Lecture 14

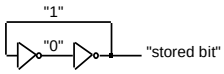
- Memory storage elements
  - Latches
  - Flip-flops
- State Diagrams

Example from last time

- Door combination lock
  - Inputs: Sequence of numbers, reset, new
  - Outputs: Door open/close
  - Memory: Must remember combination
  - Memory: Must remember current state

How do we store information?

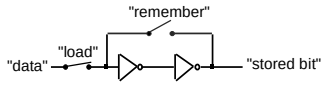
- Feedback
- Example: Two inverters can hold a bit (as long as power is applied)



  - What is missing?
    - How do we change the stored bit?

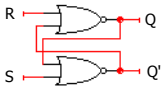
How do we store information?

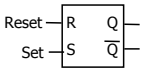
- Storing a new memory
  - Temporarily break the feedback path



SR latch

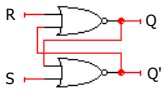
- Cross-coupled NOR gates
  - Can set (S=1, R=0) or reset (R=1, S=0) the output



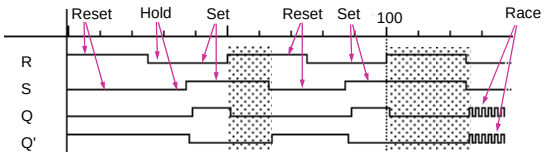


S	R	Q
0	0	hold
0	1	0
1	0	1
1	1	disallow

SR latch behavior

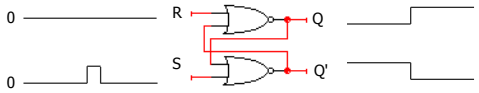


S	R	Q
0	0	hold
0	1	0
1	0	1
1	1	disallow



SR latch is glitch sensitive

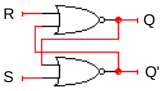
- Static 0 glitches can set/reset latch
  - Glitch on S input sets latch
  - Glitch on R input resets latch



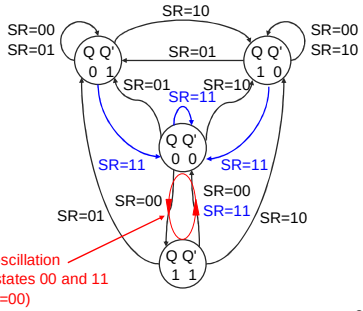
State diagrams

- How do we characterize logic circuits?
  - Combinational circuits: Truth tables
  - Sequential circuits: State diagrams
- First draw the states
  - States ≡ Unique circuit configurations
- Second draw the transitions between states
  - Transitions ≡ Changes in state caused by inputs

Example: SR latch



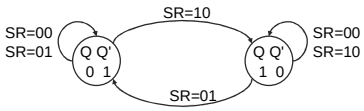
S	R	Q
0	0	hold
0	1	0
1	0	1
1	1	disallow



possible oscillation between states 00 and 11 (when SR=00)

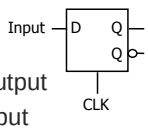
Observed SR latch behavior

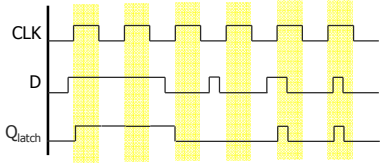
- The 1–1 state is transitory
  - Either R or S "gets ahead"
  - Latch settles to 0–1 or 1–0 state ambiguously
  - Race condition → non-deterministic transition
    - Disallow (R,S) = (1,1)



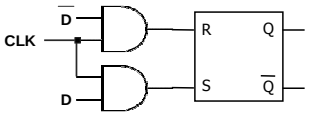
D ("data") latch

- Output depends on clock
  - Clock high: Input passes to output
  - Clock low: Latch holds its output



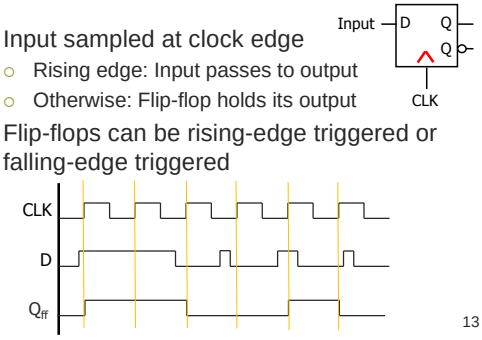


Making a D latch

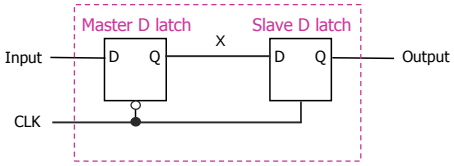


D flip-flop

- Input sampled at clock edge
  - Rising edge: Input passes to output
  - Otherwise: Flip-flop holds its output
- Flip-flops can be rising-edge triggered or falling-edge triggered

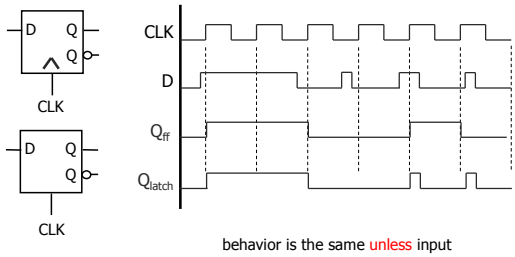


Master-slave D-type flip-flop



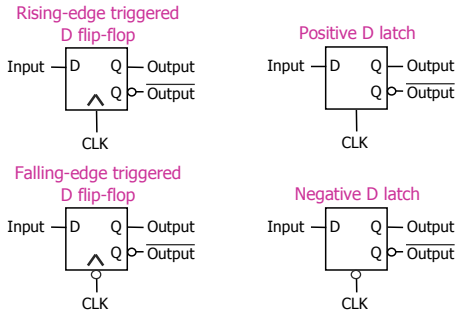
- How to make into negative edge-triggered D-type flip-flop?

Latches versus flip-flops



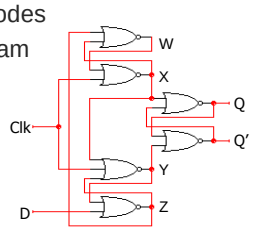
behavior is the same unless input changes while the clock is high

Terminology and notation



How to make a D flip-flop?

- Edge triggering is difficult
  - Label the internal nodes
  - Draw a timing diagram
  - Start with Clk=1



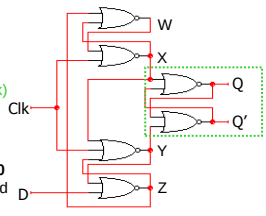
How to make a D flip flop?

Falling edge-triggered flip-flop

If **Clk=1** then **X=Y=0** and SR latch block holds previous values of **Q,Q'**, also **Z=D'** and **W=Z'=D**.

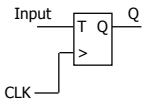
When **Clk→ 0** then **Y** (set for SR latch block) becomes **Z'=D** and **X** (reset for SR latch block) becomes **W'=D'** so **Q** becomes **D**.

While **Clk=0**, if **D** switches then **Z** becomes **0** (because inputs to **Z** are **D** and **D'**) and **X** and **W** hold their previous values and **Y=X'=D** as before.



T ("toggle") flip-flop

- Output toggles when input is asserted
  - If T=1, then Q → Q' when CLK ↑
  - If T=0, then Q → Q when CLK ↑



Input(t)	Q(t)	Q($t + \Delta t$)
0	0	0
0	1	1
1	0	1
1	1	0