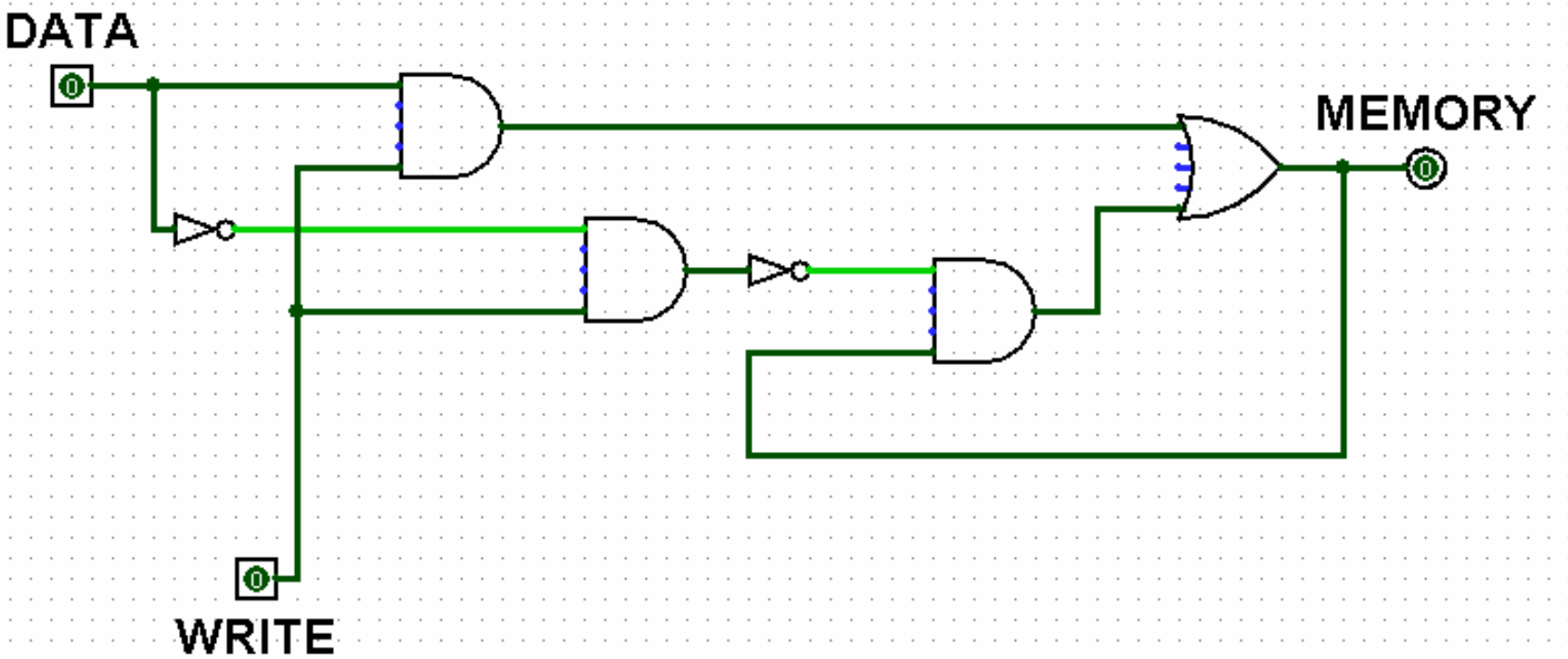
Computer Organization 1: CPUs and RAM

3/29/2006

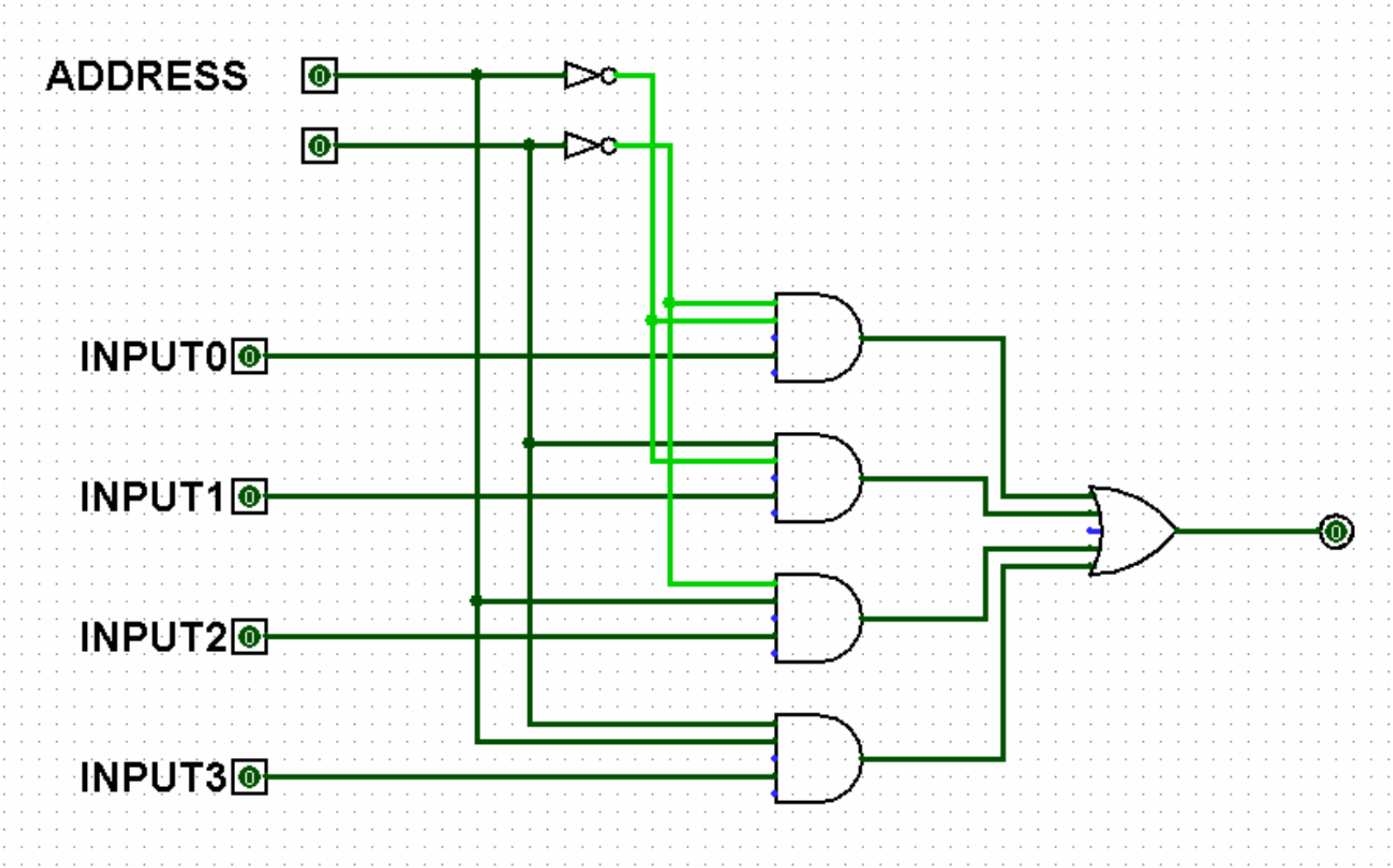
COS 116

Instructor: Umar Syed

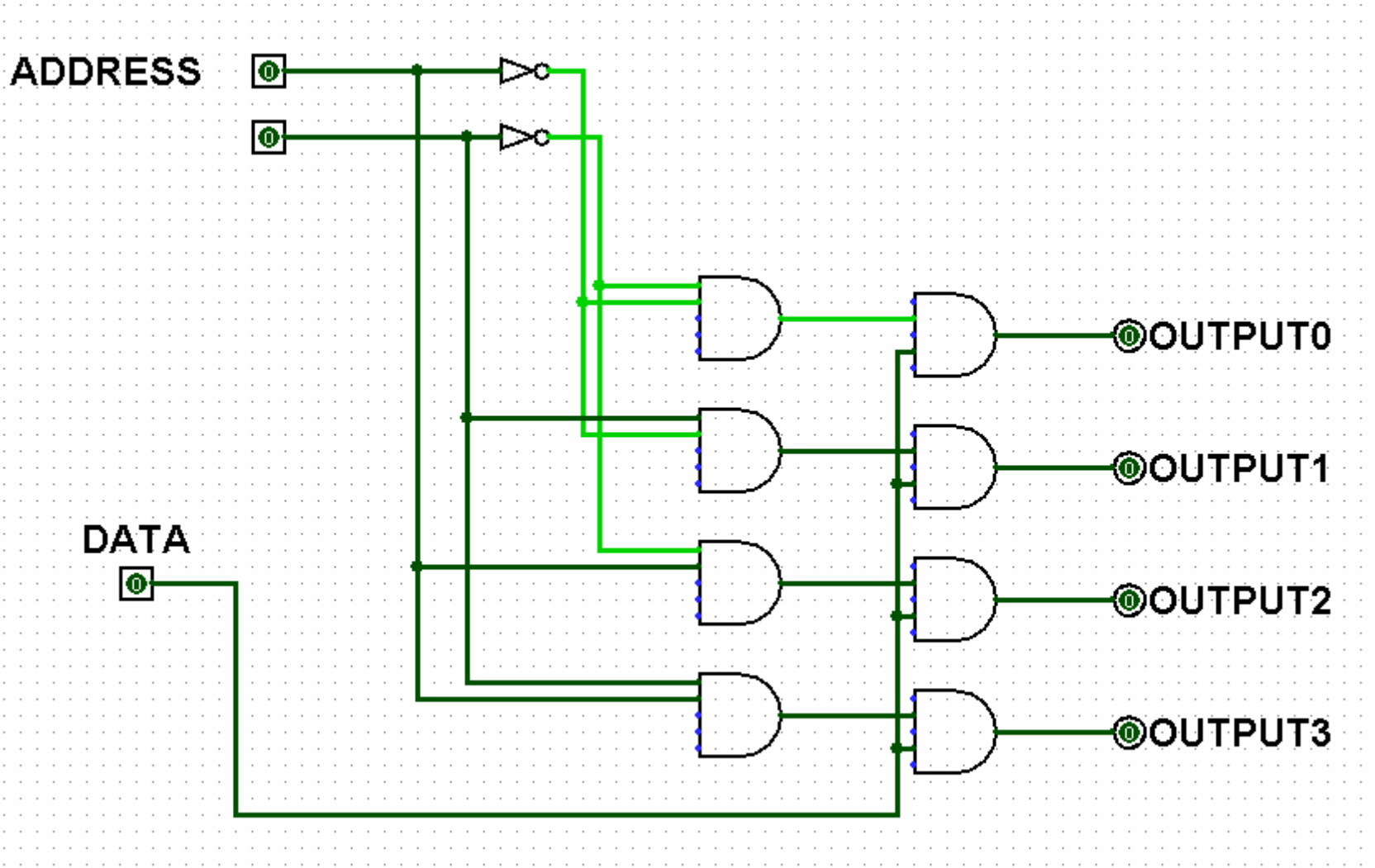
D Flip Flop (demo)



Multiplexer (demo)



Demultiplexer (demo)

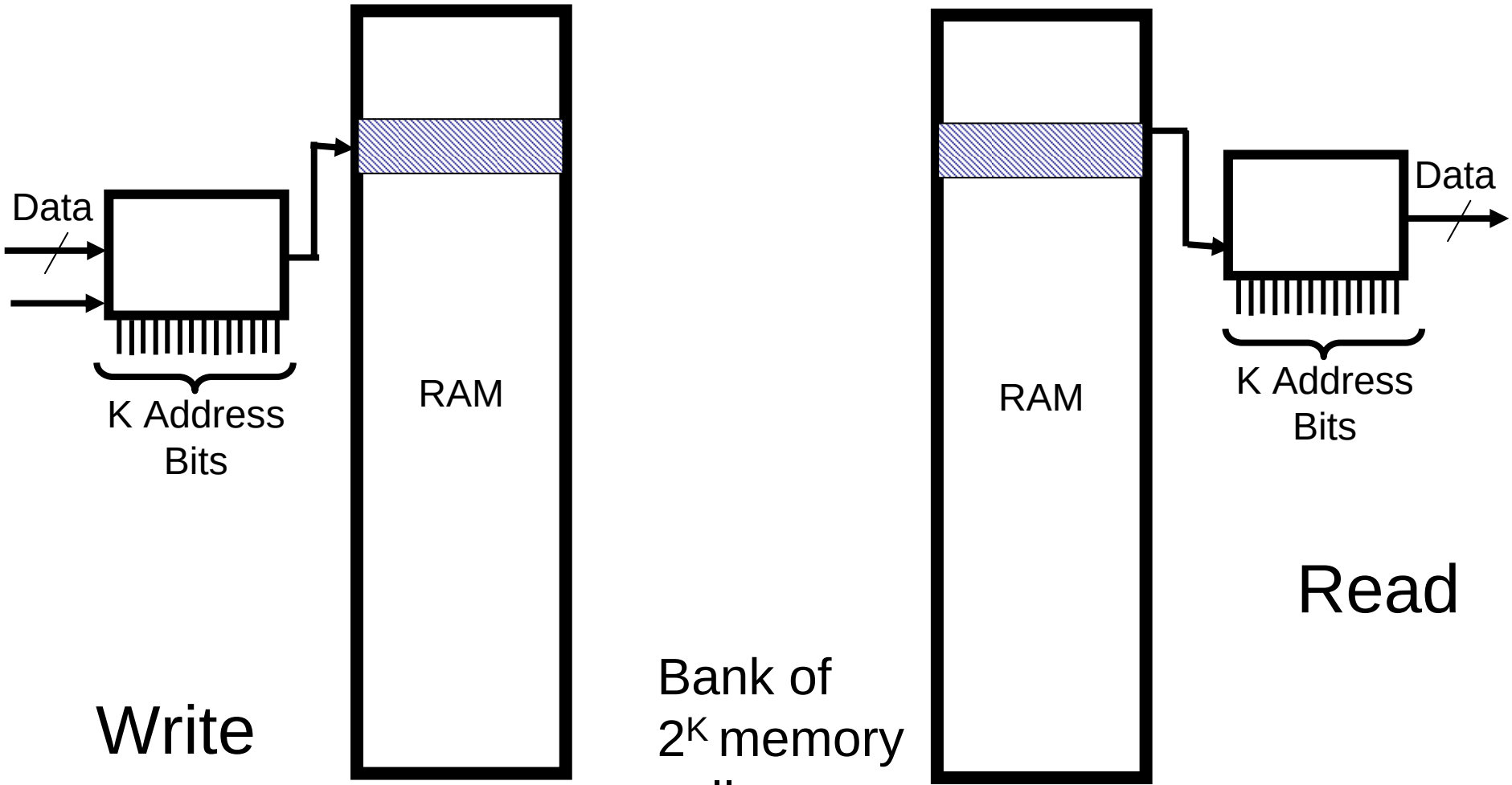


Before proceeding further…

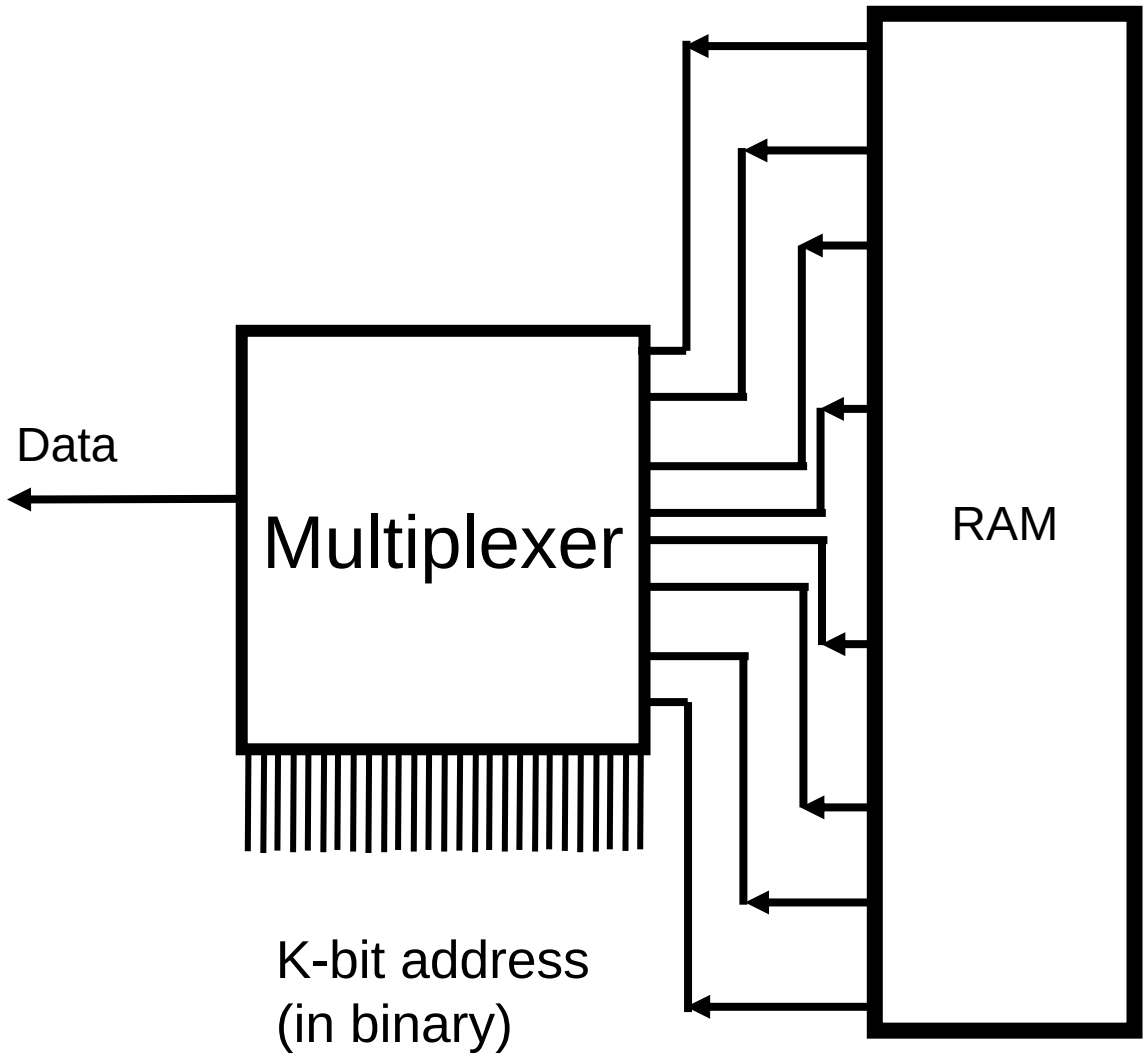
A word about Random Access Memory (RAM)

Memory where each location has an address

RAM

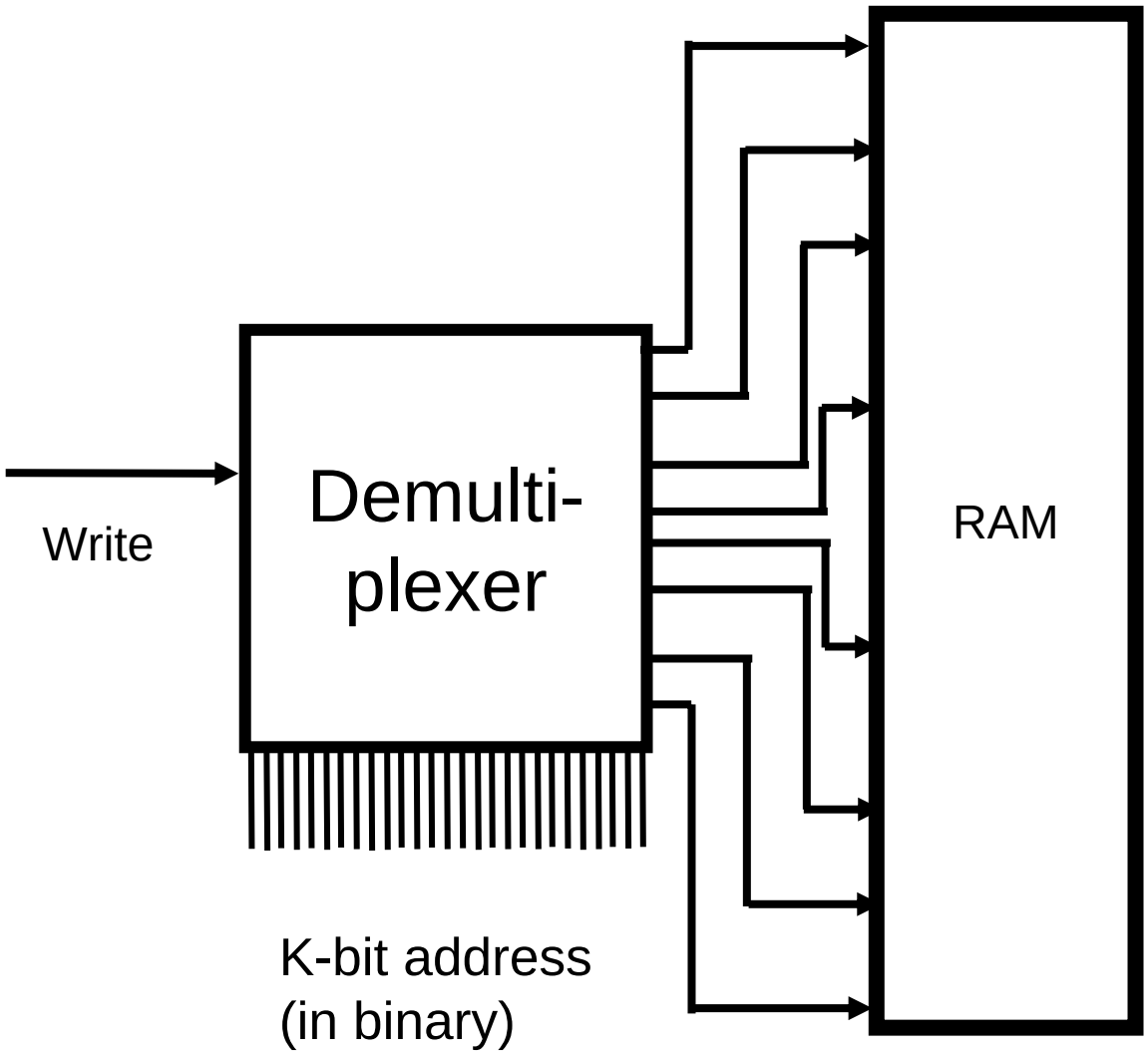

Data
K Address Bits
RAM
RAM
Data
K Address Bits
Write
Bank of 2^K memory cells
Read

RAM: Read



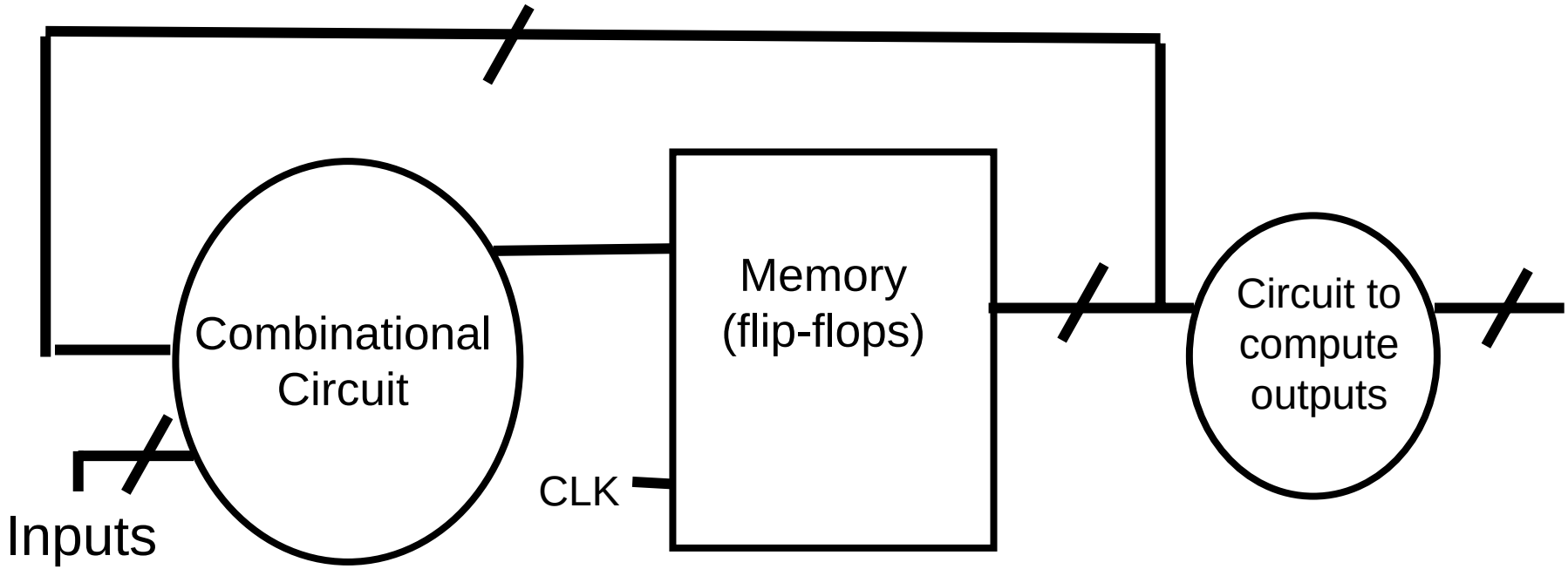
The multiplexer is connected to all 2^K cells in the RAM; selects the appropriate cell based upon the K-bit address

RAM: Write

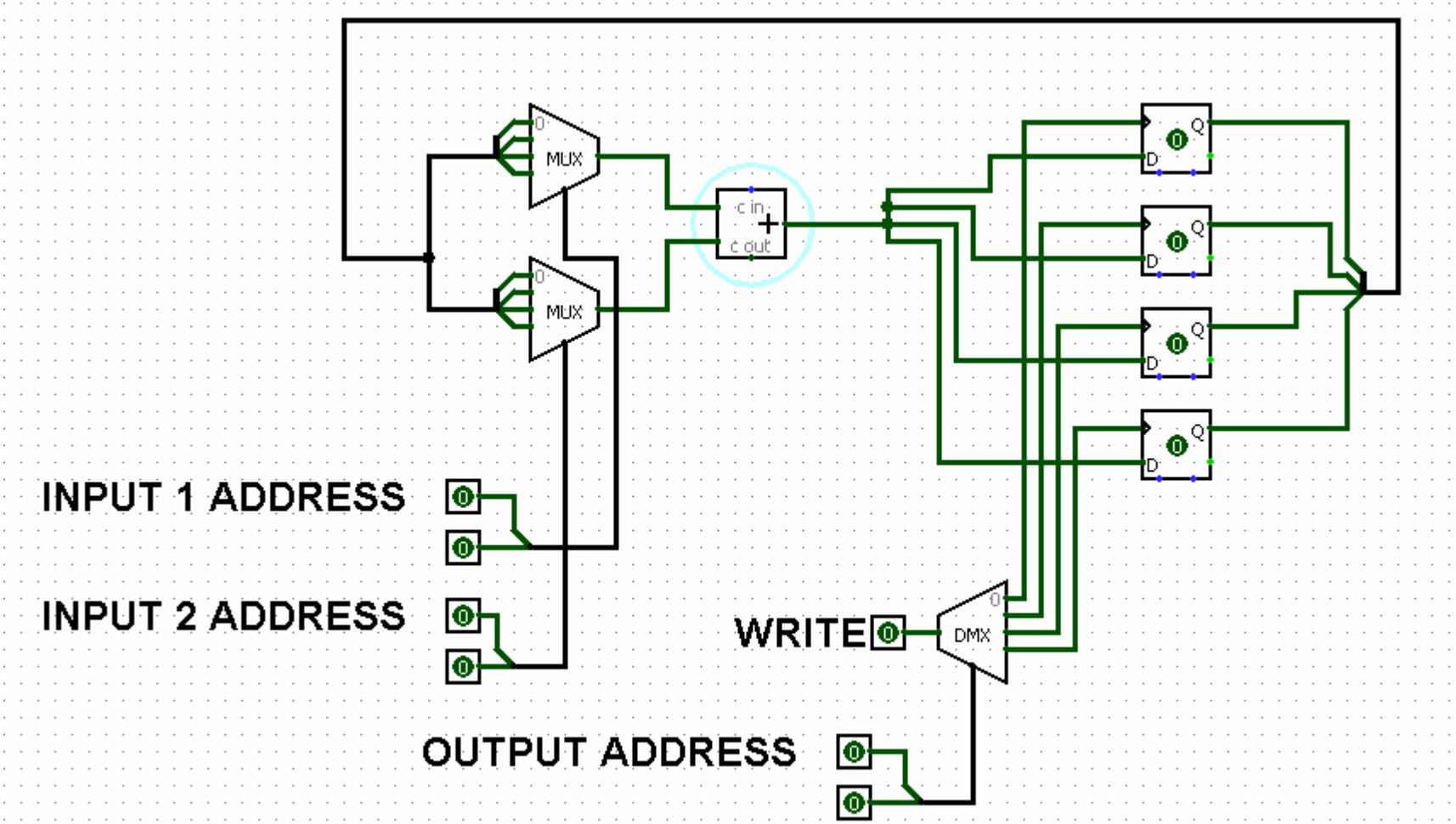


The k-bit address selects which of the 2^K cells in the RAM gets its “Write” input toggled

Recall the clocked synchronous circuit



Mini-CPU (demo)



Finally, the secret revealed...

How computers execute programs.

What is a program?

- A program is a sequence of binary numbers -- *instructions*.

- Each bit of each instruction corresponds to a control line in a programmable circuit (e.g. Pentium processor).

Scribbler Control Panel Program

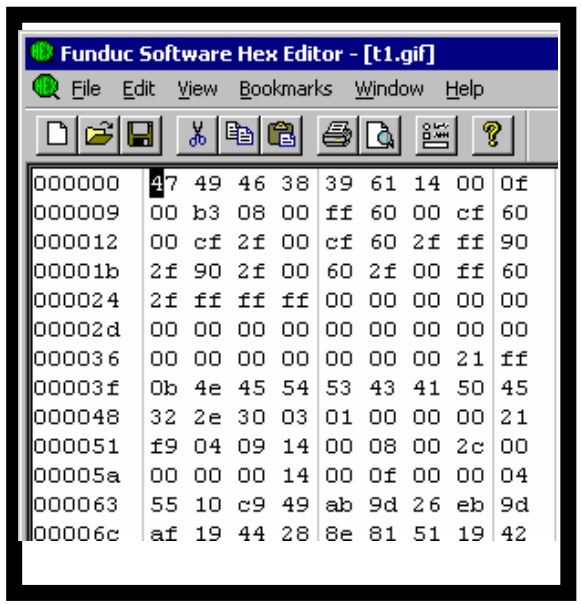
```
If <Obstacle on Either Side> Then
{
   Play Sound for 1s at Frequency 440Hz
}
Else
{
   LED: ON, ON, ON
}
END
```

F5

“Download to Robot”

(Compilation)

Machine Executable Code



Similar to:

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

Examples of Compilation

X ← Y + Z ↕ **ADD 10 11 12** ↕ 100 1010 1011 1100	1. Human writes this. 2. "Add contents of Location 11 and 12, and store result in Location 10" ▪ X in Location 10 ▪ Y in Location 11 ▪ Z in Location 12 3. Convert to binary

Recall Davis's binary encoding of T-P programs.

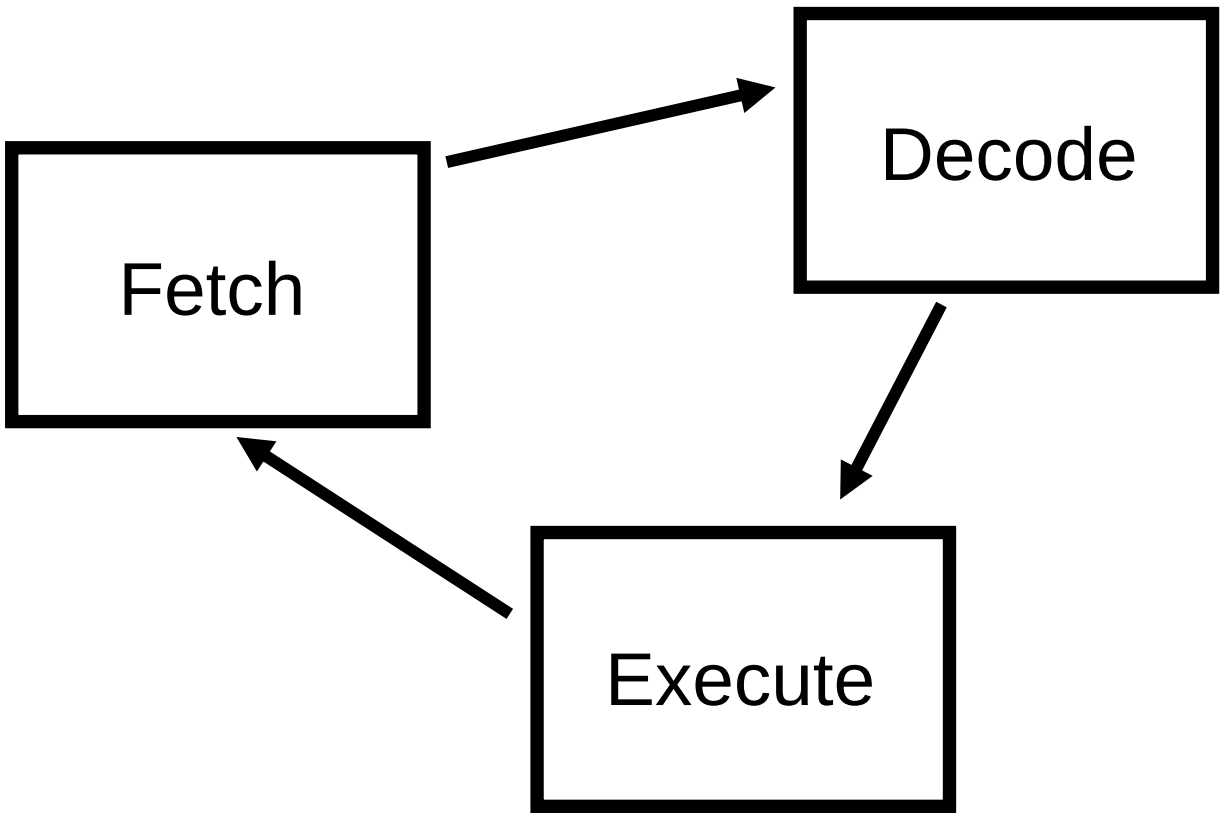
Examples of Compilation

```
If (x ≥ 0) Then …
        ↕
JUMPNEG 10      100
        ↕
101     1010    1100100
```

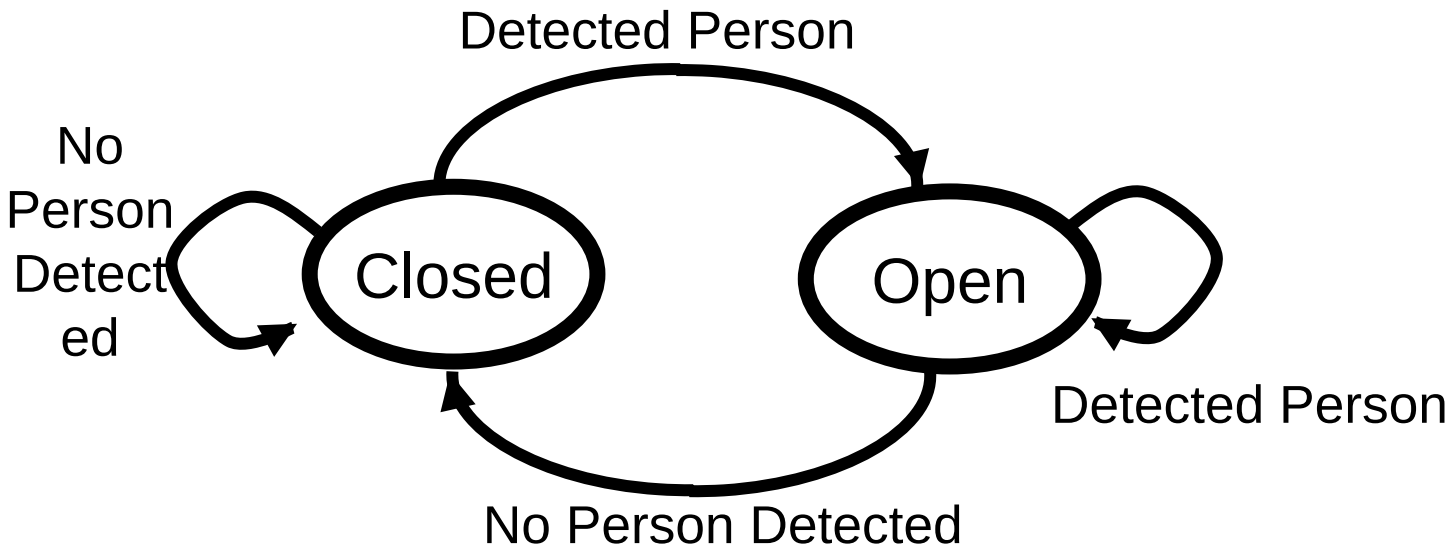
1) Human writes this.
2) "If Location 10 has a number < 0, set Instruction Pointer to 100"
   - X in Location 10
   - "Jump" to the Else block (starts at Location 100)
3) Convert to binary

Meet the little green man..

The Fetch – Decode – Execute cycle

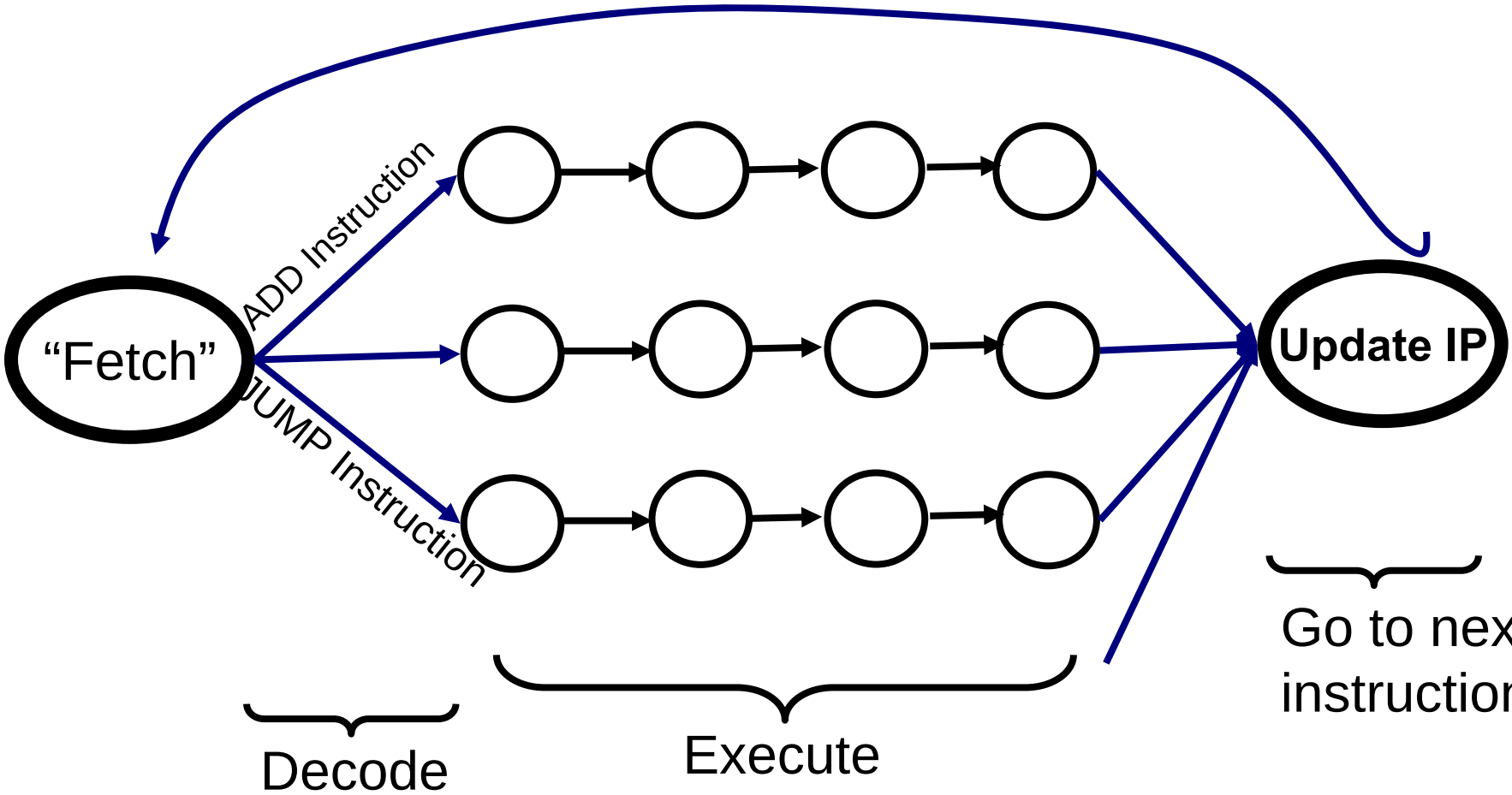


Recall: FSMs of Moore Types

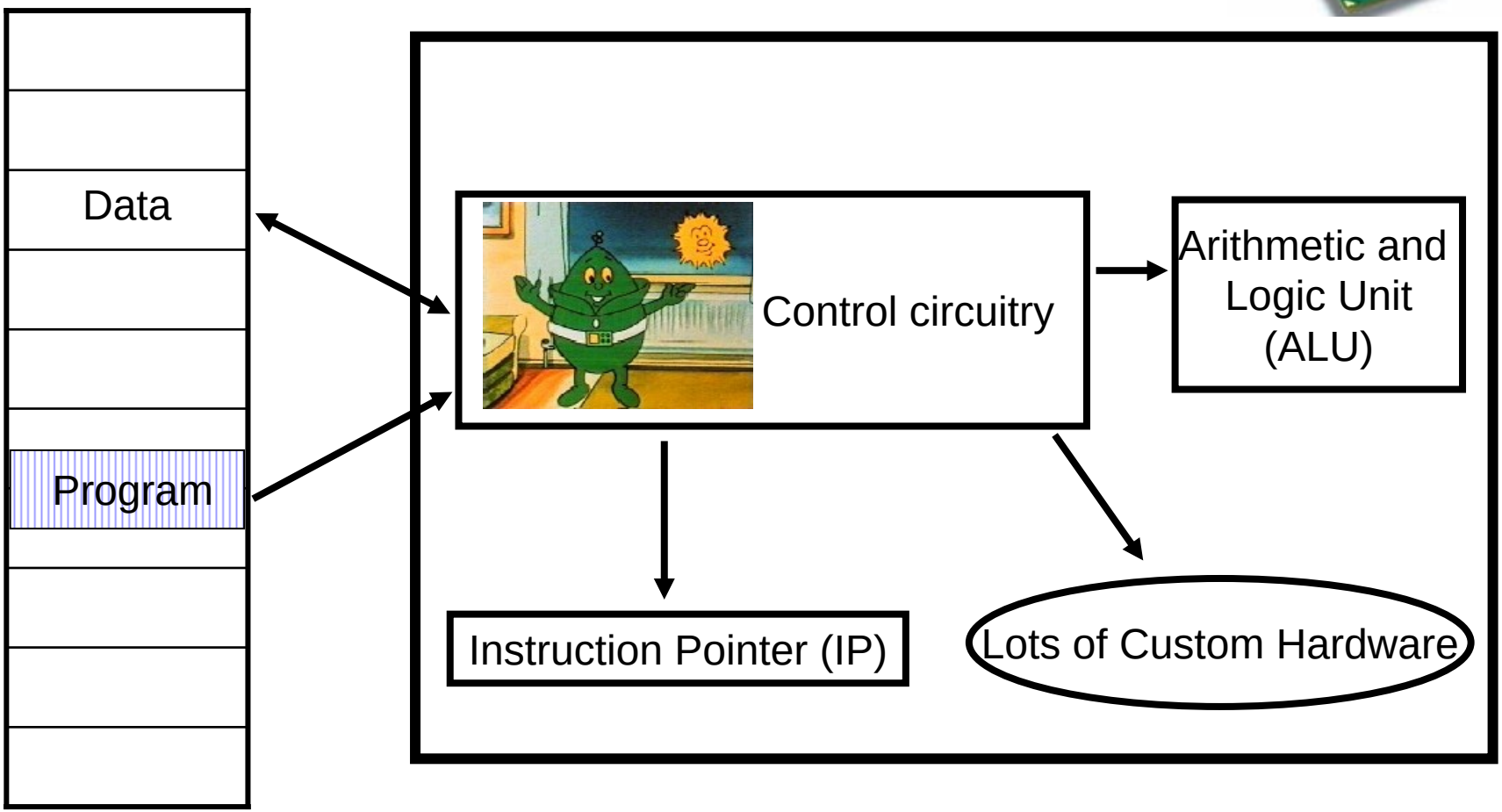


- Finite number of states
- Machine can produce outputs, these depend upon current state
- Machine can accept one or more bits of input; reading these causes transitions among states.

Fetch – Decode – Execute FSM



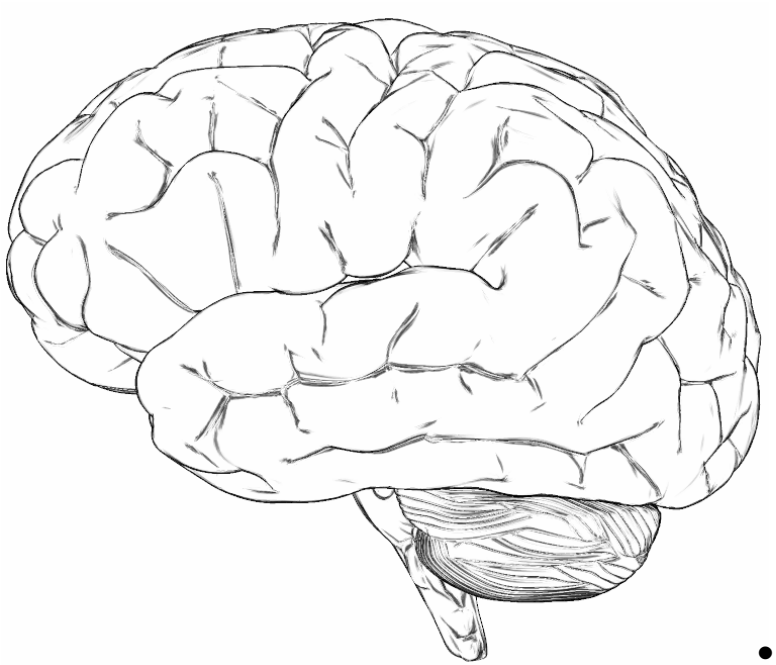
Greatly simplified view of modern CPUs.

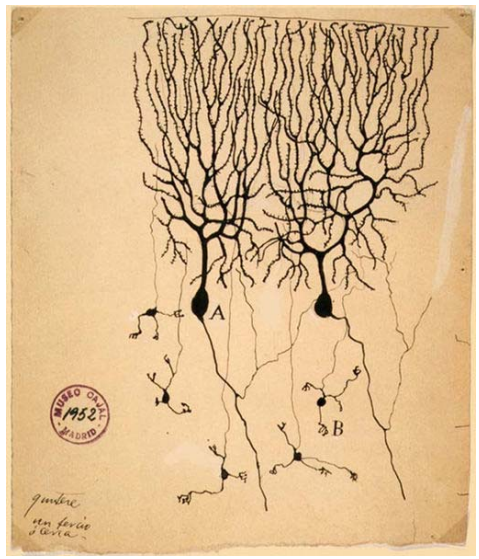


Main Insight

= FSM controlling a large memory.

Speculation: Brain as FSM?





- Network 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.