Birla Institute of Technology & Science, Pilani
Hyderabad Campus

Innovate achieve lead

CS/ECE/EEE/INSTR F215:Digital Design

Lecture 27: *Design of clocked sequential circuits_2*

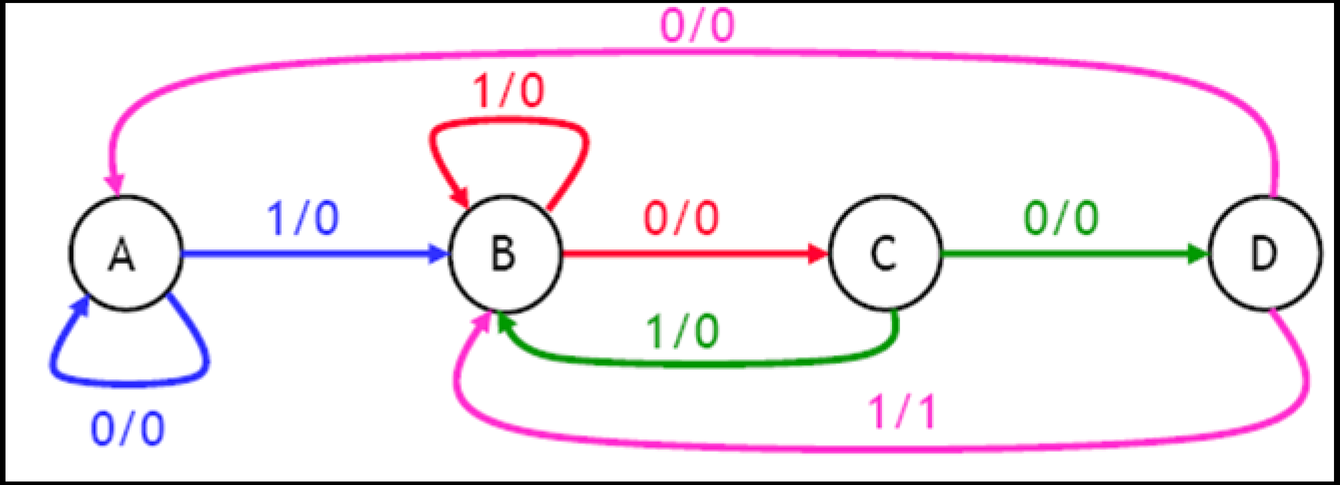
Sat, 13 Nov 2021

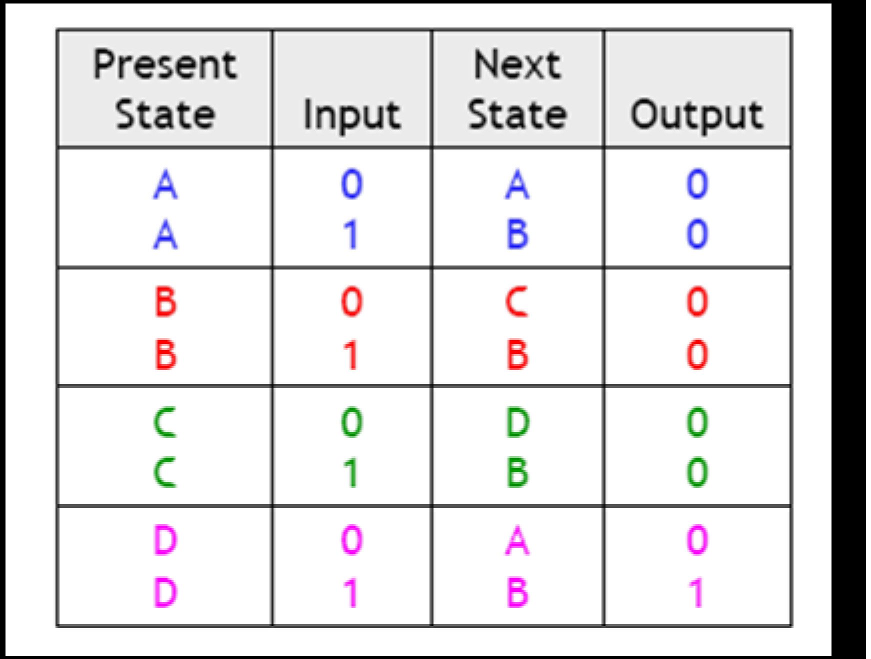
BITS Pilani
Hyderabad Campus

Dr. R. N. Ponnalagu, EEE

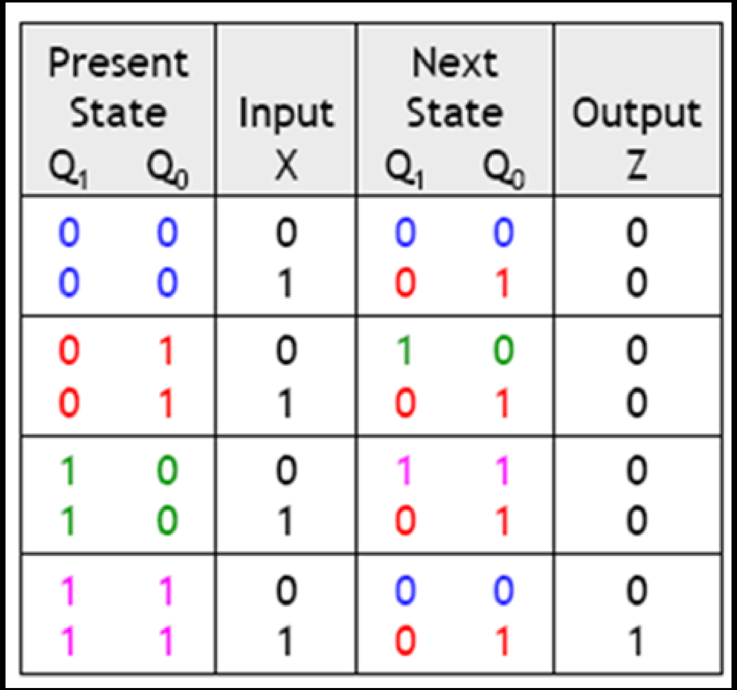
If you have choice
choose the best,
If you have No choice
do the best

➢To detect a pattern 1001

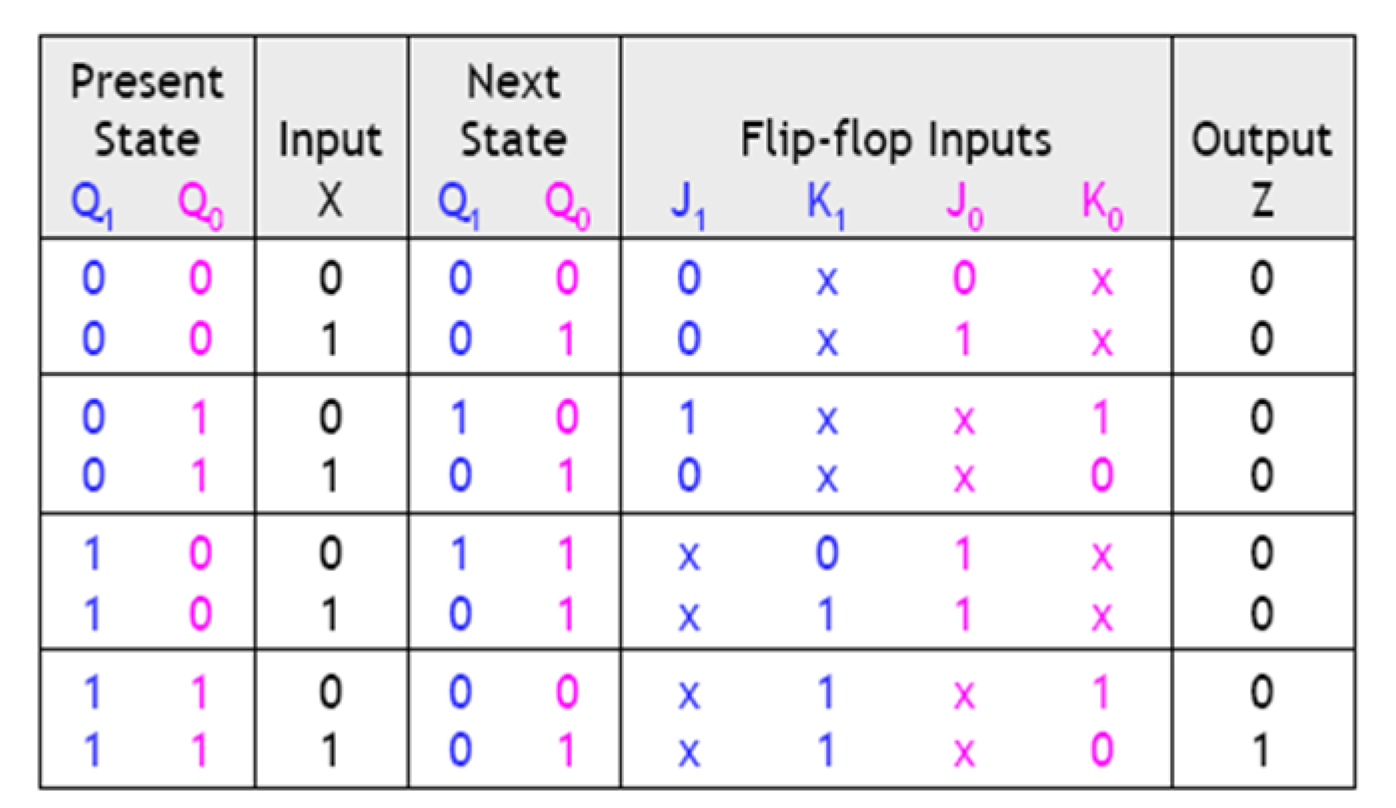




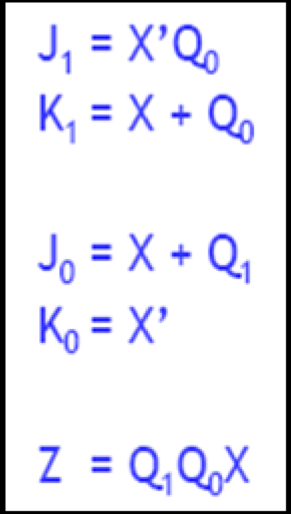
Present State	Input	Next State	Output
A	0	A	0
A	1	B	0
B	0	C	0
B	1	B	0
C	0	D	0
C	1	B	0
D	0	A	0
D	1	B	1



Present State		Input	Next State		Output
Q_1	Q_0	X	Q_1	Q_0	Z
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	0	1	0
1	1	0	0	0	0
1	1	1	0	1	1



Present State		Input	Next State		Flip-flop Inputs				Output
Q_1	Q_0	X	Q_1	Q_0	J_1	K_1	J_0	K_0	Z
0	0	0	0	0	0	x	0	x	0
0	0	1	0	1	0	x	1	x	0
0	1	0	1	0	1	x	x	1	0
0	1	1	0	1	0	x	x	0	0
1	0	0	1	1	x	0	1	x	0
1	0	1	0	1	x	1	1	x	0
1	1	0	0	0	x	1	x	1	0
1	1	1	0	1	x	1	x	0	1



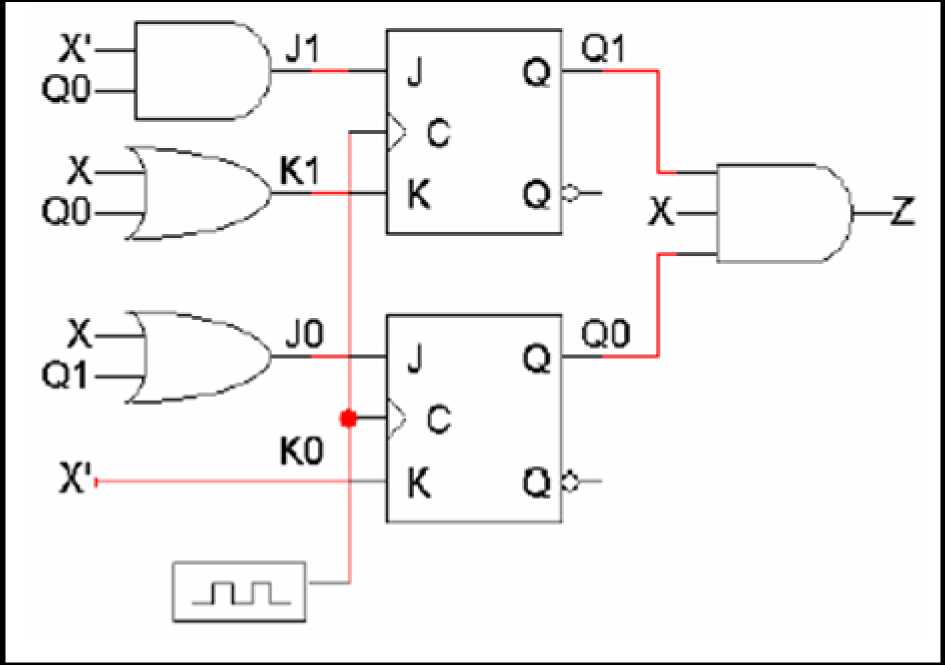
$J_1 = X'Q_0$

$K_1 = X + Q_0$

$J_0 = X + Q_1$

$K_0 = X'$

$Z = Q_1Q_0X$



Building the same circuit with D flip-flops

- What if you want to build the circuit using D flip-flops instead?
- We already have the state table and state assignments, so we can just start from Step 3, finding the flip-flop input values.
- D flip-flops have only one input, so our table only needs two columns for D_1 and D_0.

Present State		Input	Next State		Flip-flop Inputs		Output
Q_1	Q_0	X	Q_1	Q_0	D_1	D_0	Z
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	1	0	1	0	0
0	1	1	0	1	0	1	0
1	0	0	1	1	1	1	0
1	0	1	0	1	0	1	0
1	1	0	0	0	0	0	0
1	1	1	0	1	0	1	1

Finding equations (Step 4)

- If you use K-maps again, you should find the following equations.

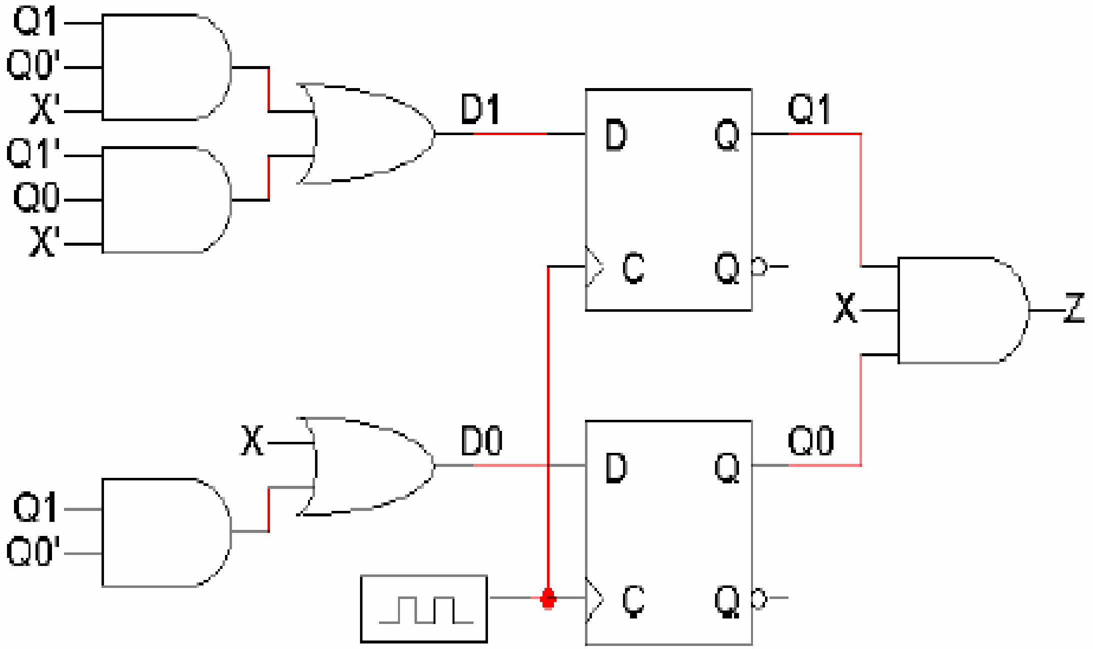
$$\underline{D_1 = Q_1Q_0'X' + Q_1'Q_0X'}$$

$$\underline{D_0 = X + Q_1Q_0'}$$

$$Z = Q_1Q_0X$$

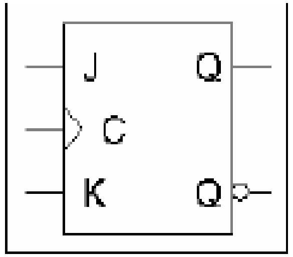
Present State		Input	Next State		Flip-flop Inputs		Output
Q_1	Q_0	X	Q_1	Q_0	D_1	D_0	Z
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	1	0	1	0	0
0	1	1	0	1	0	1	0
1	0	0	1	1	1	1	0
1	0	1	0	1	0	1	0
1	1	0	0	0	0	0	0
1	1	1	0	1	0	1	1

Building the circuit (Step 5)

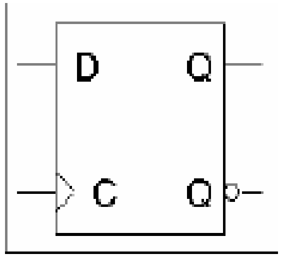


Flip-flop comparison

- JK flip-flops are good because there are many don’t care values in the flip-flop inputs, which can lead to a simpler circuit.



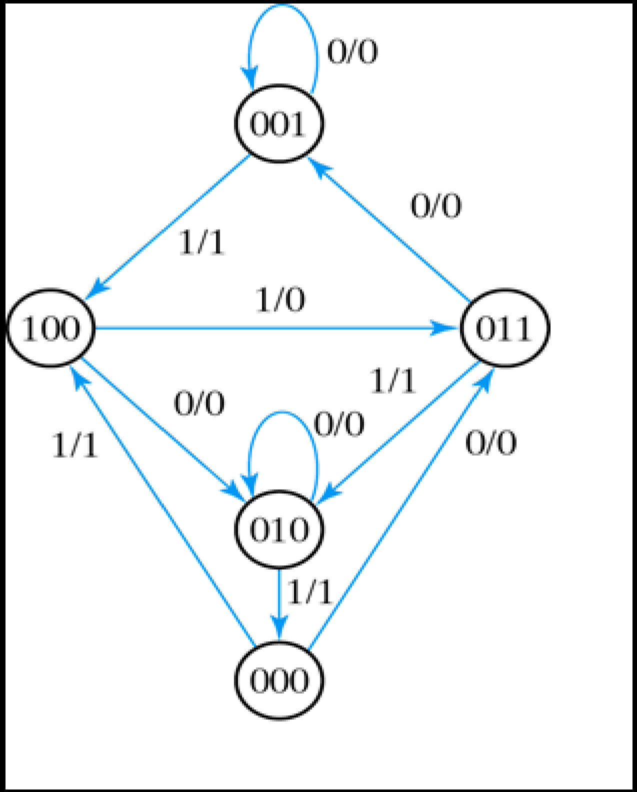
- D flip-flops have the advantage that you don’t have to set up flip-flop inputs at all, since Q(t+1) = D. However, the D input equations are usually more complex than JK input equations.



- In practice, D flip-flops are used more often.
  - There is only one input for each flip-flop, not two.
  - There are no excitation tables to worry about.
  - D flip-flops themselves are simpler to implement than JK flip-flops.

DESIGN WITH UNUSED STATES

- **Design the circuit using ~~D~~ T Flip Flops**



P.S			I/P	N.S			O/P	ff i/ps		
A	B	C	x	A	B	C	y	TA	TB	TC
0	0	0	0	0	1	1	0	0	1	1
0	0	0	1	1	0	0	1	1	0	0
0	0	1	0	0	0	1	0	0	0	0
0	0	1	1	1	0	0	1	1	0	1
0	1	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	1	0	1	0
0	1	1	0	0	0	1	0	0	1	0
0	1	1	1	0	1	0	1	0	0	1
1	0	0	0	0	1	0	0	1	1	0
1	0	0	1	0	1	1	0	1	1	1

PS				NS						
A	B	C	X	A	B	C	T ~~D_a~~	T ~~D_b~~	T ~~D_c~~	Y
0	0	0	0	0	1	1				0
0	0	0	1	1	0	0				1
0	0	1	0	0	0	1				0
0	0	1	1	1	0	0				1
0	1	0	0	0	1	0				0
0	1	0	1	0	0	0				1
0	1	1	0	0	0	1				0
0	1	1	1	0	1	0				1
1	0	0	0	0	1	0				0
1	0	0	1	0	1	1				0

PS				**NS**						
A	**B**	**C**	**X**	**A**	**B**	**C**	**D_A**	**D_B**	**D_C**	**Y**
0	0	0	0	0	1	1	0	1	1	0
0	0	0	1	1	0	0	1	0	0	1
0	0	1	0	0	0	1	0	0	1	0
0	0	1	1	1	0	0	1	0	0	1
0	1	0	0	0	1	0	0	1	0	0
0	1	0	1	0	0	0	0	0	0	1
0	1	1	0	0	0	1	0	0	1	0
0	1	1	1	0	1	0	0	1	0	1
1	0	0	0	0	1	0	0	1	0	0
1	0	0	1	0	1	1	0	1	1	0

$T_A = A + \bar{B}X$

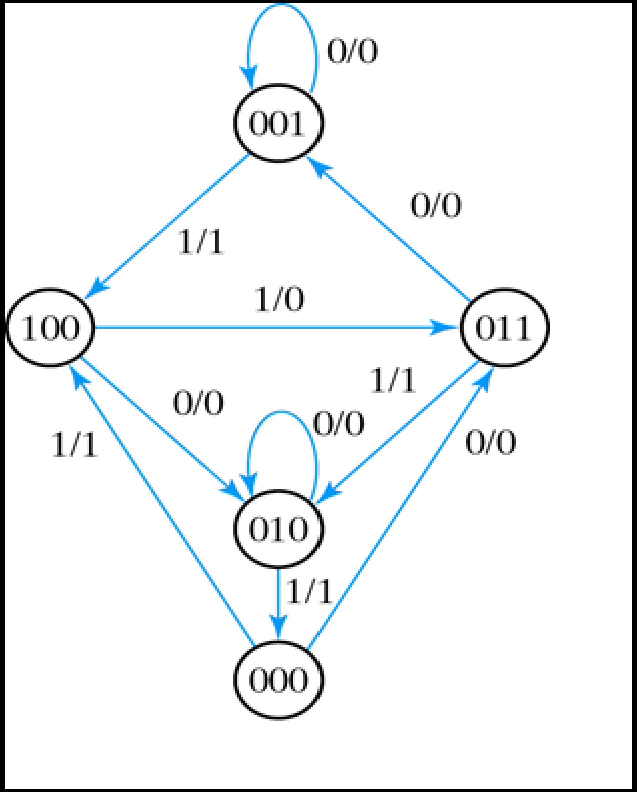
$T_B = A + \bar{B}\bar{C}X$

AB \ CX	00	01	11	10
00	0	1	1	0
01	0	0	0	0
11	X	X	X	X
10	0	0	X	X

AB \ CX	00	01	11	10
00	1			
01	1		1	
11	X	X	X	X
10	1	1	X	X

AB \ CX	00	01	11	10
00	1			1
01				1
11	X	X	X	X
10		1	X	X

AB \ CX	00	01	11	10
00		1	1	
01		1	1	
11	X	X	X	X
10			X	X



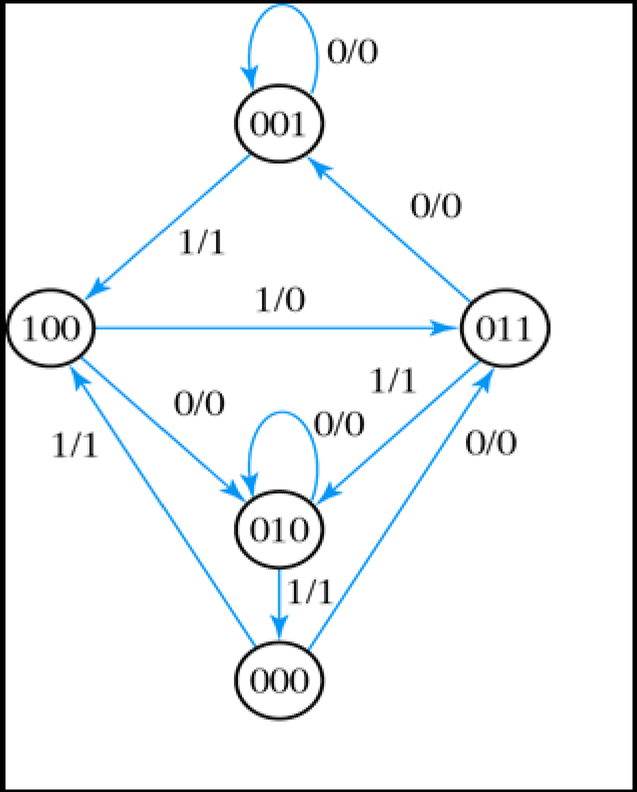
Answer:

D_a=A’B’x

D_b=C’x’+A+BCx

D_c=A’B’x’+Cx’+Ax

Y=A’x



Answer:

D_a=A’B’x

D_b=C’x’+A+BCx

D_c=A’B’x’+Cx’+Ax

Y=A’x

The circuit is self correcting

PS				NS						
A	B	C	X	A	B	C	T_a	T_b	T_c	Y
0	0	0	0	0	1	1				0
0	0	0	1	1	0	0				1
0	0	1	0	0	0	1				0
0	0	1	1	1	0	0				1
0	1	0	0	0	1	0				0
0	1	0	1	0	0	0				1
0	1	1	0	0	0	1				0
0	1	1	1	0	1	0				1
1	0	0	0	0	1	0				0
1	0	0	1	0	1	1				0

Answer:

T_a=A+B’x

T_b=B’C’x’+A+BCx’ +BC’X

T_c=A’B’C’x’+Cx+Ax

Y=A’x

000
001
010
011
100
used states

101
110
111
unused states

Self correcting

A B C	x	TA	TB	TC	N.S A B C
101	0	1	1	0	0 1 1
	1	1	1	1	0 1 0
101	0				
	1				
110	0				
	1				
110	0				
	1				
111					

011
010

~~011~~ 010

~~010~~ 011
010

Flip flop Conversions

SR
D
JK
T

SR to $\boxed{JK}$

Char. table

J	K	Q_t	$Q_{(t+1)}$	S	R
0	0	0	0	0	X
0	0	1	1	X	0
0	1	0	0	0	X
0	1	1	0	0	1
1	0	0	1	1	0
1	0	1	1	X	0
1	1	0	1	1	0
1	1	1	0	0	1

Excitation table

S

J \ KQ_t	$\bar{K}\bar{Q}_t$	$\bar{K}Q_t$	KQ_t	$K\bar{Q}_t$
$\bar{J}$	0	X	0	0
J	1	X	0	1

$S = J\bar{Q}_t$

R

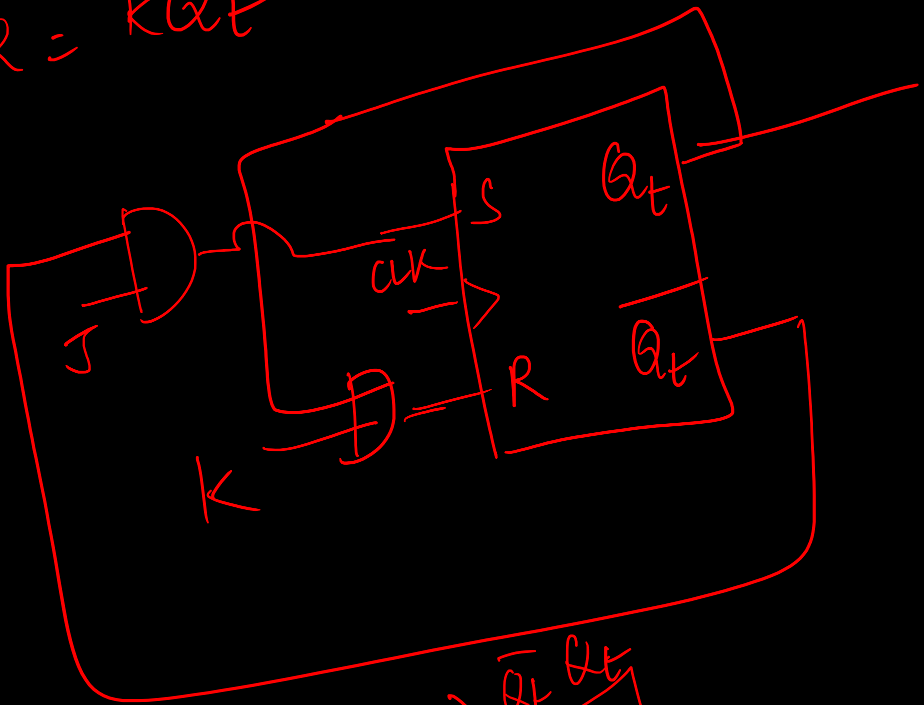
	X	0	1	X
	0	0	1	0

$R = KQ_t$

SR to JK

$S = J\overline{Q_t}$

$R = KQ_t$



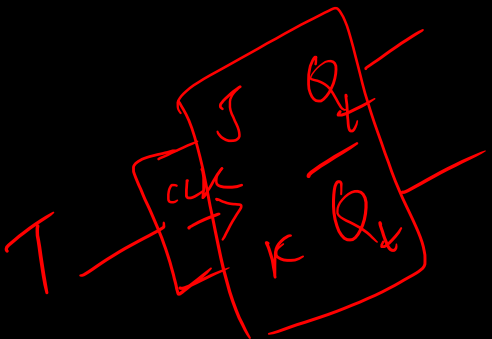
	$\overline{Q_t}$	Q_t
$\overline{T}$	0	x
T	1	x

JK to T

T	Q_t	Q_{t+1}	J	K
0	0	0	0	x
0	1	1	x	0
1	0	1	1	x
1	1	0	x	1

$J = T$

$K = T$



Registers

- Register is an extension of a Flip-flop that can store multiple bits

Register consists of group of flip-flops and some combinational gates. Each FF is capable of storing one bit of information.

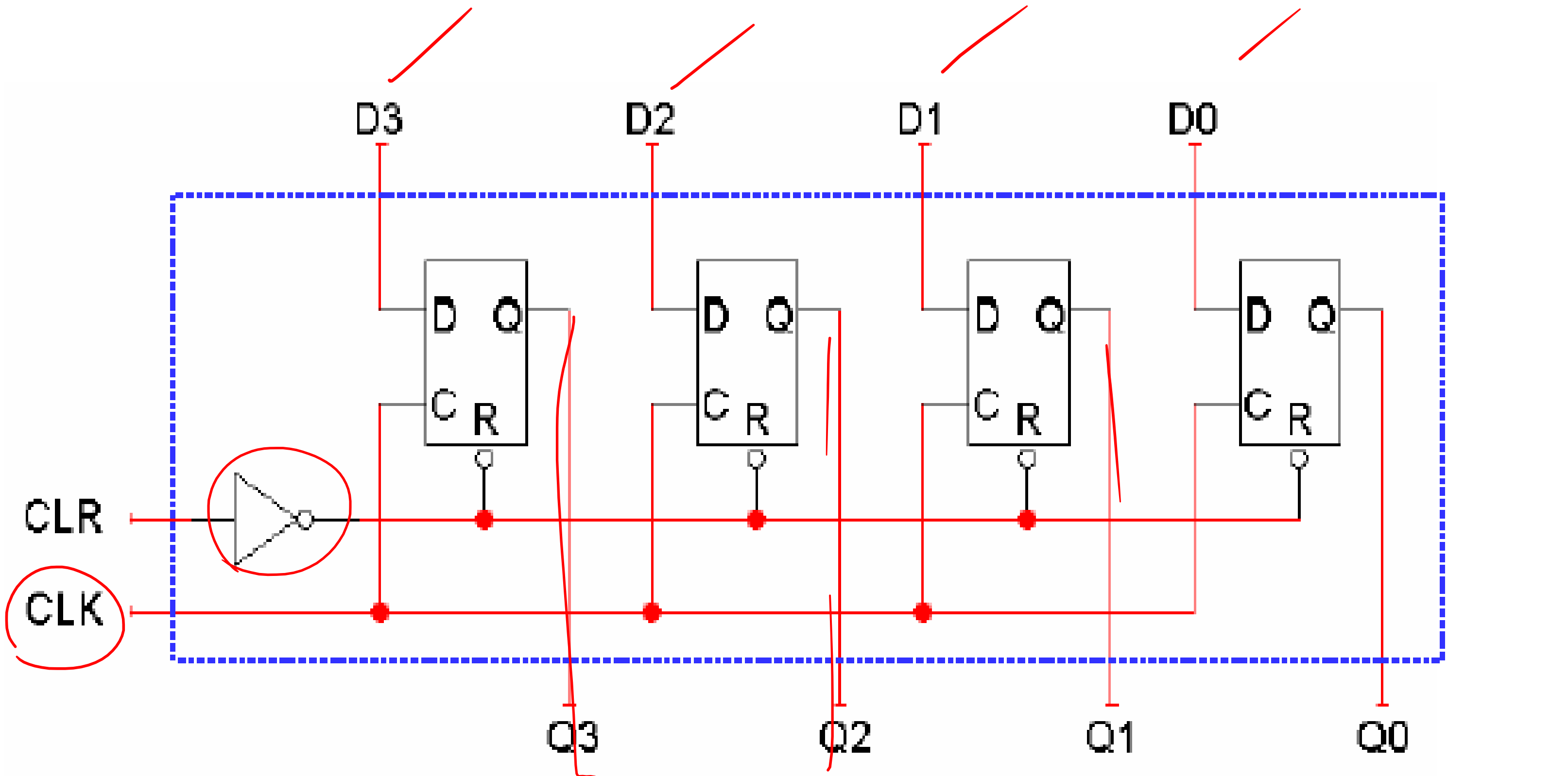
N-bit register has group of n-FFs capable of storing n-bits of binary information. The logic gates determine how the information is transferred into the register.

- Ex: Used as a Temporary storage in Microprocessors

A basic register

An example: Register constructed using D FFs.

D0-D3 Data inputs; Q0-Q3 data output.


D3
D2
D1
D0
D
Q
C
R
CLR
CLK
Q3
Q2
Q1
Q0