Sri SAI RAM ENGINEERING COLLEGE

INSTITUTE OF TECHNOLOGY

West Tambaram, Chennai - 44

YEAR	SEM
II	III

CS 8351

DIGITAL PRINCIPLES AND SYSTEM DESIGN
(Common to CSE & IT)

UNIT NO. 4

4.5 RACE FREE STATE ASSIGNMENT

Version: 1.0

RACE FREE STATE ASSIGNMENT

RACES:

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.

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 (a) 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.
In (a), the final state is (y1y2x= 111), and in (b), it is (y1y2x= 011).

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

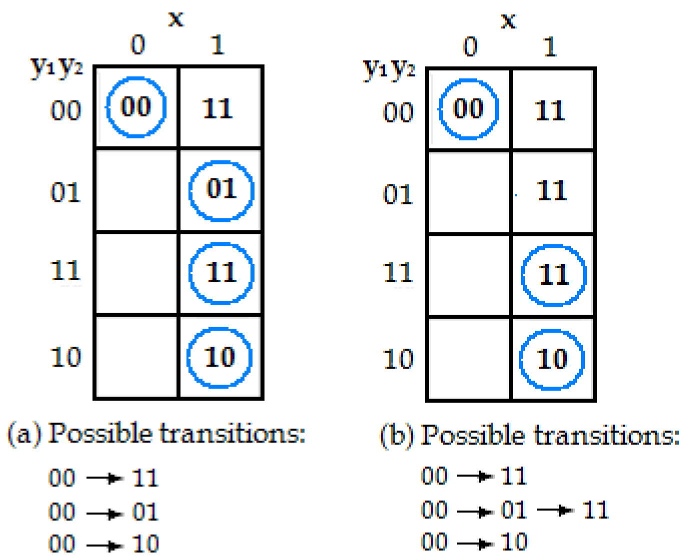
Examples for Non Critical Races

Critical races:

A race becomes critical if the correct next state is not reached during a state transition. 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 it is a critical race. For proper operation, critical races must be avoided.

The below transition table illustrates critical race condition. The transition table (a) starts in stable state (y1y2x= 000), and then change the input from 0 to 1. The state variables must then change from 00 to 11. If they change simultaneously, the final total stable state is 111. In the transition table (a), if, because of unequal propagation delay, Y2 changes to 1 before Y1 does, then the circuit goes to the total stable state 011 and remains there. If, however, *Y1* changes first, the internal state becomes 10 and the circuit will remain in the stable total state 101.

Hence, the race is critical because the circuit goes to different stable states, depending on the order in which the state variables change.



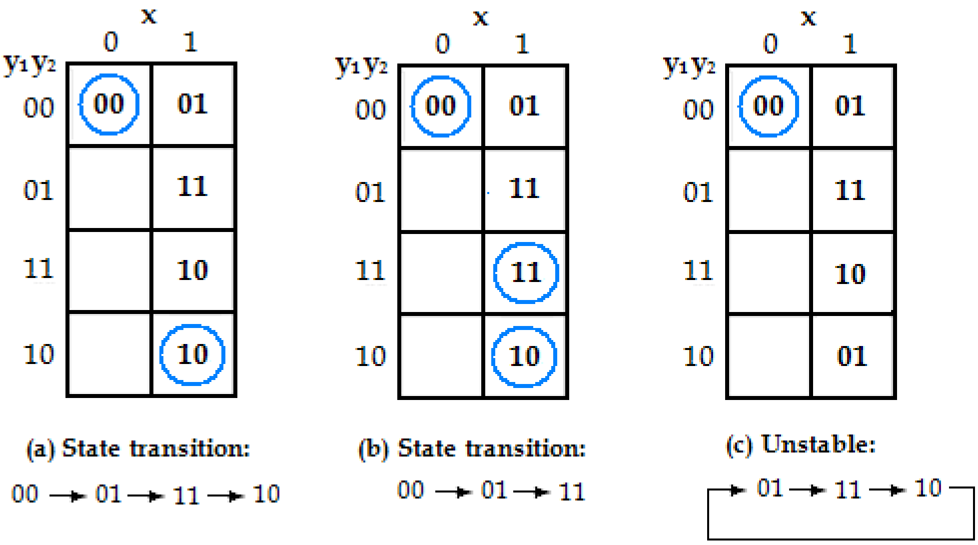
Examples of Critical Races

CYCLES

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.

Again, we start with y1y2 = 00 and change the input from 0 to 1. The transition table (a) gives a unique sequence that terminates in a total stable state 101. The table in (b) shows that 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 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. This is demonstrated in the transition table (c).

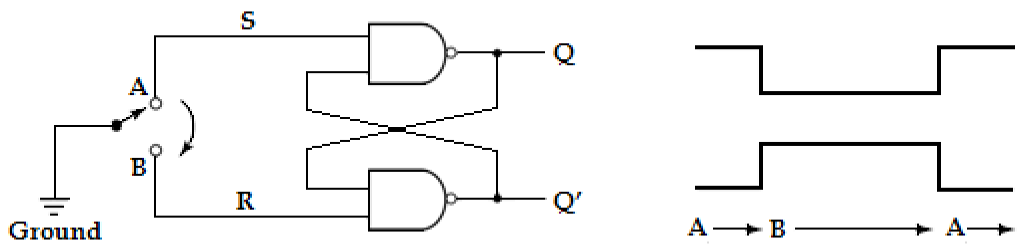
Examples of Cycles



Debounce Circuit:

Input binary information in binary information can be generated manually be means of mechanical switches. One position of the switch provides a voltage equivalent to logic 1, and the other position provides a second voltage equivalent to logic 0. Mechanical switches are also used to start, stop, or reset the digital system. A common characteristic of a mechanical switch is that when the arm is thrown from one position to the other the switch contact vibrates or bounces several times before coming to a final rest. In a typical switch, the contact bounce may take several milliseconds to die out, causing the signal to oscillate between 1 and 0 because the switch contact is vibrating.

A debounce circuit is a circuit which removes the series of pulses that result from a contact bounce and produces a single smooth transition of the binary signal from 0 to 1 or from 1 to 0. One such circuit consists of a single-pole, double-throw switch connected to an SR latch, as shown below. The center contact is connected to ground that provides a signal equivalent to logic 0. When one of the two contacts, A or *B,* is not connected to ground through the switch, it behaves like a logic-1 signal. When the switch is thrown from position *A* to position *B* and back, the outputs of the latch produce a single pulse as shown, negative for *Q* and positive for *Q'.* The switch is usually a push button whose contact rests in position A. When the pushbutton is depressed, it goes to position B and when released, it returns to position A.



Debounce Circuit

The operation of the debounce circuit is as follows: When the switch resets in position A, we have the condition S = 0, R = 1 and Q = 1, Q' = 0. When the switch is moved to position B, the ground connection causes R to go to 0, while S becomes a 1 because contact A is open. This condition in turn causes output Q to go to 0 and Q' to go to 1. After the switch makes an initial contact with B, it bounces several times. The output of the latch will be unaffected by the contact bounce because Q' remains 1 (and Q remains 0) whether R is equal to 0 (contact with ground) or equal to 1 (no contact with ground). When the switch returns to position A, S becomes 0 and Q returns to 1. The output again will exhibit a smooth transition, even if there is a contact bounce in position A.