

Version 1.0 November 30, 2004

This is a list of problems that the developers here have run into and decided to list for the benefit of future users.

1 HDL

- Is the endianness of your signals correct?
- Are the bits that you are reading/writing actually the bits that you think are getting written/read by the other device? Could the actual position of the info you are trying to read/write be shifted over by a few bits? Could it be backwards? Check in Modelsim using radix 2 if you were using radix 16 before, since sometimes that can mask a wrong value.
- Did you declare the correct width for your signals? Make sure you aren't doing things like:

```
wire [0:9] foo;  
wire [0:8] bar;  
...  
assign foo = {bar, 2'd3};
```

- Are different parts of your system coming out of reset at different times? If not, should they be? For example, you probably don't want peripherals to start transmitting data until the DCMs in the system have locked and the FSLs have come out of reset.

2 Simulation

- Is your ISE project configured to generate the simulation language that you think it is supposed to be generating? If not, this can cause weird errors.
- How do I create a testbench without knowing testbench syntax? Use HDL Bencher in ISE. This is described in the ISE QuickStart tutorial.
- How can you debug a design that has to talk to the ZBT? Use Samsung's Verilog behavioural model of the ZBT. This must be compiled in the simulator and not in XST.
- Why can't ModelSim find my Xilinx primitives or CoreGen stuff? When you call vsim, did you use -L with the required library? Unisim is for functional models. XilinxCoreLib is for CoreGen models. Simprim is for Timing models.
- How do I clean up the way the signals are named in the Wave window? I don't want to see the whole path. Save the format as a .do script. Use the ECE532 label.do script on your .do file.
- How do I streamline my simulation flow when I am debugging things repeatedly? Make your own version of the .fdo or .ldo file that the Xilinx tools generate. Have it open your customized .do file in the wave window.

3 Timing

- Is the device connected to the correct clock? Note that the board has 2 27MHz clocks.
- Does your system contain a DCM to lock the clock to the specified frequency? It should.
- Do you need to divide/multiply the clock? You need a DCM to do so. Do not toggle a Flipflop to do so. You can use CoreGen in ISE to create and customize a DCM for your clock quite easily.
- Make sure that the DCM output CLK0 goes through a BUFG before feeding back.

- Are you using the ZBT? You must have a separate DCM generate the clocks from the memory because the memory DCM's feedback goes off chip through the FPGA pins, whereas the internal DCM's feedback travels internally in the FPGA.
- Are you putting the clock through a gate? Don't.

4 ISE

- Are you using the correct device/package/speed grade for the Multimedia board?
- Don't know what one of the error/warning messages mean? Right click on the little red or yellow "WEB" icons in the message window and click to view the Answer Record.
- Can't find the Answer Record for an error/warning message? Try searching on Google. It may have been discussed on one of the fpga news groups somewhere.
- Is your system/peripheral not coming out of reset? If you are using something like

```
internal_rst = FSL1_Rst | FSL2_Rst | ...
```

then make sure that any instances of the FSL have the clock connected. For the asynchronous FSL, at least the master clock must be connected. See the FSL HDL if you want to know why.

- Confused about how some of the Xilinx primitives work exactly? You can find out how it works and how to instantiate it in the Xilinx Libraries Guide at: <Xilinx-installation-dir>/doc/usenglish/books/docs/lib folder.

5 EDK

- Confused about how some of the Xilinx-provided IP works exactly? For IP provided as HDL with EDK, you can find the HDL and read through it at: <EDK-installation-dir>/hw/XilinxProcessorIPLib/pcores. For IP in the Xilinx library that is not available as IP, you can find out how it works and how to instantiate it in the Xilinx Libraries Guide at: <Xilinx-installation-dir>/doc/usenglish/books/docs/lib folder.

6 Other

- During system integration, you can isolate problems to specific modules by creating test cores that interact in a similar manner to the real cores but are very simple, to keep behaviour predictable.