

Lecture #1: Introduction

Paul Hartke

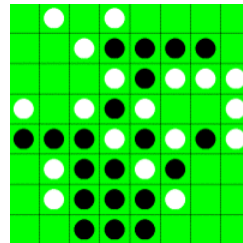
Phartke@stanford.edu

Stanford EE183

April 3, 2002

What is EE183?

- EE183 is continuation of EE121
- Digital Logic Design is a “a minute to learn, a lifetime to master”
 - Othello Board Game slogan
- How do you go about a “million-gate” design?
 - Need good methodologies to handle the complexity
 - Separation of Datapath and Control
 - SoC



EE121 Topics

- Basic Boolean Algebra
 - K-maps, SOP, POS, Hazards
- Basic FPGA Structure
 - logic is cheap (consider 4 input XOR in VLSI) but wires are very expensive.
- Ground Bounce and CV^2f Power
- Combinational Logic Idioms
 - Decoders, [priority] encoders, comparators, basic adder and multiplier architecture
 - Only use power of two dividers (which are equivalent to shifts).

EE121 Topics (cont.)

- **State Machine Design**
 - Why design fully synchronously?
 - “never gate the clock”
 - Mealy versus Moore
 - One-hot versus binary encoding
 - Include synchronous resets and enables
- Sequential Circuit Idioms
 - Counters, shift registers, and LFSRs, edge detectors
 - Why is a ripple counter “bad”?

State Machine Design Process

1. Determination of inputs and outputs.
2. Determination of machine states.
3. Create State/Bubble Diagram—should this be a Mealy or Moore machine?
4. State Assignment—assign each state a particular value.
5. Create Transition/Output Table
6. Derive Next State Logic for each state element—using K-maps as necessary.
7. Derive Output logic.
8. Implement in Xilinx.

EE121 Topics (cont 2.)

- FSM Timing
 - Skew, jitter, H clock distribution tree
 - Max path, min path, critical path
- Metastability, latches and flops
 - Async Input Synchronizer Circuit
- Memory Architectures
 - ROM, SRAM, Dual Port SRAM, DRAM

Differences

- Instead of using Schematic Entry, EE183 is going to use the Verilog Hardware Description Language (HDL).
 - Shallower learning curve than VHDL (the alternative)
 - Verilog is C-like
 - Other stanford classes use it
 - (EE271, 272, 282, 318, 319, 371, etc).
 - More popular in the Valley
 - VHDL is probably better for verification tasks but cycle-accurate model comparison or a special-purpose language (ie, Vera) is used nowadays.

So what are we going to do?

- More practice with digital logic design
 - After completing 183, you will be able to get an entry level chip design job
- If you do not know FSM design like the back of your hand, this is the time to learn it 😊!!!

Course Logistics

- One incomparable TA:
 - David Black-Schaffer (davidbbs@stanford.edu)
- Webpage is <http://www.stanford.edu/class/ee183>
 - Email list is ee183@lists.stanford.edu (you should all be signed up)
 - Newsgroup is su.class.ee183@news.stanford.edu
- You get 24/7 Lab Access to Packard 129 (lab) and Packard 128 (cluster)
 - Can work from home just fine but need network connectivity for license file--240baud modem ok
 - How many have PCs/laptops at home?

Course Logistics (deux)

- Four Labs and one “Quiz”
 - Each lab is half design/getting it to work and half report
 - Quiz is based on interview type questions
 - Not meant to trick.
- 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.

Labs 3 and 4

- Traditionally Labs 3 and 4 have been to pipeline a 5-stage RISC processor and then add some interesting capabilities to it.
 - Does that interest people?
 - EE282 does a very similar project.
 - How many people will be taking EE282?

Verilog Intro

- We will be using fully structural verilog for this course.
 - Hand instantiate all Flip-flips from provided library—ie, no inferred state elements in your logic.
 - Can use Xilinx CoreGen elements
 - the syntax appears in the verilog “Language Assistant”—be sure to include the simulation *.v file in the project.