Overview

- Virtex-6 DDR3 Memory Interface
- ML605 Board
- ML605 Setup
- Generate MIG Example Design
- Modifications to Example Design
- Compile Example Design
- Run MIG Example Design
- Adjust Data Pattern using VIO Console
- Example Design VIO Consoles
- References

Note: This presentation applies to the ML605
Virtex-6 DDR3 Memory Interface

- Pre-Engineered Controller and Physical Layer (PHY) memory interface
- 300-533 MHz (600-1066 Mb/s) Performance
  - Center Column Interfaces
  - 400 MHz in a -1 speed device
  - 533 MHz in -2, -3 devices
- 1 Gb density memory device support
- X4, x8, x16 device support
- Configurable data bus widths
  - Multiples of 8 bits, up to 72 bits

Note: Presentation applies to the ML605
Virtex-6 Memory Controller and Interfaces

- **Improved performance**
  - Higher data rates
    - Faster circuitry (40 nm)
    - Enhanced I/O (50 ps IODELAY)
    - Dedicated clocking paths
    - Real-time calibration
  - Higher effective bandwidth
    - Reordering controller (DDR3/DDR2)

- **Improved functionality**
  - DDR3 DIMM write leveling

- **Easy to use**
  - MIG for ISE design flow
  - MPMC for EDK design flow

*Note: Presentation applies to the ML605*

Xilinx makes it easier and faster to design with Virtex-6
**DDR3 User Interfaces**

- **Virtex-6 FPGA user interface similar to Virtex-5 architecture**
  - Native interface option available for the advanced users to achieve lower latency

---

**User Interface**

- **Addrs/Cmd**
  - **WR_Data/DM**
  - **Data_valid**
  - **Rd_Data**

**Native Interface**

- **Addrs/Cmd**
  - **WR_Data/DM**
  - **Rd_Data**
  - **Rd_data**
  - **Rd_data_en**
  - **Rd_Data_Addrs**

**Reordering Controller**

**Physical Layer**

---

**Note:** Presentation applies to the ML605
Reordering for Higher Effective Bandwidth

- **Half-frequency DDR2/DDR3 controller**
  - Control state machine runs at half the memory clock rate

- **Reorder READs to avoid precharge time penalty**
  - Example: Execute out-of-order READs to a different bank while performing precharge for the current bank

- **Regroup READs and WRITEs to minimize bus turnaround**
  - Example: Read A - Write B - Read C - Write D
  - Reordered to: Read A - Read C - Write B - Write D

- **Reordering controller looks ahead several commands**
  - Efficiency is dependent on applications (address / command patterns)

*Reordering can more than double the throughput*
Xilinx ML605 Board

Note: Presentation applies to the ML605
ISE Software Requirement

- Xilinx ISE 11.4 software

Note: Presentation applies to the ML605
ChipScope Pro Software Requirement

- Xilinx ChipScope Pro 11.4 software

Note: Presentation applies to the ML605
Generate MIG Example Design

- Open the CORE Generator
  Start → All Programs → Xilinx ISE Design Suite 11 →
  ISE → Accessories → CORE Generator
- Create a new project; select File → New Project

Note: Presentation applies to the ML605
Generate MIG Example Design

- Create a project directory: ml605_mig_design
- Name the project: ml605_mig_design.cgp
- Set the Part (as shipped on the ML605):
  - Family: Virtex6
  - Device: xc6vlx240t
  - Package: ff1156
  - Speed Grade: -1

Note: Presentation applies to the ML605
Generate MIG Example Design

- Select Generation
- Set the Design Entry to Verilog
- Click OK

Note: Presentation applies to the ML605
Generate MIG Example Design

- Right click on MIG Version 3.3
  - Select Customize

Note: Presentation applies to the ML605
Generate MIG Example Design

- Leave this page as is
  - Click Next

The Memory Interface Generator (MIG) creates memory controllers for Xilinx FPGAs. MIG creates complete customized Verilog or VHDL RTL source code, pin-out and design constraints for the FPGA selected, and script files for implementation and simulation.

CORE Generator Options

This GUI includes all configurable options along with explanations to aid in generation of the required controller. Please note that some of the options selected in the CORE Generator Project Options will be used in generation of the controller. It is very important that the correct CORE Generator Project Options are selected. These options are listed below.

Selected CORE Generator Project Options:

- FPGA Family: Virtex-6
- FPGA Part: xc5vlx240t-ff1156
- Speed Grade: -1
- Synthesis Tool: XST
- Design Entry: VERILOG

If any of these options are incorrect, please click on "Cancel", change the CORE Generator Project Options, and restart MIG.
Generate MIG Example Design

- Leave this page as is
  - Click Next
Generate MIG Example Design

- Leave this page as is
  - Click Next
Generate MIG Example Design

- **Select Memory Type**
  - DDR3 SDRAM
  - Click Next
Generate MIG Example Design

- **Select**
  - Type: SODIMMs
  - Part: MT4JSF6464HY-1G1
  - Ordering: Strict
Generate MIG Example Design

- Leave this page as is
  - Click Next
Generate MIG Example Design

- **Select**
  - Debug: ON
Generate MIG Example Design

- Leave this page as is
  - Click Next
Generate MIG Example Design

- On this screen select the banks as used on the ML605 SODIMM interface
- Click Deselect Banks
Generate MIG Example Design

- All Banks Deselected
Generate MIG Example Design

- Select
  - Bank 36: Address/Control
Generate MIG Example Design

- Select
  - Bank 26: Data
  - Bank 25: Data
  - Bank 35: Data
Generate MIG Example Design

- Select
  - Bank 34: System Clock
Generate MIG Example Design

- **Select**
  - Master Bank: 25
  - Click Next
Generate MIG Example Design

Leave this page as is
– Click Next
Generate MIG Example Design

- Accept Simulation license, if desired
  - Otherwise, Decline license
  - Click Next
Leave this page as is
- Click Next
Generate MIG Example Design

DDR3 SDRAM Design for Virtex-6 FPGAs

**Design Notes**

1. This design is tested with ISE 11.4 and Synplify Pro C-2009.06-SP1
2. Inner and Outer Column banks are permitted for selection
3. Only Single Rank memory parts are supported
4. In case of DCI Cascading, a Master bank must be selected. In between the Master and Slave banks or in between two Slave banks, there cannot be a bank with different IO Standard (viz., System Clock bank)
5. Users with components must lay out their board using fly by topology as if on a DIMM
6. The frequency range is preliminary and subject to characterization
7. Hardware test bench logic does not support: ECC enable designs and Burst Length "On the Fly"
8. Design control and status pins (viz., sys_rst, error, phy_init_done,..) are allocated in System Clock bank
9. ModelSim simulation error "# sim_tb_top.comp_inst.mem_b_4.gen_mem[2].u_comp_ddr3.dqs at time 18790687.0 ps WARNING: tWLS violation on DQS bit 0 positive edge. Indeterminate CK capture is possible."
Generate MIG Example Design

- After the MIG core finishes generating, click Close on the Datasheet window

Note: Presentation applies to the ML605
Generate MIG Example Design

- MIG design appears in Project IP

Note: Presentation applies to the ML605
Modifications to Example Design

- **RDF0011.zip includes**
  - ChipScope Project File, UCF, and Verilog Files

- **Modifications to RTL Files for ML605 Example Design**
  - Changed design to support a single 200 MHz LVDS clock input
  - Added Debug display code to drive LEDs
  - Added ChipScope ILA and VIO port assignments for ML605 board debug
  - Removed IIC Signals – sda, scl
  - Changed various parameter to match the ML605 board
    - CLKFBOUT_MULT_F = 6 (was 3 in MIG 3.3 output)
    - OUTPUT_DRV to “HIGH”
    - nDQS_COLx
    - DQS_LOC_COLx
    - RST_ACT_LOW = 0 (was 1)

*Note: Presentation applies to the ML605*
Modifications to Example Design

- Updates to UCF file specifically required for ML605 board:
  - Updated IO Locations to match ML605
  - Remove IIC Signals – sda, scl
  - Merged Default two clocks into one clock for ML605
  - Moved sys_reset to CPU_RESET
  - Edited DCI_CASCADE to match ML605
  - Removed CONFIG_PROHIBIT lines
  - Added LOC for GPIO LED signals (2.5V bank voltage)
  - Added LOCs for RSYNC OSERDES and IODELAY

Note: Presentation applies to the ML605
 Modifications to Example Design

- Unzip the rdf0011.zip file to your C:\ml605_mig_design directory
  - This adds modifications to the example design (1)
  - A fully pre-built ML605 example design is included in the zip file (2)
    - Use the included bitstream to run MIG with ChipScope
    - Run `ise_flow.bat` in `<design directory>\ml605_prebuilt_example_design\mig_33\example_design\par` to recompile the pre-built example design

Note: Overwrites Core Generator output files with ML605 specific files (1)
Compile Example Design

- Start a windows command shell and enter these commands:
  
  ```
  cd ml605_mig_design\mig_33\example_design\par
  ise_flow.bat
  ```

**Note:** Presentation applies to the ML605
Run MIG Example Design

- Power on the ML605 board
- Connect a USB Type-A to Mini-B cable to the USB JTAG connector on the ML605 board
  - Connect this cable to your PC

Note: Presentation applies to the ML605
Run MIG Example Design

- After the design compiles, open ChipScope Pro Analyzer
  - Click on the Open Cable Button (1)
  - Click OK (2)

Note: Presentation applies to the ML605
Run MIG Example Design

- Select Device → DEV:0 MyDevice0 (XC6VLX240T) → Configure...
- Select <Design Path>\mig_33\example_design\par\example_top.bit

Note: Presentation applies to the ML605
Run MIG Example Design

- Select File → Open Project...
- Select <Design Path>\mig_33\example_design\par\ML605_SODIMM_example_design.cpj

Note: Presentation applies to the ML605
Run MIG Example Design

- Click on Trigger Setup to view trigger settings
- The error bit value should be set to 1

Note: Presentation applies to the ML605
- Click on Waveform; click the Arm Trigger button (1)
- Detection of an error will cause ChipScope Pro to trigger

Note: Presentation applies to the ML605
Run MIG Example Design

- The Example Design should run error free (no trigger on error)
- To force a trigger, click the T! button (1)

Note: Presentation applies to the ML605
Adjust Data Pattern using VIO Console

- Select VIO Console 4
- Set tg_mod_en_sel to 1

Note: Presentation applies to the ML605
Adjust Data Pattern using VIO Console

- Set `tg_data_mode` to "3" for `HAMMER_DATA_MODE`
Adjust Data Pattern using VIO Console

- Select VIO Console 1
- Note error is active

Note: Presentation applies to the ML605
Adjust Data Pattern using VIO Console

- Press and release the CPU RESET switch, SW10, after each change to tg_mod_en_sel or tg_data_mode

Note: Presentation applies to the ML605
Adjust Data Pattern using VIO Console

- Error is now cleared

Note: Presentation applies to the ML605
Adjust Data Pattern using VIO Console

- Click on Waveform; click the Arm Trigger button (1)
- Force a trigger by clicking the T! button (2)

Note: Presentation applies to the ML605
Adjust Data Pattern using VIO Console

- **Hammer PRBS Data Mode**
  - 64 bit DQ data bus hammer pattern

**Note:** Presentation applies to the ML605
Adjust Data Pattern using VIO Console

- Set `tg_data_mode` to "7" for PRBS data pattern
- Push CPU Reset, click Arm Trigger button, click T! button

Note: Presentation applies to the ML605
Adjust Data Pattern using VIO Console

- **PRBS Data Mode**

*Note:* Presentation applies to the ML605
Example Design VIO Consoles

- Useful for PHY layer logic debug and status
- Available if “debug” option is checked in MIG GUI
  - Monitor PHY outputs
    - Status of write calibration
    - Status of read calibration
  - Phase detector control
  - Read data capture clock adjustment
  - Disable selected PHY features
- Reference documentation in UG406
  - “PHY Layer Debug Port” section
  - Table 1-25 for signal definitions and descriptions
- VIO port assignments (4 cores) defined in “example_top.v”
Example Design VIO Consoles

- **VIO Console 1**
  - Write Path Calibration Status
  - Read Leveling Done, Read Leveling Error
  - Initialization complete, PLL reset
  - Note: Press CPU RESET to clear error status in this VIO console
Example Design VIO Consoles

- **VIO Console 2 & VIO Console 3**
  - Read Path Calibration Status

```plaintext
<table>
<thead>
<tr>
<th>Bus/Signal</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>cpt_taps_0</td>
<td>12</td>
</tr>
<tr>
<td>cpt_taps_1</td>
<td>12</td>
</tr>
<tr>
<td>cpt_taps_2</td>
<td>13</td>
</tr>
<tr>
<td>cpt_taps_3</td>
<td>15</td>
</tr>
<tr>
<td>cpt_taps_4</td>
<td>16</td>
</tr>
<tr>
<td>cpt_taps_5</td>
<td>15</td>
</tr>
<tr>
<td>cpt_taps_6</td>
<td>17</td>
</tr>
<tr>
<td>cpt_taps_7</td>
<td>17</td>
</tr>
<tr>
<td>rd_active_dly</td>
<td>12</td>
</tr>
<tr>
<td>rd_bitslip_0</td>
<td>1</td>
</tr>
<tr>
<td>rd_bitslip_1</td>
<td>1</td>
</tr>
<tr>
<td>rd_bitslip_2</td>
<td>1</td>
</tr>
<tr>
<td>rd_bitslip_3</td>
<td>1</td>
</tr>
<tr>
<td>rd_bitslip_4</td>
<td>1</td>
</tr>
<tr>
<td>rd_bitslip_5</td>
<td>1</td>
</tr>
<tr>
<td>rd_bitslip_6</td>
<td>1</td>
</tr>
</tbody>
</table>
```
Example Design VIO Consoles

- **VIO Console 2 & VIO Console 3**
  - Read Path Calibration Status
Example Design VIO Consoles

- **VIO Console 4**
  - Phase Detector Controls
  - Read Data Capture Clock Adjustment
Generate MIG ACE File (Optional)

- Type these commands in a windows command shell:
  
  ```
  cd C:\ml605_mig_design\ready_for_download
  make_ace.bat
  ```

Note: Presentation applies to the ML605
References

- **Virtex-6 Memory**
  - Virtex-6 FPGA Memory Interface Solutions User Guide – UG406
  - Virtex-6 FPGA Memory Interface Solutions – DS186
Documentation

- **Virtex-6**
  - Virtex-6 FPGA Family

- **ML605 Documentation**
  - Virtex-6 FPGA ML605 Evaluation Kit
  - ML605 Hardware User Guide
  - ML605 Reference Design User Guide