DHD- Mark

① Given is Circuit:

(i) Find eq^n of o/p of FF & i/p FF

(ii) Make TT of PS, i/p, FF i/p, NS

(iii) Make state table w.r.t PS, NS & how it changes w.r.t i/p

(iv) Make State Diagram

② There are two types of ques:

(1) Circuit → eq^ → table → diagram

(ii) ~~Given → table →~~

(4) Given → diagram → table → K-map → Circuit ↵

Steps for Designing a Finite State Machine

1. Define the purpose of the Machine in simple words.
   It should be realizable using finite number of memory elements.

2. Draw a State Transition Diagram

3. Draw a State Table

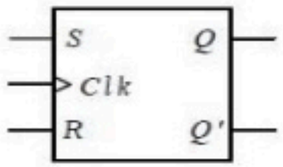
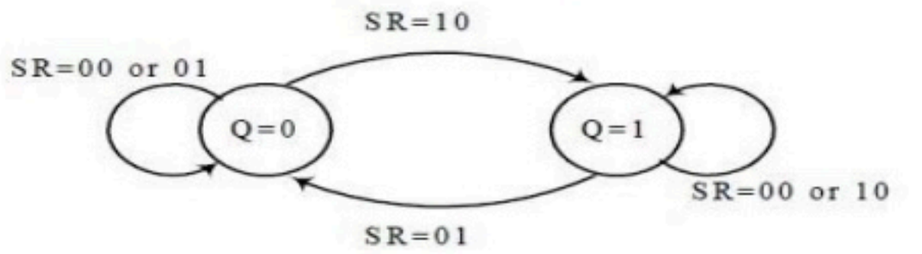
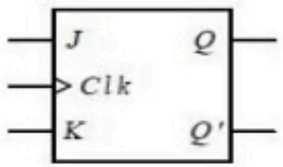
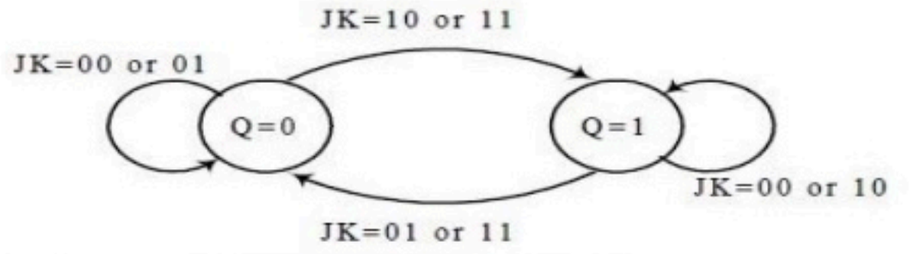
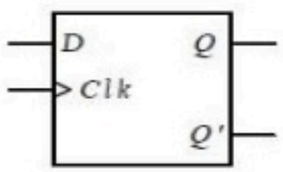
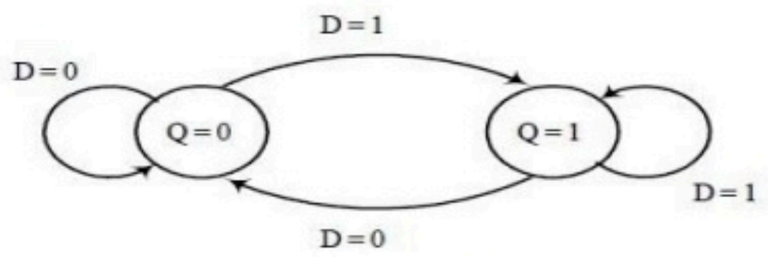
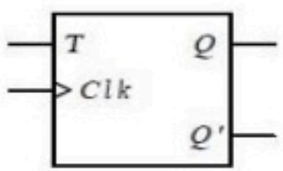
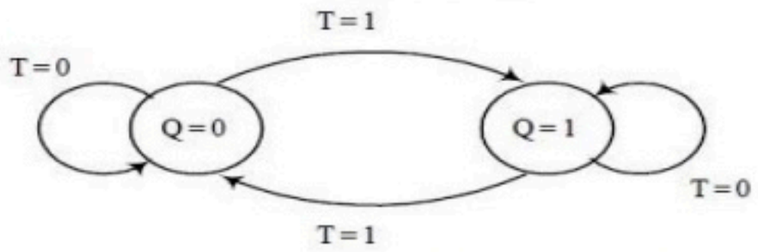
4. Remove the redundant states

5. Select the Type of Flip-Flop (D, T, JK or SR) and draw the excitation table

6. Find the Boolean Function for each flip-flop in terms of state variables and inputs.
In a same manner, also fine the output Boolean Function.

7. Draw a Logic circuit

ALL ABOUT ELECTRONICS

Name / Symbol	Characteristic (Truth) Table	State Diagram / Characteristic Equations	Excitation Table
SR  	S R Q Q_{next} 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 ×	 $Q_{next} = S + R'Q$ $SR = 0$	Q Q_{next} S R 0 0 0 × 0 1 1 0 1 0 0 1 1 1 × 0
JK  	J K Q Q_{next} 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	 $Q_{next} = J'K'Q + JK' + JKQ'$ $= J'K'Q + JK'Q + JK'Q' + JKQ'$ $= K'Q(J'+J) + JQ'(K'+K)$ $= K'Q + JQ'$	Q Q_{next} J K 0 0 0 × 0 1 1 × 1 0 × 1 1 1 × 0
D  	D Q Q_{next} 0 × 0 1 × 1	 $Q_{next} = D$	Q Q_{next} D 0 0 0 0 1 1 1 0 0 1 1 1
T  	T Q Q_{next} 0 0 0 0 1 1 1 0 1 1 1 0	 $Q_{next} = TQ' + T'Q = T \oplus Q$	Q Q_{next} T 0 0 0 0 1 1 1 0 1 1 1 0

PS	X=0	X=1	Z
a	g	c	0
b	f	h	0
c	e	d	1
d	a	c	0
e	c	a	1
f	f	b	1
g	a	c	0
h	c	g	1

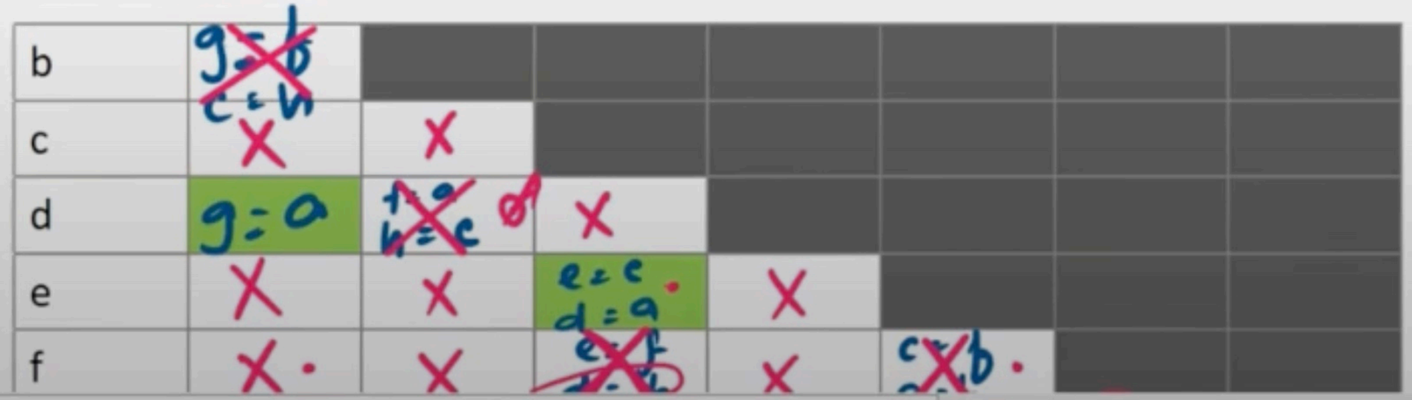
Row

Col.

STEPS TO FILL THE SQUARES:

1. Place cross in squares whose states have different outputs
2. Starting from top left square, write the pair of implied states
3. Now scan each implied pair. If a cross is place on the square corresponding to that pair, then place a 'X' on this square too.
4. On completion of the above steps, the squares without a cross indicate equivalent states.

Implication Table:



(a,c) (b,c) (d
(a,e) (b,e) (d
(a,f) (b,f) (d
(a,h) (b,h) (d

15:07 / 16:50

Date .....

① Implication Chart:

(i) Columns = no. of states - 1 ⇒ starts from 1st state ⇒ goes till 2nd last

(ii) Rows = " " " ⇒ " " 2nd " ⇒ " " last

(iii) Compare column wise/ rows ⇒ if o/p = 0 ⇒ put cross "X"

(iv) For un-crossed boxes, write possible equivalent NS for PS, do it vertically

(v) For each case, check for potential crosses "X", if there is one (in column x row wise) then cross out the whole box

(vi) Ensure once again after you're done once

(vii) Finally for un-crossed, write (column, row) for each, they're the equal/ redundant cases & can be removed

(viii) Make state table again & remove the unnecessary states while also replacing the states correctly.

① Partitioning Method:

(i) Write P_1 = all Present State

(ii) " P_2 = (group 1 w/ same o/p(s)) (group 2 w/ same o/p(s))

(iii) Check P_2 for $x=0$ & $x=1$, if o/p(s) of group 1,2 belong to the same group ⇒ nothing happens, but if an output belongs to the other group, that becomes a group on its own & hence P_3 is obtained.

(iv) Process is repeated till no further partitioning can be done, ~~that~~ and the groups that remain are equal

(v) Make state table again, removed redundant/equal states as found in above step.

Date .....

(1)

PS	NS n=0	NS n=1	O/P
A	D	C	0
B	F	H	0
C	E	D	1
D	A	E	0
E	C	A	1
F	F	B	1
G	B	H	0
H	C	G	1

$P_1 = A\,B\,C\,D\,E\,F\,G\,H$

$P_2 = (A\,B\,D\,G)$ (n=0) → A (B) D G | C E F H ; D F A B | E C F C

$(C\,F\,F\,H)$ (n=1) → A B D G | C E F H ; C H E H | D A B G

$P_3 = (A\,D\,G)\,(C\,E\,F\,H)\,(B)$

n=0 → A D (G) | C E F H | B ; D A B | E C F C | F

n=1 → A D G | C E (F) H | B ; C E G | D A B G | F

~~$P_4 = (AD)\,(G)\,(CEFH)\,(B)$~~

$P_4 = (AD)\,(G)\,(CEH)\,(B)\,(F)$

n=0 → A D | C E (H) | G | B | F ; D A | C | B | F | F

n=1 → A D | C E (H) ; C E | D A S

$P_5 = (AD)\,(CE)$

n=0 → A D | C E ; D A | E C

n=1 → A D | C E ; C E | D A

∴ A = D, C = E

P.S	N.S n=0	N.S n=1	O/P
A	A	C	0
B	F	H	0
C	C	A	1
F	F	B	1
G	B	H	0
H	C	G	1

Ans.

	A	B	C	D	E	F	G
B	D=F X C=H						
C	X	X					
D	D=A C=E	F=A X H=E	X				
E	X	X	E=C D=A	X			
F	X	X	E=F X D=B	X	C=F X A=B		
G	D=G X C=H	F=B X	X	A=B X E=H	X	X	
H	X	X	E=C D=G X	X	A=G X	F=C X B=G	X

(A, D) (C, E) — Same Latest

Example of PARTITIONING & IMPLICATION CHART

Spiral

① Direct Method (Counter):

(i) Identify the seq.

(ii) no. of flipflops = n $\Rightarrow$ $2^{n-1} \geq$ no. of bits in seq

(iii) Assign Q_0 vertically the seq

(iv) " $Q_2 Q_1$ such that all are unique, and rest should be "don't care"

(v) Make table for counting, assign flipflops, get eq^{n}(s)

(vi) Make circuit

(vii) You get "Synchronous Counter"

(2) Indirect Method:

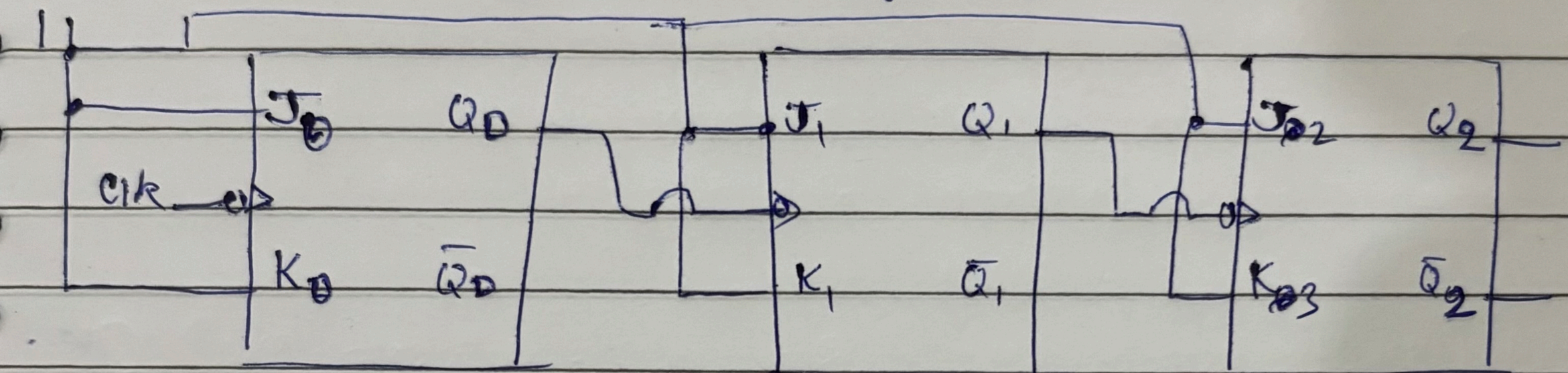
(i) Identify the seq.

(ii) Let o/p "f" be the seq. ⇒ cosider rest as "don't care"

(iii) Give unique states Q_2 Q_1 Q_0

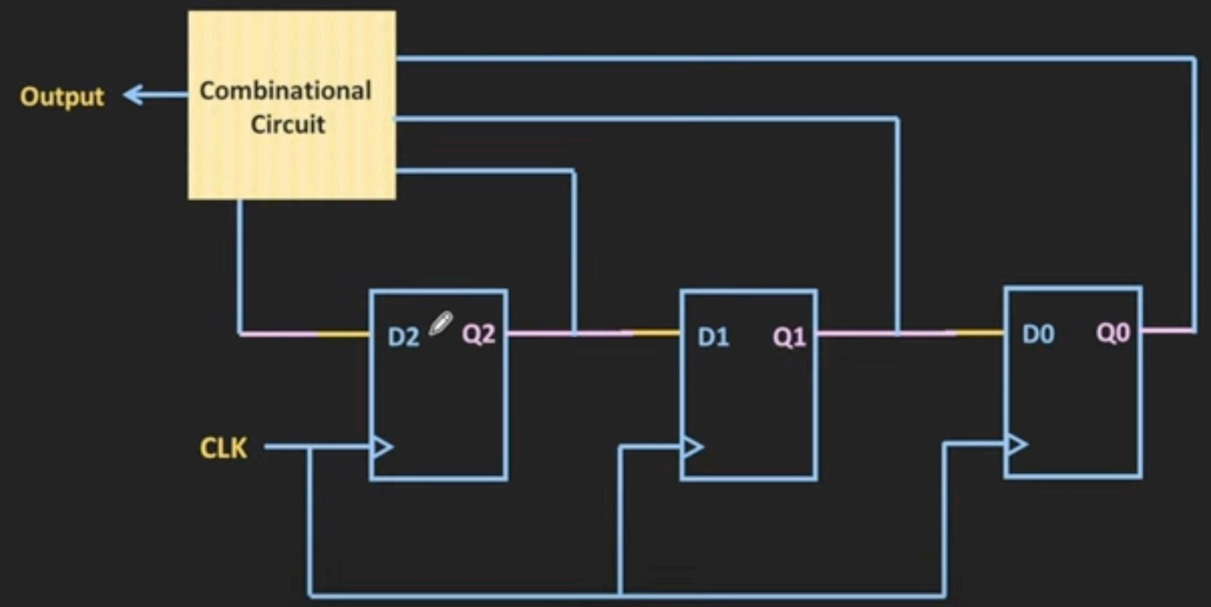
(iv) Get eqn of "f"

(v) Use "Ripple Counter" (Asynchronous)



Sequence Generator

F	Q2	Q1	Q0
1	1	0	1
1	1	1	0
0	1	1	1
1	0	1	1



ALL ABOUT ELECTRONICS

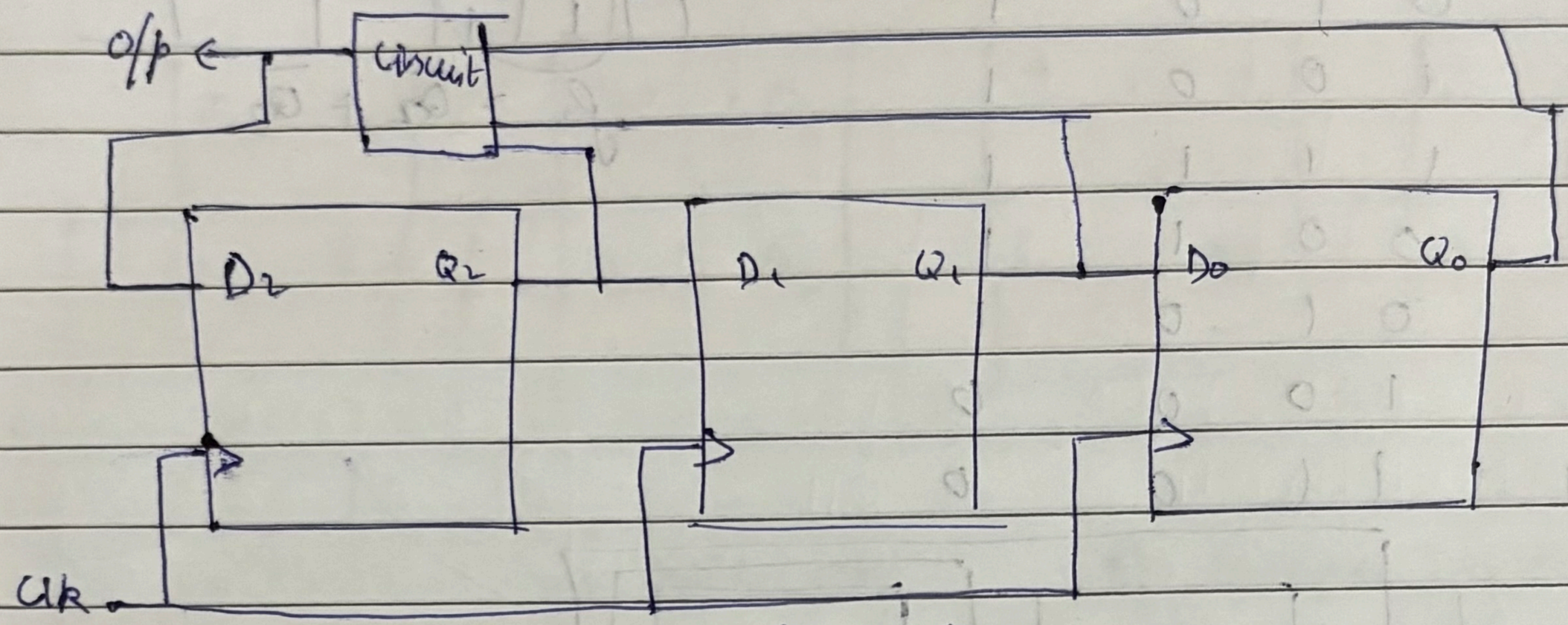
① Shift Registers:

(i) Find no. of ff ⇒ L (length of i/p stream) $\leq 2^n - 1$

(ii) Write Q_2 vertically ⇒ stream given

(iii) Write Q_1 & Q_0 such that from top, the bits go diagonally down & last bit is starting bit of Q_1 or Q_0

(iv) Same goes for o/p "f"



General Circuit

① 1 0 1 1 1 1 0 ⇒ seq = 1011110

length = L = 7

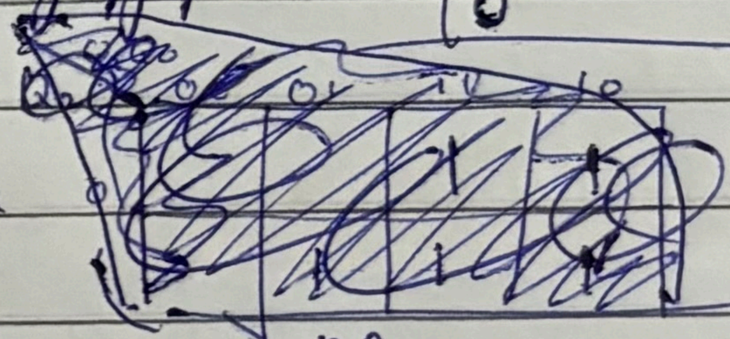
$L \leq 2^n - 1$

$7 \leq 2^3 - 1$

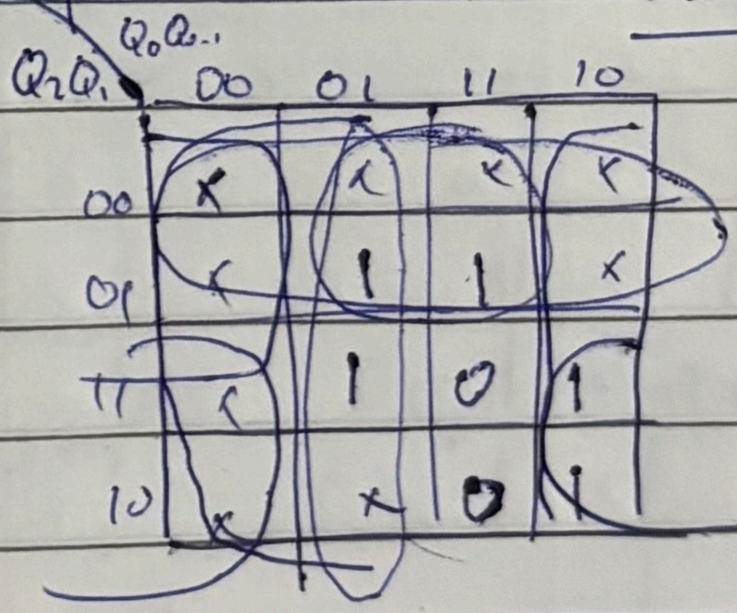
$n = 3$ ⇒ 3 flipflops

$f = \bar{Q}_2 + \bar{Q}_0 + \bar{Q}_0 Q_{-1}$

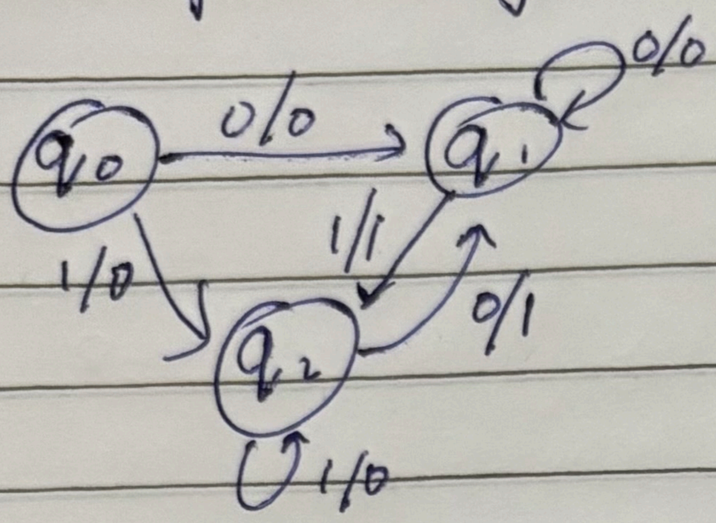
f	Q_2	Q_1	Q_0	Q_{-1}
0	1	0	1	1
1	0	1	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0
0	1	1	1	1
1	0	1	1	1



Just make like the circuit above



① Conversion from Mealy to Moore :

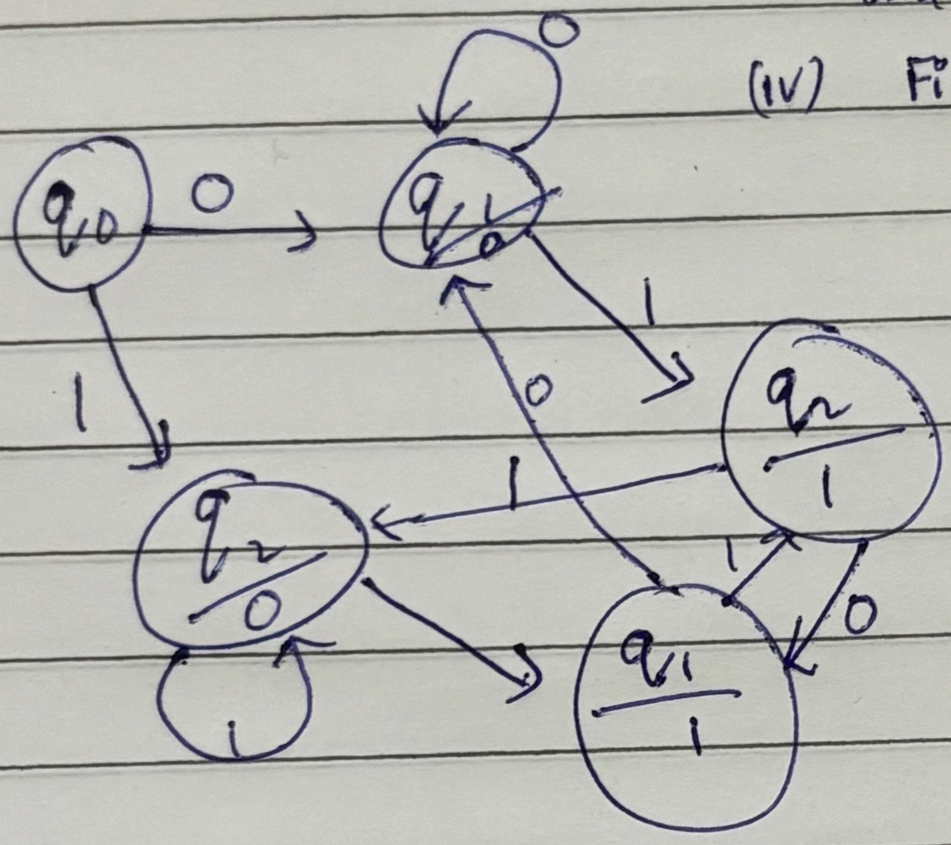


(i) Start from 1st state, and fill for 0 & 1

(ii) When it makes a transition to next state, if o/p is "0", write "0" under next state for Moore

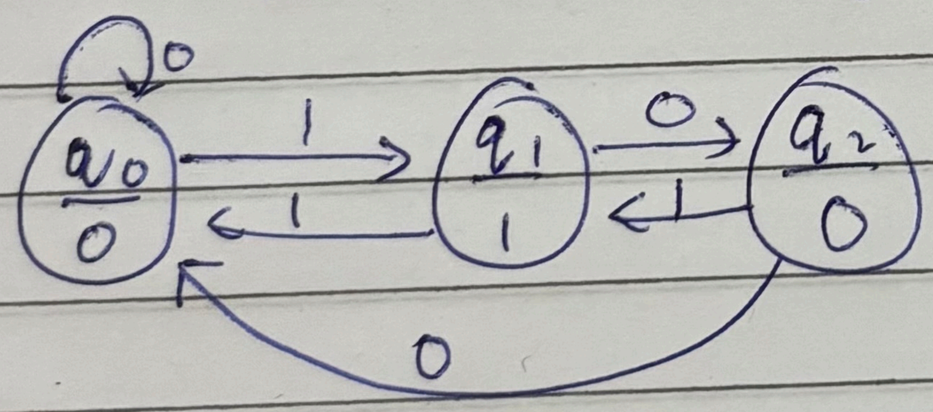
(iii) If o/p doesn't match from previous design, just add another state w/ same ~~the~~ name

(iv) Fill $n=0$ & $n=1$ for all states

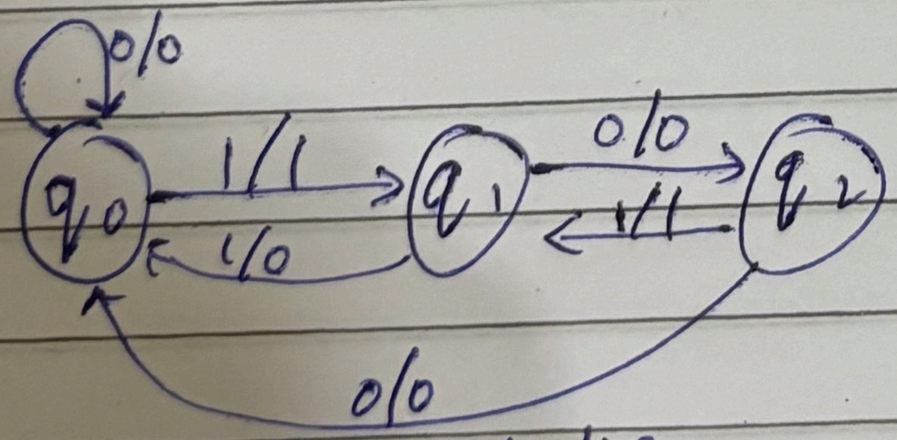


Imp. n state ~~Moore~~ Mealy machine w/ n o/p(s) can give $n \times n$ state Moore machine

② Conversion from Moore to Mealy :



(i) Start from state 1st, and do for 0, 1 while writing the o/p w/ a slash (/) for the state (ii) you're jumping to



Imp. n state in o/p Moore Machine $\Rightarrow$ "n" state Mealy Machine

① Either do it directly from diagram OR make transition table & then make the diagram

① JK FF:

$$A_{t+1} = J\bar{Q} + \bar{K}Q$$

J	K	Q_n	Q_{n+1}
0	x	0	0
1	x	0	1
x	1	1	0
x	0	1	1

② D FF: $Q_{t+1} = D$

D	Q_n	Q_{n+1}
0	0	0
1	0	1
0	1	0
1	1	1

③ SR FF:

$$A_{t+1} = S + R'Q$$

S	R	Q_n	Q_{n+1}
0	x	0	0
1	0	0	1
0	1	1	0
x	0	1	1

④ T-FF: $Q_{t+1} = \bar{T}Q + \bar{Q}T$ $= T \oplus Q$

T	Q_n	Q_{n+1}
0	0	0
1	0	1
1	1	0
0	1	1

POSSESSION OF MOBILES IN EXAM IS UFM PRACTICE

Name. Anurag Dwivedi Enrollment No. 9922102042

Jaypee Institute of Information Technology, Noida
T1 Examination, Even Semester 2025
B. Tech. VIth Semester

Course Title: Digital Hardware Design **Maximum Time: 1 hr**
Course Code: 17B1NEC741 **Maximum Marks: 20**

CO1	Design synchronous circuits using Finite State Machine approach.
CO2	Design and analyze asynchronous circuits.
CO3	Understand the advanced adders and multiplier circuit.
CO4	Apply the concept of different ways of pulse or pattern generation.
CO5	Design digital circuits using VHDL.

Note: Attempt all the questions.

Q1. Give short answers for the following:

a) What is the primary difference between Mealy and Moore state machine.
b) How does state minimization improve FSM efficiency?
c) What are the advantages of using One-hot encoding in FSM design?
d) Name the 2 methods used for state minimization in FSM.

[CO1 (Analyzing), 1x4 = 4 Marks]

Q2. Give short answers for the following:

a) How is a shift register used in sequence generation?
b) What is the role of clock dividers in pulse train generation?
c) How does a pulse train generator produce a specific pattern?
d) What is the function of a feedback circuit in pulse generation?

[CO4 (Analyzing), 1x4 = 4 Marks]

Q3. An FSM has the following state transitions:

Present State (PS)	Next State (NS) for W=0	Next State (NS) for W=1	Output (Z)
A	B	C	1
B	D	F	1
C	F	E	0
D	B	G	1
E	F	G	0
F	E	D	0
G	F	G	0

a) Use the Implication Chart method and determine the equivalent states.
b) Merge the equivalent states found in (a) and construct the reduced state table.
c) Calculate the minimum number of flip flops required to design the reduced state table.

[CO1 (Analyzing), 3+1+1 = 5 Marks]

Q4. A shift register is used to generate a **5-bit** pulse train with the sequence **10110.**

a) Determine the minimum number of flip flops required to generate the given sequence.
b) Construct the state transition table for implementing the given sequence using the shift register.
c) Derive the state equations for the shift register using D-Flip Flop.

[CO4 (Analyzing), 1+2+2 = 5 Marks]

Q5. Design a Moore machine to generate 1's compliment of a binary number 1011. Draw the state diagram only.

[CO1 (Analyzing), 2 Marks]

- **Q4.** A circuit is given below in Fig. 2. There are 4 input ports a, b, d, e and one output port g of STD_LOGIC type. The instances U1 and U2 of the two AND gates are connected to the U3 OR instance using c and f wires. Write the VHDL code for this circuit using structural (entity-Architecture) design style only for top-level entity-architecture. The component has to be declared and instantiated in the main code of the architecture.

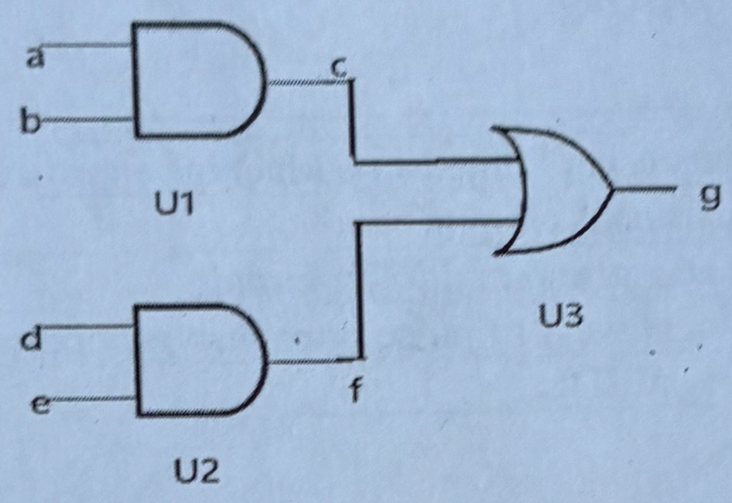


Fig. 2

[CO5 (Analyzing), 5 Marks]

Q5. Write a behavioral level code to implement a 4-bit synchronous binary counter. Consider the increment in count at the positive edge of the clock.

[CO5 (Analyzing), 4 Marks]

POSSESSION OF MOBILES IN EXAM IS UFM PRACTICE

Name. Anurag Dwivedi Enrollment No. 9922102048

Jaypee Institute of Information Technology, Noida
T2 Examination, Even Semester 2025
B. Tech. VIth Semester

Course Title: Digital Hardware Design **Maximum Time: 1 hr**
Course Code: 17B1NEC741 **Maximum Marks: 20**

CO1	Design synchronous circuits using Finite State Machine approach.
CO2	Design and analyze asynchronous circuits.
CO3	Understand the advanced adders and multiplier circuit.
CO4	Apply the concept of different ways of pulse or pattern generation.
CO5	Design digital circuits using VHDL.

Note: Attempt all the questions.

Q1. Give short answers for the following:

a) State the difference between an inertial and transport delay.
b) State the difference between Signal and Variable.
c) State the difference between dataflow and behavioral modeling style.
d) State the difference between EXIT and NEXT command.
e) Write the syntax for WHEN-ELSE command.
f) Write the syntax for Generate commands.

[CO5 (Analyzing), 1x6=6 Marks]

Q2. A signal 'A' takes the various transitions with respect to time 't' as shown in Fig. 1 below. Another signal 'S' follows the signal 'A' after 2ns (S <= A after 2ns). Considering the inertial delay model, capture the transition of signal 'S' from one state to another state with respect to time, and plot the signal 'S' against signal 'A'.

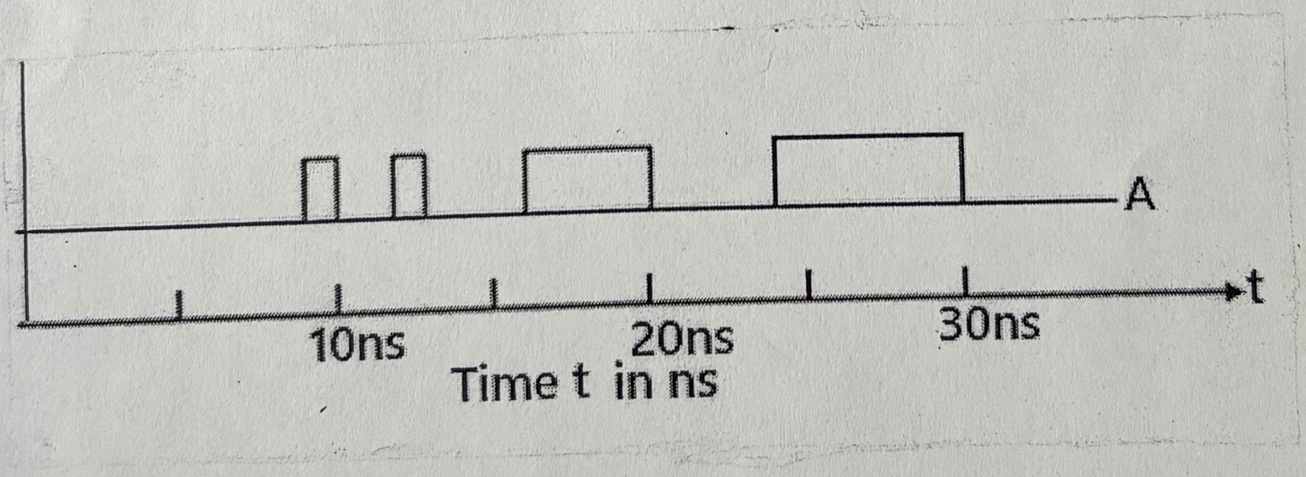


Fig. 1

[CO5 (Analyzing), 2 Marks]

Q3. A user has two sensors, namely a smoke sensor and water level detector, in the home. He has to turn on the respective alarm if any input is high. Write down the behavioral level VHDL code to implement the logic.

respective

[CO5 (Analyzing), 3 Marks]

P.T.O.....

Asynchronous Hax

① If table / circuit / word problem is given:

(i) Write eq^n(s) (Make circuit too)

(ii) Make State Table (PS, NS, Stable/Unstable, Output)

(iii) Make Transition Table like K-map ⇒

(iv) Assign states: a=00 b=01 c=11 d=10

(v) Square out the stable states in →

(vi) Make Output Table like K-map ⇒

"0" ⇒ for o/p 0, "1" ⇒ for o/p 1, "-" for nothing

(vii) Make Flow Table, basically just write o/p wrt comma in above table

	y_1y_2 00	01	11	10
x_1x_2				
00				
01				
11				
10				

② If Flow Table is given: (and you have to make circuit)

③ Race: Step 1: Find a races, if PS ≠ NS for both variables (11→00, 10→01)

Step 2: For each race cases, find by changing: (i) both y_1, y_2

(ii) y_1 ⇒ const. y_2 ⇒ change

(iii) y_2 ⇒ const. y_1 ⇒ change

[Stable states don't have race], find final stable state for all cases, if encounter unstable, take that as PS and find its NS till stable state.

Step 3: If all final stable state are equal for a given i/p ⇒ non-critical

o/w ⇒ critical

Date .....

(6)

y_1y_2 \ x	0	1
00	00	11
01	00	(01)
11	10	(11)
10	00	(10)

Step 1 : Races : y_1y_2 : 00 $\xrightarrow{x=1}$ 11 ; y_1y_2 : $\xrightarrow{}$

Step 2 : y_1y_2 $x=$ $\xrightarrow{}$ 01 $\xrightarrow{x=1}$ (01) ⇒ Find Stable State

y_1y_2 $x=1$ $\xrightarrow{}$ 10 $\xrightarrow{x=1}$ (10) ⇒ "

Step 3 : ∵ unstable final stable state ⇒ Critical

(7)

y_1y_2 \ x_1x_2	00	01	11	10
00	11	(00)	10	01
01	11	00	11	(01)
11	(11)	00	10	(11)
10	11	(10)	(10)	11

Step 1 : Find Race :

y_1y_2	x_1x_2
00	00 → 11
11	01 → 00

Step 2 : Find by changing x_1x_2 :

(A) (i) if y_1y_2 both changes :

00 $\xrightarrow{00}$ 11 $\xrightarrow{00}$ (11)

(ii) if y_1 is const. y_2 chgs :

00 $\xrightarrow{00}$ 01 $\xrightarrow{00}$ 11 $\xrightarrow{00}$ (11)

(iii) if y_2 is const. y_1 chgs :

00 $\xrightarrow{00}$ 10 $\xrightarrow{00}$ 11 $\xrightarrow{00}$ (11)

⇒ Same → ⇒ non-critical

(B) (i) if y_1y_2 both chgs :

11 $\xrightarrow{01}$ 00 $\xrightarrow{01}$ (00)

(ii) if y_1 is const. & y_2 chgs :

11 $\xrightarrow{01}$ 10 $\xrightarrow{01}$ (10)

(iii) if y_2 is const. & y_1 chgs :

11 $\xrightarrow{01}$ 01 $\xrightarrow{01}$ (00)

⇒ Not same ⇒ critical

(1)

y_1y_2 \ x	00	01	11	10
00	10	(00)	(00)	11
01	(01)	00	10	(01)
11	01	(11)	(11)	10
10	(10)	(10)	00	(10)

Races : y_1y_2 : 00 $\xrightarrow{10}$ 11 ; y_1y_2 : $\xrightarrow{}$

01 $\xrightarrow{11}$ 10

Changes : (A) if both y_1y_2 chgs :

00 $\xrightarrow{10}$ 11 $\xrightarrow{10}$ 10 $\xrightarrow{10}$ (10)

if y_1 is const. y_2 chgs :

00 $\xrightarrow{10}$ 01 $\xrightarrow{10}$ (01)

if y_2 is const. y_1 chgs

00 $\xrightarrow{10}$ 10 $\xrightarrow{10}$ (10)

⇒ Critical

(B) 01 $\xrightarrow{11}$ 10 $\xrightarrow{11}$ 00 → (00)

01 $\xrightarrow{11}$ 11 $\xrightarrow{11}$ (11) →

01 $\xrightarrow{11}$ 00 $\xrightarrow{11}$ 00

⇒ Critical