IGLOO2 FPGA

Fabric DDR Controller Configuration

Libero SoC v11.6 and later

Microsemi

Table of Contents

Introduction . . . 3
System Builder . . . 3
DDR Memory Controller Core from Catalog . . . 3

1 Fabric External Memory DDR Controller Configurator . . . 4

2 FDDR Controller Configuration . . . 8
Fabric DDR Control Registers . . . 8
Exporting DDR Configuration Files . . . 13

3 Port Description . . . 15

A Product Support . . . 20
Customer Service . . . 20
Customer Technical Support Center . . . 20
Technical Support . . . 20
Website . . . 20
Contacting the Customer Technical Support Center . . . 20
ITAR Technical Support . . . 21

Introduction

The IGLOO2 FPGA has an embedded Fabric DDR (FDDR) controller for access to an external DDR memory from the FPGA Fabric. The FDDR controls off-chip DDR memories.

When you use System Builder to build a system block, System Builder configures the FDDR controller for you based on your entries and selections.

Refer to the IGLOO2 High Speed DDR Interfaces User Guide for details.

System Builder

1. Invoke System Builder from the Design Flow window.
2. In the Device Features page of System Builder, check the Fabric External DDR Memory (FDDR) check box. Click **Next** to open the DDR Memory Controller Configurator.

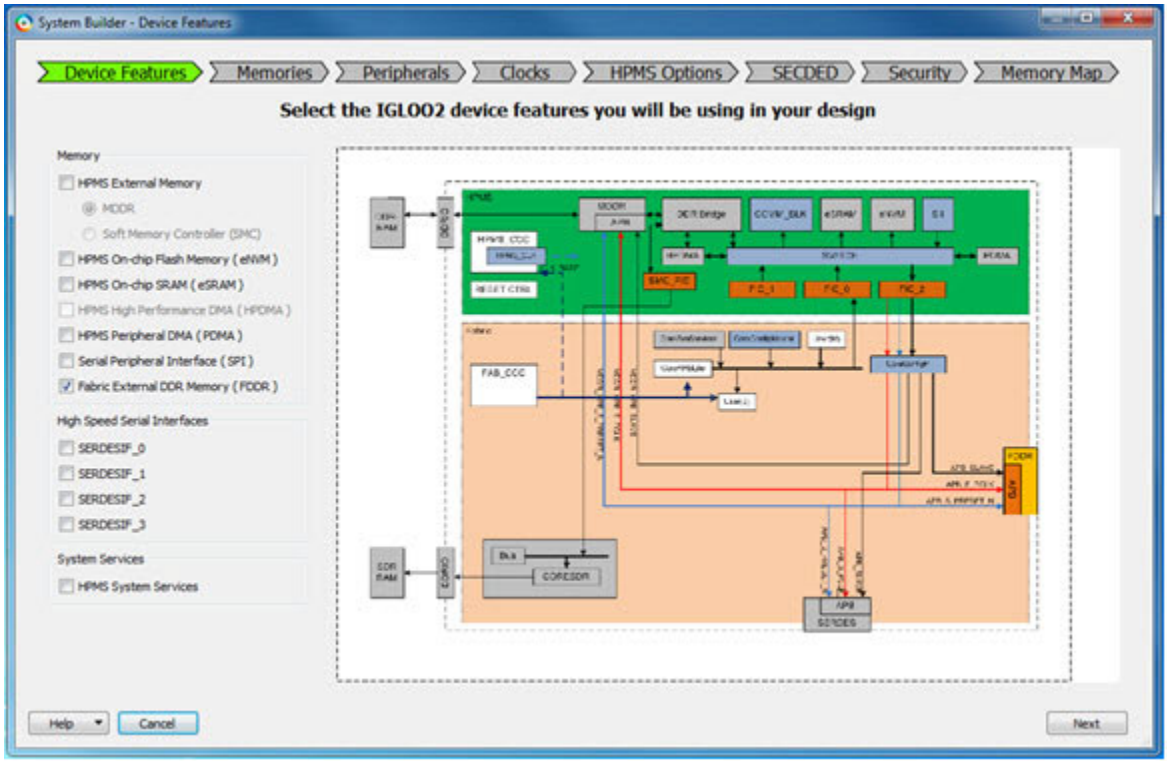

***Figure 1* • System Builder—Device Features Page**

3. Click the **General**, **Memory Initialization**, and **Memory Timing** tabs of the Configurator to configure the DDR Memory Controller.

DDR Memory Controller Core from Catalog

Alternatively, drag the DDR Memory Controller core from the Catalog and drop it into your SmartDesign Canvas. Double-click the Core to open the Configurator.

1 – Fabric External Memory DDR Controller Configurator

The Fabric External Memory DDR (FDDR) Configurator is used to configure the overall datapath and the external DDR memory parameters for the Fabric DDR Controller.

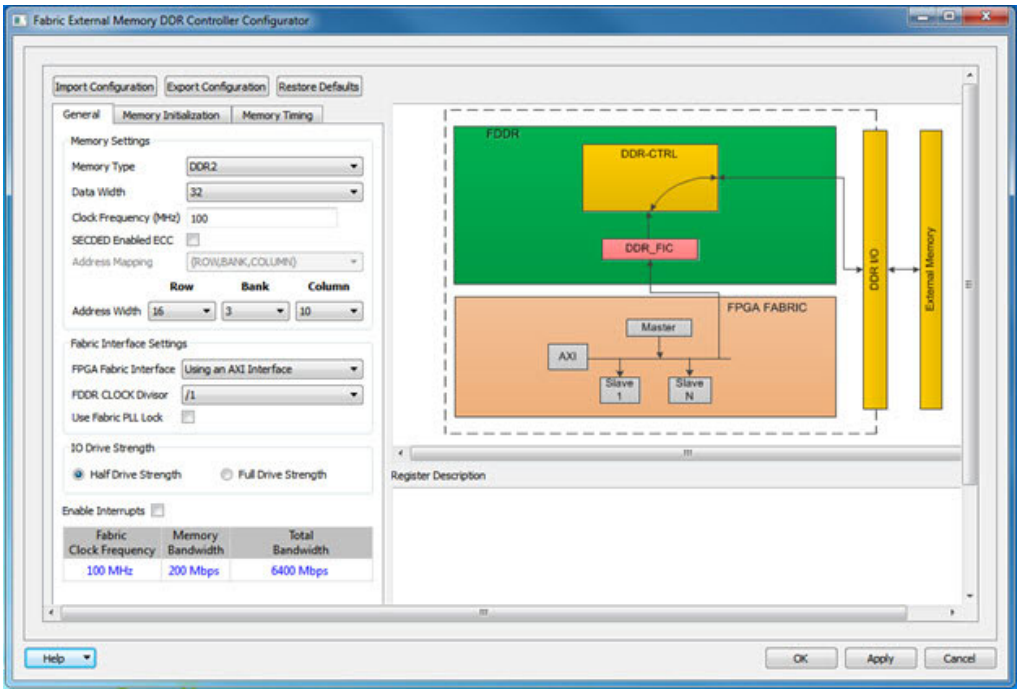

***Figure 1-1* • FDDR Configurator Overview**

Memory Settings

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

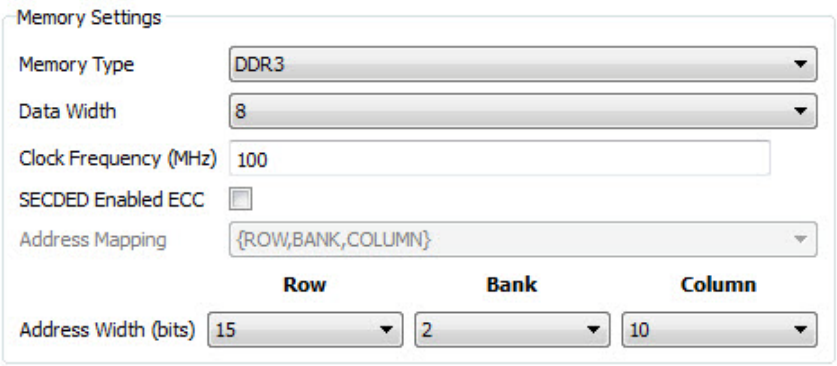

***Figure 1-2* • Memory Settings**

- **Memory Type** - LPDDR, DDR2, or DDR3
- **Data Width** - 32-bit, 16-bit or 8-bit
- **Clock Frequency** - Any value (Decimal/Fractional) in the range of 20 MHz to 333 MHz. This is DDR_CLK, the clock generated from FDDR to DDR.
- **SECDED Enabled ECC** - ON or OFF
- **Address Width (bits)** - Refer to the data sheet of your DDR memory vendor for the number of row, bank, and column address bits for the LPDDR/DDR2/DDR3 memory you use. Click 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. Take 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

FPGA Fabric Interface - This is the data interface between the FDDR and the FPGA design. Because the FDDR is a memory controller, it is intended to be a slave on an AXI or AHB bus. The Master of the bus initiates bus transactions, which are in turn interpreted by the FDDR as memory transactions and communicated to the off-chip DDR Memory. FDDR fabric interface options are:

- **Using an AXI Interface** - One master accesses the FDDR through a 64-bit AXI interface.
- **Using a Single AHB Interface** - One master accesses the FDDR through a single 32-bit AHB interface.
- **Using Two AHB Interfaces** - Two masters access the FDDR using two 32-bit AHB interfaces.

FDDR CLOCK Divisor - Specifies the frequency ratio between the DDR Controller clock (CLK_FDDR) and the clock controlling the fabric interface (CLK_FIC64). The CLK_FIC64 frequency should be equal to that of the AHB/AXI subsystem that is connected to the FDDR AHB/AXI bus interface. For example, if you have a DDR RAM running at 200 MHz and your Fabric/AXI Subsystem runs at 100 MHz, you must select a divisor of 2 (Figure 1-3).

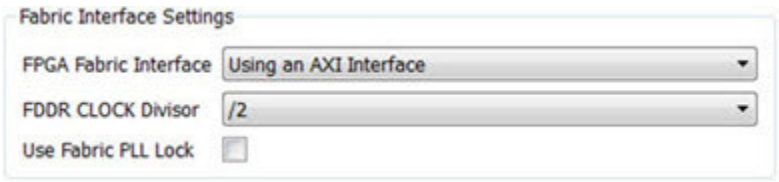

***Figure 1-3* • Fabric Interface Settings - AXI Interface and FDDR Clock Divisor Agreement**

Use Fabric PLL LOCK - If CLK_BASE is sourced from a Fabric CCC, you can connect the fabric CCC LOCK output to the FDDR FAB_PLL_LOCK input. CLK_BASE is not stable until the Fabric CCC locks. Therefore, Microsemi recommends that you hold the FDDR in reset (i.e., assert the CORE_RESET_N input) until CLK_BASE is stable. The LOCK output of the Fabric CCC indicates that the Fabric CCC output clocks are stable. By checking the Use FAB_PLL_LOCK option, you can expose the FAB_PLL_LOCK input port of the FDDR. You can then connect the LOCK output of the Fabric CCC to the FAB_PLL_LOCK input of the FDDR.

IO Drive Strength (DDR2 and DDR3 only)

Select one of the following drive strengths for your DDR I/O's:

- Half Drive Strength
- Full Drive Strength

Depending on your DDR Memory type and the I/O Strength you select, Libero SoC sets the DDR I/O Standard for your FDDR system as follows:

DDR Memory Type	Half Drive Strength	Full Drive Strength
DDR3	SSTL15I	SSTL15II
DDR2	SSTL18I	SSTL15II

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 IGLOO2 High Speed DDR Interfaces User Guide.

Enable Interrupts

The FDDR is capable of raising interrupts when certain predefined conditions are satisfied. Check Enable Interrupts in the FDDR configurator if you would like to use these interrupts in your application. This exposes the interrupt signals on the FDDR instance. You can connect these interrupt signals as your design requires. The following Interrupt signals and their preconditions are available:

- **FIC_INT** - Generated when there is an error in the transaction between the Master and the FDDR
- **O_CAL_INT** - Enables you to recalibrate DDR I/O's by writing to DDR controller registers via the APB configuration interface. When calibration is complete, this interrupt is raised. For details, refer to the IGLOO2 High Speed DDR Interfaces User Guide.
- **PLL_LOCK_INT** - Indicates that the FDDR FPLL has locked
- **PLL_LOCKLOST_INT** - Indicates that the FDDR FPLL has lost lock
- **FDDR_ECC_INT** - Indicates a single or two-bit error has been detected

Fabric Clock Frequency

Clock frequency calculation based on your current Clock frequency and CLOCK divisor, displayed in MHz.

Fabric Clock Frequency (in MHz) = Clock Frequency / CLOCK divisor

Memory Bandwidth

Memory bandwidth calculation based on your current Clock Frequency value in Mbps. This is the bandwidth where the FDDR operates the DDR memory.

Memory Bandwidth (in Mbps) = 2 * Clock Frequency

Total Bandwidth

Total bandwidth calculation based on your current Clock Frequency, Data Width and CLOCK divisor, in Mbps. This is the maximum bandwidth available with a fabric master.

Total Bandwidth (in Mbps) = (2 * Clock Frequency * Data Width) / CLOCK Divisor

2 – FDDR Controller Configuration

When you use the Fabric 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. In this section, we describe how to configure the Fabric DDR controller registers and how the configuration data is managed as part of the overall Peripheral Initialization solution. Refer to the IGLOO2 FPGA High Speed DDR Interface User's Guide for detailed information about the Peripheral Initialization solution.

Fabric DDR Control Registers

The Fabric DDR Controller has a set of registers that need to be configured at runtime. The configuration values for these registers represent different parameters, e.g. DDR mode, PHY width, burst mode, ECC, etc. For details about DDR controller configuration registers, refer to the IGLOO2 High Speed DDR Interfaces User Guide.

Fabric DDR Registers Configuration

Use the Memory Initialization (Figure 2-1) and Memory Timing (Figure 2-2) 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 Window (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.

.

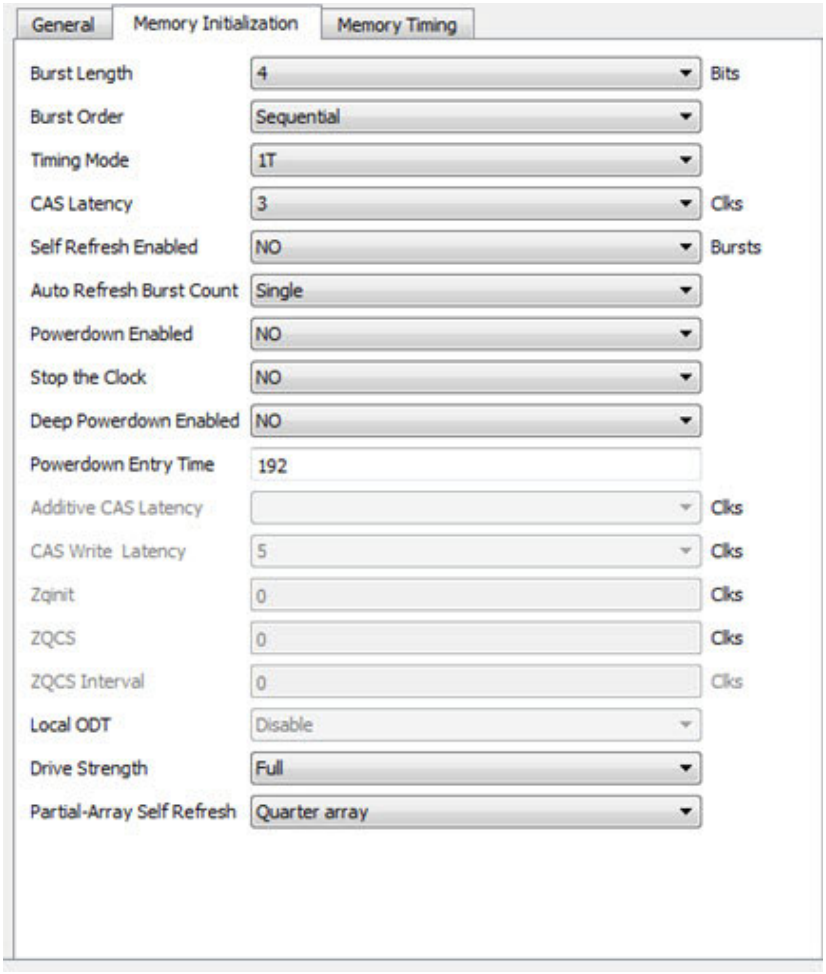

***Figure 2-1* • FDDR Configuration - Memory Initialization (LPDDR)**

- **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.

.

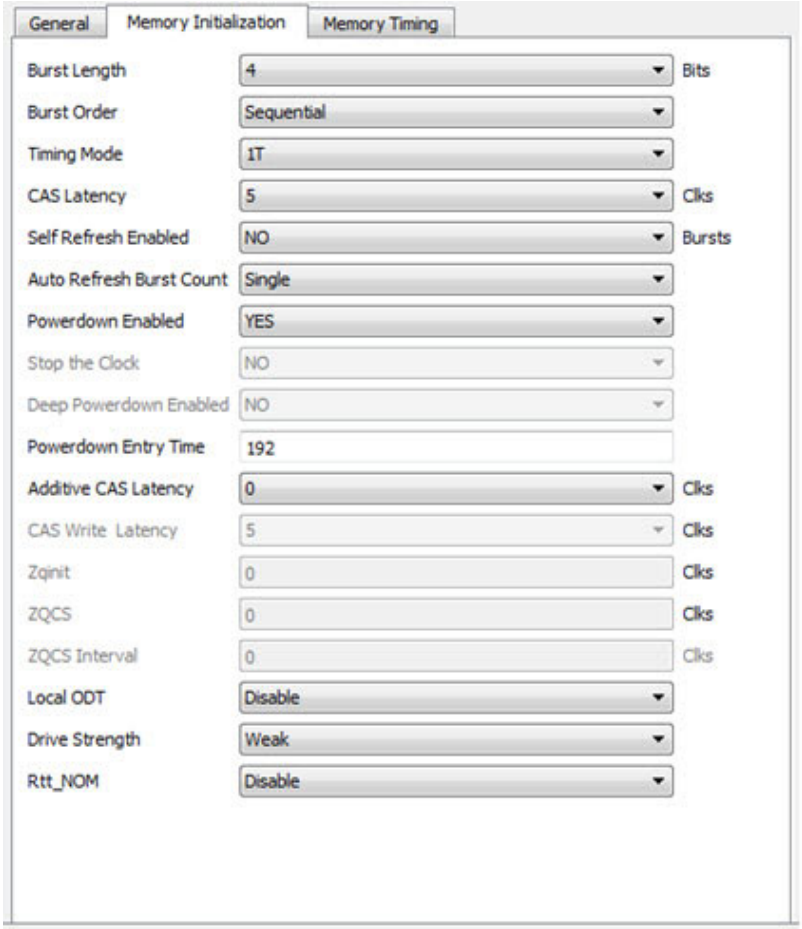

Figure 2-2 **• FDDR Configuration - Memory Initialization (DDR2)**

.

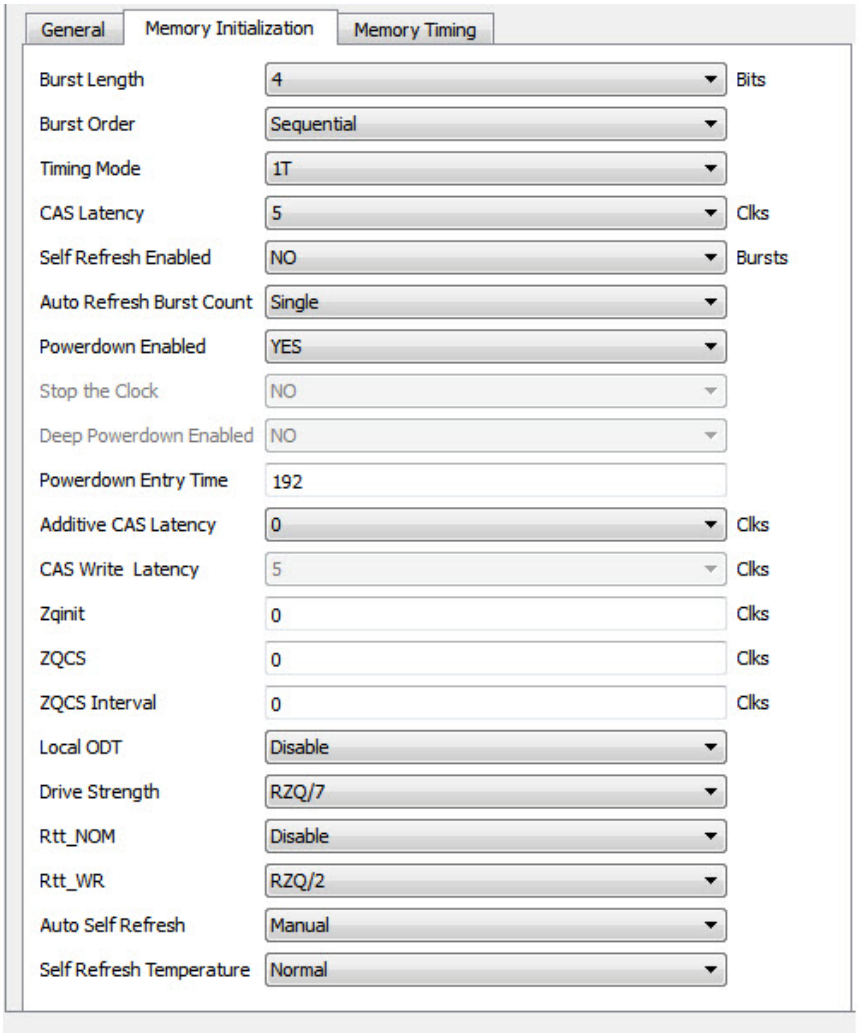

Figure 2-3 **• FDDR Configuration - Memory Initialization (DDR3)**

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. .

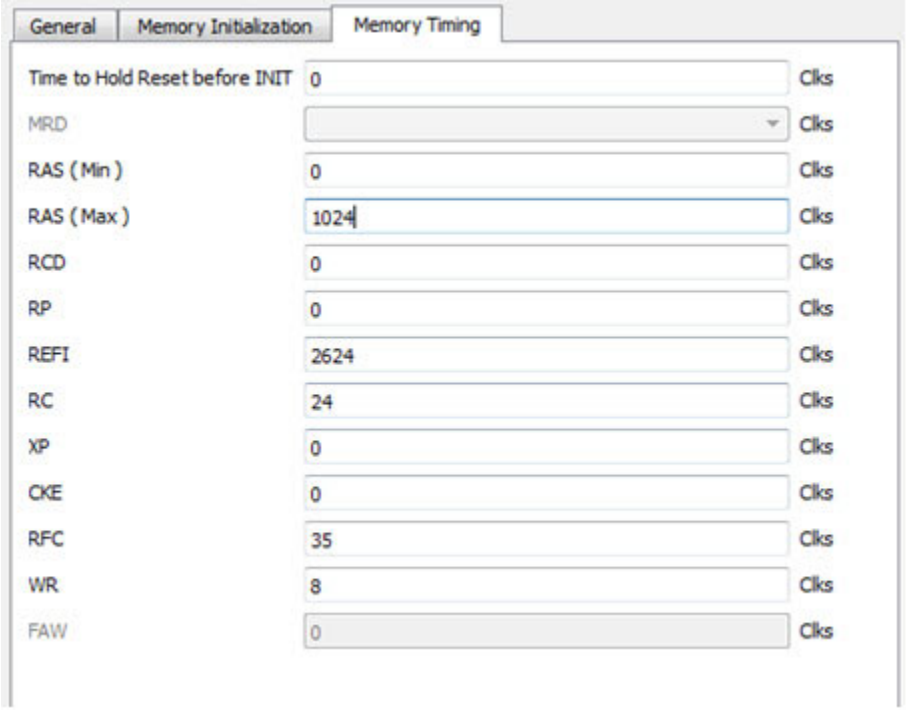

***Figure 2-4* • FDDR Configuration - Memory Timing Tab**

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 **Import Configuration** 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. For details, refer to the IGLOO2 High Speed DDR Interfaces User Guide.

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 box.

If you want to undo changes you have made to the DDR register configuration, you can do so with Restore Default. 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 FDDR Configurator computes values for all DDR configuration registers. 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**

Libero automatically stores this configuration data in the eNVM. Upon FPGA reset, this configuration data will automatically be copied into the FDDR. See Figure 2-7 and Figure 2-8.

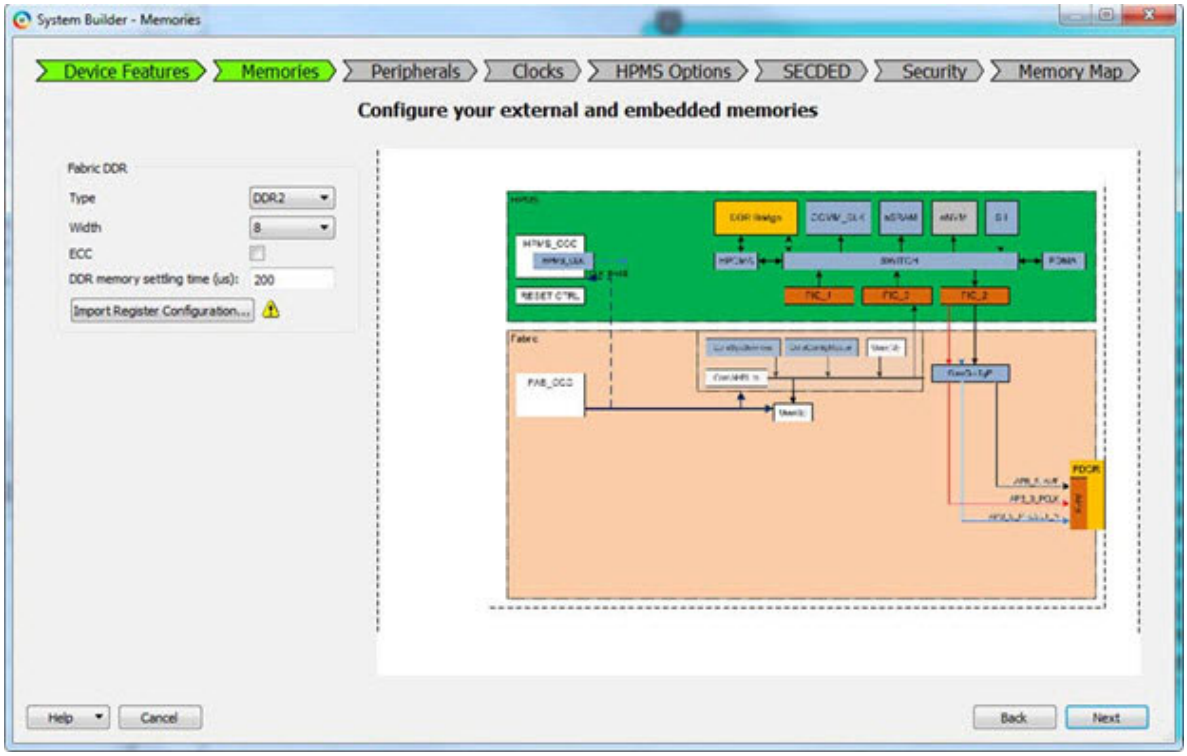

***Figure 2-7* • System Builder and FDDR**

```
ddrc_dyn_soft_reset_CR 0x00 ;
ddrc_dyn_refresh_1_CR 0x27DE ;
ddrc_dyn_refresh_2_CR 0x30F ;
ddrc_dyn_powerdown_CR 0x02 ;
ddrc_dyn_debug_CR 0x00 ;
ddrc_ecc_data_mask_CR 0x0000 ;
ddrc_addr_map_col_1_CR 0x3333 ;
```

***Figure 2-8* • Register Configuration File Syntax**

3 – Port Description

DDR PHY Interface

These ports are exposed at the top level of the System Builder generated block. For details, refer to the IGLOO2 High Speed DDR Interfaces User Guide.

***Table 3-1* • DDR PHY Interface**

Port Name	Direction	Description	Remarks
FDDR_CAS_N	OUT	DRAM CASN	
FDDR_CKE	OUT	DRAM CKE	
FDDR_CLK	OUT	Clock, P side	
FDDR_CLK_N	OUT	Clock, N side	
FDDR_CS_N	OUT	DRAM CSN	
FDDR_ODT	OUT	DRAM ODT	Ignore this signal for LPDDR Interface. For LPDDR, mark it unused.
FDDR_RAS_N	OUT	DRAM RASN	
FDDR_RESET_N	OUT	DRAM Reset for DDR3	Ignore this signal for LPDDR Interface. For LPDDR, mark it unused.
FDDR_WE_N	OUT	DRAM WEN	
FDDR_ADDR[15:0]	OUT	DRAM Address bits	Address MSB vary with the number of rows in the DDR memory. See Table 3-2 below.
FDDR_BA[2:0]	OUT	DRAM Bank Address	For LPDDR, BA[2] is not used. Slice the bus and mark BA[2] unused. Connect BA[1:0] to LPDDR.
FDDR_DM_RDQS[4:0]	INOUT	DRAM Data Mask	
FDDR_DQS[4:0]	INOUT	DRAM Data Strobe Input/Output - P Side	
FDDR_DQS_N[4:0]	INOUT	DRAM Data Strobe Input/Output - N Side	
FDDR_DQ[35:0]	INOUT	DRAM Data Input/ Output	
FDDR_FIFO_WE_IN[2:0]	IN	FIFO in signal	
FDDR_FIFO_WE_OUT[2:0]	OUT	FIFO out signal	
FDDR_DM_RDQS ([3:0]/[1:0]/[0])	INOUT	DRAM Data Mask	For LPDDR interface, this port direction is OUT. RDQS function is not supported. Only DM function is supported. Connect it to DRAM_DM input port of LPDDR.
FDDR_DQS ([3:0]/[1:0]/[0])	INOUT	DRAM Data Strobe Input/Output - P Side	

***Table 3-1* • DDR PHY Interface (continued)**

Port Name	Direction	Description	Remarks
FDDR_DQS_N ([3:0]/[1:0]/[0])	INOUT	DRAM Data Strobe Input/Output - N Side	
FDDR_DQ ([31:0]/[15:0]/[7:0])	INOUT	DRAM Data Input/ Output	
FDDR_DQS_TMATCH_0_IN	IN	FIFO in signal	For LPDDR, connect this signal to FDDR_DQS_TMATCH_0_OUT.
FDDR_DQS_TMATCH_0_OUT	OUT	FIFO out signal	For LPDDR, connect this signal to FDDR_DQS_TMATCH_0_IN.
FDDR_DQS_TMATCH_1_IN	IN	FIFO in signal (32-bit only)	For LPDDR, connect this signal to FDDR_DQS_TMATCH_1_OUT.
FDDR_DQS_TMATCH_1_OUT	OUT	FIFO out signal (32-bit only)	For LPDDR, connect this signal to FDDR_DQS_TMATCH_1_IN.
FDDR_DM_RDQS_ECC	INOUT	DRAM ECC Data Mask	
FDDR_DQS_ECC	INOUT	DRAM ECC Data Strobe Input/Output - P Side	
FDDR_DQS_ECC_N	INOUT	DRAM ECC Data Strobe Input/Output - N Side	
FDDR_DQ_ECC ([3:0]/[1:0]/[0])	INOUT	DRAM ECC Data Input/Output	
FDDR_DQS_TMATCH_ECC_IN	IN	ECC FIFO in signal	
FDDR_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.

***Table 3-2* • Address MSB Value for LPDDR**

LPDDR Memory Width	64 MB	128 MB	256 MB	512 MB	1 GB	2 GB
16-bit	11	11	12	12	13	13
32-bit	10	11	11	12	12 OR 13	13

Note: This Address MSB table is for individual DDR components. A DDR dim/board may use several identical components to increase the size (for example, 2 x16 2GB to create x32 4GB). In that case, it is the individual component width/depth that controls ADDR MSB. It is also equivalent to the row address width parameter in the configurator.

FDDR Core Ports

***Table 3-3* • FDDR Core Ports**

Port Name	Direction	Description
CORE_RESET_N	IN	FDDR Controller Reset
CLK_BASE	IN	FDDR Fabric Interface Clock
FPLL_LOCK	OUT	FDDR PLL Lock output - high when FDDR PLL is locked
CLK_BASE_PLL_LOCK	IN	Fabric PLL Lock Input. This input is exposed only when the Use FAB_PLL_LOCK option is selected.

Interrupt Ports

This group of ports is exposed when you select the Enable Interrupts option.

***Table 3-4* • Interrupt Ports**

Port Name	Direction	Description
PLL_LOCK_INT	OUT	Asserts when FDDR PLL locks
PLL_LOCKLOST_INT	OUT	Asserts when FDDR PLL lock is lost
ECC_INT	OUT	Asserts when an ECC Event occurs
IO_CALIB_INT	OUT	Asserts when I/O calibration is complete
FIC_INT	OUT	Asserts when there is an error in the AHB/AXI protocol on the Fabric interface

APB3 Configuration Interface

***Table 3-5* • APB3 Configuration Interface**

Port Name	Direction	Description
APB_S_PENABLE	IN	Slave Enable
APB_S_PSEL	IN	Slave Select
APB_S_PWRITE	IN	Write Enable
APB_S_PADDR[10:2]	IN	Address
APB_S_PWDATA[15:0]	IN	Write Data
APB_S_PREADY	OUT	Slave Ready
APB_S_PSLVERR	OUT	Slave Error
APB_S_PRDATA[15:0]	OUT	Read Data
APB_S_PRESET_N	IN	Slave Reset
APB_S_PCLK	IN	Clock

AXI Bus Interface

***Table 3-6* • AXI Bus Interface**

Port Name	Direction	Description
AXI_S_AWREADY	OUT	Write address ready
AXI_S_WREADY	OUT	Write address ready
AXI_S_BID[3:0]	OUT	Response ID
AXI_S_BRESP[1:0]	OUT	Write response
AXI_S_BVALID	OUT	Write response valid
AXI_S_ARREADY	OUT	Read address ready
AXI_S_RID[3:0]	OUT	Read ID Tag
AXI_S_RRESP[1:0]	OUT	Read Response
AXI_S_RDATA[63:0]	OUT	Read data
AXI_S_RLAST	OUT	Read Last - This signal indicates the last transfer in a read burst.
AXI_S_RVALID	OUT	Read address valid
AXI_S_AWID[3:0]	IN	Write Address ID
AXI_S_AWADDR[31:0]	IN	Write address
AXI_S_AWLEN[3:0]	IN	Burst length
AXI_S_AWSIZE[1:0]	IN	Burst size
AXI_S_AWBURST[1:0]	IN	Burst type
AXI_S_AWLOCK[1:0]	IN	Lock type - This signal provides additional information about the atomic characteristics of the transfer.
AXI_S_AWVALID	IN	Write address valid
AXI_S_WID[3:0]	IN	Write Data ID tag
AXI_S_WDATA[63:0]	IN	Write data
AXI_S_WSTRB[7:0]	IN	Write strobes
AXI_S_WLAST	IN	Write last
AXI_S_WVALID	IN	Write valid
AXI_S_BREADY	IN	Write ready
AXI_S_ARID[3:0]	IN	Read Address ID
AXI_S_ARADDR[31:0]	IN	Read address
AXI_S_ARLEN[3:0]	IN	Burst length
AXI_S_ARSIZE[1:0]	IN	Burst size
AXI_S_ARBURST[1:0]	IN	Burst type
AXI_S_ARLOCK[1:0]	IN	Lock Type
AXI_S_ARVALID	IN	Read address valid
AXI_S_RREADY	IN	Read address ready

***Table 3-6* • AXI Bus Interface (continued)**

Port Name	Direction	Description
AXI_S_CORE_RESET_N	IN	MDDR Global Reset
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.

AHB0 Bus Interface

***Table 3-7* • AHB0 Bus Interface**

Port Name	Direction	Description
AHB0_S_HREADYOUT	OUT	AHBL slave ready - When high for a write indicates the slave is ready to accept data and when high for a read indicates that data is valid.
AHB0_S_HRESP	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.
AHB0_S_HRDATA[31:0]	OUT	AHBL read data - Read data from the slave to the master.
AHB0_S_HSEL	IN	AHBL slave select - When asserted, the slave is the currently selected AHBL. slave on the AHB bus.
AHB0_S_HADDR[31:0]	IN	AHBL address - byte address on the AHBL interface.
AHB0_S_HBURST[2:0]	IN	AHBL Burst Length.
AHB0_S_HSIZE[1:0]	IN	AHBL transfer size - Indicates the size of the current transfer (8/16/32 byte transactions only).
AHB0_S_HTRANS[1:0]	IN	AHBL transfer type - Indicates the transfer type of the current transaction.
AHB0_S_HMASTLOCK	IN	AHBL lock - When asserted the current transfer is part of a locked transaction.
AHB0_S_HWRITE	IN	AHBL write - When high indicates that the current transaction is a write. When low indicates that the current transaction is a read.

***Table 3-7* • AHB0 Bus Interface (continued)**

Port Name	Direction	Description
AHB0_S_HREADY	IN	AHBL ready - When high, indicates that the slave is ready to accept a new transaction.
AHB0_S_HWDATA[31:0]	IN	AHBL write data - Write data from the master to the slave.

AHB1 Bus Interface

***Table 3-8* • AHB1 Bus Interface**

Port Name	Direction	Description
AHB1_S_HREADYOUT	OUT	AHBL slave ready - When high for a write indicates the slave is ready to accept data and when high for a read indicates that data is valid.
AHB1_S_HRESP	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.
AHB1_S_HRDATA[31:0]	OUT	AHBL read data - Read data from the slave to the master.
AHB1_S_HSEL	IN	AHBL slave select - When asserted, the slave is the currently selected AHBL. slave on the AHB bus.
AHB1_S_HADDR[31:0]	IN	AHBL address - byte address on the AHBL interface.
AHB1_S_HBURST[2:0]	IN	AHBL Burst Length.
AHB1_S_HSIZE[1:0]	IN	AHBL transfer size - Indicates the size of the current transfer (8/16/32 byte transactions only).
AHB1_S_HTRANS[1:0]	IN	AHBL transfer type - Indicates the transfer type of the current transaction.
AHB1_S_HMASTLOCK	IN	AHBL lock - When asserted the current transfer is part of a locked transaction.
AHB1_S_HWRITE	IN	AHBL write - When high indicates that the current transaction is a write. When low indicates that the current transaction is a read.
AHB1_S_HREADY	IN	AHBL ready - When high, indicates that the slave is ready to accept a new transaction.
AHB1_S_HWDATA[31:0]	IN	AHBL write data - Write data from the master to the slave.

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, 408.643.6913

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

Visit the Customer Support website (www.microsemi.com/soc/support/search/default.aspx) for more information and support. Many answers available on the searchable web resource include diagrams, illustrations, and links to other resources on the website.

Website

You can browse a variety of technical and non-technical information on the SoC 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. 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

©2015 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.

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for communications, defense and 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; security technologies and scalable anti-tamper products; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif. and has approximately 3,400 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.