Prerequisites

Basic Logic Design and Boolean Algebra
   AND, OR, NAND, NOR gates
   Boolean Algebra
   Karnaugh Maps
   D Flip-flops, registers, and Memories
   Binary numbers, 2's complement, negation, overflows

Verilog

C/C++/Java programming

If you don’t know this material, **DO NOT TAKE THE CLASS**

If you don’t remember this material, **REVIEW NOW.**
Joint Work Policy

The processor design and homeworks will be done in groups of 1-2.
Students may not collaborate between groups on the specifics of homework or on the projects.
Let me know if you need help forming groups.

OK:
- Studying together for exams
- Discussing lectures or readings
- Talking about general approaches
- Help in debugging, CAD tools peculiarities, etc.

Not OK:
- Developing a design between groups
- Implementing the CPU between groups
- Checking homework answers between groups

Violation of these rules is at minimum grounds for failing the class.

Late Policy

All assignments due by the end of the class period

Late penalties:
- -10% for the first 24 hours
- -20% for the second 24 hours (total -30%)
- -30% for the third 24 hours (total -60%)
- -40% for all additional hours (total -100%)
Computer Architecture

Readings: 1.1-1.4

Interaction between hardware and software
Hardware sets realities, requirements
Area, power, performance
Software places demands on hardware
Processor only as good as software it runs

Implementing Software – The Compilation Process

C, C++, Java, …

/* Swap the i'th and (i+1)'th element of an array */
swap(int v[], int k) {
    int temp = v[k];
    v[k] = v[k+1];
    v[k+1] = temp;
}

SWAP:
    sll $2, $5, 2 // Compute address of v[k]
    add $2, $4, $2 // get v[k]
    lw $15, 0($2) // get v[k+1]
    lw $16, 4($2) // save new value to v[k]
    sw $16, 0($2) // save new value to v[k+1]
    sw $15, 4($2) // return from subroutine
    jr $31

000000 00000 00101 00010 00010 00000
000000 00100 00010 00010 00000 100000
100011 00010 01111 00000 00000 00000
100011 00010 10000 00000 00000 00000
101011 00010 01111 00000 00000 000100
101011 00010 01111 00000 00000 00000
000000 11111 00000 00000 00000 001000
Computer Organization

Five classic components

- **Computer**
  - Processor
    - Control
    - Datapath
  - Memory
  - Devices
    - Input
    - Output

Memory: Store instructions, data
Datapath: Perform operations (Add, subtract, …)
Control: Orchestrate operations (who does what when)
Input: Get information from the outside world
Output: Provide results

Execution cycle

- **Instruction Fetch**
  - Obtain instruction from program storage
- **Instruction Decode**
  - Determine required actions and instruction size
- **Operand Fetch**
  - Locate and obtain operand data
- **Execute**
  - Compute result value or status
- **Result Store**
  - Deposit results in storage for later use
- **Next Instruction**
  - Determine successor instruction