Computer architecture

Von Neumann architecture
— architecture where program stored in memory

Computer architecture

Von Neumann architecture —
(bottleneck — memory slower than processor)

Registers:

- general purpose
- special purpose
  - program counter
  - instruction register
  - others...

Computer architecture

Adding 2 values from memory:

1. Get first value in a register
2. Get second value in a register
3. Add results in ALU — result in a register
4. Store result in memory (or a register)

Computer architecture

Example machine language — Appendix C

Instruction:

- 4 bits op-code
- 12 bits operands
    - 4 bits register
    - 8 bits address — 256 words in memory

How many general purpose registers are there?

A. 4

B. 8

C. 12

D. 16

E. 32

Vote at m.socrative.com. Room number 415439.

Example machine language

Instructions:

Op-code	Operands	Meaning
1	RXY	Load reg R from memory cell XY
2	RXY	Load reg R with value XY
3	RXY	Store contents of reg R in cell XY
4	$0RS$	Move contents of reg R to reg S
5	RST	Add two's compl. contents of reg S to reg T; store result in R
6	RST	Foating point add
7	RST	OR
8	RST	AND
9	RST	XOR
A	$R0X$	Rotate reg R X bits to right
B	RXY	Jump to XY if $c(R) = c(0)$
C	000	HALT

Note operands are hexadecimal.

Example machine language

One word (cell) is 1 byte.
One instruction is 16 bits.

Machine cycle:

- fetch — get next instr., increment program counter by 2
- decode
- execute (instr)

Example machine language

Computer Architecture
Example Machine Language
External Devices
Computer Architecture

Example: check if low-order 4 bits of value in reg 1 = 0

2000	load	load zero into reg 0
220F	load	load string 00001111 into reg 2
8312	AND	c(reg 1) AND c(reg 2) —> reg 3 — masking
B3XY	JMP	jump to address XY if c(reg 3) = c(reg 0)

Example machine language

How can we complement a byte in reg 1?

A. load 11 in register 2; OR 3,1,2;

B. load FF in register 2; OR 3,1,2;

C. load 00 in register 2; XOR 3,1,2;

D. load 11 in register 2; XOR 3,1,2;

E. load FF in register 2; XOR 3,1,2;

Vote at m.socrative.com. Room number 415439.

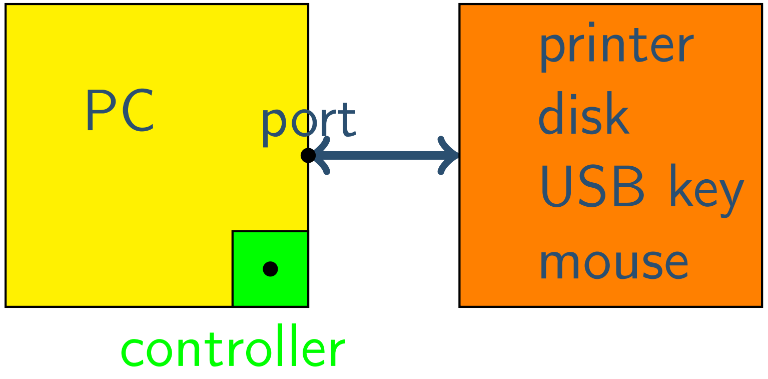
Computer architecture

RISC — reduced instr. set — fast per instr. — cell phones
CISC — complex instruction set — easier to program — PC

Clock

- coordinates activities
- faster clock → faster machine cycle
- Hz — one cycle per second
- MHz — mega Hz (1 million Hz)
- GHz — giga Hz (1000 MHz)
- flop — floating point ops / sec
- benchmark — program to run on different machines for comparison

External devices



motherboard — main circuit board (with CPU, memory)

controller — on motherboard or plugged into motherboard
To reduce number — universal serial bus (USB), FireWire, Thunderbolt
Serial — 1 bit at a time (vs. parallel) — fast for short distances

DMA — CPU not involved after starting — (read sector of disk)

If everything uses bus, von Neumann bottleneck.

External devices

Initial connection

- handshaking (also for protocols)
- often status word — is printer OK, paper out, jam,...

Communication rates

- bits per second (bps) / bytes per second (Bps)
- Kbps — standard phone lines
- Mbps — 1,000,000 bps — USB, FireWire 100s of Mbps
- Gbps — 1,000,000,000 bps — USB 3.0, Thunderbolt — Gbps

External devices

(Time-division) multiplexing

	telephone voice	data from computer	telephone voice	. . .	

data from computer can be modem, xDSL, cable TV

bandwidth – max rate
broadband – high rate

Making computers faster

- Pipelining —

ADD	RXY	fetch instruction
ADD	R'X'Y'	decode
ADD	R"X"Y"	perform add
		possibly further divided

- Supercomputers
  — multiprocessor machines now
  (10s of 1000s, with over 3,000,000 cores)
  — SIMD, MIMD
- Multi-core — in single integrated circuit, package
  - dual-core — 2 processors
  - quad-core — 4 processors
  - ...
  - 2 at 2 GHz not as good as 1 at 4 GHz