AC489
Application Note
Building the PolarFire SoC MSS Subsystem

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

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.

51900489. 1.0 12/19

Contents

Figures

Tables

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 -PolarFire SoC License	Required for Libero SoC 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**

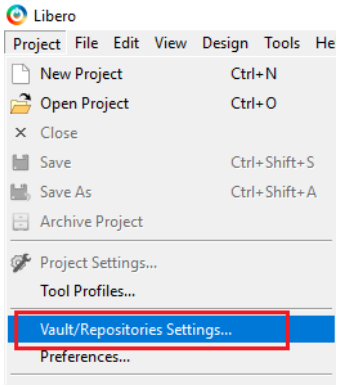
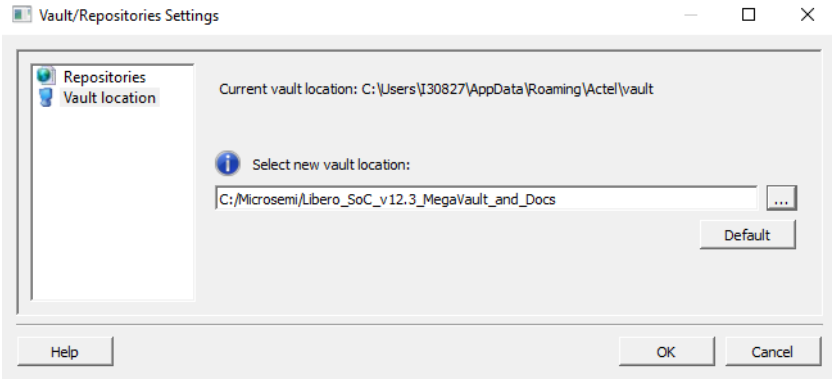


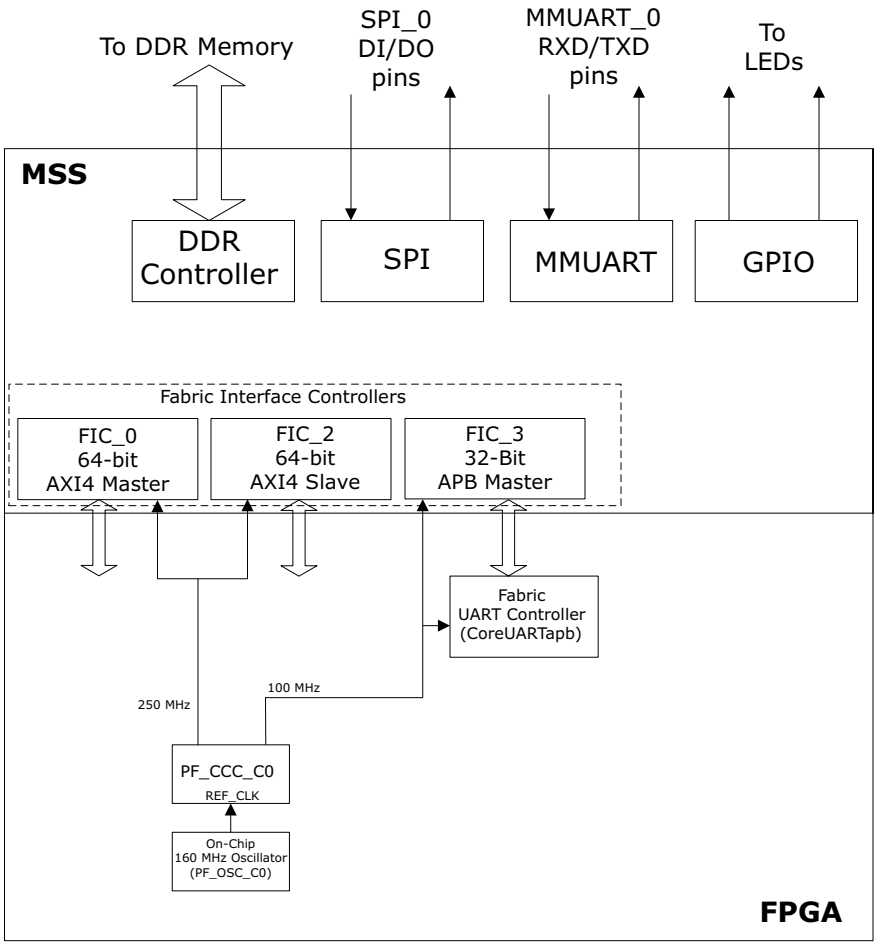
Figure 2 • **Browse Vault Location**



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.

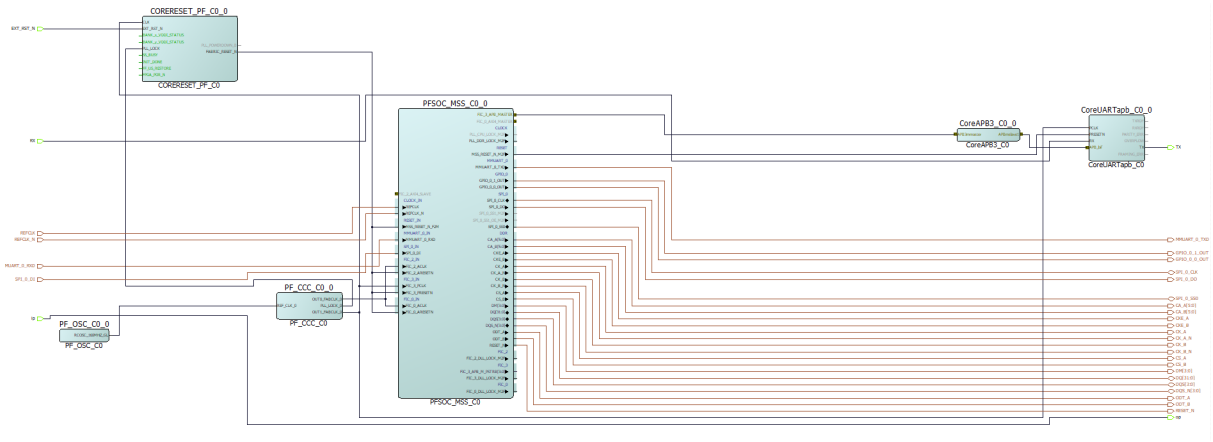
Figure 3 • **Block Diagram**



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

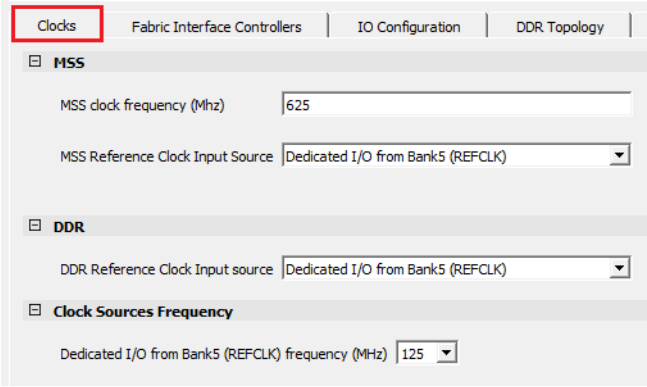
1.5.2.1 Clocks Configuration

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.

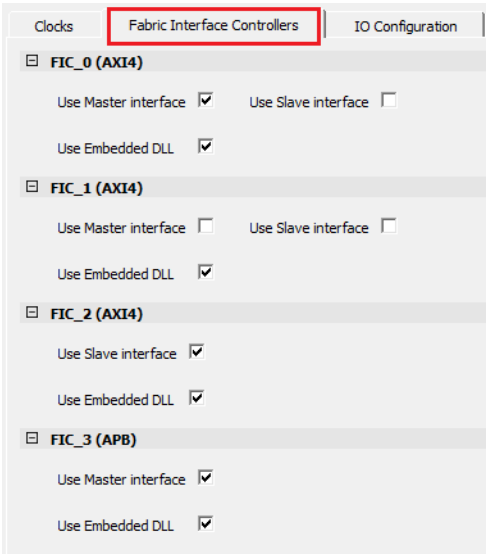
Figure 5 • **Clocks Configuration**



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

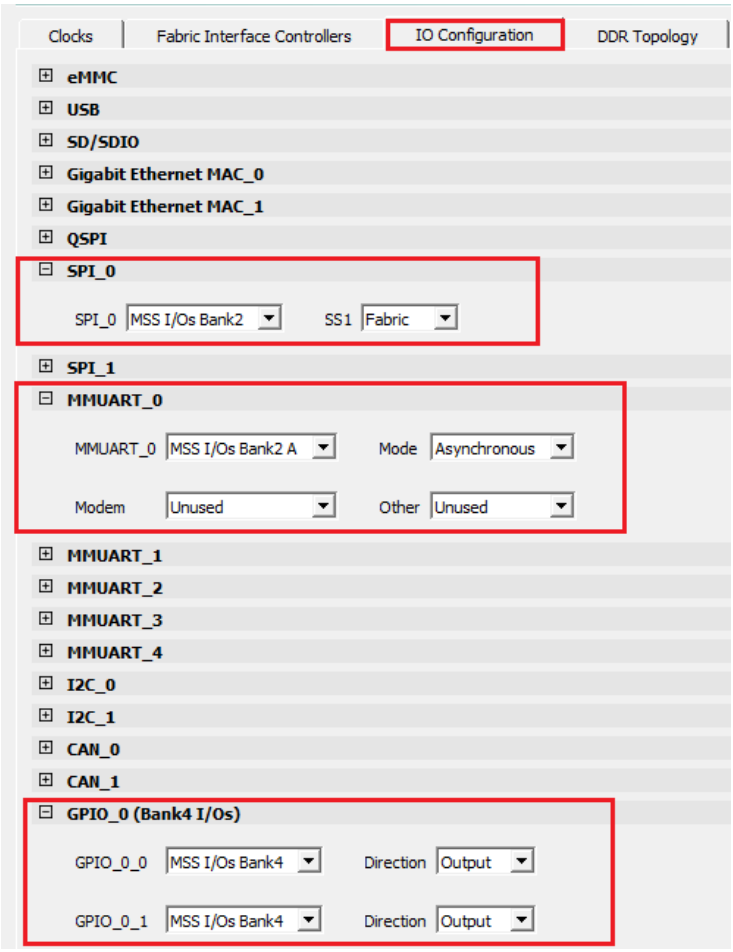
MSS supports a wide range of peripherals. The MSS Configurator -> IO Configuration tab is used to select the peripherals and assign the I/Os for those peripherals. By default, all peripherals are marked as unused. To include peripherals that are required in the design, use the drop-down corresponding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

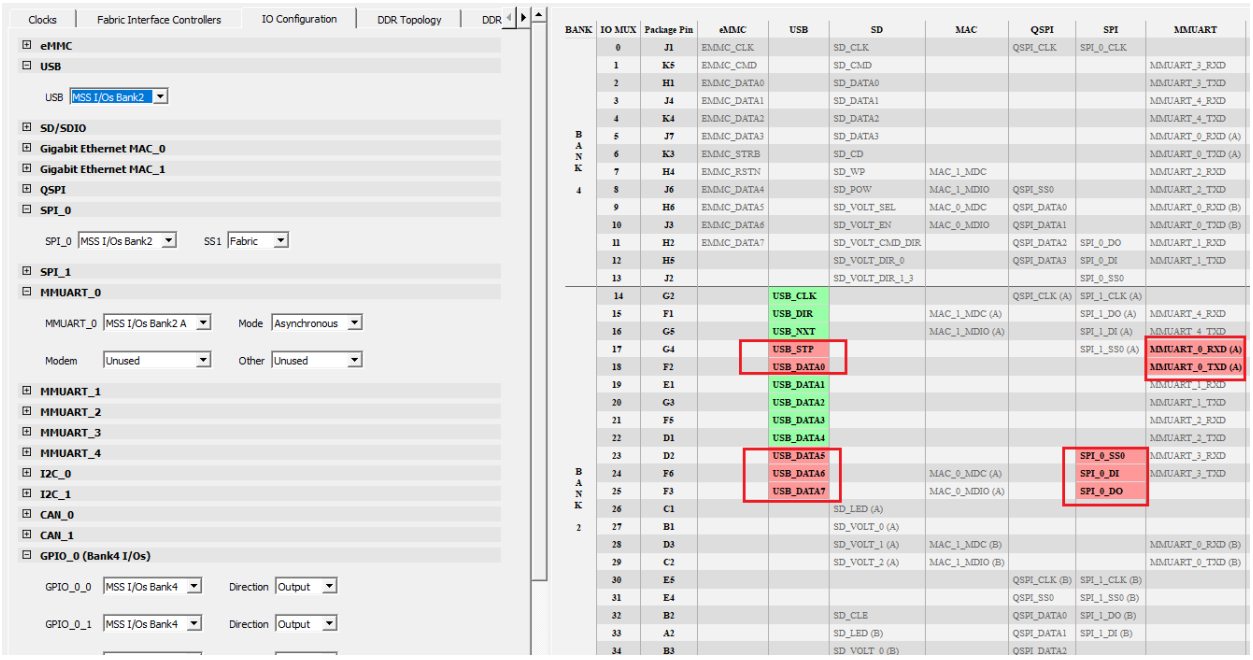
The reference design uses MMUART, GPIO, SPI as shown in Figure 7.

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**



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**

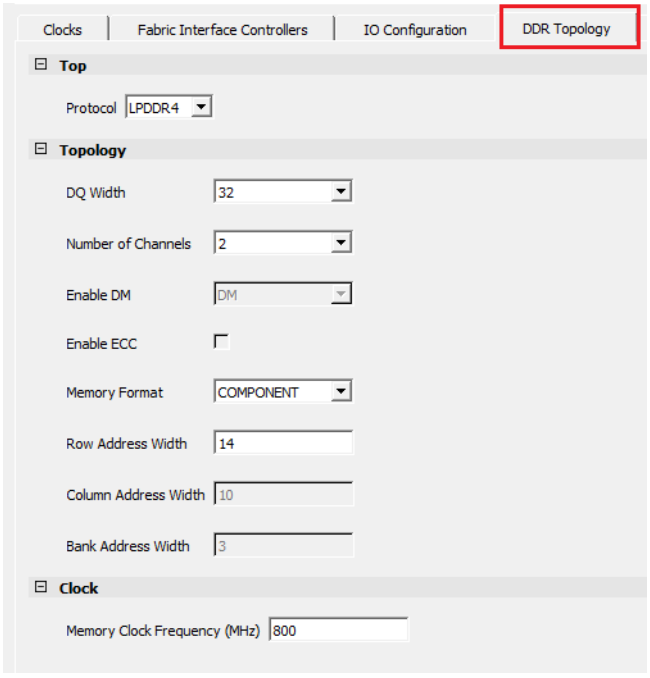


Figure 10 • **DDR Memory Initialization**

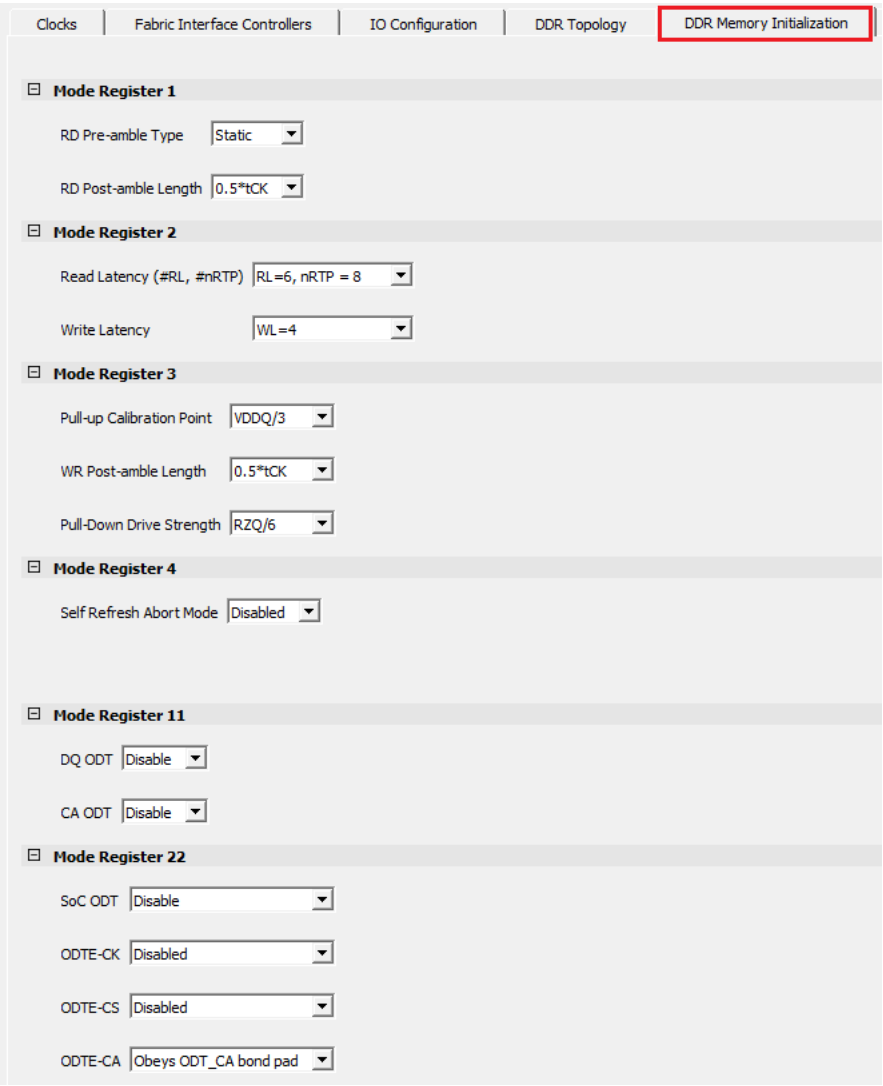


Figure 11 • **DDR Memory Timing**

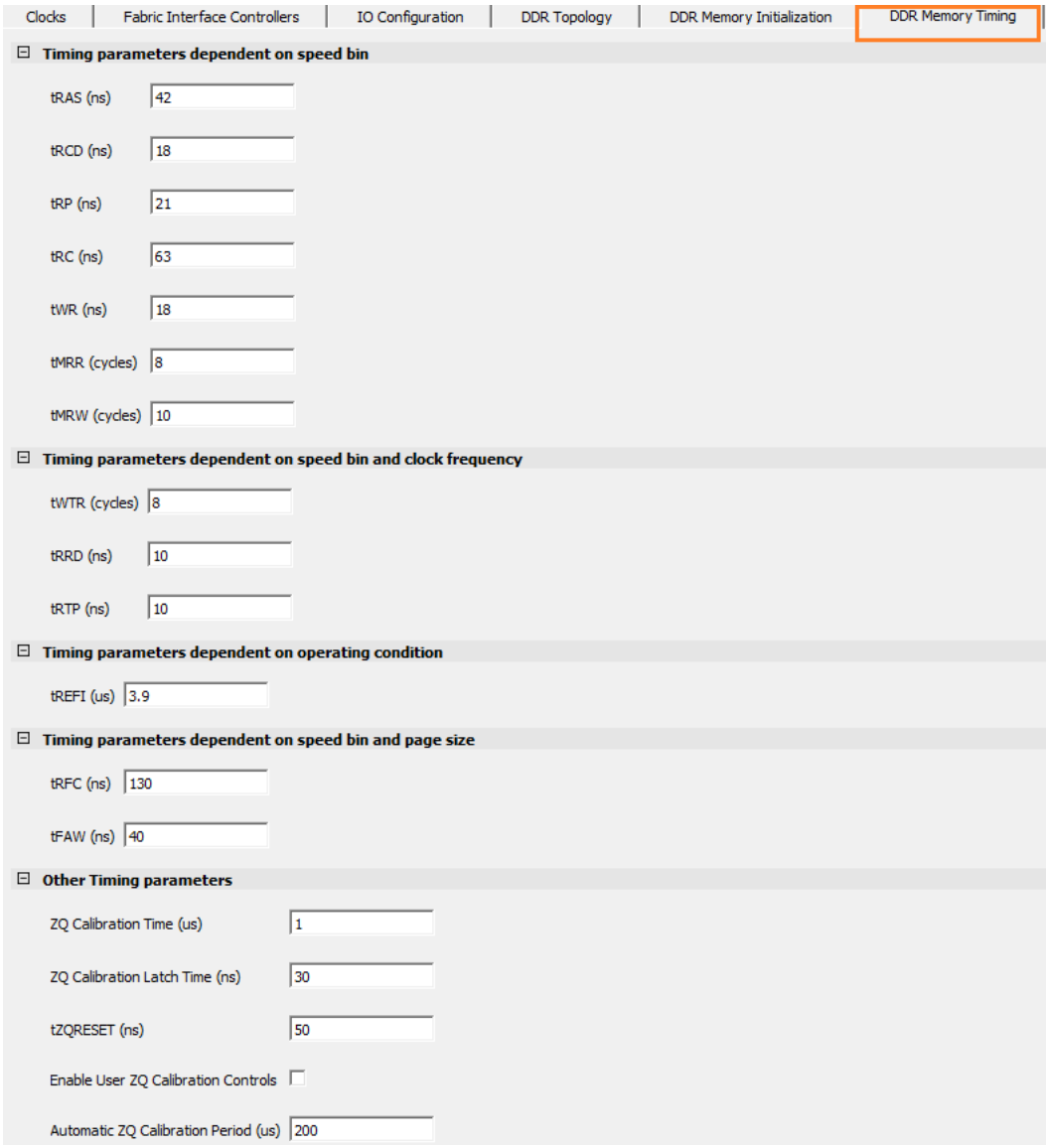


Figure 12 • **DDR Controller Efficiency**

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:

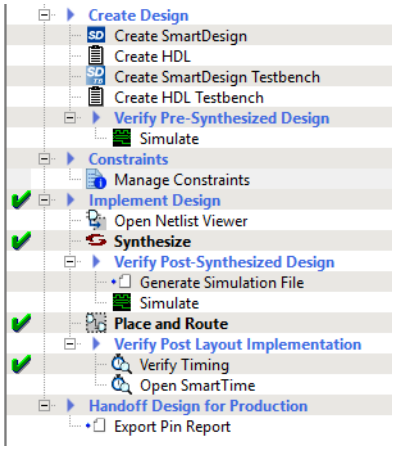
- Synthesize
- Place and Route
- Verify Timing

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 flies location:

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

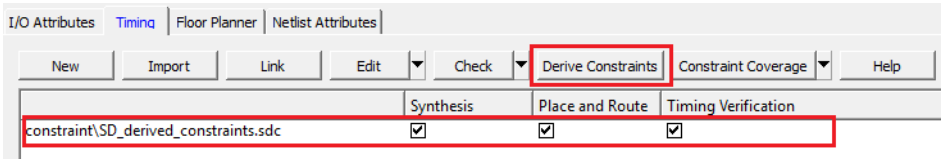
The following figure shows these options in the Design Flow tab.

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.