

AC489
Application Note
Building the PolarFire SoC MSS Subsystem



a  **MICROCHIP** company



a  MICROCHIP company

Microsemi 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

Email: sales.support@microsemi.com

www.microsemi.com

©2019 Microsemi, a wholly owned subsidiary of Microchip Technology Inc. All rights reserved. Microsemi and the Microsemi logo are registered trademarks of Microsemi Corporation. All other trademarks and service marks are the property of their respective owners.

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.

About Microsemi

Microsemi, a wholly owned subsidiary of Microchip Technology Inc. (Nasdaq: MCHP), offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Learn more at www.microsemi.com.

Contents

1	Building the MSS Subsystem	6
1.1	Design Requirements	6
1.2	Prerequisites	6
1.3	Vault Setting in Libero	7
1.4	Block Diagram	8
1.5	Hardware Implementation	9
1.5.1	I/O Ports	9
1.5.2	PFSOC_MSS SgCore IP	10
1.5.3	CORERESET_PF IP	16
1.5.4	PF_OSC IP	16
1.5.5	PF_CCC IP	16
1.5.6	COREAPB3 IP	16
1.5.7	CoreUARTapb IP	16
2	Libero Design Flow	17
2.1	Synthesize	17
2.2	Place and Route	17
2.3	Verify Timing	18
3	Revision History	19
3.1	Revision 1.0	19

Figures

Figure 1	Vault Settings Option	7
Figure 2	Browse Vault Location	7
Figure 3	Block Diagram	8
Figure 4	Hardware Design	9
Figure 5	Clocks Configuration	10
Figure 6	FIC Configuration	11
Figure 7	IO Configuration	12
Figure 8	IO Overlap Warning	13
Figure 9	DDR Topology	13
Figure 10	DDR Memory Initialization	14
Figure 11	DDR Memory Timing	15
Figure 12	DDR Controller Efficiency	15
Figure 13	Libero Design Flow	17
Figure 14	Derived Constraints	17

Tables

Table 1	Design Requirements	6
Table 2	I/O Ports	9

1 Building the MSS Subsystem

Microchip's PolarFire SoC FPGAs include industry's first RISC-V based Microprocessor Subsystem (MSS) and the PolarFire fabric. The MSS includes 5x 64-bit RISC-V processor cores, DDR Controller, Fabric Interface Controllers (FIC), and a rich set of peripherals. For more information about the PolarFire SoC MSS and its components, see *UG0880: PolarFire SoC FPGA MSS User Guide*.

The Libero SoC design suite includes a wide range of IP cores, which enable FPGA designers to build PolarFire SoC FPGA designs for various application domains. Libero SoC includes the PFSOC_MSS SgCore IP which is used to build the MSS subsystem in PolarFire SoC FPGA designs.

This application note describes how to build the MSS subsystem with peripherals using the PFSOC_MSS SgCore IP in Libero SoC. A reference design is also included enabling designers to quickly rebuild the MSS subsystem using the options available in the PFSOC_MSS SgCore IP Configurator.

1.1 Design Requirements

Table 1 lists hardware and software required to build the MSS subsystem.

Table 1 • Design Requirements

Software	Description
-Libero SoC	v12.3
Hardware	Description
-Host PC	Windows 10
License	Description
-Libero Gold License or higher	Required for Libero SoC
-PolarFire SoC License	Required for the PolarFire SoC family

1.2 Prerequisites

Before you start:

1. Download the reference design files using the following link:
http://soc.microsemi.com/download/rsc/?f=mpfs_ac489_df
2. Download and install Libero SoC using the following link:
<https://www.microsemi.com/product-directory/design-resources/1750-libero-soc#downloads>
3. Contact the local sales team to get the PolarFire SoC license along with the Libero SoC license to design for the PolarFire SoC family.
4. Download and install the [Libero SoC Mega Vault](#). Mega Vault is the repository of all the IP cores (including the PFSOC_MSS Sgcore IP) required to design for the PolarFire SoC family.

1.3 Vault Setting in Libero

To use the PFSOC_MSS IP core, the vault location must be set to the Mega Vault in Libero SoC.

Figure 1 • Vault Settings Option

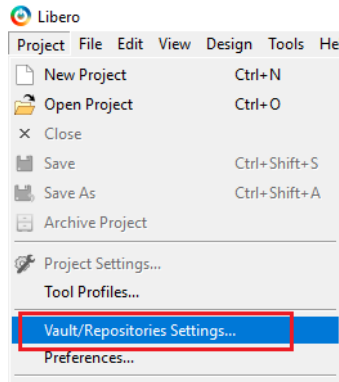
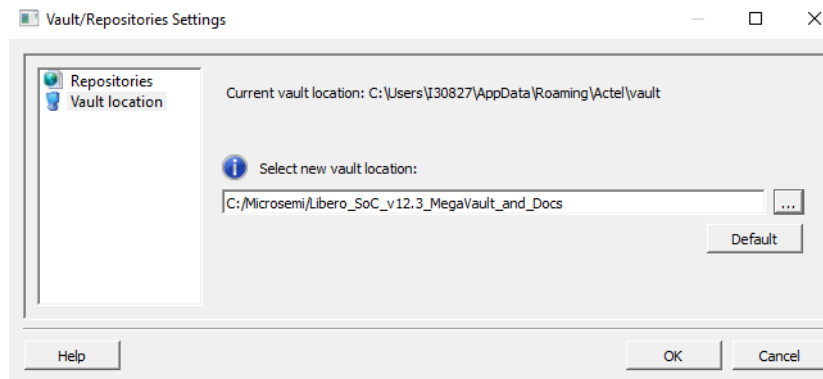


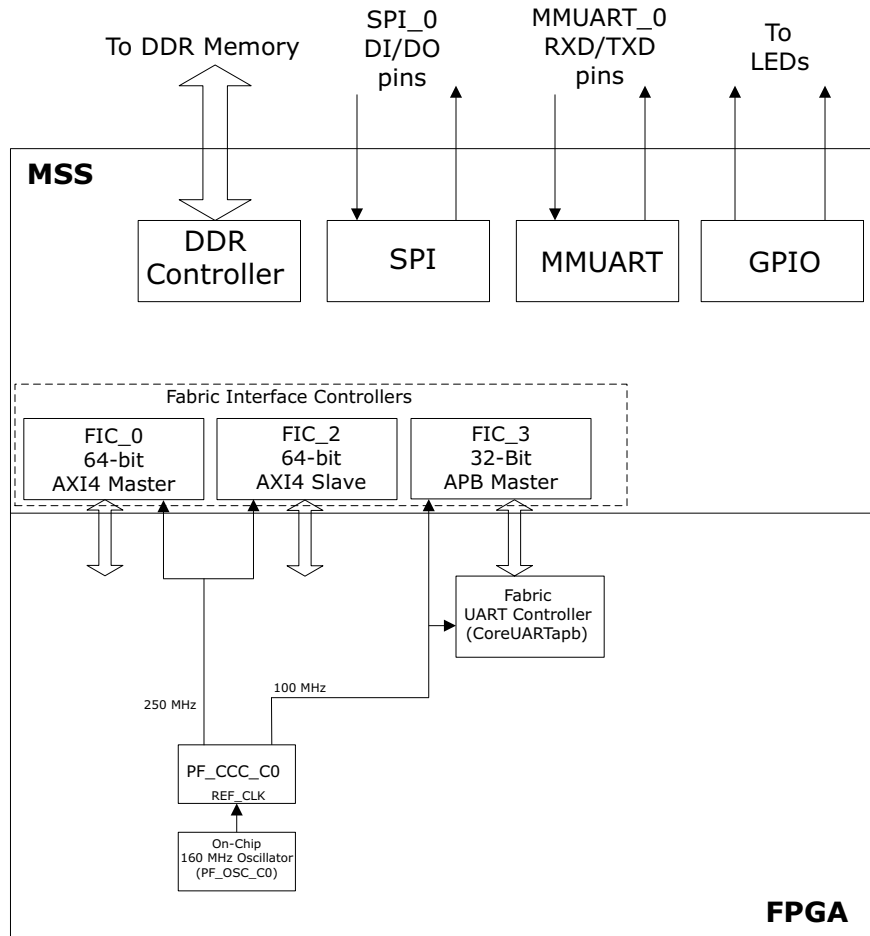
Figure 2 • Browse Vault Location



1.4 Block Diagram

The MSS includes a wide range of peripherals and in the reference design, the MSS is configured with DDR controller, SPI, MMUART, GPIO, and three FICs (FIC_0, FIC_2, and FIC_3). Figure 3 shows the high-level overview of the PolarFire SoC MSS reference design.

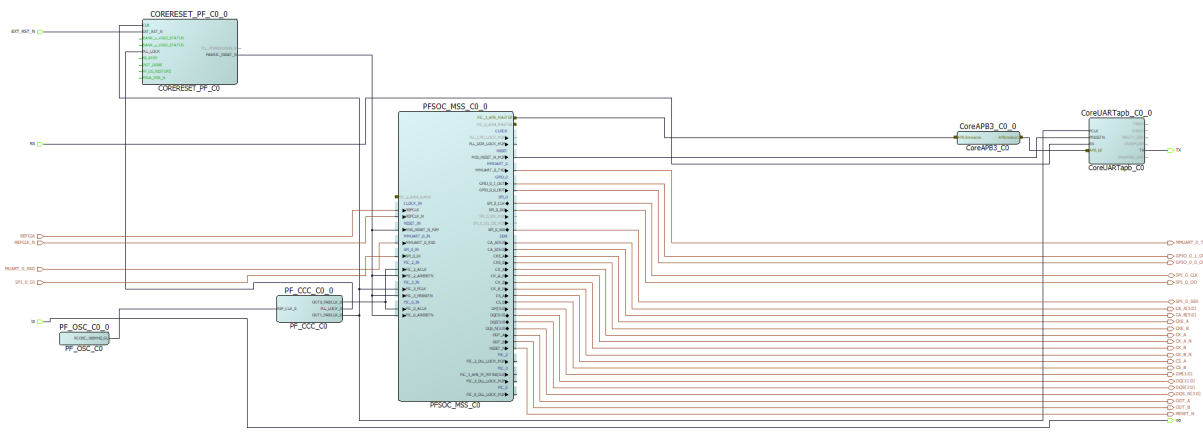
Figure 3 • Block Diagram



1.5 Hardware Implementation

Figure 4 shows the MSS subsystem implemented in the reference design.

Figure 4 • Hardware Design



As shown in Figure 4, the MSS subsystem contains the following IP blocks:

- PFSOC_MSS SgCore IP (hard IP)
- CORERESET_PF IP (soft IP)
- PF_OSC IP (hard IP)
- PF_CCC IP (hard IP)
- COREAPB3 IP (soft IP)
- CoreUARTapb IP (soft IP)

1.5.1 I/O Ports

Table 2 lists the important I/Os in the reference design.

Table 2 • I/O Ports

Port Name	Direction	Description
EXT_RST_N	Input	Active-low user reset
REFCLK REFCLK_N	Input	Reference clock input received from the dedicated I/O from Bank 5 (REFCLK)
RX TX	Input Output	UART interface to the Host PC from the FPGA UART interface to the FPGA from the Host PC
SPI_0_DI SPI_0_DO	Input Output	SPI Data In interface to the MSS SPI Data Out interface from the MSS
MMUART_0_RXD MMUART_0_TXD	Input Output	MMUART interface to the Host PC from the MSS MMUART interface to the MSS from the Host PC
GPIO_0_0_OUT GPIO_0_1_OUT	Output Output	GPIO signal mapped to an on-board LED GPIO signal mapped to an on-board LED

1.5.2 PFSOC_MSS SgCore IP

This IP is instantiated as PFSOC_MSS_C0_0 and used to build the MSS subsystem. In the reference design, the MSS is configured with DDR Controller, FIC0, FIC2, FIC3, GPIO, SPI, and MMUART.

Building the MSS involves the following configurations:

- [Clocks Configuration](#)
- [FIC Configuration](#)
- [I/O Configuration](#)
- [DDR Configuration](#)

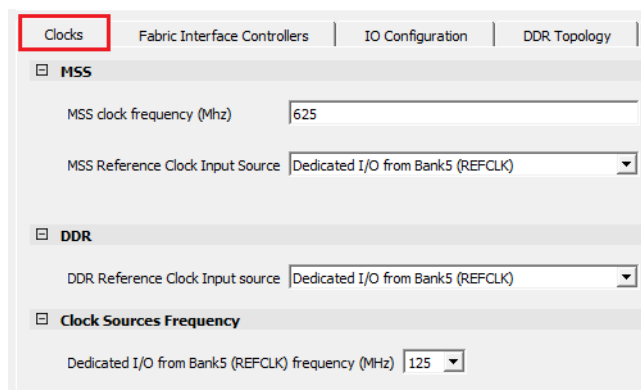
1.5.2.1 Clocks Configuration

The MSS Configurator -> Clocks tab is used to configure the following:

- **MSS clock frequency (MHz):** In the reference design, this value is set to 625 MHz because the RISC-V processors operate at 625 MHz.
- **MSS Reference Clock Input Source:** In the reference design, the I/Os for MSS clock are selected from Bank 5.
- **DDR Reference Clock Input Source:** In the reference design, the DDR reference clock I/Os are selected from Bank 5.
- **Clock Sources Frequency:** In the reference design, a 125 MHz clock source is selected.

Figure 5 shows the clocks configuration used in the reference design.

Figure 5 • Clocks Configuration



The screenshot shows the 'Clocks' tab of the MSS Configurator. The 'MSS' section is expanded, showing the following settings:

- MSS clock frequency (Mhz): 625
- MSS Reference Clock Input Source: Dedicated I/O from Bank5 (REFCLK)

The 'DDR' section is also expanded, showing:

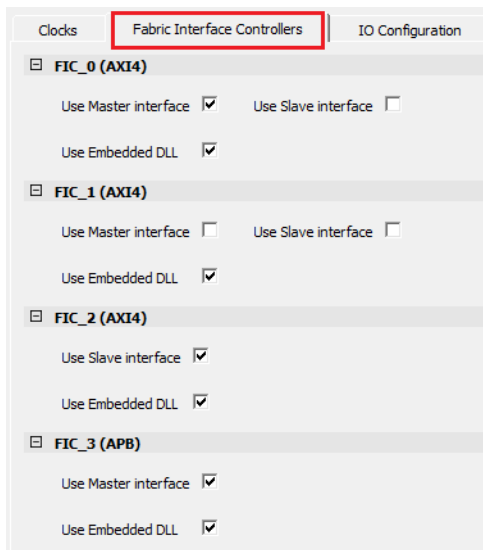
- DDR Reference Clock Input source: Dedicated I/O from Bank5 (REFCLK)

The 'Clock Sources Frequency' section is expanded, showing:

- Dedicated I/O from Bank5 (REFCLK) frequency (MHz): 125

1.5.2.2 FIC Configuration

MSS includes four FICs for data transfer between MSS and the fabric. The MSS Configurator -> Fabric Interface Controllers tab is used to configure the FICs. FIC_0 and FIC_1 have both master and slave interfaces to/from the FPGA fabric. In the reference design, FIC_0 (master interface), FIC_2 (slave interface), and FIC_3 (master interface) are used as shown in Figure 6.

Figure 6 • FIC Configuration

Note: When a master interface is enabled for a FIC, that master interface must be connected to a slave in the fabric as shown in [Figure 4](#). When a slave interface is enabled for a FIC, that slave interface must be connected to a master in the fabric as shown in [Figure 4](#).

1.5.2.3 I/O Configuration

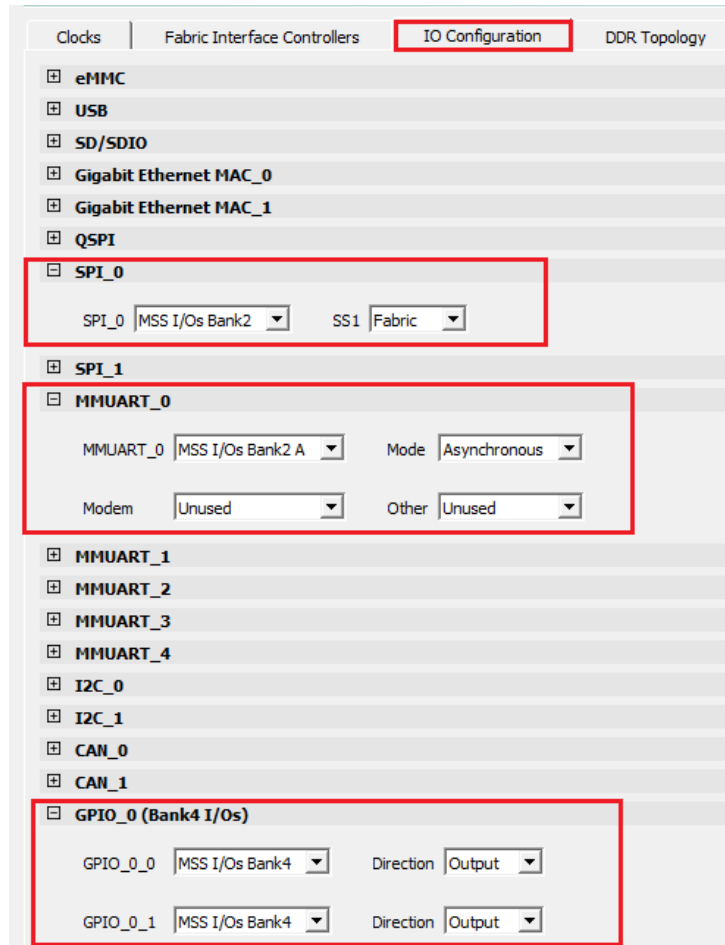
MSS supports a wide range of peripherals. The MSS Configurator -> IO Configuration tab is used to select the peripherals and assign the I/Os for those peripherals. By default, all peripherals are marked as unused. To include peripherals that are required in the design, use the drop-down corresponding to each peripheral and assign MSS I/Os or fabric I/Os for that peripheral.

The following peripherals are dedicated and cannot be assigned to fabric I/Os:

- USB peripherals are dedicated in Bank 4
- eMMC peripherals are dedicated in Bank 2
- SD/SDIO peripherals are dedicated in Bank 2

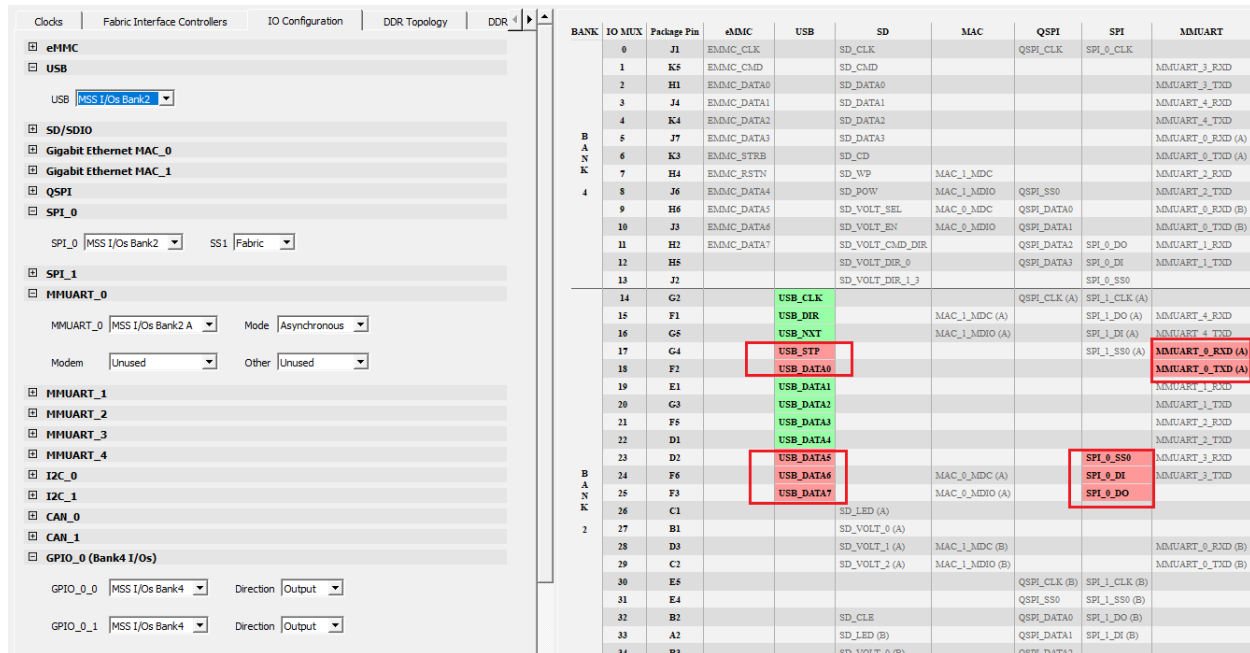
The reference design uses MMUART, GPIO, SPI as shown in [Figure 7](#).

Figure 7 • IO Configuration



Note: If the I/Os for a peripheral are selected in a Bank, selecting the same I/Os for another peripheral from the same Bank is not allowed. The user is warned by red highlights on overlapping I/Os as shown in Figure 8.

Figure 8 • IO Overlap Warning

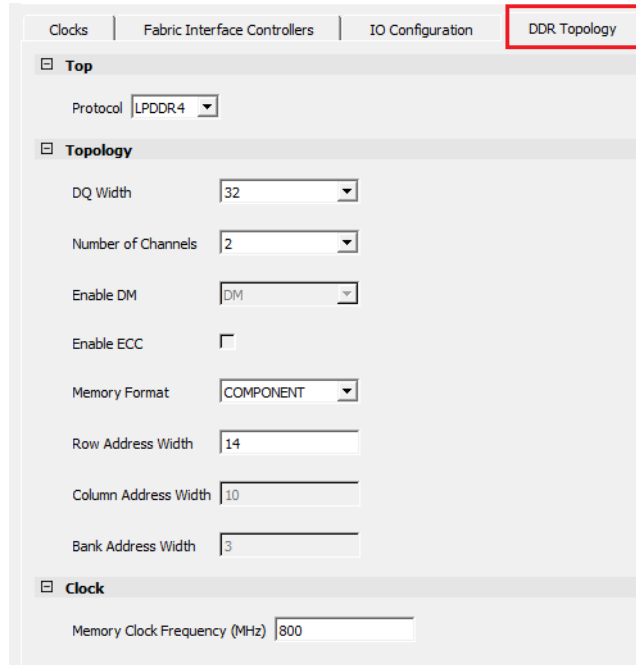


BANK	IO MUX	Package Pin	oDMIC	USB	SD	MAC	QSPI	SPI	MMUART
	0	J1	EMMC_CLK		SD_CLK		QSPI_CLK	SPI_0_CLK	
	1	K5	EMMC_CMD		SD_CMD				MMUART_3_RXD
	2	H1	EMMC_DATA0		SD_DATA0				MMUART_3_TXD
	3	J4	EMMC_DATA1		SD_DATA1				MMUART_4_RXD
	4	K4	EMMC_DATA2		SD_DATA2				MMUART_4_TXD
BANK 4	5	J7	EMMC_DATA3		SD_DATA3				MMUART_0_RXD (A)
	6	K3	EMMC_STRB		SD_CD				MMUART_0_TXD (A)
	7	H4	EMMC_RSTN		SD_WP	MAC_1_MDC			MMUART_2_RXD
	8	J6	EMMC_DATA4		SD_P0W	MAC_1_MDIO	QSPI_S80		MMUART_2_TXD
	9	H6	EMMC_DATA5		SD_VOLT_SEL	MAC_0_MDC	QSPI_DATA0		MMUART_0_RXD (B)
	10	J3	EMMC_DATA6		SD_VOLT_EN	MAC_0_MDIO	QSPI_DATA1		MMUART_0_TXD (B)
	11	H2	EMMC_DATA7		SD_VOLT_CMD_DIR		QSPI_DATA2	SPI_0_DO	MMUART_1_RXD
	12	H5			SD_VOLT_DIR_0		QSPI_DATA3	SPI_0_DI	MMUART_1_TXD
	13	J2			SD_VOLT_DIR_1_3			SPI_0_S80	
	14	G2			USB_CLK			QSPI_CLK (A)	SPI_1_CLK (A)
15	F1			USB_DIR		MAC_1_MDC (A)		SPI_1_DO (A)	MMUART_4_RXD
16	G5			USB_NXT		MAC_1_MDIO (A)		SPI_1_DI (A)	MMUART_4_TXD
17	G4			USB_STP				SPI_1_S80 (A)	MMUART_0_RXD (A)
18	F2			USB_DATA0					MMUART_0_TXD (A)
19	E1			USB_DATA1					MMUART_1_RXD
20	G3			USB_DATA2					MMUART_1_TXD
21	F5			USB_DATA3					MMUART_2_RXD
22	D1			USB_DATA4					MMUART_2_TXD
23	D2			USB_DATA5					MMUART_3_RXD
24	F6			USB_DATA6		MAC_0_MDC (A)			MMUART_3_TXD
25	F3			USB_DATA7		MAC_0_MDIO (A)			
BANK 2	26	C1			SD_LED (A)				
	27	B1			SD_VOLT_0 (A)				
	28	D3			SD_VOLT_1 (A)	MAC_1_MDC (B)			MMUART_0_RXD (B)
	29	C2			SD_VOLT_2 (A)	MAC_1_MDIO (B)			MMUART_0_TXD (B)
	30	E5					QSPI_CLK (B)	SPI_1_CLK (B)	
	31	E4					QSPI_S80	SPI_1_S80 (B)	
	32	B2			SD_CLE		QSPI_DATA0	SPI_1_DO (B)	
	33	A2			SD_LED (B)		QSPI_DATA1	SPI_1_DI (B)	
	34	B3			SD_VOLT_0 (B)		QSPI_DATA2		

1.5.2.4 DDR Configuration

The reference design configures the DDR controller for LPDDR4 memory. The following figures show the DDR Topology, memory initialization, memory timing, address ordering used in the reference design.

Figure 9 • DDR Topology



Clk | Fabric Interface Controllers | IO Configuration | **DDR Topology**

Top

Protocol: LPDDR4

Topology

DQ Width: 32

Number of Channels: 2

Enable DM: DM

Enable ECC:

Memory Format: COMPONENT

Row Address Width: 14

Column Address Width: 10

Bank Address Width: 3

Clock

Memory Clock Frequency (MHz): 800

Figure 10 • DDR Memory Initialization

Clocks	Fabric Interface Controllers	IO Configuration	DDR Topology	DDR Memory Initialization
Mode Register 1				
RD Pre-amble Type <input type="text" value="Static"/>				
RD Post-amble Length <input type="text" value="0.5*tCK"/>				
Mode Register 2				
Read Latency (#RL, #nRTP) <input type="text" value="RL=6, nRTP = 8"/>				
Write Latency <input type="text" value="WL=4"/>				
Mode Register 3				
Pull-up Calibration Point <input type="text" value="VDDQ/3"/>				
WR Post-amble Length <input type="text" value="0.5*tCK"/>				
Pull-Down Drive Strength <input type="text" value="RZQ/6"/>				
Mode Register 4				
Self Refresh Abort Mode <input type="text" value="Disabled"/>				
Mode Register 11				
DQ ODT <input type="text" value="Disable"/>				
CA ODT <input type="text" value="Disable"/>				
Mode Register 22				
SoC ODT <input type="text" value="Disable"/>				
ODTE-CK <input type="text" value="Disabled"/>				
ODTE-CS <input type="text" value="Disabled"/>				
ODTE-CA <input type="text" value="Obey ODT_CA bond pad"/>				

Figure 11 • DDR Memory Timing

Clocks	Fabric Interface Controllers	IO Configuration	DDR Topology	DDR Memory Initialization	DDR Memory Timing
--------	------------------------------	------------------	--------------	---------------------------	-------------------

Timing parameters dependent on speed bin

tRAS (ns)

tRCD (ns)

tRP (ns)

tRC (ns)

tWR (ns)

tMRR (cycles)

tMRW (cycles)

Timing parameters dependent on speed bin and clock frequency

tWTR (cycles)

tRRD (ns)

tRTP (ns)

Timing parameters dependent on operating condition

tREFI (us)

Timing parameters dependent on speed bin and page size

tRFC (ns)

tFAW (ns)

Other Timing parameters

ZQ Calibration Time (us)

ZQ Calibration Latch Time (ns)

tZQRESET (ns)

Enable User ZQ Calibration Controls

Automatic ZQ Calibration Period (us)

Figure 12 • DDR Controller Efficiency

Clocks	Fabric Interface Controllers	IO Configuration	DDR Topology	DDR Memory Initialization	DDR Memory Timing	DDR Controller
--------	------------------------------	------------------	--------------	---------------------------	-------------------	----------------

Efficiency

Enable Activate/Precharge look-ahead

Address Ordering

1.5.3 CORERESET_PF IP

This soft IP is instantiated as CORERESET_PF_C0_0 and is used to reset the MSS block. CORERESET_PF_C0_0 IP resets the MSS using the FABRIC_RESET_N signal. The FABRIC_RESET_N signal is asserted when the EXT_RST_N and PLL_LOCK signals are asserted.

1.5.4 PF_OSC IP

This hard IP is instantiated as PF_OSC_C0_0 (on-chip 160 MHz RC oscillator) and feeds the reference clock to the PF_CCC_C0_0. This clock source from the fabric is used to generate two clocks to drive the FIC DLLs.

1.5.5 PF_CCC IP

This hard IP is instantiated as PF_CCC_C0_0 and generates the following clocks using the 160 MHz input reference from PF_OSC_C0_0:

- OUT0_FABCLK_0 (250 MHz): Drives the FIC_2_ACLK clock signal. This clock is used by a master in the fabric for DDR operations, and the FIC_0_ACLK clock signal used by a slave in the fabric for general data transfers.
- OUT1_FABCLK_0 (100 MHz): Drives the FIC_3_PCLK clock signal. This clock is used by a APB slave in the fabric.

1.5.6 COREAPB3 IP

This soft IP is used as a bridge to connect to the fabric UART controller (CoreUARTapb_C0_0).

1.5.7 CoreUARTapb IP

This soft IP is instantiated as CoreUARTapb_C0_0 and controls the UART interface between the SoC FPGA and a host.

2 Libero Design Flow

This chapter describes the Libero design flow for running this demo design, which includes:

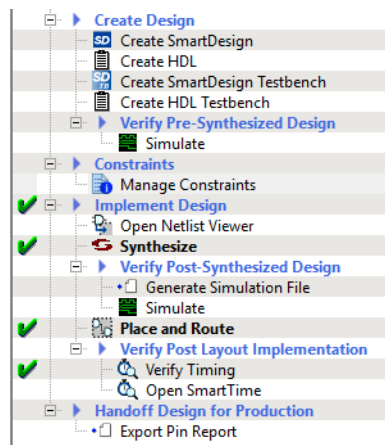
- Synthesize
- Place and Route
- Verify Timing

Note: Libero SoC EAP1 release supports up to timing verification. Future releases will support bitstream generation and programming features. To view the reference design, launch Libero SoC, use the Open Project option, and select the `hw_project.prjx` file from the following design files location:

```
<${Download_Directory}>\mpfs_ac489_df\Libero_Project
```

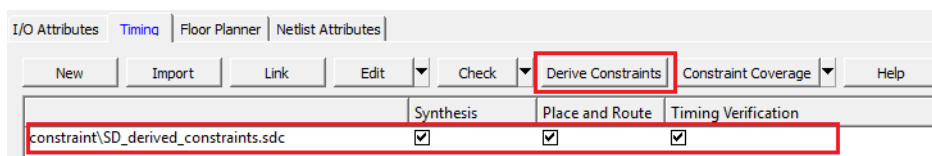
The following figure shows these options in the Design Flow tab.

Figure 11 • Libero Design Flow



Note: The timing constraints of the reference design are generated using the Derive Constraints option from **Design Flow** -> **Manage Constraints** -> **Timing** tab. The timing constraints are generated in the `SD_derived_constraints.sdc` file and it is selected for Synthesis, Place and Route, and Timing Verification as shown in [Figure 12](#).

Figure 12 • Derived Constraints



2.1 Synthesize

To synthesize the design:

1. On the Design Flow tab, double-click Synthesize. When the synthesis is successful, a green tick mark appears next to Synthesize, as shown in [Figure 11](#).
2. Right-click Synthesize and select View Report to view the synthesis report and log files in the Reports tab.

2.2 Place and Route

To place and route the design:

1. On the Design Flow tab, double-click Place and Route.
When place and route is successful, a green tick mark appears next to Place and Route, as shown [Figure 11](#).
2. Right-click Place and Route and select View Report to view the place and route report and the log files in the Reports tab.

2.3 Verify Timing

To verify timing:

On the Design Flow tab, double-click Verify Timing.

1. When the design successfully meets the timing requirements, a green tick mark appears next to Verify Timing, as shown in [Figure 11](#).
2. Right-click Verify Timing and select View Report to view the verify timing report and log files in the Reports tab.

3 Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

3.1 Revision 1.0

The first publication of this document.