Interfacing SmartFusion2 SoC FPGA with External LPDDR Memory through MDDR Controller - Libero SoC v11.6

DG0568 Demo Guide

Microsemi®

Revision History

Date	Revision	Change
16 November, 2015	4	Fourth release
16 October, 2015	3	Third release
19 February, 2015	2	Second release
22 August, 2014	1	First release

Confidentiality Status

This document is a Non-Confidential.

Table of Content

Preface

About this document

This demo guide is for SmartFusion®2 system-on-chip (SoC) field programmable gate array (FPGA) devices. It provides instructions on how to use the corresponding reference design.

Intended Audience

The SmartFusion2 devices are used by:

- FPGA designers
- System-level designers
- Embedded designers

References

Microsemi Publications

- *UG0446: SmartFusion2 SoC FPGA and IGLOO2 FPGA High Speed DDR Interfaces User Guide*
- *SmartFusion2 System Builder User Guide*
- *UG0594: M2S090TS-EVAL-KIT SmartFusion2 Security Evaluation Kit User Guide*

Refer to the following web page for a complete and up-to-date listing of the SmartFusion2 device documentation: *http://www.microsemi.com/products/fpga-soc/soc-fpga/smartfusion2#documentation*.

Interfacing SmartFusion2 SoC FPGA with External LPDDR Memory through MDDR Controller

Introduction

This demo shows the microcontroller subsystem (MSS) double-data rate (DDR) controller accessing the external DDR SDRAM memories in the SmartFusion2 devices. The demo has two parts:

- Demo using simulation
- Demo using the SmartFusion2 Security 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 Evaluation Kit board using the microcontroller subsystem DDR (MDDR) controller. A utility, `SF2_MDDR_Demo` is provided along 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.

Design Requirements

Table 1 shows the hardware and software design requirements.

Table 1 • **Design Requirements**

Design Requirements	**Description**
Hardware Requirements	
SmartFusion2 Security Evaluation Kit: • FlashPro4 programmer • 12 V adapter • USB A to Mini-B cable	Rev D or later
Host PC or Laptop	Any 64-bit Windows Operating System
Software Requirements	
Libero® System-on-Chip (SoC)	v11.6
FlashPro programming software	v11.6
Microsoft .NET Framework 4	–
Host PC Drivers	USB to UART drivers

Demo Design

Introduction

The demo design files are available for download from the following path in the Microsemi website:

http://soc.microsemi.com/download/rsc/?f=m2s_dg0568_liberov11p6_df

Design files include:

- Demo_utility
- Libero_project
  - SF2_MDDR_Demo
- Programming_file
- Source_files
- readme.txt

Figure 1 shows the top-level structure of the design files. For further details, refer to the `readme.txt` file.

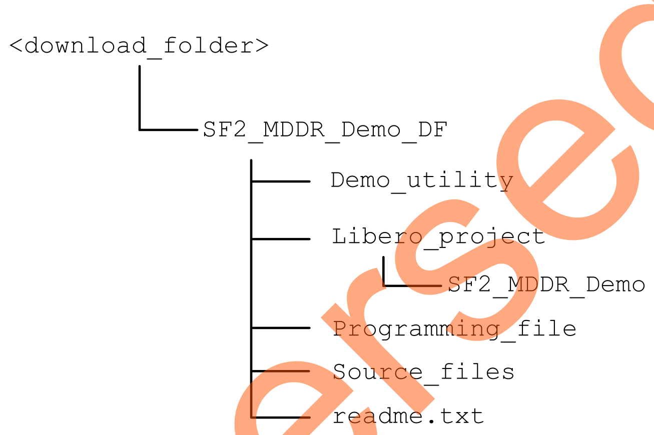


Figure 1 **• Demo Design Files Top-Level Structure**

In the demo design, the AXI Master implemented in the FPGA Fabric accesses the LPDDR memory present in the SmartFusion2 Security Evaluation Kit board using the MDDR controller. The AXI Master logic communicates to the MDDR controller through Core AXI interface and the DDR_FIC interface. The read/write operations initiated by the `SF2_MDDR_Demo` utility are sent to the UART_IF block using the UART protocol. The AXI Master receives the address and the 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 two port static random-access memory (TPSRAM). When the read operation is complete, the read data is sent to the Host PC through UART.

Figure 2 shows the top-level view of demo design.

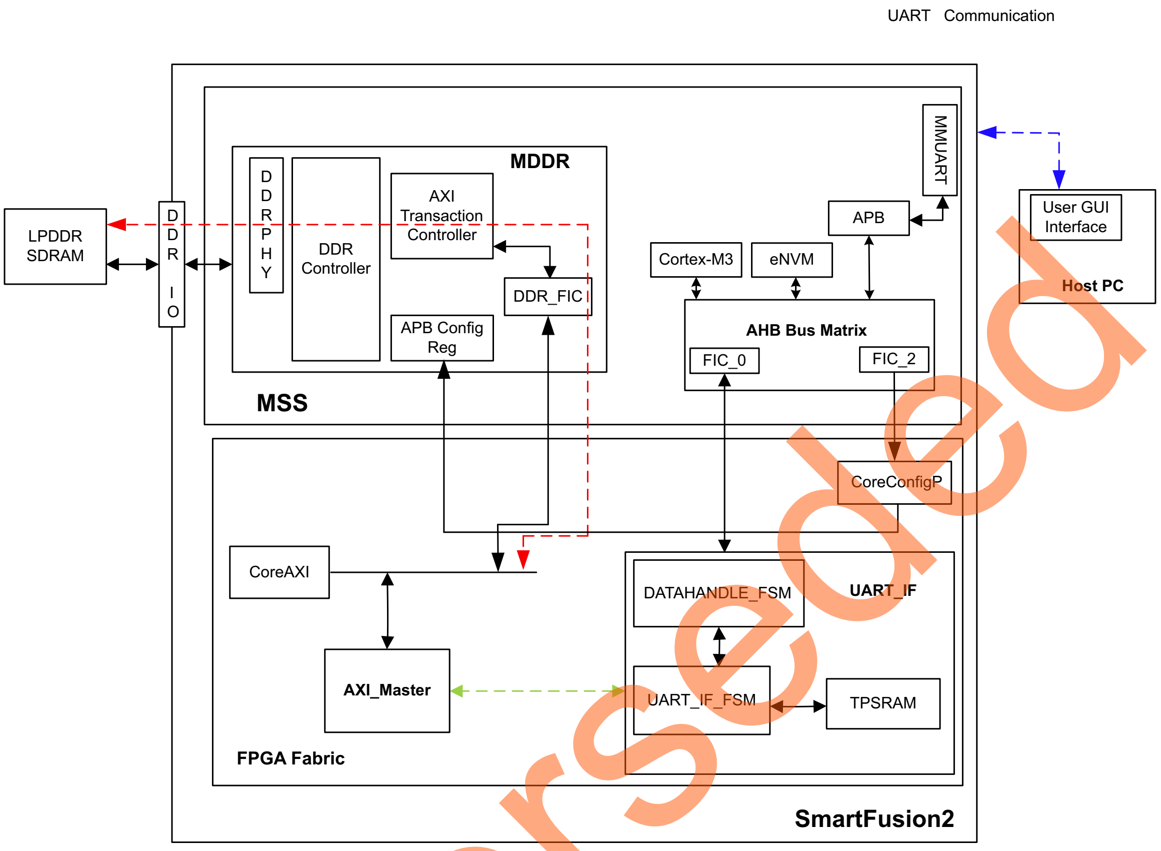


Figure 2 • **SmartFusion2 MDDR Demo Block Diagram**

In this demo design, different blocks are configured as shown below:

- 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 AXI bus interface.
- Both 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

Refer to "Appendix A: Configuring MDDR Controller" on page 28 for information on how to configure the DDR controller.

Demo Design Features

The SmartFusion2 MDDR demo design has the following features:

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

Demo Design Description

The demo design consists of the following SmartDesign components:

- **MDDR_Demo_top_0**: This 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 Evaluation Kit board.

Figure 3 shows the **MDDR_Demo_top_0** and **UART_IF_0** connections.

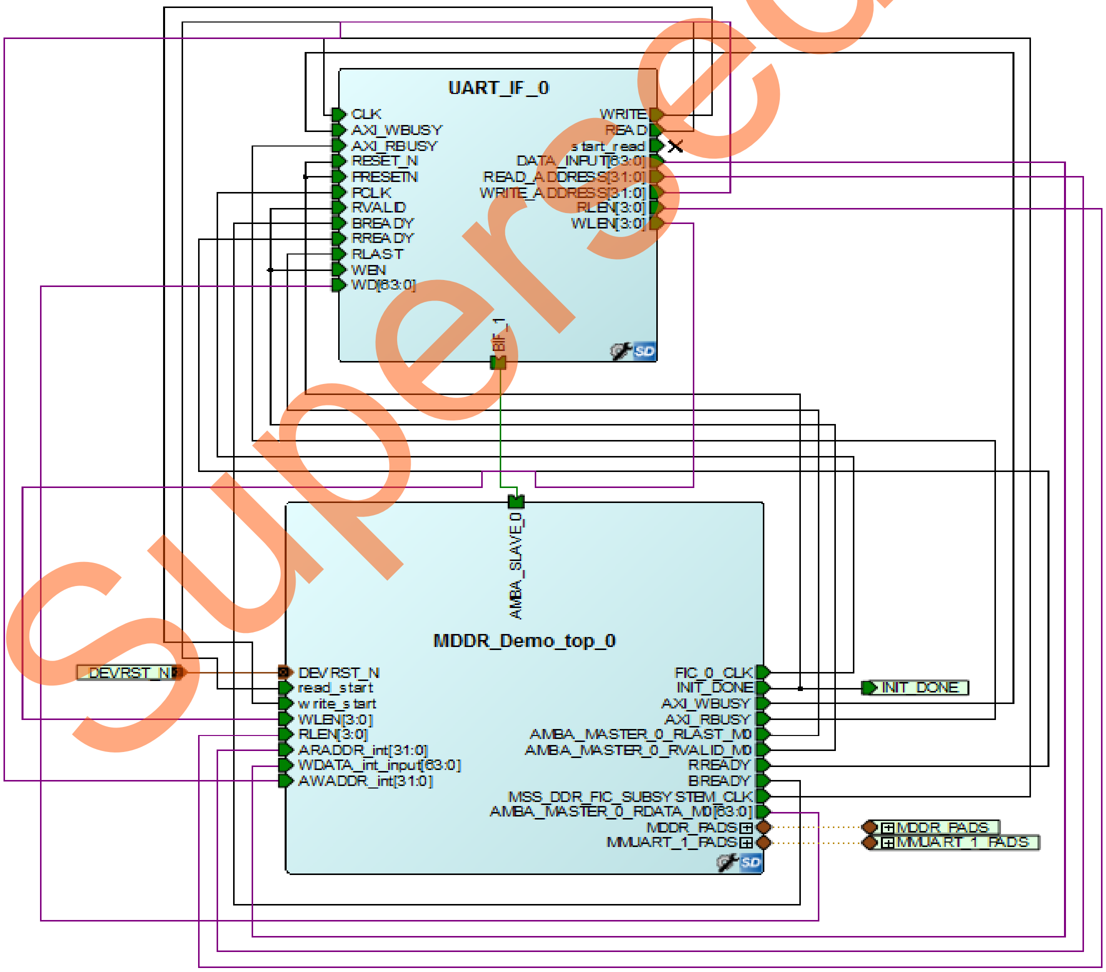


***Figure 3* • SF2_MDDR_Demo SmartDesign**

MDDR_Demo_top_0

This 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 4 shows the MDDR_Demo_top_0 SmartDesign component.

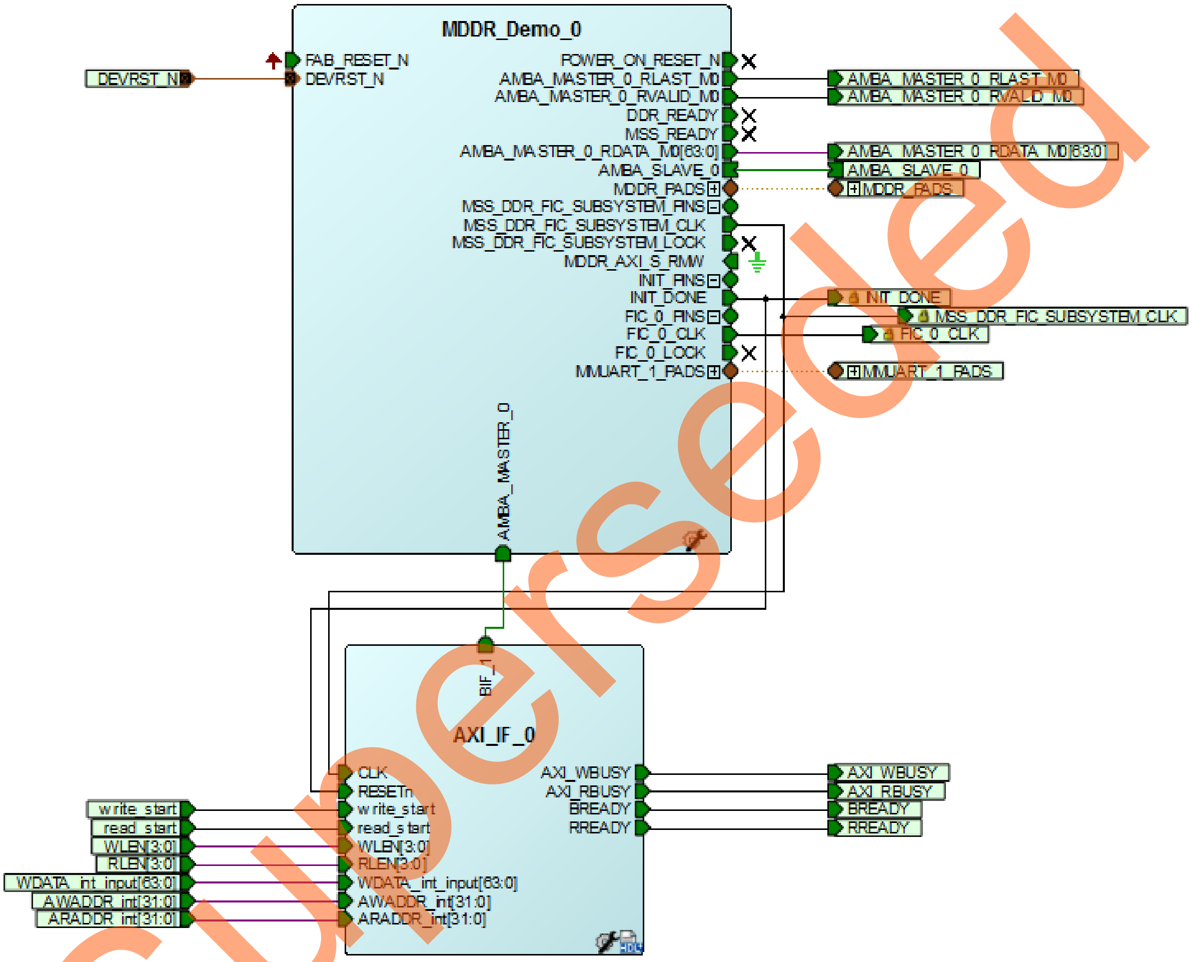


Figure 4 **• MDDR_Demo_top_0 SmartDesign Component**

UART_IF_0

The UART_IF_0 SmartDesign component handles the 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.

Figure 5 shows the UART_IF_0 SmartDesign component.

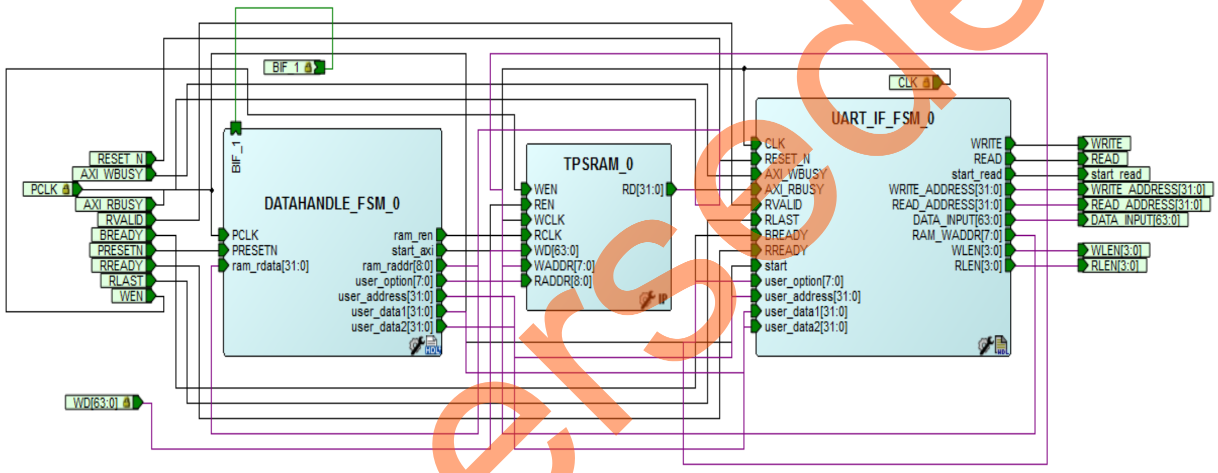


***Figure 5* • UART_IF_0 SmartDesign Component**

Running the Demo using Simulation

Introduction

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

The simulation is set to run the following:

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

Figure 6 shows the AXI_LPDDR_Simulation SmartDesign testbench. The AXI_testbench provides the read or write operations, burst length, address, and data to the MDDR_Demo_top_0 SmartDesign component.

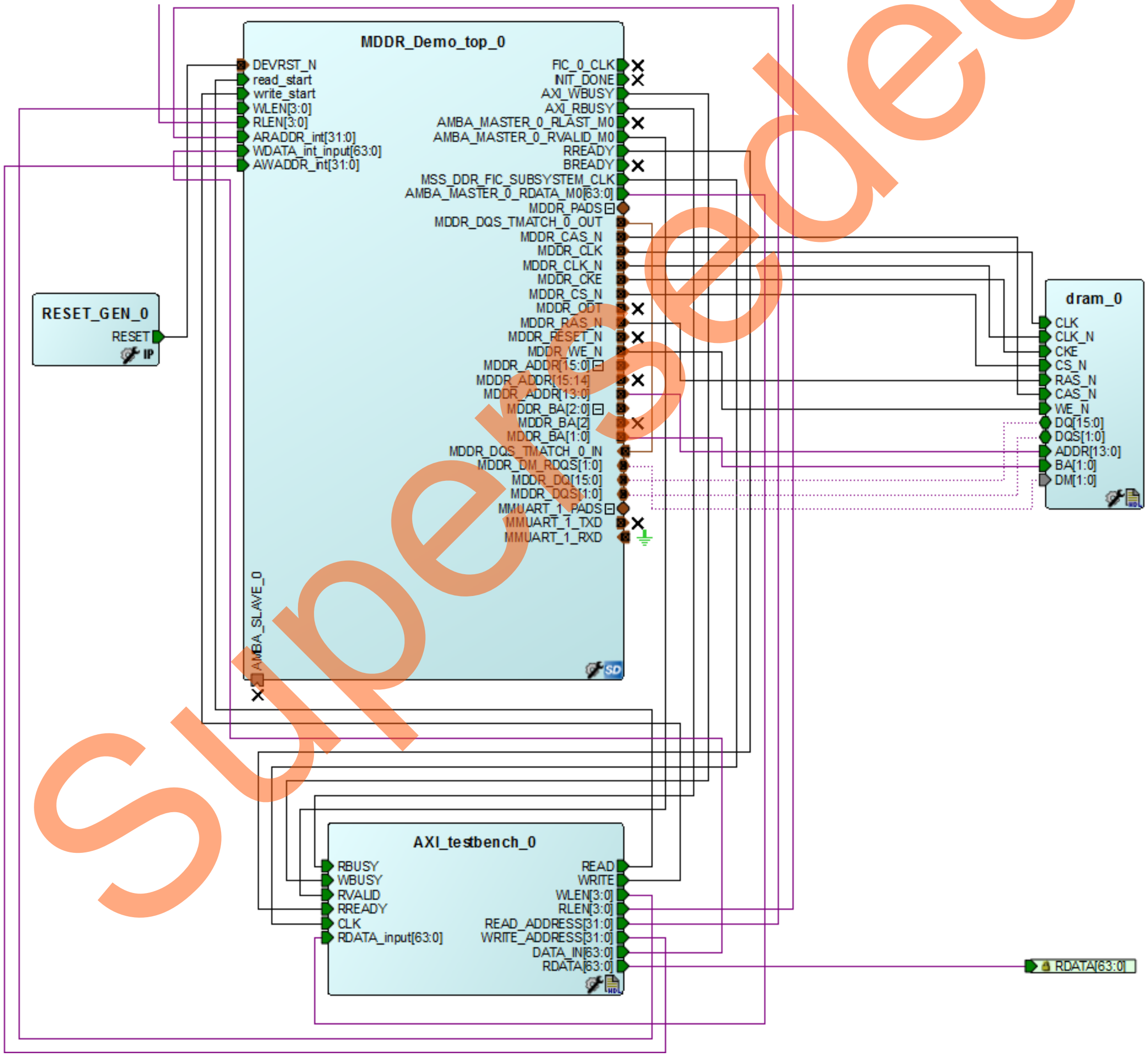


Figure 6 • **AXI_LPDDR_Simulation SmartDesign Testbench**

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

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

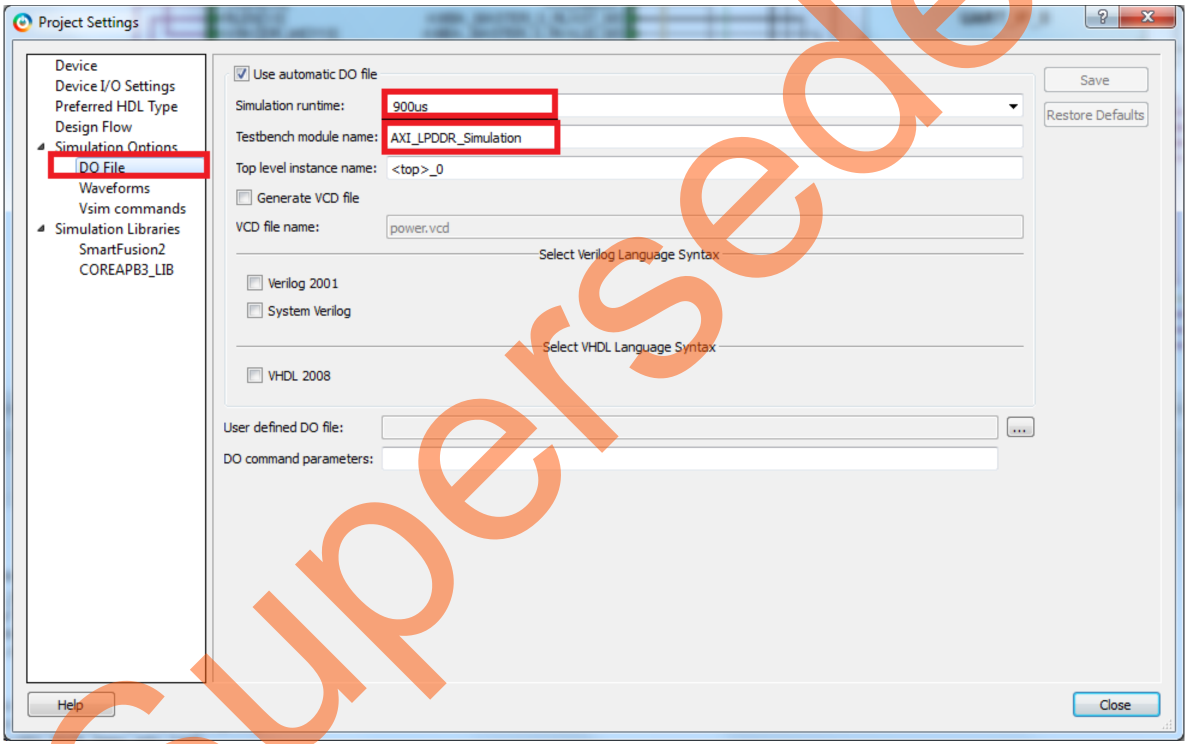
The default location of the files is:

`<Download folder>\SF2_MDDR_Demo_DF\Libero_project\SF2_MDDR_Demo\stimulus`

Simulation

Simulation setup configuration can be set properly using the following steps:

1. Launch the Libero SoC software.
2. Browse the *SF2_MDDR_Demo* 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 7.



☑ Use automatic DO file

Simulation runtime: 900us

Testbench module name: AXI_LPDDR_Simulation

Top level instance name: <top>_0

☐ Generate VCD file

VCD file name: power.vcd

Select Verilog Language Syntax

☐ Verilog 2001

☐ System Verilog

Select VHDL Language Syntax

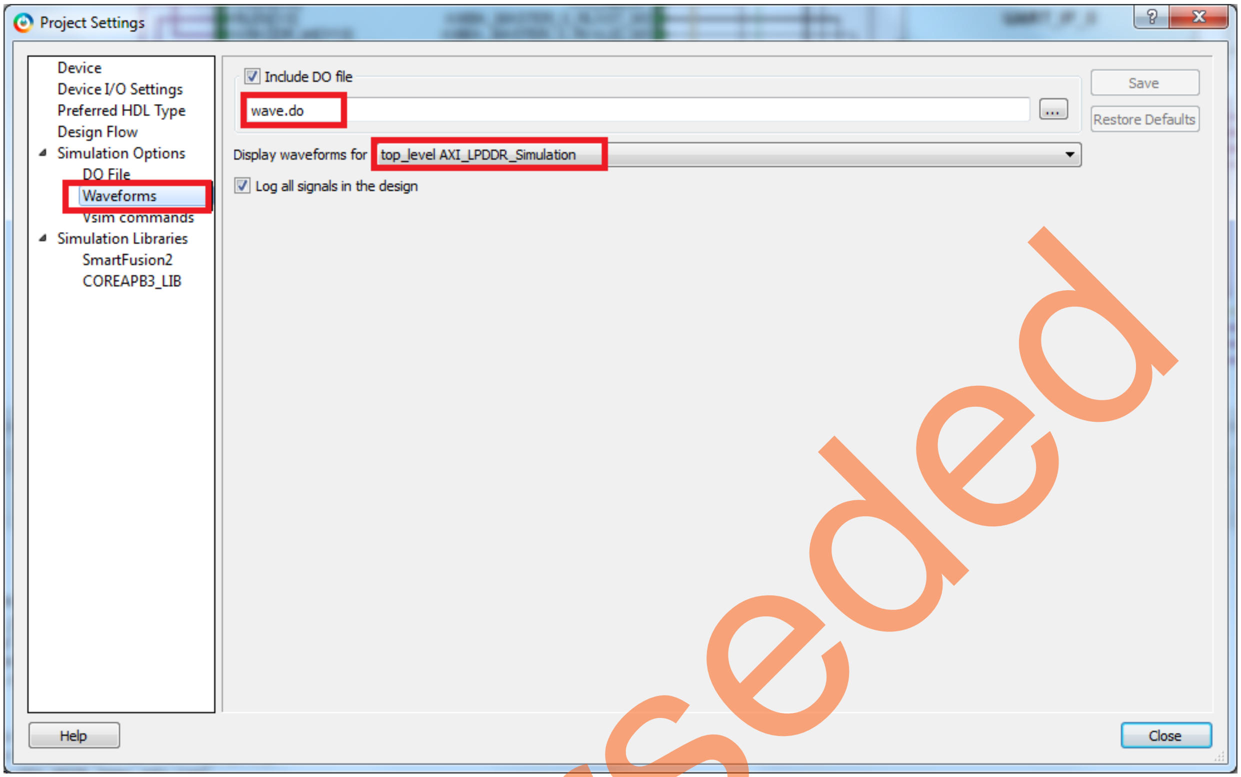
☐ VHDL 2008

User defined DO file:

DO command parameters:

***Figure 7* • DO File Settings**

5. Ensure that the **Waveforms** tab has the configuration as shown in Figure 8.



☑ Include DO file

wave.do

Display waveforms for top_level AXI_LPDDR_Simulation

☑ Log all signals in the design

***Figure 8* • Waveforms Settings**

6. Go to **Design Flow** tab.
7. Right-click **Simulate** under **Verify Pre-Synthesized Design** and then select, **Organize Input Files** > **Organize Stimulus Files** as shown in Figure 9.

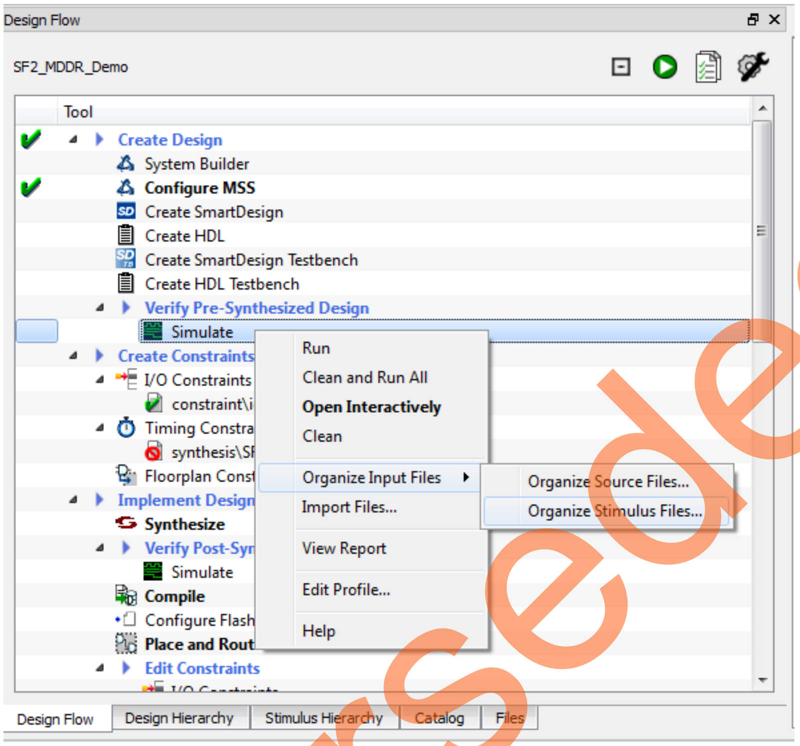
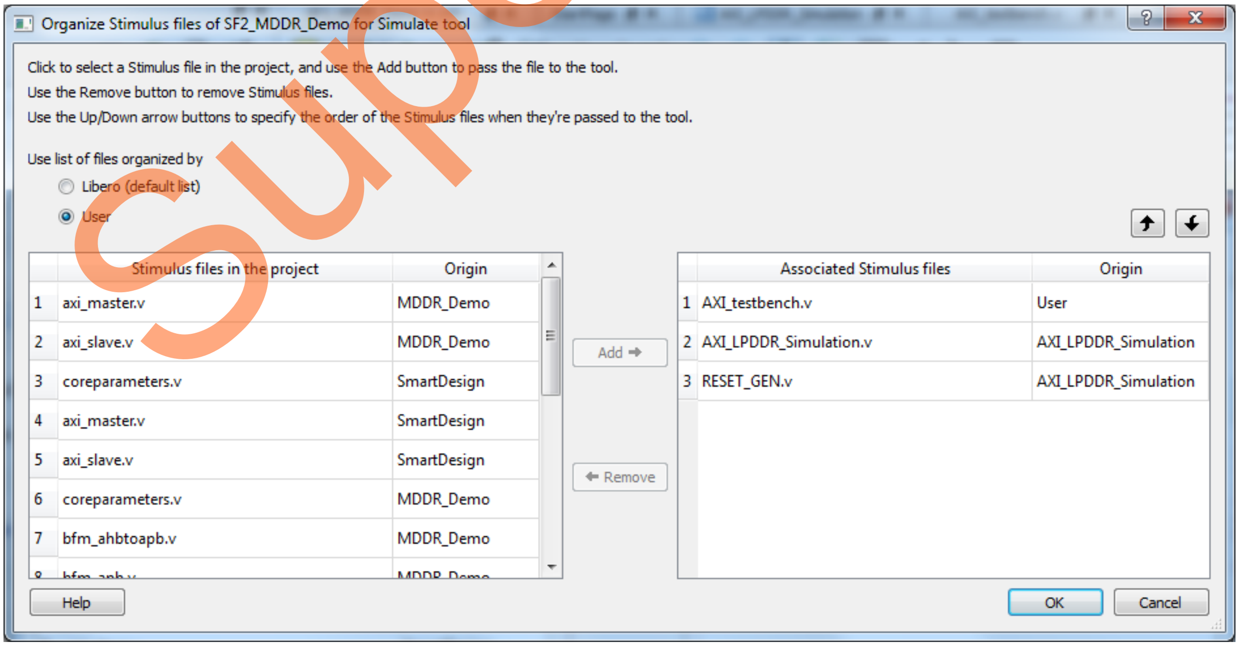


Figure 9 • **Invoking Organize Stimulus Files Window**

8. Ensure that the **Organize Stimulus files** window has the configuration as shown in Figure 10.



Click to select a Stimulus file in the project, and use the Add button to pass the file to the tool.
Use the Remove button to remove Stimulus files.
Use the Up/Down arrow buttons to specify the order of the Stimulus files when they're passed to the tool.

Use list of files organized by
○ Libero (default list)
● User

	Stimulus files in the project	Origin
1	axi_master.v	MDDR_Demo
2	axi_slave.v	MDDR_Demo
3	coreparameters.v	SmartDesign
4	axi_master.v	SmartDesign
5	axi_slave.v	SmartDesign
6	coreparameters.v	MDDR_Demo
7	bfm_ahbtoapb.v	MDDR_Demo
8	bfm_apb.v	MDDR_Demo

	Associated Stimulus files	Origin
1	AXI_testbench.v	User
2	AXI_LPDDR_Simulation.v	AXI_LPDDR_Simulation
3	RESET_GEN.v	AXI_LPDDR_Simulation

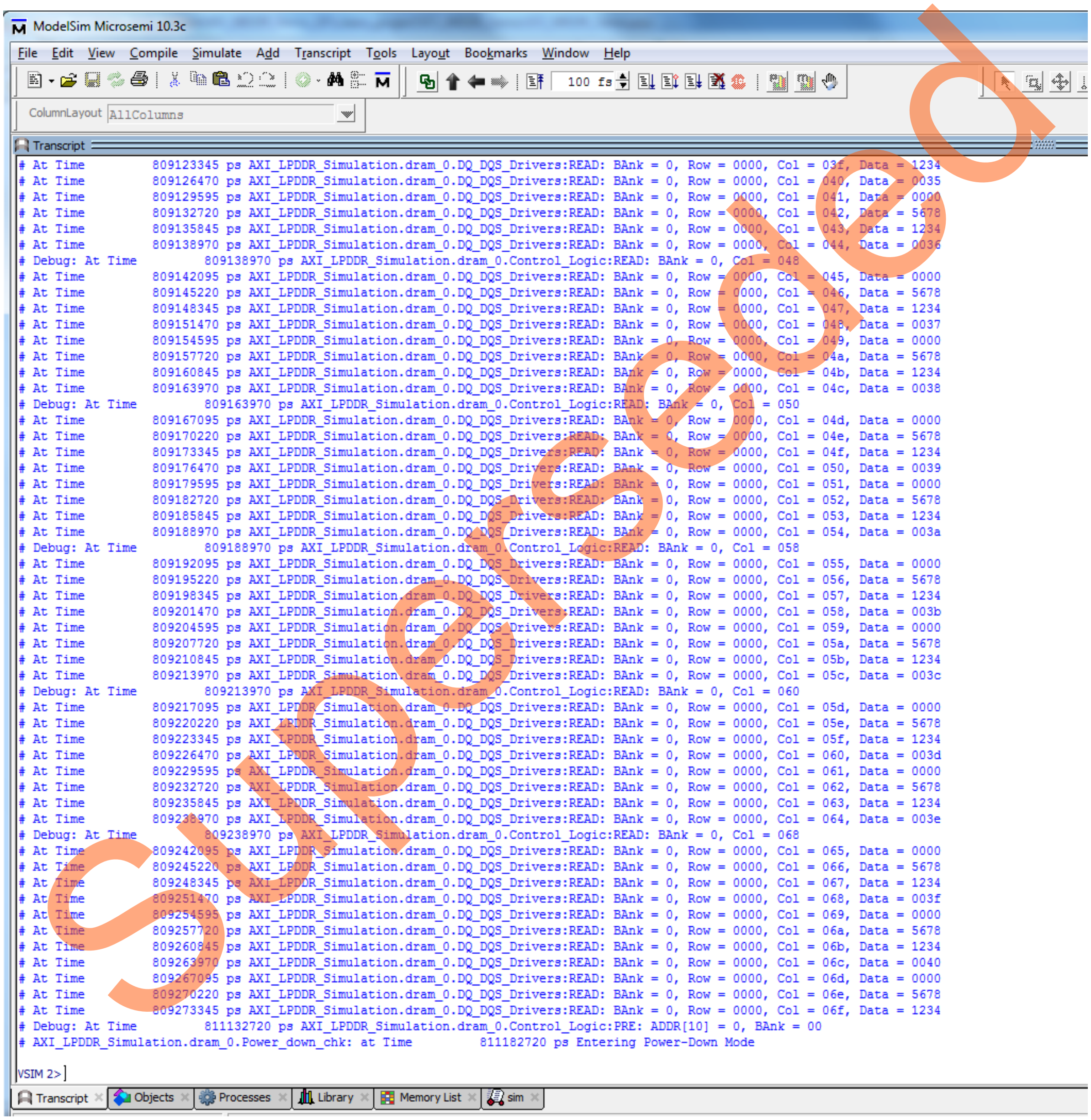
Figure 10 • **Organize Stimulus Files Window**

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. Simulation requires 900 µs to complete as mentioned in the 3rd point under "Simulation" section on page 12.

Figure 11 shows the transcript window of the simulation.


ModelSim Microsemi 10.3c
File Edit View Compile Simulate Add Transcript Tools Layout Bookmarks Window Help
100 fs
ColumnLayout AllColumns
Transcript

```
# At Time          809123345 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 03f, Data = 1234
# At Time          809126470 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 040, Data = 0035
# At Time          809129595 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 041, Data = 0000
# At Time          809132720 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 042, Data = 5678
# At Time          809135845 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 043, Data = 1234
# At Time          809138970 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 044, Data = 0036
# Debug: At Time          809138970 ps AXI_LPDDR_Simulation.dram_0.Control_Logic:READ: BAnk = 0, Col = 048
# At Time          809142095 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 045, Data = 0000
# At Time          809145220 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 046, Data = 5678
# At Time          809148345 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 047, Data = 1234
# At Time          809151470 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 048, Data = 0037
# At Time          809154595 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 049, Data = 0000
# At Time          809157720 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 04a, Data = 5678
# At Time          809160845 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 04b, Data = 1234
# At Time          809163970 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 04c, Data = 0038
# Debug: At Time          809163970 ps AXI_LPDDR_Simulation.dram_0.Control_Logic:READ: BAnk = 0, Col = 050
# At Time          809167095 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 04d, Data = 0000
# At Time          809170220 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 04e, Data = 5678
# At Time          809173345 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 04f, Data = 1234
# At Time          809176470 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 050, Data = 0039
# At Time          809179595 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 051, Data = 0000
# At Time          809182720 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 052, Data = 5678
# At Time          809185845 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 053, Data = 1234
# At Time          809188970 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 054, Data = 003a
# Debug: At Time          809188970 ps AXI_LPDDR_Simulation.dram_0.Control_Logic:READ: BAnk = 0, Col = 058
# At Time          809192095 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 055, Data = 0000
# At Time          809195220 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 056, Data = 5678
# At Time          809198345 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 057, Data = 1234
# At Time          809201470 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 058, Data = 003b
# At Time          809204595 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 059, Data = 0000
# At Time          809207720 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 05a, Data = 5678
# At Time          809210845 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 05b, Data = 1234
# At Time          809213970 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 05c, Data = 003c
# Debug: At Time          809213970 ps AXI_LPDDR_Simulation.dram_0.Control_Logic:READ: BAnk = 0, Col = 060
# At Time          809217095 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 05d, Data = 0000
# At Time          809220220 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 05e, Data = 5678
# At Time          809223345 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 05f, Data = 1234
# At Time          809226470 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 060, Data = 003d
# At Time          809229595 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 061, Data = 0000
# At Time          809232720 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 062, Data = 5678
# At Time          809235845 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 063, Data = 1234
# At Time          809238970 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 064, Data = 003e
# Debug: At Time          809238970 ps AXI_LPDDR_Simulation.dram_0.Control_Logic:READ: BAnk = 0, Col = 068
# At Time          809242095 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 065, Data = 0000
# At Time          809245220 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 066, Data = 5678
# At Time          809248345 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 067, Data = 1234
# At Time          809251470 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 068, Data = 003f
# At Time          809254595 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 069, Data = 0000
# At Time          809257720 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 06a, Data = 5678
# At Time          809260845 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 06b, Data = 1234
# At Time          809263970 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 06c, Data = 0040
# At Time          809267095 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 06d, Data = 0000
# At Time          809270220 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 06e, Data = 5678
# At Time          809273345 ps AXI_LPDDR_Simulation.dram_0.DQ_DQS_Drivers:READ: BAnk = 0, Row = 0000, Col = 06f, Data = 1234
# Debug: At Time          811132720 ps AXI_LPDDR_Simulation.dram_0.Control_Logic:PRE: ADDR[10] = 0, BAnk = 00
# AXI_LPDDR_Simulation.dram_0.Power_down_chk: at Time          811182720 ps Entering Power-Down Mode

VSIM 2>
```

Figure 11 **• Simulation Completed**

Figure 12 shows the single AXI write and AXI read operation.

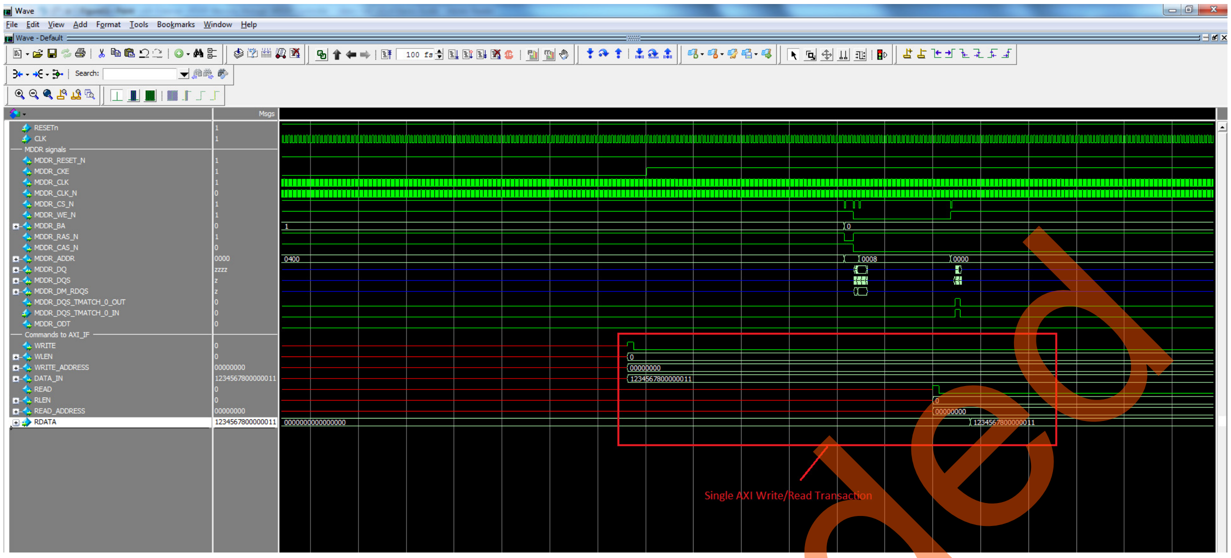


Figure 12 • Single Write and Read Operation

Figure 13 shows the 16-beat AXI burst write and read operation.

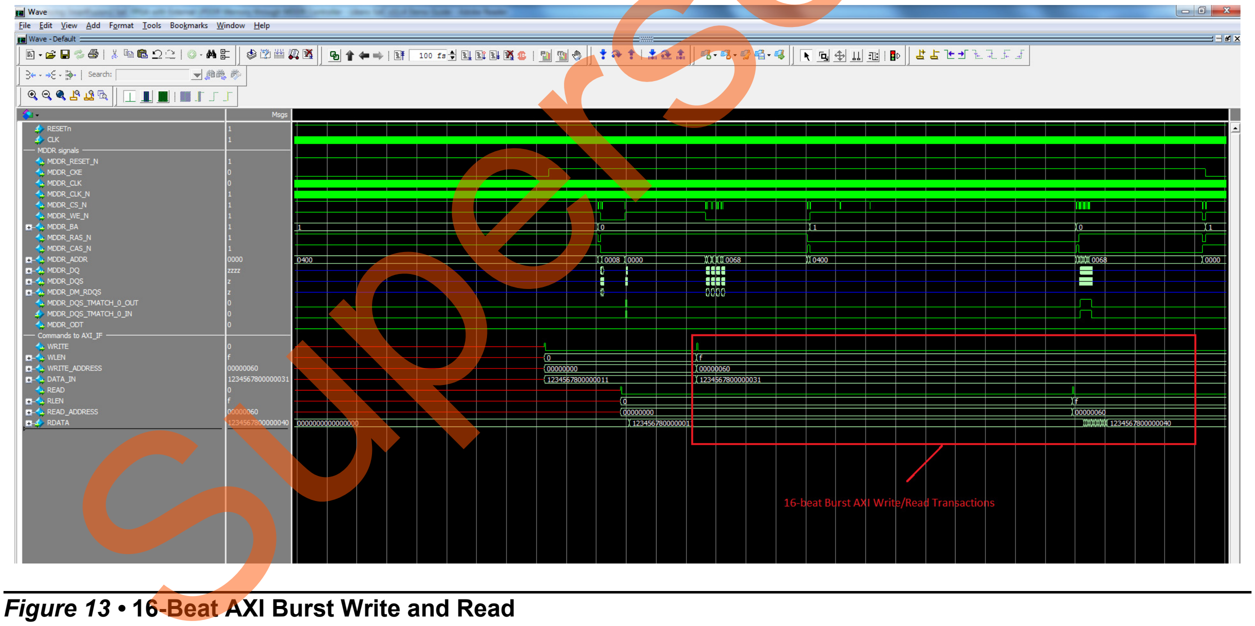


Figure 13 • 16-Beat AXI Burst Write and Read

Setting Up the Hardware Demo

The following steps describe how to setup the hardware demo:

1. Connect the jumpers on the SmartFusion2 Security Evaluation Kit as listed in Table 2.

Table 2 • **SmartFusion2 Security 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 Evaluation Kit.
4. Connect the Host PC USB port to the SmartFusion2 Security Evaluation Kit board's **J18** USB connector using the USB mini-B cable.

Figure 14 shows the board setup for running the SmartFusion2 MDDR demo on the SmartFusion2 Security Evaluation Kit.

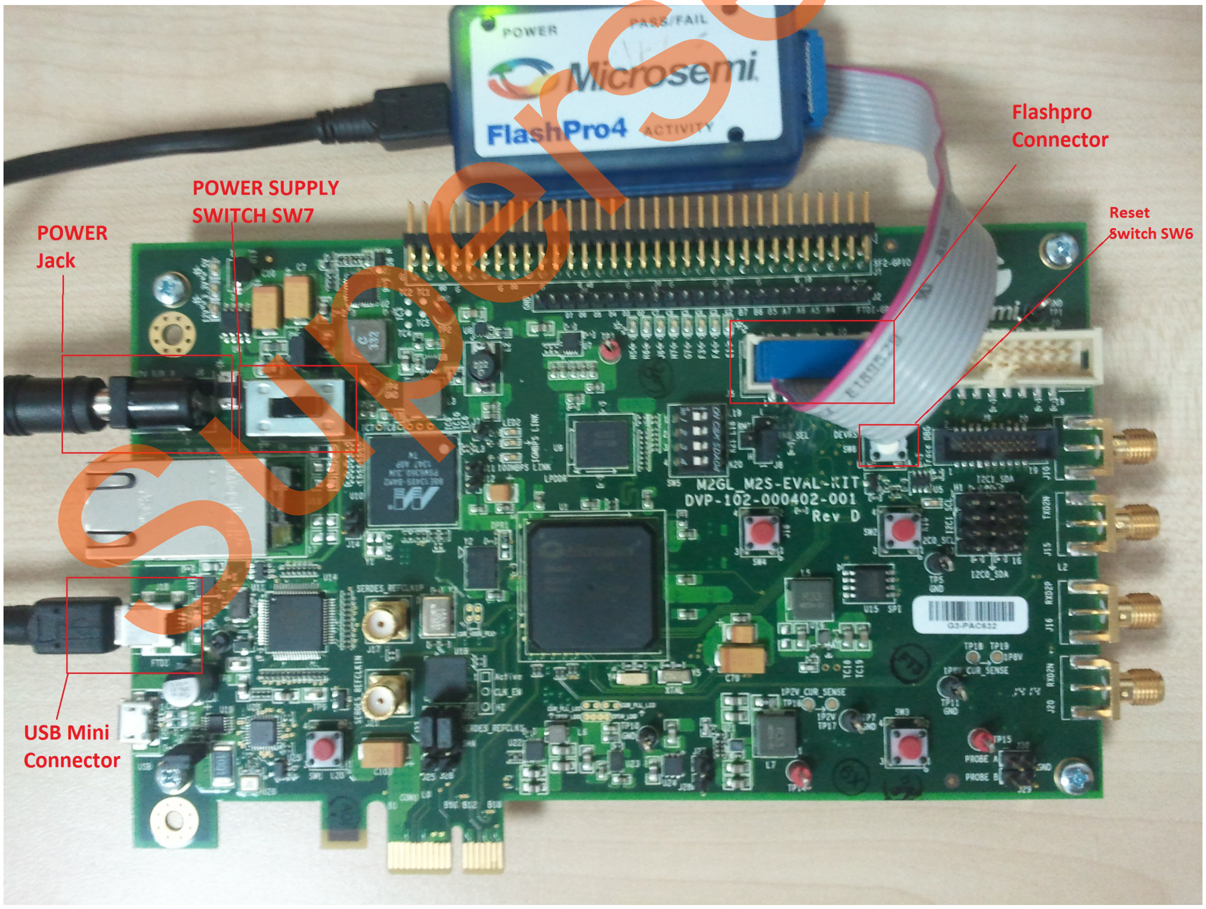
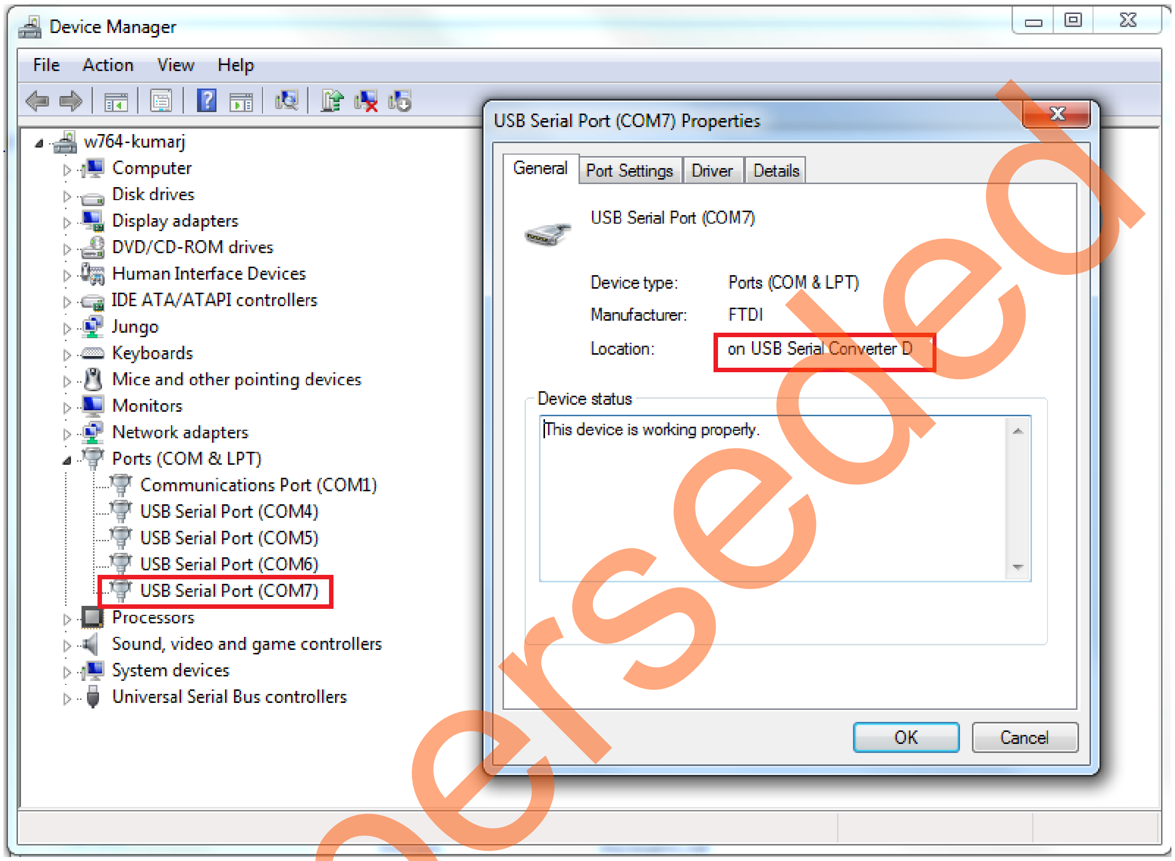


Figure 14 • **SmartFusion2 Security Evaluation Kit**

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. Figure 15 shows the USB 2.0 Serial port properties. As shown in Figure 15, COM7 is connected to USB Serial Converter D. Refer to "Appendix B: Finding Correct COM Port Number when Using the USB 3.0" on page 32 for finding the correct COM port in USB 3.0.



USB Serial Port (COM7)

Device type: Ports (COM & LPT)

Manufacturer: FTDI

Location: on USB Serial Converter D

Device status

This device is working properly.

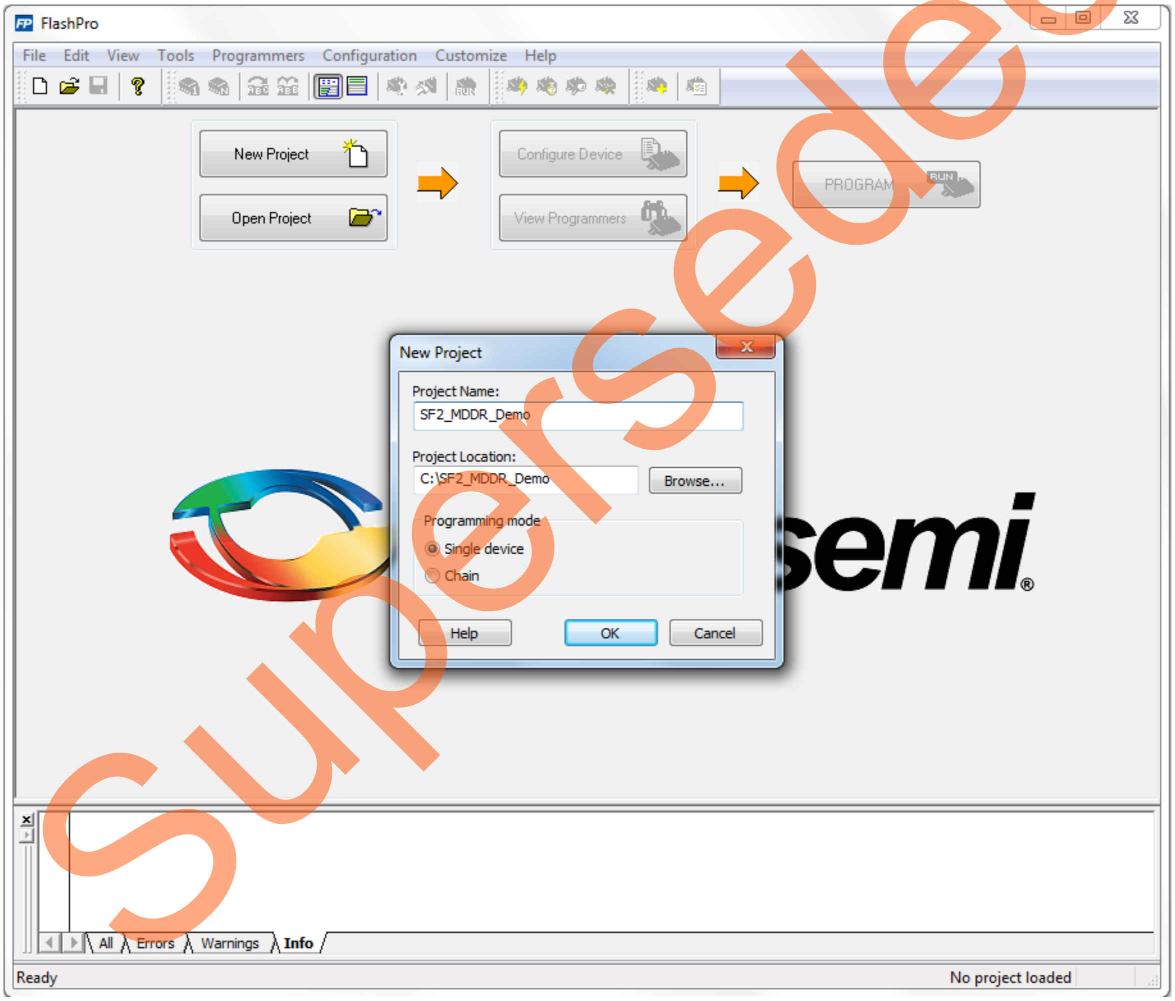
Figure 15 **• USB Serial 2.0 Port Properties**

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*.

Programming the Demo Design

The following steps describe how to program the demo design:

1. Download the demo design from the following link:
   http://soc.microsemi.com/download/rsc/?f=m2s_dg0568_liberov11p6_df
2. Switch **ON** the power supply switch **SW7**.
3. Launch the FlashPro software.
4. Click **New Project**.
5. In the **New Project** window, type the project name as `SF2_MDDR_Demo`.
6. Click **Browse** and navigate to the location where you want to save the project.
7. Select **Single device** as the **Programming mode**.
8. Click **OK** to save the project.



Project Name: SF2_MDDR_Demo

Project Location: C:\SF2_MDDR_Demo

Programming mode
● Single device
○ Chain

Ready

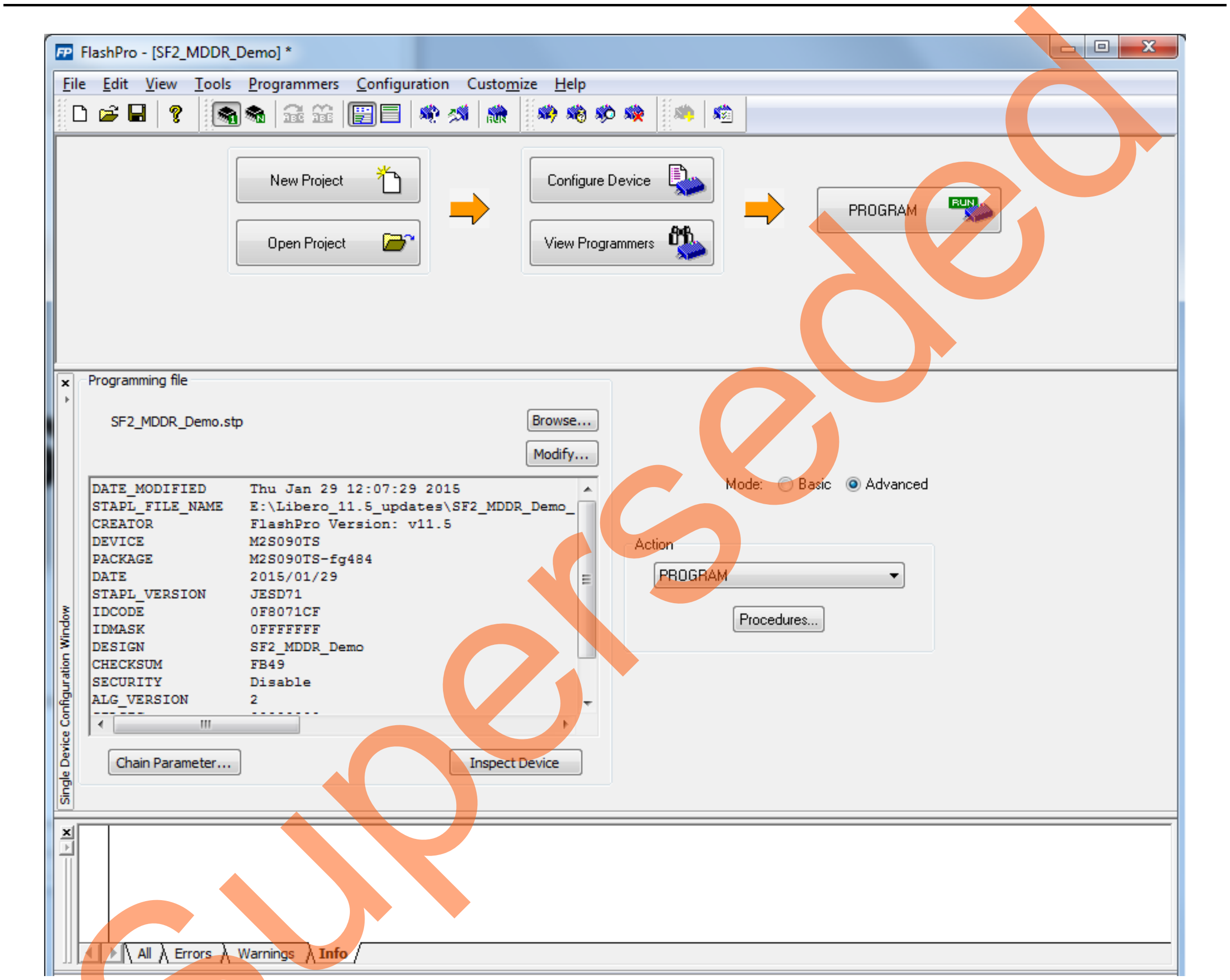
No project loaded

Figure 16 **• FlashPro New Project**

Setting Up the Device

The following steps describe how to configure the device:

1. Click **Configure Device** on the FlashPro GUI.
2. Click **Browse** and navigate to the location where the `SF2_MDDR_Demo.stp` file is located and select the file. The default location is: *`<download_folder>\SF2_MDDR_Demo_DF\Programming_file\`*.
3. Click **Open**. The required programming file is selected and is ready to be programmed in the device.



Programming file

SF2_MDDR_Demo.stp

```
DATE_MODIFIED      Thu Jan 29 12:07:29 2015
STAPL_FILE_NAME    E:\Libero_11.5_updates\SF2_MDDR_Demo_
CREATOR            FlashPro Version: v11.5
DEVICE             M2S090TS
PACKAGE            M2S090TS-fg484
DATE               2015/01/29
STAPL_VERSION      JESD71
IDCODE             0F8071CF
IDMASK             0FFFFFFF
DESIGN             SF2_MDDR_Demo
CHECKSUM           FB49
SECURITY           Disable
ALG_VERSION        2
```

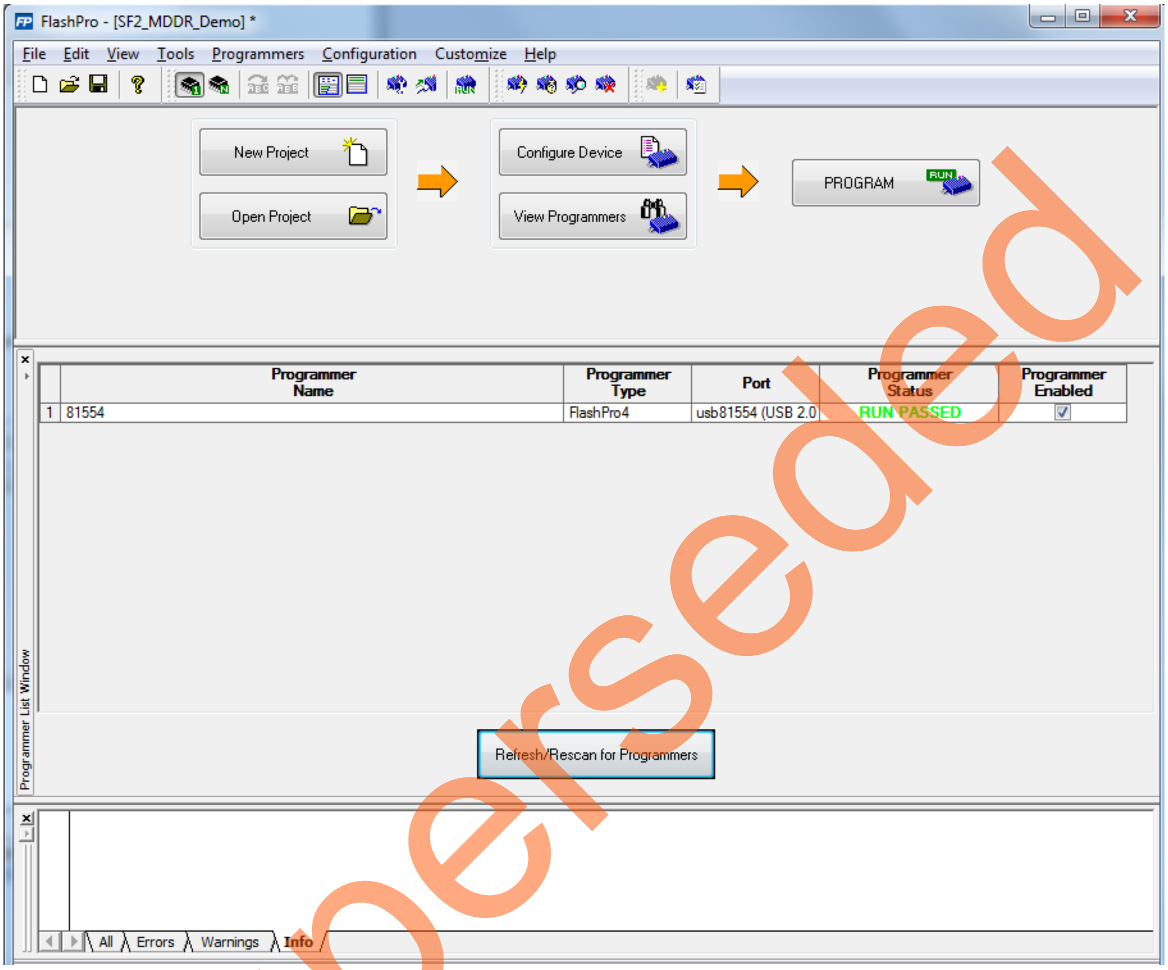
Mode: ○ Basic ● Advanced

Action: PROGRAM

Figure 17 • FlashPro Project Configuration

Programming the Device

Click **PROGRAM** to start programming the device. Wait until Programmer Status is changed to **RUN PASSED**.

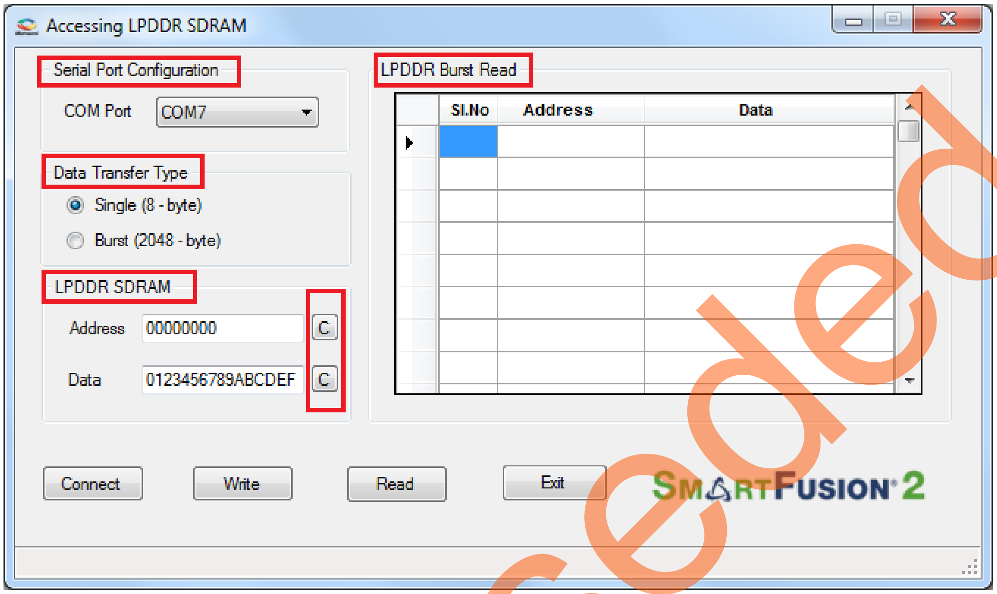


	Programmer Name	Programmer Type	Port	Programmer Status	Programmer Enabled
1	81554	FlashPro4	usb81554 (USB 2.0	RUN PASSED	☑

Figure 18 • **FlashPro Program Passed**

Running the Hardware Demo

The SmartFusion2 MDDR demo comes with utility, `SF2_MDDR_Demo` that runs on the Host PC to communicate with the SmartFusion2 Security Evaluation Kit board. The UART protocol is used as the underlying communication protocol between the Host PC and the SmartFusion2 Security Evaluation Kit board. Figure 19 shows the initial screen of the `SF2_MDDR_Demo` utility.



COM Port: COM7

● Single (8 - byte)

○ Burst (2048 - byte)

Address: 00000000

Data: 0123456789ABCDEF

***Figure 19* • SF2_MDDR_Demo Utility**

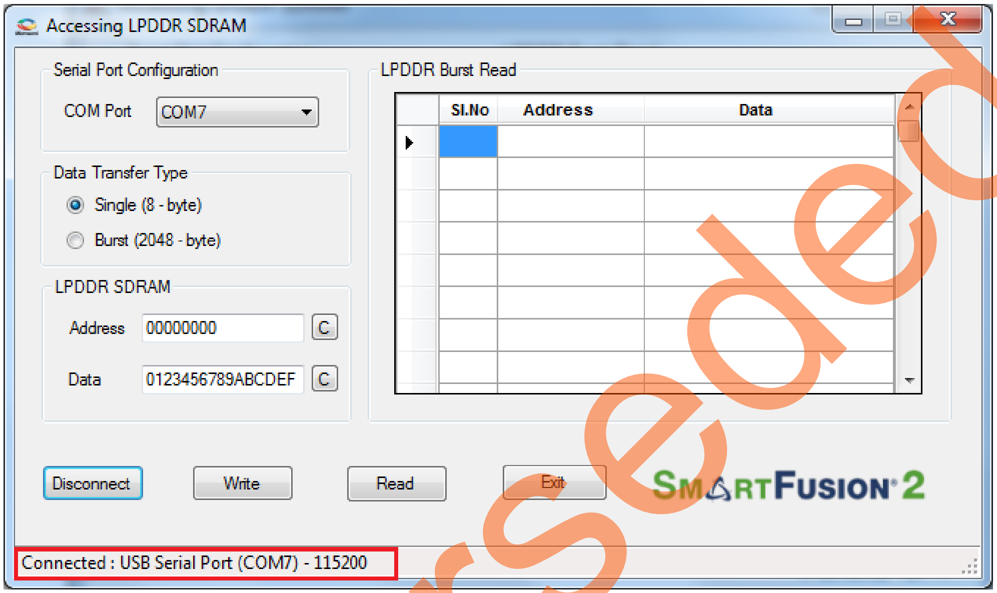
The SF2_MDDR_Demo utility has the following sections:

- **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.

Steps to Run GUI

The following steps describe how to run the GUI:

1. Launch the utility. The default location is:
   `<download_folder>\\SF2_MDDR_Demo_DF\Demo_Utility\SF2_MDDR_Demo.exe.`
2. Select the appropriate COM port from drop down menu. In this case, it is COM 7.
3. Click **Connect**. The connection status along with the COM Port and Baud rate is shown in the left bottom corner of the screen. Figure 20 shows the connection status of the utility.



Accessing LPDDR SDRAM

Serial Port Configuration
COM Port: COM7

Data Transfer Type
● Single (8 - byte)
○ Burst (2048 - byte)

LPDDR SDRAM
Address: 00000000
Data: 0123456789ABCDEF

LPDDR Burst Read

Sl.No	Address	Data

Connected : USB Serial Port (COM7) - 115200

Figure 20 **• SF2_MDDR_Demo – Connection Status**

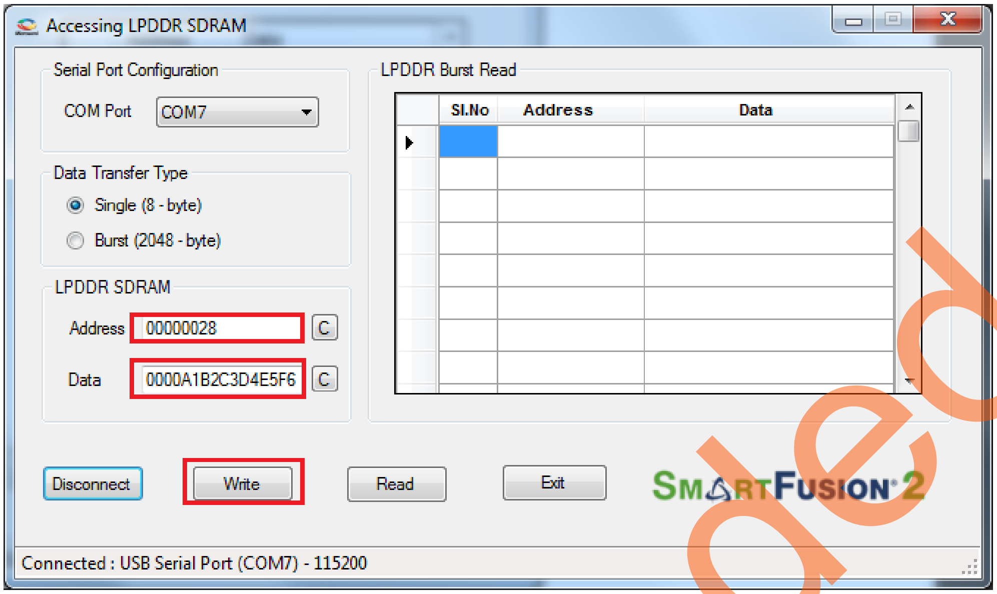
Performing a Single Data Transfer

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 the **Data Transfer Type as Single (8 bytes)**.
2. A 64-bit aligned address is required in the address field. Enter a 32-bit HEX Address in the range 0x00000000 - 0x03FFFFF8. When a non 64-bit aligned address is provided, the GUI converts it to 64-bit aligned address and performs the write or read. Refer to "Appendix C: Performing Write/Read Operation when Non 64-Bit Aligned Address is Provided" on page 34 to perform write or read when non 64-bit aligned address is provided.
3. In the **Data** field, enter a 64-bit data in HEX format.
4. Click **Write**. The entered data is written to the LPDDR memory.

Figure 21 shows the **Address** and **Data** values entered for a Single Write operation.



Accessing LPDDR SDRAM

Serial Port Configuration

COM Port: COM7

Data Transfer Type

● Single (8 - byte)

○ Burst (2048 - byte)

LPDDR SDRAM

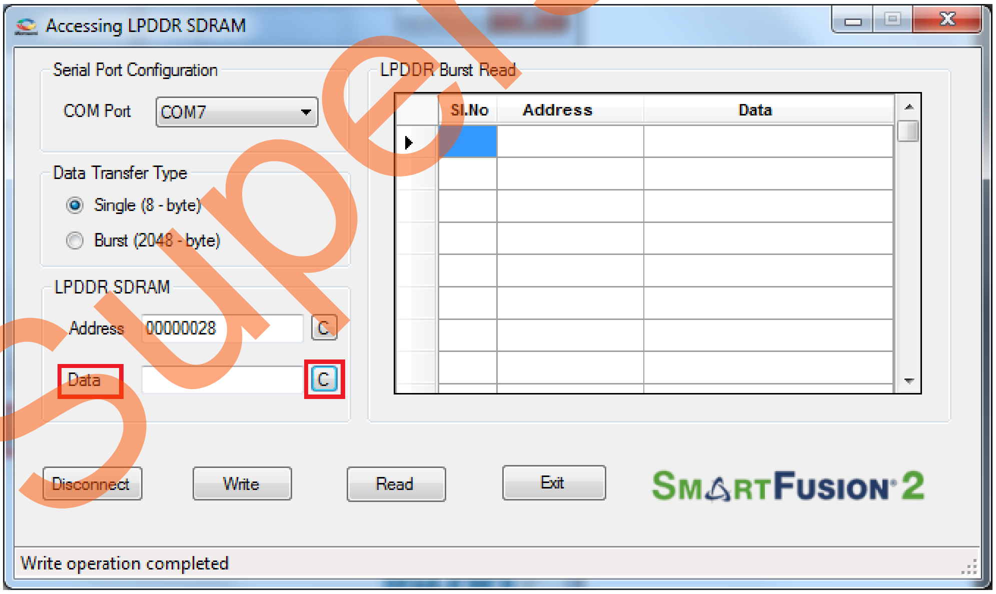
Address: 00000028

Data: 0000A1B2C3D4E5F6

Connected : USB Serial Port (COM7) - 115200

Figure 21 **• Single Write Operation**

5. To verify the write operation, perform a read operation to the same address where the data was written.
6. Press **C** to clear the data present in the **Data** field. Figure 22 highlights the Clear button, C.



Accessing LPDDR SDRAM

Serial Port Configuration

COM Port: COM7

Data Transfer Type

● Single (8 - byte)

○ Burst (2048 - byte)

LPDDR SDRAM

Address: 00000028

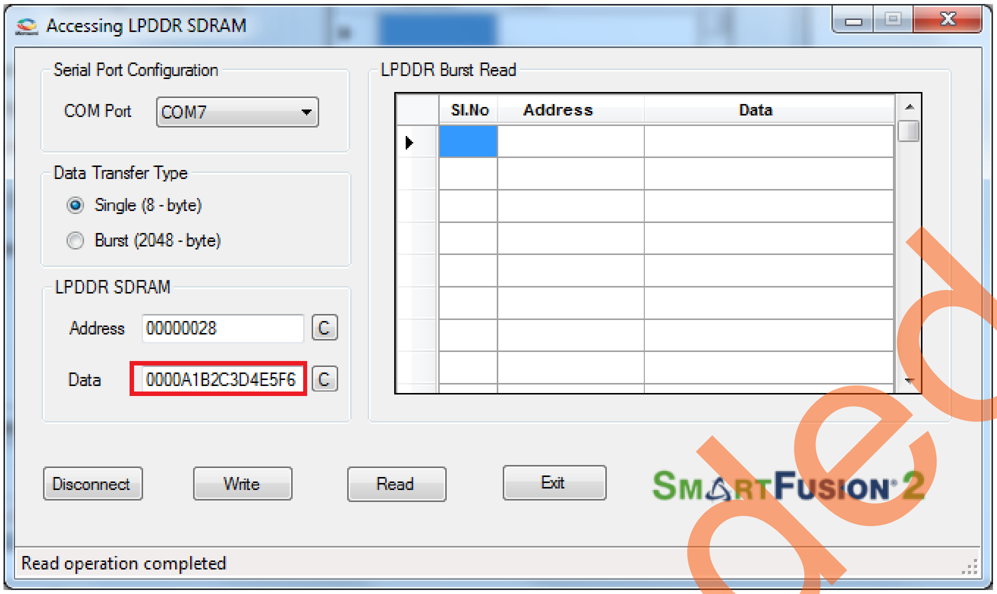
Data:

Write operation completed

Figure 22 **• Clear Data Field**

7. Click **Read** to read the data from the LPDDR SDRAM.

Figure 23 shows the data read from the LPDDR SDRAM.



COM Port: COM7

● Single (8 - byte)
○ Burst (2048 - byte)

Address: 00000028

Data: 0000A1B2C3D4E5F6

Read operation completed

Figure 23 **• Single Read Operation**

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

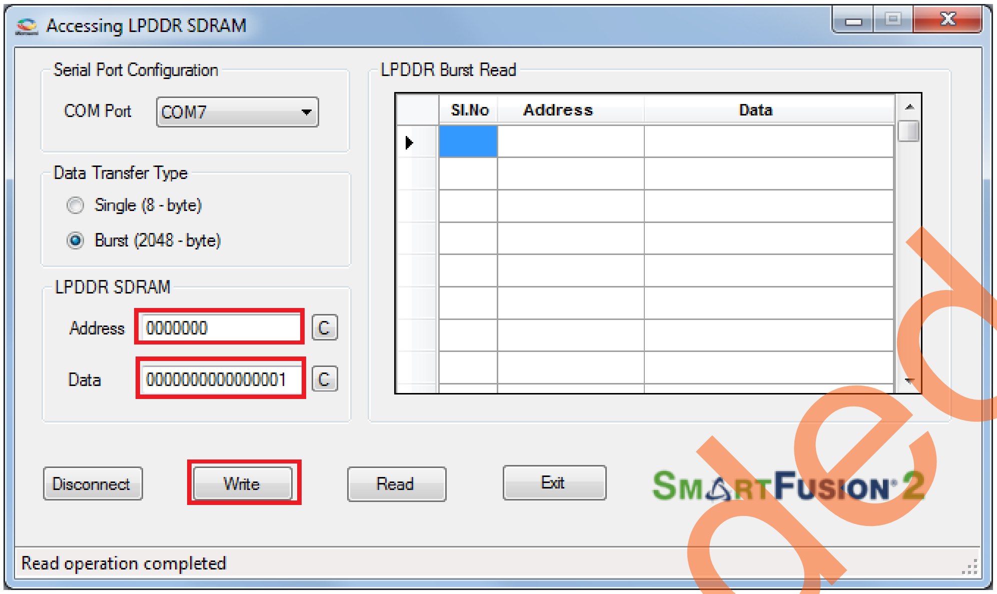
Performing Burst Data Transfer

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 is implemented, that is, 16 (transfers) x 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 the **Data Transfer Type** as **Burst (2048 bytes)**.
2. A 64-bit aligned address is required in the address field. Enter a 32-bit HEX Address in the range 0x00000000 - 0x03FFF7F8. When a non 64-bit aligned address is provided, the GUI converts it into 64-bit aligned address and performs the write or read operation. Refer to "Appendix C: Performing Write/Read Operation when Non 64-Bit Aligned Address is Provided" on page 34 to perform write or read when non 64-bit aligned address is provided.
3. In the **Data** field, enter a 64-bit data in HEX format.
4. Click **Write**. The entered data is written to the Address location specified in the Address filed 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.

Figure 24 shows the **Address** and **Data** values entered for a Burst Write operation.



Serial Port Configuration

COM Port: COM7

Data Transfer Type

○ Single (8 - byte)
● Burst (2048 - byte)

LPDDR SDRAM

Address: 0000000 [C]
Data: 0000000000000001 [C]

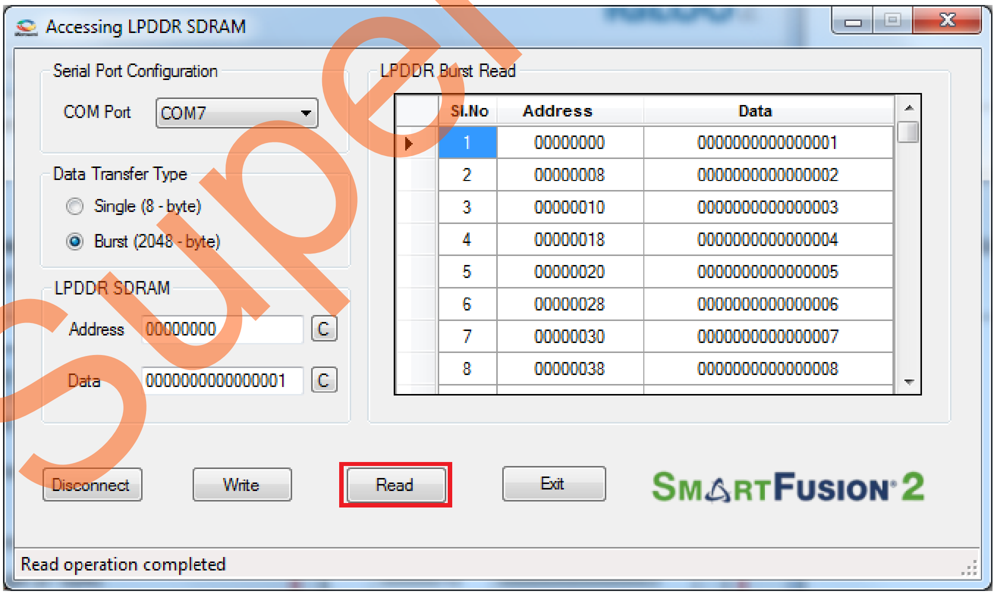
LPDDR Burst Read

Sl.No	Address	Data

Read operation completed

Figure 24 **• Burst Write Operation**

5. To verify the write operation, perform a read operation to the same address where the data was written.
6. Click **Read**. All the 2048 bytes of data that was written to the LPDDR was read and the read data was displayed in the **LPDDR Burst Read** panel. Figure 25 shows the burst read data.



Serial Port Configuration

COM Port: COM7

Data Transfer Type

○ Single (8 - byte)
● Burst (2048 - byte)

LPDDR SDRAM

Address: 00000000 [C]
Data: 0000000000000001 [C]

LPDDR Burst Read

Sl.No	Address	Data
1	00000000	0000000000000001
2	00000008	0000000000000002
3	00000010	0000000000000003
4	00000018	0000000000000004
5	00000020	0000000000000005
6	00000028	0000000000000006
7	00000030	0000000000000007
8	00000038	0000000000000008

Read operation completed

Figure 25 **• Burst Read Operation**

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

Conclusion

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

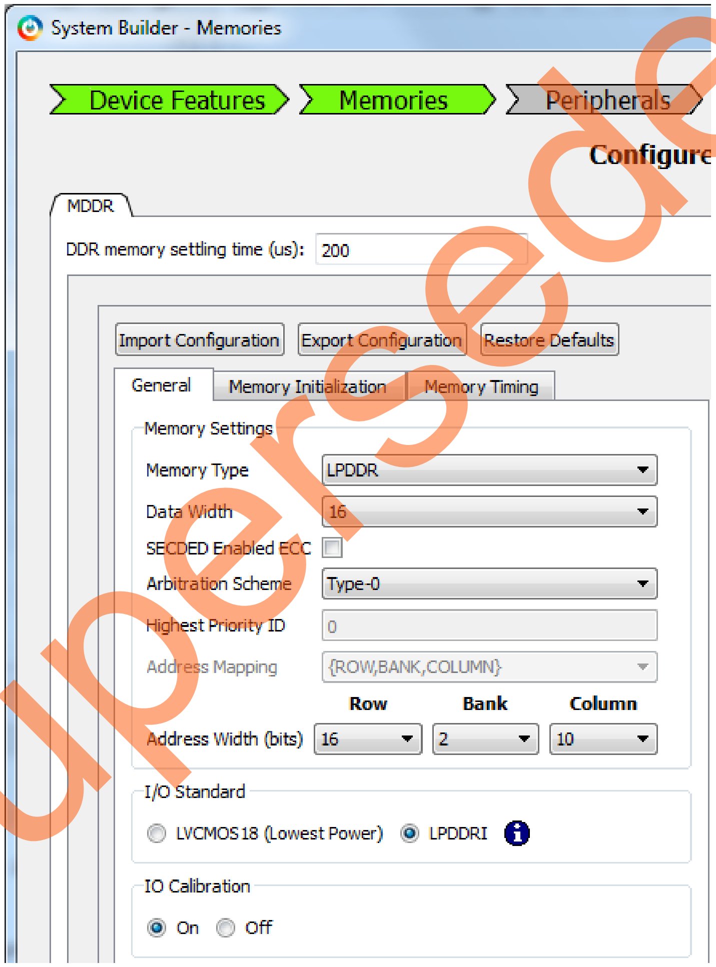
Appendix A: Configuring MDDR Controller

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 System Builder.

The SmartFusion2 Security 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 for download from Micron website.

Figure 26 shows the **MDDR** tab.



DDR memory settling time (us): 200

Memory Type: LPDDR

Data Width: 16

SECDED Enabled ECC □

Arbitration Scheme: Type-0

Highest Priority ID: 0

Address Mapping: {ROW,BANK,COLUMN}

	Row	Bank	Column
Address Width (bits)	16	2	10

I/O Standard

○ LVCMOS18 (Lowest Power) ● LPDDRI

IO Calibration

● On ○ Off

Figure 26 **• System Builder - Memories - MDDR Tab**

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 Evaluation Kit uses the LPDDR memory. Therefore, the DDR controller has to wait at least 200 us. Provide 200 as the value for the field, **DDR memory settling time (us)**.

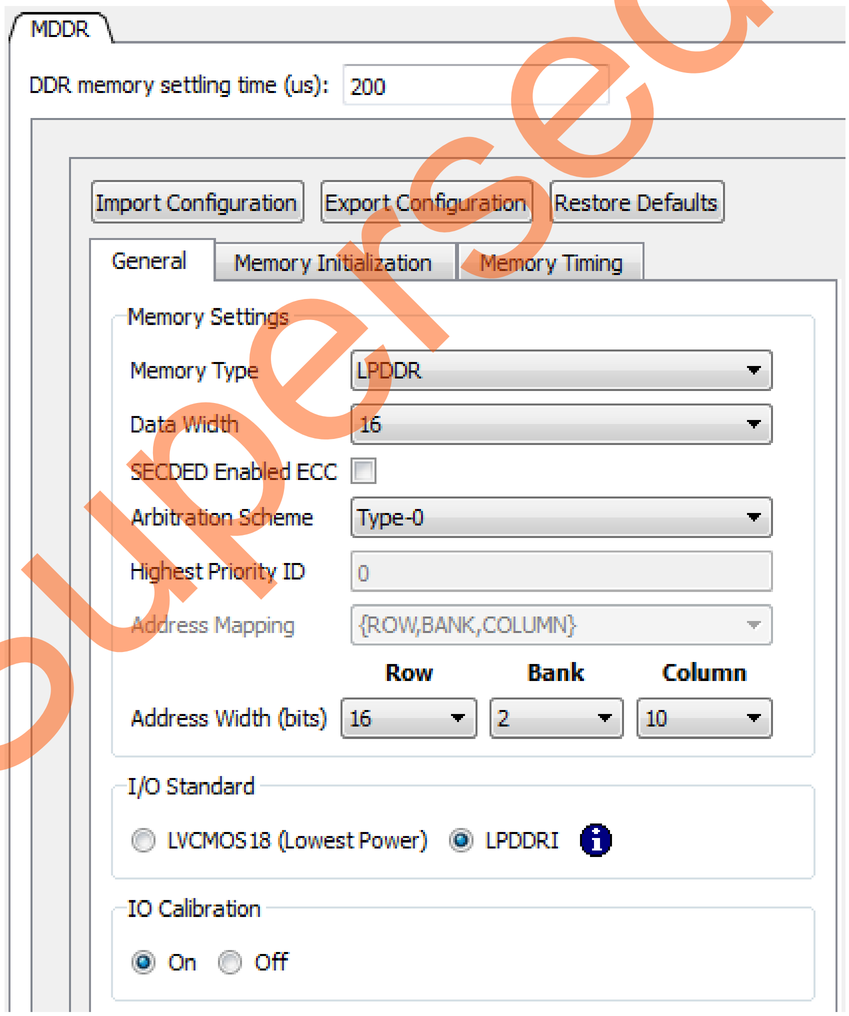
Note: All the values provided here are from the Micron datasheet. The parameters can be configured according to the user requirements.

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 27 shows the **General** tab after configuration parameters are set.



DDR memory settling time (us): 200

Memory Type: LPDDR

Data Width: 16

SECDED Enabled ECC □

Arbitration Scheme: Type-0

Highest Priority ID: 0

Address Mapping: {ROW,BANK,COLUMN}

	Row	Bank	Column
Address Width (bits)	16	2	10

I/O Standard: ○ LVCMOS18 (Lowest Power) ● LPDDRI

IO Calibration: ● On ○ Off

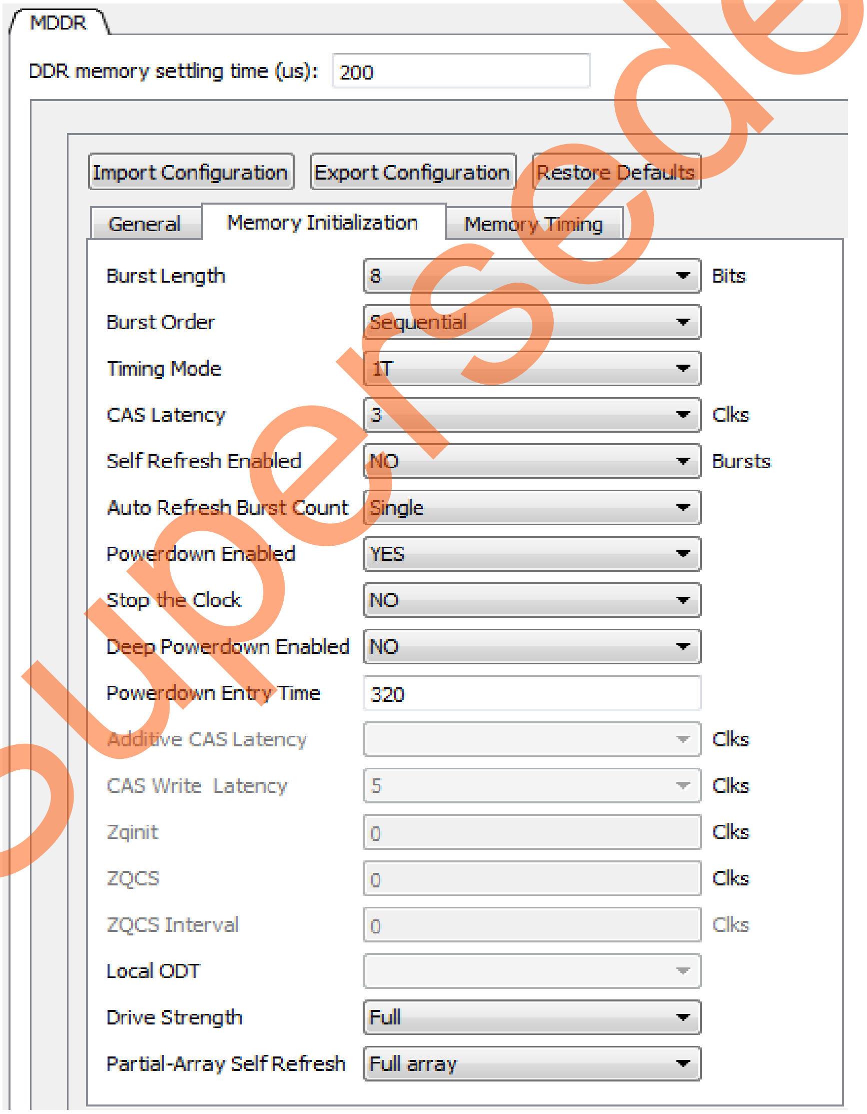
Figure 27 • **System Builder MDDR Configuration – General Tab**

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
- **Power Down Enabled**: YES
- **Stop the clock:** NO
- **Deep Power Down enabled**: NO
- **No Activity clocks for Entry**: 320

Figure 28 shows the **Memory Initialization** tab after configuration parameters are set.



DDR memory settling time (us): 200

Burst Length	8	Bits
Burst Order	Sequential	
Timing Mode	1T	
CAS Latency	3	Clks
Self Refresh Enabled	NO	Bursts
Auto Refresh Burst Count	Single	
Powerdown Enabled	YES	
Stop the Clock	NO	
Deep Powerdown Enabled	NO	
Powerdown Entry Time	320	
Additive CAS Latency		Clks
CAS Write Latency	5	Clks
Zqinit	0	Clks
ZQCS	0	Clks
ZQCS Interval	0	Clks
Local ODT		
Drive Strength	Full	
Partial-Array Self Refresh	Full array	

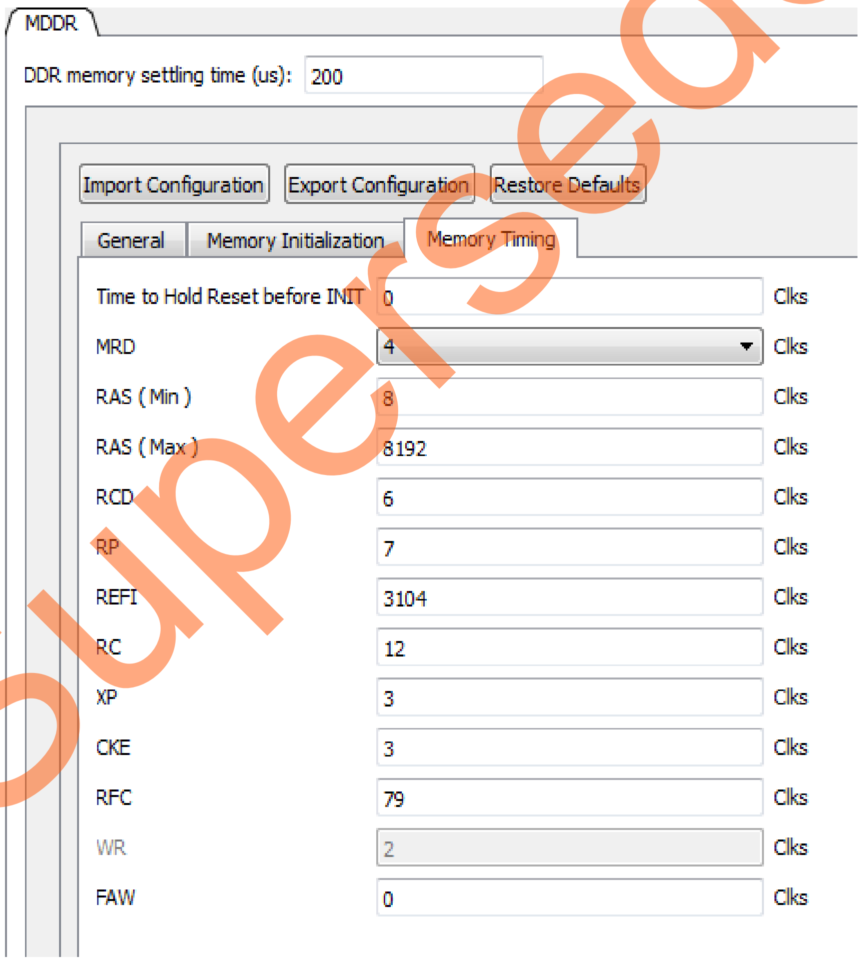
***Figure 28* • System Builder MDDR Configuration – Memory Initialization Tab**

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



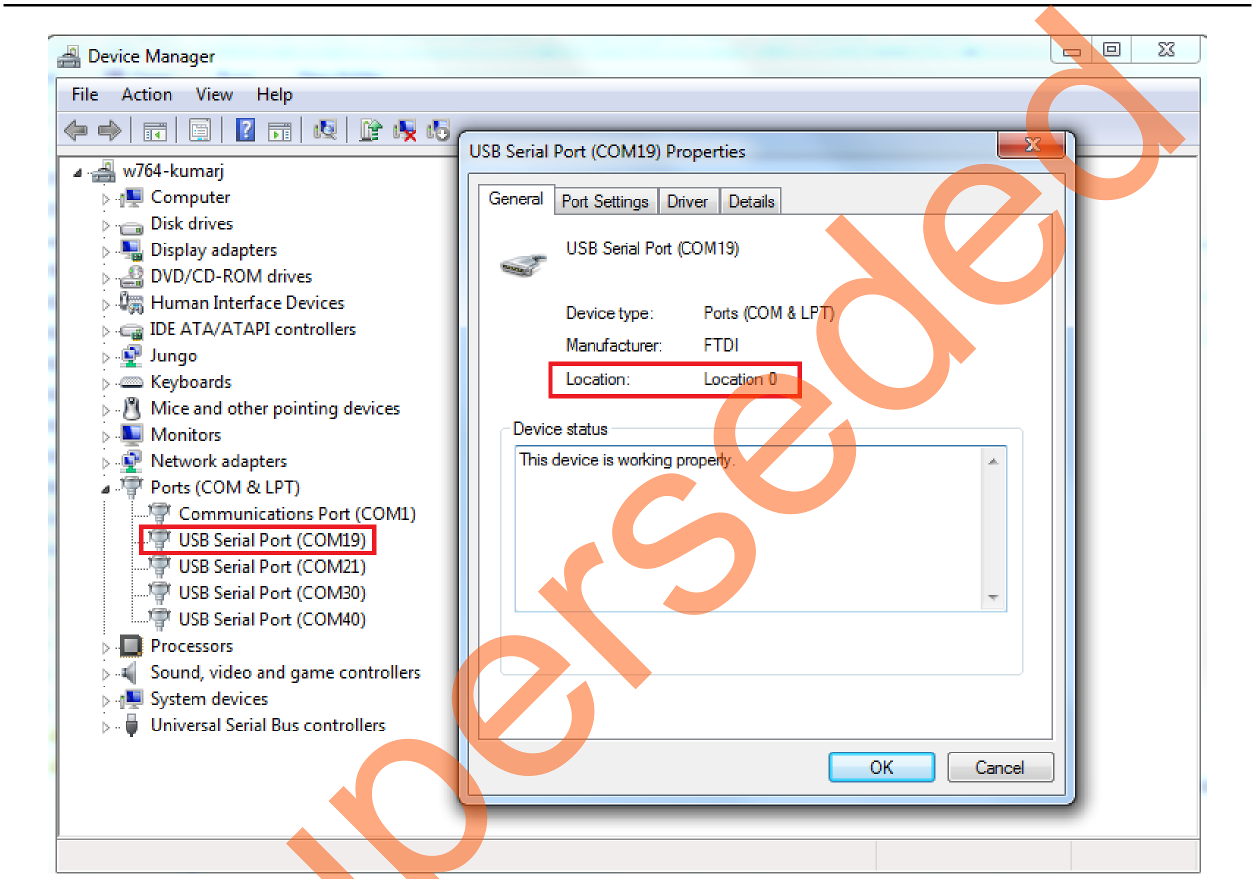
DDR memory settling time (us): 200

Time to Hold Reset before INIT	0	Clks
MRD	4	Clks
RAS (Min)	8	Clks
RAS (Max)	8192	Clks
RCD	6	Clks
RP	7	Clks
REFI	3104	Clks
RC	12	Clks
XP	3	Clks
CKE	3	Clks
RFC	79	Clks
WR	2	Clks
FAW	0	Clks

Figure 29 **• System Builder MDDR Configuration – Memory Timing Tab**

Appendix B: Finding Correct COM Port Number when Using the 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. Figure 30 shows the USB 3.0 Serial port properties.



USB Serial Port (COM19)

Device type: Ports (COM & LPT)

Manufacturer: FTDI

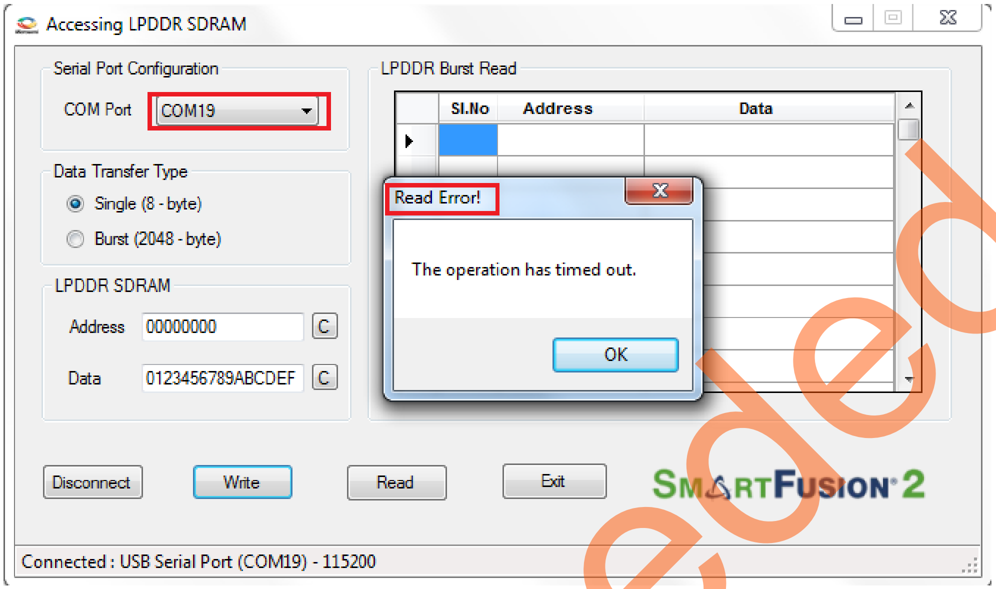
Location: Location 0

Device status

This device is working properly.

Figure 30 • **USB 3.0 Serial Port Properties**

To find out the correct COM port, program the SmartFusion2 Security Evaluation Kit board with provided programming file. Connect each available COM port and click **Write**. If wrong COM port is selected, the GUI displays the read error. Try with all four available COM ports until this message disappears. Figure 31 shows the read error message.



COM Port: COM19

● Single (8 - byte)

○ Burst (2048 - byte)

Address: 00000000 C

Data: 0123456789ABCDEF C

Read Error!

The operation has timed out.

Connected : USB Serial Port (COM19) - 115200

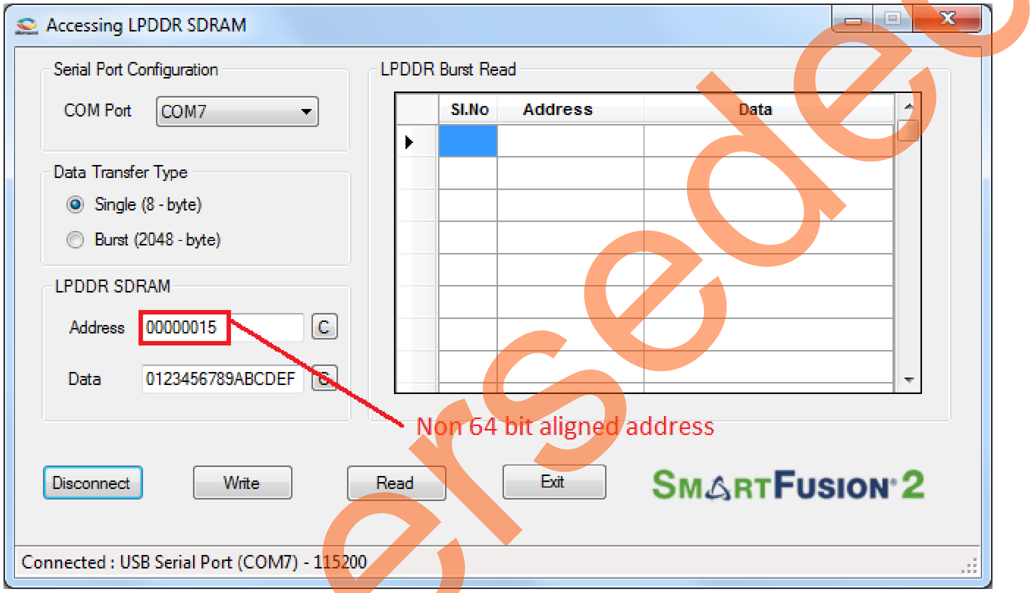
Figure 31 **• Read Error**

Appendix C: 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 (0, 8, 10, 18, 20, 28, 30, 38…) 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.

Figure 32 shows the non 64-bit aligned address entered in the GUI.



COM Port: COM7

● Single (8 - byte)
○ Burst (2048 - byte)

Address: 00000015

Data: 0123456789ABCDEF

Sl.No	Address	Data

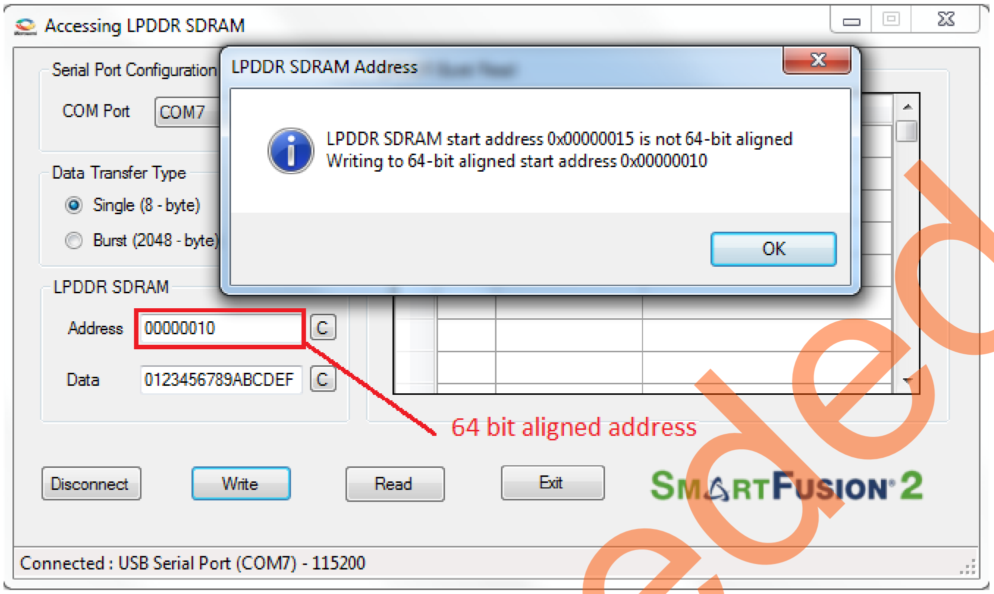
Non 64 bit aligned address

Connected : USB Serial Port (COM7) - 115200

Figure 32 • **Non 64-bit Aligned Address**

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

Figure 33 shows the GUI pop-up information message and converted 64-bit aligned address.



Accessing LPDDR SDRAM

Serial Port Configuration

COM Port: COM7

Data Transfer Type

● Single (8 - byte)

○ Burst (2048 - byte)

LPDDR SDRAM

Address: 00000010

Data: 0123456789ABCDEF

LPDDR SDRAM Address

LPDDR SDRAM start address 0x00000015 is not 64-bit aligned
Writing to 64-bit aligned start address 0x00000010

64 bit aligned address

Connected : USB Serial Port (COM7) - 115200

Figure 33 **• Converted 64-bit Aligned Address**

List of Changes

The following table shows important changes made in this document for each revision.

Date	Changes	Page
Revision 4 (November 2015)	Changed AXI : MDDR ratio as 1:1 and updated Figure 3, Figure 4, Figure 5, and Figure 6 (SAR 73230).	1-8, 1-9, 1-10, and 1-11
Revision 3 (October 2015)	Updated the document for Libero v11.6 software release (SAR 71692).	NA
Revision 2 (February 2015)	Updated the document for Libero v11.5 software release (SAR 64895).	NA
Revision 1 (August 2014)	Initial release	NA

Product Support

Microsemi SoC Products Group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, electronic mail, and worldwide sales offices. This appendix contains information about contacting Microsemi SoC Products Group and using these support services.

Customer Service

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

From North America, call 800.262.1060
From the rest of the world, call 650.318.4460
Fax, from anywhere in the world, 408.643.6913

Customer Technical Support Center

Microsemi SoC Products Group staffs its Customer Technical Support Center with highly skilled engineers who can help answer your hardware, software, and design questions about Microsemi SoC Products. The Customer Technical Support Center spends a great deal of time creating application notes, answers to common design cycle questions, documentation of known issues, and various FAQs. So, before you contact us, please visit our online resources. It is very likely we have already answered your questions.

Technical Support

For Microsemi SoC Products Support, visit

http://www.microsemi.com/products/fpga-soc/design-support/fpga-soc-support.

Website

You can browse a variety of technical and non-technical information on the SoC home page, at http://www.microsemi.com/products/fpga-soc/fpga-and-soc.

Contacting the Customer Technical Support Center

Highly skilled engineers staff the Technical Support Center. The Technical Support Center can be contacted by email or through the Microsemi SoC Products Group website.

Email

You can communicate your technical questions to our email address and receive answers back by email, fax, or phone. Also, if you have design problems, you can email your design files to receive assistance. We constantly monitor the email account throughout the day. When sending your request to us, please be sure to include your full name, company name, and your contact information for efficient processing of your request.

The technical support email address is soc_tech@microsemi.com.

My Cases

Microsemi SoC Products Group customers may submit and track technical cases online by going to My Cases.

Outside the U.S.

Customers needing assistance outside the US time zones can either contact technical support via email (soc_tech@microsemi.com) or contact a local sales office. Visit About Us for sales office listings and corporate contacts.

ITAR Technical Support

For technical support on RH and RT FPGAs that are regulated by International Traffic in Arms Regulations (ITAR), contact us via soc_tech@microsemi.com. Alternatively, within My Cases, select Yes in the ITAR drop-down list. For a complete list of ITAR-regulated Microsemi FPGAs, visit the ITAR web page.

Microsemi

Microsemi Corporate 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

E-mail: sales.support@microsemi.com

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for communications, defense & security, aerospace 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; security technologies and scalable anti-tamper products; Ethernet Solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif., and has approximately 3,600 employees globally. Learn more at **www.microsemi.com**.

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.