Printout

Monday, October 31, 2022 3:59 PM

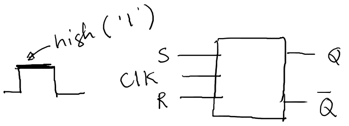
COE328
Digital Systems

Lecture 13

Dr. Shazzat Hossain

Latch

S-R , D

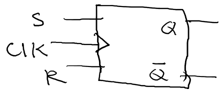


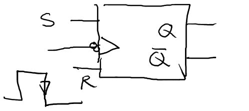
set Reset

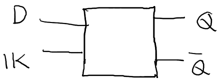
ClK	S	R	Q_{n+1}
0	×	×	NC Q_n
1	0	0	NC
1	0	1	0
1	1	0	1
1	1	1	Invalid

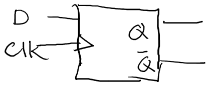
Flip-Flop

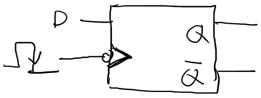
S-R, D











Clk	D	Q
0	×	NC ← No change (Memory)
1	0	0
1	1	1

Next state

$Q_{n+1} = Q_n$ ← Present state

1 — ↑ → ↓ ← negative

↑ Positive

Clk	D	Q
0	×	NC
↑	0	0
↑	1	1

Case I

T = 0

Next o/p Present o/p

$D = Q_{n+1} = \bar{0} \cdot Q_n + 0 \cdot \bar{Q}_n = Q_n + 0 = Q_n$

Truth Table

Clk | T | Q

/ Next o/p

Truth Table

clk	T	Q
0	x	NC
1	0	NC
1	1	Toggle $\bar{Q}_n = Q_{n+1}$

Next o/p

$D = Q_{n+1} = \bar{T}.Q + T.\bar{Q}$

Next

$D = Q_{n+1} = \bar{0} \cdot Q_n + 0 \cdot \bar{Q}_n = Q_n + 0 = Q_n$

Case II : T = 1

Toggle

$D = Q_{n+1} = \bar{1} \cdot Q_n + 1 \cdot \bar{Q}_n$
$= 0 + \bar{Q}_n = \bar{Q}_n$

T Flip-Flop

Logic function for $D = \bar{T}Q + T\bar{Q}$

Case 1:

T=0; Q(t)=1; $\bar{Q}(t) = 0$; D=1.1+0.0=1 ; Q(t+1)=1 i.e. Q(t+1)=Q(t)

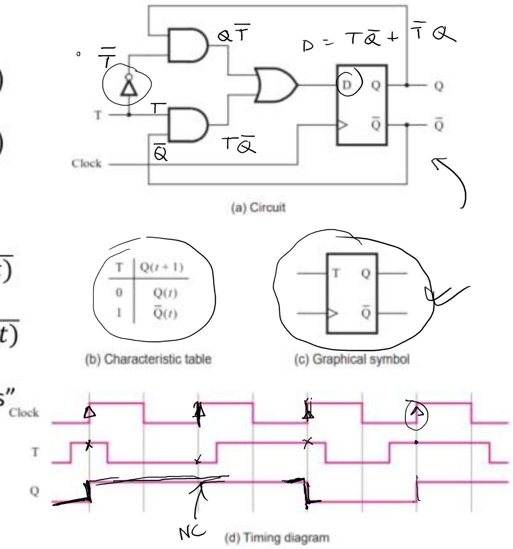
T=0; Q(t)=0; $\bar{Q}(t) = 1$; D=0.0+0.1=0 ; Q(t+1)=0 i.e. Q(t+1)=Q(t)

Case 2:

T=1; Q(t)=1; $\bar{Q}(t) = 0$; D=0.1+1.0=0 ; Q(t+1)=0 i.e. Q(t+1)=$\overline{Q(t)}$

T=1; Q(t)=0; $\bar{Q}(t) = 1$; D=0.0+1.1=1 ; Q(t+1)=1 i.e. Q(t+1)= $\overline{Q(t)}$

T flip-flop derives from the behavior of the circuit, which “toggles” its state when T = 1. The toggle feature makes the T flip-flop a useful element for building counter circuits.



(a) Circuit

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

(b) Characteristic table

(c) Graphical symbol

(d) Timing diagram

Figure 7.16 T flip-flop.

$J = T$

J-K Flip-flop

Two inputs J and K.

$= 1 \cdot \bar{Q} + \bar{0} \cdot Q = \bar{Q} + Q = 1$

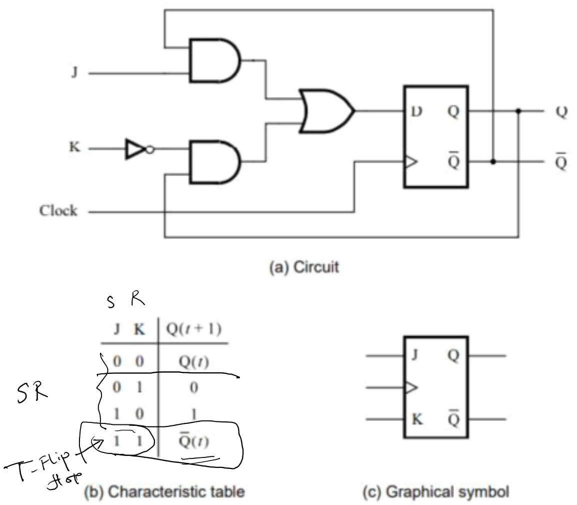
Input D is defined as $D = J\bar{Q} + \bar{k}Q$

$= 0 \cdot \bar{Q} + \bar{1} \cdot Q = 0 + 0 = 0$

A JK flip-flop is a combined effect SR flip-flop and T flip-flop.

It behaves as the SR flip flop where J=S and K=R for all combinations of J and, except J=1=K.

For J=1=K it behaves as the T flip-flop.



(a) Circuit

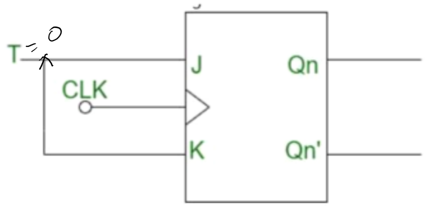
J	K	Q(t + 1)
0	0	Q(t)
0	1	0
1	0	1
1	1	$\bar{Q}(t)$

(b) Characteristic table

(c) Graphical symbol

Figure 7.17 JK flip-flop.

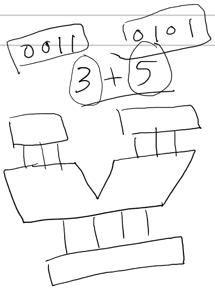
J-K Flip-flop into T Flip-flop



T	Q
0	NC
1	Togee

Memory →

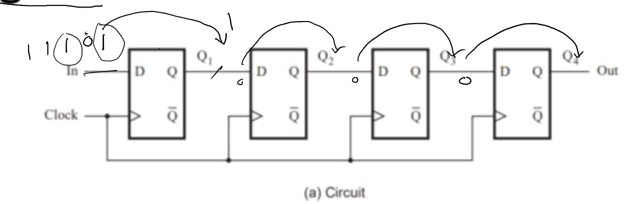
J	K	Q
0	0	NC
1	1	Tiggle



ASU

Register

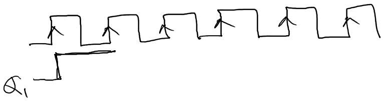
- Flip-flop is a one-bit memory cell.
- To increase the storage capacity, we have to use a group of flip-flops. The group of flip-flops is called a register.
- Shift Register: A 4-bit shift register is shown in figure.



(a) Circuit

	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

(b) A sample sequence

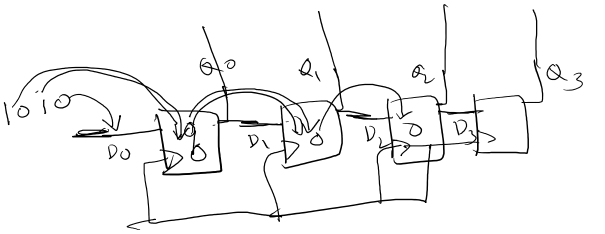


(b) A sample sequence

Parallel-access Shift Register

Shift/Load =0 , serial input and shift

Shift/load= 1, parallel load



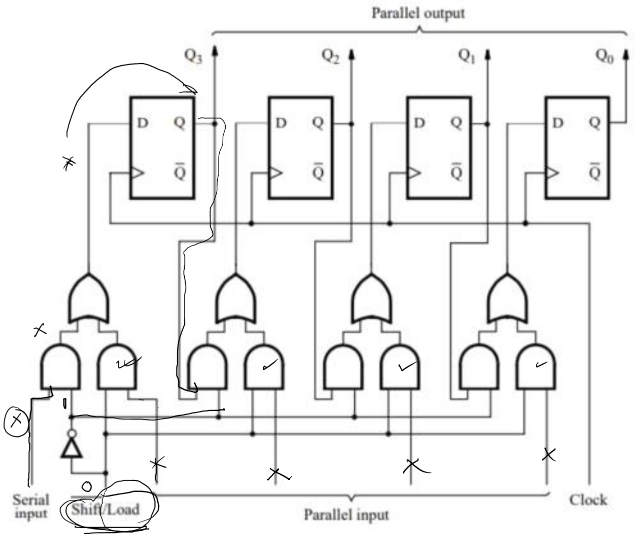


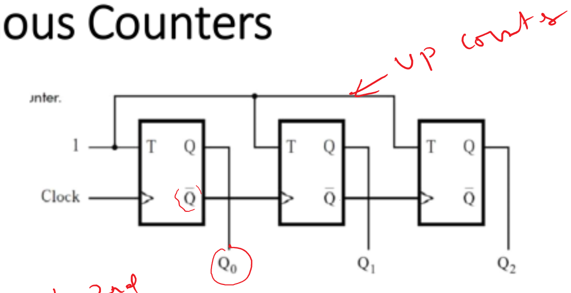
Figure 7.19 Parallel-access shift register.

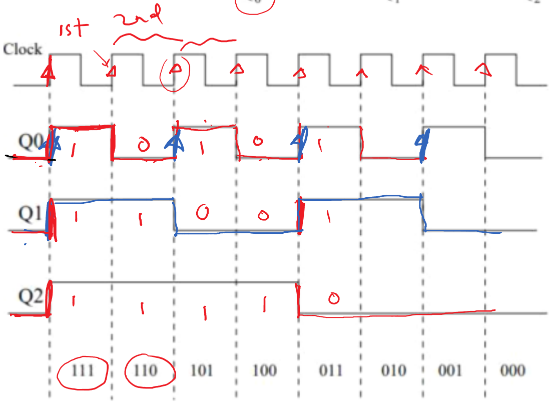
Counters: Asynchronous Counters

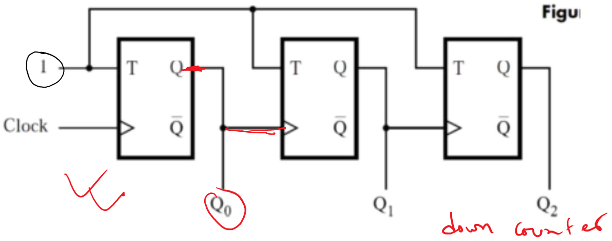
Asynchronous Counters: Asynchronous counters are those counters which **do not operate on simultaneous clocking**. In an asynchronous counter, only the first flip-flop is externally clocked using a clock pulse while the clock input for the successive flip-flops will be the output from a previous flip-flop.

This means that only a single clock pulse is not driving all the flip-flops in the arrangement of the counter.

- Ring Counter
- Modulo-8 counter (counter with 3 flip-flops)
- UP counter and down counter



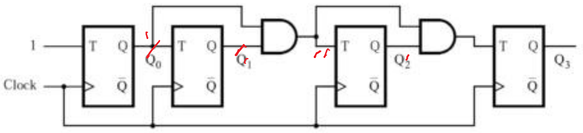




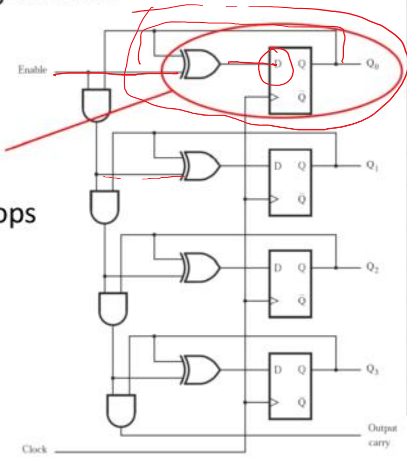
$D = Q\bar{T} + T\bar{Q} = \boxed{T \oplus Q}$

Counters: Synchronous counter

Synchronous Counters: Synchronous counters are those counters which **operate on simultaneous clocking**. All first flip-flops are externally clocked using a clock pulse. Thus, reducing the delay.



0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1



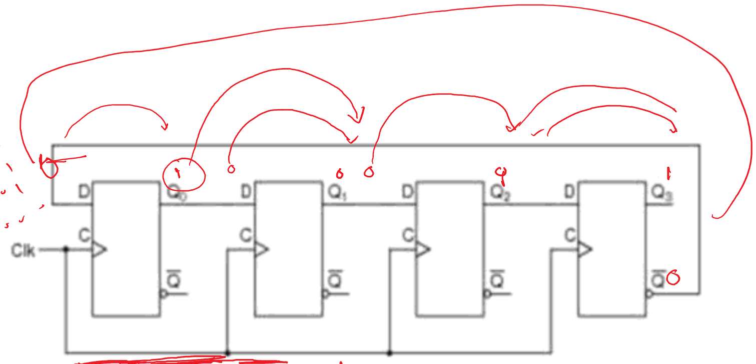
Synchronous counter with D- Flip-flops

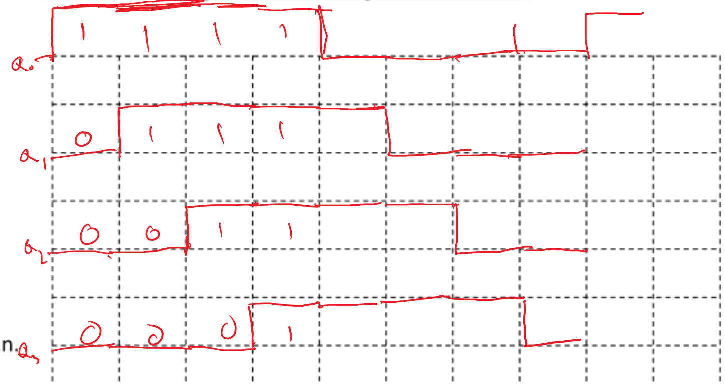
$$D = Q\bar{T} + \bar{Q}T = Q \oplus T$$

0 0 0 0
1 0 0 0
φ 1 0 0
1 1 1 0
1 1 1 1
0 1 1 1
0 0 1 1
0 0 0 1

Johnson's Counters

CLR	Clk	Qo	Q_1	Q_2	Q4
0	x	0	0	0	0
1	↑	1	0	0	0
1	↑	1	1	0	0
1	↑	1	1	1	0
1	↑	1	1	1	1
1	↑	0	1	1	1
1	↑	0	0	1	1
1	↑	0	0	0	1
1	↑	0	0	0	0





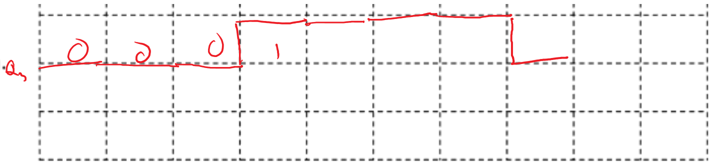
The $\bar{Q}$ output of the last stage is feedback to the first stage.

The n-bit counter generates a counting sequence of 2n.

Only a single bit has a different value of two

The Q output of the last stage is feedback to the first stage.
The n-bit counter generates a counting sequence of 2n.
Only a single bit has a different value of two consecutive codes

n

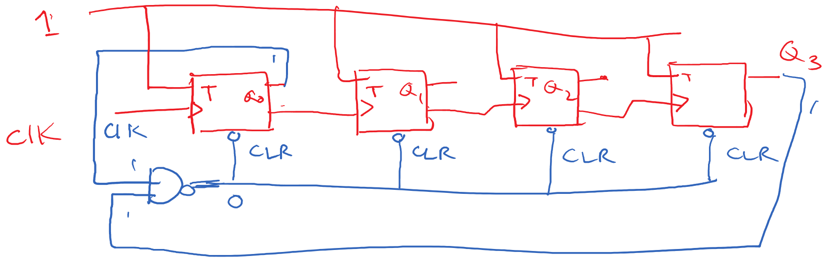


0 — 1 — - - - 9

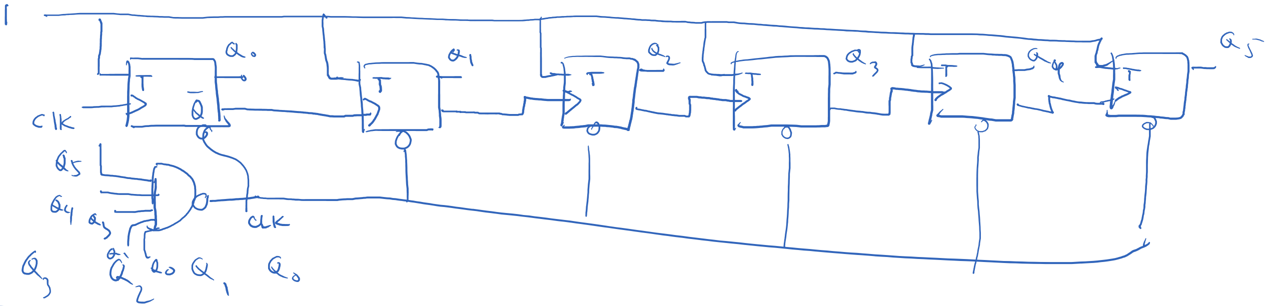
Mod-10 Counter

MSB LSB

Q_3	Q_2	Q_1	Q_0
1	0	0	1



Mod-60 counter



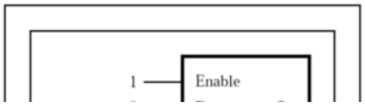
Q_5	Q_4	Q_3	Q_2	Q_1	Q_0
1	1	1	0	1	1

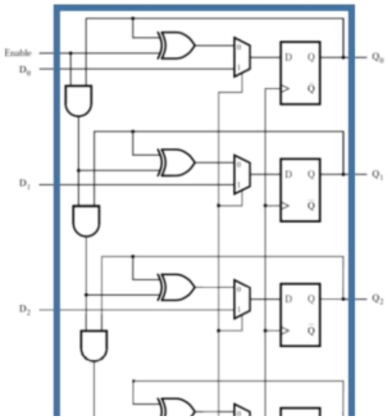
2 → 3 → 4 → 5 → 6 → 7

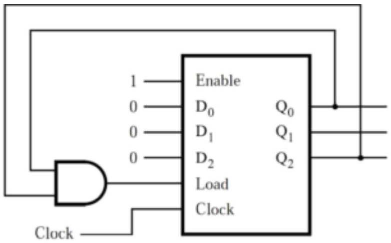
Counters with Parallel load

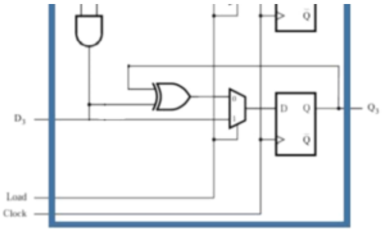
Initial count can be loaded with clear and preset input.
Alternatively, with some logic gates with the help of 2 input multiplexers.
One input of the MUX is used for normal counting and the other input is for initial value loading. Load=0 for counting an load =1 for initialization.

Modulo-6 counter with synchronous reset

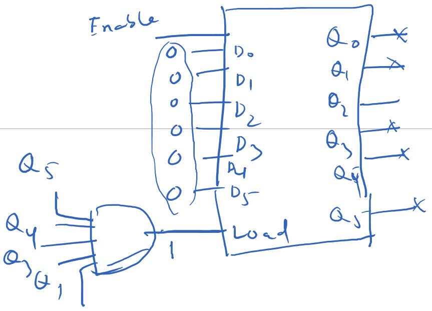








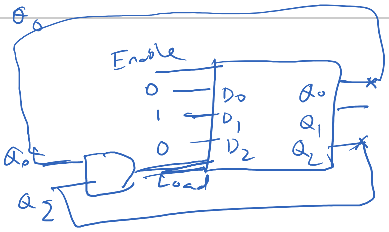
Mod-60



2 - 5

Q0 Q2

010 101



2 - 5

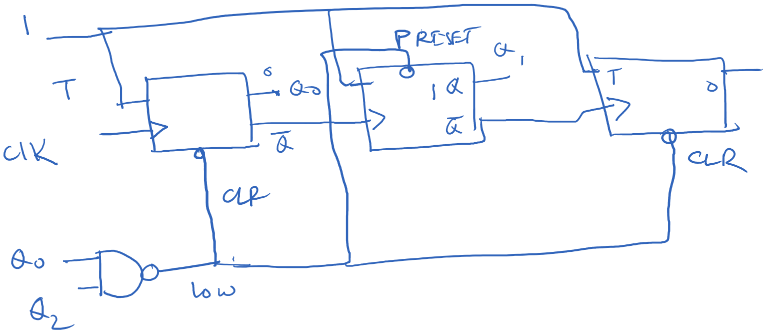
↓ ↓

010 101

↓ ↓ ↓

Q1 Q2 Q0

Q2


1
T
ClK
Q0
Q̄
CLR
PRESET
Q1
1Q
Q̄
T
0
CLR
Q0
Q2
low