ 32-bits, clocked at 100 MHz
- This design has a single PPC405, the PPC405 is clocked at 300 MHz
- This design has one PLB port, one OPB port, and one CDMAC port, all clocked at 100MHz.
- The PLB has a PLB_TFT rev_c controller to test bus mastership (Burst 32)
- The OPB has all OPB peripherals.
- The CDMAC has a LL_TFT controller and 1 LL Data generator
- To swap between PLB_TFT and LL_TFT, use the W and E buttons on the ML403.

ml403_ddr_n_100mhz

- Target Platform: ML403,
- Target Memory: DDR @ 32-bits, clocked at 100 MHz
- This design has a single PPC405, the PPC405 is clocked at 300 MHz
- This design has one NPI port, which attaches to a PLB interface.

ml403_ddr_idpoc_100mhz_gsrd (aka "GSRD2")

- Target Platform: ML403,
- Target Memory: DDR @ 32-bits, clocked at 100 MHz
- This design has a single PPC405, the PPC405 is clocked at 300 MHz
- This design has one PLB port, one OPB port, and one CDMAC port, all clocked at 100MHz.
- The PLB is unused.
- The OPB has all OPB peripherals.
- The CDMAC has a LL_TEMAC.
- Please review README.TXT in the project directory for important information

ml402_ddr_xxo_100mhz

- Target Platform: ML402
- Target Memory: DDR @ 32-bits, clocked at 100 MHz
- This design has a single MicroBlaze, clocked at 100 MHz.
- This design has 2 XCL ports and one OPB port, all clocked at 100MHz.
- The OPB has all OPB peripherals.

ml410_ddr_xxo_100mhz

- Target Platform: ML410
- Target Memory: DDR @ 32-bits, clocked at 100 MHz
- This design has a single MicroBlaze, clocked at 100 MHz.
- This design has 2 XCL ports and one OPB port, all clocked at 100MHz
- The OPB has all OPB peripherals.
- DDR designs may not function on ML410 Rev B boards, ML410 Rev C or later boards are required.

ml501_ddr2_xxp_150mhz_75mhz

- Target Platform: ML501
- Target Memory: DDR2 @ 64-bits, clocked at 150 MHz
- This design has a single MicroBlaze, clocked at 75 MHz.
- This design has 2 XCL ports and one PLB port, all clocked at 75MHz

ml505_ddr2_xxp_150mhz_75mhz

- Target Platform: ML505
- Target Memory: DDR2 @ 64-bits, clocked at 150 MHz
- This design has a single MicroBlaze, clocked at 75 MHz.
- This design has 2 XCL ports and one PLB port, all clocked at 75MHz

ml505_ddr2_xxp_150mhz_75mhz_ecc

- Target Platform: ML505
- Target Memory: DDR2 @ 32-bits + ECC support enabled, clocked at 150 MHz
- This design has a single MicroBlaze, clocked at 75 MHz.
- This design has 2 XCL ports and one PLB port, all clocked at 75MHz

ml561_ddr_xxo_100mhz

- Target Platform: ML561
- Target Memory: DDR @ 32-bits, clocked at 100 MHz
- This design has a single MicroBlaze, clocked at 100 MHz.
- This design has 2 XCL ports and one OPB port, all clocked at 100MHz
- The OPB has all OPB peripherals.

v4fx12lc_ddr_idpp_100mhz

- Target Platform: V4FX12LC,
- Target Memory: DDR @ 16-bits, clocked at 100 MHz
- This design has a PPC405 and two PLB ports.
- The PPC405 is clocked at 300 MHz, while its ISPLB/DSPLB are clocked at 100 MHz
- The first PLB has a PLB2OPB bridge and all OPB peripherals.
- The second PLB is unused.

Known Issues / Frequently Encountered Issues:

GUI Usage Notes and Configuration Limitations

This section describes supported and unsupported MPMC2 configurations in the GUI in addition to usage information about the GUI. Violating any GUI configuration requirements may produce a non-functional MPMC2 design. Use caution if your desired MPMC2 configuration violates any of the following requirements.

1. If using SRL FIFO configuration, it is strongly recommended that all pipeline stages be enabled in Data Path. This is required for 64-bit memories when using SRL FIFOs.
2. For new designs, system bring-up, and debugging, it is strongly recommended that the user enable all pipeline stages in the Data Path and Arbiter. This configuration offers the best timing and is more robust. Advanced users can explore latency/area tradeoffs by removing pipeline stages but should retest their systems. Pipeline settings other than what is used on reference designs may not be tested.
3. In the Arbiter Configuration tab, only Algorithm 0 is supported. Values set for Algorithm 1, 2, or 3 are not used by the MPMC2 core. The Active Algorithm will always be algorithm 0 regardless of which algorithm is chosen in the MPMC2 GUI.
4. Using a memory clock > 200 MHz or a PLB/OPB/CDMAC PIM clock > 100 MHz is untested. The GUI will provide a warning for every PIM /memory that has a higher clock frequency, but the user is still permitted to create the design.
5. MPMC2 GUI should be locally installed and run from the install directory. Multi-user installation is not supported.
6. Only two DSPLB PIMs are allowed to be instantiated currently. While the MPMC2 GUI allows for more that two, the code will not compile through XST.
7. Users should not add extraneous/unrelated files to the sub directories used by the GUI. This may result in improper operation of the GUI. **Warning: Do not modify or move ANY files in the MPMC2 install directory!**
8. Please set the Device field correctly in the GUI so that MPMC is properly configured for the device architecture being used
9. The GUI does not correctly write address information to the control.txt file/MPD file. This should not affect designs, since the BASE/HIGH addresses are normally set in the system.mhs file which overrides these values.
10. On the Arbiter Configuration tab, “Control Path Pipeline” must be enabled.
11. On the Data Path Configuration tab, the “Memory Side” checkbox in the “Write Output Pipeline” box must be enabled.
12. On the Data Path Configuration tab, the “Memory Side” and ”Port Side” check boxes in the “Read Pipeline” and “Write Pipeline” boxes have only been tested with all values enabled, or all values disabled.

MPMC2 Design Notes and Parameter Value Requirements

1. The MPMC2 does not currently support asynchronous clocking on any of its ports. The ports must be an even integer harmonic of the clock rate to memory.
2. C_<PIM_Name>_<PortNum>_ADDR_WIDTH must be set to 32. 36 bit addressing is not supported
3. DDR and DDR2 SDRAM are currently the only supported memory types.
4. C_RD_DATAPATH_TML_PIPELINEx parameters have only been tested with values of 0x00.
5. C_RD_DATAPATH_TML_PIPELINE cannot be set directly in the MPMC2 GUI, it can only currently be set in the refcore files (data/refcores/*.txt). Only the value 0x00 has been tested with replication parameters.
6. When using a 64-bit memory, MPMC2_PIM_<Port_Num>_WrFIFO_Push must be asserted after MPMC2_PIM_<Port_Num>_AddrAck.

7. When using a 64-bit memory, all MPMC2_PIM_<Port_Num>_WrFIFO_Push's for prior transfers must complete before asserting a word write request. (MPMC2_PIM_<Port_Num>_AddrReq while MPMC2_PIM_<Port_Num>_RNW equals 0 and MPMC2_PIM_<Port_Num>_Size equals 0). The PIMs shipped with MPMC2 all conform to this specification, all user created NPI PIMs must also conform.
8. Write FIFO Reset is not currently supported.
9. When using the NPI, please ensure that all write data is in the write data path FIFOs before the memory requires it. This is particularly important to pay attention to when using 64-bit memories or when the custom NPI PIM has a slower clock than MPMC2_0_Clk0. For safest operation assert MPMC2_PIM_<Port_Num>_AddrReq after all MPMC2_PIM_<Port_Num>_WrFIFO_Push's.
10. Serial Presence Detect ROMs are not currently supported.
11. Per Port Address Offset and Per Port Address Aliasing features are untested and unsupported.
12. When using DSPLB/ISPLB PIM, PowerPC read/write access to an address not covered by memory or bridges to PLB/OPB PIMs will result in a lock-up on that interface where no other transactions can be completed on that interface. If access to unmapped address space cannot be avoided, size the address range of memory/PLB/OPB space to cover all memory space. Alternatively one, can use PLB PIMs to connect processor to memory, but this will result in reduced performance.
13. 200 MHz or higher memory clock rates typically require -11 or -12 (preferable) speed grade Virtex-4 FPGAs or -3 speed grade Virtex-5 parts. For systems with higher numbers of ports, -12 speed grade should be used with Virtex-4.
14. If CDMAC running 1:1 clock ratio to memory, then memory clock Fmax is around 100 MHz. If CDMAC is running 1:2 clock ratio to memory, then memory clock Fmax is in the range of 130 to 150 MHz.
15. On some refresh operations, precharge may be asserted for two cycles.
16. The PLB PIM does not support indeterminate length burst. Some IP such as plb_pci may utilize indeterminate length bursts and are therefore not compatible with the PLB PIM.
17. In multi-rank/multi-DIMM systems, the MPMC2's MIG PHY interface will only calibrate its IDELAY timing to one of the ranks on one of the DIMMs. Differences in timing between ranks and DIMMs will reduce timing margin and can affect the frequency range of operation.
18. Users are strongly recommended that the MPMC2 GUI be used to alter number of ports, PIM type, style of FIFO interface, etc. It is also strongly recommended that ***only*** the user parameters listed within the MPMC2 user guide be varied without using the MPMC2 GUI. There are many non-obvious parameter value dependencies within MPMC2. The MPMC2 GUI has specialized knowledge of how to set those dependencies.
19. Modelsim with mixed VHDL/verilog language support should be used for simulation. Other simulators have not be tested.
20. Setting ECC Write Enable Flag (Bit 31 of ECCCR) to 0 is not supported. This bit must be set to 1 for memory writes to work correctly.

EDK/ISE Tool Notes and Reference Design Notes

1. Before generating a bit stream for each included project, the user <u>must</u> set the CONFIG STEPPING constraint in the UCF file. This constraint specifies the silicon stepping level of the specific device on the user's board. Please see the project's UCF file for more details.
2. All Pcores are individually made for a specific purpose. Use the MPMC2 GUI if you make any substantive changes to the core. It is not recommended for the user to try to reconfigure the MPMC2 core manually.
3. If performance monitors are instantiated and MPMC2 memory is set to operate at 200MHz the MPMC2 core may fail to meet timing
4. If the xxxx.pao file is blank in a generated pcore, it is likely that cygwin tools are not installed properly. Be sure that the Cygwin bin directory is in the computer's path, and be sure that multiple versions of Cygwin are not conflicting.
5. All designs have been hardware and software tested on their respective boards.

6. DCR addresses for CDMAC and performance monitors in project generation are not set correctly. You must modify the system.mhs and set this manually. Otherwise xps will return an error that the addresses are not set correctly.
7. When trying to read/write to the OPB2DCR bridge to access DCR registers via XMD, you may get an error that says that the address space is not accessible via the processor. To work around this, exit out of XMD and reconnect to XMD using the following command line: "xmd -opt ect/<opt_file>.opt". Note that if you are also using OCM to run code, not having the "-xmp system.xmp" command line option as part of your XMD command, you may not be able to download code to OCM.

Tips and Hints for Debug and Bring-Up of MPMC2 Designs

This section describes ideas for debugging and bringing up an MPMC2 system. It is recommended that new users review and experiment with these suggestions. When creating a webcase, please try to include information about which of these suggestions/tips you have implemented and what the results are. When filing a webcase for support, provide as much information as possible.

1. It is recommended users have some experience with EDK, ISE, and FPGA design before attempting an MPMC2 design. EDK provides tutorials and simple design examples for new users. Xilinx also offers training classes on EDK to quickly start new users with the EDK tools.
2. It is recommended that new users start from one of the included reference design that most closely resembles the system they are trying to build and use the reference design as a starting point to model.
3. If possible, new users should begin their exploration of MPMC2 using a development board that is supported by a reference design. Many of these developments boards (i.e. ML403, ML405, etc) are available for purchase from Xilinx or our partners.
4. Make sure your design has memory at the boot vector of the processor (Address 0xFFFFFFFC on PPC and 0x00000000 on MicroBlaze) so that the processor starts up and has valid code to run out of reset. The boot code can be as simple as a branch to itself instruction ("bootloop"). After the processor boots and is stable, user can connect with XMD to write/read memory or load larger programs. User should not allow processor to have undefined memory at the boot vector or else erratic behavior may result.
5. Make sure your design meets timing and that proper timing constraints are in effect when running a design through the tools.
6. When bringing up MPMC2 on a new board, start with as simple of a design as possible and establish a working connection to memory before adding more ports.
7. If experiencing problems, try running at slower end of clock frequency range supported by your memory. This provides more timing margin for initial testing before trying higher frequencies
8. Use Chipscope to look at transactions across the PIMs or inside MPMC2 to help isolate problems
9. Try to run simulations of your design.
10. Note: During memory initialization (following reset), the memory physical interface writes to the bottom locations in memory to set up the write training pattern. This can affect addresses between 0x00000000 and 0x000000FF. Therefore, after reset, these locations in memory will be overwritten. It is recommended that user SW code stored in memory not start at address 0x0000000 of the memory. SW code should start at an address that is 0x100 offset into memory and the boot vector set to jump to this offset address. For example, a PPC405 system would be set to boot at 0xFFFFFFFC and then jump to 0x00000100 in memory to start executing code out of memory. As an example, look at the SW compile options in the MPMC2 reference designs.
11. When using ECC in debug mode, hardware considerations need to be taken into account when forcing errors. As a simple example, let us take a design with a memory that has 32 data bits and 4 ECC bits. In this case, every clock cycle, 64-bits of data will be written to memory. If you force a single bit error, the error will show up somewhere in the 64-bit range. If you did a 32-bit write to address 0x0 in your software, the error could either show up at address 0x0 or at 0x4. Additionally, since the burst length to memory is 4 data beats (4x32-bits), and since ECC requires read-modify-write, you will actually write one data error at either address 0x0 or 0x4, and one

error at either 0x8 or 0xC. Now, if you read data from 0x0, your ECC status errors will report one error. However, if you write to address 0x0, you are going to do a read modify write, so you will read 4x32-bits of data, then modify address 0x0 and write it back. This will report two errors rather than one since you also read addresses 0x8 and 0xC.

Dynamic Arbitration

A user can implement Dynamic Arbitration by bringing the arbitration signals from the MPMC2 control path (mpmc2_ctrl_path.v) to the top level MPMC2 file (mpmc2.v) and creating a DCR interface to communicate with the arbiter interface. Note that mpmc2.v is a generated file from the MPMC2 GUI, so this could easily break the pcore or be overwritten by the GUI. Without modifying the core the design will only support arbitration Algorithm 0 as set in the GUI. Up to 16 arbitration algorithms can be stored in the arbitration BRAM. Software could determine that a different algorithm is more appropriate. For example, if the software knows that the processor needs a lot of bandwidth and that direct memory access (DMA) needs very little bandwidth, it can change the algorithm. Then when more DMA bandwidth is needed, the software can modify the algorithm number and select a more appropriate algorithm.

Answer Records pertaining to this release:

Xilinx Answer Record	Short Description
23593	MPMC2 GUI - Error - Sed is not installed, or it is not in your PATH
24912	Creating NPI PCORE
25052	Virtex2 Pro and Spartan3/3e/3a Family Support to be released in the future as a patch
25307	How to Migrate MPMC v1.7 designs to MPMC v1.8