Video Lecture 2
From combinational to sequential logic

ENGN 4213/6213

Digital Systems and Microprocessors

What's in this lecture

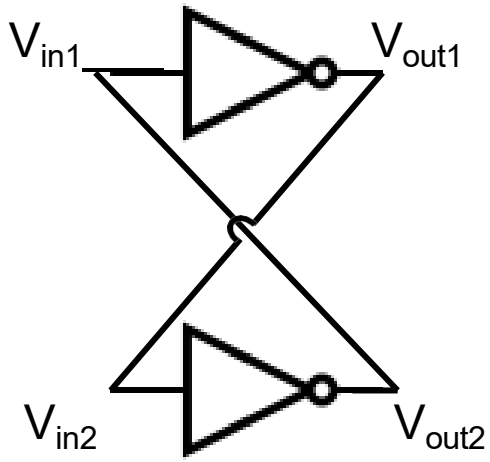
- Evolving from combinational circuits to sequential circuits:
  - feedback, bistability and metastability
  - combinational latches
  - flip-flops
  - synchronous vs asynchronous

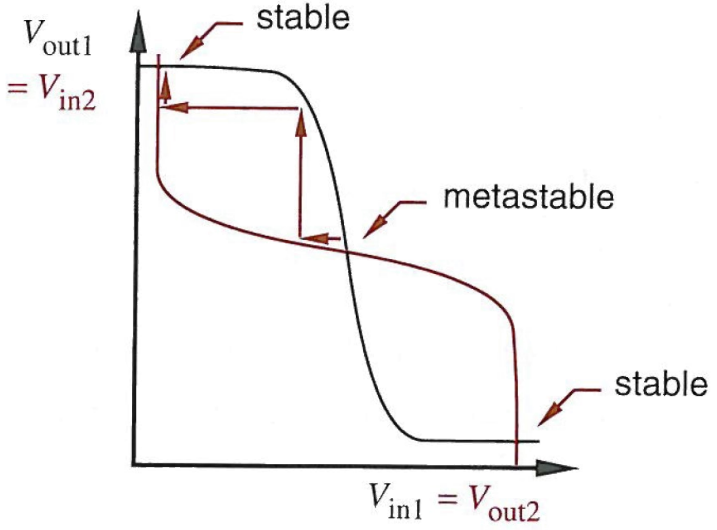
Resources

- Wakerly (5th edition) 10.1 – 10.2.6 for latches and flip-flops
- The old reading brick if you like the way it is written (I will stop mentioning it from now on)

Bistability and metastability

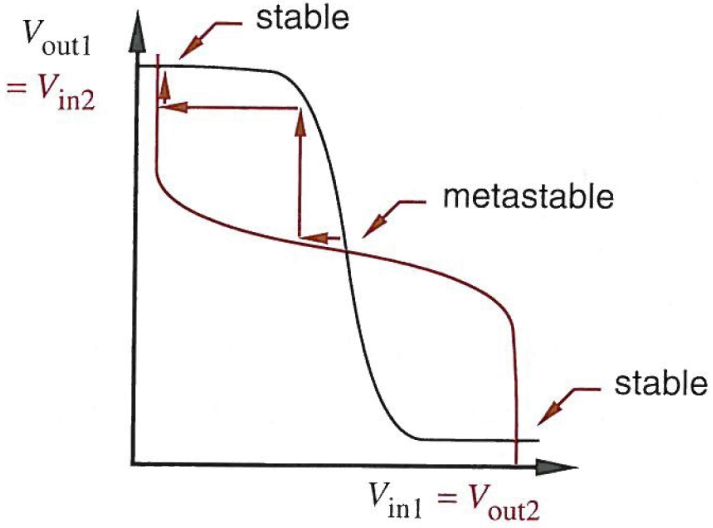
- We said that **feedback in combinational circuits is bad** and should be avoided. We will now add a bit more sophistication to this reasoning.
- The following circuit is **bistable**, i.e. there are two voltages which can determine a stable configuration.
  - we can see this from the transfer functions of the two inverters

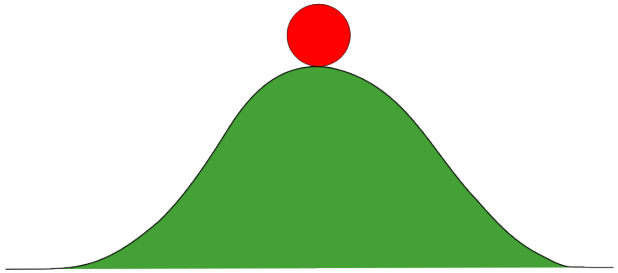




Bistability and metastability (2)

- The curves show that there are ***three* equilibrium points**.
- The middle point represents an unstable equilibrium condition, called ***metastability***.
- A *minimal shift* (e.g. electric noise) from the metastability point will cause the system to converge to one of the stable points. A bit like a ball on top of a hill.

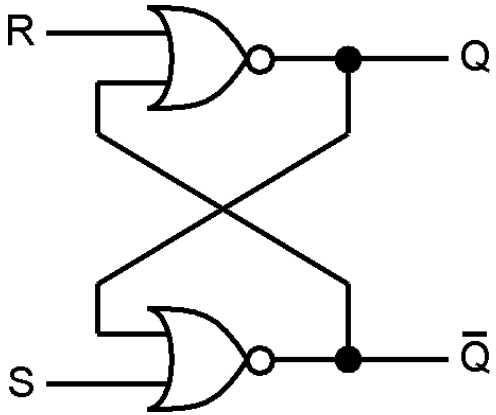




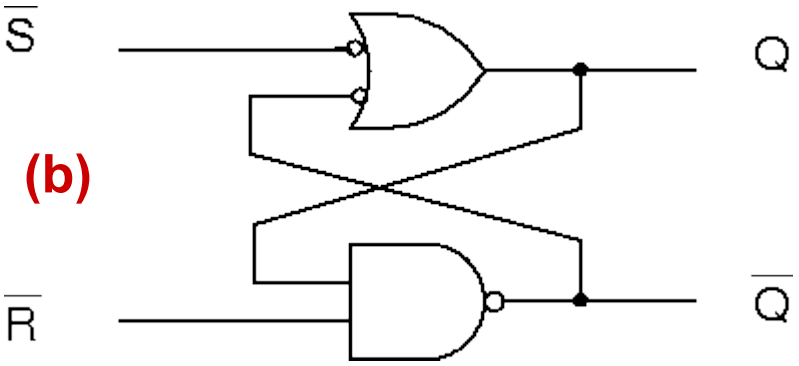
If a system enters a metastable state **it is not possible to know** (stochastic analysis):
- How long it will take to resolve
- Which way it will resolve

Set-Reset (S-R) Latch

(a)



(b)

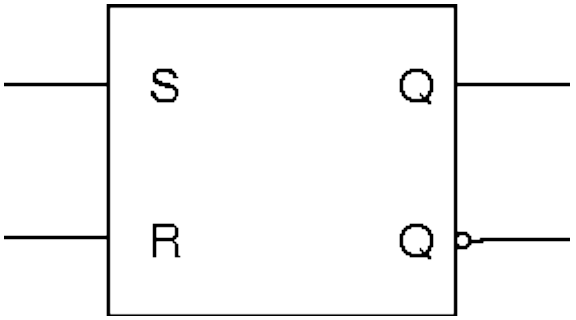


- Two equivalent realisations shown. The NAND-based one is more common in CMOS logic.
- The **S=1, R=1** combination has different output for the two realisations. It is meaningless and should be disallowed. It carries a risk of metastability upon release.

S	R	Q	QN	
0	0	Last Q	Last QN	
1	0	1	0	
0	1	0	1	
1	1	0	0	(a)
1	1	1	1	(b)

SR Latch (2)

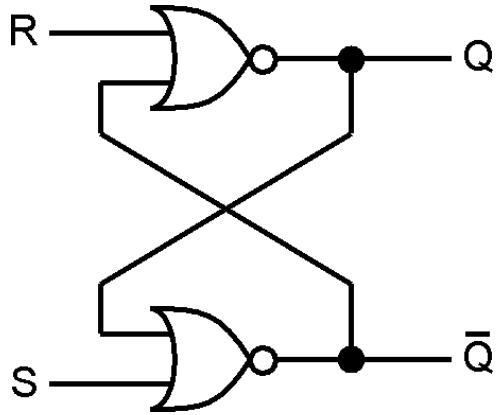
- The standard circuit symbol for the S-R latch



- **Risk of metastability**
  - If the system is switched from the inputs S=1 R=1 to S=0 R=0 what happens to the output?
  - It is impossible to predict accurately. It will depend on the slightest asynchrony between the two changing signals. This is called a **race condition.**

    ***R*=0 before *S*=0 → *Q*=1**

    S=0 before *R*=0 → *Q*=0



Sequential logic

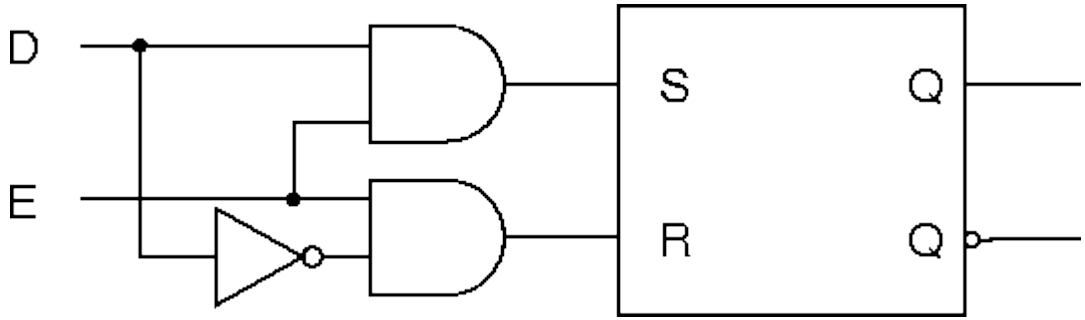
- The S-R latch is our first example of **sequential logic.**
- In a ***combinational*** logic circuit (the ones we have studied so far) the output is a direct function of the input.
- In a ***sequential*** logic circuit the output depends on both the ***current and past inputs.***

- In the case of the latch, once the circuit has been "set", changes in the values of the input S have no impact on the output until the "reset" input R=1 is applied. (typical use: raising a "flag" when an event occurs)

- Can you see how this circuit has some sort of ***memory behaviour****?*

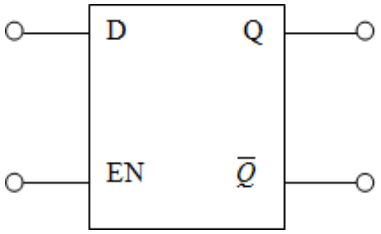
D Latch

- D stands for "Data". It is an extension of the SR latch and is typically used to store data bits as required.

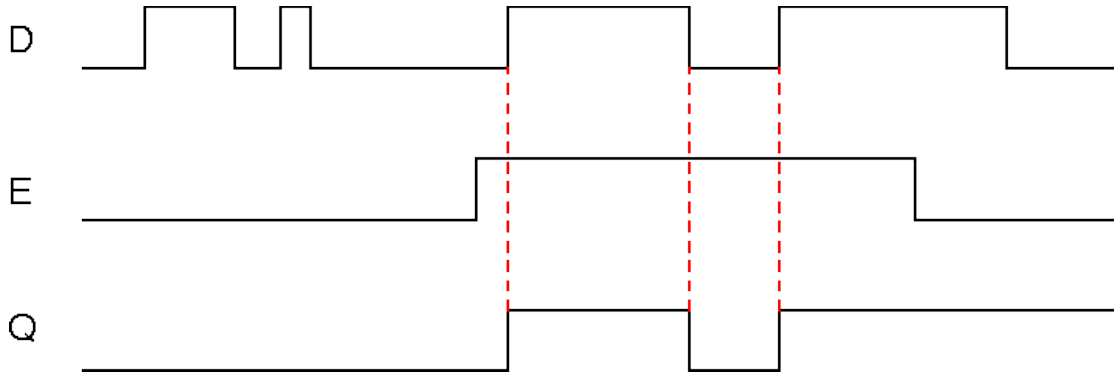


- The D input determines both set and reset events and is gated so that S and R can't equal 1 at the same time.
- The E input acts as an "Enable" function, i.e., the set and reset functions can only be activated if E=1.

D Latch (2)



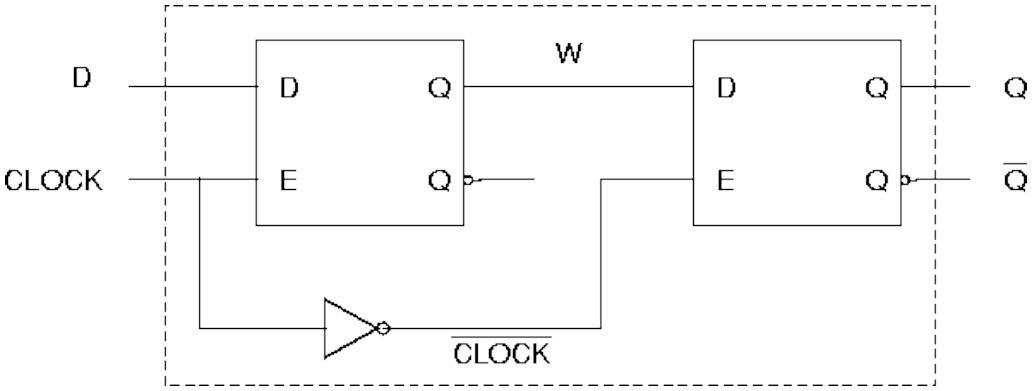
- Standard schematic symbol for a D latch
- An example of (ideal) timing diagram for the D latch



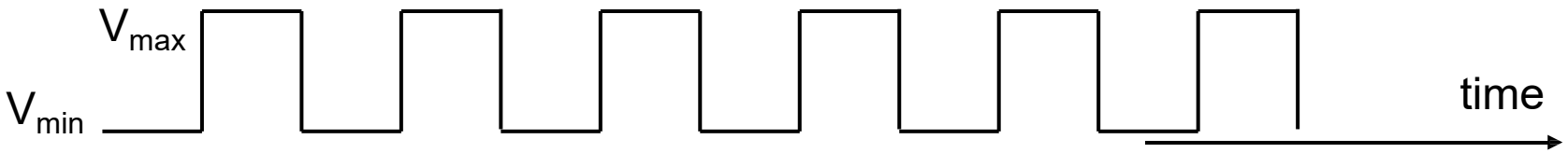
- Note that *Q* responds to changes in *D only* while *E* is active - this is called ***transparency***. i.e., the output *Q* is *transparent* with respect to the input *D* on the condition that *E=1*.

Edge-triggered D Flip-Flop

- A basic circuit made up of **two D Latches** in a *master-slave configuration.* Often called simply D Flip-Flop.

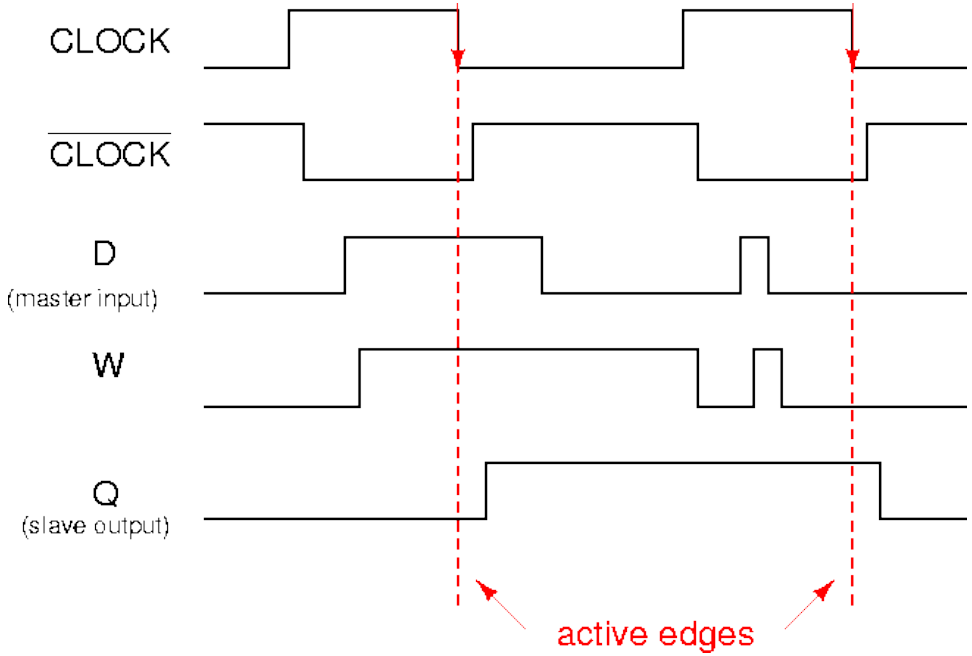


- Note that a ***clock signal*** has been connected to the enable inputs. A clock is a periodic square wave signal.



Edge-triggered D Flip-Flop (2)

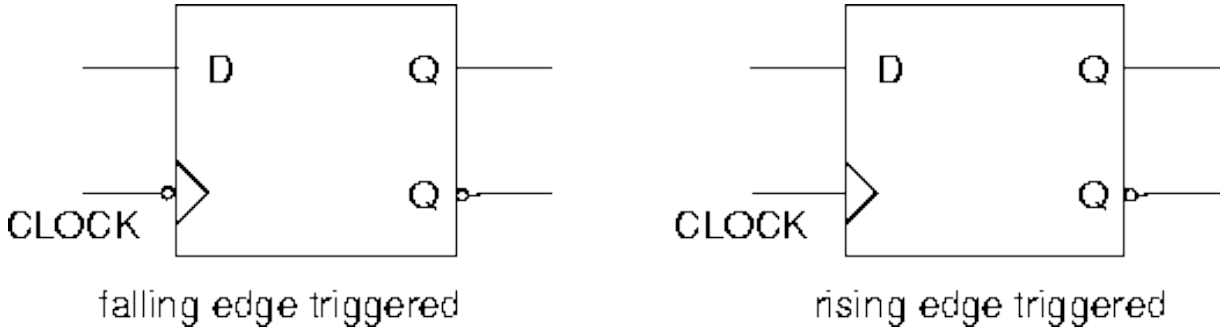
- Below is a timing diagram for a D flip-flop (propagation delays are essential here)



- A D flip-flop stores the data presented at the D input as the *active edge* of the clock occurs (falling edge here).

Edge-triggered D Flip-Flop (3)

- Symbols for rising and falling edge-triggered D flip-flops.



- Flip-flops are extremely important as they can be used to ***synchronise*** circuit events with a precise timing event such as the almost-instantaneous rising/falling edge of a periodic square wave. They are an essential component in sequential designs.

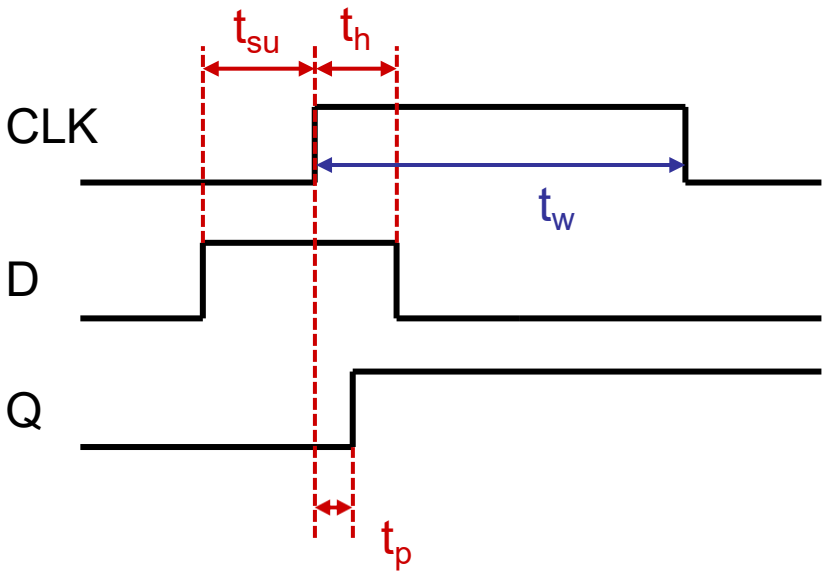
- There are other types of flip-flops but we will only learn about D flip-flops in this course.

D flip-flops and metastability

- The D flip flop is a device with **some complexity**
  - several gates and two latches inside.
  - it has internal dynamics and propagation delays
- The description *a flip-flop samples the input at the time of the active edge* is convenient to remember but must be used with care
- Flip-flops have **timing requirements** which must be met in order to ensure proper operation
- If requirements are not met, **metastability** may occur, resulting in an ***undefined / unpredictable output voltage***

D flip-flops and metastability (2)

- **Essential timing for a D f-f**

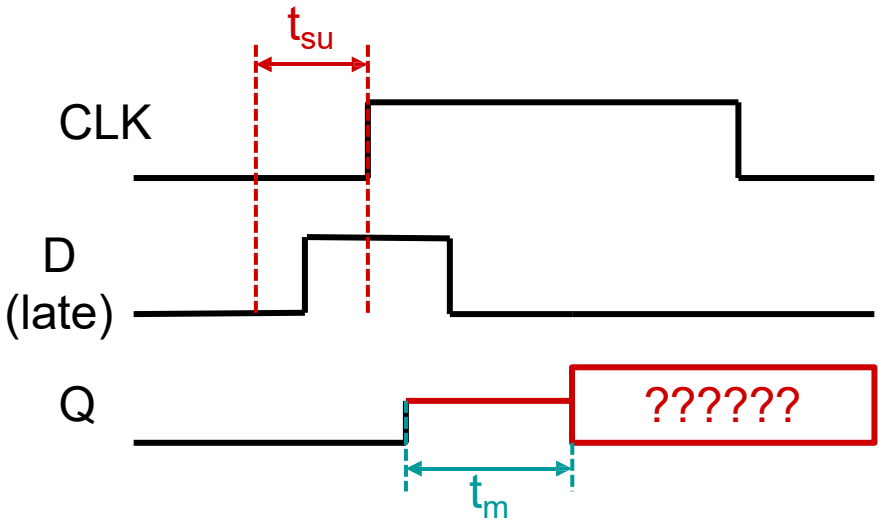


- t_{su} - ***set up time***: the minimum time period immediately prior to an active clock edge during which D must remain stable.
- t_h - ***hold time***: the minimum time period immediately after an active clock edge during which D must remain stable.
- t_w - ***pulse width***: width of the clock pulse.
- t_p - ***propagation delay***: time taken for Q to stabilize after each active clock edge.
- The ***sampling interval*** t_{su}+t_h: D should be stable for this whole time period.

- Flip flop data sheets specify the values for these (and other) parameters which the designer must take into account.

D flip-flops and metastability (3)

- If the setup and hold time requirements are not met, correct operation of the f-f cannot be guaranteed.
  - The example below shows a violation of set-up time

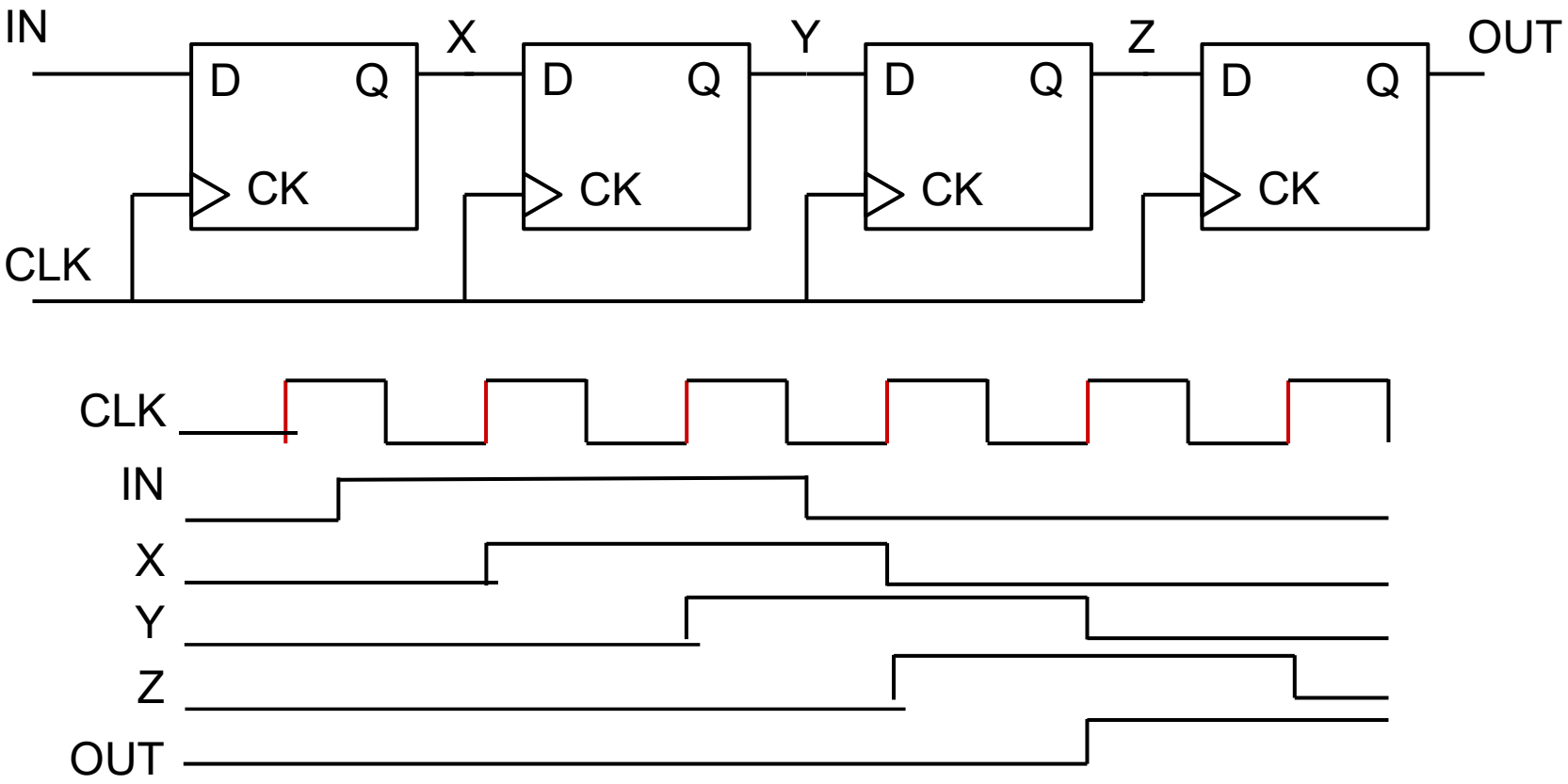


  - The late arriving data can push *Q* half-way from LOW to HIGH, i.e., somewhere in the invalid range of voltages
  - t_m refers to the ***metastability time,*** which is of unknown and uncertain duration. Metastability will resolve by itself but the next output of the f-f is **unknown** (could be either 0 or 1)

Synchronous VS Asynchronous

- ***Synchronous*** (from Greek *syn* + *chronos* = same time)
  - Systems in which signal transitions occur in a periodic/precisely timed manner. A typical example is “clocked” systems.
  - **Sequential digital designs are often synchronous** as this makes it **easier to deal with timing constraints and propagation delays**: as long as successive clock cycles are far enough apart, the system will work.
  - **We will focus on synchronous designs in this course.**
- ***Asynchronous*** systems do not have precise timing
  - Timing of transitions is determined by changes in the input(s) and the internal propagation delays, not by the clock signal.
  - Purely combinational circuits are asynchronous.
- Synchronous systems are an **artificial construct**. Real-world inputs are most likely to be *asynchronous,* and will need to be *synchronised* in order to be used effectively in synchronous designs.
  - We will return on this concept (*synchronisation*) in the next lectures.

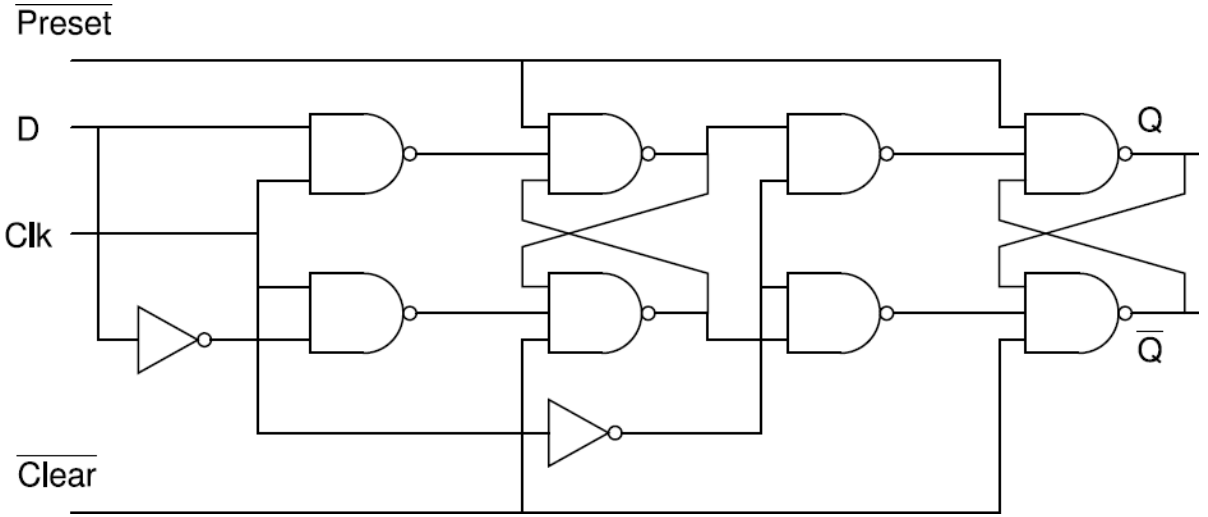
Synchronous VS Asynchronous (2)



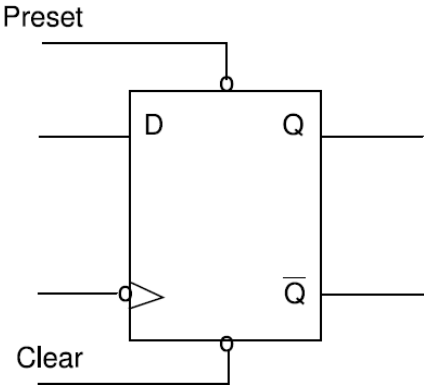
- Synchronous circuit example (delay circuit - *register*)

D Flip-Flop with *asynchronous* Preset and Clear

Logic schematic

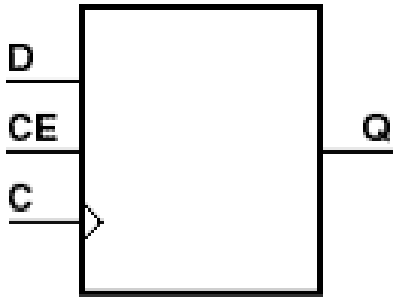


Circuit symbol

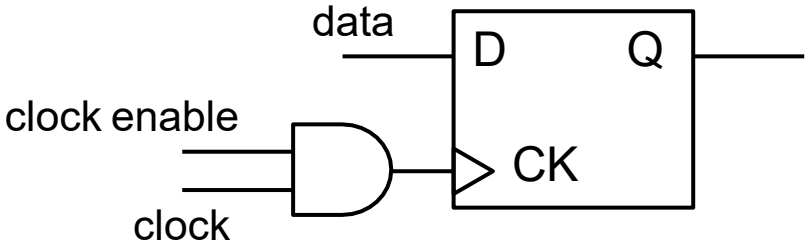


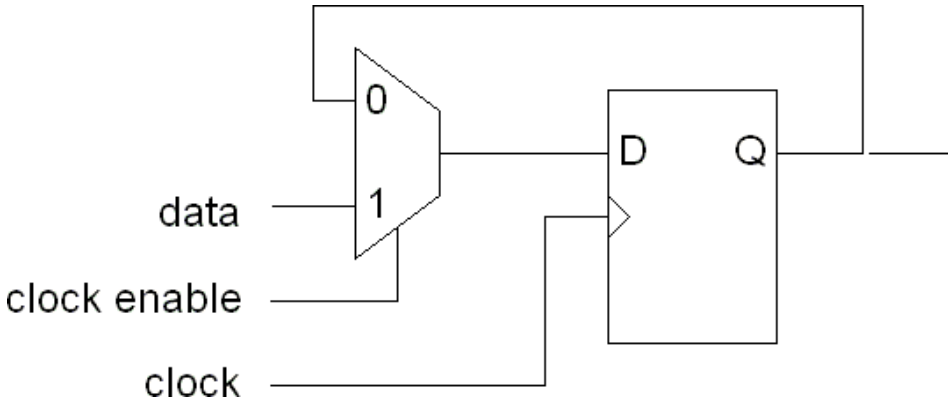
D Flip-Flop with *clock enable (CE)*

Enable = signal which determines whether a block is active or not



Out of the two possible implementations below, which do you think is best (and why)?





Summing up

- We have described some "acceptable" combinational logic systems with **feedback**
  - Latches: S/R and D
  - Use in creating a memory feature in a circuit
- We have met our first ***sequential* logic** component: the **D flip-flop**
  - Derived from the D latch
  - Samples the value of the input as the active edge of an enable signal (clock) occurs
  - Has specific timing requirements
  - Some DFF versions have additional inputs
- We also learned more about **metastability** and the difference between ***synchronous*** and ***asynchronous*** designs