CS152
Computer Architecture and Engineering
Lecture 1

Introduction and Five Components of a Computer

January 21, 2004

John Kubiatowicz (www.cs.berkeley.edu/~kubitron)

lecture slides: http://inst.eecs.berkeley.edu/~cs152/

Overview

- Intro to Computer Architecture (30 minutes)
- Administrative Matters (5 minutes)
- Course Style, Philosophy and Structure (15 min)
- Break (5 min)
- Organization and Anatomy of a Computer (25) min)

What is "Computer Architecture"

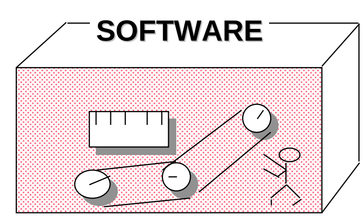
Computer Architecture =

Instruction Set Architecture +

Machine Organization + …..

Instruction Set Architecture (subset of Computer Arch.)

... the attributes of a [computing] system as seen by the programmer, *i.e.* the conceptual structure and functional behavior, as distinct from the organization of the data flows and controls the logic design, and the physical implementation.
– Amdahl, Blaaw, and Brooks, 1964

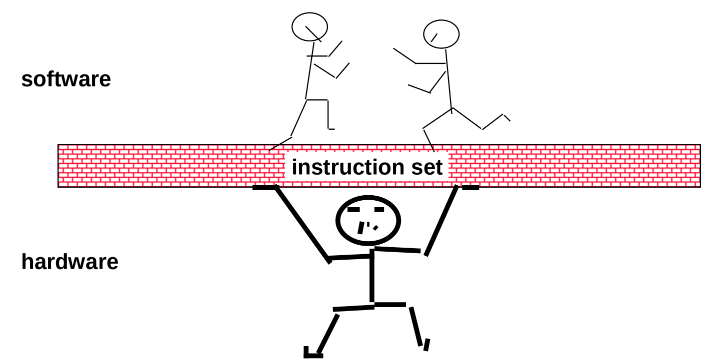


- -- Organization of Programmable Storage
- -- Data Types & Data Structures: Encodings & Representations
- -- Instruction Set
- -- Instruction Formats
- -- Modes of Addressing and Accessing Data Items and Instructions
- -- Exceptional Conditions

Computer Architecture's Changing Definition

- 1950s to 1960s: Computer Architecture Course: Computer Arithmetic
- 1970s to mid 1980s: Computer Architecture Course: Instruction Set Design, especially ISA appropriate for compilers
- 1990s: Computer Architecture Course: Design of CPU, memory system, I/O system, Multiprocessors, Networks
- 2010s: Computer Architecture Course: Self adapting systems? Self organizing structures? DNA Systems/Quantum Computing?

The Instruction Set: a Critical Interface



Example ISAs (Instruction Set Architectures)

Digital Alpha	(v1, v3)	1992-97
HP PA-RISC	(v1.1, v2.0)	1986-96
Sun Sparc	(v8, v9)	1987-95
SGI MIPS	(MIPS I, II, III, IV, V)	1986-96
Intel	(8086,80286,80386, 80486,Pentium, MMX, ...)	1978-96

MIPS R3000 Instruction Set Architecture (Summary)

- Instruction Categories
  - Load/Store
  - Computational
  - Jump and Branch
  - Floating Point
    - coprocessor
  - Memory Management
  - Special

Registers

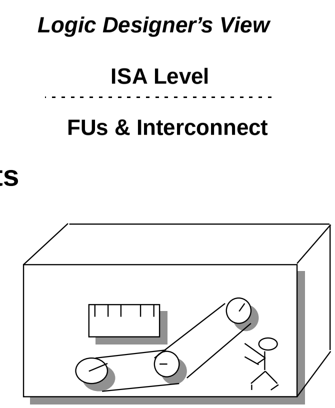
R0 - R31
PC
HI
LO

3 Instruction Formats: all 32 bits wide

OP	rs	rt	rd	sa	funct
OP	rs	rt	immediate		
OP	jump target				

Q: How many already familiar with MIPS ISA?

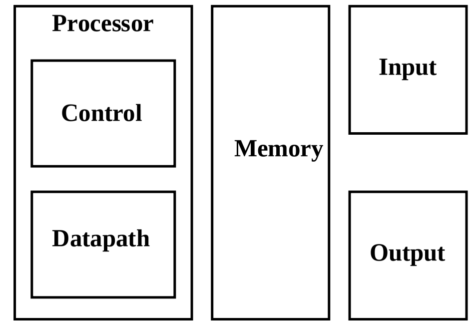
Organization



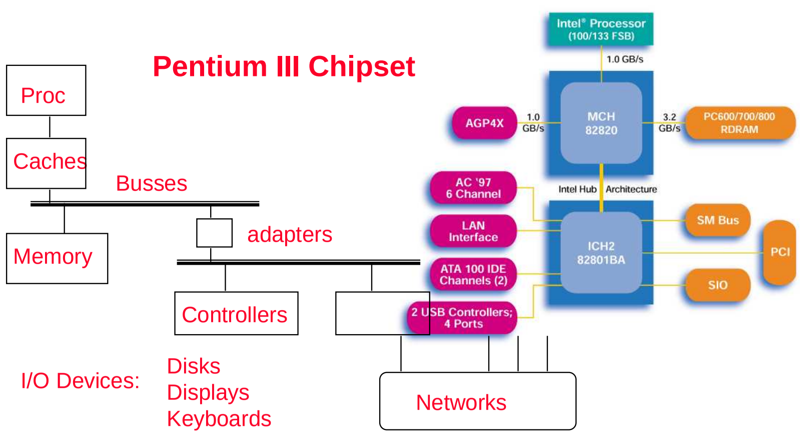
- **Capabilities & Performance Characteristics of Principal Functional Units**
  - **(e.g., Registers, ALU, Shifters, Logic Units, ...)**
- **Ways in which these components are interconnected**
- **Information flows between components**
- **Logic and means by which such information flow is controlled.**
- **Choreography of FUs to realize the ISA**
- **Register Transfer Level (RTL) Description**

The Big Picture

- Since 1946 all computers have had 5 components

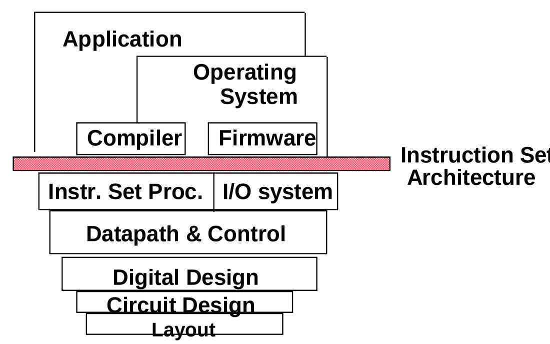


Sample Organization: It's all about communication



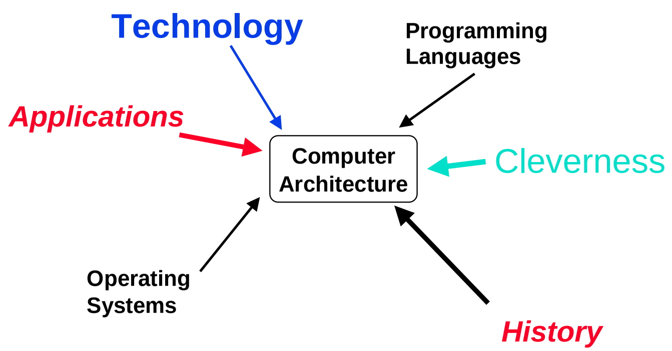
- **All have interfaces & organizations**
- **Um…. It's the network stupid???!**

What is "Computer Architecture"?



- **Coordination of many *levels of abstraction***
- **Under a rapidly changing set of forces**
- **Design, Measurement, *and* Evaluation**

Forces on Computer Architecture

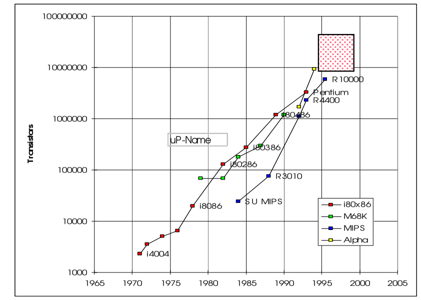


Technology

DRAM chip capacity

DRAM	
Year	Size
1980	64 Kb
1983	256 Kb
1986	1 Mb
1989	4 Mb
1992	16 Mb
1996	64 Mb
1999	256 Mb
2002	1 Gb

Microprocessor Logic Density



° **In ~1985 the single-chip processor (32-bit) and the single-board computer emerged**
  - **=> workstations, personal computers, multiprocessors have been riding this wave since**

° **In the 2002+ timeframe, these may well look like mainframes compared single-chip computer (maybe 2 chips)**

Technology => dramatic change

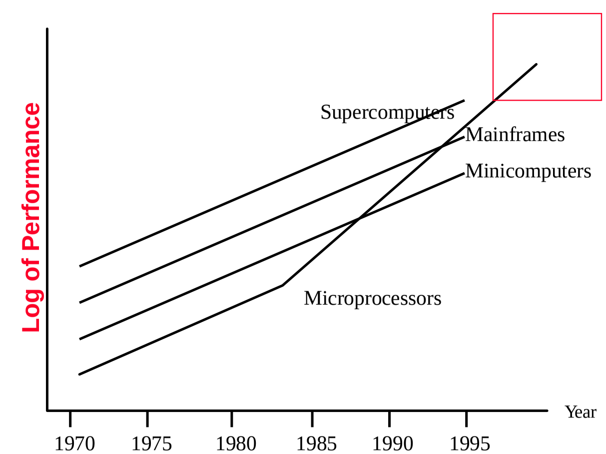
° **Processor**
  - **logic capacity: about 30% per year**
  - **clock rate: about 20% per year**

° **Memory**
  - **DRAM capacity: about 60% per year (4x every 3 years)**
  - **Memory speed: about 10% per year**
  - **Cost per bit: improves about 25% per year**

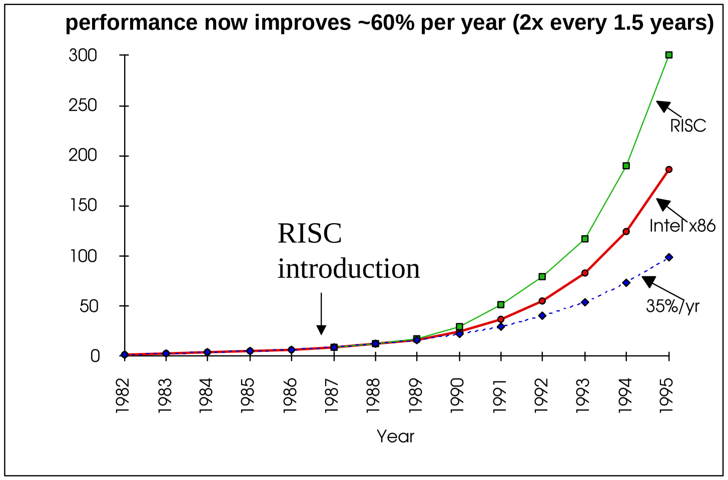
° **Disk**
  - **capacity: about 60% per year**
  - **Total use of data: 100% per 9 months!**

° **Network Bandwidth**
  - **Bandwidth increasing more than 100% per year!**

Performance Trends



Processor Performance (SPEC)

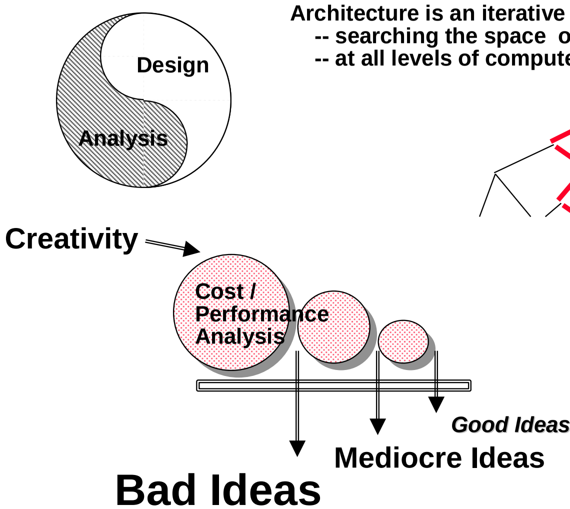


Did RISC win the technology battle and lose the market war?

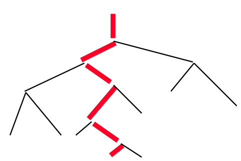
Applications and Languages

- CAD, CAM, CAE, . . .
- Lotus, DOS, . . .
- Multimedia, . . .
- The Web, . . .
- JAVA, . . .
- The Net => ubiquitous computing
- ???

Measurement and Evaluation



Architecture is an iterative process
-- searching the space of possible designs
-- at all levels of computer systems

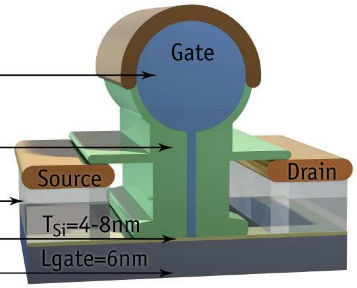


Why do Computer Architecture?

- **CHANGE**
- **It's exciting!**
- # It has never been more exciting!
- **It impacts every other aspect of electrical engineering and computer science**

Computers in the News: New IBM Transistor

Region	Material
Gate:	Polysilicon
Spacers:	Dielectric
RSD:	Silicon
Channel:	Silicon
Box:	Oxide

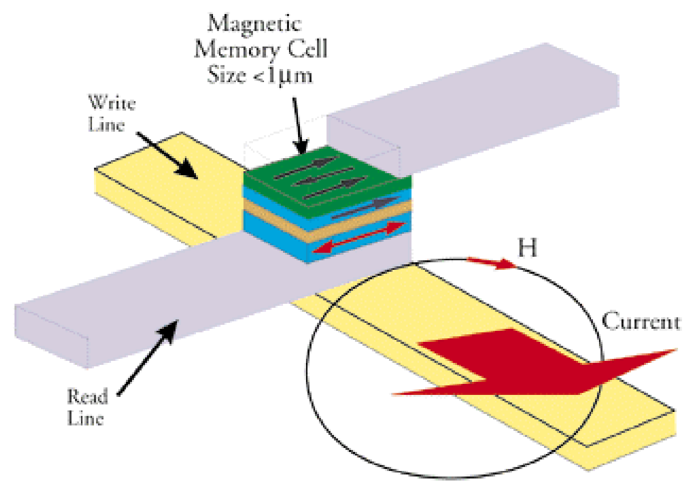


RSD stands for "Raised Source Drain"

Box stands for "Buried Oxide"

- ° **Announced 12/10/02**
- ° **6nm gate length!!!**
- ° **Details: Still to be announced**

Computers in the news: Tunneling Magnetic Junction



Computers in the News: Sony Playstation 2000

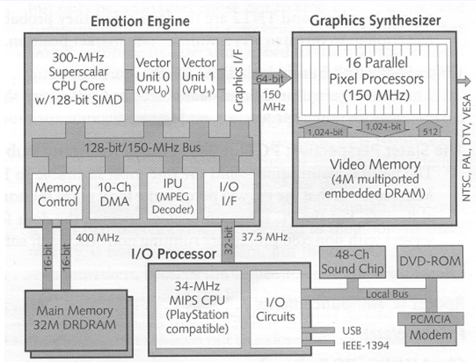


Figure 1. PlayStation 2000 employs an unprecedented level of parallelism to achieve workstation-class 3D performance.

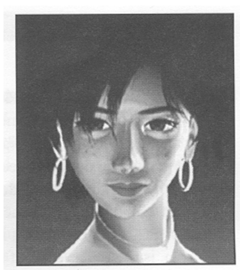
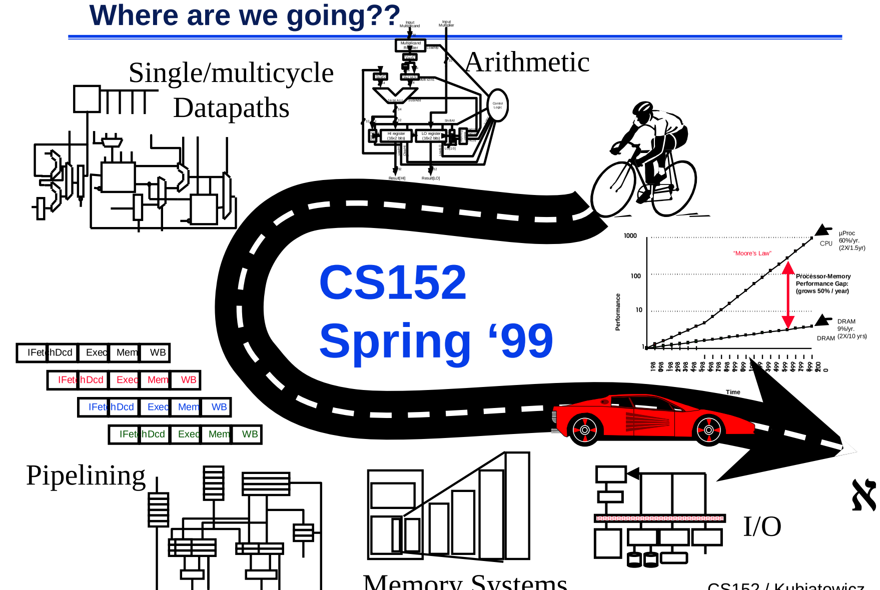


Figure 2. PlayStation 2000 screenshot. (Source: Namco)

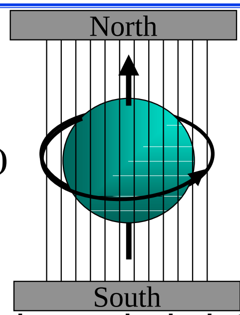
- ° (as reported in Microprocessor Report, Vol 13, No. 5)
  - Emotion Engine: 6.2 GFLOPS, 75 million polygons per second
  - Graphics Synthesizer: 2.4 Billion pixels per second
  - Claim: *Toy Story* realism brought to games!

Where are we going??



Maybe even Quantum Computing: Use of "Spin"

Spin ½ particle:
(Proton/Electron)



Representation:
|0> or |1>

- Particles like Protons have an intrinsic "Spin" when defined with respect to an external magnetic field
- Kane Proposal: use of impurity Phosphorus in silicon
  - Spin of odd proton is used to represent the bit
  - Manipulation of this bit via "Hyperfine" interaction with electrons
- Quantum Computers: Factor numbers in Polynomial time!
  - Classically this is (sub)exponential problem
  - Just cool?

CS152: So what's in it for me?

- **In-depth understanding of the inner-workings of modern computers, their evolution, and trade-offs present at the hardware/software boundary.**
  - Insight into fast/slow operations that are easy/hard to implementation hardware
  - Out of order execution and branch prediction
- **Experience with the *design process* in the context of a large complex (hardware) design.**
  - Functional Spec --> Control & Datapath --> Physical implementation
  - Modern CAD tools
- **BUILD A REAL PROCESSOR**
  - You will build pipelines that operate in realtime
  - Some of you may even design out-of-order processors
- **Designer's "Conceptual" toolbox.**

Conceptual tool box?

- Evaluation Techniques/Testing methodologies
- Levels of translation (e.g., Compilation)
- Levels of Interpretation (e.g., Microprogramming)
- Hierarchy (e.g, registers, cache, mem,disk,tape)
- Pipelining and Parallelism
- Static / Dynamic Scheduling
- Indirection and Address Translation
- Synchronous and Asynchronous Control Transfer
- Timing, Clocking, and Latching
- CAD Programs, Hardware Description Languages, Simulation
- Physical Building Blocks (e.g., CLA)
- Understanding Technology Trends

Course Structure

- Design Intensive Class --- 75 to 150 hours per semester per student

  MIPS Instruction Set ---> Standard-Cell implementation

- Modern CAD System (WorkView):

  Schematic capture and Simulation

  Design Description

  Computer-based "breadboard"
  - Behavior over time
  - Before construction

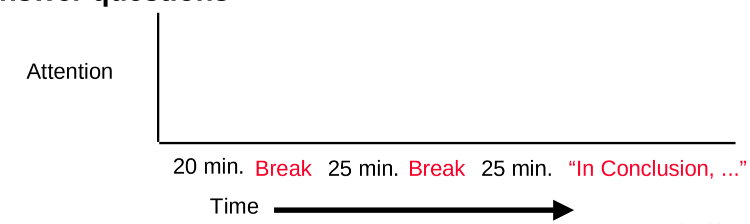
- **Lectures (rough breakdown):**
  - Review: 2 weeks on ISA, arithmetic, Logic, Verilog
  - 1 1/2 weeks on technology, HDL, and arithmetic
  - 3 1/2 weeks on testing, standard Proc. Design and pipelining
  - 1 1/2 weeks on advanced pipelining and modern superscalar design
  - 2 weeks on memory and caches
  - 1 1/2 weeks on Memory and I/O
  - ?? Guest lectures/Special lectures (Quantum computing?)
  - 2 weeks exams, presentations

Format: Lecture - Disc - Lab

- Mon: Lecture
- Tue: No class
- Wed: Lecture
- Thu: Discussion Section
  - Labs Due/Demo
  - Supplemental Information/Clarification of material from class
  - No discussion section this week!

Typical Lecture Format

- 20-Minute Lecture
- 5- Minute Administrative Matters
- 25-Minute Lecture
- 5-Minute Break (water, stretch)
- 25-Minute Lecture
- Instructor will come to class early & stay after to answer questions



Course Administration

- Instructor: John Kubiatowicz (kubitron@cs)
  673 Soda Hall
  Office Hours(Tentative): M 4:00-5:30
- TAs: Jack Kang (jackkang@uclink.berkeley.edu)
  Kurt Meinz (kurtm@mail.com)
- Labs: Windows 2000 accounts in 119 Cory
- Materials: http://inst.eecs.berkeley.edu/~cs152
  Mirror: http://www.cs.berkeley.edu/~kubitron/cs152
- Newsgroup: ucb.class.cs152
- Sign up for the mailing list: Go to homepage, click on link
- Text: *Computer Organization and Design: The Hardware/Software Interface, Second Edition*, Patterson and Hennessy
  - Q: Need 2nd Edition?
    yes! >> 50% text changed, all exersizes changed all examples modernized, new sections, ...

Course Exams

- Reduce the pressure of taking exams
  - Midterms: (approximately) March 5th and May 5th
  - 3 hrs to take 1.5-hr test (5:30-8:30 PM, 277 Cory?).
  - Our goal: test knowledge vs. speed writing
  - Review meetings: Sunday before?
  - Both mid-terms can bring summary sheets
- Students/Staff meet over pizza after exam at LaVals!
  - Allow me to meet you
  - I'll buy!

Course Workload

- Reasonable workload (if you have good work habits)
  - No final exam: Only 2 mid-terms
  - Every lab feeds into the project
  - Project teams have 4 or 5 members
- Spring 1995 HKN workload survey (1 to 5, 5 being hardest)

CS 150	4.2	CS 164	3.1
CS 152	3.4/3.5	CS 169	3.6
CS 162	3.9/4.0	CS 184	4.6

- Spring 1997 HKN workload survey (1 to 5, 5 being hardest)

CS 150	3.8	CS 164	4.0
CS 152	3.2	CS 169	3.2
CS 162	3.3	CS 184	3.3

- Revised Science/Design units: now 3 Science, 2 Design

Homework Assignments and Project

- Most assignment consists of two parts
  - Individual Effort: Exercises from the text book
  - Team Effort: Lab assignments
  - First Homework: out later today on Website.
- Assignments (usually) go out on Wednesday
  - Exercises due on a later Wednesday at beginning of lecture
    - *Brief (15 minute) quiz on assignment material in lecture*
    - *Must understand assignment to do quiz*
    - No late assignments!
  - Labs reports due by midnight via *submit* program on Thursday.
- Lab Homeworks returned in discussion section
  - To spread computer workload
  - put section time on them homeworks
- Discussion sections start next week
  - 101 Th 2:00 – 4:00 in 3107 Etcheverry
  - 102 Th 4:00 – 6:00 in 3107 Etcheverry
  - Must turn in survey to be considered enrolled (online tomorrow)

My Goal

- Show you how to understand modern computer architecture in its rapidly changing form.
- Show you how to design by leading you through the process on challenging design problems
- Learn how to test things.
- NOT to talk <u>at</u> you
- so...
  - ask questions
  - come to office hours
  - find me in the lab
  - ...

Project/Lab Summary

- Tool Flow runs on many workstations in Cory, but :
  - 119 Cory is primary CS152 lab.
  - 125 Cory is secondary CS152 lab (some machines shared with cs150)
- Get card-key access to Cory now (3rd floor...)
- Lab assignments:
  - Lab 1 C -> MIPS, SPIM (1½ weeks)
  - Lab 2 Fast Multiplier Design (2 week) + Intro to hardware synthesis
  - Lab 3 Single Cycle Processor Design (2 weeks)
  - Lab 4 Pipelined Processor Design (2 weeks)
  - Lab 5 Cache & DMA Design (3 weeks)
  - Lab 6 *Open ended work for final project*
- 2-hour discussion section for later in term. Early sections may end in 1 hour. Make sure that you are free for both hours however!
- <u>team in same section!</u>
- Oral presentation and written report

Project Focus

- Design Intensive Class ---
100 to 200 hours per semester per student
MIPS Instruction Set ---> FPGA implementation
- Modern CAD System:
Schematic capture and Simulation

Design Description

Computer-based "breadboard"
- Behavior over time
- Before construction

Xilinx FPGA board
- Running design at 25 MHz to 50 MHz

(~ state-of-the-art clock rate a decade ago)

Grading

- Grade breakdown
  - Two Midterm Exams: 35% (combined)
  - Labs and Design Project: 35%
  - Homework Completion: 5%
  - Quizzes: 15%
  - Project Group Participation 5%
  - Class Participation: 5%
- No late homeworks or labs:
our goal grade, return in 1 week
- Grades posted on home page/glookup?
  - Don't forget secret code on survey
  - Written/email request for changes to grades
- CS Division guideline upper division class GPA between 2.7 and 3.1.
  - average 152 grade will be a B or B+; set expectations accordingly

Course Problems

- Can't make midterm
  - Tell early us and we will schedule alternate time
- Forgot to turn in homework/ Dog ate computer
  - NO late homeworks or labs.
- What is cheating?
  - Studying together in groups is encouraged
  - Work must be your own
  - Common examples of cheating: running out of time on a assignment and then pick up output, take homework from box and copy, person asks to borrow solution "just to take a look", copying an exam question, ...
  - Better off to skip assignment (homeworks: 5% of grade!)
  - Labs worth more. However, each lab worth ~5% of grade.
  - Doesn't help on quiz (15%of grade) anyway

Class decides on penalties for cheating; staff enforces

- Exercises (book):
  - 0 for problem
  - 0 for homework assignment
  - subtract full value for assignment
  - subtract 2X full value for assignment
- Labs leading to project (groups: only penalize individuals?)
  - 0 for problem
  - 0 for laboratory assignment
  - subtract full value of laboratory
  - subtract 2X full value of laboratory
- Exams
  - 0 for problem
  - 0 for exam

Project Simulates Industrial Environment

- Project teams have 4 or 5 members in same discussion section
  - Must work in groups in "the real world"
- Communicate with colleagues (team members)
  - Communication problems are natural
  - What have you done?
  - What answers you need from others?
  - You must document your work!!!
  - Everyone must keep an on-line notebook
- Communicate with supervisor (TAs)
  - How is the team's plan?
  - Short progress reports are required:
    - What is the team's game plan?
    - What is each member's responsibility?

Things We Hope You Will Learn from 152

- Keep it simple and make it work
  - Fully test everything individually and then together
  - Retest everything whenever you make any changes
  - Last minute changes are big "no nos"
- Group dynamics. Communication is the key to success:
  - Be open with others of your expectations and your problems
  - Everybody should be there on design meetings when key decisions are made and jobs are assigned
- Planning is very important:
  - Promise what you can deliver; deliver more than you promise
  - Murphy's Law: things ***DO*** break at the last minute
    - Don't make your plan based on the best case scenarios
    - Freeze you design and don't make last minute changes
- Never give up! It is not over until you give up.

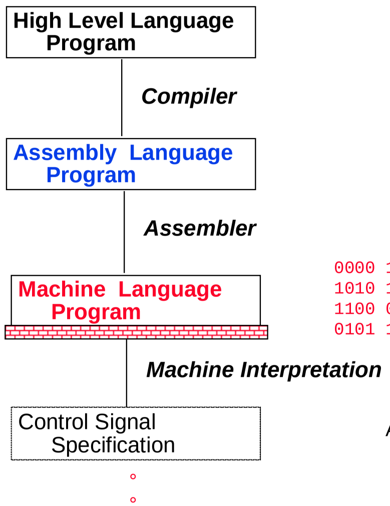
What you should know from 61C, 150

- Basic machine structure
  - processor, memory, I/O
- Read and write basic C programs
  - compile, link, load & execute
- Read and write in an assembly language
  - MIPS preferred
- Understand the concept of virtual memory
- Logic design
  - logical equations, schematic diagrams, FSMs, components
- Single-cycle processor

Getting into CS 152

- If not preenrolled, Fill out petition form
- Fill out survey and return Monday in class
- Know the prerequisites
  - CS 61C - assembly language, logic design and simple computer organization
- Prerequisite quiz on Monday 2/2; Pass/Fail
  - UC doesn't always enforce prerequisites
  - TA's will hold review sessions in section next Thursday+1 other time
  - Need to pass prerequisite quiz to take CS 152
  - Previous preq quizzes on web pages.
  - New material: something about single-cycle processor design….

Levels of Representation (61C Review)



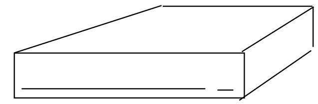

```
temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;
```

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

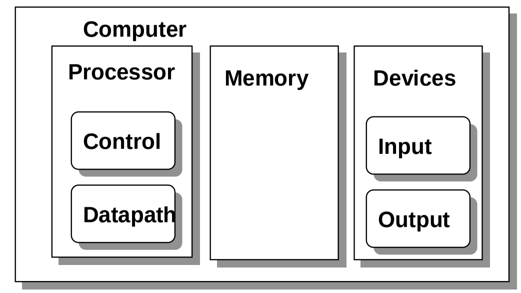
```
0000 1001 1100 0110 1010 1111 0101 1000
1010 1111 0101 1000 0000 1001 1100 0110
1100 0110 1010 1111 0101 1000 0000 1001
0101 1000 0000 1001 1100 0110 1010 1111
```

ALUOP[0:3] <= InstReg[9:11] & MASK

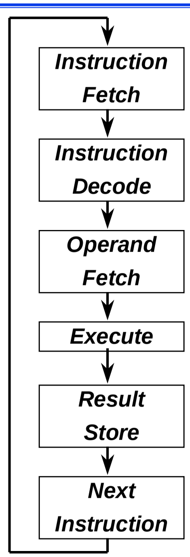
Levels of Organization



Workstation Design Target:
25% of cost on Processor
25% of cost on Memory
(minimum memory size)
Rest on I/O devices,
power supplies, box



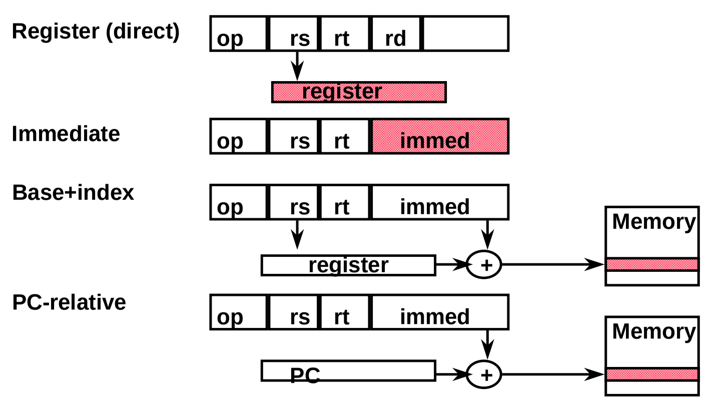
Instruction Set Architecture: What Must be Specified?



- **Instruction Format or Encoding**
  - **how is it decoded?**
- **Location of operands and result**
  - **where other than memory?**
  - **how many explicit operands?**
  - **how are memory operands located?**
  - **which can or cannot be in memory?**
- **Data type and Size**
- **Operations**
  - **what are supported**
- **Successor instruction**
  - **jumps, conditions, branches**
  - ***fetch-decode-execute is implicit!***

MIPS Addressing Modes/Instruction Formats

- All instructions 32 bits wide



MIPS I Operation Overview

° **Arithmetic Logical:**

- Add, AddU, Sub, SubU, And, Or, Xor, Nor, SLT, SLTU
- AddI, AddIU, SLTI, SLTIU, AndI, OrI, XorI, LUI
- SLL, SRL, SRA, SLLV, SRLV, SRAV

° **Memory Access:**

- LB, LBU, LH, LHU, LW, LWL,LWR
- SB, SH, SW, SWL, SWR

Miscellaneous MIPS I instructions

- ° **break** — A breakpoint trap occurs, transfers control to exception handler
- ° **syscall** — A system trap occurs, transfers control to exception handler
- ° **coprocessor instrs.** — Support for floating point
- ° **TLB instructions** — Support for virtual memory: discussed later
- ° **restore from exception kernel/user** — Restores previous interrupt mask & mode bits into status register
- ° **load word left/right** — Supports misaligned word loads
- ° **store word left/right** — Supports misaligned word stores

And in conclusion...

° **Continued rapid improvement in Computing**

- 2X every 1.5 years in processor speed;
  every 2.0 years in memory size;
  every 1.0 year in disk capacity;
  Moore's Law enables processor, memory
  (2X transistors/chip/ ~1.5 yrs)

° **5 classic components of all computers**

$\underbrace{\textbf{Control} \quad \textbf{Datapath}}_{\text{Processor}}$ **Memory Input Output**