EECS 150 Lab Lecture 3

Ian Juch
Electrical Engineering and Computer Sciences
UC Berkeley
14 September 2012
This week's lab:

List Processor and Chipscope

- Prelab:
  - As always, read the entire lab document
  - Write your verilog ahead of time
  - In particular, read and understand all of section 2, which details the list accumulator you will be implementing
  - Complete the worksheet. It is posted on the website.
This week's lab:

List Processor and Chipscope

- Objectives:
  - implement a relatively complex design using the skills you have gained in previous labs
  - learn to use Chipscope
List Accumulator

- Data and pointer stored in subsequent memory addresses
- Pointer points to next node
- First node always at address 0
- Pointer field of last node is always 0
List Accumulator

The design you will be implementing:

Traversa
Lab Files:

- You should only need to edit two files:
  - Lab3Datapath.v
  - Lab3Control.v
  - These correspond directly to the the List Accumulator schematic in the lab document
- src/blk_ram/small_list.coe
  - contains the list that your list accumulator is processing
Block Ram:

- We will be using built-in block rams on the FPGA
- Configured using Coregen
  - Single/Dual port
  - Different depths/widths
  - Separate clocks
- For this class, we provide most of what you need so you shouldn't need to mess with this
- Just run `./build` in the `blk_ram` directory
Chipscope

- powerful debugging tool (if used correctly)
- synthesizes along with your design allowing you to view signals on the actual FPGA
- should only be used as a last resort
  - ModelSim is still your best friend
- much more difficult to use than simulation because you are very limited in what you can see
  - limited by amount of available RAM on the FPGA
Why we need Chipscope:

- sometimes (often), there will be a discrepancy between simulation and reality
  - Simulation is great for checking that your logic is sound, and often that is good enough
  - Simulation does not model precise timing behavior
  - It is sometimes hard to build a simulation environment that exactly models your fpga setup
  - Chipscope allows you to look at a small subset of your internal signals to look for such behaviors

- It should be used in conjunction with your simulation
Chipscope

ILA (ChipScope Pro - Integrated Logic Analyzer)

Component_Name: chipscope_ila

Trigger Port Settings:
- Number Of Trigger Ports: 1
- Max Sequence Levels: 1
- Use RPMs
- Enable Trigger Output Port

Storage Settings:
- Sample On: Rising
- Sample Data Depth: 1024
- Enable Storage Qualification
- Data Same As Trigger
- Data Port Width: 77 (Range: 1..4096)
Chipscope

ILA (ChipScope Pro - Integrated Logic Analyzer)

Trigger Port 1

- Trigger Port Width: 1
- Match Units: 1
- Counter Width: Disabled
- Match Type: basic with edges
- Bit Values: 0,1,x,r,f,b
- Functions: =,<>

Exclude Trigger Port from Data Storage
Chipscope
Checkoff Requirements

- show correctly functioning circuit on the board
- show organized waveforms in chipscope
Additional Tips:

- Understand the circuit before you start coding the verilog
- Think FSM for the controller
- Make sure you remember to copy over your ALU.v and ALUdec.v and corresponding header files
- Don't forget to generate the core for the block ram
- Chipscope adds significantly to synthesis time, and you have to re-make every time you decide you want to look at a new set of signals, so choose carefully
FAQ

- What? There's checkoff questions?
  - Yes, yes there are. Please read the WHOLE lab document.
FAQ

• How do I know that it's working?
  ○ Set DIP switches 1 and 6 (funct 0'b100001 = add)
  ○ Hit reset (center compass button)
  ○ Center compass LED should be on, signaling that your list accumulator has finished processing the list
  ○ The list sum is 36, so LED's 2 and 5 should be on
FAQ

- So wait, I got my design to work without using chipscope, can I just get checked off?
  - The whole point of this lab was to teach you to use chipscope. You MUST show chipscope waveforms to get checked off!
FAQ

- Why can't we just see all the signals?
  - Chipscope is basically a module that sits inside your design and captures signal values. It uses available block RAM on the FPGA, so it is limited by however much space is available. You can't possibly get the kind of visibility you can get from simulation.
FAQ

• But chipscope is such a pain in the ass, do I really have to use it?
  ○ If your simulation looks good but your design doesn't work, then chipscope is your only remaining tool. It takes some work to setup, but it can actually be a powerful tool.
FAQ

- My simulation passes but it doesn't work on the board, what do I do?
  - Use Chipscope
Project is coming...

- start thinking about who you want to work with
- teams of 2 or less
- lab 4 will have you working in pairs
- more details to come
Other Questions