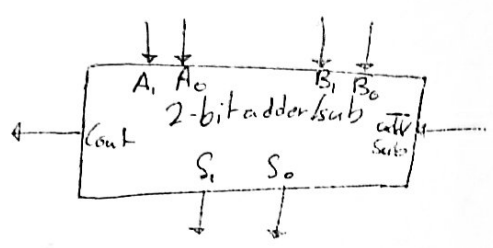


$\overline{\text{add/sub}} = 0 \Rightarrow \text{addition}$
 $\text{add/sub} = 1 \Rightarrow \text{subtraction}$

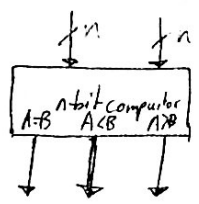
$x \oplus 1 = \bar{x}$
 $2x \oplus 0 = x$



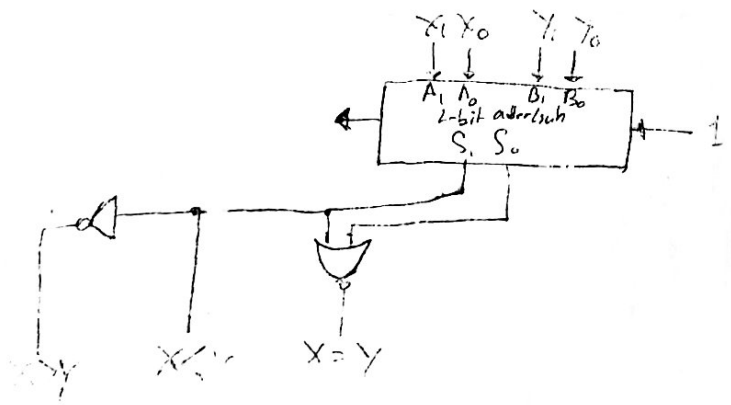
read about 7-carry decoder from the slides

Slide 22 \rightarrow 205

4.3.7 Magnitude Comparator



A	B	C_0	C_1	C_2
0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
1	1	1	0	0



3-bit comparator

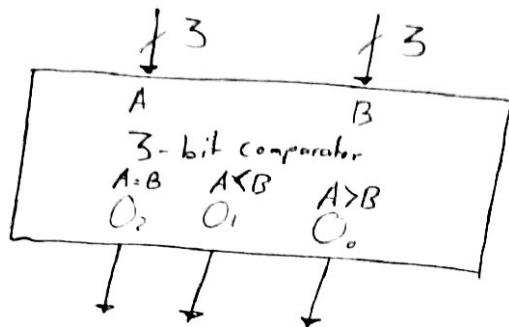
1/11/2019

A_2, A_1, A_0
 B_2, B_1, B_0

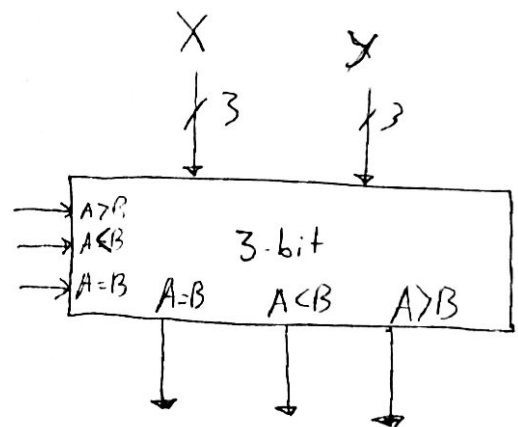
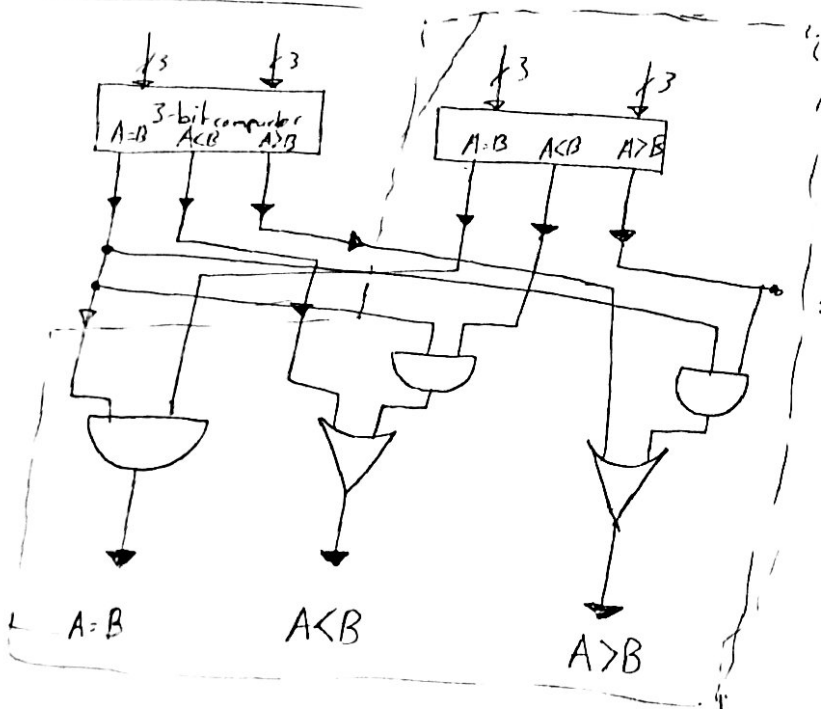
$$O_2 = \underbrace{(A_2 \odot B_2)}_{x_0} \underbrace{(A_1 \odot B_1)}_{x_1} \underbrace{(A_0 \odot B_0)}_{x_2}$$

$$O_1 = \overline{A_2} B_2 + (A_2 \odot B_2) (\overline{A_1} B_1) + (A_2 \odot B_2) (A_1 \odot B_1) (\overline{A_0} B_0)$$

$$O_0 = A_2 \overline{B_2} + (x_1) (A_1 \overline{B_1}) + (x_2) (x_1) (A_0 \overline{B_0})$$

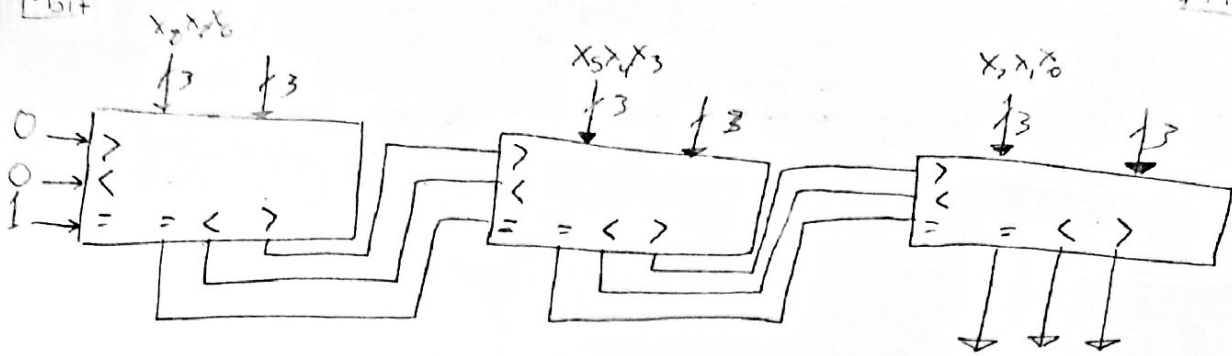


6-bit comparator



9-bit

17/7/2019



4.4 Multiplication and division by 2^n

$$30011 \xrightarrow{\times 2} 60110$$

$$60110 \xrightarrow{\times 2} 121100$$

*multiplication by 2^n is achieved by
shifting the number to left by
 n bits and feeding zeros from
the ~~left~~ right

4.4.2

unsigned

$$121100 \div 2 \rightarrow 60110$$

$$60110 \div 2 \rightarrow 30011$$

↑ zero feed

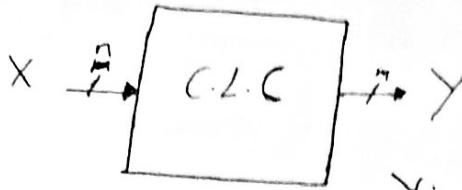
Signed

$$\begin{array}{c} \text{①} \ 1 \ 0 \ 0 \\ \uparrow \\ \text{replicate sign} \end{array} \xrightarrow{\div 2} \begin{array}{c} 1 \ 1 \ 1 \ 0 \\ \text{-2} \end{array}$$

-4

5.1 Introduction

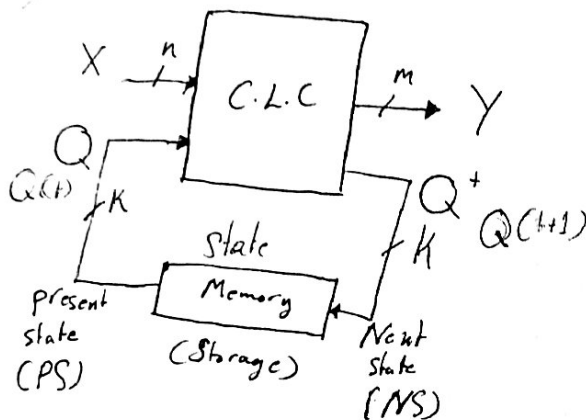
Combinational



$$Y_i = f(x_{n-1}, x_{n-2}, \dots, x_0)$$

- in C.L.C, the outputs, are only dependent on current combination

Sequential



$$Y_i = f(\underbrace{x_{n-1}, x_{n-2}, \dots, x_0}_{\text{INPUTS}}, \underbrace{Q_{K-1}, Q_{K-2}, \dots, Q_0}_{\text{PS}})$$

• Sequential circuits have two types:-

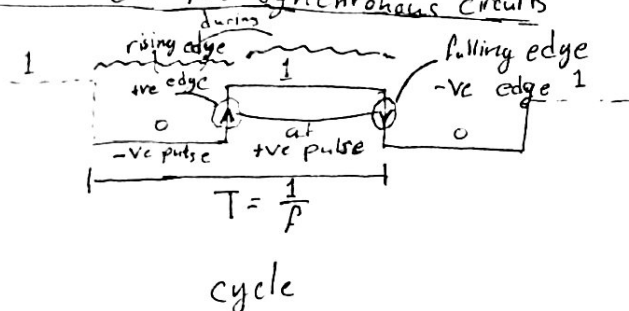
(i) Asynchronous:-

The stored value can be changed at any time. (Faster), (more complex to analyze and design)

(ii) Synchronous:- *

The stored value is ^{allowed} ~~allowed~~ to change during/at sometime only!

* Clocking in Synchronous circuits



5.2.8

5.2 Storage elements :-

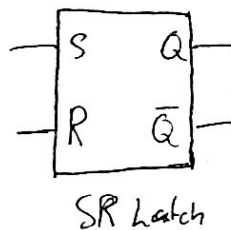
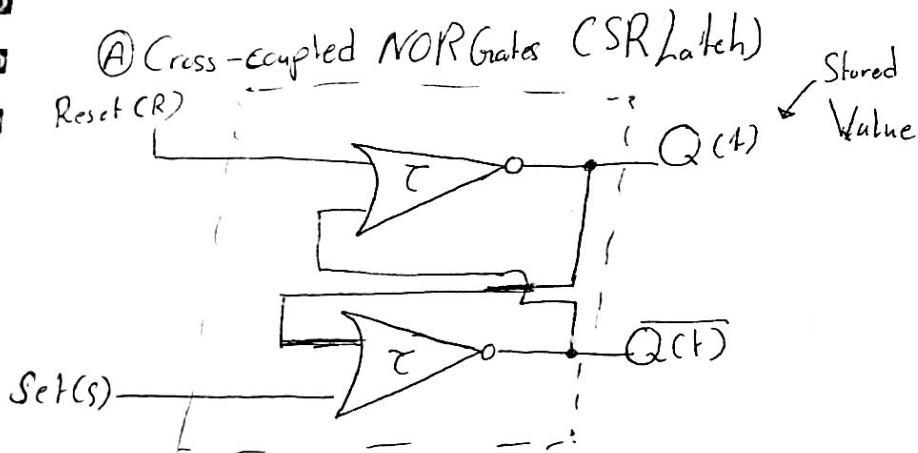
21/1/2019

5.2.1 Latches

* ~~Define~~ A Latch is a circuit that can store one bit.

* The stored value can be specified through the following operations:

- (i) ~~Store~~ Set (S) (store 1)
- (ii) Reset (R) (store 0)
- (iii) Hold (no change)
- (iv) Complement (toggle) $1 \rightarrow 0$
 $0 \rightarrow 1$



	S	R	$Q(t)$	$\overline{Q}(t)$	
t_0	X	X	X	X	
t_1	1	0	1	0	Set
t_2	→ 0	0	1	0	No Change (Hold)
t_3	0	1	0	1	Reset
t_4	→ 0	0	0	1	Hold
t_5	1	1	0	0	?? undefined

TT

S	R	$Q(t)$	$Q(t+1)$
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	0
1	1	1	0

	$Q(t+1)$		R	
	0	1	0	1
S	1	1	0	0

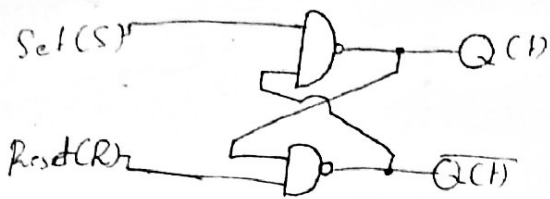
$\overline{Q(t)}$

$$Q(t+1) = S\overline{R} + \overline{R}Q(t)$$

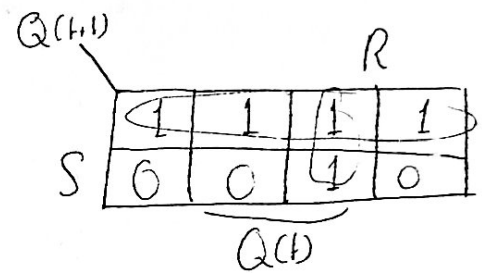
Characteristic
Equation

(B) Cross-Coupled NAND Gates ($\overline{S}\overline{R}$ Latch)

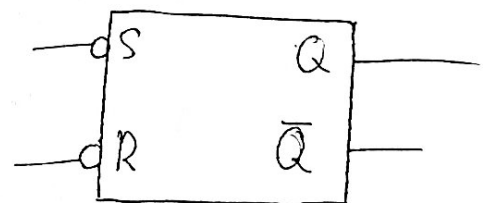
22/7/2019



S	R	Q	\overline{Q}	
x	x	x	x	
0	1	1	0	set
1	1	1	0	hold
1	0	0	1	reset
1	1	0	1	hold
0	0	1	1	? undefined



$$Q(t+1) = \overline{S} + RQ$$



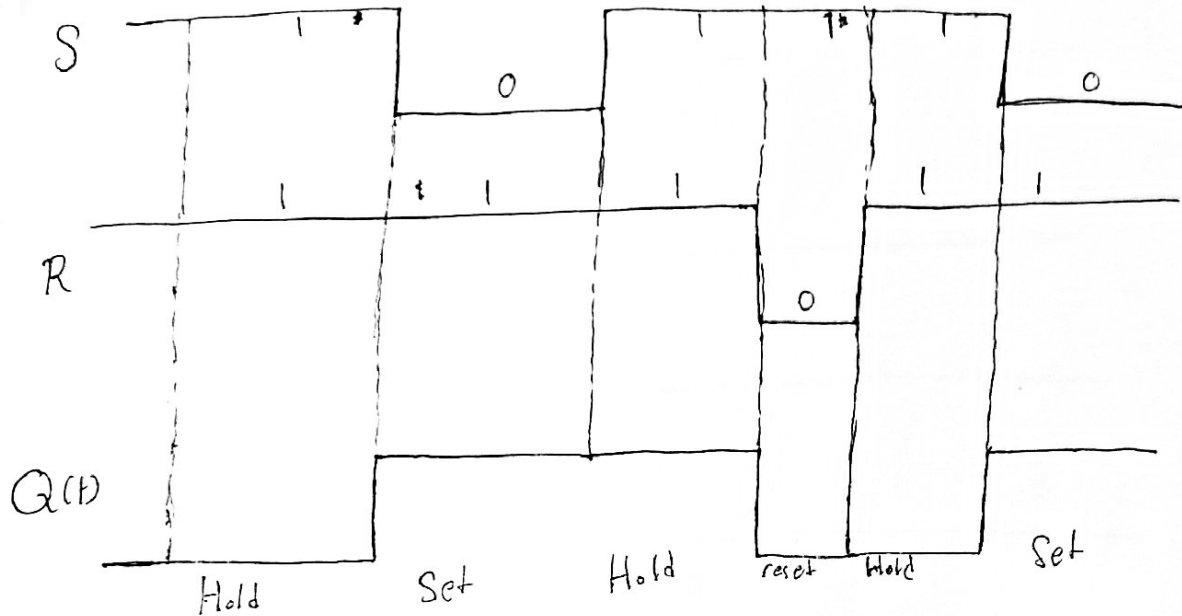
$\overline{S}\overline{R}$ latch

low input SR latch

	S	R	Q(t)	Q(t+1)
?	0	0	0	1
	0	0	1	1
Set	0	1	0	1
	0	1	1	1
Reset	1	0	0	0
	1	0	1	0
hold	1	1	0	0
	1	1	1	1

22/11/2011

ex. Complete the following timing diagram:-



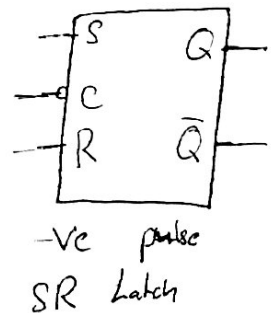
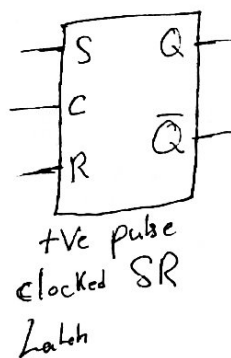
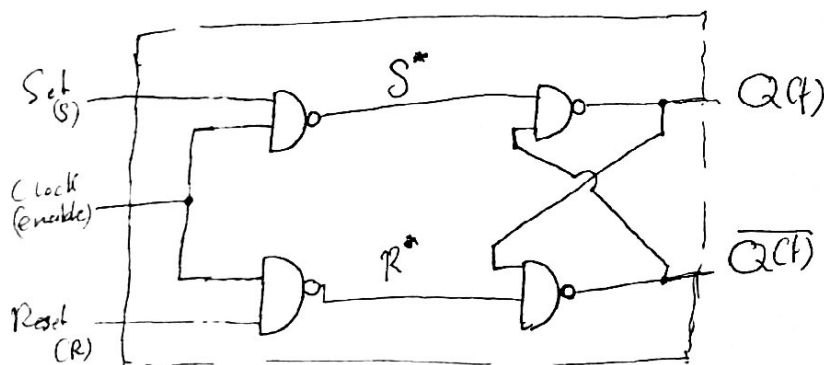
5.2.2 Clocked Latches (Latches with enable):-

* Latches may change the stored value as soon as the input changes.

* what if ~~we~~ we need to

$$\text{Clock} = 0 \Rightarrow \begin{matrix} S^* = 1 \\ R^* = 1 \end{matrix} \Rightarrow \text{Hold}$$

$$\text{Clock} = 1 \Rightarrow \begin{matrix} S^* = S \\ R^* = R \end{matrix} \Rightarrow \text{Normal Operation}$$



clock

S

R

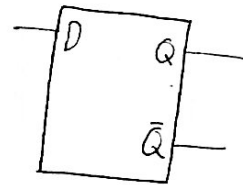
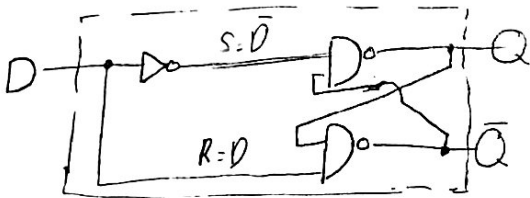
Q₁

using +ve pulse SR latch

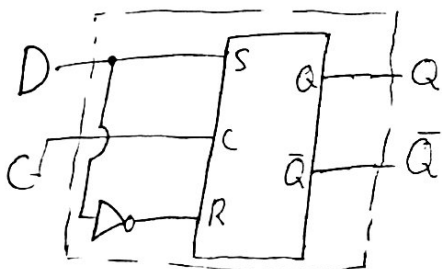
Q₂

using -ve pulse SR latch

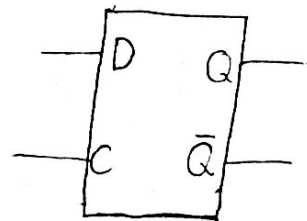
* Clocked D-Latch



D-Latch



Clocked D-Latch



+ve pulse
clocked D-Latch

$$C=0 \Rightarrow Q(t+1) = Q(t) \text{ hold}$$

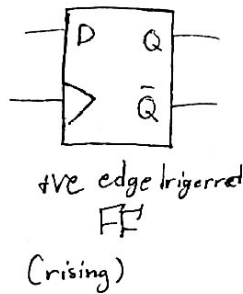
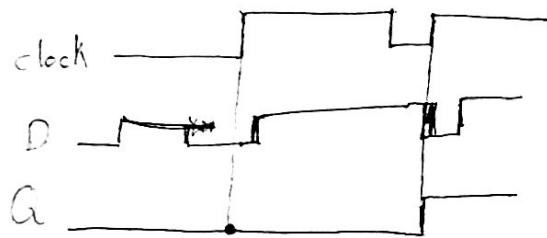
$$C=1 \Rightarrow Q(t+1) = D$$

2.2.3 Flip Flops

- * In clocked latches, the input is allowed to change the stored value when the clock is +ve
- * What if we need to allow the input to affect the stored value at specified instant of time?

- * Flip-Flops
 - Master-slave FFs
 - Edge triggered FFs

(A) D-FFs



Characteristic eqn
 $Q^+ = D$
 otherwise $Q^+ = Q$

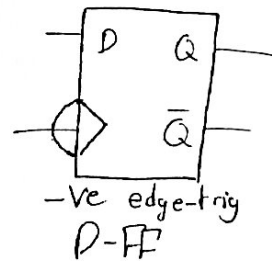
D	$Q(t)$	$Q(t+1)$
0	0	0
0	1	0
1	0	1
1	1	1

Characteristic Table

\Rightarrow

D	$Q(t+1)$
0	0
1	1

(Analysis)

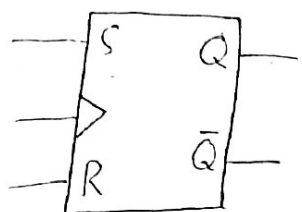


$Q(t)$	$Q(t+1)$	D
0	0	0
0	1	1
1	0	0
1	1	1

Excitation Table (Design)

23/7/2011

③ SR FF



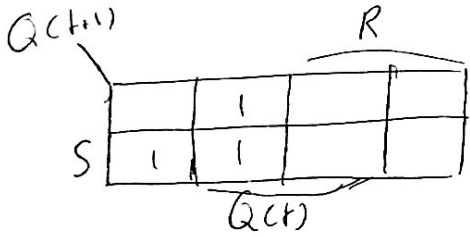
↑ve edge-triggered
SR FF

can be -ve edge-triggered

S	R	$Q(t+1)$
0	0	$Q(t)$
0	1	0
1	0	1
1	1	??

Characteristic
table

S	R	$Q(t)$	$Q(t+1)$
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	??



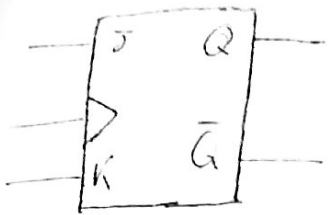
$$Q(t+1) = S\bar{R} + \bar{R}Q(t+1)$$

$Q(t)$	$Q(t+1)$	S	R
0	0	0	x
0	1	1	0
1	0	0	1
1	1	x	0

Excitation
Table

28/7/2019

JK FF



Can be -ve edge-trigg JKFF

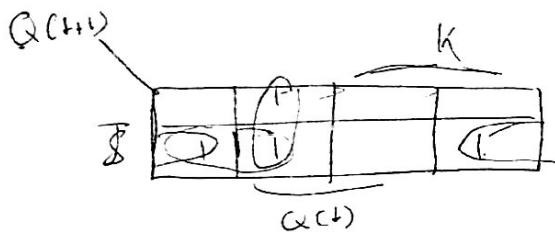
+ve edge-trigg

JKFF

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

Characteristic Table

J	K	Q(t)	Q(t+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

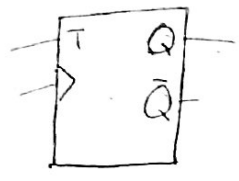


$$Q(t+1) = \bar{Q}(t) + JQ(t)$$

Q(t)	Q(t+1)	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

Excitation Table

① TFF



+ve
-ve

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

T	Q(t)	Q(t+1)
0	0	0
0	1	1
1	0	1
1	1	0

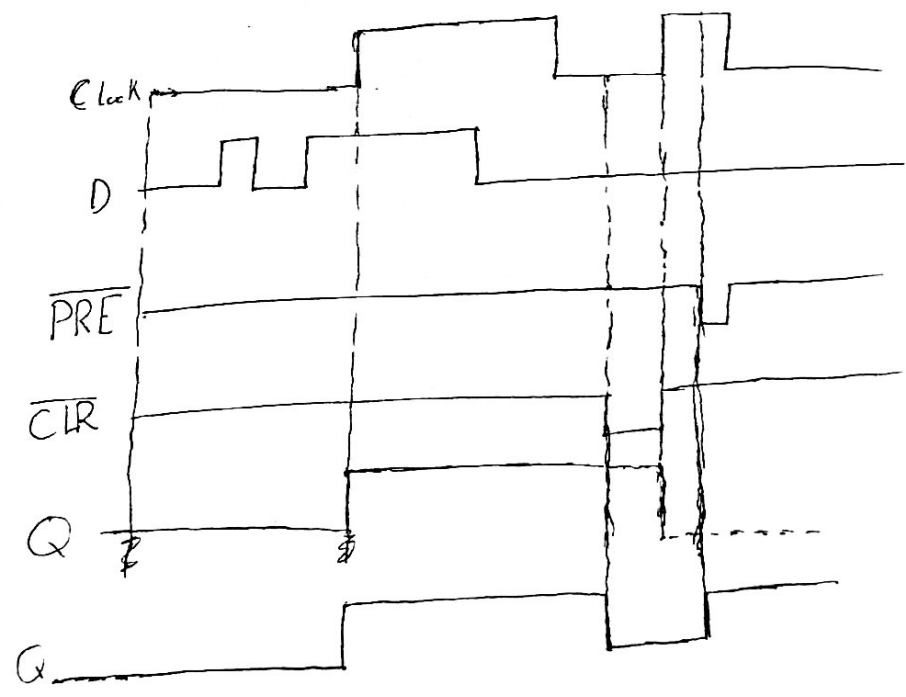
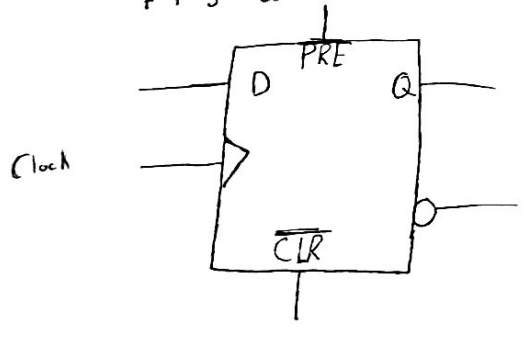
$Q(t+1) = T \oplus Q(t)$ char. eqn

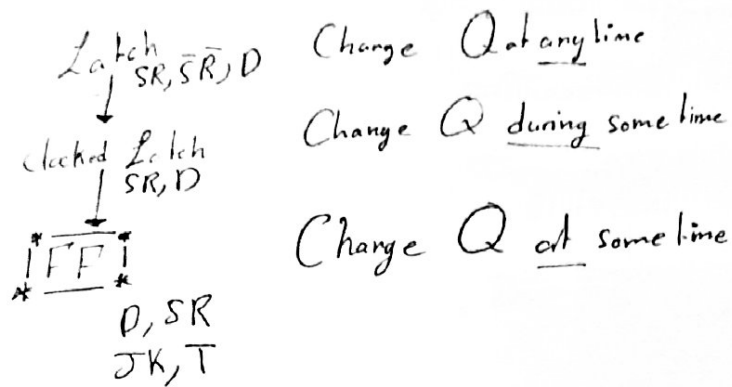
Q	Q ⁺	T
0	0	0
0	1	1
1	0	1
1	1	0

excitation
table

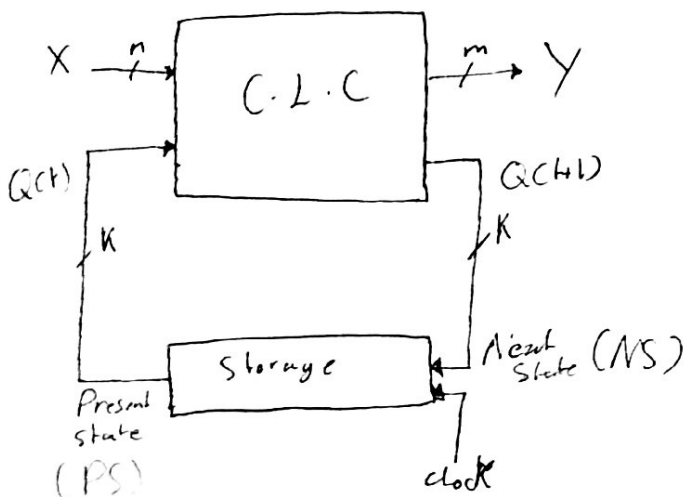
Note:-

TFFs with direct inputs



Summary:-5.3 Analysis of Sequential Circuit

Given the Logic diagram of some sequential circuit (comb + storage), we want to determine the behavior of the circuit as the input(s) changes over time. The behavior is basically the changes in the output(s) and the stored value (state).



* Analysis Steps

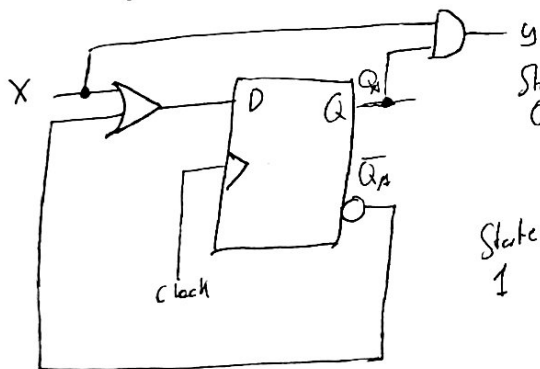
25/11/2019

- ① Obtain the FF input equations directly from the diagram
- ② Obtain the output(s) equations directly from the diagram.
- ③ Write the ~~FF~~ state table:
 - (i) It is similar to the TT,
 - (ii) It has the external input variables and the present state variables in the input section.
 - (iii) It has the outputs and the next state values in the output section.

~~④~~

- ④ Optionally, draw the state diagram.

ex Obtain the state table of the following circuit.



one input X

1 FF \rightarrow QA \rightarrow two states

PS		IN		NS		OUT
$Q_A(t)$		X		$Q_A(t+1)$		y
State 0	0	0	1	1	0	0
	0	1	1	1	0	1
State 1	1	0	0	0	0	0
	1	1	1	1	1	1

• FF Input equation

$$D = X + \overline{Q_A(t)}$$

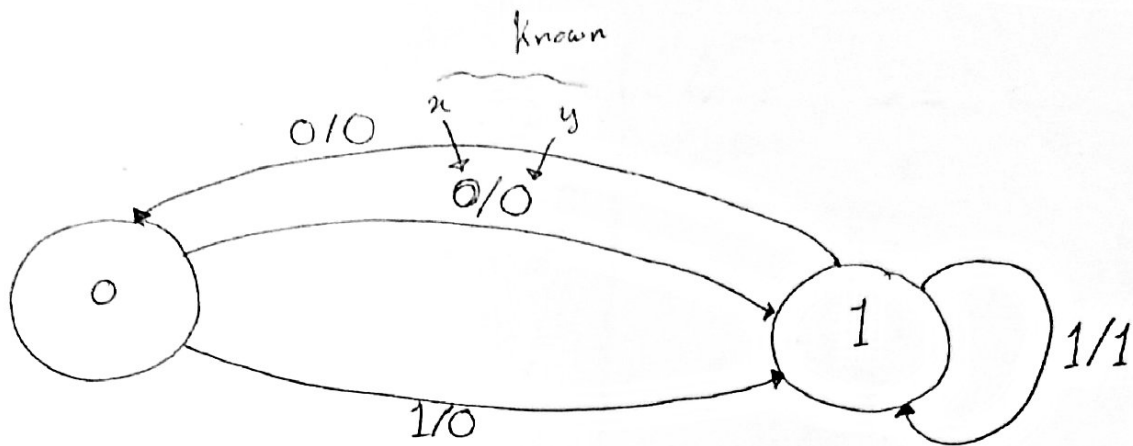
• Output equation

$$Y = X \cdot Q_A(t)$$

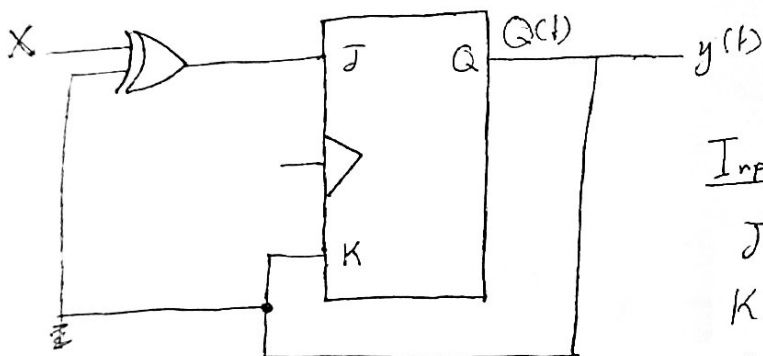
Mealy Sequential Circuit (outputs is a function of PS and IN)

Using state diagram.

24/7/2019



Ex Draw the state diagram of the following circuit:-



Input equations

$$J = X \oplus Q(t)$$

$$K = Q(t)$$

Output

$$y = Q(t)$$

Moore Sequential Circuit

Output is a function of Present state only

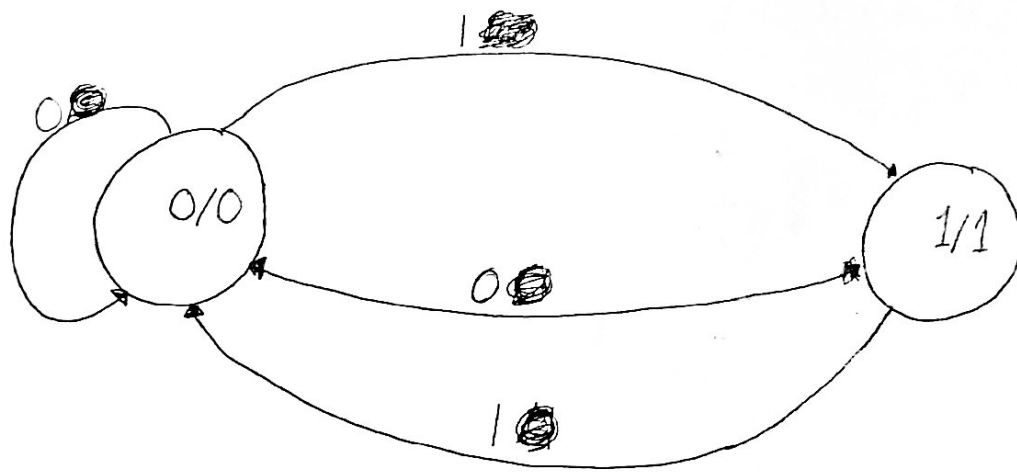
FF = 1 \rightarrow Q(t) \rightarrow 2 states

Inputs = 1 X

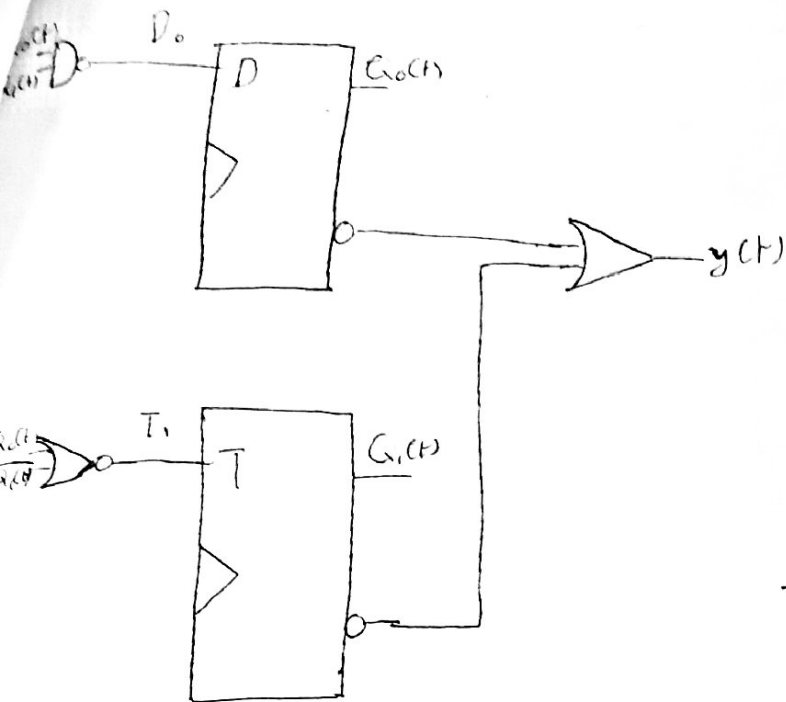
Outputs = 1 y

~~PS~~

PS Q(n)	TN X	J	K	NS Q(n+1)	OUT y
0	0	0	0	0	0
0	1	1	0	1	0
1	0	1	1	0	1
1	1	0	1	0	1



24/7/2019



FF Input equation:

$$D_0 = \overline{Q_0(t)} Q_1(t)$$

$$T_1 = Q_0(t) + \overline{Q_1(t)}$$

FF: 2 \rightarrow $Q_1 Q_0 \rightarrow$ 4 states

inputs = 0

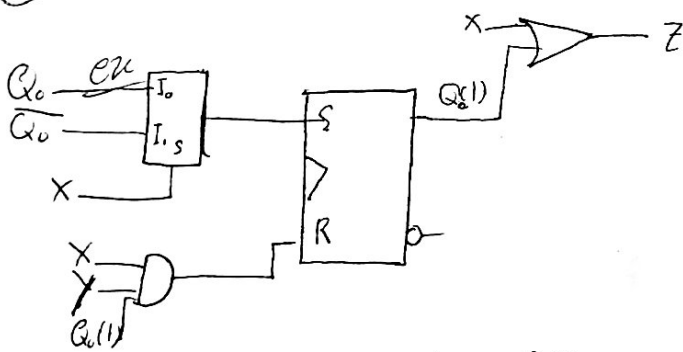
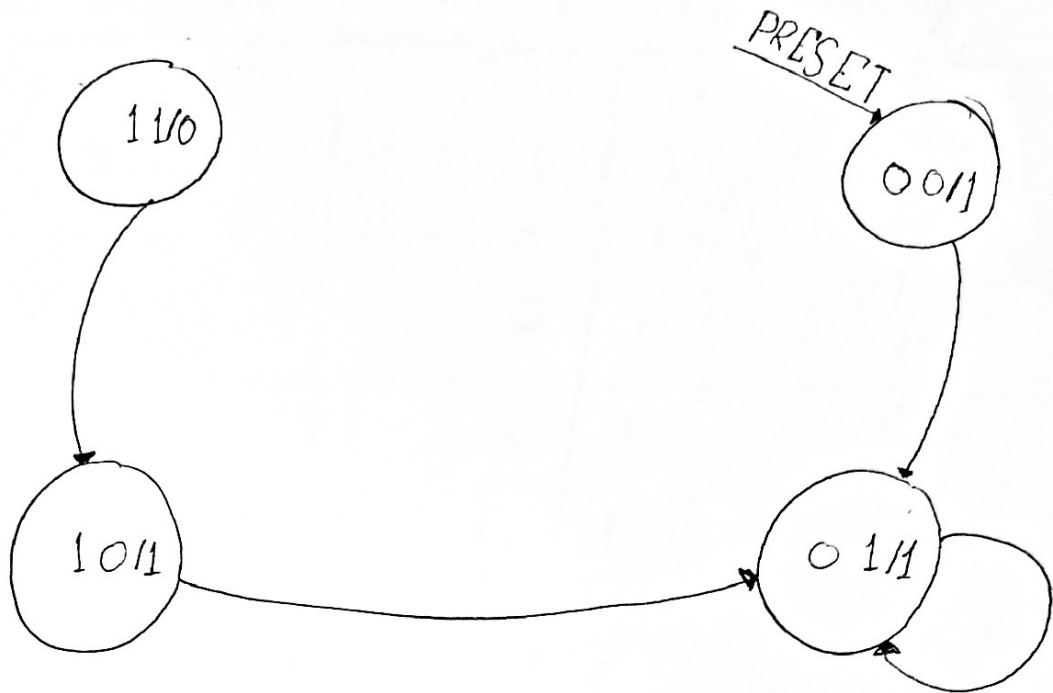
~~FF~~ outputs = 1 \rightarrow y

Output

$$y = \overline{Q_0(t)} + \overline{Q_1(t)}$$

<u>PS</u>				<u>NS</u>		
$Q_1(t)$	$Q_0(t)$	D_0	T_1	Q_1^+	Q_0^+	y
0	0	1	0	0	1	1
0	1	1	0	0	1	1
1	0	1	1	0	1	1
1	1	0	0	1	0	0

24/11/2021



$$S = 2 - 1_0 - 1_{min}$$

$$R = X \cdot Y \cdot Q_0(t)$$

$$Z = X \cdot Q_0(t)$$

#FF's $\rightarrow 1 \rightarrow 2 \text{ states} \rightarrow Q(t)$

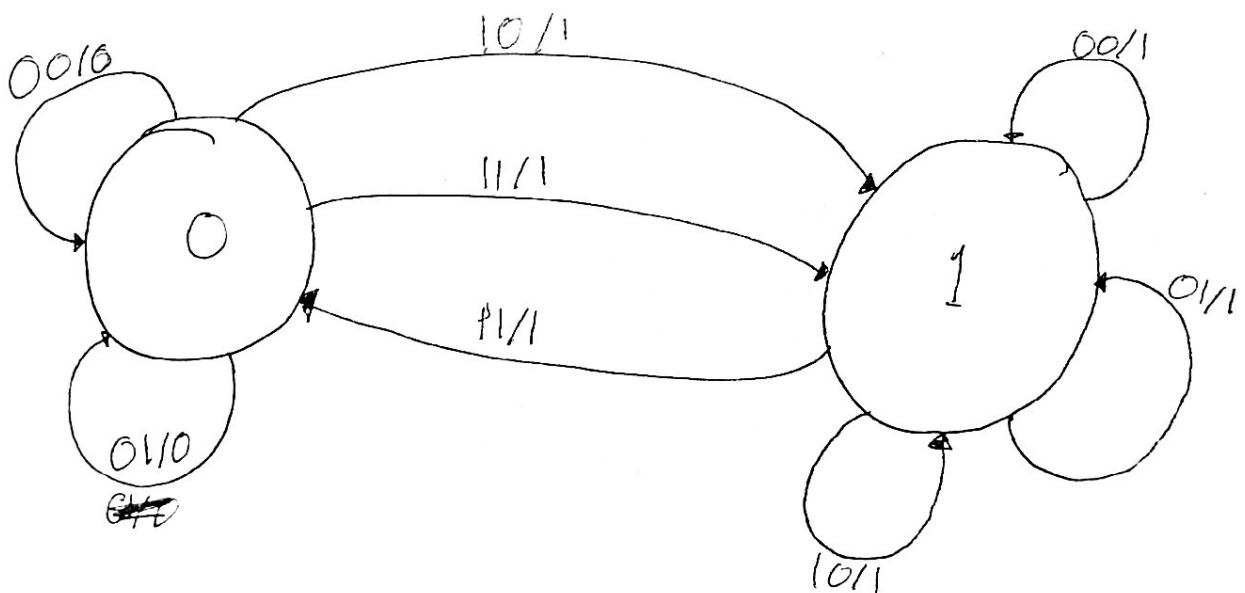
#IN = 2 $\rightarrow x, y$

#OUT = 1 $\rightarrow Z$

21/7/2019

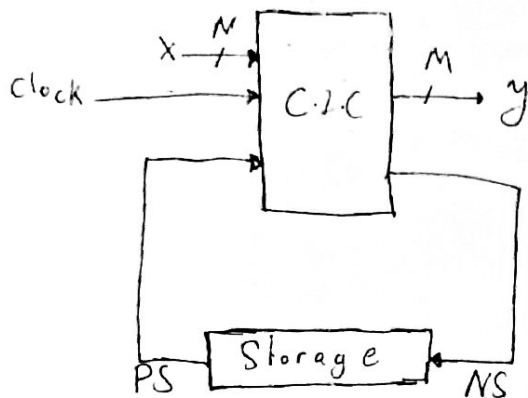
OUT

<u>PS</u>	<u>IN</u>		<u>NS</u>		<u>OUT</u>	
Q_0	X	Y	S	R	Q_0^+	Z
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	1	0	1	1
0	1	1	1	0	1	1
1	0	0	1	0	1	1
1	0	1	1	0	1	1
1	1	0	0	0	1	1
1	1	1	0	1	0	1



5.4 Design of Sequential Circuits

28/7/2019



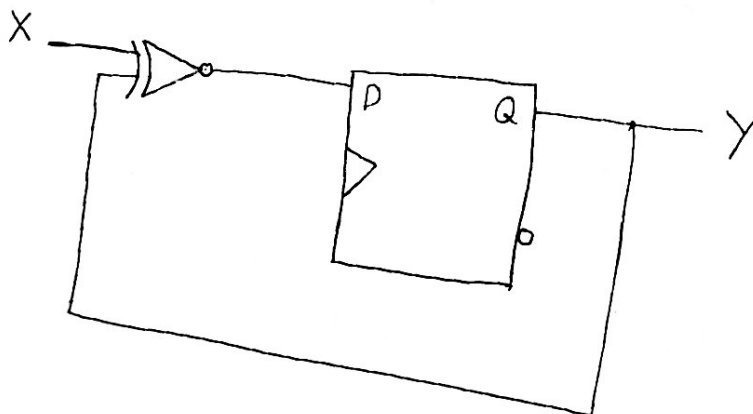
Ex Design the sequential circuit described by the following state table. Use DFFs

PS	IN	NS	OUT	
Q(t)	X	Q(t+1)	Y	D
0	0	1	0	1
0	1	0	0	0
1	0	0	1	0
1	1	1	1	1

States = 2 \rightarrow # FFs = 1

inputs = 1 \rightarrow X

Outputs = 1 \rightarrow Y



Ex. Repeat the previous example but with JK FFs

22/11/04

$Q(t)$	X	$Q(t+1)$	Y	J	K
0	0	1	0	1	x
0	1	0	0	0	x
1	0	0	1	x	1
1	1	1	1	x	0

J

\bar{x}	x
$Q(t)$	
0	1
1	x

$J = \bar{x}$

K

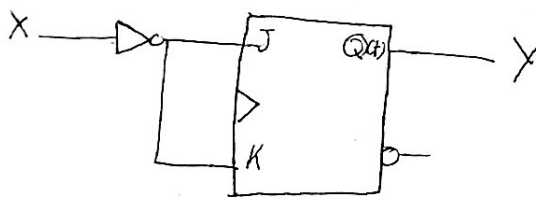
	x
$Q(t)$	
0	x
1	0

$K = \bar{x}$

Y

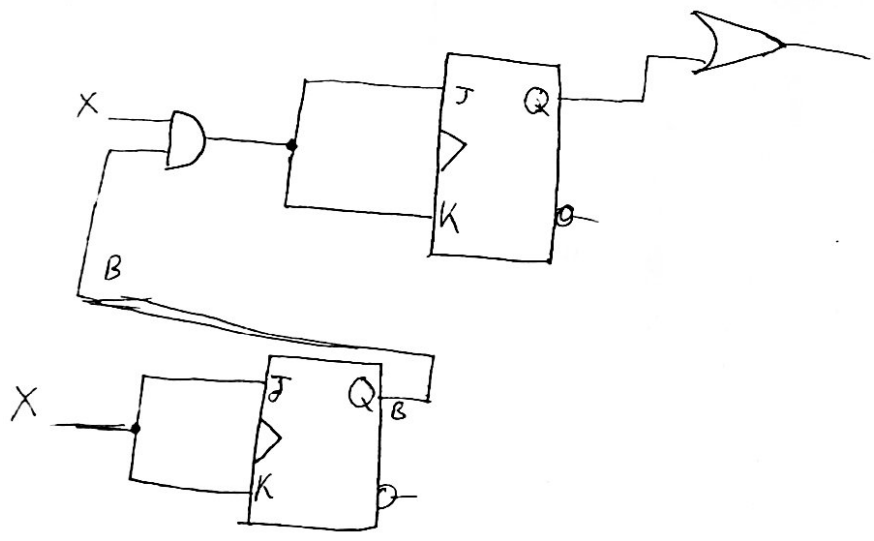
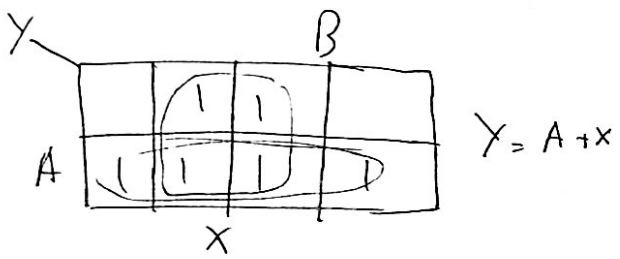
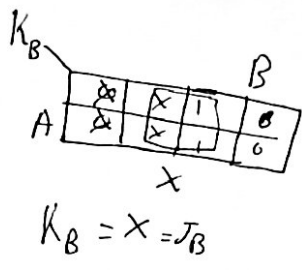
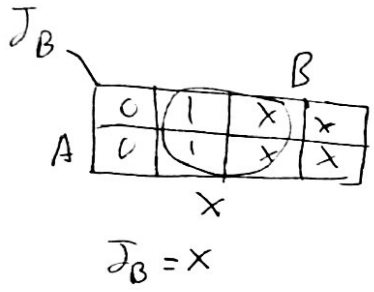
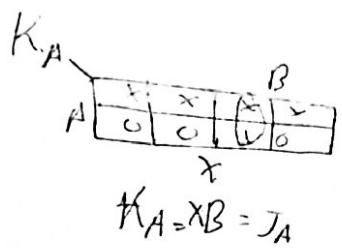
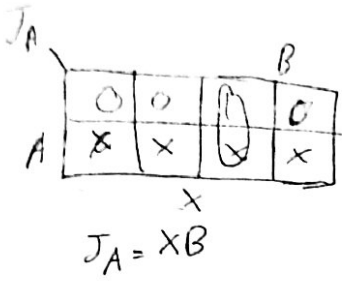
	x
$Q(t)$	
0	0
1	1

$Y = Q(t)$



Ex. Design the circuit described by the following state table.
(i) Use JK FFs

\underline{PS}		\underline{IN}	\underline{NS}		\underline{OUT}	J_A		K_A		J_B		K_B	
A	B	X	A ⁺	B ⁺	Y								
0	0	0	0	0	0	0	x	0	x	0	x		
0	0	1	0	1	1	0	x	1	x	1	x		
0	1	0	0	1	0	0	x	x	0	x	0		
0	1	1	1	0	1	1	x	x	1	x	1		
1	0	0	1	0	1	x	0	0	x	0	x		
1	0	1	1	1	1	x	0	0	1	1	x		
1	1	0	1	1	1	x	0	x	0	1	0		
1	1	1	0	0	1	x	1	x	1	x	1		

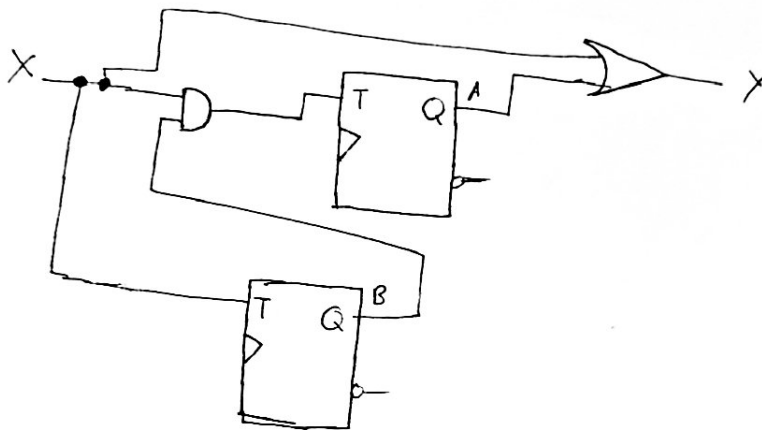


T_A	T_B
0	0
0	1
0	0
1	1
0	0
0	1
0	1
1	1

$$T_A = XB$$

0	1	1	0		
0	1	1	0		

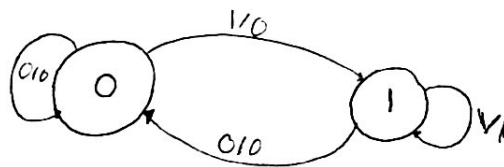
$$T_B = X$$



ex Design the sequential circuit described by the following state diagram

(i) DFFs

(ii) JKFFs

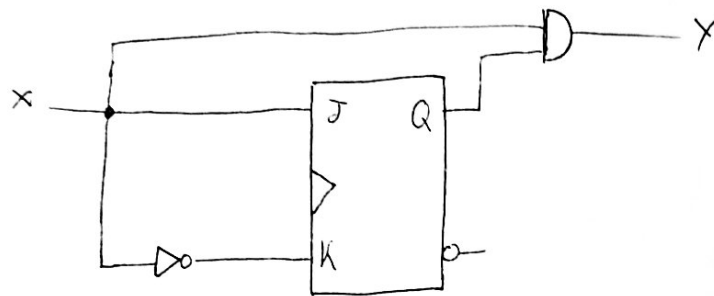
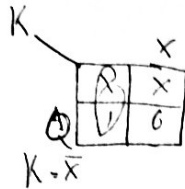
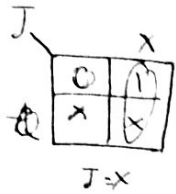
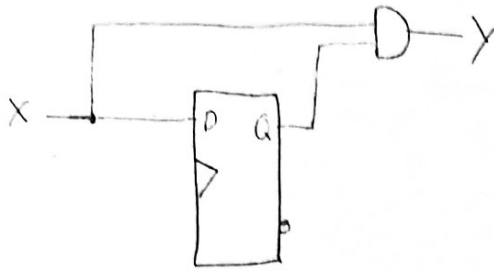


(I.)

PS Q	IN X	NS Q^+	OUT Y	D	J	K
0	0	0	0	0	0	x
0	1	1	0	1	0	x
1	0	0	0	0	x	1
1	1	1	1	1	x	0

(11)

23/7/2019



Ex Design a circuit that counts continuously through the following pattern: 0, 1, 2, 3, 0, 1, 2, 3, 0, ... \rightarrow moore circuit

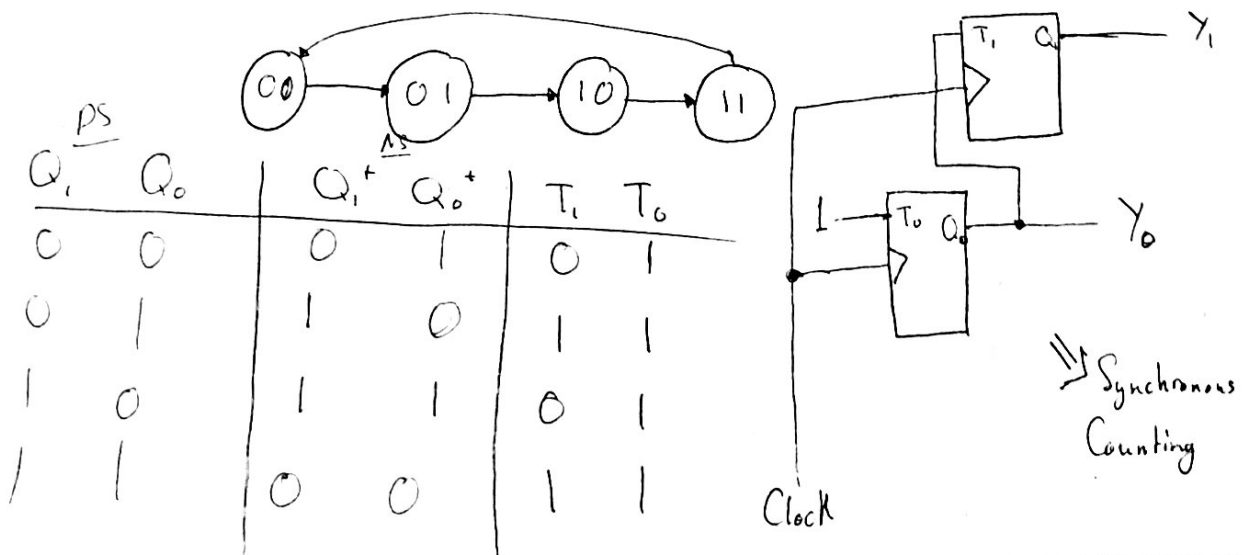
$$\#FF = \lceil \log_2 \#states \rceil = \lceil \log_2 4 \rceil = 2 \Rightarrow Q_1, Q_0$$

$$T_0 = 1$$

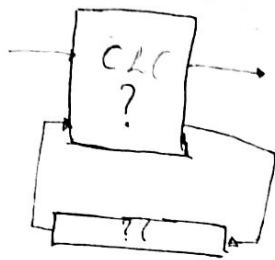
$$T_1 = Q_0^{(t)}$$

$$\#IN = 0$$

#OUT = Same as PS \rightarrow the count value



counts: 1, 3, 5, 7, 1, 3, 5, 7...
using D FFs

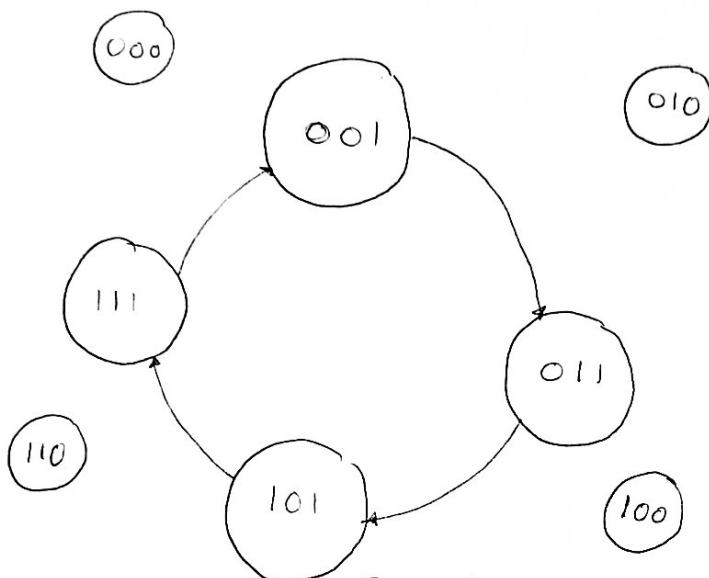


$$\#FF = \lceil \log_2 4 \rceil = 2$$

We need 3 bits to represent the count value!

$$\#FF = 3 = Q_2 Q_1 Q_0$$

$$\#IN = 0 \quad \#EN = PS$$



Q_2	Q_1	Q_0	D_2	D_1	D_0	D_2	D_1	D_0
* 0	0	0						
0	0	1	0	1	1			
* 0	1	0						
0	1	1	1	0	1			
* 1	0	0						
1	0	1	1	1	1			
+ 1	1	0						
1	1	1	0	0	1			

29/7/2019

D_2

x	0	1	x
x	1	0	x

D_1

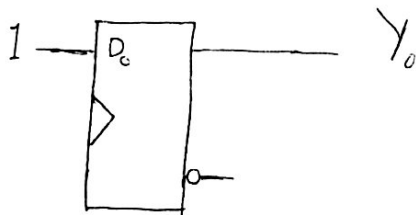
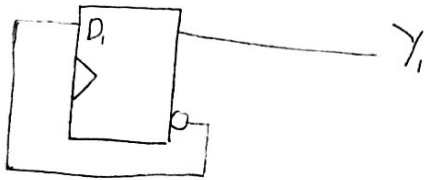
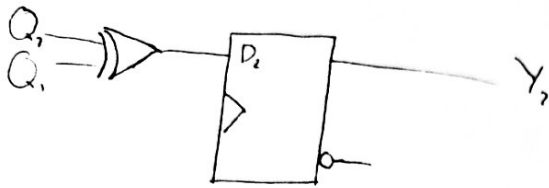
x	x	0	x
x	1	0	x

D_0

x	1	1	x
x	1	1	x

$$D_2 = Q_2 \oplus Q_1$$

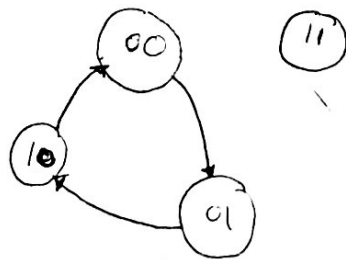
$$D_1 = \overline{Q_1} \quad D_0 = 1$$



ex

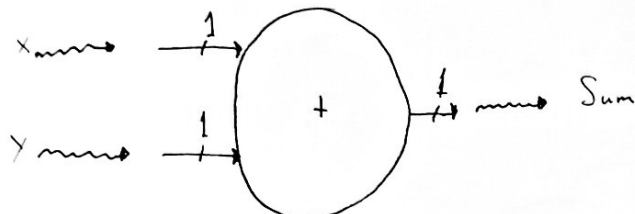
Design

(i) P.C
(ii)



X ... 0010 memorize the carry

Y ... 0011



Sequential Circuit

#inputs = 2 \Rightarrow x, y

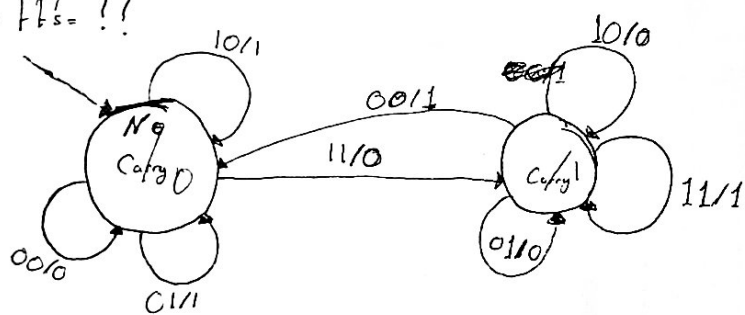
#Outputs = 1 \rightarrow sum

FFs = ??

01/0

10/0

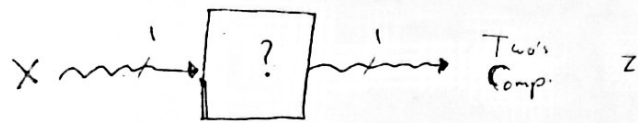
11/1



\underline{ps}	\underline{IN}		\underline{NS}	\underline{OUT}						
Q	X	Y	Q ⁺	Sum	D	T	J	K	S	R
0	0	0	0	0						
0	0	1	0	1						
0	1	0	0	1						
0	1	1	1	0						
1	0	0	0	1						
1	0	1	1	0						
1	1	0	1	0						
1	1	1	1	1						

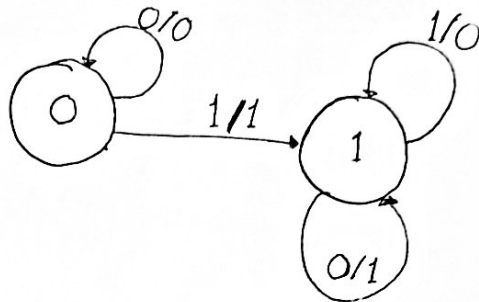
Qx Design a Serial Two's
Comp. Circuit

29/7/2019



IMS=1 X

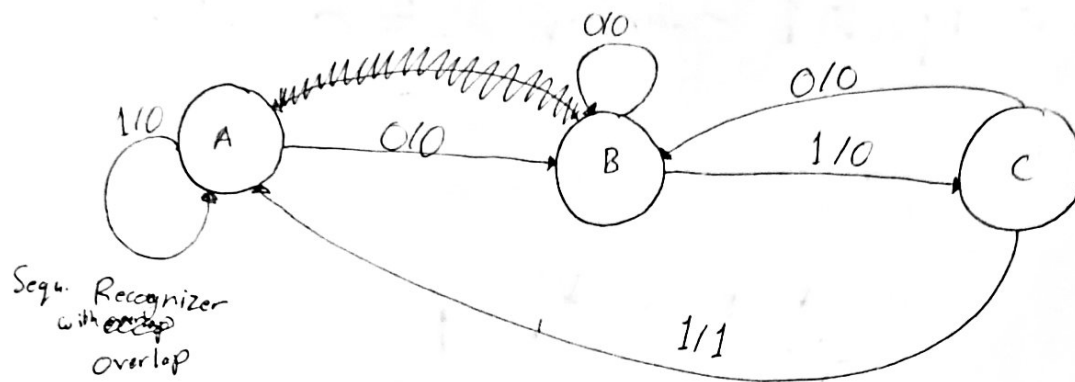
OUT=1 Z



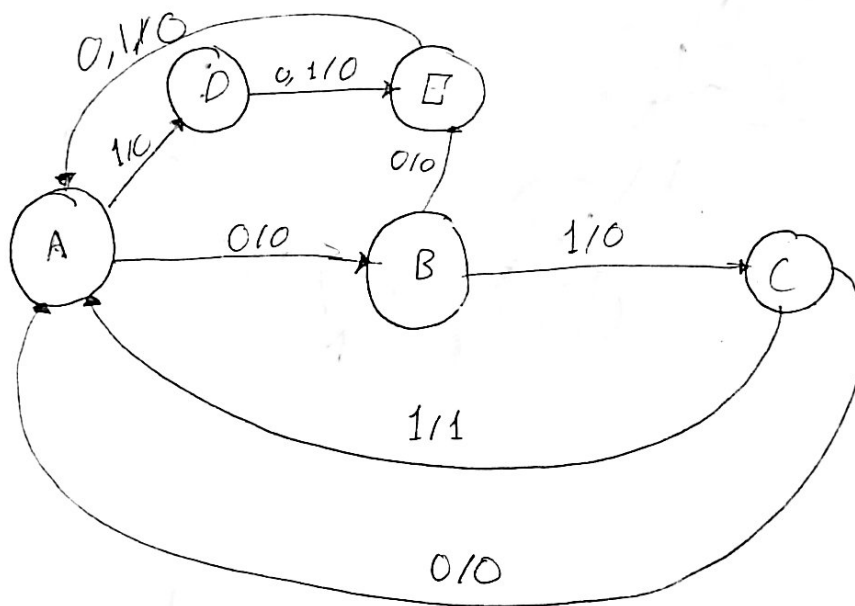
<u>PS</u>	<u>IN</u>	<u>NS</u>	<u>OUT</u>						
Q	X	Q ⁺	Z	D	T	J	K	R	S
0	0	0	0	0	0	0	1X	0	x
0	1	1	1	1	1	1	x	1	0
1	0	1	1	1	0	x	0	x	0
1	1	1	0	1	0	x	0	x	0

Design a circuit that streams
of bits serially and output 1
whenever the pattern (110) is
detected

1 V 11/10/19



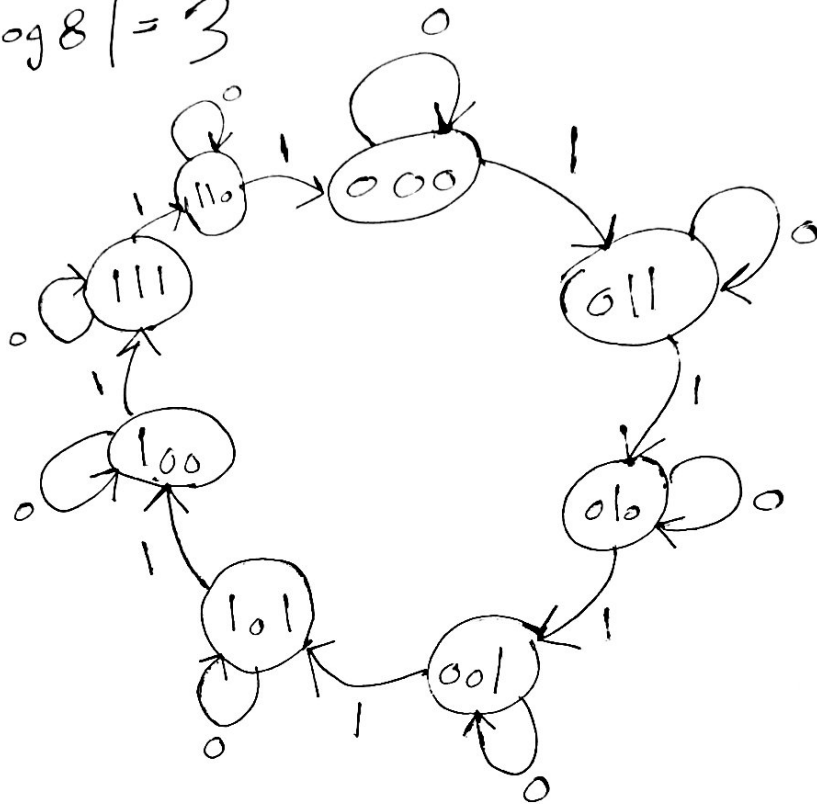
PS IN MS OUT



EX: Design a 3-bit binary counter that has one input to operation such that:

- * when its input 0 the counter pauses counting
- * when the input 1 counter counts 0, 3, 2, 1, 5, 4, 7, 6, 0, 3, 2, ...

$$\#FFs = \lceil \log 8 \rceil = 3$$



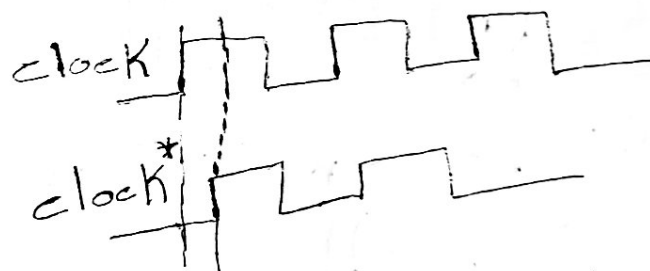
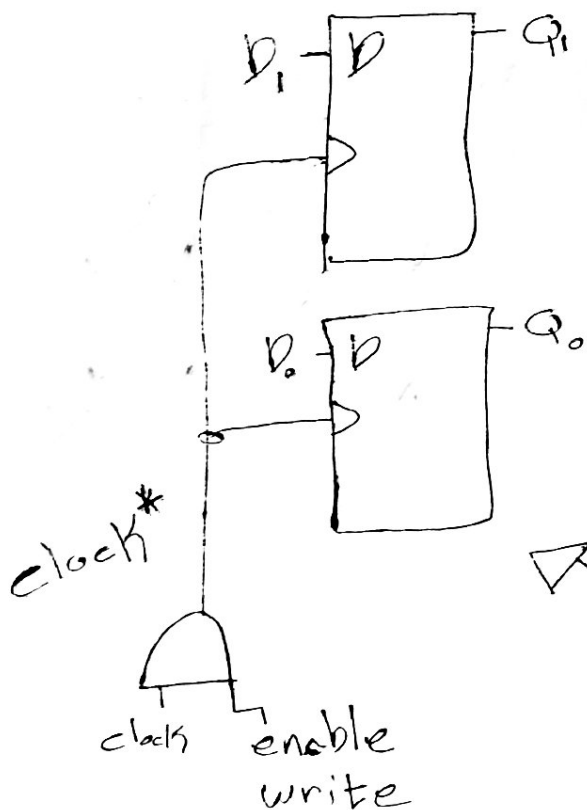
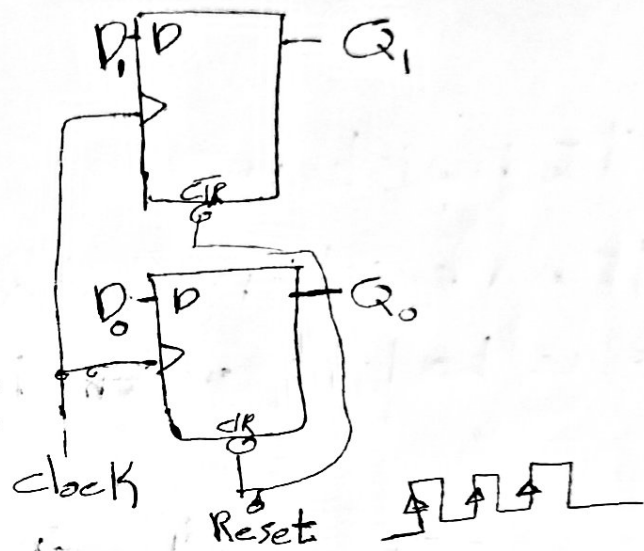
Registers & Counter

Registers

An n -bit register is a set of n storage elements that can store n -bit value

* 2-bit Register

* the stored value may change in every cycle
what if we want to hold the value for a longer time?

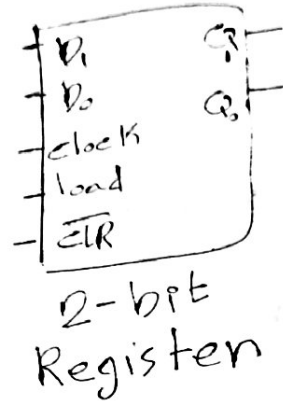
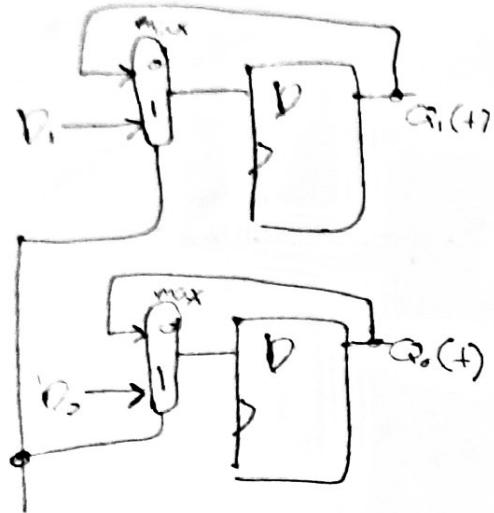


① control the clock

* There is delay

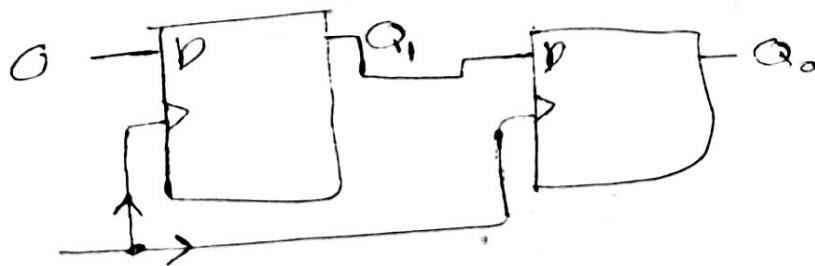
②

* this way is better (no delay)

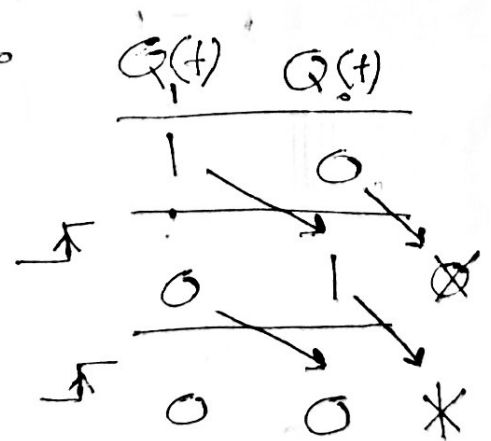


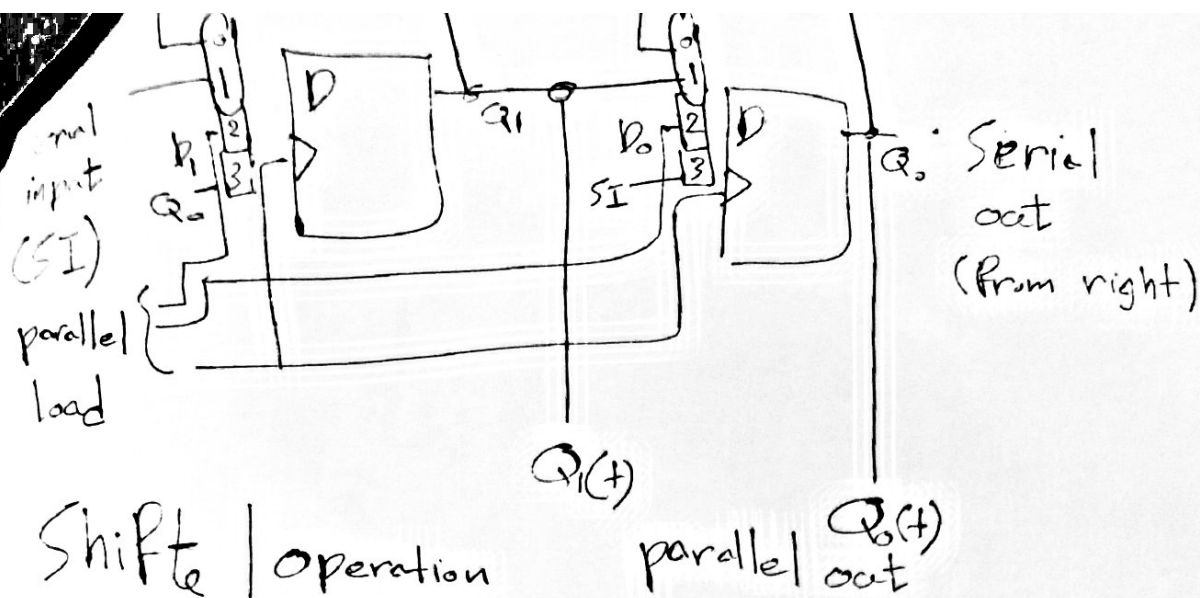
* Shift registers :

Is a register that can shift the stored value to left or ~~and~~ Right

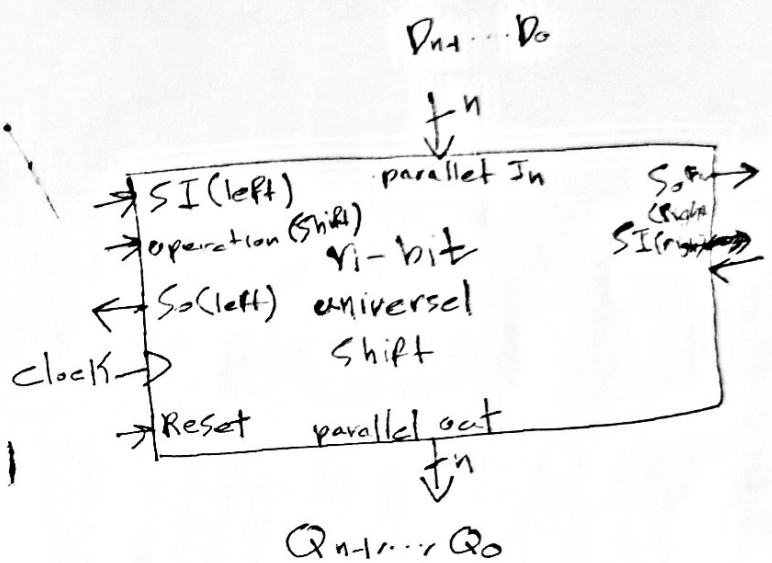


* this Register shift on every cycle !!





Shift t_0	operation
0 0	hold
0 1	Shift right by 1
1 0	load
1 1	Shift left by 1



6.2 Counter

Seq circuit that goes through a predefined seq

~~* Seq~~

* Synchronous counters (same FF) clock is

* Asynch counters (Ripple counters)

