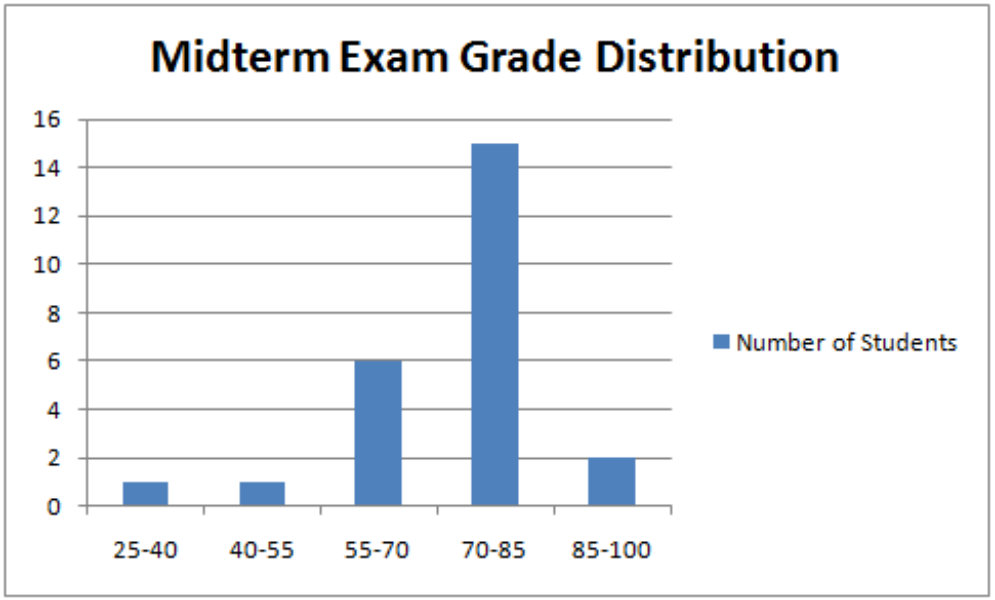
How circuits acquire memory: Sequential & Clocked Circuits.

COS 116, Spring 2011

Sanjeev Arora

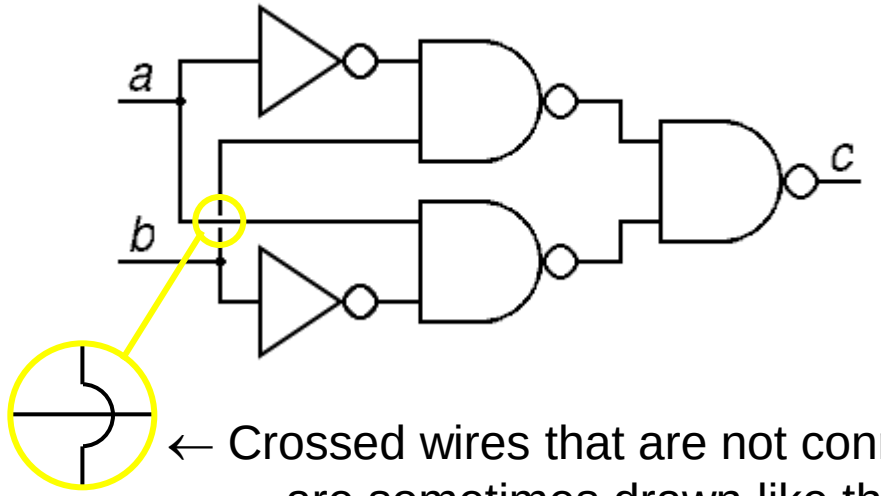
Midterm

Midterm exam scores



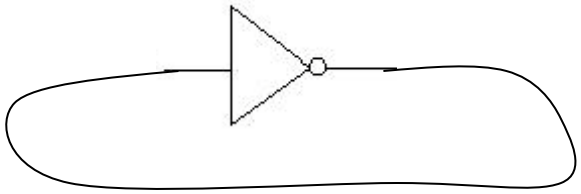
Formula for overall grade:
50% exam + 25% HW + labs, 25% participation

Recap: Combinational Circuits



← Crossed wires that are not connected are sometimes drawn like this.

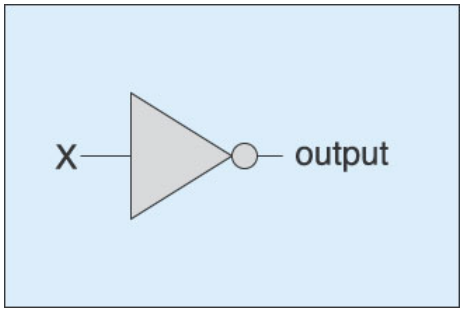
Wires: transmit voltage (and hence value)

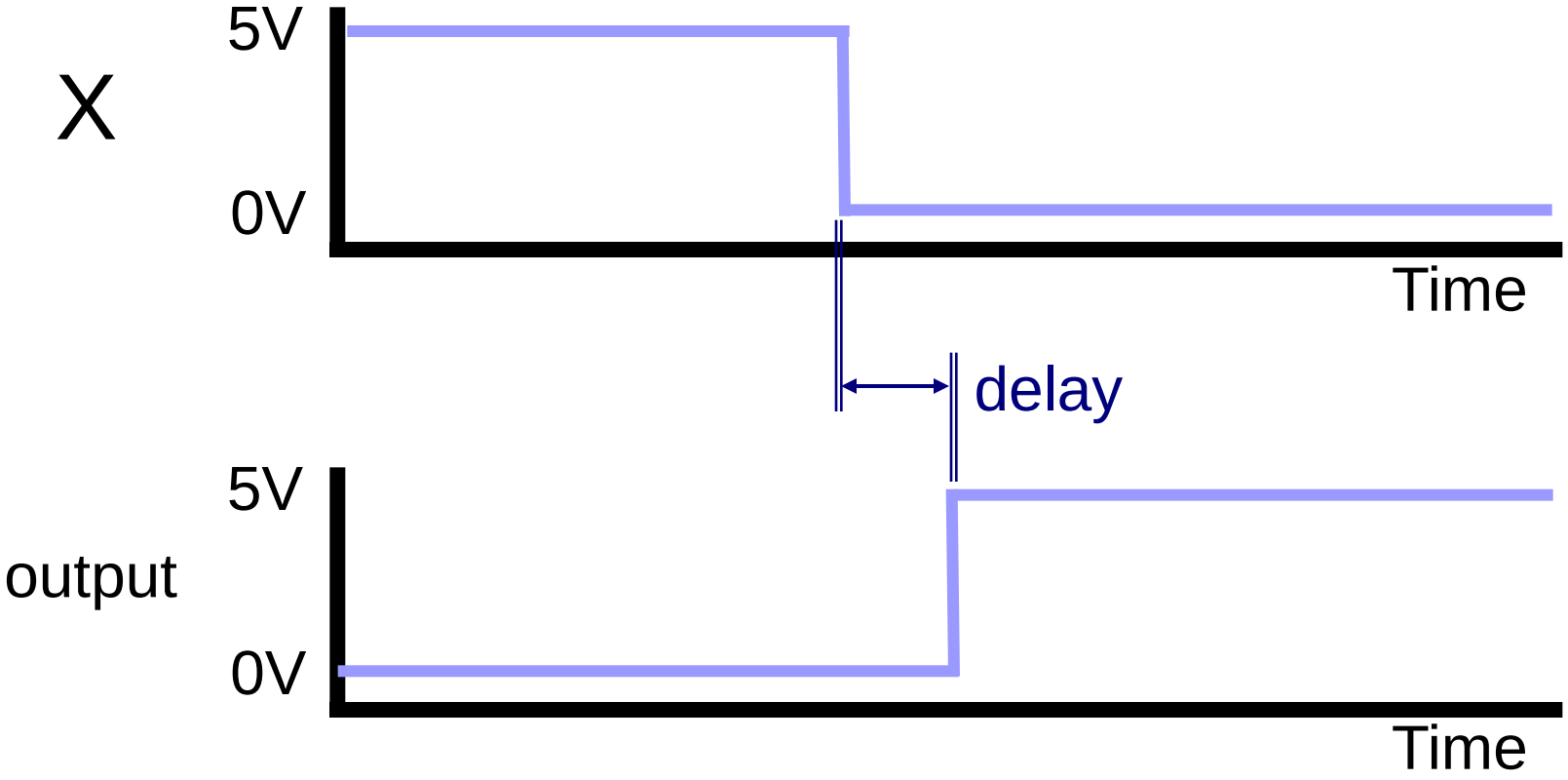


No loops allowed (direct or indirect)

Timing Diagram

NOT gate





Recap: Combinational circuit for binary addition?

25	11001
+29	11101
54	110110

- Want to design a circuit to add any two *N*-bit integers (say N =64).

Is the truth table method useful? Ideas?

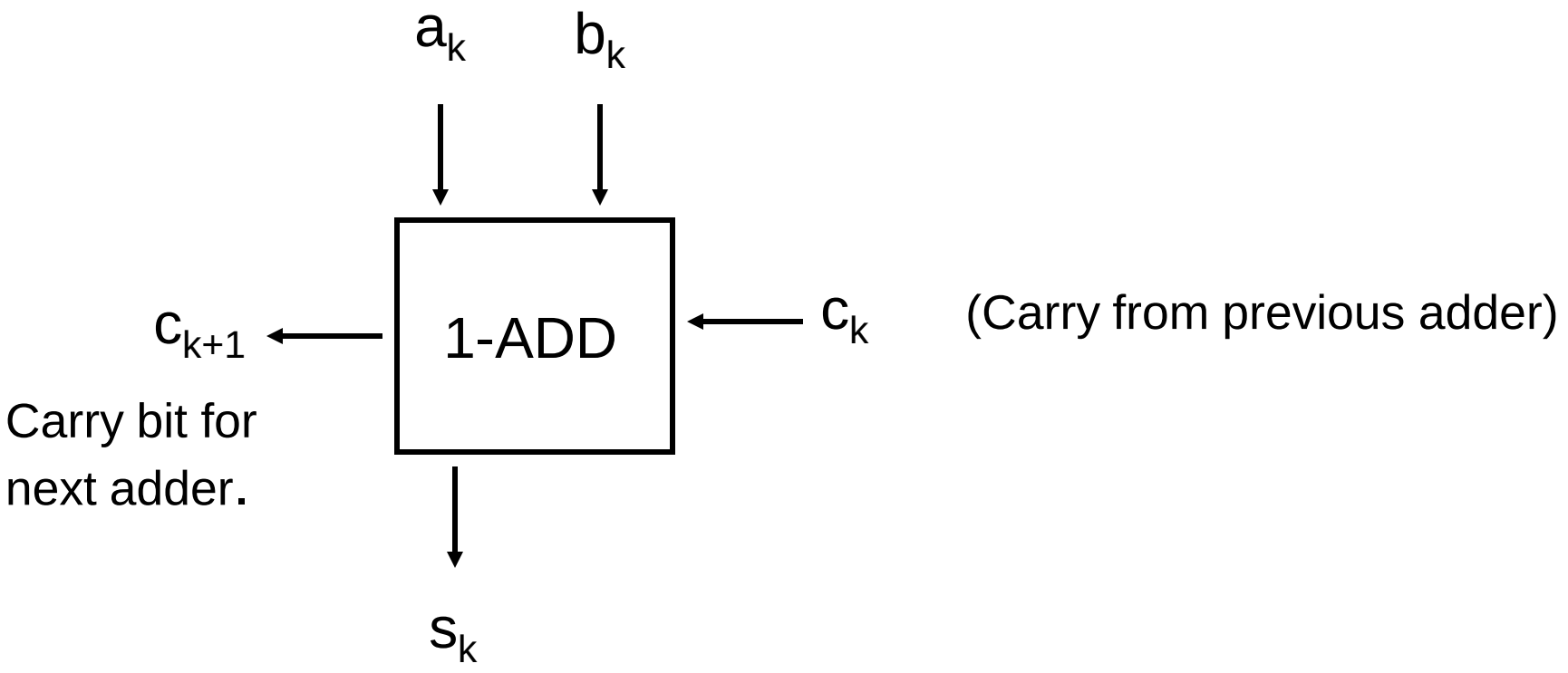
Modular design

Have small number
of basic components.

Put them together to achieve
desired functionality

Basic principle of modern industrial design;
recurring theme in next few lectures.

1-bit adder

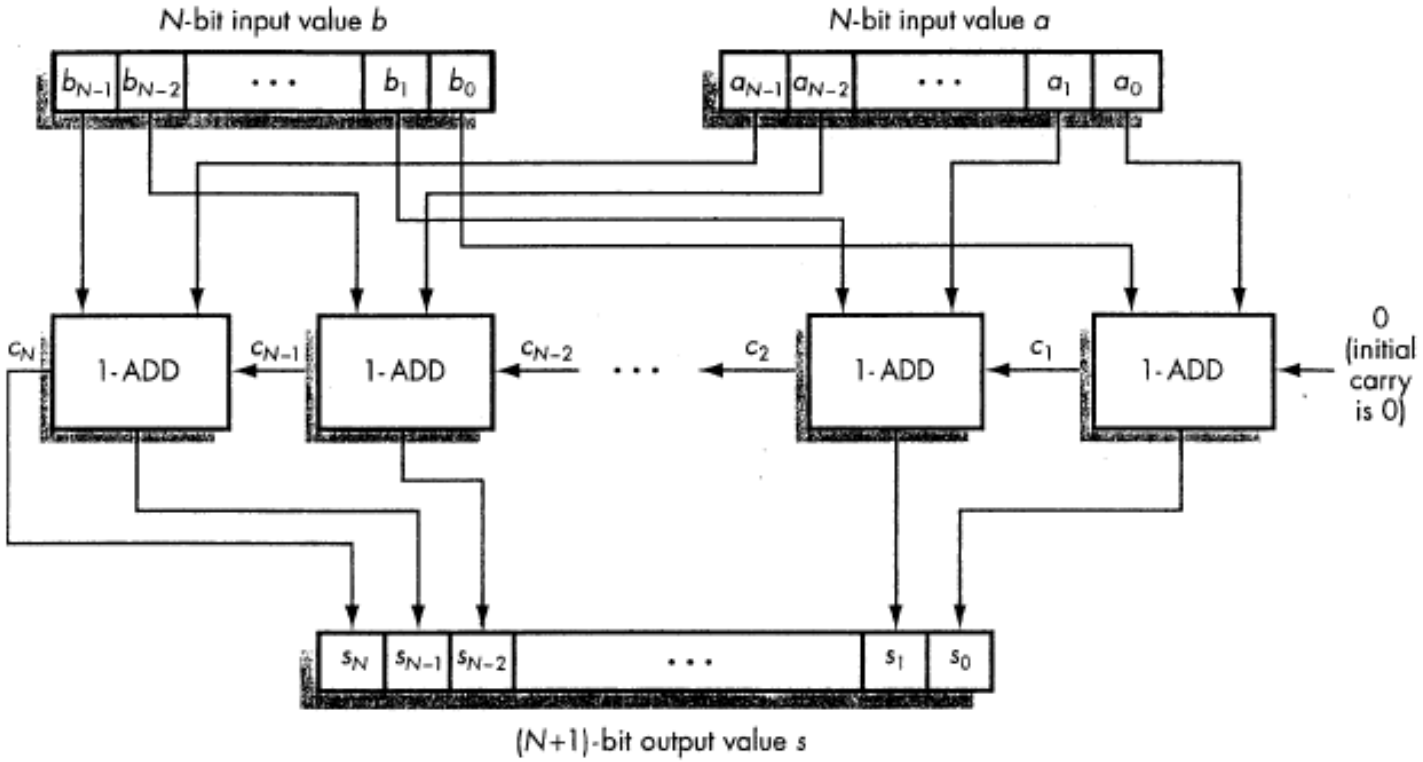


Hand in on Mar 22: Truth table, circuit for 1-bit adder.

Modular Design for boolean circuits

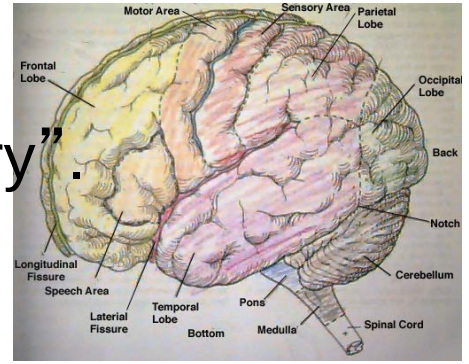
An N-bit adder using N 1-bit adders
(will do Mar 22)

A Full Adder (from handout)



Memory

Rest of this lecture:
How boolean circuits can have “memory”.



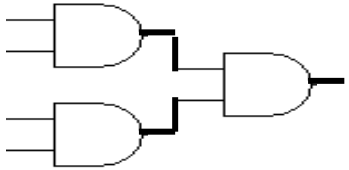
What do you understand by ‘memory”?

How can you tell that a 1-year old child has it?

Behaviorist’s answer: His/her actions depend upon past events.

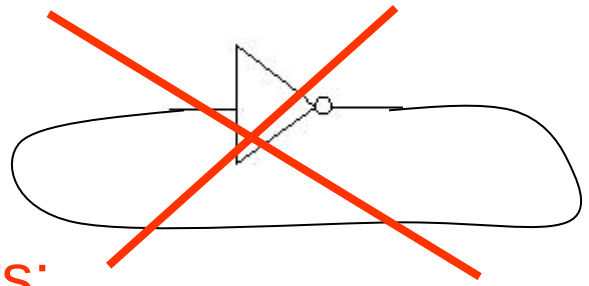
Why combinational circuits have no "memory"

- Boolean gates connected by wires



Wires: transmit voltage (and hence value)

- Important: no loops allowed

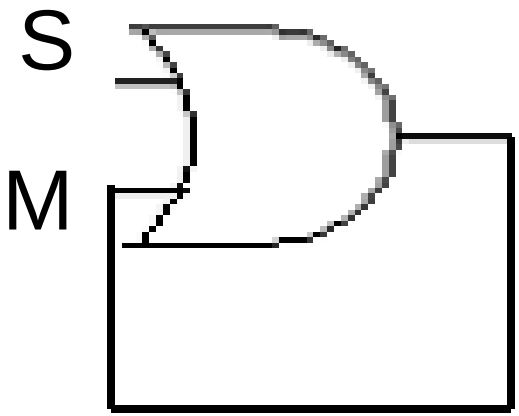


Output is determined by current inputs; no "memory" of past values of the inputs.

Today: Circuits with loops; aka "Sequential Circuits"

Matt likes Sue but he doesn't like changing his mind

- Represent with a circuit: Matt will go to the party if Sue goes or if he already wanted to go



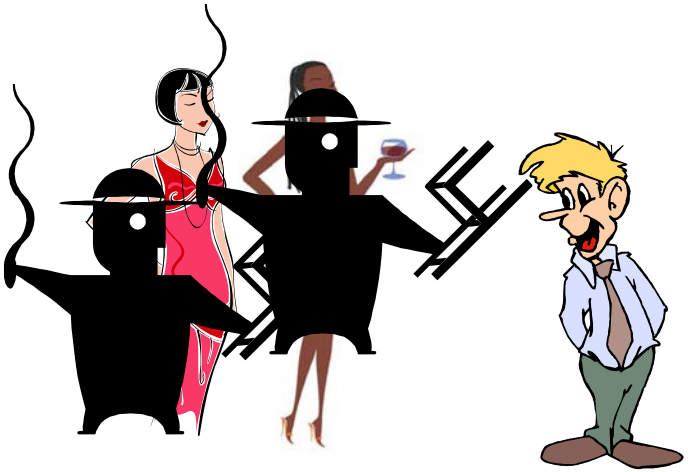
Is this well-defined?

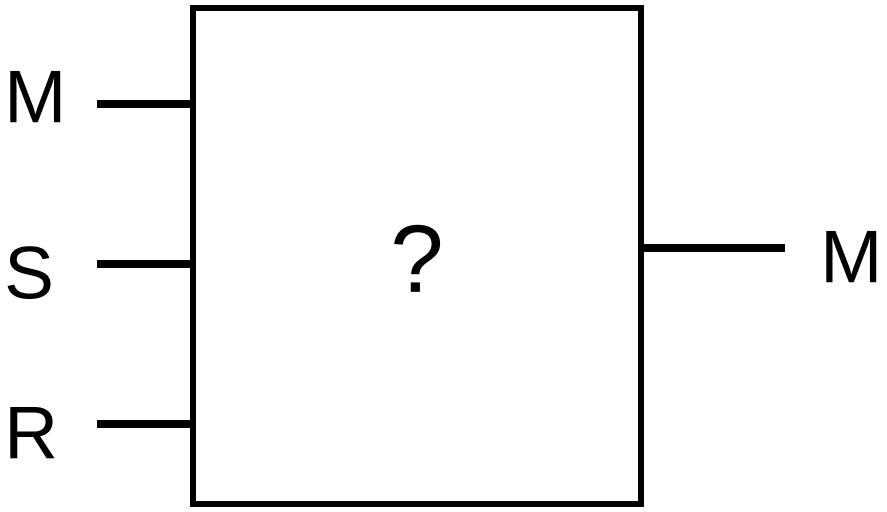
Sequential Circuits

- Circuits with AND, OR and NOT gates.
- Cycles ***are*** allowed (ie outputs can feed back into inputs)
- Can exhibit “memory”.
- Sometimes may have “undefined” values
- How to write the “truth table”? Suggestions?

Enter Rita

- Matt will go to the party if Sue goes OR if the following holds: if Rita does not go *and* he already wanted to go.

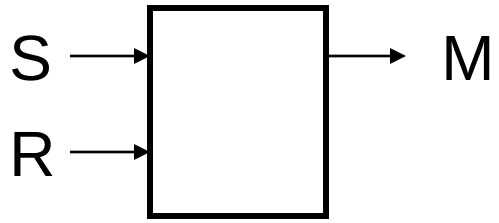


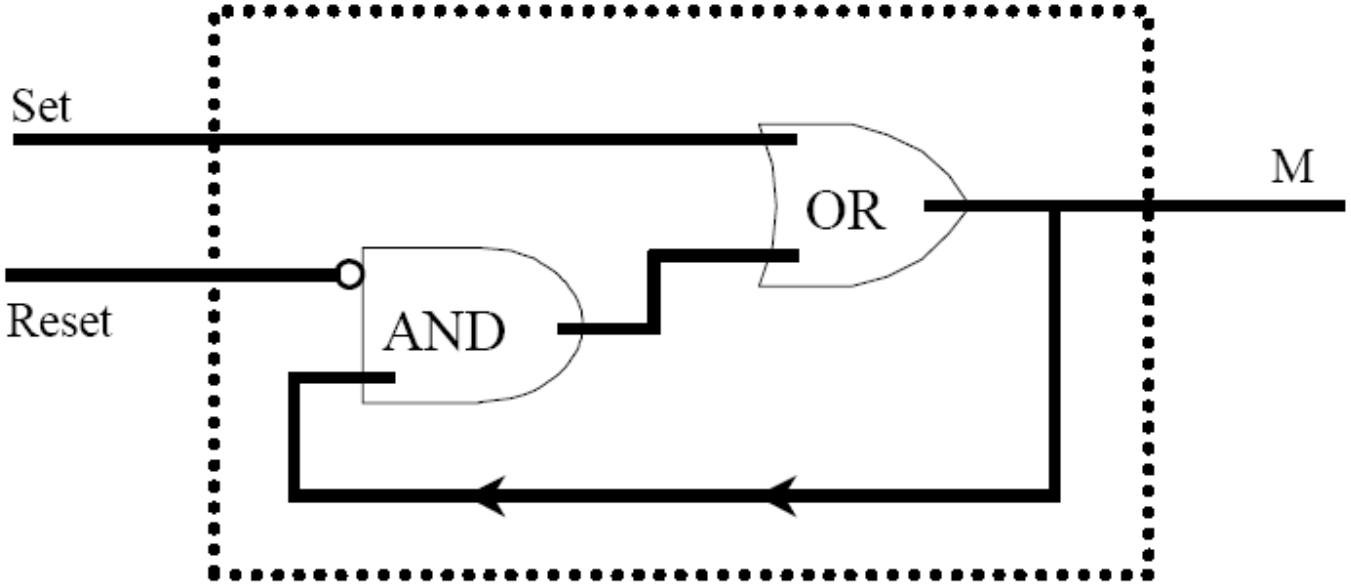


R, S: "**control**" inputs

What combination of R, S changes M?

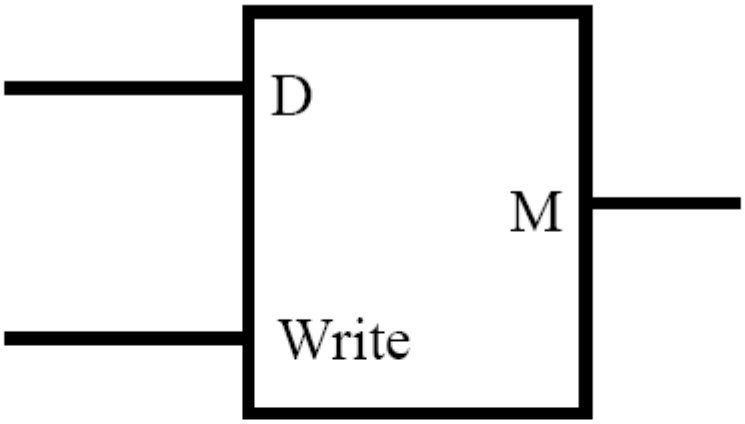
R-S Latch





- M becomes 1 if Set is turned on
- M becomes 0 if Reset is turned on
- Otherwise (if both are 0), M just remembers its value

A more convenient form of memory

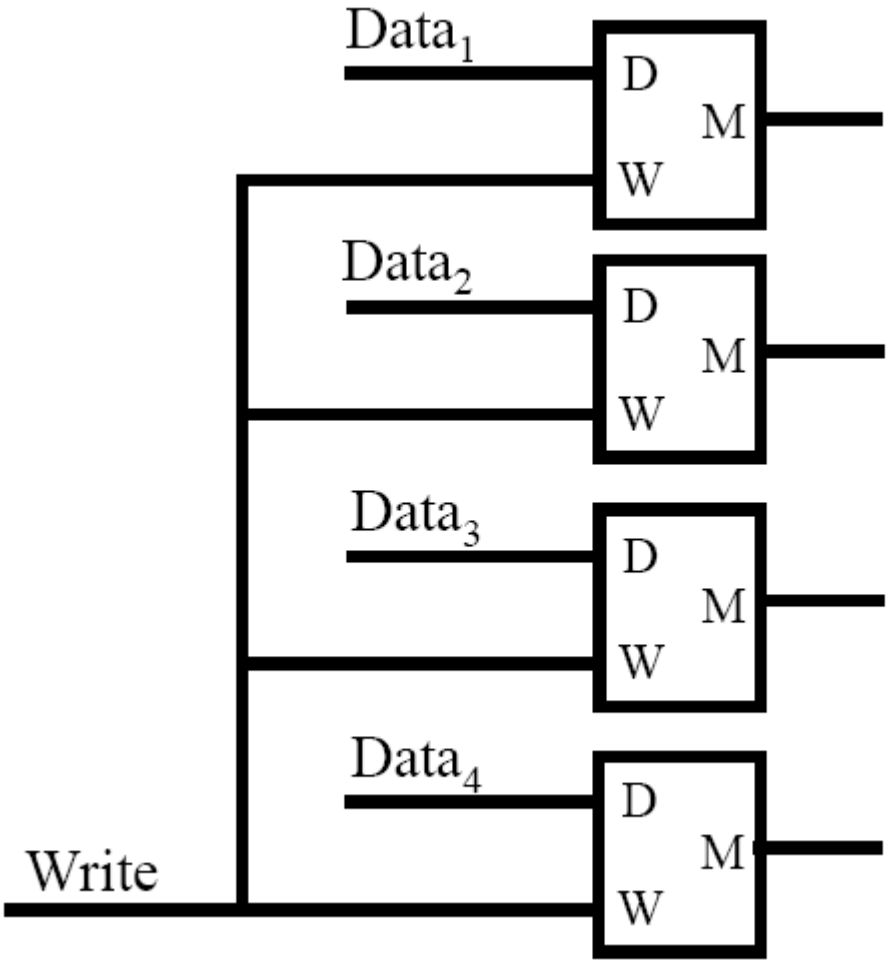


No "undefined" outputs ever!

- If Write = 0, M just keeps its value. (It ignores D.)
- If Write = 1, then M becomes set to D

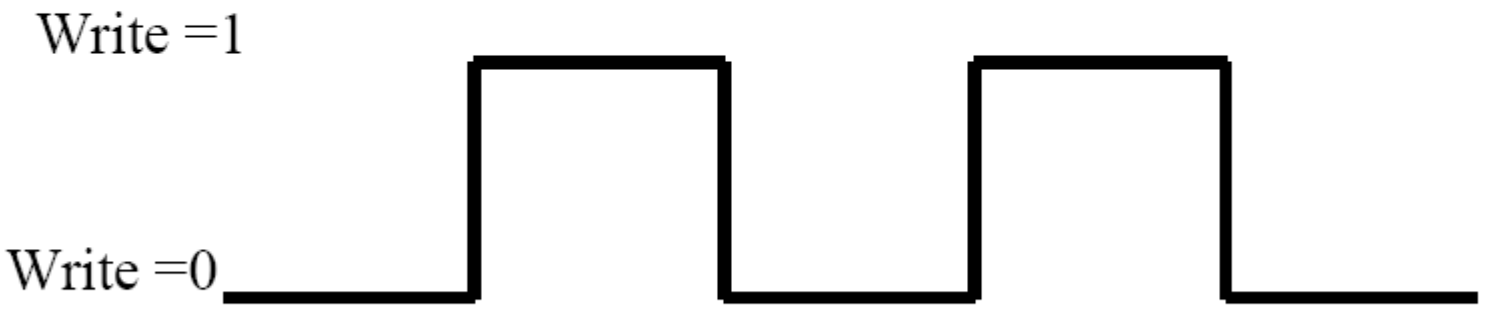
Fact: "Data Flip-Flop" or "D flip flop";
can be implemented using two R-S flip flops.

“Register” with 4 bits of memory



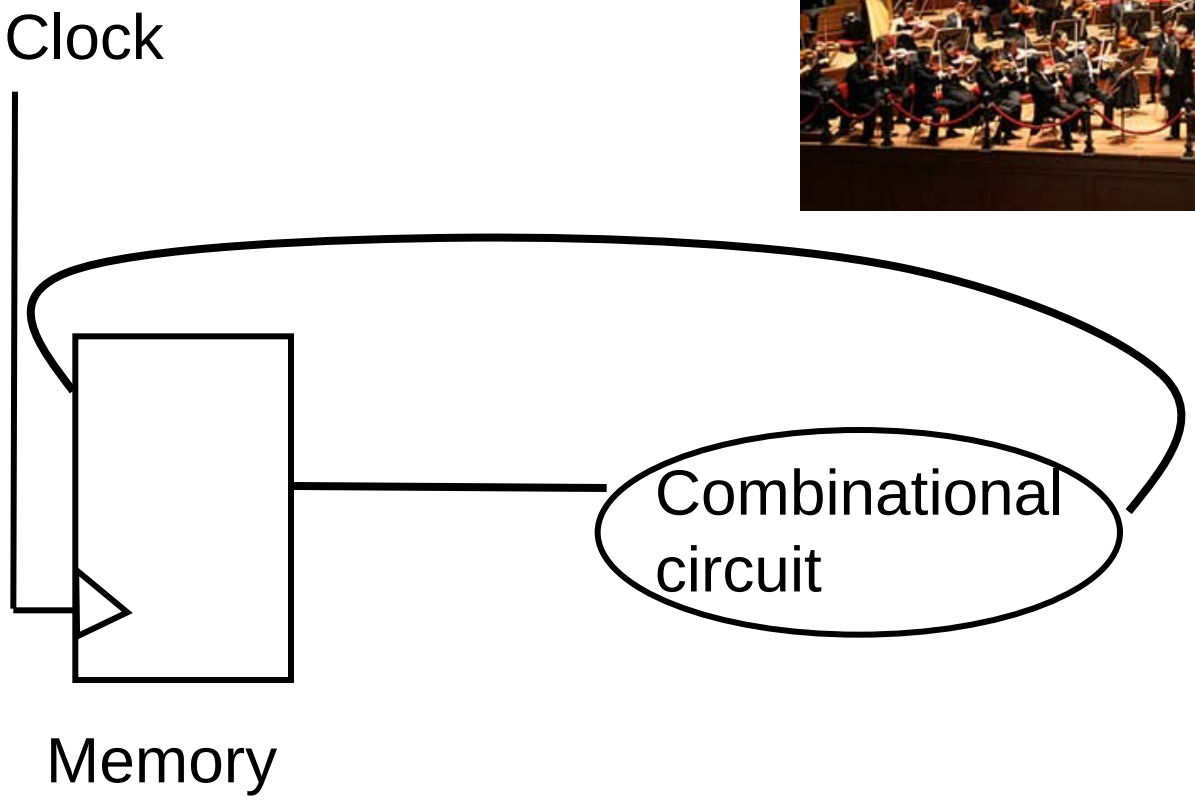
What controls the “Write” signal?

- Often, the system clock!
- “clock” = device that sends out a fluctuating voltage signal that looks like this



“Computer speed” often refers to the clock frequency (e.g. 2.4GHz)

The “symphony” inside a computer



Clocked Sequential Circuit (aka Synchronous Circuits)

Timing diagram

(analog of truth table for sequential circuits)

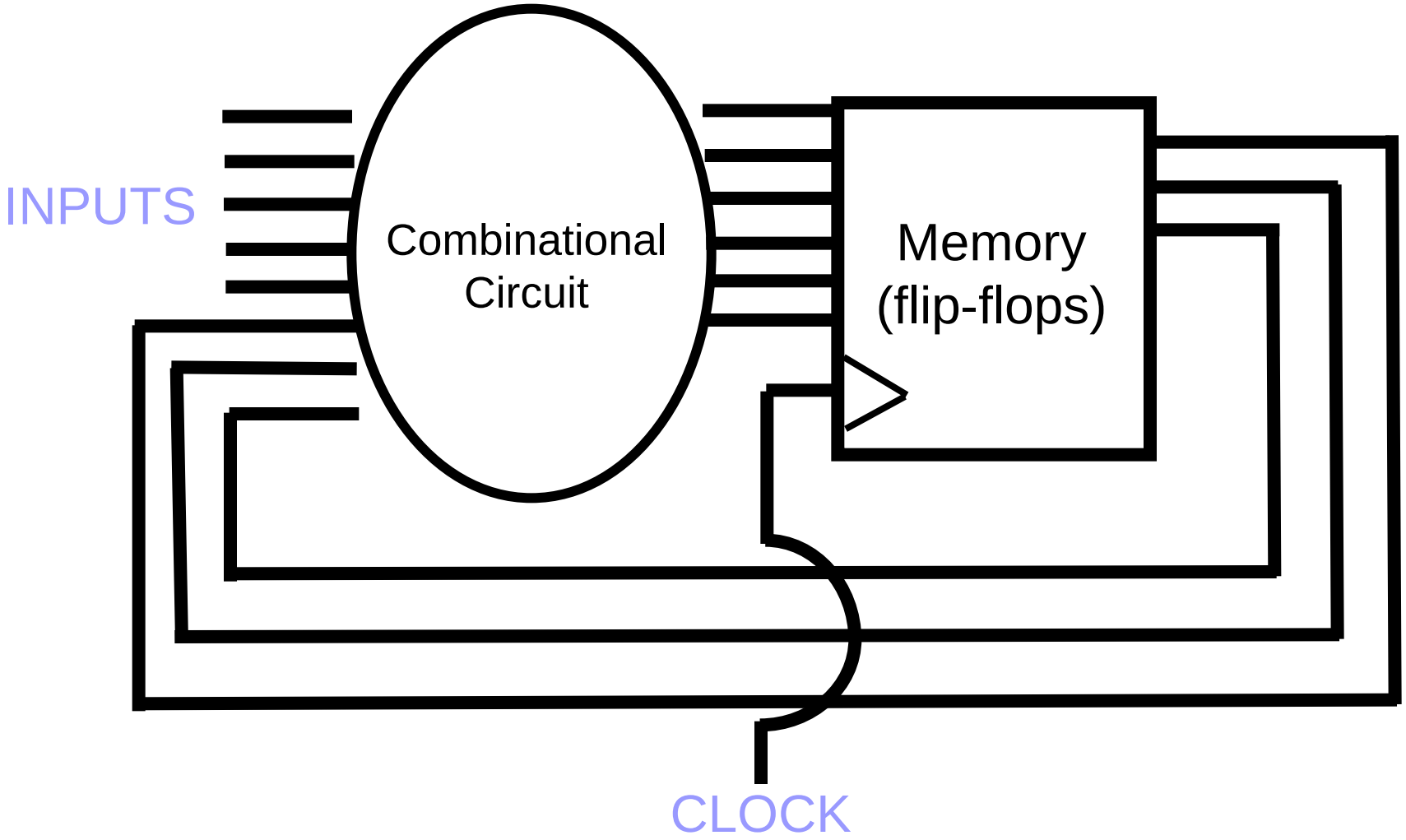
R	S	M(t)	M(t+1)
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Fill in timing diagram for RS latch and hand in on a piece of paper.

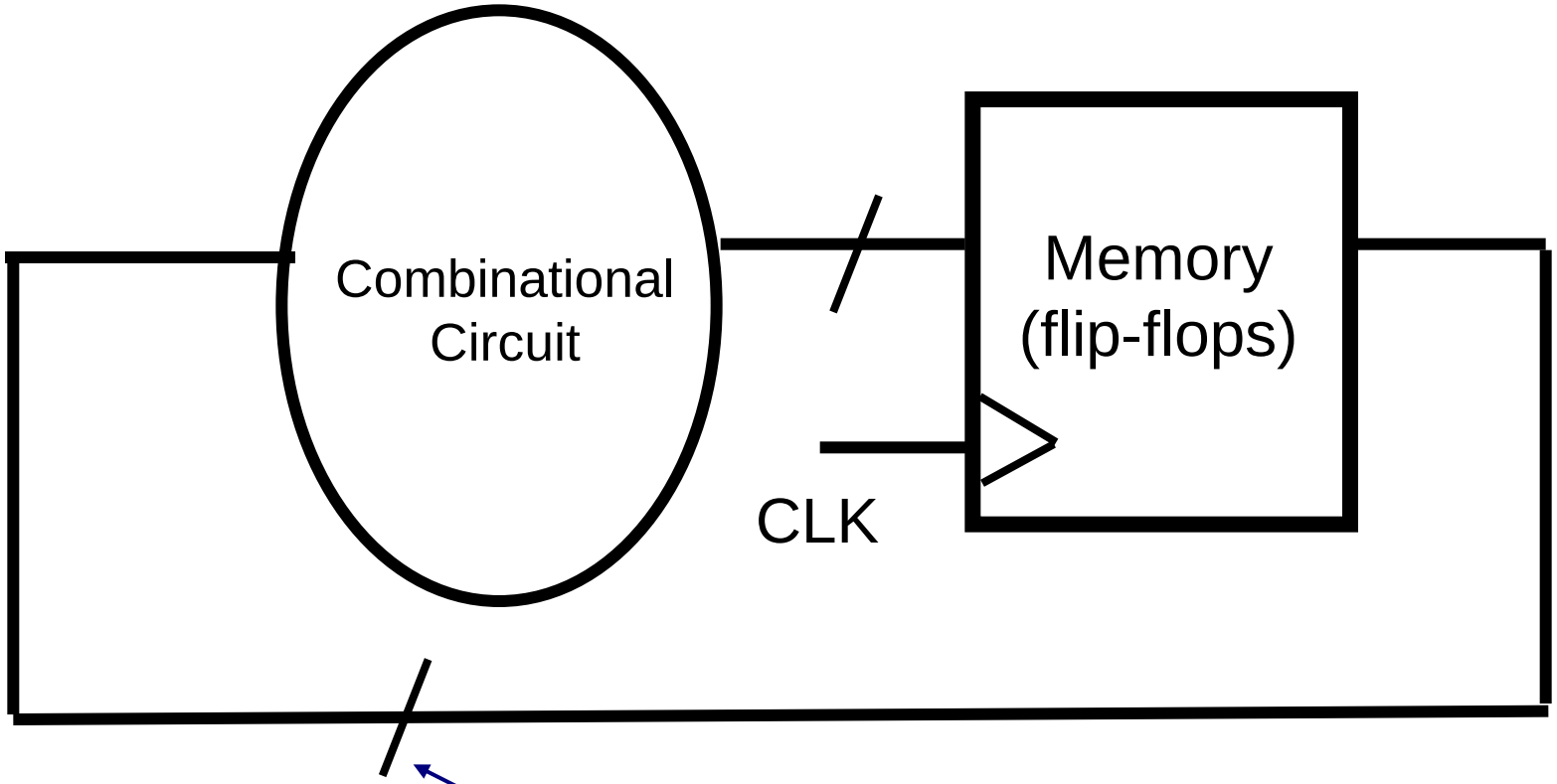
Clocked Sequential Circuits

Synchronous Sequential Circuit

(aka Clocked Sequential Circuit)



Shorthand

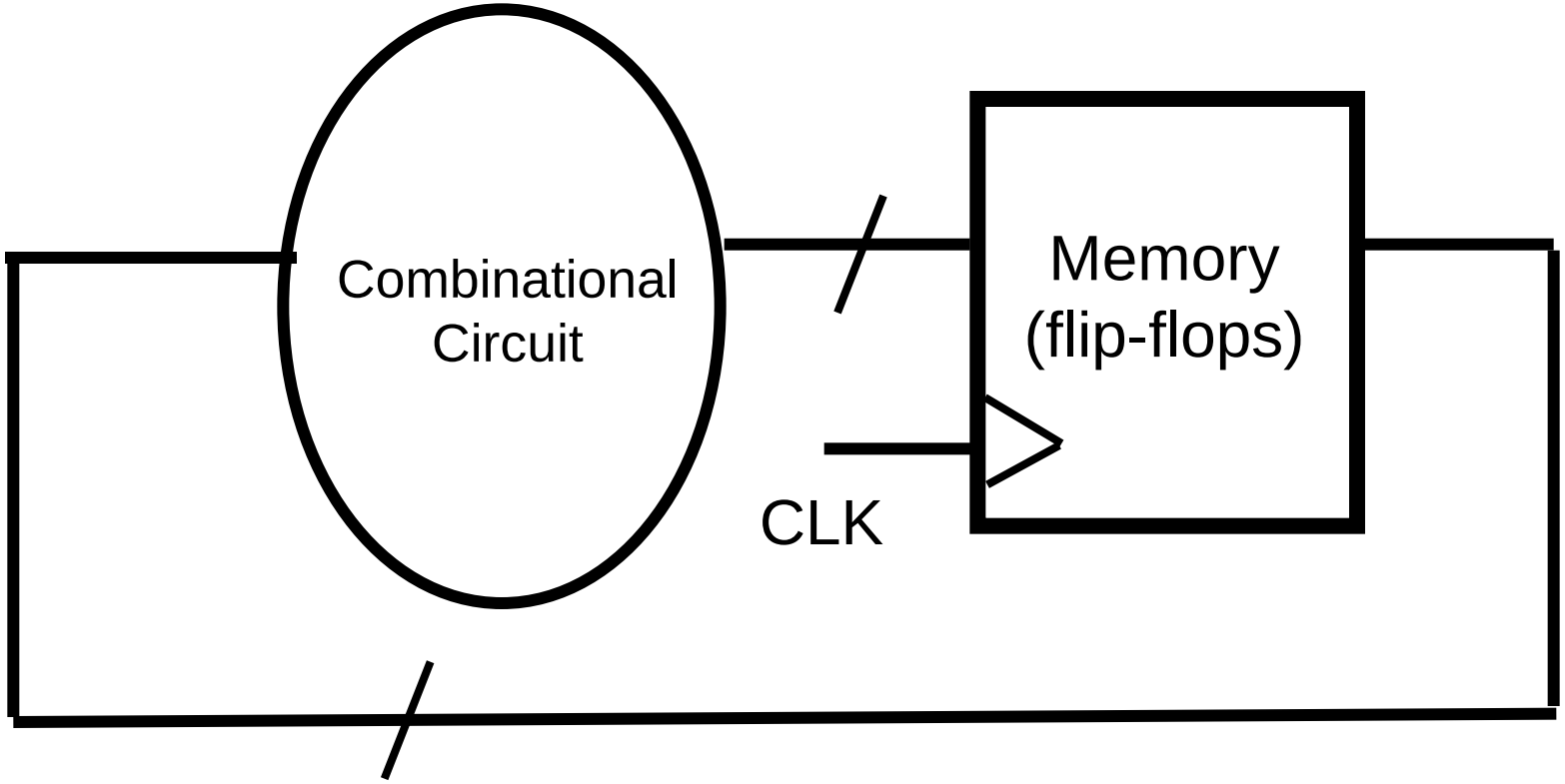

Combinational
Circuit
Memory
(flip-flops)
CLK
This stands for “lots of wires”
(aka “bus”)

Clock Speeds

1974	Intel 8080	2 MHz (Mega = Million)
1981	Original IBM PC	4.77 MHz
1993	Intel Pentium	66 MHz
2005	Pentium 4	3.4 GHz (Giga = Billion)

Heinrich Hertz
1857-94

What limits clock speed?

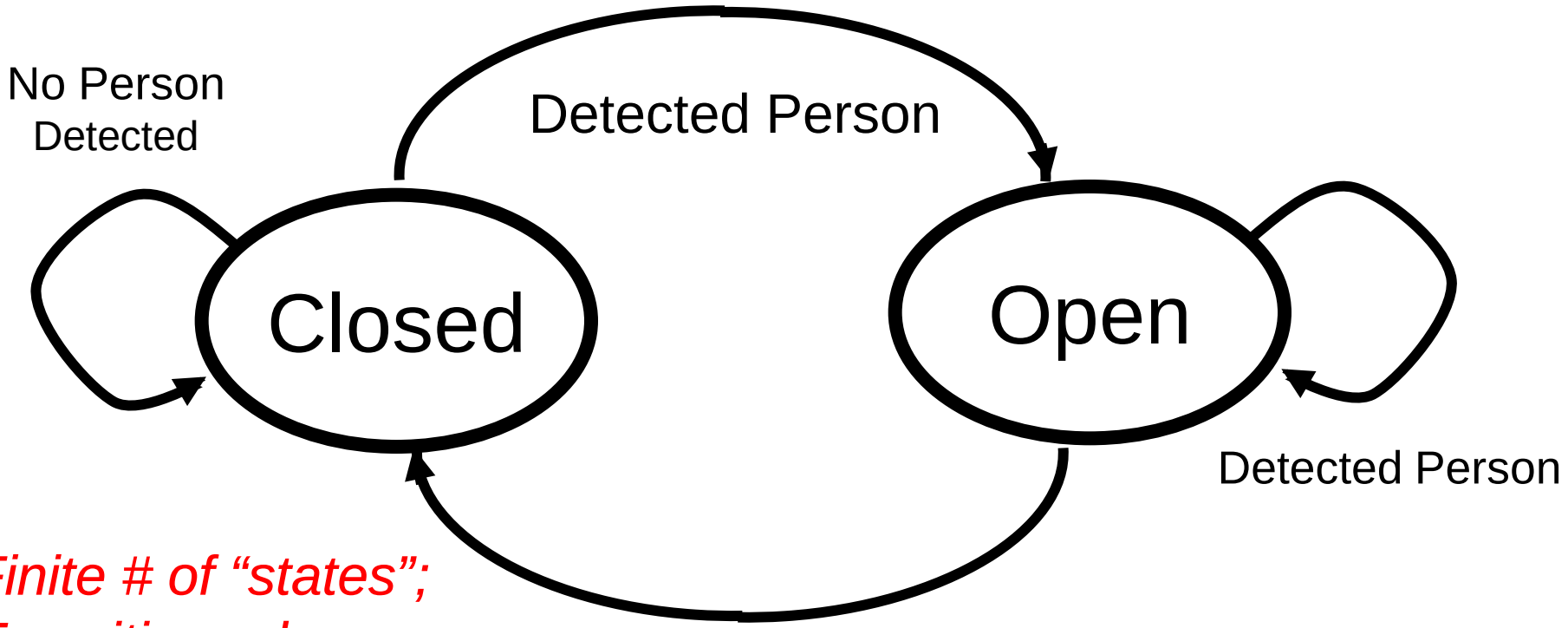


Delays in combinational logic (remember the adder).
Clock cycle = time needed for circuit value to settle.
During 1 clock cycle of Pentium 4, light travels: **4 inches**

Finite State Machines

Read handout (Brian Hayes article) for next time.

Example: State diagram for automatic door



Finite # of "states";
Transition rules between states

Next lectures…

- FSMs
- Computer organization: CPUs and RAM
- Lessons from computer architecture.