UNIT- III & V

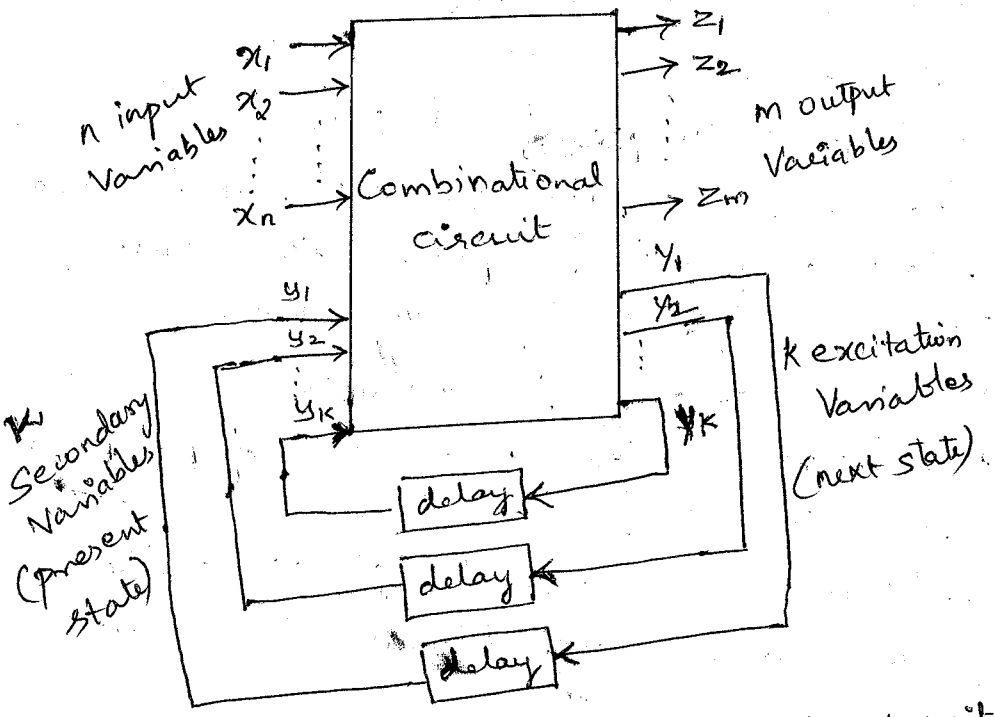
ASYNCHRONOUS SEQUENTIAL CIRCUIT.

* A sequential circuit is specified by a time sequence of inputs, Outputs and internal states.
* In asynchronous sequential circuits, donot use clock pulses. The change of internal state occurs when there is a change in input variables.
* The memory elements in asynchronous sequential circuits are either unclocked flip flops or time delay elements.
* An asynchronous sequential circuit quite often resembles a combinational circuit with feedback.
* The design of asynchronous sequential circuit is more difficult than that of synchronous circuit because of timing problem involved in feedback path.
* The asynchronous circuit doesnot use a clock, the state of the system is allowed to change immediately after the input changes.

Application

1. Used in digital system since it doesnot wait for the clock pulse, i.e speed of operation is more.
2. It is more economical since they are not going to the expense of providing a circuit for generating clock pulses.
3. Used in applications where the input signals to the system may change at any time, independently of an internal clock.

BLOCK DIAGRAM OF AN ASYNCHRONOUS SEQUENTIAL CIRCUIT



* It consist of combinational circuit and delay elements connected to form feedback loops.
* There are 'n' input Variables, 'm' Output Vaeiables and 'k' internal states.
* The present state and next state Variables in asyncheonous sequential circuit are called Secondary Variables and excitation Variables.
* When an input Variable changes in value, the 'y' secondary Variables donot change instaneously.
* It takes a certain amount of time for the signal to propagate from the input terminals, through the combinational circuit, to the 'Y' excitation Variables, which generate new values for the next state.
* These Values propagate through the delay elements and become the new present state for the secondary Variables. (ie) $y_1, y_2, \ldots y_K$.

Two Types: 1. Fundamental mode circuits
2. Pulse mode circuit.

Fundamental mode circuits assumption:

* The input variables change only when the circuit is stable.
* Only one input variable can change at a given time
* Inputs are levels and not pulses.
* Delay lines are used as a memory elements.

Pulse mode assumption.

* The input variables are pulses instead of levels.
* The width of pulses is long enough for the circuit to respond to the input
* The pulse width must not be so long that it is still present after the new state is reached.
* pulse should not occur simultaneously on two or more input lines.
* Flip flops are commonly used as a memory element.
* Memory element transitions are initiated only by input pulses.
* Input variables are used only in the uncomplemented or the complemented forms, but not both.

ANALYSIS OF FUNDAMENTAL MODE SEQUENTIAL CIRCUIT.

* The circuit has two input variables I_1 and I_0 and one output variable Z.
* has two feedback path providing input to gates creating latching and generates state variables X_0 and X_1.

$$X_1^+ = X_0 \bar{I}_1 + X_0 X_1 I_0$$
$$X_0^+ = X_0 I_0 I_1 + X_1 \bar{I}_0$$

} Next secondary state

$$Z = X_0 X_1 I_0$$ Output equation.

For the given input and Secondary state if next secondary state does not change then the state is said to be stable.

Secondary Present state x_1 x_0	Inputs I_1 I_0	Next secondary state x_1 x_0	stable state	Z
0 0	0 0	0 0	Yes	0
0 0	0 1	0 0	Yes	0
0 0	1 0	0 0	yes	0
0 0	1 1	0 0	yes	0
0 1	0 0	1 0	No	0
0 1	0 1	1 0	No	0
0 1	1 0	0 0	No	0
0 1	1 1	0 1	yes	0
1 0	0 0	0 1	Yes	0
1 0	0 1	0 0	No	0
1 0	1 0	0 1	No	0
1 0	1 1	0 0	No	0
1 1	0 0	1 1	yes	0
1 1	0 1	1 0	No	1
1 1	1 0	0 1	No	0
1 1	1 1	1 1	yes	1

Next secondary state

Secondary x_1x_0 \ Input I_1I_0	00	01	11	10
00	(00) 0	(00) 1	(00) 3	(00) 2
01	10 4	10 5	(01) 7	00 6
11	(11) 12	10 13	(11) 15	01 14
10	01 8	00 9	00 11	01 10

Circle represent stable state

unstable state

* The circle represents the stable state
* arrows indicates the transitions from unstable states to stable states

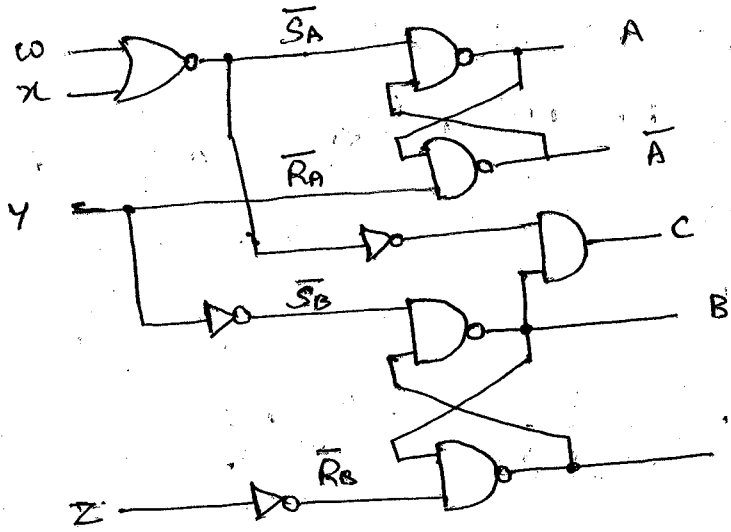
Output map

x_1x_0 \ I_1I_0	00	01	11	10
00	0 0	0 1	0 3	0 2
01	- 4	- 5	0 7	- 6
11	0 12	- 13	1 15	- 14
10	- 8	- 9	- 11	- 10

* Output is mapped for all stable states.
* For unstable states output is mapped unspecified.

ANALYSIS OF PULSE MODE SEQUENTIAL CIRCUIT.

Let us consider the asynchronous sequential circuit which is driven by the pulses. The two NAND gate latches generates the state variables A and B. The circuit has four input variables W, X, Y and Z and one output variable C.



excitation equation

$$\overline{S}_A = \overline{W+X}, \quad S_A = W+X$$
$$\overline{R}_A = Y$$
$$\overline{S}_B = \overline{Y}, \quad S_B = Y$$
$$\overline{R}_B = \overline{Z}$$

output equation

$$C = (W+X)\cdot B. \qquad \text{— (1)}$$

The characteristic equation for $\overline{S}\overline{R}$ flipflop is

$$Q_n^+ = S + \overline{R}Q_n \qquad \text{— (2)}$$

State variable next-state equation.

Sub (1) in (2). Let $Q_A = A$

$Q_B = B$.

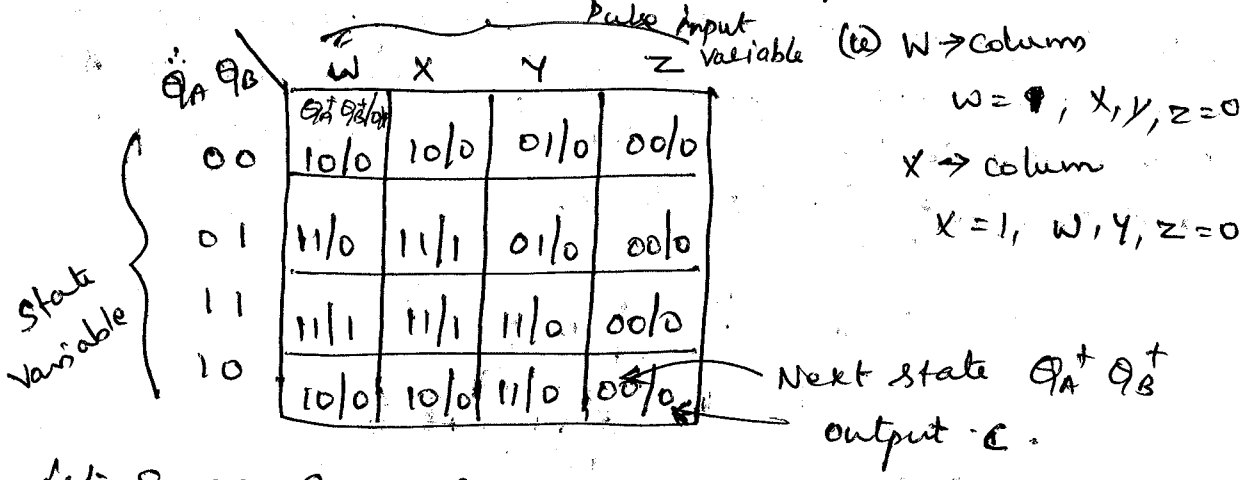
$n = A \Rightarrow \quad Q_A^+ = S_A + \overline{R}_A Q_A.$

$= W + X + Y.A \qquad \text{— (3)}.$

$n = B \Rightarrow \quad Q_B^+ = S_B + \overline{R}_B Q_B.$

$= Y + \overline{Z}B \qquad \text{— (4)}$

Transition table indicating state variable, input variables, next state value and output state.

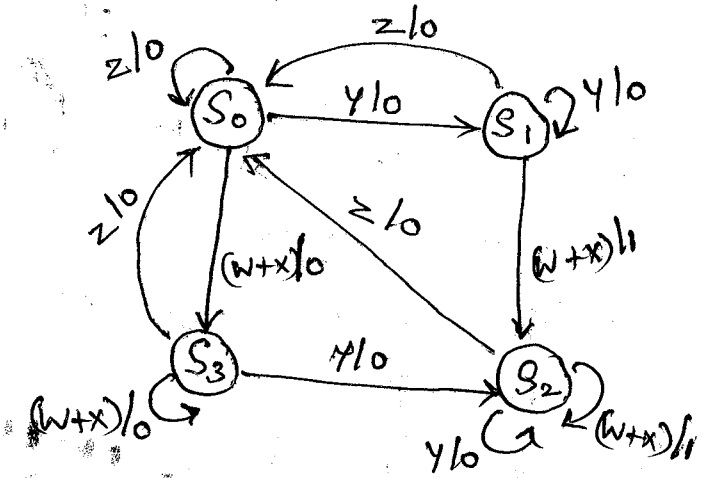


Let $S_0 = 00$, $S_1 = 01$, $S_2 = 11$, $S_3 = 10$ → State variable.

Flow table

$Q_A Q_B$	W	X	Y	Z
S_0	S_3/0	S_3/0	S_1/0	S_0/0
S_1	S_2/1	S_2/1	S_1/0	S_0/0
S_2	S_2/1	S_2/1	S_2/0	S_0/0
S_3	S_3/0	S_3/0	S_2/0	S_0/0

Flow diagram.



DESIGN OF FUNDAMENTAL MODE SEQUENTIAL CIRCUIT

Design process is exactly reverse of analysis process. We know the behaviour of the circuit and we have to develop the sequential circuit from scratch.

STEPS:

1. Construction of a primitive flow table from the problem statement. An intermediate step may include the development of a state diagram.
2. Primitive flow table is reduced by ~~eliminating~~ eliminating redundant states by using state

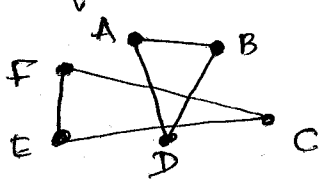
reduction techniques.
3. State assignment is made.
4. Primitive flow table is realized using appropriate logic elements.

1. DERIVATION OF PRIMITIVE FLOW TABLE.

* A primitive flow table is a special case of flow table.
* defined as a flow table which has exactly one stable state for each row in the table.
* The design process begins with the construction of primitive flow table.

2. REDUCTION OF PRIMITIVE FLOW TABLE.

* Next step is to reduce primitive flow table using state reduction technique.
* we use merger graph, it has same number of vertices as the states in the flow table.
* Two rows of primitive flow table can be merged into a single row if there are no output conflict and no state conflict in any column.
* stable state and unstable state are merged to form a stable state.



two compatible pairs
$(A, B, D) \rightarrow S_0$, $(C, E, F) \rightarrow S_1$.

3. RACE FREE STATE ASSIGNMENT

* In synchronous circuits, the state assignments are made with the objective of circuit reduction.
* In asynchronous circuits, the objective of state assignment is to avoid critical races.

Races:

* Race condition occurs in asynchronous sequential circuit when two or more binary state variables change their value in response to change in input variable.
* In unequal delays, a race condition may cause the state variables to change in an unpredictable manner.
* change in two state variable from 00 to 11.
* 1. first variable may change faster than second
  ∴ 00 to 10 then 11.
  2. second variable may change faster than first
  00 to 01 then 11.

* Noncritical race: If the final stable state changes does not depend on the order in which the state variable changes, race condition is not harmful.
* Critical race: If the final stable state depends on the order of the changes in the state variable then race condition is harmful, this condition must be avoided.

Noncritical races:

Let Y_1, Y_2 → state variables, X → input.

Stable state $Y_1Y_2X = 000$ input changes from 0 to 1

Y_1Y_2 \ X	0	1
00	(00)	11
01		11
11		(11)
10		11

00 → 11
00 → 01 → 11
00 → 10 → 11

* there are 3 possibilities.
* 1st → 00 to 11
* 2nd → 00 to 10 then 11
* 3rd → 00 to 01 then 11

final state is 11 which results in noncritical race condition

Y_1Y_2 \ x	0	1
00	00	11
01		11
11		10
10		(10)

possible transition

$00 \rightarrow 11 \rightarrow 10$

$00 \rightarrow 10$

$00 \rightarrow 01 \rightarrow 11 \rightarrow 10$.

Critical races.

Y_1Y_2 \ x	0	1
00	(00)	11
01		(01)
11		(11)
10		(10)

Possible transition

$00 \rightarrow 11$

$00 \rightarrow 01$

$00 \rightarrow 10$

* Circuit has stable state $Y_1Y_2x = 000$ and there is a change in input from 0 to 1.

* if changes simultaneously 00 to 11

Stable state $Y_1Y_2x = 111$.

* if Y_2 change before Y_1

$00 \rightarrow 01$ unequal delay

then circuit goes to stable state 011.

* if Y_1 change before Y_2

$00 \rightarrow 10$ unequal delay

then circuit goes to stable state 101 & remains

* Races can be avoided by directing the circuit through intermediate unstable state with a unique state - variable change.

CYCLE: When the circuit goes through a unique sequence of unstable states it is called cycles.

Y_1Y_2 \ x	0	1
00	(00)	01
01		11
11		10
10		(10)

a) state transition

$00 \rightarrow 01 \rightarrow 11 \rightarrow 10$

Y_1Y_2 \ x	0	1
00	(00)	01
01		11
11		(11)
10		(10)

b) state transition

$00 \rightarrow 01 \rightarrow 11$

Y_1Y_2 \ x	0	1
00	(00)	01
01		11
11		10
10		01

c) unstable

$01 \rightarrow 11 \rightarrow 10 \rightarrow 01$

Start with 00 and change input from 0 to 1

a) transition stable gives a unique sequence that terminates total stable state 101.

b) even though the state variables change from 00 to 11, the cycle provides a unique transition from 00 to 01 and then to 11.

* Care must be taken when using a cycle that it terminates with a stable state. If a cycle doesnot terminate with a stable state, the circuit will keep going from one unstable state to another, making the entire circuit unstable.

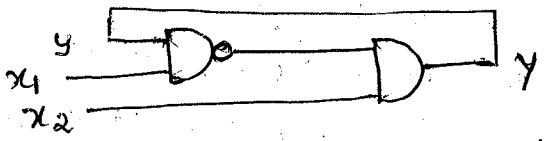
STABILITY CONSIDERATION.

In asynchronous circuits feedback connections exist, therefore care must be taken to ensure that circuit doesnot become unstable

* An unstable condition will cause the circuit to oscillate between unstable states.

example of an unstable circuit

$$Y = (x_1 y)' x_2 = (x_1' + y') x_2 = x_1' x_2 + x_2 y'$$



y \ x_1x_2	00	01	11	10
0	(0)	1	1	(0)
1	0	(1)	0	0

* Let $x_1=1$, $x_2=1$, and $y=1$. The output of NAND gate is equal to 0, and the output of AND gate is equal to 0, making Y equal to 0, with the result that $Y \neq y$.

* if $y=0$, the output of NAND gate is equal to 1, the AND gate is 1, making $Y=1$, with result $Y \neq y$.

* the instability that may occur in asyncheonous sequential circuit is undesirable and must be avoided

TWO TECHNIQUES FOR CRITICAL-RACE FREE STATE ASSIGNMENT

1. Shared row state assignment 2. One hot state assignment

SHARED ROW STATE ASSIGNMENT

* Races can be avoided by proper binary assignment

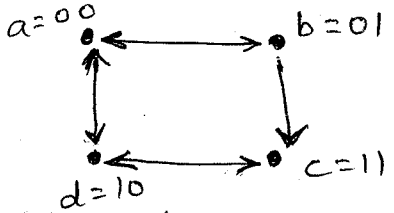
* When state transition occurs there should be only one state change in state variable.

Let the transition diagram start the transition from state a to state b and transition from state a to state c. a = 00, c = 11.

(Transition diagram)

This causes critical race during transition from a to c since there are two changes in binary state variable.

∴ A race free assignment can be obtained by introducing addition binary state say 'd' with binary value 10, which is adjacent to both 'a' and 'c'.



binary variable change from 00 → 10 → 11 which satisfy the condition that only one binary variable changes during each state transition, thus avoiding the critical race. This technique is called shared row state assignment. This extra state is shared between two stable states

ONE HOT STATE ASSIGNMENT.

Flow table

State Variables f_4 f_3 f_2 f_1	State	Inputs X_1X_2 00	01	11	10
0 0 0 1	A	(A)	B	C	C
0 0 1 0	B	A	(B)	C	D
0 1 0 0	C	A	B	(C)	(C)
1 0 0 0	D	(D)	B	C	(D)

* Four state variable are used to represent four rows in the table.
* from f_1 to F_2 there is a transition $F_1 = 1$ to 0 then 0 to $F_2 = 1$.
∴ additional rows are included that is One hot state assignment.

example:
* When $X_1X_2 = 01$ the transition from A to B is passing through dummy state E.

One hot state assignment flow table

	State Variable f_4 f_3 f_2 f_1	State	Input X_1X_2 00	01	11	10
Original table	0 0 0 1	A	(A)	B/E	C/F	C/F
	0 0 1 0	B	A/E	(B)	C	D
	0 1 0 0	C	A/F	B/G	(C)	(C)
	1 0 0 0	D	(D)	B/H	C/I	(D)
Added rows	0 0 1 1	E	A	B	—	—
	0 1 0 1	F	A	—	C	—
	0 1 1 0	G	—	B	C	—
	1 0 1 0	H	—	B	—	D
	1 1 0 0	I	—	—	C	D

ANALYSIS OF ASYNCHRONOUS SEQUENTIAL CIRCUIT.

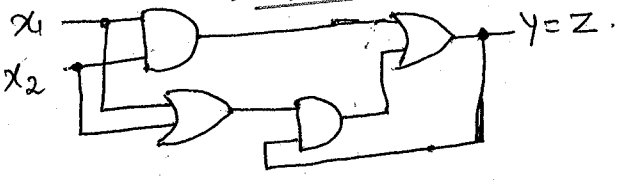
1. An asynchronous sequential circuit is described by the following excitation and output function $Y = x_1x_2 + (x_1+x_2)y$

$Z = y$.

(i) Draw the logic diagram of the circuit.
(ii) Derive the transition table and output map.
(iii) Describe the behaviour of the circuit

Soln:

i. logic diagram.



ii)

Present total state			Next total state		internal Y state	state total	Z
x_1	x_2	y	x_1	x_2		Yes / No	
0	0	0	0	0	0	Yes	0
0	0	1	0	0	0	No	0
0	1	0	0	1	0	Yes	0
0	1	1	0	1	1	Yes	1
1	0	0	1	0	0	Yes	0
1	0	1	1	0	1	Yes	1
1	1	0	1	1	1	No	1
1	1	1	1	1	1	Yes	1

Internal state (Y)

y \ x_1x_2	00	01	11	10
0	(0)	(0)	1	(0)
1	0	(1)	(1)	(1)

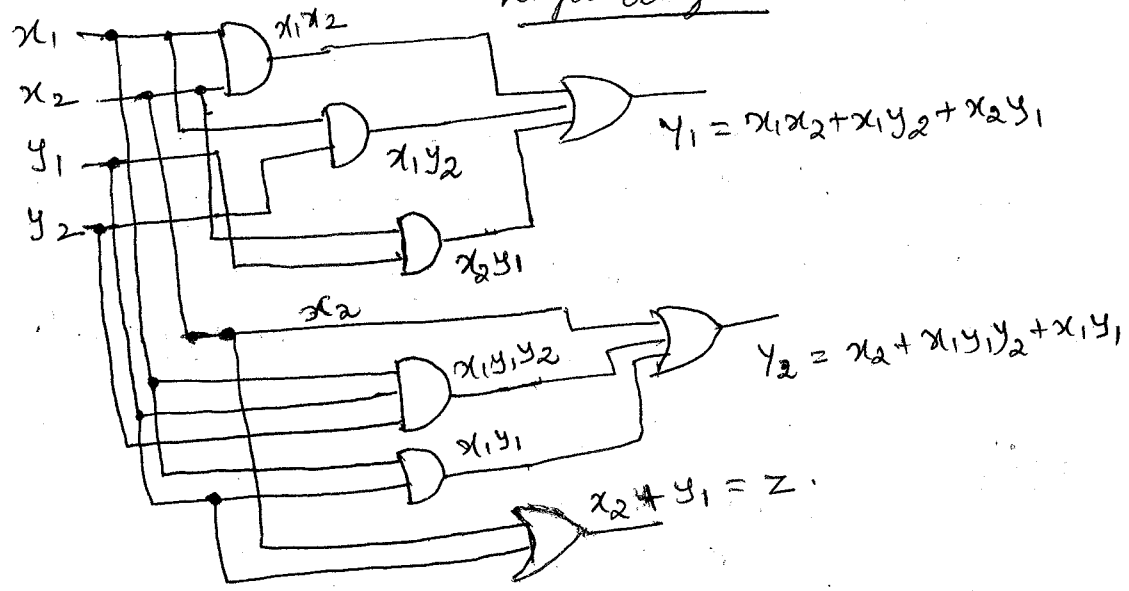
Output (Z)

y \ x_1x_2	00	01	11	10
0	0	0	–	0
1	–	1	1	1

2. An asynchronous sequential circuit has two internal states and one output. The excitation output function describing the circuit are as follows.

$$Y_1 = x_1x_2 + x_1y_2 + x_2y_1$$
$$Y_2 = x_2 + x_1y_1y_2 + x_1y_1$$
$$Z = x_2 + y_1$$

Logic diagram



Present state				Next state				stable		z
x_1	x_2	y_1	y_2	x_1	x_2	Y_1	Y_2	Yes	No	
0	0	0	0	0	0	0	0	Yes		0
0	0	0	1	0	0	0	0	No		0
0	0	1	0	0	0	0	0	No		1
0	0	1	1	0	0	0	0	No		1
0	1	0	0	0	1	0	1	No		1
0	1	0	1	0	1	0	1	Yes		1
0	1	1	0	0	1	1	1	No		1
0	1	1	1	0	1	1	1	Yes		1
1	0	0	0	1	0	0	0	Yes		0
1	0	0	1	1	0	1	0	No		0
1	0	1	0	1	0	1	1	No		1
1	0	1	1	1	0	1	1	Yes		1
1	1	0	0	1	1	1	1	No		1
1	1	0	1	1	1	1	1	No		1
1	1	1	0	1	1	1	1	No		1
1	1	1	1	1	1	1	1	Yes		1

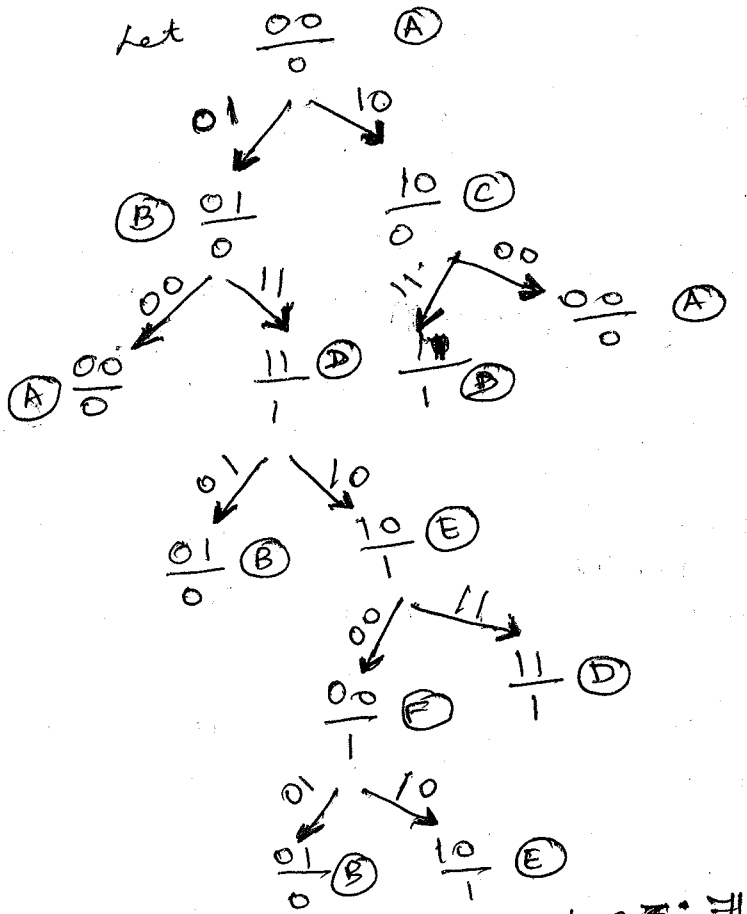
y_1y_2 \ x_1x_2	00	01	11	10
00	(00)	01	11	(00)
01	00	(01)	11	10
11	00	(11)	(11)	(11)
10	00	11	11	01

0	—	—	0
—	1	—	—
—	1	1	1
—	—	—	—

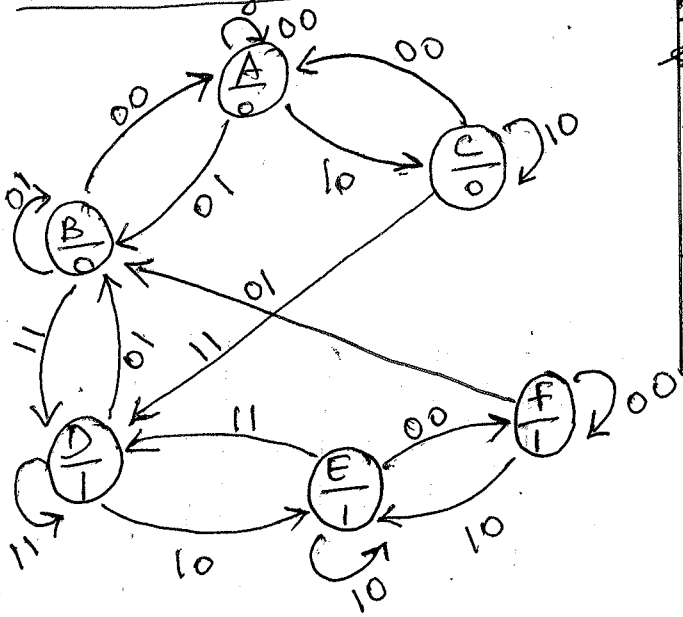
DESIGN OF ASYNCHRONOUS SEQUENTIAL CIRCUIT.

1. Design an asynchronous sequential circuit with two inputs x and y and with one output z. Whenever y is 1, input x is transferred to z. When y is 0, the output doesnot change for any change in x. (May/June '03)

Step:1



step 2:
State diagram



step 3: Flow table

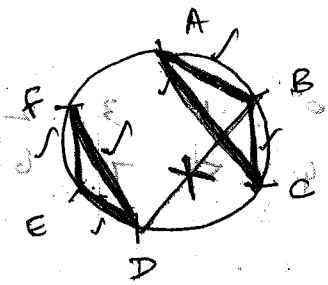
Present state	Next state, o/p z for xy inputs			
	00	01	11	10
A	(A),0	B,–	–,–	C,–
B	A,–	(B),0	D,–	–,–
C	A,–	–,–	D,–	(C),0
D	F,–	B,–	(D),1	E,–
E	F,–	–,–	D,–	(E),1
F	(F),–	B,–	–,–	E,–

Implication table

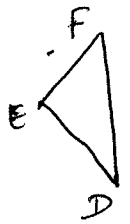
	A	B	C	D	E
B	✓	–	–	–	–
C	✓	✓	–	–	–
D	X	✓	X	–	–
E	X	X	X	✓	–
F	X	X	X	✓	✓

(A,B) (A,C) (B,C) (B,D) (D,E) (D,F) (E,F)

step 4: reduce flow table using Merger graph.



(ii)



A, B, C = S_0

D, E, F = S_1

step 5: Reduced flow table

Present state (F)	Next state (F^+), Output z for : xy inputs.			
	00	01	11	10
S_0	S_0, 0	S_0, 0	S_1, –	S_0, 0
S_1	S_1, 1	S_0, –	S_1, 1	S_1, 1

step 6: Assigning binary Value. $S_0 \to 0, S_1 \to 1$

Present state	Next state F^+, o/p z for xy inputs			
	00	01	11	10
0	0, 0	0, 0	1, –	0, 0
1	1, 1	0, –	1, 1	1, 1

step 7: For F_+

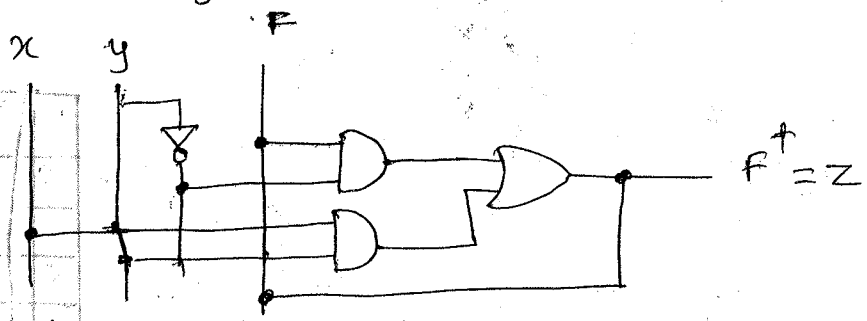
f \ xy	00	01	11	10
0	0 (0)	0 (1)	1 (3)	0 (2)
1	1 (4)	0 (5)	1 (7)	1 (6)

$F^+ = F\bar{y} + xy$

For z

F \ xy	00	01	11	10
0	0 (0)	0 (1)	X (3)	0 (2)
1	1 (4)	X (5)	1 (7)	1 (6)

$Z = F\bar{y} + xy$.



Implementation using SR latch.

For input xy = 01, the second row of transition table, it requires a transition from F = 1 to $F^+ = 0$. The excitation table specifies S = 0, R = 1 for this change. The corresponding squares in the S map is marked with a 0 and the One in the R map with a 1.

Excitation table:

F	F^+	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

K-map

For S

f \ xy	00	01	11	10
0	0 (0)	0 (1)	1 (3)	0 (2)
1	X (4)	0 (5)	X (7)	X (6)

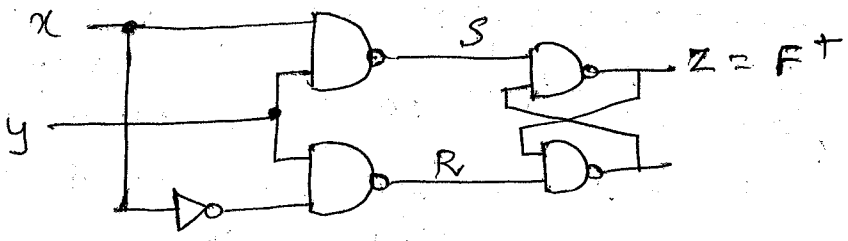
$S = xy$.

For R

f \ xy	00	01	11	10
0	X (0)	X (1)	0 (3)	X (2)
1	0 (4)	1 (5)	0 (7)	0 (6)

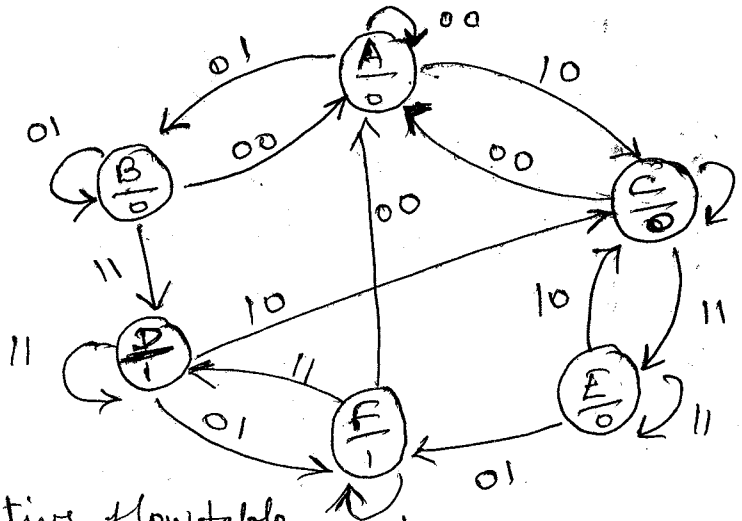
$R = \bar{x}y$.

Logic diagram



2. Design an asynchronous sequential circuit that has two inputs x_2 and x_1 and one output Z. When $x_1 = 0$, the output Z is 0. The first change in x_2 that occurs while x_1 is 1 will cause output Z to be 1. The output Z will remain 1 until x_1 returns to 0.

Step 1:



Step 2:

Primitive flow table

Present state	Next state, output Z for x_2x_1 Input			
	00	01	11	10
A	(A), 0	B, —	—, —	C, —
B	A, —	(B), 0	D, —	—, —
C	A, —	—, —	E, —	(C), 0
D	—, —	F, —	(D), 1	C, —
E	—, —	F, —	(E), 0	C, —
F	A, —	(F), 1	D, —	—, —

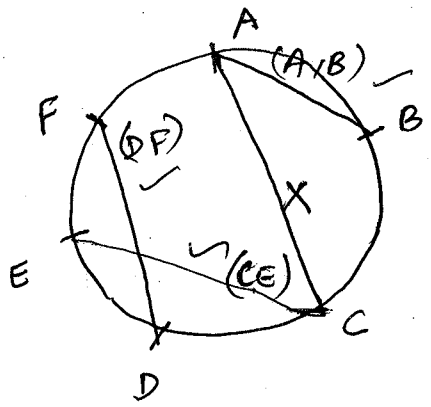
Step 3

Implication table

	A	B	C	D	E
B	✓				
C	X	✓			
D	X	X	X		
E	X	X	✓	X	
F	X	X	X	✓	X

(A, B), (A, C), (C, E) (D, E)

4. Merger graph.



Compatible pairs

$(A, B) \rightarrow S_0$

$(C, E) \rightarrow S_1$

$(D, F) \rightarrow S_2$.

Let $S_0 = 00$

$S_1 = 01$

$S_2 = 10$

5. REDUCED FLOW TABLE

Present state	Next state f_+, Output Z for $x_2 x_1$ inputs.			
	00	01	11	10
$S_0 \rightarrow 00$	(S0), 0	(S0), 0	S_2, –	S_1, –
$S_1 \rightarrow 01$	S_0, –	S_2, –	(S1), 0	S_1, 0
$S_2 \rightarrow 10$	S_0, –	(S2), 1	(S2), 1	S_1, –

6. FLOW TABLE WITH STATE ASSIGNMENT

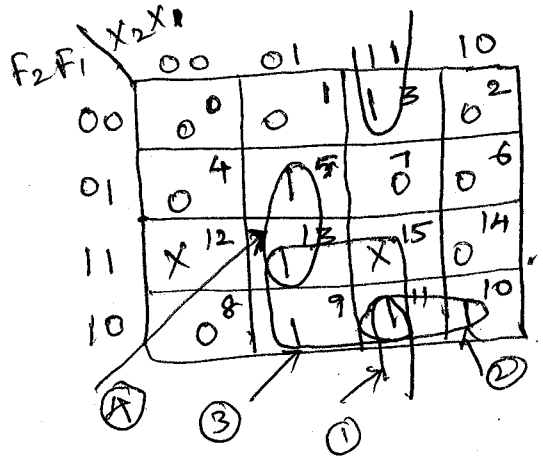
Present state $f_2 f_1$	Next state f_+, output z for $x_2 x_1$ inputs.			
	00	01	11	10
$S_0 \rightarrow 00$	(S0), 0	(S0), 0	S_2, –	S_1, –
$S_1 \rightarrow 01$	S_0, –	S_3, –	(S1), 0	(S1), 0
$S_2 \rightarrow 10$	S_0, –	(S2), 1	(S2), 1	S_3, –
$S_3 \rightarrow 11$	–, –	S_2, –	–, –	S_1, –

7. FLOW TABLE CONVERTED TO TRANSITION TABLE.

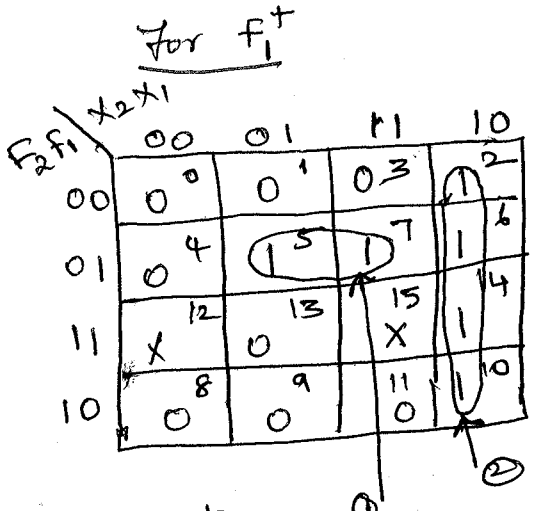
Present State f_2 f_1	Next state $f_2^+ f_1^+$, Output z for $x_2 x_1$ inputs			
	00	01	10	11
0 0	00, 0	00, 0	10, –	01, –
0 1	00, –	11, –	01, 0	01, 0
1 0	00, –	10, 1	10, 1	11, –
1 1	–, –	10, –	–, –	01, –

8. K-map simplification.

For f_2^+.



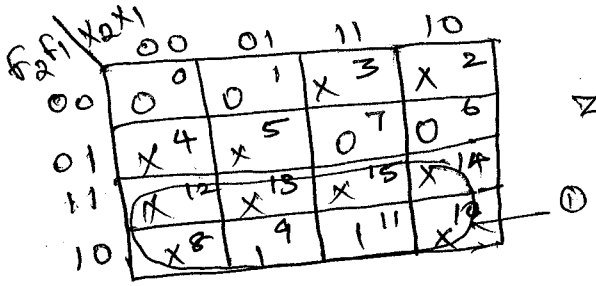
For f_1^+



$f_2^+ = \bar{F}_1 X_2 X_1 + F_1 \bar{X}_2 X_1 + F_2 X_1 + F_2 \bar{F}_1 X_2$

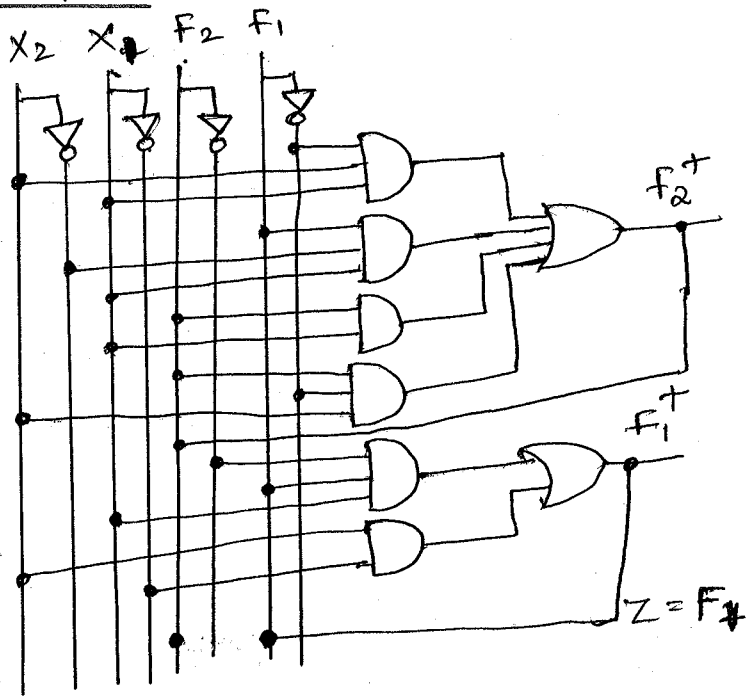
$f_1^+ = \bar{F}_2 F_1 X_1 + X_2 \bar{X}_1$

For Z



$Z = f_2$.

9. LOGIC DIAGRAM.



3. Design a T flip-flop from logic gates. [Nov/Dec '07, '08 '09]

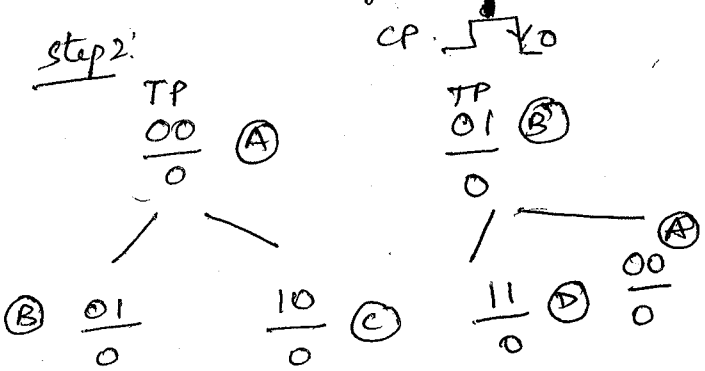
The flip flop changes its state if T=1 and when the clock (P) changes from 1 to 0. Under all other input conditions, Output Q will remain constant.

Assume T and clock pulse (P) do not change simultaneously.

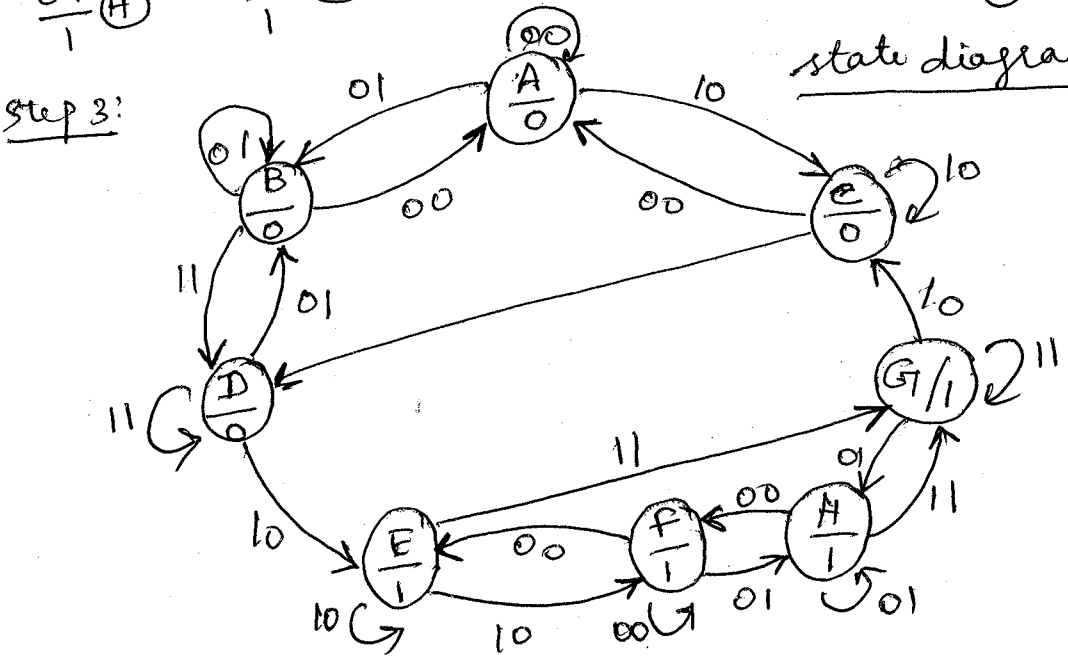
Table 'Step 1':

CP	Q_n	T	Q_{n+1}
↓	0	0	0
↓	0	1	1
↓	1	0	1
↓	1	1	0
↑	x	x	No change.

Step 2:



Step 3:

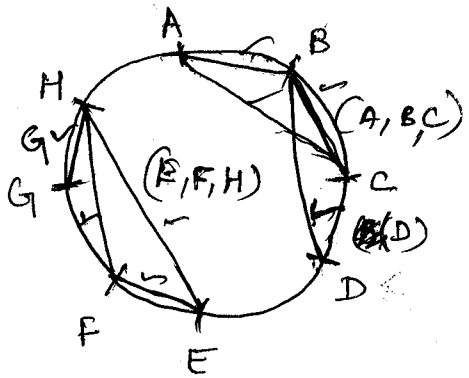


state diagrams

4. Primitive flow table

Present state	Next state, output for TP inputs			
	00	01	11	10
A	(A), 0	B, –	–, –	C, –
B	A, –	(B), 0	D, –	–, –
C	A, –	–, –	D, –	(E), 0
D	–, –	B, –	(D), 0	E, –
E	F, –	–, –	G, –	(E), 1
F	(F), 1	H, –	–, –	E, –
G	–, –	H, –	(G), 1	C, –
H	F, –	(H), 1	G, –	–, –

5. Merger graph.



Compatible pairs

(A, B, C) → S_0

D → S_1

(E, F, H) → S_2

G → S_3

6. Reduced primitive flow table.

Present state	Next state, output for TP inputs			
	00	01	11	10
S_0	(S_0), 0	(S_0), 0	S_1, –	(S_0), 0
S_1	–, –	S_0, 1	(S_1), 0	S_2, –
S_2	(S_2), 1	(S_2), 1	S_3, –	(S_2), 1
S_3	–, –	S_2, –	(S_3), 1	S_0, –

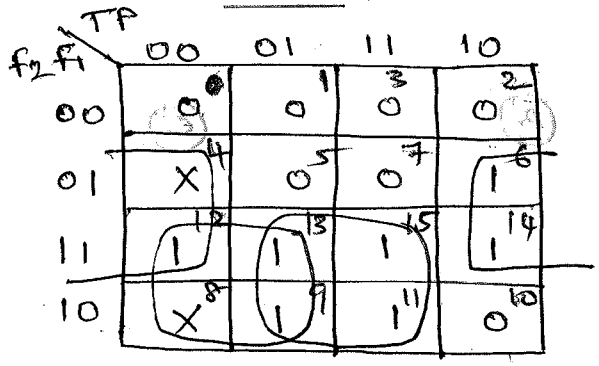
* Assign $S_0 \to 00$, $S_1 \to 01$, $S_2 \to 11$, $S_3 \to 10$.

Transition table

Present state		Next state, output Q for TP Inputs. Next state			
		TP			
f_2	f_1	00	01	11	10
0	0	$f_2^+ f_1^+$ Q: 00, 0	00, 0	01, –	00, 0
0	1	–, –	00, –	01, 0	11, –
1	1	11, 1	11, 1	10, –	11, 1
1	0	–, –	11, –	10, 1	00, –

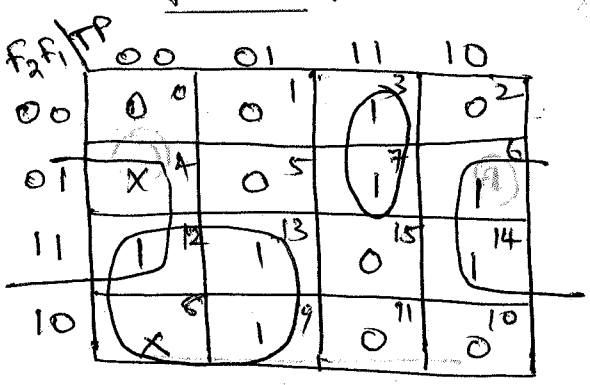
8. K-map

For F_2^+



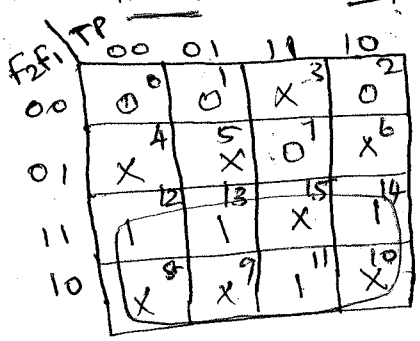
$f_2^+ = f_1 \bar{P} + f_2 \bar{T} + f_2 P$

For F_1^+



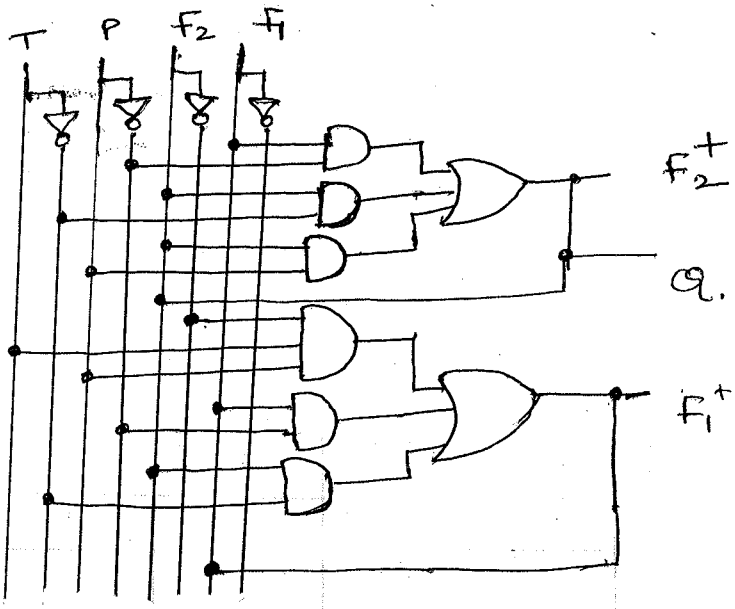
$F_1 = \bar{F}_2 TP + F_1 \bar{P} + F_2 \bar{T}$

For Q.



$Q = F_2$.

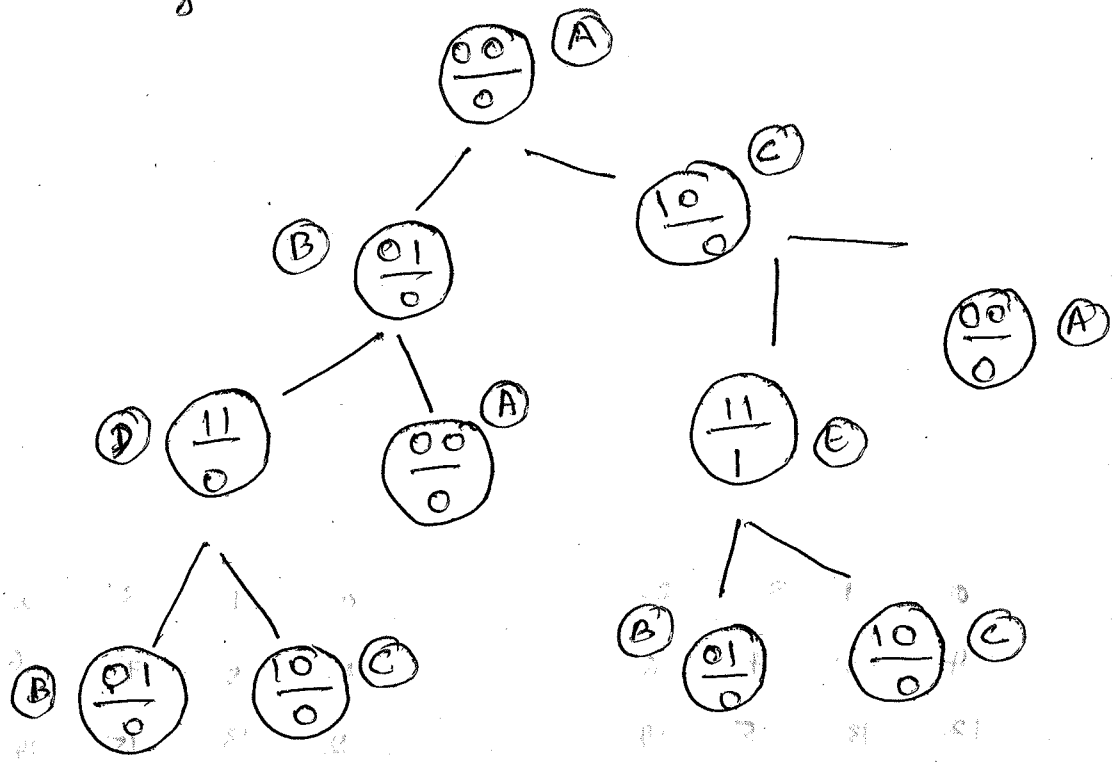
Step 9:



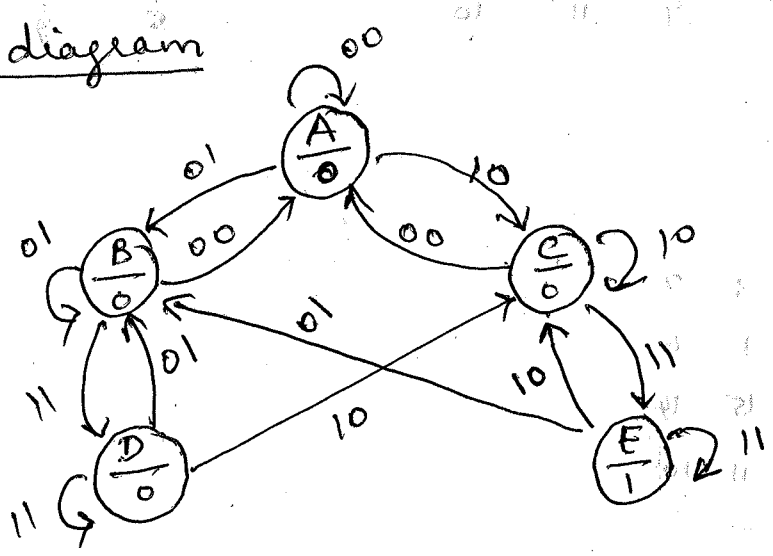
4. Design a circuit with input X and Y to give an output Z=1 when XY=11 but only if X becomes 1 before Y, by drawing total state

diagram, Primitive flow table and output map in which transient state is included. (May/June '06)

State diagram



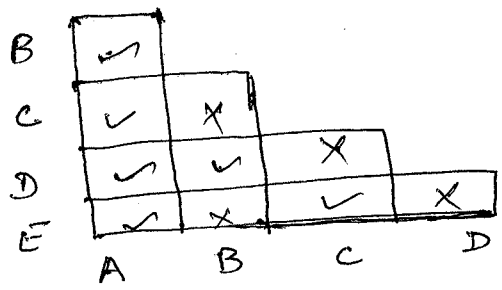
2) State diagram



3. State table

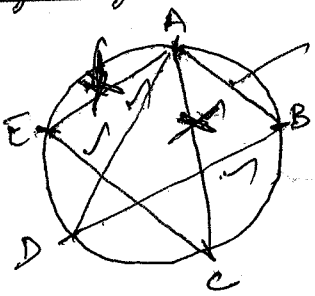
Present state	Next state (xy).			
	00	01	11	10
A	(A), 0	B, –	–, –	C, –
B	A, –	(B), 0	D, –	–, –
C	A, –	–, –	E, –	(C), 0
D	–, –	B, –	(D), 0	e, –
E	–, –	B, –	(E), 1	C, –

4: Implication table.



(A,B), (A,C), (A,D), (A,E)
(B,D), (C,E)

5. Merger graph



$(A, B, D) \xrightarrow{S0}$, $(C, E) = S_1$

6. Reduced flow table

Present state	Next state			
	00	01	11	10
S_0 (A,B,D)	(S0),0	S_0, –	(S0),0	S_1, –
S_1 (C,E)	S_0, –	S_0, –	(S1),1	(S1),0

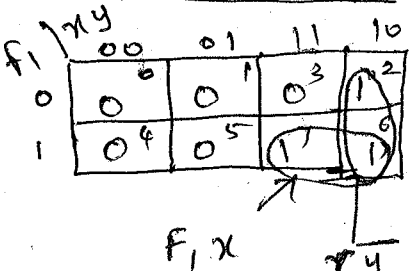
7. Assign values.

$S_0 \rightarrow 0$, $S_1 = 1$.

Present state f_1	Next state f_{1+}, for input xy output is Q.			
	00	01	11	10
0	0,0	0, –	0,0	1, –
1	0, –	0, –	1,1	1,0

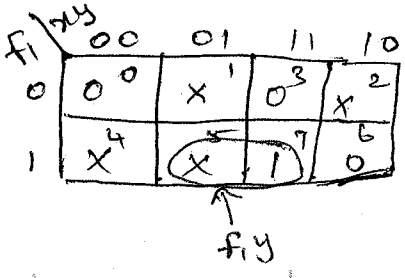
8. k-map:

f_{1+} (Next state)



$f_{1+} = f_1 x + x\bar{y}$

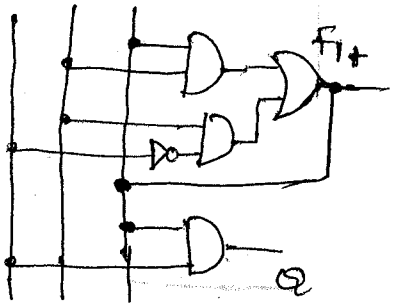
Output Q



$Q = f_1 y$.

9. Logic diagrams

y x f_1



HAZARDS

Hazards are unwanted switching transients that may appear at the output of a circuit because different paths exhibit different propagation delays. Hazards occurs in combinational circuits, where they may cause a temporary false-output value. When this condition occurs in asynchronous sequential circuits, it may result in transition to a wrong stable state.

HAZARDS IN COMBINATIONAL CIRCUITS

A hazard is a condition where a single variable change produces a momentary output change when no output change should occur.

Types: static (static 0, static 1), Dynamic

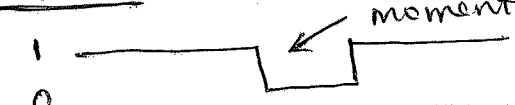
static hazard.

In digital systems, there are only two possible outputs '0' or '1'. The hazard may cause wrong '0' or a wrong '1'.

static-0:

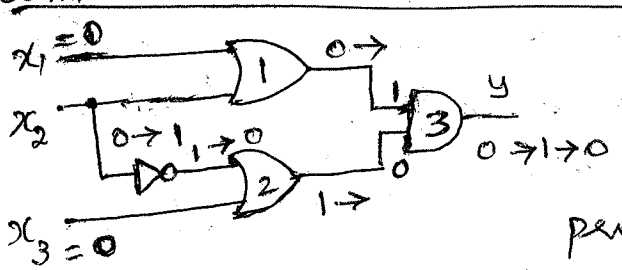
When the output of the circuit is to remain at 0, and a momentary 1 output is possible during the transmission between the two inputs, then the hazard is called static-0 hazard.

static-1:



When the output of the circuit is to remain at 1, and a momentary 0 output is possible during the transmission between the 2 inputs then the hazard is called static-1 hazard.

Combinational circuit with static 0 hazard.

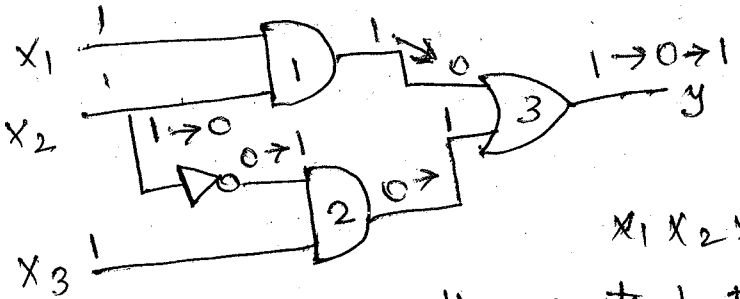


Assume that the inverter has an appreciably greater propagation delay time than the other gates. In this case there is a static-0 hazard in the transition between the input states $X_1 X_2 X_3 = 000$ and $X_1 X_2 X_3 = 010$ since it is possible for a logic 1 signal to appear at both input terminals of the AND gate for a short duration.

The delay in the inverter may cause the output of gate 1 to change to 1 before the output of gate 2 changes to 0. In that case, both inputs of gate 3 are momentarily equal to 0, causing the output to go to 1 for the short interval of time that the input signal from x_2 is delayed while it is propagating through the inverter circuit.

Thus static-0 hazard exist during the transition between the input states $X_1 X_2 X_3 = 000$ and $X_1 X_2 X_3 = 010$.

Circuit with static-1 hazard.



Assume that all three inputs are initially equal to 1 (ie $X_1 X_2 X_3 = 111$. This cause the output of the gate 1 to be 1, that of gate 2 to be 0, and the output of the circuit

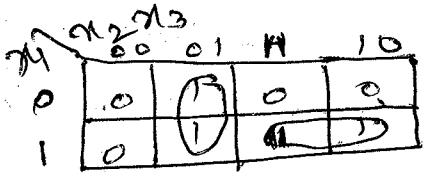
to be equal to 1. Now consider a change of x_2 from 1 to 0 (b) $x_1x_2x_3 = 101$. The output of gate 1 changes to 0 and that of gate 2 changes to 1, leaving the output at 1. The output may momentarily go to 0 if the propagation delay through the inverter is taken into consideration

The delay in the inverter may cause the output of gate 1 to change to 0 before the output of gate 2 changes to 1. In that case, both inputs of gate 3 are momentarily equal to 0, causing the output to go to 0 for the short interval of time that the input signal from x_2 is delayed while it is propagating through the inverter circuit.

Thus, a static-1 hazard exists during the transition between the input states $x_1x_2x_3 = 111$ and $x_1x_2x_3 = 101$.

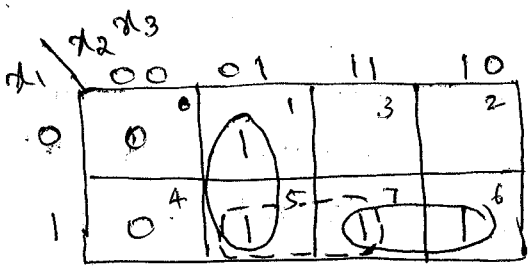
Eliminating Hazard.

A hazard can be detected by inspection of the map of the particular circuit. To illustrate, consider the map in the circuit with static-0 hazard, which is a plot of function implemented. The change in x_2 from 1 to 0 moves the circuit from minterm 111 to minterm 101. The hazard exists because the change in input results in a different product term covering two minterms.



$Y = x_1x_2 + x_2'x_3$

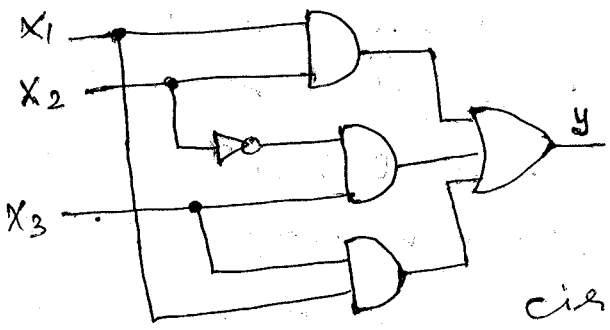
hazard circuits



$$Y = x_1x_2 + x_2'x_3 + x_1x_3$$

The minterm 111 is covered by product term implemented in gate 1 and minterm 101 is covered by the product term implemented in gate 2. Whenever the circuit must move from one product term to another, there is a possibility of momentary interval when neither term is equal to 1, giving rise to an undesirable 0 output.

The remedy for eliminating a hazard is to enclose the two minterms in question with another product term that overlaps both groupings. This situation is shown in map above, where two terms that causes the hazard are combined into one product term. The hazard free circuit obtained by this combinational circuit



The extra gate in the circuit generates the product term x_1x_3. The hazards in combinational circuits can be removed by covering any two minterms that may produce a hazard with a product term common to both. The removal of hazards requires the addition of redundant gates to the circuit.

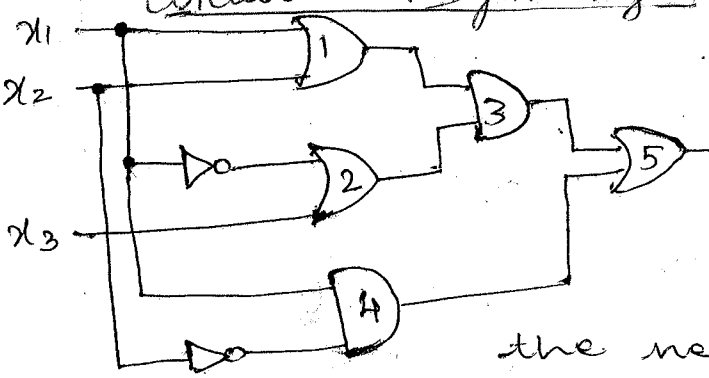
Dynamic Hazard.

A dynamic hazard is defined as a transient change occurring three or more times at an output terminal of a logic network when the output is supposed to change only once during a transition between two input states differing in the value of one variable.

Now consider the input states $x_1x_2x_3 = 000$ and $x_1x_2x_3 = 100$. For the first input state, the steady state output is 0, while for the second input state, the steady-state output is 1.

Assume there are no propagation delays of the other three gates are such that G_1 can switch faster than G_{12} and G_{12} can switch faster than G_{14}.

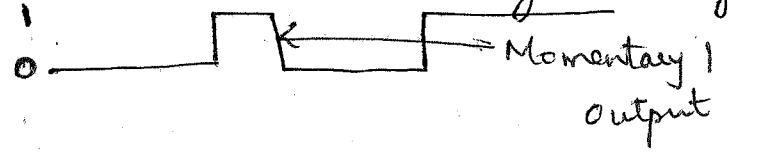
Circuit with Dynamic Hazard



When x_1 changes from 0 to 1, the change propagates through gate G_1 before gate G_{12} with the net effect that the inputs to gate G_3 are simultaneously 1 and the network output changes from 0 to 1. Then when x_1 change propagate through gate G_{12}, the lower input to gate G_3 becomes 0 and network output changes back to 0.

Finally, when the $x_1 = 1$ signal propagates through gate G_{14}, the lower input to gate G_5 becomes 1 and the network output again changes to 1.

It is therefore seen that during the change of X_1 variable from 0 to 1 the output undergoes the sequence, $0 \to 1 \to 0 \to 1$, which results in three changes when it should have undergone only a single change.



Essential Hazard.

An essential hazard is caused by unequal delays along two or more paths that originate from the same input. An excessive delay through an inverter circuit in comparision to the delay associated with the feedback path may cause such a hazard.

Essential hazards elimination

essential hazards can be eliminated by adjusting the amount of delays in the affected path. To avoid essential hazards, each feedback loop must be handled with individual care to ensure that the delay in the feedback path is long enough compared with delays of other signals that originate from the input terminals.

Design of Hazard free circuits.

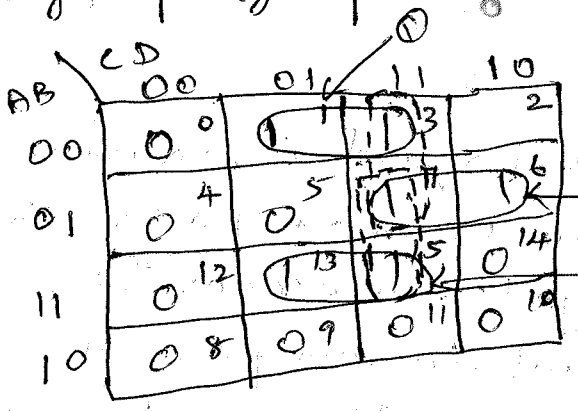
1. Design a hazard free circuit to implement the following function $F(A,B,C,D) = \Sigma m(1,3,6,7,13,15)$

AB \ CD	00	01	11	10
00	0 0	1 1	1 3	0 2
01	0 4	0 5	1 7	1 6
11	0 12	1 13	1 15	0 14
10	0 8	0 9	0 11	0 10

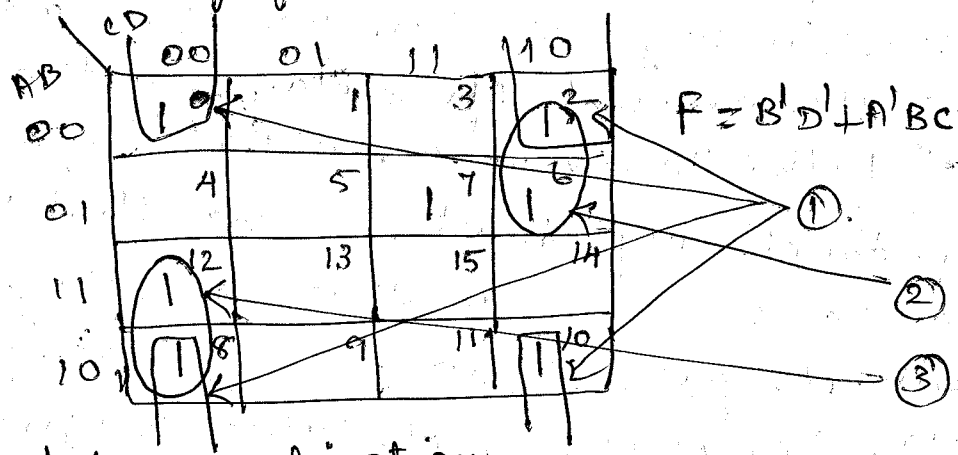
$F = A'B'D + A'BC + ABD.$

hazard-free realization

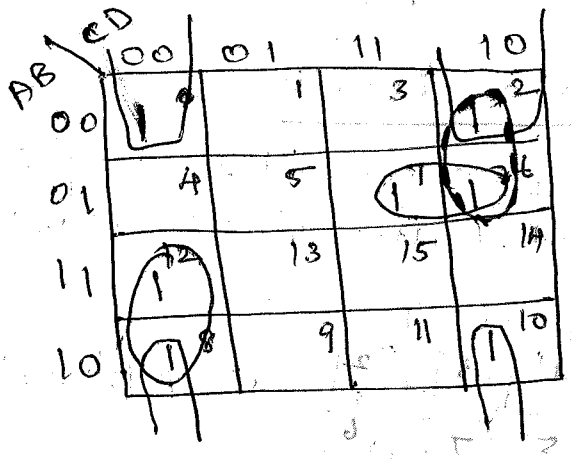
The first additional product term A'CD, overlapping two groups (group 1 and 2) and the second additional product term, BCD, overlapping the two groups (group 2 and 3).



2. Design a hazard free circuit to implement the following function $F(A,B,C,D) = \Sigma m(0,2,6,7,8,10,12)$



Hazard free realization



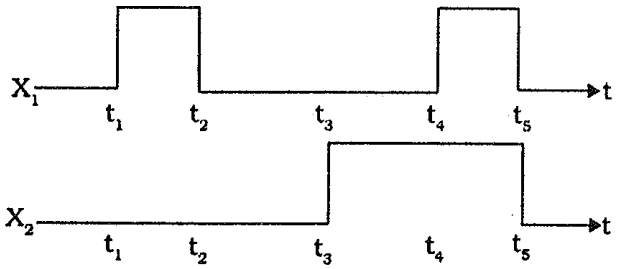
The additional product term, A'CD' overlapping two groups (group 1 & 2) for hazard free realization, Group 1 and 3 are already overlapped hence they donot require additional minterm for grouping

$F = B'D' + A'BC + AC'D' + A'CD'$

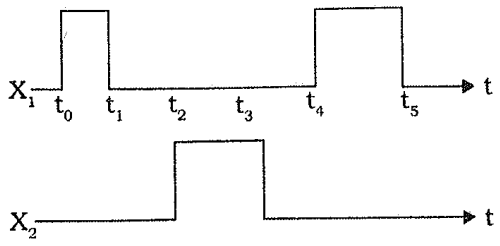
6.6 ANALYSIS OF PULSE MODE SEQUENTIAL CIRCUIT

In pulse mode circuit all the inputs are pulses in the width of the input pulses is critical to the circuit operation. The input pulse must be long enough for the circuit to respond to the input but it must not be so long as to the present even after a new state is reached. The minimum pulse width required is based upon the propagation delay through the next state logic. The maximum pulse width is determined by the total propagation delay through the next state logic and memory elements. When pulse occurs on any one input, while the circuit is in stable state, pulse must not arrive at any other input.

Let us consider two input pulse x_1 and x_2. In figure 6.82 (a) at time t_3 pulse at input x_2 arise



(a) Unacceptable pulse mode input changes



(b) Acceptable pulse mode input changes

Figure 6.82

While this pulse is still present, another pulse at x_1 input arrives at t_4, therefore this kind of presence of pulse input is not allowed.

Example 6.16 Let us consider the asynchronous sequential circuit which is driven by the pulses. The two NAND gate latches generates the state variables A and B. The circuit has four input variables w, x, y and z and one output variable C is shown in figure 6.83

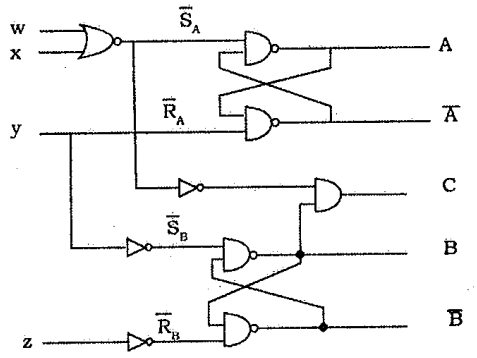


Figure 6.83 Logic Diagram

Solution

The Circuit Excitation and Output equation are given below

Excitation Equations

$$\bar{S}_A = \overline{w+x},\ S_A = w+x$$
$$\bar{R}_A = y$$
$$\bar{S}_B = \bar{y}, S_B = y$$
$$\bar{R}_B = \bar{z}$$

Output equations

$$c = (w+x).B$$

...(6.3)

The characteristic equation for SR flip flop is

$$Q_{n+1} = S + \bar{R}Q_n \qquad ...(6.4)$$

There are two latches $\bar{S}_A, \bar{R}_A$ and $\bar{S}_B, \bar{R}_B$. Therefore state variable for the next secondary equation are derived separately for Latch A and B by substituting equation (6.3) in (6.4). The next state equation Q_{A+1} and Q_{B+1} are given below in equation (6.5) and (6.6)

State variable next state equation substitute (6.3) in (6.4)

Let $Q_A = A$

$Q_B = B$

$n = A \Rightarrow Q_{A+1} = S_A + \bar{R}_A Q_A$

$= w + x + y.A$ ...(6.5)

$n = B \Rightarrow Q_{B+1} = S_B + \bar{R}_B Q_B$

$= y + \bar{z}B$ ...(6.6)

The state transition table indicating state variable, input variables, next state value and output state are shown in figure 6.84.

(i) If w column, assume $w = 1$, $x, y, z = 0$

(ii) If x column assume $x = 1$, $w, y, z = 0$

(iii) If y column, assume $y = 1$, $w, x, z = 0$

(iv) If z column assume $z = 1$, $w, x, y = 0$

$Q_A Q_B$ / $Q_A^+ Q_B^+$/C	w	x	y	z
00	10/0	10/0	01/0	00/0
01	11/1	11/1	01/0	00/0
11	11/1	11/1	11/0	00/0
10	10/0	10/0	11/0	00/0

Output C

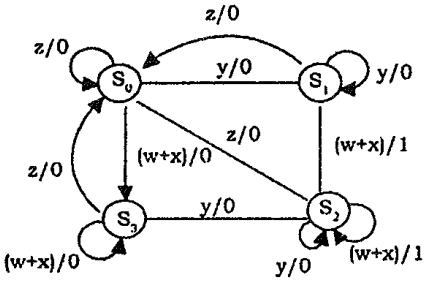
Next State $Q_A^+ Q_B^+$

Figure 6.84 Transition Table

The value of next state are assigned variables such as, $S_0 = 00$, $S_1 = 01$, $S_2 = 11$, $S_3 = 10$ and these values are replaced by the state variable in figure 6.84. The flow Table for the circuit is given below in figure 6.85

$Q_A Q_B$	w	x	y	z
S_0	$S_3/0$	$S_3/0$	$S_1/0$	$S_0/0$
S_1	$S_2/1$	$S_2/1$	$S_1/0$	$S_0/0$
S_2	$S_2/1$	$S_2/1$	$S_2/0$	$S_2/0$
S_3	$S_3/0$	$S_3/0$	$S_2/0$	$S_0/0$

The state Diagram is shown in figure 6.86



Example 6.17

Consider the asynchronous sequential circuit driven by the pulses shown in figure 6.87 analyze the circuit and draw the timing diagram

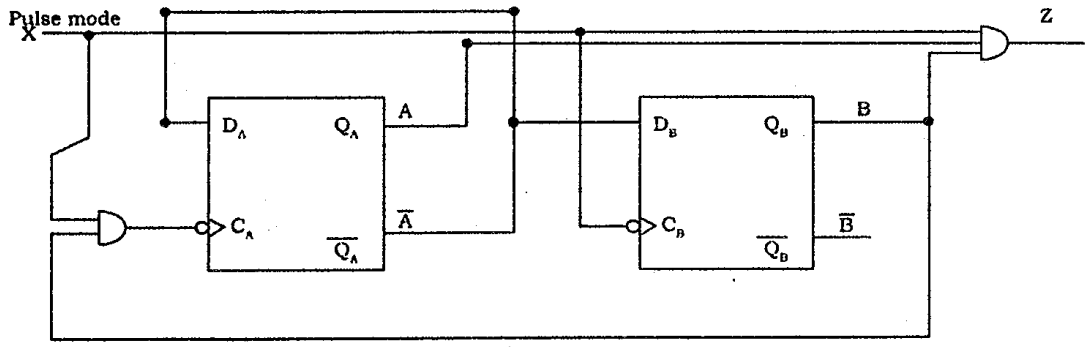


Figure 6.87 Logic Diagram

Solution

The Circuit excitation and output equation are given below

The excitation equations for the above circuit are

$$D_A = \bar{A}$$
$$D_B = \bar{A} \qquad \ldots(6.7)$$

The Output equation is given below

$$Z = XAB \qquad \ldots(6.8)$$

The Clock pulse CP equation is given as

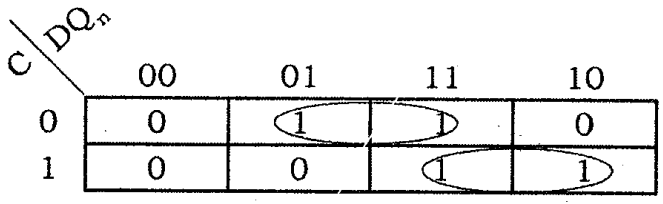
$$C_A = XB$$
$$C_B = X \qquad \ldots(6.9)$$

The truth table for the D-flip flop is given in table 6.32

Table 6.32 Truth table

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

The characteristic equation for D flip flop is derived from the figure 6.88 K-map simplification



$$Q_{n+1} = CD + \bar{C}Q_n \qquad ...(6.10)$$

Figure 6.88 K-map for D flip flop

To obtain the State variable for next state equation substitute equation (6.7), and (6.9) in equation (6.10). The obtained equations are r Q_{A+1} and Q_{B+1} are given below in equation (6.11) and (6.12)

$$n = A,\ Q_{A+1} = D_A C_A + \bar{C}_A Q_A \qquad \boxed{Let\ Q_A = A,\ Q_B = B}$$

$$= \bar{A}XB + \left(\overline{XB}\right)A$$

$$= A \oplus (XB) \qquad ...(6.11)$$

$$n = B,\ Q_{B+1} = D_B C_B + \bar{C}_B Q_B$$

$$= \bar{A}X + \bar{X}B \qquad ...(6.12)$$

The State transition table indicating state variable, input variable next state value and output state is shown in figure 6.89.

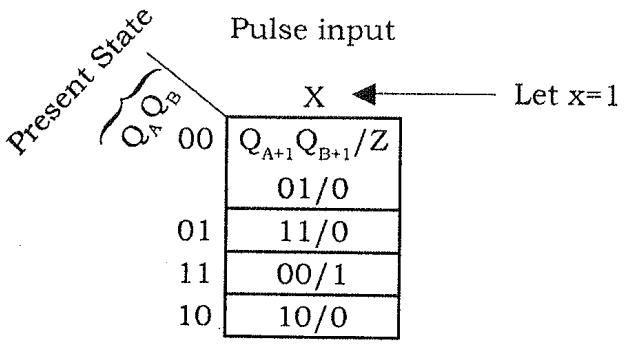


Figure 6.89 Transition table

The value of next state are assigned variables such as S_0 =00, S_1=01, S_2=11, S_3=10, and these values are replaced by the state variable in figure 6.89. The flow table for the circuit is given below in figure 6.90.

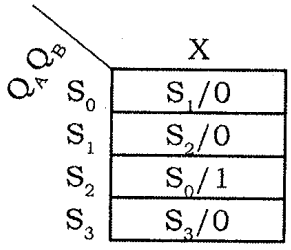


Figure 6.90 Flow table

The state diagram for the given circuit is shown in figure 6.91

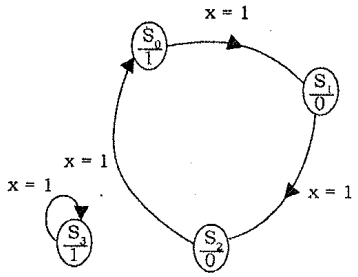


Figure 6.91 State diagram

6.7 DESIGN OF PULSE MODE CIRCUITS

The steps to design pulse mode circuits are as follows,

Step 1: Define states and draw a state diagram and/or state table of circuit

Step 2: Minimize the state table

Step 3: Do the state assignment

Step 4: Choose the type of latch or flip-flop to be used and determine excitation equation

Step 5: Construct the excitation table for circuit

Step 6: Determine the output equation and the flip-flop input equation using k-map simplification.

Step 7: Draw the logic diagram using flip flops and gates.

Example 6.18 Design a pulse mode circuit having two input lines x_1 and x_2 and one output line z as shown in figure 6.92. The circuit should produce an output pulse to coincide with last input pulse in sequence $x_1 \rightarrow x_2 \rightarrow x_2$. No other input sequence should produce an output pulse. Assign T Flip-flop

AU. QP A/M 2009

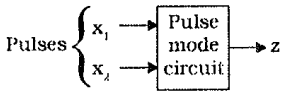


Figure 6.92 Pulse mode block diagram

Solution

Step 1: Let S_0 be the last input was x_1

S_1 be the sequence x_1 - x_2 occurred

S_2 be the Sequence x_1- x_2 -x_2 occurred

Step 2: With the condition given, a table is formed with the input x_1 and x_2 is shown in table 6.33

Table 6.33 State table

	Present state	Next state, output z	
		x_1	x_2
x_1	S_0	$S_0/0$	$S_1/0$
x_1-x_2	S_1	$S_0/0$	$S_2/1$
x_1-x_2-x_2	S_2	$S_0/0$	$S_2/0$

Step 3: The State Diagram for the circuit diagram is given in figure 6.93

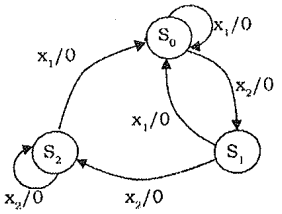


Figure 6.93 State Diagram

Step 4: Let $S_0 = 00$, $S_1 = 01$, $S_2 = 10$ be the three state variable and its state assignment shown in table 6.34

Table 6.34 State table with assignment

	Present state	Next state	
		x_1	x_2
S_0	00	00/0	01/0
S_1	01	00/0	10/0
S_2	10	00/0	10/0

Step 5: Using the excitation table of T flip flop from the state table for the circuit with the flip flop inputs shown in table 6.35

Table 6.35 State table

Present state		**Next state**				**Flip flop inputs**				**Output z**	
		x_1		**x_2**		**T_A**		**T_B**			
A	B	A_{t+1}	B_{t+1}	A_{t+1}	B_{t+1}	x_1	x_2	x_1	x_2	x_1	x_2
0	0	0	0	0	1	0	0	0	1	0	0
0	1	0	0	1	0	0	1	1	1	0	1
1	0	0	0	1	0	1	0	0	0	0	0

Step 6: The expression for flip-flop inputs T_A, T_B and output Z are obtained from K-map as shown in figure 6.94

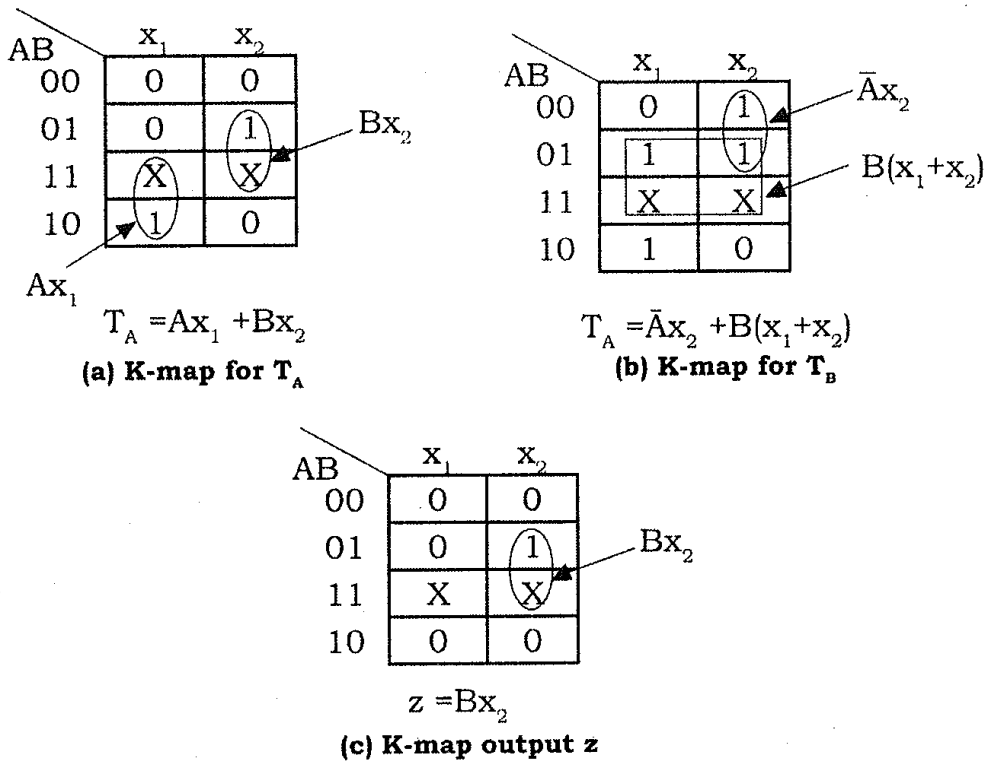


Figure 6.94 K-map Simplification

Step 7: The logic diagram for the given circuit is shown in figure 6.95

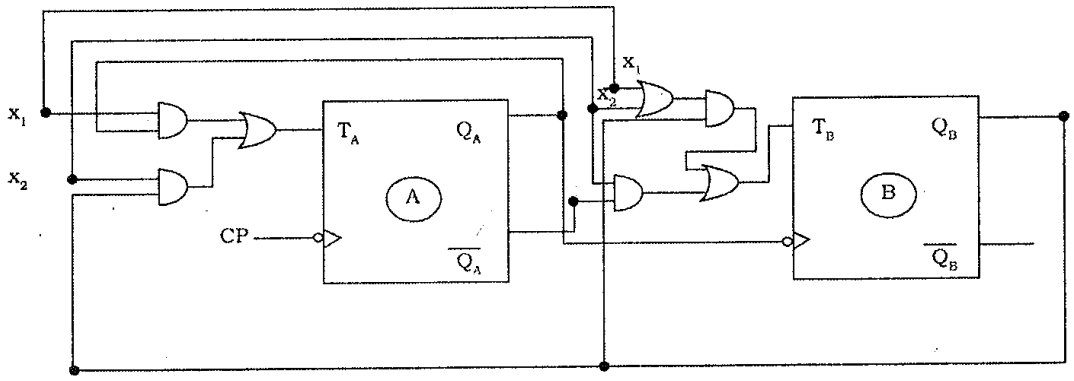


Figure 6.95 Logic Diagram