❖ Introduction

The circuits in which the output is a function of both present and past inputs are called sequential logic circuits. These circuits consist of a combinational logic circuits, storage element, and feedback. The sequential logic circuits are divided into asynchronous which are self-timing circuits, the output of these circuits changes directly according to the changing of its inputs and synchronous which are pulse-timing circuits, the output in this case depends on the inputs and clock pulse. The classical way of representing a sequential circuit is by either using state diagram or state table (also called a state transition diagram). The transition tables, state diagrams, and state equations of the four flip-flop types are:

1- $S - R$ flip-flop

The truth table and excitation table of $S - R$ flip-flop is given below:

S	R	$Q(t)$
0	0	No change
0	1	0(Reset)
1	0	1(Set)
1	1	Independent value

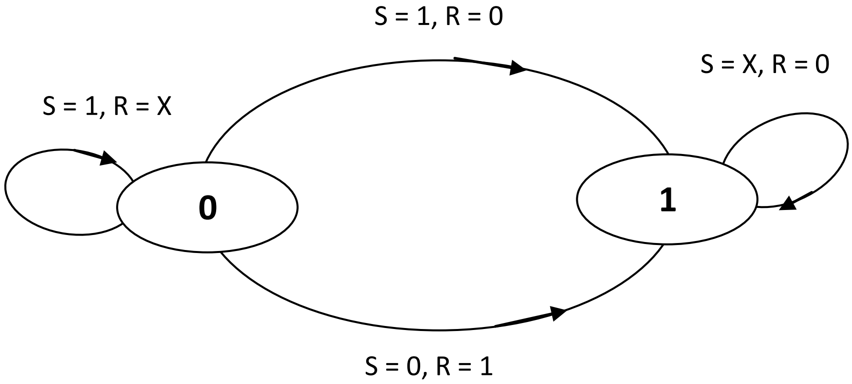
Q(t)	Q(t+1)	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

To find the next state of S-R flip-flop, the following truth table must be considered.

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	X
1	1	1	X

S \ RQ(t)	$\bar{R}\bar{Q}(t)$	$\bar{R}$Q(t)	RQ(t)	R$\bar{Q}(t)$
$\bar{S}$	0	1	0	0
S	1	1	X	X

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



2- *D* flip-flop

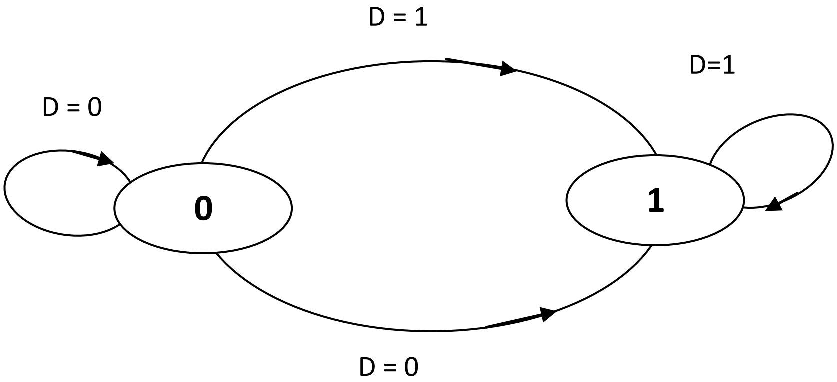
D	Q(t)	Q(t+1)
0	0	0(Reset)
0	1	0(Reset)
1	0	1(Set)
1	1	1(Set)

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

Q(t+1) = D

The excitation table and state diagram of (D) flip-flop is shown below

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



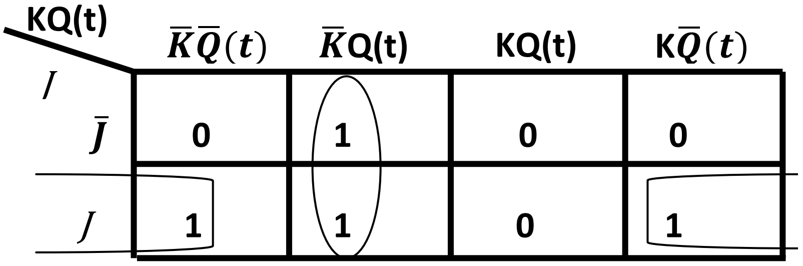
3- $J-K$ flip-flop

The excitation table, and truth table of this flip-flop is

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

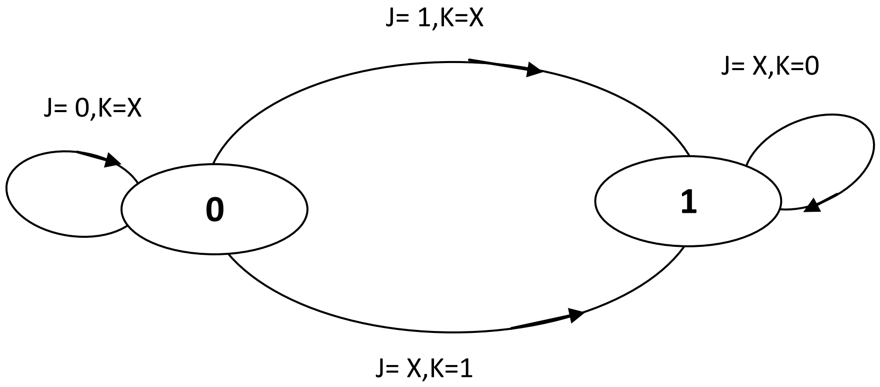
J	K	Q(t)
0	0	No change
0	1	0(Reset)
1	0	1(Set)
1	1	$\overline{\textbf{Last state}}$

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



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

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



4- *T* flip-flop

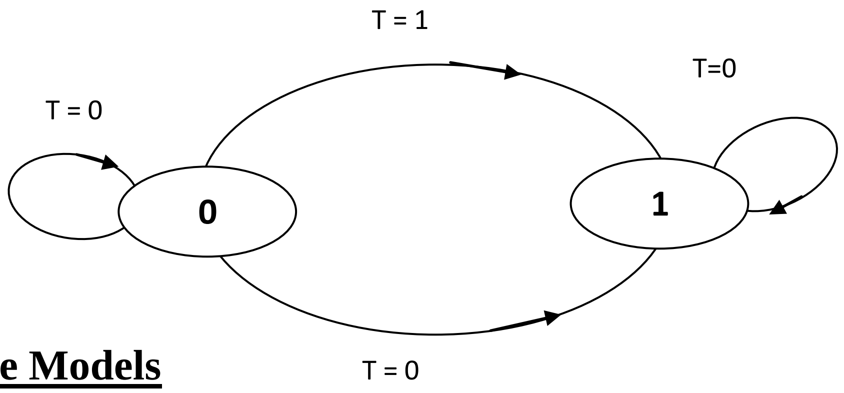
T	Q(t)	Q(t+1)
0	0	0(Reset)
0	1	1(Set)
1	0	1(Set)
1	1	0(Reset)

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

Q(t+1) = T ⊕ Q(t)

The excitation table is given below:

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



❖ Mealy and Moore Models

Generally, the most models of a sequential circuit have inputs, outputs and internal states. It is customary to distinguish between two models of sequential circuits Mealy and Moore. In Mealy model, the output is a function of both the present state and the inputs of the logic circuit, while the output of Moore logic circuit is a function of the present state of the logic circuit only.

Ex5/ draw the state table and state diagram for the synchronous sequential logic circuit shown in figure (13).

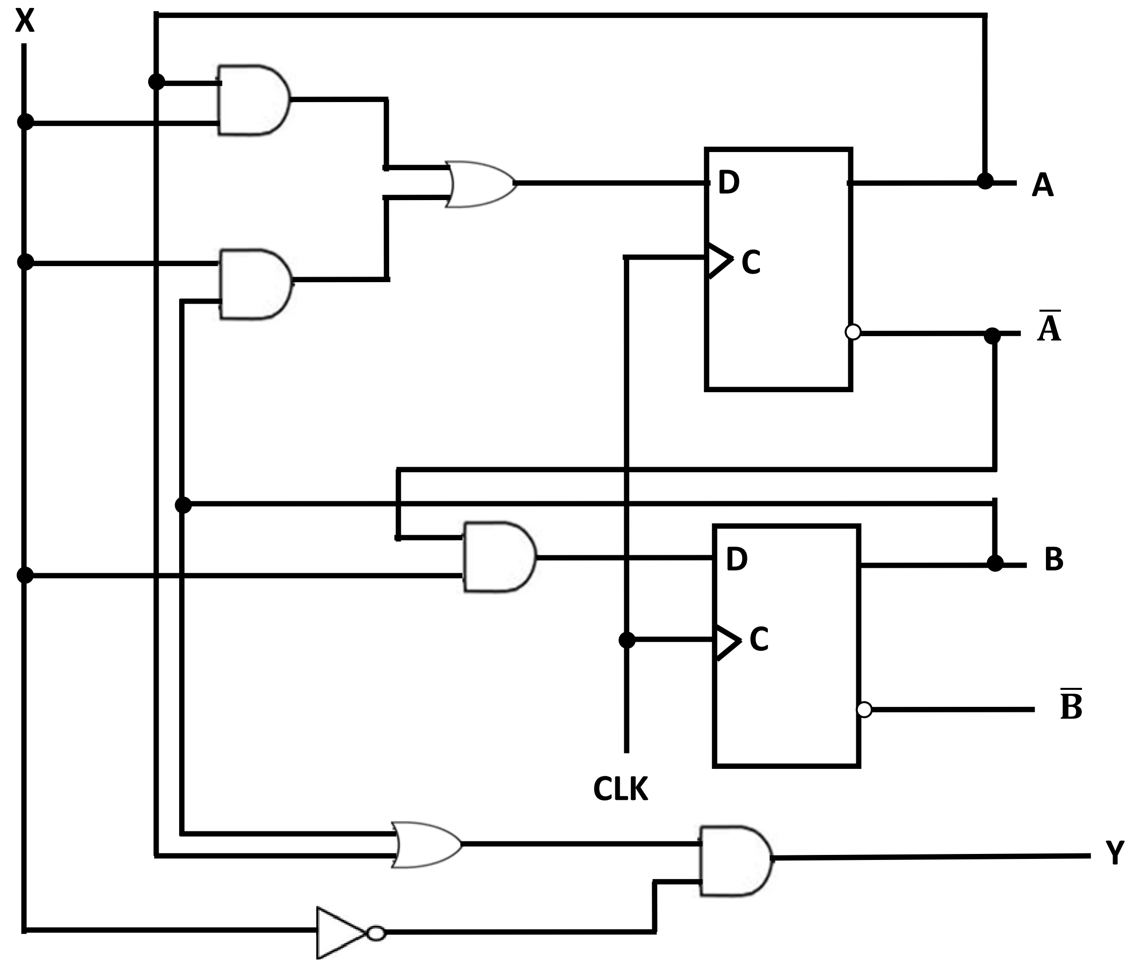


Fig 13 example of a sequential logic circuit

Sol: to draw the state table, the equations of outputs and next state must be found.

D_A= AX+BX $\dashrightarrow$ A(t+1) = AX+BX

D_B=$\overline{\mathbf{A}}$X $\dashrightarrow$ B(t+1) = $\overline{A}$X

In addition Y =$\bar{\mathbf{X}}$.(A+B)

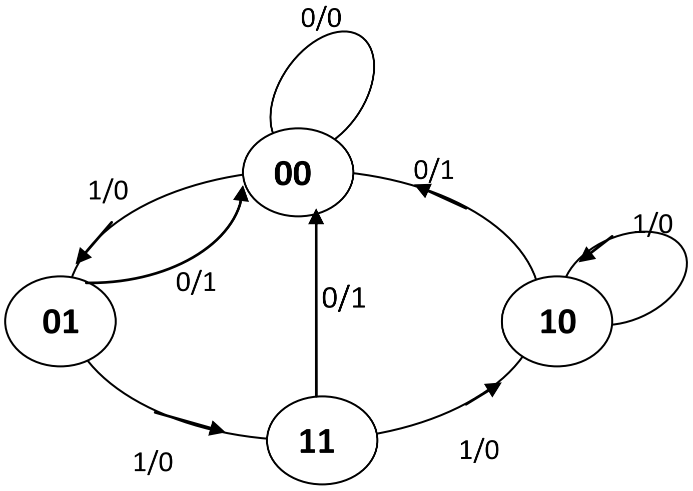
P.S.		**I/P**	**N.S.**		**O/P**
A	B	X	A	B	Y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0

Note that:

P.S is Present State

N.S is Next State

The state diagram is



This circuit is Mealy Model since the output is a function of both input (X) and present state of (A) and (B).

Ex6/ find the state table and draw the state diagram for the synchronous sequential circuit that shown in figure (14)

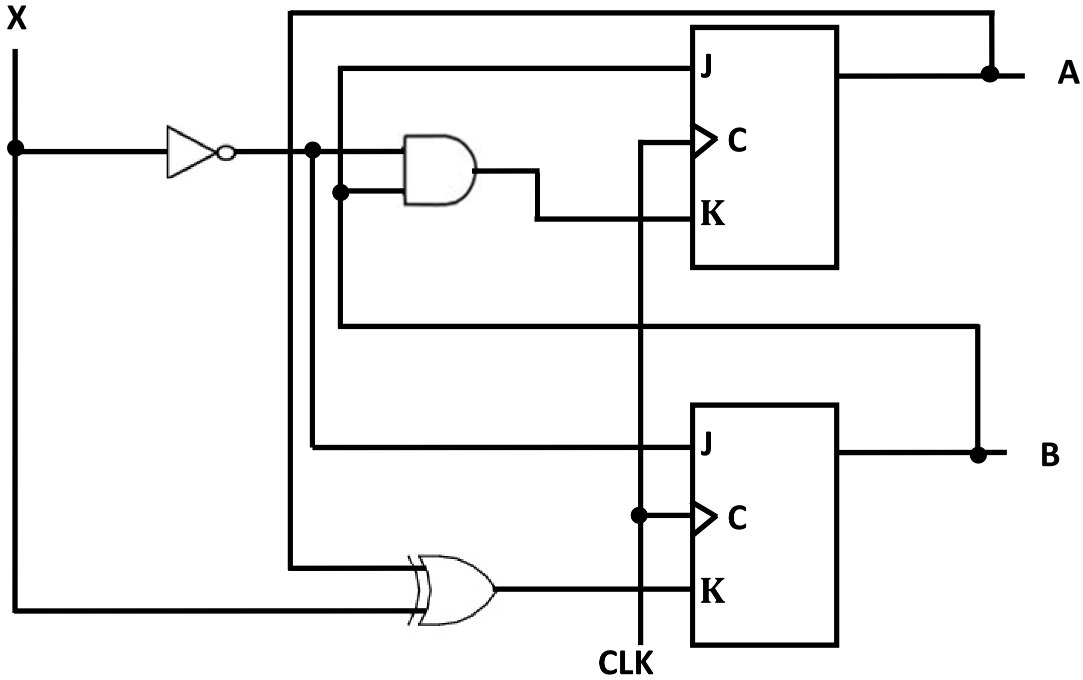


Fig 14 sequential logic circuit with J-K flip-flop

Sol:

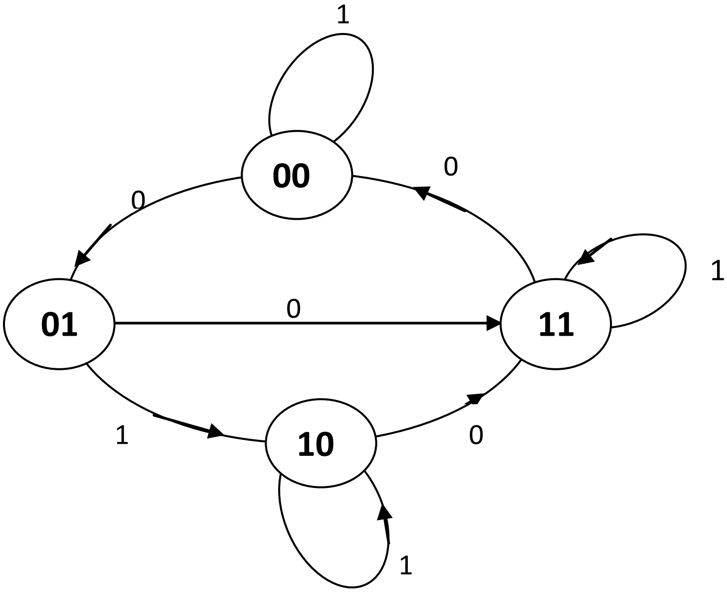
P.S.		I/P	N.S.		Flip-flop inputs			
A	B	X	A	B	J_A	K_A	J_B	K_B
0	0	0	0	1	0	0	1	0
0	0	1	0	0	0	0	0	1
0	1	0	1	1	1	1	1	0
0	1	1	1	0	1	0	0	1
1	0	0	1	1	0	0	1	1
1	0	1	1	0	0	0	0	0
1	1	0	0	0	1	1	1	1
1	1	1	1	1	1	0	0	0

The above state table can be found from the flip-flop input equations:

$J_A = B$ $K_A = B\overline{\mathbf{X}}$

$J_B = \bar{X}$ $K_B = \bar{A}X + A\bar{X}$

The state diagram of this circuit is



This circuit is Moore Model since the output is a function of present state of (A) and (B) only.

HW$_2$: for the sequential logic circuit shown in figure (15), draw the state table and state diagram.

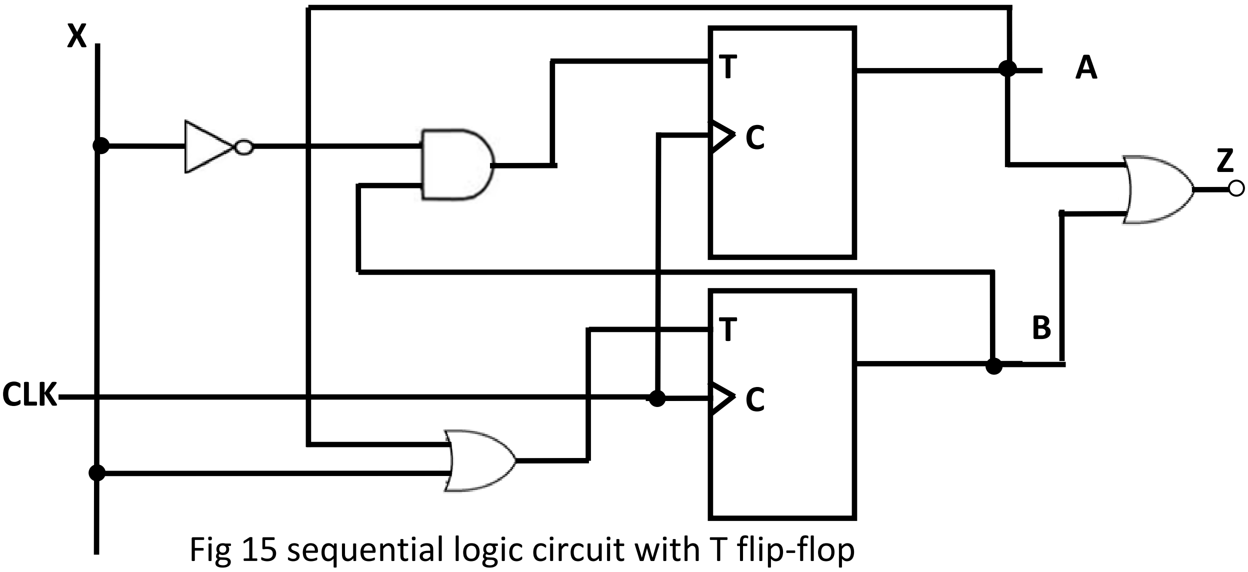


Fig 15 sequential logic circuit with T flip-flop

In the previous examples, synchronous sequential logic circuits were considered know asynchronous sequential logic circuits will be illustrated. Asynchronous sequential logic circuits are useful in a variety of applications. They are used when the speed of operation is important, especially in those cases where the digital system must respond quickly without having to wait for the clock pulse.

Ex7/ for the logic asynchronous circuit shown in figure (16), find the transition, state, and flow tables.

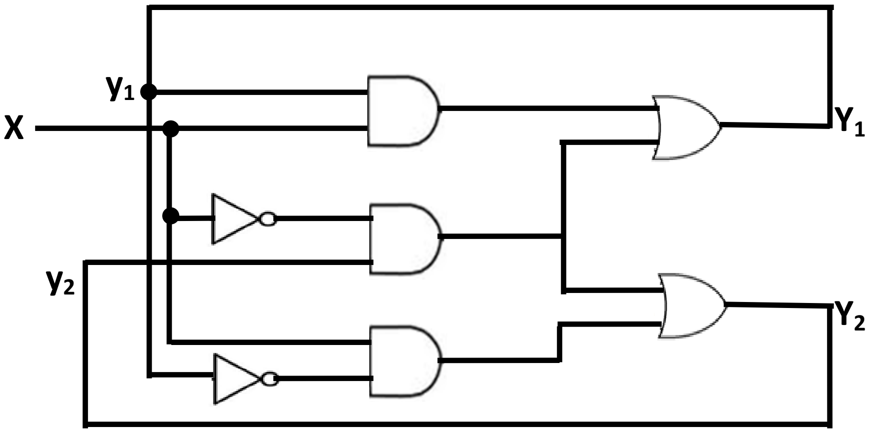


Fig 16 example of asynchronous sequential circuit

Sol: the excitation variables are: $Y_1 = Xy_1+\bar{X}y_2$, $Y_2 = X\bar{y}_1+\bar{X}y_2$

$y_1\,y_2$ \ X	0	1
00	0	0
01	1	0
11	1	1
10	0	1

$Y_1 = Xy_1+\bar{X}y_2$

$y_1\,y_2$ \ X	0	1
00	0	0
01	1	0
11	1	1
10	0	1

$Y_2 = X\bar{y}_1+\bar{X}y_2$

$y_1\,y_2$ \ X	0	1
00	00	01
01	11	01
11	11	10
10	00	10

Transition table

In the transition table {$Y = Y_1Y_2$ and $y = y_1y_2$}, when Y = y then this case represent the stable case and circled in the transition table while when Y≠ y then this case is unstable case. The effect of input (X) must be considered, when X = 0and y_1y_2 =00 then Y_1Y_2 =00 which mean stable state while when X changes to 1 then Y_1Y_2 = 01which unstable sates and this makes y_1y_2 change until reach to the stable state 01, this is another difference between the synchronous and asynchronous circuits. The output of asynchronous circuits change immediately when input changes. The state table can be obtained from the transition table, which is:

A	B	X=0		X=1	
0	0	0	0	0	1
0	1	1	1	0	1
1	0	0	0	1	0
1	1	1	1	1	0

The flow table is the same as the transition table except that the binary values are changed with letters by assigned [a =00, b=01, c =11, and d =10], and then the flow table is given as:

X	0	1
a	ⓐ	b
b	c	ⓑ
c	ⓒ	d
d	a	ⓓ

Flow table

This table is called primitive flow table since it consists of one stable state in each raw

Ex8/ draw the flow table of the circuit shown in figure (17)

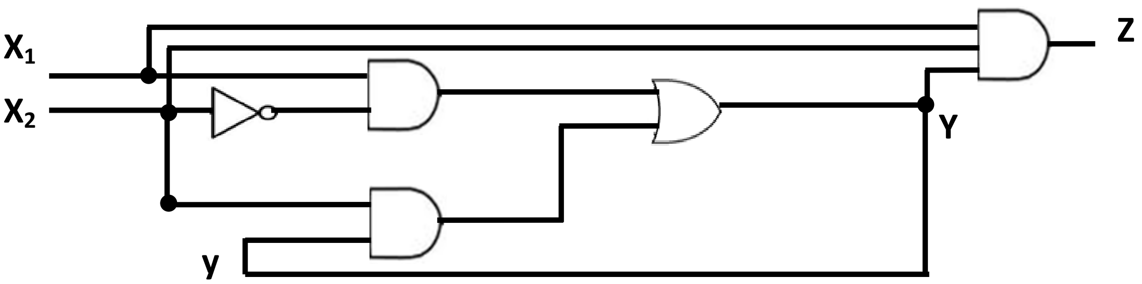


Fig 17 example of asynchronous sequential circuit

Sol: the transition table of Y and K-Map of the output Z are given below:

X_1X_2 \ y	0	1
00	0	0
01	0	0
11	0	1
10	1	1

$Y = X_1\overline{X}_2+X_1\,y$

$y_1\,y_2$ \ X	0	1
00	0	0
01	0	0
11	0	1
10	0	0

$Z = X_1X_2\,y$

The flow table is:

X_1X_2 \ y	a	b
00	a,0	a,0
01	a,0	a,0
11	a,0	b,1
10	b,0	b,0

Two states with two inputs and one output

Ex9/ find the logic circuit that gives the following flow table.

$y_1 y_2$ \ X_1X_2	00	01	11	10
a	b,0	c,0	c,1	a,0
b	b,0	c,0	c,1	a,0
c	c,0	c,0	a,0	a,0
d	d,0	d,1	a,1	a,0

Sol: the transition table of the logic circuit is :

$y_1 y_2$ \ X_1X_2	00	01	11	10
00	01	11	11	00
01	01	11	11	00
11	11	11	00	00
10	10	10	00	00

$y_1 y_2$ \ X_1X_2	00	01	11	10
00	1	1	1	0
01	1	1	1	0
11	1	1	0	0
10	0	0	0	0

$Y_2=\overline{X}_1\overline{y}_1+X_2\overline{y}_1+\overline{X}_1y_2$

$y_1 y_2$ \ X_1X_2	00	01	11	10
00	0	1	1	0
01	0	1	1	0
11	1	1	0	0
10	1	1	0	0

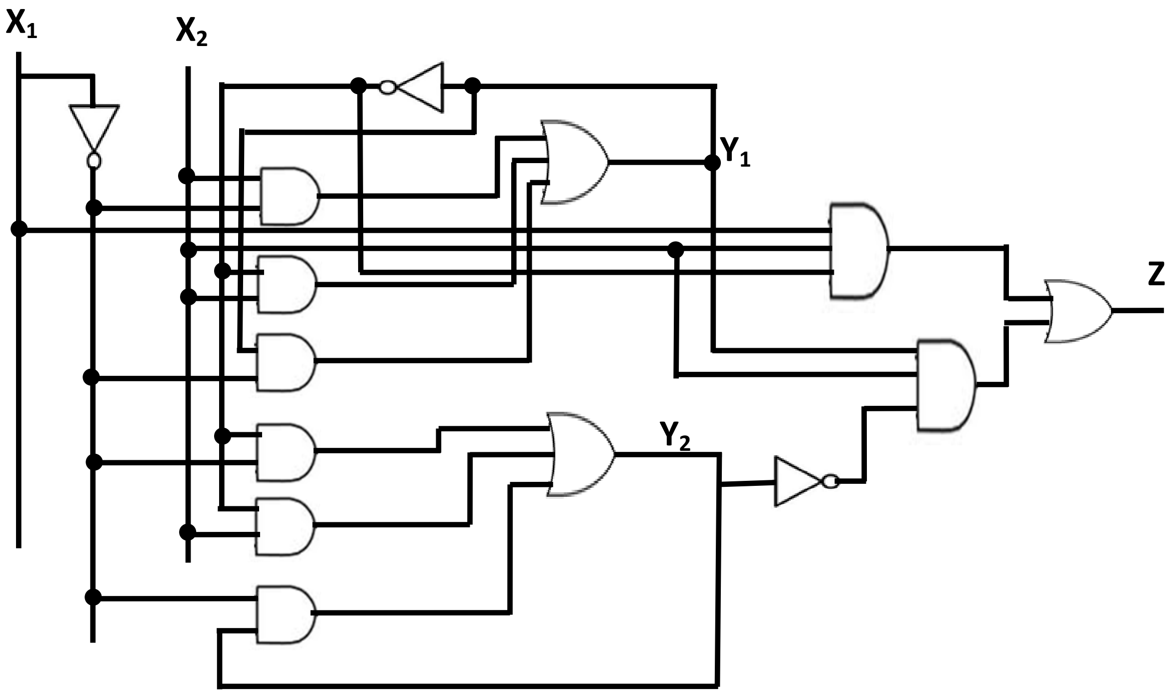
$Y_1=\overline{X}_1X_2+X_2\overline{y}_1+\overline{X}_1y_1$

The output assume to be Z can be found from the following K-Map.

$y_1 y_2$ \ X_1X_2	00	01	11	10
00	0	0	1	0
01	0	0	1	0
11	0	0	0	0
10	0	1	1	0

$$Z=X_1X_2\bar{y}_1+X_2y_1\bar{y}_2$$

The sequential logic circuit is:

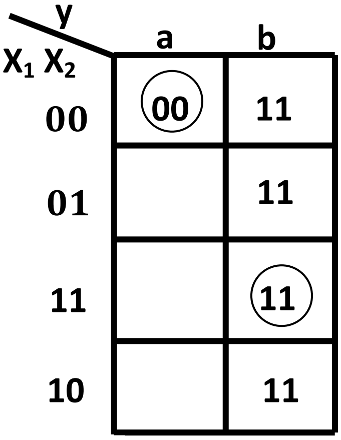


HW$_3$: for the following flow table, find the logic circuit.

y \ X_1 X_2	00	01	11	10
0	a,01	b,10	b,01	a,11
1	b,00	b,00	a,00	a,00

❖ Race Condition

A race condition is said to exist in an asynchronous sequential circuit when two or more binary state variables change value in response to a change in an input variable. When unequal delays are encountered, a race condition may cause the state variables to change in an unpredictable manner. For example, if the state variables must change from 00 to 11, the difference in delays may cause the first variable to change sooner than the second, with the result that the state variables change in sequence hm 00 to 10 and then to 11. If the second variable changes sooner than the first, the state variables will change from 00 to 01 and then to 11. Thus, the order by which the state variables change may not be known in advance. If the final stable state that the circuit reaches does not depend on the order in which the state variables change, the race is called a noncritical race. If it is possible to end up in two or more different stable states, depending on the order in which the state variables change, then the race is a critical race. For proper operation, critical races must be avoided. The following two figures (18&19)show the noncritical race and critical race cases respectively.



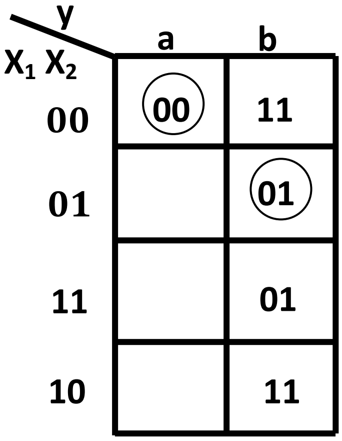
X_1X_2 \ y	a	b
00	00	11
01		11
11		11
10		11

00 → 11

00 → 01 → 11

00 → 10 → 11

(18-a)



X_1X_2 \ y	a	b
00	00	11
01		01
11		01
10		11

00 → 01

00 → 11 → 01

00 → 10 → 11 → 01

(18-b)

Fig 18 Noncritical race

X_1X_2 \ y	a	b
00	$\textcircled{00}$	11
01		$\textcircled{01}$
11		$\textcircled{11}$
10		$\textcircled{10}$

00 → 11

00 → 01

00 → 10

(19-a)

X_1X_2 \ y	a	b
00	$\textcircled{00}$	11
01		11
11		$\textcircled{11}$
10		$\textcircled{10}$

00 → 11

00 → 01 → 11

00 → 10

(19-b)

Fig 19 Critical race

❖ Stability Condition

The system may be unstable if the changing of its inputs gives unstable case and makes {Y≠y} or gives a cycle of unstable cases such as given in the following figures (20-a, 20-b).

X_1X_2 \ y	a	b
00	$\textcircled{00}$	11
01		11
11		10
10		01

→00 → 11 → 10 ⟲

20-a

y \ X_1X_2	00	01	11	10
0	$\textcircled{0}$	1	1	$\textcircled{0}$
1	0	$\textcircled{1}$	0	0

When X_1X_2 change to 11 then Y will never equal to y

20-b

Fig 20 unstable cases

❖ S-R Latch

From the logic circuit of S-R latch, the logic circuit using NOR and using NAND gates are shown in figures (21&22) respectively.

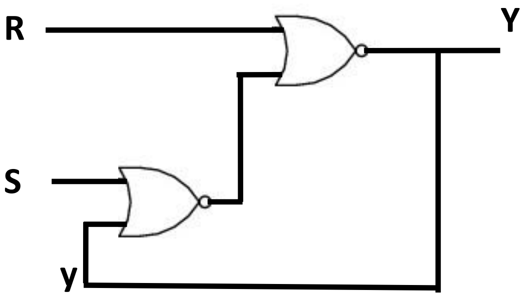


Fig 22 S-R Latch using NAND

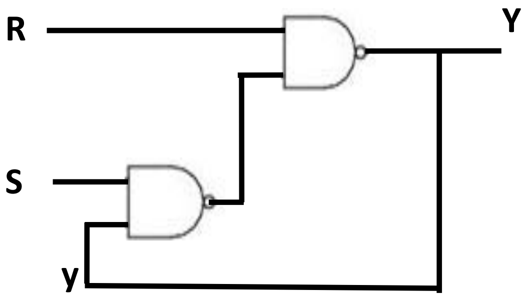


Fig 22 S-R Latch using NAND

The excitation function of the logic circuit shown in figure (21) is

$$Y = Q = \overline{\overline{S + y} + R} \quad \Longrightarrow \quad Y = Q = (S + y)\bar{R} = S\bar{R} + \bar{R}\,y$$

For this circuit the inputs are (S&R) therefore the effect of changing them on the output must be studied carefully,

1- SR = 10 then Y = 1
2- SR = 00 then Y = 1
3- SR = 01 then Y = 0
4- SR = 00 then Y = 0
5- SR = 11 then Y = independent case since ($Q = \bar{\mathbf{Q}} = 0$)

The transition table of this circuit is

y \ $X_1 X_2$	00	01	11	10
0	⓪	⓪	⓪	1
1	①	0	0	①

An important note that SR≠1 or always SR=0 from this note, it can be found that $S\overline{\mathbf{R}}+SR = S(R+\overline{\mathbf{R}}) = S$ then $S\overline{\mathbf{R}}+SR = S$ and this mean that $S\overline{\mathbf{R}}=S$, now the excitation function $Y = S + \overline{\mathbf{R}}\,y$ as $SR = 0$.

The excitation function of the logic circuit shown in figure (22) is

$$Y=\overline{\overline{S(Ry)}} \quad \Longrightarrow \quad Y=\overline{S}+Ry \text{ for } \overline{S}\,\overline{R}=0$$

The transition table of this circuit is

y \ X_1X_2	00	01	11	10
0	1	1	0	0
1	1	1	1	0

Ex10/ find the transition table for the circuit shown in figure (23).

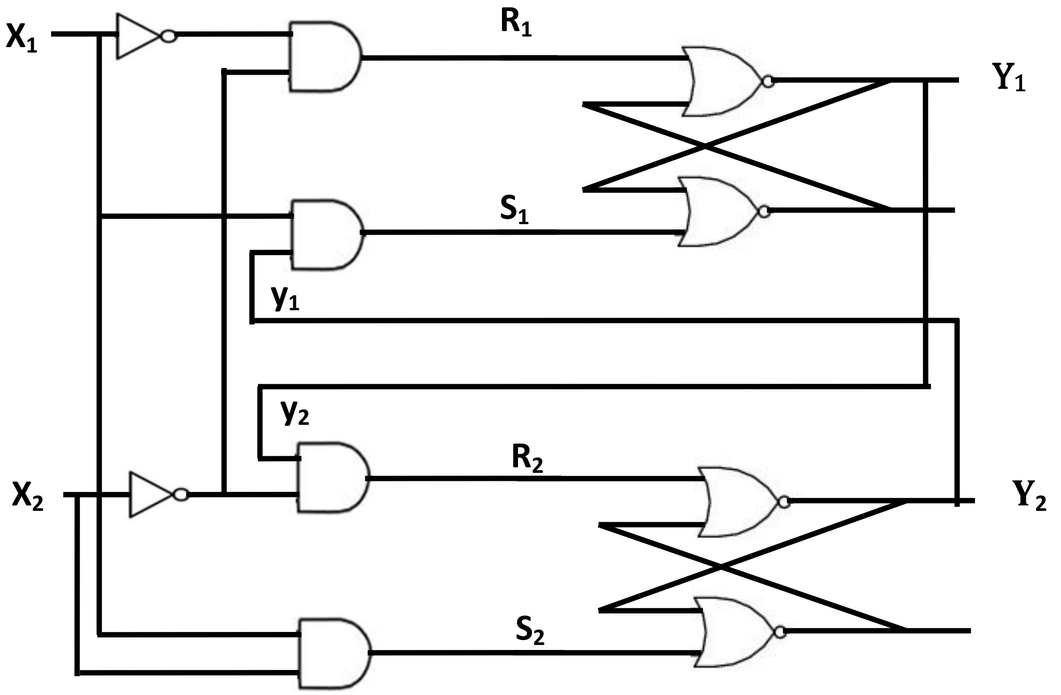


Fig 23 example of a sequential circuit with S-R Latch

The Boolean expressions for the S-R latch inputs are:

$S_1= X_1y_1$, $R_1=\bar{X}_1\bar{X}_2$, and $S_2= X_1X_2$, $R_2=\bar{X}_2y_1$

The proper condition for the S-R latch is SR = 0, then

$S_1R_1=0 \Longrightarrow X_1y_1\bar{X}_1\bar{X}_2 =0$ and $S_2R_2=0 \Longrightarrow X_1X_2\,\bar{X}_2y_1 =0$

These results are zeros since $X_1\,\bar{X}_1 = 0$& $X_2\,\bar{X}_2 = 0$,

$Y_1 = S_1 + \bar{R}_1y_1 = X_1y_2 +(X_1+X_2)\,y_1 = X_1y_2 + X_1y_1 +X_2y_1$

$Y_2 = S_2 + \bar{R}_2y_2= X_1X_2 +(X_2+\bar{y}_1)\,y_2 = X_1X_2 + X_2y_2 +\bar{y}_1y_2$

The transition table of this logic circuit is

X_1X_2 / $y_1\,y_2$	00	01	11	10
00	$\textcircled{00}$	$\textcircled{00}$	01	$\textcircled{00}$
01	$\textcircled{01}$	$\textcircled{01}$	11	11
11	00	$\textcircled{11}$	$\textcircled{11}$	10
10	00	$\textcircled{10}$	11	$\textcircled{10}$

❖ Hazard

Hazards are unwanted switching transients that may appear at the output of a circuit because different paths exhibit different propagation delays. Hazards occur in combinational circuits, where they may cause a temporary false output value. When they occur in asynchronous sequenti1 circuits, hazards may result in transients to a wrong stable state. A hazard is a condition in which a change in a single variable produces a momentary change in output when no change in output &odd occur.

Generally, there is two types of hazard which are:

1- Static Hazard

Considering the logic circuit shown in figure (24) which consists of an real NOT gate and ideal X-NOR gate, it is found that the output of NOT gate will delay and these leads to {A.$\bar{\mathbf{A}}$ ≠ 0 and A+$\bar{\mathbf{A}}$ ≠1}. Timing diagram shows the effect of the delay of NOT gate as illustrated in figure (25).

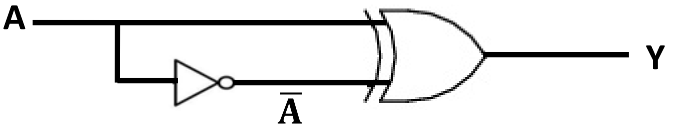


Fig 24 Example of 0-Hazard

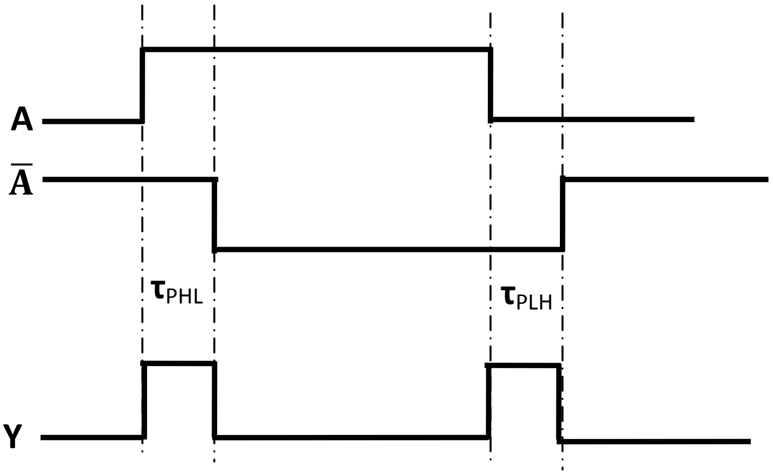


Fig 25 Timing diagram

Ex11/ what is the type of hazard for the circuit given in figure (24)

Sol: the Boolean expression of the output is

$$Y = \overline{A \oplus \bar{A}} = \overline{A\bar{\bar{A}} + \bar{A}\bar{A}} \Longrightarrow Y = \overline{AA+\bar{A}\bar{A}}$$

$$= \overline{A+\bar{A}}$$

$$= \bar{1} \Longrightarrow Y = 0$$

This shows that for an ideal gate the output Y will always be 0 because [A + A = 1]. However, with a real NOT gate this cannot be guarateed (because of the propagation delay of the gate) and so the hazard occurs when this condition is not met.

Ex12/ avoid the hazard occurs in the circuit shown in figure (26).

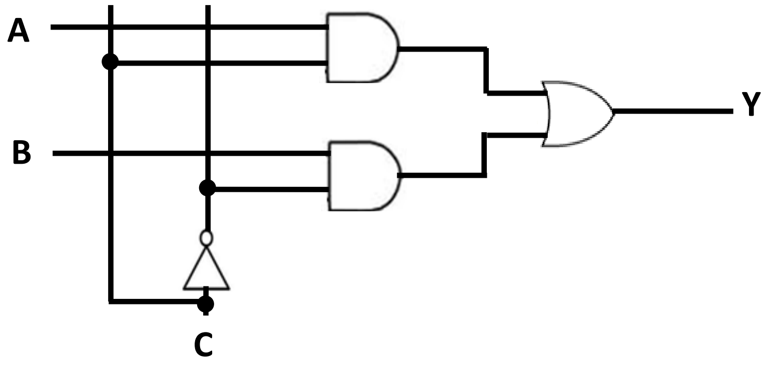
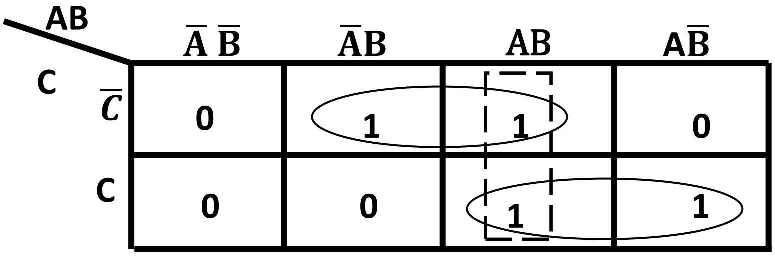


Fig 26 2*1 MUX logic circuit

Sol: the truth table of this logic circuit is

Inputs		Selector	Output
A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

The Boolean expression that can be obtained from the truth table using the following K-Map is $Y = AC+B\bar{C}$.



$C \backslash AB$	$\bar{A}\,\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
$\bar{C}$	0	1	1	0
C	0	0	1	1

The important point is that this minimization depends upon the use of the Boolean identity C+C= 1, which we know, because of the action of the NOT gate, is not always true and may introduce a hazard. This suggests how we may eliminate the static hazard, which is to include the non-essential prime implicant so that the expression for Y will not then have been minimized using this identity. To see the effect of including the AB term, we note that if we use the Boolean expression that obtained as:

$$Y = \bar{A}B\bar{C} + A\bar{B}C+AB\bar{C}+ABC$$

$$= \bar{A}B\bar{C} + A\bar{B}C+AB(C+\bar{C})$$

$$= \bar{A}B\bar{C} + A\,[B+\bar{B}C]$$

$$= \bar{A}B\bar{C} + A\,[B\,(1+C) +\bar{B}C]$$

$$= \bar{A}B\bar{C} + A\,[B+BC +\bar{B}C]$$

$$= \bar{A}B\bar{C} + A\,[B+C\,(B +\bar{B})]$$

$$= \bar{A}B\bar{C} + AB+AC$$

$$= AB+ AC+B\bar{C}$$

Then if ((A = 1) AND (B= 1)) (the conditions which lead to the hazard, and define the non-essential prime implicant) this expression becomes:

$Y = C + 1 + \bar{C} = 1$.

The logic circuit will be as given in figure (27)

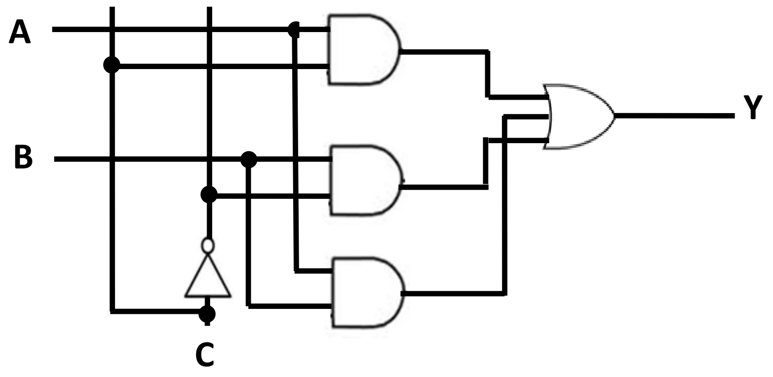


Fig 26 (2*1) MUX logic circuit

Generally, static hazards can be eliminated by the inclusion of nonessential prime implicants in the 'minimized' Boolean expression. More specifically they will be non-essential prime implicants whose 'removal' relies upon the use of X+ X= 1, where X is a variable for which

Ex13/ Derive the product of sums form of a 2-to-1 multiplexer and then, performing the corresponding analysis to that for the sum of products form, determine whether any static hazards occur, and if they do how they may be eliminated.

Sol: For K-Map in (Ex12) looping and grouping the zeros gives:

$$\bar{Y}=\bar{A}C+\bar{B}\bar{C}$$

$$\Longrightarrow$$

$$Y=(A+\bar{C})(B+C)$$

Due to the final product produced we anticipate a hazard if both racing versions of C reach the AND gate. For a hazard to occur requires

$C.\bar{C}$ Which needs both A=0and B=0.static-0 hazard produced as C goes high.

Using Boolean algebra to confirm this, from the Karnaugh map using all'prime implicants' for $\bar{Y}$:

$\bar{Y}=\bar{A}C+\bar{B}\bar{C}+\bar{A}\,\bar{B}$

Getting the product of sums form, and using the fact that C. C=0:

$Y=(A+\bar{C})\,(B+C)\,(A+B)$

$=(A+\bar{C})\,(B+C)\,(A+B+\bar{C}C)$

$=(A+\bar{C})\,(A+B+\bar{C})\,(B+C)\,(A+B+C)$

$=(A+\bar{C})\,(1+B)\,(B+C)\,(1+A)$

$=(A+\bar{C})\,(B+C)$

So the minimization process relies upon the fact that $C\bar{C}=0$ which is where the hazard arises from. The equivalent 'blanking gate' in this product of sums implementation is (A + B) since for a hazard to occur both A and B must be 0 meaning A +B=0. This gate will hold the output, Y, low, thus preventing the static-0 hazard. Note the similarity between the sum of products and product of sums forms which is again a consequence of duality. A static hazard arise because of the fact that $C+\bar{C}\neq 1$, and therefore will occur when C changes state. In addition for a static hazard to occur the change in C must not cause a change in the output (Y).

2- Dynamic hazards

Dynamic hazards can occur when a signal has three or more paths through a combinational logic circuit. Their effect is to cause a signal, which is expected to change state to do so, then transiently change back to the original state, before making the final transition to the expected state, (e.g. the signal gives 1010 rather than just 10). Consider the circuit shown in figure (27) with its truth table and K-Map.

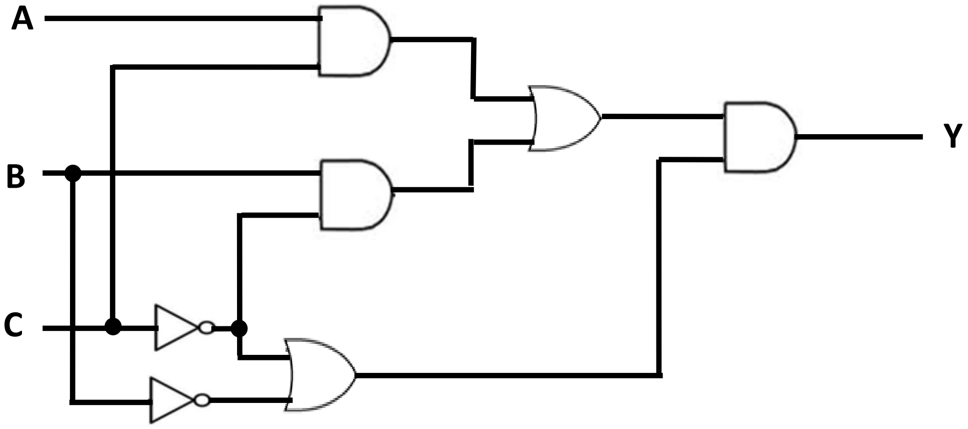
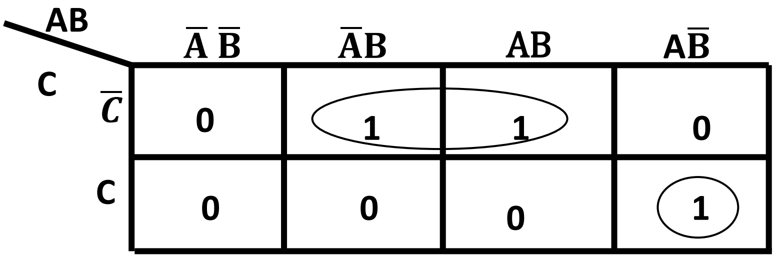


Fig 27 example of combinational logic circuit



AB \ C	$\overline{A}\,\overline{B}$	$\overline{A}B$	AB	$A\overline{B}$
$\overline{C}$	0	1	1	0
C	0	0	0	1

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

From this implementation, the value of output is:

Y= ($\overline{\mathbf{B}}$+$\overline{\mathbf{C}}$) (AC+B$\overline{\mathbf{C}}$)

In addition, note that C has three paths through the circuit. Therefore, there is the possibility of race conditions in three paths, which may lead to a dynamic hazard.

From the truth table we see that for inputs of (A, B) of (0, 1), (1, 0) and (1, 1), the output, Y, changes as C changes. Therefore, there is the possibility of a dynamic hazard as the output may change state, then go back to the initial value, before changing again to the final expected output. Continuing our analysis we see that if B = 0, then $\bar{\mathbf{B}}$ = 1 and so the output, (1+$\bar{\mathbf{C}}$) is always 1. This means C now only has two paths through the circuit and so a dynamic hazard is not possible. Similarly, if A =0 then the output will be 0 and so again there are only two paths for C through the circuit. For A = 1 and B=1, the output equation is:

$$Y= (\bar{\mathbf{B}}+\bar{\mathbf{C}}_1)\,(AC_2+B\bar{\mathbf{C}}_3)\rightarrow Y= (\mathbf{0}+\bar{\mathbf{C}}_1)\,(1.C_2+1.\bar{\mathbf{C}}_3)$$

$$= \bar{\mathbf{C}}_1.\,(C_2+\bar{\mathbf{C}}_3)$$

The truth table for the transient values of Y as a function of the three values of C is shown in the following table.

C_1	C_2	C_3	Y
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

For instance for C changing from 0 to 1, then if C_3 changes first, then C_2 followed by C_1, then the circuit will move from row 1 to 8,transiently visiting rows 2 and 4. This will give transient outputs of 0 and 1 (for rows 2 and 4) and hence a possible dynamic hazard since the output will give 1010.This is the only possible hazard for C going from 0 to 1 because of the need for the output to go to 0 first to give the incorrect transient state. For C changing from 1 to 0, we note the possible dynamic hazard if the transition is from row 8 to 1 via rows 4 and 2 (the reverse route to the above). Again this is the only possibility since we need the output, Y, to go HIGH first (so it can go LOW again to give the hazard). The output here will be 0101. So we have identified two possible situations in which a dynamic hazard may occur, depending upon the relative speed of propagation of C_1, C_2 and C_3. The timing diagram in figure (28) illustrates how this dynamic hazard will occur for these conditions. Note that C must go from 1 to 0 and so C, will do the opposite as shown in the figure. This concludes our analysis of this circuit for dynamic hazards

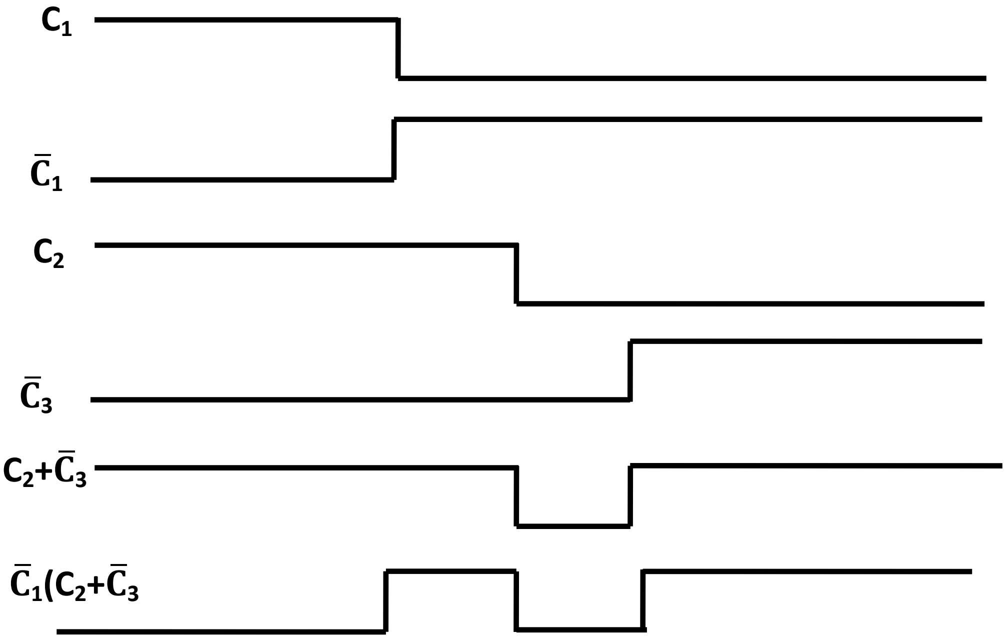


Fig 28 Timing diagram illustrating the dynamic hazard for the circuit. in figure(27)

❖ Hazards in Sequentla1 Circuits

In normal combinational circuit design associated with synchronous sequential circuits, hazards are of no concern, since momentary erroneous signals are not generally troublesome. However, if a momentary erroneous signal is fed back in an asynchronous sequential circuit, it may cause the circuit to go to the wrong stable state. This situation is illustrated in figure (29). If the circuit is in total stable state YX_1X_2 = 1 1 1 and input X_2 changes from 1 to 0. The next total stable state should be 110. However, because of the hazard, output Y may go to 0 momentarily.

If this false signal feeds back into gate 2 before the output of the inverter goes to 1, the output Y of gate 2 will remain at 0 and the circuit will switch to the incorrect total stable state 010. This malfunction can be eliminated by adding an extra gate.

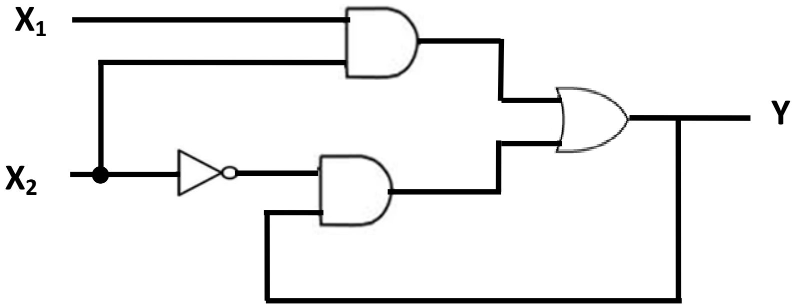


Fig 27 Hazard in an asynchronous sequential logic circuit

3- Essential Hazards

Thus far, we have considered what are known as static and dynamic hazards. Another type of hazard that may occur in asynchronous sequential circuits is called an essential hazard. This type of 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 comparison to the delay associated with the feedback path may cause such a hazard. Essential hazards cannot be corrected by adding redundant gates as in static hazards. The problem that they impose can be corrected by adjusting the amount of delay 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. This problem tends to be specialized, as it depends on the particular circuit used and the size of the delays that are encountered in its various paths.