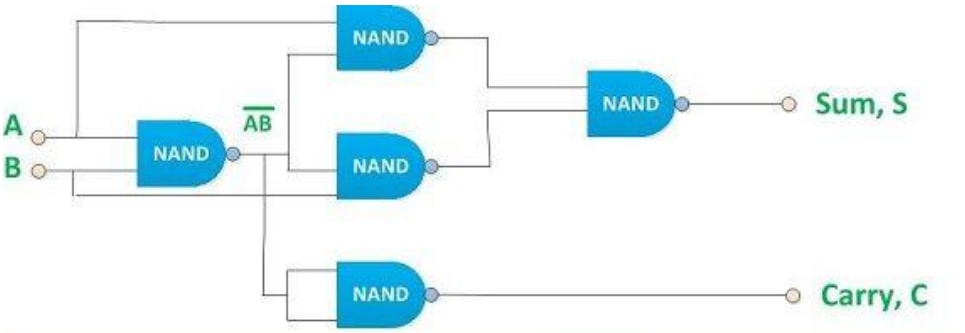
Half Adder using NAND Gates

The half adder can also be designed with the help of NAND gates. NAND gate is considered as a universal gate. A universal gate can be used for designing of any digital circuitry. It is always simple and efficient to use the minimum number of gates in the designing process of our circuit. The minimum number of NAND gates required to design half adder is 5.



Half Adder Using NAND Gates

Electronics Coach

The first NAND gate takes the inputs which are the two 1-bit numbers. The resultant NAND operated inputs will be again given as input to 3- NAND gates along with the original input. Out of these 3 NAND gates, 2-NAND gates will generate the output which will be given as input to the NAND gate connected at the end. The gate connected at the end will generate the sum bit. Out of the 3 considered NAND gates, the third NAND gate will generate the carry bit.

The NAND operation can be understood more clearly with the help of equation given below. These equations are written in the form of operation performed by NAND gates.

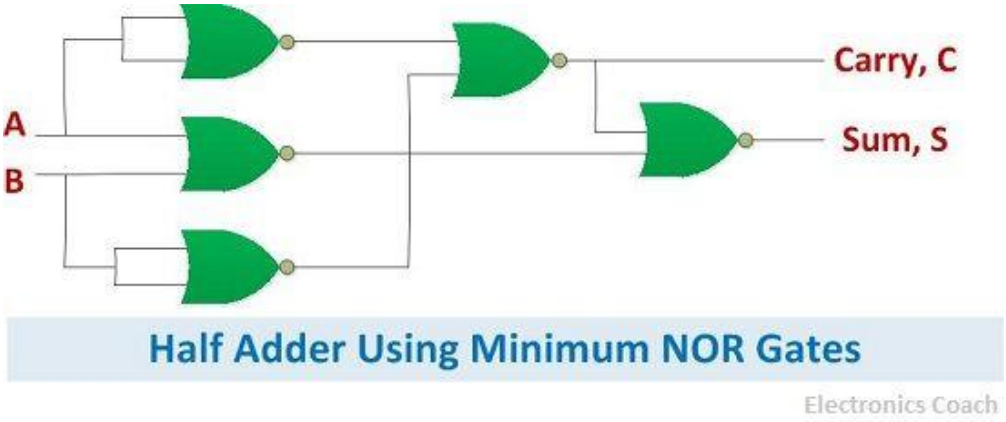
$$\text{SUM, } S = \overline{A.\overline{AB}} \; . \; \overline{B.\overline{AB}}$$

$$\text{SUM, } S = A\overline{B} + \overline{A}B$$

$$\text{Carry, } C = AB = \overline{\overline{AB}}$$

Half Adder using NOR Gates

The NOR gate is also a universal gate. Thus, it can also be used for designing of any digital circuit. The Half adder can be designed using 5 NOR gates. This is the minimum number of NOR gates to design half adder.



Firstly, three NOR gates are used in the designing and the output from two of these NOR gates is given to fourth NOR gate. The output from second NOR gate is given to the gate connected at the end. This will generate the sum bit of the addition of two 1-bit numbers.

The operation of the above circuit diagram can be understood more clearly with the help of equation. The sum bit and carry bit can be written in terms of NOR operations performed by the logic gates.

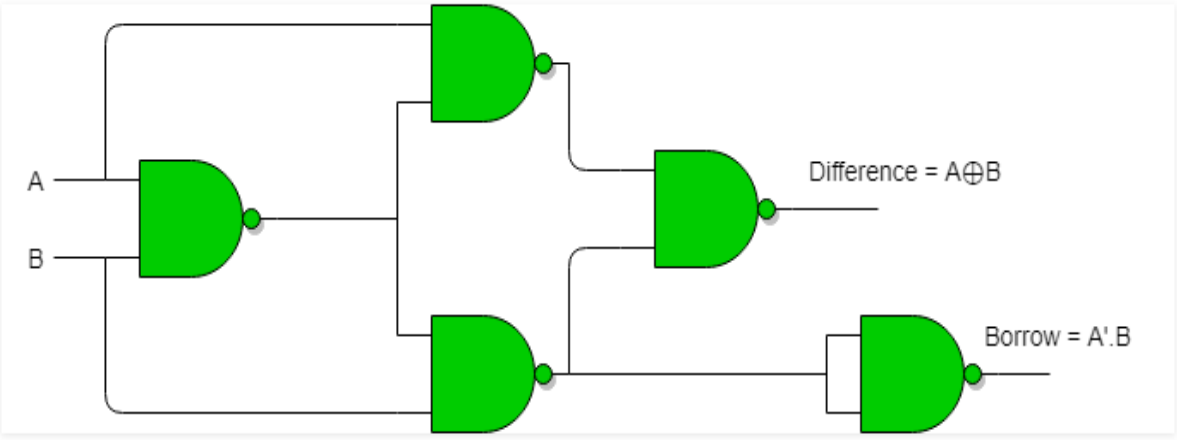
$$\text{SUM, } S = \overline{\overline{A+B} + \overline{A+\overline{B}}}$$

$$\text{SUM, } S = A\overline{B} + \overline{A}B$$

$$\text{Carry, } C = \overline{\overline{A}+\overline{B}}$$

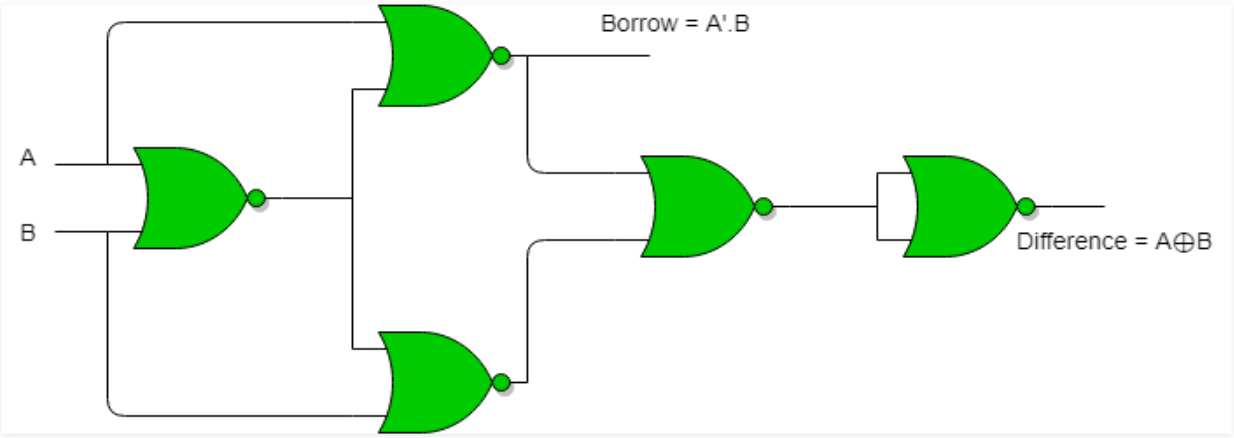
Implementation of Half Subtractor using NAND gates :

Total 5 NAND gates are required to implement half subtractor.



Implementation of Half Subtractor using NOR gates :

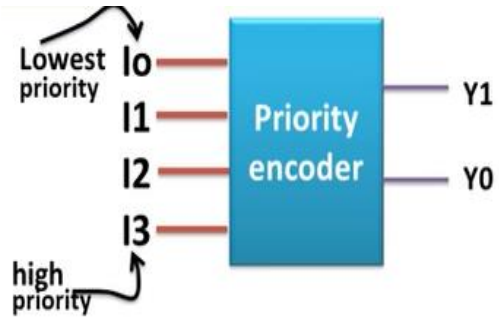
Total 5 NOR gates are required to implement half subtractor.



Priority Encoder

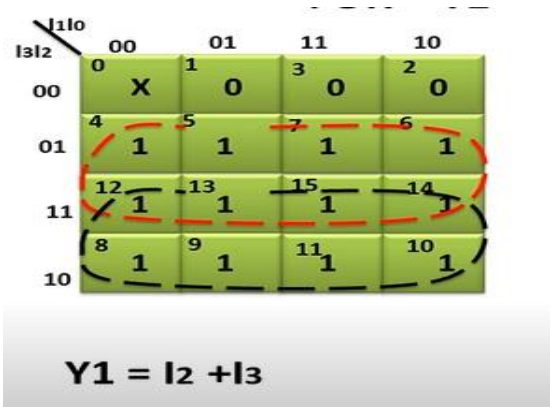
Priority Encoders take all of their data inputs one at a time and converts them into an equivalent binary code at its output. The job of a priority encoder is to produce a binary output address for the input with the highest priority. Generally, digital encoders produce outputs of 2-bit, 3-bit or 4-bit codes depending upon the number of data input lines. An "n-bit" binary encoder has 2^n input lines and n-bit output lines with common types that include 4-to-2, 8-to-3 and 16-to-4 line configurations. The *priority encoders* output corresponds to the currently active input which has the highest priority. So when an input with a higher priority is present, all other inputs with a lower priority will be ignored.

4 to 2 Priority Encoder:

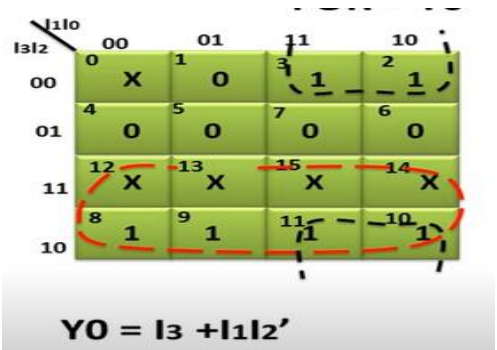


I3	I2	I1	I0	Y1	Y0
0	0	0	0	X	X
0	0	0	1	0	0
0	0	1	X	0	1
0	1	X	X	1	0
1	X	X	X	1	1

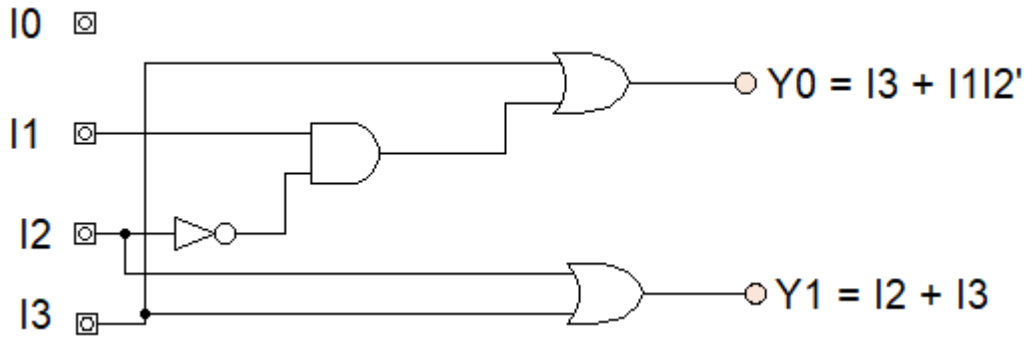
Equation for Y1



Equation for Y0:



Logic diagram for 4 to 2 priority Encoder:



1. Mention any two differences between the edge triggering and level triggering.

Level Triggering:

1) The input signal is sampled when the clock signal is either HIGH or LOW.
2) It is sensitive to Glitches.

Example: Latch.

Edge Triggering:

1) The input signal is sampled at the RISING EDGE or FALLING EDGE of the clock signal.
2) It is not-sensitive to Glitches.

Example: Flip flop.

2. What is meant by programmable counter? Mention its application.

- A counter that divides an input frequency by a number which can be programmed into decades of synchronous down counters.
- Decades, with additional decoding and control logic, give the equivalent of a divide-by *N* counter system, where *N* can be made equal to any number.

Application:

- Microprocessor.
- Traffic light controller.
- Street light controller.

3. **Write the characteristic equation of a JK flip-flop.**

The characteristic equation of a JK flip-flop is given by

Q(next) = JQ' + K'Q

3. Compare the logics of synchronous counter and ripple counter.

Asynchronous counter:

1. In this type of counter flip flop are connected in such a way that output of first flip-flop drives the clock for next flip-flop.
2. All the flip-flop are not clocked simultaneously.
3. Logic circuit is very simple even for more number of states.

Synchronous counter:

1. In this type there is no connection between output of first flip-flop and clock input of the next flip-flop.
2. All the flip-flop are clocked simultaneously.
3. Design involves complex logic circuit as number of states increases.

The difference between Synchronous and Asynchronous Counters are as follows:

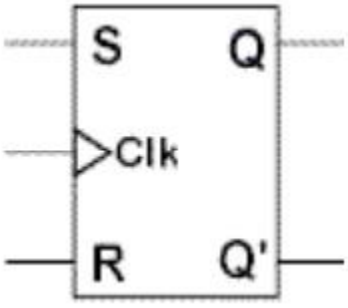
S.No	Asynchronous Counters	Synchronous Counters
1.	These are low-speed Counters.	These are high-speed Counters.
2.	The Flip flops of these counters are not clocked simultaneously.	In these counters, the flip-flops are cl simultaneously.
3.	Simple logic circuits are there for more number of states.	Complex logic circuits are there when states increases.

What are the applications of Flip-Flops?

The applications of flip-flops are:

1. Flip-flops are used as the delay element.
2. These are used for Data transfer.
3. Flip-flops are used in Frequency Division and Counting.
4. Flip-Flops are used as the memory element.

4. Sketch the logic diagram of a clocked SR flip-flop.



5. How do you eliminate the race around condition in a JK flip-flop?

- When the input to the JK flip-flop is j=1 and k=1, the race around condition occurs, i.e it occurs when the time period of the clock pulse is greater than the propagation delay of the flip flop.
- the output changes or toggles in a single clock period. If it toggles even number of times the output is same but if it toggles odd number of times then the output is complimented.

To avoid race around condition we can't make the clock pulse smaller than the propagation delay so we use

1. Master slave JK flip flop
2. Positive or negative edge triggering

6. Define latches.

Latch is a simple memory element, which consists of a pair of logic gates with their inputs and outputs inter connected in a feedback arrangement, which permits a single bit to be stored.

7. Write short notes on Digital Clock.

A digital clock is a simplified logic diagram of a digital clock that displays seconds, minutes, and hours. First, a 60 Hz sinusoidal ac voltage is converted to a 60 Hz pulse waveform and divided own to a 1Hz pulse waveform by a divide-by-60 counter formed by a divide-by-10 counter allowed by a divide-by-6 counter. Both the seconds and minutes counts are also produced by divide-by-60 counters.

8. What are shift register counters? List two widely used shift register counters.
Solution:

If the output of a shift register is fed back to the input. a ring counter results. The data pattern contained within the shift register will recirculate as long as clock pulses are applied.

9. Why is Flip Flop also known as Latch?
Solution:

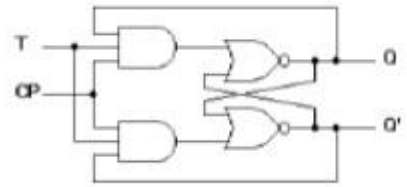
The difference between a latch and a flip-flop is that a latch does not have a clock signal, whereas a flip-flop always does. Latch is a level sensitive device while flip-flop is an edge sensitive device. Latch is sensitive to glitches on enable pin, whereas flip-flop is immune to that.

10. What are state diagram and state tables?
Solution:

The time sequence of inputs, outputs and flip-flop states may be enumerated in a state table and the information available in a state table may be represented graphically called a state diagram. For the design of sequential counters we have to relate present states and next states. The table, which represents the relationship between present states and next states, is called state table.

11. Draw the logic diagram for T Flip Flop

Solution:



12. How many flip-flops are required to design a mod-7 up-down counter?
Solution:

Flip Flops required are 2n >= N where N=7

2n >= 7 n = Number of Flip Flops.

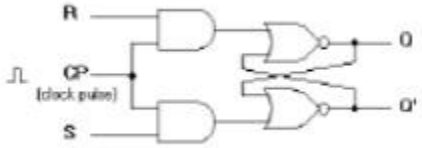
n = 3 Hence 3 Flip Flops are required

13. What is a sequential circuit?

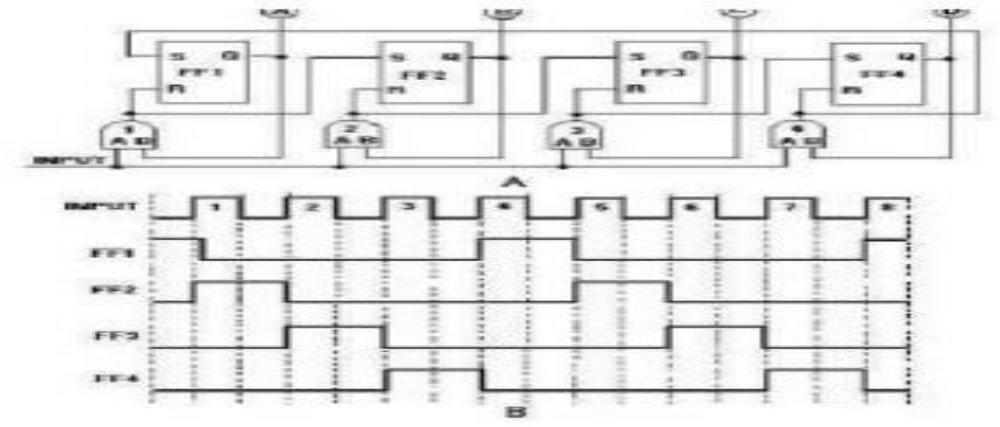
Solution:

The logic circuits whose outputs at any instant of time depend not only on the present inputs but also on the past outputs are called sequential circuits. Example : flip-flops.

14. Draw the logic diagram of SR flip-flop.
Solution:



15. Draw the timing diagram of 4-bit ring counter.



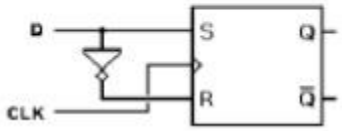
16. Write the characteristic equation of JK flip-flop.

Solution:

$Q(t + 1) = JQ' + K'Q$

Excitation table for above conversion

Input	Present state	Next state	Flip-flop inputs	
D	Qn	Qn+1	S	R
0	0	0	0	X
0	1	0	0	1
1	0	1	1	0
1	1	1	X	0



17. State a limitation of SR flip-flop.

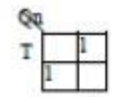
Solution:

The last input condition in SR flip-flop is S=1 and R=1. This Condition will produce 0 at the output of both the NOR gate. Hence Qn+1=0 and Q'n+1= 0. This condition violates the fact that the outputs Qn+1 and Q'n+1 are the complements of each other. In normal operation, this condition must be avoided by making sure that 1s are not a applied to both inputs simultaneously.

18. Convert a D flip-flop into T flip-flop.

Excitation table for above conversion

Input	Present state	Next state	Flip-flop inputs
T	Qn	Qn+1	D
0	0	0	0



$D = TQ' + T'Q = T \oplus Q$

19. What is the race- around condition?

Solution:

In JK FF output is fedback to the input and therefore change in the output results in change in the input.If the FF is level trigged, in the positive half of the clock pulse if j and k are both high then output toggles continuously. This condition is known as race around condition.

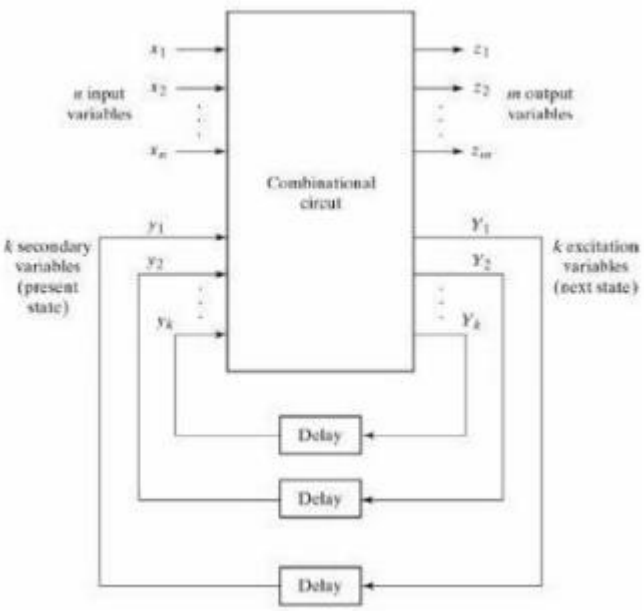
Analysis Procedure of Asynchronous Sequential Circuits

ASYNCHRONOUS SEQUENTIAL CIRCUITS

Asynchronous sequential circuits:

- ¬ Do not use pulses. The change of internal state occurs when there is a change in the input variable.
- ¬ Their memory elements are either unclocked flip flops or time delay elements.
- ¬ They often resemble combinational circuit with feedback.
- ¬ Their synthesis is much difficult than the synthesis of clocked synchronous sequential circuit.
- ¬ They are used when speed of operation is important.

The communication of two units, with each unit having its own independent clock, must be done with asynchronous circuits.



There are n input variables, m output variables, and k internal states.

The present state variables (y_1 and y_2) are called secondary variables. The next state variables (y_1 and y_2) are called excitation variables.

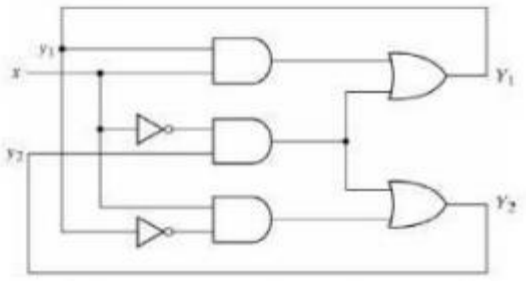
Fundamental mode operation assumes that the input signals change one at a time and only when the circuit is in a condition.

1. ANALYSIS PROCEDURE

The analysis of asynchronous sequential circuits proceeds in much the same way as that of clocked synchronous sequential circuits. From a logic diagram. Boolean expressions are written and then transferred into tabular form.

1. TRANSITION TABLE

An example of an asynchronous sequential circuit is shown below:



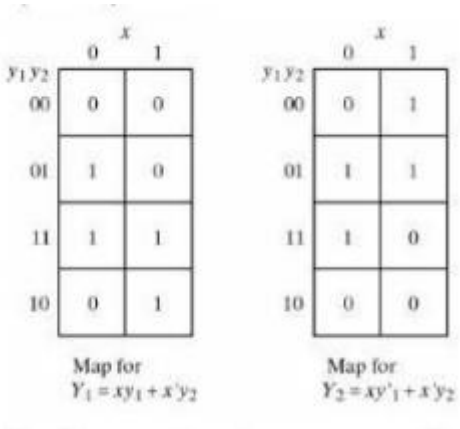
The analysis of the circuit starts by considering the excitation variables (y_1 and y_2).

As outputs and the secondary variables (y_1 and y_2) as inputs.

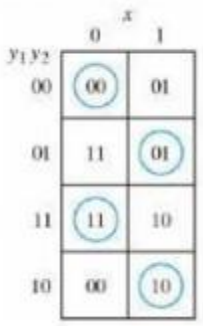
The Boolean expressions are:

$$Y_1 = xy_1 + x'y_2$$
$$Y_2 = xy_1' + x'y_2$$

The next step is to plot the Y_1 and Y_2 functions in a map:



Combining the binary values in corresponding squares the following transition table is obtained.



The transition table shows the values of $Y=Y_1Y_2$ inside each square. Those where Y=y are circled to indicate a stable condition.

The circuits has four stable total states –y_1y_2x= 000, 011, 110, and 100 and four unstable total states 001, 010, 111, and 100.

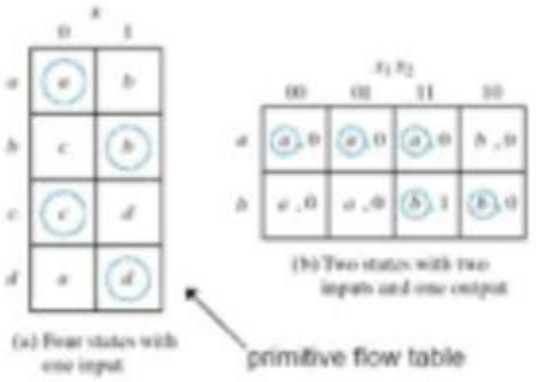
The state table of the circuit shown below:

Present State		Next State			
		$x = 0$		$x = 1$	
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 table provides the same information as the transition table.

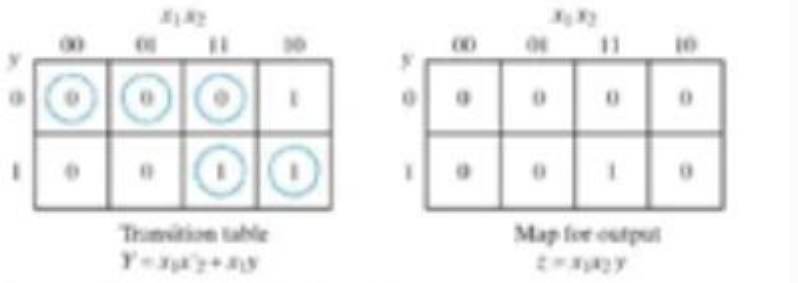
2. FLOW TABLE

In a flow states are named by letter symbols. Examples of flow tables are as follows.

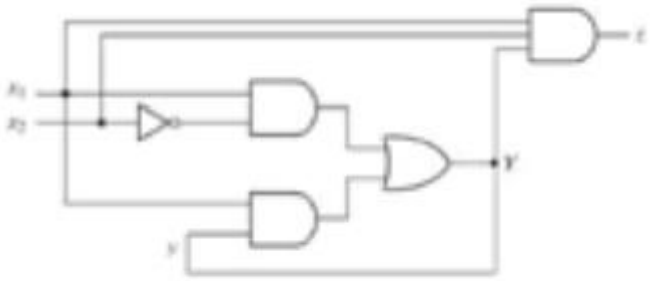


In order to obtain the circuit described by a flow table, it is necessary to assign to each state distinct value.

This assignment converts the flow table into a transition table. This is shown below:



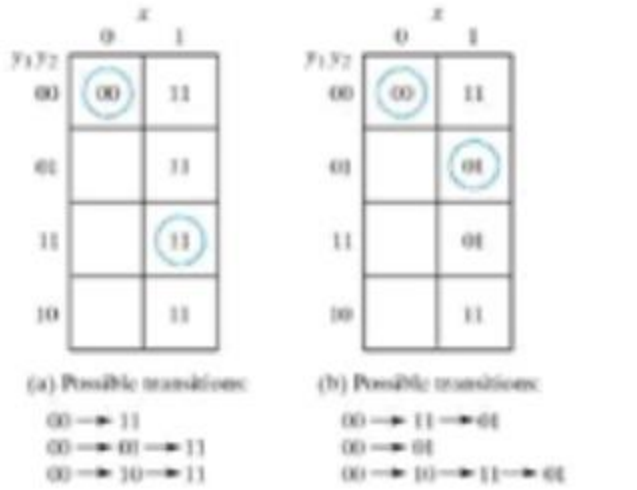
The resulting logic diagram is shown below:



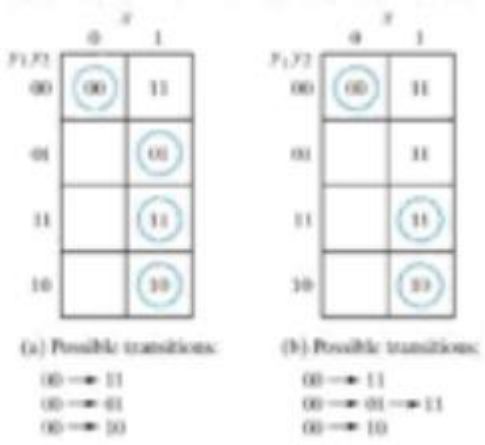
3. RACE CONDITIONS

A race condition exists in an asynchronous 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 variable to change in an unpredictable manner.

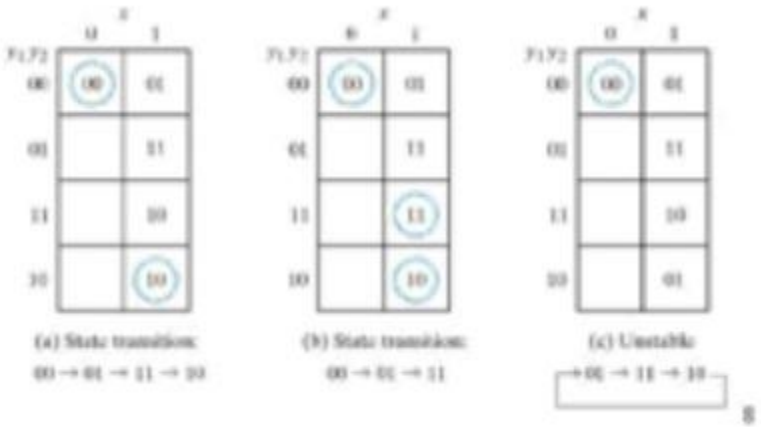
If the final stable state that the circuit reaches does not depend on the order in which the state variable change, the races are illustrated in the transition tables below:



The transition tables below illustrate critical races:



Races can be avoided by directing the circuit through a unique sequence of intermediate unstable states when a circuit does that, it is said to have a cycle. Examples of cycles are:



STATE ASSIGNMENTS

If one is able to reduce the total number of states, one may be able to save on the number of flip flops required for a design. This is the optimal situation. For example if a finite state machine drops from 7 states to 4 states and compact state assignment are used, the design drops from three flip flops to two flip flop. A sub optimal situation is when the number of states is reduced, but the number of flip flops is not. This does add don't cares to the combinational logic that generates the next state equations. This will most likely drop the over all cost of the finite state machine.

Once the number states is at a minimum, then a judicious assignment of states may further reduce the cost of the next state equation and or the cost of the output equations. A set of heuristic rules is proposed where each rule is directed toward the reduction of the combinational logic in the finite state machine design.

As opposed to compact state assignments, one may propose a one hot state assignment. One hot is a set of state assignment in which a unique bit is one in the assignment for each state. This often lends to a reduction in the logic cost for the outputs, because in one and only one state a given output is asserted.

All of these topics will be discussed in more detail below.