



# ML505/506/507 MIG Design Creation

**May 2010**

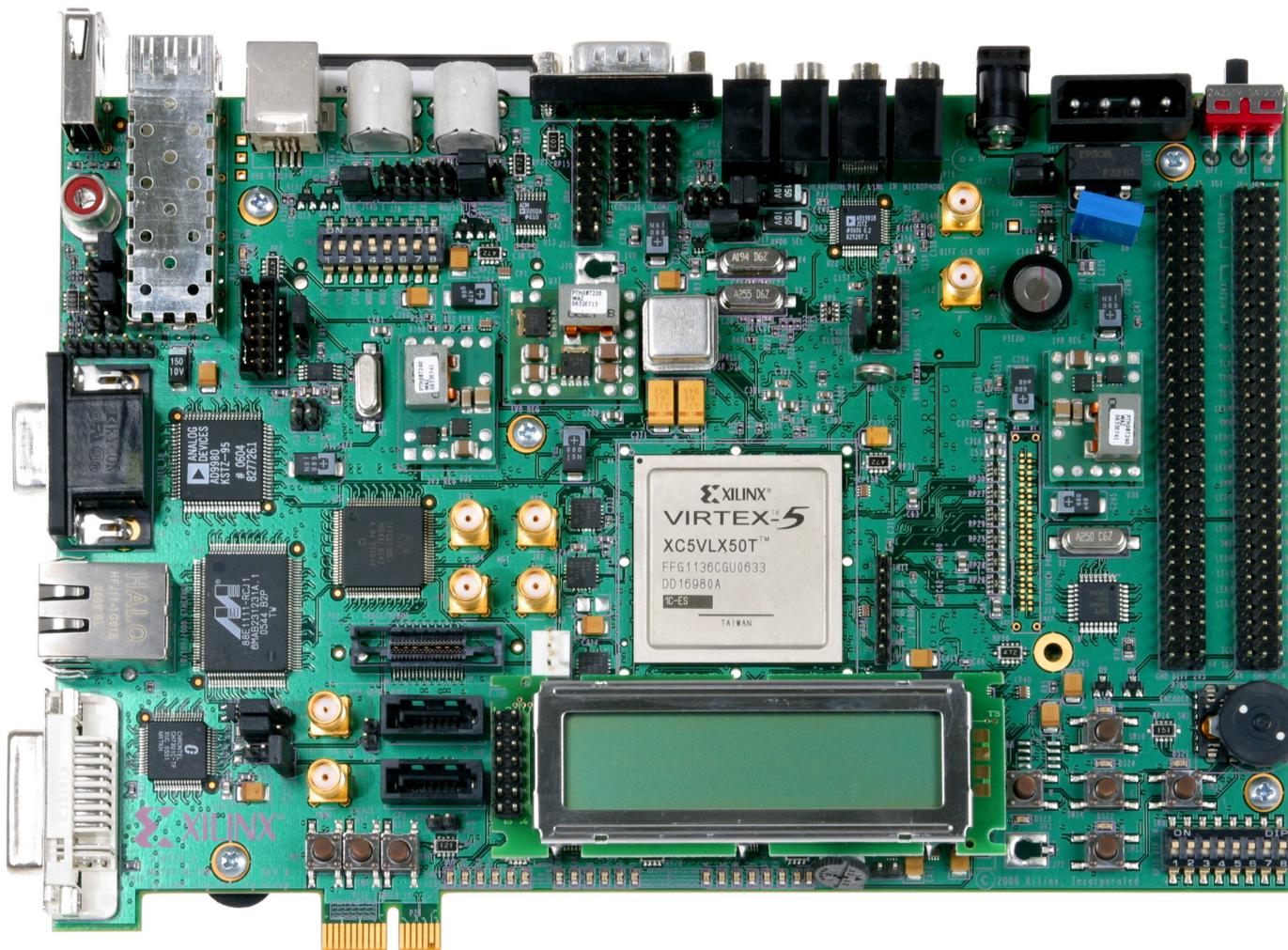
# Overview

- **Hardware Setup**
- **Software Requirements**
- **CORE Generator™ software**
  - Memory Interface Generator (MIG)
- **Modify Design**
  - Add ChipScope Pro Cores to Design
- **Compile and Test Memory Interface**
- **Appendix**
- **References**

# Virtex-5 DDR2 Capabilities

- **MIG DDR2 SDRAM design supports frequencies up to 333 MHz in a -3 speed grade device**
  - The [MIG user guide](#) addresses MIG performance across device speed grades
- **The ML505 ships with a –1 speed grade device**
  - See the [Virtex-5 Datasheet](#) for a list of Virtex-5 supported memory interface speeds
  - MIG DDR2 SODIMM interface design built for 266 MHz operation
    - Maximum SODIMM memory interface frequency

# Xilinx ML505 Board



Note: Presentation applies to the ML505, ML506, and ML507

# Additional Setup Details

- Refer to [ml505\\_overview\\_setup.ppt](#) for details on:

- Software Requirements
- ML505/506/507 Board Setup
- Equipment and Cables
- Software
- Network

- [Terminal Programs](#)

- This presentation requires the 9600-8-N-1 Baud terminal setup



# ISE Software Requirements

- Xilinx ISE 12.1 software



# ChipScope Pro Software Requirement

- Xilinx ChipScope Pro 12.1 software



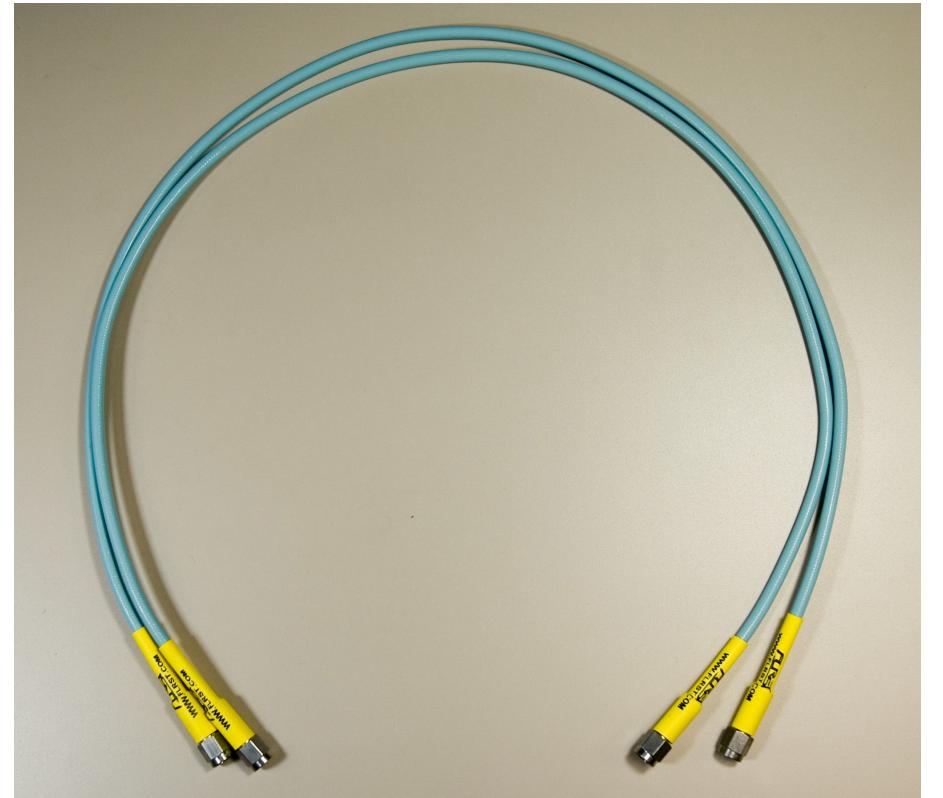
# Hardware Setup

- **Connect the Xilinx Platform Cable USB to the ML505 board**



# Hardware Setup

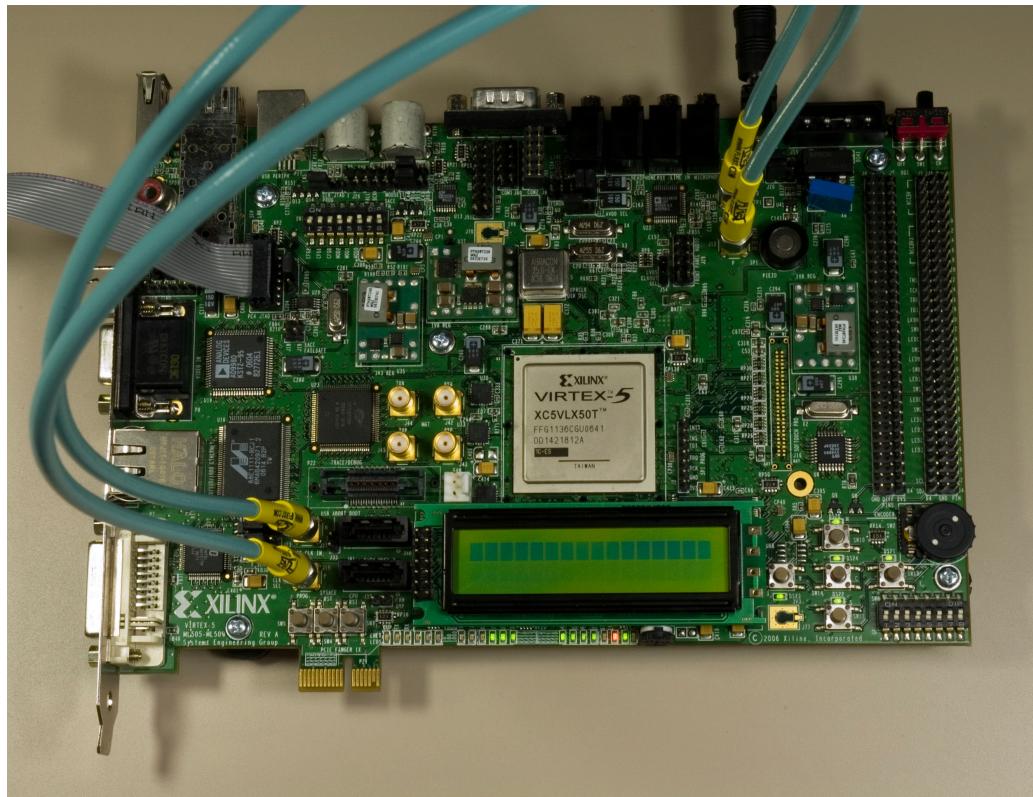
- **MIG DDR2 SODIMM interface design built for 266 MHz operation**
- **Memory Clock via SMA Cable**
  - Using on-board frequency synthesizer
  - Using user supplied external oscillator (See Appendix)
- **SMA Cable**
  - [www.flrst.com](http://www.flrst.com)
  - P/N: ASPI-024-ASPI-S402



# Hardware Setup

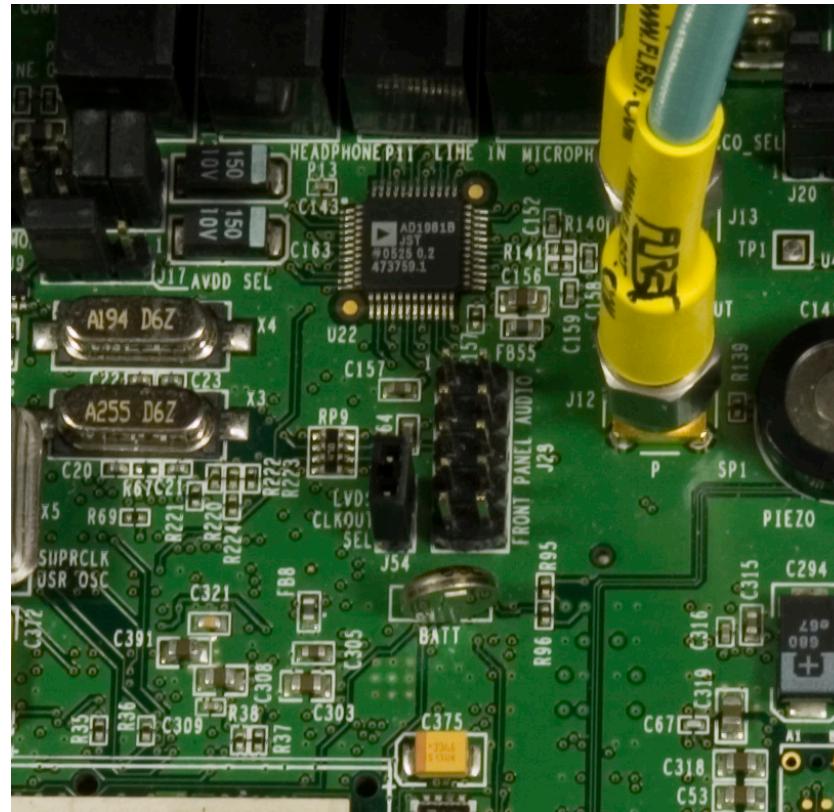
- **Connect SMA Cables from J12/13 to J10/11**

- This provides access to an on-board 200MHz frequency synthesizer
- See appendix for connecting a user provided external oscillator



# Hardware Setup

- **Jumper J54 must be connected**
    - This enables the output of the on-board ICS frequency source



**Note:** Presentation applies to the ML505, ML506, and ML507

# Hardware Setup

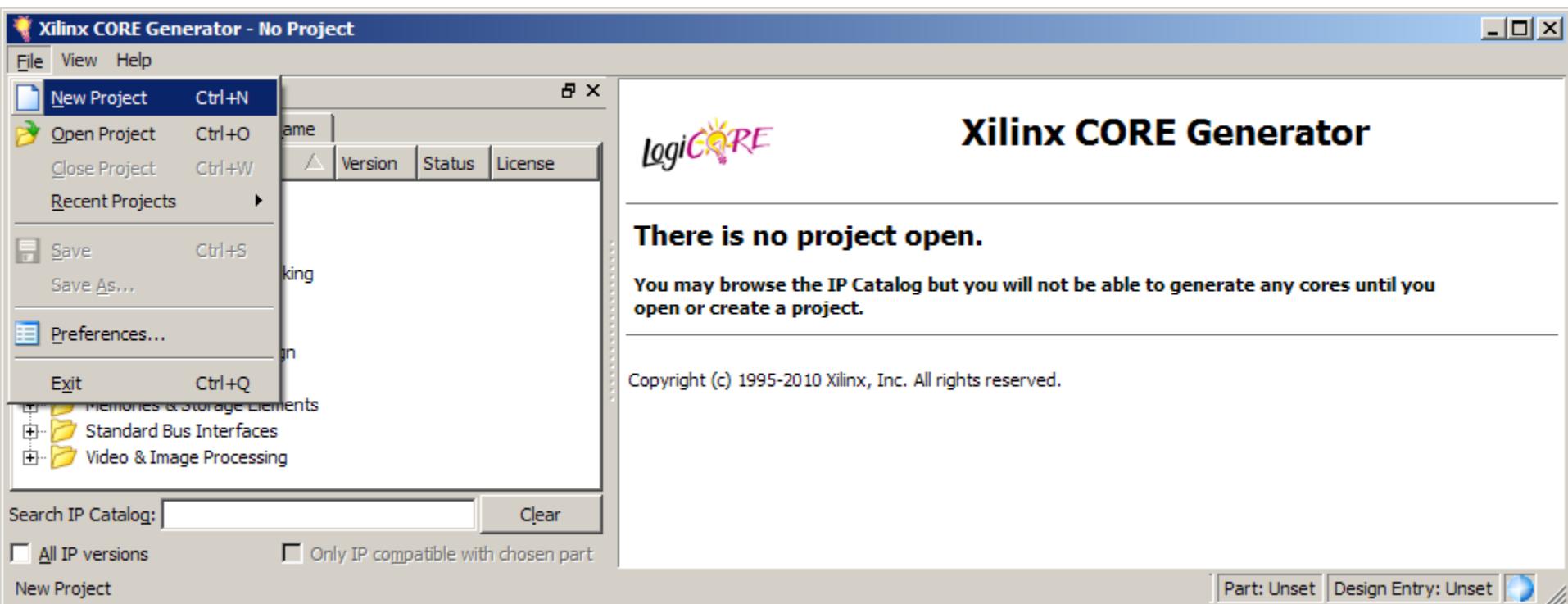
- **Use SW6 to set the memory clock frequency**
- **Set SW6 to 200 MHz (1)**
  - Use SW6 to set the on-board synthesizer's frequency
  - Set SW6 to 200 MHz (maximum synthesizable in-range memory clock)
  - 200 MHz = 010 010 10
  - External oscillator can be used for 266 MHz operation (See Appendix)

# CORE Generator

- Open the CORE Generator

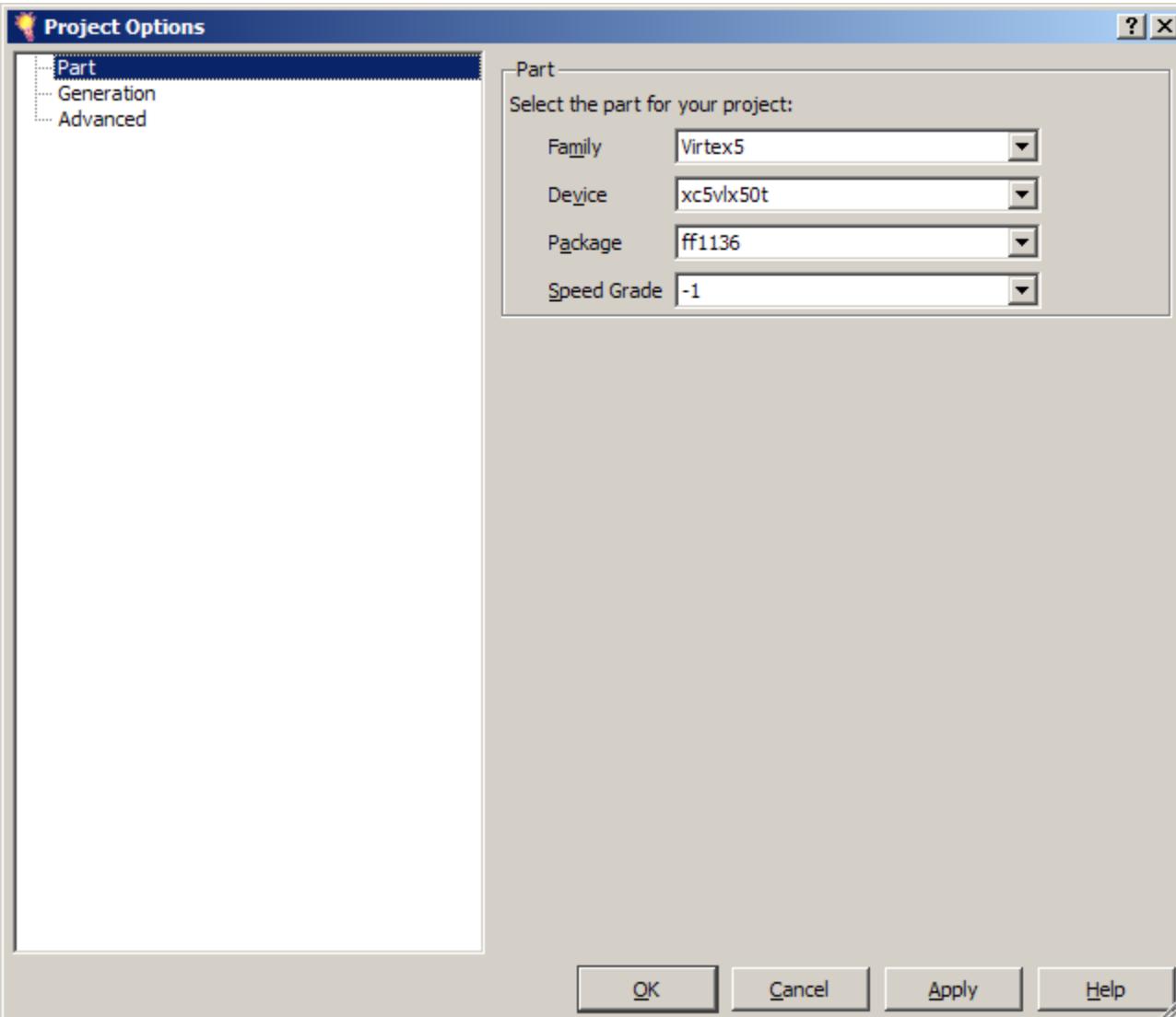
Start → All Programs → Xilinx ISE Design Suite 12.1 →  
ISE → Accessories → CORE Generator

- Create a new project; select File → New Project



Note: Presentation applies to the ML505, ML506, and ML507

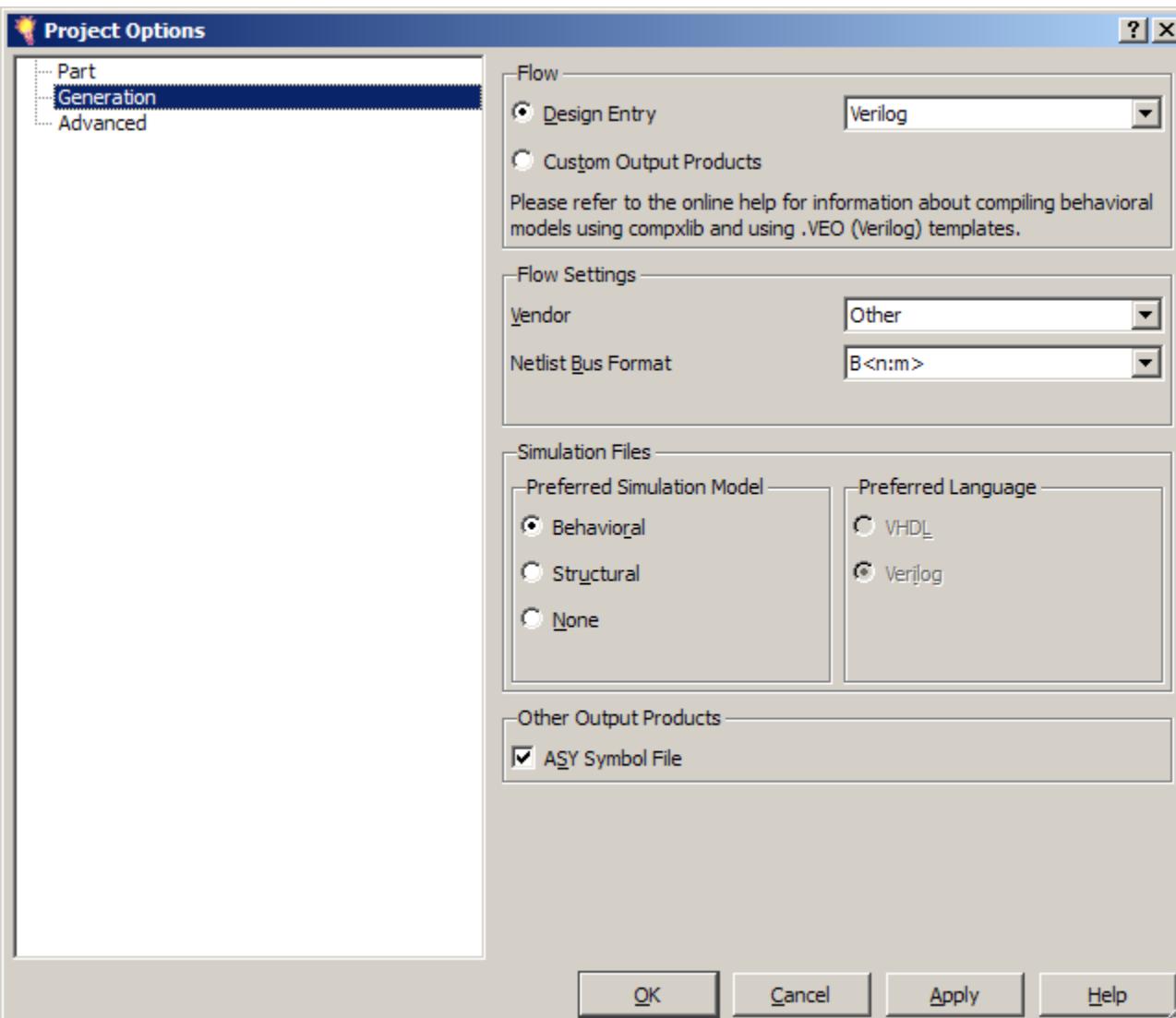
# Generate MIG Core



- **Create a project directory:** `ml505_mig_design`
- **Name the project:** `ml505_mig_design.cgp`
- **Set the Part (as shipped on the ML505):**
  - Family: Virtex5
  - Device: xc5vlx50t
  - Package: ff1136
  - Speed Grade: -1

**Note:** Select the xc5vsx50t for the ML506, and the xc5vfx70t for the ML507

# Generate MIG Core

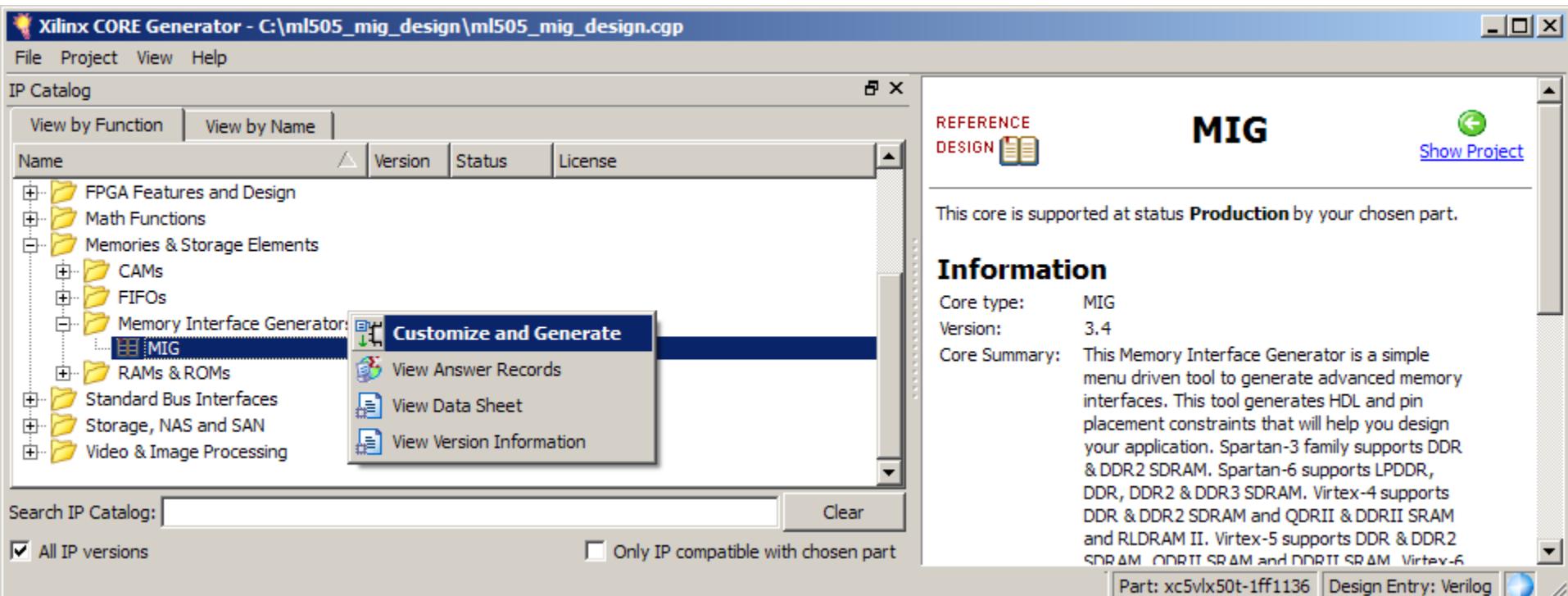


- Select Generation
- Set the Design Entry to Verilog
- Click OK

Note: Presentation applies to the ML505, ML506, and ML507

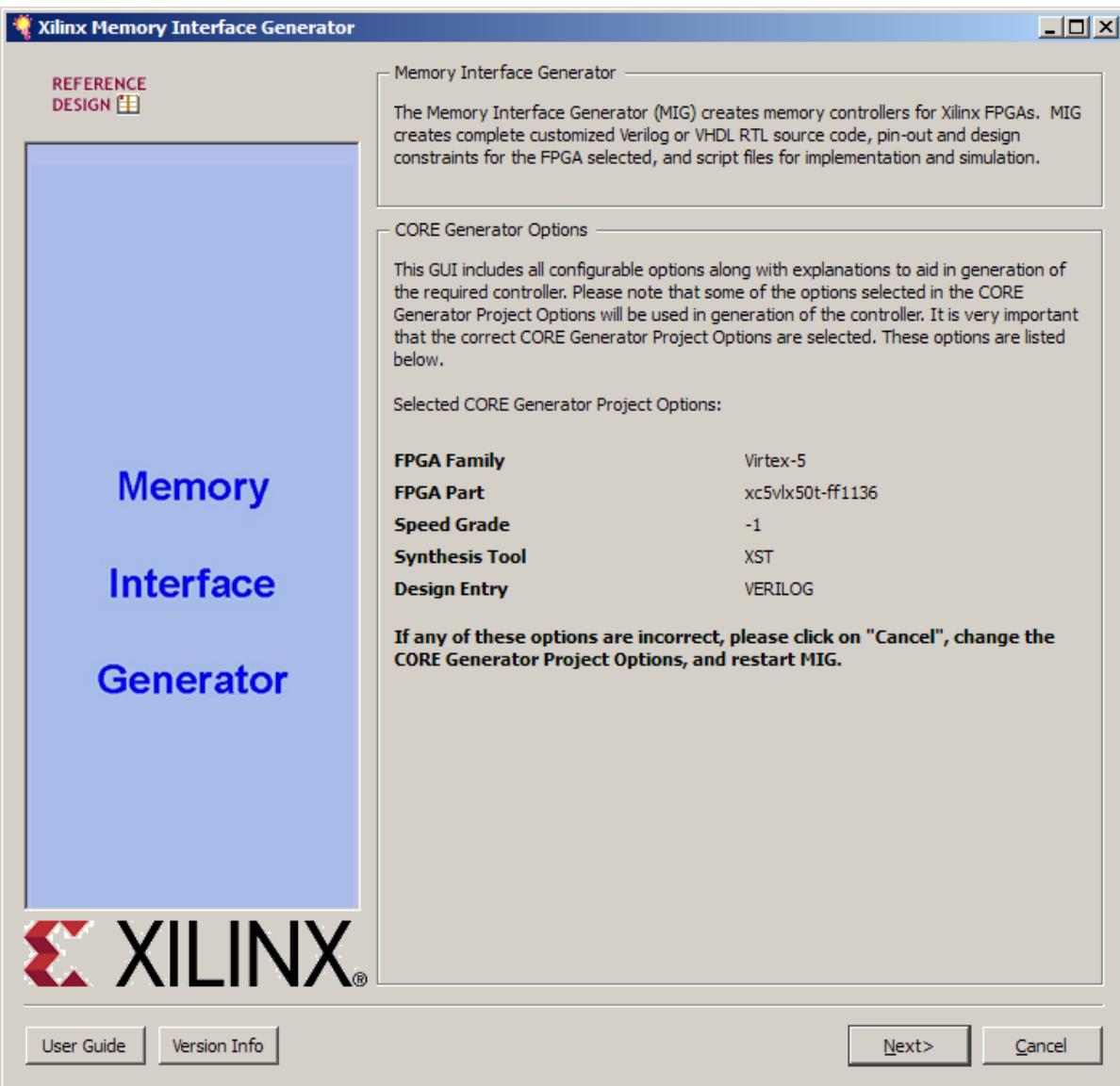
# Generate MIG Core

- Right click on MIG Version 3.4
  - Select Customize and Generate



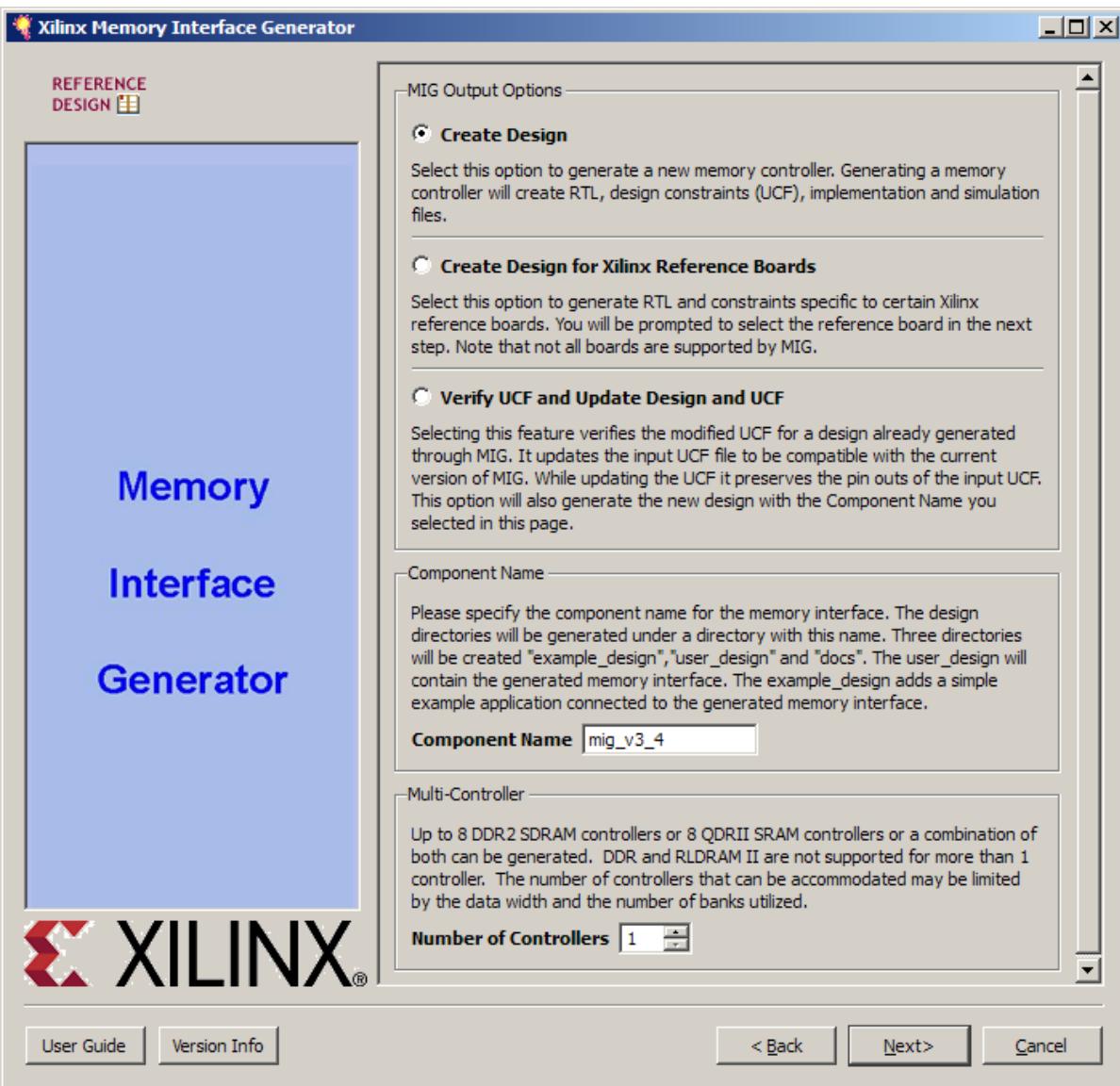
Note: Presentation applies to the ML505, ML506, and ML507

# Generate MIG Core



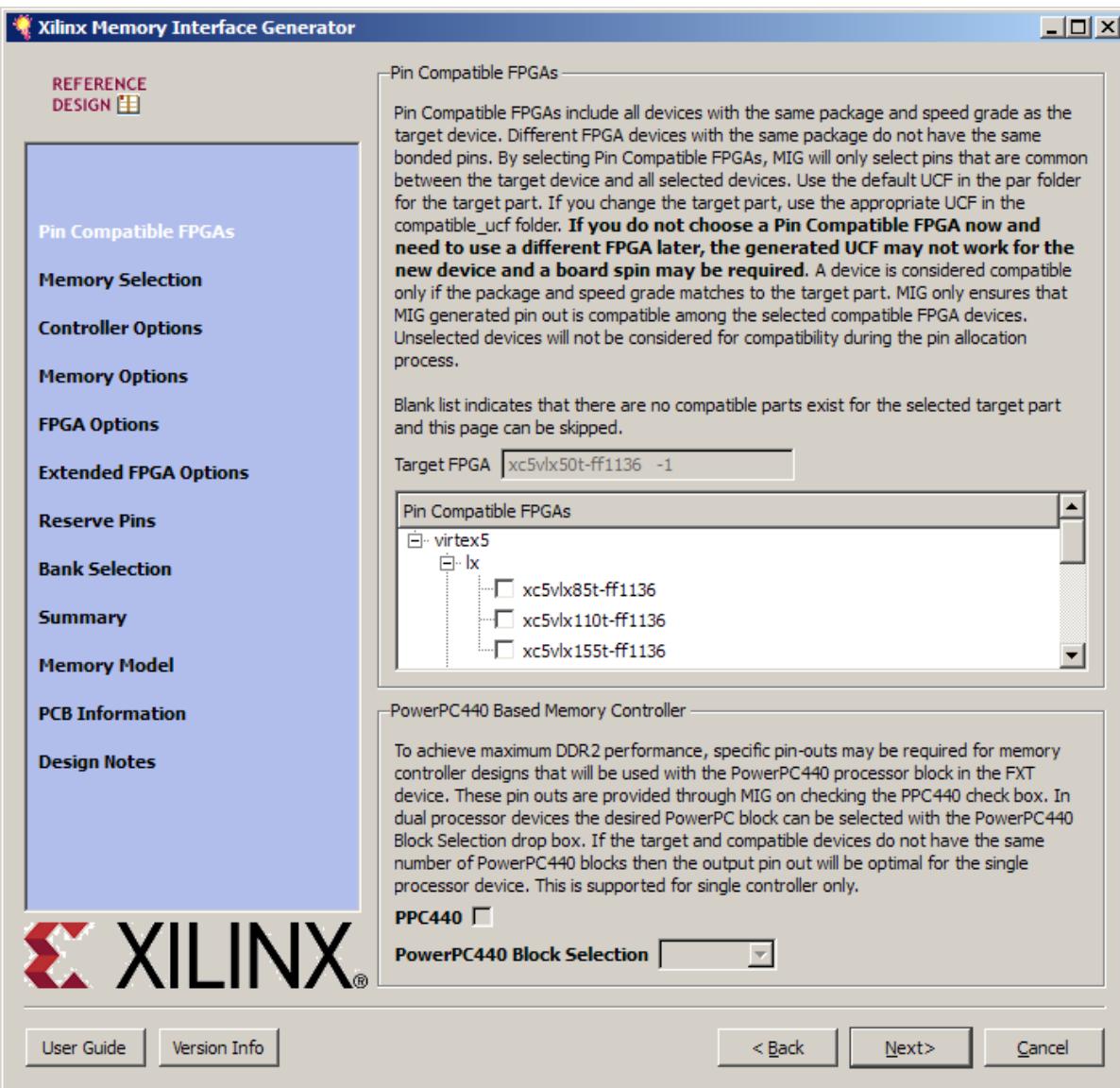
- Leave this page as is
  - Click Next

# Generate MIG Core



- Leave this page as is
  - Click Next

# Generate MIG Core



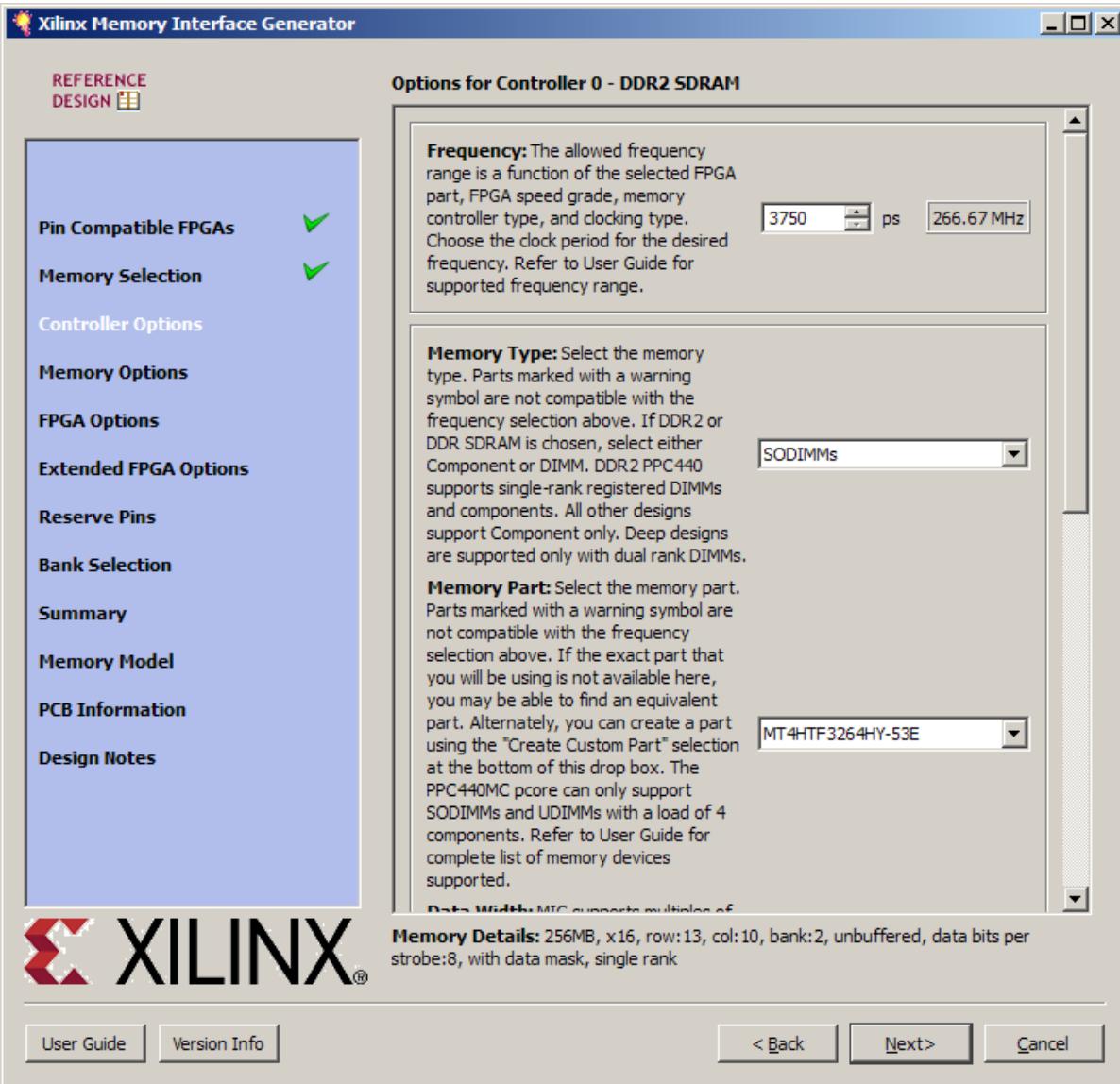
- Leave this page as is
  - Click Next

# Generate MIG Core



- **Select DDR2 SDRAM**
  - Click Next

# Generate MIG Core



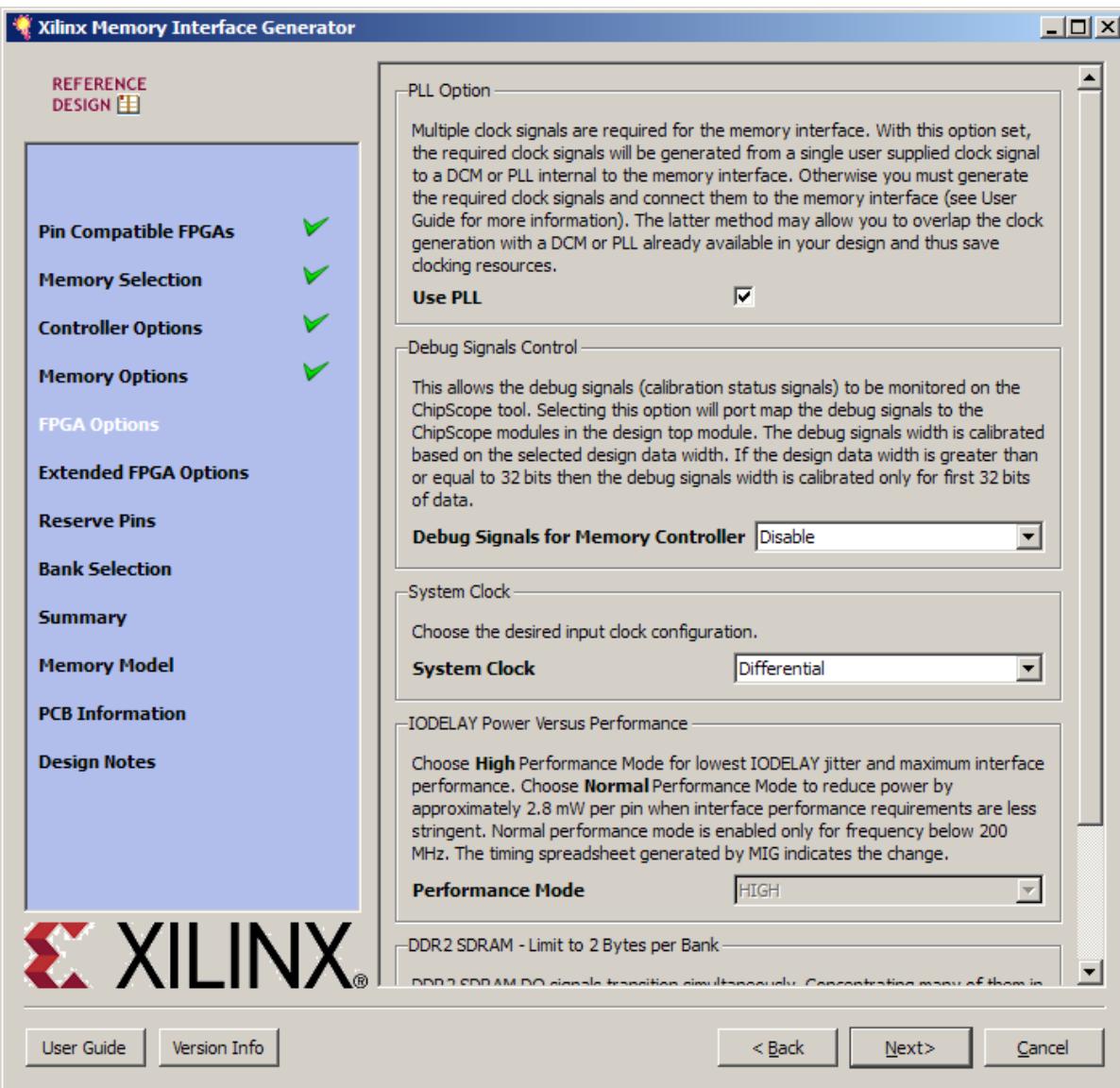
- **MIG defaults to a maximum frequency of 266 for a Virtex-5 –1 speed grade selection**
  - See [UG086](#)
- **Make the following settings:**
  - Set the Memory Type to SODIMMs
  - Set the Memory part to MT4HTF3264HY-53E
- **Click Next**

# Generate MIG Core



- Leave this page as is
  - Click Next

# Generate MIG Core



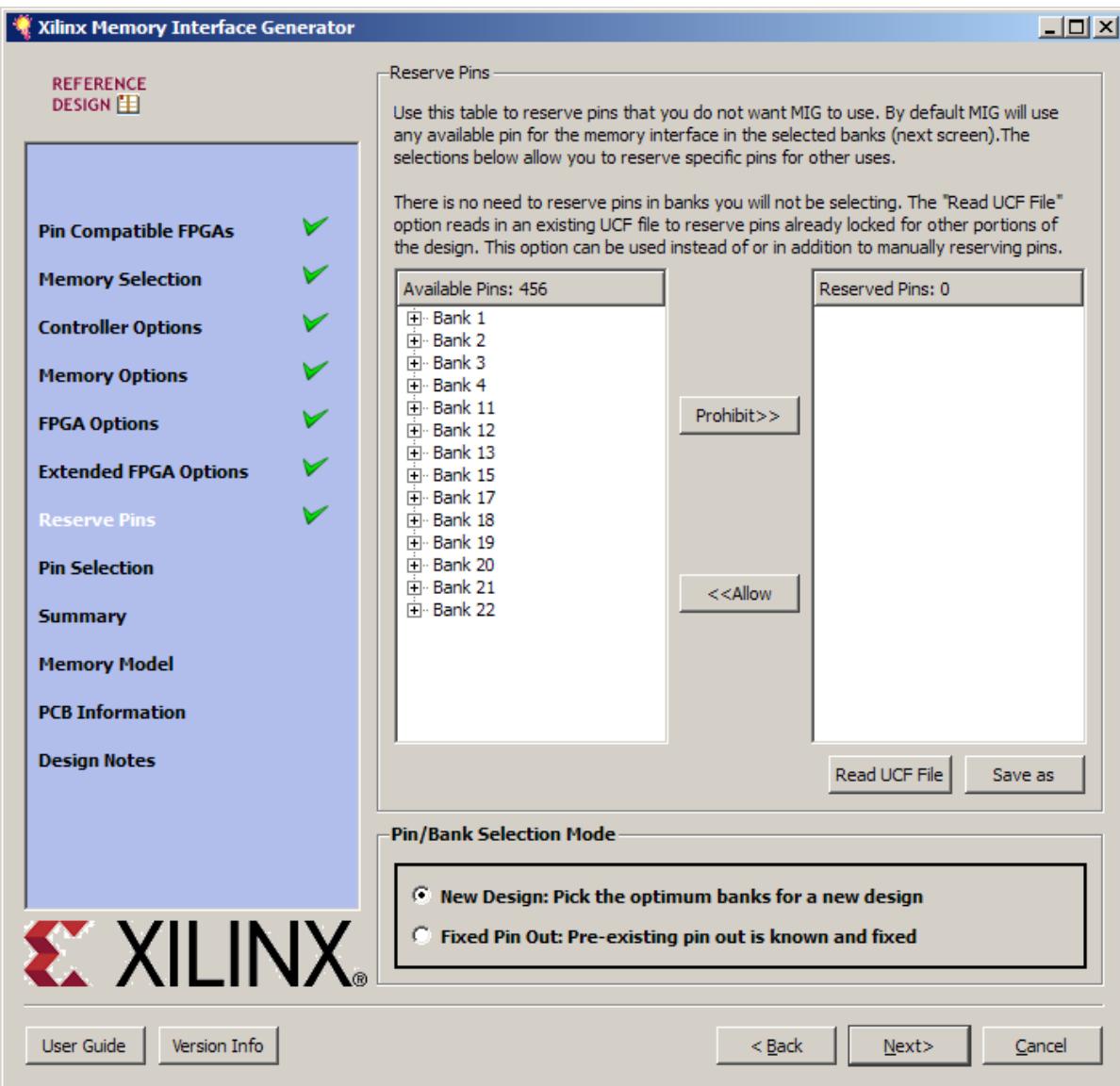
- Leave this page as is
  - Click Next

# Generate MIG Core



- Leave this page as is
  - Click Next

# Generate MIG Core



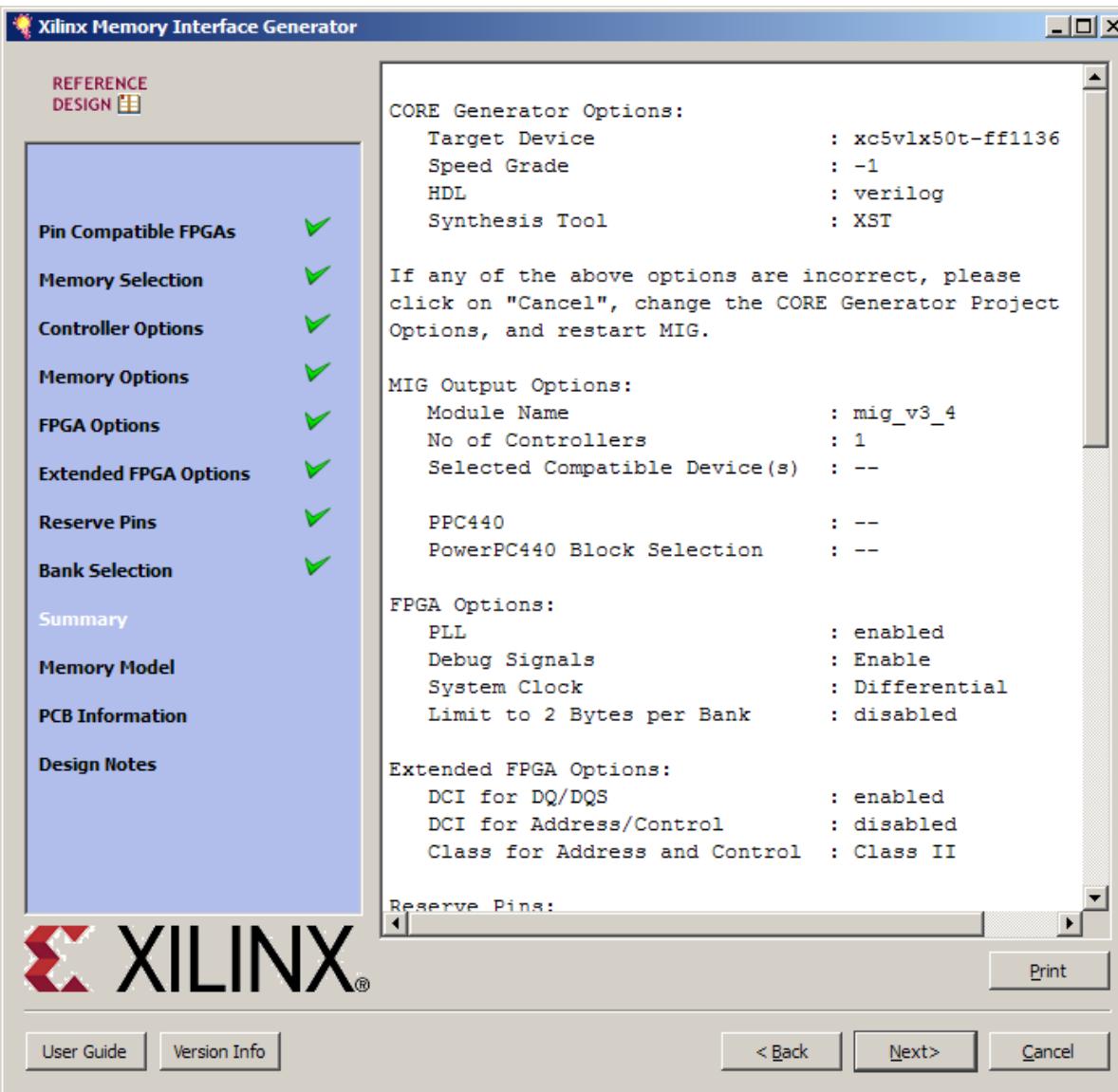
- Select New Design
  - Click Next

# Generate MIG Core



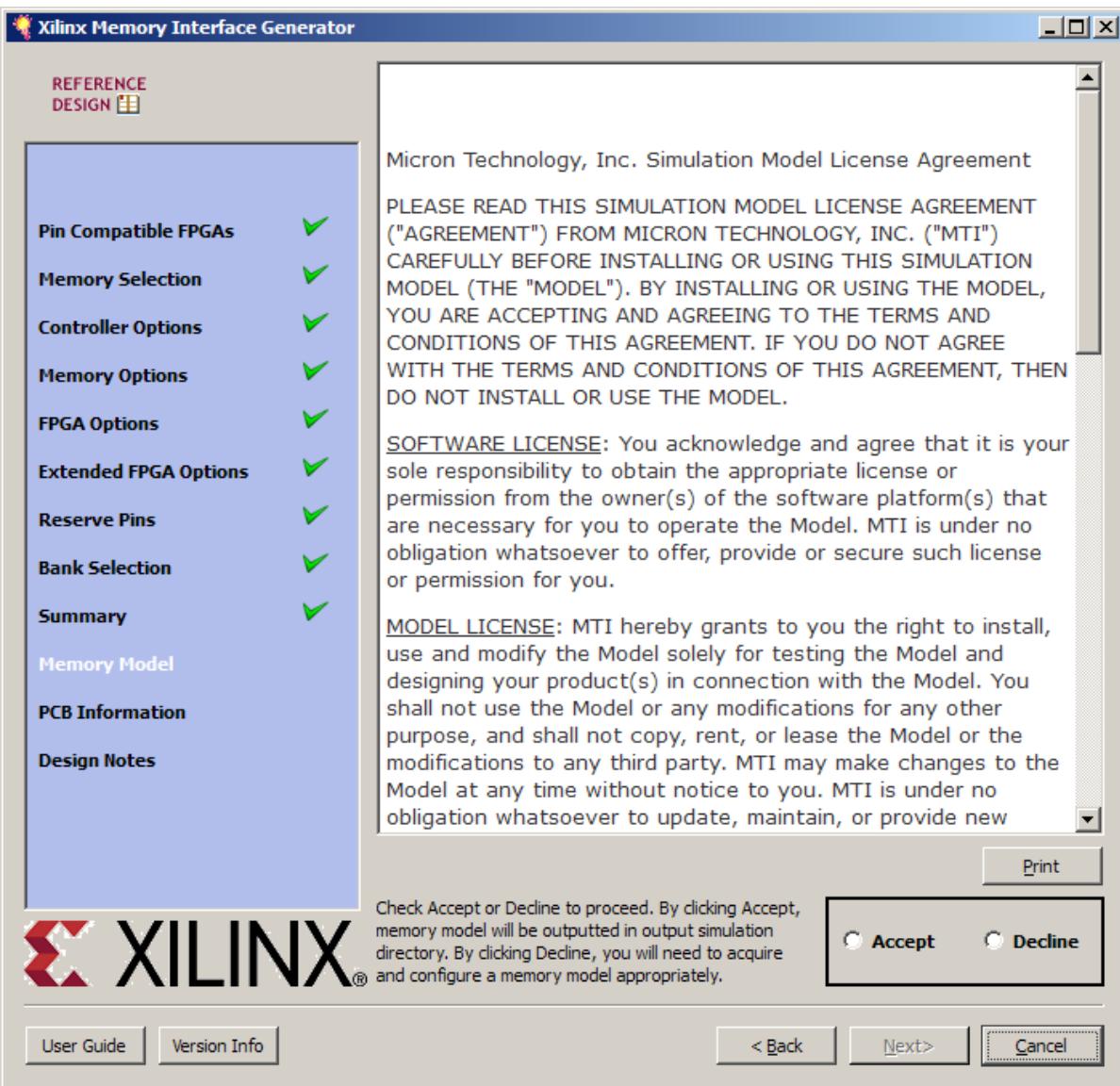
- Leave this page as is
  - Click Next

# Generate MIG Core



- Leave this page as is
  - Click Next

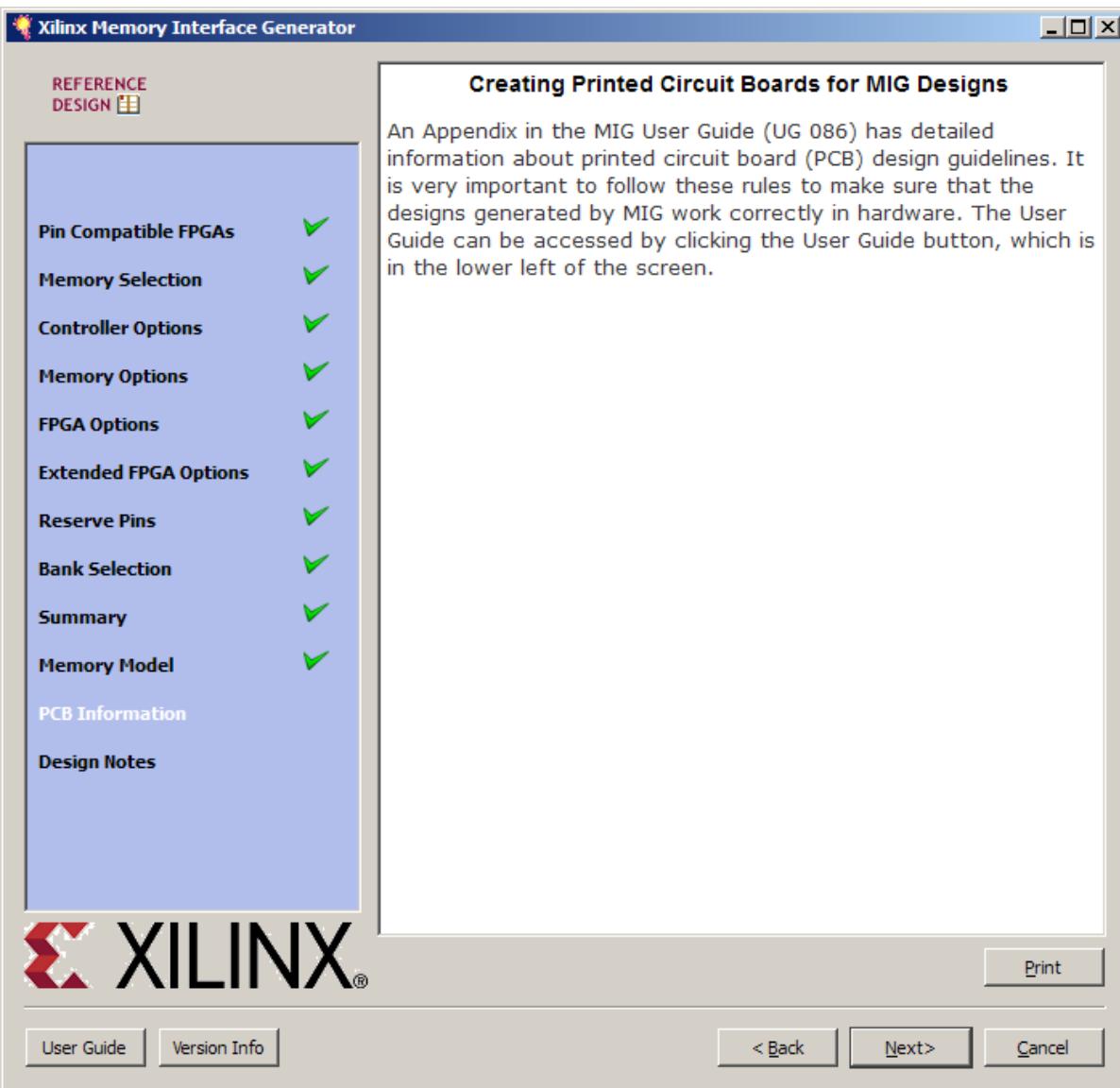
# Generate MIG Core



- **Accept Simulation license, if desired**

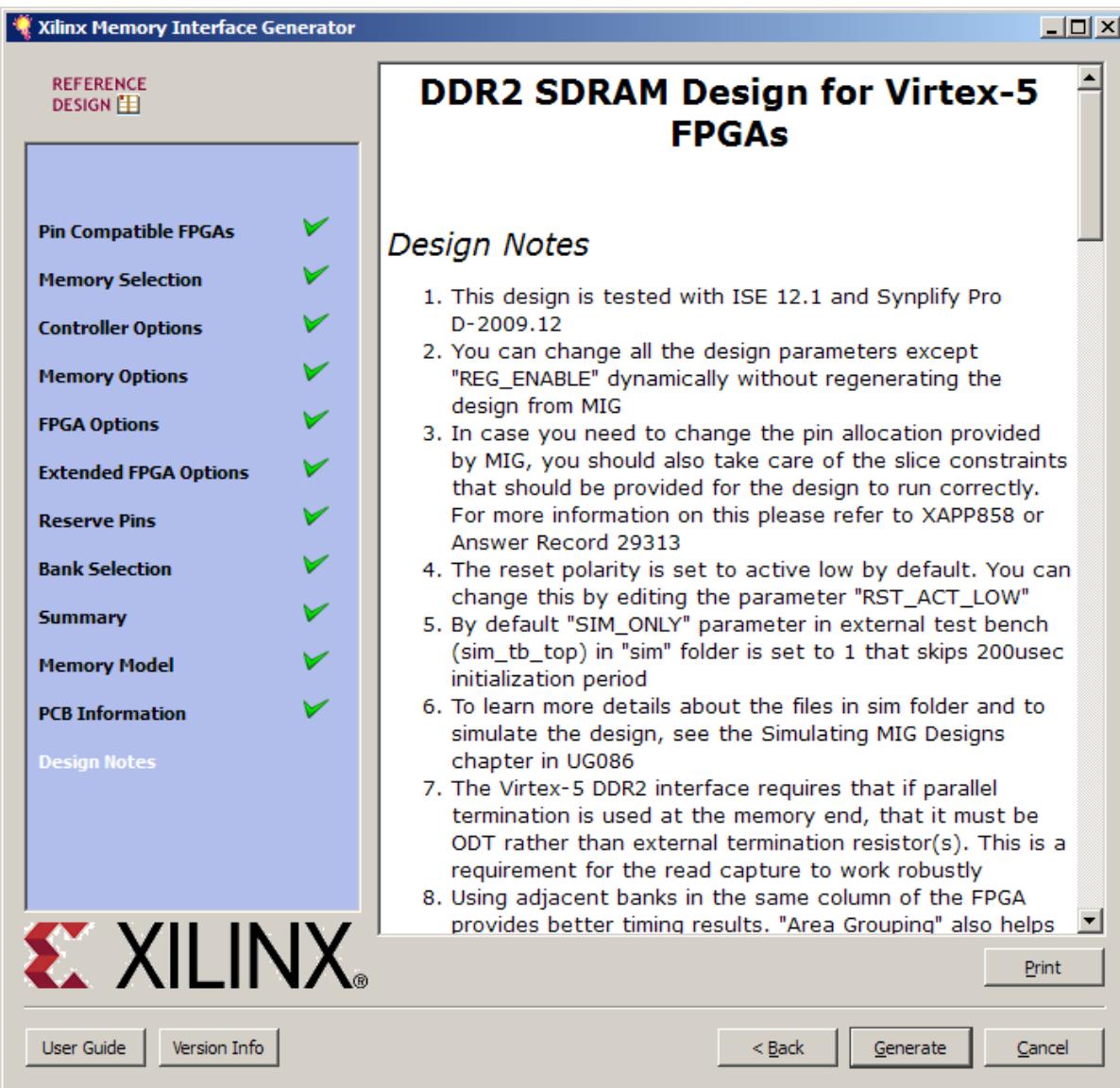
- Otherwise, Decline license
- Click Next

# Generate MIG Core



- Leave this page as is
  - Click Next

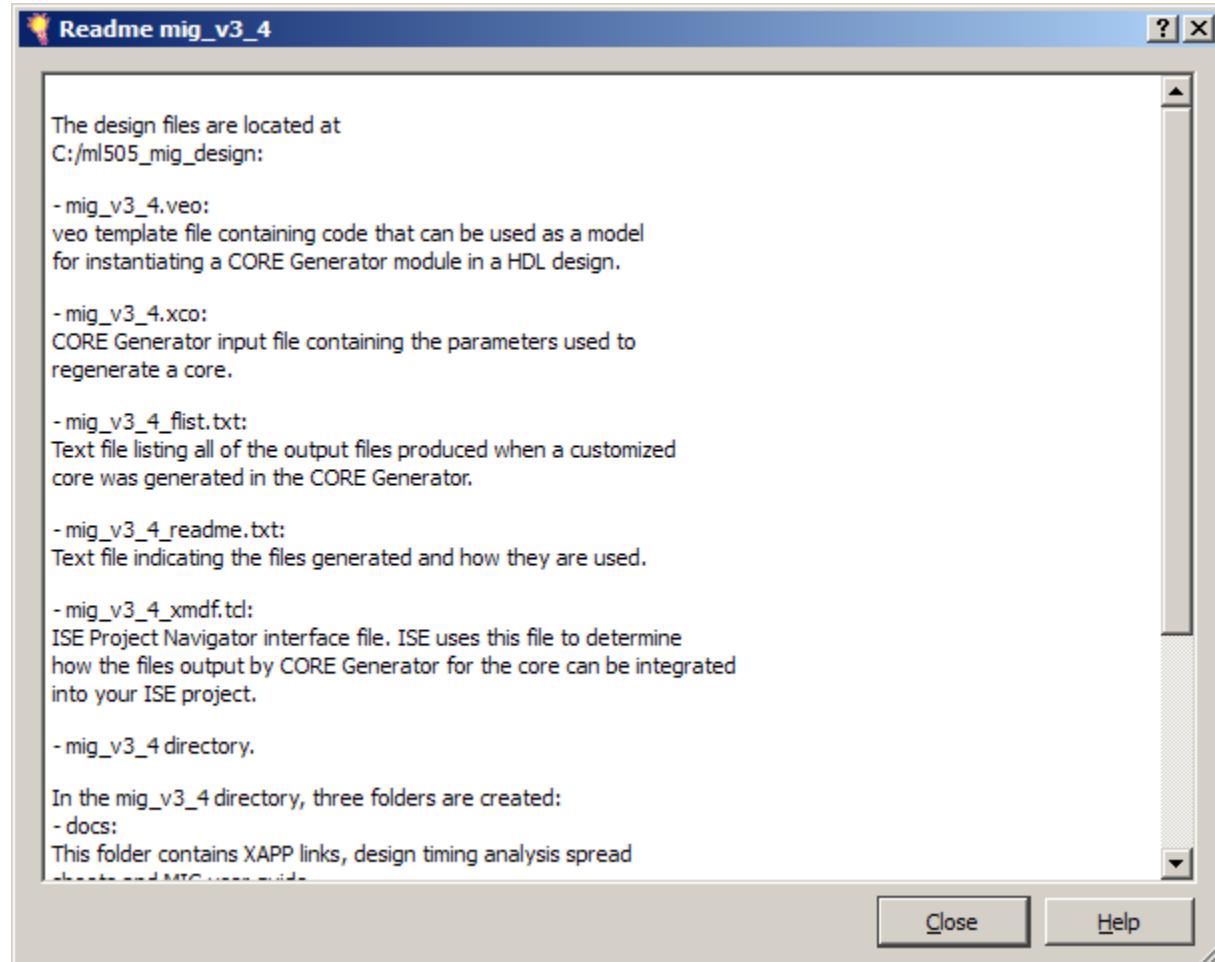
# Generate MIG Core



- Leave this page as is
  - Click Generate

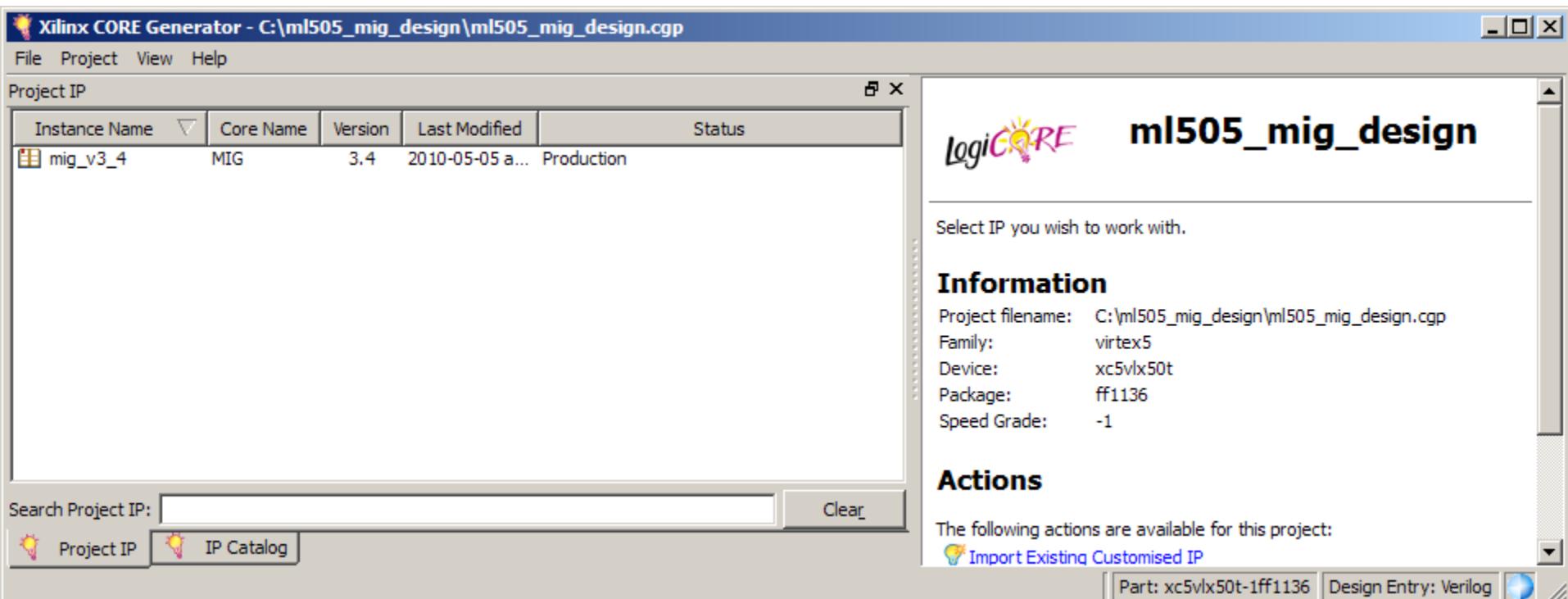
# Generate MIG Core

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



# Generate MIG Core

- MIG core appears in Project IP



Note: Presentation applies to the ML505, ML506, and ML507

# Design Modifications

- **Add overlay files**

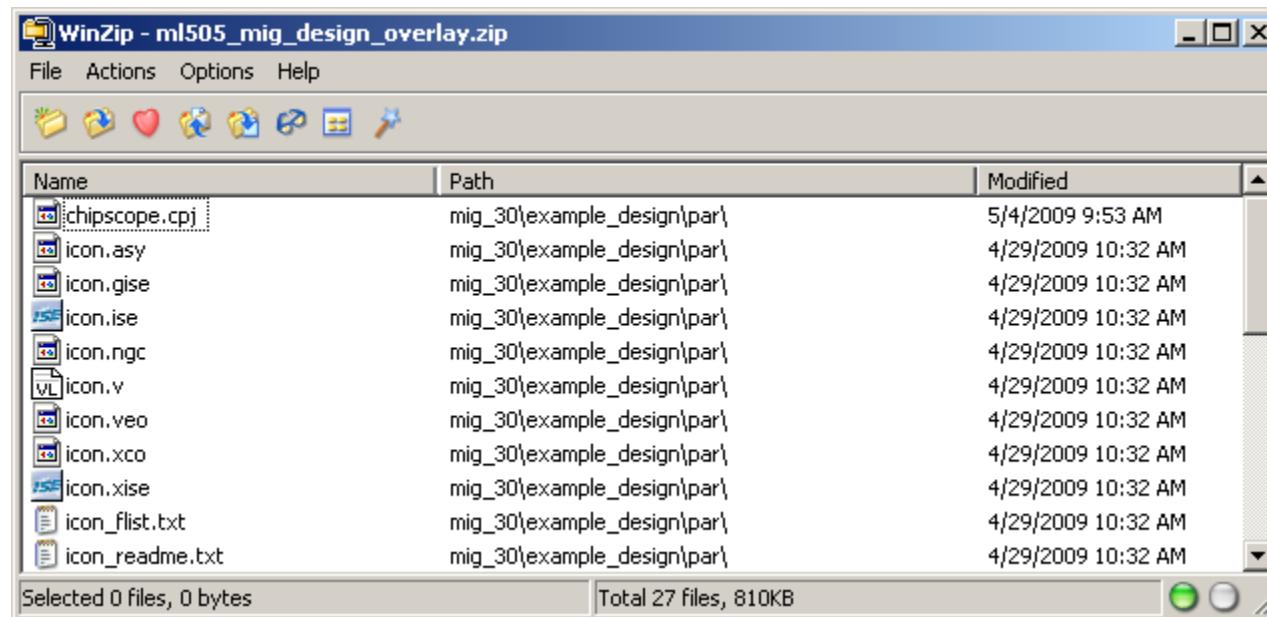
- ML50x specific UCF file as per [AR29313](#)
- Pre-compiled ChipScope Pro design files used to validate the design

- **Modify top level Verilog file**

- Support for ML50x Board as per [AR29313](#)
- Add ChipScope Pro to design

# Add Design Files

- **Unzip the ml505\_mig\_design\_overlay.zip file**
  - Unzip to the ml505\_mig\_design directory
  - See ChipScope Pro documentation for details on generating/instantiating the ICON and ILA cores

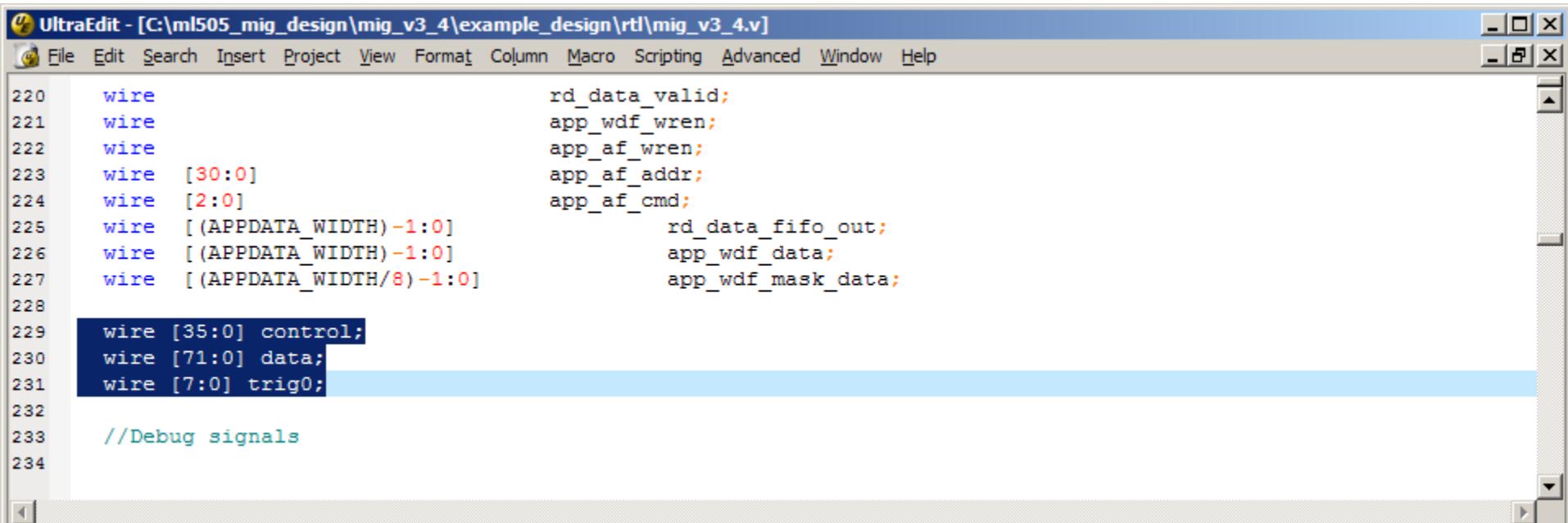


Note: ML506: [ml506\\_mig\\_design\\_overlay.zip](#) ML507: [ml507\\_mig\\_design\\_overlay.zip](#)

# Add ChipScope Pro Cores

- Add these lines to the top-level Verilog file (mig\_v3\_4.v):

```
    wire [35:0] control;  
    wire [71:0] data;  
    wire [7:0] trig0;
```



```
UltraEdit - [C:\ml505_mig_design\mig_v3_4\example_design\rtl\mig_v3_4.v]  
File Edit Search Insert Project View Format Column Macro Scripting Advanced Window Help  
220     wire                               rd_data_valid;  
221     wire                               app_wdf_wren;  
222     wire                               app_af_wren;  
223     wire [30:0]                         app_af_addr;  
224     wire [2:0]                          app_af_cmd;  
225     wire [(APPDATA_WIDTH)-1:0]          rd_data_fifo_out;  
226     wire [(APPDATA_WIDTH)-1:0]          app_wdf_data;  
227     wire [(APPDATA_WIDTH/8)-1:0]         app_wdf_mask_data;  
228  
229     wire [35:0] control;  
230     wire [71:0] data;  
231     wire [7:0] trig0;  
232  
233 //Debug signals  
234
```

# Add ChipScope Pro Cores

- Add these lines to the top-level Verilog file before endmodule:

```
icon i_icon
(
    .CONTROL0(control)
);

ila i_ilal
(
    .CLK(clk0),
    .CONTROL(control),
    .TRIG0(trig0),
    .DATA(data)
);

assign data[63:0] = app_af_addr;
assign data[64]   = app_wdf_afull;
assign data[65]   = app_af_afull;
assign data[66]   = app_wdf_wren;
assign data[67]   = app_af_cmd;
assign data[68]   = phy_init_done;
assign data[69]   = idelay_ctrl_rdy;
assign data[70]   = sys_rst_n;
assign data[71]   = error;
assign trig0[0]   = phy_init_done;
assign trig0[1]   = idelay_ctrl_rdy;
assign trig0[2]   = sys_rst_n;
assign trig0[3]   = error;
assign trig0[4]   = app_wdf_afull;
assign trig0[5]   = app_af_afull;
assign trig0[6]   = app_wdf_wren;
assign trig0[7]   = app_af_cmd;
```

```
655 ila i_ilal
656 (
657     .CLK(clk0),
658     .CONTROL(control),
659     .TRIG0(trig0),
660     .DATA(data)
661 );
662
663 assign data[63:0] = app_af_addr;
664 assign data[64]   = app_wdf_afull;
665 assign data[65]   = app_af_afull;
666 assign data[66]   = app_wdf_wren;
667 assign data[67]   = app_af_cmd;
668 assign data[68]   = phy_init_done;
669 assign data[69]   = idelay_ctrl_rdy;
670 assign data[70]   = sys_rst_n;
671 assign data[71]   = error;
672 assign trig0[0]   = phy_init_done;
673 assign trig0[1]   = idelay_ctrl_rdy;
674 assign trig0[2]   = sys_rst_n;
675 assign trig0[3]   = error;
676 assign trig0[4]   = app_wdf_afull;
677 assign trig0[5]   = app_af_afull;
678 assign trig0[6]   = app_wdf_wren;
679 assign trig0[7]   = app_af_cmd;
680
681 endmodule
```

# Add ChipScope Pro Cores

- Add these lines to the top-level Verilog file after endmodule:

```
module icon (
    CONTROL0
);
    inout [35 : 0] CONTROL0;
endmodule

module ila (
    CLK, CONTROL, TRIGO, DATA
);
    input CLK;
    inout [35 : 0] CONTROL;
    input [7 : 0] TRIGO;
    input [71 : 0] DATA;
endmodule
```

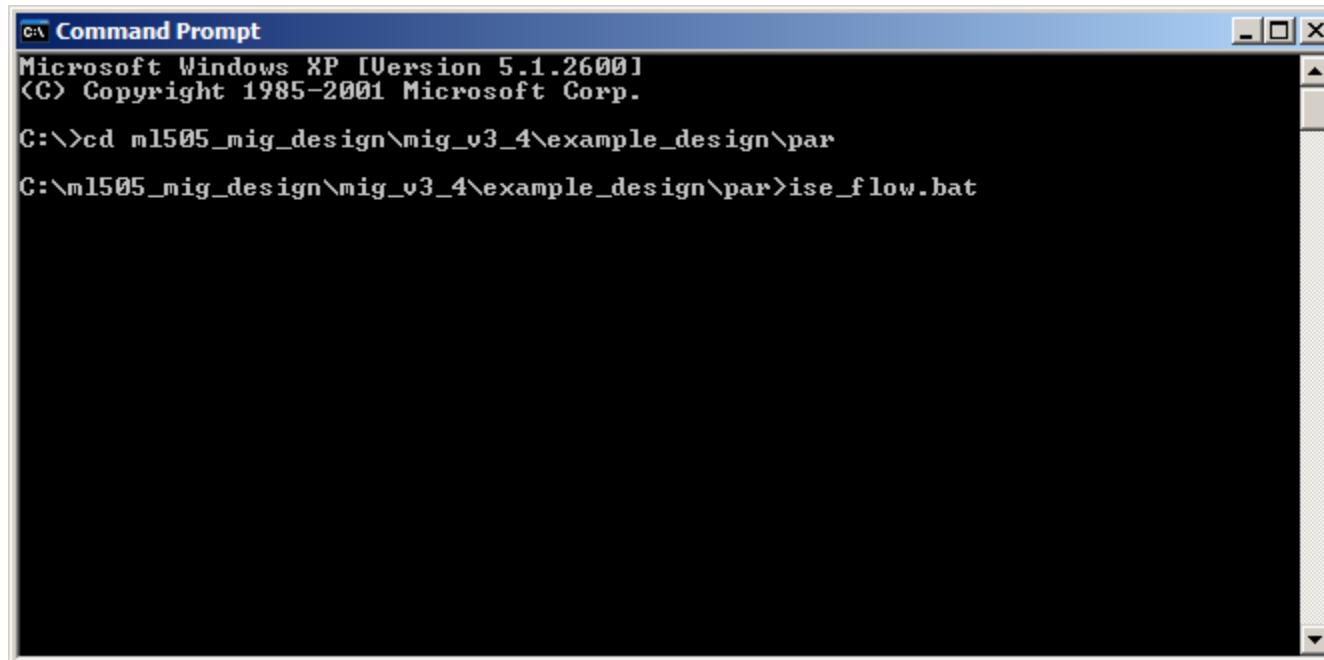
The screenshot shows a Windows application window titled "UltraEdit - [C:\ml505\_mig\_design\mig\_v3\_4\example\_design\rtl\mig\_v3\_4.v]". The code editor displays a Verilog file with several modules defined. The "ila" module is currently selected. The code includes assignments for various signals like data[71], trig0[0-7], and control signals, followed by an "endmodule" statement. Below this, there are two other module definitions for "icon" and "ila", each with their respective port lists and "endmodule" statements. The code is color-coded for syntax highlighting.

```
671 assign data[71] = error;
672 assign trig0[0] = phy_init_done;
673 assign trig0[1] = idelay_ctrl_rdy;
674 assign trig0[2] = sys_rst_n;
675 assign trig0[3] = error;
676 assign trig0[4] = app_wdf_afull;
677 assign trig0[5] = app_af_afull;
678 assign trig0[6] = app_wdf_wren;
679 assign trig0[7] = app_af_cmd;
680
681 endmodule
682
683 module icon (
684     CONTROL0
685 );
686     inout [35 : 0] CONTROL0;
687 endmodule
688
689 module ila (
690     CLK, CONTROL, TRIGO, DATA
691 );
692     input CLK;
693     inout [35 : 0] CONTROL;
694     input [7 : 0] TRIGO;
695     input [71 : 0] DATA;
696 endmodule
697
```

# Generate Bitstream

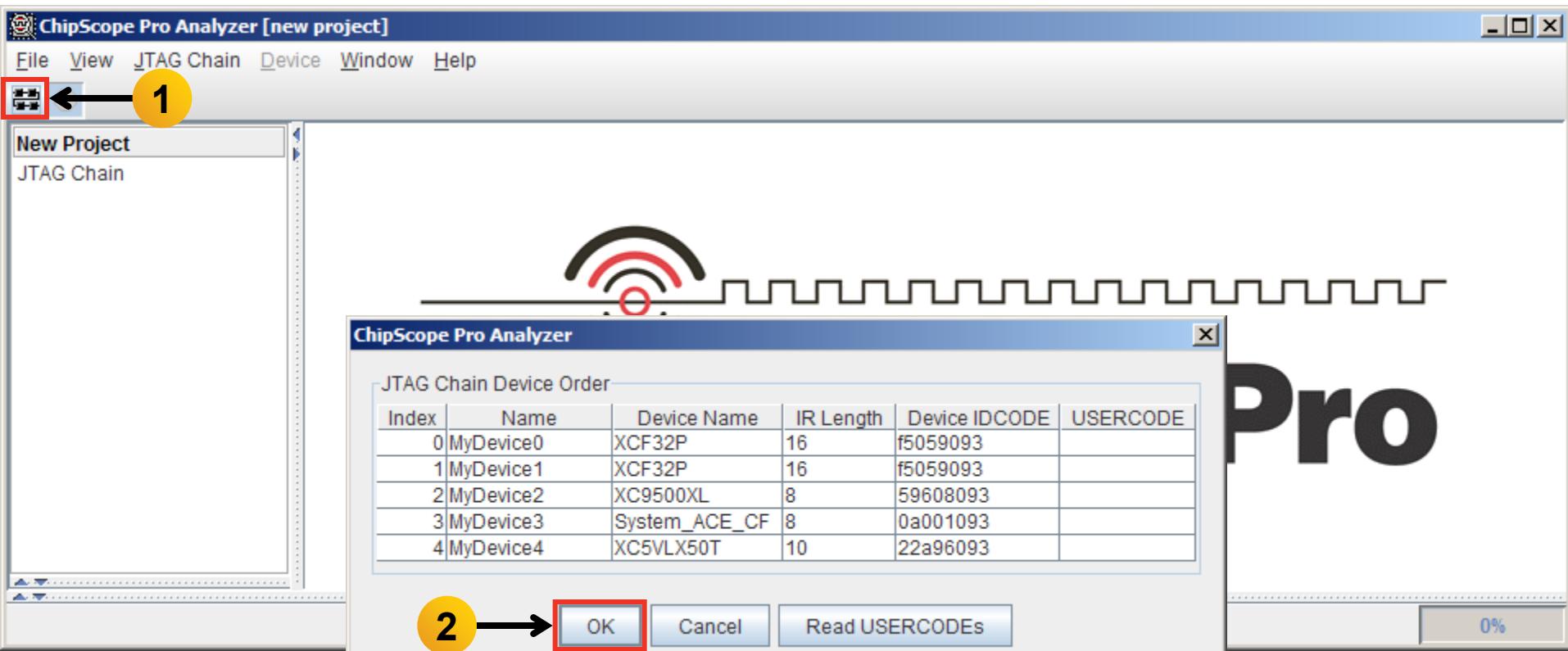
- Start a windows command shell and enter these commands:

```
cd ml505_mig_design\mig_v3_4\example_design\par  
ise_flow.bat
```



# ChipScope Pro Setup

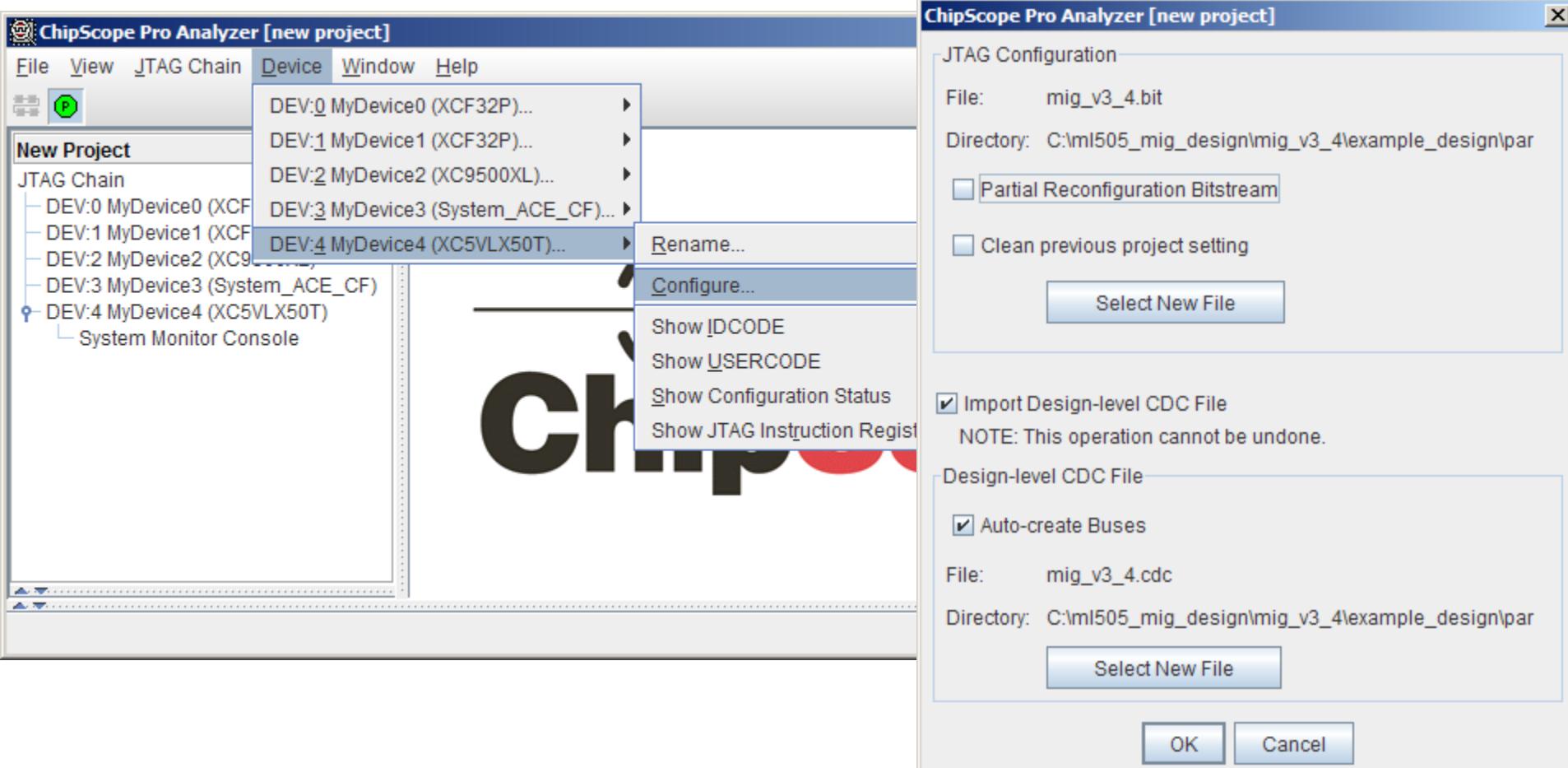
- After the design compiles, open ChipScope Pro Analyzer
  - Click on the Open Cable Button (1)
  - Click OK (2)



Note: Presentation applies to the ML505, ML506, and ML507

# ChipScope Pro Setup

- Select Device → DEV:4 MyDevice4 (XC5VLX50T) → Configure...
- Select <Design Path>\mig\_v3\_4\example\_design\par\ mig\_v3\_4.bit



# ChipScope Pro Setup

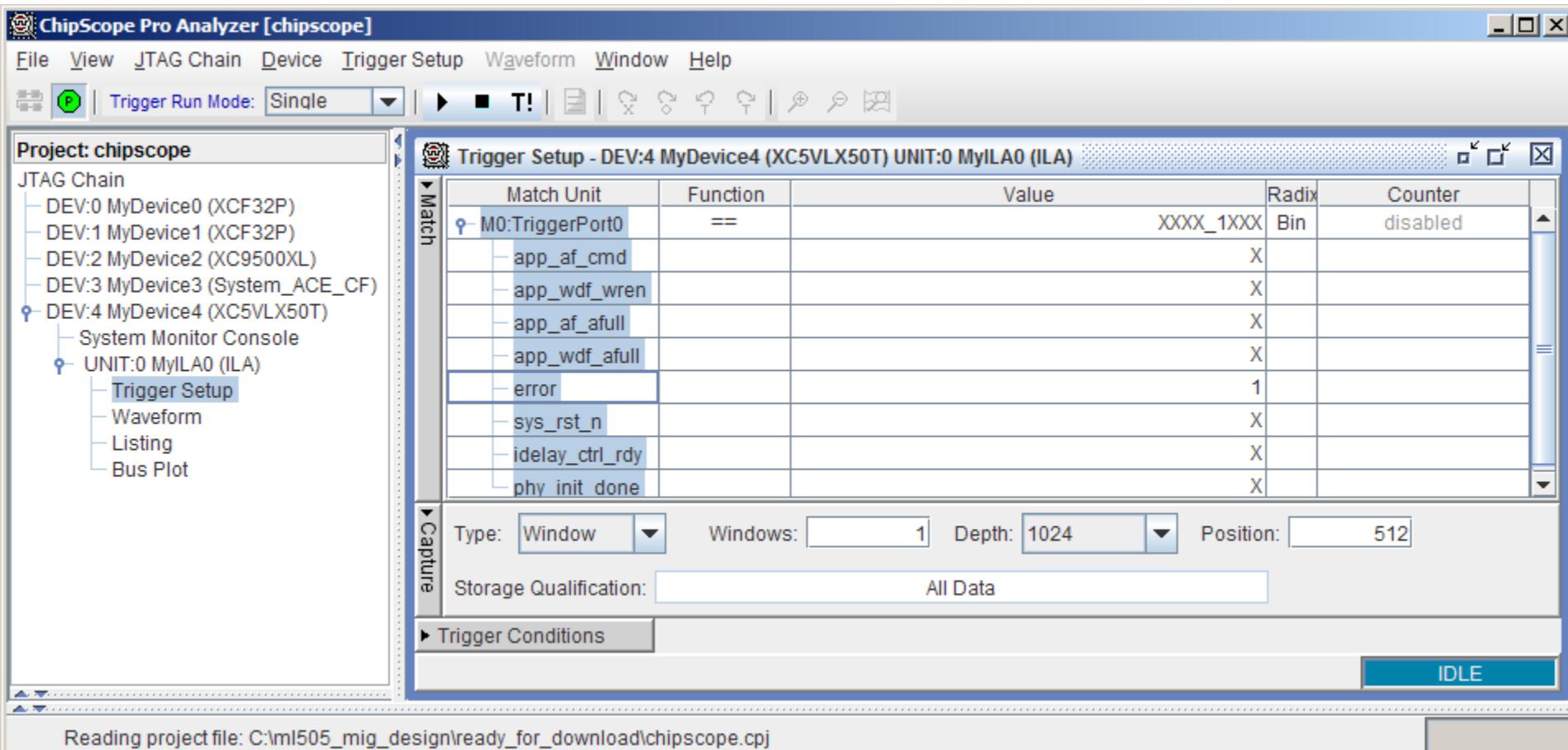
- Select File → Open Project...
- Select <Design Path>\ready\_for\_download\chipscope.cpj



Note: Presentation applies to the ML505, ML506, and ML507

# Run MIG

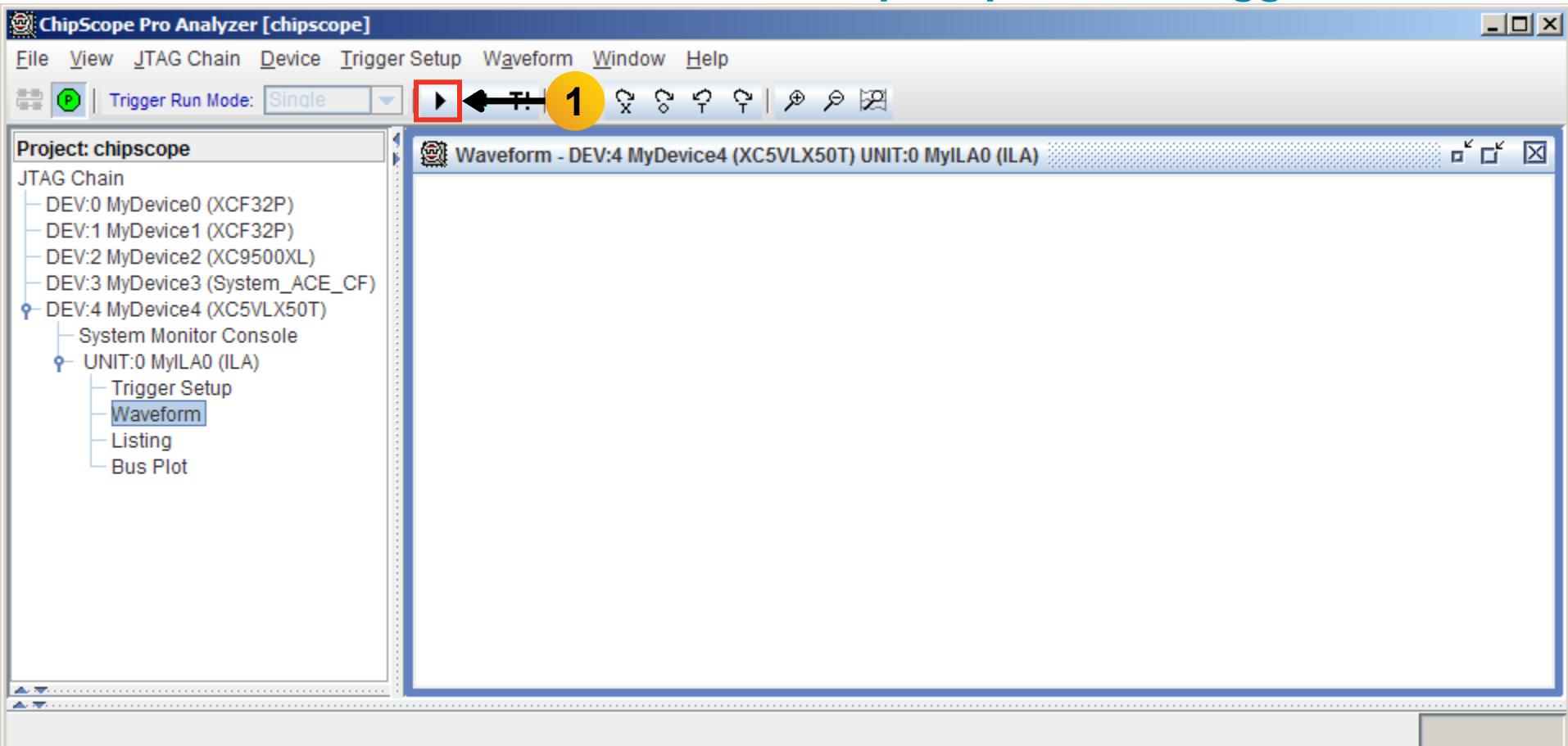
- Click on Trigger Setup to view trigger settings
- The error bit value should be set to 1



Note: Presentation applies to the ML505, ML506, and ML507

# Run MIG

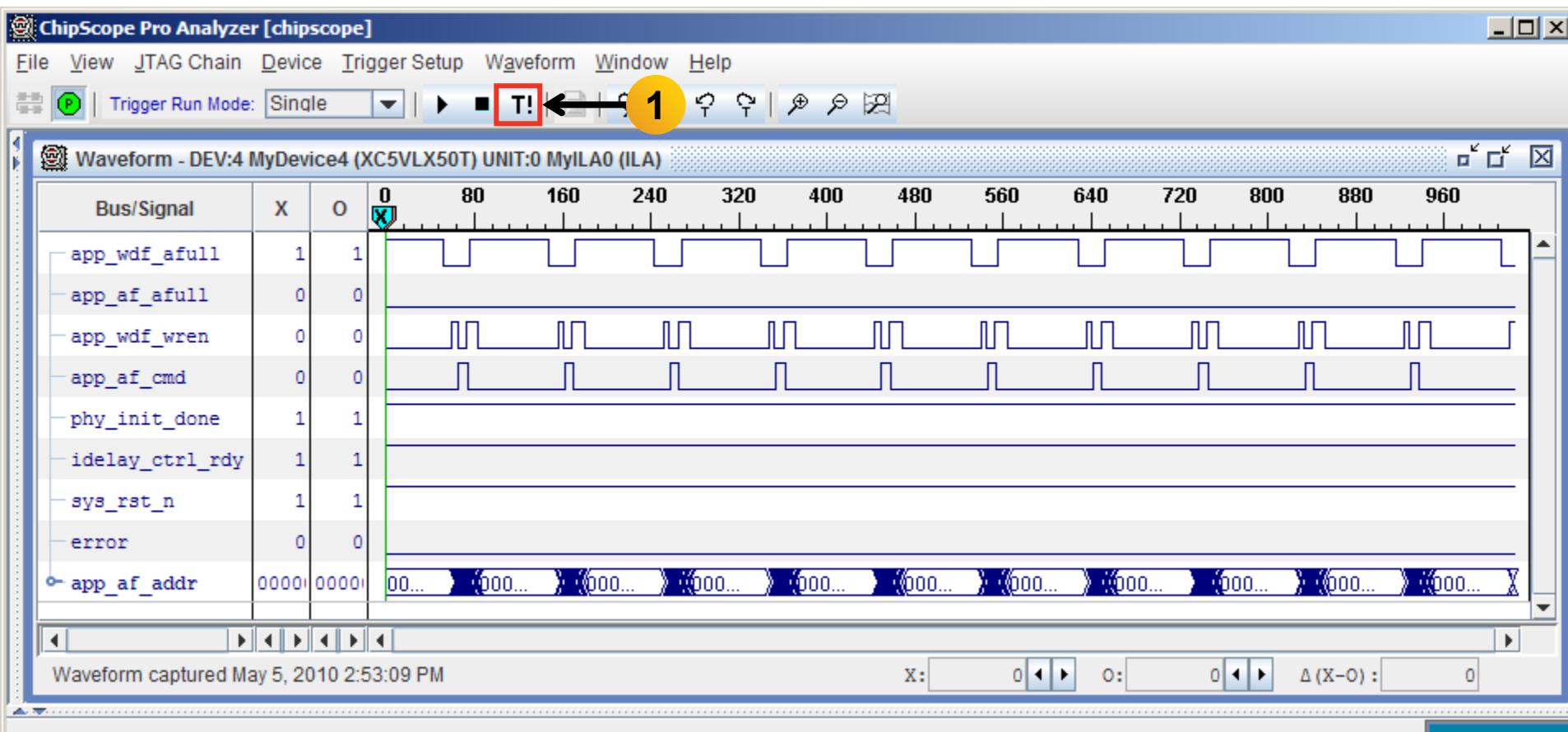
- Click on Waveform; click the Arm Trigger button (1)
- Detection of an error will cause ChipScope Pro to trigger



Note: Presentation applies to the ML505, ML506, and ML507

# Run MIG

- To force a trigger, in order to view the waveform, click the T! button (1)



DONE

# ML505 DDR2 Verification

- **Virtex-5 –1 speed grade device supports 266 MHz DDR2 operation as stated in the [Virtex-5 datasheet](#)**
- **The ML505 SODIMM supports 266 MHz**
  - Supplied DDR2 SODIMM is rated for 266 MHz operation
- **The MIG test design and ChipScope Pro Analyzer verify operation of the ML505 with a Virtex-5 –1 speed grade device, with a memory clock of 200 MHz using the on-board frequency synthesizer**
- **See appendix for 266 MHz operation with user supplied oscillator**

# Appendix

# ML505 DDR2 at 266 MHz

- An external signal generator was used to clock the MIG test design on the ML505 at the maximum clock rate selectable in the MIG tool, 266 MHz
- Supplied DDR2 SODIMM is rated for 266 MHz operation
- The MIG test design and ChipScope Pro Analyzer verify operation of the ML505 at the maximum selectable MIG clock rate of 266 MHz for a -1 speed grade device
  - See [UG086](#)
- If using an external signal generator, connect output to J10 and J11



# References

# Documentation

## ▪ **Memory Solutions**

- Demos on Demand – Memory Interface Solutions with Xilinx FPGAs  
[http://www.demosondemand.com/clients/xilinx/001/page\\_new2/index.asp#35](http://www.demosondemand.com/clients/xilinx/001/page_new2/index.asp#35)
- Xilinx Memory Corner  
[http://www.xilinx.com/products/design\\_resources/mem\\_corner](http://www.xilinx.com/products/design_resources/mem_corner)
- Additional Memory Resources  
<http://www.xilinx.com/support/software/memory/protected/index.htm>
- Xilinx Memory Interface Generator (MIG) User Guide  
[http://www.xilinx.com/support/documentation/ip\\_documentation/ug086.pdf](http://www.xilinx.com/support/documentation/ip_documentation/ug086.pdf)

# References

- **Virtex-5**
  - Virtex-5 FPGA Family  
<http://www.xilinx.com/products/virtex5/index.htm>
- **ChipScope Pro**
  - ChipScope Pro Software and Cores User Guide  
[http://www.xilinx.com/support/documentation/sw\\_manuals/xilinx12\\_1/chipscope\\_pro\\_sw\\_cores\\_ug029.pdf](http://www.xilinx.com/support/documentation/sw_manuals/xilinx12_1/chipscope_pro_sw_cores_ug029.pdf)

# Documentation

# Documentation

- **ML505/506/507 Documentation**

- ML505 Overview

- <http://www.xilinx.com/ml505>

- ML506 Overview

- <http://www.xilinx.com/ml506>

- ML507 Overview

- <http://www.xilinx.com/ml507>

- ML505/506/507 Evaluation Platform User Guide – UG347

- [http://www.xilinx.com/support/documentation/boards\\_and\\_kits/ug347.pdf](http://www.xilinx.com/support/documentation/boards_and_kits/ug347.pdf)

- ML505/506/507 Getting Started Tutorial – UG348

- [http://www.xilinx.com/support/documentation/boards\\_and\\_kits/ug348.pdf](http://www.xilinx.com/support/documentation/boards_and_kits/ug348.pdf)

- ML505/506/507 Reference Design User Guide – UG349

- [http://www.xilinx.com/support/documentation/boards\\_and\\_kits/ug349.pdf](http://www.xilinx.com/support/documentation/boards_and_kits/ug349.pdf)