ECE 2050 Digital Logic and Systems

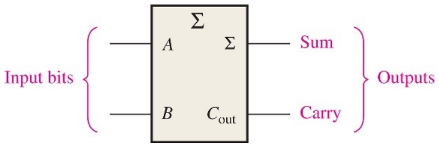
Chapter 7 : Sequential Logic Design

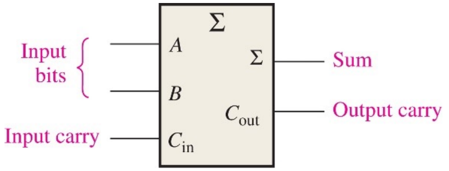
Instructor: Tinghuan CHEN, Ph.D.

香港中文大學（深圳）
The Chinese University of Hong Kong, Shenzhen

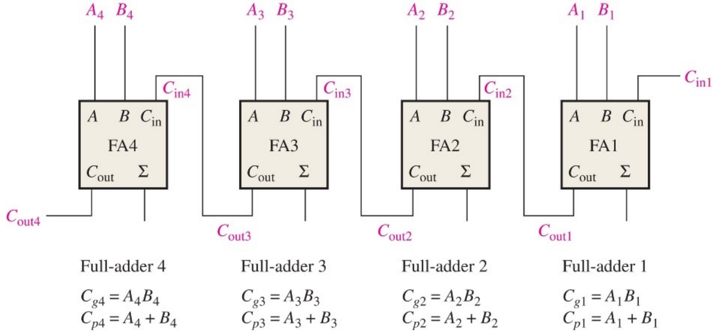
Last Week

- □ Half and Full Adders
- □ Parallel Binary Adders
- □ Ripple Carry and Look-Ahead Carry Adders
- □ Comparators
- □ Decoders (n input lines → max. 2^n output lines)
- □ Encoders (e.g. Decimal-to-BCD Priority Encoder)
- □ Multiplexers (To route several inputs onto a single output line)
- □ Demultiplexers
- □ Timing





How to design a multiplier by FA/HA?



State Elements

Introduction

- Outputs of sequential logic depend on current *and* prior input values – it has ***memory***.

- Some definitions:
  - **State:** all the information about a circuit necessary to explain its future behavior

  - **Latches and flip-flops:** state elements that store one bit of state

  - **Synchronous sequential circuits:** Sequential circuits using flip-flops sharing a common clock

Sequential Circuits

- Give sequence to events
- Have memory (short-term)
- Use feedback from output to input to store information

State Elements

- **State**: everything about the prior inputs to the circuit necessary to predict its future behavior
  - Usually just 1 bit, the last value captured

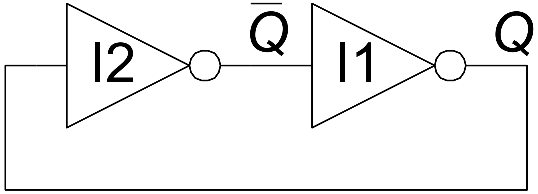
- State elements store state
  - Bistable circuit
  - SR Latch
  - D Latch
  - D Flip-flop

Bistable Circuit

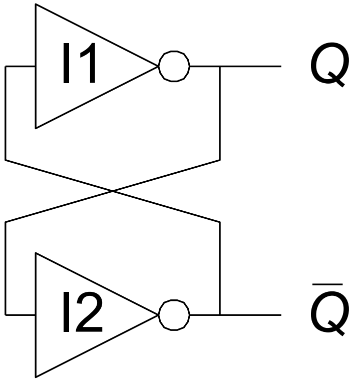
Bistable Circuit

- Fundamental building block of other state elements
- Two outputs: Q, $\overline{Q}$
- No inputs

Same circuit!



Back-to-back inverters



Cross-coupled inverters

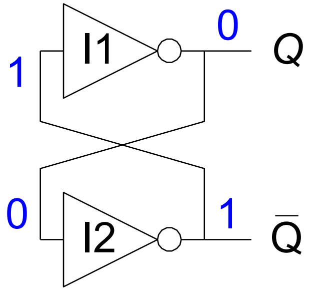
Bistable Circuit Analysis

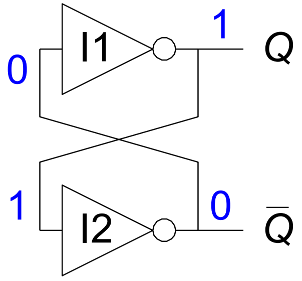
- Consider the two possible cases:
  - ***Q*** **= 0:**

    then $Q = \overline{1}$, $Q = 0$ (consistent)

  - ***Q*** **= 1:**

    then $Q = \overline{0}$, $Q = 1$ (consistent)



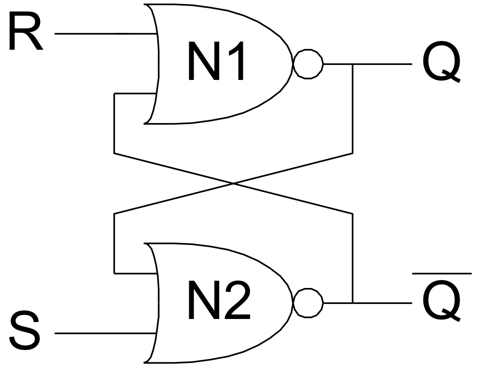


- Stores 1 bit of state in the state variable, Q (or $\overline{Q}$)
- But there are **no inputs to control the state**

SR Latch

SR (Set/Reset) Latch

- **SR Latch**



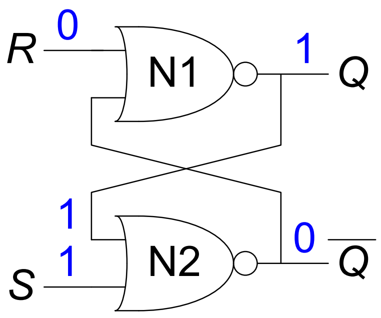
- Consider the four possible cases:
  - ***S* = 1, *R* = 0**
  - ***S* = 0, *R* = 1**
  - ***S* = 0, *R* = 0**
  - ***S* = 1, *R* = 1**

SR Latch Analysis

- ***S* = 1**, *R* = 0**:**

  then ***Q* = 1** and $\overline{Q}$ = 0

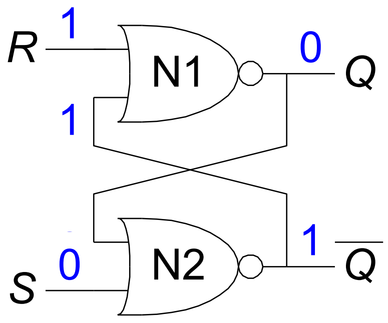
  ***Set* the output**



- *S* = 0, ***R* = 1:**

  then ***Q* = 0** and $\overline{Q}$ = 1

  ***Reset* the output**

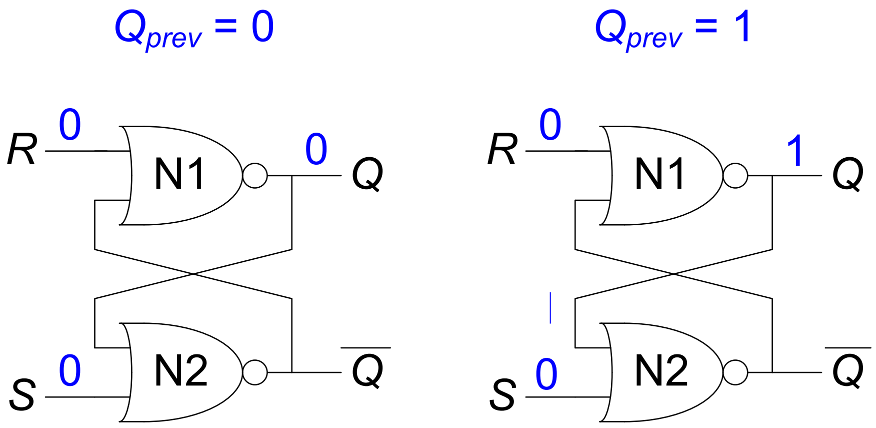


SR Latch Analysis

– ***S* = 0, *R* = 0:**

then $Q = Q_{prev}$

Memory!

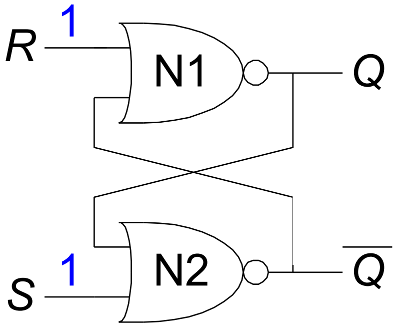


– ***S* = 1, *R* = 1:**

then $Q = 0$, $\overline{Q} = 0$

Invalid State

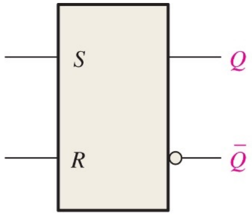
$\overline{Q} \neq \text{NOT } Q$



SR Latch

- **SR** stands for **S**et/**R**eset Latch
  - Stores one bit of state (Q)
- Control what value is being stored with S, R inputs
  - **Set:** Make the output 1
    - $\boldsymbol{S = 1}$, $R = 0$, $Q = \mathbf{1}$
  - **Reset:** Make the output 0
    - $S = 0$, $\boldsymbol{R = 1}$, $Q = \mathbf{0}$
  - **Memory:** Retain value
    - $S = 0$, $R = 0$, $Q = \boldsymbol{Q_{prev}}$

SR Latch symbol



- **Must do something to avoid invalid state (when $S = R = 1$)**

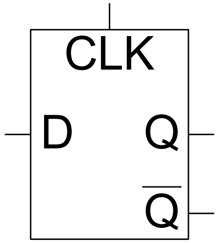
D Latch

D Latch

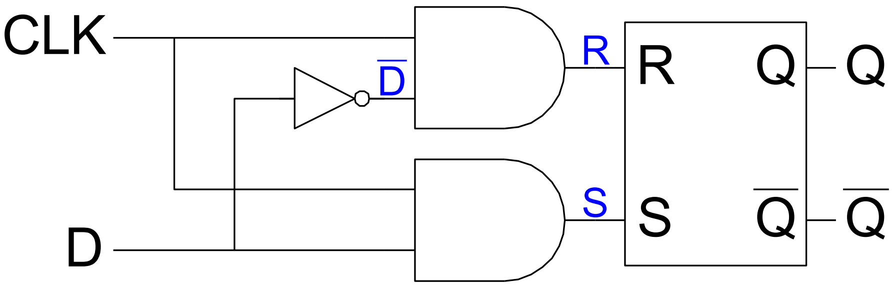
- **Two inputs:** *CLK*, *D*
  - ***CLK*:** controls *when* the output changes
  - ***D*** (the data input): controls *what* the output changes to

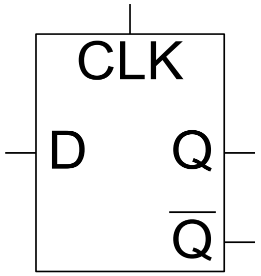
- **Function**
  - When ***CLK* = 1**,
  - *D* passes through to *Q* (*transparent*)
  - When ***CLK* = 0**,
  - *Q* holds its previous value (*opaque*)

D Latch Symbol



D Latch Internal Circuit



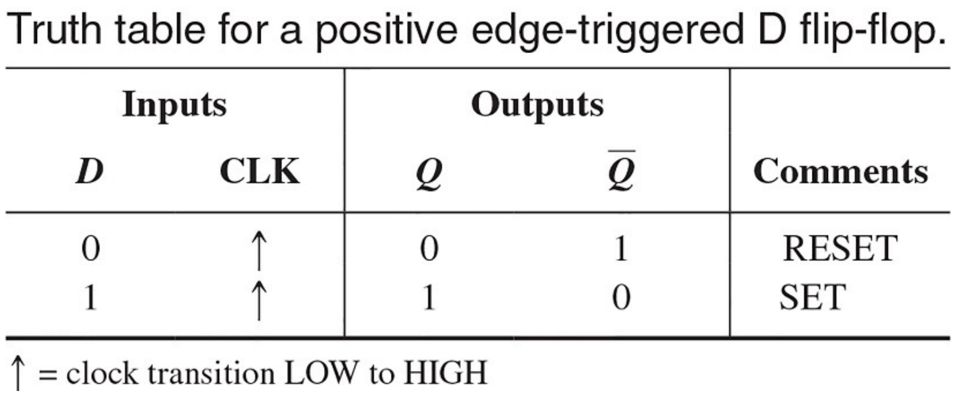


CLK	*D*	$\overline{D}$	*S*	*R*	*Q*	$\overline{Q}$
0	X	$\overline{X}$	0	0	Q_{prev}	$\overline{Q}_{prev}$
1	0	1	0	1	0	1
1	1	0	1	0	1	0

D Flip Flop

D Flip-Flop

- **Inputs:** *CLK*, *D*
- **Truth Table:**

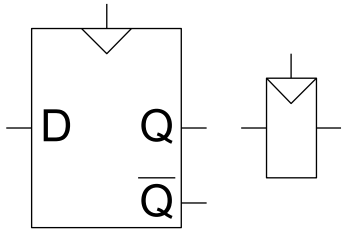


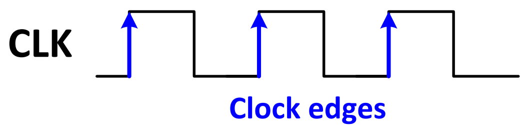
Truth table for a positive edge-triggered D flip-flop.

Inputs		Outputs		
D	CLK	*Q*	$\overline{Q}$	Comments
0	↑	0	1	RESET
1	↑	1	0	SET

↑ = clock transition LOW to HIGH

D Flip-Flop Symbols





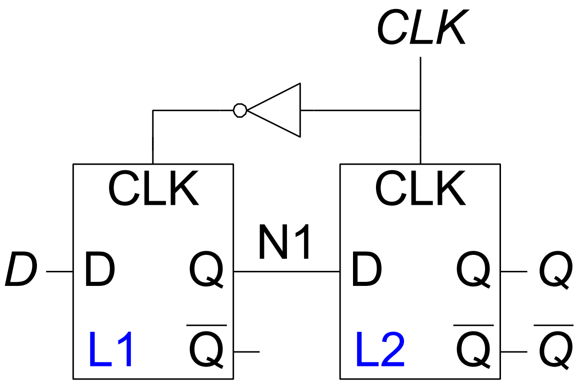
- Called ***edge-triggered***
  - Activated on the *clock edge*

D Flip-Flop Internal Circuit

- **Two back-to-back D latches** (L1 and L2) controlled by complementary clocks

- When ***CLK*** **= 0**
  - L1 is transparent
  - L2 is opaque
  - ***D*** passes through to **N1**

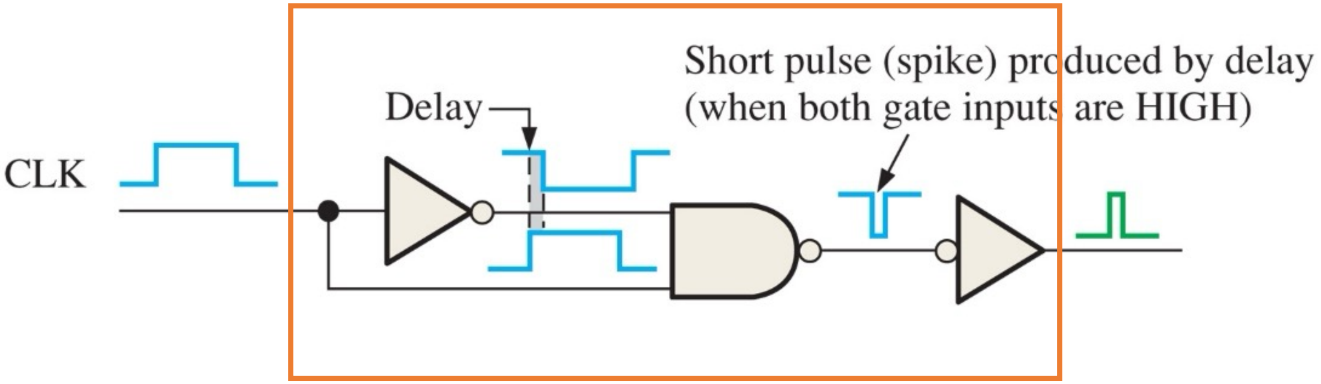
- When ***CLK*** **= 1**
  - L2 is transparent
  - L1 is opaque
  - **N1** passes through to ***Q***



- Thus, on the edge of the clock (when ***CLK*** **rises from 0 → 1**)
  - ***D*** passes through to ***Q***

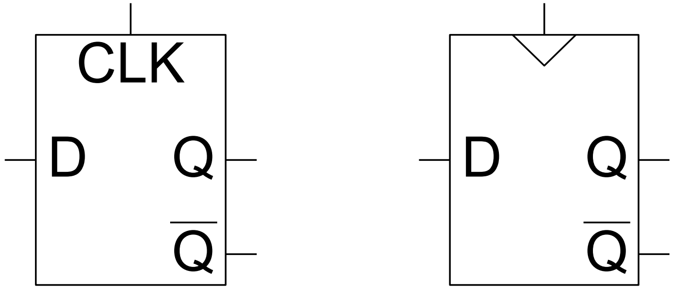
Edge-Triggered Operation

- By exploiting a small delay through the inverter on one input to the NAND gate, generate a very short-duration spike on the positive-going transition of the clock pulse.


Short pulse (spike) produced by delay
(when both gate inputs are HIGH)
Delay
CLK

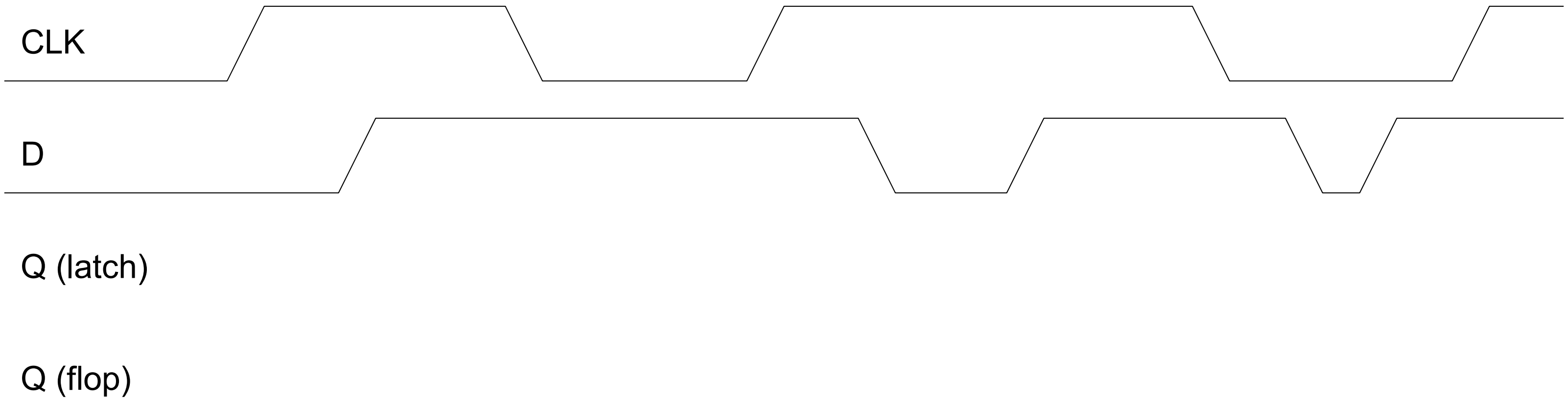
A type of pulse transition detector

D Latch vs. D Flip-Flop



D Latch

D Flip-flop

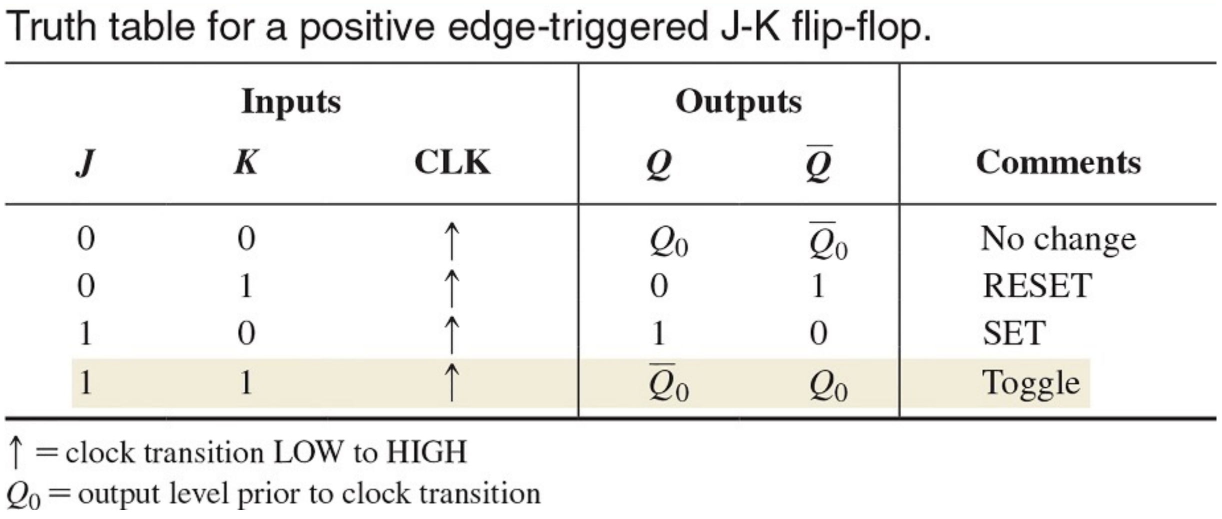


J-K Flip Flop

J-K Flip Flop

- **Inputs:** *CLK, J, K*

- **Truth Table:**



Truth table for a positive edge-triggered J-K flip-flop.

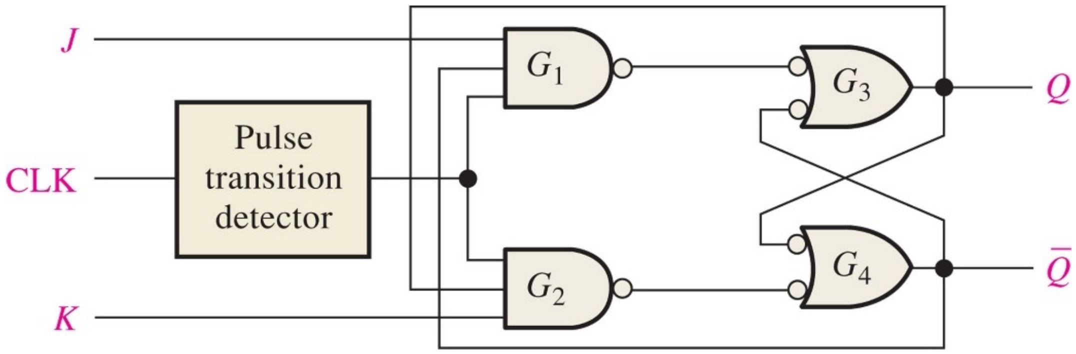
Inputs			Outputs		
J	***K***	**CLK**	***Q***	$\overline{Q}$	**Comments**
0	0	↑	Q_0	$\overline{Q}_0$	No change
0	1	↑	0	1	RESET
1	0	↑	1	0	SET
1	1	↑	$\overline{Q}_0$	Q_0	Toggle

↑ = clock transition LOW to HIGH

Q_0 = output level prior to clock transition

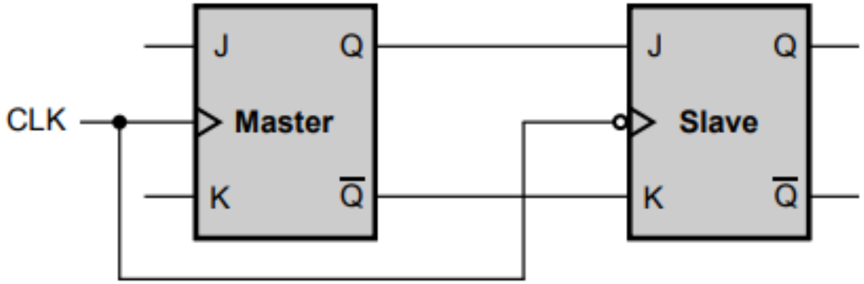
- ***Toggle State***

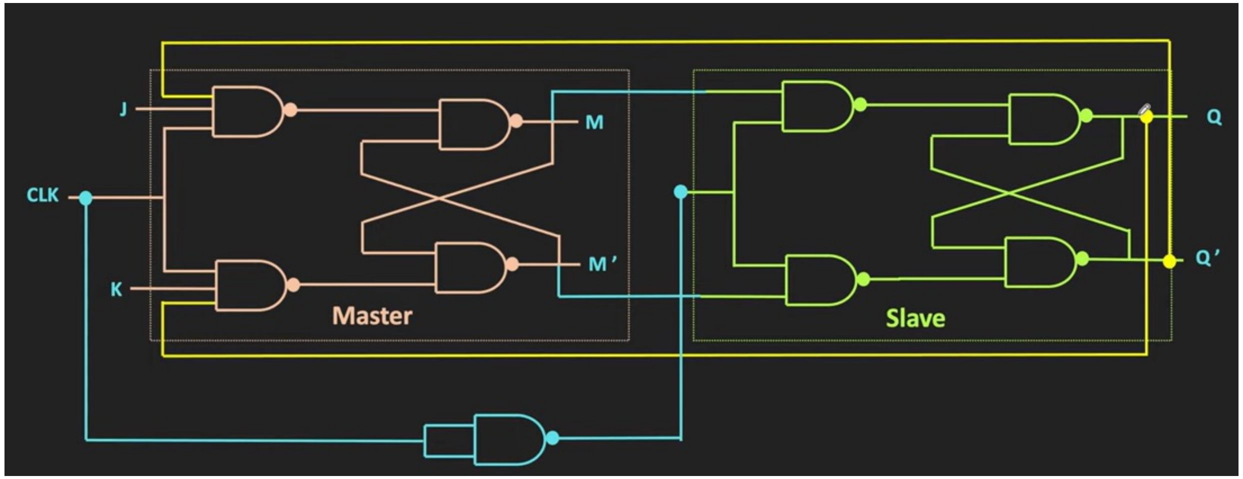
J-K Flip-Flop Internal Circuit

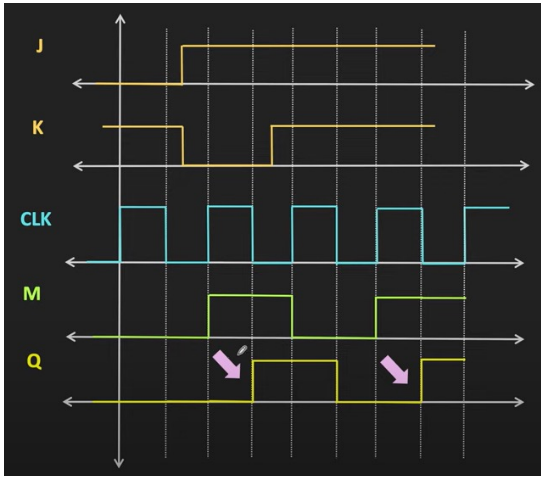


J	K	Q_N	Q_{N+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Master-Slave JK Flip-Flop (Optional)







Ref: Master Slave JK Flip-Flop Explained | Digital Electronics
https://www.youtube.com/watch?v=XgRmLl9uRfI

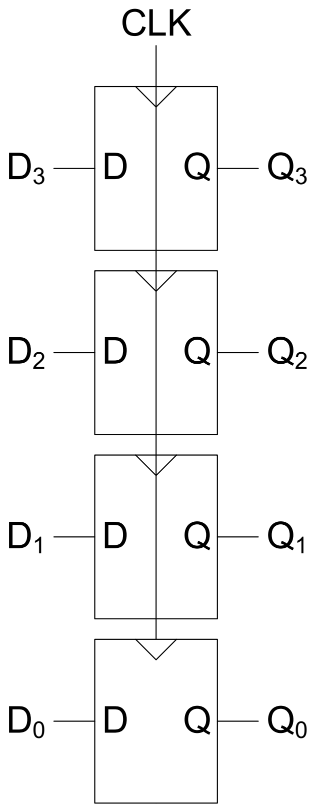
Master-Slave vs PTD-based Design (Optional)

	Master-Slave	PTD-based
Architecture	Two back-to-back latches + clock inversion	Single latch + pulse generator (delay + AND gate)
Pros	Robust, PVT-tolerant, clean timing	Compact, low power, faster
Cons	Larger area, higher power, legacy compatibility	Sensitive to pulse width, metastability risk
Applications	CPUs, FPGAs, industrial controls	High-speed I/Os, low-power IoT, RF circuit

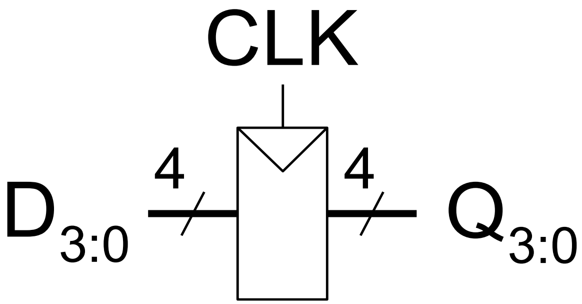
Takeaway: *Choose Master-Slave for reliability, PTD for speed/power.*

Variations on a Flop

Registers: One or More Flip-Flops



4-bit Register

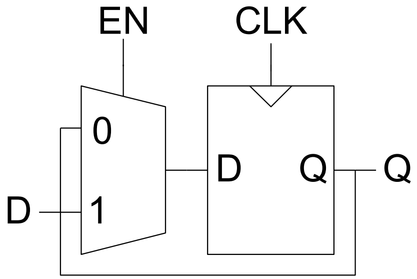


4-bit Register

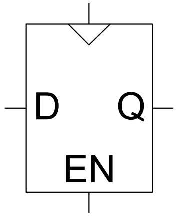
Enabled Flip-Flops

- **Inputs:** *CLK*, *D*, *EN*
  - The enable input (*EN*) controls when new data (*D*) is stored
- **Function**
  - ***EN* = 1:** *D* passes through to *Q* on the clock edge
  - ***EN* = 0:** the flip-flop retains its previous state

Internal Circuit



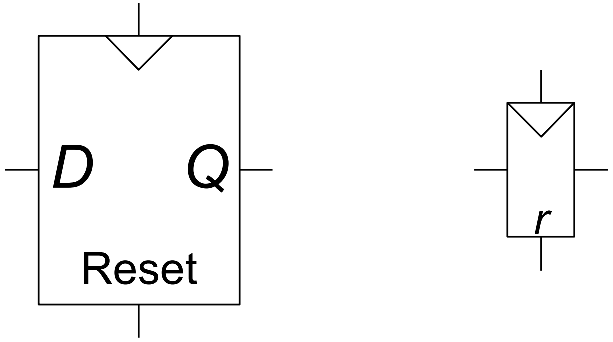
Symbol



Resettable Flip-Flops

- **Inputs:** *CLK*, *D*, *Reset*
- **Function:**
  - ***Reset* = 1:** *Q* is forced to 0
  - ***Reset* = 0:** flip-flop behaves

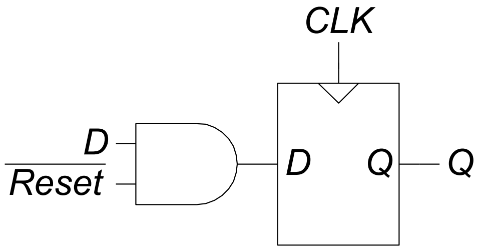
Symbols



Resettable Flip-Flops

- Two types:
  - **Synchronous:** resets at the clock edge only
  - **Asynchronous:** resets immediately when *Reset* = 1
- **Asynchronously** resettable flip-flop requires changing the internal circuitry of the flip-flop
- **Synchronously** resettable flip-flop?

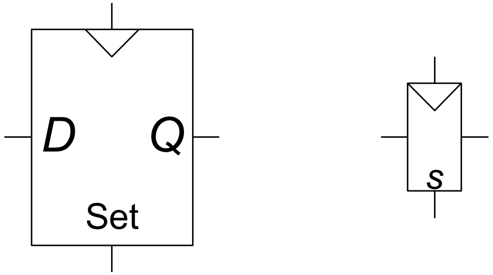
Internal Circuit



Settable Flip-Flops

- **Inputs:** *CLK*, *D*, *Set*
- **Function:**
  - ***Set* = 1:** *Q* is set to 1
  - ***Set* = 0:** the flip-flop behaves as ordinary D flip-flop

Symbols



Chapter Review

- ❑ State
- ❑ SR Latch, D Latch
- ❑ D Flip Flop
- ❑ J-K Flip Flop
- ❑ Variations on a Flop
    - ❑ Registers
    - ❑ Enabled Flip-Flops
    - ❑ Resettable Flip-Flops
    - ❑ Settable Flip-Flops

True/False Quiz

- ✗ A latch has one stable state.
- ✓ A latch is considered to be in the RESET state when the Q output is low.
- ✓ Flip-flops and latches are both bistable devices.
- ✗ An edge-triggered D flip-flop changes state whenever the D input changes.
- ✓ A clock input is necessary for an edge-triggered flip-flop.
- ✓ When both the J and K inputs are HIGH, an edge-triggered J-K flip-flop changes state on each clock pulse.