

CIS 372 Computer Organization and Design Lab

Prof. Amir Roth

Unit 0: Introduction

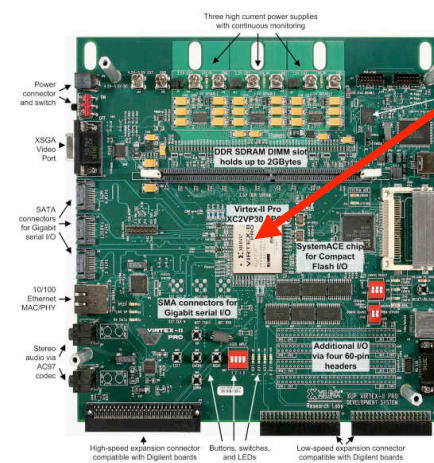
Computer Organization and Design

- CIS 240: computer organization ...
 - How a computer works
 - Hardware: instruction sets, datapath & control, integer arithmetic
 - Software: "assembly language" (LC3 / LC4), C
- CIS 371: ... and design ...
 - How a computer works fast! And using little power!
 - Focus on hardware: pipelining, caches, speculation, wide issue, multiprocessing, multithreading, virtualization, etc.
- CIS 372: ... **lab**
 - Build an LC4 processor! Run programs (maelstrom) on it!

Actual Hardware

- We'll use **FPGAs (Field Programmable Gate Array)**
 - Also called Programmable Logic Devices (PLDs)
- An FPGA is a special type of programmable chip
 - Conceptually, contains a grid of gates
 - The wiring connecting them can be reconfigured electrically
 - Using more transistors as switches
 - Once configured, the FPGA can emulate any digital logic design
 - Synthesis tool converts **gate-level design** to configuration
- Uses
 - **Hardware prototyping** (what "we" are doing)
 - Low-volume special-purpose hardware
 - New: computational offload

In Our Lab: Digilent XUP-V2P Boards

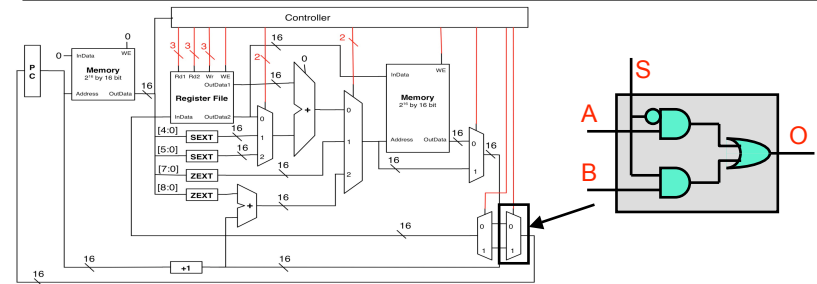


- Program FPGA to run p37x
- "The project"
- Hook up keyboard
- And VGA
- Game on!
- Boards have many features
- Use some for debugging
 - LEDs, switches
- Others, not at all
 - Ethernet, flash reader
 - 256MB SDRAM, audio in/out
 - Can boot Linux!

Hardware Design Methodologies

- How do we get there from here?
- How is hardware designed?
- Many design methodology choices
 - Level: transistors vs. **gates** (no other choice with FPGAs)
 - Layout: manual vs. **synthesized**
 - Tool: schematic vs. **hardware description language (HDL)**

Schematics



- Draw pictures
 - Use a schematic entry program to draw wires, logic blocks, gates
 - Support hierarchical design (arbitrary nesting)
 - + Good match for hardware which is inherently spatial, purty
 - Time consuming, "non-scalable" (large designs are unreadable)
 - Rarely used in practice ("real-world" designs are big)

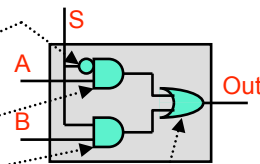
Hardware Description Languages (HDLs)

- Write "code" to describe hardware
 - HDL vs. SDL
 - Specify wires, gates, modules (also hierarchical)
 - + Easier to create, edit, modify, scales well
 - Disconnect: must still "think" visually (gets easier with practice)

```

module mux2to1(S, A, B, Out);
  input S, A, B;
  output Out;
  wire S_, AnS_, BnS;

  not (S_, S);
  and (AnS_, A, S_);
  and (BnS, B, S);
  or (Out, AnS_, BnS);
endmodule
    
```



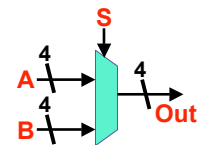
(Hierarchical) HDL Example

- Build up more complex modules using simpler modules
 - Example: 4-bit wide mux from four 1-bit muxes

```

module mux2to1_4(S, A, B, Out);
  input [3:0] A;
  input [3:0] B;
  input S;
  output [3:0] Out;

  mux2to1 mux0 (S, A[0], B[0], Out[0]);
  mux2to1 mux1 (S, A[1], B[1], Out[1]);
  mux2to1 mux2 (S, A[2], B[2], Out[2]);
  mux2to1 mux3 (S, A[3], B[3], Out[3]);
endmodule
    
```



Verilog HDL

- **Verilog**: HDL we will be using
 - Syntactically similar to C (by design)
 - ± Ease of syntax hides fact that this isn't C (or any SDL)
 - We will use a few lectures to learn Verilog

```
module mux2to1_4(S, A, B, Out);  
  input [3:0] A;  
  input [3:0] B; ← These aren't variables  
  input S;  
  output [3:0] Out;  
  
  mux2to1 mux0 (S, A[0], B[0], Out[0]);  
  mux2to1 mux1 (S, A[1], B[1], Out[1]);  
  mux2to1 mux2 (S, A[2], B[2], Out[2]); ← These aren't function calls  
  mux2to1 mux3 (S, A[3], B[3], Out[3]);  
endmodule
```

CSE 372 (Martin/Roth): Introduction

9

Why Study Digital Design?

- Learn by doing
 - If you can build it, you understand it
- Exposure to design issues
 - Lessons learned will be applicable in many domains
 - Testing, design partitioning, bottom-up vs. top-down, etc.
- Get a job
 - Can't live off your parents forever
 - Project experience looks good
- Pretty cool
 - "Look ma, I built my own processor!"
 - "Hey, what's your name? Wanna see my processor?"

CSE 372 (Martin/Roth): Introduction

10

Job Posting Example

- Actual job posting...

Stone Ridge Technology is seeking an Electrical or Computer Engineer with FPGA design experience. The qualified candidate will have experience in FPGA development for complex algorithmic processing. Experience should include FPGA design and implementation with VHDL or Verilog for Xilinx or Altera devices. The candidate should also have simulation tools experience with ModelSim and synthesis tools experience with Synplify or XST. Place & route tool experience with ISE or Quartus is also required. A BS/MS EE with specialization in one or more of the following is desired:

-Computer architectures	Skills: FPGA design, VHDL design for synthesis.	C/C++
-Compilers and operating systems		
-Computer systems: architecture, parallel processing -		

Integrated circuits, VLSI design, testing and cad

Stone Ridge Technology is an early stage technology startup working in the High Performance Computing Industry. The company has clients in both government and industry and is currently working on software applications in quantum chemistry, classical and quantum molecular dynamics, seismic processing, reservoir simulation and computational finance. On the leading edge of HPC, it is developing a proprietary compiler and operating system that enable the use of FPGAs for accelerated general computing. Candidates should be comfortable with an aggressive, high-energy startup environment.

CSE 372 (Martin/Roth): Introduction

11

Bye-bye 372

- This is the last instance of 372 :(
 - Starting 2010, 371/372 combine into a single 1.0 credit course
 - Workload will be reduced proportionately
 - Changes coming in 380/381 also
 - Why? More flexibility
- This is the transition year
 - Somewhat reduced workload
 - 371: 4 fewer lectures
 - 372: 3 labs instead of 4, no individual labs
 - Still 1.5 credits

CSE 372 (Martin/Roth): Introduction

12

The Labs

- Lab 0: due Thurs. 2/5
 - Getting started: tools intro, timer device
- Lab 1: due Thurs. 2/26 (before midterm)
 - Single-cycle LC4 processor: register file, datapath, controller
- Lab 2: due Thurs. 4/23 (last week of class)
 - A: Pipelined LC4 processor + B: bypassing + C: branch prediction
- Tiered grading
 - Labs 0 and 1 are mandatory for a B
 - Lab 2A, 2B, 2C optional for B+, A-, A
- All done in groups of 2
 - Boards and lockers handed out after class next Wednesday 1/28
 - Make sure partner wants to go as far as you do
 - Note: final exam will decouple your grade somewhat

CSE 372 (Martin/Roth): Introduction

13

Lab Logistics

- K-Lab: Moore 204
 - Home of the boards, computers, and later in semester ... you
 - 24 hour access, keycode for door lock
 - TA Office hours, project demos here, too
- Tools
 - Digilent XUP-V2P boards
 - Xilinx ISE v8.2i
 - Warning: all such tools notorious for being buggy and fragile
- Working from home
 - Student version of Xilinx ISE is available
 - ICARUS-0.8.7: fast, but simulation only, and not fully featured
 - All projects must run on the boards in the lab

CSE 372 (Martin/Roth): Introduction

15

372 Lectures

- 5 372 lectures over the course of the semester
 - Stolen from 371 lectures, we will try not to use the Friday slot
 - 1: Verilog (Lab 0 prep)
 - 2: VPI and Verilog simulation
 - 3: FPGAs and synthesis (Lab 1 prep)
 - 4: Design and testing (Lab 2 prep)
 - 5: 372 final exam (40 minutes)
- Not covered
 - Transistor-level design, VLSI, layout, circuit analysis
 - Synthesis algorithms, place & route
 - Embedded systems, hardware/software co-design, ASICs/ASIPs

CSE 372 (Martin/Roth): Introduction

14

Resources

- Instructor
 - Amir Roth (amir@cis)
 - Office: Levine 603, Hours: TR 1:30-3
- TAs: all undergrads from 372 last year
 - Nate (conradj), Michajlo (michajlo), Andres (andres)
 - Office: K-lab, Hours: MRF 3-6:30
- Web
 - Class web page: <http://www.seas.upenn.edu/~cse372/>
 - Google group: <http://groups.google.com/group/upenn-cis372-s09>
- Academic Conduct
 - Anything with your name on it must be your own
 - Penn's code: <http://www.vpul.upenn.edu/osl/acadint.html>

CSE 372 (Martin/Roth): Introduction

16