3.3 EXCESS 3 AND ALPHANUMERIC CODES

In ***combinational logic circuits***, the outputs atany instant of time depend only on the input signals present at that time. For a change in input, the output occurs immediately.

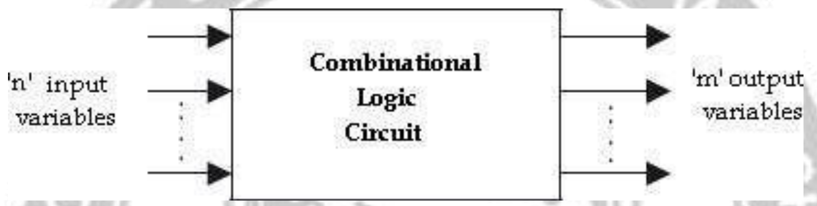


Fig: 3.1-CombinationalCircuit-BlockDiagram

In ***sequential logic circuits***, it consists of combinational circuits to which storage elements are connected to form a feedback path. The storage elements are devices capable of storing binary information either 1 or 0.

The information stored in the memory elements at any given time defines the present state of the sequential circuit. The present state and the external circuit determine the output and the next state of sequential circuits.

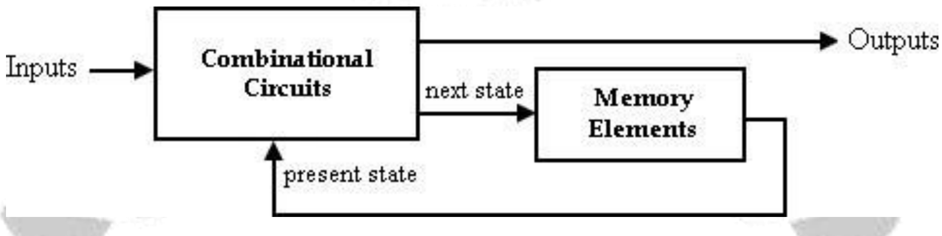


Fig:3.2-SequentialCircuit-BlockDiagram

Thus in sequential circuits, the output variables depend not only on the present input variables but also on the past history of input variables.

The rotary channel selected knob on an old-fashioned TV is like acombinational.Itsoutputselectsachannelbasedonlyonitscurrentinput–the position of the knob. The channel-up and channel-down push buttons on a TV is like a sequential circuit. The channel selection depends on the past sequence of up/down pushes.

Table:3.1-Thecomparisonbetweencombinational andsequentialcircuits

S.No	Combinationallogic	Sequentiallogic
1	The output variable,atalltimes depends on the combination of input variables.	The output variable depends not only on the present inputbutalso depend upon the past history of inputs.
2	Memoryunit isnot required	Memoryunit is requiredtostorethe pasthistoryofinputvariables.
3	Faster inspeed	Slowerthancombinationalcircuits.
4	Easytodesign	Comparativelyhardertodesign.
5	Eg.Parallel adder	Eg.Serialadder

<u>ClassificationofLogicCircuits</u>

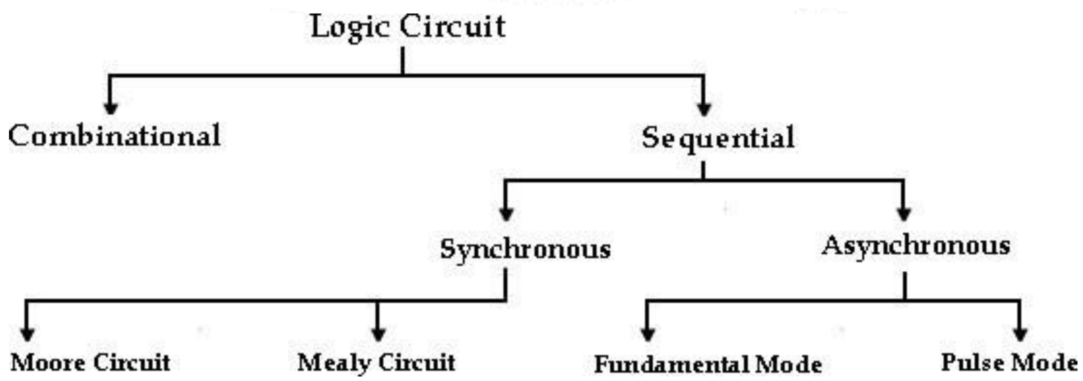


Fig: 3.3–Classificationofsequential Circuits

The sequential circuits can be classified depending on the timing of their signals:

- Synchronoussequentialcircuits
- Asynchronoussequentialcircuits.

In synchronous sequential circuits, signals can affect the memory elements only at discrete instants of time. In asynchronous sequential circuits change in input signals can affectmemory element at any instant oftime.Thememory elements used in both circuits are Flip-Flops, which are capable of storing 1- bit information.

Table: 3.2 - The comparison between Synchronous and asynchronous sequential circuits

S.No	Synchronoussequential circuits	Asynchronoussequentialcircuits
1	Memoryelementsare clocked Flip-Flops	Memory elements are either unclockedFlip-Flops or timedelay elements.
2	Thechangeininputsignals can affect memory element uponactivationofclocksignal.	Thechangeininputsignalscanaffect memoryelement atany instant of time.
3	The maximum operating speedofclockdependson time delays involved.	Because of theabsence of clock,it can operate faster than synchronouscircuits.
4	Easiertodesign	Moredifficulttodesign

ALPHA NUMERIC CODES

Latches and Flip-Flops are the basic building blocks of the most sequential circuits. Latches are used for a sequential device that checks all of its inputs continuously and changes its outputs accordingly at any time independent of clocking signal. Enable signal is provided with the latch. When enable signal isactive output changes occur as the input changes. But when enable signal is not activated input changes do not affect the output.

Flip-Flop is used for a sequential device that normally samples its inputs and changes its outputs only at times determined by clocking signal.

SRLatch:

The simplest type of latch is the set-reset(SR) latch. It can be constructed from either two NOR gates or two NAND gates.

SRlatchusingNORgates:

The two NOR gatesare cross-coupled so that the output of NOR gate 1 is connected to one of the inputs of NOR gate 2 and vice versa. The latch has two outputs Q and Q' and two inputs, set and reset.

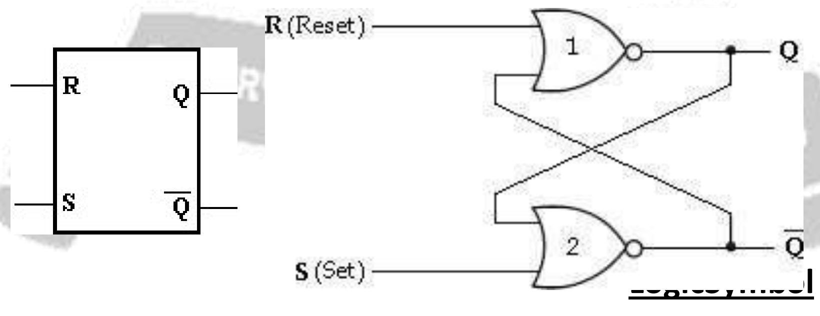


Fig:3.4-SRlatchusingNORgates

Before going to analyse the SR latch, we recall that a logic 1 at any input of a NOR gate forces its output to a logic 0. Let us understand the operation of this circuit for various input/ output possibilities.

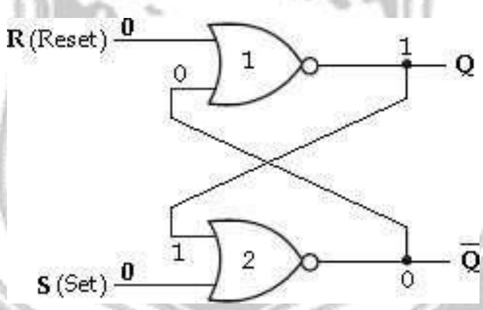
EXCESS 3 CODES

Case 1:S=0andR=0

Initially,Q=1andQ'=0

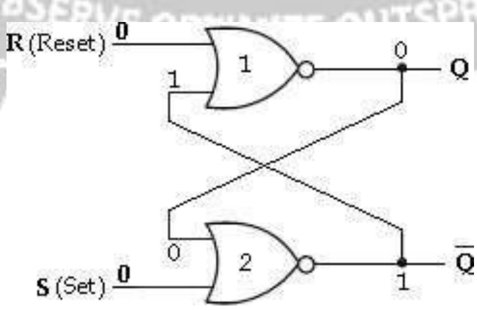
Let us assume that initially Q=1 and Q'=0. With Q'=0, both inputs to NOR gate 1 are at logic 0. So, its output, Q is at logic 1. With Q=1, one input of NOR gate 2 is at logic

1.Hence its output, Q' is at logic0. This shows that when Sand R both are low, the output does not change.



Initially,Q=0andQ'=1

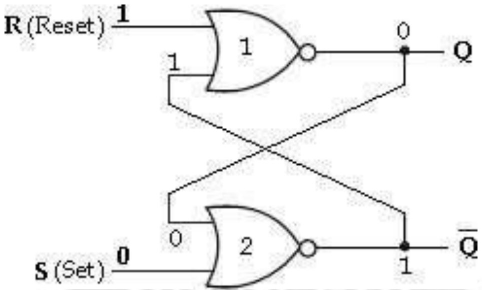
With Q'=1, one input of NOR gate 1 is at logic 1, hence its output, Q is at logic 0.With Q=0, both inputs to NOR gate 2 are at logic 0. So, its output Q' is at logic 1. In this case also there is no change in the output state.



Case 2:S=0andR=1

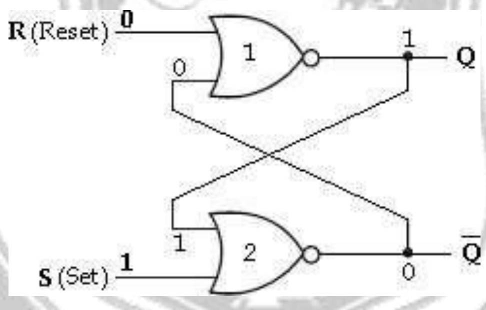
Inthiscase,RinputoftheNORgate1isatlogic 1,henceits output,Qis atlogic0.

BothinputstoNORgate2are nowatlogic 0. Sothatitsoutput,Q’is atlogic1.



Case 3:S=1andR=0

In this case, S inputof the NOR gate 2 is at logic 1, hence its output, Q is at logic 0. Both inputs to NOR gate 1 are now at logic 0. So that its output, Q is at logic 1.



Case 4:S=1andR=1

When R and S both are at logic 1, they force the outputs of both NOR gates to the low state, i.e., (Q=0 and Q’=0). So, we call this an indeterminate or prohibited state, and represent this condition in the truth table as an asterisk (*). This condition also violates the basic definition of a latch that requires Q to be complement of Q’. Thus in normal operation this condition must be avoided by making sure that 1’s are not applied to both the inputs simultaneously.

xWhen S= 0 and R= 0, the output, Q_{n+1}remains in its present state, Q_n. xWhen S= 0 and R= 1, the latch is reset to 0. xWhen S= 1 and R= 0, the latch is set to 1.

xWhenS= 1 and R= 1, the output of both gates willproduce 0.

i.e., **Qn+1= Qn+1'= 0.**

ThetruthtableofNORbasedSRlatchisshownbelow.

S	R	Qn	Qn+1	State
0	0	0	0	No Change
0	0	1	1	(NC)
0	1	0	0	Reset
0	1	1	0	
1	0	0	1	Set
1	0	1	1	
1	1	0	xx	Indeterminate
1	1	1		*

The SR latch can also be implemented using NAND gates. The inputs of this Latch are S and R.To understand how this circuit functions, recall that a low on any input to a NAND gate forces its output high.

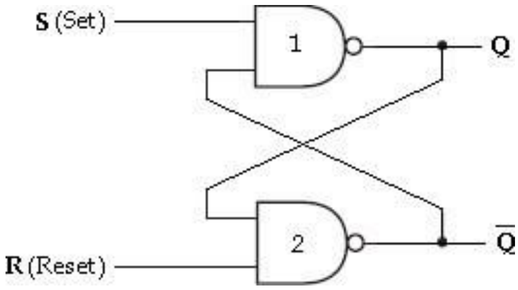


Fig: 3.5-SRlatch usingNANDgates

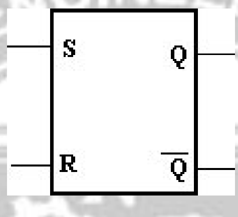


Fig:3.6- LogicSymbol We

can summarize the operation of SR latch as follows:

xWhen S= 0 and R= 0, the output of both gateswill produce 0.

i.e., **Qn+1= Qn+1'= 1.**

xWhen S= 0and R= 1, the latch is reset to 0. xWhen S= 1 and R= 0, the latch is set to 1. xWhen S= 1 and R= 1, the output, Q_{n+1}remains in its present state, Q_n.

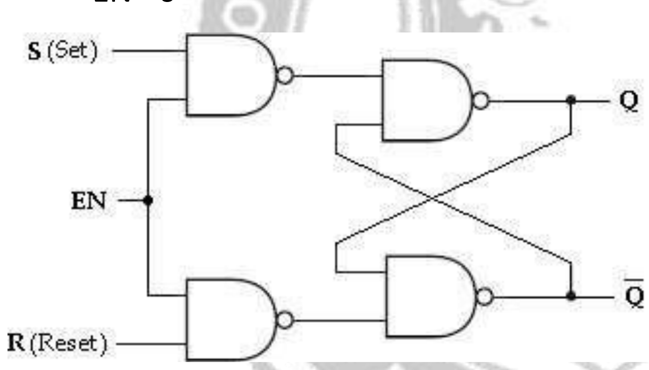
S	R	Qn	Qn+1	State
0	0	0	xx	Indeterminate
0	0	1		*
0	1	0	1	Set
0	1	1	1	
1	0	0	0	Reset
1	0	1	0	

1	1	0	0	NoChange
1	1	1	1	(NC)

In the SR latch, the output changes occur immediately after the input changes i.e, the latch is sensitive to its S and R inputs all the time.

A latch that is sensitive to the inputs only whenanenable input is active. Such a latch with enable input is known as gated SR latch.

xThe circuit behaves like SR latch when EN= 1. It retains its previous state when EN= 0



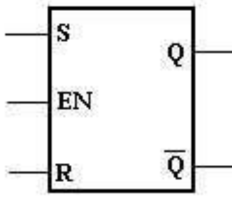


Fig : 3.6 - SR Latch with enable input using NAND gates Fig:3.7-LogicSymbol

The truth table of gated SR latch is show below.

EN	S	R	Qn	Qn+1	State
1	0	0	0	0	NoChange(NC)
1	0	0	1	1	
1	0	1	0	0	Reset
1	0	1	1	0	

1 1	1 1	0 0	0 1	1 1	Set
1 1	1 1	1 1	0 1	xx	Indeterminate *
0 0	xx	xx	0 1	0 1	NoChange(NC)

When S is HIGHand R is LOW,a HIGH on the EN input setsthe latch. When S is LOW and R is HIGH, a HIGH on the EN input resets the latch.

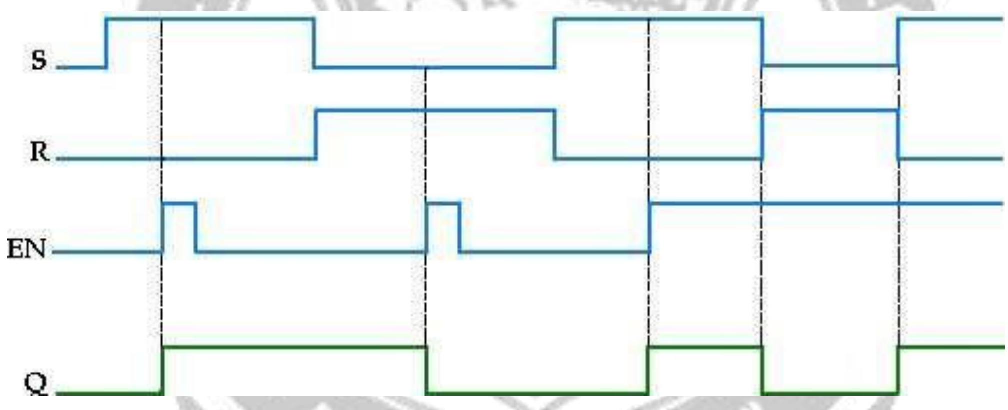
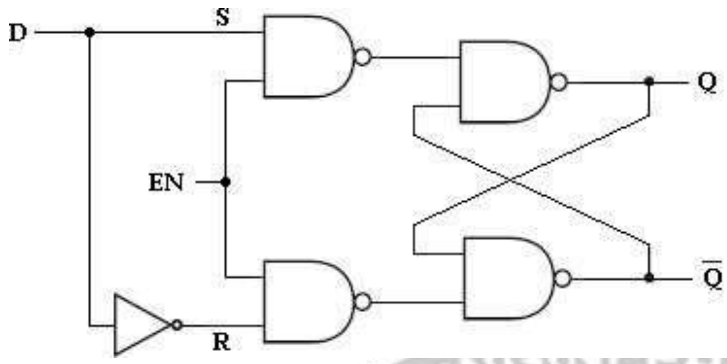


Fig:3.8–Timingdiagram

InSRlatch,whenbothinputsaresame(00or11),theoutputeitherdoesnot change or it is invalid. In many practical applications, these input conditions are not required. These input conditions can be avoided by making them complement of each other. This modified SR latch is known as **D latch.**



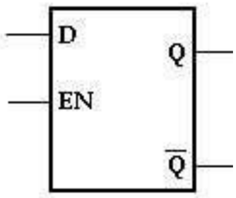


Fig:3.9-DLatch Fig:3.10 -LogicSymbol

As shown in the figure,D input goes directly to the S input, and its complement is applied to the R input. Therefore, only two input conditions exists, either S=0 and R=1 or S=1 and R=0. The truth table for D latch is shown below.

EN	D	Qn	Qn+1	State
1	0	x	0	Reset
1	1	x	1	Set
0	x	x	Qn	NoChange(NC)

As shown in the truth table, the Q outputfollows the D input. For this reason,D latch is called ***transparent latch***.

When Dis HIGH and EN is HIGH. Q goes HIGH. When Dis LOW and EN is HIGH, Q goes LOW. When EN is LOW, the state of the latch is not affected by the D input.

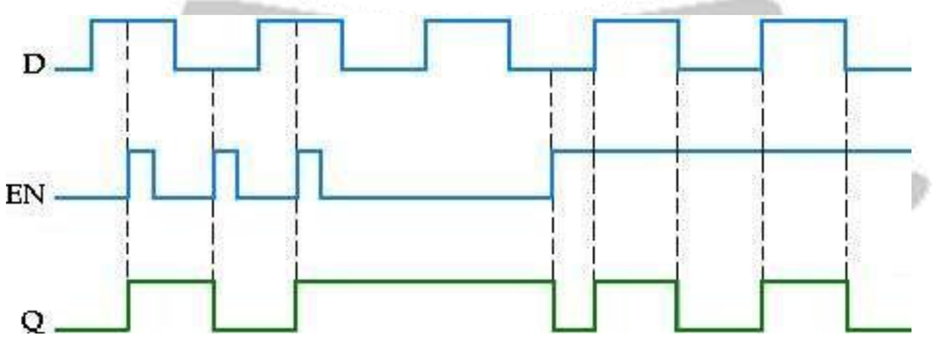


Fig:3.11–Timingdiagram