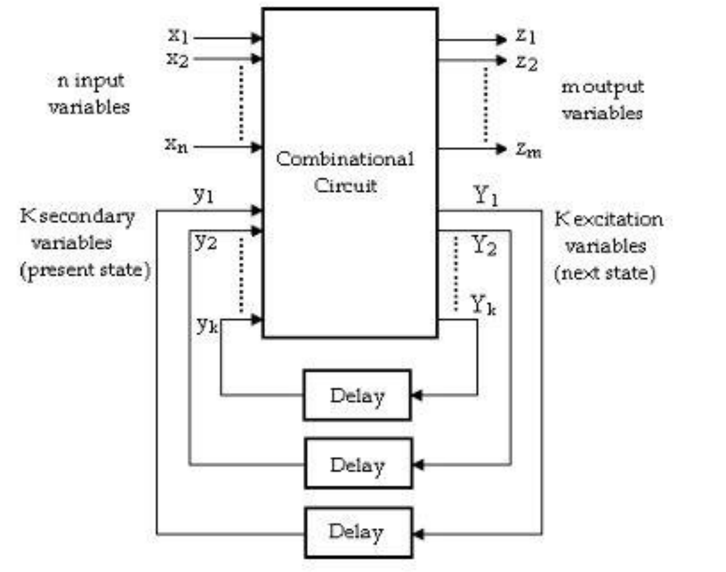
Digital System Design EC 503

Analysis and Synthesis of Asynchronous Sequential Circuits

Subject Coordinator
Dr. GAURAV KUMAR BHARTI
Department of ECE

Asynchronous sequential circuits

- Memory elements are either unclocked flip flops or time delay elements.
- The change in input signals can affect memory element at any instant of time.
- Because of the absence of clock, it can operate faster than synchronous circuits.
- Asynchronous sequential circuits do not use clock pulses.
- The memory elements in asynchronous sequential circuits are either unclocked flip-flops (Latches) or time-delay elements.



Block diagram of Asynchronous sequential circuits

- According to input variables there are two types
  - ➢Fundamental mode circuit

    –The input variables change only when the circuit is stable. Only one input variable can change at a given time.

    –Inputs are levels (0, 1) and not pulses.
  - ➢ Pulse mode circuit

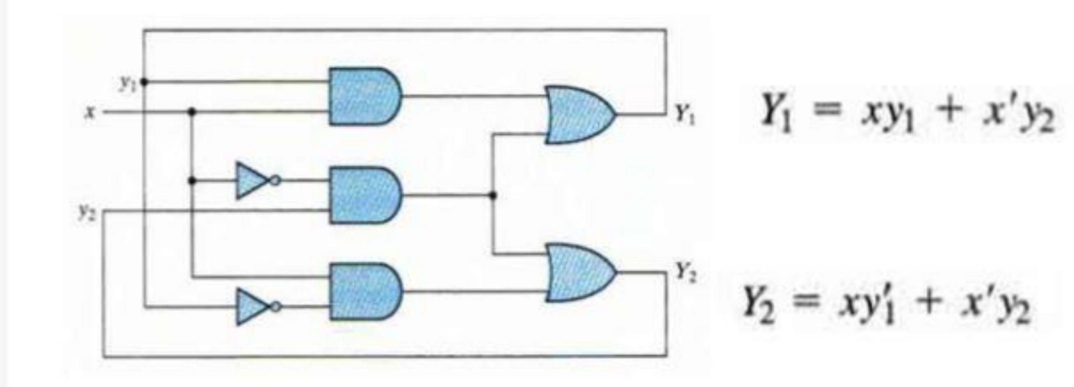
    –The input variables are pulses (True, False) instead of levels.

    –The width of the 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.
  - ➢Analysis of Sequential Circuits

    The analysis of Asynchronousous sequential circuits consists of obtaining a table or a diagram that describes the sequence of internal states and outputs as a function of changes in the input variables.

Analysis Procedure

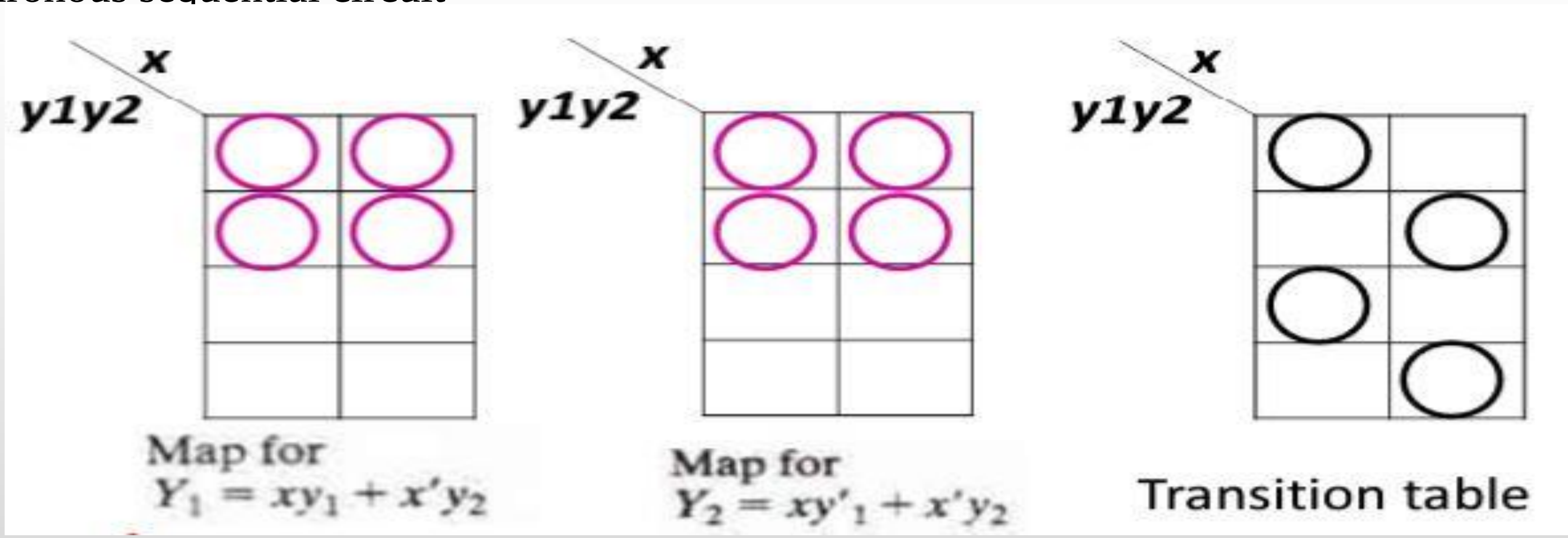


$Y_1 = xy_1 + x'y_2$

$Y_2 = xy_1' + x'y_2$

Fig: Example of Asynchronous sequential circuit

Total State
Four stable total states – y1y2x = 000, 011, 110, and 101
Four unstable total states – 001, 010, 111, and 100



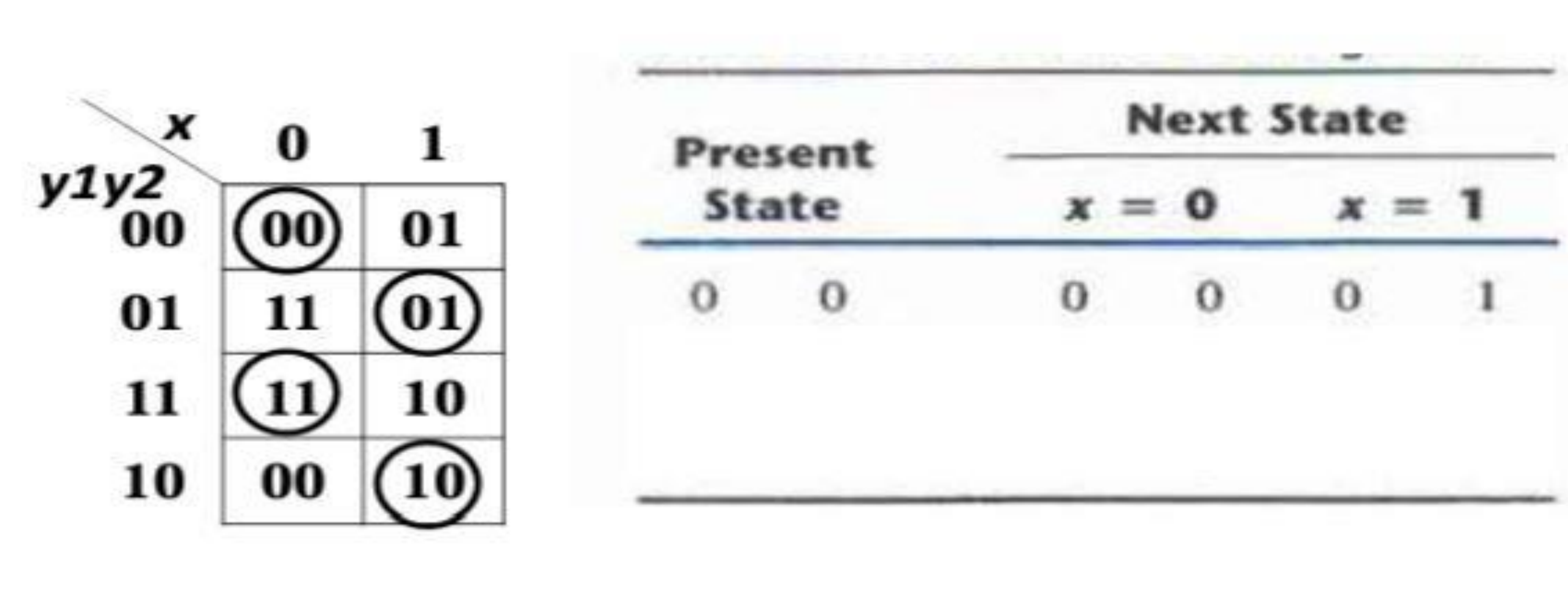
Map for $Y_1 = xy_1 + x'y_2$

Map for $Y_2 = xy'_1 + x'y_2$

Transition table

Fig: Transition Table

- The transition table of asynchronous sequential circuit is similar to the state table used for synchronous circuits



y1y2 \ x	0	1
00	00	01
01	11	01
11	11	10
10	00	10

Present State		Next State			
		$x = 0$		$x = 1$	
0	0	0	0	0	1

The procedure for obtaining a transition table from the given circuit diagram is as follows.

1. Determine all feedback loops in the circuit.

2. Designate the output of each feedback loop with variable Y1 and its corresponding inputs y1, y2,….yk, where k is the number of feedback loops in the circuit.

3. Derive the Boolean functions of all Y's as a function of the external inputs and the y's.

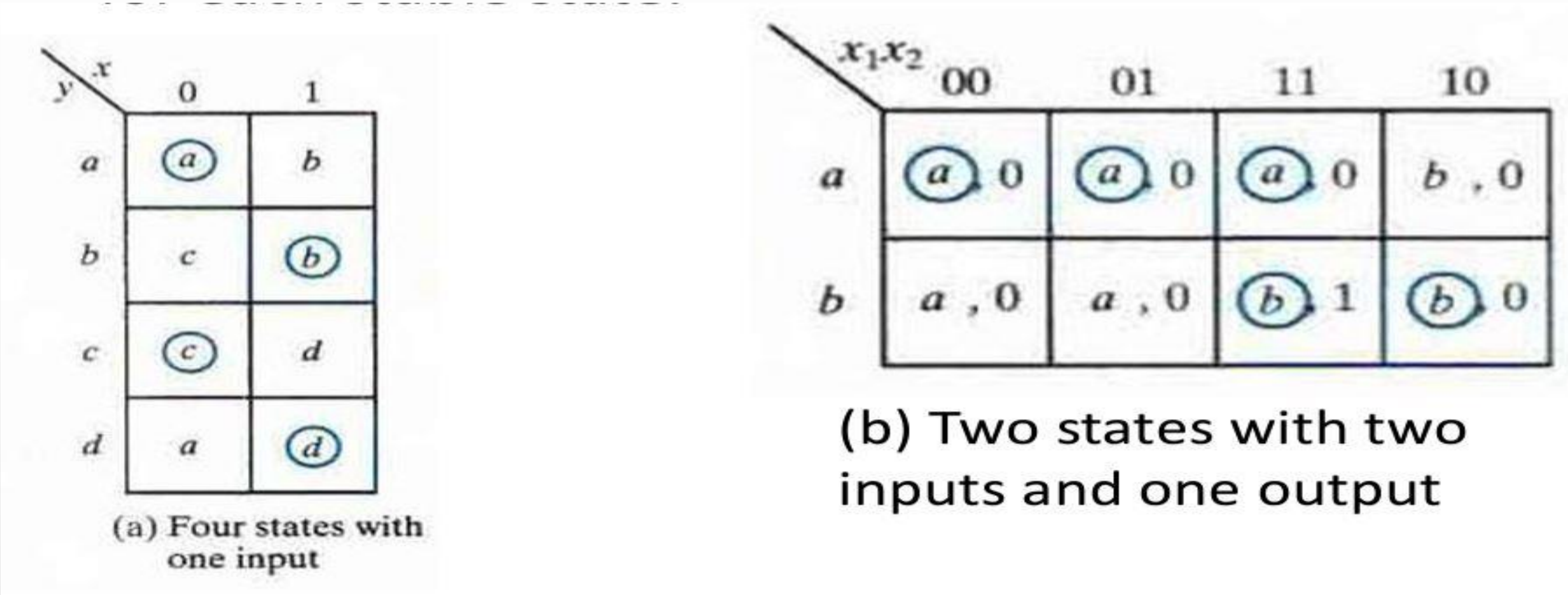
4. Plot each Y function in a map, using y variables for the rows and the external inputs for the columns.

5. Combine all the maps into one table showing the value of Y= Y1, Y2,….Yk inside each square.

6. Circle all stable states where Y=y. The resulting map is the transition table. Once the transition table is available, the behavior of the circuit can be analyzed by observing the stale transition as a function of changes in the input variables.

Flow Table

- During the design of asynchronous sequential circuits, it is more convenient to name the states by letter symbols than binary values.
- Such a table is called an flow table and is similar to a transition table, except that the internal states are symbolized with letters rather than binary numbers.
- The flow table also includes the output values of the circuit for each stable state.



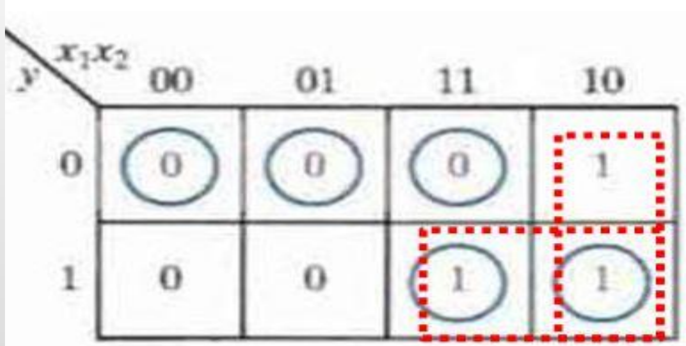
y \ x	0	1
a	ⓐ	b
b	c	ⓑ
c	ⓒ	d
d	a	ⓓ

(a) Four states with one input

\ x_1x_2	00	01	11	10
a	ⓐ, 0	ⓐ, 0	ⓐ, 0	b , 0
b	a , 0	a , 0	ⓑ, 1	ⓑ, 0

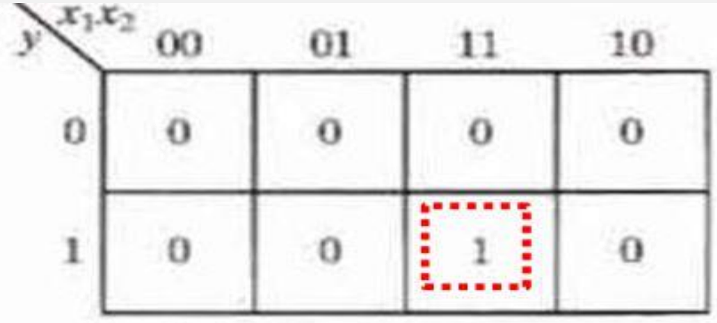
(b) Two states with two inputs and one output

- If a transition table has only one stable state in each row then it is called as ***primitive flow table***
- Figure (a) is called a primitive flow table because it has only one stable state in each row.
- Figure (b) shows a now table with more than one stable state in the same row.
- The binary value of the output variable is indicated inside the square next to the state symbol and is separated from the state symbol by a comma.
- To obtain the circuit described by a flow table, it is necessary to assign a distinct binary value to each state.
- Such an assignment converts the flow table into a transition table from which we can derive the logic diagram.
- Assign 0 to state a and 1 to state b, the result is the transition table
- The output map is obtained directly from the output values in the flow table



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

(a) Transition table
$Y = x_1x'_2 + x_1y$



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

(b) Map for output
$z = x_1x_2y$

$$Y = x_1x'_2 + x_1y'$$

$$z = x_1x_2\,y$$

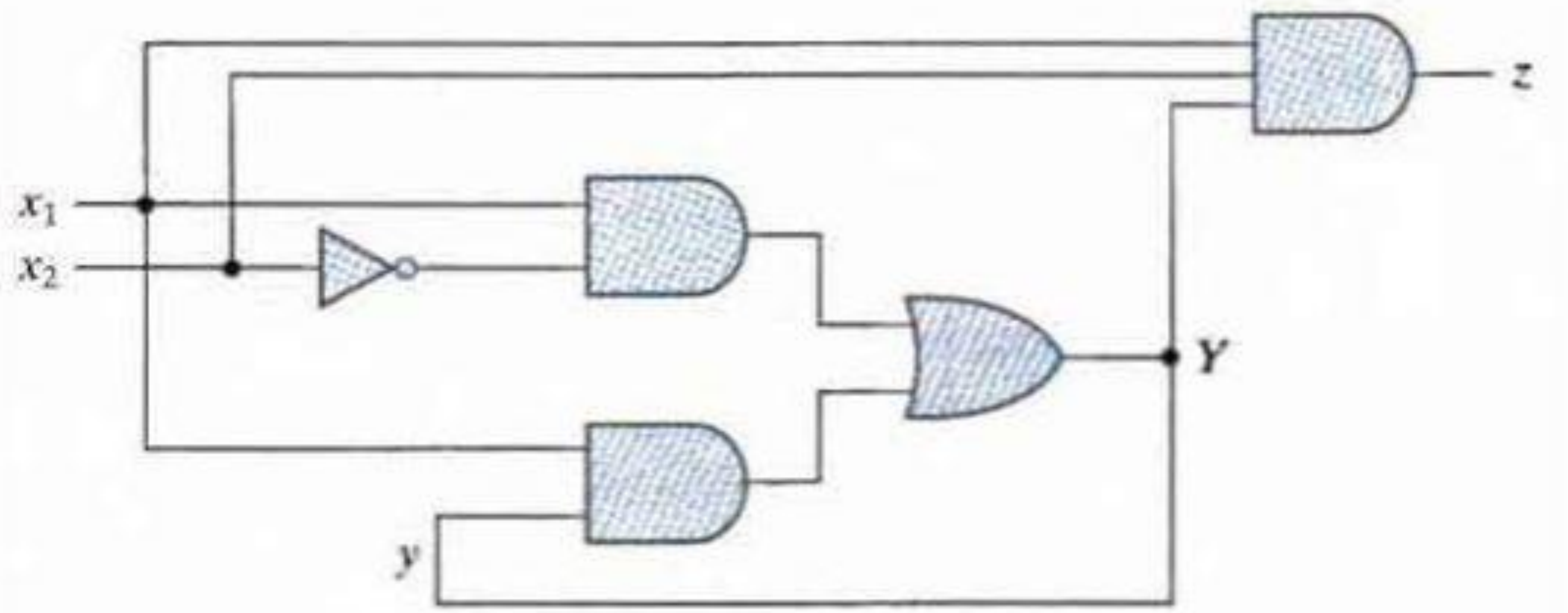


Fig: Logic Diagram

Race Conditions

- 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 stale variables to change in an unpredictable manner.

- Races are classified as:

  i. Non-critical races

  ii. Critical races

Non-critical races

- 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 non-critical race.
- If a circuit, whose transition table starts with the total stable state y1y2x= 000 and then change the input from 0 to 1. The state variables must then change from 00 to 11, which define a race condition.
- The possible transitions are:

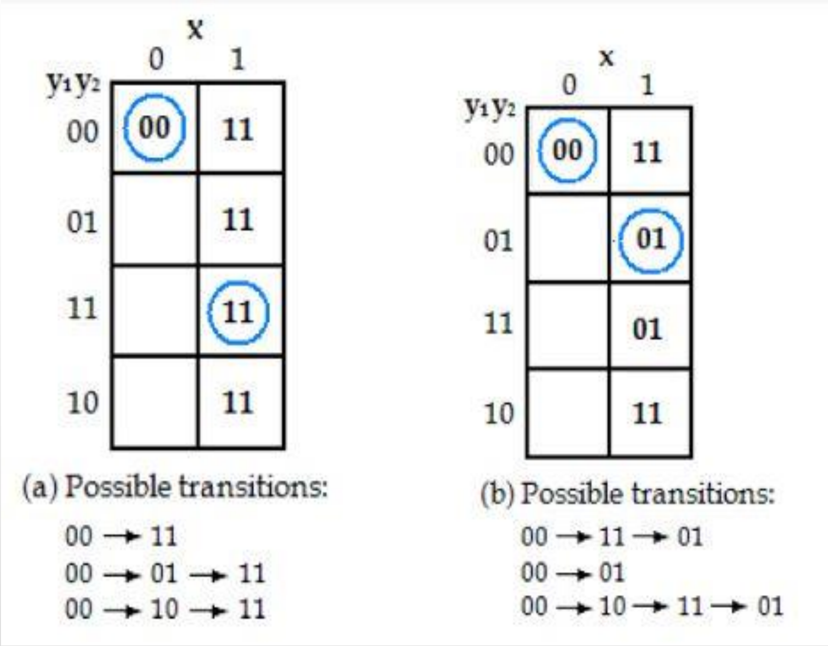
00 → 11

00 → 01 → 11

00 → 10 → 11

In all cases, the final state is the same,

which results in a non-critical condition



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

(a) Possible transitions:

00 → 11
00 → 01 → 11
00 → 10 → 11

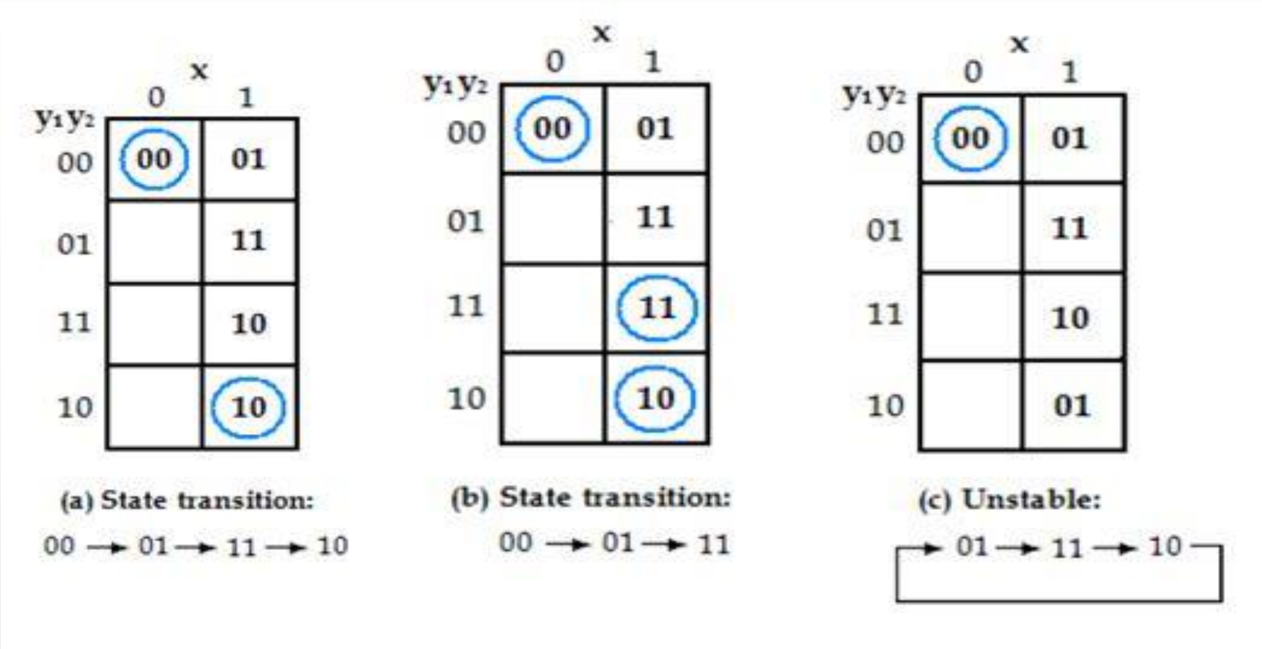
y_1y_2 \ x	0	1
00	00	11
01		01
11		01
10		11

(b) Possible transitions:

00 → 11 → 01
00 → 01
00 → 10 → 11 → 01

Fig: Examples of Non-critical Races

- Races can be avoided by directing the circuit through intermediate unstable states with a unique state-variable change.
- When a circuit goes through a unique sequence of unstable states, it is said to have a cycle.
- Care must be taken when using a cycle that terminates with a stable state.
- If a cycle does not terminate with a stable state, the circuit will keep going from one unstable state to another, making the entire circuit unstable.



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

(a) State transition:

00 → 01 → 11 → 10

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

(b) State transition:

00 → 01 → 11

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

(c) Unstable:

→ 01 → 11 → 10

Fig: Examples of Cycles

Hazards

- 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 this condition occurs in asynchronous sequential circuits, it may result in a 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 of Hazards:**

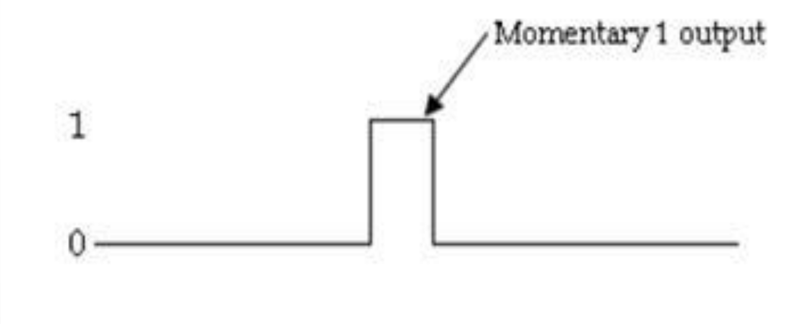
  - Static Hazard
  - Dynamic hazard

- **Static Hazard**
- In digital systems, there are only two possible outputs, a ‘0’ or a ‘1’. The hazard may produce a wrong ‘0’ or a wrong ‘1’. Based on these observations, there are three types,

➢ 1. Static- 0 hazard 2. Static- 1 hazard 3. Dynamic Hazard

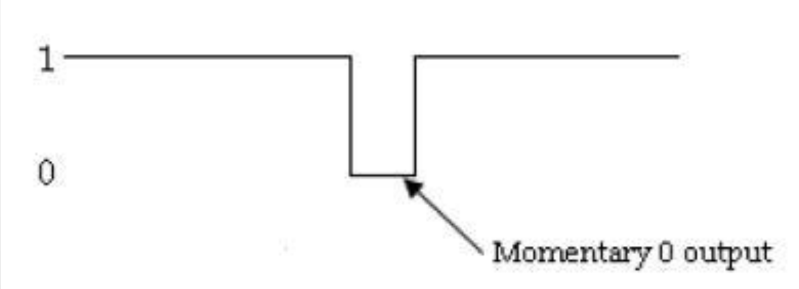
➢ Static- 0 hazard

- 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 a static 0-hazard.



➢ Static- 1 hazard

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



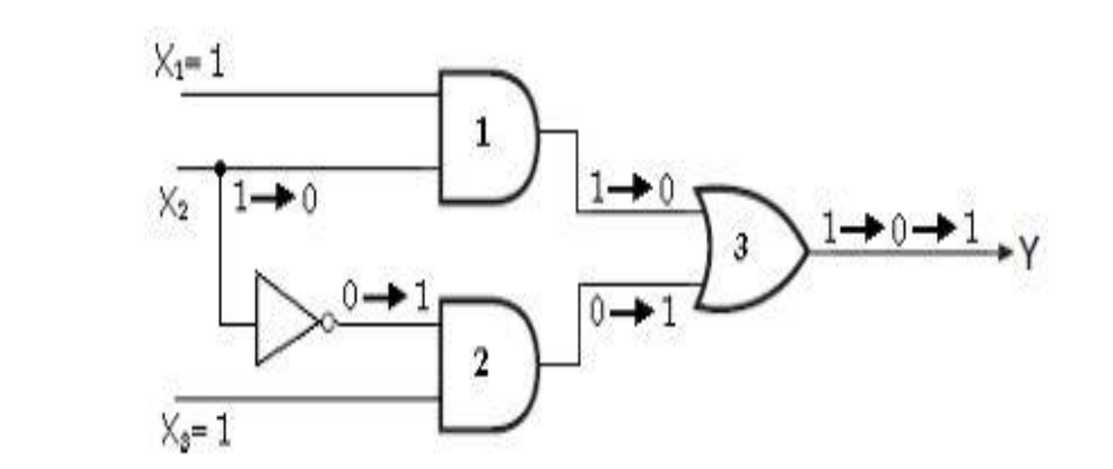


Fig: Circuit with static-1 hazard

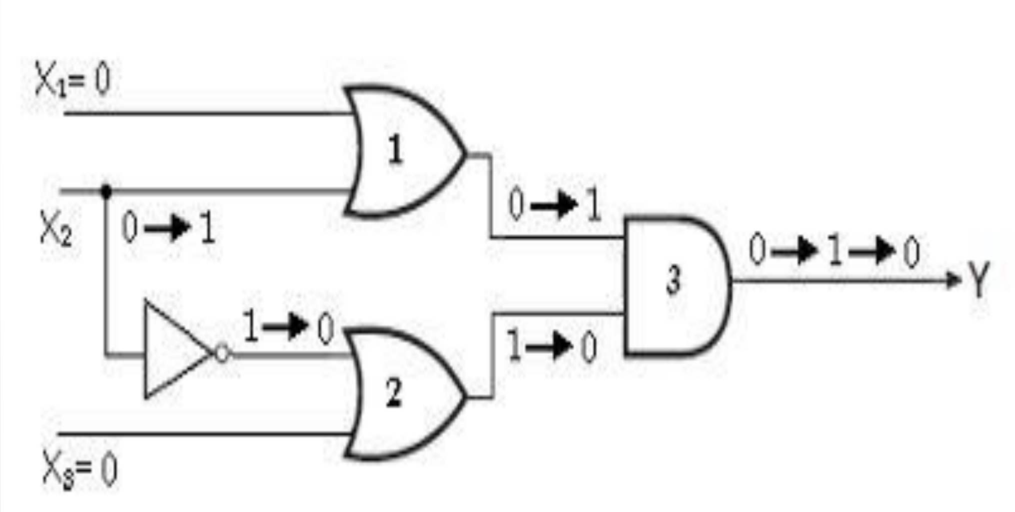
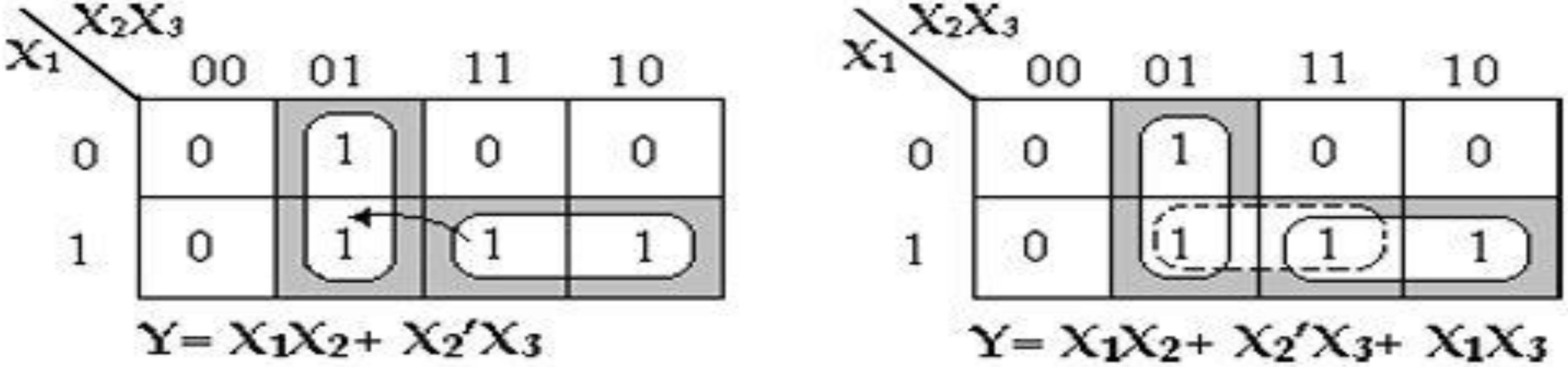


Fig: Circuit with static-0 hazard



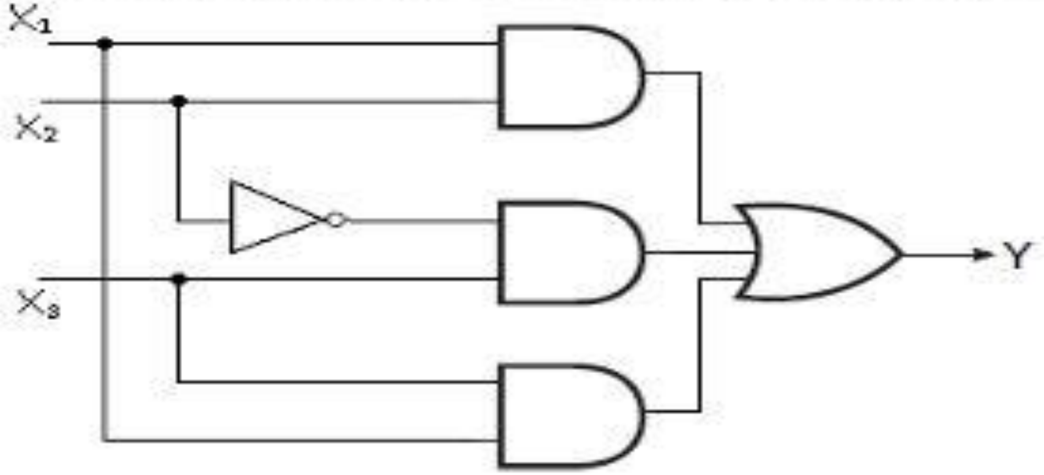
X_1 \ X_2X_3	00	01	11	10
0	0	1	0	0
1	0	1	1	1

$Y = X_1X_2 + X_2'X_3$

X_1 \ X_2X_3	00	01	11	10
0	0	1	0	0
1	0	1	1	1

$Y = X_1X_2 + X_2'X_3 + X_1X_3$

Maps demonstrating a Hazard and its Removal



Hazard-free Circuit

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 comparison to the delay associated with the feedback path may cause such a hazard.
- 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.

Thank You