DG0625
Demo Guide
Interfacing RTG4 with External DDR3 Memory
Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer’s responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided “as is, where is” and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.

About Microsemi

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

1 Revision History ......................................................... 1
   1.1 Revision 6.0 ....................................................... 1
   1.2 Revision 5.0 ....................................................... 1
   1.3 Revision 4.0 ....................................................... 1
   1.4 Revision 3.0 ....................................................... 1
   1.5 Revision 2.0 ....................................................... 1
   1.6 Revision 1.0 ....................................................... 1

2 Interfacing RTG4 FPGA with External DDR3 Memory Through FDDR ............. 2
   2.1 Design Requirements ............................................. 2
   2.2 Demo Design ..................................................... 2
      2.2.1 Introduction ................................................. 2
   2.3 Demo Design Features .......................................... 5
   2.4 Demo Design Description ....................................... 5
      2.4.1 DDR_AXI_0 .................................................... 6
      2.4.2 UART_IF_0 ................................................... 7
   2.5 Clocking Structure ............................................. 8
   2.6 Reset Structure ................................................ 8
   2.7 Resource Utilization .......................................... 9
   2.8 Demo Design Simulation ...................................... 9
      2.8.1 Simulation Setup ......................................... 13
      2.8.2 Running the Simulation ................................... 13
   2.9 Demo Setup .................................................... 15
   2.10 Programming the Demo Design ................................ 17
   2.11 Running the Demo ............................................. 20
      2.11.1 Steps to Run the GUI .................................... 21
      2.11.2 Performing a Single Data Transfer .................... 22
      2.11.3 Performing Burst Data Transfer ....................... 24
   2.12 Conclusion .................................................... 25

3 Appendix: RTG4 DDR Memory Controller Configuration and Initialization ......... 26
   3.1 Importing DDR Configuration Files ................................ 28
   3.2 FDDR Initialization ............................................ 28
   3.3 DDR Memory Settling Time .................................... 28

4 Appendix: Finding Correct COM Port Number when Using USB 3.0 ............... 30

5 Appendix: Performing Write or Read Operation when Non 64-bit Aligned Address is Provided 32
# Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>Demo Design Files Top-Level Structure</td>
<td>3</td>
</tr>
<tr>
<td>Figure 2</td>
<td>RTG4 DDR Demo Block Diagram</td>
<td>4</td>
</tr>
<tr>
<td>Figure 3</td>
<td>RTG4_DDR_Demo SmartDesign</td>
<td>5</td>
</tr>
<tr>
<td>Figure 4</td>
<td>DDR_AXI_0 SmartDesign Component</td>
<td>6</td>
</tr>
<tr>
<td>Figure 5</td>
<td>UART_IF_0 SmartDesign Component</td>
<td>7</td>
</tr>
<tr>
<td>Figure 6</td>
<td>Clocking Structure</td>
<td>8</td>
</tr>
<tr>
<td>Figure 7</td>
<td>Reset Structure</td>
<td>8</td>
</tr>
<tr>
<td>Figure 8</td>
<td>DO File Settings</td>
<td>9</td>
</tr>
<tr>
<td>Figure 9</td>
<td>Waveforms Settings</td>
<td>10</td>
</tr>
<tr>
<td>Figure 10</td>
<td>Invoking Organize Stimulus Files Window</td>
<td>11</td>
</tr>
<tr>
<td>Figure 11</td>
<td>Organize Stimulus Files Window</td>
<td>12</td>
</tr>
<tr>
<td>Figure 12</td>
<td>Transcript Window</td>
<td>13</td>
</tr>
<tr>
<td>Figure 13</td>
<td>Single AXI Write and Read Operation</td>
<td>14</td>
</tr>
<tr>
<td>Figure 14</td>
<td>16-Beat AXI Burst Write and Read</td>
<td>14</td>
</tr>
<tr>
<td>Figure 15</td>
<td>RTG4 Development Kit</td>
<td>16</td>
</tr>
<tr>
<td>Figure 16</td>
<td>USB Serial 2.0 Port Properties</td>
<td>17</td>
</tr>
<tr>
<td>Figure 17</td>
<td>FlashPro New Project</td>
<td>18</td>
</tr>
<tr>
<td>Figure 18</td>
<td>FlashPro Project Configuration</td>
<td>18</td>
</tr>
<tr>
<td>Figure 19</td>
<td>FlashPro Program Passed</td>
<td>19</td>
</tr>
<tr>
<td>Figure 20</td>
<td>RTG4_DDR_Demo_Utility</td>
<td>20</td>
</tr>
<tr>
<td>Figure 21</td>
<td>RTG4_DDR_Demo - Connection Status</td>
<td>21</td>
</tr>
<tr>
<td>Figure 22</td>
<td>Single Write Operation</td>
<td>22</td>
</tr>
<tr>
<td>Figure 23</td>
<td>Clear Data Field</td>
<td>23</td>
</tr>
<tr>
<td>Figure 24</td>
<td>Single Read Operation</td>
<td>23</td>
</tr>
<tr>
<td>Figure 25</td>
<td>Burst Write Operation</td>
<td>24</td>
</tr>
<tr>
<td>Figure 26</td>
<td>Burst Read Operation</td>
<td>25</td>
</tr>
<tr>
<td>Figure 27</td>
<td>RTG4 FDDR Configuration Window</td>
<td>26</td>
</tr>
<tr>
<td>Figure 28</td>
<td>FDDR Configurator - Memory Initialization tab</td>
<td>27</td>
</tr>
<tr>
<td>Figure 29</td>
<td>FDDR Configurator - Memory Timing tab</td>
<td>28</td>
</tr>
<tr>
<td>Figure 30</td>
<td>USB 3.0 Serial Port Properties</td>
<td>30</td>
</tr>
<tr>
<td>Figure 31</td>
<td>Read Error</td>
<td>31</td>
</tr>
<tr>
<td>Figure 32</td>
<td>Non 64-bit Aligned Address</td>
<td>32</td>
</tr>
<tr>
<td>Figure 33</td>
<td>Converted 64-bit Aligned Address</td>
<td>33</td>
</tr>
</tbody>
</table>
Tables

Table 1  Design Requirements ................................................................. 2
Table 2  Resource Utilization ................................................................. 9
Table 3  RTG4 Development Kit Jumper Settings ....................................... 15
## Revision History

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

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1.1      | **Revision 6.0**  
Updated the document for Libero SoC v11.9. |
| 1.2      | **Revision 5.0**  
Updated the document for Libero v11.8 SP2. |
| 1.3      | **Revision 4.0**  
Updated the document for Libero v11.7 SP1. |
| 1.4      | **Revision 3.0**  
Updated the document for Libero v11.7. |
| 1.5      | **Revision 2.0**  
Updated the document for Libero v11.6. |
| 1.6      | **Revision 1.0**  
The first publication of this document. |
2 Interfacing RTG4 FPGA with External DDR3 Memory Through FDDR

This demo shows how the FPGA fabric logic can access external double-data rate three (DDR3) memories using built-in fabric double-data rate (FDDR) in RTG4 devices.

The demo has two parts:

- Simulation
- Running the demo on the RTG4 Development Kit

In the demo design, the advanced extensible interface (AXI) master in the FPGA fabric accesses the DDR memory present in the RTG4 Development Kit using the FDDR. A host utility, RTG4_DDR_Demo_Utility is provided along with the demo design files. Using the utility, the AXI master logic is driven. The AXI master converts the commands from the utility to AXI transactions for the FDDR to perform the read or write operations on the external DDR3 memory.

2.1 Design Requirements

Table 1, page 2 shows the design requirements.

<table>
<thead>
<tr>
<th>Design Requirements</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hardware Requirements</strong></td>
<td></td>
</tr>
<tr>
<td>RTG4 FPGA Development Kit:</td>
<td>RT4G150-CB1657PROTO FPGA</td>
</tr>
<tr>
<td>– USB 2.0 cable</td>
<td></td>
</tr>
<tr>
<td>– FlashPro4 programmer</td>
<td></td>
</tr>
<tr>
<td>– 12 V, 5A AC power adapter and cords</td>
<td></td>
</tr>
<tr>
<td>Host PC or Laptop</td>
<td>Any 64-bit Windows Operating System</td>
</tr>
<tr>
<td><strong>Software Requirements</strong></td>
<td></td>
</tr>
<tr>
<td>Libero® System-on-Chip (SoC)</td>
<td>v11.9 SP1</td>
</tr>
<tr>
<td>FlashPro programming software</td>
<td>v11.9 SP1</td>
</tr>
<tr>
<td>Host PC Drivers</td>
<td>USB to UART drivers</td>
</tr>
</tbody>
</table>

2.2 Demo Design

2.2.1 Introduction

The demo design files are available for download from the following path in the Microsemi website: http://soc.microsemi.com/download/rsc/?f=rt4g_dg0625_liberov11p9sp1_df

The demo design files include:

- Demo_Utility
- Libero_project
- RTG4_DDR_Demo
- Programming_file
- Source_files
- readme.txt

Figure 1, page 3 shows the top-level structure of the design files. For further details, see the readme.txt file.
**Figure 1** • Demo Design Files Top-Level Structure

```
rt4g_dg0625_liberov11p9spl_df
   RTG4_DDR_Demo_DF
      Demo_Utility
      Libero_project
      Programming_file
      Source_file
      readme.txt
```

**Figure 1**, page 3 shows the top-level view of demo design. In the demo design, the AXI master implemented in the FPGA fabric accesses the DDR3 memory present on the RTG4 Development Kit using the FDDR. The AXI master logic communicates to the FDDR through CoreAXI interface and the DDR_FIC interface. The read or write operations initiated by the RTG4_DDR_Demo_Utility are sent to the UART_IF block using the UART protocol. The 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. The AXI master reads the data from DDR3 memory and stores it in TPSRAM. When the read operation is complete, the data read is sent to the host PC through UART.
In this demo design, different blocks are configured, as shown below:

- FDDR is configured for DDR3 memory available on the RTG4 Development Kit. The DDR3 memory is a Micron DRAM (Part Number: MT41K256M8DA-125 IT:K).
- DDR_FIC is configured for AXI bus interface.
- AXI clock is configured for 80 MHz and DDR3 clock is configured for 320 MHz.
- CoreUART IP has the following configuration:
  - Baud Rate: 115200
  - Data Bits: 8
  - Parity: None
- RTG4TPSRAM has the following configuration:
  - Write port depth: 256
  - Write port width: 64
  - Read port depth: 1024
  - Read port width: 16

For more information about the RTG4TPSRAM, see *RTG4 FPGA Two-Port Large SRAM Configuration User's Guide*.

Refer to Appendix: RTG4 DDR Memory Controller Configuration and Initialization, page 26 for information on how to configure the FDDR.
2.3 Demo Design Features

The RTG4 DDR demo design has the following features:

- Single AXI read or write transactions
- 16-beat burst AXI read or write transactions
- DDR3 memory model simulation using testbench
- Design validation using the RTG4 Development Kit that has the DDR3 memory
- Initiation of the read or write transactions using RTG4_DDR_Demo_Utility

2.4 Demo Design Description

The demo design consists of the following SmartDesign components:

- DDR_AXI_0: Handles the data transactions between the FDDR and the DDR3 SDRAM.
- UART_IF_0: Handles the communication between the host PC and the RTG4 Development Kit.

Figure 3, page 5 shows the DDR AXI_0 and UART IF_0 connection.

Figure 3 • RTG4_DDR_Demo SmartDesign
2.4.1 **DDR_AXI_0**

DDR_AXI_0 consists of the FDDR subsystem 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 DDR3 memory through the FDDR. Figure 4, page 6 shows the DDR_AXI_0 SmartDesign component.

*Figure 4 • DDR_AXI_0 SmartDesign Component*
### 2.4.2 UART_IF_0

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 the COREUART_0, collects the data from the COREUART_0 IP and converts the data to the relevant AXI_IF_0 master signals. For more information about CoreUART, see CoreUART Handbook.

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 is 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 receives the read data from the AXI_IF_0 master logic and stores it in the RTG4TPSRAM_0. After completion of the read burst transactions, the UART_IF_FSM_0 wrapper fetches the stored data from the RTG4TPSRAM_0 and sends it to the COREUART IP. Figure 5, page 7 shows the UART_IF_0 SmartDesign component. For more information about RTG4TPSRAM, see RTG4 FPGA Two-Port Large SRAM Configuration User's Guide from Libero.

*Figure 5 • UART_IF_0 SmartDesign Component*
2.5 Clocking Structure

The on-chip 50 MHz oscillator provides the reference clock to the FCCC block. The FCCC block provides two clocks, GL0 (80 MHz) and GL1 (50 MHz). The GL0 clock drives the AXI_IF_0, UART_IF_0, and FDDRC_With_INIT_0 blocks. The GL1 clock drives the CoreABC processor inside the FDDRC_With_INIT_0 block. The following figure shows the clocking structure of the design.

For more information about FCCC, see UG0590: RTG4 FPGA Clock Conditioning Circuit with PLL Configuration User Guide from Libero.

2.6 Reset Structure

The POWER_ON_RESET and the LOCK signal are ANDed and the output signal (INIT_RESET_N) is used to reset the FDDRC_With_INIT_0 block. After the reset of FDDRC_With_INIT_0, INIT_DONE signal is used to reset the AXI_IF_0 and UART_IF_0 blocks. The following figure shows the reset structure of the design.
2.7 **Resource Utilization**

The following table lists the resource utilization of the design after place and route. These values may vary slightly for different Libero runs, settings, and seed values. For IP-wise utilization, see the respective handbooks.

<table>
<thead>
<tr>
<th>Type</th>
<th>Used</th>
<th>Total</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>4LUT</td>
<td>1120</td>
<td>151824</td>
<td>0.74</td>
</tr>
<tr>
<td>DFF</td>
<td>863</td>
<td>151824</td>
<td>0.57</td>
</tr>
<tr>
<td>I/O Register</td>
<td>0</td>
<td>2151</td>
<td>0.00</td>
</tr>
<tr>
<td>Logic Element</td>
<td>1343</td>
<td>151824</td>
<td>0.88</td>
</tr>
</tbody>
</table>

2.8 **Demo Design Simulation**

The demo design can be simulated using testbench and Micron DDR3 memory model.

The simulation is set to run the following:

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

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

- ddr3.v
- ddr3_parameters.vh
- testbench.v

The default location of the files is:
<Download folder>\RTG4_DDR_Demo_DF\Libero_project\RTG4_DDR_Demo\stimulus

2.8.1 **Simulation Setup**

The following steps describe how to setup the simulation:

1. Launch the Libero SoC software.
2. Browse the RTG4_DDR_Demo project provided in the design file.
3. Choose **Project > Project Settings > Simulation Options**.
4. Ensure that the DO File tab has the configuration, as shown in Figure 8, page 9.

![DO File Settings](image-url)
5. Ensure that the **Waveforms** tab has the configuration, as shown in Figure 9, page 10.

**Figure 9 • Waveforms Settings**
6. Go to **Design Flow** tab.
7. Right-click **Simulate** under **Verify Pre-Synthesized Design** and select **Organize Input Files > Organize Stimulus Files...**, as shown in **Figure 10**, page 11.

**Figure 10** • Invoking Organize Stimulus Files Window
8. Ensure that the **Organize Stimulus files** window has the configuration, as shown in Figure 11, page 12.

*Figure 11 • Organize Stimulus Files Window*
2.8.2 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.

Simulation run time is 700 µs, as shown in Figure 8 on page 9. Figure 12, page 13 shows the transcript window of the simulation.

Figure 12 • Transcript Window
Figure 13, page 14 shows the single AXI write and AXI read operation as per AXI specification.

**Figure 13 • Single AXI Write and Read Operation**

These waveforms show the AXI4 single and burst transfers initiated by AXI4 master. These AXI4 transactions are input to the FDDR, which converts them to read/write operations across external DDR3 memory.
2.9 Demo Setup

This demo design uses the RTG4 Development Kit. Figure 15, page 16 shows the RTG4 Development Kit. The following steps describe how to setup the hardware demo:

1. Connect the jumpers on the RTG4 Development Kit, as shown in Table 3, page 15.

<table>
<thead>
<tr>
<th>Jumper</th>
<th>Pin (from)</th>
<th>Pin (to)</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>J32</td>
<td>1</td>
<td>2</td>
<td>Default</td>
</tr>
<tr>
<td>J27</td>
<td>1</td>
<td>2</td>
<td>Default</td>
</tr>
<tr>
<td>J26</td>
<td>1</td>
<td>2</td>
<td>Default</td>
</tr>
<tr>
<td>J23</td>
<td>1</td>
<td>2</td>
<td>Default</td>
</tr>
<tr>
<td>J21</td>
<td>1</td>
<td>2</td>
<td>Default</td>
</tr>
<tr>
<td>J19</td>
<td>1</td>
<td>2</td>
<td>Default</td>
</tr>
<tr>
<td>J11</td>
<td>1</td>
<td>2</td>
<td>Default</td>
</tr>
<tr>
<td>J16</td>
<td>2</td>
<td>3</td>
<td>Default</td>
</tr>
</tbody>
</table>

CAUTION: Ensure that the power supply switch SW6 is switched OFF while connecting the jumpers.

2. Connect the power supply to the J9 connector, switch ON the power supply switch, SW6.
3. Connect the host PC USB port to the RTG4 Development Kit J47 USB connector using the USB A to mini-B cable.
4. Ensure that the USB to UART bridge drivers are automatically detected. This can be verified in the Device Manager of the host PC. The future technology devices international (FTDI) USB to UART converter enumerates four COM ports. For USB 2.0, note down the USB Serial Converter C COM port number to use it in the GUI. Figure 16, page 17 shows the USB 2.0 serial port properties. As shown in Figure 16, page 17, COM6 is connected to the USB serial converter C. Refer to "Appendix: Finding Correct COM Port Number when Using USB 3.0" on page 30 for finding the correct COM port in USB 3.0.
5. 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.

### 2.10 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=rt4g_dg0625_liberov11p9sp1_df
2. Switch ON the power supply switch SW6.
3. Launch the FlashPro software.
4. Click New Project.
5. In the New Project window, type the project name as RTG4_DDRDemo.
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.
9. Click **Configure Device** on the FlashPro GUI.
10. Click **Browse** and navigate to the location where `DDR_demo_top.stp` file is located, and select the file. The default location is: `<download_folder>\RTG4_DDR_Demo_DF\Programming_file`.
11. Click **Open**. The required programming file is selected and is ready to be programmed in the device.
12. Click **PROGRAM** to start programming the device. Wait until the Programmer Status is changed to **RUN PASSED**.

*Figure 19 • FlashPro Program Passed*
2.11 Running the Demo

The RTG4 DDR demo comes with utility, RTG4_DDR_Demo_Utility that runs on the host PC to communicate with the RTG4 Development Kit. The UART protocol is used as the underlying communication protocol between the host PC and the RTG4 Development Kit. Figure 20, page 20 shows initial screen of the RTG4_DDR_Demo_Utility.

Figure 20 • RTG4_DDR_Demo_Utility

The RTG4_DDR_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.
- **DDR SDRAM**: Provides Address and Data.
- **DDR Burst Read**: Displays the Burst Read values for the corresponding address.
- **C**: Clears the existing data.
2.11.1 Steps to Run the GUI

The following steps describe how to run the GUI:

1. Launch the utility. The default location is:
   `<download_folder>\RTG4_DDR_Demo_DF\Demo_Utility\RTG4_DDR.exe`.
2. Select the appropriate COM port from drop down menu. In this case, it is COM 28.
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 21, page 21 shows the connection status of the utility.

*Figure 21 • RTG4_DDR_Demo - Connection Status*
2.11.2 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 DDR3 SDRAM. For a read operation, the utility sends a 32-bit address and receives 64-bit data from DDR3 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 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. See "Appendix: Performing Write or Read Operation when Non 64-bit Aligned Address is Provided" on page 32 to perform write or read operation 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 DDR3 memory. **Figure 22**, page 22 shows the **Address** and **Data** values entered for a Single Write operation.

**Figure 22** • Single Write Operation
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. Figure 23, page 23 highlights the Clear button, C.

**Figure 23 • Clear Data Field**

7. Click Read to read the data from the DDR3 SDRAM. Figure 24, page 23 shows the data read from the DDR3 SDRAM.

**Figure 24 • 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 DDR3 SDRAM are successful.
### 2.11.3 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-bit burst operations are implemented (16 transfers x 16-bit 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 DDR3 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 0x00000000 - 0x03FFFFF8. 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. See "Appendix: Performing Write or Read Operation when Non 64-bit Aligned Address is Provided" on page 32 to perform write or read operation 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 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. Figure 25, page 24 shows the Address and Data values entered for a Burst Write operation.

*Figure 25 • Burst Write Operation*
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 DDR3 SDRAM is read, and the read data is displayed on the **DDR Burst Read** panel. **Figure 26**, page 25 shows the burst read data.

**Figure 26 • Burst Read Operation**

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

### 2.12 Conclusion

This demo shows how to perform read or write operations to DDR3 SDRAM using RTG4 FDDR. Options are provided to simulate the design using a testbench and validate the design on the RTG4 Development Kit using a GUI interface.
This section describes how to configure and initialize the RTG4 DDR memory controller.

Launch RTG4 DDR Memory Controller with Initialization configurator using the RTG4 DDR Memory Controller with initialization SgCore in Libero. Use the RTG4 DDR Memory Controller Configurator to configure the FDDR, select its datapath bus interface (AXI or AMBA® high-performance bus (AHB)), and select the DDR clock frequency as well as the fabric datapath clock frequency.

1. In the RTG4 DDR Memory Controller with Initialization configurator, under General tab, configure the settings as shown in Figure 1, page 26.

**Figure 1 • RTG4 FDDR Configuration Window**
2. Set the register values for the FDDR registers to match the external DDR memory characteristics. The FDDR has a set of registers that must be configured at runtime. The configuration values for these registers represent different parameters. Use the **Memory Initialization** tab, as shown in Figure 2, page 27 and **Memory Timing** tab, as shown in Figure 3, page 28 to enter parameters that correspond to the DDR Memory and application. Consult the DDR Memory vendor’s datasheet for values to enter in **Memory Initialization** and **Memory Timing** tabs. Values entered in these tabs are automatically translated to the appropriate register values. When a specific parameter is clicked, a brief description of the corresponding register is described in the register description window of the **Fabric External Memory FDDR Configurator**. For more details on FDDR configuration registers, see the **UG0573: RTG4 FPGA High Speed DDR Interfaces User Guide**.

**Figure 2 • FDDR Configurator - Memory Initialization tab**
3. Instantiate the FDDR as part of a user application and make the datapath connections.

### 3.1 Importing DDR Configuration Files

In addition to entering DDR Memory parameters using the Memory Initialization and Timing tabs, the DDR register values can be imported from a file. Click **Import Configuration** and navigate to the text file with the DDR register names and values. The DDR3 configuration file for the Micron memory available on the RTG4 Development Kit is provided along with the design files.

**Note:** If the register values are imported and not entered using the GUI, then all the required register values must be specified.

### 3.2 FDDR Initialization

The RTG4 DDR Memory Controller with initialization Sgcore has a built-in initialization state machine. On the assertion or de-assertion of the INIT_RESET_N (Active Low) signal, the FDDR block is initialized with the user configurations. When the configuration phase is complete, the INIT_DONE signal is asserted and the FDDR block is ready for normal operations. The FDDR initialization can start automatically at power up by connecting the INIT_RESET_N (Active Low) input of the FDDR block to the POWER_ON_RESET_N (Active Low) signal of the SYSRESET macro.

The clock used for initialization must be a 50 MHz clock which is connected to the INIT_CLK 50 MHz signal.

### 3.3 DDR Memory Settling Time

The RTG4 DDR memory controller block is hard-coded with a DDR memory settling time of 200 µs, assuming that the clock period of INIT_CLK is 20 ns (frequency 50 MHz). Microsemi recommends that the initialization frequency be kept at 50 MHz.
Refer to the DDR Memory vendor’s datasheet for the correct memory settling time to use. An incorrect memory settling time may result in the failure of the DDR memory initialization, during operation. If a different memory settling time is required for the DDR memory or a different INIT_CLK frequency is chosen by the user, than the recommended 50 MHz, then the program code must be edited in the Program tab of CoreABC to change the load value of the register used to compute the settling time.
Appendix: 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. Figure 1, page 30 shows the USB 3.0 Serial port properties.

Figure 1 • USB 3.0 Serial Port Properties
The following steps describe how to find out the correct COM port:

1. Program the RTG4 Development Kit with provided programming file.
2. Connect each available COM port and click Write.
   If wrong COM port is selected, the GUI displays the Read Error message.
3. Try with all four available COM ports until this message disappears.
   Figure 2, page 31 shows the Read Error message.

**Figure 2 • Read Error**
Appendix: Performing Write or 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 or 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 1, page 32 shows the non 64-bit aligned Address entered in the GUI.

Figure 1 • 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 2, page 33 shows the GUI pop-up information message and converted 64-bit aligned address.

**Figure 2** • Converted 64-bit Aligned Address