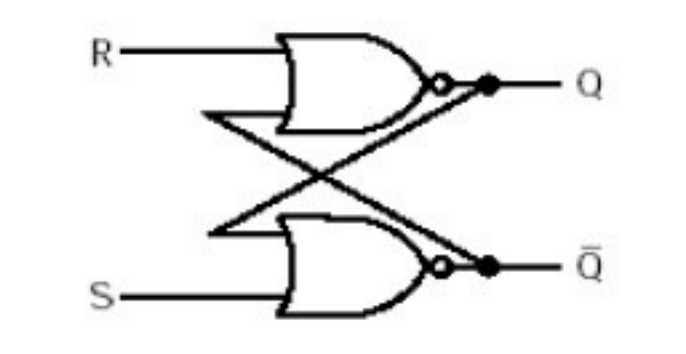
COMP1003 Computer Organization

Lecture 6 From Gates to Circuits II: Sequential Circuits

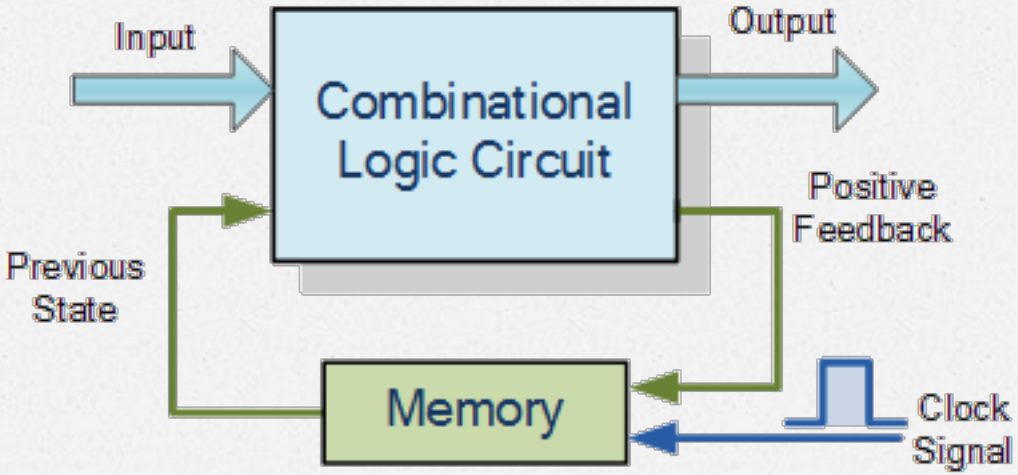


United International College

Sequential Circuit

- Combinational circuits' output depends solely on its current inputs (memoryless)
- We need some circuits that can "remember" its previous inputs
- Sequential circuits' output depends not only on its current inputs, but also its previous inputs (current state)
- In some sense, we may say sequential circuits have memory

Sequential Circuits = Combinational Circuits + Memory



What exactly is memory?

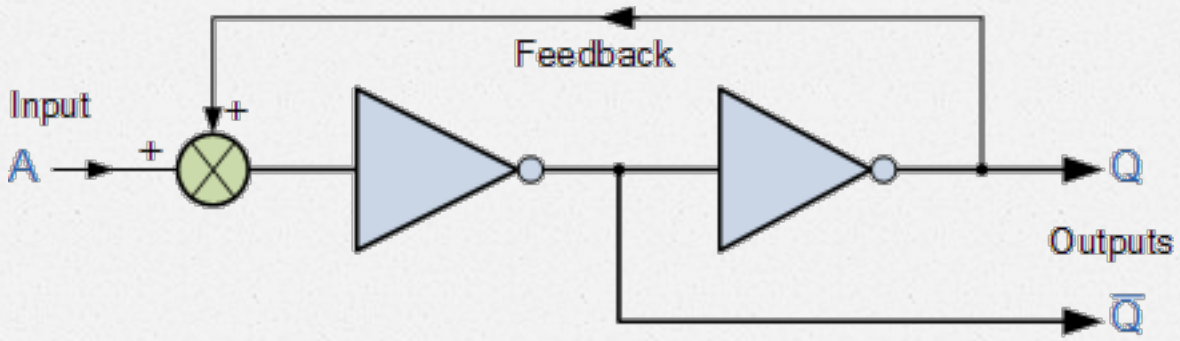
- Write: You should be able to change (write) the value that's saved.
- Hold the value: It should be able to hold a value.
- Read: You should be able to read the value that was saved.
- We will begin with the simplest case: 1 bit memory

One bit memory

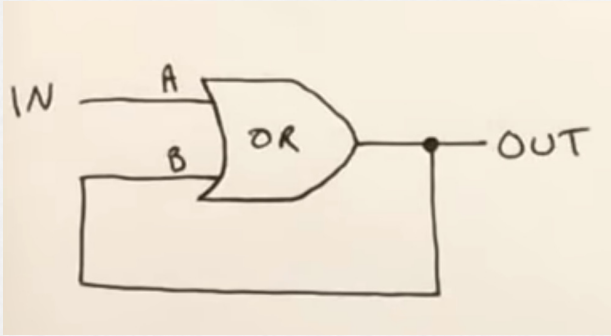
- It should be able to hold a single bit, 0 or 1.
- You should be able to read the bit that was saved.
- You should be able to change the bit. There are only two choices:
  - – Set the bit to 1
  - – Reset, or clear, the bit to 0.

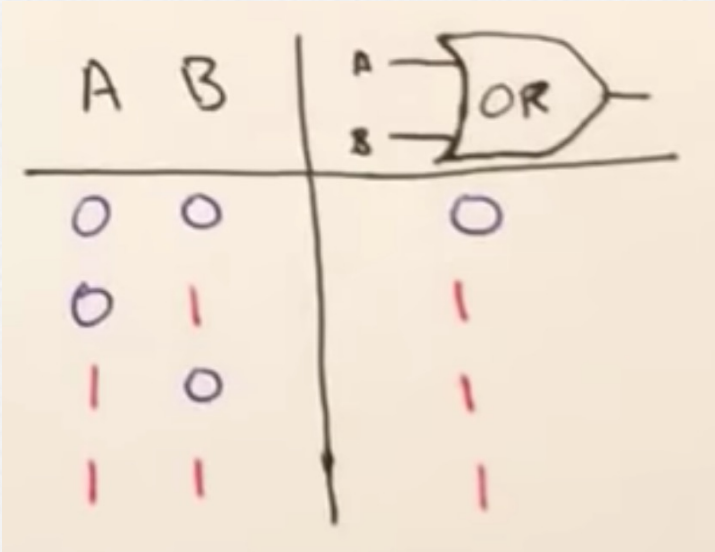
Feedback

- The output of a circuit is fed back as an input to the same circuit
- A simple example (can this be a one-bit memory?)
  - – If Q is 0, it will always be 0
  - – If Q is 1, it will always be 1

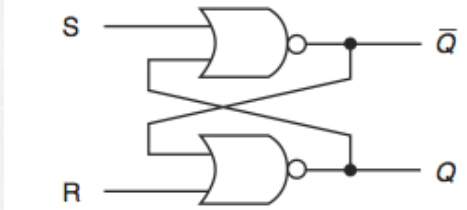


Set (to 1) and Remeber





SR NOR Latch (or Flip-flop)

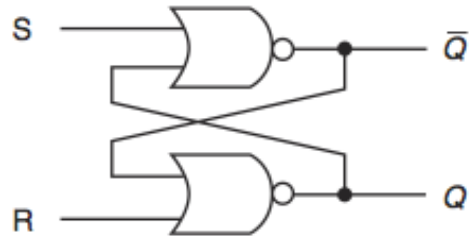


S	R	Q (t+1)
0	0	Q(t) (no change)
0	1	0 (reset to 0)
1	0	1 (set to 1)
1	1	undefined

- SR: Set/Reset
- The building blocks of sequential circuits (like gates for combinational circuits)
- Characteristic table (corresponds to truth table)

State Change of SR Latch

- When SR=00, Q(t+1) = Q(t) (store the value)
- When SR=01, Q(t+1) = 0 (reset to 0)
- When SR=10, Q(t+1) = 1 (set to 1)
- When SR=11, unstable (undefined)



S	R	Q (t+1)
0	0	Q(t) (no change)
0	1	0 (reset to 0)
1	0	1 (set to 1)
1	1	undefined

Truth Table

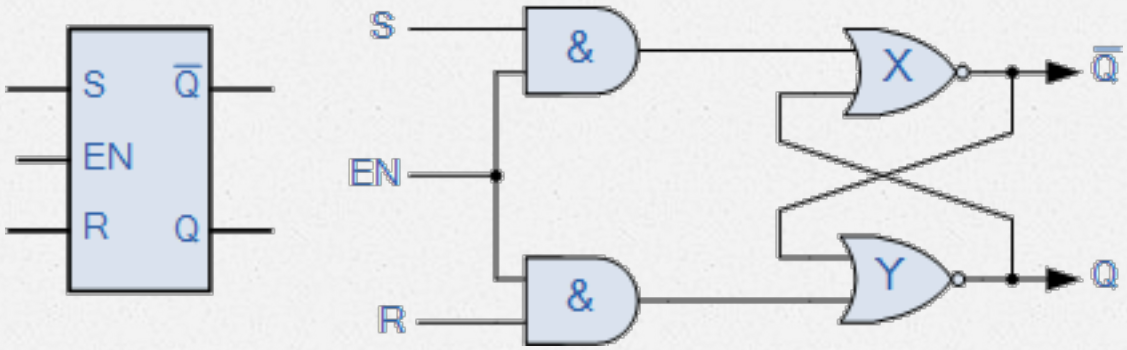
S	R	Present State Q(t)	Next State Q(t+1)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	undefined
1	1	1	undefined

SR Latch is Sequential

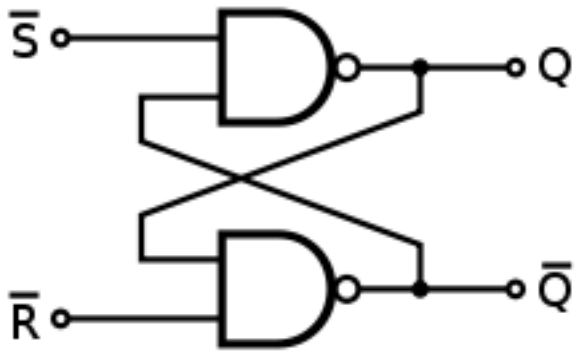
- For inputs SR = 00, the next value of Q could be either 0 or 1, depending on the current value of Q.
- So the same inputs can yield different outputs, depending on whether the latch was previously set or reset.
- This is very different from the combinational circuits that we've seen so far, where the same inputs always yield the same outputs.

Gated SR Latch

- EN: an additional enable input



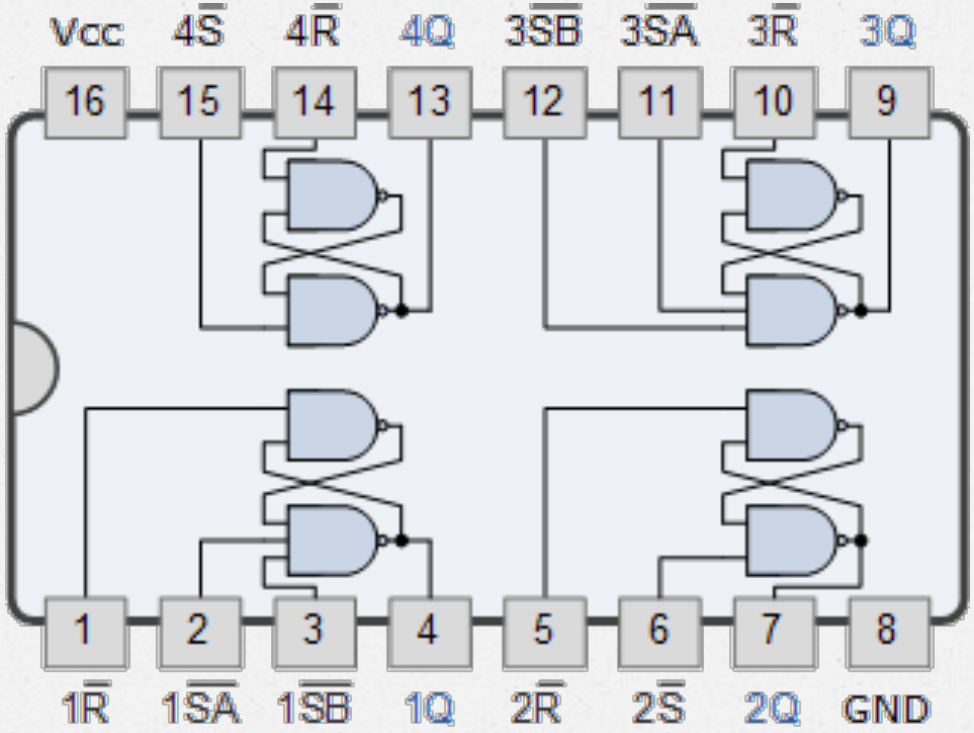
SR-NAND Latch



$\overline{SR}$ latch operation

$\overline{S}$	$\overline{R}$	Action
0	0	Not allowed
0	1	Q = 1
1	0	Q = 0
1	1	No change

SR Latch IC: The 74LS279

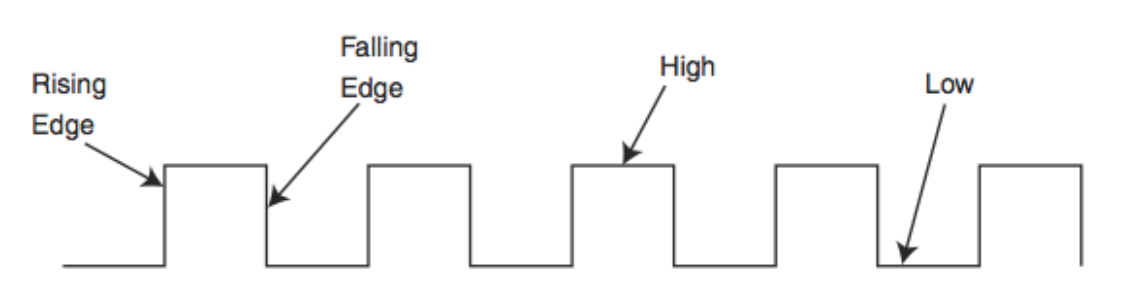


Glitch

- There is a finite time delay between a change in the inputs of a gate and any change in the output. This time is called gate delay.
- In order to avoid glitches, we want to design storage elements that only accept input when ordered to so
- We use a clock to be the control input that gives orders to the circuit about when to change states

Clock

- A clock is a special circuit that produces electrical pulses
- Clock speed is generally measured in megahertz (Mhz), or millions of pulses per second
- A clock is used by a sequential circuit to decide when to update the state of the circuit
- inputs to the circuit can only affect the storage element at give

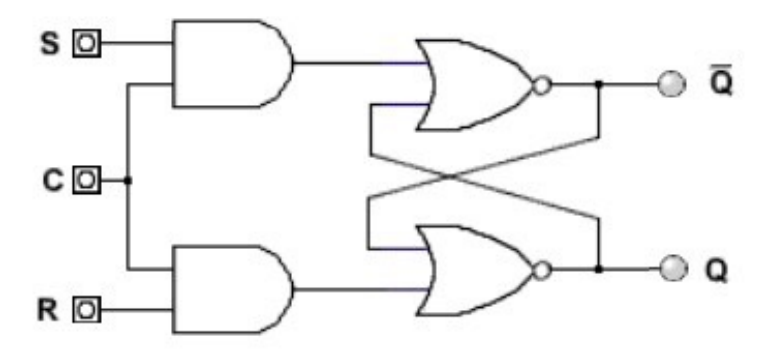


Edge-Triggered vs Level-Triggered

- Edge-triggered: allowed to change their states on either the rising or falling edge of the clock signal
- Level-triggered: allowed to change state whenever the clock signal is either high or low
- Technically, a latch is level triggered, whereas a flip-flop is edge triggered

Clocked S-R Latch

- When C is 1, the circuit acts just like the NOR gate S-R latch.
- When C is 0, the Set and Reset inputs are disabled
- The latch can change only when C is true

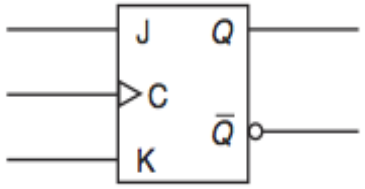


Illegal State

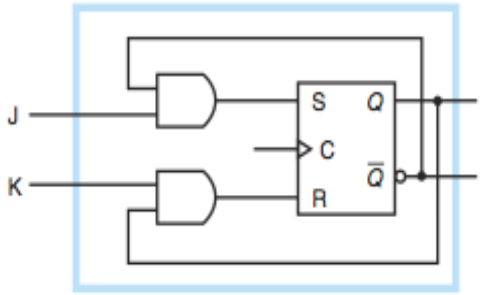
- What happens if both S and R are set to 1 at the same time?
- This forces both Q and NOT Q to 1, illegal!
- We can simply add some conditioning logic to ensure that the illegal state never arises.
- This results in a J-K flip-flop

The JK Flip Flop

- The JK flip-flop is basically an SR flip flop with feedback which enables only one of its two input terminals, either SET or RESET to be active at any one time

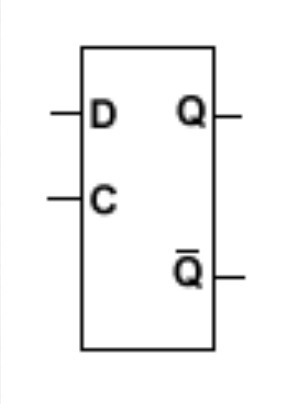


J	K	Q(t +1)
0	0	Q(t) (no change)
0	1	0 (reset to 0)
1	0	1 (set to 1)
1	1	$\overline{Q}(t)$



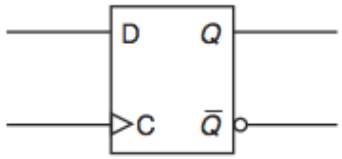
D Flip-flop: A Real One-bit Memory

- The need for explicitly setting and resetting the S-R latch is added complexity
- What we would really like is a circuit that has a data input D and a data output Q.
- When the clock signal is high, whatever appears on D should be stored in Q
- It is called the D Flip-flop

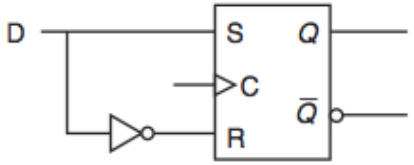


D Flip-flop

- It stores one bit of information
- The output changes only when the value of D changes
- an output value of 1 means the circuit is currently "storing" a value of 1
- A D flip-flop is a true representation of physical computer memory

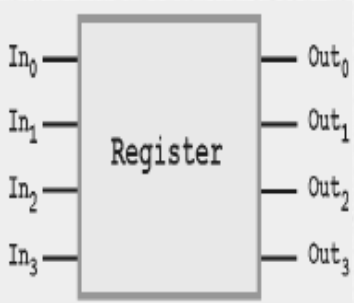


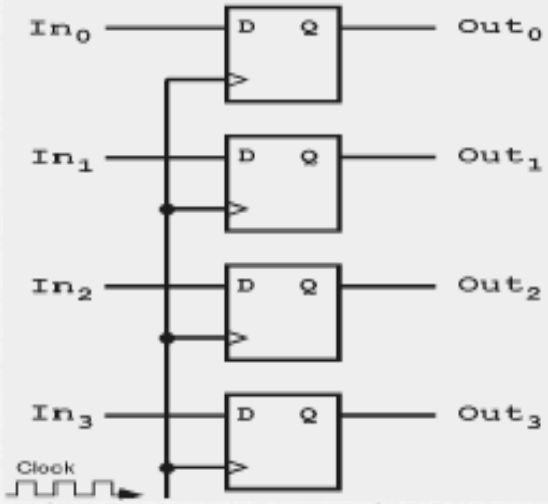
D	Q (t +1)
0	0
1	1



A 4-bit Register

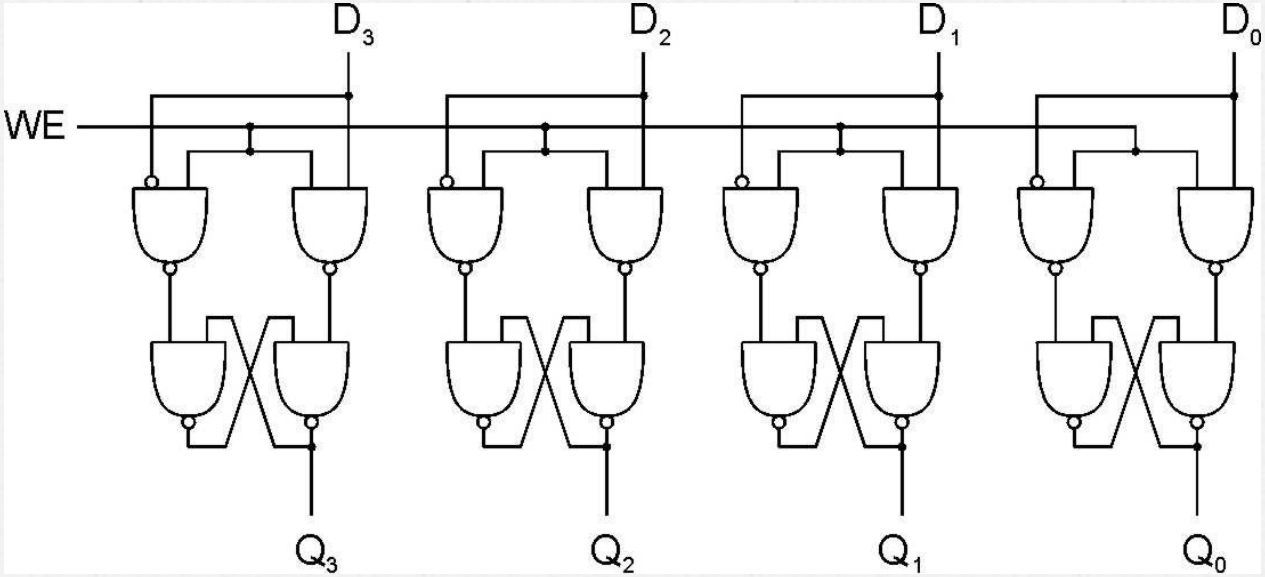
In reality, physical components have additional lines for power and for ground, as well as a clear line (which gives the ability to reset the entire register to all zeros)





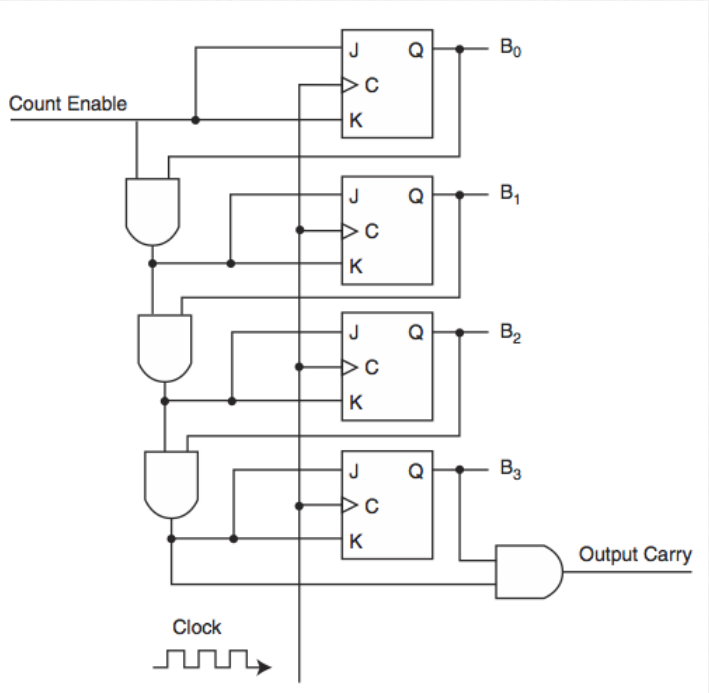
Register

- 4 input lines, 4 output lines and a WE (Write Enable) line



A Binary Counter

- How does this circuit output the binary numbers from 0000 to 1111?



J	K	Q(t +1)
0	0	Q(t) (no change)
0	1	0 (reset to 0)
1	0	1 (set to 1)
1	1	$\bar{Q}(t)$

B_0 (t+1) = JK (1, 1) = NOT B_0(t)

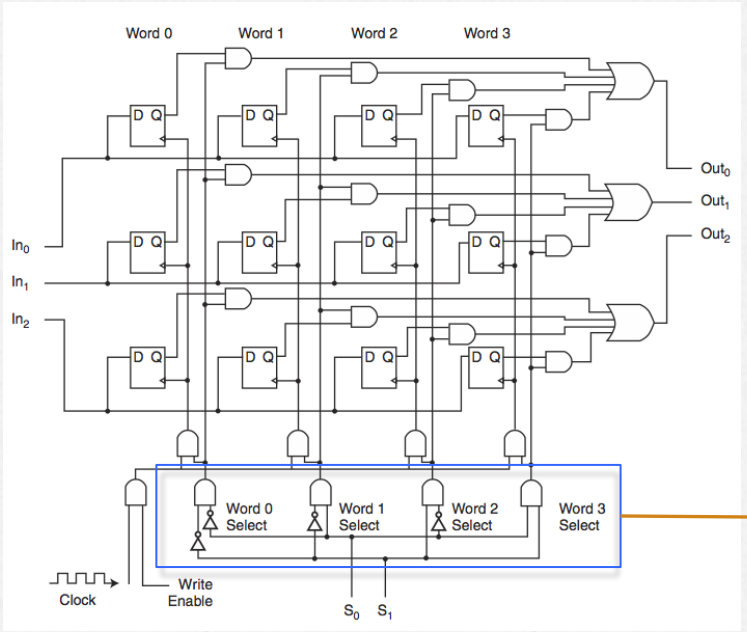
B_1 (t+1) = JK (B_0(t), B_0(t))

B_2 (t+1)= JK (B_0(t)B_1(t), B_0(t)B_1(t))

B_3 (t+1)= JK (B_0(t)B_1(t)B_2(t), B_0(t)B_1(t)B_2(t))

A 4x3 Memory

- Each column in the circuit represents one 3-bit word
- The inputs lines: In0, In1, and In2
- Address lines: S0 and S1



decoder

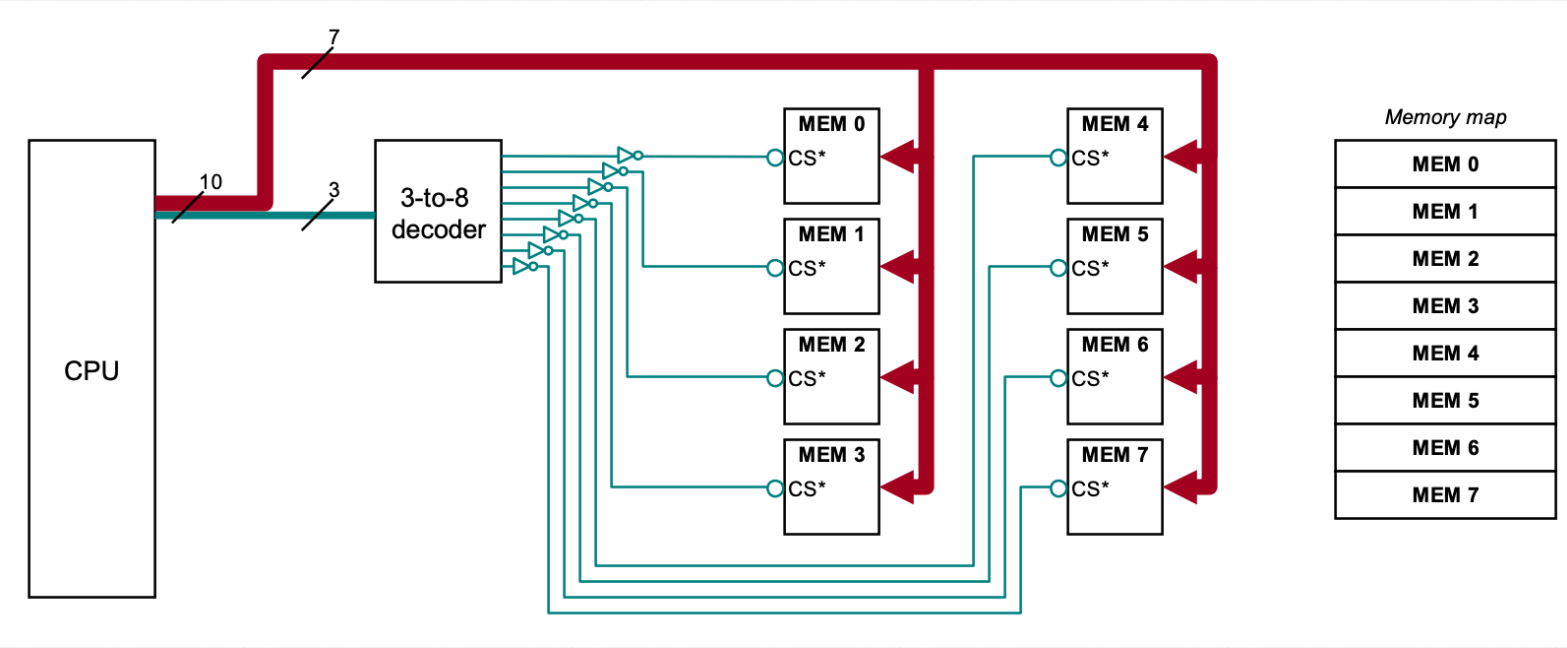
Write a word to memory

1. An address is asserted on S0 and S1.
2. WE (write enable) is set to high.
3. The decoder using S0 and S1 enables only one AND gate, selecting a given word in memory.
4. The line selected in Step 3 combined with the clock and WE select only one word.
5. The write gate enabled in Step 4 drives the clock for the selected word.
6. When the clock pulses, the word on the input lines is loaded into the D flip-flops.

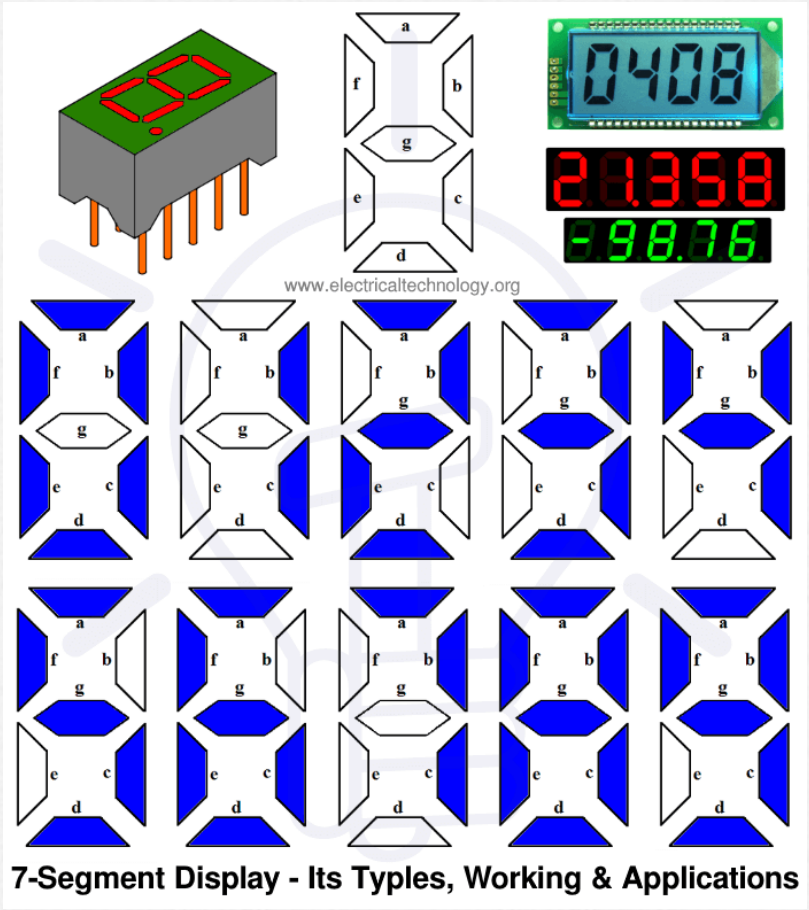
Memory Addressing

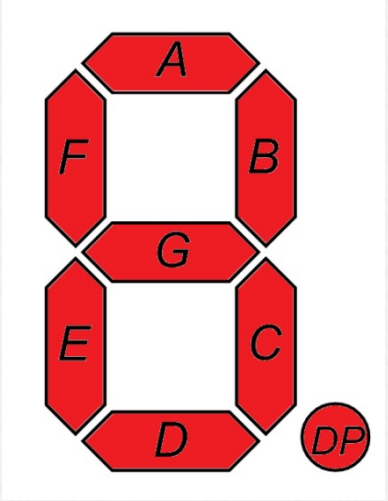
- Let's assume a very simple microprocessor with 10 address lines (1KB memory)
- Let's assume we wish to implement all its memory space and we use 128 x 8 memory chips
- SOLUTION
  - We will need 8 memory chips (8x128=1024)
  - Chip selection: We will need 3 address lines to select each one of the 8 chips
  - Memory location selection inside each chip: Each chip will need 7 address lines to address its internal memory cells
  - Address: 000 0000000

128 x 8 memory chips



7-Segment Display





Truth Table for Numbers from 0 to 9

Numerals	A	B	C	D	E	F	G
0	1	1	1	1	1	1	0
1	0	1	1	0	0	0	0
2	1	1	0	1	1	0	1
3	1	1	1	1	0	0	1
4	0	1	1	0	0	1	1
5	1	0	1	1	0	1	1
6	1	0	1	1	1	1	1
7	1	1	1	0	0	0	0
8	1	1	1	1	1	1	1
9	1	1	1	1	0	1	1

Question

- How to display numbers from 0 to 9, one after another?

Summary

- Sequential circuits can remember their previous inputs
- Sequential circuits require clocks to control their changes of states
- The basic sequential circuit unit is the flip-flop: SR, JK and D flip-flop
- Examples: registers, binary counter and memory

What's Next

- Microarchitecture: how to use circuits to implement the ISA