# *Preview*

**I Stack:**

**-operands on top of stack**

**II Accumulator:**

**-1 operand is implicitly the accumulator**

**III General Purpose Register:**

**-Register/Memory: Mem thru Instr.**

**-Load/Store: Access Mem with L or S**



# *Addressing Modes*

**I Data Addressing Modes**

**42% Average - Displacement**

**33% Average - Immediate**

**13% Average - Register Indirect**

**12% - Rest**

**II Little Endian versus Big Endian**

# *Processor Metrics*

**Equations:**

**Execution time = Clock Cycles/Program * Clock Cycle time**

**Execution time = Clock Cycles/Program/Clock Rate**

**Clock Cycles/Program = Instr./Program * CPI**

**CPI = Clock Cycles/Program/(Instr./Program)**

**CPI = (CPU Time * Clock Rate)/Instruction Count**



# *Amdahl's Law*

**Speedup due to Enhancement**

$$= \frac{\text{Execution time without Enhancement}}{\text{Execution time with Enhancement}}$$

**Equation:**

**ExecutionTimeW/Enhance = ExecTimeWithoutEnhance*(F/S+(1-F))**

# *SPARC STATION*

**I** **Datapath + Control (CPU)**

**II** **CPU + Coprocessors**

**III** **Memory + Controller**

**IV** **BUS**

**V** **I/O: Hard Disk, Floppy, Keyboard**



# *COST*

**Equations:**

$$\text{Die Cost} = \frac{\text{Wafer Cost}}{\text{Dies per Wafer} * \text{Die Yield}}$$

$$\text{Die Yield} = \frac{\text{Wafer Yield}}{(1+(\text{Defects per unit area} * \text{Die Area})/a)^a}$$

$$\text{Dies/Wafer} = \frac{\text{pi}*(\text{Wafer D}/2)2}{\text{die area}} - \frac{\text{pi}*\text{Wafer D}}{\text{sqrt}(2*\text{DieArea})} - \text{Testdies}$$

# *Instruction Set Architecture*

**I VAX (CISC)**

- **A Variable Length Instructions**
- **B Many, many, many instr. types**

**II MIPS (RISC)**

- **A Fixed length**
- **B About 50-80 Instructions (How?)**
- **C Simple Instructions**



# *MIPS Assembly Language*

**I Crucial to know MIPS Assembly**

**II Laboratory #2**

**III Read Appendix + Recommended book**

**IV Swap code**

```
lw $15, 0($2)          # load v[0] to reg15
lw $16, 4($2)          # load v[4] to reg16
sw $16, 0($2)          # store old v[0]->v[4]
sw $15, 4($2)          # store old v[4]->v[0]
```

# *Fun Activity*

**I Pair up**

**II Communicate**

**III Plan**

**IV Tie your shoe laces**

**V For more challenge: Start with lace completely seperated from the shoe**



# *The Real Stuff: Review*

**I Whot's Compoota?**

- **A Brief History - ENIAC, EDSAC, ...**
- **B CISC versus RISC**
- **C Future MPP, NOW**

**II Structure**

- **A Pipeline**
- **B ISA + Organization**
- **C Microprocessor Performance (Yikes!)**

# *Discussion Organization*

**I Review**

**II Administrative**

**III Discussion (Homework & Lab)**

**IV Break**

**V Preview**

**VI Group Meeting (Once formed)**



# *CS152: What is it?*

**I Computer Architecture Course**

**II Build RISC Computer - CAD tools**

**III Network Interfaces**

**IV Lots of work**

**V First rate Project Report**

**VI Cooperation + Knowledge + Confidence**

**VII Excellent Jobs**

# *Introduction*

**Name: Young H. Cho**

**e-mail:**

**youngc@cs.berkeley.edu**

**homepage:**

**http://http.cs.berkeley.edu/~youngc**

**phone: 642-9076, 540-0508**

**office hour: Thurs 10-11a, by Appointments**



# *T.A.s*

**I Young Cho, youngc@cs, 466 Soda Hall**

**II Brent Chun, bnc@cs, 467 Soda Hall**

**III Steve Fink, sfink@cs**

**IV Scott McPeak, smcpeak@cory**

**V Tao Ye, tye@cs**

# CS 152

**Discussion Section 102: Friday 12-2**

Discussion Note #1


## T.A.: Young H. Cho

E-mail: youngc@cs.berkeley.edu
Homepage: http://http.cs.berkeley.edu/~youngc
Office: 466 Soda Hall  Phone: 642-9076




# *Agenda*

**I Introduction**

**II T.A.s**

**III CS152**

**IV Review**

**V Preview**