CSSE2010

CSSE2010/CSSE7201
Lecture 5

Flip-flops

School of Information Technology and Electrical Engineering
The University of Queensland

Today

- Admin
- Recap from week 2
- Circuits that remember values
  - Flip-flops
  - Latches

Admin

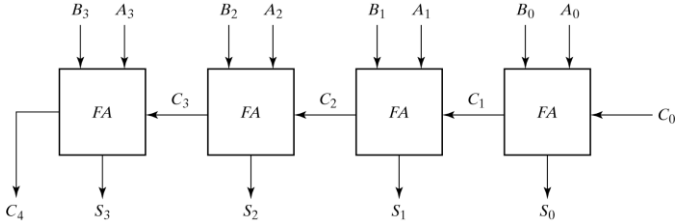
- Weekly quizzes:
  - Quiz 1 – closed – xxx attempts, xx/17 average, xx/17 median
  - Quiz 2 – open now and due Friday 13-Aug 4pm AEST
- Labs:
  - **IN students** – use Logisim to simulate logic circuits and also construct on breadboard if you have borrowed a kit
  - **EX students** – use Logisim to simulate the logic circuits. Start acquiring your Arduino based hardware items required from week 7. Details on Blackboard

Prac Equipment - IN students

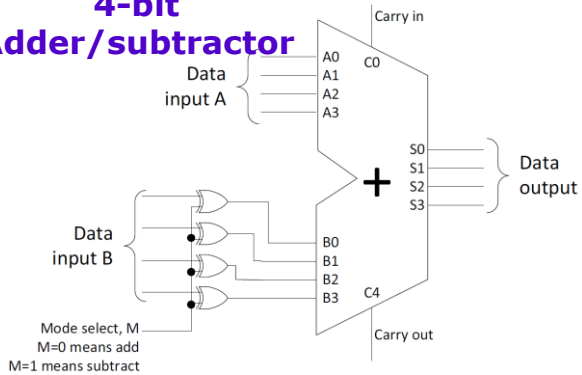
- Treat it carefully
- Report damage / missing parts
- Don’t use parts you’re not familiar with and wait until we get into the relevant prac activity.
- Pay extra attention to the I/O board USB connector when you are using it
- Switch the power off before you build/change circuits
  - Check your wiring before you apply power

Recap from Last Week

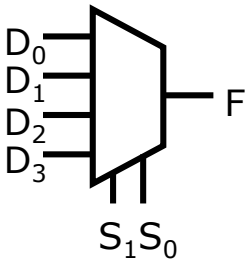
4-bit Ripple carry adder



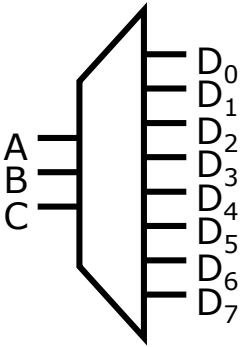
4-bit Adder/subtractor



4:1 MUX



3:8 Decoder



Combinational Logic Circuits – Adder, Adder/subtractor, Multiplexer, Decoder

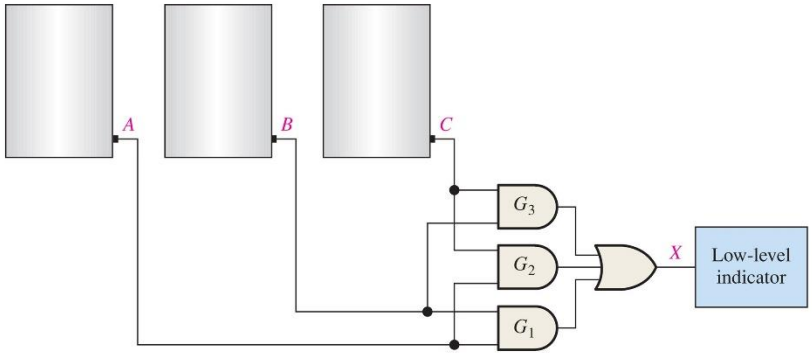
⇧

Logic Gates – NOT, AND, OR, NAND, NOR, XOR, XNOR and Boolean algebra

⇧

Binary representations – unsigned, sign-mag, 1's comp, 2's comp, excess-2^{N-1}

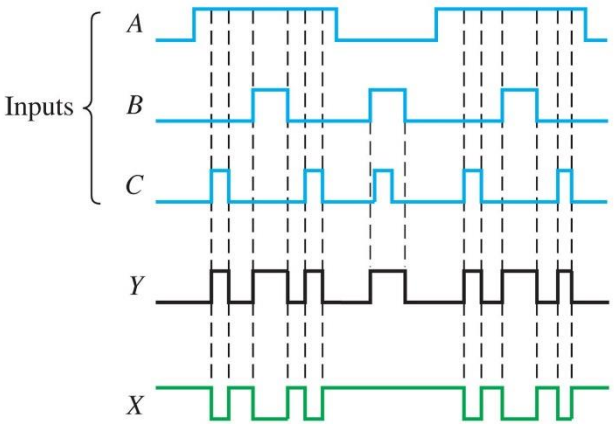
Recap from Last Week (cont...)

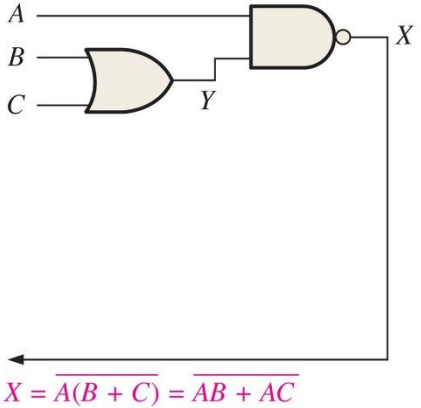


Practical application of the 3-input majority function **X=AB+BC+AC**

When a majority of tanks is low the output will be driven high indicating low-level of water

Timing diagrams to represent logic circuits



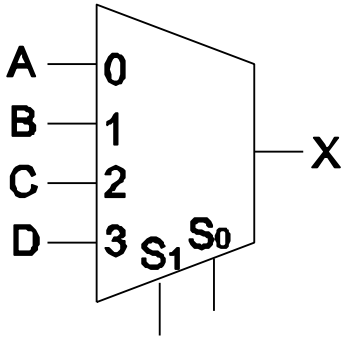


Images taken from Instructor Resources of Digital Fundamentals, Thomas L. Flyod, Pearson Educations Inc., 2015

Clicker Question

Consider the multiplexer shown. What must the inputs A,B,C,D be so that the multiplexer output is

$$X = S_1.S_0 + \overline{S_1}.G$$



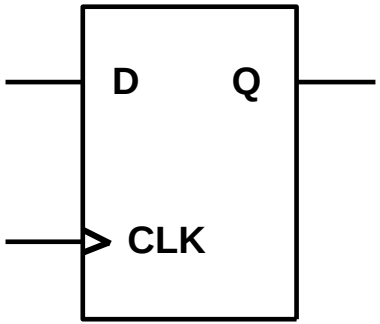
1. 17% A=0, B=0, C=0, D=1
2. 17% A=G, B=G, C=0, D=1
3. 17% A=G, B=0, C=0, D=1
4. 17% A=0, B=G, C=0, D=1
5. 17% None of the above
6. 17% I don't know

Circuits that remember values

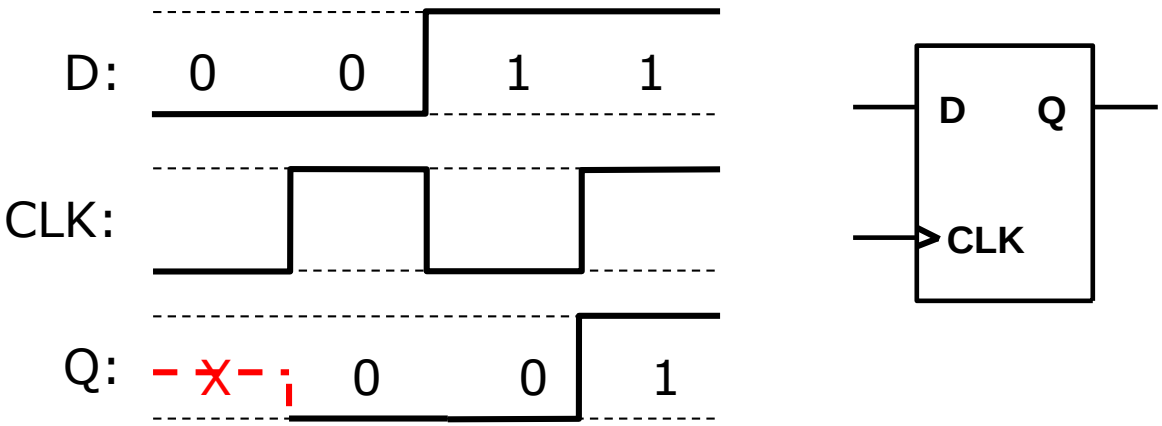
- The output of any logic gate or **combinational circuit** is dependent on the **current value of inputs only**
- If an input changes, the output can also change and the previous value is lost forever
- **Sequential circuits**: the current output depends not only on the current inputs but also on the past outputs.
- Circuits with **memory** can remember values, even if the input changes

Memory element: D Flip Flop

- **D** is input
- **Q** is output
- **CLK** (clock) is control input
- How does it work?
  - Q copies the value of D (and remembers it) whenever CLK goes from 0 to 1 (*rising edge*)



D Flip Flop



- Remember: Q changes ONLY when clock (CLK) goes from 0 to 1

Characteristic Tables

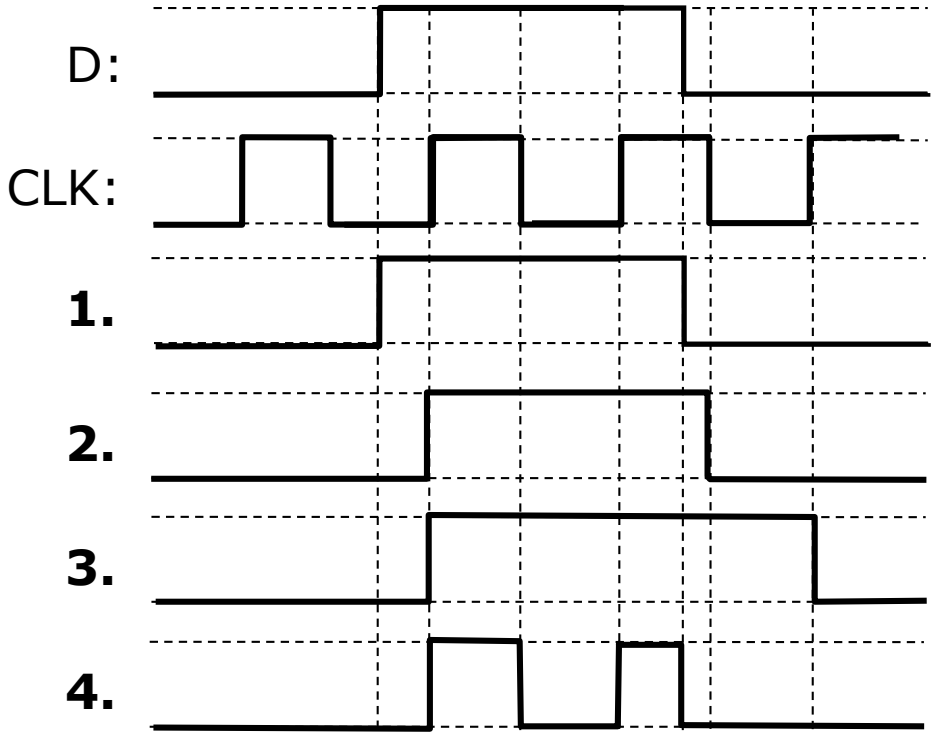
- **Characteristic table** defines operation of flip-flop in tabular form
  - D flip-flop

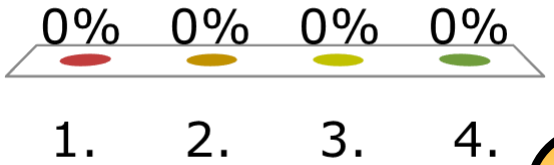
D	Q(t+1)	
0	0	Reset
1	1	Set

Input

What the output will be on the next clock edge (i.e., CLK goes from 0 to 1).

Using the D flip-flop presented previously, what is the output waveform for Q?





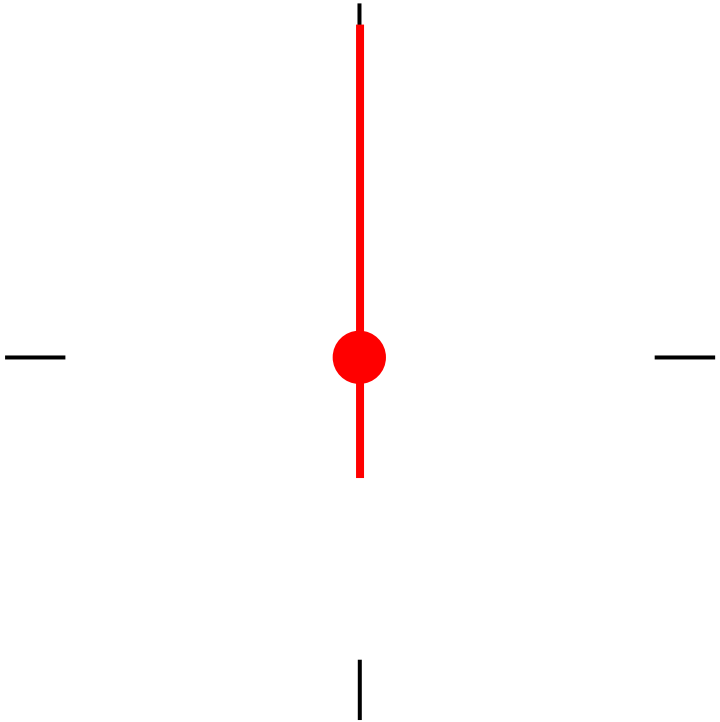
10

D Flip-flops

- Summary: D flip-flop remembers either a "1" or "0" – i.e. a single bit. That is the flip-flop remembers the value till the next clock edge, upon which the D input is transferred to output Q.
- So to remember *n* bits, you will need *n* D flip-flops
- Definition: A *n*-bit **register** can be made using *n* D flip-flops
- There are other types of flip-flops, e.g.
  - JK flip-flops
  - T flip-flops
- Flip-flops can be made out of logic gates

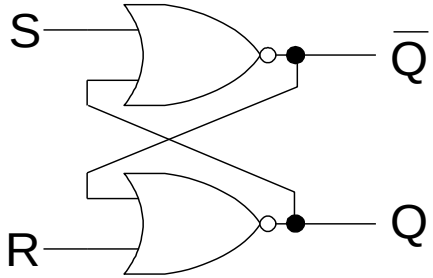
Short Break

- Stand up and stretch



SR Latch – to be completed in class

Try to complete this at home before the lecture. You can assume an initial value for Q output or treat it generic as Q and work out the rest for different combinations of inputs S and R



S	R	Q	$\overline{Q}$

Remember, this is what a NOR gate does

A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0

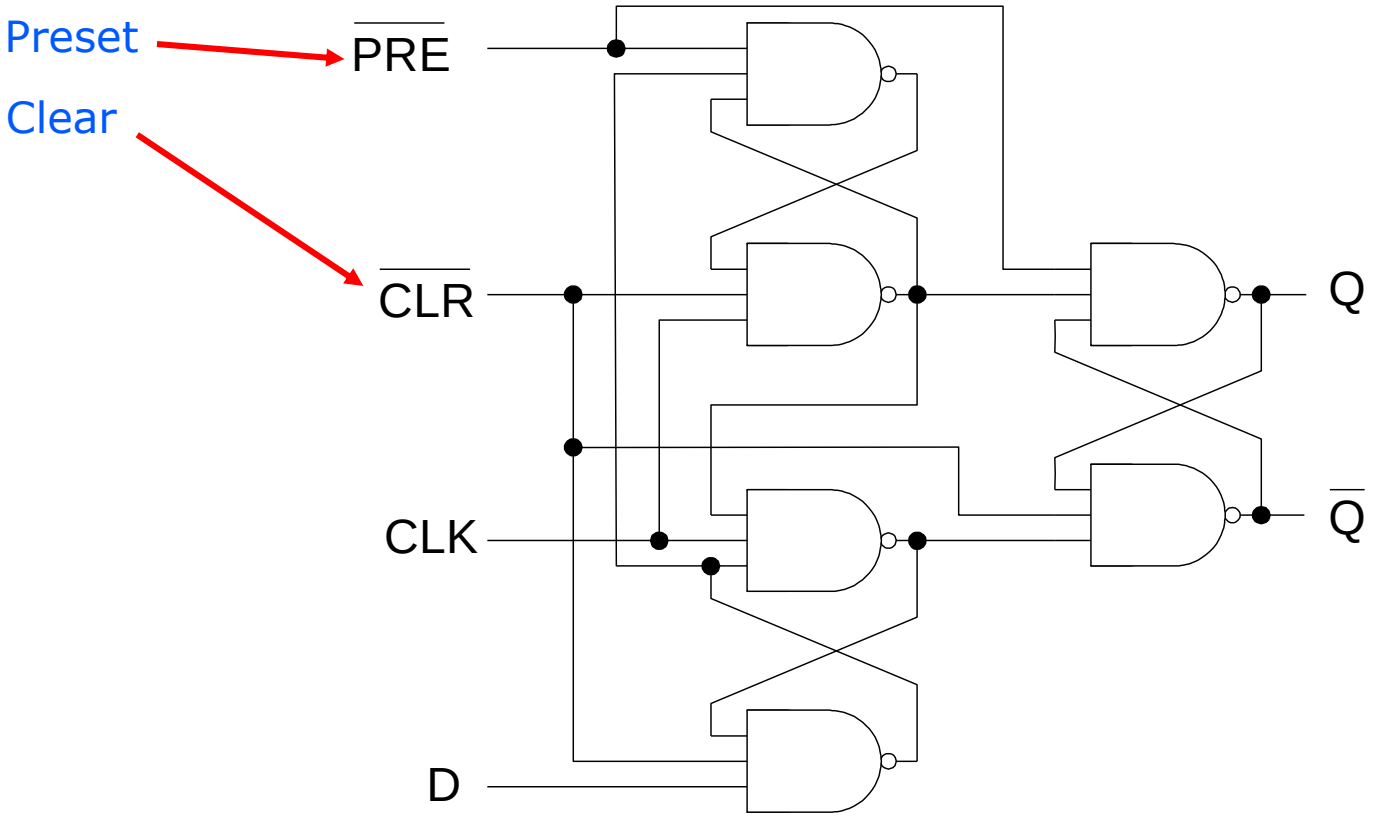
Latches from NAND gates

Homework: Analyse the S-R latch circuit in the previous slide when NOR gates are replaced with NAND gates and complete the truth table.

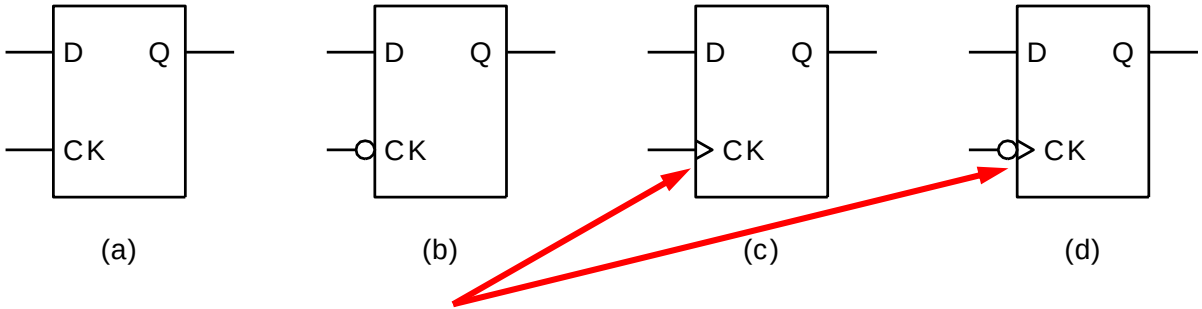
Flip-Flops vs Latches

- **Latches** are **level triggered** devices – i.e. able to latch the output and respond to changes of logic levels on the inputs
- Latch circuits can be modified such that they become sensitive to an **edge** (i.e. **momentary transitions**) of a control input (i.e. a clock signal)
- Such circuits are called "**flip flops**" and a flip flop can **store 1 bit of information while being sensitive to a clock edge** (i.e., flip flop will change its output only at the clock edges, based on the inputs)
- **Latches are level triggered** devices while **flip flops are edge triggered** devices
- A clock signal has two edges
  - Positive edge – 0 to 1 transition
  - Negative edge – 1 to 0 transition
- There are different types of flip flops
  - **D flip flop – we will only discuss this**
  - JK flip flop – not covered in this course
  - T flip flop – not covered in this course
- So, a D flip flop can be positive edge triggered or negative edge triggered
- **Positive edge triggered D flip flop → Input D is copied to output Q at the positive edge of the clock. In between the clock edges, flip flop is not responsive, thus stores the value.**

A Real D Flip-flop



D Latches and Flip-flops – Symbols Used

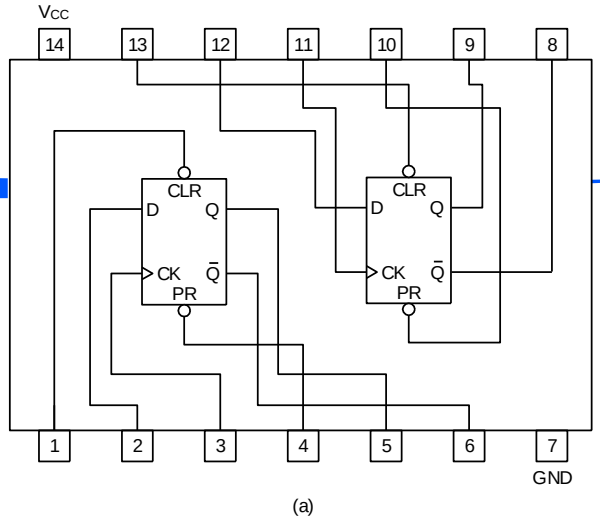


- Triangle indicates edge-triggered (therefore flip-flop)
  - (c) sensitive to rising edge of clock
  - (d) to falling edge
- **State** of a flip-flop is the value stored
- Flip-flops more useful than latches

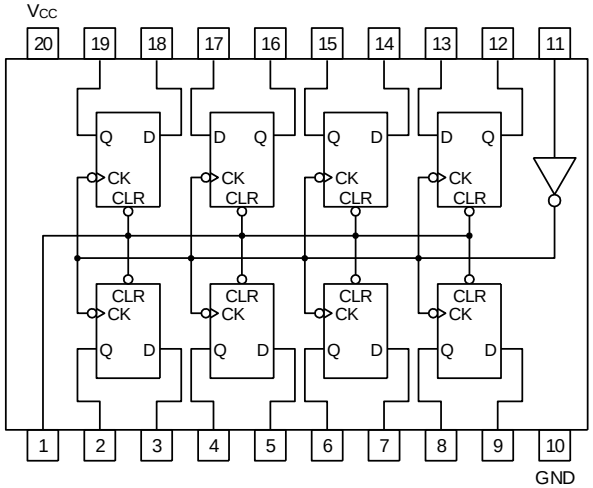
D Flip-flop Chips

- (a) 74HCT74 chip (Dual D flip-flop)
- (b) 74HCT273 chip (Eight D flip-flops)
  - can hold one byte of information (8 bits).

See device symbols PDF on Blackboard



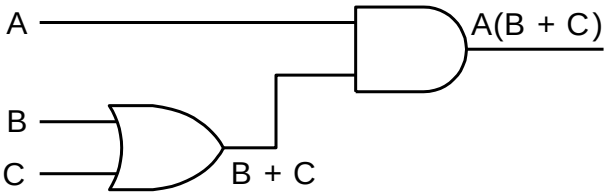
(a)



(b)

Combinational vs. Sequential Circuits

- **Combinational** Circuits (last week)
  - Logic gates only (no flip-flops)
  - Output is uniquely determined by the inputs
    - i.e. you'll always get the same output for a given set of inputs
  - Example from previously:



Adders
Multiplexers
Decoders
Demultiplexers
Encoders

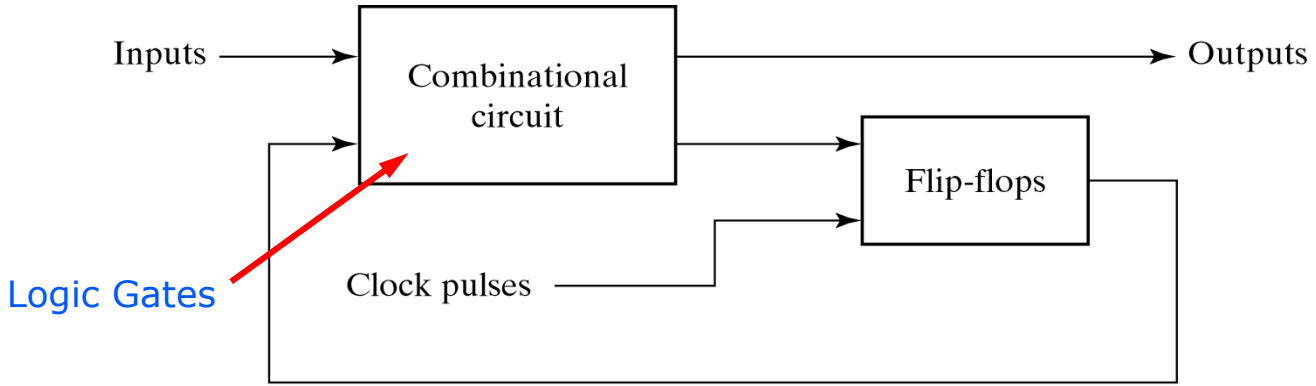
Counters

Registers

- **Sequential** Circuits
  - Include flip-flops
  - Output determined by current inputs and current **state** (values in the flip-flops)
  - Output can only change when clock 'ticks'

Sequential Circuits

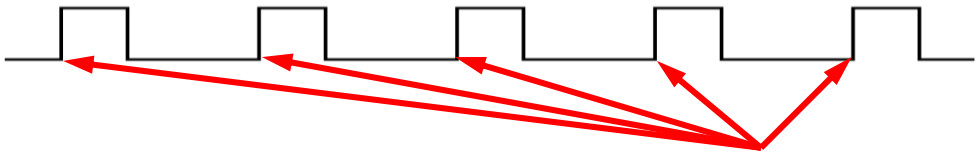
- **State** = value stored in flip-flops
- Output depends on input and state
- Next state depends on inputs and state



[Figure from Mano, "Digital Design", 3rd edition]

Synchronous Sequential Circuit

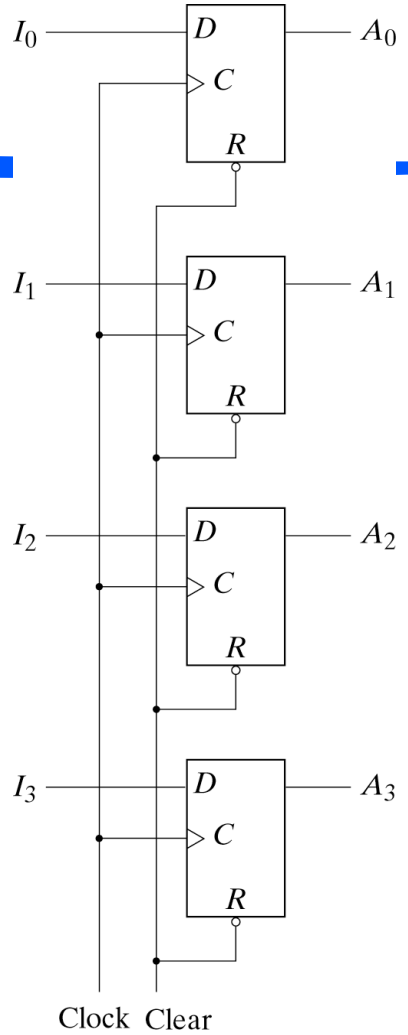
- Storage elements (flip-flops) can only change at discrete instants of time
- Assume
  - We have a clock signal:



  - Output of storage elements change only on the edges of control signal
    - (compare with logic gates whose output changes whenever the input changes)

Registers

- A **register** is a group of flip-flops
  - *n*-bit register consists of *n* flip-flops capable of storing *n* bits
- Example
  - 4-bit register
- A register is a sequential circuit *without* any combinational logic



[Mano, Digital Design, 3rd ed.]

Coming up…

- Quiz 2 – due Friday 4pm this week
- Lab 4
  - Combinational Logic
  - Make sure you attempt the preparation task
  - Use logic ICs or Logisim software to test your circuits
- Lab 5
  - Flip-flops
  - Use Logisim software to test the circuits