SmartFusion2

System Builder User's Guide

Table of Contents

Introduction

System Builder is a graphical design wizard designed specifically for SmartFusion2 based systems. System Builder takes you through the following steps:

- Selecting the Device Features for your system
- Adding any additional needed peripherals in the fabric
- Setting required configuration options for each selected feature
- Building a correct-by-design complete system

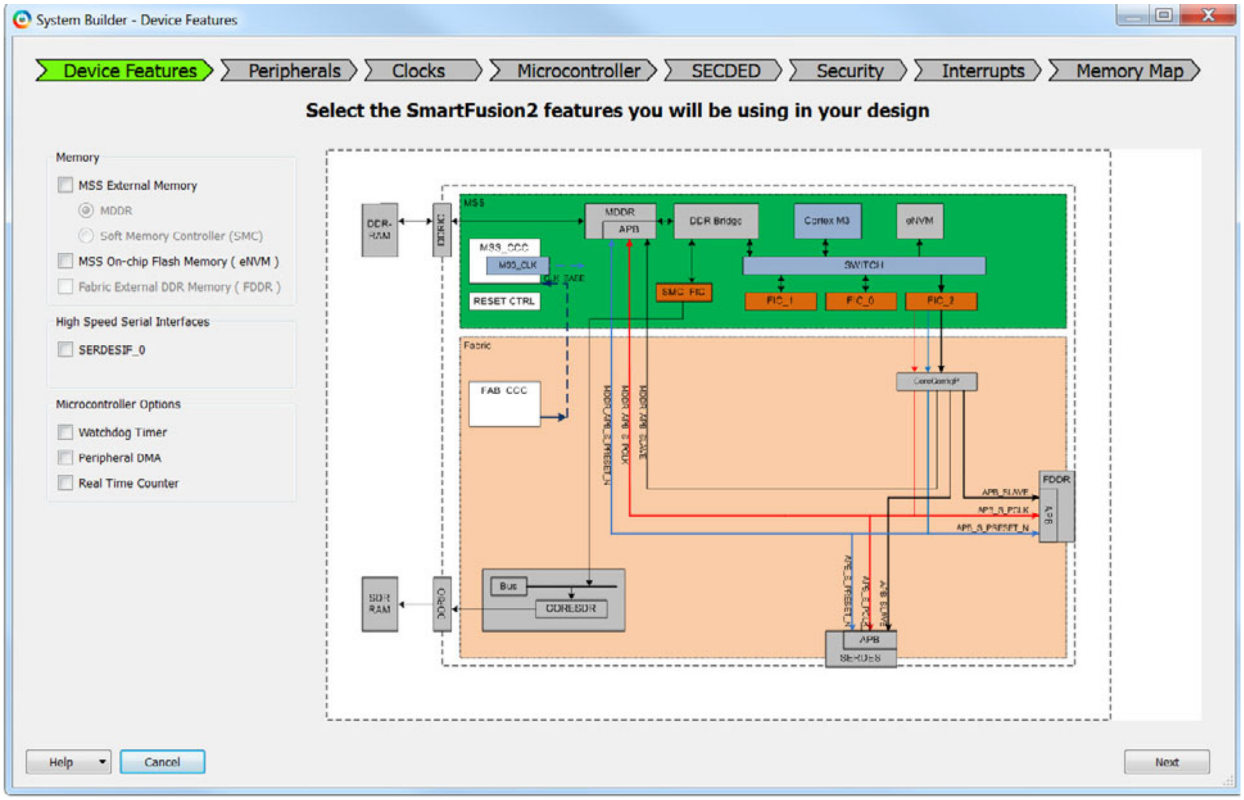
SmartFusion2 System Builder wizard (Figure 1) creates your design based on high level design specifications that define your intended system. System Builder enables you to focus on your design specializations instead of on the specific silicon requirements of a SmartFusion2 based design.

This simplifies the design creation process. The built-in design rule check feature prevents you from moving forward if there are mistakes or conflicts.

The design that is produced by the System Builder follows all the SmartFusion2 silicon design rules.

You can also extend the System Builder generated design with your own custom peripherals and logic by specifying your options and then using SmartDesign to connect up your custom peripherals.

System Builder supports the SmartFusion2 family.



System Builder - Device Features

Select the SmartFusion2 features you will be using in your design

Memory

- ☐ MSS External Memory
  - ● MDDR
  - ○ Soft Memory Controller (SMC)
- ☐ MSS On-chip Flash Memory (eNVM)
- ☐ Fabric External DDR Memory (FDDR)

High Speed Serial Interfaces

- ☐ SERDESIF_0

Microcontroller Options

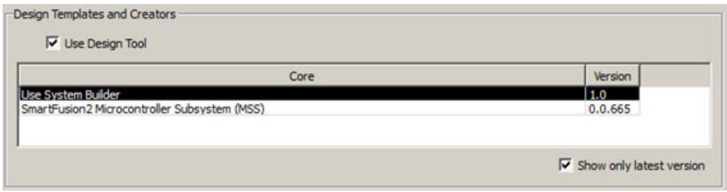
- ☐ Watchdog Timer
- ☐ Peripheral DMA
- ☐ Real Time Counter

Figure 1 **• Selecting Device Features in System Builder**

System Builder uses your feature selections to instantiate, configure, and connect the necessary low level blocks to achieve your requirements. This automates what you would have had to do manually in SmartDesign.

1 – Accessing System Builder

You can access System Builder when you first create a project for SmartFusion2 in the Design Templates and Creators panel (Figure 1-1).



Design Templates and Creators

☑ Use Design Tool

Core	Version
Use System Builder	1.0
SmartFusion2 Microcontroller Subsystem (MSS)	0.0.665

☑ Show only latest version

Figure 1-1 **• Design Templates and Creators Panel - System Builder**

If you already have your project open you can invoke System Builder from the Design Flow window (Figure 1-2).

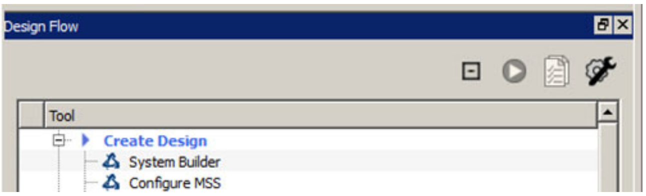


Figure 1-2 **• System Builder in the Design Flow Window**

System Builder has a set of required cores that it must have to generate a valid design. This list of required cores is downloaded automatically if you are connected to the Internet when the System Builder is invoked. If these cores are not available or present, then System Builder displays a message indicating that certain cores are not available in your vault and that you will need to download them.

2 – Configuration Pages

Device Features

Use this page to select the SmartFusion2 features you will be using in your design, including Memory and Microcontroller Options:

To access MSS External Memory, you may select either MDDR or SMC:

- **MDDR** - If you choose MDDR to access External Memory, you need to configure the MDDR in the Memories Page
- **Soft Memory Controller (SMC)** - If you choose SMC to access the MSS External Memory, System Builder builds a MSS SMC FIC Subsystem automatically into your design.

Note: You can select either MDDR or SMC but NOT both. This is a silicon limitation.

- **MSS On-Chip Flash Memory (eNVM)** - If you choose eNVM, you need to configure the ENVM in the Memories Page.
- **Fabric External DDR Memory (FDDR)** - If you choose FDDR, you need to configure the FDDR in the Memories Page.

SERDES (High Speed Serial Interfaces)

Depending on the device you have selected for your project, you may have zero to four SERDES blocks available for your system. Select the number of SERDES you need. System Builder exposes the interface ports at the top level of your system for you to connect the SERDES to your system. Use SmartDesign to instantiate the SERDES at the top level of your design and connect the ports to the top level SERDES Interface Ports.

Libero automatically stores SERDES configuration data from the SERDES Configurator in the eNVM. Upon system reset, this configuration data is automatically copied into the appropriate configuration registers of the SERDES block over the APB connection from the System Builder component to the SERDESIF block.

Refer to the SmartFusion2 High Speed Serial Interfaces User's Guide for a complete description of all the SERDES options and registers.

Microcontroller Options

You can select the following Microcontroller options if you want them enabled in your design. If unselected, they will be disabled.

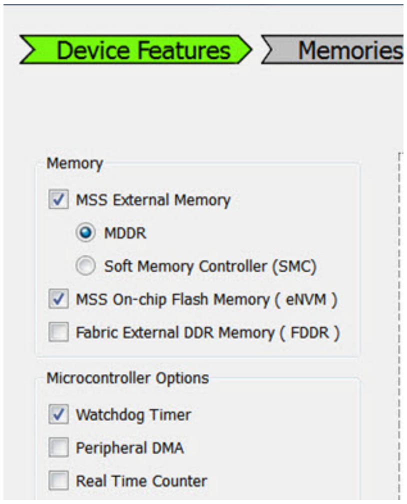
- Watchdog Timer
- Peripheral DMA
- Real Time Counter

Memories

This page is available if you select MSS DDR (MDDR), Fabric DDR (FDDR) or MSS On-chip Flash Memory (eNVM) in the Device Features configuration page.

MSS External Memory

For accessing an external memory, you can either choose to access a double data rate memory (DDR) using MDDR or single data rate memory (SDRAM) using SMC (Figure 2-1).



Device Features

Memories

Memory

☑ MSS External Memory

● MDDR

○ Soft Memory Controller (SMC)

☑ MSS On-chip Flash Memory (eNVM)

□ Fabric External DDR Memory (FDDR)

Microcontroller Options

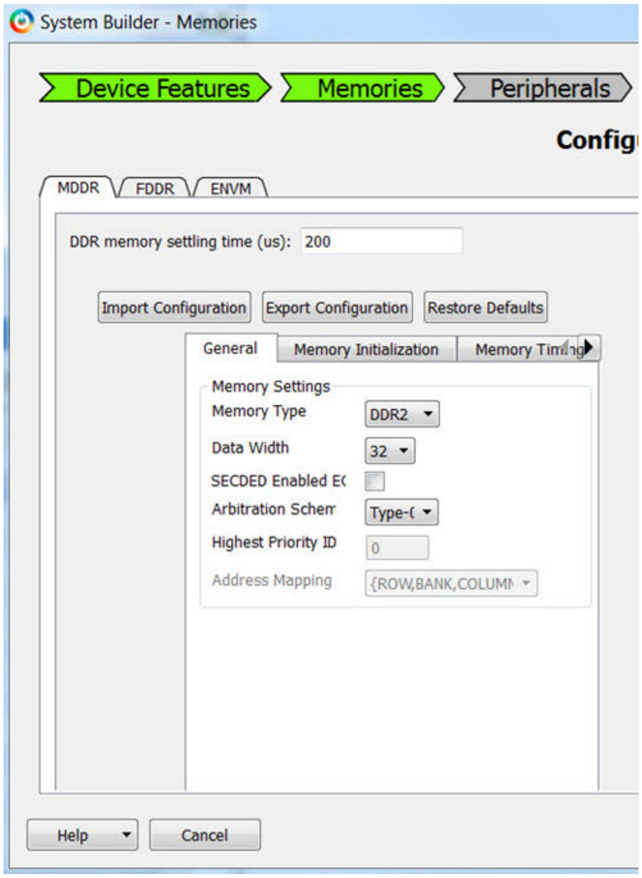
☑ Watchdog Timer

□ Peripheral DMA

□ Real Time Counter

Figure 2-1 **• MSS External Memory - Partial View**

If you have selected the double data rate memory (MDDR), then you must select the memory standard that matches your External Memory: DDR2, DDR3, or LPDDR (Figure 2-2).



DDR memory settling time (us): 200

Memory Settings

Memory Type: DDR2

Data Width: 32

SECDED Enabled E(□

Arbitration Schem: Type-(

Highest Priority ID: 0

Address Mapping: {ROW,BANK,COLUMN

Figure 2-2 • **MDDR Configuration - Partial View**

Specify the DDR memory settling time; this is the amount of time it will take for the DDR memory to be initialized. Refer to the MDDR Configuration Guide for details on how this settling time can be computed. The Default value is 200 us.

The MDDR controller must be configured to match the external selected DDR memory standard. The configuration of the MDDR can be defined in a file and the file can be imported using the System Builder via the Import Configuration button. The configuration is done through the CoreConfigP soft IP core, which is the master of the configuration data initialization process.

Upon reset, the soft IP core CoreSF2Config copies the data from embedded nonvolatile memory (eNVM) to the configuration registers of the DDR through the FIC_2 advanced peripheral bus (APB) interface.

For more details on Peripheral Initialization, review the CoreSF2Config and CoreSF2Reset application notes, available on the Microsemi website.

You can save your MDDR configuration and export it to a file using the Export Configuration button.

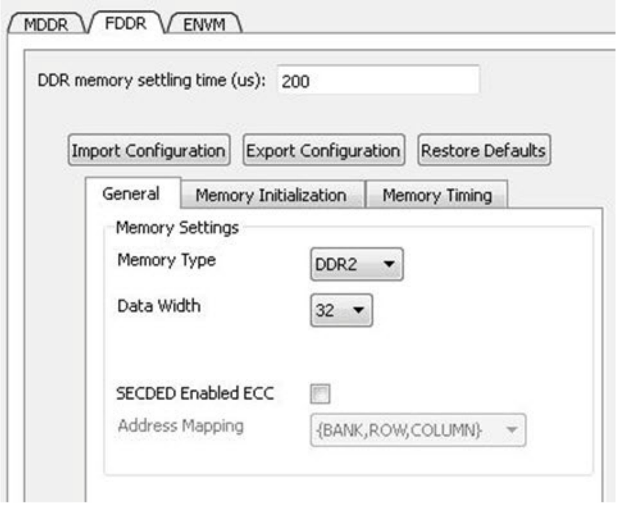
Refer to the DDR Interfaces User’s Guide for a complete description of all the DDR configuration options and registers.

If you have selected the SMC to control a Single Data Rate external Memory (SDR), a Soft Memory Controller (SMC) is automatically instantiated in the Fabric to support this data path. You may want to select this option if you have a non-DDR based external memory chip that you want to interface to.

Fabric DDR

For the FDDR memory, you must select the memory standard: DDR2, DDR3, or LPDDR (Figure 2-3).

Specify the DDR memory settling time. This is the amount of time it will take for the Fabric DDR memory to be initialized. The default value is 200 us. Refer to the Microsemi Fabric DDR Configuration Guide for details on how to compute the setting time.



DDR memory settling time (us): 200

Memory Settings

Memory Type: DDR2

Data Width: 32

SECDED Enabled ECC □

Address Mapping: {BANK,ROW,COLUMN}

Figure 2-3 • **FDDR Configuration - Partial View**

The FDDR controller must be configured to match the external selected DDR memory standard. The configuration of the FDDR can be defined in a file and the file can be imported using the System Builder via the Import Configuration button. The configuration is executed via the CoreConfigP soft IP core, which is the master of the configuration data initialization process.

Upon reset, the soft IP core CoreConfigP copies the data from embedded nonvolatile memory (eNVM) to the configuration registers of the DDR through the FIC_2 advanced peripheral bus (APB) interface.

For more details on Peripheral Initialization, review the CoreConfigP and CoreResetP application notes available on the Microsemi website.

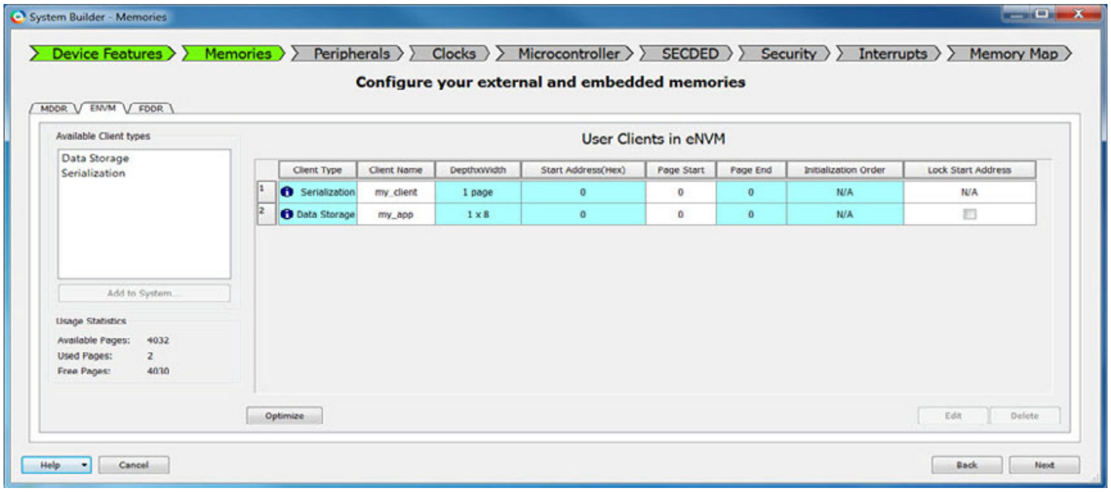
You can also save and export your FDDR Configuration to a file using the Export Configuration button on this page.

Embedded Flash (eNVM)

Embedded Flash gives you the option of specifying data storage clients in the Flash Memory to store your application firmware image or other types of data, such as data co-efficients.

If you want to add data storage clients to the embedded Flash Memory, select MSS On-chip Flash Memory (eNVM) from the Device Features page of System Builder. Proceed to the Memories page and select the ENVM tab to configure it. Select Data Storage or Serialization for the Client type and select Add to System (Figure 2-4). Click **Edit** to modify the configuration of your eNVM client.

Note: The number of available pages in the Flash Memory is device-dependent. In general, the larger the device the more pages are available in the Flash Memory. Refer to your device datasheet for specific information on Flash Memory size.



Configure your external and embedded memories

Available Client types

Data Storage
Serialization

Usage Statistics

Available Pages:	4032
Used Pages:	2
Free Pages:	4030

User Clients in eNVM

	Client Type	Client Name	DepthxWidth	Start Address(Hex)	Page Start	Page End	Initialization Order	Lock Start Address
1	Serialization	my_client	1 page	0	0	0	N/A	N/A
2	Data Storage	my_app	1 x 8	0	0	0	N/A	□

Figure 2-4 **• eNVM Configurator**

Peripherals

The Peripherals page is separated into two parts, a panel of available cores and a table that lists your subsystem.

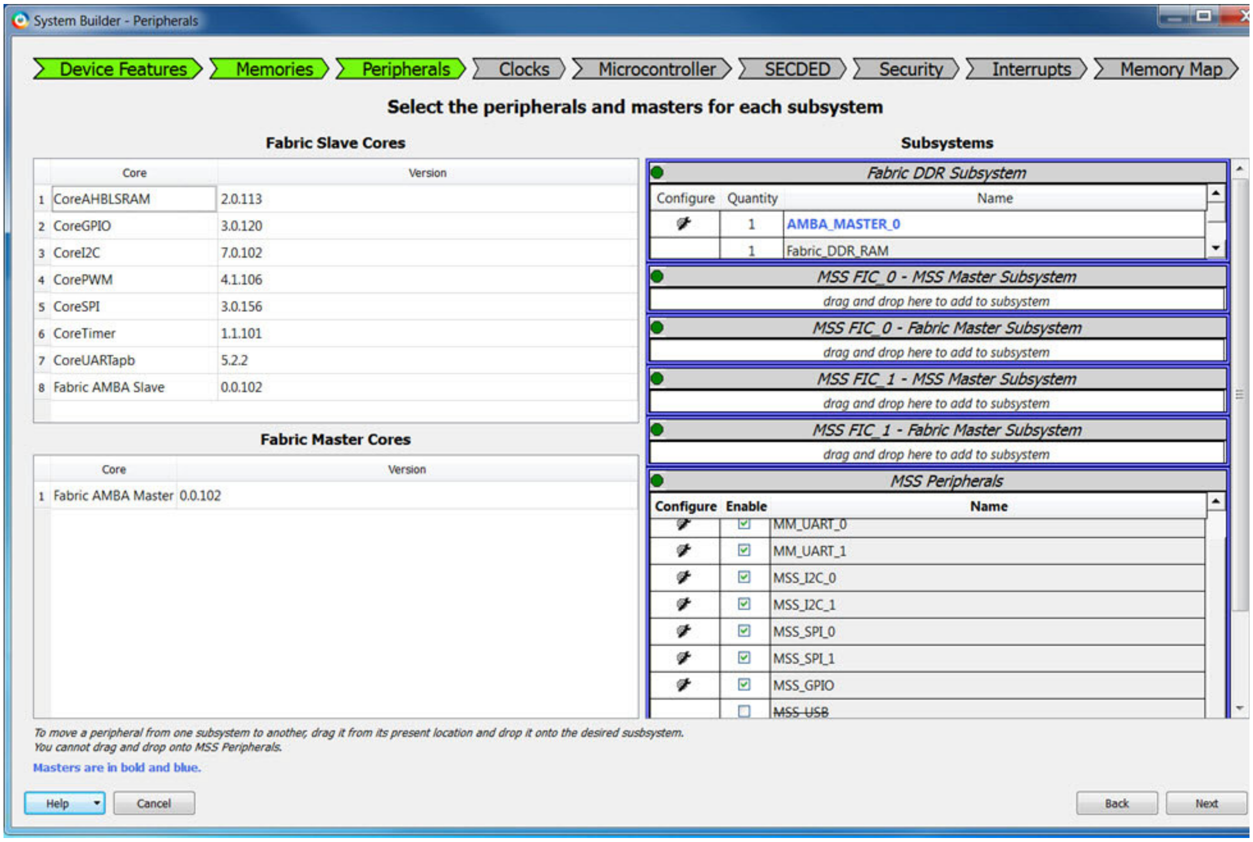
The available cores panel lists the Fabric Slave and Fabric Master cores that you can add to a subsystem.

The Subsystem Table lists the available subsystems based on your previous configurations. You can add/delete/move masters and peripherals around the subsystems to define how you want the various masters and peripherals to communicate.

Special cores in the Fabric Slave and Fabric Master list (Fabric AMBA Slave and Fabric AMBA Master, respectively) enable you to attach any custom peripherals onto the generated design. By instantiating (drag from the available cores panel into a Subsystem in the subsystem panel) and configuring these cores into a subsystem, the correct bus interface pins, clocks, and resets are exposed at the top level System Builder generated design. At the top level SmartDesign, you can then connect these ports to your custom peripheral and they will automatically be visible to the other masters/peripherals in that subsystem.

For certain cores, you can click the Configure icon to configure the options of that core. You can, for example, configure an AMBA Slave to have 0 to 8 interrupts to the MSS. This limit of 8 interrupts is on a per-slave basis. You can also specify the quantity, which determines how many cores with that configuration are instantiated in the generated system.

You can also choose to delete any peripherals that you may have added to a subsystem; right-click and choose **Delete**.



Select the peripherals and masters for each subsystem

Fabric Slave Cores

	Core	Version
1	CoreAHBLSRAM	2.0.113
2	CoreGPIO	3.0.120
3	CoreI2C	7.0.102
4	CorePWM	4.1.106
5	CoreSPI	3.0.156
6	CoreTimer	1.1.101
7	CoreUARTapb	5.2.2
8	Fabric AMBA Slave	0.0.102

Fabric Master Cores

	Core	Version
1	Fabric AMBA Master	0.0.102

Subsystems

Fabric DDR Subsystem

Configure	Quantity	Name
	1	AMBA_MASTER_0
	1	Fabric_DDR_RAM

MSS FIC_0 - MSS Master Subsystem

drag and drop here to add to subsystem

MSS FIC_0 - Fabric Master Subsystem

drag and drop here to add to subsystem

MSS FIC_1 - MSS Master Subsystem

drag and drop here to add to subsystem

MSS FIC_1 - Fabric Master Subsystem

drag and drop here to add to subsystem

MSS Peripherals

Configure	Enable	Name
	☑	MM_UART_0
	☑	MM_UART_1
	☑	MSS_I2C_0
	☑	MSS_I2C_1
	☑	MSS_SPI_0
	☑	MSS_SPI_1
	☑	MSS_GPIO
	□	~~MSS-USB~~

To move a peripheral from one subsystem to another, drag it from its present location and drop it onto the desired susbsystem.
You cannot drag and drop onto MSS Peripherals.

Masters are in bold and blue.

Figure 2-5 **• Select Peripherals - Fabric Slave, Fabric Master and Subsystems**

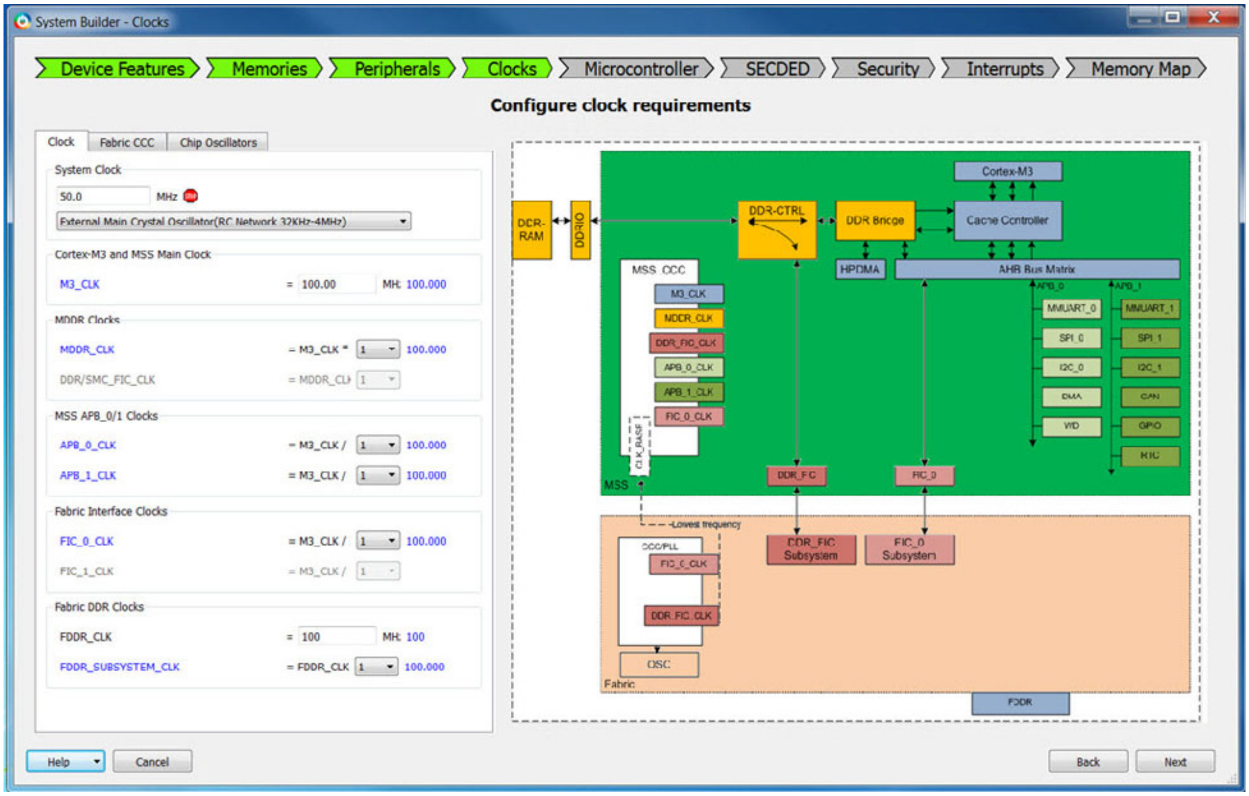
Clocks

The Clocks page enables you to enter and configure clock parameters for your systems. There are three tabs in the Clocks page:

- Clock
- Fabric CCC
- Chip Oscillators

Clock Tab

The Clock tab (Figure 2-6) enables you to select the clock source and the clock frequency to drive each of your subsystems. System Builder automatically instantiates and configures the required PLL (if needed).



System Builder - Clocks

Configure clock requirements

Clock | Fabric CCC | Chip Oscillators

System Clock

50.0 MHz

External Main Crystal Oscillator(RC Network 32KHz-4MHz)

Cortex-M3 and MSS Main Clock

M3_CLK = 100.00 MH 100.000

MDDR Clocks

MDDR_CLK = M3_CLK * 1 100.000

DDR/SMC_FIC_CLK = MDDR_CLK 1

MSS APB_0/1 Clocks

APB_0_CLK = M3_CLK / 1 100.000

APB_1_CLK = M3_CLK / 1 100.000

Fabric Interface Clocks

FIC_0_CLK = M3_CLK / 1 100.000

FIC_1_CLK = M3_CLK / 1

Fabric DDR Clocks

FDDR_CLK = 100 MH 100

FDDR_SUBSYSTEM_CLK = FDDR_CLK 1 100.000

Figure 2-6 **• Clocks - Clock Tab**

The following controls enable you to specify the System Clock. Your clock source options include:

- FPGA Fabric Input
- Dedicated Input Pad
- On-chip 1 MHz RC Oscillator
- On-chip 25/50 MHz RC Oscillator
- External Main Crystal Oscillator (Ceramic Resonator 0.5MHz to 20MHz)
- External Main Crystal Oscillator (Crystal 32KHz - 20MHz)
- External Main Crystal Oscillator (RC Network 32KHz - 4MHz)

Your clock source options determine what frequency you can specify. The other clocks in the system are derived from the System Clock setting. The tool enforces these dependency rules. Refer to the CCC Configuration Guide for more detailed information on Clock Settings.

Cortex-M3 and MSS Main Clock - Specifies how fast you want your Cortex-M3 to run.

MDDR Clocks - Consists of two clock domains: MDDR_CLK and DDR/SMC_FICCLK. The MDDR_CLK specifies how fast your MSS DDR RAM runs. The DDR/SMC_FIC_CLK option is enabled if you have selected the Single data rate memory from MSS option in the Memory page. In that case you can also specify the DDR/SMC_FIC_CLK frequency.

MSS APB_0/1 Clocks - The APB_0_CLK and APB_1_CLK is the frequency of the MSS Peripherals on the APB_0 and APB_1 busses.

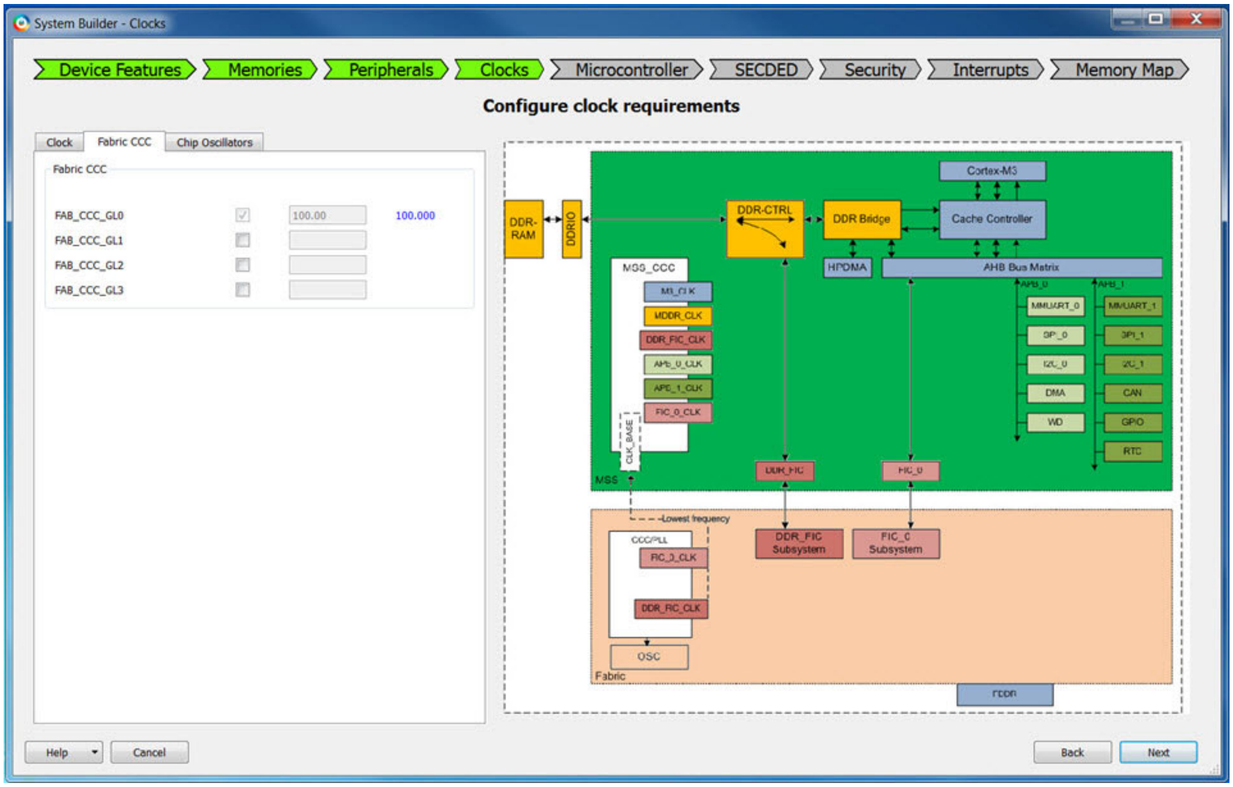
Fabric Interface Clocks - The FIC_0_CLK is the frequency at which the MSS FIC_0 subsystems operates. The FIC_1_CLK is the frequency at which the MSS FIC_1 subsystems will operate. If the FIC clocks are grayed out means you have not added any peripherals to those subsystems in the Peripherals page and there is no need to configure them.

Fabric DDR Clocks - The FDDR_CLK specifies how fast your Fabric DDR RAM runs. This refers to the I/O side of the DDR. The FDDR_SUBSYSTEM_CLK is the frequency at which your fabric logic interfaces with the Fabric DDR controller. If FDDR is grayed out, you are not using it in your design.

Fabric CCC Tab

The Fabric CCC tab (Figure 2-7) enables you to select additional fabric CCC resources not used by any of your configured subsystems. These Fabric CCC clocks are grouped under FAB_CCC_PINS group and promoted to the top level of the System Builder block. Configure these clocks and use them to drive your own fabric logic. Depending on what subsystems you have configured, there may be up to four additional FAB_CCC clocks:

- FAB_CCC_GL0
- FAB_CCC_GL1
- FAB_CCC_GL2
- FAB_CCC_GL3



System Builder - Clocks

Configure clock requirements

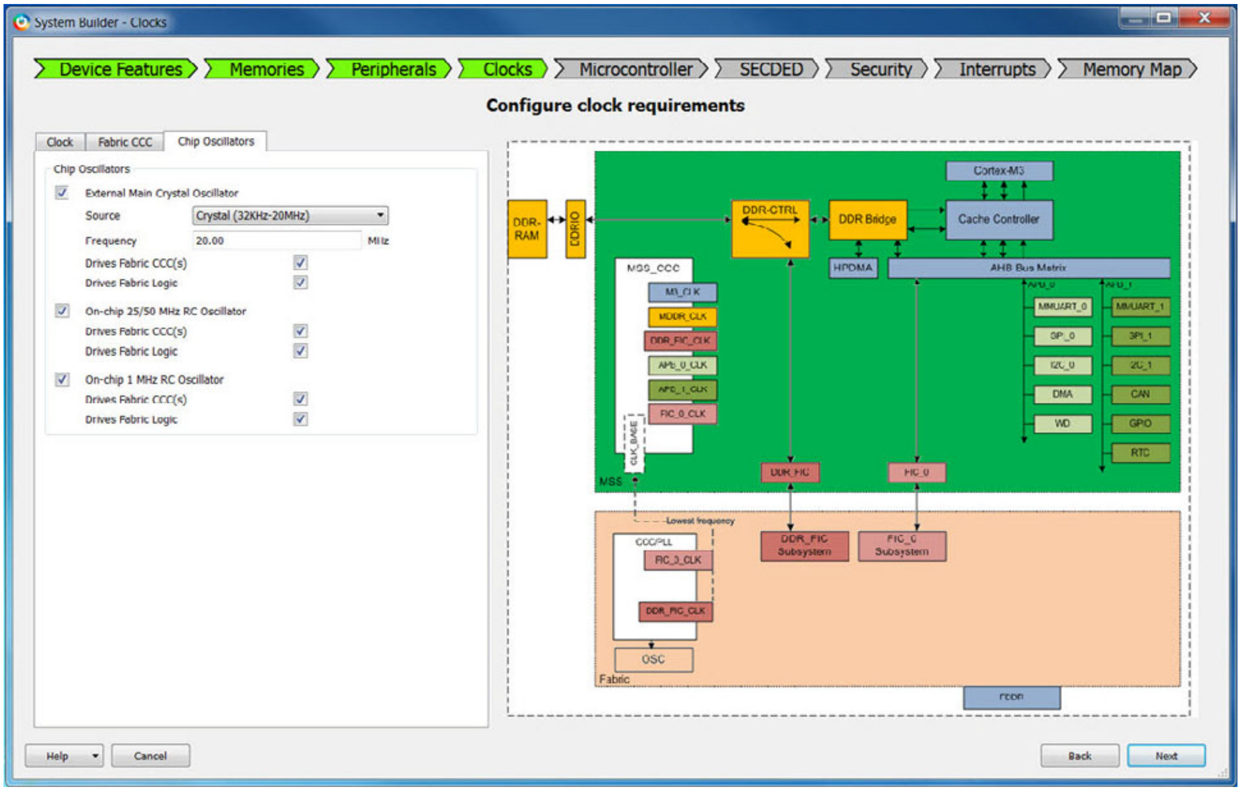
Fabric CCC

FAB_CCC_GL0	☑	100.00	100.000
FAB_CCC_GL1	☐		
FAB_CCC_GL2	☐		
FAB_CCC_GL3	☐		

Figure 2-7 • **Clocks - Fabric CCC Tab**

Chip Oscillators Tab

The Chip Oscillators tab shows all the physical clock oscillator resources (Figure 2-8). If you want to use a Fabric Oscillator to drive the Fabric CCC or the Fabric Logic, check the appropriate checkbox. System Builder exposes/promotes the appropriate pins to the top level under the CHIP_OSC_PINS group.



Configure clock requirements

Chip Oscillators

☑ External Main Crystal Oscillator

Source: Crystal (32KHz-20MHz)

Frequency: 20.00 MHz

Drives Fabric CCC(s) ☑

Drives Fabric Logic ☑

☑ On-chip 25/50 MHz RC Oscillator

Drives Fabric CCC(s) ☑

Drives Fabric Logic ☑

☑ On-chip 1 MHz RC Oscillator

Drives Fabric CCC(s) ☑

Drives Fabric Logic ☑

***Figure 2-8* • Clocks - Chip Oscillators Tab**

Note: In the Clock tab of the Clocks page, if the System Clock's source is configured to be any of the External Main Crystal Oscillator (Ceramic/Crystal/RC Network), the Source and Frequency fields under the External Main Crystal Oscillator section in the Chip Oscillators tab are automatically configured and are grayed out. You can check the Drives Fabric CCC(s) and Drive Fabric Logic checkboxes to expose the corresponding pins.

Fabric Interface Controllers (FICs) and Subsystems

The SmartFusion2 Microcontroller Sub-System (MSS) offers five different Fabric Interface Controllers (FICs):

- DDR_FIC
- SMC_FIC
- FIC_0
- FIC_1 (Available on bigger devices only)
- FIC_2

These interface blocks enable the MSS to interface with logic implemented in the FPGA fabric and vice versa.

The DDR_FIC is used when you configure the MSS DDR block (MDDR) such that the external DDR memory can be accessed from an FPGA fabric master via an AXI interface or 2 AHBLite AMBA interfaces.

The SMC_FIC is used when you configure the MSS DDR Block in the Single Date Rate (SDR) mode. In this configuration, the MSS accesses external Single Data Rate DRAM or Asynchronous memories via a soft memory controller instantiated in the FPGA fabric such as CoreSDR_AXI and CoreSDR_AHB. The SMC_FIC is an AXI or AHBLite slave AMBA interface. The DDR_FIC and SMC_FIC interfaces are mutually exclusive; only one is active at a time.

The FIC interfaces enable you to naturally extend the MSS AMBA Bus into the FPGA fabric. There are up to two FIC instances per MSS depending on the selected device. The first instance is named FIC_0 (which is available on every device) and the second instance is named FIC_1 (may not be present in smaller devices).

You can configure the FIC as either an APB3 or AHBLite AMBA interface, depending on your design needs. In each mode, a master and a slave bus interface is available. That is, a master in the fabric can interface to a slave in the MSS and a master in the MSS can interface to a slave in the fabric.

FIC_2 provides the APB Configuration Path from the MSS to the DDR and SERDES subsystems.

Figure 2-9 shows the MSS and Fabric Interface Controllers.

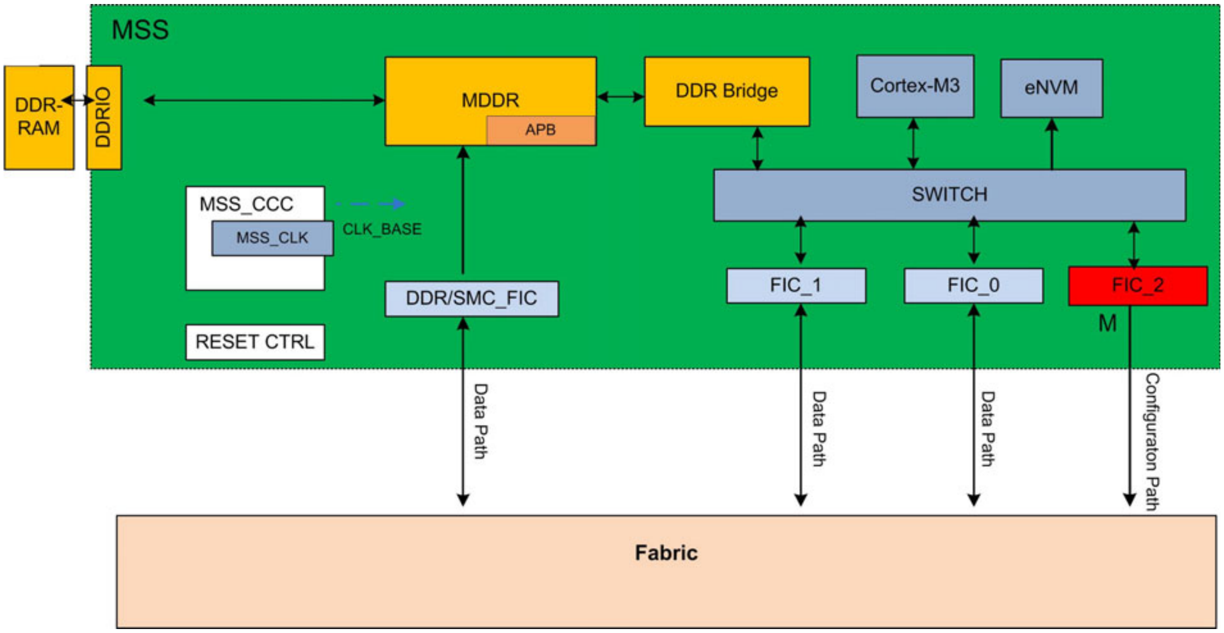


Figure 2-9 • **MSS and Fabric Interface Controllers**

Each Fabric Interface Subsystem can operate on a different clock frequency, defined as a ratio of the MSS main clock M3_CLK. The SmartFusion2 architecture imposes a certain number of rules related to clocking domains between the Fabric Interfaces and the FPGA Fabric. System Builder accounts for these rules and generates a correct-by-construction design based on your inputs. The following subsections describe the different types of MSS-Fabric subsystems supported by System Builder.

SmartFusion2 Design Subsystems

MSS FIC_0 - MSS Master Subsystem

This subsystem is available by default. It enables a MSS Master to access the MSS memory space through the Fabric Interface Controller FIC_0 (Table 2-1).

Figure 2-10 shows the MSS FIC_0 - MSS Master Subsystem.

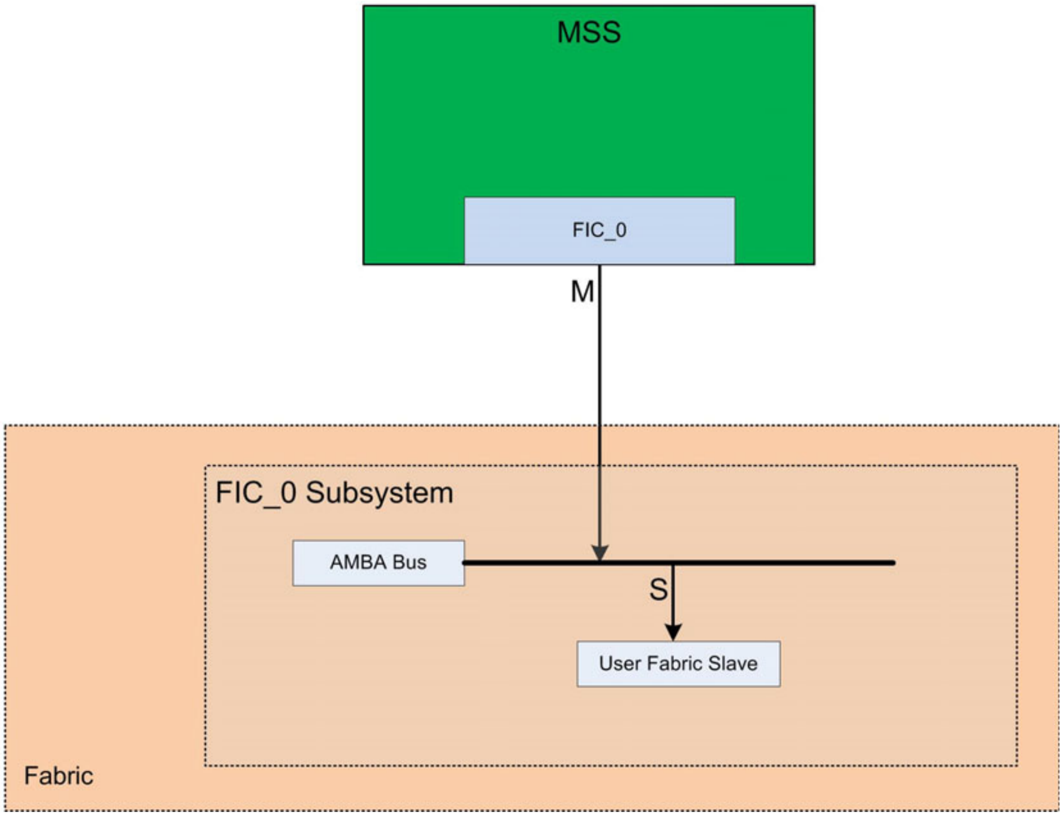


Figure 2-10 **• MSS FIC_0 - MSS Master Subsystem**

Table 2-1 **• MSS FIC_0 - MSS Master Subsystem Port List**

Pin/BIF	Direction	Functionality and Usage
System Builder Slave BIF	Slave	Connect to User Fabric Slave's AMBA Slave BIF. For each User Fabric Slave, connect its Slave BIF to the corresponding Slave BIF on the System Builder block.
FIC_0_CLK	Out	FIC_0 subsystem clock. Connect to User Fabric Slaves
FIC_0_LOCK	Out	Asserts when FIC_0_CLK is valid. If your User Fabric Slave has a PLL LOCK, connect it to this pin.

MSS FIC_0 - Fabric Master Subsystem

This subsystem is available by default. It enables a Fabric Master to access the MSS memory space through FIC_0 in addition to other Fabric peripherals (Table 2-2). If you are using this subsystem you must drag a Fabric AMBA Master into this subsystem and then connect your AMBA peripheral in the resulting design. The subsystem supports up to four AHBLite fabric masters.

Figure 2-11 shows the MSS FIC_0 - Fabric Master Subsystem.

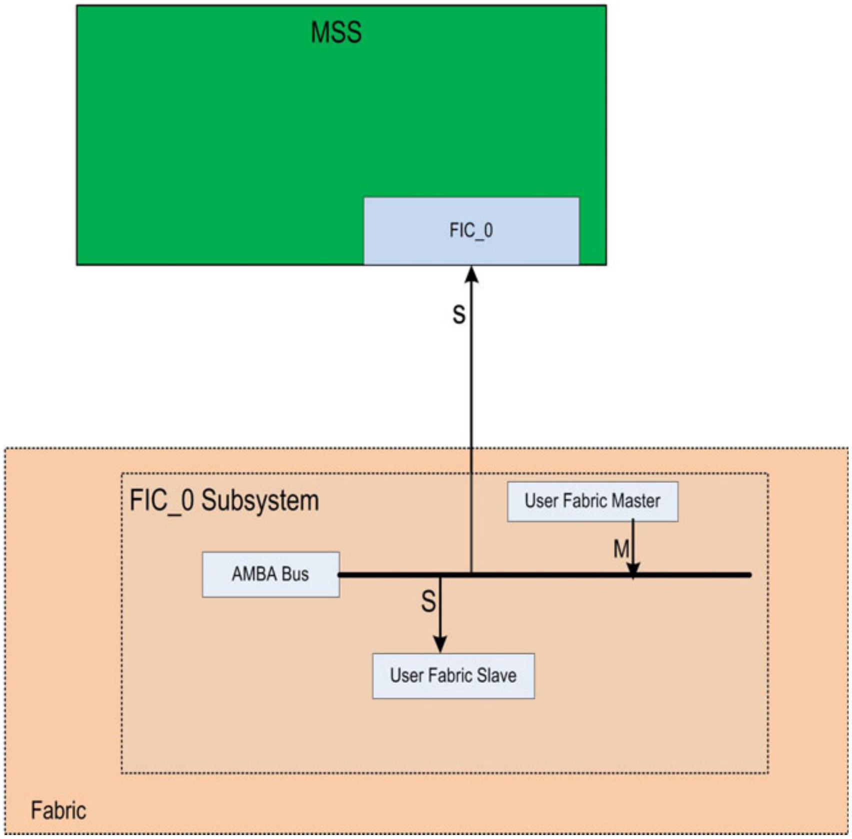


Figure 2-11 **• MSS FIC_0 - Fabric Master Subsystem**

Table 2-2 **• MSS FIC_0 - Fabric Master Subsystem Port List**

Pin/BIF	**Direction**	**Functionality and Usage**
System Builder Master BIF (AHBLite/APB3 BIF)	Master	Connect to User Fabric Master's AHBLite/APB3 Master BIF
System Builder Slave BIF	Slave	Connect to User Fabric Slave's AMBA Slave BIF. For each User Fabric Slave, connect its Slave BIF to the corresponding Slave BIF on the System Builder block.
FIC_0_CLK	Out	FIC_0 subsystem clock. Connect to User Fabric Master.
FIC_0_LOCK	Out	Asserts when FIC_0_CLK is valid. If your User Fabric Master/ slave has a PLL LOCK input, connect it to this pin.

MSS FIC_1 MSS Master Subsystem - M2S050 and Larger Devices

This subsystem is available by default. It enables an MSS Master to access the MSS memory space through FIC_1 (Table 2-3).

Figure 2-12 shows the MSS FIC_1 - MSS Master Subsystem.

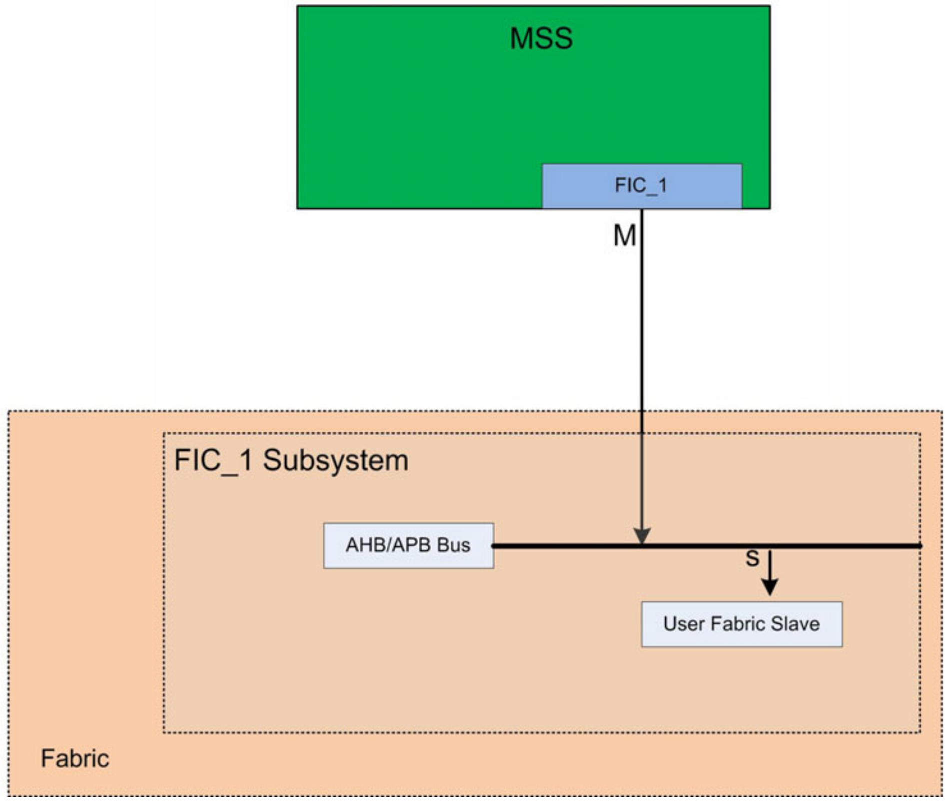


Figure 2-12 **• MSS FIC_1 - MSS Master Subsystem**

Table 2-3 **• MSS FIC_1 MSS Master Subsystem Port List**

Pin/BIF	**Direction**	**Functionality and Usage**
System Builder Slave BIF	Slave	Connect to User Fabric Slave's AMBA Slave BIF. For each User Fabric Slave, connect its Slave BIF to the corresponding Slave BIF on the System Builder block.
FIC_1_CLK	Out	FIC_1 subsystem clock. Connect to User Fabric Slaves
FIC_1_LOCK	Out	Asserts when FIC_1_CLK is valid. If your User Fabric Slave has a PLL LOCK input, connect it to this pin.

MSS FIC_1 - Fabric Master Subsystem - M2S050 and Larger Devices

This subsystem is available by default. It enables a Fabric Master to access the MSS memory space through FIC_1 in addition to other Fabric peripherals (Table 2-4). If you are using this subsystem you must drag a Fabric AMBA Master into this subsystem and then connect your AMBA peripheral in the resultant design. The subsystem supports up to four AHBLite fabric masters.

Figure 2-13 shows the MSS FIC_1 - Fabric Master Subsystem.

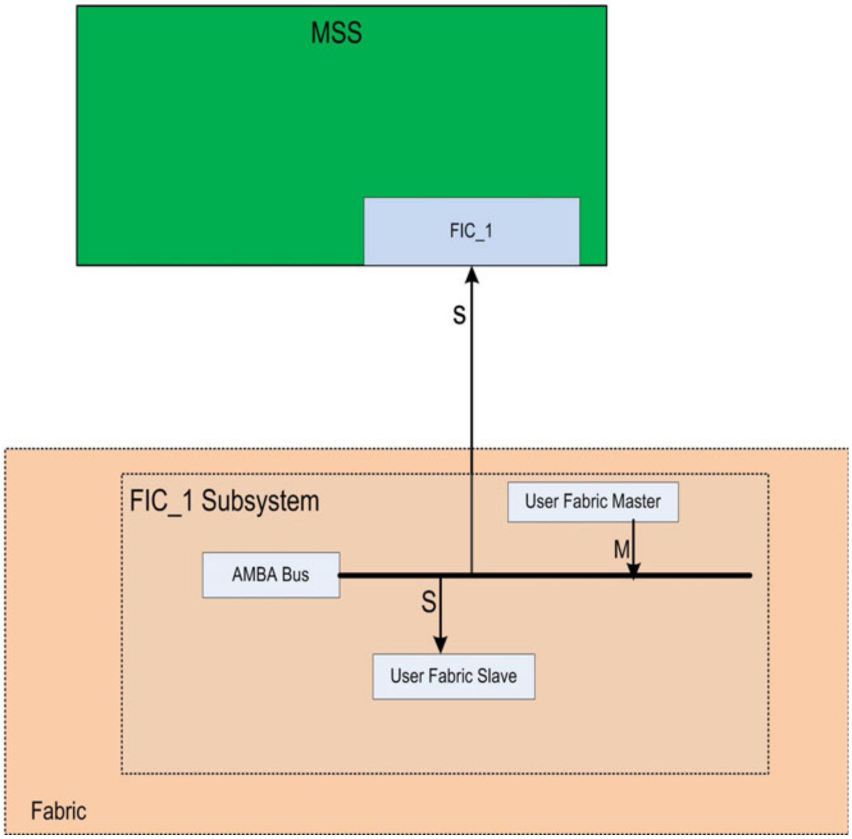


Figure 2-13 **• MSS FIC_1 - Fabric Master Subsystem**

Table 2-4 **• MSS FIC_1 - Fabric Master Subsystem Port List**

Pin/BIF	Direction	Functionality and Usage
System Builder Master BIF (AHBLite/ APB3 BIF)	Master	Connect to User Fabric Master's AHBLite/APB3 Master BIF.
System Builder Slave BIF	Slave	Connect to User Fabric Slave's AMBA Slave BIF. For each User Fabric Slave, connect its Slave BIF to the corresponding Slave BIF on the System Builder block.
FIC_1_CLK	Out	FIC_1 subsystem clock. Connect to User Fabric Master.
FIC_1_LOCK	Out	Asserts when FIC_1_CLK is valid. If your User Fabric Master/ CoreSysServices has a PLL LOCK input, connect it to this pin.

MSS DDR FIC Subsystem

This subsystem is available if you check the **MSS External Memory** checkbox and the **MDDR** radio button on the Device Features page (Table 2-5).

When available, the MSS_DDR_RAM peripheral displays automatically. The Cortex-M3 is a master on this subsystem.

Optionally, you may choose to have a Fabric Master access this DDR by instantiating a Fabric AMBA Master core into this subsystem (drag Fabric AMBA Master and drop it in the MSS DDR FIC Subsystem to instantiate). Refer to the MDDR Configuration Guide for more information.

Figure 2-14 shows the MDDR Fabric Master Subsystem.

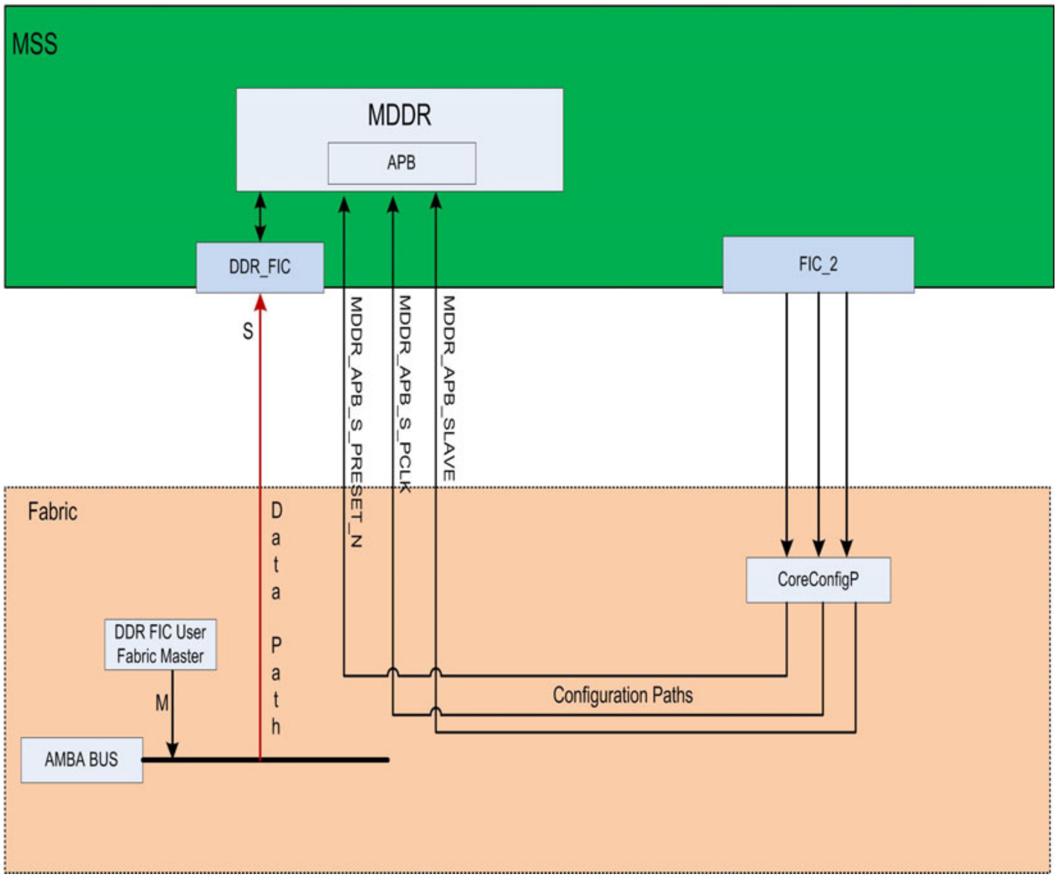


Figure 2-14 **• MDDR Fabric Master Subsystem**

Table 2-5 **• MSS DDR FIC Subsystem Port List**

Pin/BIF	**Direction**	**Functionality and Usage**
System Builder Master (AHBLite/AXI) BIF	Master	Connect to User Fabric Master's AMBA Master BIF.
System Builder Slave (AHBLite/AXI) BIF	Slave	Connect to User Fabric Master's AMBA Slave BIF.
MSS_DDR_FIC_SUBSYSTEM_PINS: MSS_DDR_FIC_SUBSYSTEM_CLK	Out	MSS DDR subsystem clock. Connect to User Fabric Master and User Fabric Slaves in this subsystem.
MSS_DDR_FIC_SUBSYSTEM_PINS: MSS_DDR_FIC_SUBSYSTEM_LOCK	Out	Asserts when MSS_DDR_FIC_SUBSYSTEM_CLK is valid. If your User Fabric Master/User Fabric Slaves have PLL LOCK inputs, connect them to this pin.

Table 2-5 • **MSS DDR FIC Subsystem Port List (continued)**

Pin/BIF	Direction	Functionality and Usage
MSS_DDR_FIC_SUBSYSTEM_PINS: MDDR_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. Tie low if not used.
MDDR_PADS	In/Out	DDR Memory Physical Interface PADs. Consult the SmartFusion2 DDR Controller Configuration document for details.

MSS SMC FIC Subsystem

This subsystem is available if you check the **MSS External Memory** checkbox and select the **Soft Memory Controller (SMC)** radio button on the Device Features page. In this configuration, the MSS accesses external Single Data Rate DRAM or Asynchronous memories via a soft memory controller instantiated in the FPGA fabric, such as CoreSDR_AXI and CoreSDR_AHB The Cortex-M3 is a master on the MSS SMC FIC subsystem.

Figure 2-15 shows the SMC_FIC Subsystem.

.

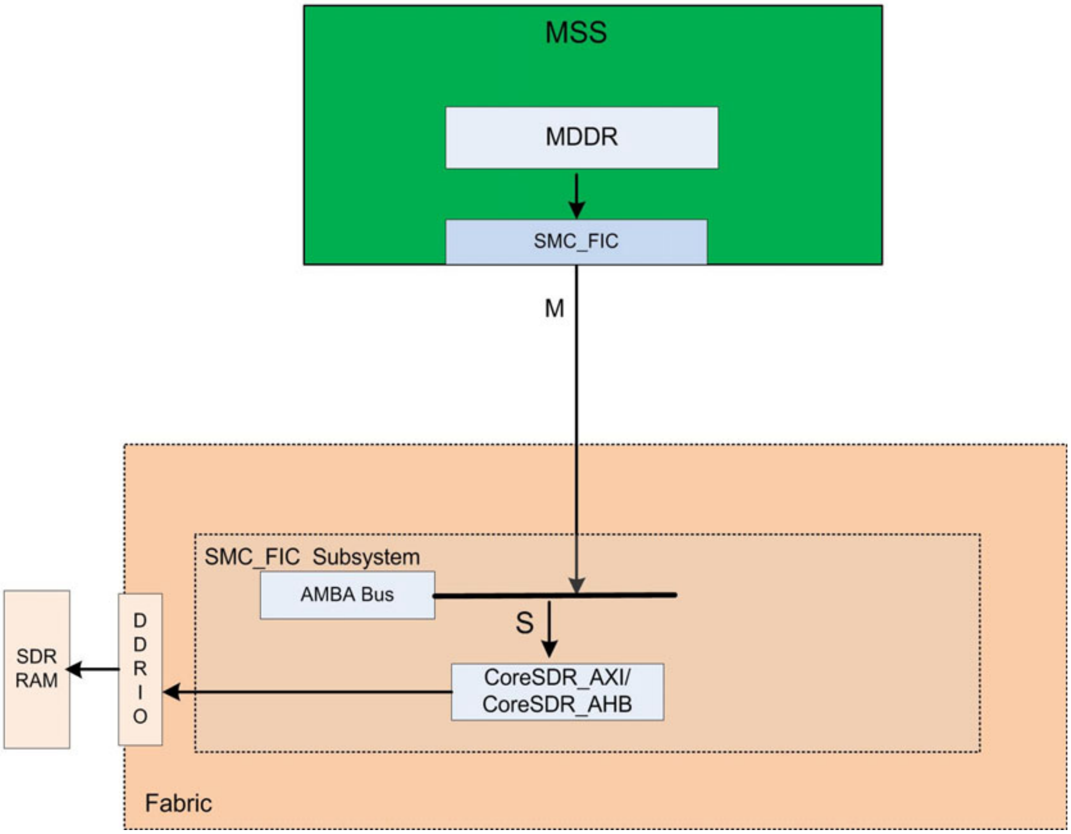


Figure 2-15 • **SMC_FIC Subsystem**

The DDR FIC and the SMC FIC Subsystems are mutually exclusive. You can have one or the other but you cannot enable both. Table 2-6 shows the port list.

Table 2-6 • **MSS DDR FIC Subsystem Port List**

Pin/BIF	Direction	Functionality and Usage
MSS_SMC_0_PINS	Pin Group	Connect to non-DDR based External Memory

Fabric DDR Subsystem

This subsystem is available if you check the Fabric External DDR Memory (FDDR) checkbox on the Device Features page. When available the Fabric_DDR_RAM peripheral is automatically shown in this subsystem. Fabric_DDR_RAM can be accessed in two ways:

- If you intend to have a Fabric Master access the Fabric DDR, click and drag a Fabric AMBA Master core into this subsystem from the available cores list.
- If you intend to have the Cortex-M3 access the Fabric DDR, click and drag the Fabric_DDR_RAM peripheral into one of the MSS FIC_<0/1> subsystems.

Figure 2-16 shows the FDDR Subsystem with a Fabric Master.

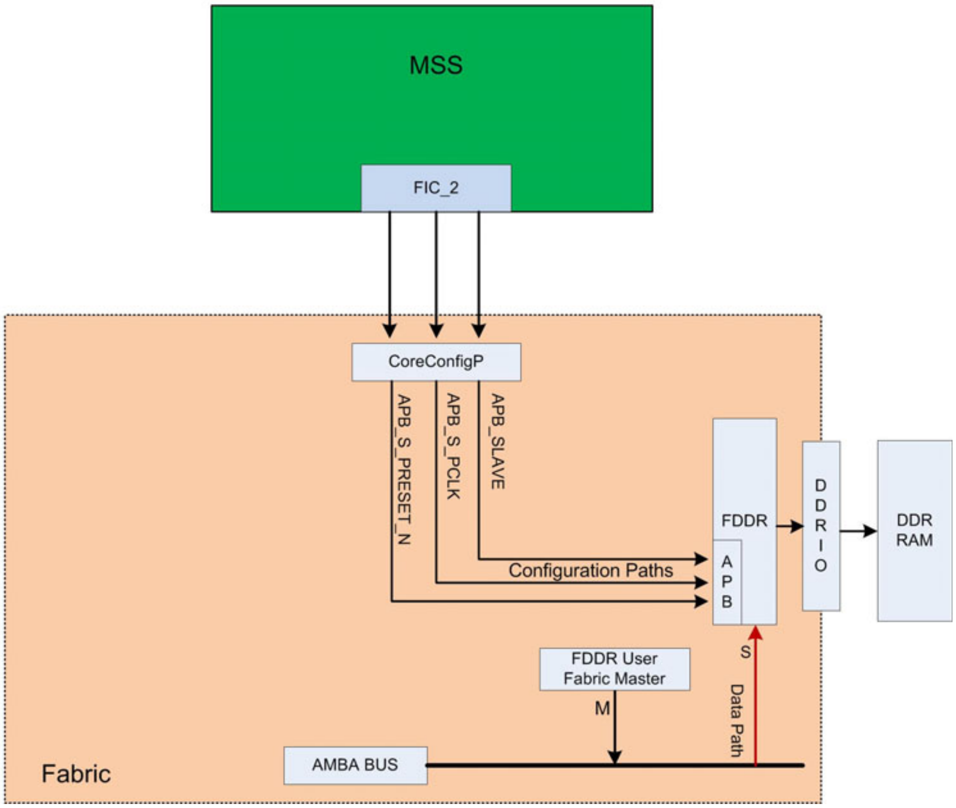


Figure 2-16 • **FDDR Subsystem with a Fabric Master**

Table 2-7 lists the Fabric DDR Subsystem Ports.

***Table 2-7* • Fabric DDR Subsystem Port List**

Pin/BIF	Direction	Functionality and Usage
System Builder Master (AHBLite /AXI) BIF	Master	Connect to User Fabric Master's AMBA Master BIF.
System Builder Slave (AHBLite/AXI) BIF	Slave	Connect to User Fabric Master's AMBA Slave BIF.

Table 2-7 **• Fabric DDR Subsystem Port List (continued)**

Pin/BIF	**Direction**	**Functionality and Usage**
FDDR_SUBSYSTEM_PINS: FDDR_SUBSYSTEM_CLK	In	Fabric DDR subsystem clock. You must provide a clock for your Fabric DDR Subsystem using one of the following options: Instantiate Fabric CCC in SmartDesign to generate the clock for the FDDR subsystem. The Fabric CCC generates a clock and PLL lock signals. Connect Fabric CCC's Clock output (GL0/1/2/3) to: - FDDR_SUBSYSTEM_CLK - User Fabric Master's Clock Input - User Fabric Slaves' Clock Inputs If you want to use the same clock frequency as one of the other subsystems (FIC_0/1 or MSS DDR), instead of instantiating a Fabric CCC, you can get the appropriate clock and lock signals from your chosen subsystem's pins. Connect your chosen clock signal (one of FIC_0_CLK, FIC_1_CLK, or MSS_DDR_FIC_SUBSYSTEM_CLK) to: - FDDR_SUBSYSTEM_CLK - User Fabric Master's Clock Input - User Fabric Slaves' Clock Inputs
FDDR_SUBSYSTEM_PINS: FDDR_SUBSYSTEM_CLK_PLL_LOCK	In	Asserts when FDDR_SUBSYSTEM_CLK is valid. Depending on your selection of the driver of FDDR_SUBSYSTEM_CLK, connect to one of the following: - Fabric CCC:LOCK - System Builder block's FIC_0_LOCK - FIC_1_LOCK - MSS_DDR_FIC_SUBSYSTEM_LOCK
FDDR_SUBSYSTEM_PIN: FDDR_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. Tie low if not used.
FDDR_PADS	In/Out	DDR Memory Physical Interface PADs. Consult the SmartFusion2 DDR Controller Configuration for details

SERDESIF<0/1/2/3> Subsystems and SERDES Configuration Path

This subsystem is available if you check the SERDESIF_<0/1/2/3> checkbox in the Device Features page of System Builder.

Based on your selection in the Device Features page, System Builder automatically generates the SERDES configuration data bus interface and the clock and reset for the configuration data bus interface.

Figure 2-17 shows the SERDESIF_<0/1/2/3> Subsystem.

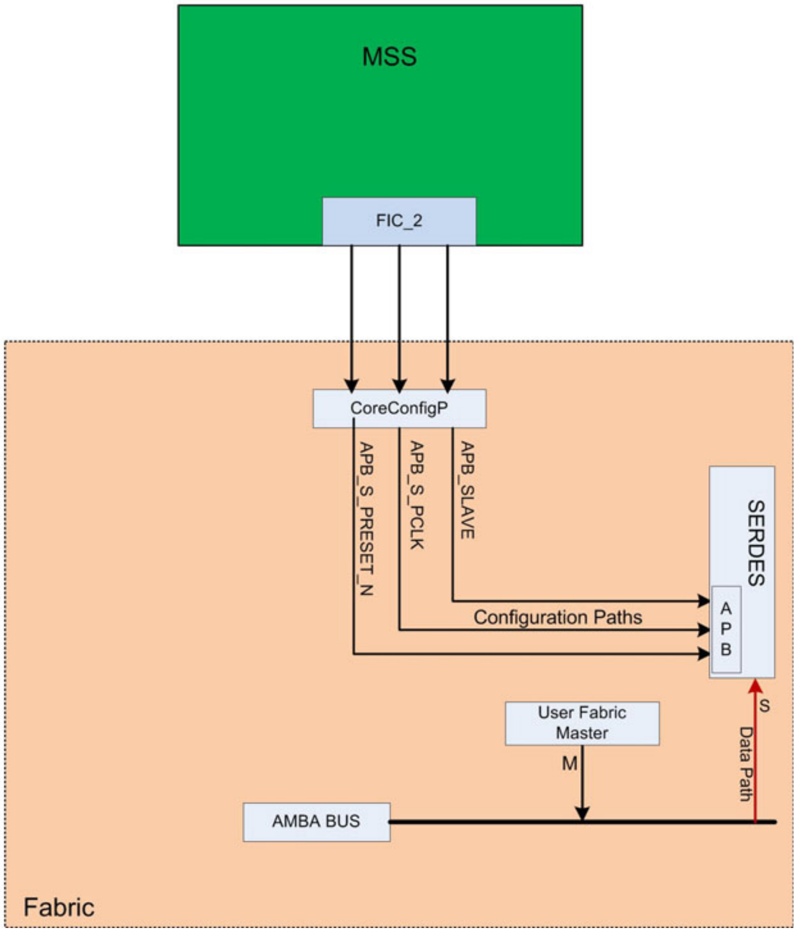
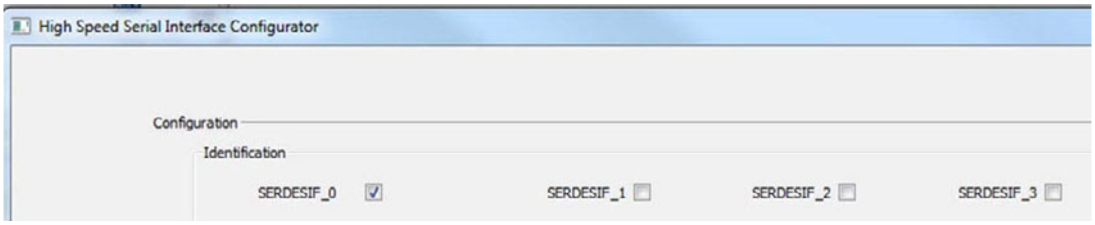


Figure 2-17 **• SERDESIF_<0/1/2/3> Subsystem**

Instantiate the SERDES blocks in SmartDesign and connect their reset and configuration pins and BIFs to System Builder as shown in Table 2-8.

The definitive name of a SERDES instance (SERDES_0/1/2/3) is visible in its configurator (Figure 2-18). Correlate this with the correct SDIF pins (Table 2-8) on the System Builder block. For example, connect pins in the SDIF_0_PINS group to SERDESIF_0, SDIF_1_PINS to SERDES_1, etc.


High Speed Serial Interface Configurator

Configuration

Identification

SERDESIF_0 ☑ SERDESIF_1 □ SERDESIF_2 □ SERDESIF_3 □

Figure 2-18 **• SERDES Instance Identification (Partial View)**

Table 2-8 **• SERDES Configuration Port List**

Pin/BIF	Direction	Functionality and Usage
SDIF(0-3)_INIT_APB BIF	Slave	For each SERDES block, System Builder generates an APB Slave BIF. These appear at the bottom of the System Builder block, and are called SDIF(0-3)_ INIT_APB. Connect the APB Slave BIF on each SERDES block to its corresponding SDIF(0-3)_APB_Slave BIF on the System Builder block.
INIT_PINS:INIT_APB_S_PCLK	Out	SERDES APB Configuration Path Clock. Connect to the INIT_APB_S_PCLK input port of all SERDES instances
INIT_PINS:INIT_APB_S_PRESET_N	Out	SERDES APB Configuration Path Reset. Connect to the INIT_APB_S_PRESET_N port of all SERDES instances
SDIF(0-3)_PINS: SDIF(0-3)_PERST_N	In	L2 and P2 are low power states for the Link and PHY interface in a PCI Express (PCIe) system. A power management component in a PCIe system controls the exit from the L2/P2 state. Part of the sequence when emerging from the low power state involves assertion and release of the PCI Express Reset (SDIF(0-3)_PERST_N pins). If you are not using PCIe, tie this pin high.
SDIF(0-3)_PINS: SDIF_(0-3)_PHY_RESET_N	Out	Deasserts to bring SERDES PHY interface out of reset. Connect to the PHY_RESET_N reset input of the corresponding SERDES instance
SDIF(0-3)_PINS: SDIF_(0-3)_CORE_RESET_N	Out	Deasserts to bring SERDES Core Logic out of reset. Connect to the CORE_RESET_N reset input of the corresponding SERDES instance
SDIF(0-3)_PINS: SDIF_(0-3)_SPLL_LOCK	In	Asserts when the SERDES instance's internal PLL has locked. Connect to the SPLL_LOCK output pin of the corresponding SERDES instance

Reset Pins

The System Builder block receives and generates reset signals that can be connected to your user logic or board. Depending on your design, you may see the reset pins shown in Table 2-9.

Table 2-9 **• Reset Pins**

Pin	Direction	Functionality and Usage
DEVRST_N (PAD)	In	FPGA Power ON Reset. Deasserts when FPGA is powered up for the first time. This active-low reset performs the same function as a power-up reset. This is an optional function and can safely be tied high in your test bench. When used, this pin must be connected to the dedicated top level port DEVRST_N.
FAB_RESET_N	In	Fabric to System Builder block reset. Can be used by your Fabric logic to reset the system (MSS/SERDES/DDR). This active-low reset performs a re-initialization of all of the peripherals controlled by the MSS. This is an optional function and can safely be tied high.

Table 2-9 **• Reset Pins (continued)**

Pin	Direction	Functionality and Usage
INIT_PINS:INIT_DONE	Out	Asserts when System Builder block components are ready for communication and initialization is complete. Connect to any fabric logic that needs to wait for System Builder initialization
POWER_ON_RESET_N	Out	When '0', indicates the system controller is in the process of booting. Goes to '0' just after the VDD ramps past the threshold voltage. When '1', indicates the system controller is finished booting.
MSS_READY	Out	Asserts when the MSS is ready for communication with the fabric logic. Connect to User logic's Reset input. MSS_READY happens before the INIT_DONE pins go high. This output only indicates the MSS FICs can be accessed, not that the entire system is operational. To wait until the entire system is operational, wait for MSS _READY and all of the applicable INIT_DONE pins to go high
DDR_READY	Out	This pin is seen only when one/both of MDDR/FDDR are used in the design. Value "1" indicates that both MDDR and FDDR (if used) blocks are ready for communication and their initialization is complete. Connect to any fabric logic that needs to wait for MDDR/FDDR initialization.
SDIF_READY	Out	This pin is seen only when one/more SERDES blocks are used in the design. Value "1" indicates that all the SERDES blocks (if used) are ready for communication and their initialization is complete. Connect to any fabric logic that needs to wait for SERDES initialization.

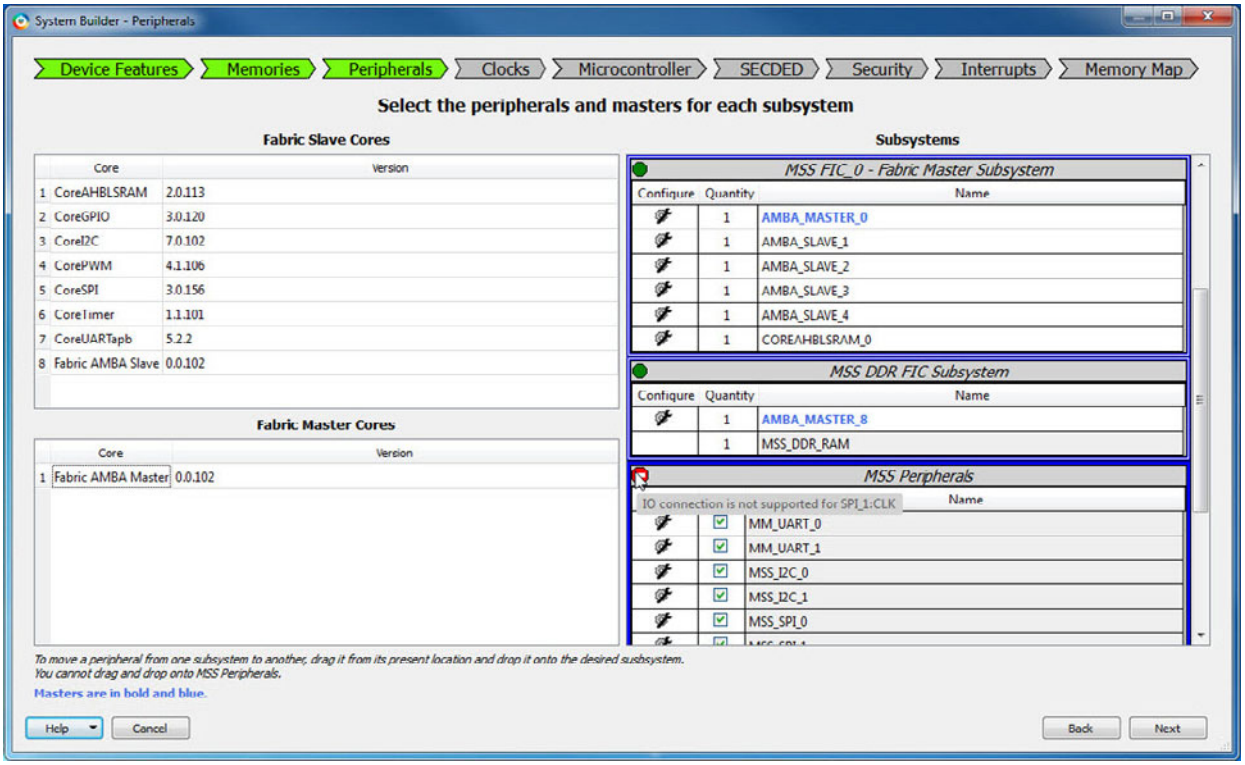
MSS Peripheral Subsystem

This subsystem is available by default. It contains the list of hard peripherals inside the MSS. You can choose to enable/disable/configure each of these peripherals in the System Builder Peripherals page. The hard peripherals include GPIO, MMUART, I2C, SPI, USB, CAN and Ethernet MAC. When these hard peripherals are enabled and configured, corresponding PADS/Fabric pins are exposed at the top level by System Builder when it generates the system.

DRC Checks

System Builder enforces Design Rule Checks (DRC) with regard to device-specific peripheral I/O and fabric availability. When the I/Os and/or fabric interface are not available for the specific peripheral/ die/ package combination, the Peripherals Page of System Builder generates an error and a tooltip message to alert you. You cannot proceed to the next page unless you reconfigure the peripheral appropriately to use supported resources.

In Figure 2-19, the tooltip is generated because the M2S005/144 VQ die/package combination does not support MSS SPI_1 I/Os. You must reconfigure MSS_SPI_1 to use fabric connections instead of I/Os to proceed further.



Select the peripherals and masters for each subsystem

Fabric Slave Cores

	Core	Version
1	CoreAHBLSRAM	2.0.113
2	CoreGPIO	3.0.120
3	CoreI2C	7.0.102
4	CorePWM	4.1.106
5	CoreSPI	3.0.156
6	CoreTimer	1.1.101
7	CoreUARTapb	5.2.2
8	Fabric AMBA Slave	0.0.102

Fabric Master Cores

	Core	Version
1	Fabric AMBA Master	0.0.102

Subsystems

MSS FIC_0 - Fabric Master Subsystem

Configure	Quantity	Name
⚙	1	AMBA_MASTER_0
⚙	1	AMBA_SLAVE_1
⚙	1	AMBA_SLAVE_2
⚙	1	AMBA_SLAVE_3
⚙	1	AMBA_SLAVE_4
⚙	1	COREAHBLSRAM_0

MSS DDR FIC Subsystem

Configure	Quantity	Name
⚙	1	AMBA_MASTER_8
	1	MSS_DDR_RAM

MSS Peripherals

IO connection is not supported for SPI_1:CLK

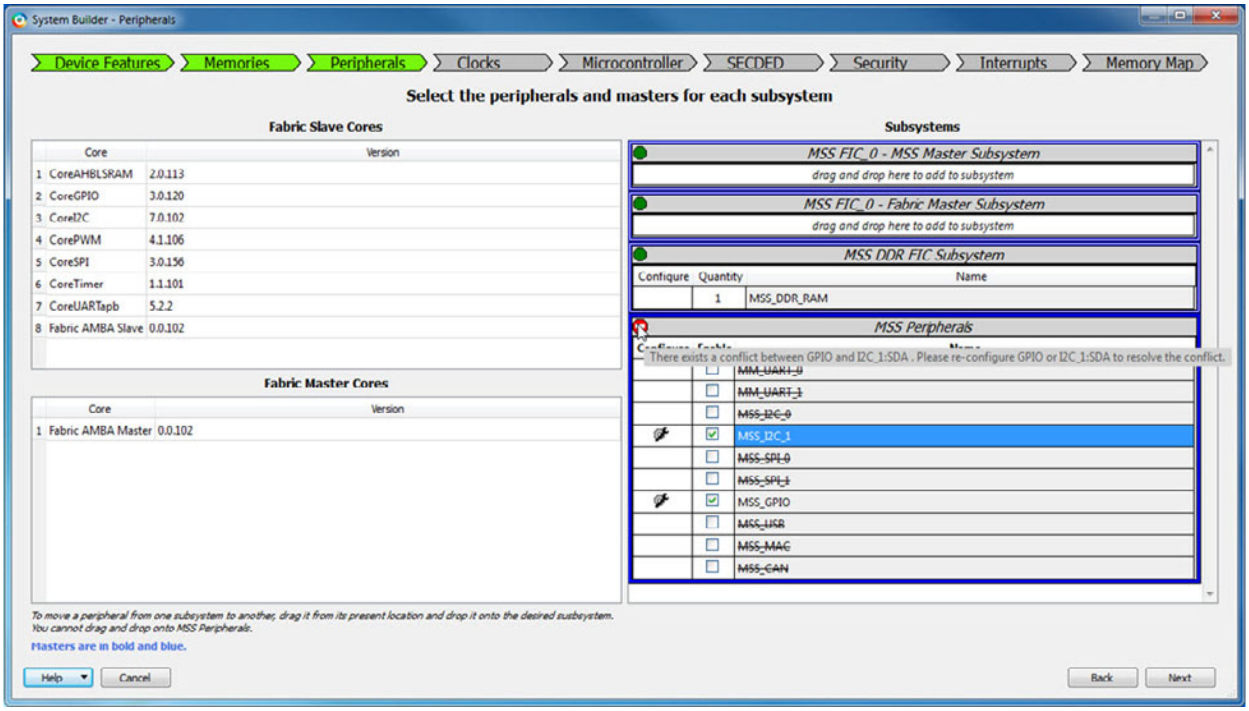
		Name
⚙	☑	MM_UART_0
⚙	☑	MM_UART_1
⚙	☑	MSS_I2C_0
⚙	☑	MSS_I2C_1
⚙	☑	MSS_SPI_0

To move a peripheral from one subsystem to another, drag it from its present location and drop it onto the desired susbsystem.
You cannot drag and drop onto MSS Peripherals.

Masters are in bold and blue.

***Figure 2-19* • DRC Check and Tooltip Information**

When you configure multiple MSS peripherals and there are conflicts of shared resources (I/Os or fabrics) between two or more MSS peripherals, the Peripherals page of System Builder displays DRC error messages on the MSS Peripherals group title. In the example below (Figure 2-20), there is a conflict between MSS Peripheral GPIO and MSS Peripheral I2C. You need to reconfigure either one of the two MSS peripherals to resolve the conflict before you can continue.



Select the peripherals and masters for each subsystem

Fabric Slave Cores

	Core	Version
1	CoreAHBLSRAM	2.0.113
2	CoreGPIO	3.0.120
3	CoreI2C	7.0.102
4	CorePWM	4.1.106
5	CoreSPI	3.0.156
6	CoreTimer	1.1.101
7	CoreUARTapb	5.2.2
8	Fabric AMBA Slave	0.0.102

Fabric Master Cores

	Core	Version
1	Fabric AMBA Master	0.0.102

Subsystems

MSS FIC_0 - MSS Master Subsystem

drag and drop here to add to subsystem

MSS FIC_0 - Fabric Master Subsystem

drag and drop here to add to subsystem

MSS DDR FIC Subsystem

Configure	Quantity	Name
	1	MSS_DDR_RAM

MSS Peripherals

There exists a conflict between GPIO and I2C_1:SDA . Please re-configure GPIO or I2C_1:SDA to resolve the conflict.

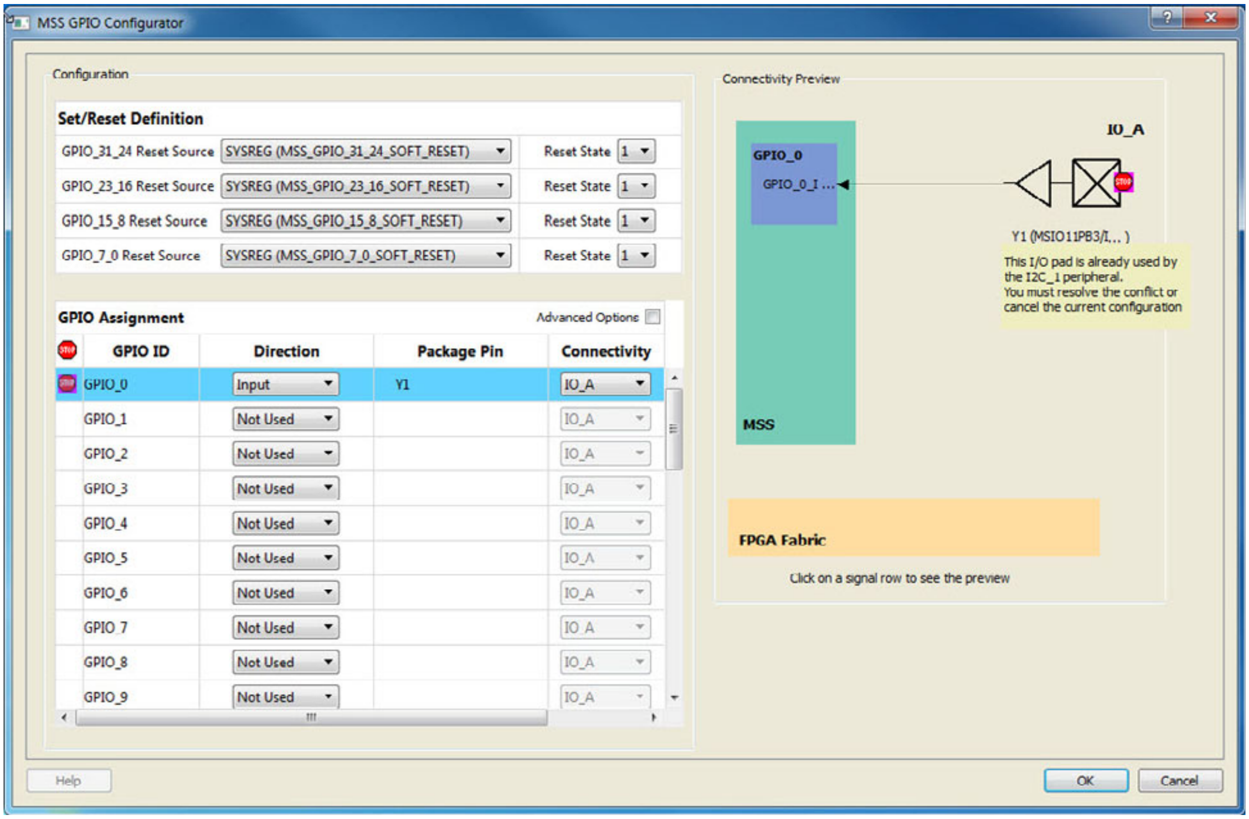
Configure	Enable	Name
	□	~~MM_UART_0~~
	□	~~MM_UART_1~~
	□	~~MSS_I2C_0~~
🔧	☑	MSS_I2C_1
	□	~~MSS_SPI_0~~
	□	~~MSS_SPI_1~~
🔧	☑	MSS_GPIO
	□	~~MSS_USB~~
	□	~~MSS_MAC~~
	□	~~MSS_CAN~~

To move a peripheral from one subsystem to another, drag it from its present location and drop it onto the desired susbsystem. You cannot drag and drop onto MSS Peripherals.

Masters are in bold and blue.

Figure 2-20 • **DRC Error Messages Showing Resource Conflict Between Multiple MSS Peripherals**

If one of the two MSS peripherals in conflict is a GPIO, as in this example, the conflicts (errors and info) are also shown inside the MSS_GPIO configurator (Figure 2-21).



MSS GPIO Configurator

Configuration

Set/Reset Definition

GPIO_31_24 Reset Source	SYSREG (MSS_GPIO_31_24_SOFT_RESET)	Reset State	1
GPIO_23_16 Reset Source	SYSREG (MSS_GPIO_23_16_SOFT_RESET)	Reset State	1
GPIO_15_8 Reset Source	SYSREG (MSS_GPIO_15_8_SOFT_RESET)	Reset State	1
GPIO_7_0 Reset Source	SYSREG (MSS_GPIO_7_0_SOFT_RESET)	Reset State	1

GPIO Assignment

Advanced Options □

	GPIO ID	Direction	Package Pin	Connectivity
	GPIO_0	Input	Y1	IO_A
	GPIO_1	Not Used		IO_A
	GPIO_2	Not Used		IO_A
	GPIO_3	Not Used		IO_A
	GPIO_4	Not Used		IO_A
	GPIO_5	Not Used		IO_A
	GPIO_6	Not Used		IO_A
	GPIO_7	Not Used		IO_A
	GPIO_8	Not Used		IO_A
	GPIO_9	Not Used		IO_A

Connectivity Preview

Y1 (MSIO11PB3/I,..)

This I/O pad is already used by the I2C_1 peripheral.
You must resolve the conflict or cancel the current configuration

Click on a signal row to see the preview

Figure 2-21 **• DRC Conflict Information Displayed in MSS GPIO Configurator**

Configure Microcontroller Options

These pages enable you to configure your Micro-controller options. More information can be found in the configuration guides for the respective microcontrollers.

- Watchdog Timer
- Real Time Counter
- Peripheral DMA
- Cortex-M3
- Cache Controller
- AHB Bus Matrix

Configure SECDED Options

This page enables you to configure your Single Error Correction / Double Error Detection logic. Refer to the SECDED Configuration Guide for more information.

Configure Security Options

The SmartFusion2 devices offer extensive configurable access controls to the MSS memory map. The Security Options page display varies depending your selected device. If enabled you may configure the

Read/Write access of eSRAM, eNVM, FICs and DDR Bridge. It also enables you to restrict access to the MSS memory region for any FPGA fabric master attempting to access the MSS through one of the two Fabric Interface Controllers (FIC).

More information on configuring the security options can be found in SmartFusion2 MSS Security Configuration.

MSS Interrupts

The MSS has 16 dedicated Interrupt lines from the Fabric. When there are 16 or fewer slaves with interrupts, the Interrupt pins of the fabric slaves 0 through 15 are connected to the MSS_INT_F2M[15:0] pins via cascaded OR-gates. In this case, each slave has a dedicated Interrupt access to the MSS. (Figure 2-28).

Figure 2-22 shows 16 slaves with dedicated interrupt access to MSS.

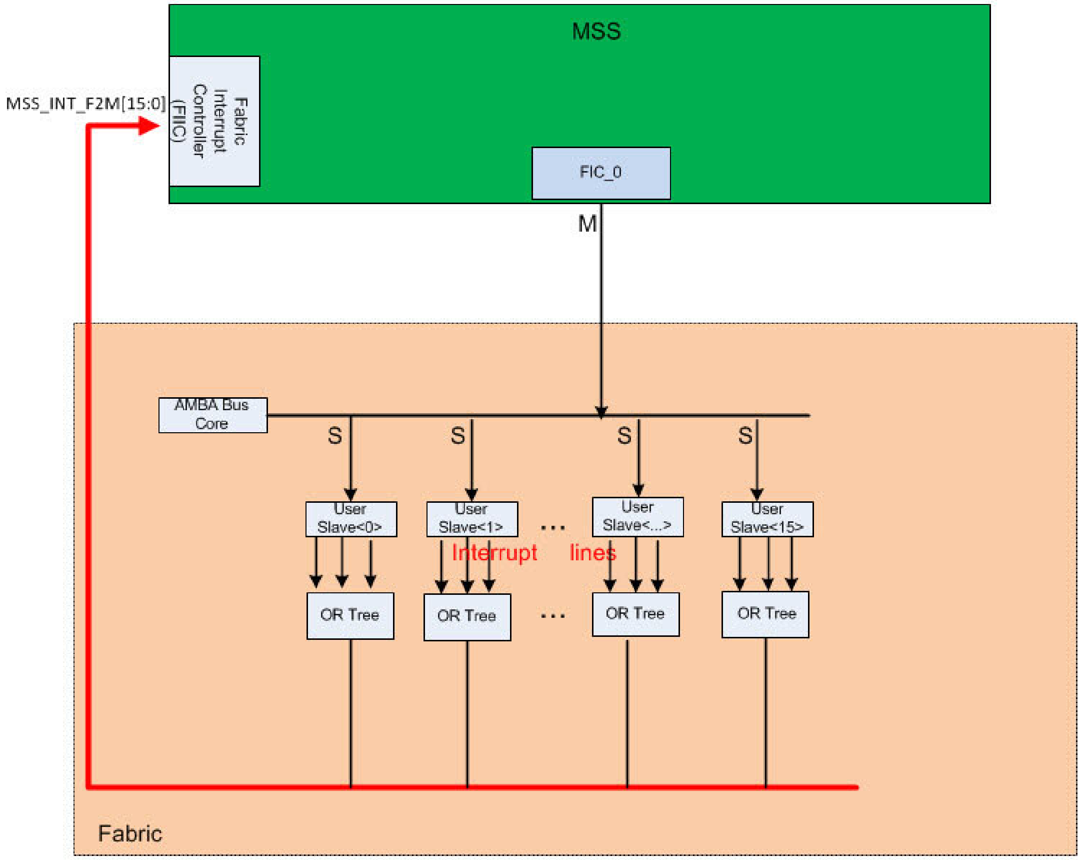


***Figure 2-22* • 16 Slaves with Dedicated Interrupt Access to MSS**

When there are more than 16 Slaves with interrupts, System Builder automatically instantiates the CoreInterrupt soft IP Core. The CoreInterrupt Core is a soft APB Interrupt Controller that System Builder automatically instantiates in the Fabric. Slave Interrupts from Slave 16 and above are connected to the 32 inputs (irqSource) of CoreInterrupt soft IP Core via cascaded OR-gates. The output of the CoreInterrupt, IRQ, is connected to MSS_INT_F2M[15]. For details on the CoreInterrupt core, refer to the CoreInterrupt Handbook.

Figure 2-23 shows 18 fabric slaves with interrupts. The interrupts from each of the 15 slaves are ORed in a cascaded OR-tree and connected to MSS_INT_F2M[14:0]. Bit 15 of MSS_INT_F2M is connected to the IRQ output of CoreInterrupt. Slave 0 through 14 have dedicated interrupt access to the MSS. Slave15, 16 and 17 have interrupt access to the MSS via the CoreInterrupt.

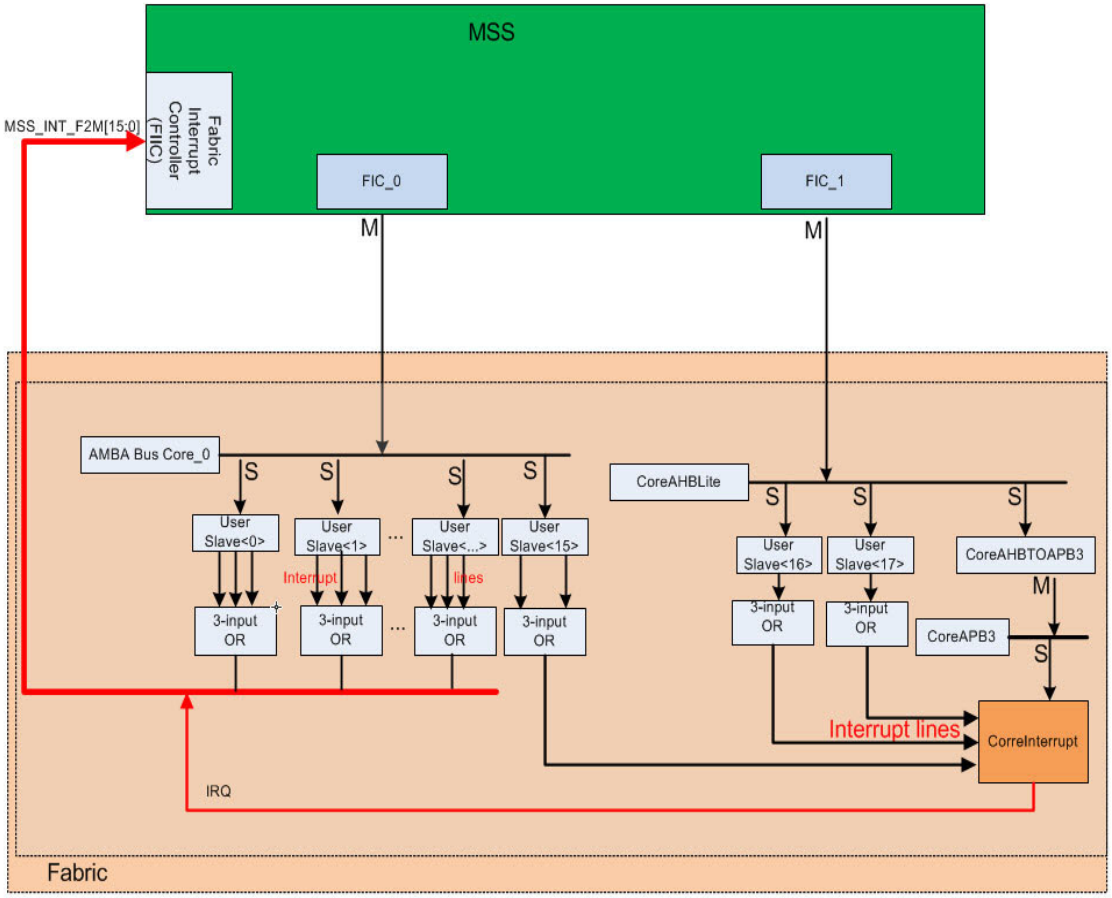


Figure 2-23 • **CoreInterrupt and Slave Interrupt Lines**

When the design has more than 47 slaves with interrupts, System Builder instantiates a second instance of CoreInterrupt to handle another 32 Slaves with Interrupts. The 32 inputs (irqSource) of the second CoreInterrutpt are connected to interrupt lines from Slave 48 through 79 (via cascaded OR-gates). The IRQ output is connected to MSS_INT_F2M[14].

When the MSS Cortex-M3 processor receives an Interrupt from the Fabric, it can query the CoreInterrupt to determine from which Fabric Slave the Interrupt originates.

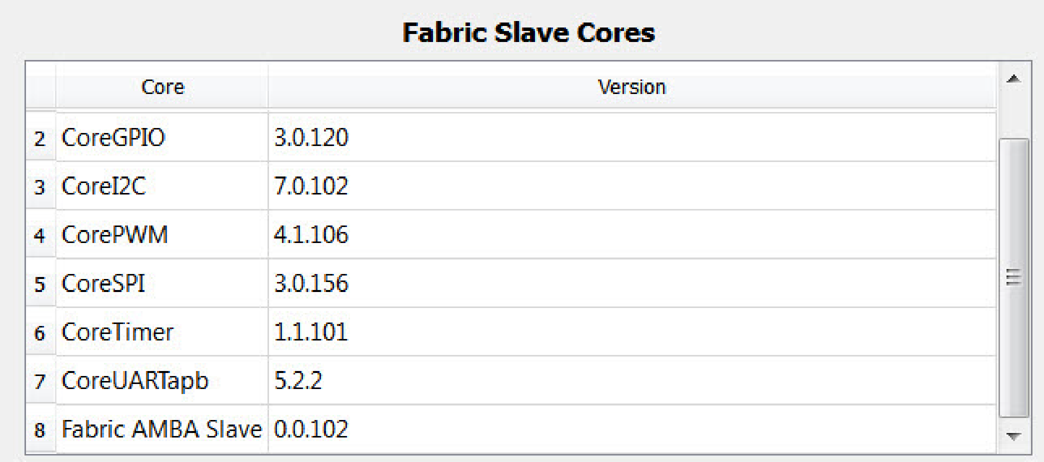
Fabric Slaves with Fixed Number of Interrupts

The Peripherals page of System Builder has a list of fabric slaves for you to build your subsystems:

- CoreGPIO
- CoreI2C
- CorePWM
- CoreSPI
- CoreTimer
- CoreUARTapb

Figure 2-24 shows fabric slaves with a fixed number of interrupts.

.



Fabric Slave Cores

	Core	Version
2	CoreGPIO	3.0.120
3	CoreI2C	7.0.102
4	CorePWM	4.1.106
5	CoreSPI	3.0.156
6	CoreTimer	1.1.101
7	CoreUARTapb	5.2.2
8	Fabric AMBA Slave	0.0.102

***Figure 2-24* • Fabric Slaves with Fixed Number of Interrupts**

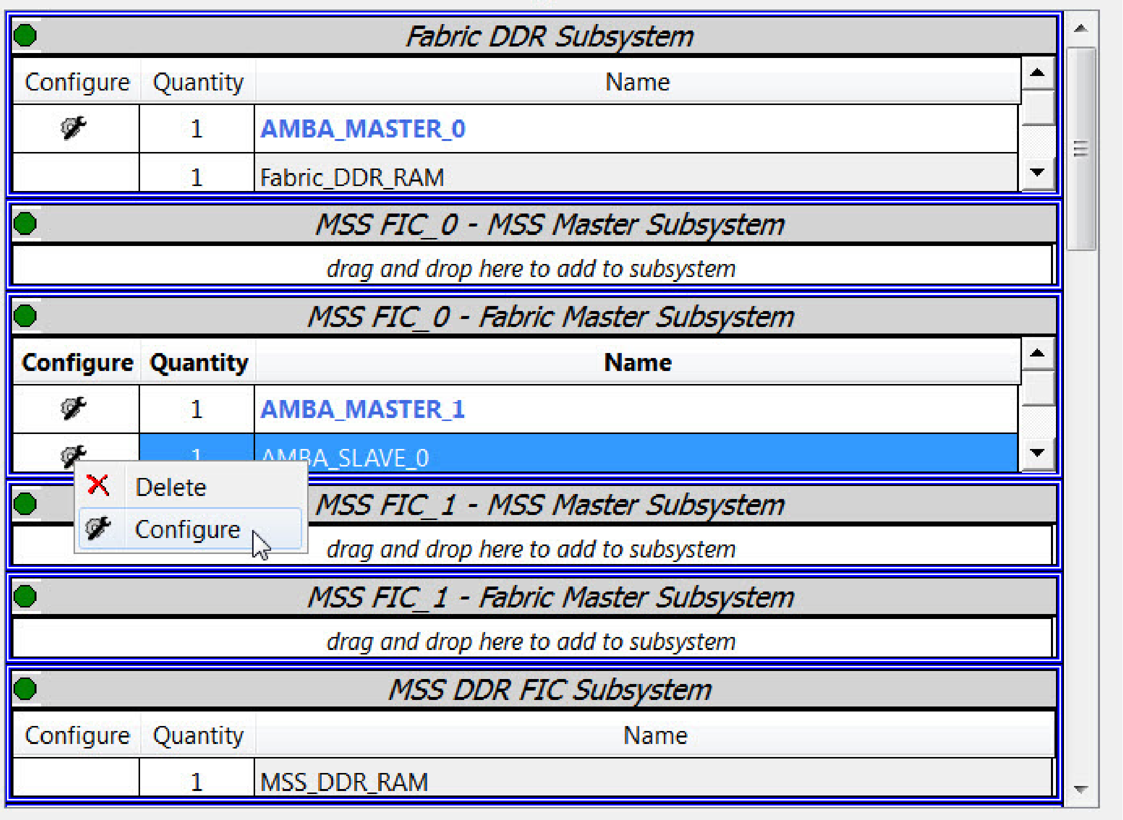
For these fabric slaves, you do not need to configure the number of interrupt signals. They have a fixed number of interrupts. System Builder automatically connects the interrupts from these fabrics to the MSS_INT_F2M pins of MSS (or irqSource pins of CoreInterrupt) via cascaded OR gates.

Fabric Slaves with Configurable Number of Interrupts

When you want to add your own AMBA(AHBLite/APB3/AXI) compliant slave peripheral logic OR when you want to add other slave peripherals from the catalog that are not listed under the 'Fabric Slave Cores' list to a subsystem with Cortex-M3(MSS) as the master, you have to add 'Fabric AMBA Slave' core from under the 'Fabric Slave Cores' list to any of the FIC_0/1 MSS Master Subsystems under the Subsystems list.

The number of Interrupts per fabric slave, up to a maximum of 8, is configurable. To specify the number of interrupts per slave, right-click the AMBA Slave of the subsystem in the Peripherals Page of System Builder to open the Configurator.

Figure 2-25 shows the AMBA Fabric Slave in the Peripherals page.



Fabric DDR Subsystem

Configure	Quantity	Name
	1	AMBA_MASTER_0
	1	Fabric_DDR_RAM

MSS FIC_0 - MSS Master Subsystem

drag and drop here to add to subsystem

MSS FIC_0 - Fabric Master Subsystem

Configure	Quantity	Name
	1	AMBA_MASTER_1
	1	AMBA_SLAVE_0

MSS FIC_1 - MSS Master Subsystem

drag and drop here to add to subsystem

MSS FIC_1 - Fabric Master Subsystem

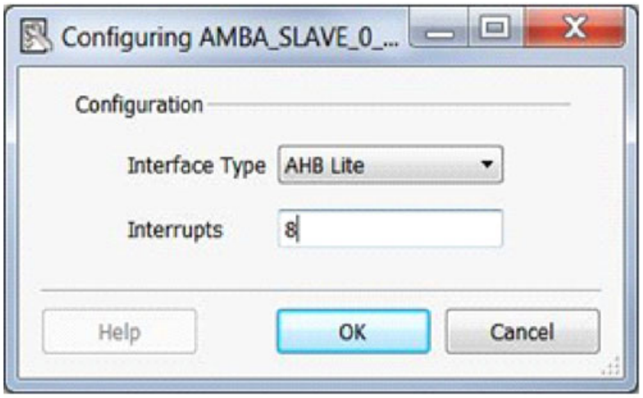
drag and drop here to add to subsystem

MSS DDR FIC Subsystem

Configure	Quantity	Name
	1	MSS_DDR_RAM

Figure 2-25 **• AMBA Fabric Slave in Peripherals Page**

In the Configurator, specify the Interface Type (AHBLite/APB3/AXI) and the number of Interrupts you need (Maximum 8). See Figure 2-26.



Configuring AMBA_SLAVE_0...

Configuration

Interface Type: AHB Lite

Interrupts: 8

Figure 2-26 **• Number of Interrupts Per Save**

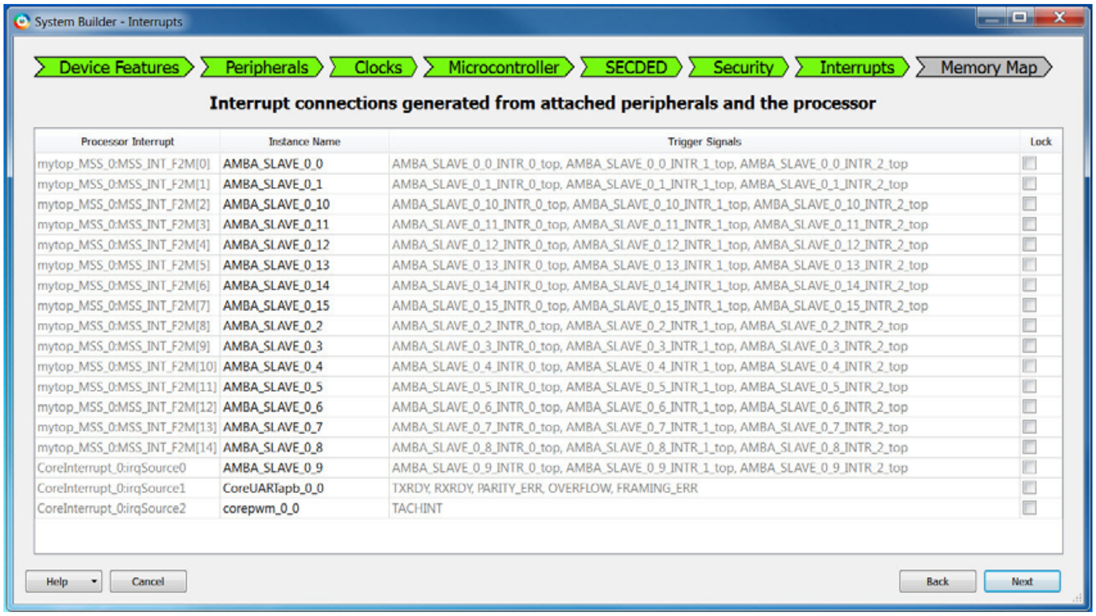
For fabric slaves with a configurable number of interrupts, System Builder exposes to the top level System Builder block the OR tree inputs corresponding to each fabric slave interrupts. You need to connect these top level input ports to the interrupt trigger signals of the slave peripheral logic outside the System Builder block.

Interrupts Page and Pin Assignment

By default, System Builder assigns pins of MSS_INT_F2M[15:0] to specific slaves. You can change the pin assignment from System Builder's Interrupt page.

To assign/unassign a fabric slave instance to a specific MSS_INT_F2M pin, click the corresponding fabric slave instance name and choose an option from the drop-down menu.

Figure 2-27 shows MSS interrupt signals in the Interrupts page.



Interrupt connections generated from attached peripherals and the processor

Processor Interrupt	Instance Name	Trigger Signals	Lock
mytop_MSS_0:MSS_INT_F2M[0]	AMBA_SLAVE_0_0	AMBA_SLAVE_0_0_INTR_0_top, AMBA_SLAVE_0_0_INTR_1_top, AMBA_SLAVE_0_0_INTR_2_top	□
mytop_MSS_0:MSS_INT_F2M[1]	AMBA_SLAVE_0_1	AMBA_SLAVE_0_1_INTR_0_top, AMBA_SLAVE_0_1_INTR_1_top, AMBA_SLAVE_0_1_INTR_2_top	□
mytop_MSS_0:MSS_INT_F2M[2]	AMBA_SLAVE_0_10	AMBA_SLAVE_0_10_INTR_0_top, AMBA_SLAVE_0_10_INTR_1_top, AMBA_SLAVE_0_10_INTR_2_top	□
mytop_MSS_0:MSS_INT_F2M[3]	AMBA_SLAVE_0_11	AMBA_SLAVE_0_11_INTR_0_top, AMBA_SLAVE_0_11_INTR_1_top, AMBA_SLAVE_0_11_INTR_2_top	□
mytop_MSS_0:MSS_INT_F2M[4]	AMBA_SLAVE_0_12	AMBA_SLAVE_0_12_INTR_0_top, AMBA_SLAVE_0_12_INTR_1_top, AMBA_SLAVE_0_12_INTR_2_top	□
mytop_MSS_0:MSS_INT_F2M[5]	AMBA_SLAVE_0_13	AMBA_SLAVE_0_13_INTR_0_top, AMBA_SLAVE_0_13_INTR_1_top, AMBA_SLAVE_0_13_INTR_2_top	□
mytop_MSS_0:MSS_INT_F2M[6]	AMBA_SLAVE_0_14	AMBA_SLAVE_0_14_INTR_0_top, AMBA_SLAVE_0_14_INTR_1_top, AMBA_SLAVE_0_14_INTR_2_top	□
mytop_MSS_0:MSS_INT_F2M[7]	AMBA_SLAVE_0_15	AMBA_SLAVE_0_15_INTR_0_top, AMBA_SLAVE_0_15_INTR_1_top, AMBA_SLAVE_0_15_INTR_2_top	□
mytop_MSS_0:MSS_INT_F2M[8]	AMBA_SLAVE_0_2	AMBA_SLAVE_0_2_INTR_0_top, AMBA_SLAVE_0_2_INTR_1_top, AMBA_SLAVE_0_2_INTR_2_top	□
mytop_MSS_0:MSS_INT_F2M[9]	AMBA_SLAVE_0_3	AMBA_SLAVE_0_3_INTR_0_top, AMBA_SLAVE_0_3_INTR_1_top, AMBA_SLAVE_0_3_INTR_2_top	□
mytop_MSS_0:MSS_INT_F2M[10]	AMBA_SLAVE_0_4	AMBA_SLAVE_0_4_INTR_0_top, AMBA_SLAVE_0_4_INTR_1_top, AMBA_SLAVE_0_4_INTR_2_top	□
mytop_MSS_0:MSS_INT_F2M[11]	AMBA_SLAVE_0_5	AMBA_SLAVE_0_5_INTR_0_top, AMBA_SLAVE_0_5_INTR_1_top, AMBA_SLAVE_0_5_INTR_2_top	□
mytop_MSS_0:MSS_INT_F2M[12]	AMBA_SLAVE_0_6	AMBA_SLAVE_0_6_INTR_0_top, AMBA_SLAVE_0_6_INTR_1_top, AMBA_SLAVE_0_6_INTR_2_top	□
mytop_MSS_0:MSS_INT_F2M[13]	AMBA_SLAVE_0_7	AMBA_SLAVE_0_7_INTR_0_top, AMBA_SLAVE_0_7_INTR_1_top, AMBA_SLAVE_0_7_INTR_2_top	□
mytop_MSS_0:MSS_INT_F2M[14]	AMBA_SLAVE_0_8	AMBA_SLAVE_0_8_INTR_0_top, AMBA_SLAVE_0_8_INTR_1_top, AMBA_SLAVE_0_8_INTR_2_top	□
CoreInterrupt_0:irqSource0	AMBA_SLAVE_0_9	AMBA_SLAVE_0_9_INTR_0_top, AMBA_SLAVE_0_9_INTR_1_top, AMBA_SLAVE_0_9_INTR_2_top	□
CoreInterrupt_0:irqSource1	CoreUARTapb_0_0	TXRDY, RXRDY, PARITY_ERR, OVERFLOW, FRAMING_ERR	□
CoreInterrupt_0:irqSource2	corepwm_0_0	TACHINT	□

Figure 2-27 • **MSS Interrupt Signals**

This page also enables you to lock the interrupt pin assignment of a fabric slave core's interrupt pins to a desired MSS_INT_F2M[n] pin. A maximum of 16 different fabric slave cores' interrupt pins can be directly connected to the MSS_INT_F2M[15:0].]. To lock the pin assignment, check the Lock checkbox at the far right.

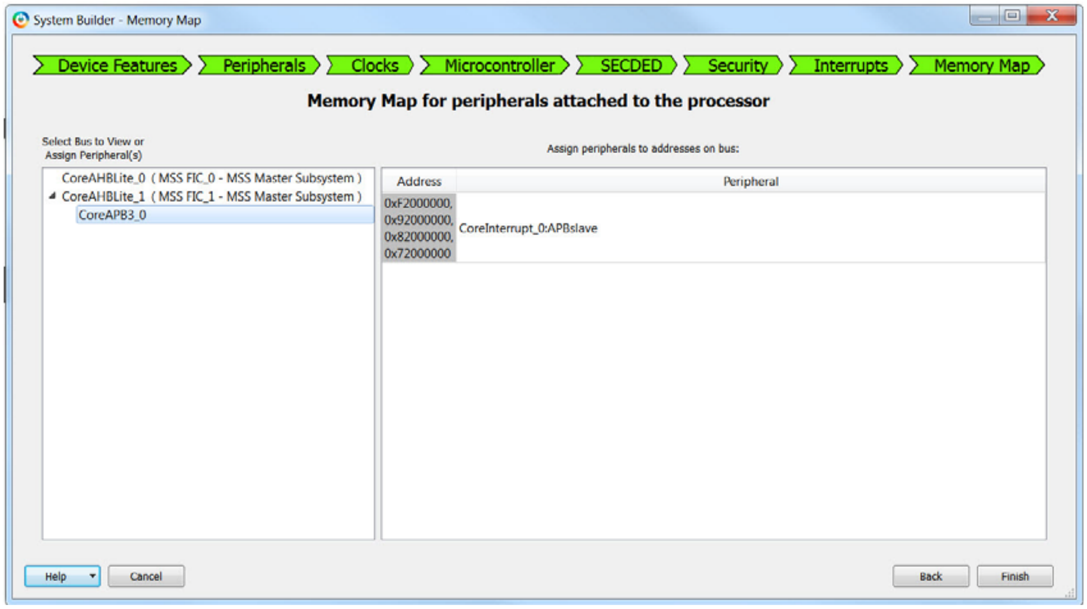
When a specific MSS_INT_F2M[n] pin or irqSource<n> pin is assigned to a specific slave, the "locked" assignment is honored when you regenerate your design in System Builder or add/remove slaves with interrupts incrementally. The pin assignment of the specific slave interrupt line to MSS_INT_F2M[n] remains unchanged after System Builder regenerates your system.

You do not need to lock the interrupt lines if either of the following is true:

- You are satisfied with the default interrupt pin assignment
- You do not want to modify or regenerate your design in System Builder (with or without changes to the slave peripherals)

If you choose not to "lock" the interrupt assignment for a slave the first time you generate the System Builder block and then after generation decide to modify the design (adding or deleting slave peripherals), the interrupt assignment for the slave may change on regeneration.

The CoreInterrupt is an APB3 slave. The Memory Map of the CoreInterrupt core from the MSS perspective displays under the CoreAPB3 bus in the Memory Map Page of System Builder (Figure 2-28). The memory address of the CoreInterrupt is design-specific. Your design may show a memory address different from this figure.



Memory Map for peripherals attached to the processor

Select Bus to View or Assign Peripheral(s)

- CoreAHBLite_0 (MSS FIC_0 - MSS Master Subsystem)
- CoreAHBLite_1 (MSS FIC_1 - MSS Master Subsystem)
  - CoreAPB3_0

Assign peripherals to addresses on bus:

Address	Peripheral
0xF2000000, 0x92000000, 0x82000000, 0x72000000	CoreInterrupt_0:APBslave

Figure 2-28 **• Memory Map of CoreInterrupt Core**

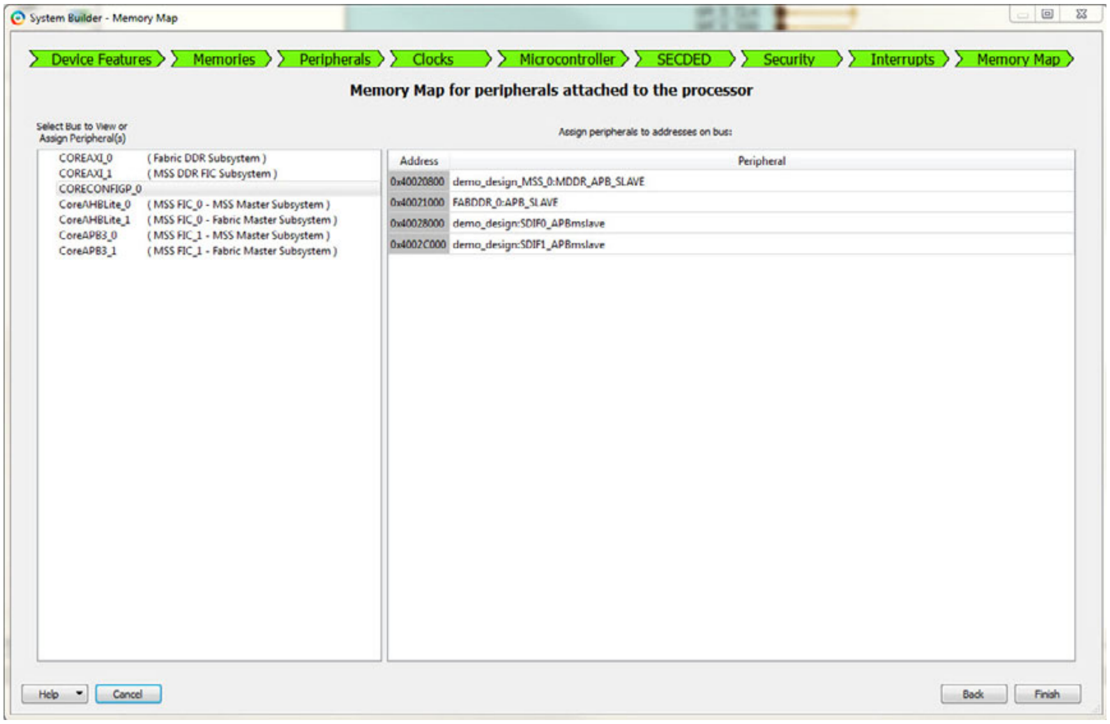
Subsystem Memory Map Configuration

This page displays the memory map for each of your subsystems. You can make limited modifications to the memory map within a subsystem. The Select Bus to View or Assign Peripheral panel displays one bus interface for each subsystem in your design. These buses are internally generated (they are part of the System Builder block).

If there are multiple slaves on a bus you can use the drop-down lists in the Assign peripherals to addresses on the bus panel to modify slave addresses.

Note: For the MSS FIC0/1 Fabric Master subsystems, address ranges 0x00000000-0x0fffffff, 0x20000000-0x2fffffff, 0x40000000-0x4fffffff, and 0x60000000-0x6fffffff all map into the MSS.

Refer to the Microsemi SmartFusion2 Fabric User's Guide for details.



Memory Map for peripherals attached to the processor

Select Bus to View or Assign Peripheral(s)

COREAXI_0	(Fabric DDR Subsystem)
COREAXI_1	(MSS DDR FIC Subsystem)
CORECONFIGP_0	
CoreAHBLite_0	(MSS FIC_0 - MSS Master Subsystem)
CoreAHBLite_1	(MSS FIC_0 - Fabric Master Subsystem)
CoreAPB3_0	(MSS FIC_1 - MSS Master Subsystem)
CoreAPB3_1	(MSS FIC_1 - Fabric Master Subsystem)

Assign peripherals to addresses on bus:

Address	Peripheral
0x40020800	demo_design_MSS_0:MDDR_APB_SLAVE
0x40021000	FABDDR_0:APB_SLAVE
0x40028000	demo_design:SDIF0_APBmslave
0x4002C000	demo_design:SDIF1_APBmslave

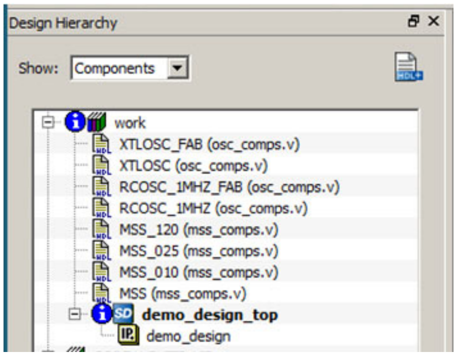
Figure 2-29 • **MSS Memory Mapping in System Builder**

3 – Generated Design

After generating your design you will see two new components in your Design Hierarchy. There will be a <design_name>_top, which is a SmartDesign that instantiates your System Builder generated component <design_name>.

The <design_name>_top is a regular SmartDesign that contains an instance of your System Builder. Double-click the <design_name> component in your Design Hierarchy to reconfigure the System Builder. All of your options are preserved, so you do not need to reconfigure them.

Figure 3-1 shows the Design Hierarchy after clicking Finish in the final System Builder page on a System Builder named demo_design.



Design Hierarchy

Show: Components

```
work
    XTLOSC_FAB (osc_comps.v)
    XTLOSC (osc_comps.v)
    RCOSC_1MHZ_FAB (osc_comps.v)
    RCOSC_1MHZ (osc_comps.v)
    MSS_120 (mss_comps.v)
    MSS_025 (mss_comps.v)
    MSS_010 (mss_comps.v)
    MSS (mss_comps.v)
    demo_design_top
        demo_design
```

Figure 3-1 **• MSS DDR Configuration with Access from FPGA Fabric**

The SmartDesign Canvas displays the instance of the System Builder generated component. You can choose to add your own custom peripherals or logic to the SmartDesign at this point.

Your System Builder design is summarized in the System Builder summary report in the Reports View. The name of this file is <design>_SYS_BLD.xml. Open the Reports View and click this file for details on what was generated in your System Builder component.

Depending on the type of subsystems you have configured for the System Builder block, the System Builder block may have the ports exposed at the top level, as shown in Figure 3-2 below.

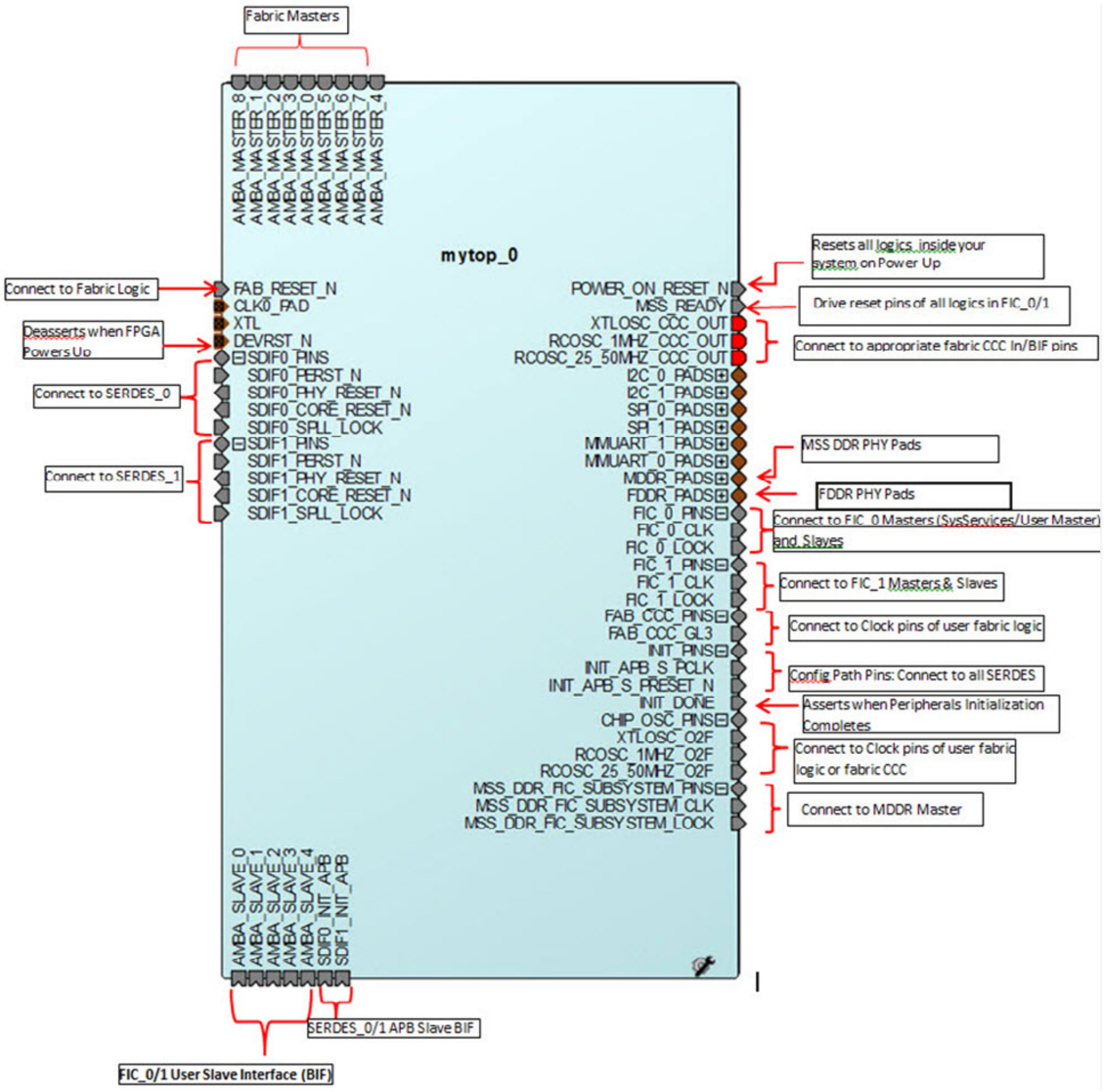


Figure 3-2 • **System Builder Block Top Level Port Configuration**

4 – Extending Your System Builder Design

You can add custom peripherals or masters to the System Builder generated design. This flexibility enables you to quickly customize the System Builder generated design with your special logic.

To extend your System Builder design, you need to instantiate and configure the special Fabric AMBA Slave and Fabric AMBA Master cores in your subsystems on the Peripherals page. These cores allow you to attach any custom peripherals onto the generated design.

By instantiating and configuring these cores into a subsystem, the necessary and appropriate bus interface pins, clocks, and resets are exposed at the top level System Builder generated design. At the top level SmartDesign, you can then connect up these ports to your custom peripheral and they will automatically be visible to the other masters/peripherals in that subsystem.

5 – Modifying/Inspecting Your System Builder Design

There may be situations where you need to modify the internals of the System Builder component or inspect what was generated.

To do so, right-click the System Builder component in the Design Hierarchy and choose **Open as SmartDesign** (Figure 5-1).

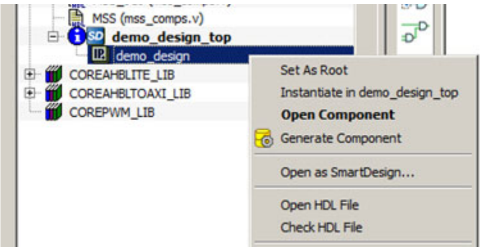


Figure 5-1 **• Open System Builder Component as SmartDesign**

Your system builder component opens as a SmartDesign, enabling you to modify the design or view what was generated. At this point, it is just a regular SmartDesign and you can operate and treat it as such.

If you want to re-open it as a System Builder, right-click the component in the Design Hierarchy and choose **Re-open as System Builder**. Any modifications that you made inside the SmartDesign are ignored and not retained during your next System Builder generation (Figure 5-2).

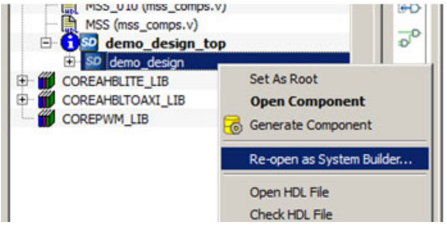


Figure 5-2 **• Open System Builder Component as SmartDesign**

Finishing Your Design

After generating and extending your System Builder, the rest of the Libero design flow is the same as a regular Libero FPGA flow. You can use the Design Flow tools to Simulate, Synthesize, Compile, Place and Route, and Program your design.

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 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996