DG0534
Demo Guide
Interfacing SmartFusion2 SOC and IGLOO2 FPGA with External LPDDR Memory through MDDR Controller

Microsemi
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

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.

50200534. 10.0 4/21

Contents

Figures

Tables

1 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.

1.1 Revision 10.0

Updated the document for Libero SoC v12.6.

1.2 Revision 9.0

The following is a summary of the changes made in this revision.

- Updated the document for Libero SoC v12.5.
- Removed the references to Libero version numbers.

1.3 Revision 8.0

Merged SmartFusion2 related content and updated the document for Libero SoC v12.2 software release.

1.4 Revision 7.0

Updated the document for Libero v11.8 SP2 software release.

1.5 Revision 6.0

Updated the document for Libero v11.7 software release changes (SAR 76992).

1.6 Revision 5.0

Changed MDDR_CLK: DDR_FIC_CLK ratio to 1:1 and updated Figure 6 on page 13, Figure 12 on page 17, and Figure 13 on page 17 (SAR 73229).

1.7 Revision 4.0

Updated the document for Libero v11.6 software release changes (SAR 72065).

1.8 Revision 3.0

Updated the document for Libero SoC v11.5 (SAR 65209).

1.9 Revision 2.0

Updated the document for Libero SoC v11.4.

1.10 Revision 1.0

Initial release.

2 Preface

2.1 Purpose

This demo guide describes the SmartFusion®2 SoC and IGLOO®2 FPGA devices. It provides instructions on how to use the corresponding reference design.

2.2 Intended Audience

This demo guide is intended for:

- FPGA designers
- System-level designers

2.3 References

The following documents are referred in this demo guide:

- *UG0446: SmartFusion2 and IGLOO2 FPGA High Speed DDR Interfaces User Guide*
- *UG0448: IGLOO2 FPGA High Performance Memory Subsystem User Guide*
- *IGLOO2 System Builder User Guide*
- *UG0478: IGLOO2 Evaluation Kit User Guide*
- *CoreUART Handbook*

For updates and additional information about the device information, visit *https://www.microsemi.com/product-directory/fpga-soc/1638-fpgas*

3 Interfacing SmartFusion2 SOC and IGLOO2 FPGA with External LPDDR Memory through MDDR Controller

3.1 Introduction

This demo shows that the High-Performance Memory Subsystem (HPMS) and the Microcontroller Subsystem (MSS) Double Data Rate (DDR) controller accessing the external DDR Synchronous Dynamic Random Access Memory (SDRAM) memories in the SmartFusion2 and IGLOO2 devices.

This demo design has two parts:

- Using the simulation
- Using the SmartFusion2 Security or IGLOO2 Evaluation Kit

In the demo design, the Advanced eXtensible Interface (AXI) master in the FPGA fabric accesses the Low Power DDR (LPDDR) memory present in the SmartFusion2 Security and IGLOO2 Evaluation Kit board, using the Microcontroller/Memory subsystem Double Data Rate (MDDR) controller. A utility, IGL2_MDDR_Demo, and SF2_MDDR_Demo is provided with the demo deliverables. Using the utility, you can drive the AXI master logic. The AXI master converts the commands from the utility to AXI transactions for the MDDR controller to perform the read/write operations on the LPDDR memory.

3.2 Design Requirements

The following table lists the resources required to run the demo:

Table 1 • **Design Requirements**

Requirement	**Version**
Operating System	64 bit Windows 7 and 10
Hardware	
SmartFusion2 Security or IGLOO2 Evaluation Kit: • FlashPro4 programmer • 12 V adapter • USB A to Mini-B cable	• IGLOO2: Rev C or later • SmartFusion2: Rev E or later
Software	
FlashPro Express	**Note:** Refer to the `readme.txt` file provided in the design files for the software versions used with this reference design
Libero® System-on-Chip (SoC)	
Host PC Drivers USB to UART drivers	USB to UART drivers

Note: Libero SmartDesign and configuration screen shots shown in this guide are for illustration purpose only. Open the Libero design to see the latest updates.

3.3 Prerequisites

Before you begin:

1. Download and install Libero SoC (as indicated in the website for this design) on the host PC from the following location.
   https://www.microsemi.com/product-directory/design-resources/1750-libero-soc

   The latest versions of ModelSim, Synplify Pro, and FTDI drivers are included in the Libero SoC installation package.
2. For IGLOO2 design files:
   http://soc.microsemi.com/download/rsc/?f=m2gl_dg0534_df
3. For SmartFusion2 design files:
   http://soc.microsemi.com/download/rsc/?f=m2s_dg0534_df

3.4 Demo Design

Design files include:

- Demo_Utility includes:
- Libero_project
- Programming_Job
- Source_files
- readme.txt

The top-level structure of the design files are shown in Figure 1 for IGLOO2 and Figure 2 for SmartFusion2. For more information, refer to the `readme.txt` file.

Figure 1 • **IGLOO2 Demo Design Files - Top-Level Structure**

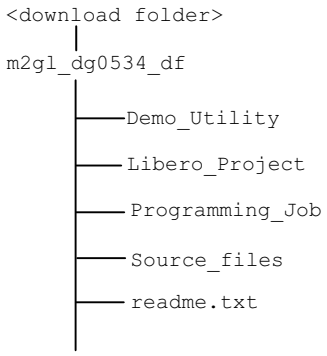
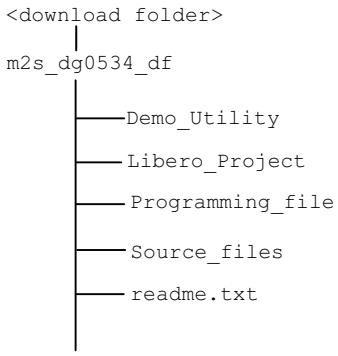


Figure 2 • **SmartFusion2 Demo Design Files - Top-Level Structure**

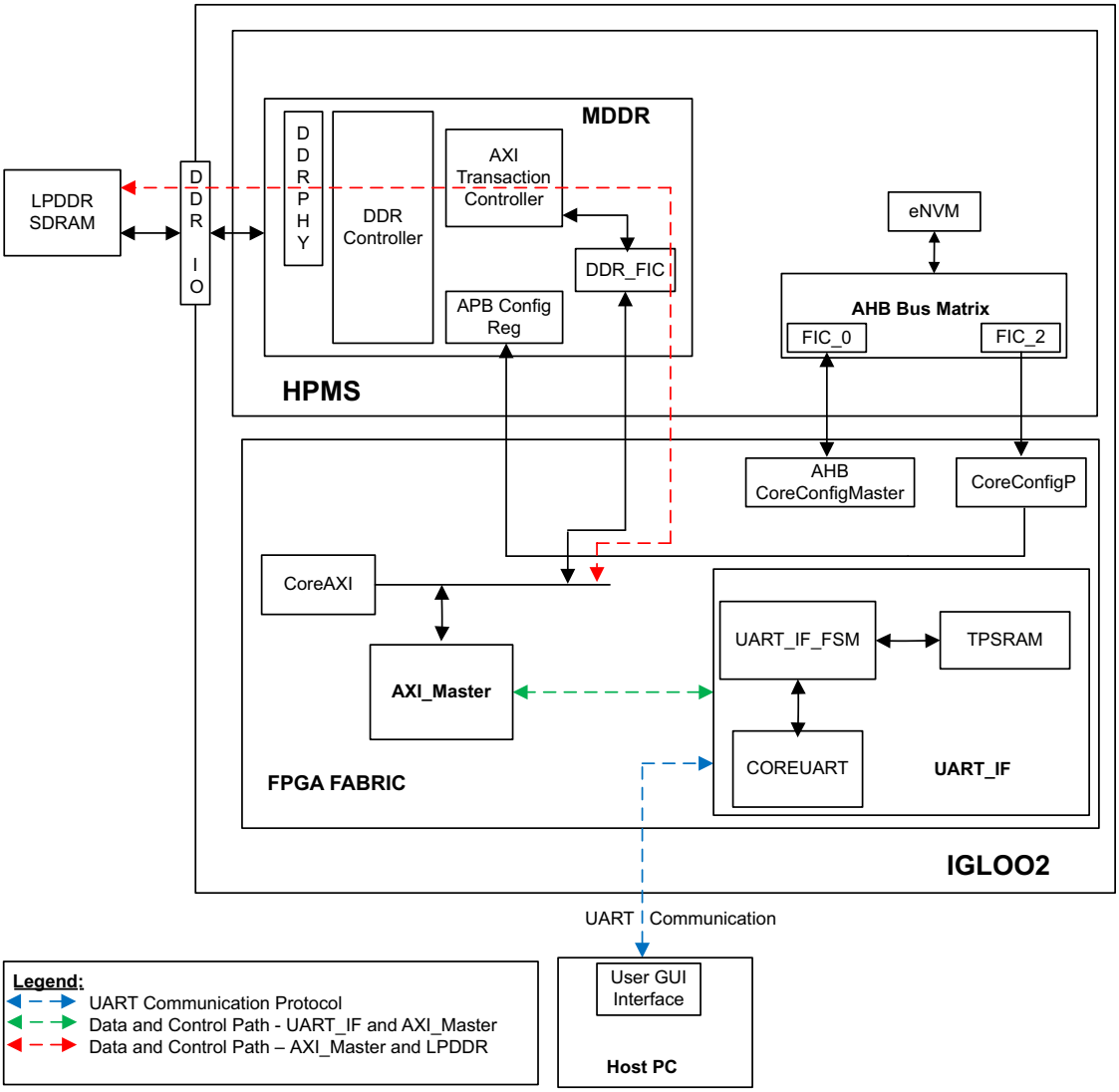


In the demo design, the AXI master implemented in the FPGA fabric accesses the LPDDR memory present in the SmartFusion2 and IGLOO2 Security Evaluation Kit board using the MDDR controller. The AXI master logic communicates to the MDDR controller via the CoreAXI interface and the DDR_FIC interface. The read/write operations initiated by the IGL2_MDDR_Demo and SF2_MDDR_Demo utility are sent to the UART_IF block using the UART protocol. AXI master receives the address and data from the UART_IF block.

During a write operation, the UART_IF block sends the address and data to the AXI master logic. During a read operation, the UART_IF block sends the address to the AXI master and stores the read data in TPSRAM. When the read operation is complete, the read data is sent to the host PC via UART.

The MDDR demo design block diagram is shown in Figure 3 for IGLOO2 and Figure 4 for SmartFusion2.

Figure 3 • **MDDR Demo Design Block Diagram - IGLOO2**



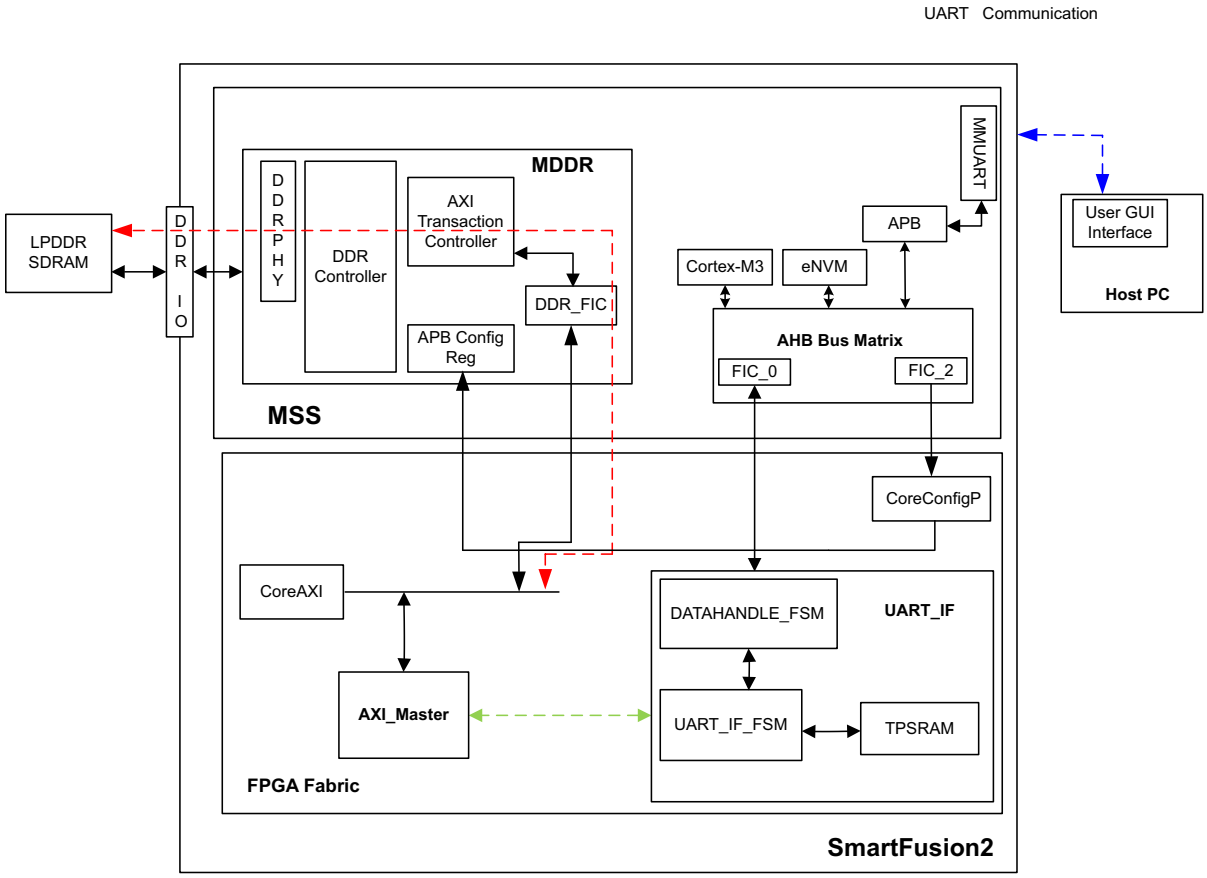
In this demo design, the following blocks are configured:

- MDDR controller is configured for LPDDR memory available in the IGLOO2 Evaluation Kit board. The LPDDR memory is a Micron® DRAM (Part Number: MT46H32M16LF).
- DDR_FIC is configured for the AXI bus interface.

- Both the AXI clock and LPDDR clock are configured for 160 MHz. IGLOO2 CoreUART IP has the following configuration:
  - **Baud Rate**: 115200
  - **Data Bits**: 8
  - **Parity**: None
- TPSRAM IP has the following configuration:
  - **Write port depth**: 256
  - **Write port width**: 64
  - **Read port depth**: 2048
  - **Read port width**: 8

Note: In the demo design, all configuration blocks are the same, except CoreUART IP configuration for IGLOO2.

Figure 4 • **MDDR Demo Design Block Diagram - SmartFusion2**



In this demo design, different blocks are configured as follows:

- MDDR controller is configured for LPDDR memory available in the SmartFusion2 Security Evaluation Kit board. The LPDDR memory is a Micron DRAM (Part Number: MT46H32M16LF)
- DDR_FIC is configured for the AXI bus interface.
- Both the AXI clock and LPDDR clock are configured for 160 MHz.
- TPSRAM IP has the following configuration:
  - Write port depth: 256
  - Write port width: 64
  - Read port depth: 2048
  - Read port width: 8

For more information on how to configure the DDR controller, refer to Appendix 2: Configuring MDDR Controller, page 28.

3.4.1 Features

The SmartFusion2 and IGLOO2 MDDR demo design have the following features:

- Single AXI read/write transactions
- 16-beat burst AXI read/write transactions
- LPDDR memory model simulation using SmartDesign testbench
- Design validation using the SmartFusion2 Security Evaluation Kit board and IGLOO2 Evaluation Kit board that has the LPDDR memory
- Initiation of the read/write transactions using `IGL2_MDDR_Demo` utility for IGLOO2 and `SF2_MDDR_Demo` utility for SmartFusion2

3.4.2 Description

The demo design consists of the following SmartDesign components:

- **MDDR_Demo_0**: IGLOO2 SmartDesign handles the data transactions between the MDDR controller and LPDDR SDRAM.
- MDDR_Demo_top_0: SmartFusion2 SmartDesign handles the data transactions between the MDDR controller and LPDDR SDRAM.
- **UART_IF_0**: This SmartDesign handles the communication between the host PC and the SmartFusion2 Security and IGLOO2 Evaluation Kit board.

Figure 5 shows the MDDR_Demo_0 and UART_IF_0 connections for IGLOO2 and Figure 6 shows the MDDR_Demo_top_0 and UART_IF_0 connections for SmartFusion2.

Figure 5 • **IGL2_MDDR_Demo SmartDesign**

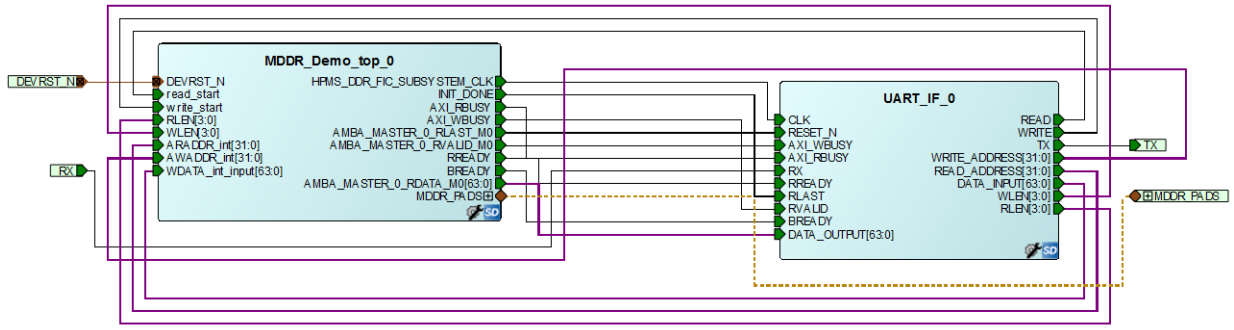
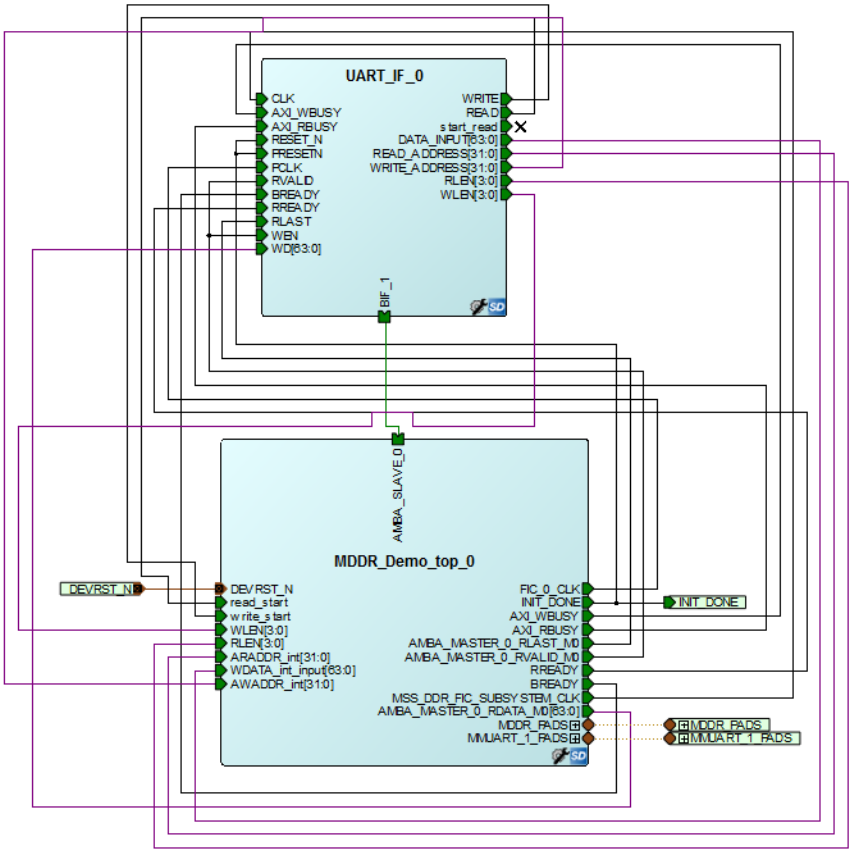


Figure 6 • SF2_MDDR_Demo SmartDesign

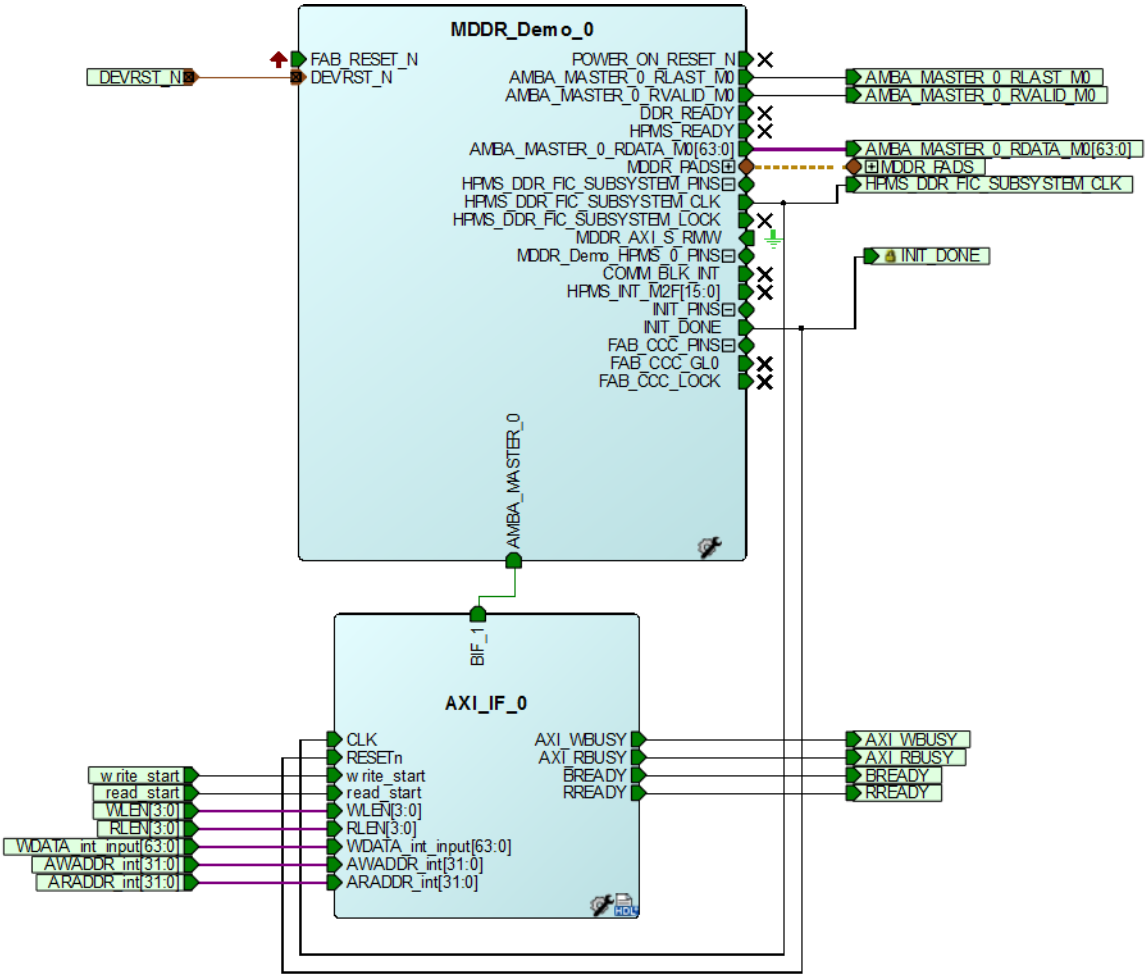


3.4.2.1 MDDR_Demo_0 for IGLOO2

MDDR_Demo_0 consists of the MDDR_Demo_sb_0 subsystem generated using the system builder and the AXI_IF_0 master logic. The AXI_IF_0 master logic is an RTL code that implements the AXI read and write transactions. It receives the read/write operations, burst length (RLEN and WLEN), address, and data as inputs. Based on inputs received, it communicates with the LPDDR memory through the MDDR controller.

Figure 7 shows the MDDR_Demo_0 SmartDesign component for IGLOO2.

Figure 7 • **MDDR_Demo_0 SmartDesign Component - IGLOO2**

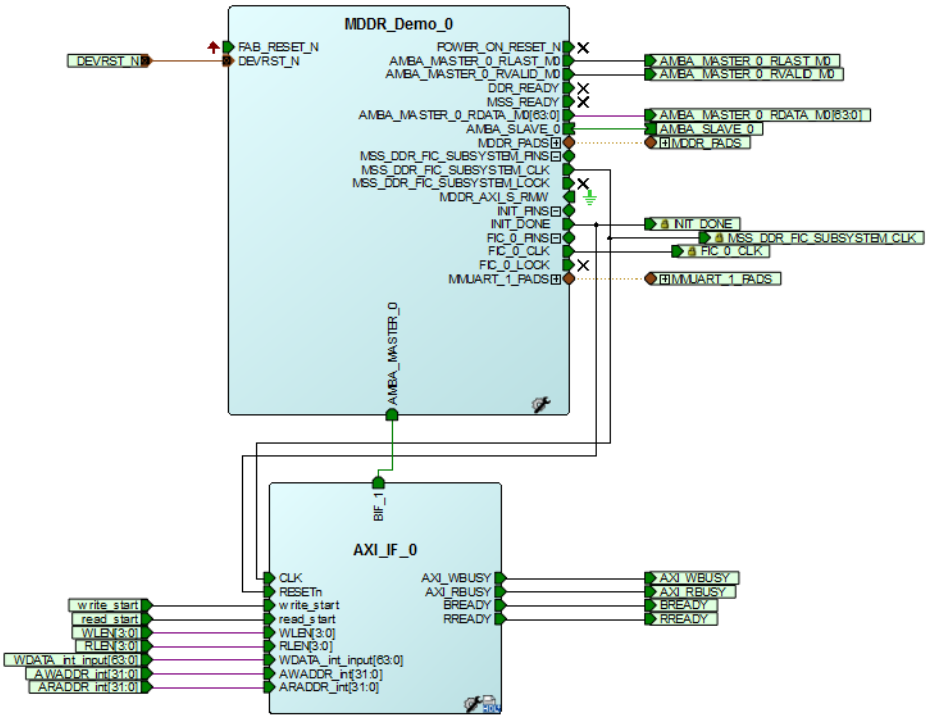


3.4.2.2 MDDR_Demo_top_0 for SmartFusion2

MDDR_Demo_top_0 consists of the MDDR_Demo_0 subsystem generated using the system builder and the AXI_IF_0 master logic. The AXI_IF_0 master logic is an RTL code that implements the AXI read and write transactions. It receives the read or write operations, burst length (RLEN and WLEN), address, and data as inputs. Based on inputs received, it communicates with the LPDDR memory through the MDDR controller.

Figure 8 shows the MDDR_Demo_top_0 SmartDesign component for SmartFusion2.

Figure 8 • **MDDR_Demo_top_0 SmartDesign Component - SmartFusion2**



3.4.2.3 UART_IF_0

For IGLOO2, the UART_IF_0 SmartDesign component handles the UART communication between the host PC demo utility and the AXI master logic. The COREUART_0 IP receives the UART signals from the host PC user interface. The UART_IF_FSM_0 is a wrapper for COREUART_0, collects the data from COREUART_0 IP, and converts data to the relevant AXI_IF_0 master signals.

For a single write operation, the UART_IF_FSM_0 wrapper receives the address and data from the demo utility. For a burst write operation, the address and data are received from the demo utility, and the subsequent incremental data are provided by the UART_IF_FSM_0 wrapper.

For a burst read operation, UART_IF_FSM_0 collects the address from the demo utility and sends that to the AXI_IF_0 master logic. It then receives the read data from the AXI_IF_0 master logic and stores it in the TPSRAM_0. After completion of the read burst transactions, the UART_IF_FSM_0 wrapper fetches the stored data from the TPSRAM_0 and sends it to the COREUART IP.

For SmartFusion2, the UART_IF_0 SmartDesign component handles the UART communication between the host PC demo utility and the AXI Master logic. The MMUART_1 block present in the MSS receives the UART signals from the host PC user interface, the ARM Cortex-M3 processor sends this user data to the DATAHANDLE_FSM block present in the FPGA fabric using the FIC_0 advanced peripheral bus (APB) slave interface. DATAHANDLE_FSM is an APB slave wrapper, which sends the received data to the UART_IF_FSM_0 block.

For a single write operation, the UART_IF_FSM_0 wrapper receives the address and data from the demo utility. For a burst write operation, the address and data are received from the demo utility, and the subsequent incremental data are provided by the UART_IF_FSM_0 wrapper.

For a burst read operation, UART_IF_FSM_0 collects the address from the demo utility and sends that to the AXI_IF_0 master logic. It then receives the read data from the AXI_IF_0 master logic and stores it in the TPSRAM_0. After completion of the read burst transactions, the Cortex-M3 processor reads the TPSRAM_0 buffer through DATAHANDLE_FSM (APB wrapper) block. The received data is sent to the host PC using the MMUART_1 block.

The UART_IF_0 SmartDesign component is shown in Figure 9 for IGLOO2 and Figure 10 for SmartFusion2.

Figure 9 • **UART_IF_0 SmartDesign Component - IGLOO2**

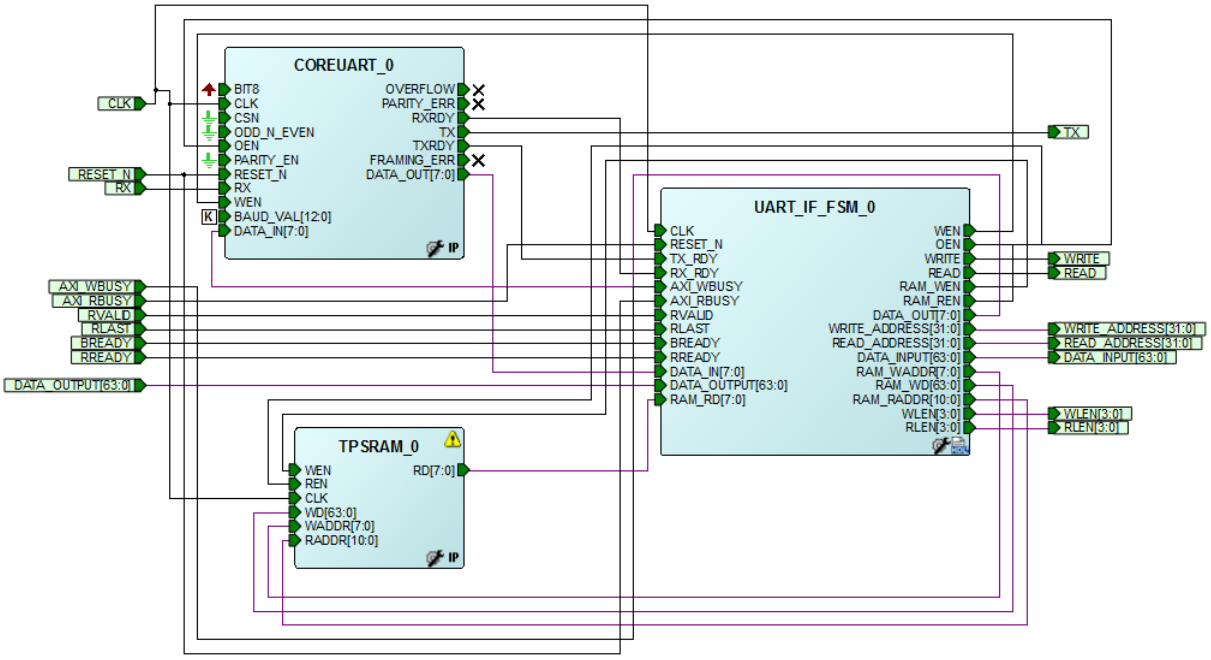
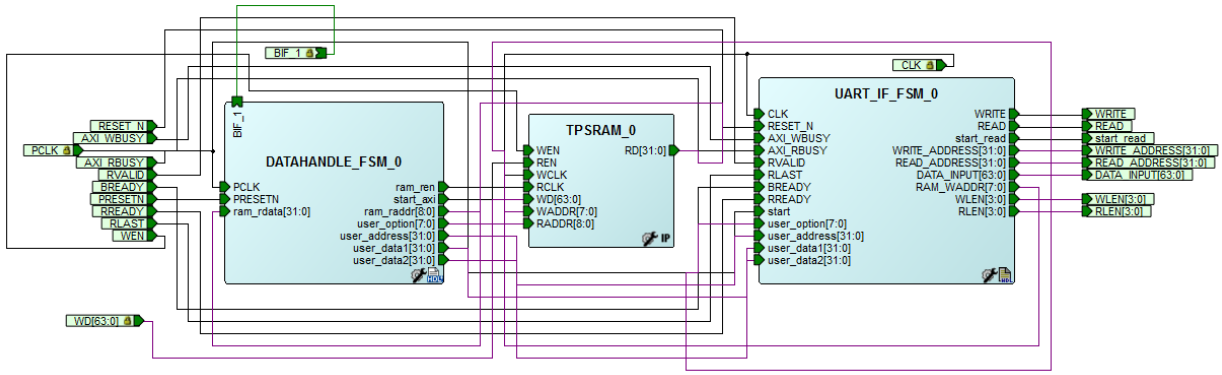


Figure 10 • **UART_IF_0 SmartDesign Component - SmartFusion2**



3.5 Running Simulation

The demo design can be simulated using SmartDesign testbench and the LPDDR memory model (MT46H32M16LF with 512 Mb density).

The simulation run the following operations:

- Single AXI write and read operation
- 16-beat AXI burst write and read operation

The AXI_LPDDR_Simulation SmartDesign testbench is shown in Figure 11 for IGLOO2 and Figure 12 for SmartFusion2. The AXI_testbench provides the read/write operations, burst length, address, and data to the MDDR_Demo_0 SmartDesign component for IGLOO2 and the MDDR_Demo_top_0 SmartDesign component for SmartFusion2.

Figure 11 • **AXI_LPDDR_Simulation SmartDesign Testbench - IGLOO2**

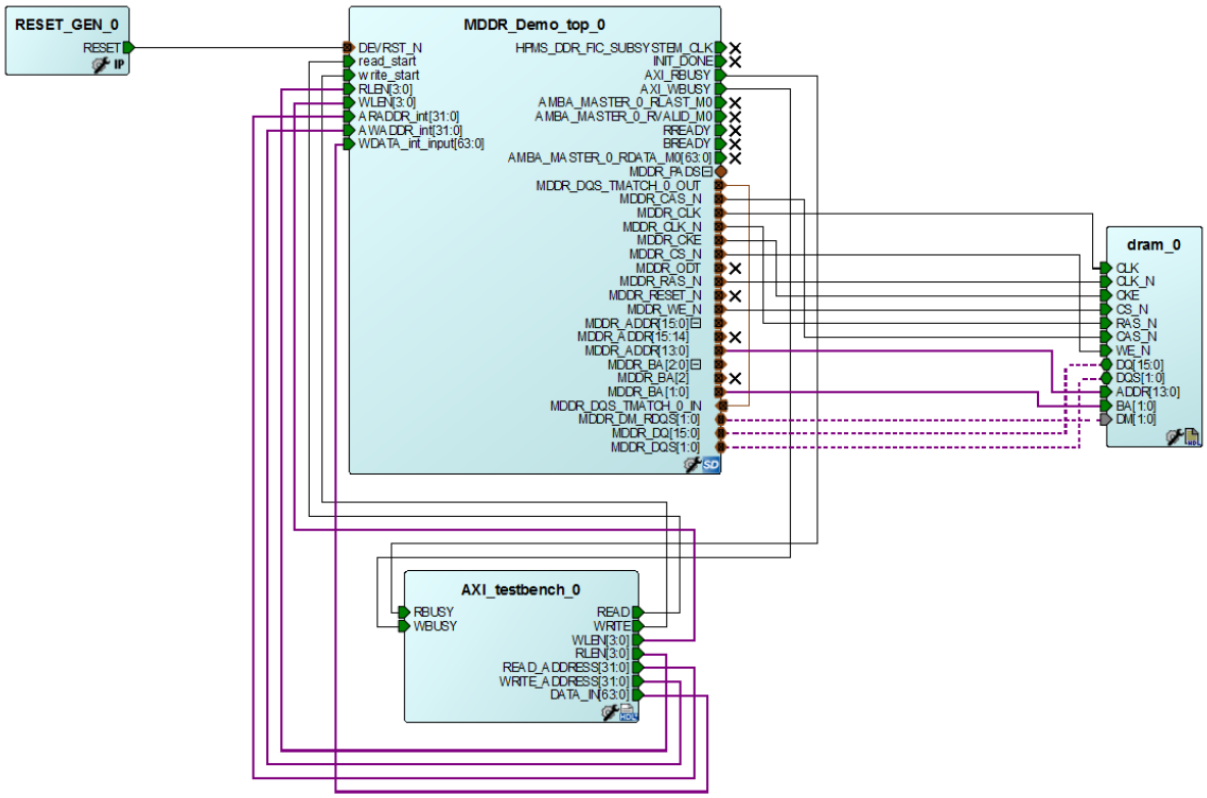
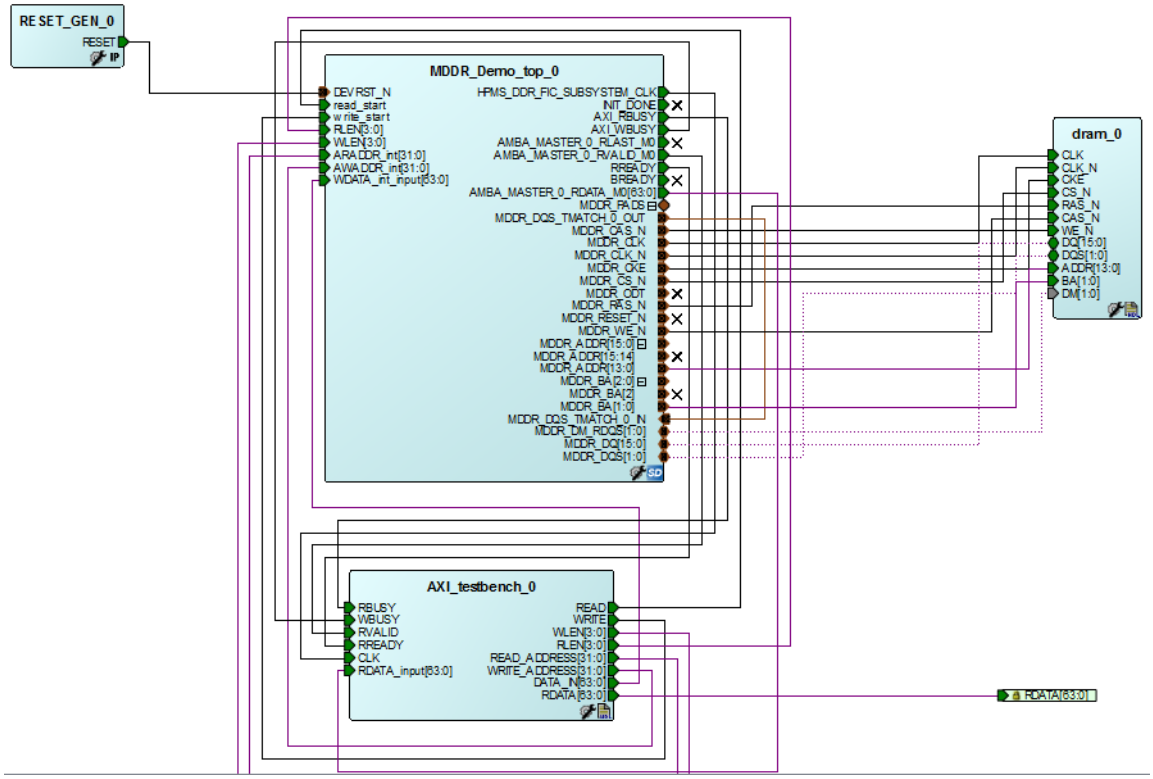


Figure 12 • **AXI_LPDDR_Simulation SmartDesign Testbench - SmartFusion2**



To run the simulation, ensure that the following files are present in the Libero SoC project:

- `dram.v`
- `dram_parameters.vh`
- `AXI_testbench.v`

The default location of the files are:

- IGLOO2:

  <Download folder>\m2gl_dg0534_df\Libero_Project\stimulus

- SmartFusion2:

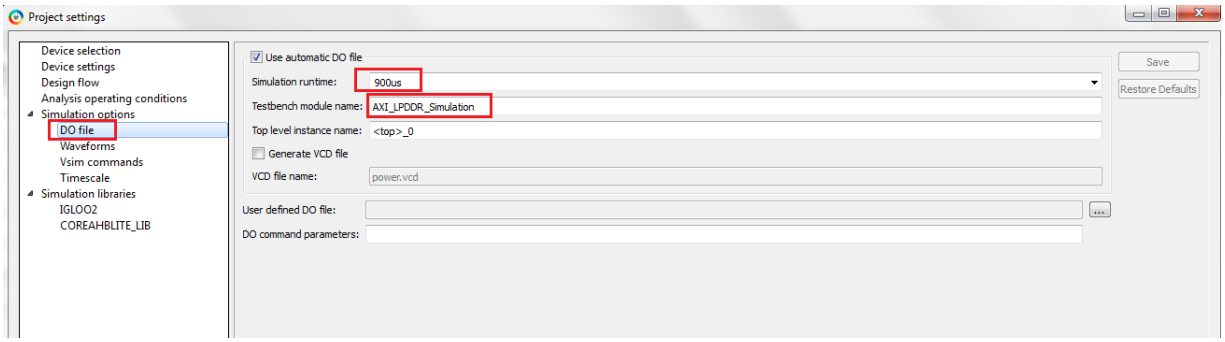
  <Download folder>\m2s_dg0534_df\Libero_Project\stimulus

3.5.1 Simulation Setup

The following are the steps to set up the Simulation set up configuration:

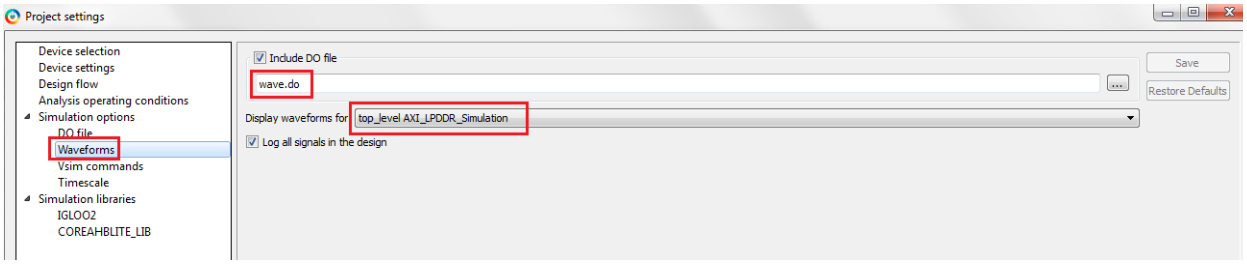
1. Launch the Libero SoC software.
2. Browse the `m2gl_dg0534_df\Libero_Project` for IGLOO or `m2s_dg0534_df\Libero_Project` for the SmartFusion2 project provided in the design file.
3. Go to **Project** > **Project Settings** > **Simulation options**.
4. Ensure that the **DO file** tab has the configuration, as shown in Figure 13.

Figure 13 • **DO File Settings**



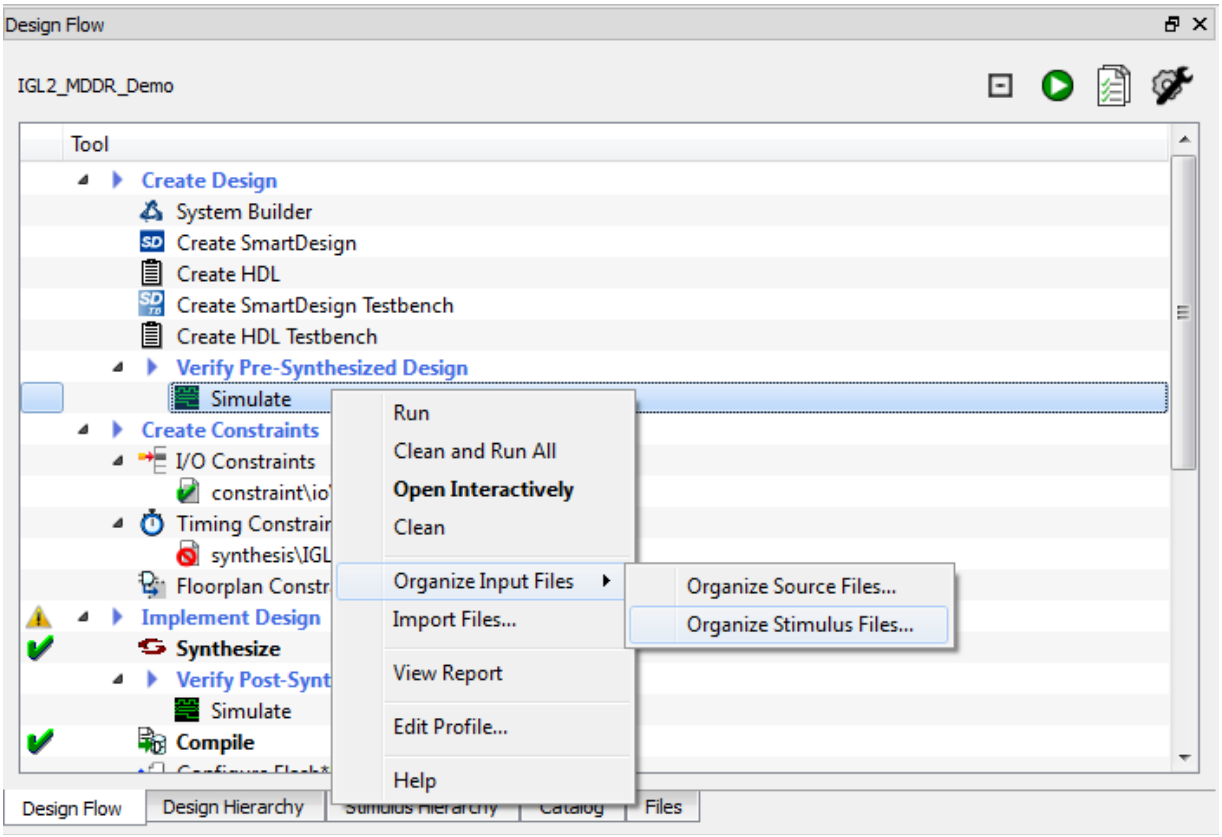
5. Ensure that the **Waveforms** tab has the configuration, as shown in Figure 14.

Figure 14 • **Waveforms Settings**



6. Go to the **Design Flow** tab.
7. Right-click **Simulate** under **Verify Pre-Synthesized Design** and select **Organize Input Files** > **Organize Stimulus Files...**, as shown in Figure 15.

Figure 15 • **Invoking Organize Stimulus Files Window**



8. Ensure that the **Organize Stimulus Files** window has the configuration, as shown in Figure 16 for IGLOO2 and Figure 17 for SmartFusion2.

Figure 16 • **Organize Stimulus Files Window - IGLOO2**

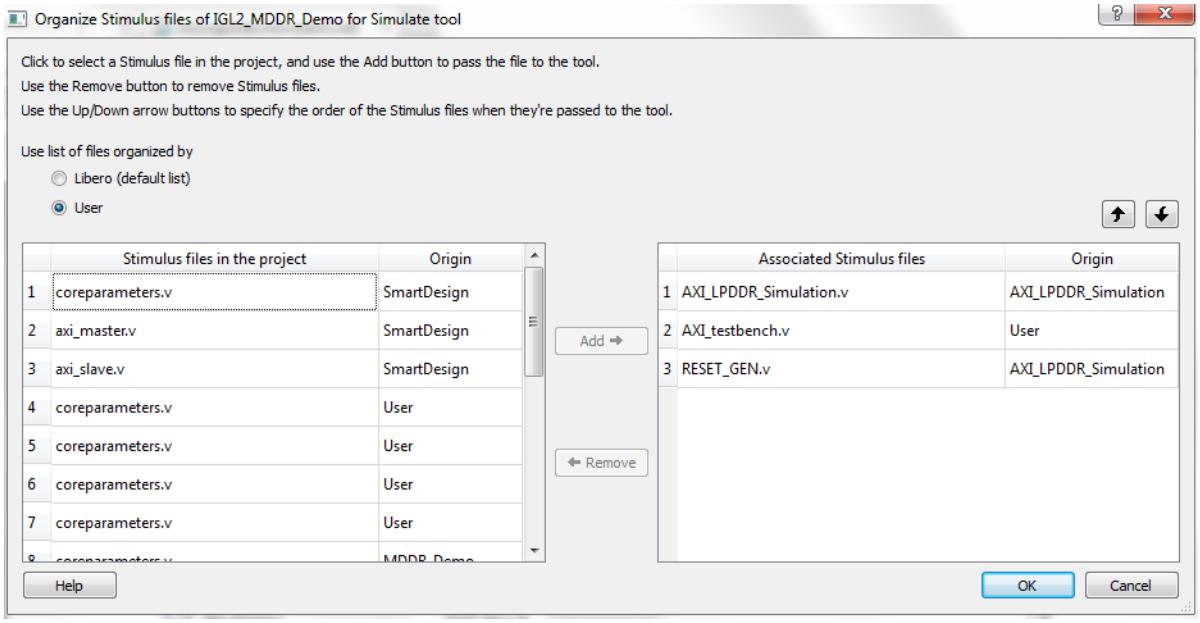
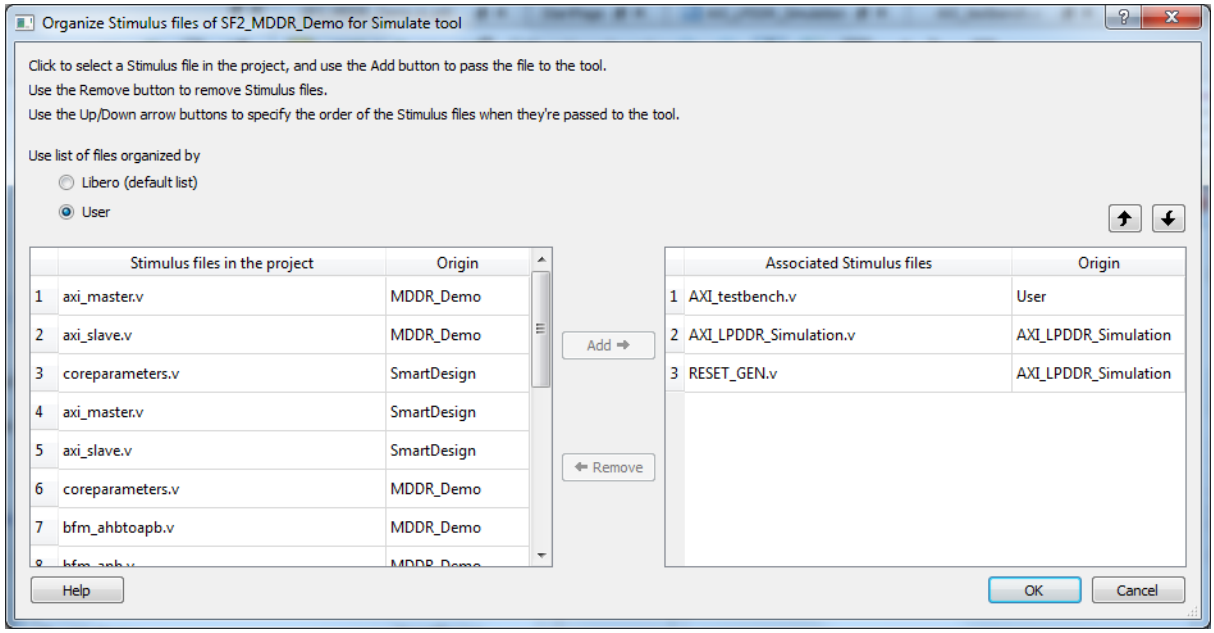


Figure 17 • **Organize Stimulus Files Window SmartFusion2**



3.5.2 Running the Simulation

The following steps describe how to run the simulation:

1. Right-click **Simulate** under **Verify Pre-Synthesized Design**.
2. Click **Open Interactively**.
3. The simulation run time is 900 µs, as shown in Figure 13.

Figure 18 shows the transcript window of the simulation.

Figure 18 • **Transcript Window**

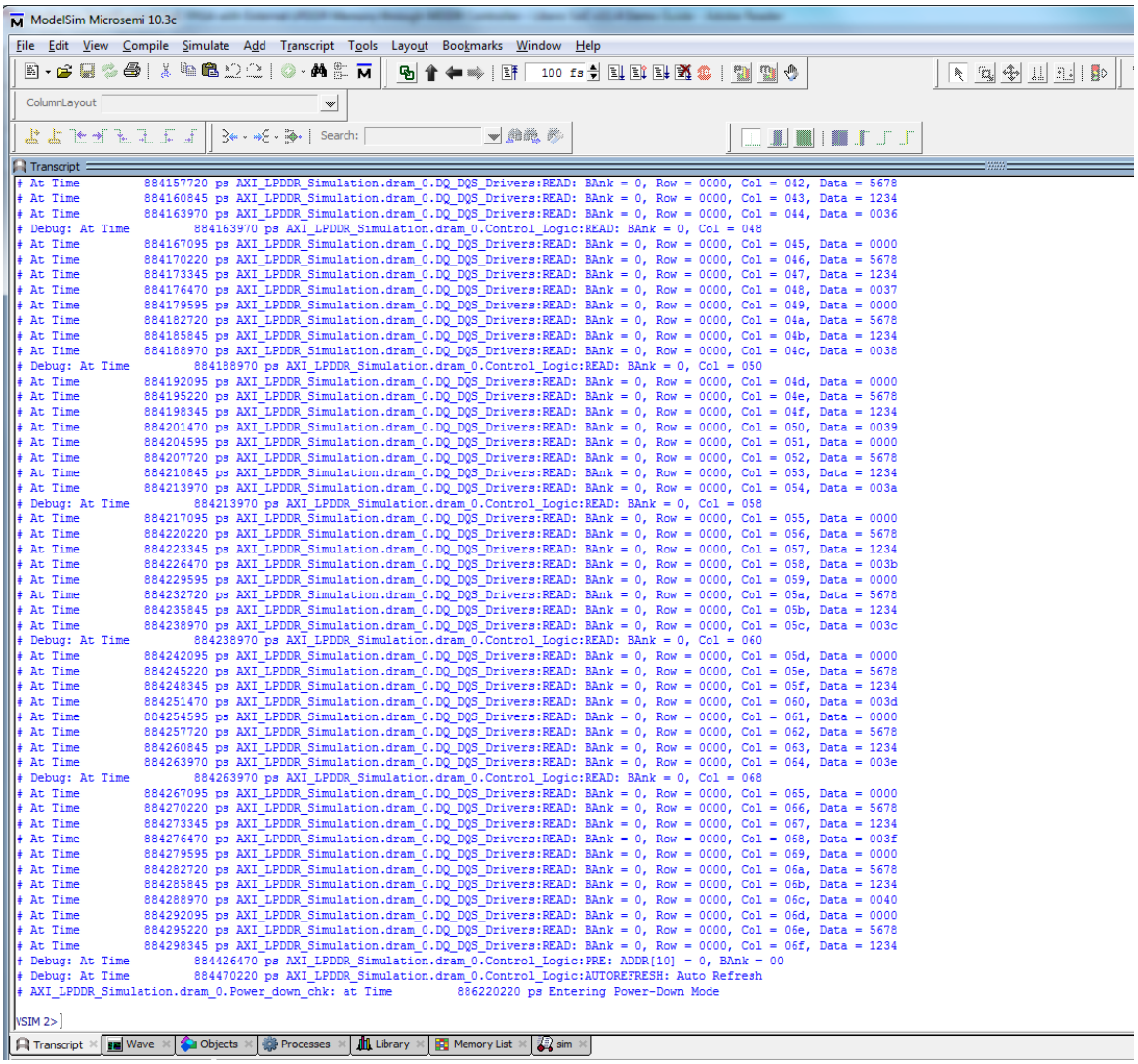


Figure 19 shows the single AXI write and AXI read operation.

Figure 19 • **Single Write and Read Operation**

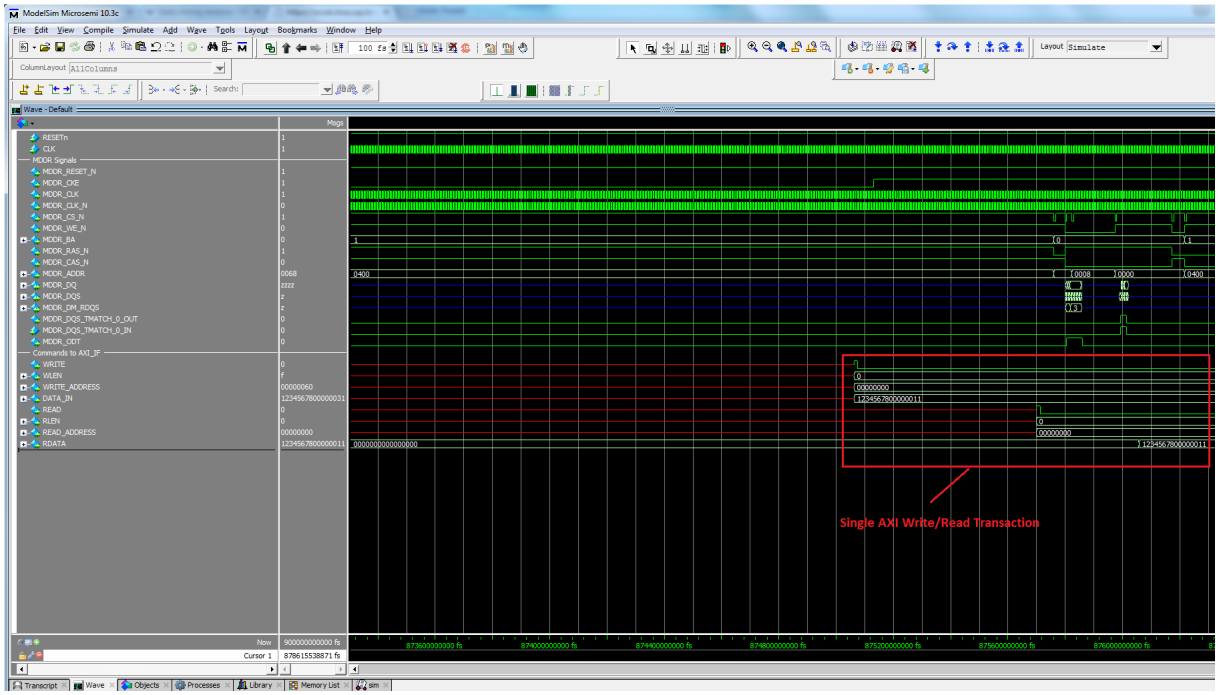
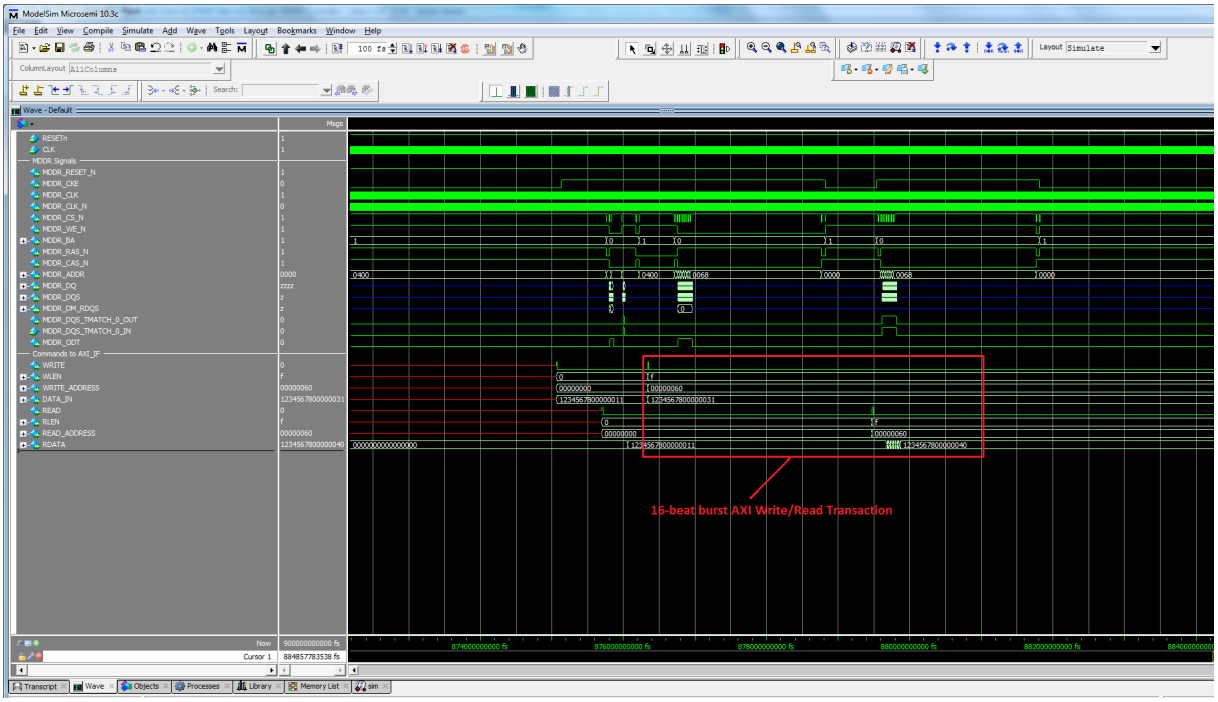


Figure 20 shows the 16-beat AXI burst write and read operation.

Figure 20 • **16-Beat AXI Burst Write and Read**



3.6 Setting Up the Hardware Demo

The following steps describe how to set up the hardware demo:

1. Connect the jumpers on the SmartFusion Security and IGLOO2 Evaluation Kit, as shown in Table 2, page 18.

Table 2 • **SmartFusion2 Security and IGLOO2 FPGA Evaluation Kit Jumper Settings**

Jumper	Pin (From)	Pin (To)	Comments
J22	1	2	Default
J23	1	2	Default
J24	1	2	Default
J8	1	2	Default
J3	1	2	Default

CAUTION: Ensure that the power supply switch **SW7** is switched OFF while connecting the jumpers.

2. Connect the Power supply to the J6 connector, switch ON the power supply switch, **SW7**.
3. Connect the FlashPro4 programmer to the **J5** connector of the SmartFusion2 Security and IGLOO2 Evaluation Kit board.
4. Connect the Host PC USB port to the SmartFusion2 Security and IGLOO2 Evaluation Kit board J18 USB connector using the USB mini-B cable.

Figure 21 shows the board setup for running the SmartFusion2 and IGLOO2 MDDR demo on the SmartFusion2 Security and IGLOO2 Evaluation Kit.

Figure 21 • **IGLOO2 Evaluation Kit Board**

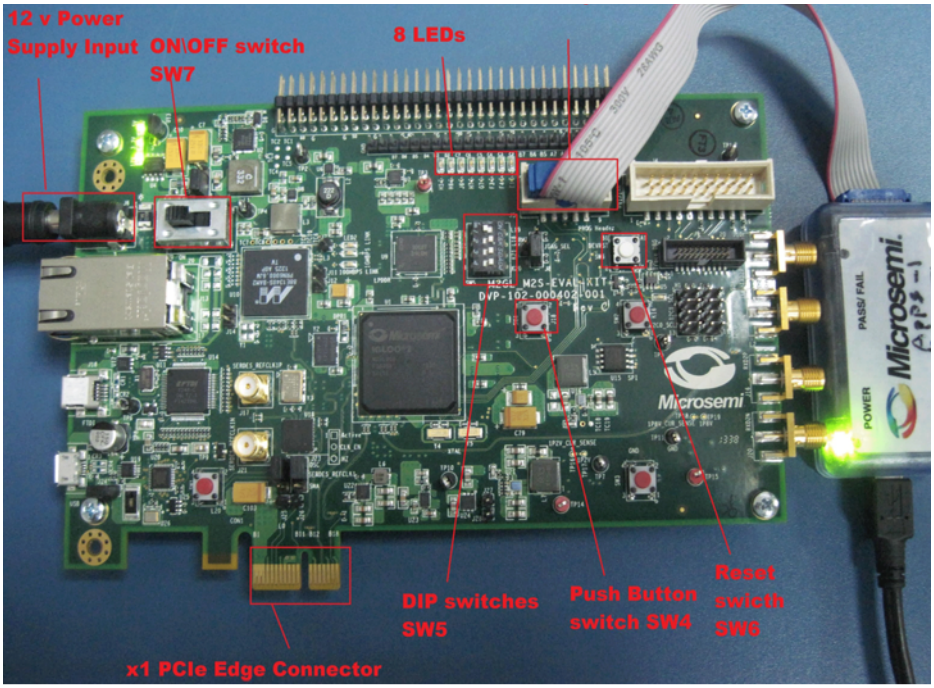
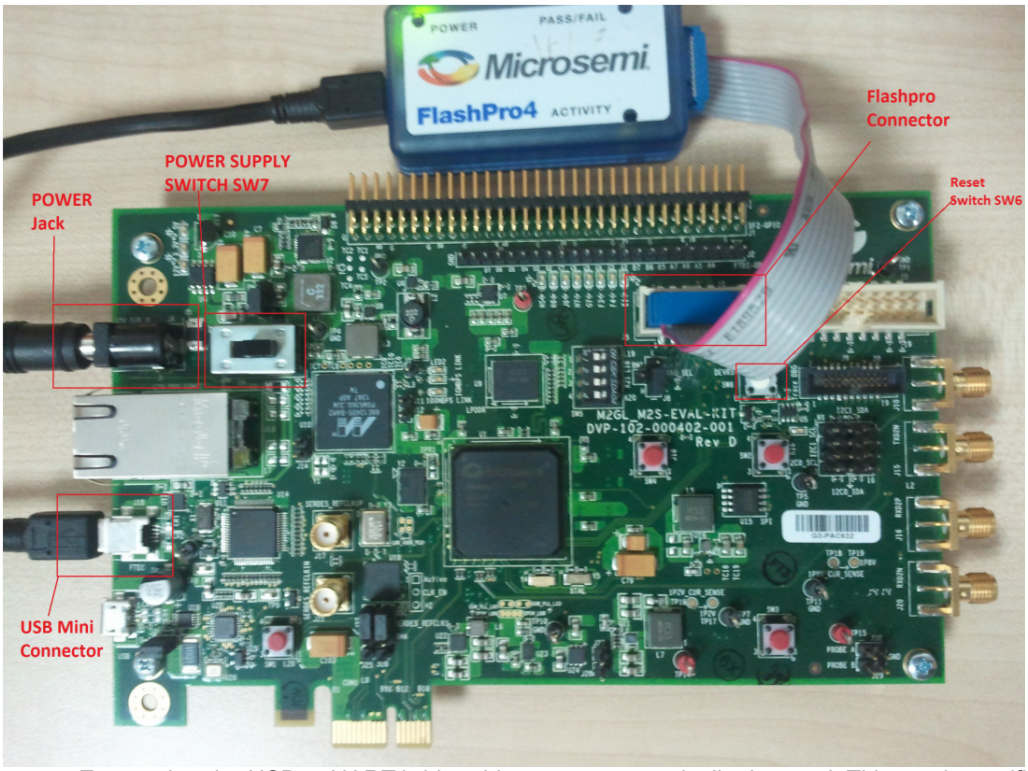


Figure 22 • **SmartFusion2 Security Evaluation Kit**



5. Ensure that the USB to UART bridge drivers are automatically detected. This can be verified in the **Device Manager** of the host PC. The FTDI USB to UART converter enumerates four COM ports. For USB 2.0, note down the USB Serial Converter D COM port number to use it in the GUI. The following figures show the USB 2.0 Serial port properties. For IGLOO2, COM10 is connected to **USB Serial Converter D,** as shown in Figure 23. For SmartFusion2, COM7 is connected to **USB Serial Converter D,** as shown in Figure 24. For more information about how to find the correct COM Port number in USB 3.0, refer to Appendix 3: Finding Correct COM Port Number when Using USB 3.0, page 36.

Figure 23 • **USB Serial 2.0 Port Properties - IGLOO2**

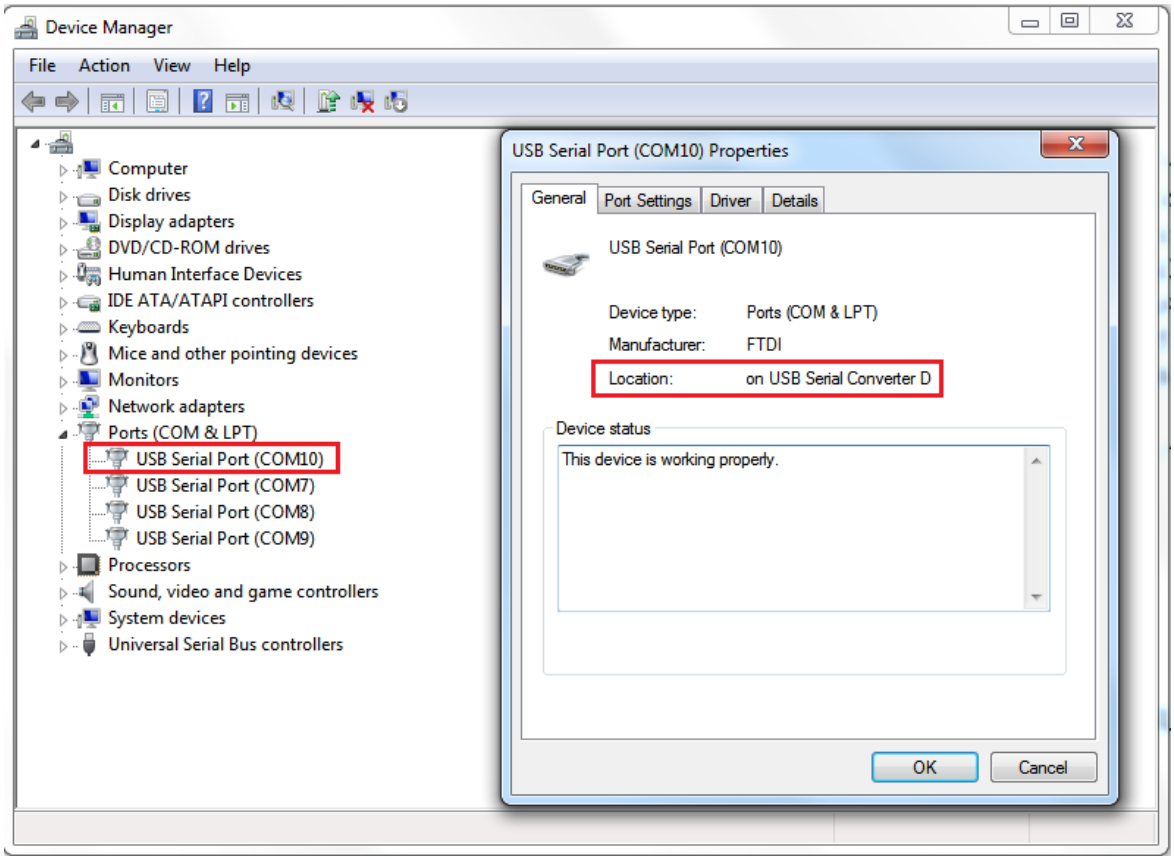
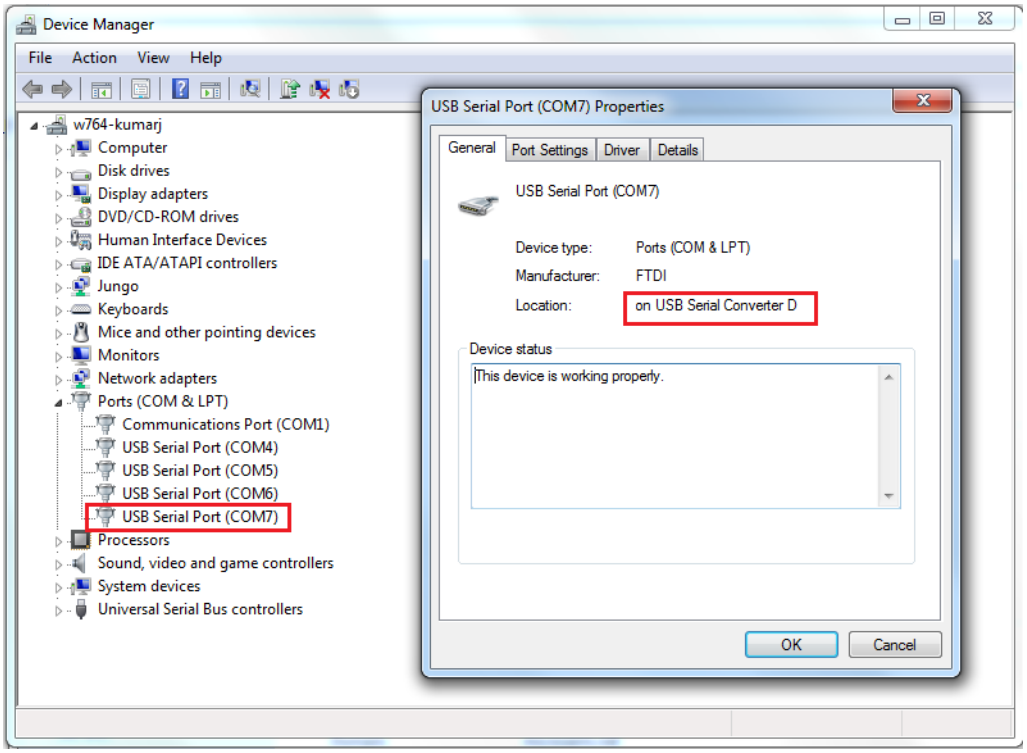


Figure 24 • **USB Serial 2.0 Port Properties - SmartFusion2**



As shown in Figure 23 and Figure 24 the port properties of COM10 and COM7 show that it is connected to USB Serial Converter D. Hence, COM10 and COM7 is selected in this example. The COM port number is system specific.

6. If the USB to UART bridge drivers are not installed, download and install the drivers from *www.microsemi.com/soc/documents/CDM_2.08.24_WHQL_Certified.zip.*

3.7 Setting Up the Device

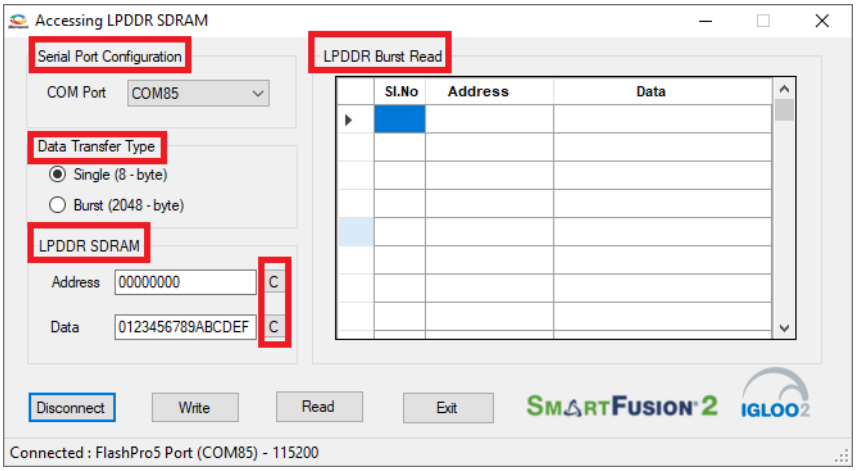
Program the SmartFusion2 Security and IGLOO2 Evaluation Kit board with the job file provided as part of the design files using FlashPro Express software, refer to Appendix 1: Programming the Device Using FlashPro Express, page 25.

3.8 Running the Hardware Demo

The SmartFusion2 and IGLOO2 MDDR demo come with a utility, `IGL2_MDDR_Demo` for IGLOO2 and `SF2_MDDR_Demo` for SmartFusion2, that runs on the host PC to communicate with the SmartFusion2 Security and IGLOO2 Evaluation Kit. The UART protocol is used as the underlying communication protocol between the host PC and the SmartFusion2 Security and IGLOO2 Evaluation Kit.

The initial screen of the `IGL2_MDDR_Demo` utility is shown in Figure 25.

Figure 25 • **MDDR_Demo Utility SmartFusion2 and IGLOO2**



The SF2_MDDR_Demo utility and IGL2_MDDR_Demo utility consists of the following sections:

- **Serial Port Configuration**: Displays the serial port. Baud rate is fixed at 115200
- **Data Transfer Type**: Single or Burst
- **LPDDR SDRAM**: Provides Address and Data
- **LPDDR Burst Read**: Displays the Burst Read Values for the corresponding address
- **C**: Clears the existing data

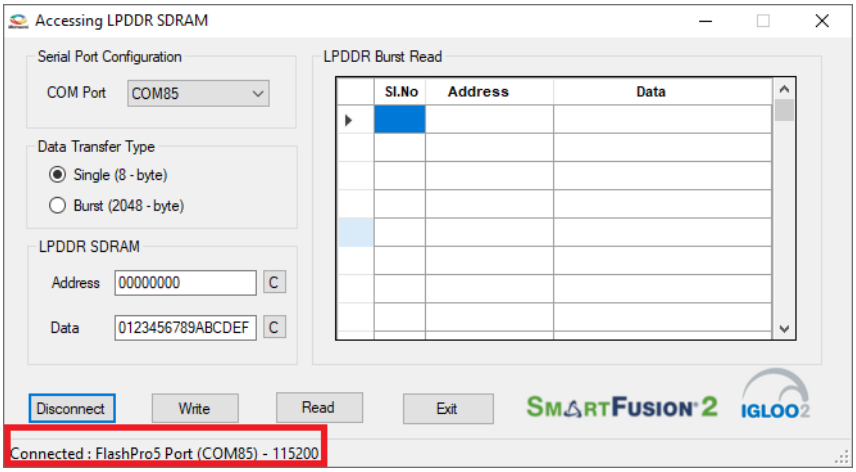
3.9 Steps to Run GUI

The following steps describe how to run the IGLOO2 GUI:

1. Launch the utility. The default location is:
   - IGLOO2:
     <download_folder>\\m2gl_dg0534_df\Demo_Utility\IGL2_MDDR_Demo.exe
   - SmartFusion2:
     <download_folder>\\m2s_dg0534_df\Demo_Utility\SF2_LPDDR.exe.
2. Select the appropriate COM port from the drop down menu. It is COM 10 for IGLOO2 and COM 7 for SmartFusion2.
3. Click **Connect**. The connection status along with the COM Port and Baud rate is shown in the left bottom corner of the screen.

The connection status of the utility is shown in Figure 26.

Figure 26 • **MDDR_Demo- Connection Status SmartFusion2 and IGLOO2**



3.10 Performing a Single Data Transfer

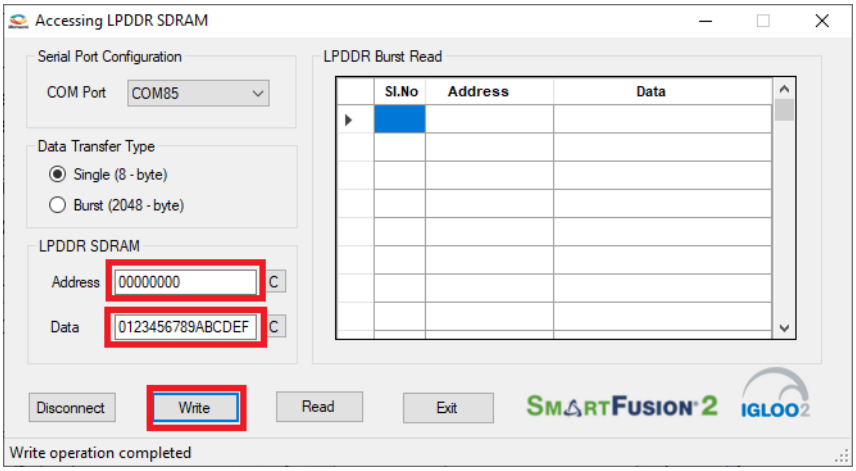
For a single write or read operation, the AXI master logic is configured to transfer a burst length of 1 (that is, 8 bytes). For a write operation, the utility sends a 32-bit address and 64-bit (8 bytes) data. The data is then written to the LPDDR SDRAM. For a read operation, the utility sends a 32-bit address and receives 64-bit data from LPDDR and is displayed in the utility.

The following steps describe how to perform a single data transfer:

1. Select **Single (8-bytes)** as **Data Transfer Type**.
2. A 64-bit aligned address is required in the address field. Enter a 32-bit HEX Address in the range 0×00000000 - 0×03FFFFF8. When a non 64-bit aligned address is provided, the GUI converts it to a 64-bit aligned address and performs the write/read. Refer to Appendix 4: Performing Write/Read Operation when Non 64-bit Aligned Address is Provided, page 38 to perform write/read when non 64-bit aligned address is provided.
3. In the **Data** field, enter 64-bit data in HEX format.
4. Click **Write**. The entered data is written to the LPDDR memory.

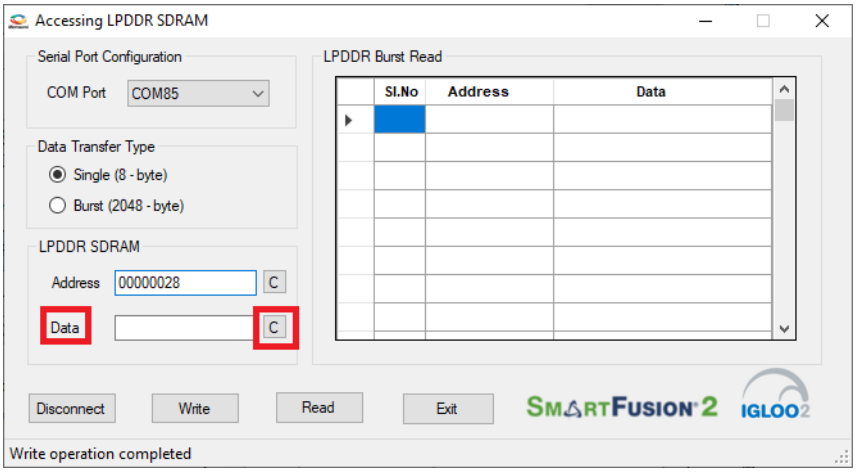
The **Address** and **Data** values entered for a Single Write operation is shown in Figure 27.

Figure 27 • **Single Write Operation - SmartFusion2 and IGLOO2**



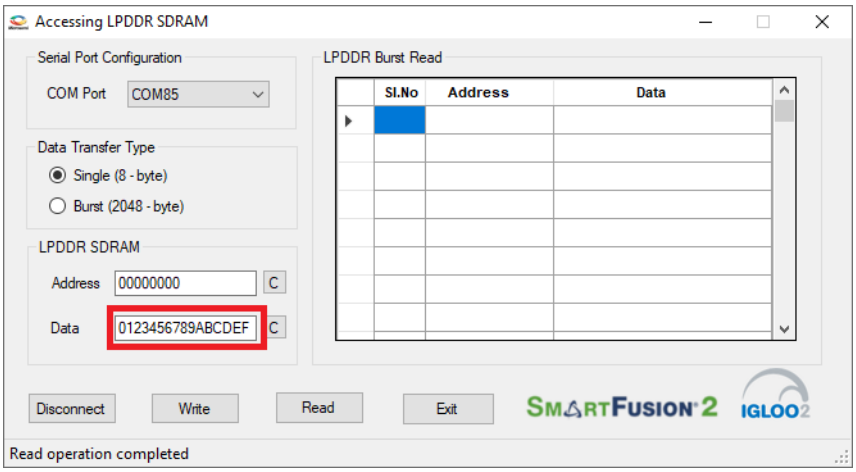
5. To verify the write operation, perform a read operation to the same address where the data is written.
6. Press **C** to clear the data present in the **Data** field, as shown in Figure 28.

Figure 28 • **Clear Data Field - SmartFusion2 and IGLOO2**



7. Click **Read** to read the data from the LPDDR SDRAM, as shown in Figure 29.

Figure 29 • **Single Read Operation - SmartFusion2 and IGLOO2**



8. Compare the read and write data. The write and read data being the same establishes that the write and read operations to the LPDDR SDRAM were successful.

3.11 Performing Burst Data Transfer

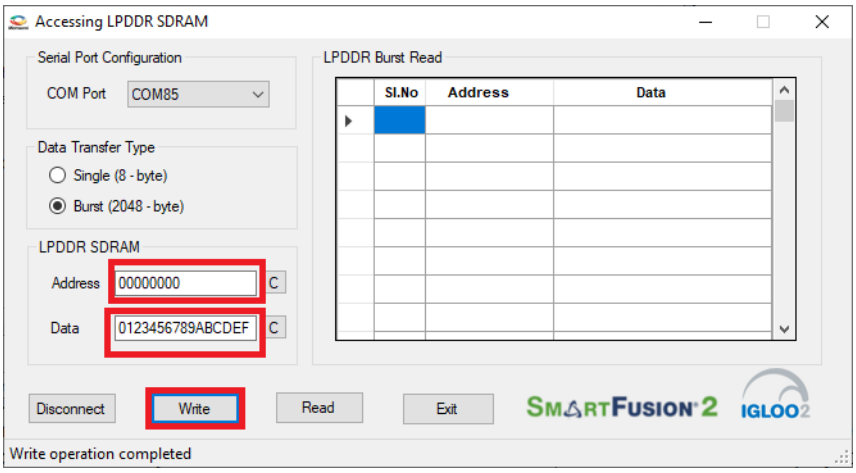
For a burst write or read operation, the AXI master logic is configured to transfer a burst length of 16 (that is, 128 bytes). In this demo, 16 transfers of 16-beat burst operations are implemented (16 transfers × 16-beat burst data = 2048 bytes data). For a write operation, the utility sends a 32-bit initial address and 64-bit (8 bytes) initial data. After the initial write operation, incremental data is written. For a read operation, the utility sends a 32-bit address and receives 2048 bytes of data from the LPDDR SDRAM, and the data is displayed in the utility.

The following steps describe how to perform a burst data transfer:

1. Select **Burst (2048-bytes)** as **Data Transfer Type**.
2. A 64-bit aligned address is required in the address field. Enter a 32-bit HEX Address in the range 0×00000000 - 0×03FFF7F8. When a non 64-bit aligned address is provided, the GUI converts it into a 64-bit aligned address and performs the write/read operation. Refer to Appendix 4: Performing Write/Read Operation when Non 64-bit Aligned Address is Provided, page 38 to perform write/read when non 64-bit aligned address is provided.
3. In the **Data** field, enter 64-bit data in HEX format.
4. Click **Write**. The entered data is written to the Address location specified in the Address field and then the data is incremented by 1 and written to the next address location. This is repeated 256 times to write all the 2048 bytes of data.

The **Address** and **Data** values entered for a Burst Write operation is shown in Figure 30.

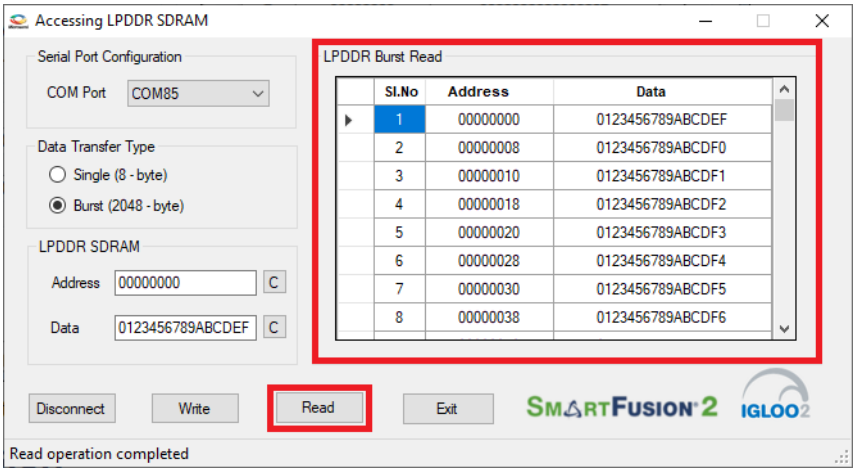
Figure 30 • **Burst Write Operation - SmartFusion2 and IGLOO2**



5. To verify the write operation, perform a read operation to the same address where the data is written.
6. Click **Read**. All the 2048 bytes of data written to the LPDDR is read, and the read data is displayed on the **LPDDR Burst Read** panel.

The burst read data is shown in Figure 31.

Figure 31 • **Burst Read Operation - SmartFusion2 and IGLOO2**



7. Click **Exit** to exit the utility.

3.12 Conclusion

This demo shows how to perform read/write operations to LPDDR SDRAM using the SmartFusion2 and IGLOO2 MDDR controller. Options are provided to simulate the design using a SmartDesign testbench and validate the design on the SmartFusion2 Security and IGLOO2 Evaluation Kit using a GUI interface.

4 Appendix 1: Programming the Device Using FlashPro Express

This section describes how to program the SmartFusion2 and IGLOO2 devices with the programming job file using FlashPro Express.

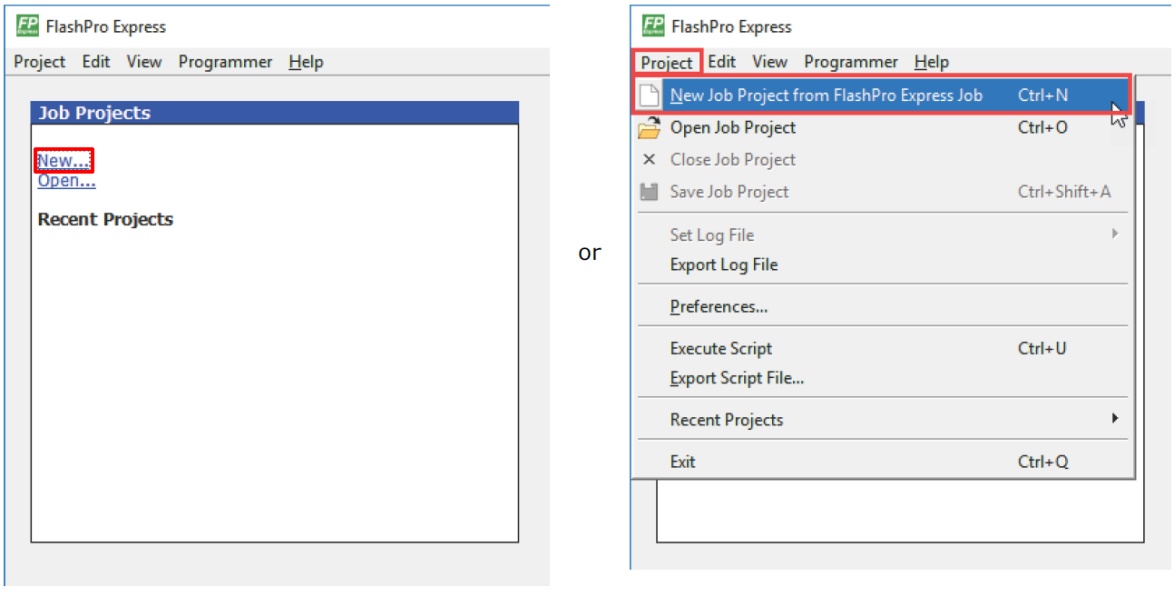
To program the device, perform the following steps:

1. Ensure that the jumper settings on the board are the same as those listed in Table 2, page 18.

Note: The power supply switch must be switched off while making the jumper connections.

2. Connect the power supply cable to the **J6** connector on the board.
3. Power **ON** the power supply switch **SW7**.
4. On the host PC, launch the **FlashPro Express** software.
5. Click **New** or select **New Job Project from FlashPro Express Job** from **Project** menu to create a new job project, as shown in the following figure.

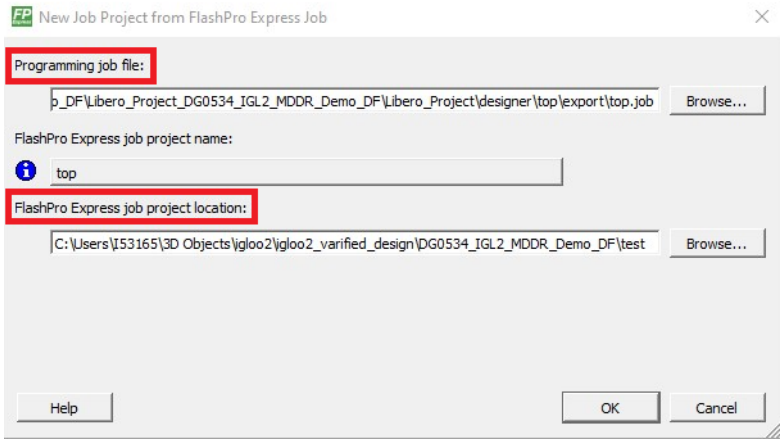
Figure 32 • **FlashPro Express Job Project**



6. Enter the following in the **New Job Project from FlashPro Express Job** dialog box:

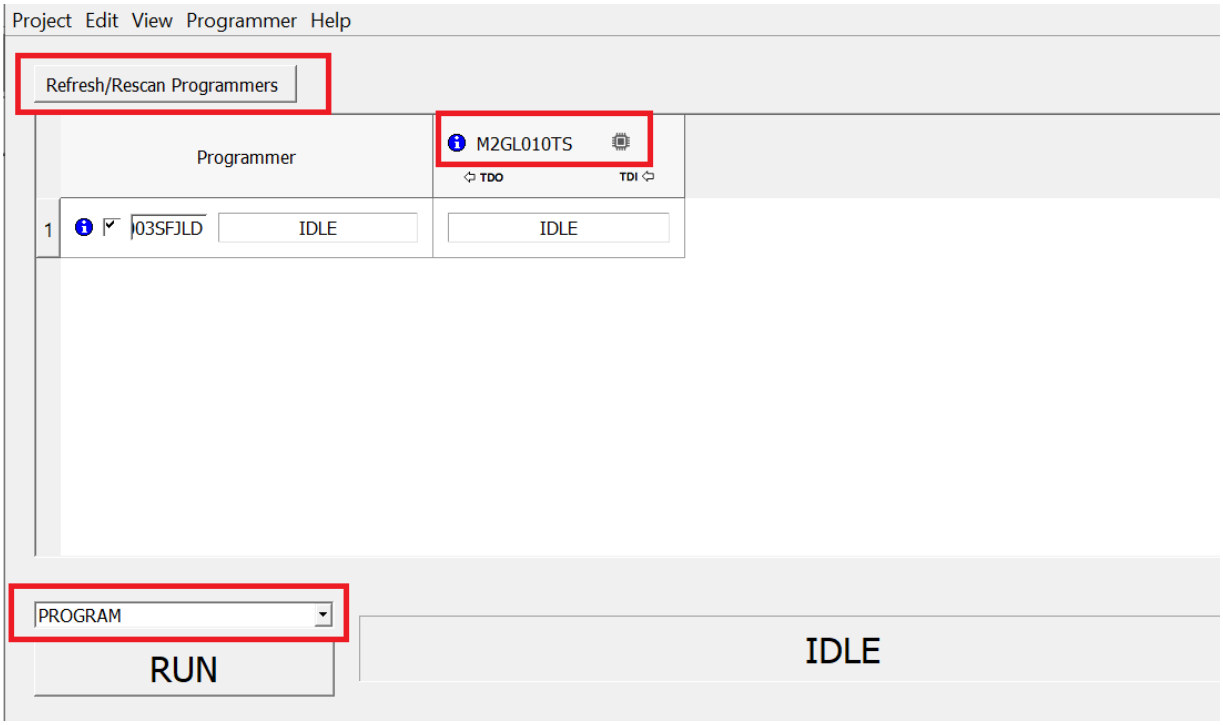
- **Programming job file:** Click **Browse**, and navigate to the location where the .job file is located and select the file.
  - `<download_folder>\m2gl_dg0534_df\Programming_Job`
  - `<download_folder>\m2s_dg0534_df\Programming_Job`
- **FlashPro Express job project name:** Click **Browse** and navigate to the location where you want to save the project.

Figure 33 • **New Job Project from FlashPro Express Job**



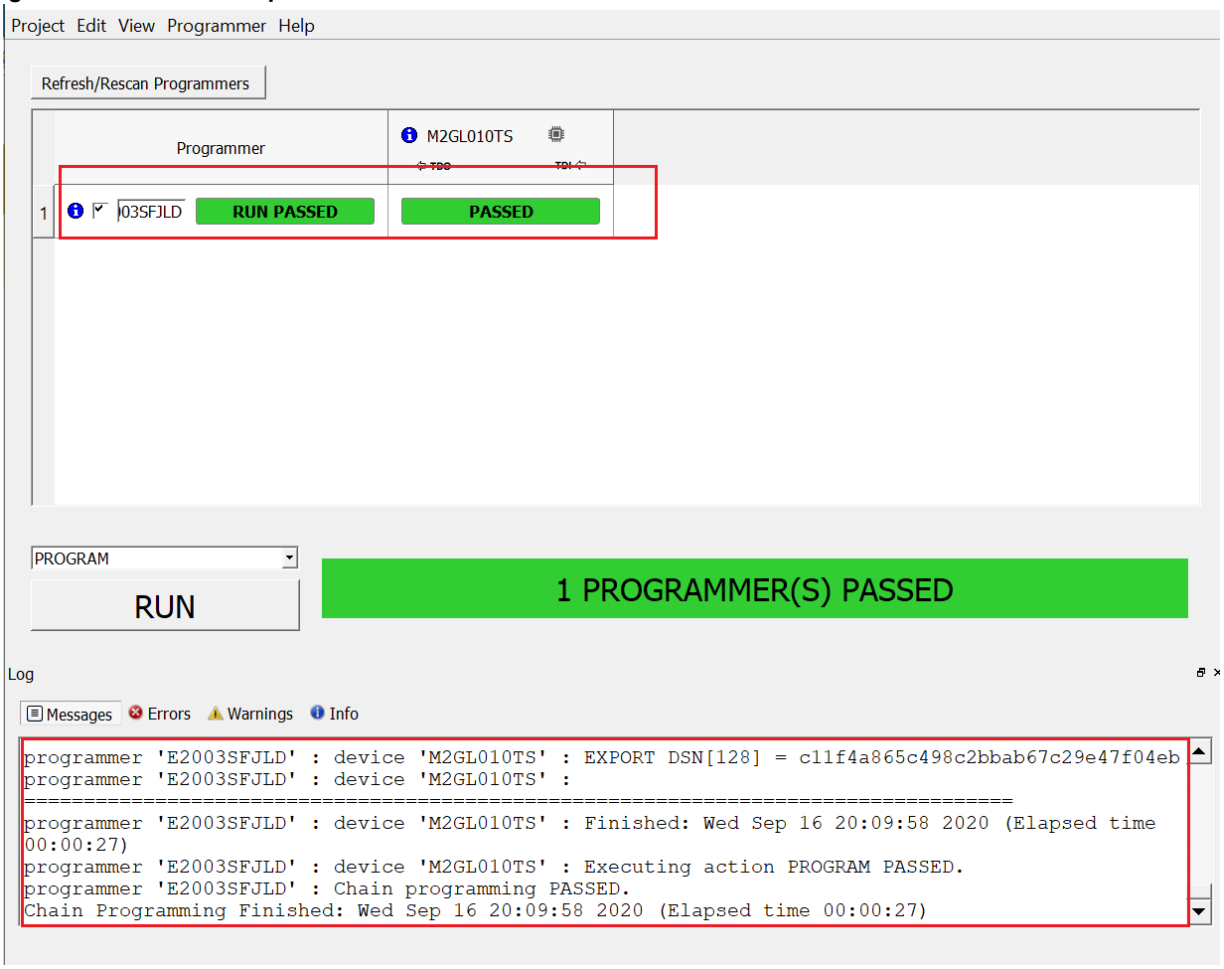
7. Click **OK**. The required programming file is selected and ready to be programmed in the device.
8. The FlashPro Express window appears as shown in the following figure. Confirm that a programmer number appears in the Programmer field. If it does not, confirm the board connections and click **Refresh/Rescan** Programmers.

Figure 34 • **Programming the Device**



9. Click **RUN**. When the device is programmed successfully, a **RUN PASSED** status is displayed as shown in the following figure.

Figure 35 • **FlashPro Express—RUN PASSED**



10. Close **FlashPro Express** or in the Project tab, click **Exit**.

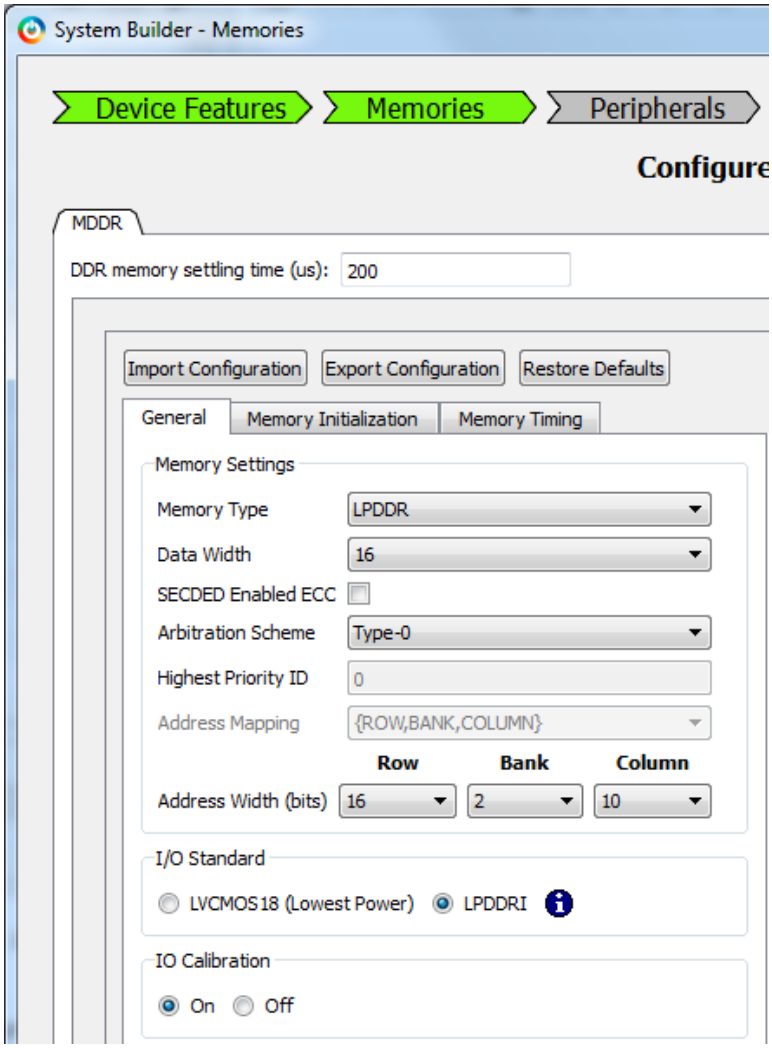
5 Appendix 2: Configuring MDDR Controller

This section describes how to configure the MDDR controller registers using Libero SoC. The configuration options for MDDR are available at the **MDDR** tab of the **Memories** tab in the system builder. Figure 36 shows the **MDDR** tab.

The SmartFusion2 Security and IGLOO2 Evaluation Kit has the LPDDR memory from Micron. All values provided here are from the Micron datasheet; part number, MT46H32M16LF.

Note: The *Automotive Mobile Low-Power DDR SDRAM Datasheet* is available to download it from Micron website.

Figure 36 • **System Builder - Memories - MDDR Tab**



5.1 MDDR Configuration Tab

When using an external memory, the memory controller must wait for the memory to initialize (settling time) before accessing it. The SmartFusion2 Security and IGLOO2 Evaluation Kit uses the LPDDR memory. Therefore, the DDR controller has to wait at least 200 µs. Provide 200 as the value for the field, **DDR memory settling time (µs)**.

Note: All the values provided here are from the Micron datasheet. The parameters can be configured according to the user's requirements.

5.1.1 General

This section shows the configurations of the **General** tab:

- **Memory Type: LPDDR**
- **Data Width**: 16
- **Address Width (bits)**:
  - **Row**: 16
  - **Bank**: 2
  - **Column**: 10

Figure 37 shows the **General** tab after configuration parameters are set.

Figure 37 • **System Builder MDDR Configuration (IGLOO2)– General Tab**

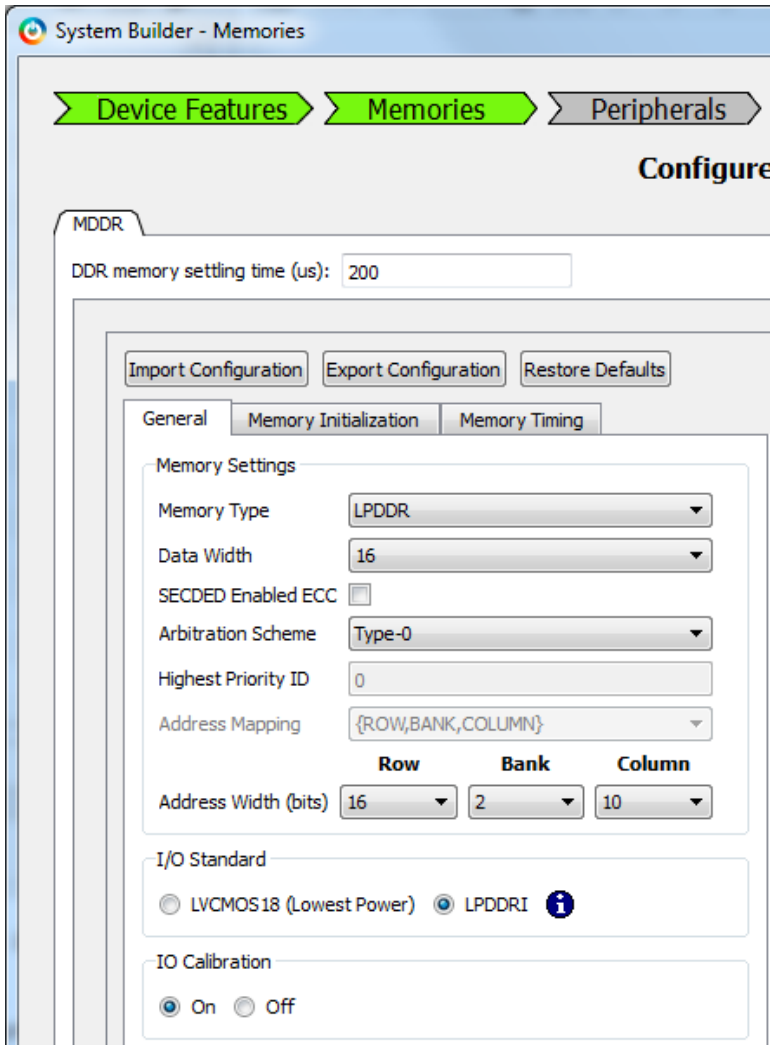
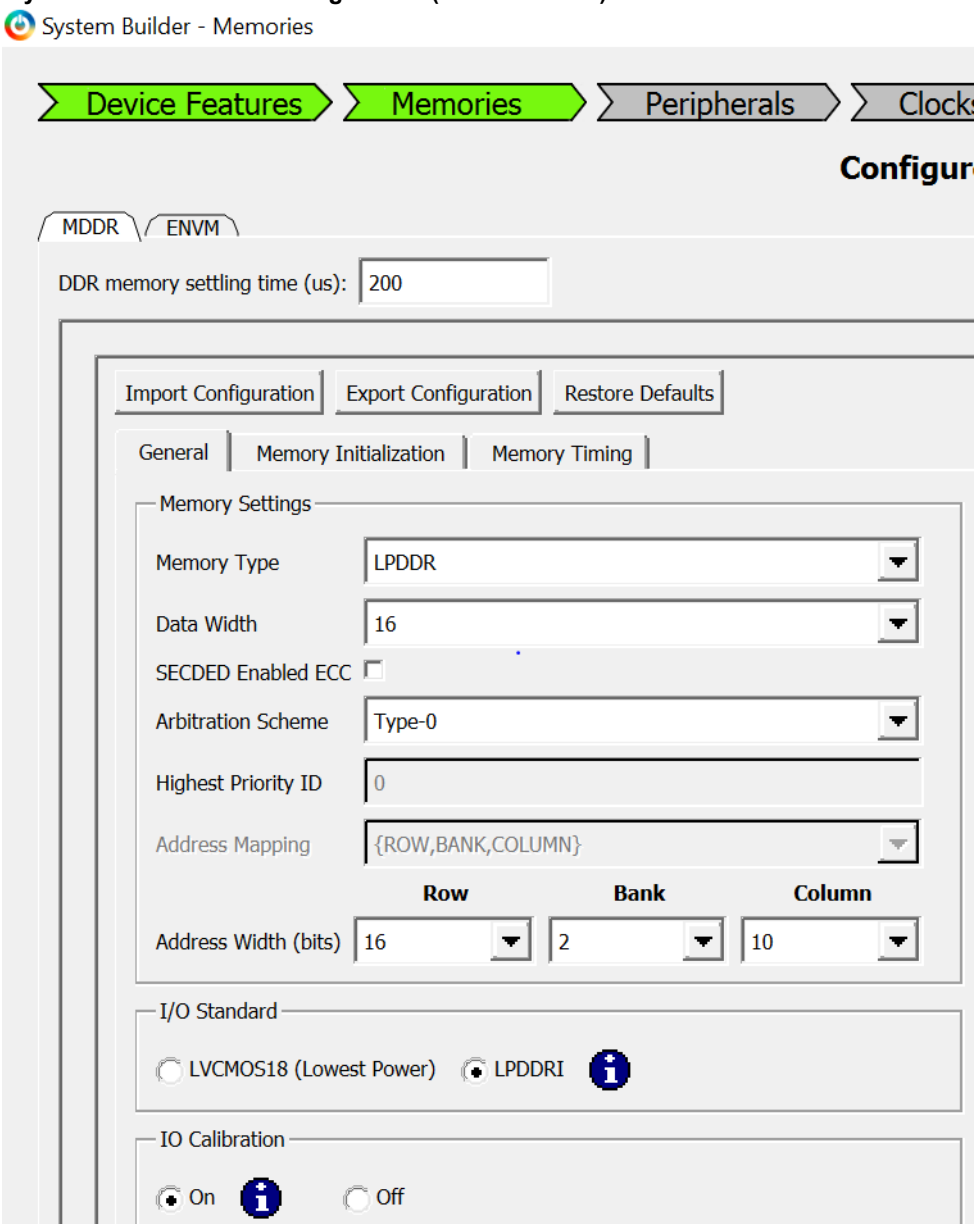


Figure 38 • **System Builder MDDR Configuration (SmartFusion2)– General Tab**



5.1.2 Memory Initialization

This section shows the configurations of the **Memory Initialization** tab:

- **Burst length**: 8
- **Burst Order**: Sequential
- **Timing Mode**: 1T
- **CAS Latency**: 3
- **Self Refresh Enabled**: NO
- **Auto Refresh Burst Count**: Single for IGLOO2 and 8 for SmartFusion2
- **Powerdown Enabled**: YES
- **Stop the Clock**: NO
- **Deep Powerdown enabled**: NO
- **Powerdown Entry Time**: 320

Figure 39 shows the **Memory Initialization** tab after configuration parameters are set.

Figure 39 • **System Builder MDDR Configuration (IGLOO2)– Memory Initialization Tab**

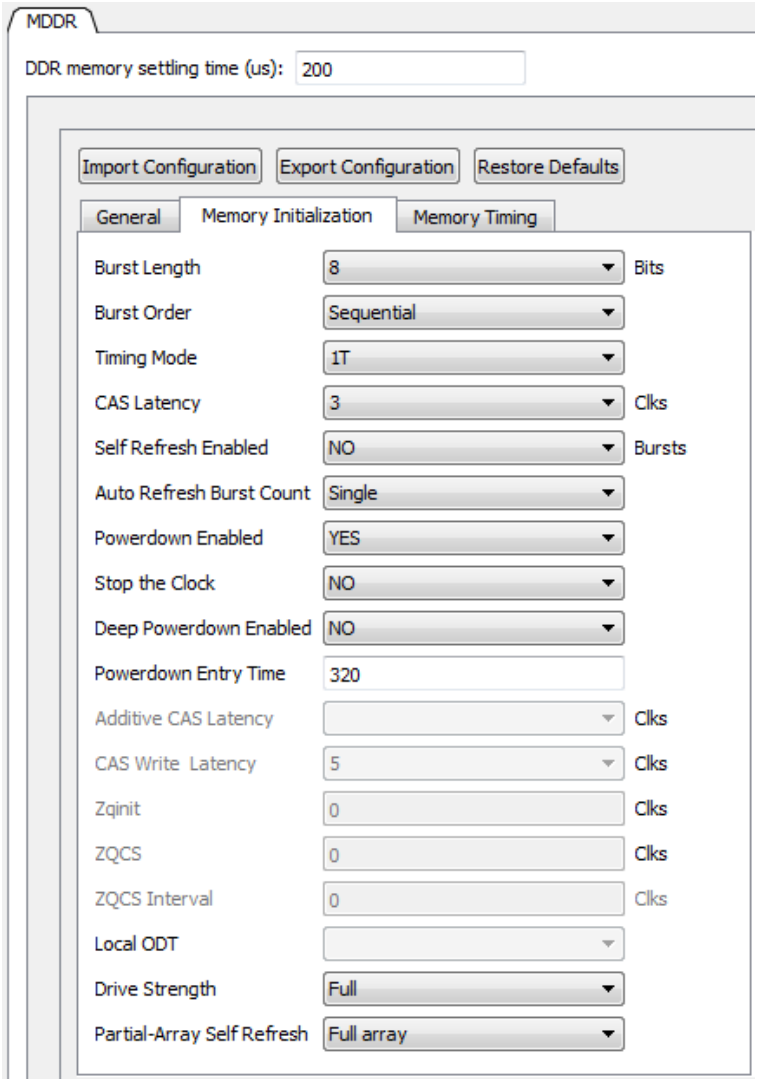
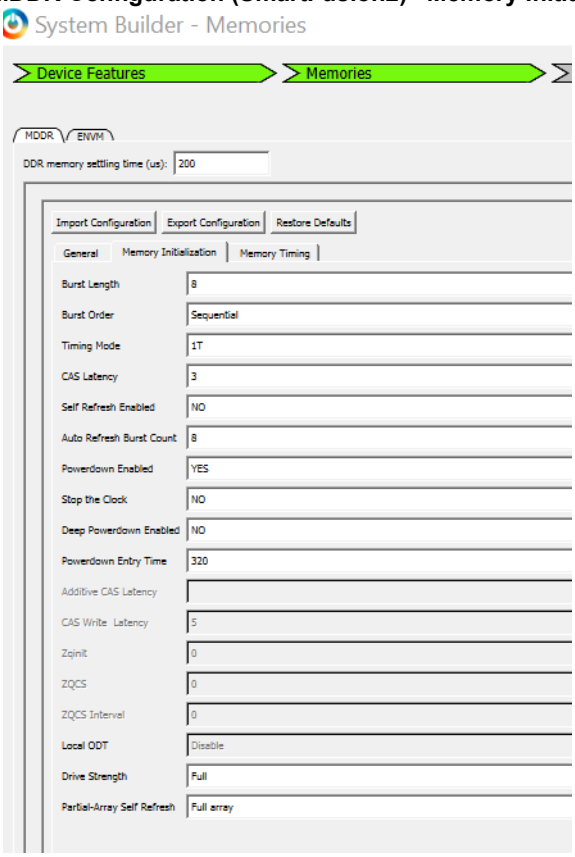


Figure 40 • **System Builder MDDR Configuration (SmartFusion2)– Memory Initialization Tab**



5.1.3 Memory Timing

This section shows the configurations of the **Memory Timing** tab:

- **Time To Hold Reset before INIT**: 0
- **MRD**: 4
- **RAS (Min)**: 8
- **RAS (Max)**: 8192
- **RCD**: 6
- **RP**: 7
- **REFI**: 3104
- **RC**: 12
- **XP**: 3
- **CKE**: 3
- **RFC**: 79
- **FAW**: 0

Figure 41 shows the **Memory Timing** tab after configuration parameters are set.

Figure 41 • **System Builder MDDR Configuration (IGLOO2)– Memory Timing Tab**

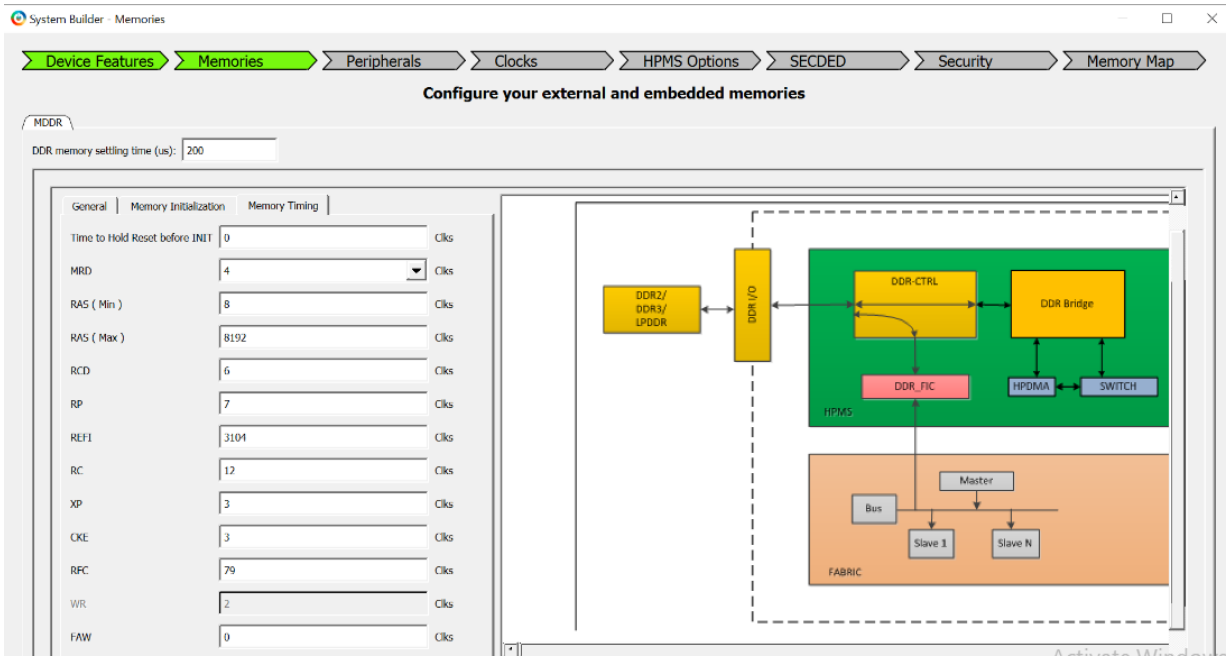
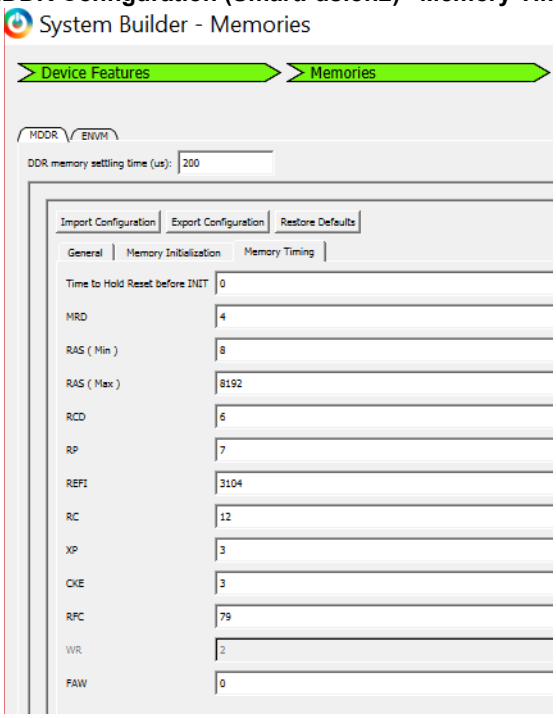


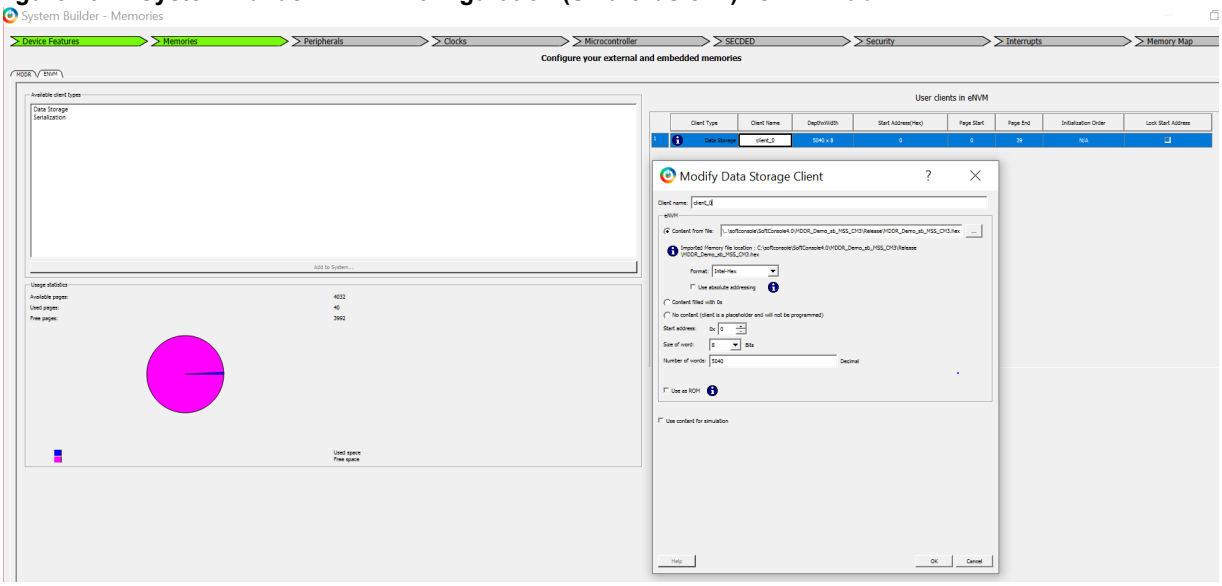
Figure 42 • **System Builder MDDR Configuration (SmartFusion2)– Memory Timing Tab**



5.2 eNVM

This section shows how to configure the **eNVM**.

Figure 43 • **System Builder MDDR Configuration (SmartFusion2)– eNVM Tab**



6 Appendix 3: Finding Correct COM Port Number when Using USB 3.0

FTDI USB to UART converter enumerates the four COM ports. In USB 3.0, the four available COM ports are in **Location 0**.

The USB 3.0 Serial port properties are shown in Figure 44 for IGLOO2 and Figure 45, page 37 for SmartFusion2.

Figure 44 • **USB 3.0 Serial Port Properties - IGLOO2**

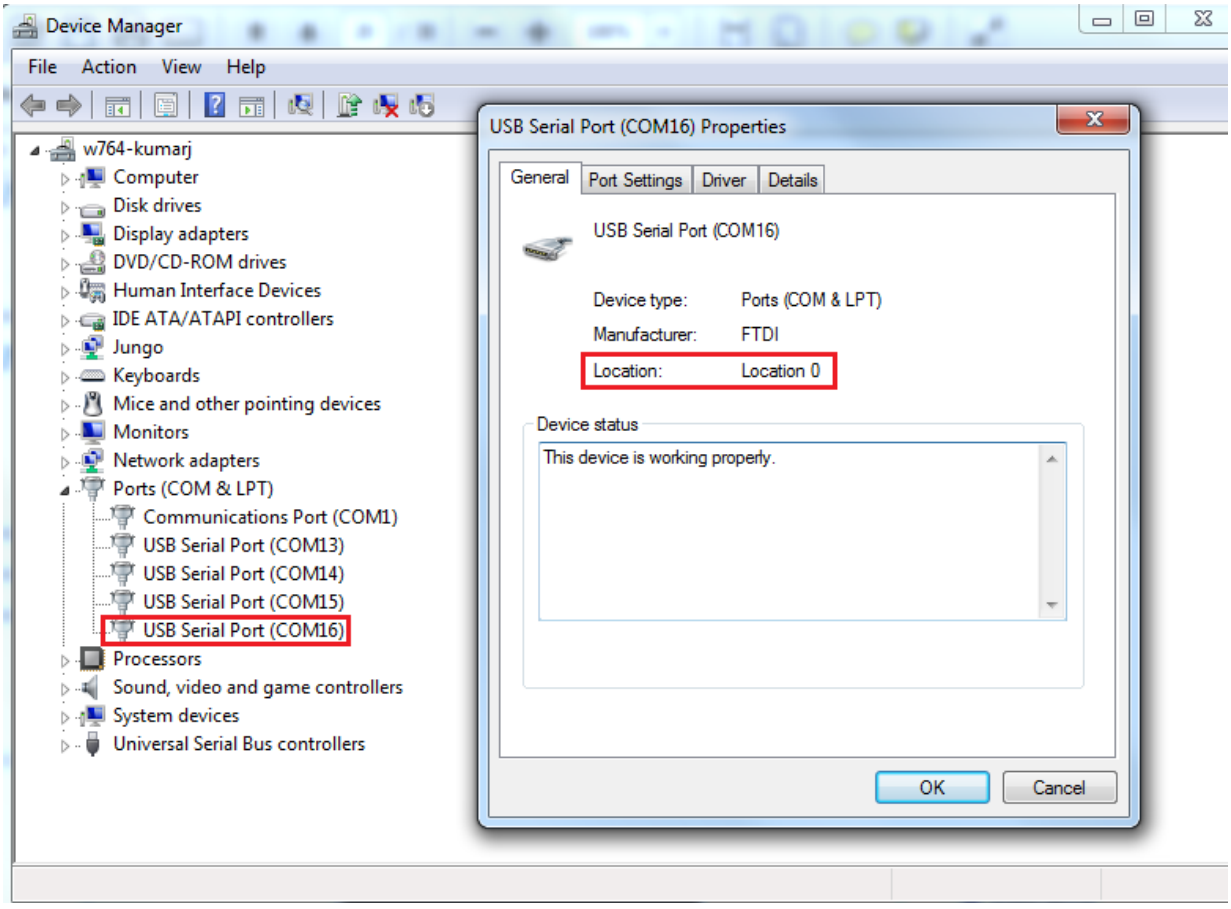
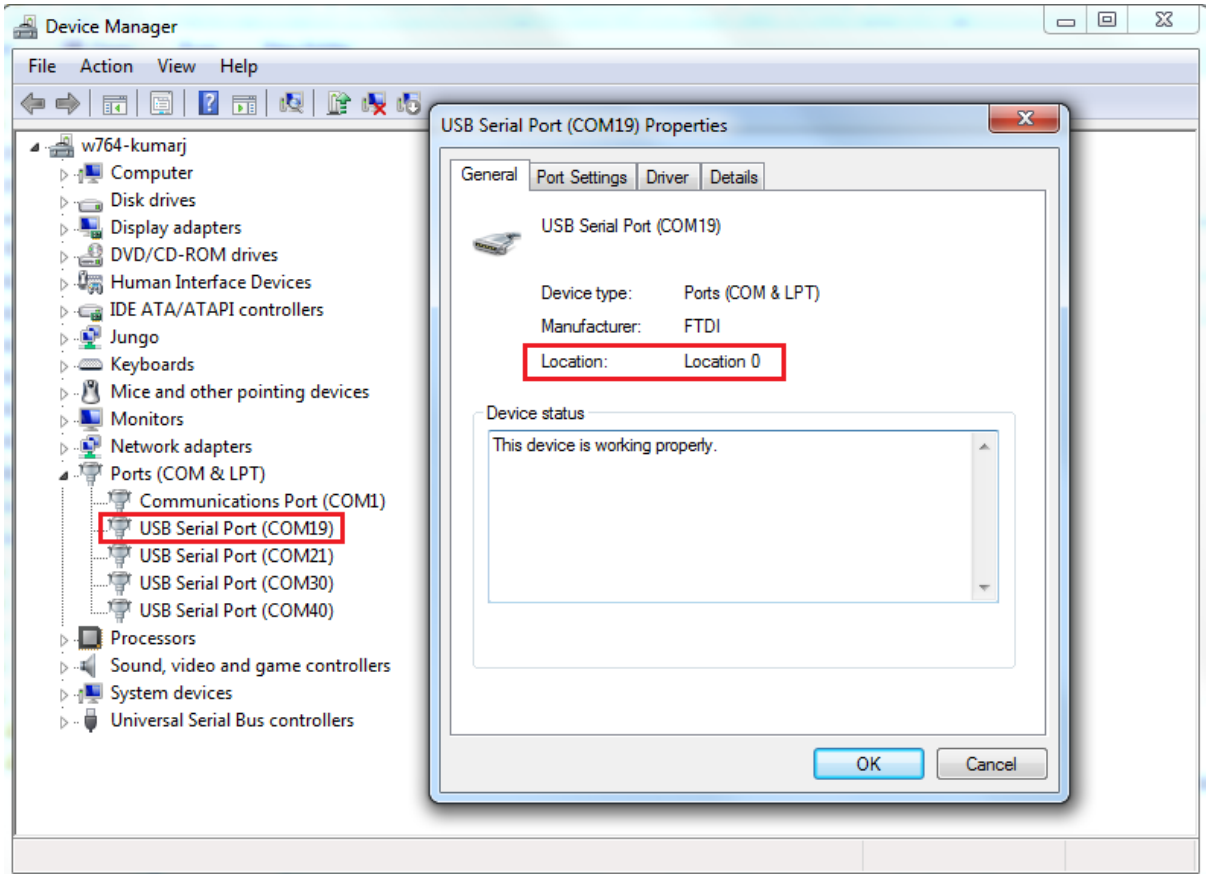


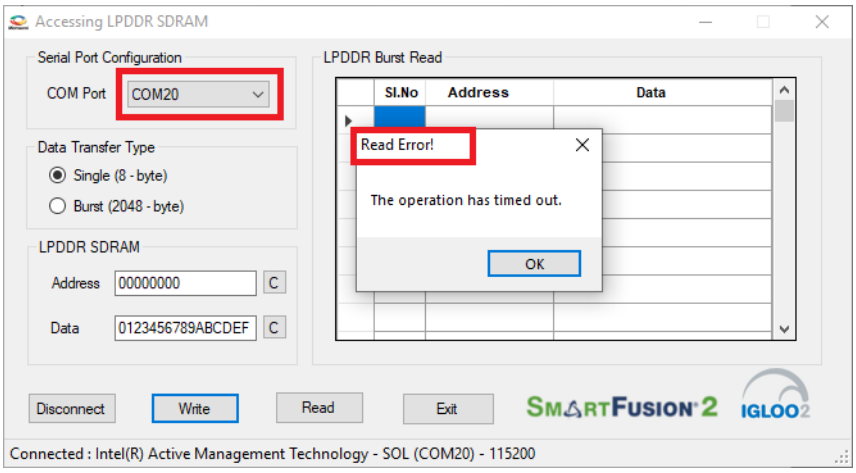
Figure 45 • **USB 3.0 Serial Port Properties - SmartFusion2**



To find out the correct COM port, program the SmartFusion2 Security and IGLOO2 Evaluation Kit board with the provided programming file. Connect each available COM port and click **Write**. If a wrong COM port is selected, the GUI displays the read error. Try with all four available COM ports until the read error message disappears.

The **Read Error**! message is shown in Figure 46.

Figure 46 • **Read Error Message - SmartFusion2/IGLOO2**



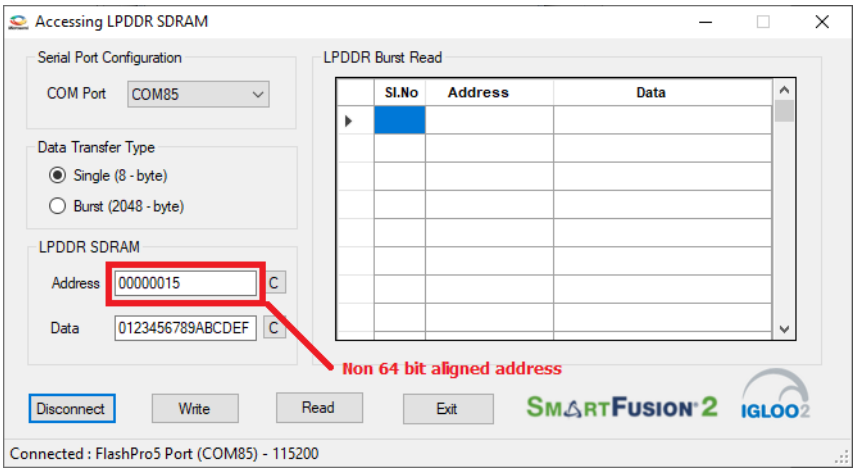
7 Appendix 4: Performing Write/Read Operation when Non 64-bit Aligned Address is Provided

When a non 64-bit aligned address is provided in the GUI, the GUI converts it into the 64-bit aligned address (0x0, 0x8, 0x10, 0x18, 0x20, 0x28, 0x30, 0x38 …) and performs the write/read operation.

1. Enter the non 64-bit aligned 32-bit address in HEX format.
2. Enter the 64-bit data in HEX format.

The non 64-bit aligned address entered in the GUI is shown in Figure 47.

Figure 47 • **Non 64-bit Aligned Address - SmartFusion2/IGLOO2**



3. Click **Write** to perform the write operation. GUI converts the address into a 64-bit aligned address and performs the write operation.

The GUI pop-up information message and converted 64-bit aligned address is shown in Figure 48.

Figure 48 • **Converted 64-bit Aligned Address - SmartFusion2/IGLOO2**