

Lecture #6: Documentation

Paul Hartke

Phartke@stanford.edu

Stanford EE183

April 22, 2002

Lab #1

- Due Wednesday at midnight
 - You can demo anytime before then
- Late labs will really hurt your grade.
 - It is very important to stay on track with this course. There will be no free late days or extensions given. The late penalty, both for demos and writeups, is 2 points per calendar day late. Labs must be completed, even for 0 points, to finish the course.
- Writeup due next Monday at midnight

Verilog `defines

- What if you wanted to make the FSM one-hot?

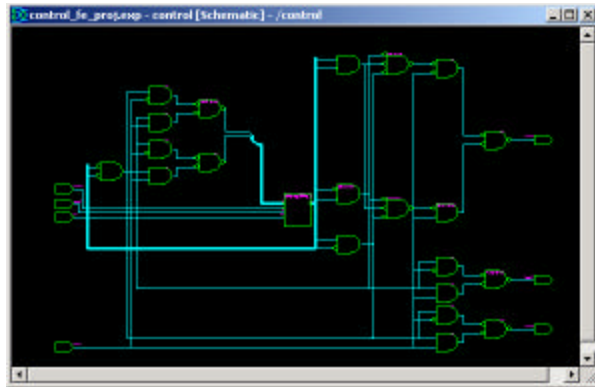
```
// State Machine Declarations...
`define FSM_NUM_DFF 2
`define STATE0 2'b00
`define STATE1 2'b01
`define STATE2 2'b10
`define STATE3 2'b11
reg [^FSM_NUM_DFF-1:0] next_state_d;
wire [^FSM_NUM_DFF-1:0] state_q;
```

Paul's control.v

- Comment your verilog!
 - Make it self-documenting
 - Include a header that says what the block does
- The same CS coding style rules apply
 - Use indentation
 - Don't write obfuscated code unless you need to
 - Remember that a pretty intelligent synthesizer will be optimizing Combinational Logic
 - Does NOT move DFFs to fix timing
 - Suggestion: end all DFF inputs in “_d” and all DFF outputs in “_q”
 - What if I forgot the default in the FSM next state logic?

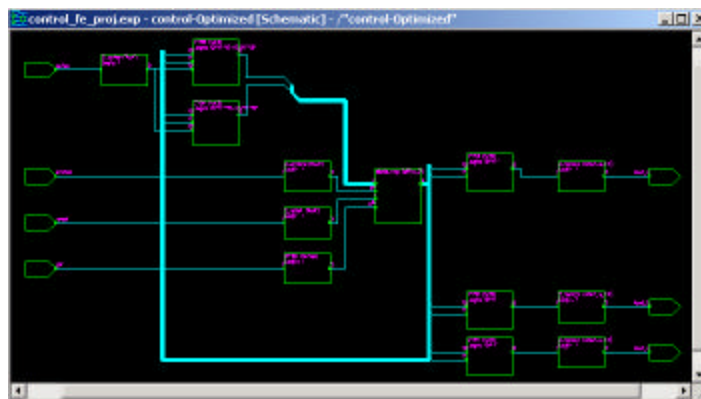
Have a synthesis structure in mind!

- Can check it in FPGA Express...



“Optimized” Schematic

- These are the actual FPGA resources...



Documentation

- What things would a competent engineer in the field need to understand to modify or use this design?
 - More often than not, that engineer is YOU in 12 months.
 - Don't lose the forest in the trees.
 - They can always "Use the Source Luke!" for the trees.
- Incrementally work on the documentation—don't leave it for after the design is complete!!

Requirements

- Overview/Abstract
 - Keep it short and to the point
 - Used to categorize the work
 - "Is this interesting?"
- Description and Block Diagrams
 - Use Drawing Toolbar in MS Word
 - Visio is more powerful
 - This is like the diagrams we drew on the board during the course discussions

Requirements (2)

- Design Aids
 - This is the meat of your write-up. This is what you're going to look at next year when you want to implement this circuit again.
 - This is also the section your boss and employer don't want to read. They're concerned with the big picture. But, you must be able to justify everything you did if asked to.
 - You can treat this section like a journal or notebook that you keep while designing the circuit.
 - Include things like state diagrams, karnaugh maps, equations.
 - Timing diagrams can go here when used to explain how components work, like the one pulse.
 - Diagrams for how to wire up things (like the Sega Genesis controller).

Requirements (3)

- Include sections of the Implementation data that show device utilization and timing
- Include a snapshot of the design in FPGA Editor or the Floorplanner (your choice)
 - Impresses non-technical people! ☺
 - Realize how much work the tools did for you
- Conclusions

Requirements (4)

- Include copy of code in Appendix
 - Should already be commented!!
 - Don't forget the UCF (pin assignments)
- Timing Diagrams/Simulation in 2nd Appendix
 - Include command files
 - Print out “interesting” portions of waveforms

All Electronic

- In general, documents that are not originally in electronic form are almost worthless.
 - Scanning doesn't work too well because nobody does it.
- Create a Word document and then convert to PDF
 - Don't email it but put it on a webpage and email the link.
- We'll do this for Labs 3 and 4
 - That is, we'll only grade a PDF.
 - How could this work for annotated waveforms?