CS 61C: Great Ideas in Computer Architecture (Machine Structures)

Instructors:
Randy H. Katz
David A. Patterson
http://inst.eecs.Berkeley.edu/~cs61c/fa10

Agenda

- Technology Trends Revisited
- Administrivia
- Technology Break
- Components of a Computer
## Agenda

- Technology Trends Revisited
- Administrivia
- Technology Break
- Components of a Computer

### Levels of Representation/Interpretation

<table>
<thead>
<tr>
<th>High Level Language Program (e.g., C)</th>
<th>Compiler</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assembly Language Program (e.g., MIPS)</td>
<td>Assembly</td>
</tr>
<tr>
<td>Machine Language Program (MIPS)</td>
<td>Assembler</td>
</tr>
</tbody>
</table>

- **Compiler**
  - `temp = v[k];
  - `v[k] = v[k+1];`  
  - `v[k+1] = temp;`  

- **Assembler**
  - `lw $t0, 0($2)`
  - `lw $t1, 4($2)`
  - `sw $t1, 0($2)`
  - `sw $t0, 4($2)`

- **Machine Language Program (MIPS)**

- **Machine Interpretation**

- **Hardware Architecture Description (e.g., block diagrams)**

- **Architecture Implementation**

- **Logic Circuit Description (Circuit Schematic Diagrams)**

- **Anything can be represented as a number, i.e., data or instructions**

- **Binary Code Examples**
  - 0000 1001 1100 0110 1010 1111 0101 1000 1100 0110 1010 1111 0101 1000 0000 1001 0101 1000 0000 1001 0101 1000 0110 1010 1111

- **Logic Gates**

- **Register File**

- **ALU**
Technology Life Cycle: Where Are We?

The life and times of a technology
Recurring phases of each great surge

Source: Carleta Perez
Technology Cost over Time

Technology Cost over Time
Successive Generations
Moore’s Law

Predicts: 2X Transistors / chip every 2 years

# of transistors on integrated circuit (IC)

Year


Gordon Moore
Intel Cofounder
B.S. Cal 1950!

Memory Chip Size

Growth in memory capacity slowing

Fall 2010 — Lecture #8

9/15/10
Technology Trends: Uniprocessor Performance (SPECint)

Improvements in processor performance have slowed. Why?

Limits to Performance: Faster Means More Power

\[ P = CV^2f \]
P = C V^2 f

- What is the effect on power consumption of:
  - “Simpler” implementation (fewer transistors)?
  - Smaller implementation (shrunk down design)?
  - Reduced voltage?
  - Increased clock frequency?

Doing Nothing Well—NOT!

- Traditional processors consume about two thirds as much power at idle (doing nothing) as they do at peak
- Higher performance (server class) processors approaching 300 W at peak
- Implications for battery life?
Limits to Performance: Processor-Storage Performance Gap

The Real Moore's Law:

- CPU speed outpaces RAM speed
- Disk speed lags behind CPU and RAM

Disk Drive Capacity/Performance Over Time

Historical Trends of CPU and IOPS performance

- CPU MHz increases over time
- HDD IOPS also increases, but at a slower rate
Computer Technology: Growing, But More Slowly

• Processor
  – Speed 2x / 1.5 years (since ’85) \([slowing!]\)
  – 100X performance last decade
  – When you graduate: 4 GHz, 32 Cores

• Memory (DRAM)
  – Capacity: 2x / 2 years (since ’96) \([slowing!]\)
  – 64x size last decade
  – When you graduate: 128 GBytes

• Disk
  – Capacity: 2x / 1 year (since ’97)
  – 250X size last decade
  – When you graduate: 8 Tbytes

• Network
  – Core: 2x every 2 years
  – Access: 100-1000 mbps from home, 1-10 mbps cellular

Agenda

• Components of a Computer
• Administrivia
• Technology Break
Announcements

• Notes: They are on line—use them!
• First HW solutions posted—see Class Web
• Experiment: move HW/Project due dates to Saturday from Friday for next three assignments (Project 1, Parts 1 & 2; HW #3)
• Exam: 6 October, 6-9 PM, 1 Pimentel
• Free book on Warehouse-Scale Computers—on the course web site

Announcements

• Grading
  — Class Participation (5%)
  — Homework (5%)
  — Labs (20%)
  — Projects (40%)
    • Computer Instruction Set Simulator
    • Data Parallelism (Map-Reduce on EC2)
    • Computer Processor Design (Logisim)
    • Performance Tuning of a Parallel Application (partnered)
  — Midterm (10%)
  — Final (20%)
Agenda

- Technology Trends Revisited
- Administrivia
- Technology Break
- Components of a Computer

Five Components of a Computer

- Control
- Datapath
- Memory
- Input
- Output

John von Neumann invented this architecture.
Reality Check: Typical MIPS Chip Die Photograph

Example MIPS Block Diagram

9/15/10
### The CPU

- **Processor (CPU):** the active part of the computer, which does all the work (data manipulation and decision-making)
- **Datapath:** portion of the processor which contains hardware necessary to perform operations required by the processor (the brawn)
- **Control:** portion of the processor (also in hardware) which tells the datapath what needs to be done (the brain)
Stages of the Datapath: Overview

- Problem: a single, atomic block which “executes an instruction” (performs all necessary operations beginning with fetching the instruction) would be too bulky and inefficient
- Solution: break up the process of “executing an instruction” into stages or phases, and then connect the phases to create the whole datapath
  - Smaller phases are easier to design
  - Easy to optimize (change) one phase without touching the others

Instruction Level Parallelism

<table>
<thead>
<tr>
<th>Instr 1</th>
<th>Instr 2</th>
<th>Instr 3</th>
<th>Instr 4</th>
<th>Instr 5</th>
<th>Instr 6</th>
<th>Instr 7</th>
<th>Instr 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>P 1</td>
<td>P 2</td>
<td>P 3</td>
<td>P 4</td>
<td>P 5</td>
<td>P 6</td>
<td>P 7</td>
<td>P 8</td>
</tr>
<tr>
<td>IF</td>
<td>ID</td>
<td>ALU</td>
<td>MEM</td>
<td>WR</td>
<td>ID</td>
<td>ALU</td>
<td>MEM</td>
</tr>
<tr>
<td>P 9</td>
<td>P 10</td>
<td>P 11</td>
<td>P 12</td>
<td>P 1</td>
<td>P 2</td>
<td>P 3</td>
<td>P 4</td>
</tr>
<tr>
<td>IF</td>
<td>ID</td>
<td>ALU</td>
<td>MEM</td>
<td>WR</td>
<td>IF</td>
<td>ID</td>
<td>ALU</td>
</tr>
<tr>
<td>MEM</td>
<td>WR</td>
<td>IF</td>
<td>ID</td>
<td>ALU</td>
<td>MEM</td>
<td>WR</td>
<td>IF</td>
</tr>
<tr>
<td>WR</td>
<td>IF</td>
<td>ID</td>
<td>ALU</td>
<td>MEM</td>
<td>WR</td>
<td>IF</td>
<td>ID</td>
</tr>
</tbody>
</table>
Stages of the Datapath (1/5)

- There is a wide variety of MIPS instructions: so what general steps do they have in common?
- **Stage 1: Instruction Fetch**
  - No matter what the instruction, the 32-bit instruction word must first be fetched from memory (the cache-memory hierarchy)
  - Also, this is where we Increment PC (that is, PC = PC + 4, to point to the next instruction: byte addressing so + 4)

Stages of the Datapath (2/5)

- **Stage 2: Instruction Decode**
  - Upon fetching the instruction, we next gather data from the fields (decode all necessary instruction data)
  - First, read the opcode to determine instruction type and field lengths
  - Second, read in data from all necessary registers
    - For add, read two registers
    - For addi, read one register
    - For jal, no reads necessary
Stages of the Datapath (3/5)

- Stage 3: ALU (Arithmetic-Logic Unit)
  - Real work of most instructions is done here: arithmetic (+, -, *, /), shifting, logic (&, |), comparisons (slt)
  - What about loads and stores?
    - lw $t0, 40($t1)
    - Address we are accessing in memory = the value in $t1 PLUS the value 40
    - So we do this addition in this stage

Stages of the Datapath (4/5)

- Stage 4: Memory Access
  - Actually only the load and store instructions do anything during this phase; the others remain idle during this phase or skip it all together
  - Since these instructions have a unique step, we need this extra phase to account for them
  - As a result of the cache system, this phase is expected to be fast
Stages of the Datapath (5/5)

• Stage 5: Register Write
  – Most instructions write the result of some computation into a register
  – E.g.,: arithmetic, logical, shifts, loads, slt
  – What about stores, branches, jumps?
    • Don’t write anything into a register at the end
    • These remain idle during this fifth phase or skip it all together
The ARM Inside the iPhone

ARM Architecture

iPhone Innards

Processor Families:
Tradeoff between Cost and Performance
Summary

• Key Technology Trends and Limitations
  – Transistor doubling BUT power constraints and latency considerations limit performance improvement
  – (Single Processor) computers are about as fast as they are likely to get, exploit parallelism to go faster

• Five Components of a Computer
  – Processor/Control + Datapath
  – Memory
  – Input/Output: Human interface/KB + Mouse, Display, Storage ... evolving to speech, audio, video

• Architectural Family: One Instruction Set, Many Implementations