DIGITAL ELECTRONICS(BCA103)

DEPARTMENT OF COMPUTER SCIENCE

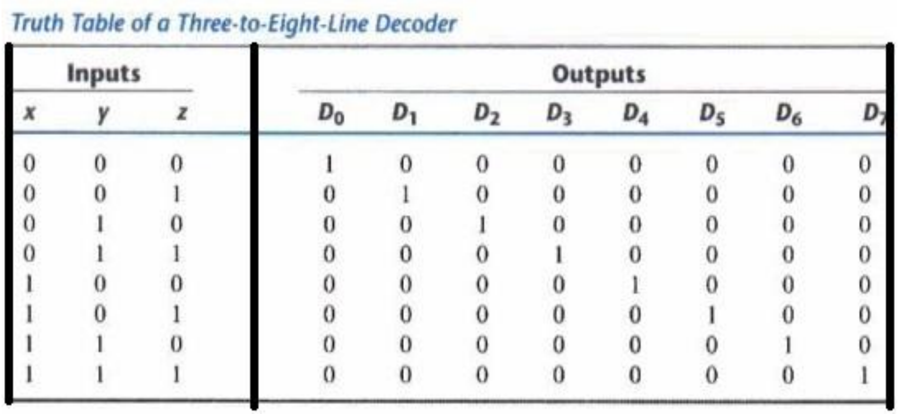
PROGRAMME: BCA

CENTRAL UNIVERSITY OF ORISSA
KORAPUT

DECODER

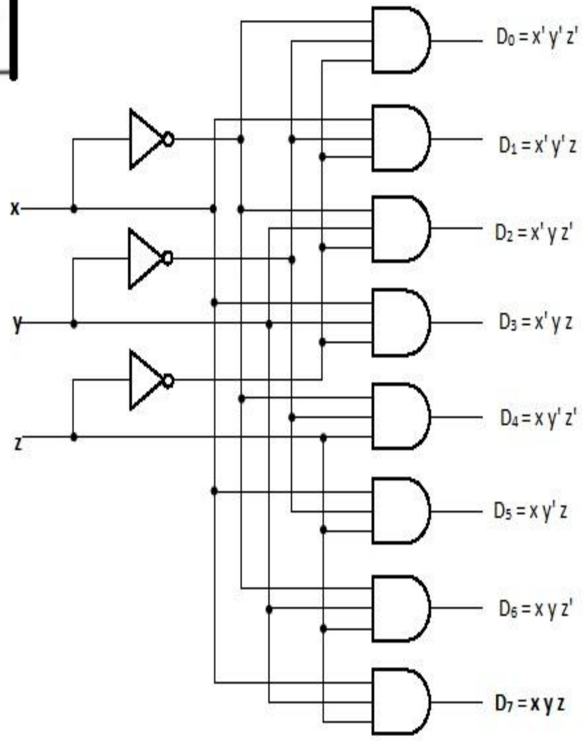
- A dec*oder* is a combinational circuit that converts binary information from ***n*** input lines to a maximum of $\mathbf{2^n}$unique output lines.
- If the **n**-bit coded information has unused combinations, the decoder may have fewer than $\boldsymbol{2^n}$ outputs.
- n-to-m line decoders, where m<= 2^n.
- Their purpose is to generate the 2^n(or fewer) mintems of n input variables.

3-to-8 Line Decoder



Truth Table of a Three-to-Eight-Line Decoder

Inputs			Outputs							
x	y	z	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

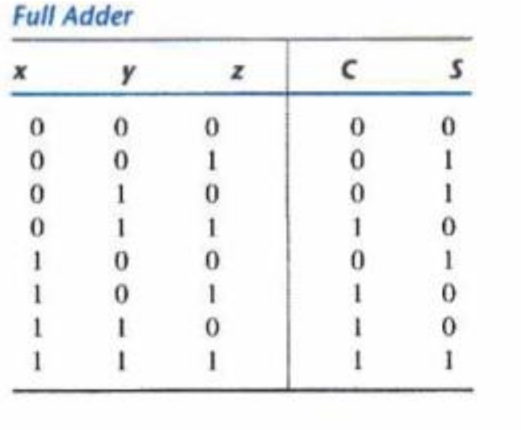


Three-to-Eight Line Decoder (3-to-8 Line Decoder)

3-to-8 Line Decoder

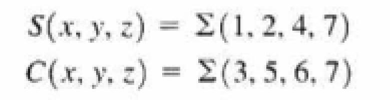
- The three inputs are decoded into eight outputs, each representing one of the minterms of the three input variables.
- The three inverters provide the complement of the inputs, and each one of the eight *AND* gates generates one of the minterms.
- Use/Application: binary-to-octal conversion.
- For each possible input combination, there are seven outputs that are equal to 0 and only one that is equal to 1.
- The output whose value is equal to 1 represents the minterm equivalent of the binary number currently available in the input lines.

Implementation of a Full Adder with a Decoder

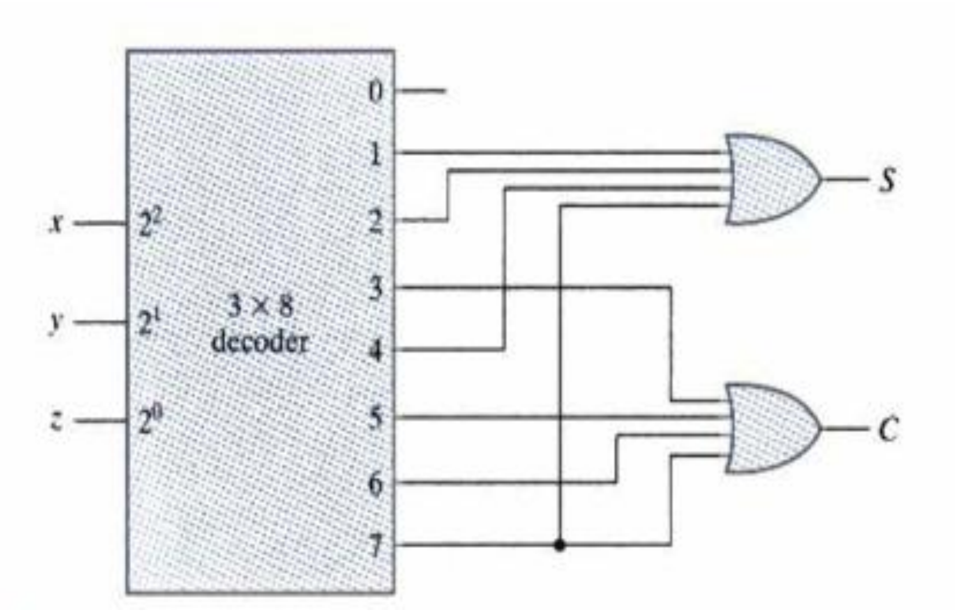


Full Adder

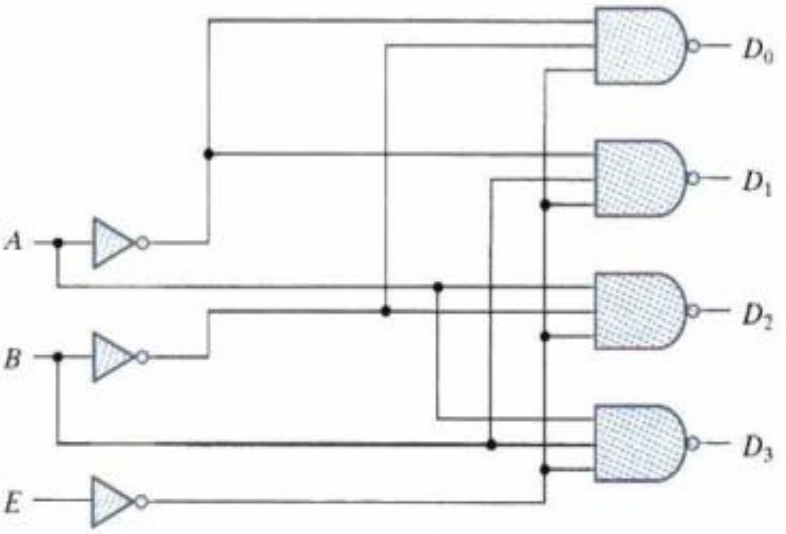
x	y	z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



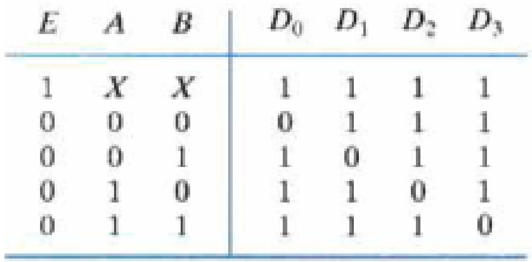
$$S(x, y, z) = \Sigma(1, 2, 4, 7)$$
$$C(x, y, z) = \Sigma(3, 5, 6, 7)$$



Demultiplexer (2-to-4 Line Decoder With Enable Input)



(a) Logic diagram



E	A	B	D_0	D_1	D_2	D_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

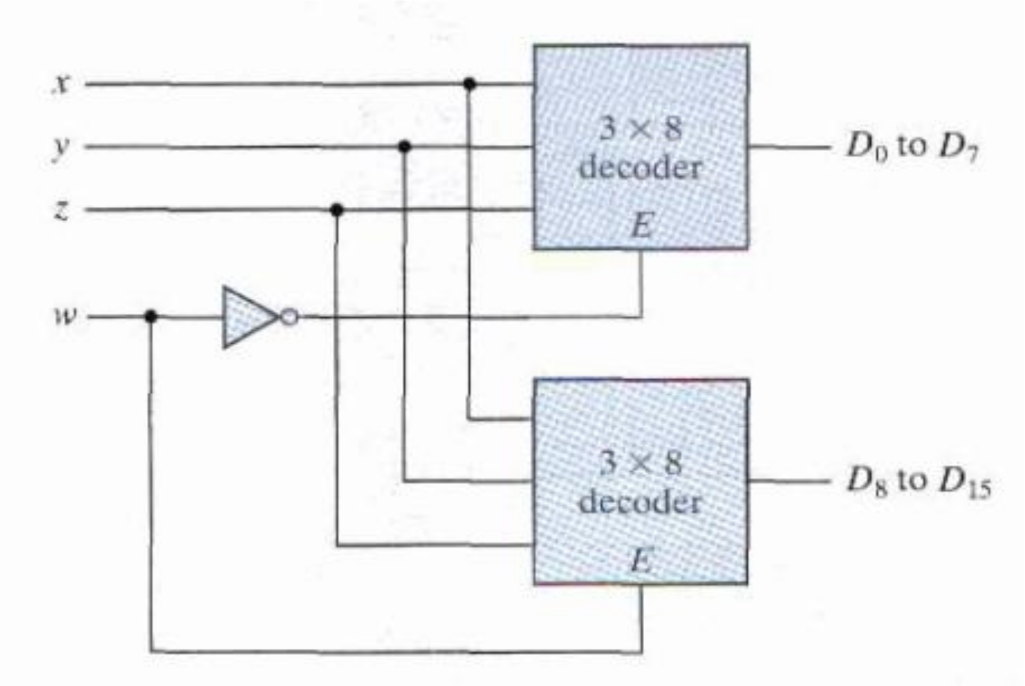
Demultiplexer (2-to-4 Line Decoder With Enable Input)

- Some decoders are constructed with NAND gates.
- Decoders include one or more enable inputs to control the circuit operation.
- The circuit operates with complemented outputs and a complement enable input.
- The decoder is enabled when E is equal to 0.
- Only one output can be equal to 0 at any given time; all other outputs are equal to 1.
- The output whose value is equal to 0 represents the minterm selected by inputs A and B.
- The circuit is disabled when E is equal to 1, regardless of the values of the other two inputs.
- When the circuit is disabled, none of the outputs are equal to 0 and none of the minterms are selected.
- In general, a decoder may operate with complemented or uncomplemented outputs.
- The enable input may be activated with a 0 or with a 1 signal.

Demultiplexer (2-to-4 Line Decoder With Enable Input)

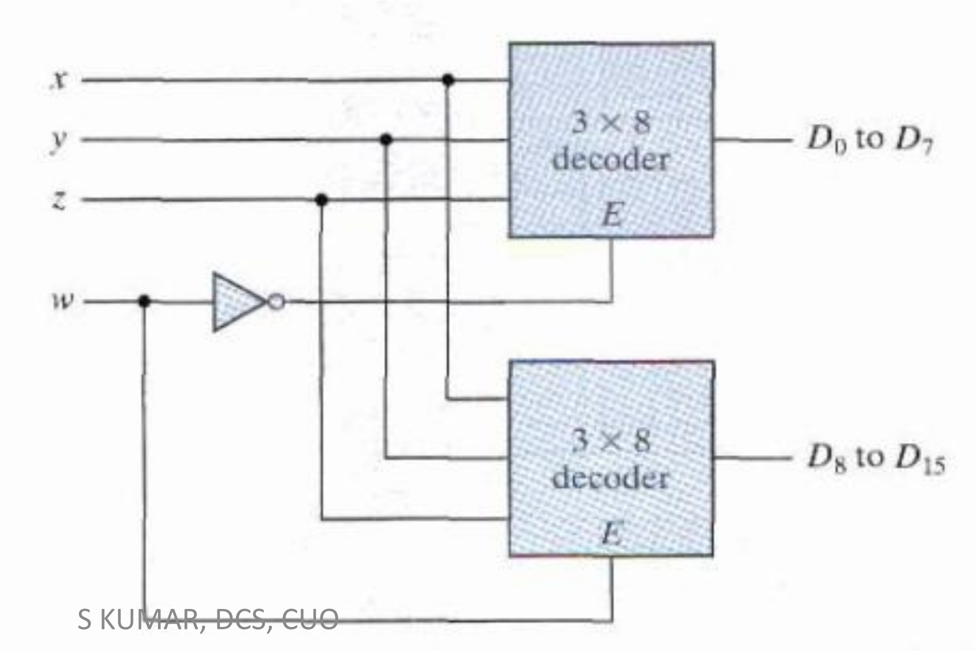
- A decoder with enable input can function as demultiplexer- circuit that receives information from a single Line and directs it to one of $\mathbf{2^n}$ possible output lines.
- The selection of a specific output is controlled by the bit combination of **n** *selection* lines.
- Because decoder and demultiplexer operations are obtained from the same circuit, a decoder with an enable input is referred to as a decoder-demultiplexer.
-

4 X 16 Decoder Constructed With Two 3 X 8 Decoders



4 X 16 Decoder Constructed With Two 3 X 8 Decoders

- Decoders with enable inputs can be connected together to form a larger decoder circuit.
- When *w* = 0,the top decoders enabled and the other is disabled.
- The bottom decoder outputs are all 0's, and the top eight outputs generate minterms *0000* to 0111.
- When *w* = 1,the enable conditions are reversed;The bottom decoder outputs generate minterms 1000 to 1111, while the outputs of *the* top decoder *are* all 0’s .



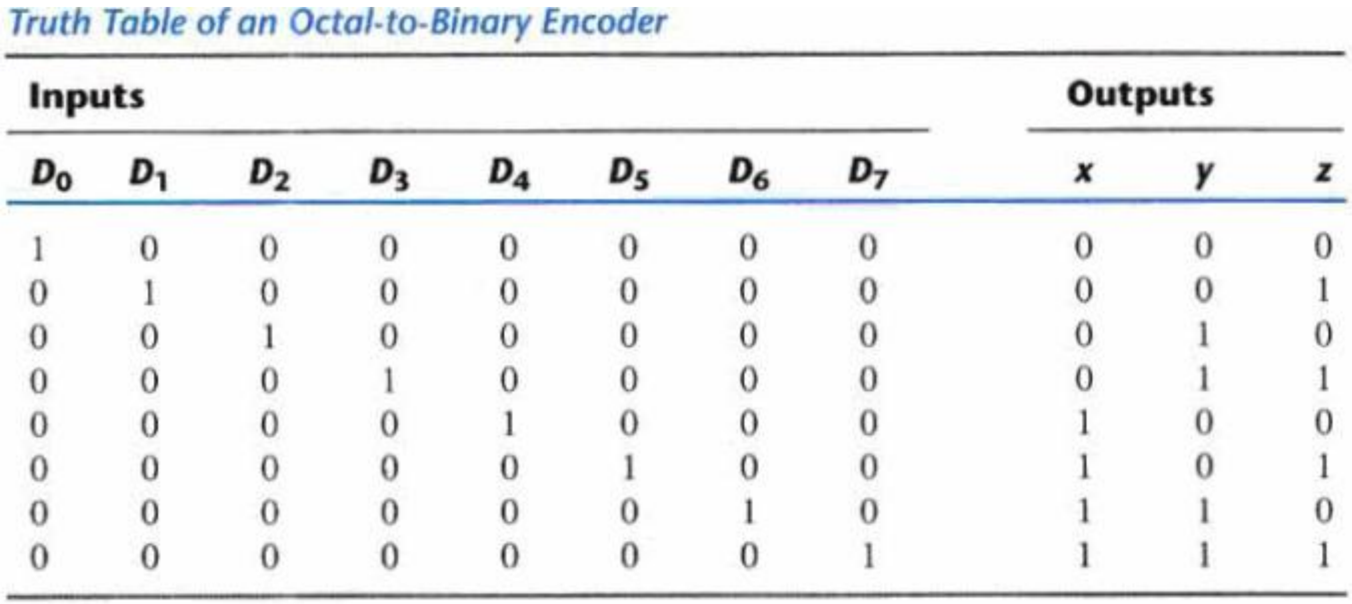
2-to-4 Line Decoder With Enable Input

- A decoder provides the 2^n minterms of *n* input variables.
- Since any Boolean function can be expressed in sum-of-minterms form, a decoder that generates the minterms of the function, together with an external OR gate that forms their logical sum, provides a hardware implementation of the function.
- In this way any combinational circuit with *n* inputs and *m* outputs can be implemented with an n-to-2^n-line decoder and *m* OR gates.
- A function with a long -list of minterms requires an OR Gate - with a large numbers of inputs.
- A function having a list of k minterms can be expressed in its complemented form F' with 2^n - k minterms.
- If the number of minterms in the function is greater than $2^n/2$, then F' can be expressed with fewer minterms.
- In such a case, it is advantageous to use a NOR gate to sum the minterms of F'.
- The output of the NOR gate complements this sum and generates the normal output F.

ENCODER

- An encoder is a digital circuit that performs the inverse operation of a decoder.
- An encoder has 2^n *(or* fewer) input lines and n output lines.
- The output lines, as an aggregate, generate the binary code corresponding to the input value.
- Example: Octal-to-Binary Encoder
- It has eight inputs (one for each of the octal digits)and three outputs that generate the corresponding binary number.
- It is assumed that only one input has a value of 1 at any given time.
- The circuit has 8 inputs and could have 2^8= 256 possible input combinations.
- Only 8 combinations have meaning, others are don’t care conditions.

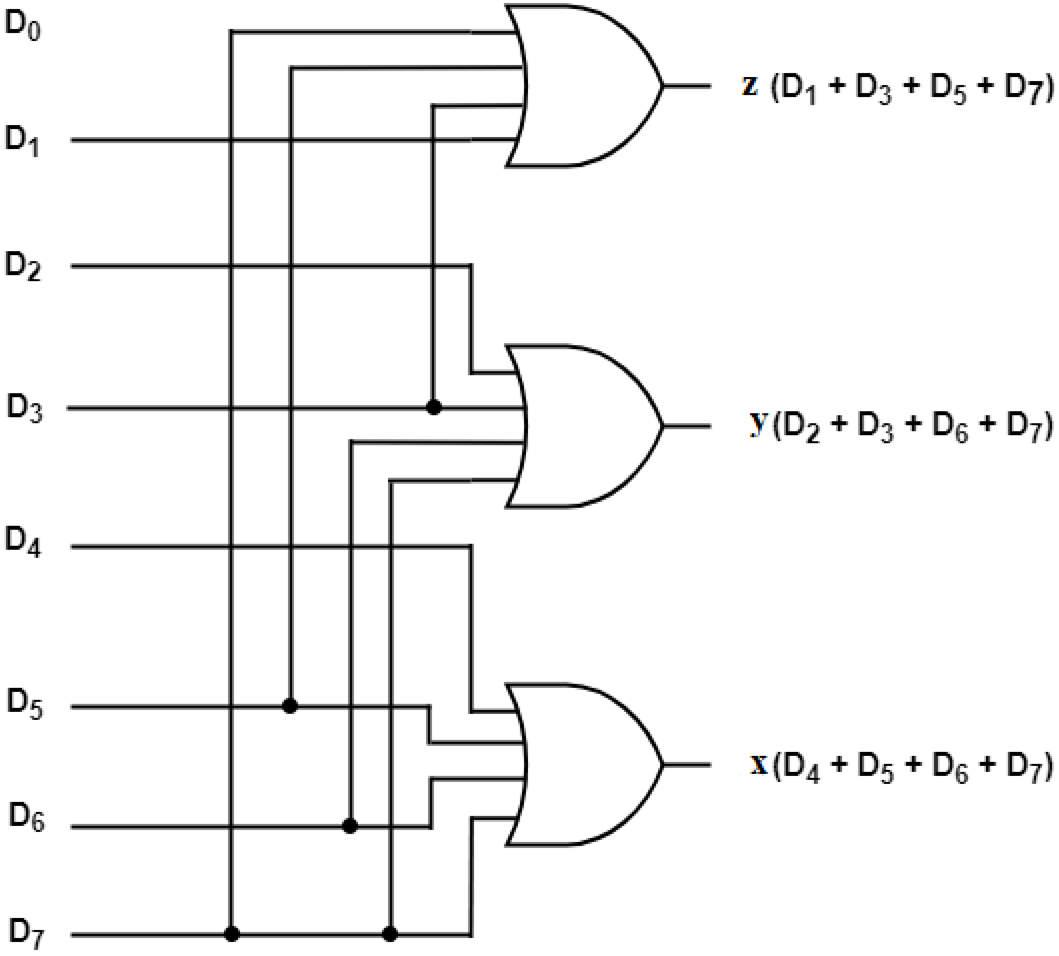
- The encoder can be implemented with OR gates whose inputs are determined directly from the truth table.
- Output ***z*** is equal to 1 when the input octal digit is 1,3,5, or 7.
- Output ***y*** is 1 for octal digits 2,3,6,or 7, and output ***x*** is 1 for digits 4,5,6, *or* 7.
- It has the limitation that only one input can be active at any given time.
- If two inputs are active simultaneously, the output produces an undefined combination.
- Another ambiguity in the octal-to-binary encoder is that an output with all 0's is generated when all the inputs are 0; but this output is the same as when is equal to 1.
- The discrepancy can be resolved by providing one more output to indicate whether at least one input is equal to 1.
- **Priority Encoder:** if two more inputs are equals to 1 at the same time, the input having the highest priority will take precedence.



Truth Table of an Octal-to-Binary Encoder

Inputs								Outputs		
D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	x	y	z
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

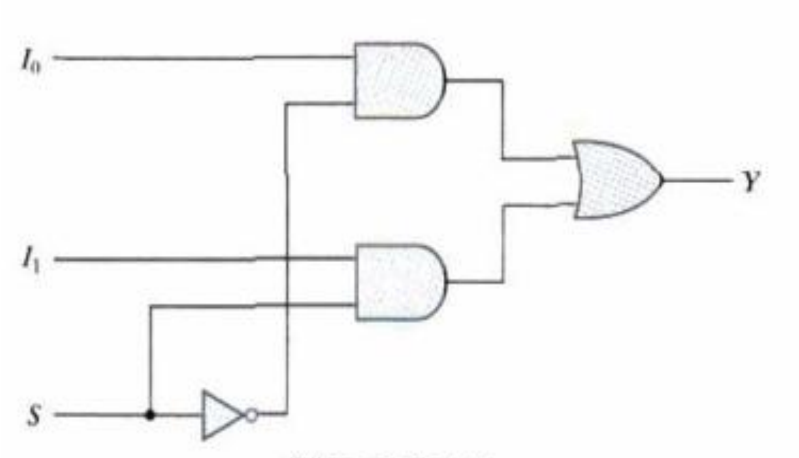
- $z = D_1 + D_3 + D_5 + D_7$
- $y = D_2 + D_3 + D_6 + D_7$
- $x = D_4 + D_5 + D_6 + D_7$


D_0
D_1
D_2
D_3
D_4
D_5
D_6
D_7
z $(D_1 + D_3 + D_5 + D_7)$
y $(D_2 + D_3 + D_6 + D_7)$
x $(D_4 + D_5 + D_6 + D_7)$

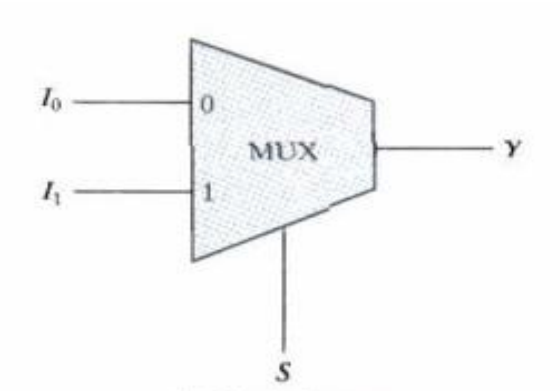
MULTIPLEXER

- It is a combinational circuit hat selects binary information from *one* of many input lines and directs it to a single output line.
- The selection of a particular input line is controlled by set of *selection* lines.
- Normally, there are 2^n input lines and *n* selection lines whose bit combinations determine which input is selected.
- A multiplexer is also called a data selector, since it selects one of many inputs and steers the binary information to the output line.
- Any Boolean function of **n** variables with a multiplexer with **n - 1** selection inputs and $\mathbf{2^{n-1}}$ data inputs .

2-to-1 Line Multiplexer



(a) Logic diagram



(b) Block diagram

The circuit has two data input lines, one output line, and one selection line ***S***.

When S = 0, the upper AND gate is enabled and ***I_0*** has a path to the output.

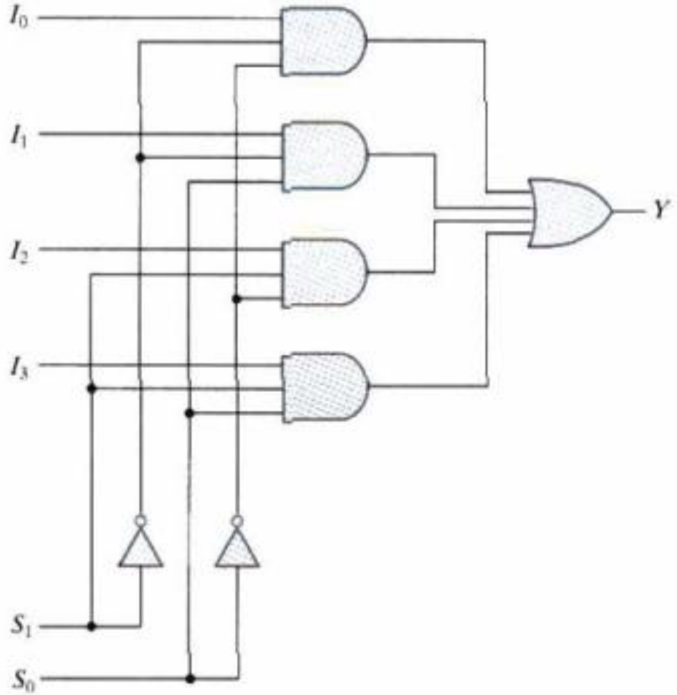
When **S = 1**, the lower AND gate is enabled and ***I_1*** has a path to the output.

The black diagram of a multiplexer is sometimes depicted by a wedge-shaped symbol .

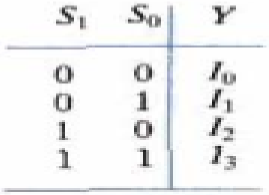
2-to-1 Line Multiplexer

- In general, a **2^n-to-1**-line multiplexer is constructed from an **n-to-2^n** decoder by adding ***2^n*** input lines to it, one to each AND gate.
- The outputs of the AND gates are applied to a single OR gate.
- The *size* of a multiplexer is specified by the number **2^n** of its data input lines and the single output line.
- The ***n*** selection lines are implied from the **2^n** data lines.
- As in decoders, multiplexers may have an enable input to control the operation of the unit.
- When the enable input is in the inactive state, the outputs are disabled, and when it is in the active state, the circuit functions as a normal multiplexer.

4-to-1 Line Multiplexer



(a) Logic diagram

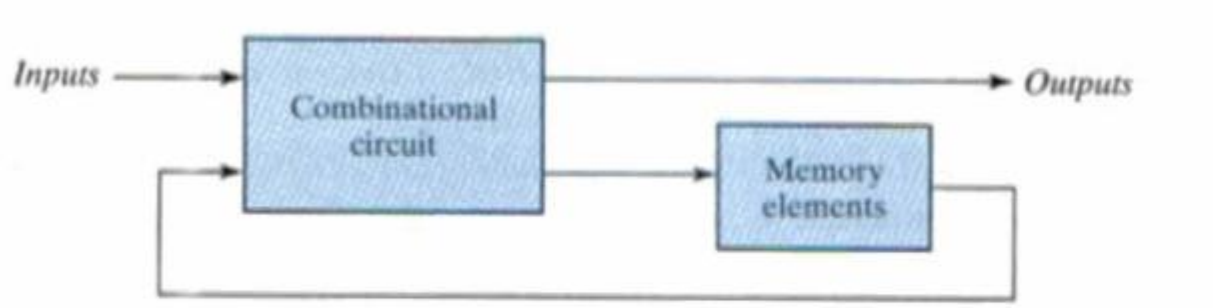


S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

(b) Function table

SEQUENTIAL CIRCUITS

- It consists of a combinational circuit to which storage elements are connected to form a feedback path.
- The storage elements are devices capable of storing binary information.
- The binary information stored in these elements at any given time defines the state of the sequential circuit at that time.
- The output of the circuit depends up on the external input and present state of the storage elements.
- Thus, a sequential circuit is specified by a time sequence of inputs, outputs, and internal states.

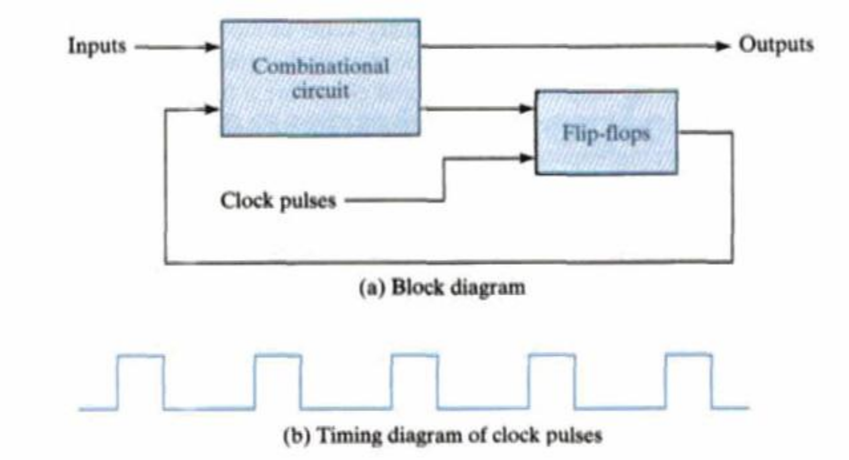


- Based on the timing of signals Sequential circuits are two types such as:
- A **synchronous sequential circuit** is a system whose behavior can be defined from the knowledge of its signals at discrete instants of time.
- The behavior of an ***asynchronous*** **sequential circuit** depends upon the input signals at any instant of time *and* the order in which the inputs change.
- An asynchronous sequential circuit may be regarded as a combinational circuit with feedback.
- Because of the feedback among logic gates, an asynchronous sequential circuit may become unstable at times.

- Synchronization is achieved by a timing device called a clock generator, which provides a clock signal having the form of a periodic train of *clock pulses*.
- The clock signal is commonly denoted by the identifiers clock and *clk*.
- The clock pulses are distributed throughout the system in such a way that storage elements are affected only with the arrival of each pulse.
- In practice, the clock pulses determine when computational activity will occur within the circuit, and other signals (external inputs and otherwise)determine what changes will take place affecting the storage elements and the outputs.

- Synchronous sequential circuits that use clock pulses to control storage elements are called clocked sequential circuits and are the type most frequently encountered in practice.
- They are called synchronous *circuits* because the activity within the circuit and the resulting updating of stored values is synchronized to the occurrence of clock pulses.
- The design of synchronous circuits is feasible.
- The storage elements (memory) used in clocked sequential circuits are called flip-flops.
- A flip-flop is a binary storage device capable of storing one bit of information.
- In a stable state, the output of a flip-flop is either 0 or 1 .
- A sequential circuit may use many flip-flops to store as many bits as necessary.
- The value that is stored in a flip-flop when the clock pulse occurs is also determined by the inputs to the circuit or the values presently stored in the flip-flop(or both).

- The new value is stored(i.e. flip-flop is updated), when a pulse of the clock signal occurs.
- If a clock signal arrives at a regular interval, the combinational logic must respond to a change in the state of the flip-flop in time to be updated before the next clock pulse arrives.
- Propagation delays play in an important role in determining the minimum interval between clock pulses that will allow the circuit to operate correctly.
- The major differences among various storage elements are in the number of inputs they possess and in the manner in which the inputs affect the binary state.
- Storage elements that operates with signal levels are referred to as latches and those controlled by clock transition are flip-flops.
- Latches are level sensitive devices and flip-flops are edge sensitive devices.

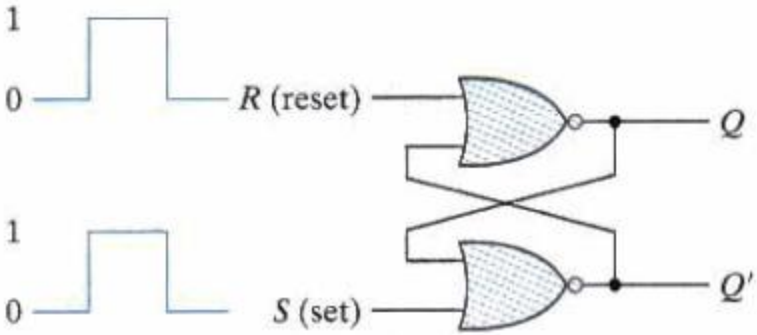


(a) Block diagram

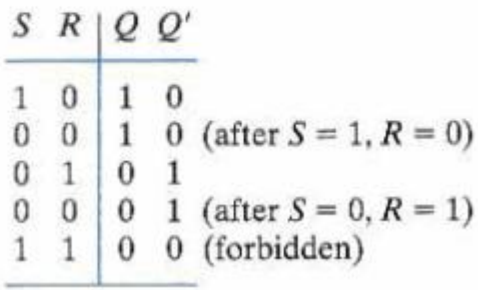
(b) Timing diagram of clock pulses

Basic Flip-Flop Circuit

- A flip-flop circuit can be constructed from two NAND gates or two NOR gates.
- Each circuit forms a basic flip-flop upon which other more complicated types can be built.
- Each flip-flop has two outputs, Q and Q`, and two inputs set and reset.
- This type of flip-flop is sometimes called a direct-coupled RS flip-flop or SR latch.

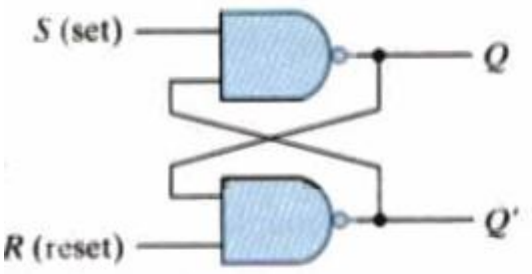


(a) Logic diagram

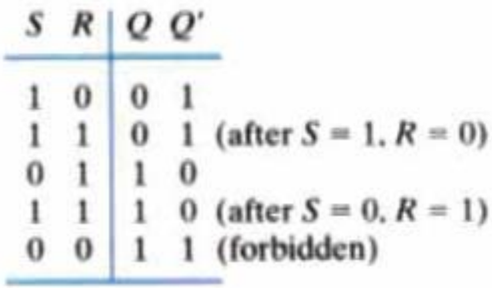


S	R	Q	Q'	
1	0	1	0	
0	0	1	0	(after $S = 1, R = 0$)
0	1	0	1	
0	0	0	1	(after $S = 0, R = 1$)
1	1	0	0	(forbidden)

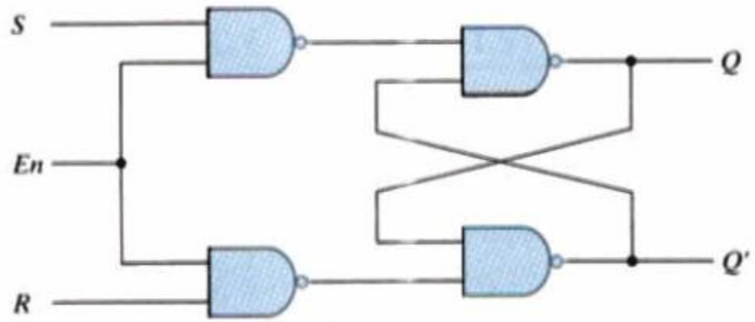
A flip-flop has two useful states.
When Q=1 and Q`=0, it is the set state(or 1-state).
When Q=0 and Q`=1, it is the clear state(or 0-state).
The outputs Q and Q` are complements of each other and are referred to as the normal and complemented outputs, respectively.
The binary state of the flip-flop is taken to be the value of the normal output.
When 1 is applied to both the set and the reset inputs, both outputs go to 0.
This state is undefined and is usually avoided.
If both inputs go to 0, the state of the flip-flop is indeterminate and depends on which input remains a 1 longer before the transition to 0.



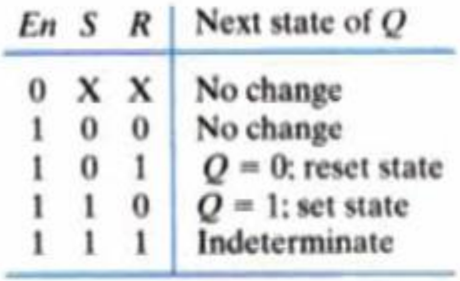
(a) Logic diagram



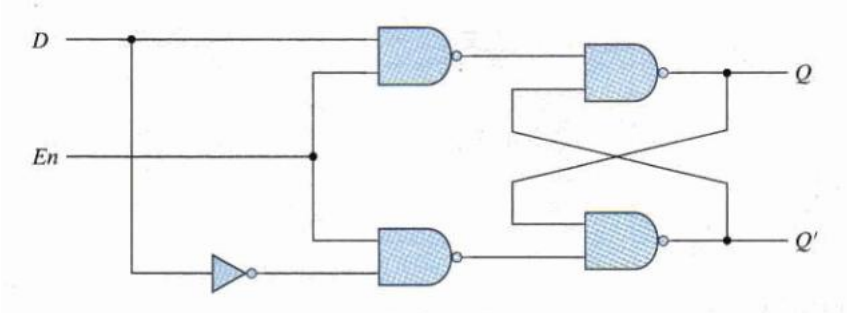
S	R	Q	Q'	
1	0	0	1	
1	1	0	1	(after $S = 1, R = 0$)
0	1	1	0	
1	1	1	0	(after $S = 0, R = 1$)
0	0	1	1	(forbidden)



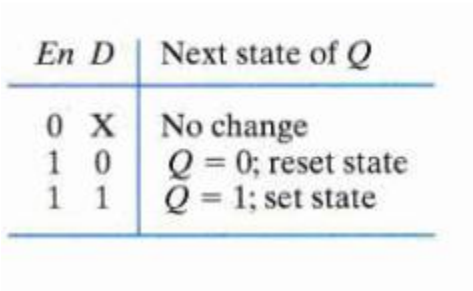
(a) Logic diagram



En	*S*	*R*	Next state of *Q*
0	X	X	No change
1	0	0	No change
1	0	1	*Q* = 0; reset state
1	1	0	*Q* = 1; set state
1	1	1	Indeterminate



(a) Logic diagram



En	*D*	Next state of *Q*
0	X	No change
1	0	$Q = 0$; reset state
1	1	$Q = 1$; set state