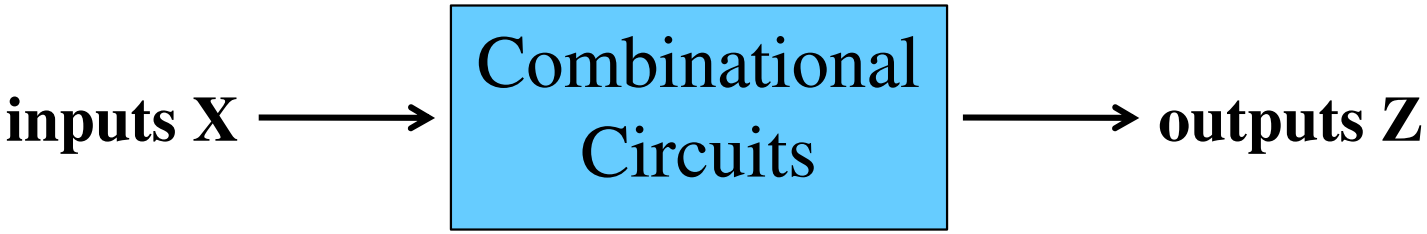
Sequential Circuits : Introduction

- Sequential Circuits
- Storage Elements (Memory)
  - Latches
  - Flip-Flops

Sequential Circuits

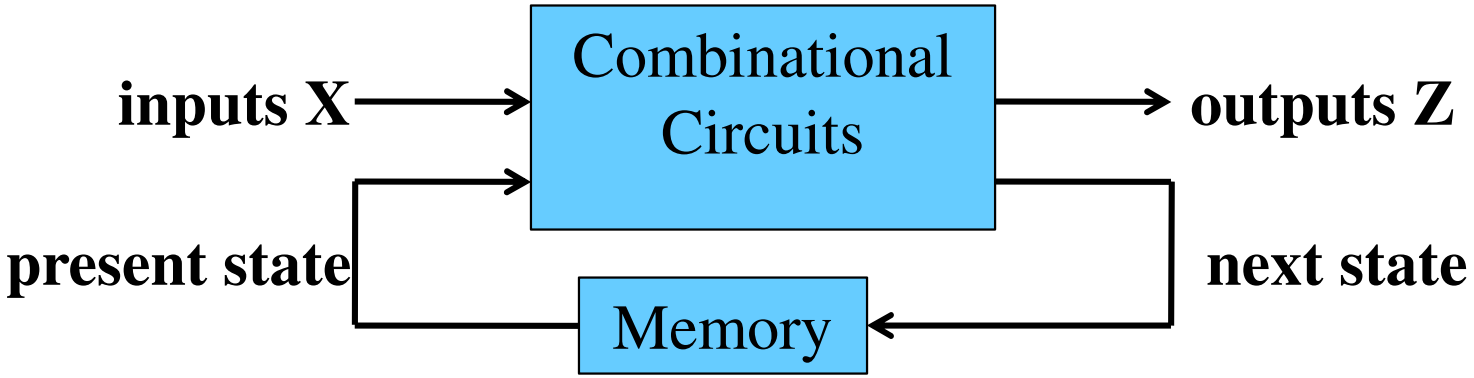
- All of the previous circuits were combinational circuits
  - Current flowed in at one end and out the other
  - Combinational circuits cannot ***retain*** values
  - If we want to build a kind of memory, we need to use a sequential circuit
    - In a sequential circuit, current flows into the circuit and stays there
    - This is done by looping the output back into the input
  - Sequential circuits will be used to implement 1-bit storage
  - We can then combine 1-bit storage circuits into groups for n-bit storage (registers, cache)
    - These circuits will be known as flip-flops because they can flip from one state (storing 1) to another (storing 0) or vice versa

COMBINATIONAL VS SEQUENTIAL



- A **combinational** circuit:
  - At any time, outputs depends only on inputs
  - Changing inputs changes outputs
  - No regard for previous inputs
  - No memory (history)
  - Time is ignored !

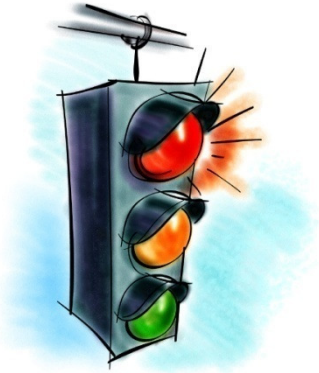
COMBINATIONAL VS SEQUENTIAL (2)



❖A **sequential** circuit:

- A combinational circuit with **feedback** through **memory**
  - The stored information at any time defines a **state**
- Outputs depends on inputs and previous inputs
  - Previous inputs are stored as binary information into memory
- Next state depends on inputs and present state

EXAMPLES OF SEQUENTIAL SYSTEMS



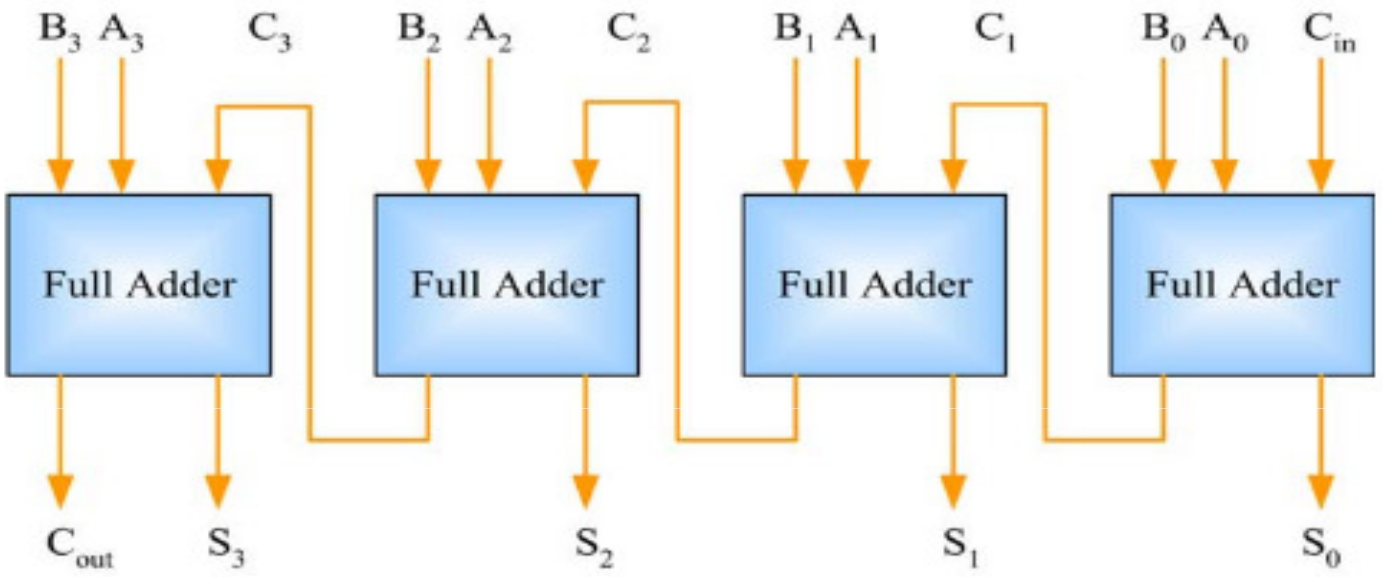
Traffic light

ATM

Vending machine

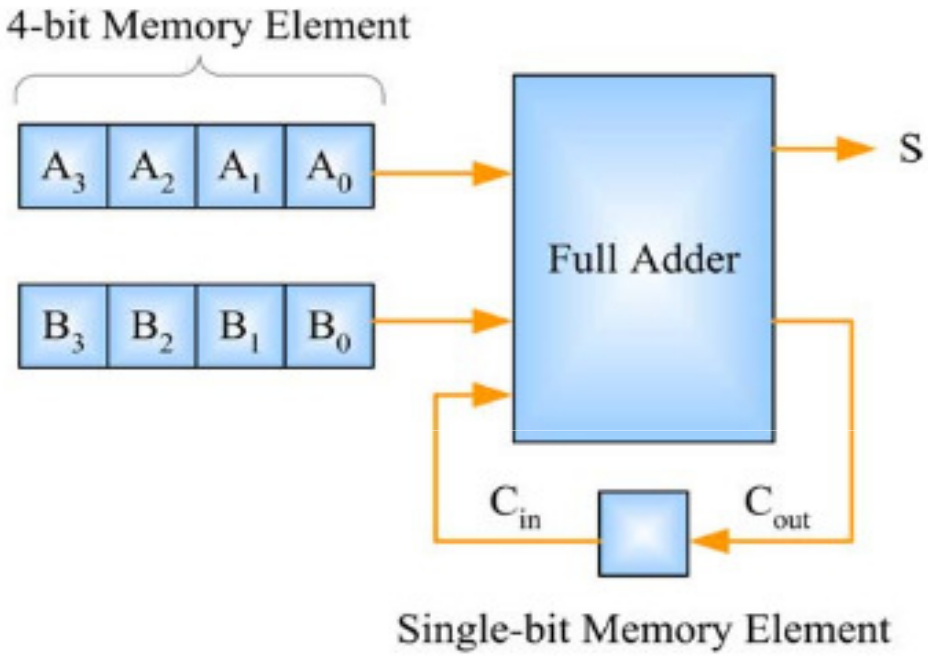
What is common between these systems?

COMBINATIONAL ADDER



- 4-bit adder (ripple-carry)
  - Notice how carry-out propagates
  - One adder is active at a time
- 4 full adders are needed

SEQUENTIAL ADDER

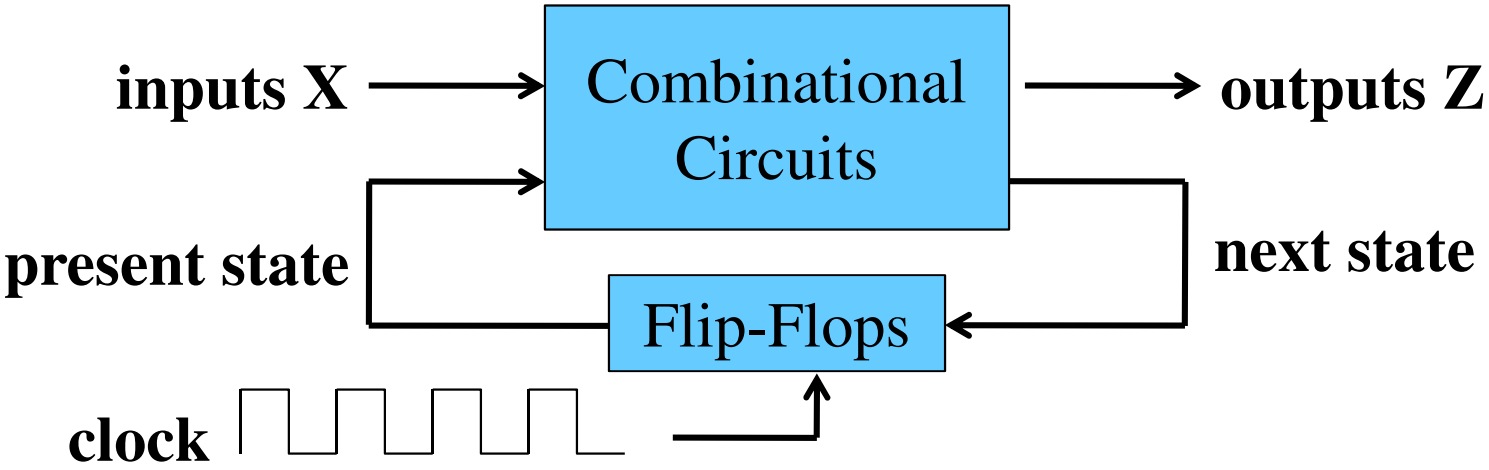


- 1-bit memory and 2 4-bit memory
- Only one full-adder!
- 4 clocks to get the output
- The 1-bit memory defines the circuit state (0 or 1)

Types of Sequential Circuits

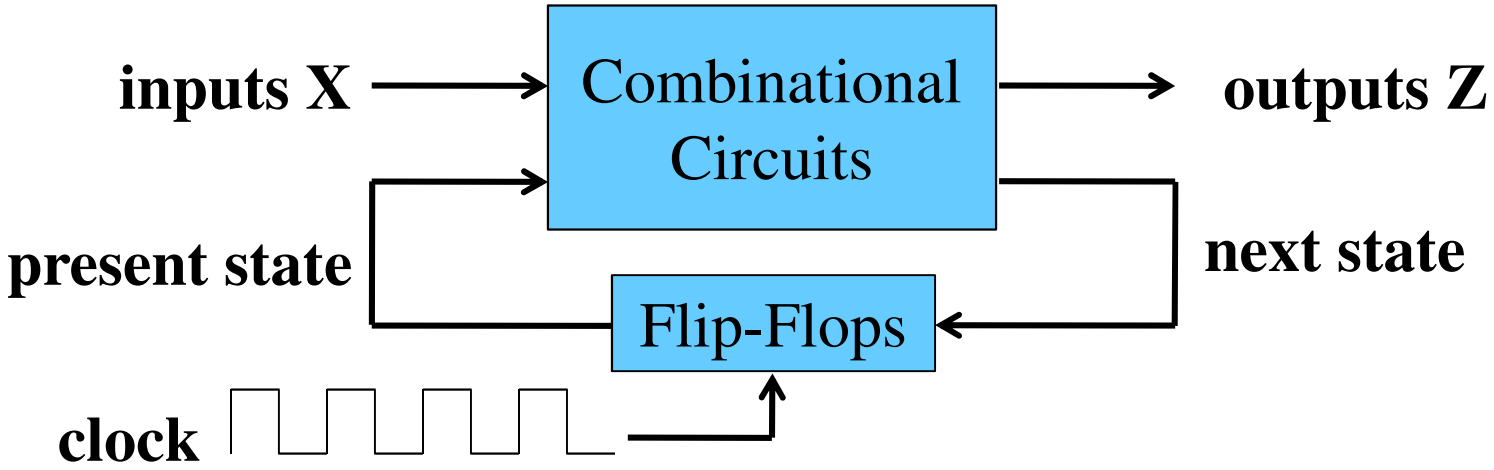
- Two types of sequential circuits:
  - **Synchronous**: The behavior of the circuit depends on the input signal at discrete instances of time (*also called clocked*)
  - **Asynchronous**: The behavior of the circuit depends on the input signals at any instance of time and the order of the inputs change
    - A combinational circuit with feedback

SYNCHRONOUS SEQUENTIAL CIRCUITS



- Synchronous circuits employs a synchronizing signal called **clock** (a periodic train of pulses; 0s and 1s)
- A clock determines **when** computational activities occur
- Other signals determines **what** changes will occur

SYNCHRONOUS SEQUENTIAL CIRCUITS



- The storage elements (memory) used in clocked sequential circuits are called **flip-flops**
  - Each flip-flop can store one bit of information 0,1
  - A circuit may use many flip-flops; together they define the circuit state
- Flip-Flops (memory/state) update **only** with the clock

Storage Elements (Memory)

- A storage element can maintain a binary state (0,1) indefinitely, until directed by an input signal to switch state
- Main difference between storage elements:
  - Number of inputs they have
  - How the inputs affect the binary state
- Two main types:
  - **Latches** (level-sensitive)
  - **Flip-Flops** (edge-sensitive)
- Latches are useful in asynchronous sequential circuits
- Flip-Flips are built with latches

CHARACTERISTIC TABLES

- A **characteristic table** defines the operation of a latch or flip-flop in a tabular form
- Next state is defined in terms of the current state and the inputs
  - Q(t) (or just Q) refers to current state (before the clock arrives)
  - Q(t+1) (or Q_+)refers to next state (after the clock arrives)
- Similar to the truth table in combinational circuits

Flip-Flop Characteristic Tables

JK Flip-Flop

J	K	Q(t + 1)	
0	0	$Q(t)$	No change
0	1	0	Reset
1	0	1	Set
1	1	$Q'(t)$	Complement

D Flip-Flop

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

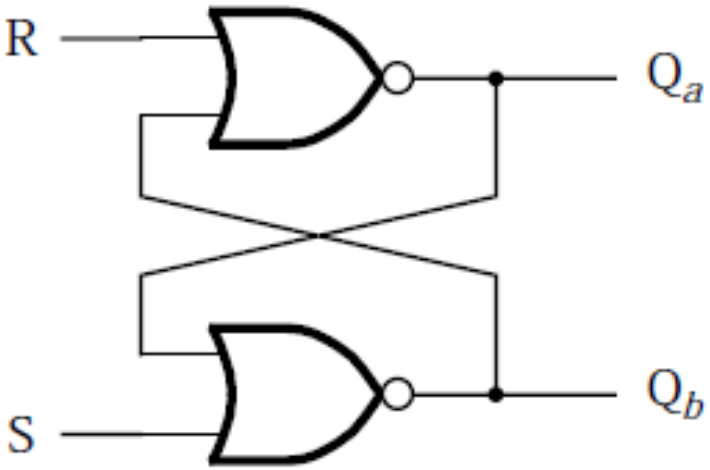
T Flip-Flop

T	Q(t + 1)	
0	$Q(t)$	No change
1	$Q'(t)$	Complement

LATCHES

- A **latch** is a binary storage element
- Can store a 0 or 1
- The most basic memory
- Easy to build
  - Built with gates (NORs, NANDs, NOT)

SR LATCH



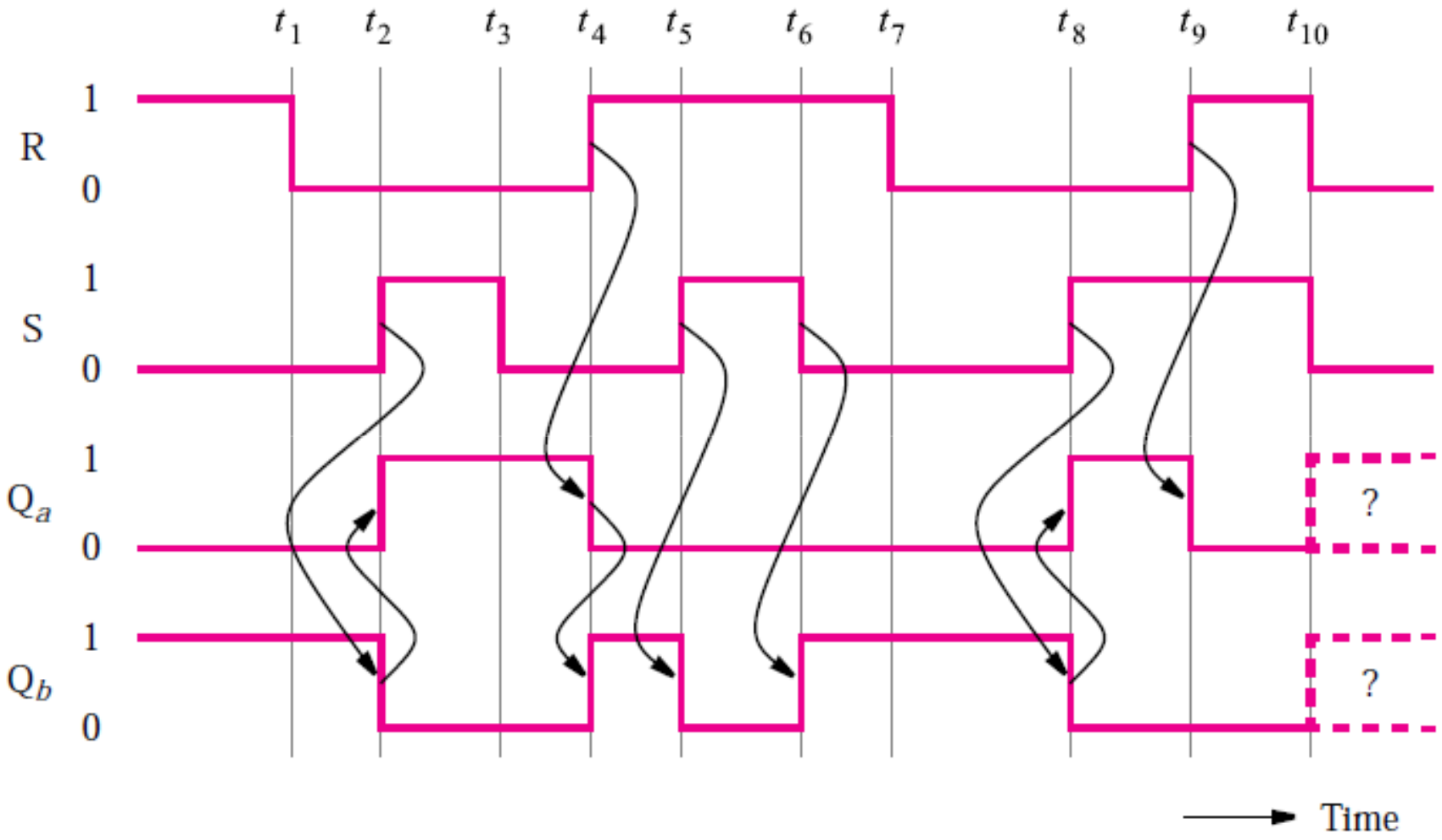
Circuit

S	R	Q_a	Q_b	
0	0	0/1	1/0	(no change)
0	1	0	1	Reset
1	0	1	0	Set
1	1	0	0	Undefined

Characteristic Table

- Two states: Set ($Q_a = 1$) and Reset ($Q_a = 0$)
- When S=R=0, Q_a remains the same, S=R=1 is not allowed!
- Normally, S=R=0 unless the state needs to be changed (memory?)
- State of the circuit depends not only on the current inputs, but also on the recent history of the inputs

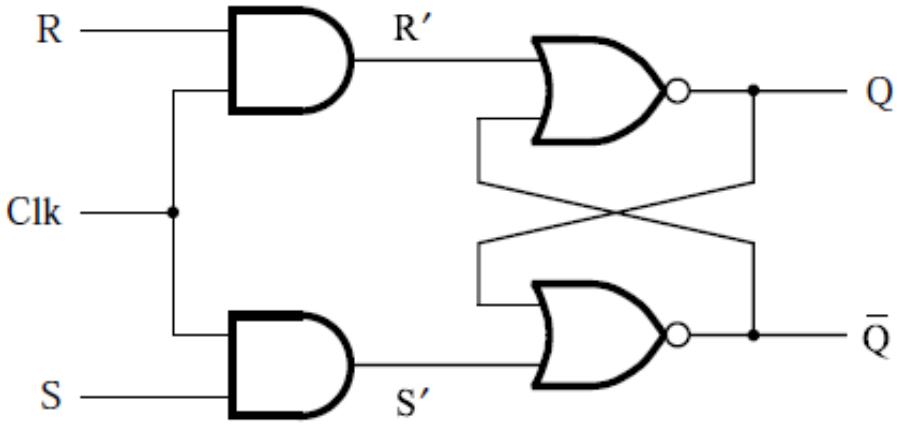
SR Latch Timing Diagram



UNSTABLE STATE (OSCILLATION)

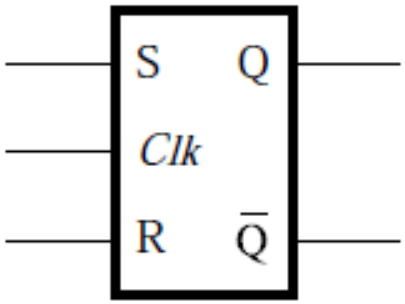
- After t_{10} : If S and R remains 0,
  - $Q_a : 0 \rightarrow 1 \rightarrow 0 \rightarrow 1$
  - $Q_b : 0 \rightarrow 1 \rightarrow 0 \rightarrow 1$
  - If the delays through two NOR gates are exactly the same, oscillation continues indefinitely → Unstable!
  - If the delays slightly different → one "uncertain" state
  - S=R=1 not allowed!

GATED SR LATCH



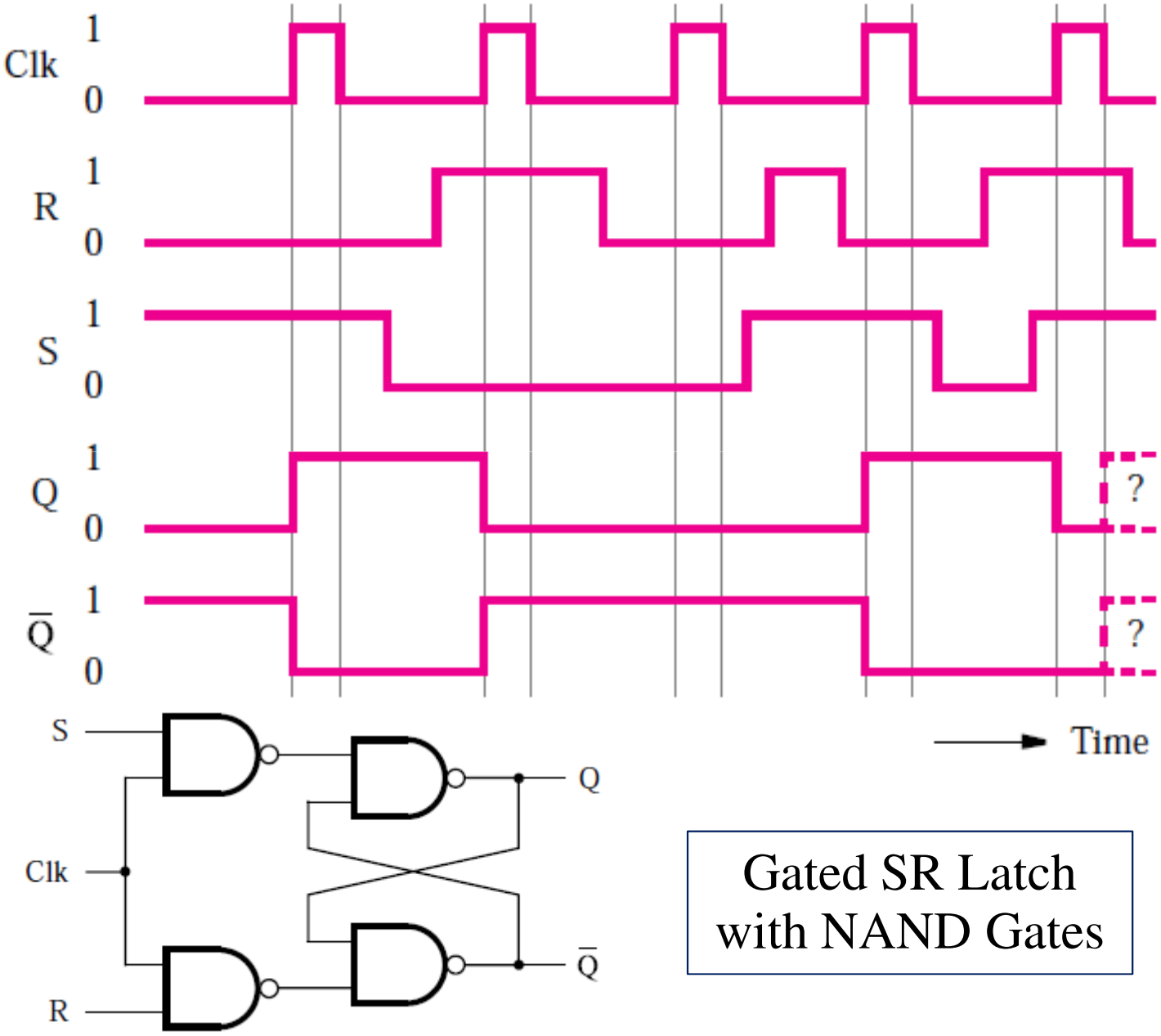
Clk	S	R	Q(t + 1)
0	x	x	Q(t) (no change)
1	0	0	Q(t) (no change)
1	0	1	0
1	1	0	1
1	1	1	x

- An SR Latch can be modified to control **when** it changes
- An additional input signal Clock (Clk)
- When C=0, the S and R inputs have no effect on the latch
- When C=1, the inputs affect the state of the latch and possibly the output



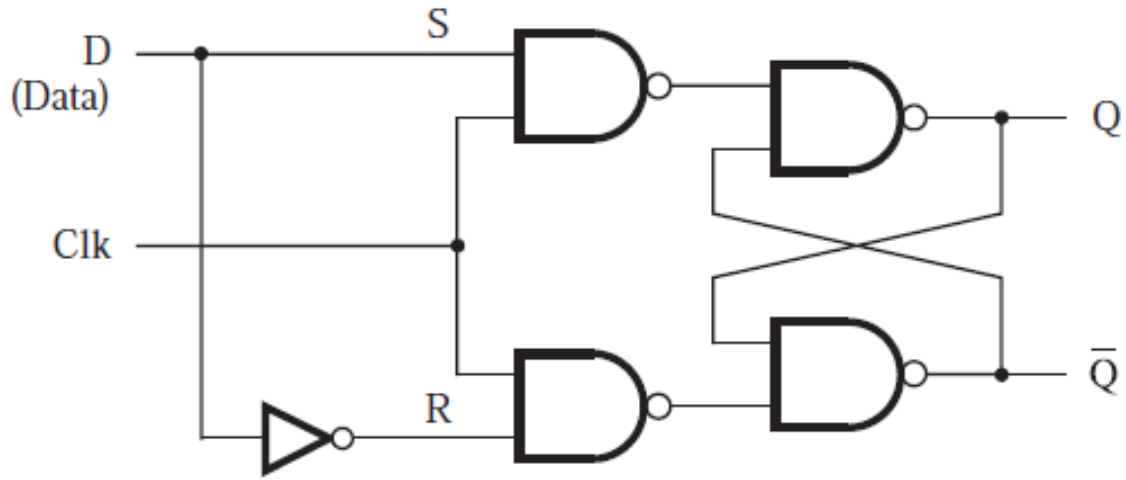
Graphical Symbol

Gated SR Latch Timing Diagram



Gated SR Latch with NAND Gates

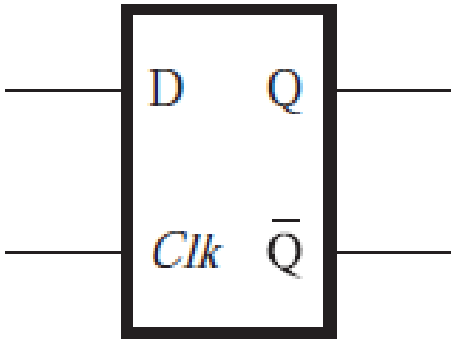
GATED D LATCH

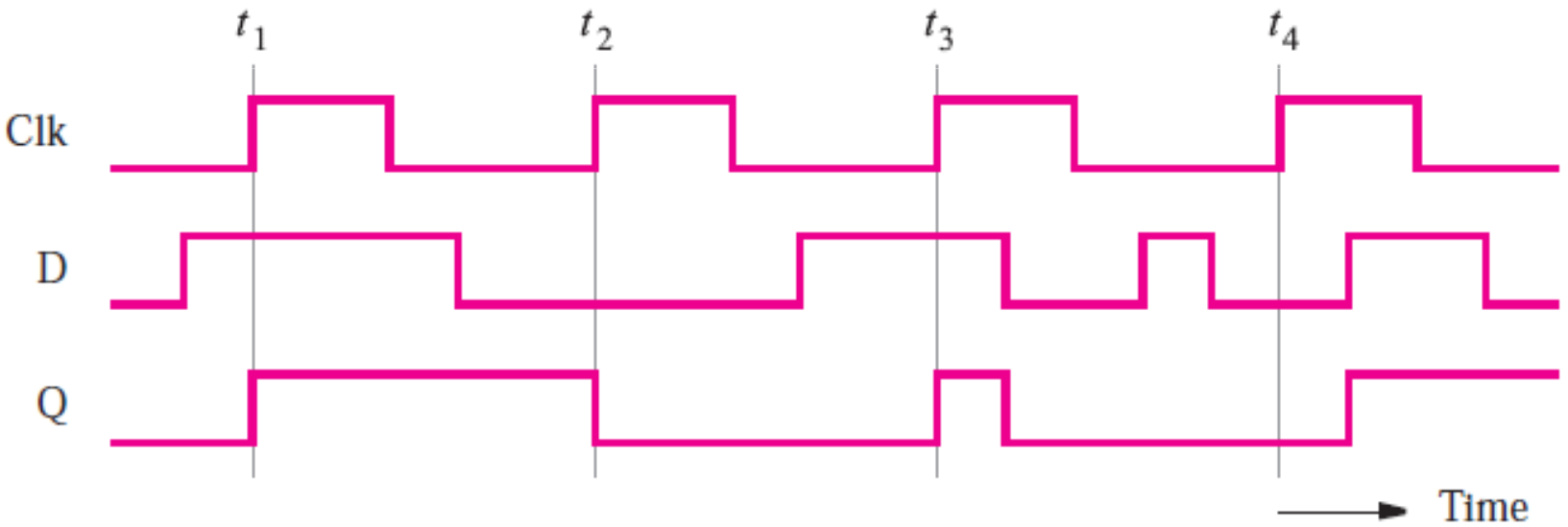


- To forbid S=R=1 input, only one input (D)
  - D connects to S
  - D' connects to R
- D stands for data
- Output follows the input when Clk = 1
- When Clk = 0, Q remains the same

GATED D LATCH

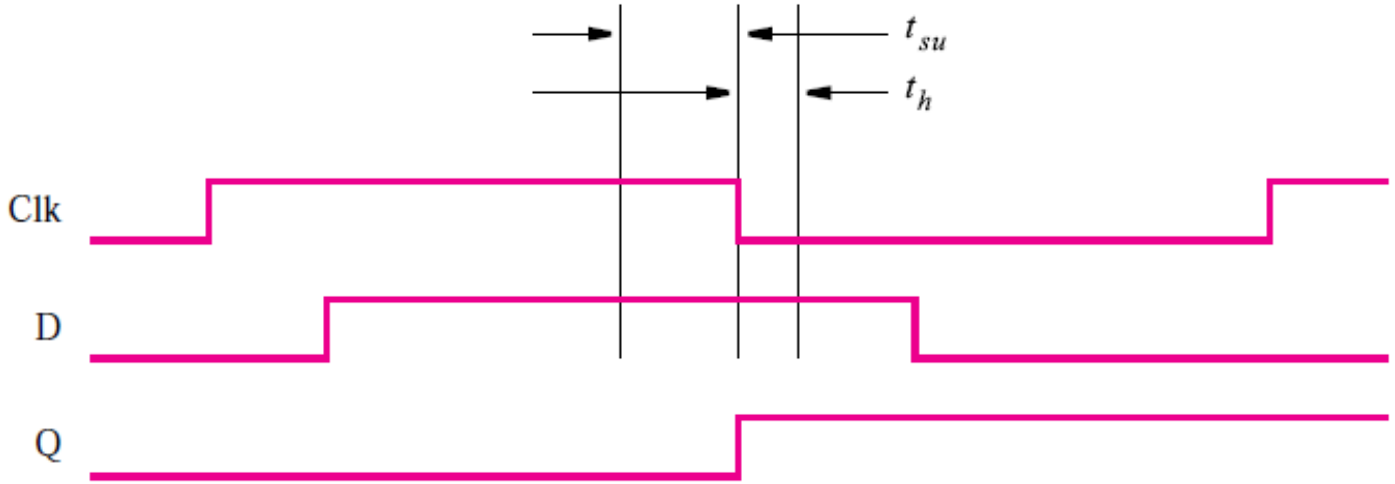
Clk	D	Q(t + 1)
0	x	Q(t)
1	0	0
1	1	1





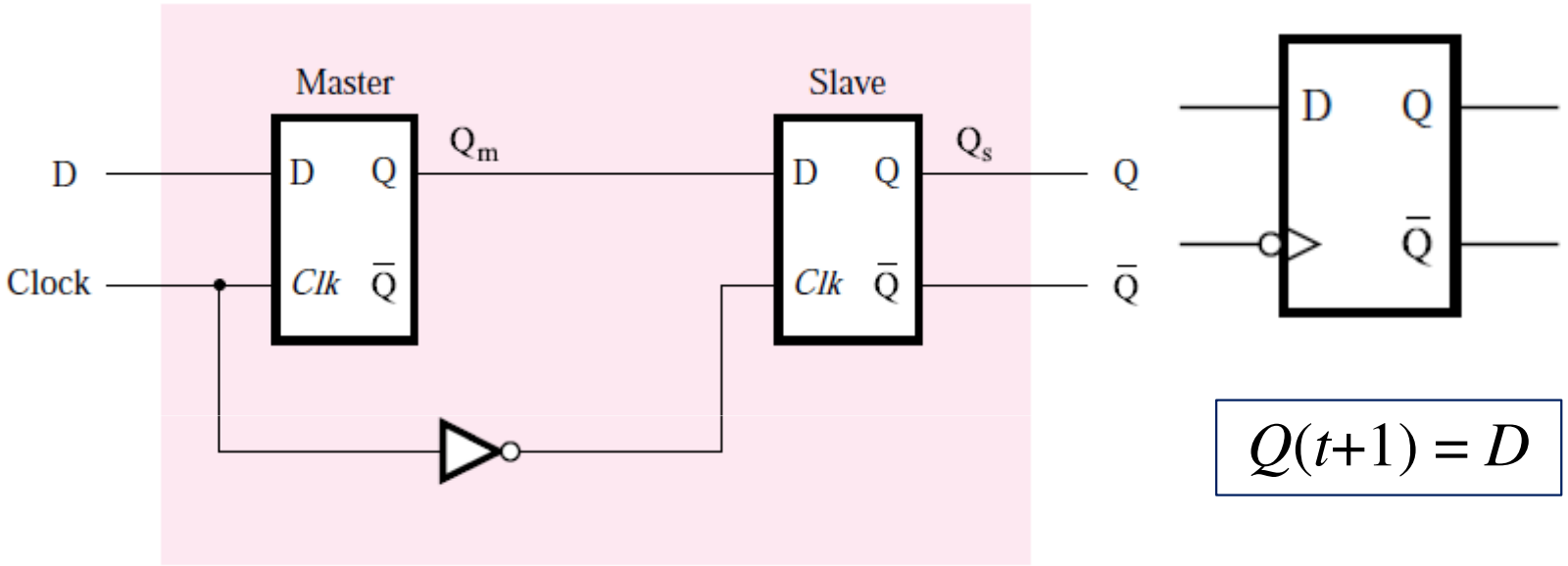
EFFECT OF PROPAGATION DELAY

- Condition : D signal is stable (not changing) at the time Clk changes (from 0 to 1, or from 1 to 0)
- Key quantities:
  - Setup time (t_{su}) : The minimum time D signal must be stable **prior** to the negative/positive edge of the Clk signal. (typical 0.3 ns)
  - Hold time (t_h) : The minimum time D signal must remain stable **after** the negative/positive edge of the Clk signal. (typical 0.2 ns)

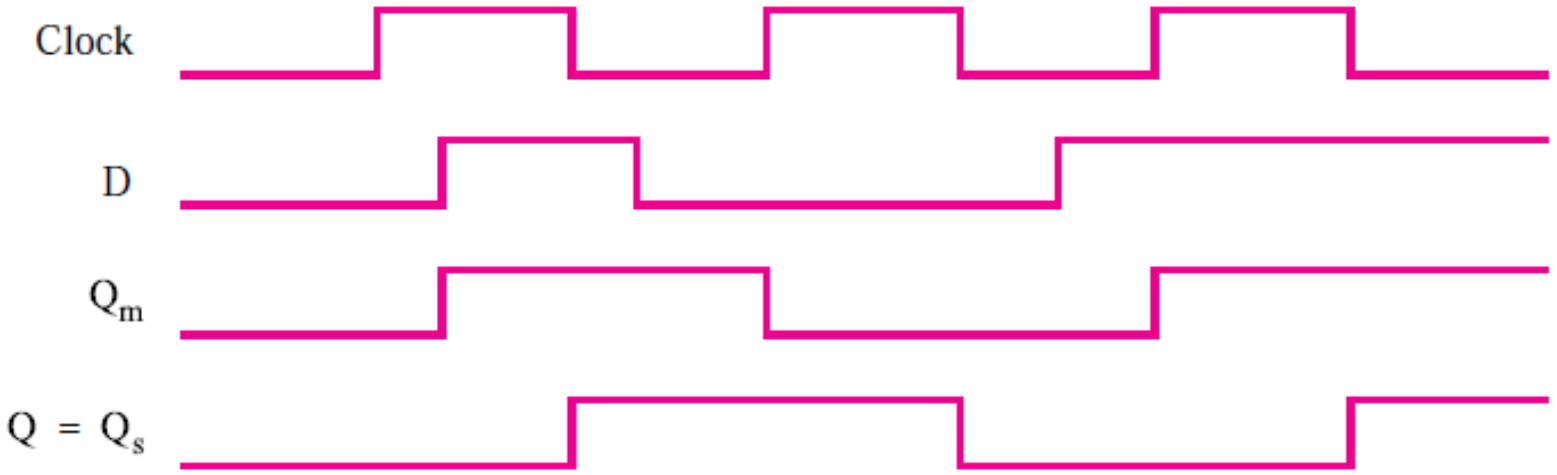


MASTER-SLAVE D FLIP-FLOP

- Q changes at the negative edge.

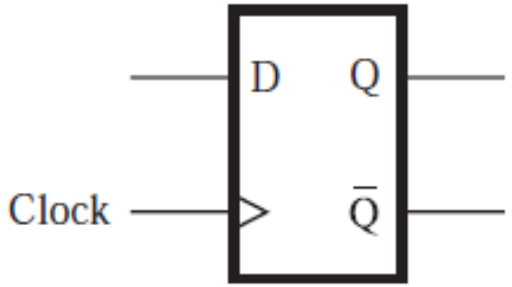


$$Q(t+1) = D$$

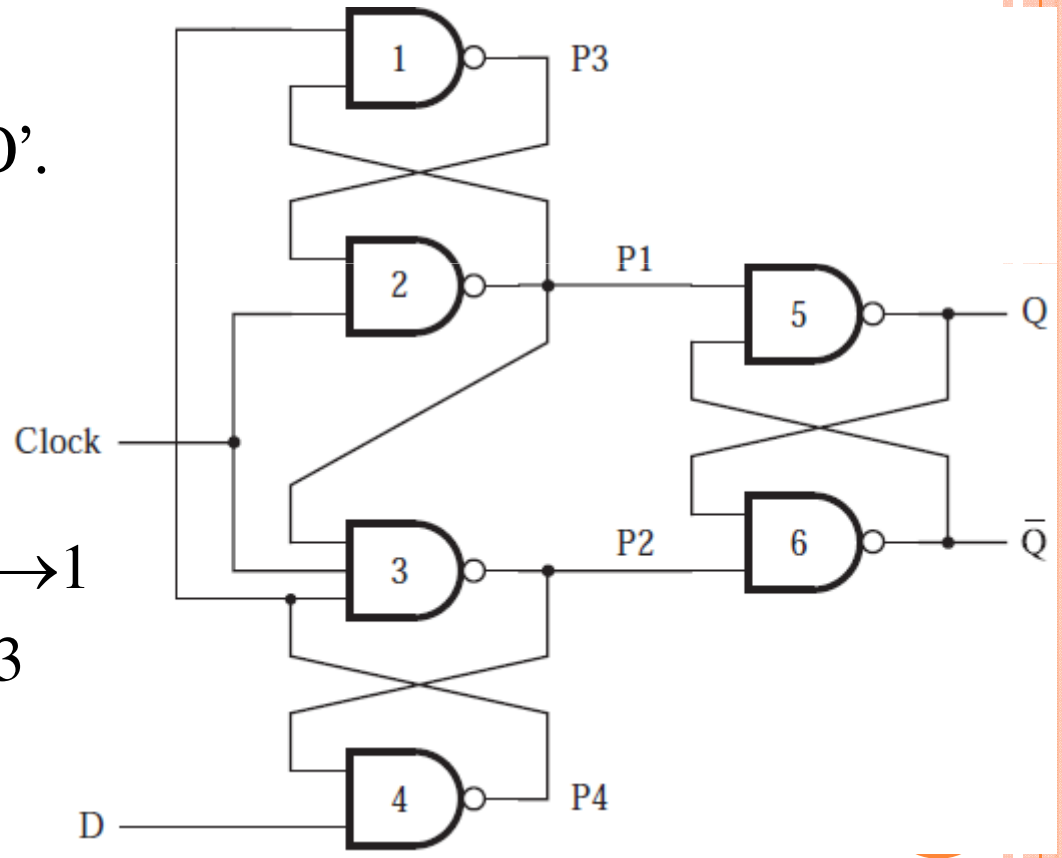


EDGE-TRIGGERED D FLIP-FLOP

- Q changes at the positive edge.
- Use fewer transistors.
- Operation:

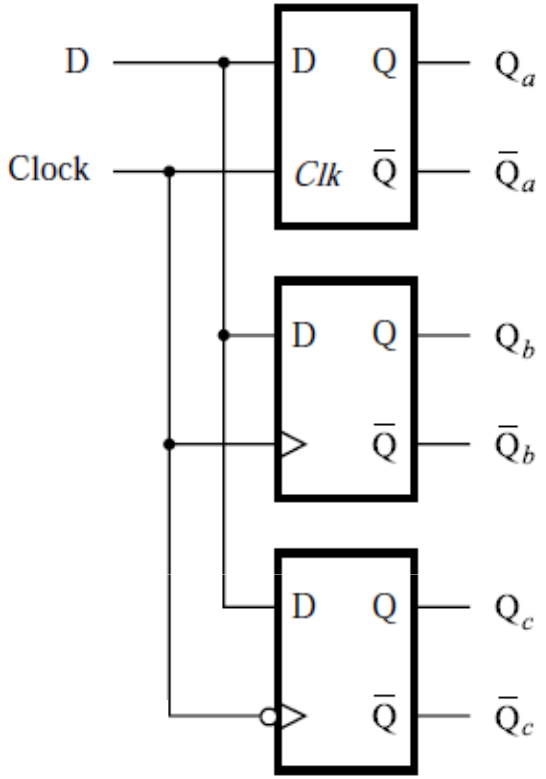


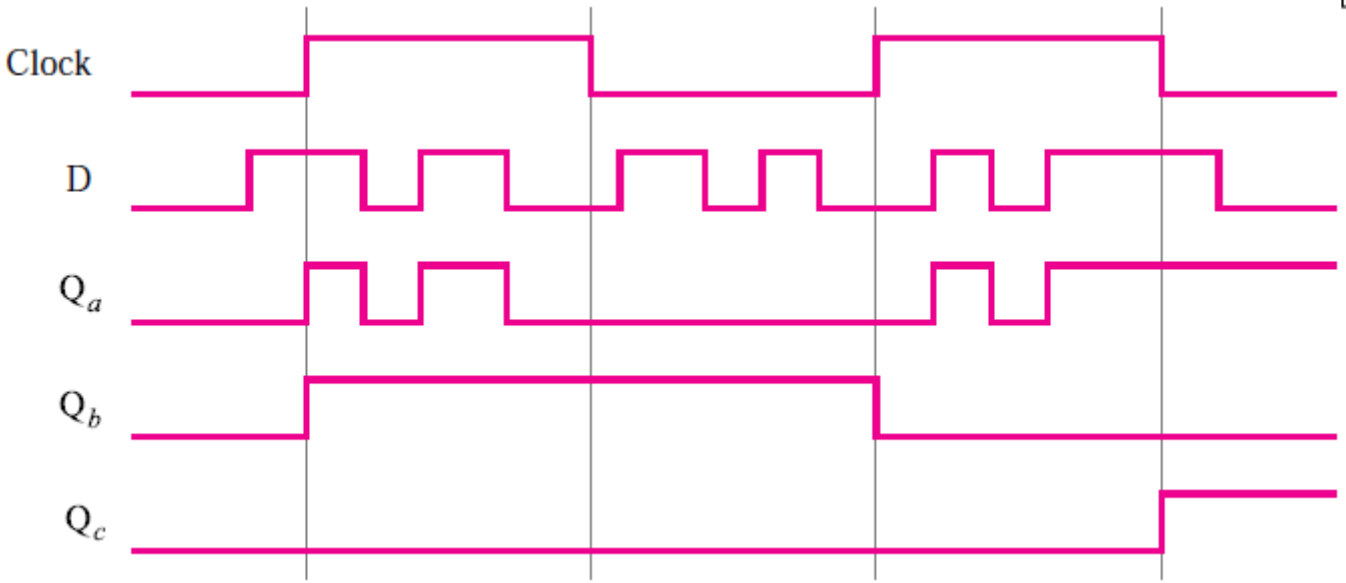
- Clock=0: P1=P2=1, Q unchanged, P3=D, P4=D'.
- Clock: 0→1: P3,P4 transmitted through G2,G3 → P1=D',P2=D →Q=D
- P3,P4 stable when Clock 0 →1
- t_{su} : Gate delay from D to P3 through G4, G1
- t_h : Gate delay through G3



D Latch VS D Flip-Flop

- D Latch : Level-sensitive
- D Flip-Flop : Edge-sensitive
  - Positive-edge Triggered (Q_b)
  - Negative-edge Triggered (Q_c)

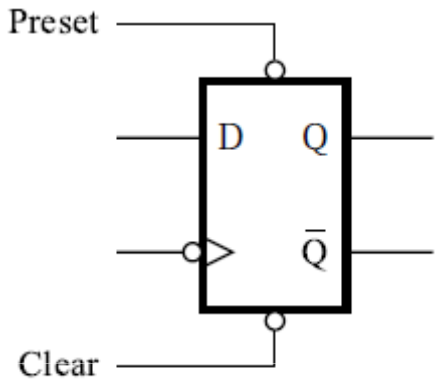


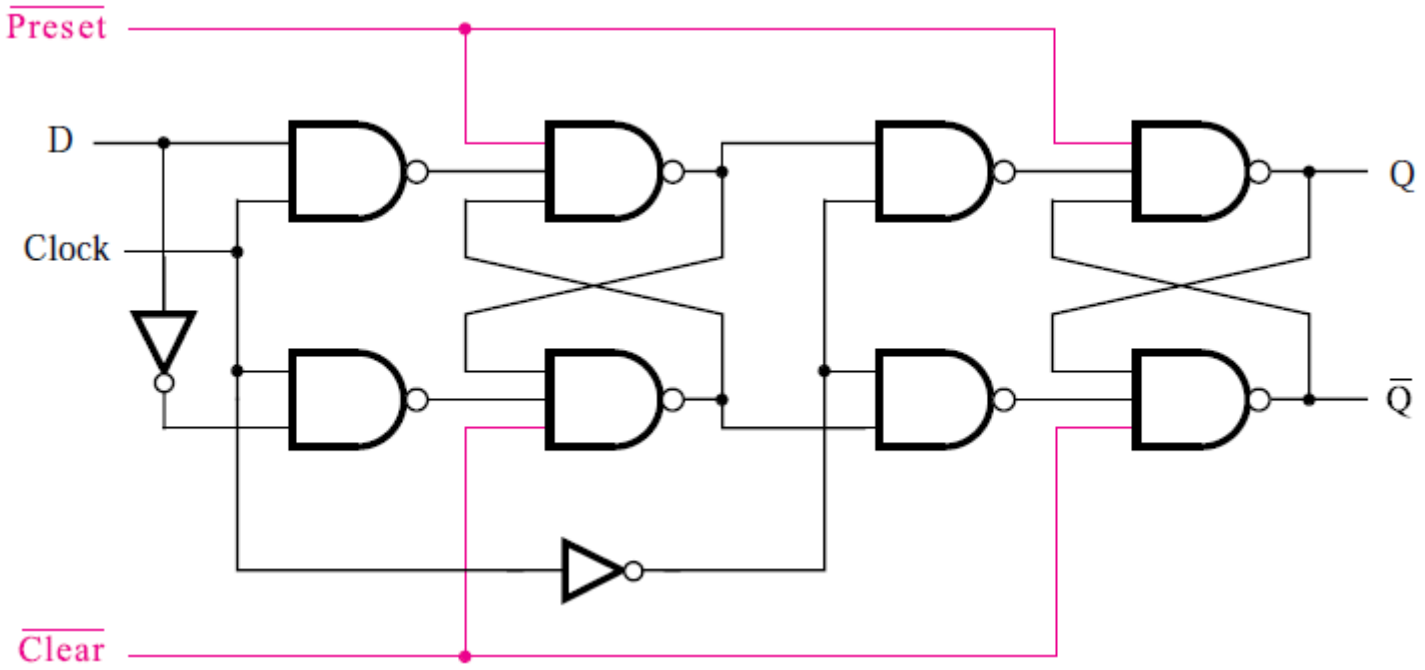


MASTER-SLAVE D FF WITH CLEAR, PRESET

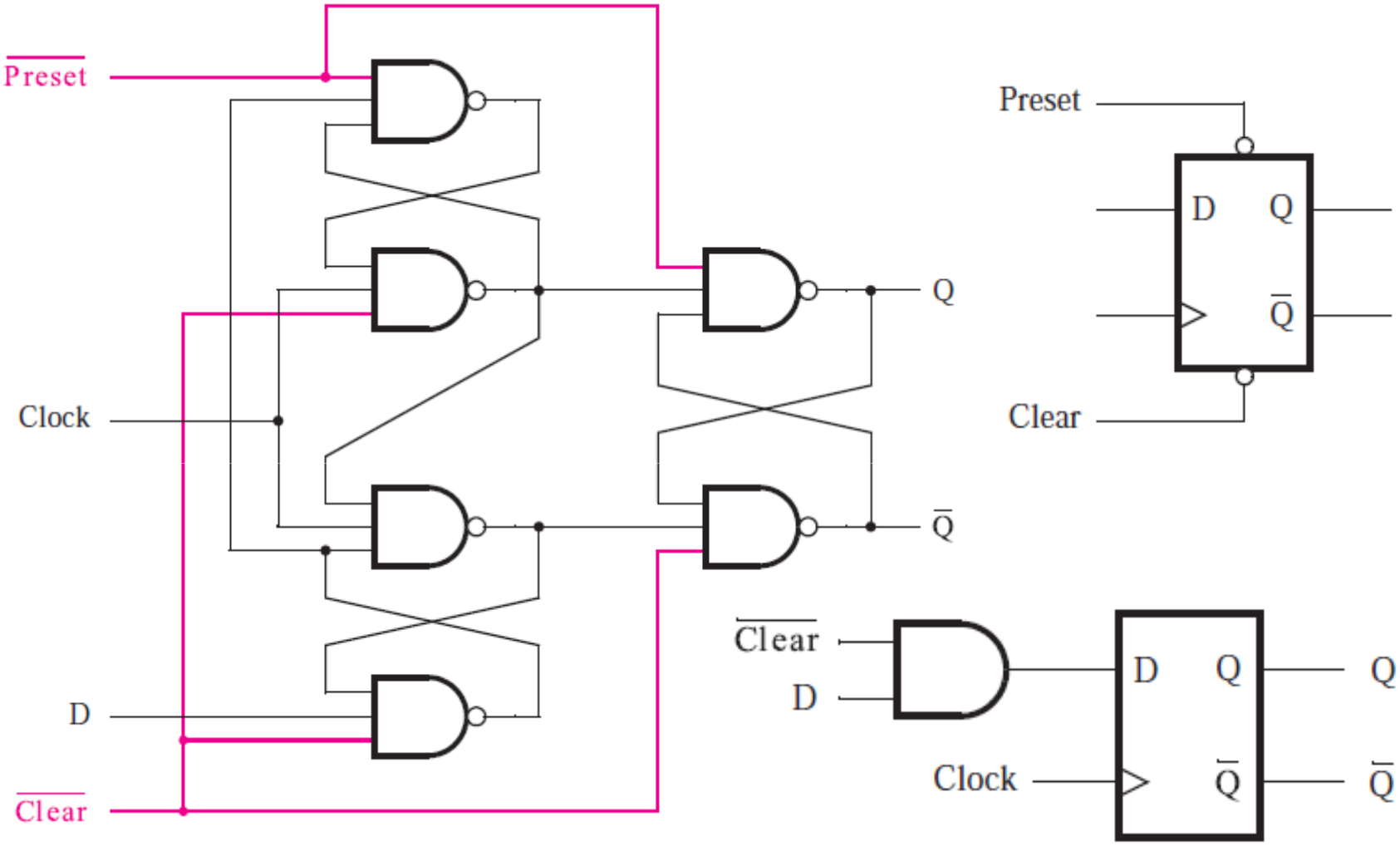
- Clear=1 → Q=0, Preset=0 → Q=1
- Clear=Preset=0 → Unstable
- "Asynchronous" clear

(unrelated to clock signal)





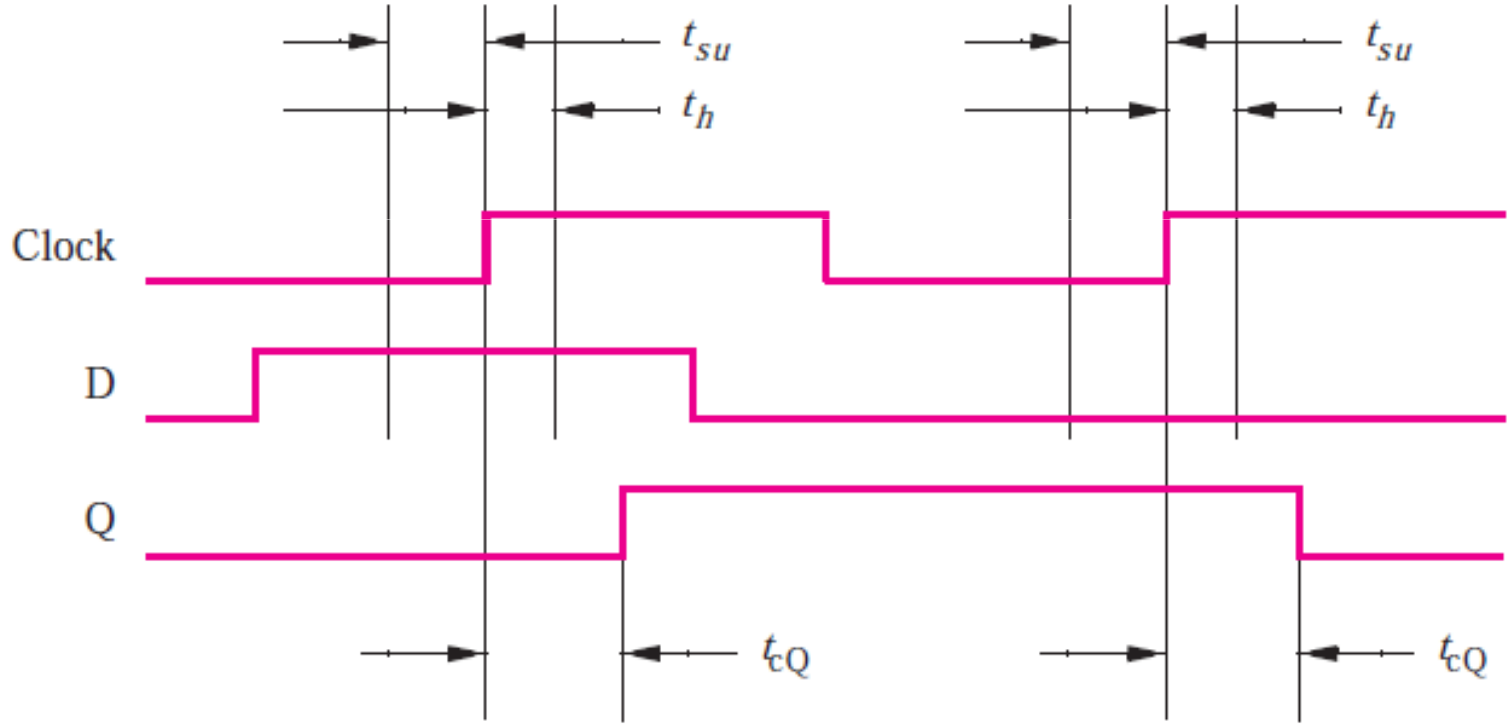
MASTER-SLAVE D FF WITH CLEAR, PRESET



Adding a synchronous clear

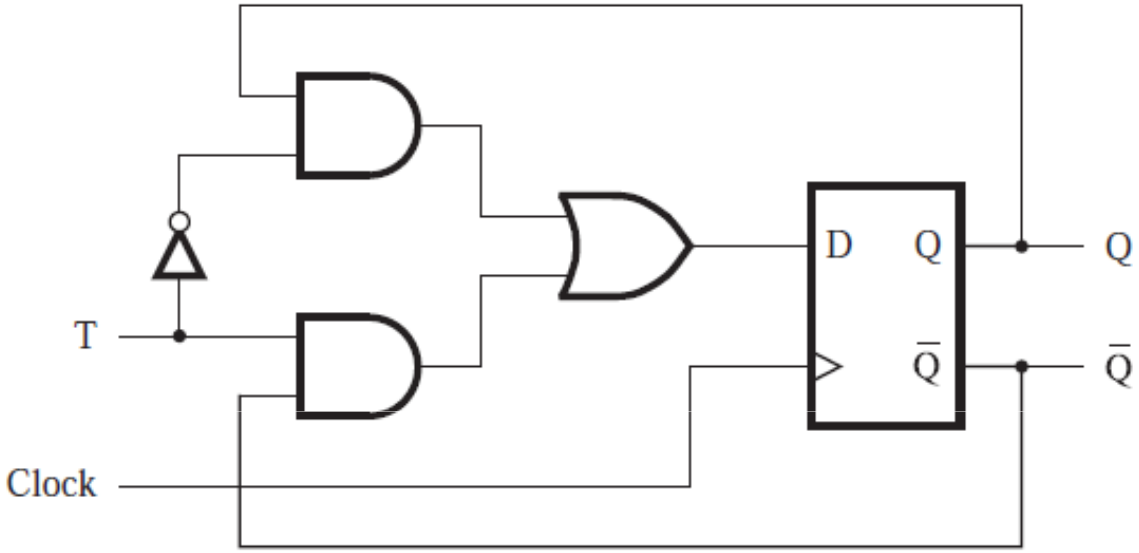
CLOCK-TO-Q PROPAGATION DELAY

- t_{cQ} : Time before Q changes after a positive clock edge.

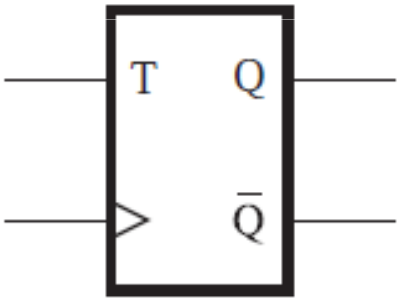


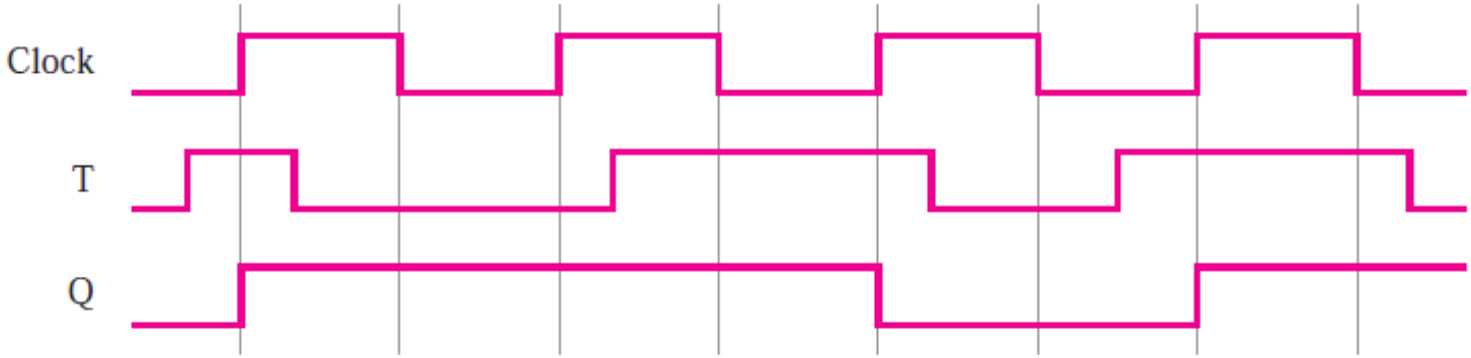
T FLIP-FLOP

- T="Toggle"→toggle when T=1,clock 0 →1



T	Q(t + 1)
0	Q(t)
1	$\bar{Q}(t)$



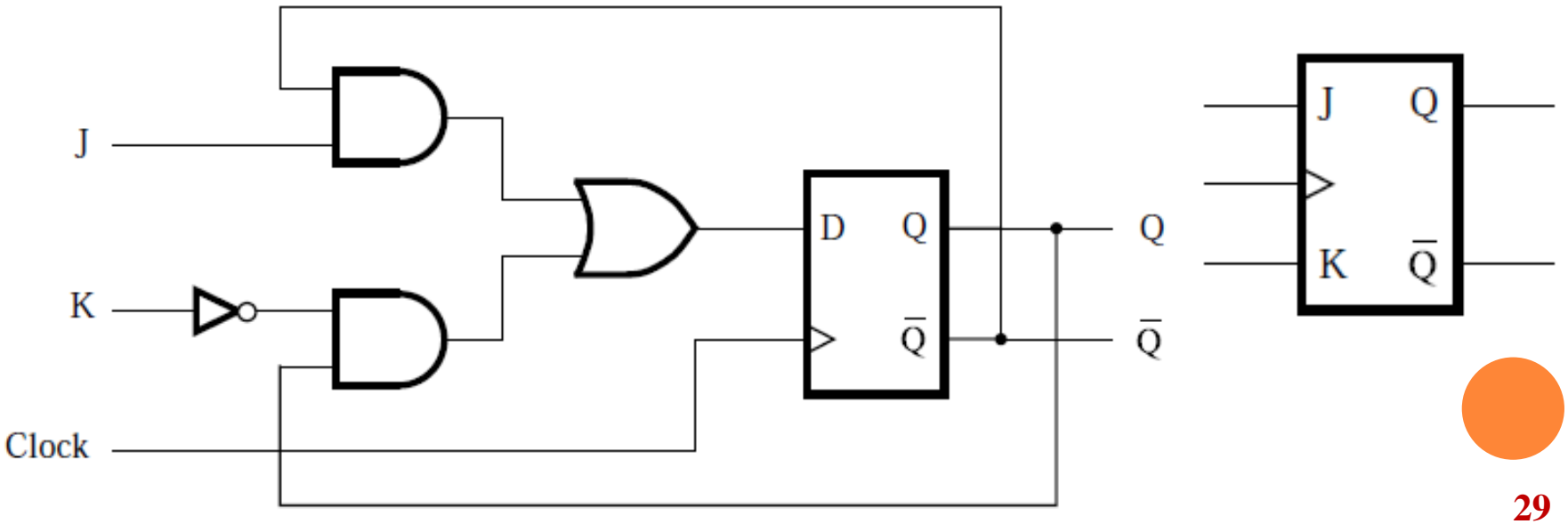


JK Flip-Flop

- Similar to SR FF with J=S, K=R
- J=K=1 → toggle
- J=K → T-FF

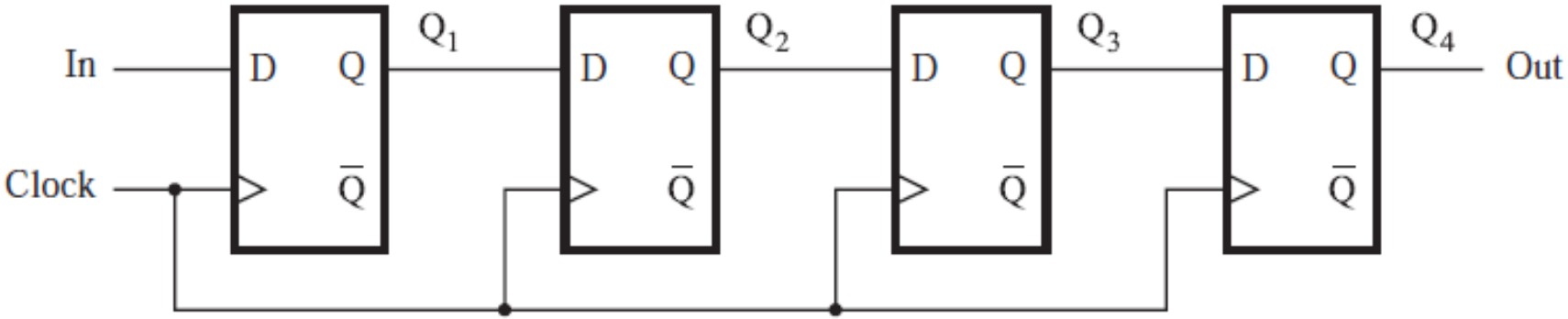
J	K	Q(t + 1)
0	0	Q(t)
0	1	0
1	0	1
1	1	$\bar{Q}(t)$

$$Q(t+1)=JQ'+K'Q$$



Shift Register

- Register = a set of *n* flip-flops

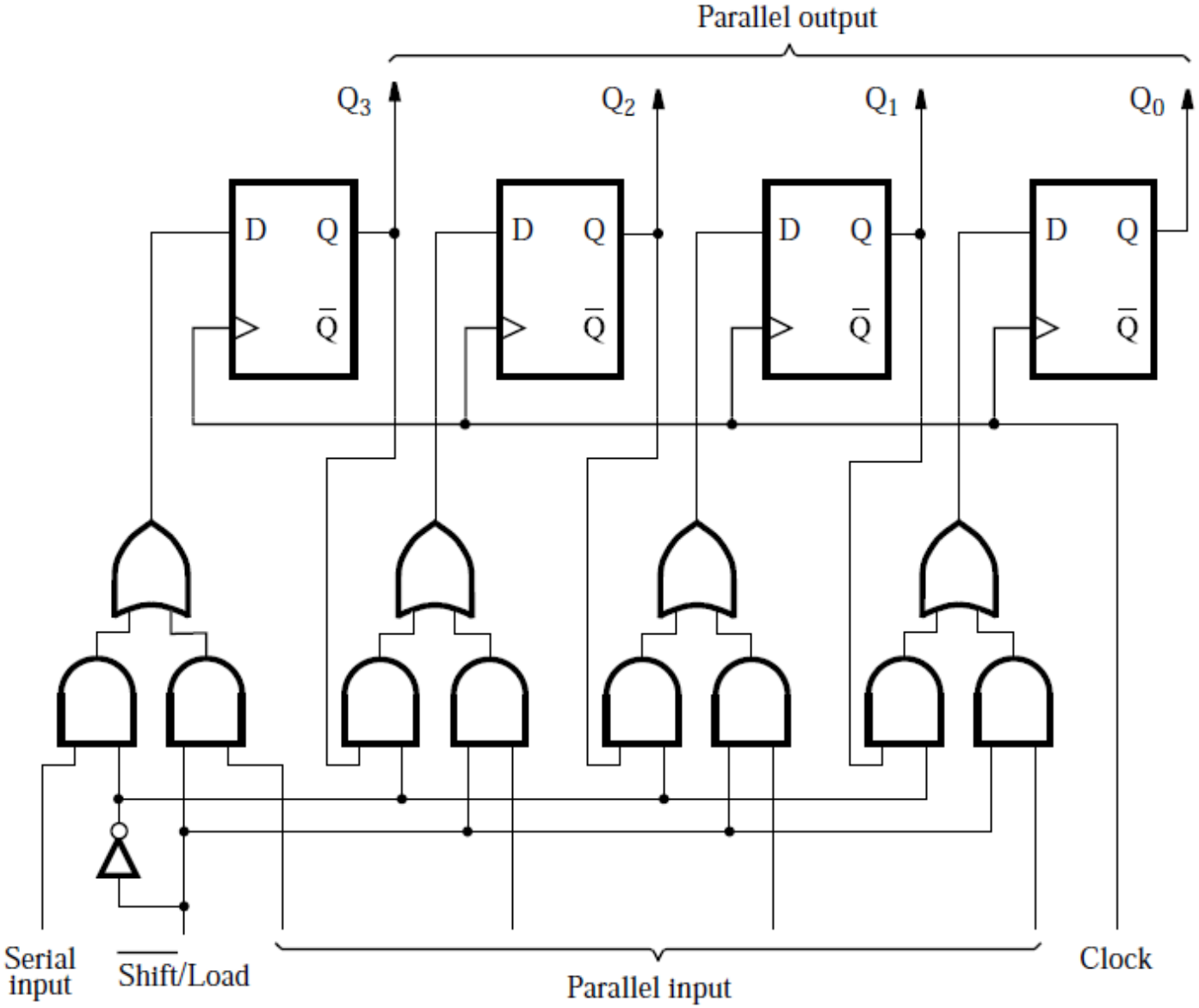


	In	Q_1	Q_2	Q_3	Q_4 = Out
t_0	1	0	0	0	0
t_1	0	1	0	0	0
t_2	1	0	1	0	0
t_3	1	1	0	1	0
t_4	1	1	1	0	1
t_5	0	1	1	1	0
t_6	0	0	1	1	1
t_7	0	0	0	1	1

Output shifts to the right at positive-edge

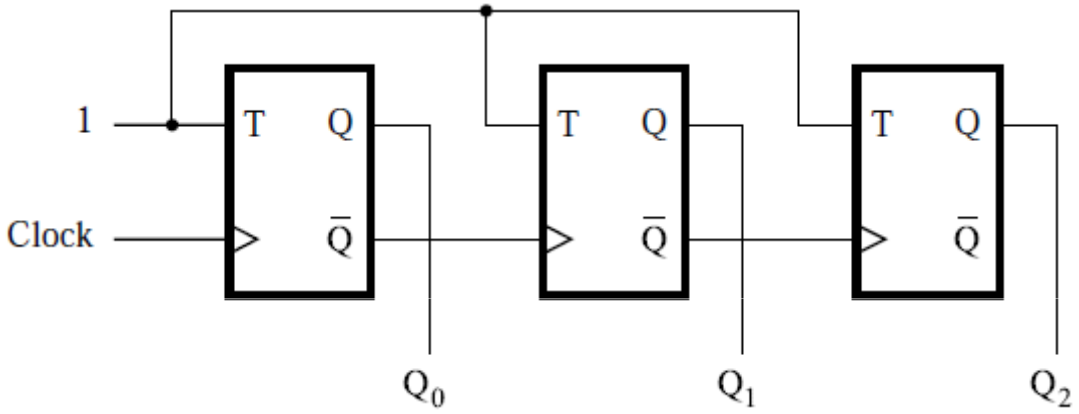
PARALLEL-ACCESS SHIFT REGISTER

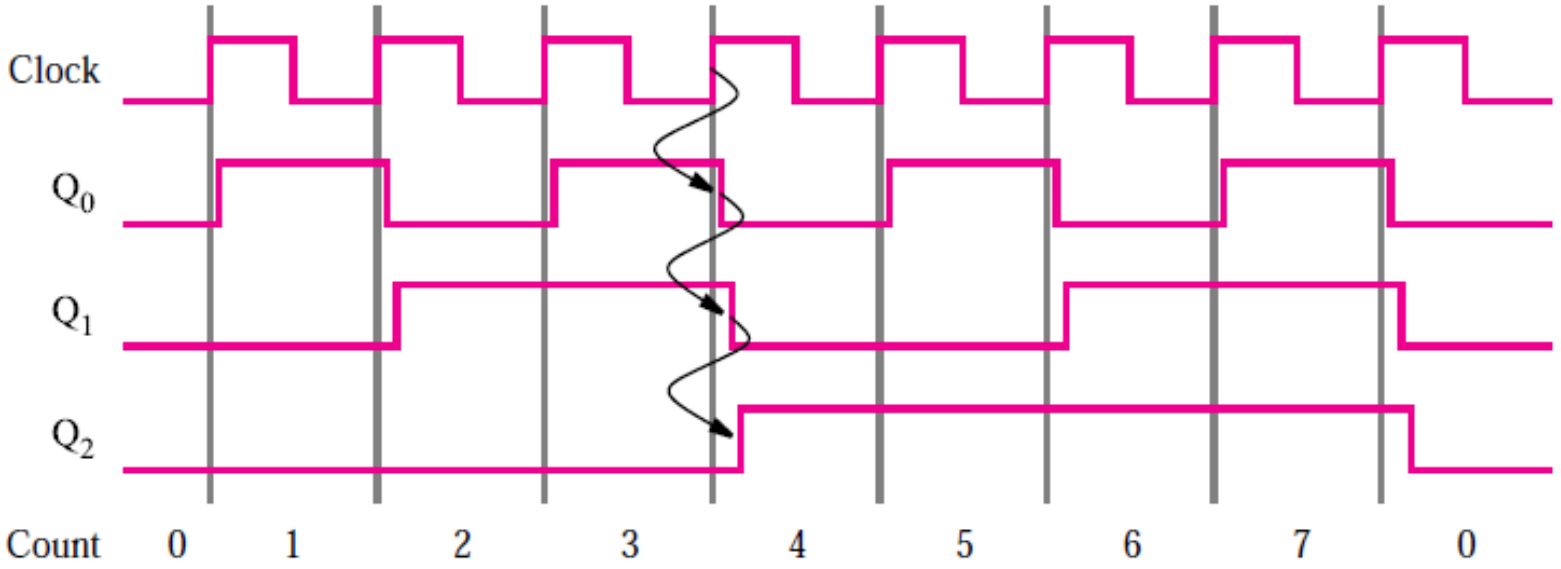
- $\overline{\text{shift}}$/load=0→shift register, $\overline{\text{shift}}$/load=1→parallel access



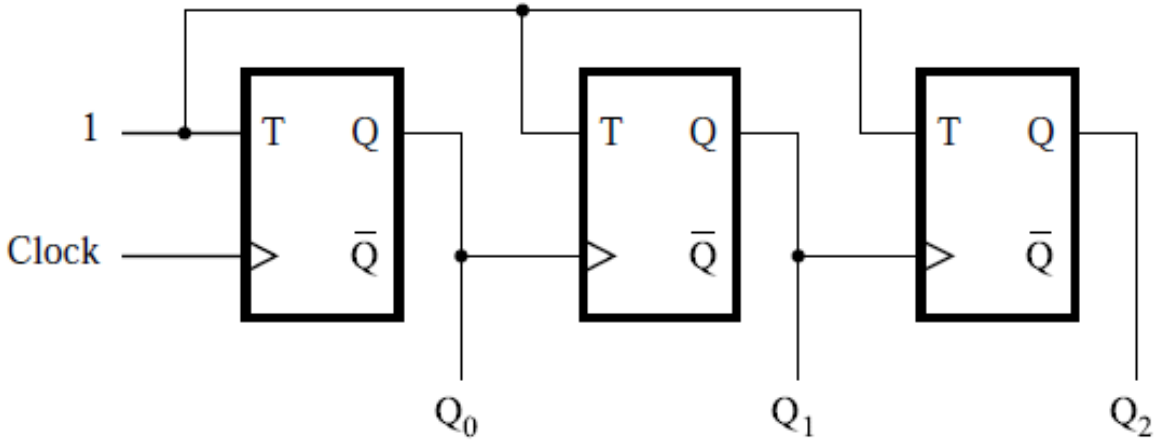
Asynchronous Counter (Up-counter)

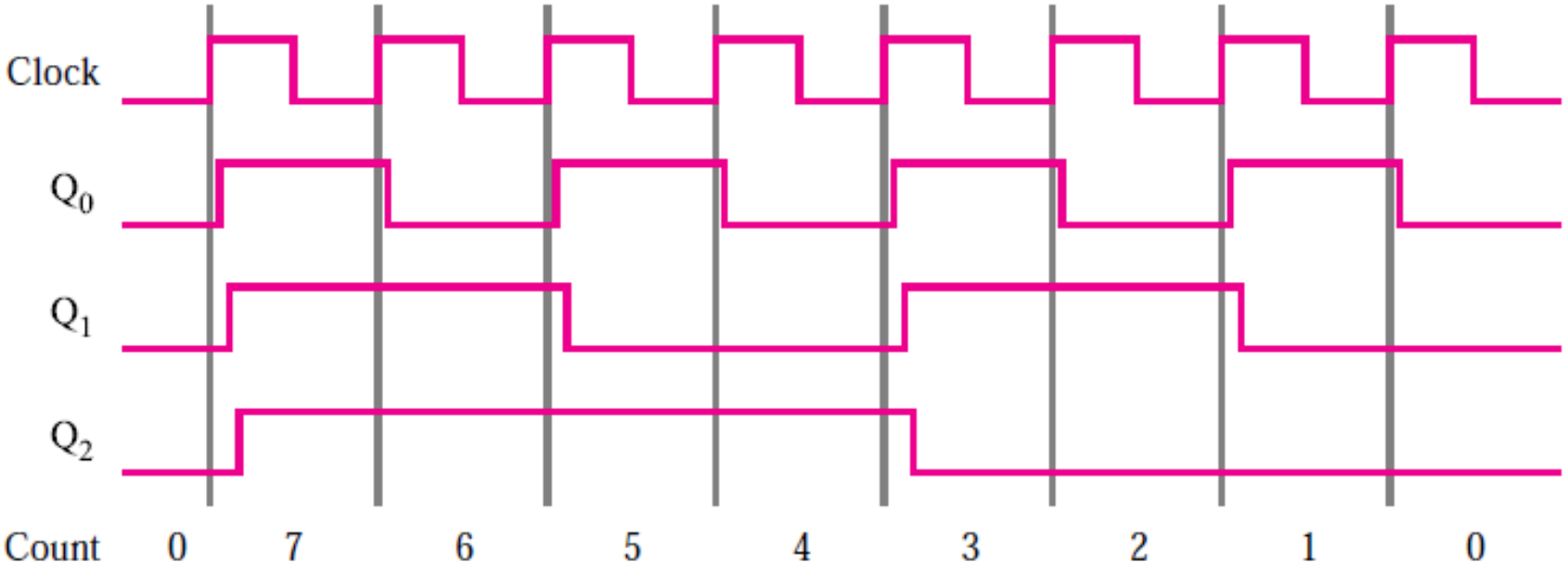
- Q1 changes after Q0, Q2 changes after Q1
  →similar to ripple-carry adder
- "Ripple" Counter
- Modulo-8, i.e., 0,1,…,7,0,…





Asynchronous Counter (Down-counter)





SYNCHRONOUS COUNTER

- Asynchronous counter : delay -> Not fast
- Synchronous counter can work faster.
- 3-bit up-counter :

table 7.1:

- For n-bit counter:

$T_0=1$

$T_1=Q_0$

$T_2=Q_0Q_1$

$T_3=Q_0Q_1Q_2$

⋮

$T_n=Q_0Q_1Q_2 \ldots Q_{n-1}$

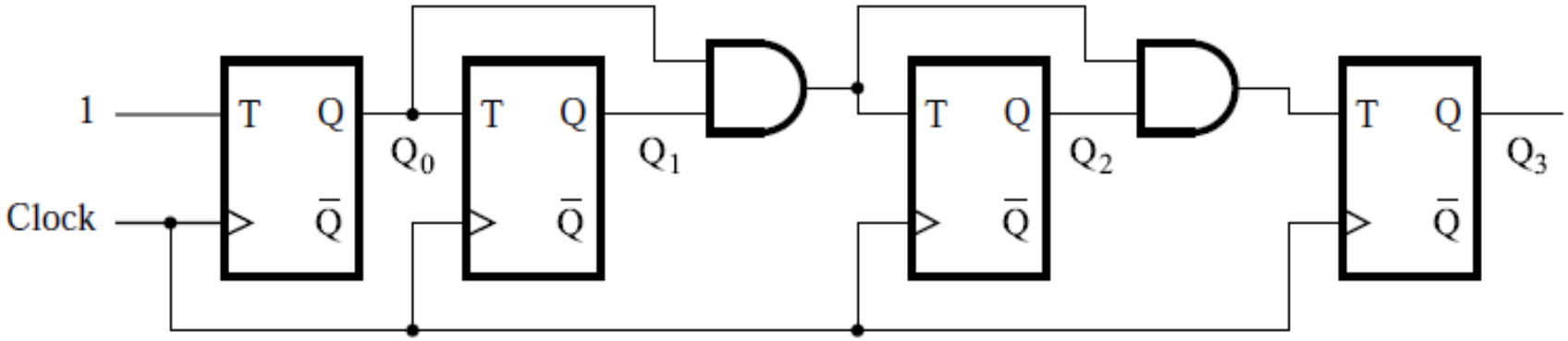
Table 7.1 Derivation of the synchronous up-counter.

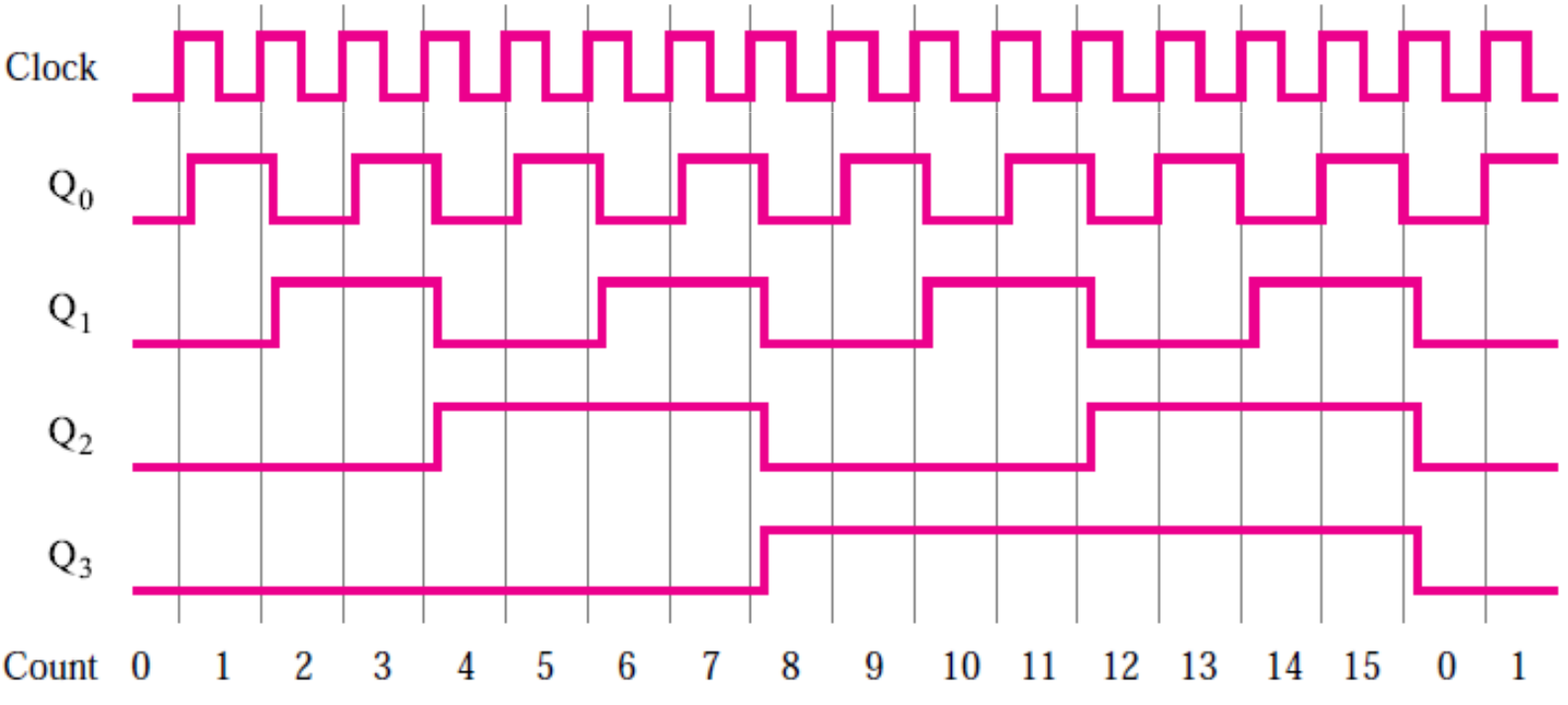
Clock cycle	Q_2	Q_1	Q_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8	0	0	0

Q_1 changes

Q_2 changes

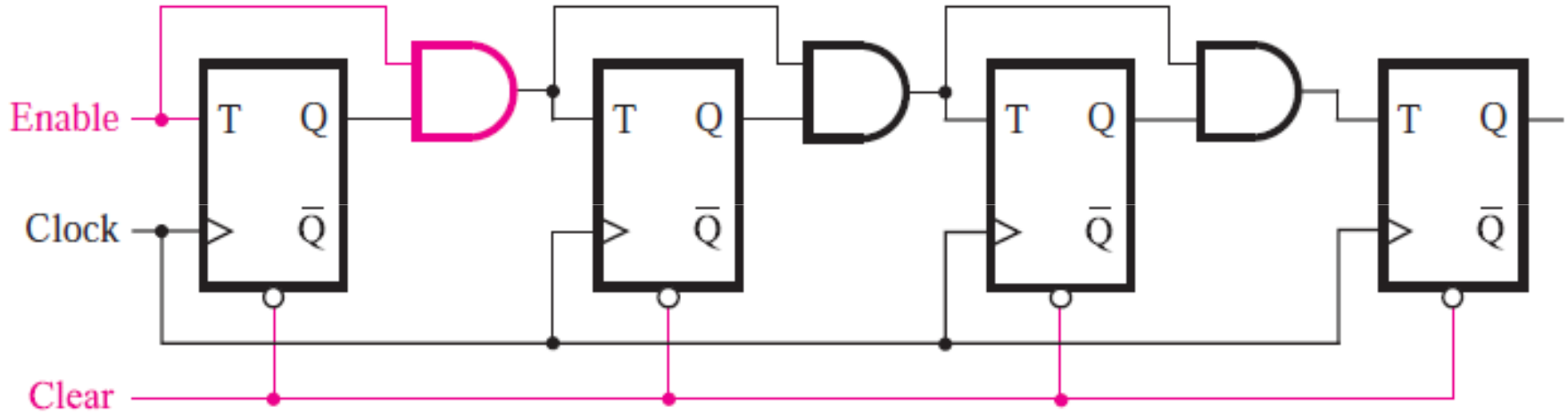
SYNCHRONOUS COUNTER (UP-COUNTER)





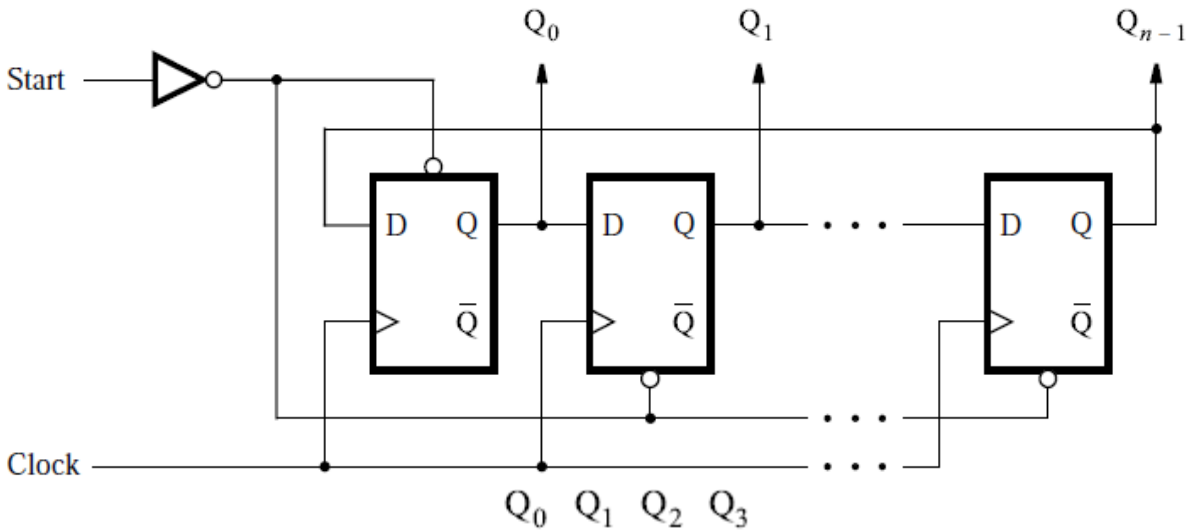
COUNTER WITH ENABLE & CLEAR

- Enable=0→T=0, Enable=1→Counter
- Clear=0→Reset ($Q_0=Q_1=Q_2=0$)

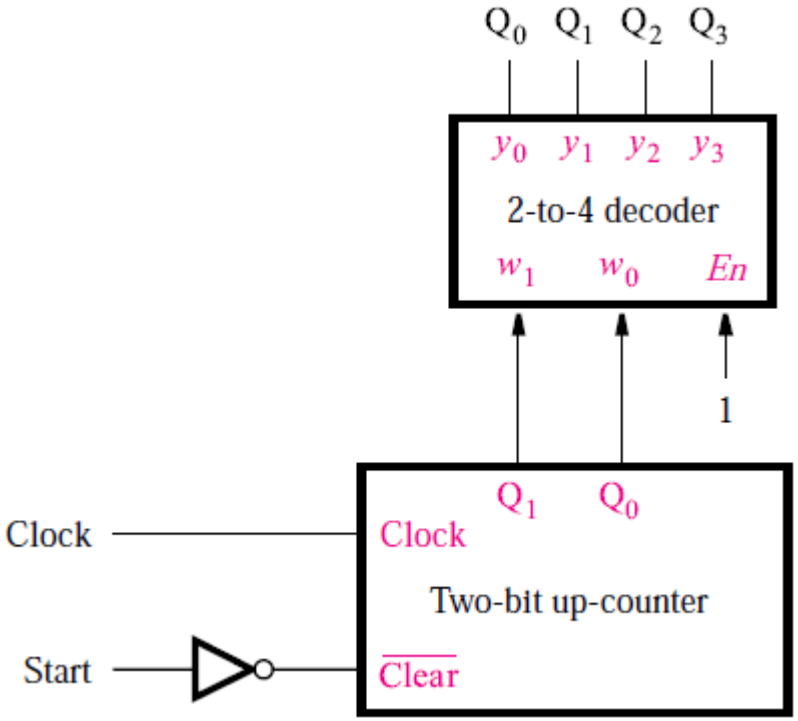


RING COUNTER

1000→0100→0010→0001→1000→… “One-Hot” Code

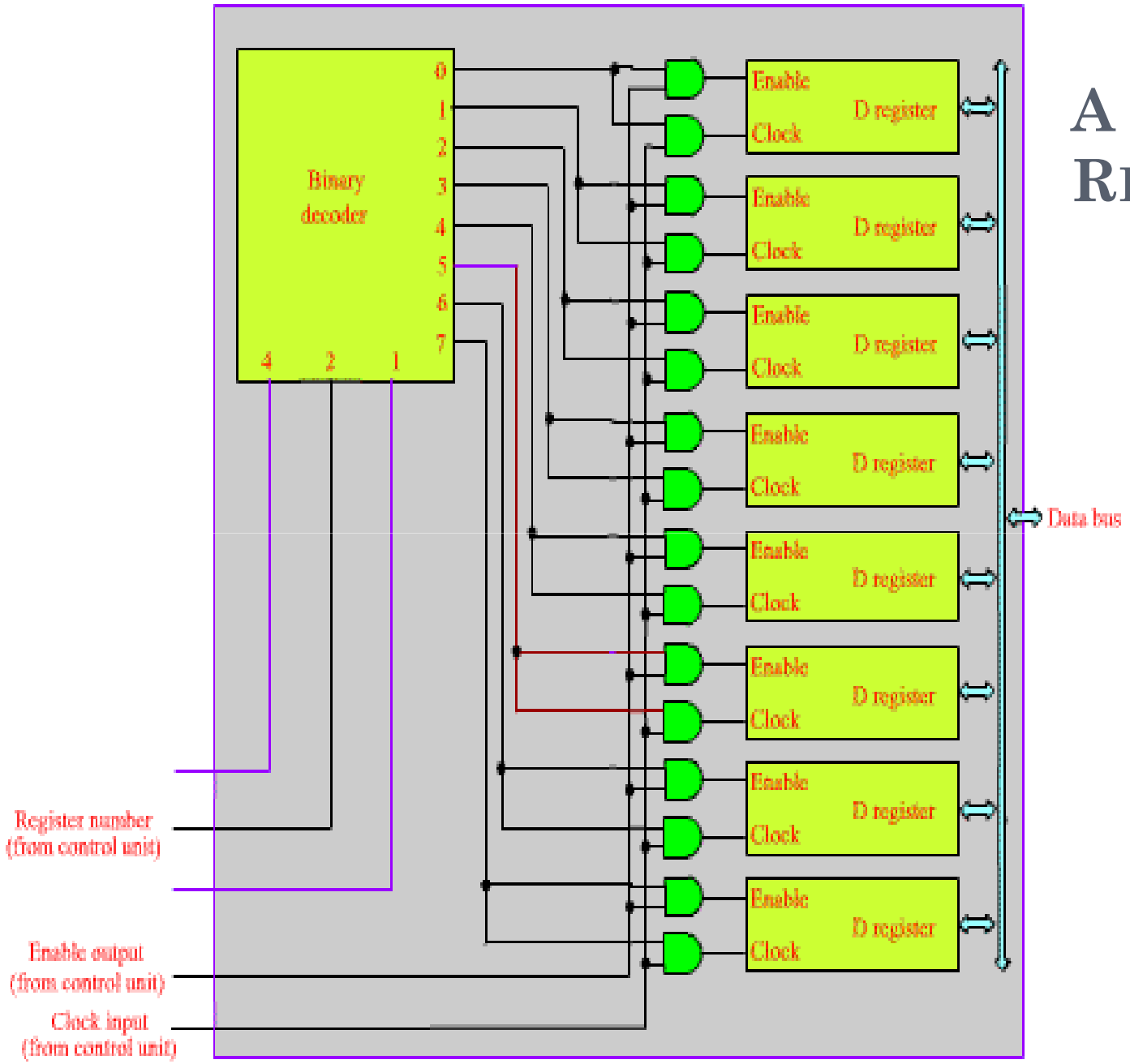


First set Start to 1, then change to 0



Start=1, Q_1Q_0=00
Up-counter output :
00, 01, 10, 11, 00, …
Decoder output :
1000,0100,0010,0001,1000,
…

A Register File

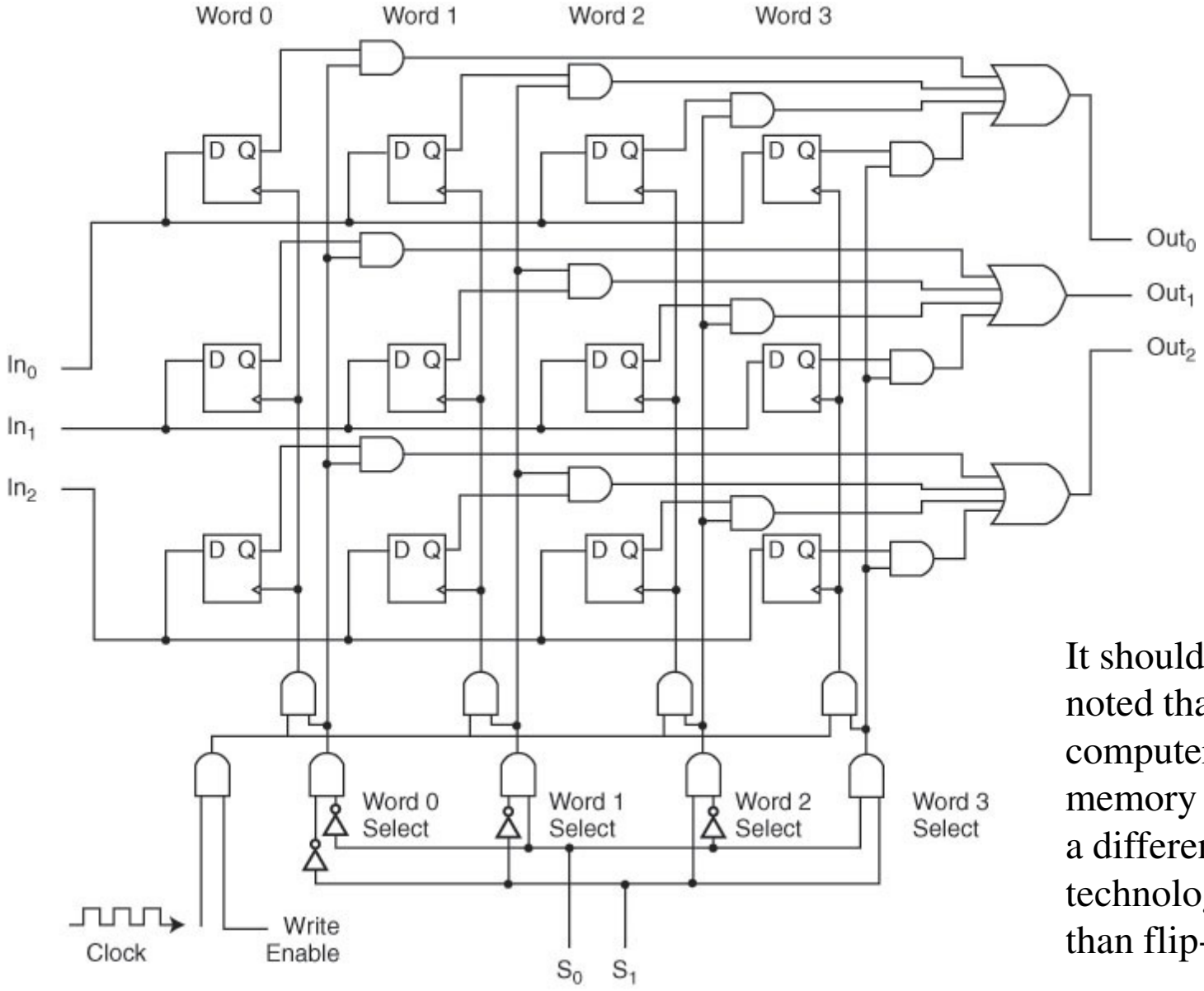


The decoder accepts a 3-bit register number from the control unit

This along with the system clock selects the register

The data bus is used for both input and output to the selected register

A 4x3 Memory



This is a collection of flip-flops that can store 4 items (each consisting of 3 bits)

The two bit selector S0 S1 chooses which of the 4 items is desired

It should be noted that computer memory uses a different technology than flip-flops

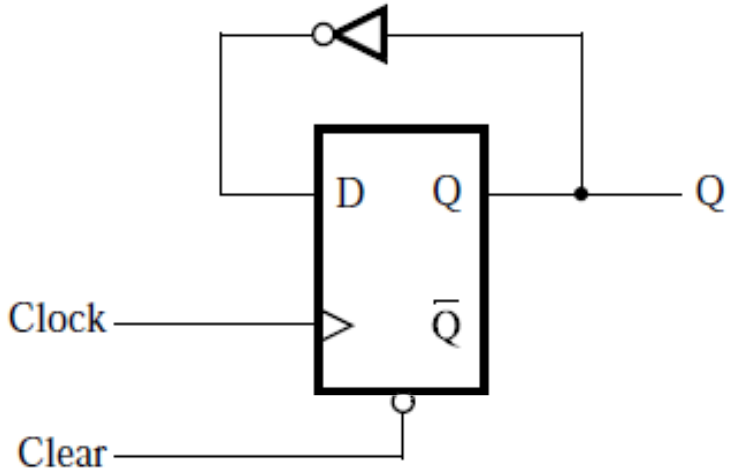
TIMING ANALYSIS OF FLIP-FLOP

❖ Want to find the **maximum clock frequency** $f_{max} = 1/T_{min}$ (T_{min} : minimum clock period)

❖ Consider the circuit on the right

❖ Operation :

1. D is loaded with positive clock edge.
2. D propagates to Q
3. Q propagates through NOT.



❖ T_{min} is given by $T_{min} = t_{cQ} + t_{NOT} + t_{su}$

❖ Example : t_{su} 0.6 ns, t_h 0.4 ns, 0.8 ns $\le t_{cQ} \le$ 1 ns, and assume t_{NOT}=1+0.1k (k : Number of input to the gate)

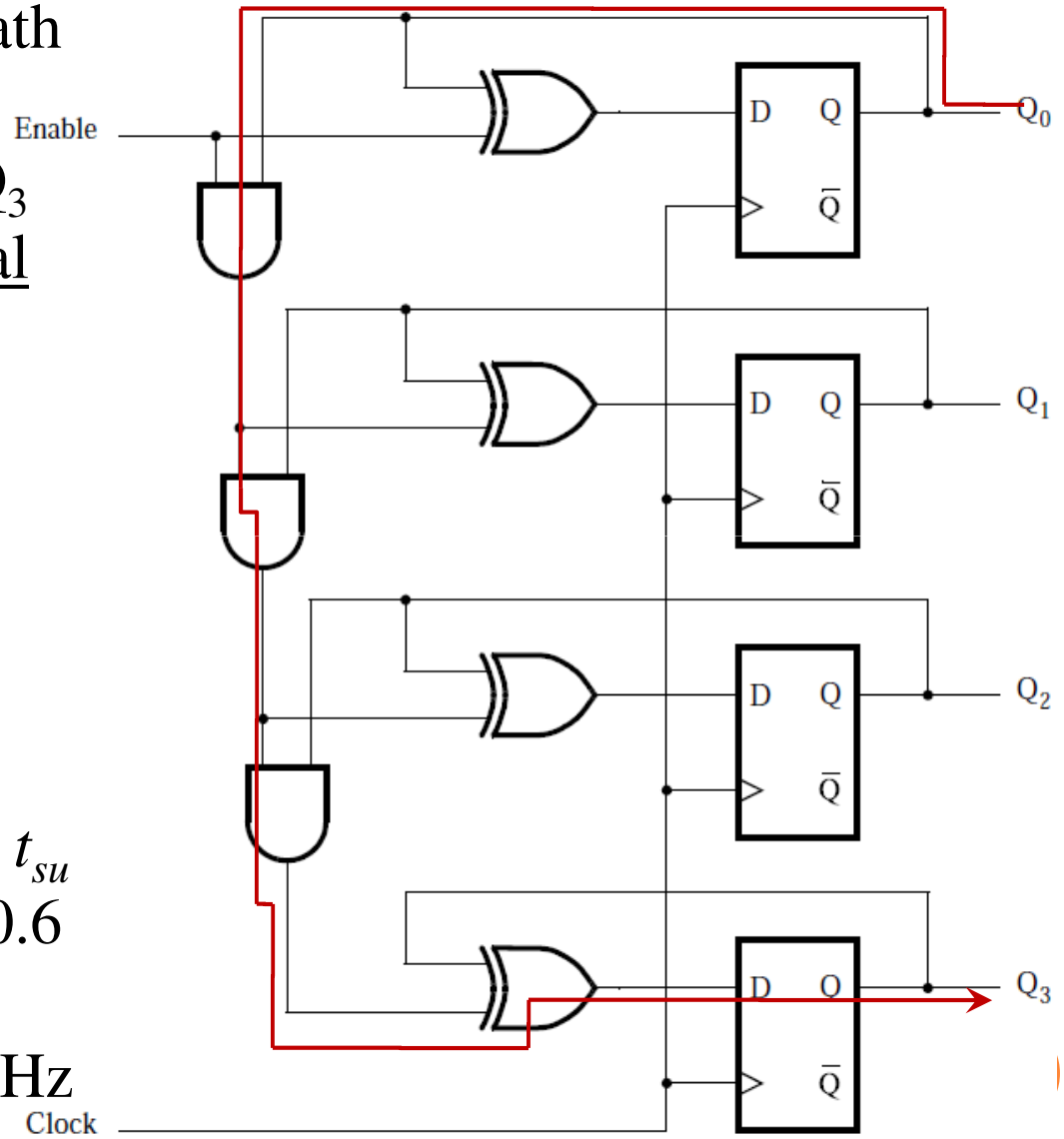
Thus, $T_{min} = 1 + 1.1 + 0.6 = 2.7$ ns.

And, $f_{max} = 1/2.7$(ns) = 370.37 MHz

TIMING ANALYSIS OF A 4-BIT COUNTER

- First find the longest path (critical path).
- In this example, $Q_0 \rightarrow Q_3$
- The delay of this critical path includes
  - clock-to-Q of Q_0
  - delay through 3 AND's
  - delay through 1 XOR
- Also need to take into account setup time of Q_3.
- Thus,

$$T_{min} = t_{cQ} + 3t_{AND} + t_{XOR} + t_{su}$$
$$= 1 + 3(1.2) + 1.2 + 0.6$$
$$= 6.4 \text{ ns}$$
$$\therefore f_{max} = 1/6.4\text{ns} = 156.25 \text{ MHz}$$



SUMMARY

- In a sequential circuit, outputs depends on inputs and previous inputs
  - Previous inputs are stored as binary information into memory
  - The stored information at any time defines a state
  - Similarly, next state depends on inputs and present state
- Two types of sequential circuits: Synchronous and Asynchronous
- Two types of Memory elements: Latches and Flip-Flops.
- Flip-flops are built with latches
- A flip-flop is described using characteristic table/equation
- Flips-flops can have direct asynchronous inputs