CSE 141L: Design your own processor

What you’ll do:
- learn Xilinux toolflow
- learn Verilog language
- propose new ISA
- implement it
- optimize it (for FPGA)
- compete with other teams

Grading

15% lab participation – webboard
85% various parts of the labs
CSE 141L: Design your own processor

Teams
- two people
- pick someone with similar goals
- you keep them to the end of the class
- more on the class website:

http://www.cse.ucsd.edu/classes/sp08/cse141L/
Course Staff: 141L

Instructor: Michael Taylor
Email: mbtaylor@cs.ucsd.edu
Office Hours:
EBU 3b 4110
Tuesday 11:30-12:20

TA: Saturnino
Email: sat@cs.ucsd.edu
Lab Hours:
TBA

(141 TA: Kwangyoon)
→ occasional cameos in 141L

http://www-cse.ucsd.edu/classes/sp08/cse141L/
Class Introductions

Stand up & tell us:
- Name
- How long until graduation
- What you want to do when you “hit the big time”
- What kind of thing you find intellectually interesting
What is an FPGA?
Next time: (Tuesday)

Start working on Xilinx assignment (due next Tuesday)
- should be posted

Sat will give a tutorial on Verilog today

Check the website regularly for updates:

http://www.cse.ucsd.edu/classes/sp08/cse141L/
CSE 141:

Computer Architecture

Professor: Michael Taylor

UCSD Department of Computer Science & Engineering

http://www.cse.ucsd.edu/classes/sp08/cse141/
Computer Architecture from 10,000 feet

foo(int x)
{ .. }

Class of application

Physics
Computer Architecture from 10,000 feet

foo(int x)
{
  ... 
}

Class of application

An impossibly large gap!

In the olden days:

“In 1942, just after the United States entered World War II, hundreds of women were employed around the country as computers...” (source: IEEE)
The Great Battles in Computer Architecture Are About How to Refine the Abstraction Layers

```c
foo(int x) {
    ..
}
```

- **Computation**
  - Language
  - Compiler
  - ISA
  - Micro Architecture
  - Register-Transfer Level
  - Circuits
  - Devices
  - Materials Science

- **Physics**
  - Fortran
  - IBM 360, VLIW
  - RISC, T'meta
  - Superscalar, caches
  - Mead & Conway
Abstractions protect us from change -- but must also change as the world changes.

Computation

Language
Compiler
ISA
Micro Architecture
Register-Transfer Level
Circuits
Devices
Materials Science

Changes in fabrication capabilities

Changes in fabrication capabilities

Slower Wires!
Denser VLSI gates!
More pins!
More Power/cm^2!
Abstraction Layers – reflected in organization of research communities

**Computation**
- Language
- Compiler
- ISA
- Micro Architecture
- Reg-Transfer Level
- Circuits
- Devices
- Materials Science

**Physics**
- International Symposium on Computer Architecture (ISCA)
- High Performance Computer Architecture (HPCA)
- Architectural Support for Programming Languages and OS (ASPLOS)
- International Symposium on Microarchitecture (MICRO)
- Design Automation Conference (DAC)
- Int. Conf. Computer Aided Design (ICCAD)
- International Solid State Circuit Conference (ISSCC)
- International Electron Devices Meeting (IEDM)
Classic ISSCC (Circuits) Paper: “How we designed a chip and how fast / low power it is.”

### TABLE IV
**Voltage/Frequency Schmoo**

<table>
<thead>
<tr>
<th>Frequency (GHz)</th>
<th>48C</th>
<th>49C</th>
<th>50C</th>
<th>51C</th>
<th>52C</th>
<th>53C</th>
<th>54C</th>
<th>55C</th>
<th>56C</th>
<th>57C</th>
<th>58C</th>
<th>59C</th>
<th>60C</th>
<th>61C</th>
<th>63C</th>
<th>61C</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2</td>
<td>4W</td>
<td>4W</td>
<td>5W</td>
<td>6W</td>
<td>7W</td>
<td>7W</td>
<td>8W</td>
<td>7W</td>
<td>8W</td>
<td>9W</td>
<td>9W</td>
<td>10W</td>
<td>10W</td>
<td>11W</td>
<td>11W</td>
<td>11W</td>
</tr>
<tr>
<td>2.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** Failed
Classic Int. Electron Device Meeting (IEDM)
Paper: How we designed a single transistor

90 nm Generation Transistor

Nickel Silicide Layer
Silicon Gate Electrode
1.2 nm SiO₂ Gate Oxide
Strained Silicon

No other company combines these transistor features at the 90 nm generation.

Figure 11: 1.2 nm gate oxide time to fail vs. electric field.

Figure 6: NMOS I_{ON} vs. I_{OFF} at 1.0V and 1.2V.
Classic Int. Electron Device Meeting (IEDM)
Paper: “How we designed a wire”

90 nm Generation Interconnects

Low-k CDO Dielectric
Copper Interconnects

Combination of copper + low-k dielectric now meeting performance and manufacturing goals

Intel
The focus of this class

Language
Compiler
ISA
Micro Architecture
Reg-Transfer Level
Circuits
Devices
Materials Science

- International Symposium on Computer Architecture (ISCA)
- High Performance Computer Architecture (HPCA)
- Architectural Support for Programming Languages and OS (ASPLOS)
- International Symposium on Microarchitecture (MICRO)
- Design Automation Conference (DAC)
- Int. Conf. Computer Aided Design (ICCAD)
- International Solid State Circuit Conference (ISSCC)
- International Electron Devices Meeting (IEDM)
Tech Trends

Since technology change is such a big influence in architecture, and because it takes 3-6 years to create a totally new design, we try to predict & exploit it (with varying degrees of success.)
Moore's Law: 2X transistors / “year”

“Cramming More Components onto Integrated Circuits”
- Gordon Moore, Electronics, 1965

# on transistors / cost-effective integrated circuit double every N months (12 ≤ N ≤ 24)

Adapted from Patterson, CSE 252 Sp06 Lecture 2 © 2006 UC Berkeley.
One Important Change: Power
Santa Clara, we have a problem

More pipeline stages, less efficient, more power.

Just can’t remove > 100 watts without great expense on a desktop.

All computing is now
Low Power Computing!
Power Density

Power doubles every 4 years
5-year projection: 200W total, 125 W/cm²!

Change: microprocessor frequency versus time

- 20 yr / 10x (12%)
- 5 yr / 10x (58%)
- 7 yr / 10x (39%)

Faster Circuits, Faster + Smaller Transistors, Fast Microarchitecture

Power Limited

[Graph showing trends over time with Intel x86 data points]
Intel

P3: 12 stages
P4 (b4 paper): 20 stages
P4/prescott: 31 stages
P5/Tejas: >> 31 stages
Intel

P3: 12 stages
P4 (b4 paper): 20 stages
P4/prescott: 31 stages
P5/Tejas: >> 31 stages
Intel

P3: 12 stages
P4 (b4 paper): 20 stages
P4/prescott: 31 stages
P5/Tejas: >> 31 stages
Back to the future

P3:
12 stages

P4 (b4 paper):
20 stages

P4/prescott:
31 stages

P5/Tejas:
>> 31 stages

Same as 1996 – I can’t sell that. I must call it something new --- Pentium...Mmmm... Great Scott, I’ve got it!
And forward to multi-core

Intel Core Duo
Future outlook

Old Trend: Frequency

New Trend: Parallel processing
   ➔ Intel is pushing multi-core instead of higher clocks (will we ever hit 10 GHz?)
   ➔ good time to know something about architecture
   ➔ your application may be feasible only if you can use the architecture efficiently
Abstractions protect us from change -- but must also change as the world changes

Changes in application space

Language
Compiler
ISA
Micro Architecture
Register-Transfer Level
Circuits
Devices
Materials Science

Physics

Virtual Homicide (Quake)
Photographic memory
Telepathic
Mathematical Genius
Etc...
And on that note: PC's are not the only important class of computer - in fact they are in the minority (~2%)!
Administrative Details

***************
Course Work and Grading

• Grading
  - 12.5% homework
  - 12.5% class participation
  - 20% midterm
  - 20% midterm
  - 35% final (which will cover the whole quarter)
• Tests
  • Closed books and no notes

http://www-cse.ucsd.edu/classes/sp08/cse141/
Textbook

Patterson & Hennessy, third edition of "Computer Organization, the Hardware/Software Interface"

- Decent book. We'll read most of it.
  - 3rd Edition, came out this year, look for errata
- Patterson is professor at Berkeley;
  - lead RISC project (foundation of SPARC processor)
  - lead RAID (redundant array of inexpensive disks) project
- Hennessy is professor at Stanford
  - now President of Stanford
  - co-founded of MIPS Computer Systems
- Note: same authors wrote the graduate textbook, "Computer Architecture, A Quantitative Approach".
Text vs. Lectures in CSE 141

Assigned readings for each lecture posted on website!

- Lectures will include material not in the text…text will include material not in the lectures.

- Resource limitations prevent us from addressing material from the prerequisites in office hours…but we are happy to refer you to the book or your classmates.
How to find out your deliverables

• Check the website. Generally, we won’t necessarily announce readings or assignment due dates in lecture.

http://www-cse.ucsd.edu/classes/sp08/cse141/

• You will have assigned reading for every lecture except when you have an exam.

Please watch the website for course updates, reading assignments and homework assignments!
Course Policies

• You may request a regrade for your exams. To do so, you must submit a written description of the mistake we made to the TA within 1 week of the return of the exam. We reserve the right to regrade the entire exam; so your grade may go up or down depending.

• No extensions or regrades for homework. HWs are each worth less than 2% of your grade.

• For those of you taking 141L, superior participation in 141L will contribute to your 141 class participation grade.

http://www-cse.ucsd.edu/classes/sp07/cse141/
Course Staff

Instructor: Michael Taylor
Email: mbtaylor@cs.ucsd.edu
Office Hours:
EBU 3b 4110
Tuesday 11:30-12:20 pm (right before 141)

TA: Kwangyoon Lee
Email: kwl002@cs.ucsd.edu
Office Hours:

Discussion Section:
M 11-11:50am, peterson 103

http://www-cse.ucsd.edu/classes/sp08/cse141/
Question Triage:  
Who to ask which Question

Me:
“In lecture, ...”
“I’m designing my own supercomputer, and...”

Kwangyoon:
“On problem 5, ...”

Sat:
“In the Xilinx 9.1 environment, ...”

Me+Kwangyoon:
“In a 2-way set associative cache...”
“In the book, ...”

Me+Sat:
“In my 141L ISA, ...”

.. and of course, talk with your classmates!!
Am I Qualified to Teach You?

- PhD, MIT, EE & CS
  - ten years at MIT studying processor design
  - 2 years consulting for chip companies
  - various research publications

Architectures designed: 4
Machines Implemented: 3
Millions of units of software shipped: > 1
Million-gate chips designed: 1
Supercomputers designed: 1
About Me

PowerMush 3
PowerMush IV
About Me

~120 million transistors
Course Outline

1. Instruction Set Architecture
2. Performance Metrics
3. CPU Organization
4. Pipelining
5. The Memory/Cache Hierarchy
6. Multiprocessors (time permitting)

Please watch the website for course updates, reading assignments and homework assignments!
Chapter 1
The Overview
(read it tonight)
What is Computer Architecture?

Computer Architecture =
Machine Organization +
Instruction Set Architecture
How to Speak Computer

High Level Language Program

Compiler

Assembly Language Program

Assembler

Machine Language Program

ISA

Machine Interpretation

Control Signal Spec

temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;

lw $15, 0($2)
lw $16, 4($2)
sw $16, 0($2)
sw $15, 4($2)

10001100011000100000000000000000
10001100111100100000000000010010
101011000110001000000000100

100011000110001000000000000000000000000000100

100011000110001000000000000000000000000000100
10001100111100100000000000010010
101011000110001000000000100

1010110001100010000000000000000000100

ALUOP[0:3] <= InstReg[9:11] & MASK
A Review: Java on the Computer

- Java Language Program
- Compiler
- Machine Language Program
- ISA
- Control Signal Spec
- Machine Interpretation
Java vs. Byte code vs. Assembly (almost machine lang)

```java
void spin()
{
    int i;
    for (i = 0; i < 100; i++)
    {
        ; // Loop body is empty
    }
}
```

```
0       iconst_0   // Push int constant 0
1       istore_0  // Store into local 0 (i=0)
2       goto 8    // First time through don't increment
5       iinc 0 1  // Increment local 0 by 1 (i++)
8       iload_0   // Push local 0 (i)
9       bipush 100 // Push int constant (100)
11      if_icmplt 5 // Compare, loop if < (i < 100)
14      return    // Return void when done
```
Java vs. Byte code vs. Assembly (almost machine lang)

dump: /* bytecode plus x86 assembly */
// BB#1
  0: iconst_0
  1: istore_0
   0x8440fa1 xor ebx -> ebx 33 db
   0x8440fa3 jmp 0x8440fb1 e9 09 00 00 00
  2: goto 8
// BB#5
  5: iinc 0 1
   0x8440fb0 inc ebx 43
// BB#2
  8: iload_0
  9: bipush 100
 11: if_icmplt 5
   0x8440fb1 cmp 100 -> ebx 83 fb 64
   0x8440fb4 jl 0x8440fb0 7c fa
// BB#3
 14: return
// BB#4 (* epilog *)
  0x8440fb6 pop ebx 5b
  0x8440fb7 ret c3
End of Method test1.spin()V

Byte Code

Assembly on an x86
The Instruction Set Architecture (ISA)

- that part of the architecture that is visible to the programmer
  - opcodes (available instructions)
  - number and types of registers
  - instruction formats
  - storage access, addressing modes
  - exceptional conditions

- It is NOT QUITE assembly language: but usually DARN close.
The Instruction Set Architecture

° is the agreed-upon interface between all the software that runs on the machine and the hardware that executes it.
Examples of ISAs

- Alpha AXP
- Intel 80x86/pentium (called x86)
- VAX
- MIPS
- SPARC
- IBM 360
- Intel IA-64 (Itanium or EPIC)
- PowerPC
Computer Organization

- Once you have decided on an ISA, you must decide how to **design the hardware** to execute those programs written in the ISA as fast as possible.
  - There are MANY different **hardware** designs possible for a given ISA.
- Hardware implementation can vary based on cost or power budget:
  - For reduced cost (low marginal cost)
    - $200 Walmart computer
  - For servers (high marginal cost)
    - Opteron / Xeon
  - For reduced power requirements
    - Transmeta’s Crusoe processor, Mobile Pentium
  - Reliability (spacecraft, 5-9’s mainframes)
    - “Rad-hardened”, ECC and/or TMR
The Challenge of Computer Architecture

• The industry changes faster than any other.
• The ground rules change every year.
  - new problems (e.g., power)
  - new opportunities
  - different tradeoffs
• It’s all about making programs run faster than the next guy’s machine can run them.
Starting in the 1940’s, had the transportation industry kept pace with computer performance increases, today you could travel to the east coast in 5 seconds at a cost of 50 cents.
The five classic components of computers:
All “computers” can fit into this general classification
Disks & Tape

Also considered I/O devices

Fine print: portion of disk used as “virtual memory” could be called “memory”.

- **Hard disks** (magnetic surface on metal)
  - Very slow access time (~ 5 ms)
  - Getting inexpensive very fast

- **Floppy disks** (magnetic surface on mylar)
  - Cheap and convenient

- **CD’s** (compact disks) - optical
  - Even cheaper
  - Slow (or impossible) to write

- **Magnetic tape**

- **Flash**

“secondary storage”

“tertiary storage”
Memory: How we use it

- 2 Volatile Types:
  - Main Memory
    - Where program (and some data) is stored
    - When you turn off the power it’s gone
    - Made of DRAM (usually)
  - Cache
    - A small, fast memory
    - Is a “buffer” for MM
      - Keeps a portion of MM available FASTER
      - Made of SRAM or SDRAM
Why care about power consumption?

• California’s energy crisis??
  - Maybe - google cares

• Heat is hard to get rid of!
  - Workstation processor might use 100 Watts
  - Limits how densely components can be packaged

• Battery power is limited!
  - Embedded processors in portable devices
Real Stuff: Chip Manufacturing

![Diagram of chip manufacturing process]

- **Silicon ingot** → **Slicer** → **Blank wafers** → **20 to 30 processing steps** → **Patterned wafers** → **Dicer** → **Individual dies (one wafer)** → **Die tester** → **Tested dies** → **Bond die to package** → **Packaged dies** → **Part tester** → **Tested packaged dies** → **Ship to customers**

**Cost per die formula:**

\[
\text{cost per die} = \frac{\text{cost per wafer}}{\text{dies per wafer}} \times \text{yield}
\]

\[
\text{dies per wafer} = \frac{\text{wafer area}}{\text{die area}}
\]

\[
\text{yield} = \frac{1}{1 + \left(\frac{\text{defects per area} \times \text{die area}}{2}\right)}^2
\]

**Typical desktop processor = 1 cm^2 of silicon → about $10 cost to fabricate**
Interesting Stuff in Book Not Covered in Lecture

- Disks
- Networks
- CRTs and displays
- Mice
- Embedded processors
Key Points

• The instruction set architecture defines how software is allowed to use the processor. Multiple computers with vastly different organizations and performance can share an ISA.

• Most every component in a computer system falls into one of five categories.
What you can expect to get out of this class

• Understand fundamental concepts in computer architecture and how they impact computer and application performance.
• Evaluate architectural descriptions of even today’s most complex processors.
• Gain experience designing a working CPU completely from scratch (with 141L).
• Learn experimental techniques used to evaluate advanced architecture design
  - you will be primed to work on architecture problems
  - or other areas that benefit from architecture knowledge
Remember:

- Check the website. Generally, we won’t necessarily announce readings or assignment due dates in lecture.

http://www-cse.ucsd.edu/classes/sp08/cse141/

- You will have assigned reading for every lecture except when you have an exam.

Please watch the website for course updates, reading assignments and homework assignments!