CSE 378
Machine Organization
and Assembly Language Programming
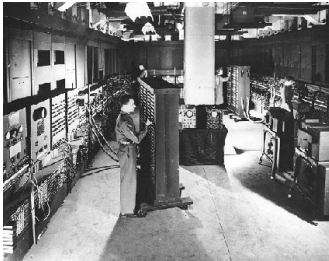
Winter 2004

John Zahorjan
Steven Balensiefer
Vadim Lobanov
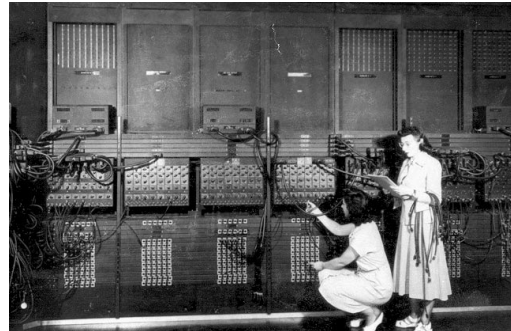

Intro To Architecture Via
a Bit of Computer History

ENIAC: 1946

Cost to build: $486,804.22
17,468 vacuum tubes, 5,000 additions/second (5 Kips)
30 feet x 50 feet, 30 tons
Cost to operate (electricity): $650/hr. (idling)

ENIAC Programming

IBM S360/67: 1967

Cost: $3,000,000
1,000,000 instructions/sec. (1 Mip)
512KB "core" memory ($1,000,000/MB)
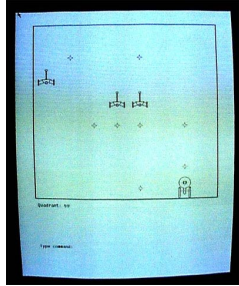352MB disk

---

VAX 11/780: circa 1980

Cost: $150,000
1 "VAX Mip"
1MB Ram

Xerox Alto: 1973
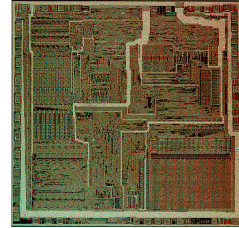
Cost: $32,000 (research)
1 Mip
Bitmap display
Mouse
"Microsoft Word"

Intel 8086 (x86): 1978

Cost: ~$350
5-10 MHz (~1Mip)
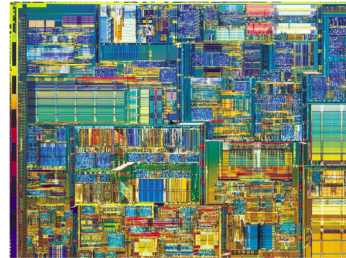29,000 transistors

Microprocessors + Workstation Concept

8/12/1981 IBM introduces its Personal Computer, which uses Microsoft's 16-bit operating system, Microsoft® MS-DOS® version 1.0, plus Microsoft BASIC, Microsoft COBOL, Microsoft Pascal, and other Microsoft products.
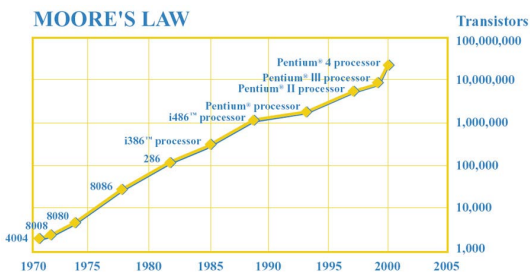

1984: Original Mac
Cost: $3,500
8 MHz
64KB RAM
No disk (400KB floppy)

Pentium 4: 2000's

Cost: $100's
2 GHz
42,000,000 transistors

Moore's Law: 1975

One Way to View Architecture as a Topic

What are we going to do with all those transistors?

or

How can we make *programs* run faster at the rate processor speeds are improving?
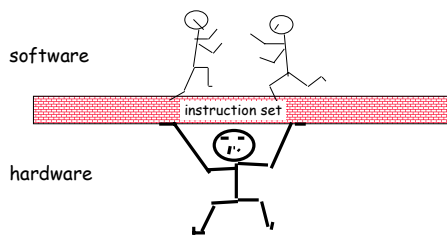
Course Overview

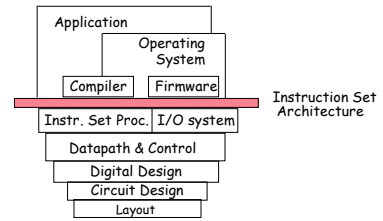---

What is "Computer Architecture"?

Computer Architecture   =
- Instruction Set Architecture +
- Machine Organization + …

The Instruction Set: a Critical Interface

software

instruction set

hardware

What is "Computer Architecture"?

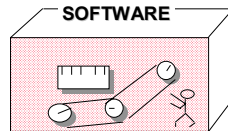| Application |
| Operating System |
| Compiler | Firmware |
| Instr. Set Proc. | I/O system |
| Datapath & Control |
| Digital Design |
| Circuit Design |
| Layout |

Instruction Set Architecture

§ Coordination of many levels of abstraction

§ Under a rapidly changing set of forces
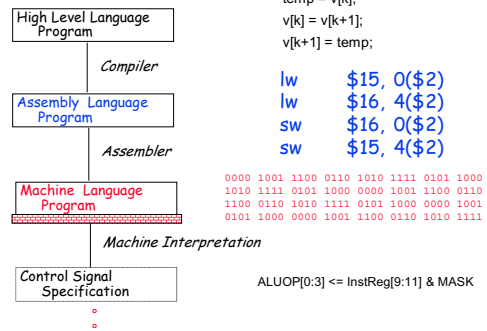
§ Design, Measurement, and Evaluation

Instruction Set Architecture
(subset of Computer Architecture)

"... 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

SOFTWARE

Levels of Representation

High Level Language Program

Compiler

Assembly Language Program

Assembler

Machine Language Program

Machine Interpretation

Control Signal Specification

```
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

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

Machine Organization

§ Since 1946 all computers have had 5 components

Processor
Control
Datapath
Memory
Input
Output
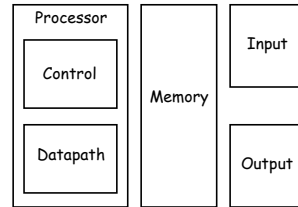
A Machine (is not just a CPU)

Pentium III Chipset

Proc
Caches
Memory

Busses
adapters

Controllers

I/O Devices: Disks Displays Keyboards

Networks

Intel® Processor (100/133 FSB)
1.0 GB/s
AGP4X
MCH 82820
1.0 GB/s
3.2 GB/s
PC800/700/600 RDRAM
Intel Hub Architecture
AC '97 6 Channel
LAN Interface
ATA 100 IDE Channels (2)
2 USB Controllers, 4 Ports
ICH2 82801BA
5M Bus
PCI
SIO

Where are We Going??

Single/multicycle Datapaths

Arithmetic

Pipelining

IFetch Dcd Exec Mem WB

Memory Systems

I/O