Design of a Bridge between AHIR system and DDR3 SDRAM

Swatantara Chakraborty(15307R011)

Indian Institute of Technology,Bombay

Under Prof. Madhav P. Desai

July 4, 2018

Understanding the Memory Controller

We start the discussion by describing the basics of the environment we are working on.The design is applicable to all Xilinx FPGAs, like VC-709, ML-605, KC-705 etc. We shall demonstrate the external memory in ML605 for example.

External Memory in ML605 card

The ML605 card provides a Virtex-6 FPGA (XC6VLX240T - 1FFG1156). A single 512 MB DDR3(Double Data Rate Type 3) Synchronous Dynamic Random Access Memory is provided for user applications. The overall block diagram is shown below:(includes only the DDR3 and the FPGA top)

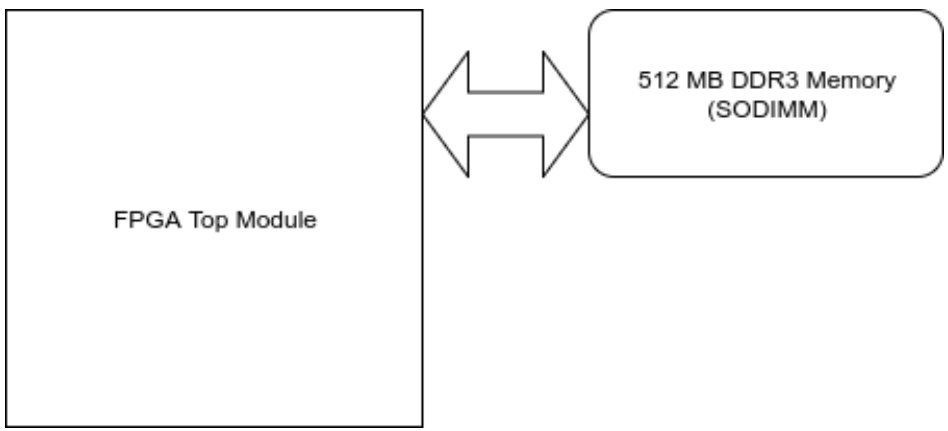

Figure : External Memory in ML605 card

Parameter Specifications for the Current Design

- For a SODIMM DDR3 SDRAM, the data width is fixed at 64 bits.
- The address width is taken as 32 bits. However, 64 and 128 bit addresses may also be selected.
- The burst length is fixed at 8.
- The clock rate can be between 75 MHz and 266 MHz for the user design because the DDR3 SDRAM clock should be double that of the design clock and the DDR3 permissible clock range is between 150 MHZ and 533 MHz.
- The target is ML605 .

MIG Memory Interface Solutions

The following diagram shows the user interface of the memory controller. It has been reproduced from ug406.pdf (Ref 1).

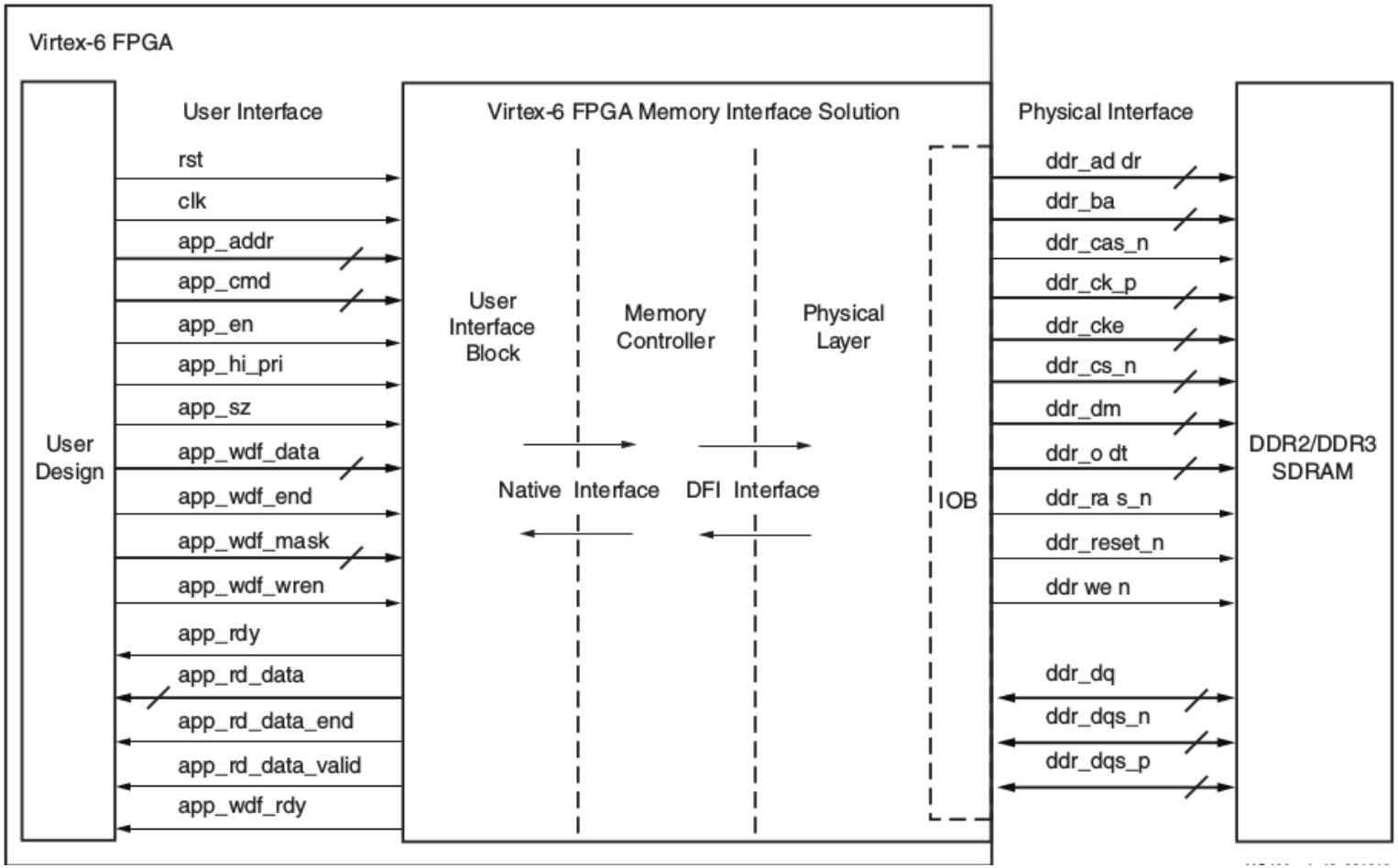

Figure : MIG Virtex 6 Memory Interface Solutions

Steps to Generate the MIG core

- Open the terminal. Make a directory, say, July4_mig. Invoke coregen, coregen&.
- File – New Project – Choose July4_mig – Save.
- Choose Family: Virtex6, Device: xc6vlx240t, Package: ff1156, Speed Grade: -1
- Go to **Generation** option. Make Design Entry Verilog. Click on Ok. Window closes.
- Right Click on Mig Virtex6 & Spartan 6 from the drop-down menu on the left. Select Customize and Generate..
- Click on Next for the next 4 windows. In the 5th window, select Memory type: SODIMMs, Ordering: Strict.
- Click on Next. Make Debug: ON. Go to Next.
- Select New Design for pin layout. Go to Next.

Steps to Generate the MIG Core

- Click on Deselect Banks. Now, select Bank36 : Address/Control, Bank26: Data, Bank25: Data, Bank35: Data,Bank34: System Clock. From the middle column on the top, select Master Bank 25.
- Click on Next for two successive windows. Click on Decline. Click on Next for the next 2 windows.
- Click on Generate.
- Go back to the command prompt.

Steps to Run a Basic Example Design

- Go to ml605_modified_main/mig39/example_design/par and run the ise_flow.sh, i.e., ./ise_flow.sh.

This is an example design that generates a pseudo random bit sequence.(explained in the next slide).The ./ise_flow.sh script will add the ILA modules for chipscope, and synthesize the example top to generate the bit file.

The Basic MIG Example-Design

The example-design of the memory controller interface that communicates with the DDR3 SDRAM.

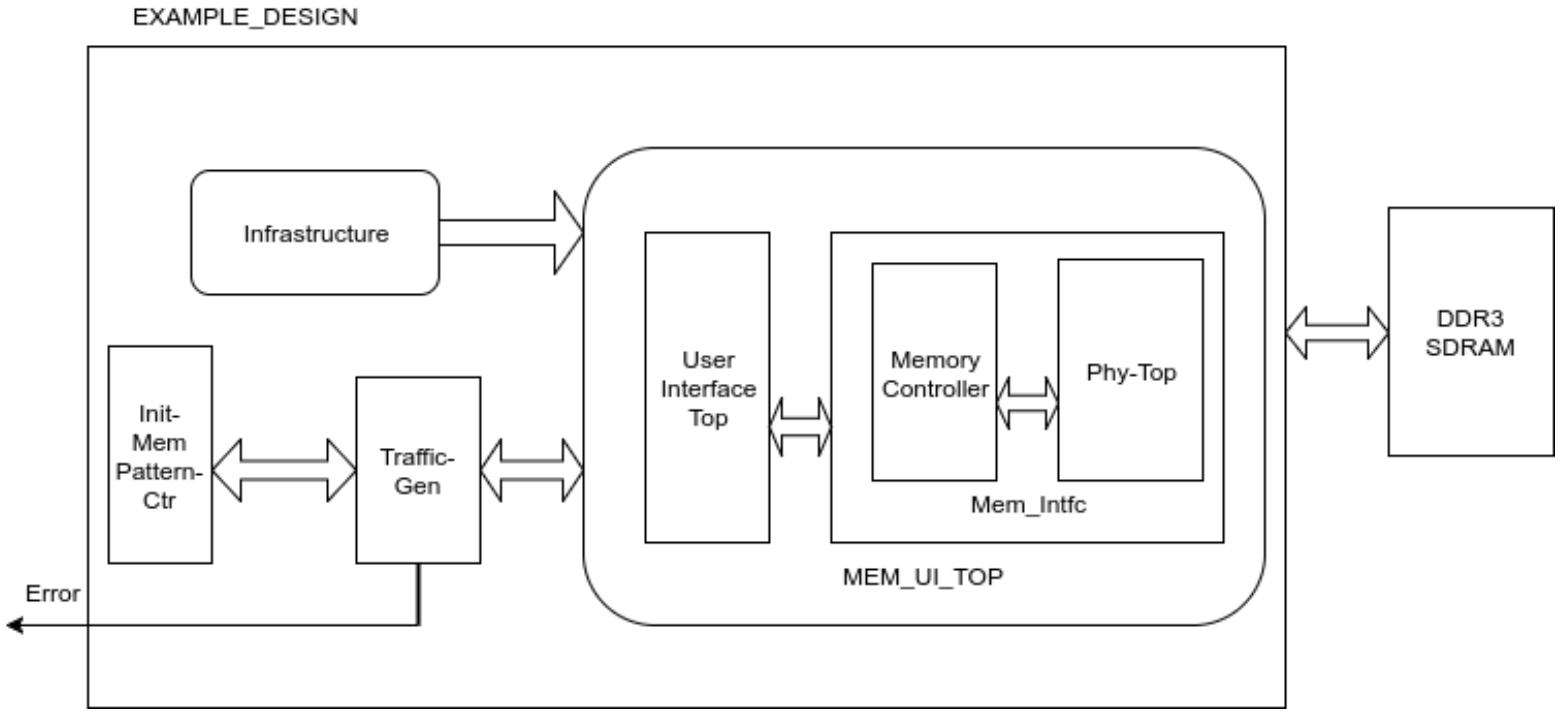

Figure : Block Diagram for the Example Design

Explanation of the MIG Example

- **Infrastructure**: This module helps in clock generation and distribution, and reset synchronization.
- **Mem_intfc** : This is the top-level memory interface block , instantiates the memory controller block and the phy block.
- **Mem_ui_top** : This is the top-level memory interface controller wrapper with the user interface.
- **Phy_Top** : Top-level memory physical layer interface.
- **MC** : Top level memory sequencer structural block.

Steps to Observe the Output of the Example on ML605

- Invoke Chipscope Analyzer, analyzer & at the command prompt.
- Click on Open Cable/Search JTAG Chain button (top left).
- Click on OK on the MYDevice prompt.
- Click on Device from the top bar, Go to Device1 and click on Configure.
- In the new window, click on Select New File. Select July4_mig/ml605_modified_main/mig39/example_design/par/exampl
- After it has finished, click on the T button on the top to trigger the chipscope pro.
- Click on the Apply setting and arm trigger button next but one to the T button.
- Check for the debug waveforms.

Steps to Observe the Output of the Example on ML605

- To adjust the PRBS (pseudo random bit sequence), go to the VIO console on Unit 4 (top left) and click.
- In the list of debug signals, make any one 1 to induce an error. Click on the U button on top to update the change.
- Now again , click on T and arm trigger like before.
- Check dbg_rddata for the PRBS. You may have to zoom into the waveforms.

Running the Compiled Design : Chipscope Analyzer

Xilinx Chipscope Analyzer ILA(Integrated Logic Analyzer) has been invoked to monitor the debug signals. To do so, the example_top.bit file and the example_top.cdc file generated when the ise_flow.sh script was run, are included in a new Chipscope project. The resultant waveforms have been obtained as illustrated in the next few slides.

Running the Compiled Design : Chipscope Analyzer

The ChipScope Pro tool allows the user to set trigger conditions to capture application and MIG signals in hardware. Captured signals can then be analyzed through the ChipScope Pro Logic Analyzer tool.

How to Debug

In order to debug we have to load the provided example_design onto the board in question. This is a known working solution with a traffic generator design that checks for data errors.

Which Signal Waveforms to Check

- **dbg_rdlvl_done** : Each bit is driven to a static 1 as each stage of read leveling is completed. The dbg_rdlvl_done[0] signal corresponds to stage 1. If both the read stages have some error, both dbg_rdlvl_done[0] and dbg_rdlvl_done[1] should be one, that is dbg_rdlvl_done as a bit vector should be shown as 3 in hex format. If both stages complete without any error, that is, the design runs correctly, then it should be 1.
- **dbg_rdlvl_err** : This output indicates if an error occurred during each stage of read leveling. If there is no error, both dbg_rdlvl_err[0] and dbg_rdlvl_err[1] should be 0.
- **dbg_rddata** : This is the capture read data synchronized to the clk clock domain. It should show the psedo-random data generated by the traffic generator.
- **phy_init_done**: If it is 0, the design is error free. If 1, there is some error.

Running the Compiled Design : Chipscope Analyzer

The debug signals to be monitored using Chipscope Analyzer should be included in the example_top.cdc file . Note that the DATA pins of the ILA are not utilised.

Issues Faced

The mig version 39 is directly compatible with the VC709 memory interface solutions. But as we have used mig38 and ML605 certain changes have to be made in the example design rtl. The modified design is provided alongside this document. Changes are made in the rtl files(example top) and the ucf file. Some of the modifications are:

- CLKFBOUT_MULT_F = 6
- OUTPUT_DRV to HIGH
- nDQS_COLx
- DQS_LOC_COLx
- RST_ACT_LOW = 0 (was 1)

Both the unchanged and the modified example design folders are provided along with this document. All the reference documents are also attached.

References

1. Virtex-6 FPGA Memory Interface Solutions User Guide, UG406 June 22, 2011
2. ML605 Reference Design User Guide, UG535 (v1.0) September 25, 2009
3. Virtex-6 FPGA Memory Interface Solutions, DS186 January 18, 2012
4. ChipScope Pro 11.4 Software and Cores, UG029 (v11.4) December 2, 2009
5. https://forums.xilinx.com/t5/Memory-Interfaces/MIG-DDR3-example-design-init-calib-complete-never-goes-high/td-p/721154
6. https://www.xilinx.com/support/answers/34319.html
7. https://www.xilinx.com/products/intellectual-property/mig.htm
8. https://www.xilinx.com/products/intellectual-property/mig.htm
9. www.instructables.com/id/Configuring-the-MIG-7-Series-to-