

AC489
Application Note
Building the PolarFire SoC MSS Design



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

©2020 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 Design	6
1.1	Design Requirements	6
1.2	Prerequisites	6
1.3	Vault Setting in Libero	7
1.4	Block Diagram	8
1.5	Design Description	9
1.5.1	I/O Ports	9
1.5.2	Clocking Structure	10
1.5.3	Reset Structure	10
1.5.4	IP Configurations	11
2	Libero Design Flow	18
2.1	Synthesize	18
2.2	Place and Route	19
2.3	Verify Timing	19
3	Revision History	20
3.1	Revision 2.0	20
3.2	Revision 1.0	20

Figures

Figure 1	Vault Settings Option	7
Figure 2	Browse Vault Location	7
Figure 3	Block Diagram	8
Figure 4	Clocking Structure	10
Figure 5	Reset Structure	11
Figure 6	Clocks Configuration	12
Figure 7	FIC Configuration	13
Figure 8	IO Configuration	13
Figure 9	IO Overlap Warning	14
Figure 10	DDR Topology	14
Figure 11	DDR Memory Initialization	15
Figure 12	DDR Memory Timing	16
Figure 13	DDR Controller	16
Figure 14	Boot Mode	17
Figure 15	Libero Design Flow	18
Figure 16	Derived Constraints	18

Tables

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

1 Building the MSS Design

Microchip's PolarFire SoC FPGAs include industry's first RISC-V based Microprocessor Subsystem (MSS) and a fabric that inherits all the features of the PolarFire family. The PolarFire SoC 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 design in PolarFire SoC FPGAs.

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

Table 1 • Design Requirements¹

Software	Description
-Libero SoC	v12.4
Hardware	
-Host PC	Windows 10
License	
-Libero Gold License or higher	Required for Libero SoC
-PolarFire SoC License	Required for the PolarFire SoC family

1. Programming the reference design on a Hardware Kit will be supported in future releases.

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. To work offline with Libero SoC, 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. To change the vault setting, see [Vault Setting in Libero](#).

1.3 Vault Setting in Libero

To work offline with Libero SoC, the vault location must be set to the Mega Vault in Libero SoC.

Figure 1 • Vault Settings Option

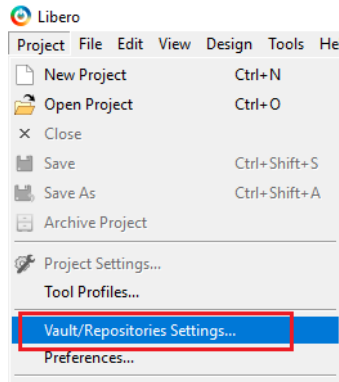
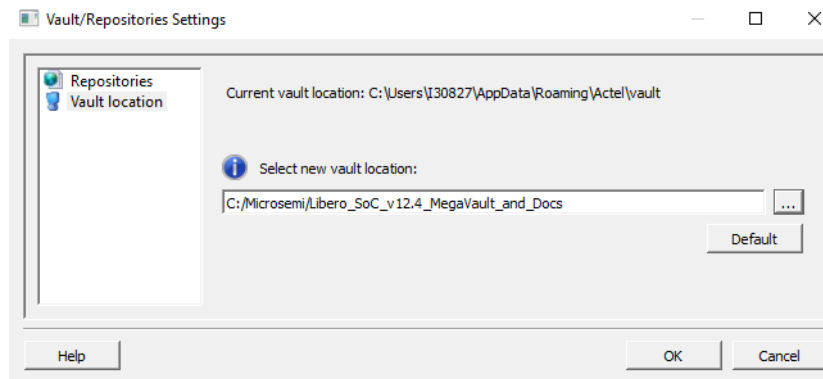


Figure 2 • Browse Vault Location



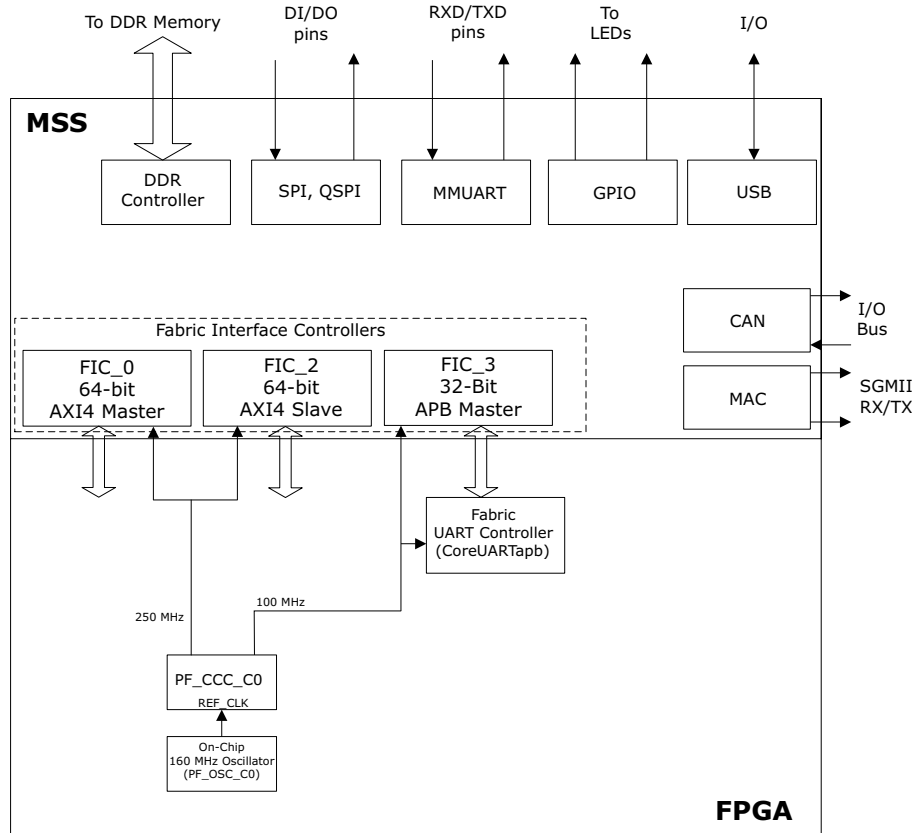
Note: Download the latest version of the MegaVault from the [Libero SoC](#) page.

1.4 Block Diagram

The MSS includes a wide range of peripherals and in the reference design, the MSS is configured by enabling DDR controller, USB, CAN, QSPI, MAC, 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 design.

For more information about the PolarFire SoC MSS and its components, see [UG0880: PolarFire SoC FPGA MSS User Guide](#).

Figure 3 • Block Diagram



1.5 Design Description

The MSS design 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)

For more details regarding the IP blocks used and their configuration, open the design files provided along with the reference design.

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	Input	UART interface to the Host PC from the FPGA
TX	Output	UART interface to the FPGA from the Host PC
SPI_1_DI	Input	SPI_1 Data In interface to the MSS
SPI_1_DO	Output	SPI_1 Data Out interface from the MSS
MAC_0_MDC MAC_0_MDIO	Output Inout	Management interface to/from Ethernet PHY
MMUART_1_RXD	Input	MMUART_1 interface to the Host PC from the MSS
MMUART_1_TXD	Output	MMUART_1 interface to the MSS from the Host PC
QSPI_DATA_F2M[3:0]	Input	QSPI Data In from fabric SPI Controller
QSPI_DATA_M2F[3:0]	output	QSPI Data Out to fabric SPI Controller
CAN_1_RXBUS	Input	CAN_1 input data
CAN_1_TXBUS	Output	CAN_1 output data
USB_CLK	Input	USB clock
USB_DIR	Input	USB data to host PC
USB_NXT	Input	Throttle signal from the USB PHY indicating next data
USB_DATA7:0	Inout	USB data bus
GPIO_0_13_IN	Input	User input
GPIO_1_22_IN	Input	User input
GPIO_1_12_IN	Input	User input
CA[5:0] CK CKE CK_N CS DM[3:0] DQ[31:0] DQS[3:0] DQS_N[3:0] ODT	To DDR memory	SDRAM signals

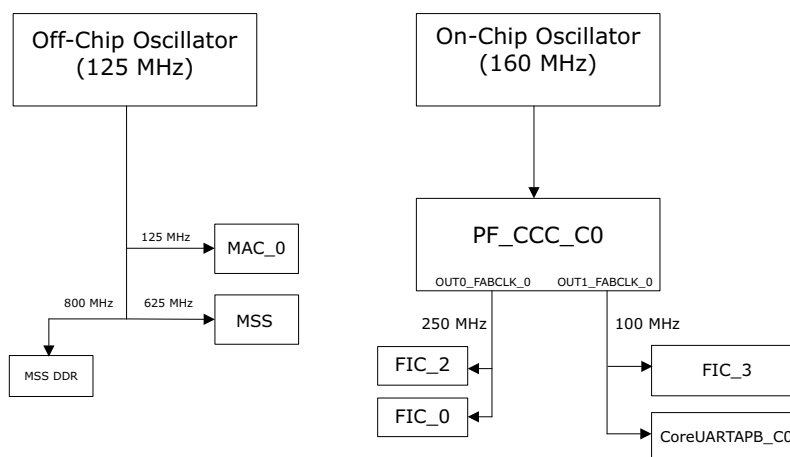
1.5.2 Clocking Structure

There are two clock sources in the design— an off-chip 125 MHz oscillator and an on-chip 160 MHz oscillator. The off-chip 125 MHz oscillator is used to source the reference clock for the MSS DDR (800 MHz), CPU Core Complex (600 MHz), and MAC_0. The on-chip 160 MHz clock source is used to source the reference clock for CoreUARTapb_C0 and FIC_3. OUT0_FABCLK0 (250 MHz) clock of the PF_CCC_C0 drives the FIC_2 clock. OUT1_FABCLK0 (100 MHz) clock of the PF_CCC_C0 drives the FIC_0 and CoreUARTAPB_C0 clocks.

The 625 MHz reference clock is further internally divided to source the AXI (312 MHz), AHB/APB (156 MHz), and eMMC clock (200 MHz) in MSS.

Figure 4 shows the clocking structure of the design.

Figure 4 • Clocking Structure

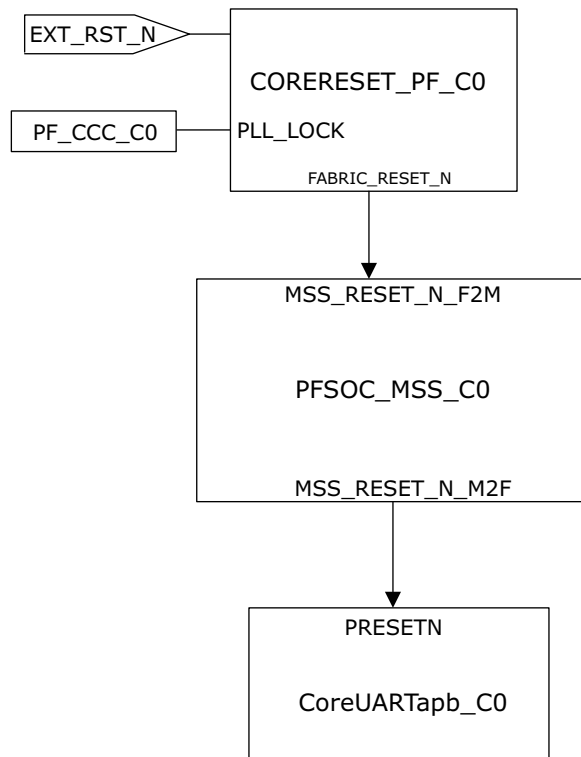


1.5.3 Reset Structure

The MSS reset is controlled by the FABRIC_RESET_N signal, which is asserted by ANDING the EXT_RST_N and the PLL_LOCK signal using the CORERESET_PF_C0 IP. After reset, MSS sends the MSS_RESET_N_M2F reset signal to the fabric to release the CoreUARTapb_C0 IP out of reset.

Figure 5 shows the reset structure of the design.

Figure 5 • Reset Structure



1.5.4 IP Configurations

This section describes each IP block of the MSS design.

1.5.4.1 PFSOC_MSS SgCore IP

This IP instantiates the MSS as **PFSOC_MSS_C0_0**. The MSS is configured with DDR Controller, FIC0, FIC2, FIC3, GPIO, SPI, and MMUART.

The MSS Configurator is used to enable the following configurations:

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

1.5.4.1.1 Clocks Configuration

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

- **MSS Reference Clock Input Source**—This is set to dedicated I/Os from Bank 5 (REFCLK) to source the reference clock for the MSS PLL.
- **MSS PLL Clock Frequency**—This is set to 625 MHz. The CPU cores can operate at a maximum frequency of 625 MHz.
- **MSS CPU cores clock frequency Divider**—This is set to /1 divider.
- **MSS AXI clock frequency Divider**—This is set to /2 divider for a 312 MHz AXI clock frequency.
- **MSS AHB/APB clock frequency Divider**—This is set to /4 divider for a 156 MHz AHB/APB clock frequency.
- **MSS CAN clock frequency**—This is set to 80 MHz. This option is exposed only when CAN is selected in the Configurator.

- **eMMC/SD/SDIO clock frequency (MHz)**—This is set to 200 MHz. This option is exposed only when eMMC.SD/SDIO is selected in the Configurator.
- **DDR Reference Clock Input Source**—This is set to dedicated I/Os from Bank 5 (REFCLK) to source the DDR reference clock.
- **MAC SGMII Reference Clock Input Source**—This is set to Dedicated I/Os from BANK 5 (REFCLK) to source the SGMII reference clock. This option is exposed only when MAC is selected in the Configurator.
- **Clock Sources Frequency**—This is set to 125 MHz to source from a 125 MHz off-chip oscillator.

Note: The **DDR Reference Clock Input Source** option is displayed only when the DDR type is selected from the **DDR Topology** tab.

Figure 6 shows the clocks configuration used in the reference design.

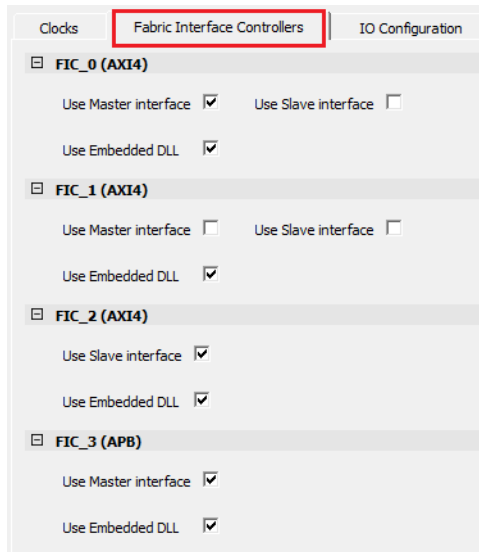
Figure 6 • Clocks Configuration

Clocks	Fabric Interface Controllers	IO Configuration	I/O REFCLK	I/O Bank4	I/O Bank2	DDR Topology	DI																										
<div style="border: 1px solid #ccc; padding: 5px;"> <div style="background-color: #f2f2f2; padding: 2px;">⊟ MSS</div> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">MSS Reference Clock Input Source</td> <td style="width: 20%;">Dedicated I/O from Bank5 (REFCLK) ▾</td> <td style="width: 20%;">MSS PLL clock frequency (Mhz)</td> <td style="width: 10%;">625</td> </tr> <tr> <td>MSS CPU cores clock frequency Divider</td> <td>/1 ▾</td> <td>MSS CPU cores clock frequency (Mhz)</td> <td>625</td> </tr> <tr> <td>MSS AXI clock frequency Divider</td> <td>/2 ▾</td> <td>MSS AXI clock frequency (Mhz)</td> <td>312</td> </tr> <tr> <td>MSS AHB/APB clock frequency Divider</td> <td>/4 ▾</td> <td>MSS AHB/APB clock frequency (Mhz)</td> <td>156</td> </tr> <tr> <td>MSS CAN clock frequency (Mhz)</td> <td>80 ▾</td> <td>eMMC/SD/SDIO clock frequency (Mhz)</td> <td>200</td> </tr> </table> <div style="background-color: #f2f2f2; padding: 2px; margin-top: 5px;">⊟ Gigabit Ethernet MAC</div> <table style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <tr> <td>MAC SGMII Reference Clock Input Source</td> <td>Dedicated I/O from Bank5 (REFCLK) ▾</td> </tr> </table> <div style="background-color: #f2f2f2; padding: 2px; margin-top: 5px;">⊟ DDR</div> <table style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <tr> <td>DDR Reference Clock Input source</td> <td>Dedicated I/O from Bank5 (REFCLK) ▾</td> </tr> </table> <div style="background-color: #f2f2f2; padding: 2px; margin-top: 5px;">⊟ Clock Sources Frequency</div> <table style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <tr> <td>Dedicated I/O from Bank5 (REFCLK) frequency (MHz)</td> <td>125 ▾</td> </tr> </table> </div>								MSS Reference Clock Input Source	Dedicated I/O from Bank5 (REFCLK) ▾	MSS PLL clock frequency (Mhz)	625	MSS CPU cores clock frequency Divider	/1 ▾	MSS CPU cores clock frequency (Mhz)	625	MSS AXI clock frequency Divider	/2 ▾	MSS AXI clock frequency (Mhz)	312	MSS AHB/APB clock frequency Divider	/4 ▾	MSS AHB/APB clock frequency (Mhz)	156	MSS CAN clock frequency (Mhz)	80 ▾	eMMC/SD/SDIO clock frequency (Mhz)	200	MAC SGMII Reference Clock Input Source	Dedicated I/O from Bank5 (REFCLK) ▾	DDR Reference Clock Input source	Dedicated I/O from Bank5 (REFCLK) ▾	Dedicated I/O from Bank5 (REFCLK) frequency (MHz)	125 ▾
MSS Reference Clock Input Source	Dedicated I/O from Bank5 (REFCLK) ▾	MSS PLL clock frequency (Mhz)	625																														
MSS CPU cores clock frequency Divider	/1 ▾	MSS CPU cores clock frequency (Mhz)	625																														
MSS AXI clock frequency Divider	/2 ▾	MSS AXI clock frequency (Mhz)	312																														
MSS AHB/APB clock frequency Divider	/4 ▾	MSS AHB/APB clock frequency (Mhz)	156																														
MSS CAN clock frequency (Mhz)	80 ▾	eMMC/SD/SDIO clock frequency (Mhz)	200																														
MAC SGMII Reference Clock Input Source	Dedicated I/O from Bank5 (REFCLK) ▾																																
DDR Reference Clock Input source	Dedicated I/O from Bank5 (REFCLK) ▾																																
Dedicated I/O from Bank5 (REFCLK) frequency (MHz)	125 ▾																																

1.5.4.1.2 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 7.

Figure 7 • FIC Configuration



Note: When a master interface is enabled for a FIC, that master interface must be connected to a slave in the fabric. When a slave interface is enabled for a FIC, that slave interface must be connected to a master in the fabric.

1.5.4.1.3 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

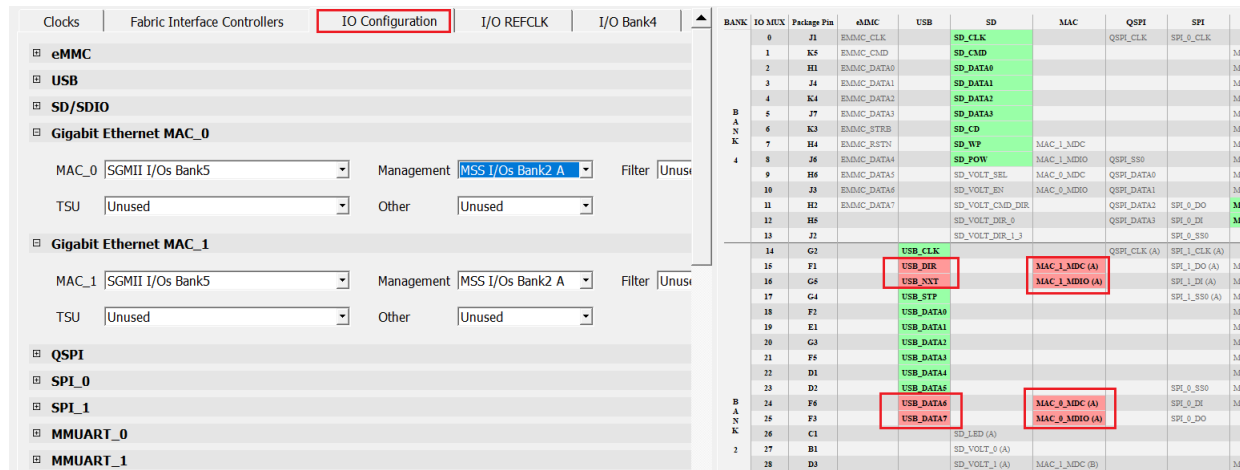
The MSS uses MMUART_1, SPI_1, CAN_1, GPIO_0_13 (Bank 4), GPIO_1_12 and GPIO_1_22 (Bank 2), QSPI, USB, and MAC_0 as shown in Figure 8.

Figure 8 • IO Configuration

BANK	IO	MEM	Package Pin	oMNC	USB	SD	MAC	QSPI	SPI	MMUART	I2C	CAN	GPIO
0	J1			EMMC_CLK		SD_CLK		QSPI_CLK	SPI_CLK				GPIO_0_0
1	K9			EMMC_CMD		SD_CMD				MMUART_3_RXD0	I2C_0_SCL		GPIO_0_1
2	H1			EMMC_DATA0		SD_DATA0				MMUART_3_TXD0	I2C_0_SDA		GPIO_0_2
3	J4			EMMC_DATA1		SD_DATA1				MMUART_4_RXD0		CAN_0_TXD0B	GPIO_0_3
4	K4			EMMC_DATA2		SD_DATA2				MMUART_4_TXD0		CAN_0_RXD0B	GPIO_0_4
5	J7			EMMC_DATA3		SD_DATA3				MMUART_0_RXD0 (A)		CAN_1_TX_RXB_N	GPIO_0_5
6	K3			EMMC_STBE		SD_CD				MMUART_0_TXD0 (A)			GPIO_0_6
7	H4			EMMC_RSTN		SD_WP	MAC_1_MDC			MMUART_2_RXD0	I2C_1_SCL		GPIO_0_7
8	J6			EMMC_DATA4		SD_POW	MAC_1_MIO0	QSPI_S00		MMUART_2_TXD0	I2C_1_SDA		GPIO_0_8
9	H6			EMMC_DATA5		SD_VOLT_0RE	MAC_0_MIO0	QSPI_DATA0		MMUART_3_RXD0 (B)			GPIO_0_9
10	J3			EMMC_DATA6		SD_VOLT_0N	MAC_0_MIO0	QSPI_DATA1		MMUART_0_TXD0 (B)			GPIO_0_10
11	H2			EMMC_DATA7		SD_VOLT_CMD_0ER		QSPI_DATA2	SPI_0_DO	MMUART_1_RXD0		CAN_1_TXD0B	GPIO_0_11
12	H8					SD_VOLT_0R_0		QSPI_DATA3	SPI_0_DE	MMUART_1_TXD0		CAN_1_RXD0B	GPIO_0_12
13	J2					SD_VOLT_0R_1_3			SPI_0_S00			CAN_1_TX_RXB_N	GPIO_0_13
14	G2			USB_CLK				QSPI_CLK (A)	SPI_1_CLK (A)				GPIO_1_0
15	F1			USB_0R			MAC_1_MDC (A)	SPI_1_DO (A)		MMUART_4_RXD0			GPIO_1_1
16	G4			USB_NXT			MAC_1_MIO0 (A)	SPI_1_DP (A)		MMUART_4_TXD0			GPIO_1_2
17	G4			USB_STP				SPI_1_S00 (A)		MMUART_0_RXD0 (A)			GPIO_1_3
18	F2			USB_DATA0						MMUART_0_TXD0 (A)			GPIO_1_4
19	E1			USB_DATA1						MMUART_1_RXD0			GPIO_1_5
20	G3			USB_DATA2						MMUART_1_TXD0	I2C_0_SCL (A)		GPIO_1_6
21	F5			USB_DATA3						MMUART_2_RXD0	I2C_0_SDA (A)	CAN_0_TX_RXB_N (A)	GPIO_1_7
22	D1			USB_DATA4						MMUART_2_TXD0		CAN_0_TXD0B (A)	GPIO_1_8
23	D1			USB_DATA5						MMUART_1_RXD0		CAN_0_RXD0B (A)	GPIO_1_9
24	F6			USB_DATA6			MAC_0_MDC (A)	SPI_0_S00		MMUART_3_RXD0	I2C_1_SCL (A)		GPIO_1_10
25	F3			USB_DATA7			MAC_0_MIO0 (A)	SPI_0_DE		MMUART_3_TXD0	I2C_1_SDA (A)		GPIO_1_11
26	C1					SD_LED (A)					I2C_1_SCL (B)		GPIO_1_12
27	H1					SD_VOLT_0 (A)					I2C_1_SDA (B)		GPIO_1_13
28	D3					SD_VOLT_1 (A)	MAC_1_MDC (B)			MMUART_0_RXD0 (B)		CAN_1_TX_RXB_N (A)	GPIO_1_14
29	C1					SD_VOLT_2 (A)	MAC_1_MIO0 (B)			MMUART_0_TXD0 (B)		CAN_1_RXD0B (A)	GPIO_1_15
30	E5							QSPI_CLK (B)	SPI_1_CLK (B)				GPIO_1_16
31	E4							QSPI_S00	SPI_1_S00 (B)			CAN_0_TXD0B (B)	GPIO_1_17
32	H5					SD_CLK		QSPI_DATA2	SPI_1_DO (B)			CAN_0_RXD0B (B)	GPIO_1_18
33	A2					SD_LED (B)				MMUART_3_RXD0 (C)		CAN_0_TX_RXB_N (B)	GPIO_1_19
34	B3					SD_VOLT_0 (B)		QSPI_DATA3				CAN_1_TXD0B (B)	GPIO_1_20
35	A3					SD_VOLT_1 (B)	MAC_0_MDC (B)	QSPI_DATA3		MMUART_0_RXD0 (C)	I2C_0_SCL (B)		GPIO_1_21
36	E3					SD_VOLT_2 (B)	MAC_0_MIO0 (B)			MMUART_0_TXD0 (C)	I2C_0_SDA (B)	CAN_1_TX_RXB_N (B)	GPIO_1_22
37	D4							QSPI_CLK (C)	SPI_0_CLK				GPIO_1_23

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 9.

Figure 9 • IO Overlap Warning



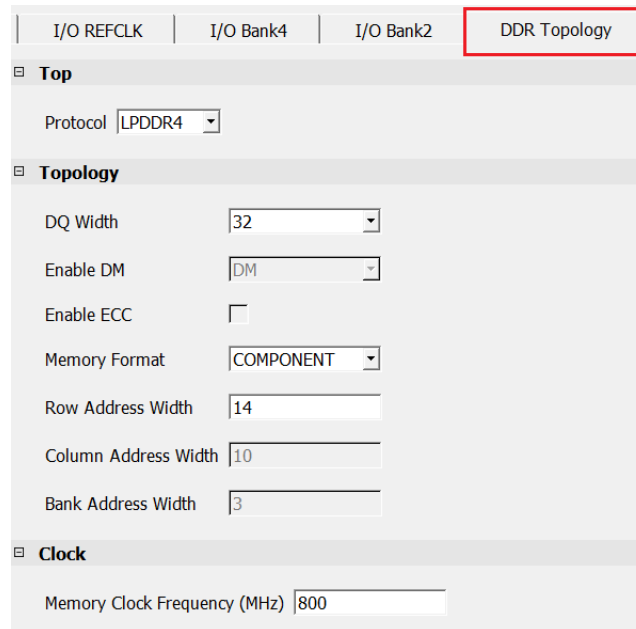
The screenshot shows the IO Configuration tool with the 'IO Configuration' tab selected. On the left, a tree view shows various peripheral categories like eMMC, USB, SD/SDIO, Gigabit Ethernet MAC_0, Gigabit Ethernet MAC_1, QSPI, SPI_0, SPI_1, MMUART_0, and MMUART_1. On the right, a table lists I/Os for Bank 4. Red boxes highlight overlapping I/Os: USB_CLK, USB_DIR, USB_NXT, USB_STP, USB_DATA0, USB_DATA1, USB_DATA2, USB_DATA3, USB_DATA4, USB_DATA5, USB_DATA6, and USB_DATA7. These are overlapped with MAC_1_MDC (A), MAC_1_MDI0 (A), MAC_0_MDC (A), and MAC_0_MDI0 (A).

BANK	I/O MUX	Package Pin	eADIC	USB	SD	MAC	QSPI	SPI
0	J1		EMMC_CLK		SD_CLK		QSPI_CLK	SPI_0_CLK
1	K5		EMMC_CMD		SD_CMD			
2	H1		EMMC_DATA0		SD_DATA0			
3	J4		EMMC_DATA1		SD_DATA1			
4	K4		EMMC_DATA2		SD_DATA2			
5	J7		EMMC_DATA3		SD_DATA3			
6	K3		EMMC_STB		SD_CD			
7	H4		EMMC_RSTN		SD_WF	MAC_1_MDC		
8	J6		EMMC_DATA4		SD_POW	MAC_1_MDI0	QSPI_SS0	
9	H5		EMMC_DATA5		SD_VOLT_SEL	MAC_0_MDC	QSPI_DATA0	
10	J3		EMMC_DATA6		SD_VOLT_SEN	MAC_0_MDI0	QSPI_DATA1	
11	H2		EMMC_DATA7		SD_VOLT_CMD_DIR		QSPI_DATA2	SPI_0_DO
12	H5				SD_VOLT_DIR_0		QSPI_DATA3	SPI_0_DI
13	J2				SD_VOLT_DIR_1_3			SPI_0_SS0
14	G2			USB_CLK			QSPI_CLK (A)	SPI_1_CLK (A)
15	F1			USB_DIR		MAC_1_MDC (A)		SPI_1_DO (A)
16	G8			USB_NXT		MAC_1_MDI0 (A)		SPI_1_DI (A)
17	G4			USB_STP				SPI_1_SS0 (A)
18	F2			USB_DATA0				
19	E1			USB_DATA1				
20	G3			USB_DATA2				
21	F8			USB_DATA3				
22	D1			USB_DATA4				
23	D2			USB_DATA5				SPI_0_SS0
24	F6			USB_DATA6		MAC_0_MDC (A)		SPI_0_DI
25	F3			USB_DATA7		MAC_0_MDI0 (A)		SPI_0_DO
26	C1							
27	B1				SD_LED (A)			
28	D3				SD_VOLT_0 (A)			
29	D3				SD_VOLT_1 (A)	MAC_1_MDC (B)		

1.5.4.1.4 DDR Configuration

The reference design configures the DDR controller for LPDDR4 memory. Figure 10, Figure 11, Figure 12, and Figure 13 show the DDR Topology, memory initialization, memory timing, and address ordering used.

Figure 10 • DDR Topology



The screenshot shows the 'DDR Topology' configuration window. The 'Protocol' is set to 'LPDDR4'. Under 'Topology', the following settings are visible: DQ Width (32), Enable DM (DM), Enable ECC (unchecked), Memory Format (COMPONENT), Row Address Width (14), Column Address Width (10), and Bank Address Width (3). Under 'Clock', the Memory Clock Frequency (MHz) is set to 800.

Figure 11 • DDR Memory Initialization

Clocks | Fabric Interface Controllers | IO Configuration | DDR Topology | **DDR Memory Initialization**

Mode Register 1

RD Pre-amble Type: Static

RD Post-amble Length: 0.5*tCK

Mode Register 2

Read Latency (#RL, #nRTP): RL=6, nRTP = 8

Write Latency: WL=4

Mode Register 3

Pull-up Calibration Point: VDDQ/3

WR Post-amble Length: 0.5*tCK

Pull-Down Drive Strength: RZQ/6

Mode Register 4

Self Refresh Abort Mode: Disabled

Mode Register 11

DQ ODT: Disable

CA ODT: Disable

Mode Register 22

SoC ODT: Disable

ODTE-CK: Disabled

ODTE-CS: Disabled

ODTE-CA: Obeys ODT_CA bond pad

Figure 12 • 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 13 • DDR Controller

I/O REFCLK	I/O Bank4	I/O Bank2	DDR Topology	DDR Memory Initialization	DDR Memory Timing	DDR Controller
------------	-----------	-----------	--------------	---------------------------	-------------------	----------------

Efficiency

Enable Activate/Precharge look-ahead

Address Ordering

Drive

DQ Drive DQS Drive

ADD/CMD Drive Clock Drive

ODT

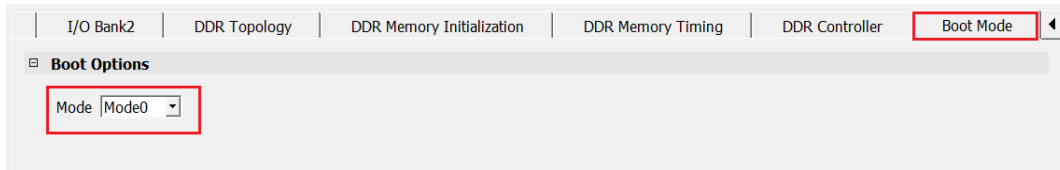
DQ ODT

DQS ODT

1.5.4.1.5 Boot Mode

Boot Mode option is set to **Mode0**. **Mode0** is non-secure boot mode. In this mode, the MSS Processor E51 monitor core boots from the boot code stored in the eNVM (Boot Flash).

Figure 14 • Boot Mode



For more information about MSS boot modes, see [UG0881: PolarFire SoC FPGA Booting And Configuration User Guide](#).

1.5.4.2 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.

1.5.4.3 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.

1.5.4.4 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.

1.5.4.5 COREAPB3 IP

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

1.5.4.5.1 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:

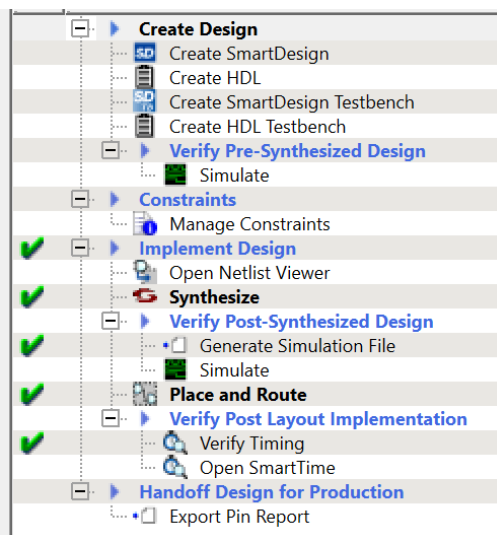
- Synthesize
- Place and Route
- Verify Timing

Note: The reference design will be targeted for hardware in future releases. This release of Libero SoC supports up to timing verification. 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\hw_project
```

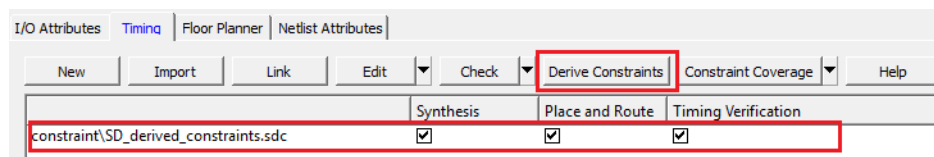
The following figure shows these options in the Design Flow tab.

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

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 2.0

Updated for Libero SoC 12.4.

3.2 Revision 1.0

The first publication of this document.