CMSC 11:
Introduction to Computer Science

Jaderick P. Pabico <jppabico@uplb.edu.ph>
Institute of Computer Science, CAS, UPLB

Review

- Flip-flop in action : how flip-flop remembers the previous output

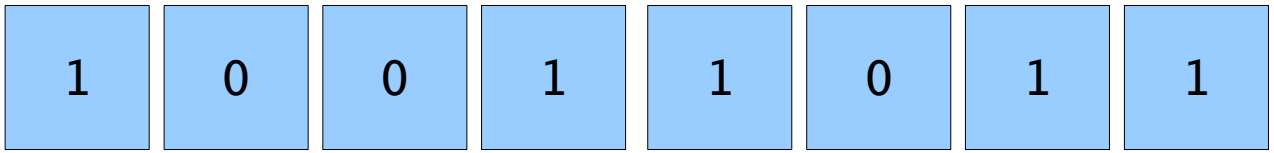
S	R	Q	Not-Q
1	1	No change	
1	0	0	1
0	1	1	0
0	0	Disallowed	

Flip-flop trivia

- Flip-flop is also called a LATCH
- Because it “locks in” data

Registers

- If the flip-flop is a device for storing one bit, a REGISTER stores several bits simultaneously
- It's like a row of boxes, each holding one bit



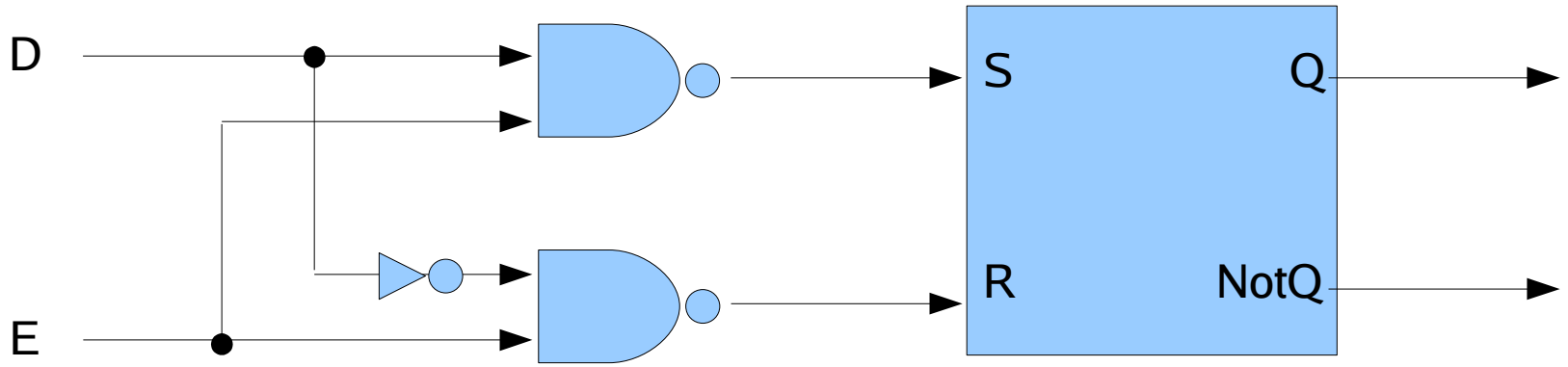
- Does a row of flip-flops do the job?

Well, sort of!

BUT, if you try and make this work by hooking up some inputs to R-S flip-flops, you may find yourself growing confused!

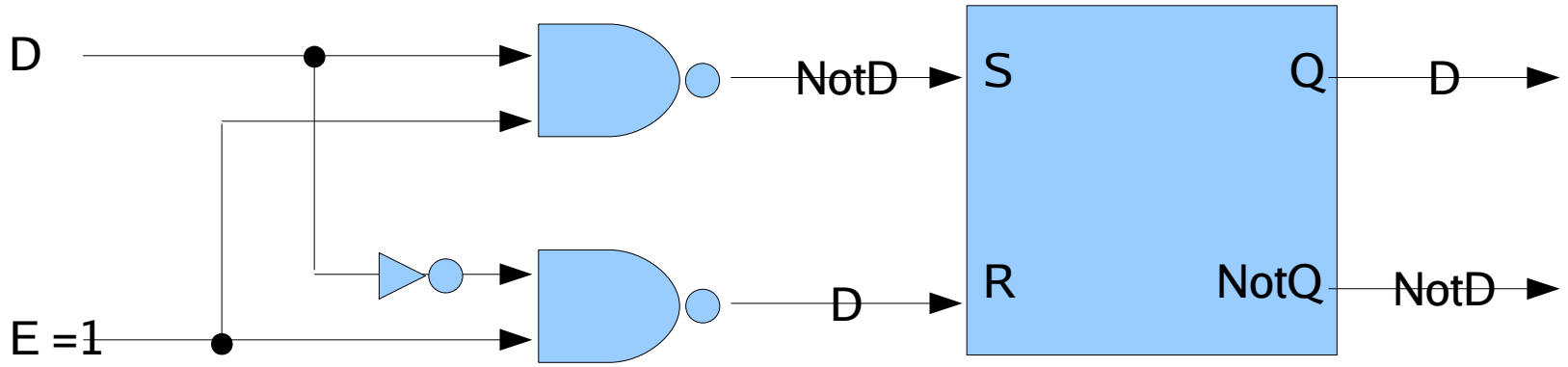
Registers

- The solution is to add a GATING NETWORK to the basic R-S flip-flop:



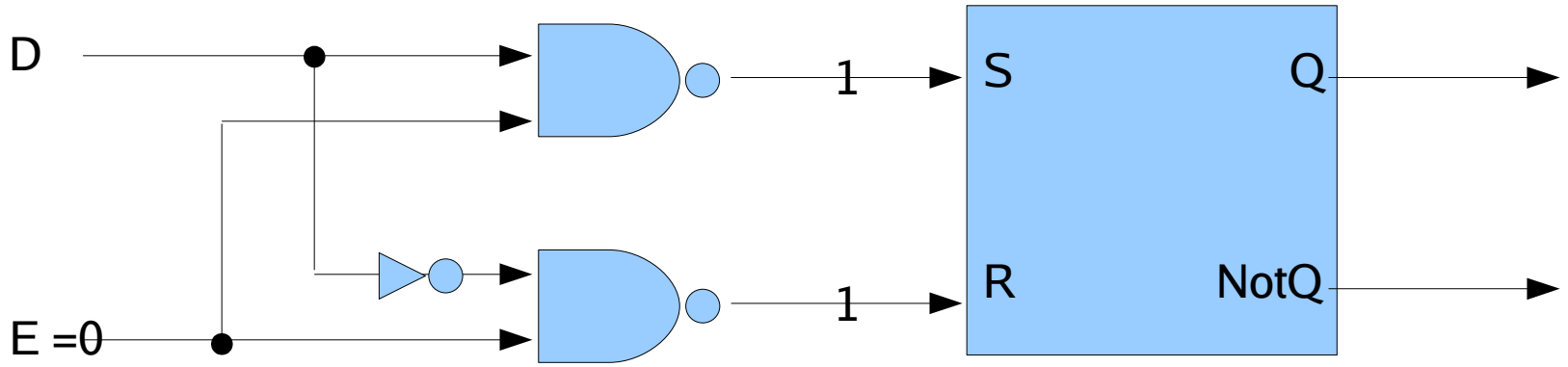
- D = data, E= enable
- The GATING NETWORK does not allow the DISALLOWED state for the flip-flop

Registers



- When E=1, then R=D and S=NotD.
- Hence, the value of D is stored at Q
- IOW, E=1 ENABLES the bit D to be loaded into the flip-flop

Registers



- When E=0, then S and R both become 1,
- and the flip-flop does not changes
- IOW, E=0 BLOCKS the arrival of more data.

Boxes in a box

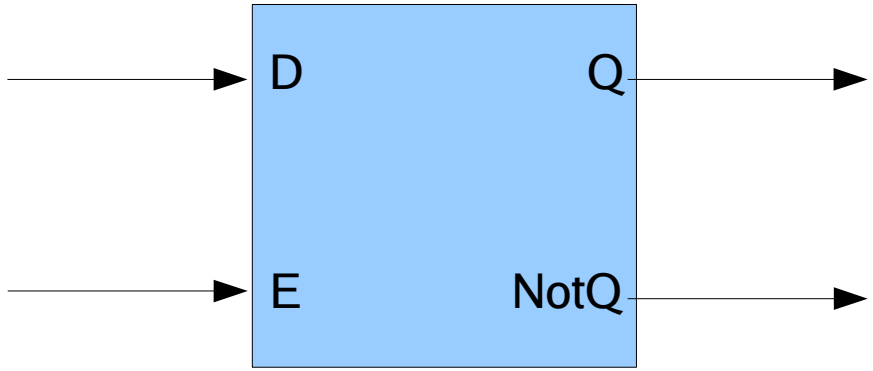
- Computers are black boxes made of boxes that are made of boxes that are made of boxes...

Gated Flip-Flop

- So, in the spirit of ignoring the inner workings once they are understood...

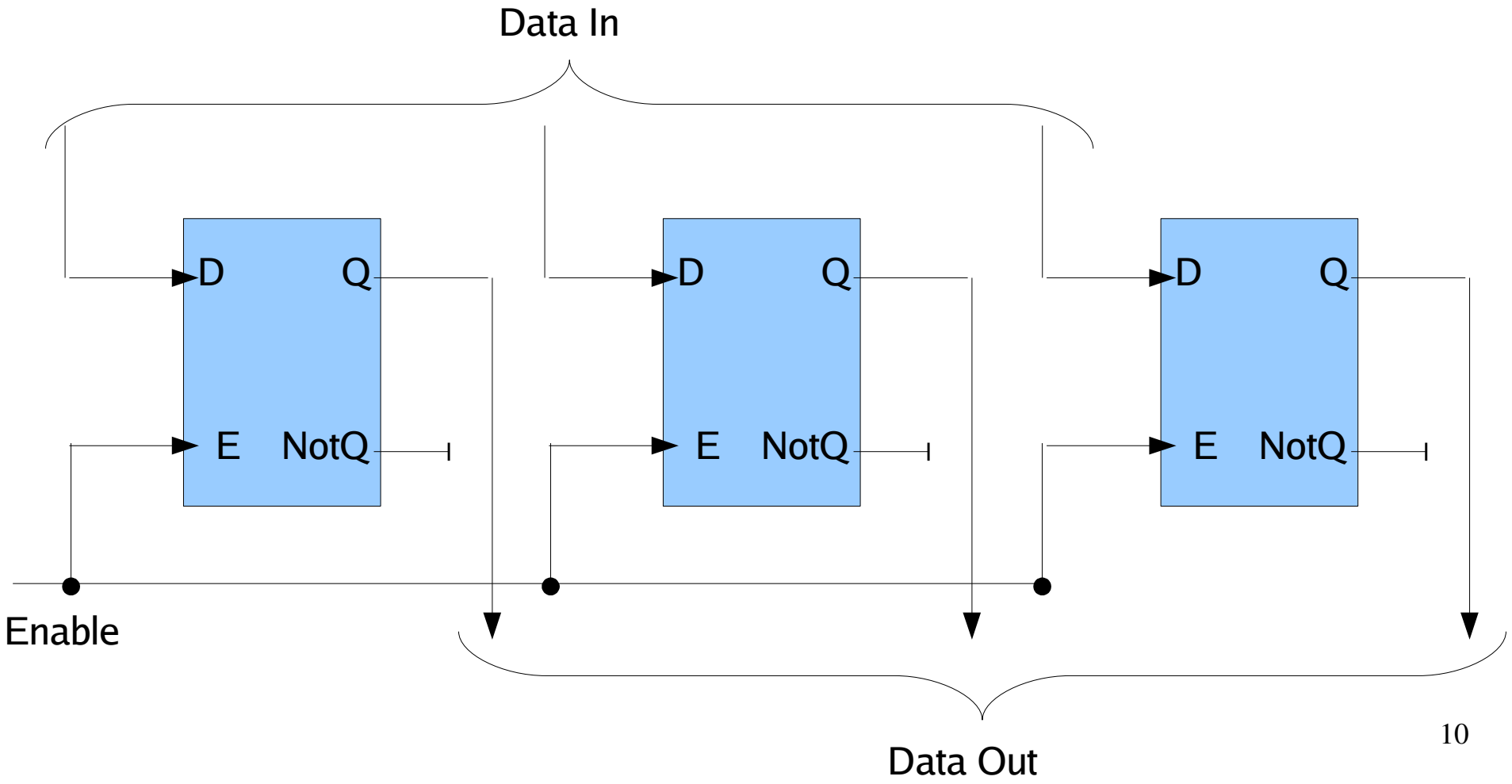
Or even without understanding them...

- We incorporate the GATING NETWORK into the box and draw the GATED FLIP-FLOP like so:



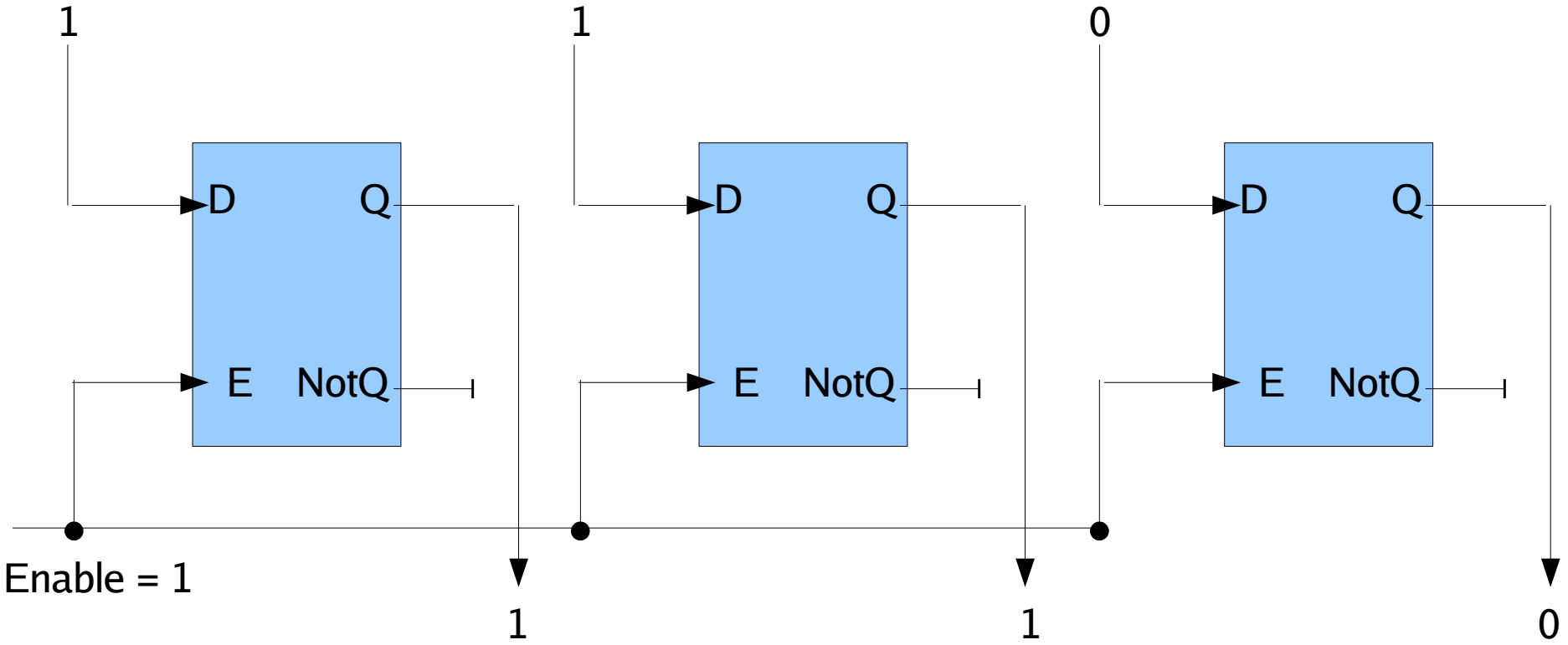
Parallel Register

- Then here is a parallel register that stores 3 bits



Parallel Register

- On E=1, 3 bits are simultaneously loaded into the register



- Now, what controls the ENABLE input?

TIME

We are a basic fact of computer life!

Computers won't be computers without us, bwa, ha, ha, ha, ha!

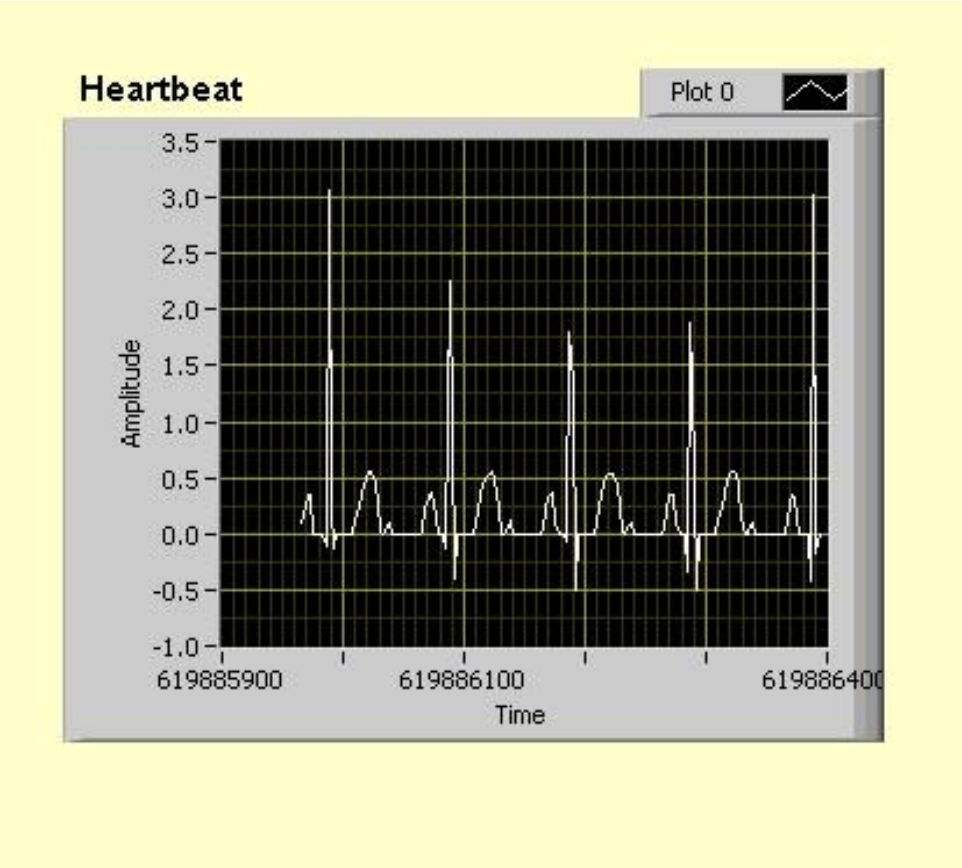
TIME

- As soon as you begin storing data, questions of TIMING arise:
  - How long do you store data?
  - When do you move data?
  - How do you synchronize signals?
- These issues are so critical that logic with memory is called SEQUENTIAL LOGIC
- Thus, to keep sequential logic in step: ALL COMPUTERS HAVE CLOCKS

We are the necessary evils!

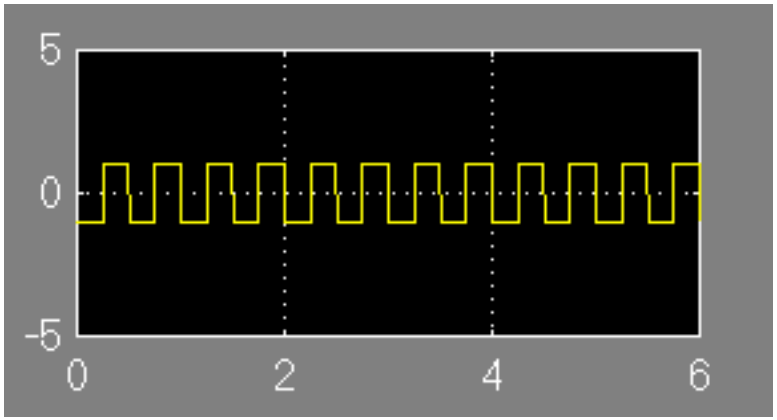
TIME

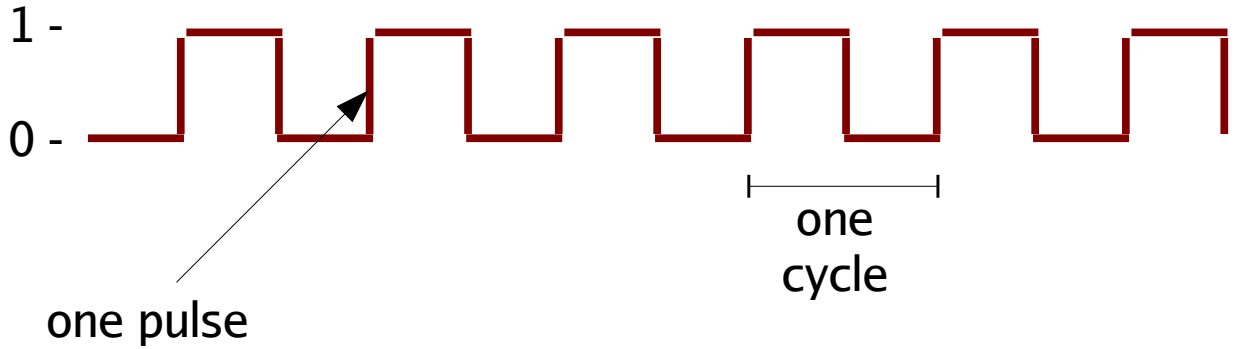
- The clock's pulse is the computer's heartbeat
- Only, instead of a warm, ragged human hearbeat, like this:



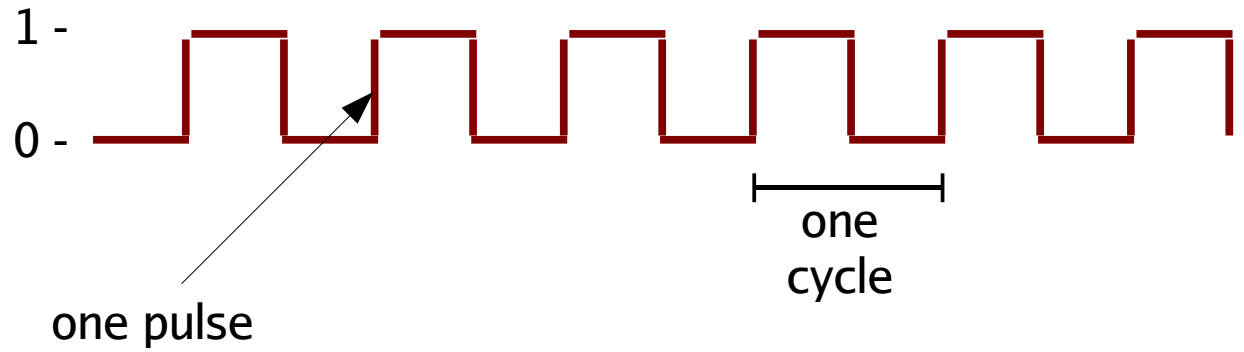
TIME

- The computer's pulse is square and cold:





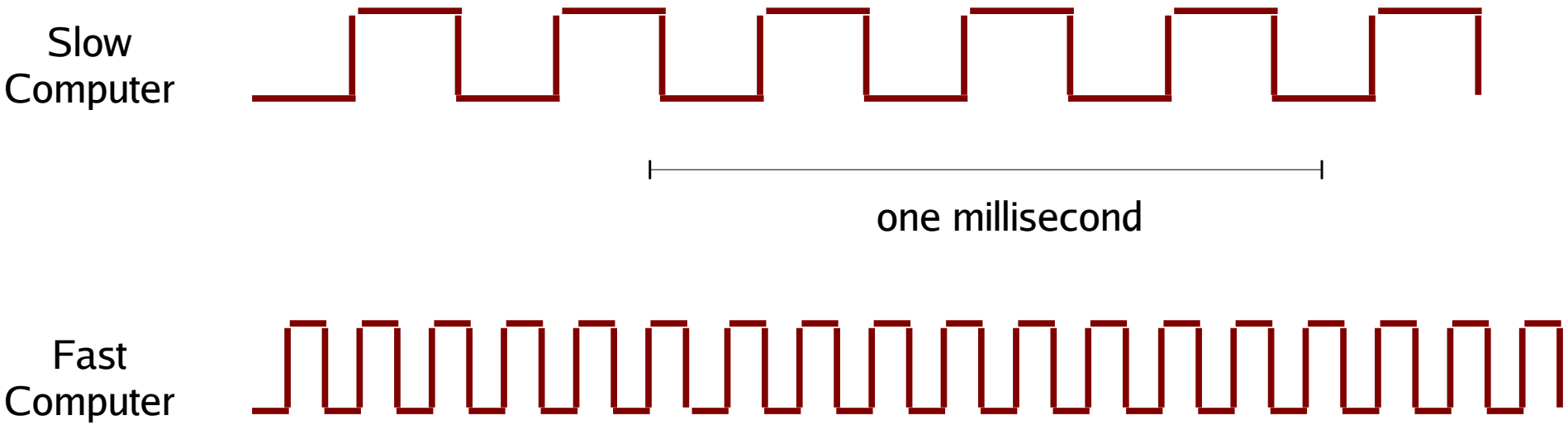
TIME



- One clock pulse is the burst of electrical current when clock output = 1
- One cycle is the interval from the beginning of a pulse to the beginning of the next.

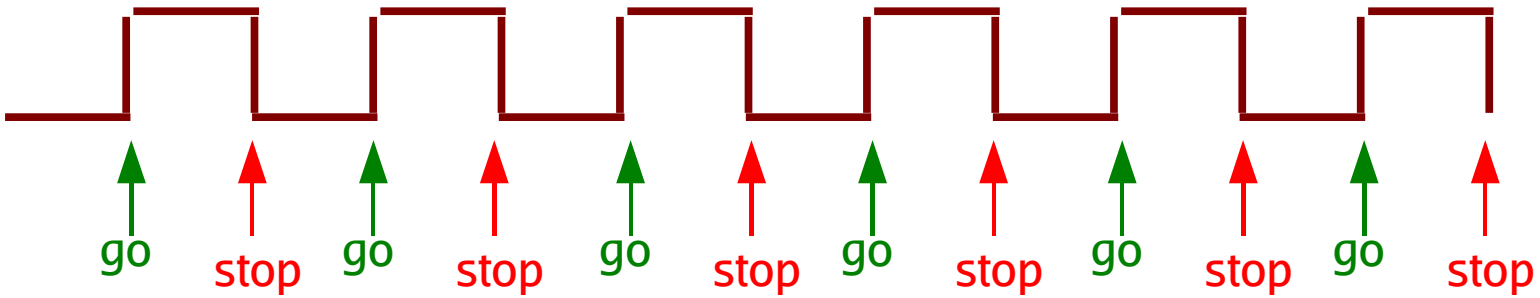
TIME

- Depending on the computer, the clock frequency may be hundreds of thousands to billions of cycles per second



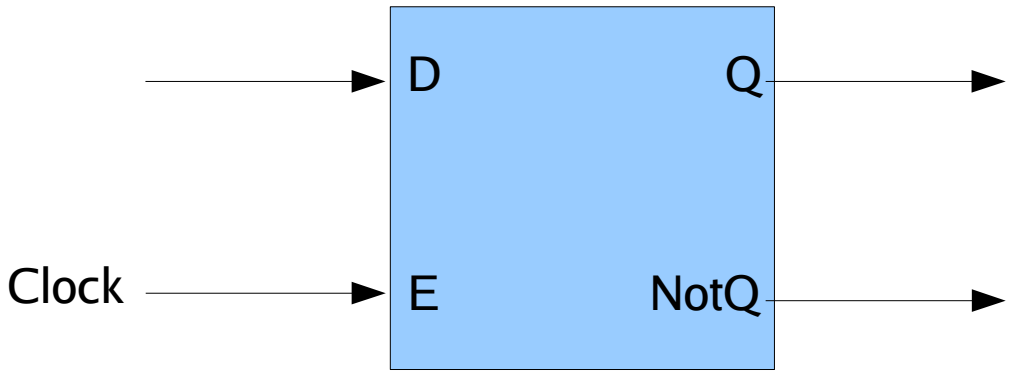
TIME

- The idea of using a clock is that the computer's logical state should change ONLY on the clock pulse.
- Ideally, when the clock hits 1, all signals move, then stop on clock=0



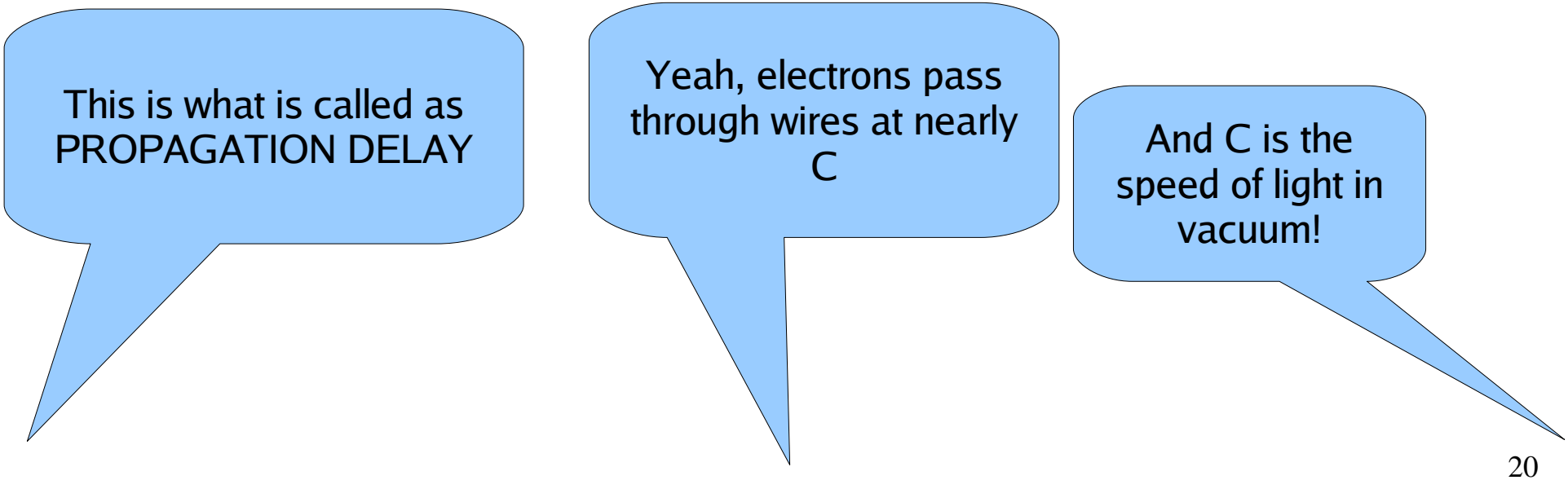
TIME

- A typical example is to attach the clock to the ENABLE input of the gated flip-flop
- in which case the flip-flop becomes known as the D Flip-Flop
- Then a new bit of data is loaded at every clock pulse



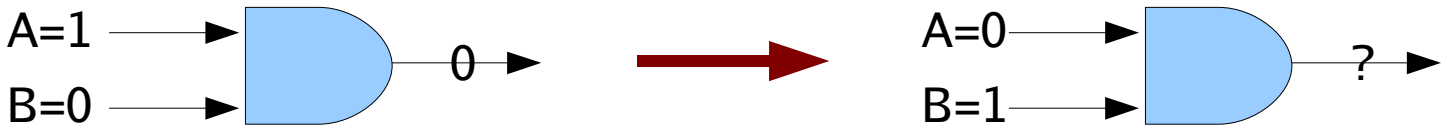
TIME

- Unfortunately, things are rarely ideal
- It takes a non-zero time for a signal to pass along a wire
- so things are never perfectly synchronized



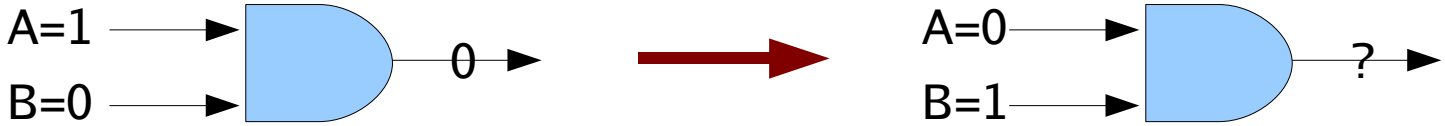
TIME

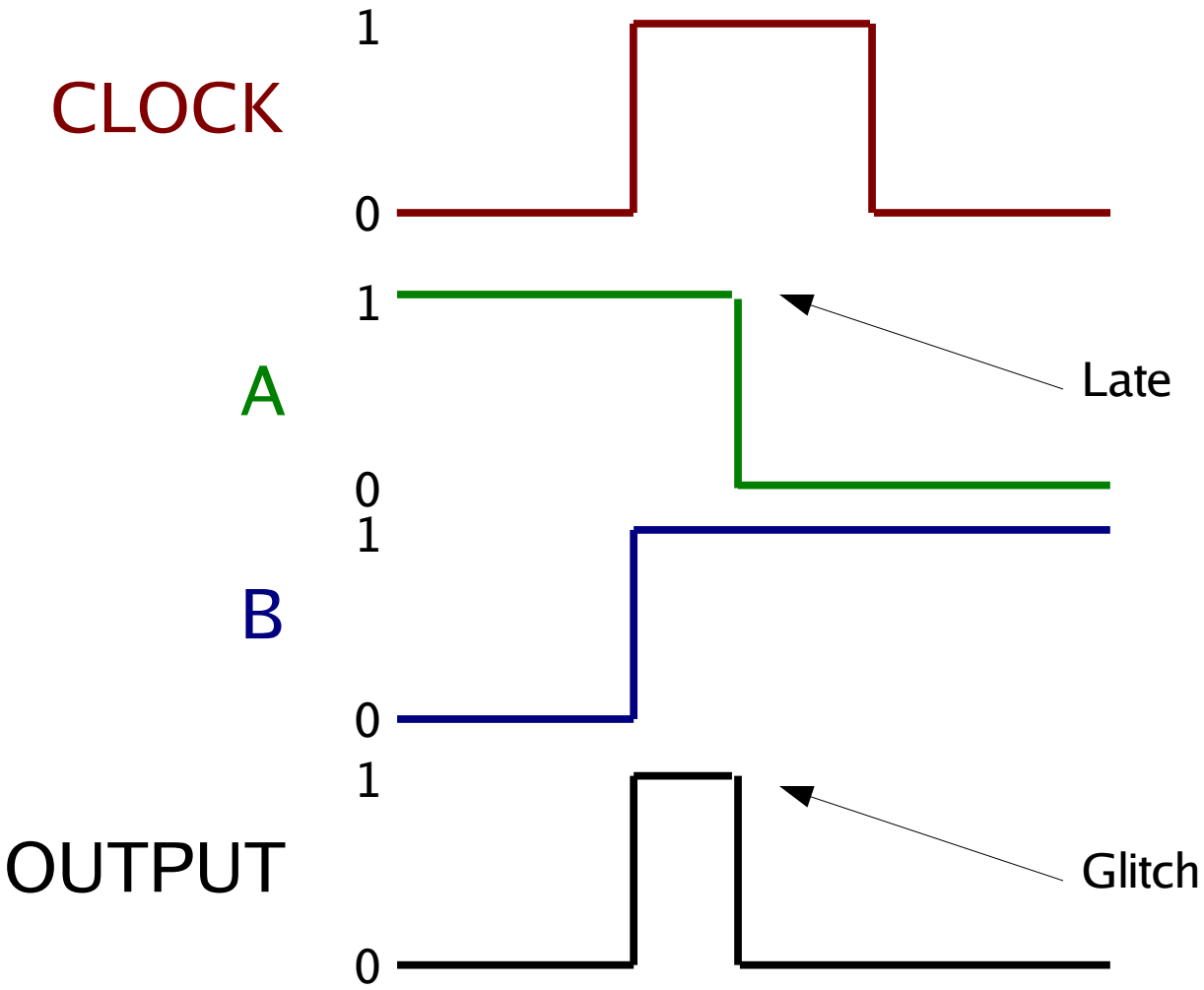
- For example, suppose at an AND-gate, one input is changing from 1 to 0, and the other from 0 to 1.



- If A and B did not change at the same time,
- Say A changes after B, the output will have an unwanted pulse.

TIME





- This pulse is a GLITCH
- And brief as it is, it can cause a flip-flop to flop!

Next Meeting

- Memory