SmartFusion2 MSS

DDR Controller Configuration

Libero SoC v11.6 and later

Table of Contents

Introduction

The SmartFusion2 MSS has an embedded DDR controller. This DDR controller is intended to control an off-chip DDR memory. The MDDR controller can be accessed from the MSS as well as from the FPGA fabric. In addition, the DDR controller can also be bypassed, providing an additional interface to the FPGA fabric (Soft Controller Mode (SMC)).

To fully configure the MSS DDR controller, you must:

1. Select the datapath using the MDDR Configurator.
2. Set the register values for the DDR controller registers.
3. Select the DDR memory clock frequencies and FPGA fabric to MDDR clock ratio (if needed) using the MSS CCC Configurator.
4. Connect the controller's APB configuration interface as defined by the Peripheral Initialization solution. For the MDDR Initialization circuitry built by System Builder, refer to the "MSS DDR Configuration Path" on page 13 and Figure 2-7.

   You can also build your own initialization circuitry using standalone (not by System Builder) Peripheral Initialization. Refer to the SmartFusion2 Standalone Peripheral Initialization User Guide.

1 – MDDR Configurator

The MDDR Configurator is used to configure the overall datapath and the external DDR Memory Parameters for the MSS DDR controller.

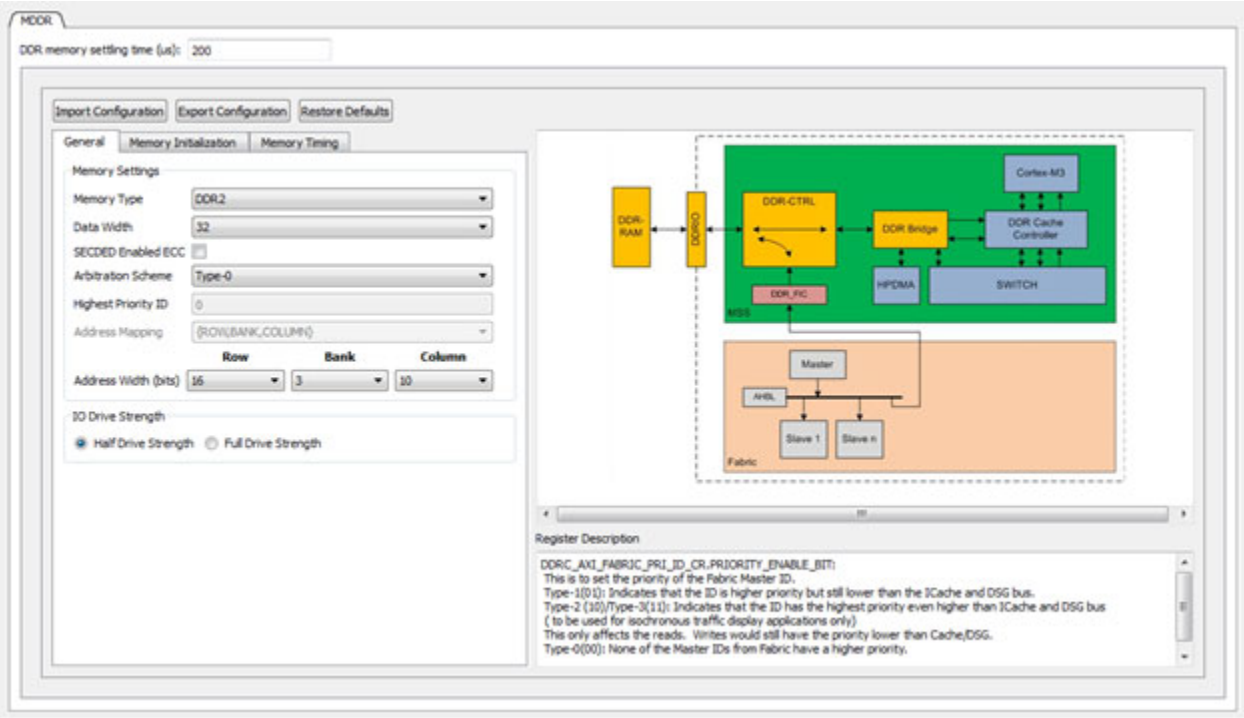

***Figure 1-1* • MDDR Configurator Overview**

The General tab sets your Memory and Fabric Interface settings (Figure 1-1).

Memory Settings

Enter the DDR Memory Settling Time. This is the time the DDR memory requires to initialize. The default value is 200 us. Refer to your DDR Memory Data Sheet for the correct value to enter.

Use Memory Settings to configure your memory options in the MDDR.

- **Memory Type** - LPDDR, DDR2, or DDR3
- **Data Width** - 32-bit, 16-bit or 8-bit
- **SECDED Enabled ECC** - ON or OFF
- **Arbitration Scheme** - Type-0, Type -1, Type-2,Type-3
- **Highest Priority ID** - Valid values are from 0 through 15
- **Address Width (bits)** - Refer to your DDR Memory Data Sheet for the number of row, bank, and column address bits for the LPDDR/DDR2/DDR3 memory you use. select the pull-down menu to choose the correct value for rows/banks/columns as per the data sheet of the LPDDR/DDR2/DDR3 memory.

Note: The number in the pull-down list refers to the number of Address bits, not the absolute number of rows/banks/columns. For example, if your DDR memory has 4 banks, select 2 (2^2=4) for banks. If your DDR memory has 8 banks, select 3 (2^3=8) for banks.

Fabric Interface Settings

By default, the hard Cortex-M3 processor is set up to access the DDR Controller. You can also allow a fabric Master to access the DDR Controller by enabling the Fabric Interface Setting checkbox. In this case, you can choose one of the following options:

- **Use an AXI Interface** - The fabric Master accesses the DDR Controller through a 64-bit AXI interface.
- **Use a Single AHBLite Interface** - The fabric Master accesses the DDR Controller through a single 32-bit AHB interface.
- **Use two AHBLite Interfaces** - Two fabric Masters access the DDR Controller using two 32-bit AHB interfaces.

The configuration view (Figure 1-1) updates according to your Fabric Interface selection.

I/O Drive Strength (DDR2 and DDR3 only)

Select one of the following drive strengths for your DDR I/Os:

- Half Drive Strength
- Full Drive Strength

Libero SoC sets the DDR I/O Standard for your MDDR system based on your DDR Memory type and I/O Drive Strength (as shown in Table 1-1).

***Table 1-1* • I/O Drive Strength and DDR Memory Type**

DDR Memory Type	Half Strength Drive	Full Strength Drive
DDR3	SSTL15I	SSTL15II
DDR2	SSTL18I	SSTL18II
LPDDR	LPDRI	LPDRII

IO Standard (LPDDR only)

Select one of the following options:

- LVCMOS18 (Lowest Power) for LVCMOS 1.8V IO standard. Used in typical LPDDR1 applications.
- LPDDRI Note: Before you choose this standard, make sure that your board supports this standard. You must use this option when targeting the M2S-EVAL-KIT or the SF2-STARTER-KIT boards. LPDDRI IO standards require that a IMP_CALIB resistor is installed on the board.

IO Calibration (LPDDR only)

Choose one of the following options when using LVCMOS18 IO standard:

- On
- Off (Typical)

Calibration ON and OFF optionally controls the use of an IO calibration block that calibrates the IO drivers to an external resistor. When OFF, the device uses a preset IO driver adjustment.

When ON, this requires a 150-ohm IMP_CALIB resistor to be installed on the PCB.

This is used to calibrate the IO to the PCB characteristics. However, when set to ON, a resistor needs to be installed or the memory controller will not initialize.

For more information, refer to AC393-SmartFusion2 and IGLOO2 Board Design Guidelines Application Note and the SmartFusion2 SoC FPGA High Speed DDR Interfaces User Guide.

2 – MDDR Controller Configuration

When you use the MSS DDR Controller to access an external DDR Memory, the DDR Controller must be configured at runtime. This is done by writing configuration data to dedicated DDR controller configuration registers. This configuration data is dependent on the characteristics of the external DDR memory and your application. This section describes how to enter these configuration parameters in the MSS DDR controller configurator and how the configuration data is managed as part of the overall Peripheral Initialization solution.

MSS DDR Control Registers

The MSS DDR Controller has a set of registers that need to be configured at runtime. The configuration values for these registers represent different parameters, such as DDR mode, PHY width, burst mode, and ECC. For complete details about the DDR controller configuration registers, refer to the SmartFusion2 SoC FPGA High Speed DDR Interfaces User's Guide.

MDDR Registers Configuration

Use the Memory Initialization (Figure 2-1, Figure 2-2, and Figure 2-3) and Memory Timing (Figure 2-4) tabs to enter parameters that correspond to your DDR Memory and application. Values you enter in these tabs are automatically translated to the appropriate register values. When you click a specific parameter, its corresponding register is described in the Register Description pane (lower portion in Figure 1-1 on page 4).

Memory Initialization

The Memory Initialization tab allows you to configure the ways you want your LPDDR/DDR2/DDR3 memories initialized. The menu and options available in the Memory Initialization tab vary with the type of DDR memory (LPDDR/DDR2/DDR3) you use.

Refer to your DDR Memory Data Sheet when you configure the options.

When you change or enter a value, the Register Description pane gives you the register name and register value that is updated. Invalid values are flagged as warnings.

Figure 2-1, Figure 2-2, and Figure 2-3 show the Initialization tab for LPDDR, DDR2 and DDR3, respectively.

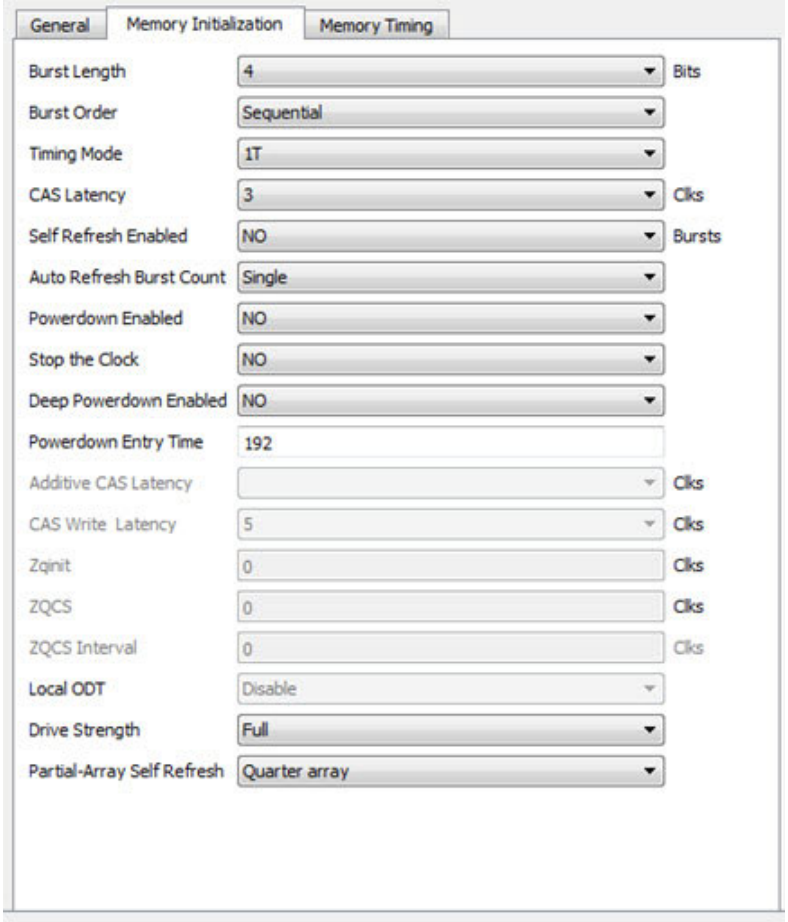

***Figure 2-1* • MDDR Configuration—Memory Initialization Parameters (LPDDR)**

- **Timing Mode** - Select 1T or 2T Timing mode. In 1T (the default mode), the DDR controller can issue a new command on every clock cycle. In 2T timing mode, the DDR controller holds the address and command bus valid for two clock cycles. This reduces the efficiency of the bus to one command per two clocks, but it doubles the amount of setup and hold time.
- **Partial-Array Self Refresh (LPDDR only)**. This feature is for power saving for the LPDDR. Select one of the following for the controller to refresh the amount of memory during a self refresh:
  - Full array: Banks 0, 1,2, and 3
  - Half array: Banks 0 and 1
  - Quarter array: Bank 0
  - One-eighth array: Bank 0 with row address MSB=0
  - One-sixteenth array: Bank 0 with row address MSB and MSB-1 both equal to 0.

For all other options, refer to your DDR Memory Data Sheet when you configure the options.

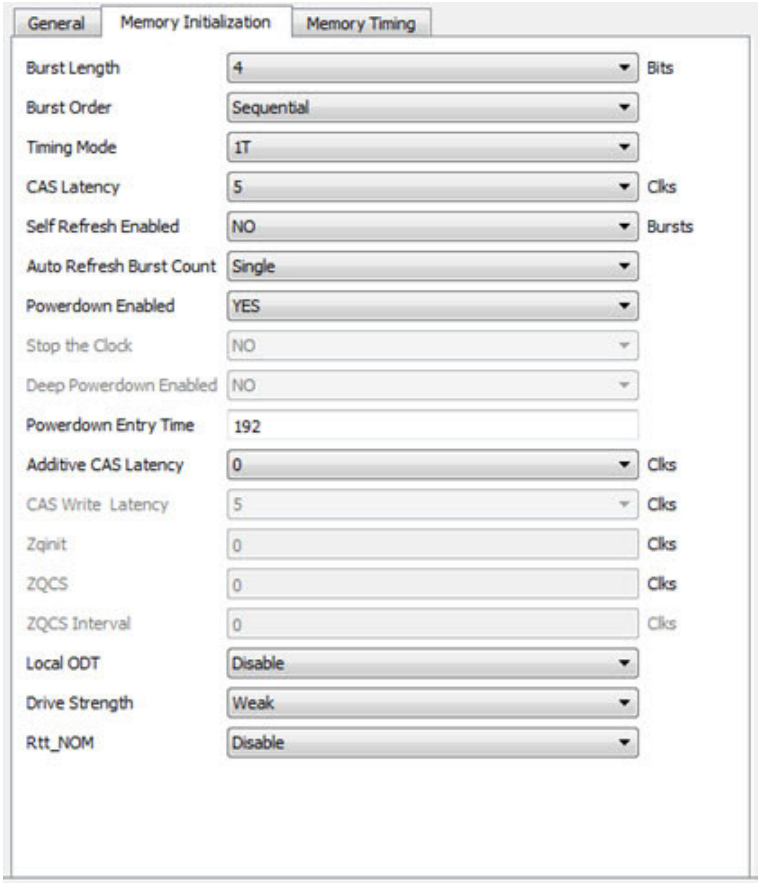

Figure 2-2 **• MDDR Configuration—Memory Initialization Parameters (DDR2)**

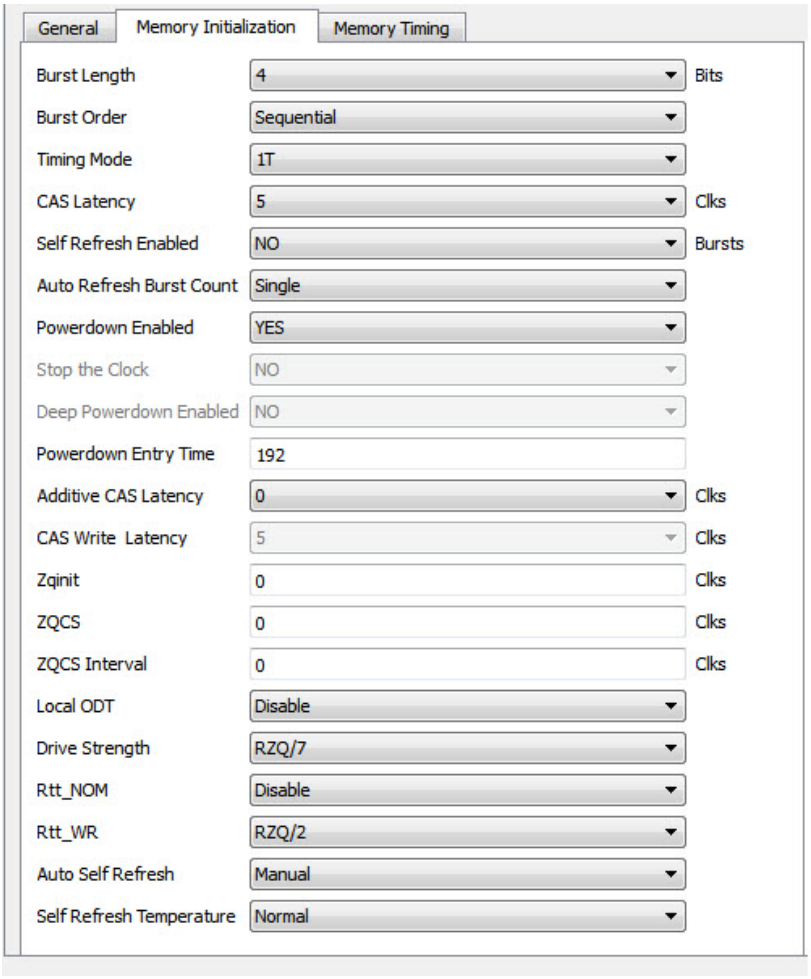

Figure 2-3 **• MDDR Configuration—Memory Initialization Paramet ers (DDR3)**

Memory Timing

This tab allows you to configure the Memory Timing parameters. Refer to the Data Sheet of your LPDDR/DDR2/DDR3 memory when configuring the Memory Timing parameters.

When you change or enter a value, the Register Description pane gives you the register name and register value that is updated. Invalid values are flagged as warnings.

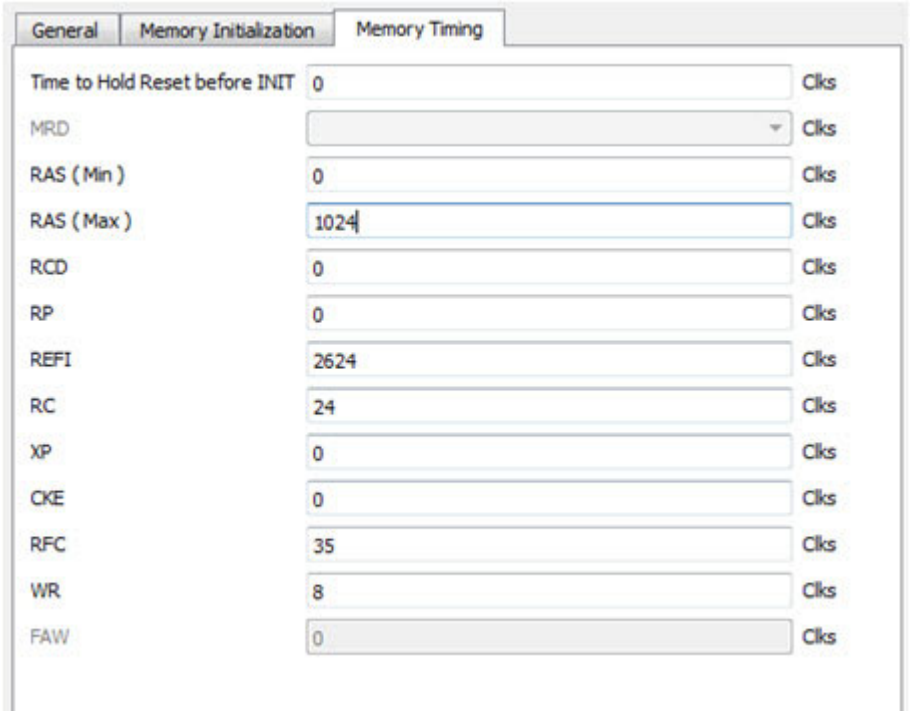

***Figure 2-4* • MDDR Configuration Memory Timing Tab**

Importing DDR Configuration Files

In addition to entering DDR Memory parameters using the Memory Initialization and Timing tabs, you can import DDR register values from a file. To do so, click the **Import Configuration** button and navigate to the text file containing DDR register names and values. Figure 2-5 shows the import configuration syntax.

```
ddrc_dyn_soft_reset_CR          0x00 ;
ddrc_dyn_refresh_1_CR           0x27DE ;
ddrc_dyn_refresh_2_CR           0x030F ;
ddrc_dyn_powerdown_CR           0x02 ;
ddrc_dyn_debug_CR               0x00 ;
ddrc_ecc_data_mask_CR           0x0000 ;
ddrc_addr_map_col_1_CR          0x3333 ;
ddrc_addr_map_col_3_CR          0x3300 ;
ddrc_init_1_CR                  0x0001 ;
ddrc_cke_rstn_cycles_CR1        0x0100 ;
ddrc_cke_rstn_cycles_CR2        0x0008 ;
ddrc_init_emr2_CR               0x0000 ;
ddrc_init_emr3_CR               0x0000 ;
ddrc_dram_bank_act_timing_CR    0x1947;
```

***Figure 2-5* • DDR Register Configuration File Syntax**

Note: If you choose to import register values rather than entering them using the GUI, you must specify all necessary register values. Refer to the SmartFusion2 SoC FPGA High Speed DDR Interfaces User's Guide for details.

Exporting DDR Configuration Files

You can also export the current register configuration data into a text file. This file will contain register values that you imported (if any) as well as those that were computed from GUI parameters you entered in this dialog.

If you want to undo changes you have made to the DDR register configuration, you can do so with Restore Default. Note that this deletes all register configuration data and you must either re-import or reenter this data. The data is reset to the hardware reset values.

Generated Data

Click **OK** to generate the configuration. Based on your input in the General, Memory Timing and Memory Initialization tabs, the MDDR Configurator computes values for all DDR configuration registers and exports these values into your firmware project and simulation files. The exported file syntax is shown in Figure 2-6.

```
# Exported: 2013-Sep-02 05:07:16
# Libero DDR Configurator GUI Version = 2.0
# DDR Controller Type = DDR2
# Bus Width = 32-bits
# Memory Bandwidth = 200 Mbps
# Total Bandwidth = 6400 Mbps
#
# Validation Status:
# Target Device Manufacturer:
# Target Device:
#
# User Comments:
#
#
DDRC_ADDR_MAP_BANK_CR.REG_DDRC_ADDRMAP_BANK_B2                    0xa
DDRC_ADDR_MAP_BANK_CR.REG_DDRC_ADDRMAP_BANK_B1                    0xa
DDRC_ADDR_MAP_BANK_CR.REG_DDRC_ADDRMAP_BANK_B0                    0xa
DDRC_ADDR_MAP_COL_1_CR.REG_DDRC_ADDRMAP_COL_B7                    0x3
DDRC_ADDR_MAP_COL_1_CR.REG_DDRC_ADDRMAP_COL_B4                    0x3
DDRC_ADDR_MAP_COL_1_CR.REG_DDRC_ADDRMAP_COL_B3                    0x3
DDRC_ADDR_MAP_COL_1_CR.REG_DDRC_ADDRMAP_COL_B2                    0x3
DDRC_ADDR_MAP_COL_2_CR.REG_DDRC_ADDRMAP_COL_B11                   0xf
DDRC_ADDR_MAP_COL_2_CR.REG_DDRC_ADDRMAP_COL_B10                   0xf
DDRC_ADDR_MAP_COL_2_CR.REG_DDRC_ADDRMAP_COL_B9                    0xf
DDRC_ADDR_MAP_COL_2_CR.REG_DDRC_ADDRMAP_COL_B8                    0x3
DDRC_ADDR_MAP_COL_3_CR.REG_DDRC_ADDRMAP_COL_B6                    0x3
DDRC_ADDR_MAP_COL_3_CR.REG_DDRC_ADDRMAP_COL_B5                    0x3
```

Figure 2-6 • **Exported DDR Register Configuration File Syntax**

Firmware

When you generate the SmartDesign, the following files are generated in the <project dir>/firmware/drivers_config/sys_config directory. These files are required for the CMSIS firmware core to compile properly and contain information regarding your current design including peripheral configuration data and clock configuration information for the MSS. Do not edit these files manually as they are re-created every time your root design is re-generated.

- sys_config.c
- sys_config.h
- sys_config_mddr_define.h - MDDR configuration data.
- Sys_config_fddr_define.h - FDDR configuration data.
- sys_config_mss_clocks.h - MSS clocks configuration

Simulation

When you generate the SmartDesign associated with your MSS, the following simulation files are generated in the <project dir>/simulation directory:

- **test.bfm** - Top-level BFM file that is first "executed" during any simulation that exercises the SmartFusion2 MSS' Cortex-M3 processor. It executes peripheral_init.bfm and user.bfm, in that order.
- **peripheral_init.bfm** - Contains the BFM procedure that emulates the CMSIS::SystemInit() function run on the Cortex-M3 before you enter the main() procedure. It essentially copies the configuration data for any peripheral used in the design to the correct peripheral configuration registers and then waits for all the peripherals to be ready before asserting that the user can use these peripherals.
- **MDDR_init.bfm** - Contains BFM write commands that simulate writes of the MSS DDR configuration register data you entered (using the Edit Registers dialog above) into the DDR Controller registers.
- **user.bfm** - Intended for user commands. You can simulate the datapath by adding your own BFM commands in this file. Commands in this file will be "executed" after peripheral_init.bfm has completed.

Using the files above, the configuration path is simulated automatically. You only need to edit the user.bfm file to simulate the datapath. Do not edit the test.bfm, peripheral_init.bfm, or MDDR_init.bfm files as these files are re-created every time your root design is re-generated.

MSS DDR Configuration Path

The Peripheral Initialization solution requires that, in addition to specifying MSS DDR configuration register values, you configure the APB configuration data path in the MSS (FIC_2). The SystemInit() function writes the data to the MDDR configuration registers via the FIC_2 APB interface.

Note: If you are using System Builder the configuration path is set and connected automatically.

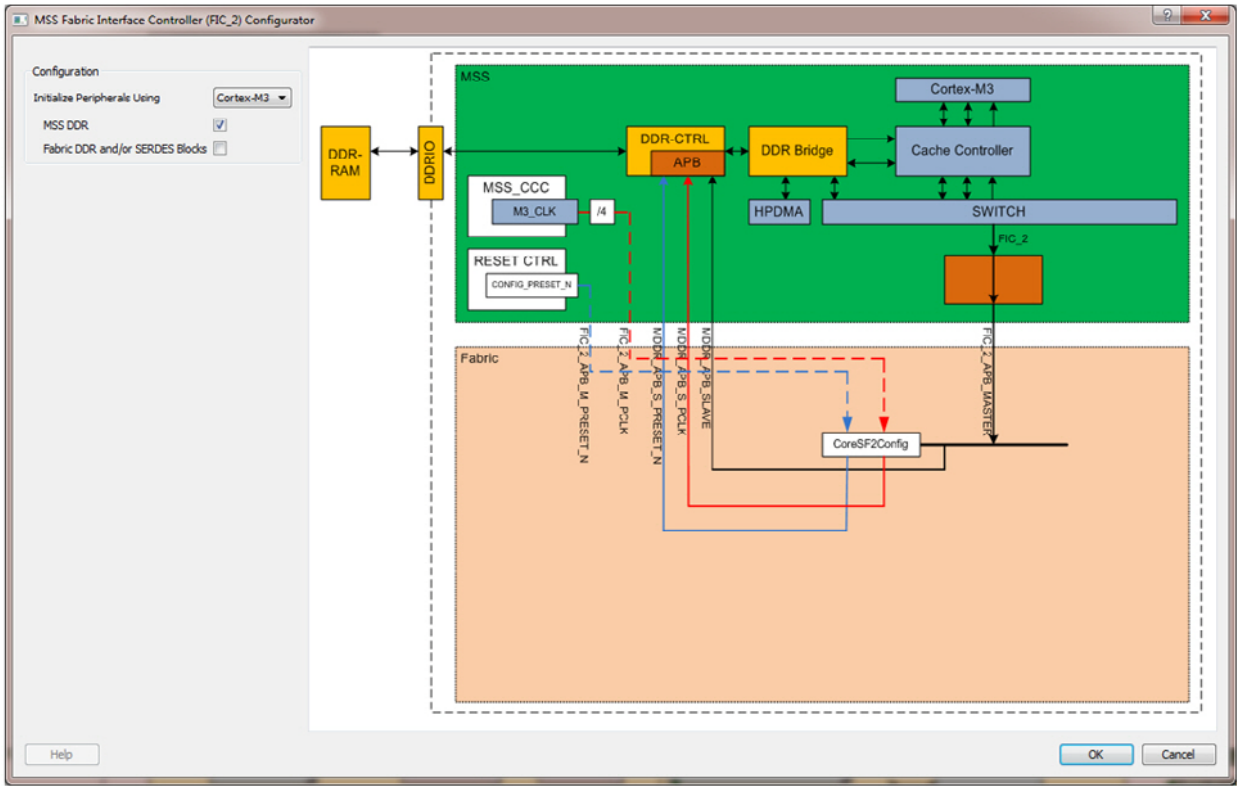

***Figure 2-7* • FIC_2 Configurator Overview**

To configure the FIC_2 interface:

1. Open the FIC_2 configurator dialog (Figure 2-7) from the MSS configurator.
2. Select the **Initialize peripherals using Cortex-M3** option.
3. Make sure that the MSS DDR is checked, as are the Fabric DDR/SERDES blocks if you are using them.
4. Click **OK** to save your settings. This will expose the FIC_2 configuration ports (Clock, Reset, and APB bus interfaces), as shown in Figure 2-8.
5. Generate the MSS. The FIC_2 ports (FIC_2_APB_MASTER, FIC_2_APB_M_PCLK and FIC_2_APB_M_RESET_N) are now exposed at the MSS interface and can be connected to the CoreConfigP and CoreResetP as per the Peripheral Initialization solution specification.

For complete details on configuring and connecting the CoreConfigP and CoreResetP cores, refer to the Peripheral Initialization User Guide.

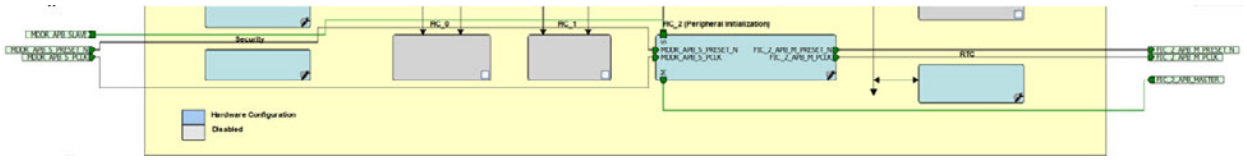

Figure 2-8 • **FIC_2 Ports**

3 – Port Description

DDR PHY Interface

Table 3-1 **• DDR PHY Interface**

Port Name	Direction	Description
MDDR_CAS_N	OUT	DRAM CASN
MDDR_CKE	OUT	DRAM CKE
MDDR_CLK	OUT	Clock, P side
MDDR_CLK_N	OUT	Clock, N side
MDDR_CS_N	OUT	DRAM CSN
MDDR_ODT	OUT	DRAM ODT
MDDR_RAS_N	OUT	DRAM RASN
MDDR_RESET_N	OUT	DRAM Reset for DDR3. Ignore this signal for LPDDR and DDR2 Interfaces. Mark it unused for LPDDR and DDR2 Interfaces.
MDDR_WE_N	OUT	DRAM WEN
MDDR_ADDR[15:0]	OUT	Dram Address bits
MDDR_BA[2:0]	OUT	Dram Bank Address
MDDR_DM_RDQS ([3:0]/[1:0]/[0])	INOUT	Dram Data Mask
MDDR_DQS ([3:0]/[1:0]/[0])	INOUT	Dram Data Strobe Input/Output - P Side
MDDR_DQS_N ([3:0]/[1:0]/[0])	INOUT	Dram Data Strobe Input/Output - N Side
MDDR_DQ ([31:0]/[15:0]/[7:0])	INOUT	DRAM Data Input/Output
MDDR_DQS_TMATCH_0_IN	IN	FIFO in signal
MDDR_DQS_TMATCH_0_OUT	OUT	FIFO out signal
MDDR_DQS_TMATCH_1_IN	IN	FIFO in signal (32-bit only)
MDDR_DQS_TMATCH_1_OUT	OUT	FIFO out signal (32-bit only)
MDDR_DM_RDQS_ECC	INOUT	Dram ECC Data Mask
MDDR_DQS_ECC	INOUT	Dram ECC Data Strobe Input/Output - P Side
MDDR_DQS_ECC_N	INOUT	Dram ECC Data Strobe Input/Output - N Side
MDDR_DQ_ECC ([3:0]/[1:0]/[0])	INOUT	DRAM ECC Data Input/Output
MDDR_DQS_TMATCH_ECC_IN	IN	ECC FIFO in signal
MDDR_DQS_TMATCH_ECC_OUT	OUT	ECC FIFO out signal (32-bit only)

Note: Port widths for some ports change depending on the selection of the PHY width. The notation "[a:0]/[b:0]/[c:0]" is used to denote such ports, where "[a:0]" refers to the port width when a 32-bit PHY width is selected, "[b:0]" corresponds to a 16-bit PHY width, and "[c:0]" corresponds to an 8-bit PHY width.

Fabric Master AXI Bus Interface

Table 3-2 • **Fabric Master AXI Bus Interface**

Port Name	Direction	Description
DDR_AXI_S_AWREADY	OUT	Write address ready
DDR_AXI_S_WREADY	OUT	Write address ready
DDR_AXI_S_BID[3:0]	OUT	Response ID
DDR_AXI_S_BRESP[1:0]	OUT	Write response
DDR_AXI_S_BVALID	OUT	Write response valid
DDR_AXI_S_ARREADY	OUT	Read address ready
DDR_AXI_S_RID[3:0]	OUT	Read ID Tag
DDR_AXI_S_RRESP[1:0]	OUT	Read Response
DDR_AXI_S_RDATA[63:0]	OUT	Read data
DDR_AXI_S_RLAST	OUT	Read Last This signal indicates the last transfer in a read burst
DDR_AXI_S_RVALID	OUT	Read address valid
DDR_AXI_S_AWID[3:0]	IN	Write Address ID
DDR_AXI_S_AWADDR[31:0]	IN	Write address
DDR_AXI_S_AWLEN[3:0]	IN	Burst length
DDR_AXI_S_AWSIZE[1:0]	IN	Burst size
DDR_AXI_S_AWBURST[1:0]	IN	Burst type
DDR_AXI_S_AWLOCK[1:0]	IN	Lock type This signal provides additional information about the atomic characteristics of the transfer
DDR_AXI_S_AWVALID	IN	Write address valid
DDR_AXI_S_WID[3:0]	IN	Write Data ID tag
DDR_AXI_S_WDATA[63:0]	IN	Write data
DDR_AXI_S_WSTRB[7:0]	IN	Write strobes
DDR_AXI_S_WLAST	IN	Write last
DDR_AXI_S_WVALID	IN	Write valid
DDR_AXI_S_BREADY	IN	Write ready
DDR_AXI_S_ARID[3:0]	IN	Read Address ID
DDR_AXI_S_ARADDR[31:0]	IN	Read address
DDR_AXI_S_ARLEN[3:0]	IN	Burst length
DDR_AXI_S_ARSIZE[1:0]	IN	Burst size
DDR_AXI_S_ARBURST[1:0]	IN	Burst type
DDR_AXI_S_ARLOCK[1:0]	IN	Lock Type
DDR_AXI_S_ARVALID	IN	Read address valid
DDR_AXI_S_RREADY	IN	Read address ready

***Table 3-2* • Fabric Master AXI Bus Interface (continued)**

Port Name	Direction	Description
DDR_AXI_S_CORE_RESET_N	IN	MDDR Global Reset
DDR_AXI_S_RMW	IN	Indicates whether all bytes of a 64 bit lane are valid for all beats of an AXI transfer. 0: Indicates that all bytes in all beats are valid in the burst and the controller should default to write commands 1: Indicates that some bytes are invalid and the controller should default to RMW commands This is classed as an AXI write address channel sideband signal and is valid with the AWVALID signal. Only used when ECC is enabled.

Fabric Master AHB0 Bus Interface

***Table 3-3* • Fabric Master AHB0 Bus Interface**

Port Name	Direction	Description
DDR_AHB0_SHREADYOUT	OUT	AHBL slave ready - When high for a write indicates the MDDR is ready to accept data and when high for a read indicates that data is valid
DDR_AHB0_SHRESP	OUT	AHBL response status - When driven high at the end of a transaction indicates that the transaction has completed with errors. When driven low at the end of a transaction indicates that the transaction has completed successfully.
DDR_AHB0_SHRDATA[31:0]	OUT	AHBL read data - Read data from the MDDR slave to the fabric master
DDR_AHB0_SHSEL	IN	AHBL slave select - When asserted, the MDDR is the currently selected AHBL slave on the fabric AHB bus
DDR_AHB0_SHADDR[31:0]	IN	AHBL address - byte address on the AHBL interface
DDR_AHB0_SHBURST[2:0]	IN	AHBL Burst Length
DDR_AHB0_SHSIZE[1:0]	IN	AHBL transfer size - Indicates the size of the current transfer (8/16/32 byte transactions only)
DDR_AHB0_SHTRANS[1:0]	IN	AHBL transfer type - Indicates the transfer type of the current transaction
DDR_AHB0_SHMASTLOCK	IN	AHBL lock - When asserted the current transfer is part of a locked transaction
DDR_AHB0_SHWRITE	IN	AHBL write - When high indicates that the current transaction is a write. When low indicates that the current transaction is a read
DDR_AHB0_S_HREADY	IN	AHBL ready - When high, indicates that the MDDR is ready to accept a new transaction
DDR_AHB0_S_HWDATA[31:0]	IN	AHBL write data - Write data from the fabric master to the MDDR

Fabric Master AHB1 Bus Interface

***Table 3-4* • Fabric Master AHB1 Bus Interface**

Port Name	Direction	Description
DDR_AHB1_SHREADYOUT	OUT	AHBL slave ready - When high for a write indicates the MDDR is ready to accept data and when high for a read indicates that data is valid
DDR_AHB1_SHRESP	OUT	AHBL response status - When driven high at the end of a transaction indicates that the transaction has completed with errors. When driven low at the end of a transaction indicates that the transaction has completed successfully.
DDR_AHB1_SHRDATA[31:0]	OUT	AHBL read data - Read data from the MDDR slave to the fabric master
DDR_AHB1_SHSEL	IN	AHBL slave select - When asserted, the MDDR is the currently selected AHBL slave on the fabric AHB bus
DDR_AHB1_SHADDR[31:0]	IN	AHBL address - byte address on the AHBL interface
DDR_AHB1_SHBURST[2:0]	IN	AHBL Burst Length
DDR_AHB1_SHSIZE[1:0]	IN	AHBL transfer size - Indicates the size of the current transfer (8/16/32 byte transactions only)
DDR_AHB1_SHTRANS[1:0]	IN	AHBL transfer type - Indicates the transfer type of the current transaction
DDR_AHB1_SHMASTLOCK	IN	AHBL lock - When asserted the current transfer is part of a locked transaction
DDR_AHB1_SHWRITE	IN	AHBL write - When high indicates that the current transaction is a write. When low indicates that the current transaction is a read.
DDR_AHB1_SHREADY	IN	AHBL ready - When high, indicates that the MDDR is ready to accept a new transaction
DDR_AHB1_SHWDATA[31:0]	IN	AHBL write data - Write data from the fabric master to the MDDR

Soft Memory Controller Mode AXI Bus Interface

***Table 3-5* • Soft Memory Controller Mode AXI Bus Interface**

Port Name	Direction	Description
SMC_AXI_M_WLAST	OUT	Write last
SMC_AXI_M_WVALID	OUT	Write valid
SMC_AXI_M_AWLEN[3:0]	OUT	Burst length
SMC_AXI_M_AWBURST[1:0]	OUT	Burst type
SMC_AXI_M_BREADY	OUT	Response ready
SMC_AXI_M_AWVALID	OUT	Write Address Valid
SMC_AXI_M_AWID[3:0]	OUT	Write Address ID
SMC_AXI_M_WDATA[63:0]	OUT	Write Data
SMC_AXI_M_ARVALID	OUT	Read address valid
SMC_AXI_M_WID[3:0]	OUT	Write Data ID tag
SMC_AXI_M_WSTRB[7:0]	OUT	Write strobes
SMC_AXI_M_ARID[3:0]	OUT	Read Address ID
SMC_AXI_M_ARADDR[31:0]	OUT	Read address
SMC_AXI_M_ARLEN[3:0]	OUT	Burst length
SMC_AXI_M_ARSIZE[1:0]	OUT	Burst size
SMC_AXI_M_ARBURST[1:0]	OUT	Burst type
SMC_AXI_M_AWADDR[31:0]	OUT	Write Address
SMC_AXI_M_RREADY	OUT	Read address ready
SMC_AXI_M_AWSIZE[1:0]	OUT	Burst size
SMC_AXI_M_AWLOCK[1:0]	OUT	Lock type This signal provides additional information about the atomic characteristics of the transfer
SMC_AXI_M_ARLOCK[1:0]	OUT	Lock Type
SMC_AXI_M_BID[3:0]	IN	Response ID
SMC_AXI_M_RID[3:0]	IN	Read ID Tag
SMC_AXI_M_RRESP[1:0]	IN	Read Response
SMC_AXI_M_BRESP[1:0]	IN	Write response
SMC_AXI_M_AWREADY	IN	Write address ready
SMC_AXI_M_RDATA[63:0]	IN	Read Data
SMC_AXI_M_WREADY	IN	Write ready
SMC_AXI_M_BVALID	IN	Write response valid
SMC_AXI_M_ARREADY	IN	Read address ready
SMC_AXI_M_RLAST	IN	Read Last This signal indicates the last transfer in a read burst
SMC_AXI_M_RVALID	IN	Read Valid

Soft Memory Controller Mode AHB0 Bus Interface

***Table 3-6* • Soft Memory Controller Mode AHB0 Bus Interface**

Port Name	Direction	Description
SMC_AHB_M_HBURST[1:0]	OUT	AHBL Burst Length
SMC_AHB_M_HTRANS[1:0]	OUT	AHBL transfer type - Indicates the transfer type of the current transaction.
SMC_AHB_M_HMASTLOCK	OUT	AHBL lock - When asserted the current transfer is part of a locked transaction
SMC_AHB_M_HWRITE	OUT	AHBL write -- When high indicates that the current transaction is a write. When low indicates that the current transaction is a read
SMC_AHB_M_HSIZE[1:0]	OUT	AHBL transfer size - Indicates the size of the current transfer (8/16/32 byte transactions only)
SMC_AHB_M_HWDATA[31:0]	OUT	AHBL write data - Write data from the MSS master to the fabric Soft Memory Controller
SMC_AHB_M_HADDR[31:0]	OUT	AHBL address - byte address on the AHBL interface
SMC_AHB_M_HRESP	IN	AHBL response status - When driven high at the end of a transaction indicates that the transaction has completed with errors. When driven low at the end of a transaction indicates that the transaction has completed successfully
SMC_AHB_M_HRDATA[31:0]	IN	AHBL read data - Read data from the fabric Soft Memory Controller to the MSS master
SMC_AHB_M_HREADY	IN	AHBL ready - High indicates that the AHBL bus is ready to accept a new transaction

A – Product Support

Microsemi SoC Products Group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, electronic mail, and worldwide sales offices. This appendix contains information about contacting Microsemi SoC Products Group and using these support services.

Customer Service

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

From North America, call **800.262.1060**
From the rest of the world, call **650.318.4460**
Fax, from anywhere in the world, **650.318.8044**

Customer Technical Support Center

Microsemi SoC Products Group staffs its Customer Technical Support Center with highly skilled engineers who can help answer your hardware, software, and design questions about Microsemi SoC Products. The Customer Technical Support Center spends a great deal of time creating application notes, answers to common design cycle questions, documentation of known issues, and various FAQs. So, before you contact us, please visit our online resources. It is very likely we have already answered your questions.

Technical Support

For Microsemi SoC Products Support, visit http://www.microsemi.com/products/fpga-soc/design-support/fpga-soc-support.

Website

You can browse a variety of technical and non-technical information on the Microsemi SoC Products Group home page, at www.microsemi.com/soc.

Contacting the Customer Technical Support Center

Highly skilled engineers staff the Technical Support Center. The Technical Support Center can be contacted by email or through the Microsemi SoC Products Group website.

Email

You can communicate your technical questions to our email address and receive answers back by email, fax, or phone. Also, if you have design problems, you can email your design files to receive assistance. We constantly monitor the email account throughout the day. When sending your request to us, please be sure to include your full name, company name, and your contact information for efficient processing of your request.

The technical support email address is soc_tech@microsemi.com.

My Cases

Microsemi SoC Products Group customers may submit and track technical cases online by going to My Cases.

Outside the U.S.

Customers needing assistance outside the US time zones can either contact technical support via email (soc_tech@microsemi.com) or contact a local sales office.

Visit About Us for sales office listings and corporate contacts.

Sales office listings can be found at www.microsemi.com/soc/company/contact/default.aspx.

ITAR Technical Support

For technical support on RH and RT FPGAs that are regulated by International Traffic in Arms Regulations (ITAR), contact us via soc_tech_itar@microsemi.com. Alternatively, within My Cases, select **Yes** in the ITAR drop-down list. For a complete list of ITAR-regulated Microsemi FPGAs, visit the ITAR web page.

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA

Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996

E-mail: sales.support@microsemi.com

©2016 Microsemi Corporation. All rights reserved. Microsemi and the Microsemi logo are trademarks of Microsemi Corporation. All other trademarks and service marks are the property of their respective owners.

About Microsemi

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for communications, defense & security, aerospace and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; Enterprise Storage and Communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif. and has approximately 4,800 employees globally. Learn more at **www.microsemi.com**.

Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer's responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided "as is, where is" and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.

5-02-00377-5/11.16