CSE 378
Machine Organization
and Assembly Language Programming

Winter 2005

John Zahorjan
Andrei Alexandrescu
Adrienne Wang

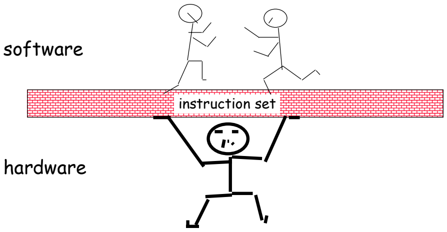
Course Overview

What is "Computer Architecture"?

Computer Architecture =

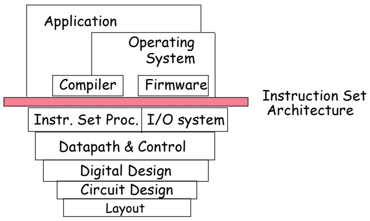
- Instruction Set Architecture (ISA) +
- Machine Organization + …

The Instruction Set: a Critical Interface



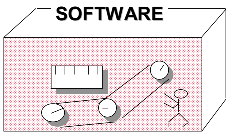
Lesson from history:
Push complex functionality into software –
it's more flexible, and it ends up being faster.

What is "Computer Architecture"?



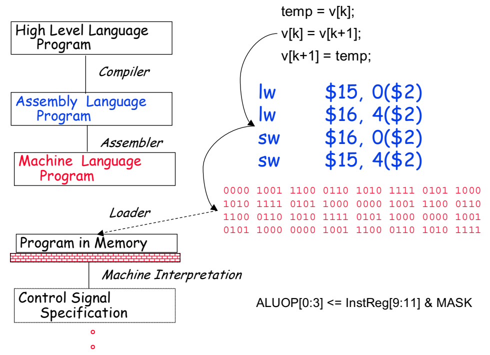
Instruction Set Architecture
(subset of Computer Architecture)

"… 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

Levels of Representation



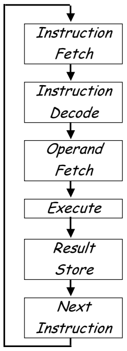
```
temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;
```

```
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

Basic Execution Cycle



Obtain instruction from program storage

Determine required actions and instruction size

Locate and obtain operand data

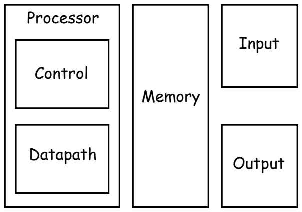
Compute result value or status

Deposit results in storage for later use

Determine successor instruction

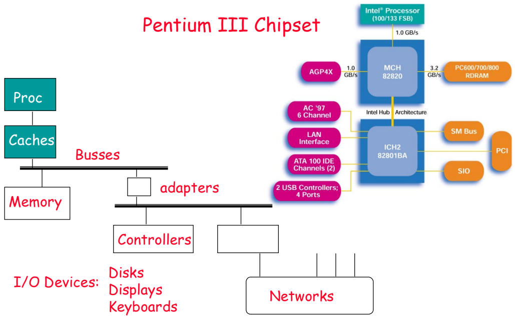
Machine Organization

§ Since 1946 all computers have had 5 components

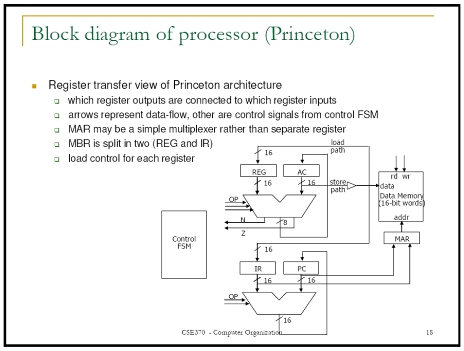


A Machine (is not just a CPU)

Pentium III Chipset



Machine Organization

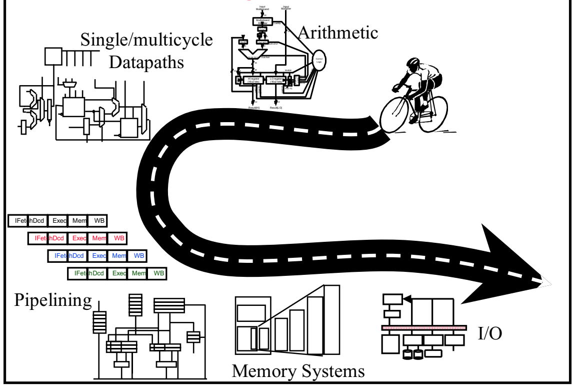


Block diagram of processor (Princeton)

- Register transfer view of Princeton architecture
  - which register outputs are connected to which register inputs
  - arrows represent data-flow, other are control signals from control FSM
  - MAR may be a simple multiplexer rather than separate register
  - MBR is split in two (REG and IR)
  - load control for each register

CSE370 - Computer Organization 18

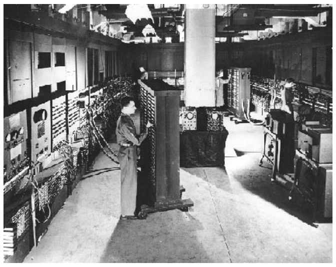
Where are We Going??



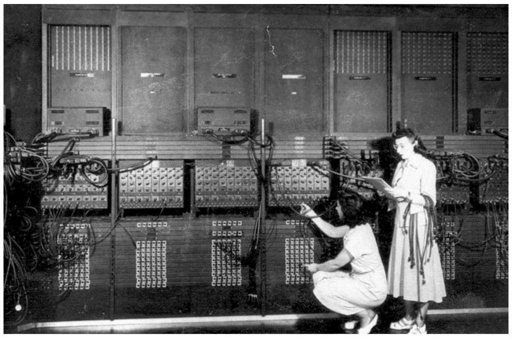
A Bit of History
(And What is Moore's Law?)

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

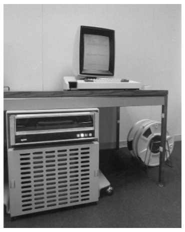
Cost: $3,000,000
1,000,000 instructions/sec. (1 Mip)
512KB "core" memory ($1,000,000/MB)
352MB disk

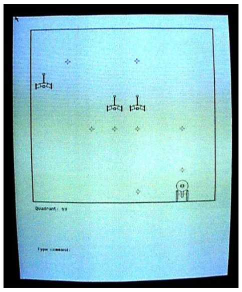
VAX 11/780: circa 1980

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

Xerox Alto: 1973

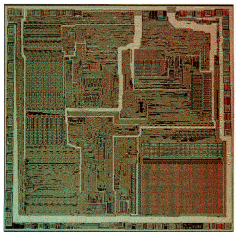
Cost: $32,000 (research)
1 Mip
Bitmap display
Mouse
"Microsoft Word"





Intel 8086 (x86): 1978

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



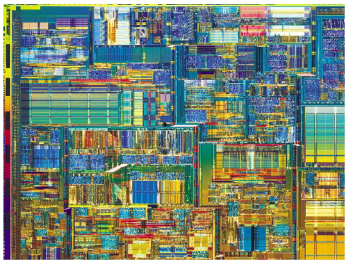
Microprocessors + Workstation Concept

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.

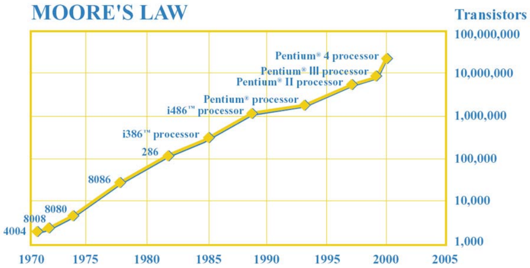
1984: Original Mac
Cost: $3,500
8 MHz
64KB RAM
No disk (400KB floppy)

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?

A Remark About the Weight of History

A *computing system* is more than just hardware – there is an enormous base of software required (e.g., OS, compilers, applications).

Architectures tend to undergo evolution, rather than revolution, since *backward compatibility* is required to gain adoption.

On the other hand, the *machine organization* (implementation of the ISA) is free to change as dramatically as the designer thinks is beneficial.