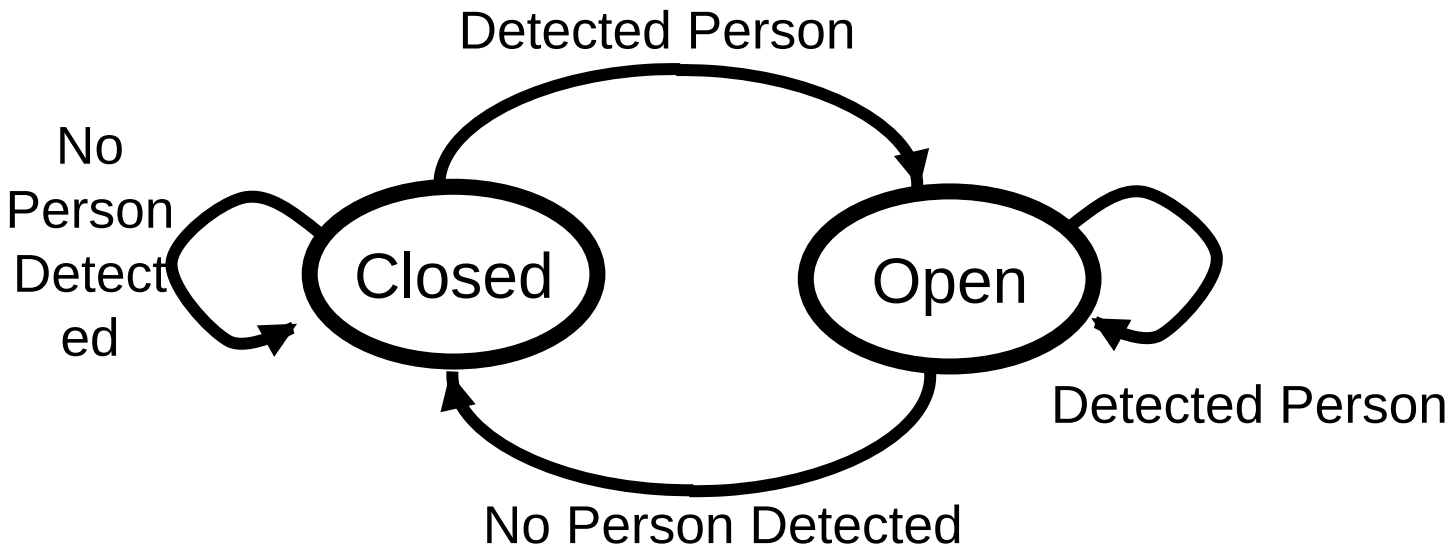
Computer Organization 1: CPUs and RAM

3/30/2006

COS 116

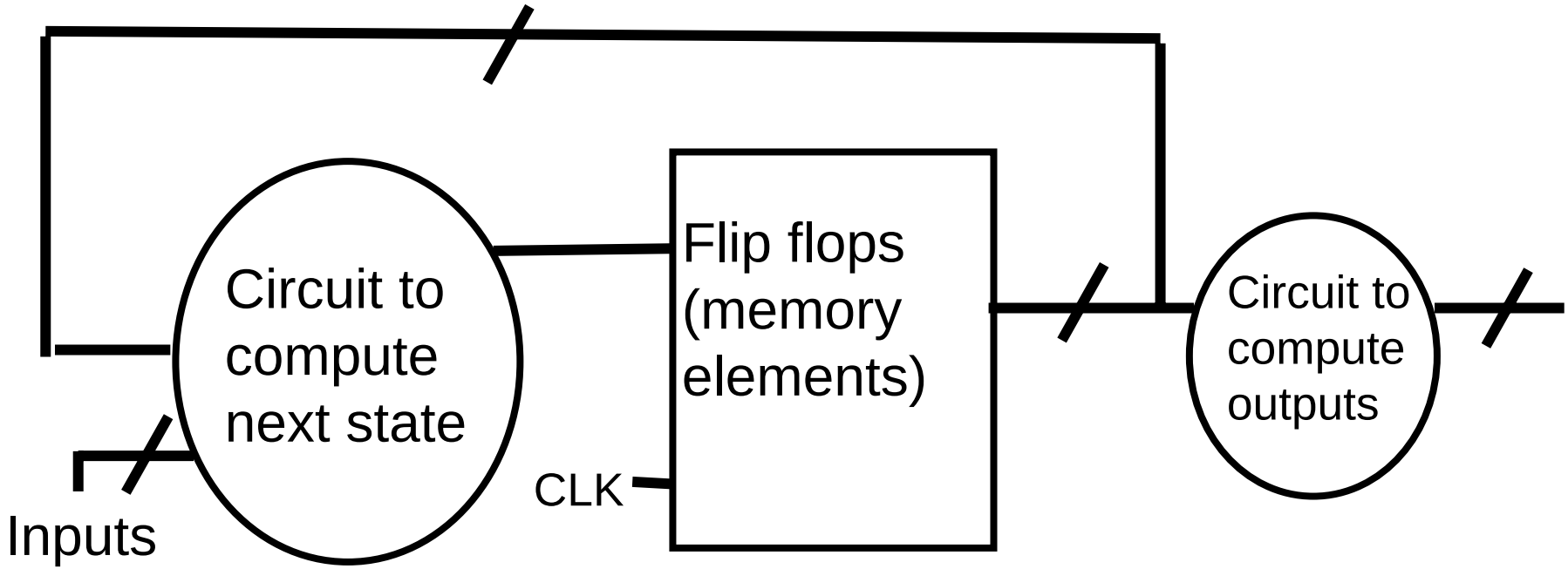
Instructor: Sanjeev Arora

Recall: FSMs of Moore Types



- Finite number of states
- Machine can produce outputs, these depend upon current state
- Machine can accept one or more bits of input; reading these causes transitions among states.

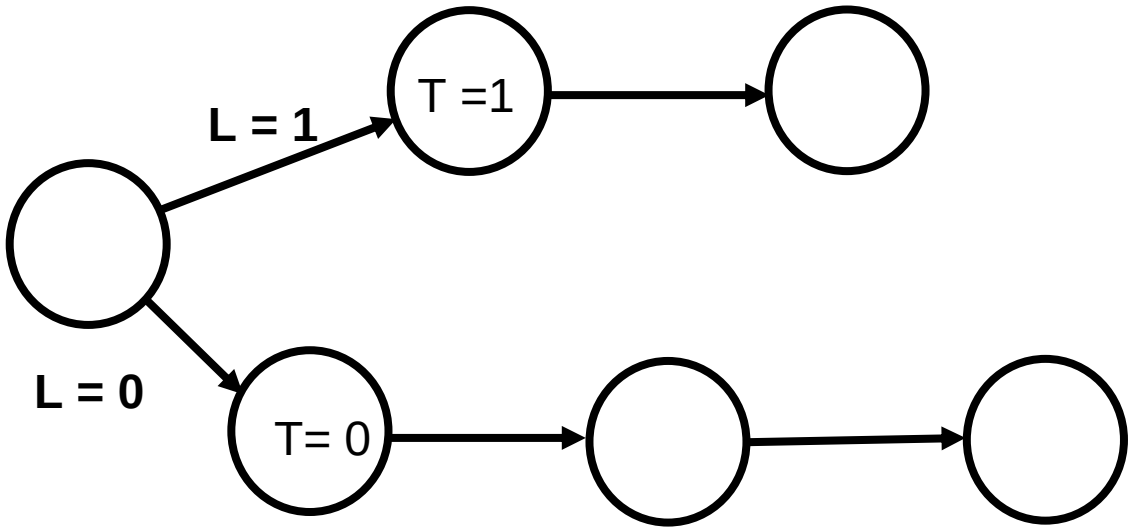
Conceptual Representation of FSM as synchronous circuit



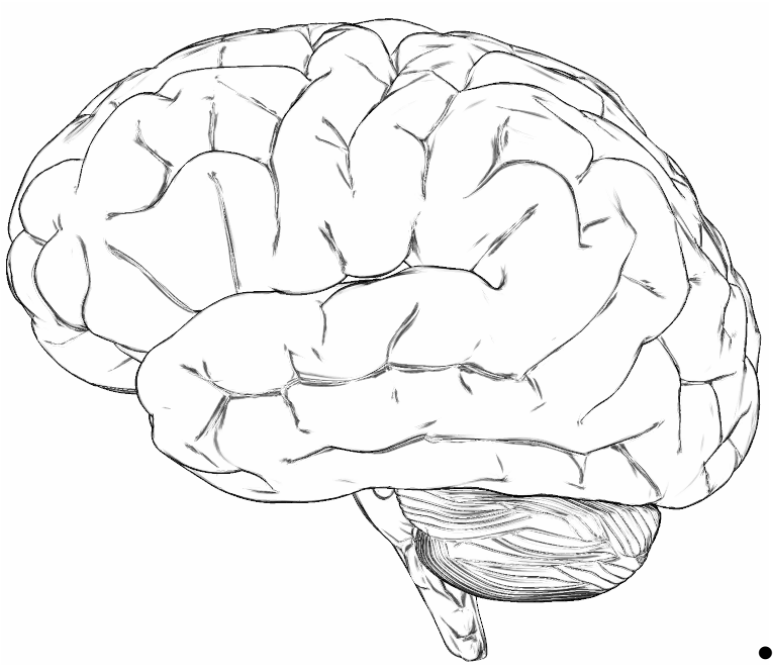
K Flip flops allow FSM to have 2^K states

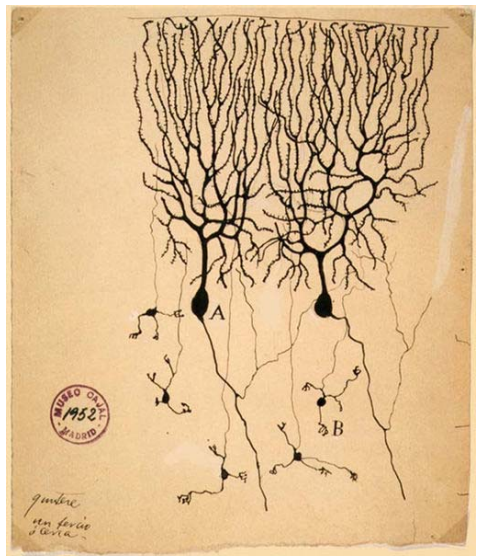
How an FSM does “reasoning”

“If left infrared sensor detects a person, turn left”



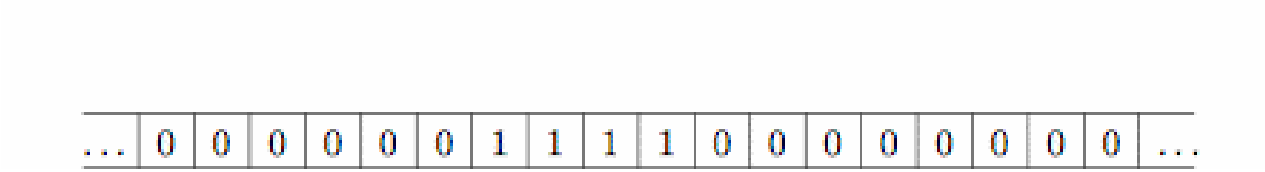
Speculation: Brain as FSM?





- Network (“graph”) of 100 billion neurons; each connected to a few thousand others
- Neuron = tiny Computational Element; “switching time” 0.01 s
- Neuron generates a voltage spike depending upon how many neighbors are spiking.

Discussion: How would you implement a Turing-Post program with a digital circuit?



1.PRINT 0
2. GO RIGHT
3. GO TO STEP 1 if 1 SCANNED
4. GO TO STEP 1 if 0 SCANNED
5. STOP

Assume “PRINT” and “SCAN” as basic operations

Main Insight

Computer = FSM controlling a larger (or infinite) memory.

Finally, the secret revealed...

How real computers execute programs.

Scribbler Control Panel Program

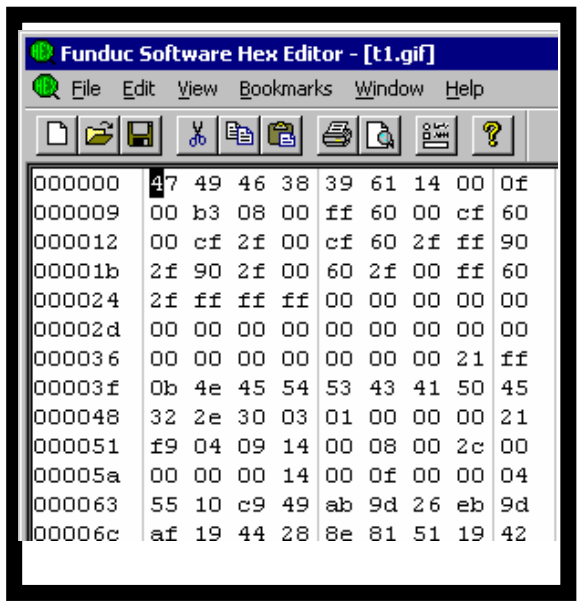
```
If <Obstacle on Either Side> Then
{
   Play Sound for 1s at Frequency 440Hz
}
Else
{
   LED: ON,  ON,  ON
}
END
```

F5

“Download to Robot”

(Compilation)

Machine Executable Code



Similar to:

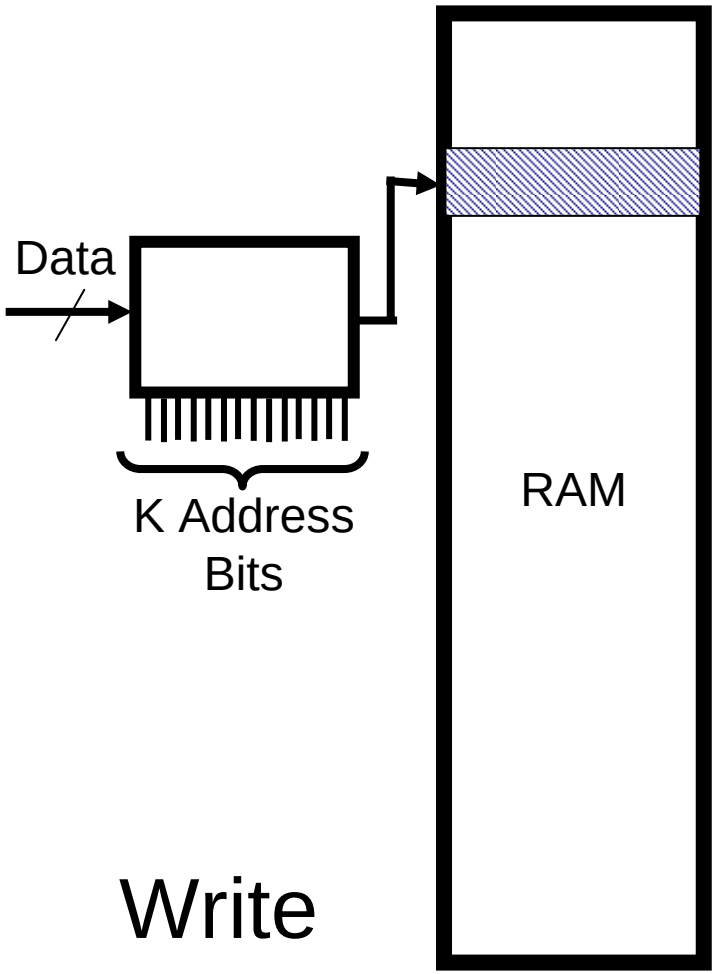
- T-P programs represented in binary
- .exe files in the Wintel world

Before proceeding further…

A word about Random Access Memory (RAM)

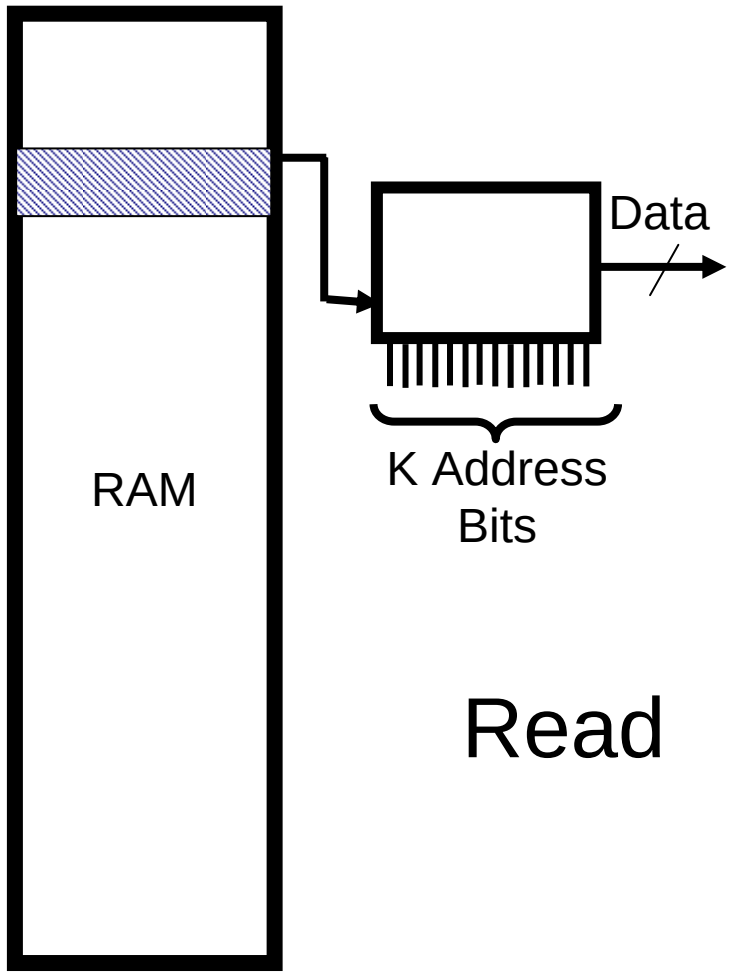
Memory where each location has an address

RAM



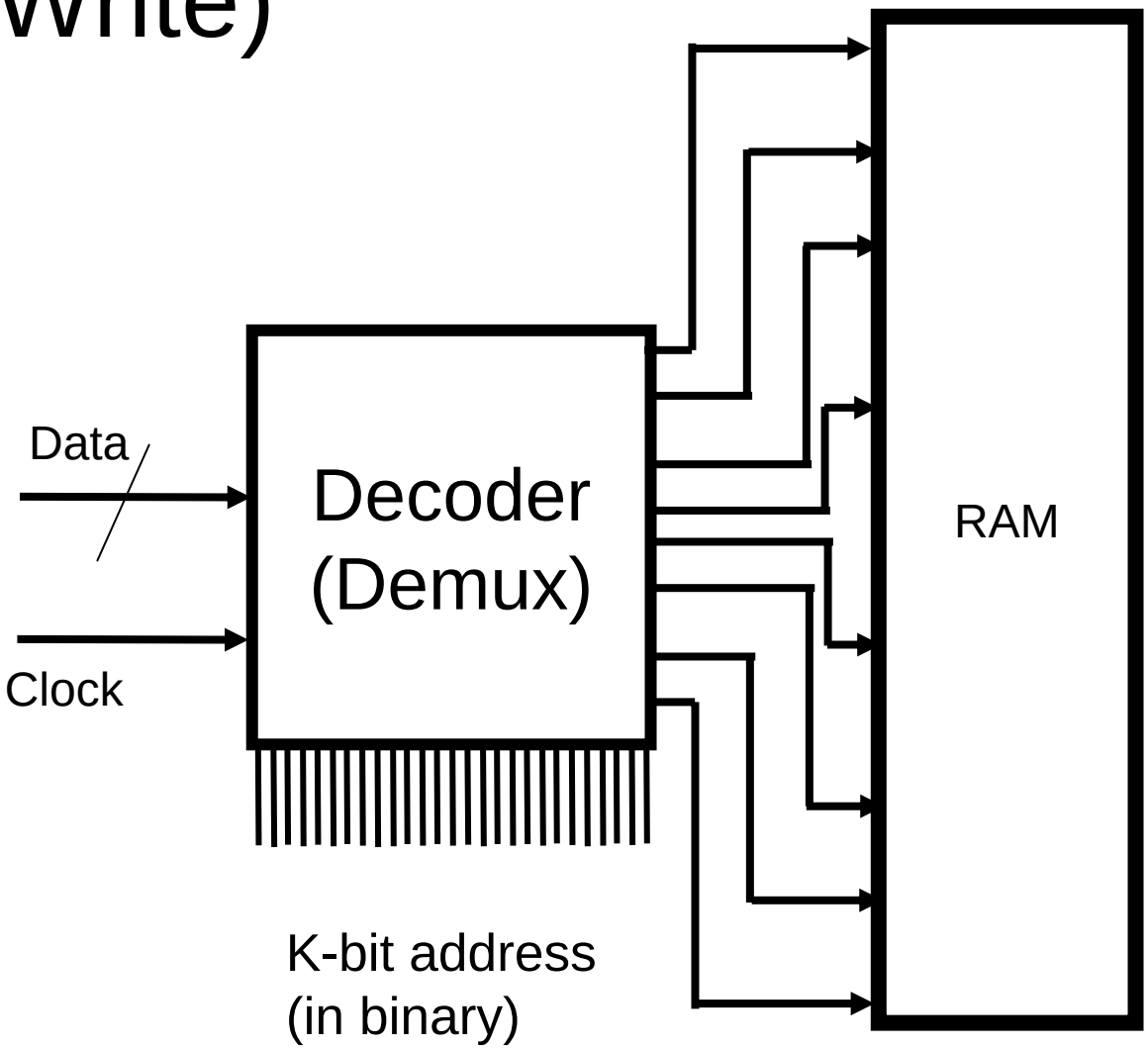
Write

2^K bits; bank of flipflops



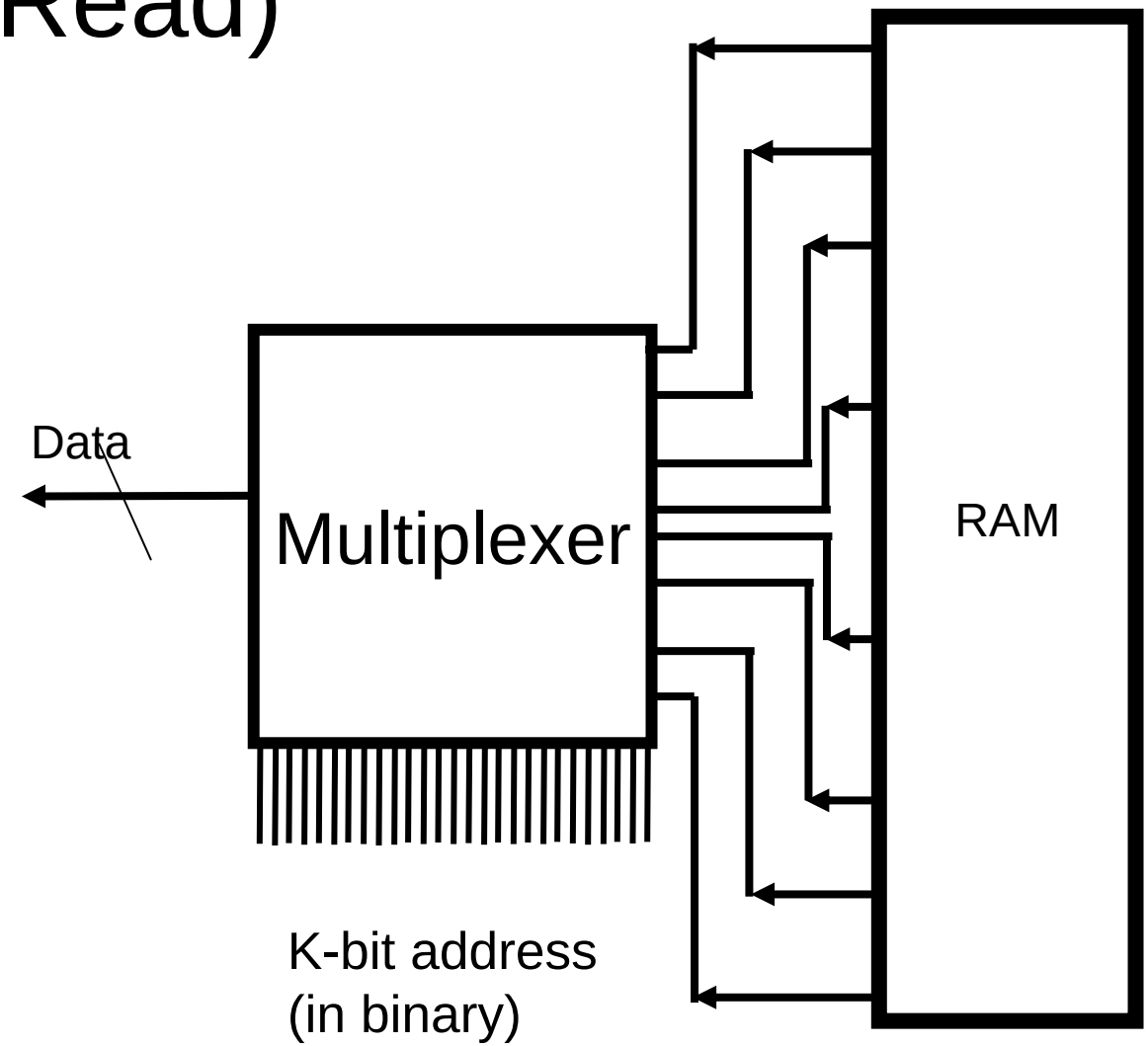
Read

RAM: a simple synchronous circuit (Write)



The decoder selects which cell in the RAM gets its “Write” input toggled

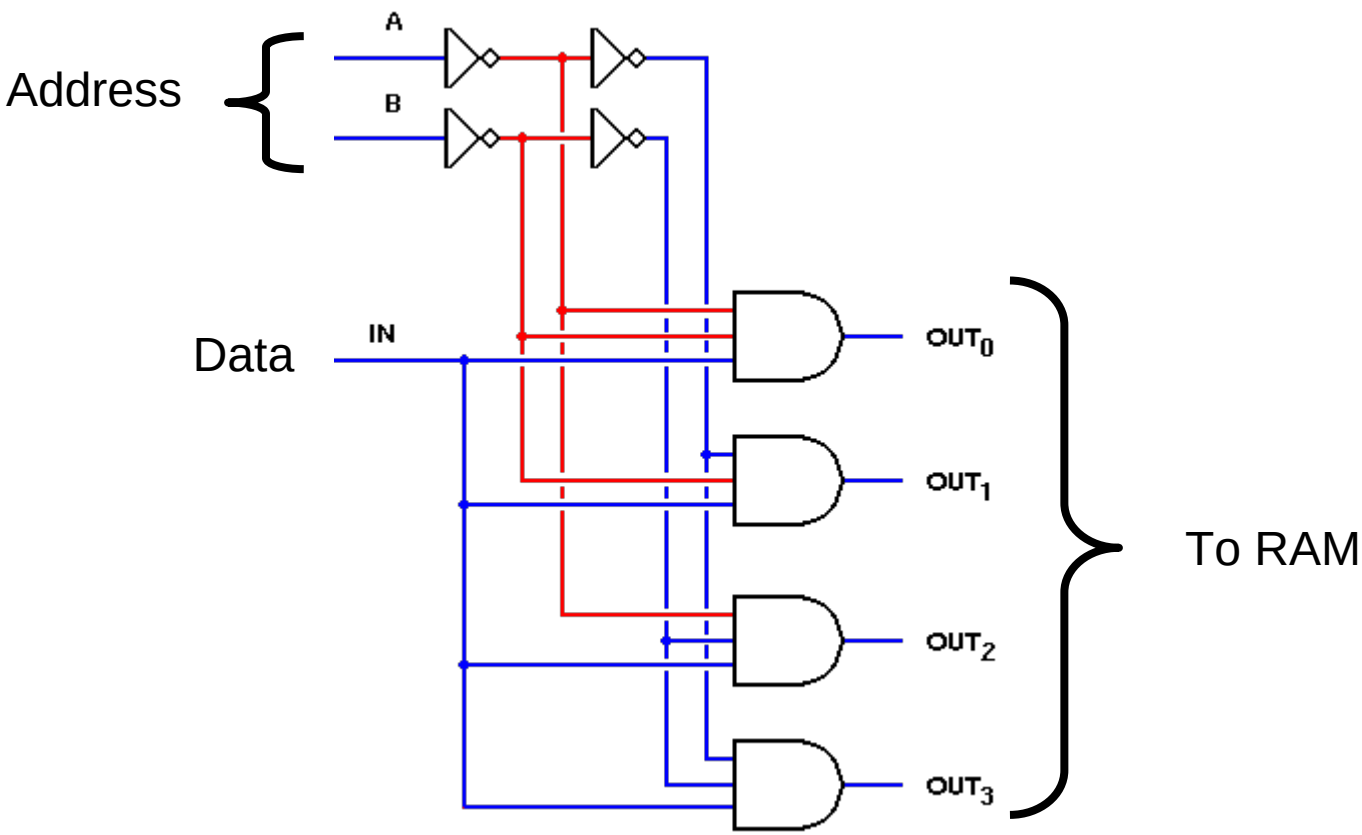
Ram: a simple synchronous circuit (Read)



The multiplexer is connected to all cells in the RAM; selects the appropriate cell based upon the k-bit address

A 2-4 Line Demultiplexer/Decoder

(see logic handout from midterm week)

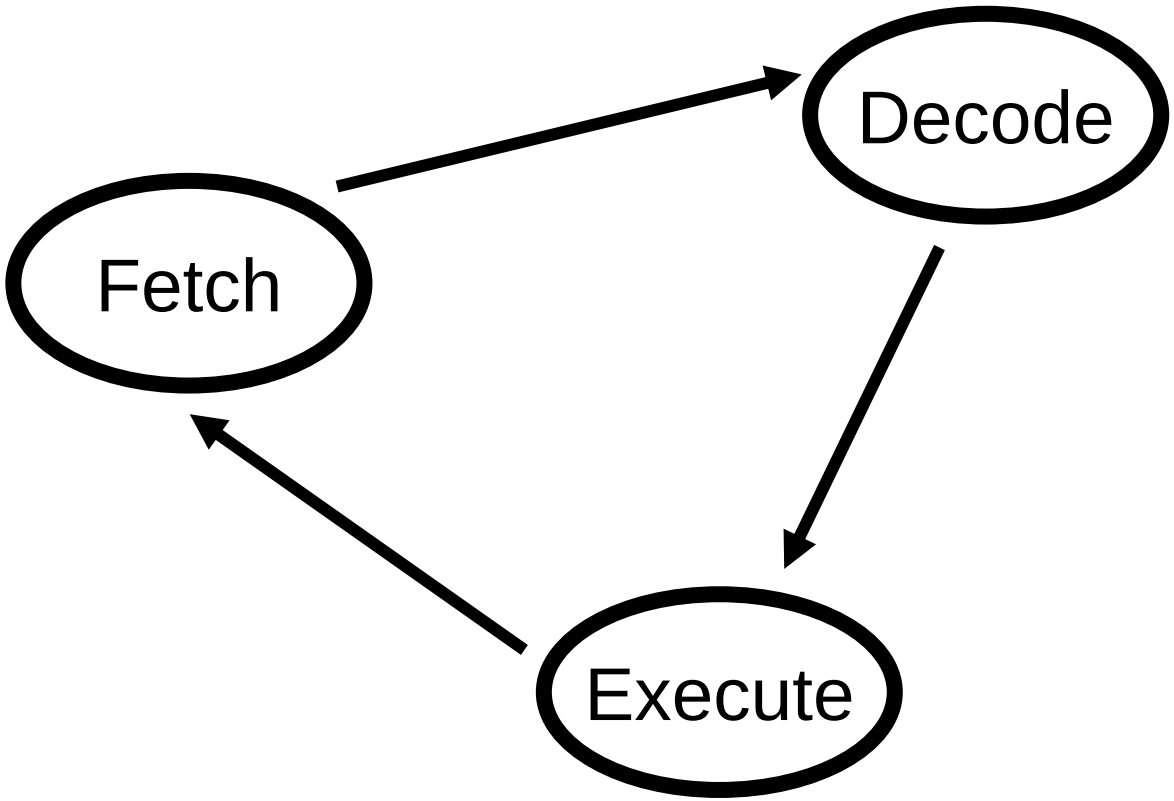


Finally, the secret revealed...

How computers execute programs.

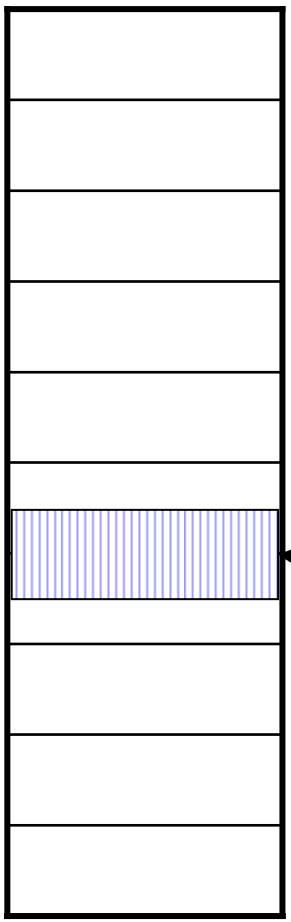
Meet the little green man…

The Fetch – Decode – Execute FSM

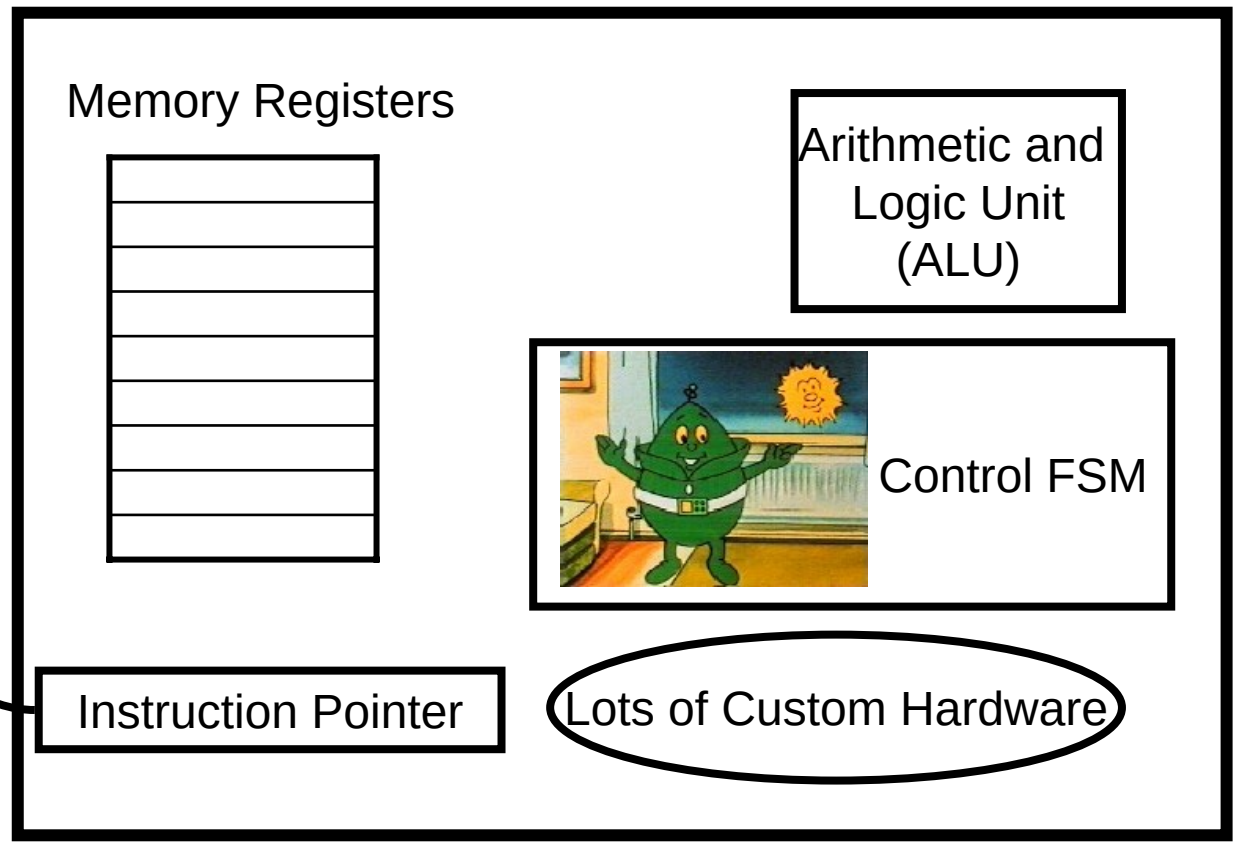


Greatly simplified view of modern CPUs.

Program (in binary) stored in memory



RAM



Examples of Machine Language Instructions

ADD 3 7 12	Add contents of Register 3 and Register 7 and store in Register 12
LOAD 3 67432	Read Location 67432 from memory and load into Register 3
JUMP 4 35876	If register 4 has a number > 0 set IP to 35876

Stored in binary (recall Davis's binary encoding of T-P programs)

Different CPUs have different machine languages

- Intel Pentium
- Power PC
- Palmpilot, etc.

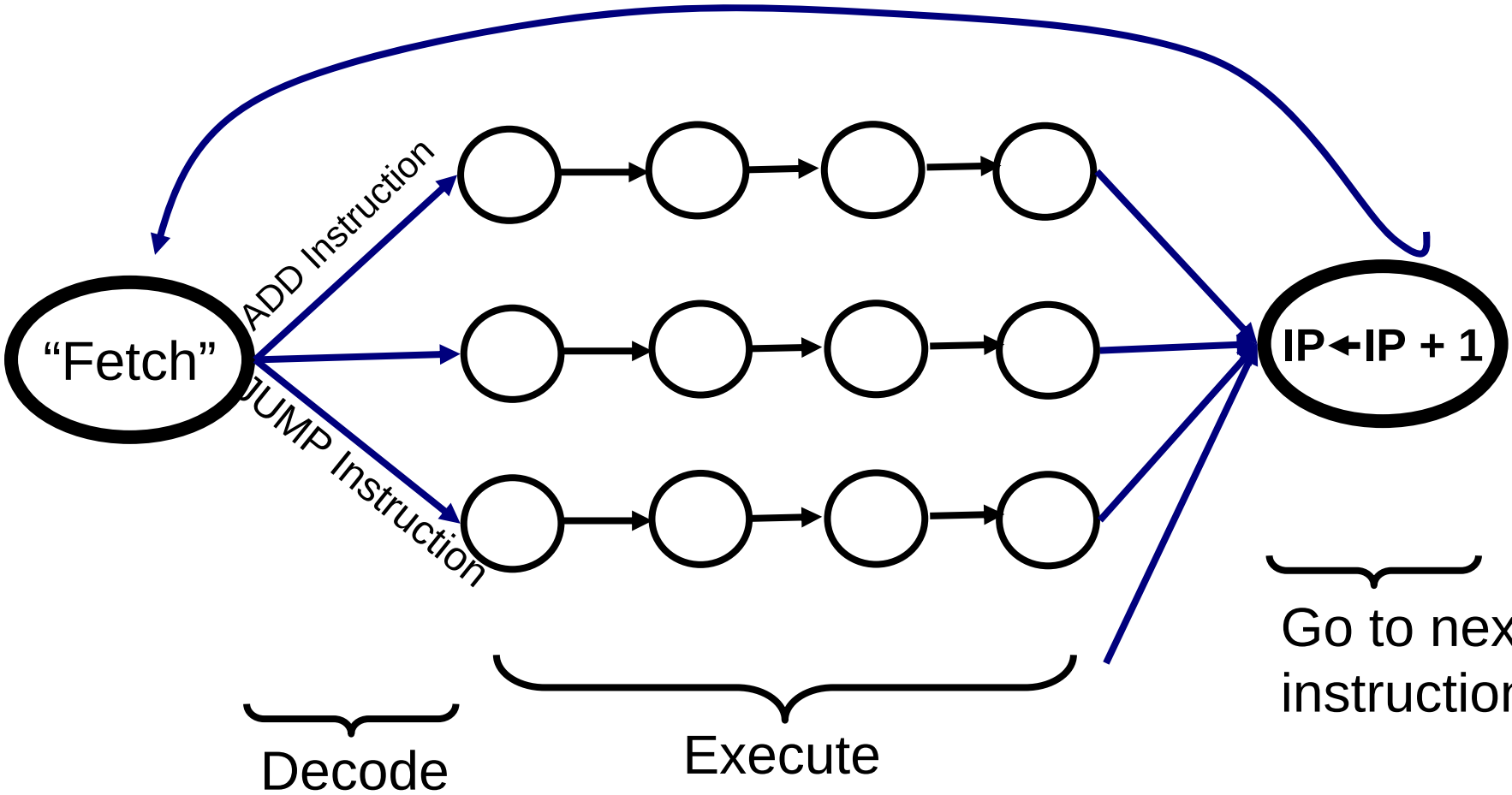
"Backwards Compatibility" – Pentium 4's machine language extends Pentium 2's machine language

Machine languages now allow complicated calculations (eg for multimedia, graphics) in a single instruction

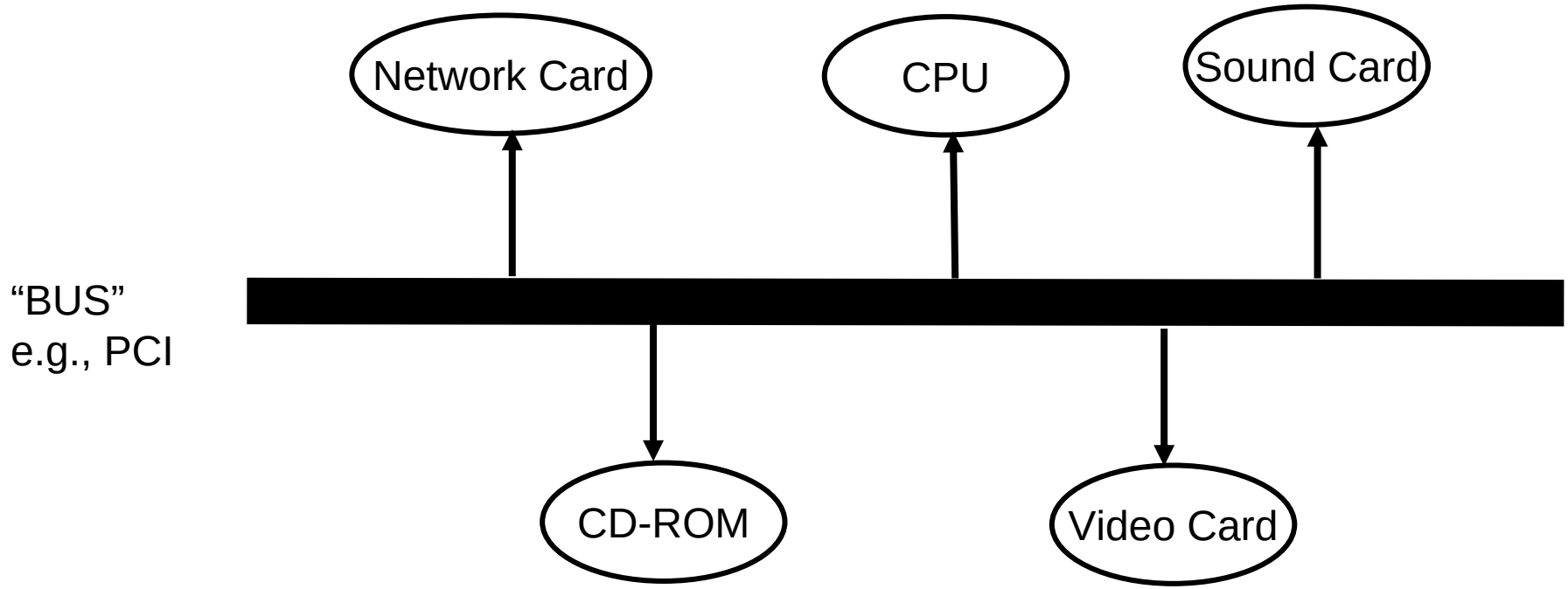
Example:
Intel press release

SANTA CLARA, Calif., June 28, 2004 - Intel Corporation today announced availability of a new Intel® Xeon™ processor-based platform and a host of new products and technologies for its Intel Xeon processor family that significantly boost performance, memory and graphics capabilities for workstation platforms. Workstations will benefit from rich set of new technologies that address the increasingly data-hungry systems and software applications that crave performance for a range of functions such as financial and scientific data modeling to digital filmmaking and design automation.

Fetch – Decode – Execute FSM



CPU as a conductor of a symphony



Bus: "Everybody hears everybody else"