Combinational Logic Design Using MSI Circuits

Introductions

Digital systems obtain binary coded data and information continuously being operated on in some manner:

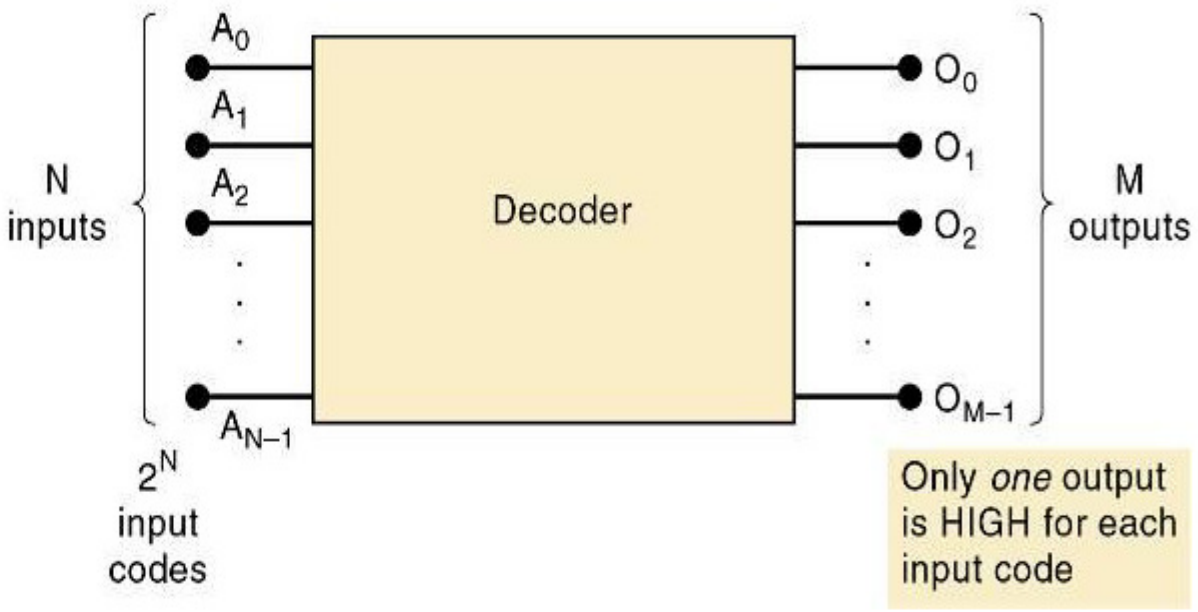
- *Decoding*
- *Encoding*
- *Multiplexing*
- *Demultiplexing*
- *Comparison*
- *Code conversion*
- *Data busing*

These and other operations have been facilitated by the availability of numerous ICs in the MSI (medium-scale-integration) category.

Decoders

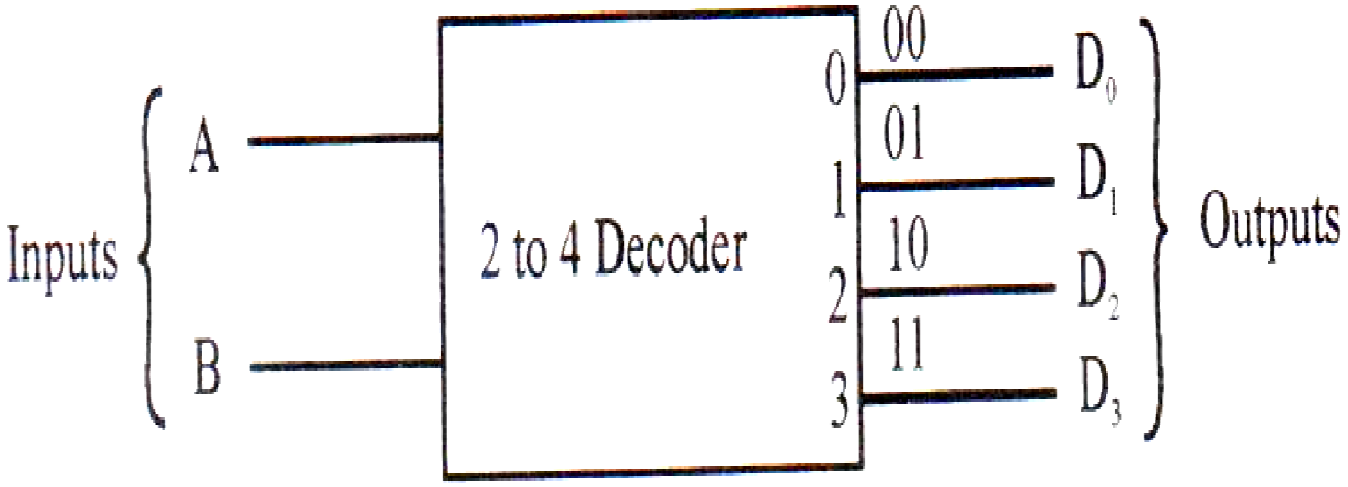
- A **decoder** accepts a set of inputs that represents a binary number—activating only the output that corresponds to the input number.

For each of these input combinations, only one of the *M* outputs will be active (HIGH); all the other outputs are LOW.



Many decoders are designed to produce active-LOW outputs, where only the selected output is LOW while all others are HIGH.

2 to 4 Decoder



Block Diagram of 2 to 4 Decoder

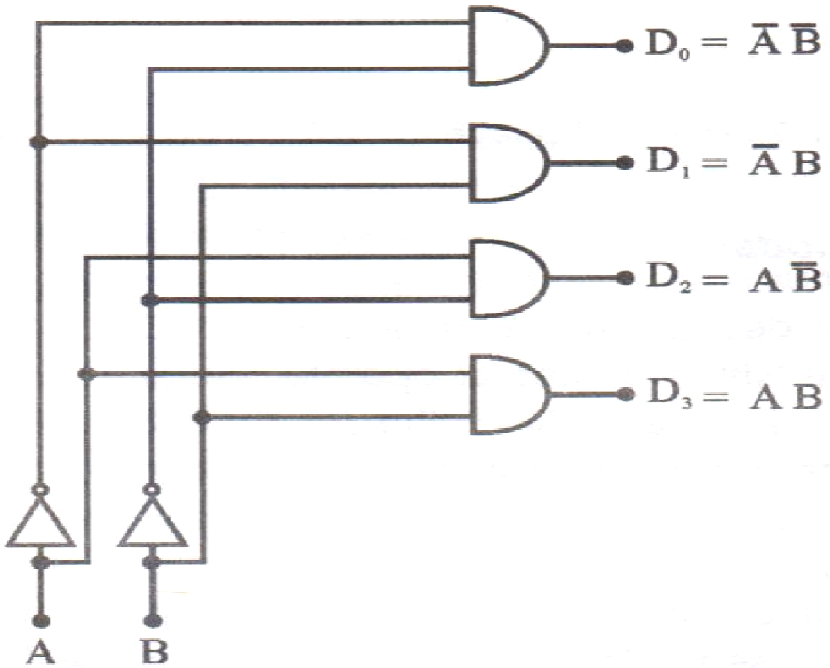
Truth Table
(2 to 4 Decoder)

Input		Output			
A	B	D_3	D_2	D_1	D_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

From the truth table the equation for D_0, D_1, D_2, D_3 can be written as:

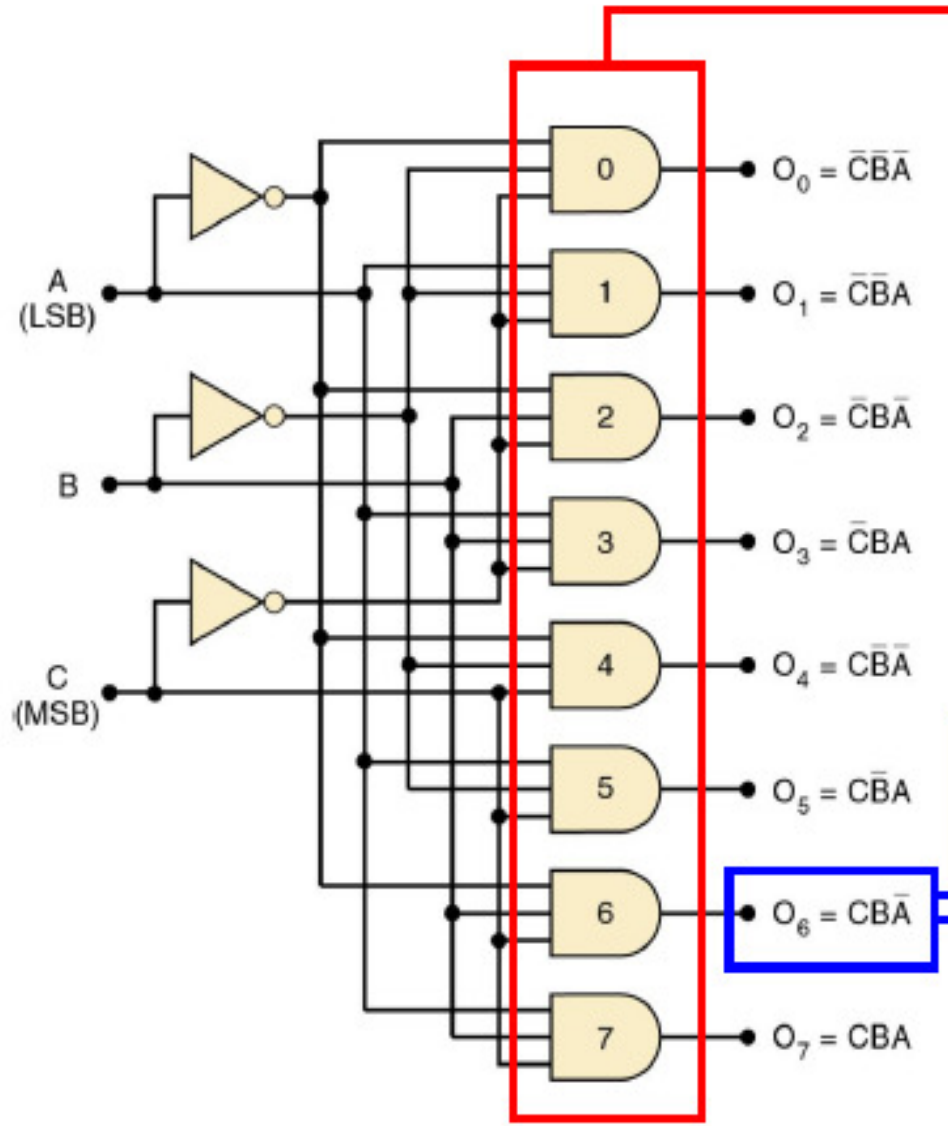
$$D_0 = \overline{A}\,\overline{B}$$
$$D_1 = \overline{A}\,B$$
$$D_2 = A\,\overline{B}$$
$$D_3 = AB$$

Realizing these equations using AND and NOT gates



Combinational circuit of 2 to 4 Decoder

Circuitry for a decoder with three inputs and 8 outputs.

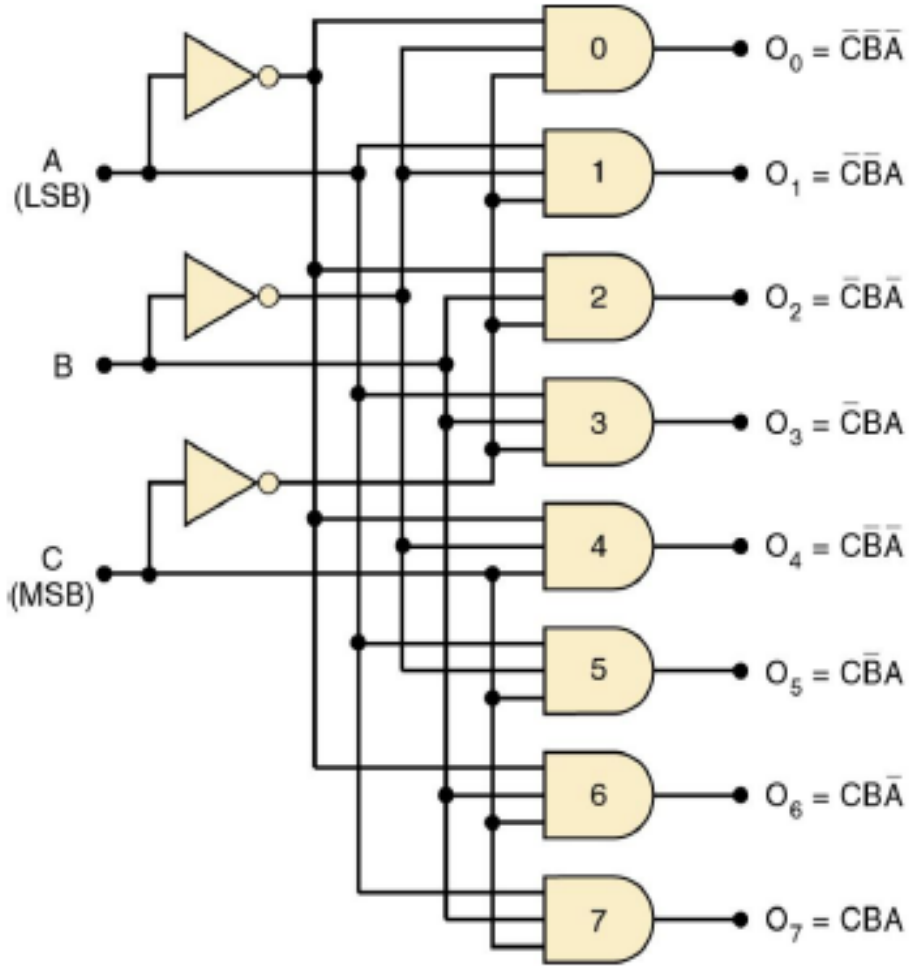


It uses all **AND** gates, so outputs are active-HIGH

C	B	A	O_7	O_6	O_5	O_4	O_3	O_2	O_1	O_0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

Output O_6 goes HIGH only when *CBA* $110_2 = 6_{10}$.

Circuitry for a decoder with three inputs and 8 outputs.



This can be called a *3-line-to-8-line decoder*—it has three input lines and eight output lines.

Also called a *binary-to-octal decoder* or *converter*—taking three-bit binary input code and activating one of eight (octal) outputs.

Also referred to as a *1-of-8 decoder*—only 1 of the 8 outputs is activated at one time.

Decoders

- Some decoders have one or more enable inputs used to control the operation of the decoder.
  - The decoder is enabled only if *ENABLE* is HIGH.
- With common *ENABLE* line connected to a fourth input of each gate:
  - If *ENABLE* is HIGH, the decoder functions normally.
    - *A, B, C* input will determine which output is HIGH.
  - If *ENABLE* is LOW, *all* outputs will be forced LOW.
    - *Regardless* of the levels at the *A, B, C* inputs.

Encoders

- **Most decoders accept an input code & produce a HIGH (or LOW) at *one* and *only one* output line.**

–A decoder identifies, recognizes, or detects a particular code.

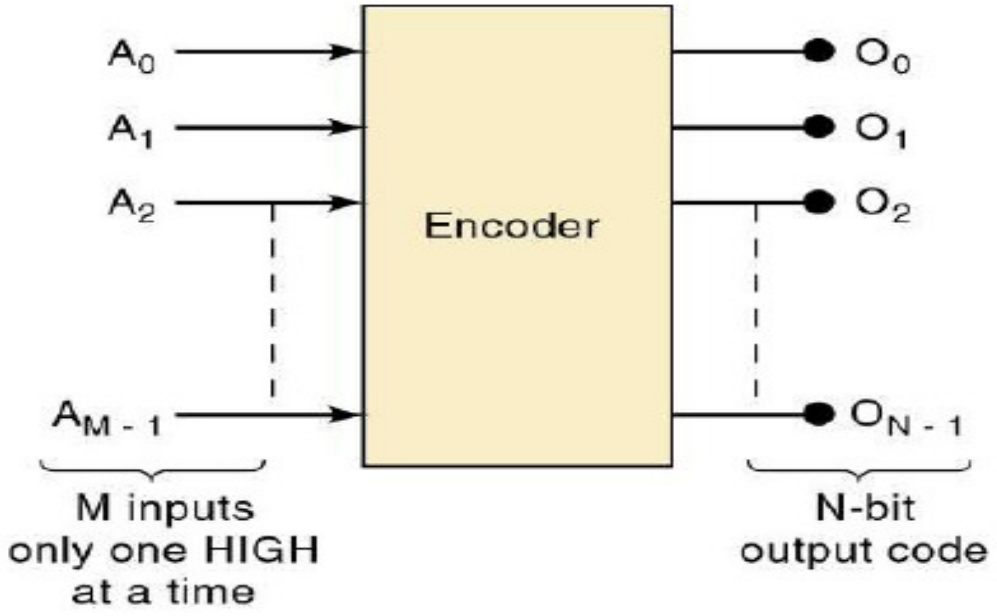
- The opposite of decoding process is **encoding**.
  - Performed by a logic circuit called an **encoder**.

An encoder has a number of input lines, only **one** of which is activated at a given time.

Shown is an encoder with *M* inputs and *N* outputs.

Inputs are active-HIGH, which means that they are normally LOW.

It produces an *N*-bit output code, depending on which input is activated.

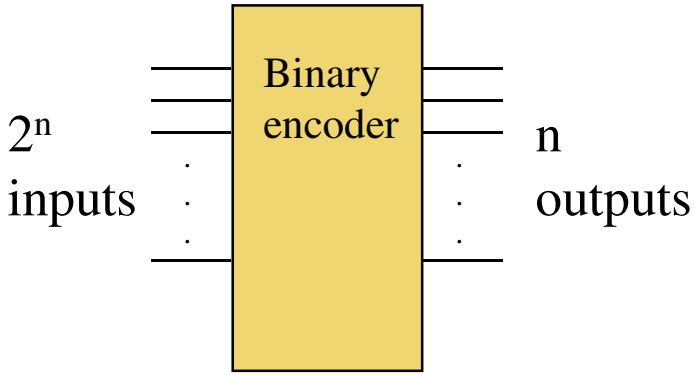


Encoder

- If the a decoder's output code has fewer bits than the input code, the device is usually called an encoder.

  e.g. 2^n-to-n

- The simplest encoder is a 2^n-to-n binary encoder
  - One of 2^n inputs = 1
  - Output is an n-bit binary number



The encoder is a logic circuit that provides the appropriate code (e.g. binary, as output for each input voltage signal.

The commonly used encoders are :

Decimal to BCD Encoder

Octal to Binary Encoder.

Decimal to BCD Encoder

Truth Table

(Decimal to BCD Encoder)

Input										Output			
D_9	D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	D	C	B	A
0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	0	1	0
0	0	0	0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	0	1	0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	0	0	1	0	1
0	0	0	1	0	0	0	0	0	0	0	1	1	0
0	0	1	0	0	0	0	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	0	1	0	0	1

The Boolean expression can be written from the truth table as

$$A = D_1+D_3+D5+D_7+D_9$$

$$B = D_2+D_3+D_6+D_7$$

$$C = D_4+D_5+D_6+D_7$$

$$D = D_8+D_9$$

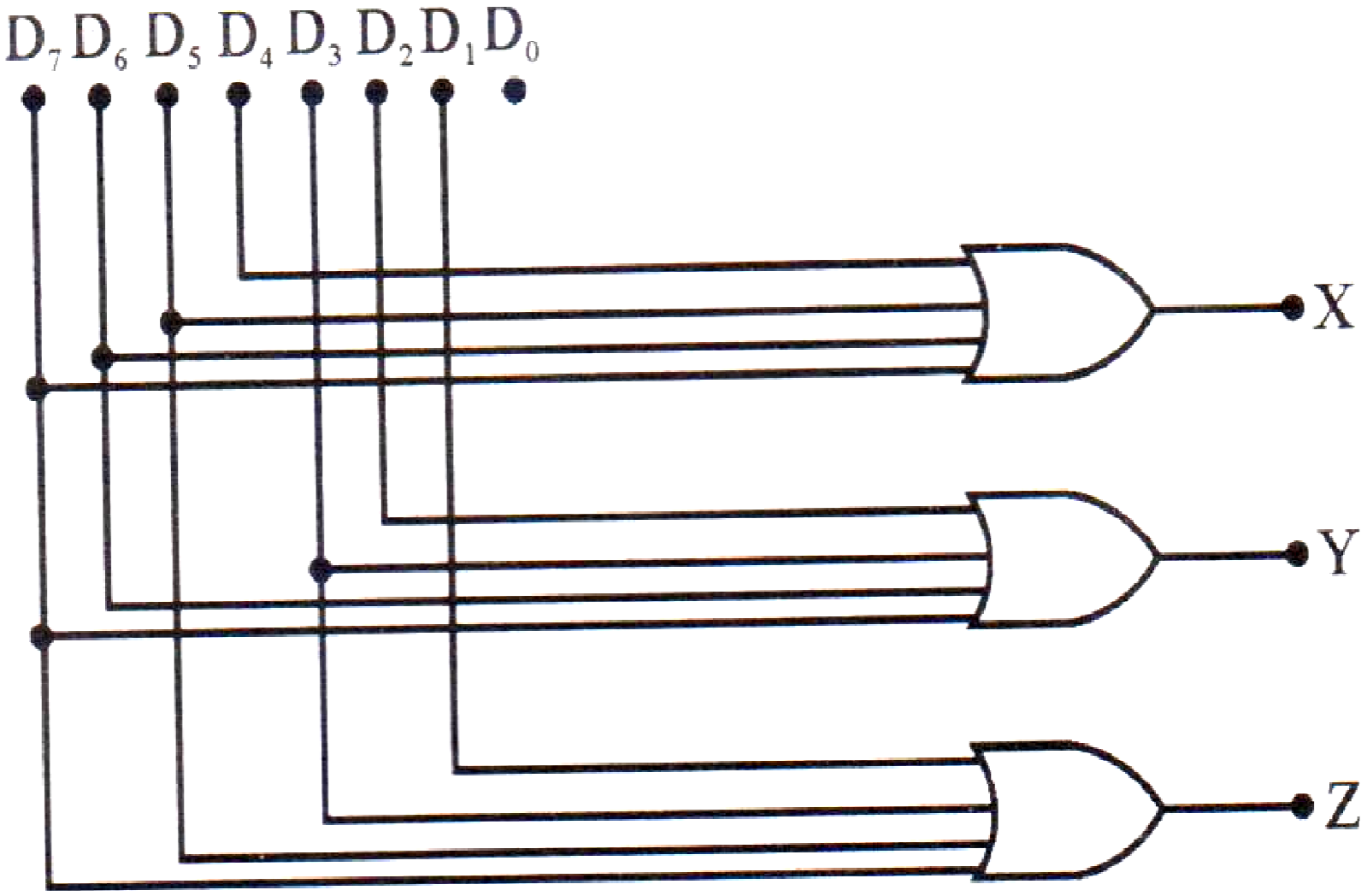
Realizing these equations as shown in the Fig.

$$Y = D_2+D_3+D_6+D_7$$

$$Z = D_1+D_3+D_5+D_7$$

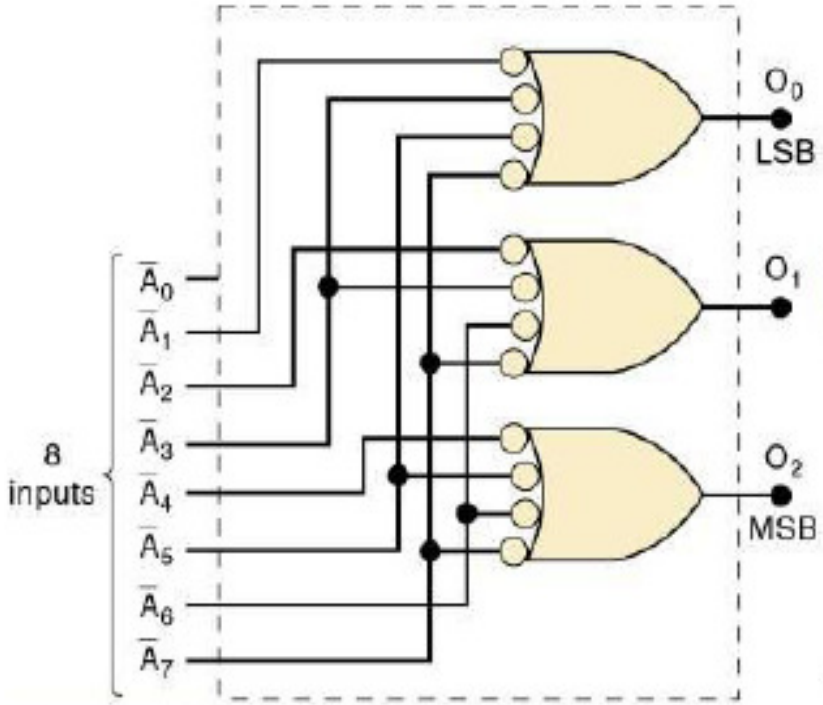
Realizing the outputs using gates

Octal to Binary Encoder Circuit



8 to 3 Encoders

- An *octal-to-binary encoder (8-line-to-3-line encoder)* accepts eight input lines, producing a three-bit output code corresponding to the input.



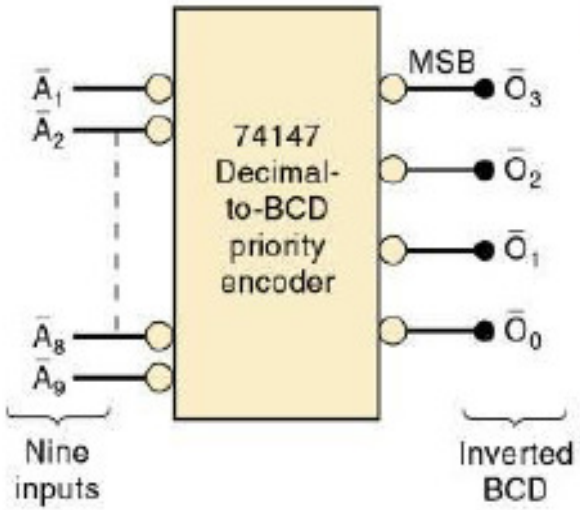
Inputs								Outputs		
$\overline{A}_0$	$\overline{A}_1$	$\overline{A}_2$	$\overline{A}_3$	$\overline{A}_4$	$\overline{A}_5$	$\overline{A}_6$	$\overline{A}_7$	O_2	O_1	O_0
X	1	1	1	1	1	1	1	0	0	0
X	0	1	1	1	1	1	1	0	0	1
X	1	0	1	1	1	1	1	0	1	0
X	1	1	0	1	1	1	1	0	1	1
X	1	1	1	0	1	1	1	1	0	0
X	1	1	1	1	0	1	1	1	0	1
X	1	1	1	1	1	0	1	1	1	0
X	1	1	1	1	1	1	0	1	1	1

*Only one LOW input at a time

Logic circuit for an octal-to-binary (8-line-to-3-line) encoder. Only one input should be active at one time.

Priority Encoder

- A **priority encoder** ensures that when two or more inputs are activated, the output code will correspond to the highest-numbered input.



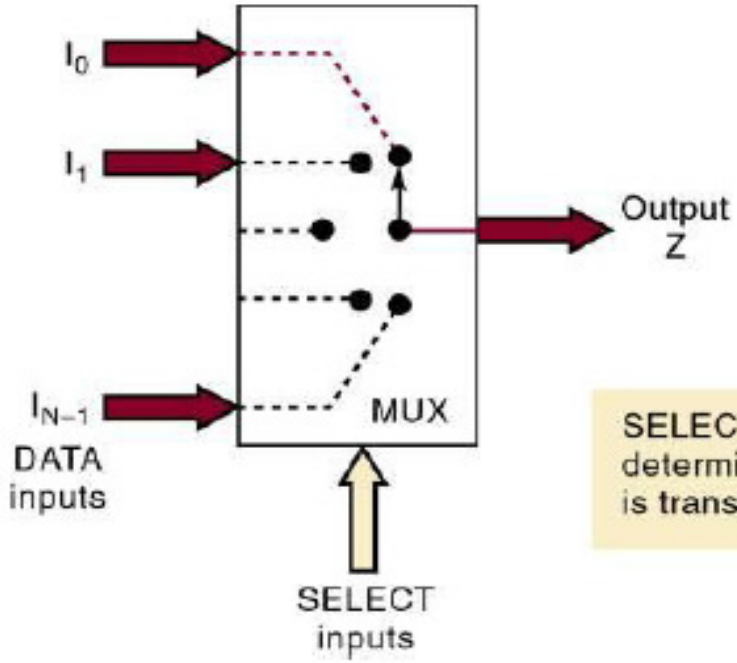
$\bar{A}_1$	$\bar{A}_2$	$\bar{A}_3$	$\bar{A}_4$	$\bar{A}_5$	$\bar{A}_6$	$\bar{A}_7$	$\bar{A}_8$	$\bar{A}_9$	$\bar{O}_3$	$\bar{O}_2$	$\bar{O}_1$	$\bar{O}_0$
1	1	1	1	1	1	1	1	1	1	1	1	1
X	X	X	X	X	X	X	X	0	0	1	1	0
X	X	X	X	X	X	X	0	1	0	1	1	1
X	X	X	X	X	X	0	1	1	1	0	0	0
X	X	X	X	X	0	1	1	1	1	0	0	1
X	X	X	X	0	1	1	1	1	1	0	1	0
X	X	X	0	1	1	1	1	1	1	0	1	1
X	X	0	1	1	1	1	1	1	1	1	0	0
X	0	1	1	1	1	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	1	1	1	0

X = either 0 or 1

It has nine active-LOW inputs represent decimal digits 1 through 9, producing *inverted* BCD code corresponding to the highest-numbered activated input.

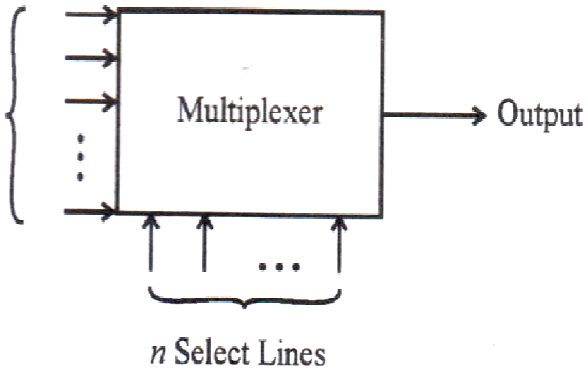
Multiplexer

- A **multiplexer (MUX)** selects 1 of *N* input data sources and transmits the selected data to a single output—called **multiplexing**.
  - A *digital multiplexer* or *data selector* is a logic circuit that performs the same task.



Routing control of desired data input to output by SELECT inputs—referred to as ADDRESS inputs.

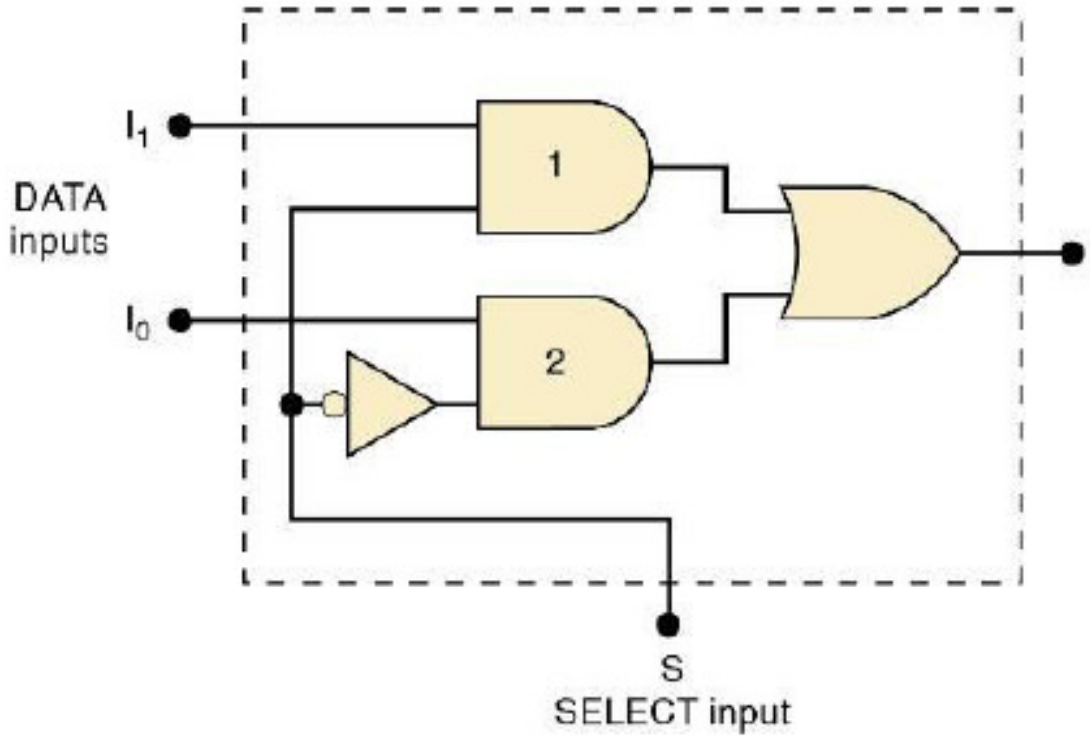
SELECT input code determines which input is transmitted to output Z.



Block Diagram of Multiplexer

Two Input MUX

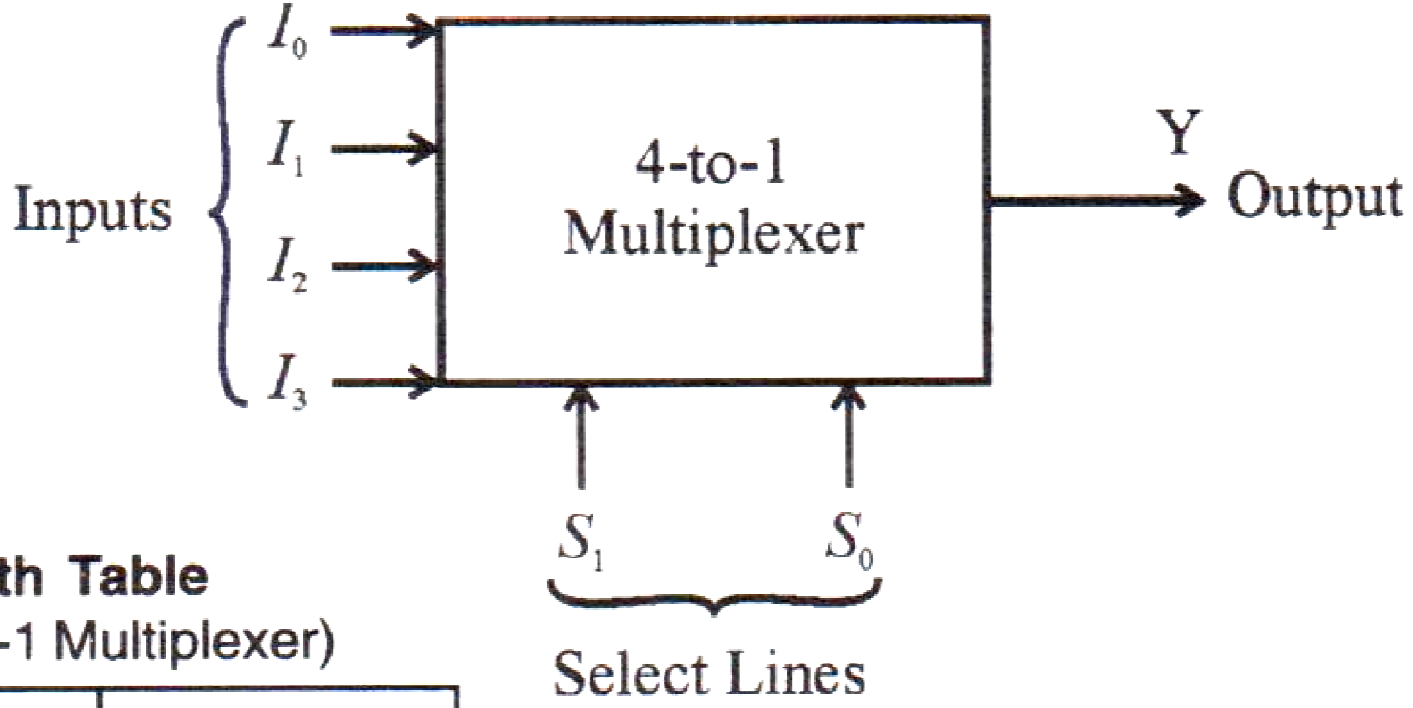
- A two-input MUX could be used in a digital system that uses two different MASTER CLOCK signals.
  - A high-speed clock in one mode and a slow-speed clock for the other.



$Z = I_0 \cdot \bar{S} + I_1 \cdot S$

S	Output
0	$Z = I_0$
1	$Z = I_1$

Four to One Multiplexer



Truth Table
(4-to-1 Multiplexer)

Select Line		Output
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

From the truth table we see that when:

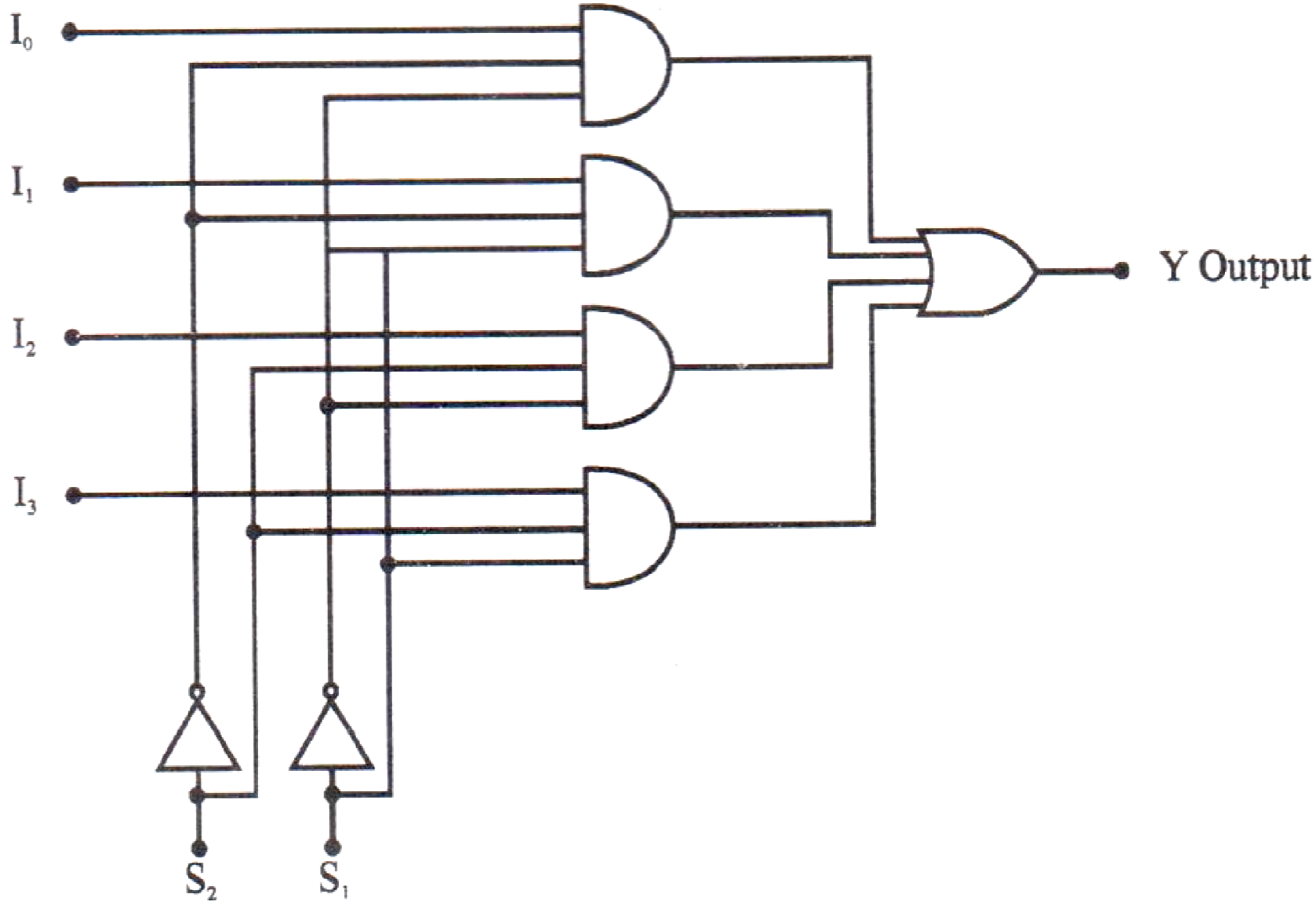
$S_1 = S_0 = 0$, then at the output we get data of I_0

$S_1 = 0$, $S_0 = 1$, then at the output we get data of I_1

$S_1 = 1$, $S_0 = 0$, then at the output we get data of 1_2.

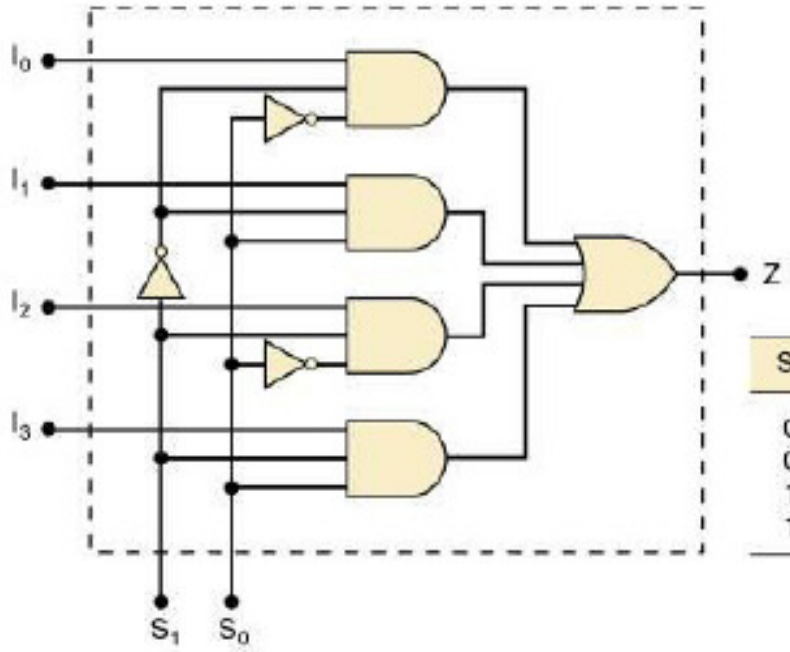
$S_1 = S_0 = 1$, then at the output we get data of I_3.

Combinational Circuit for Four to One Multiplexer

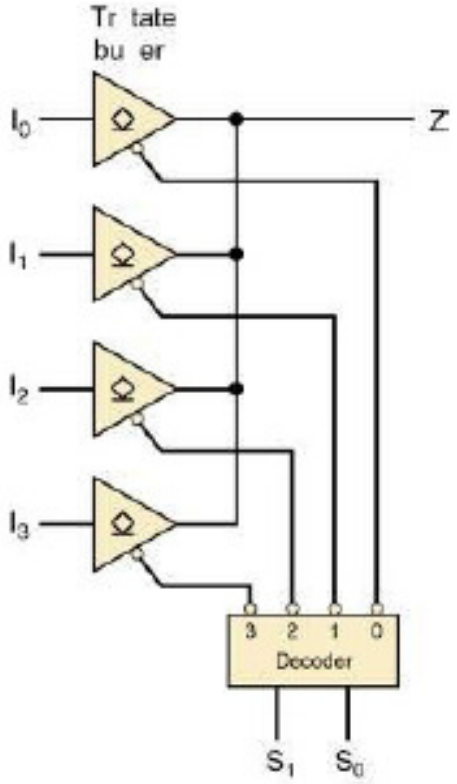


2,4,8,16 Input MUX

- Two-, four-, eight-, and 16-input multiplexers are available in the TTL and CMOS logic families.
  - These basic ICs can be combined for multiplexing a larger number of inputs.



S_1	S_0	Output
0	0	$Z = I_0$
0	1	$Z = I_1$
1	0	$Z = I_2$
1	1	$Z = I_3$

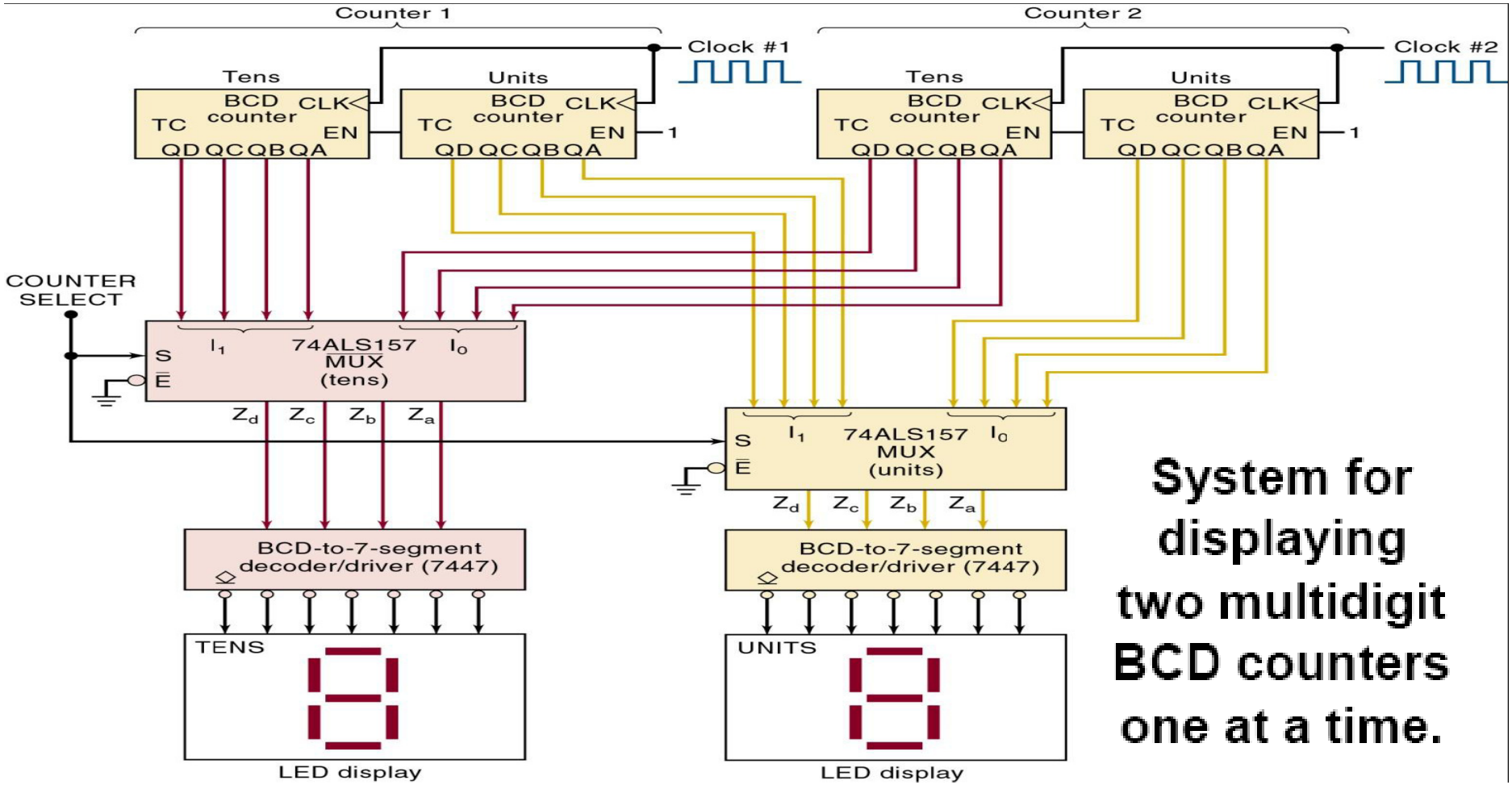


Multiplexer circuits find numerous and varied applications in digital systems of all types.

–Data selection/routing, parallel-to-serial conversion.

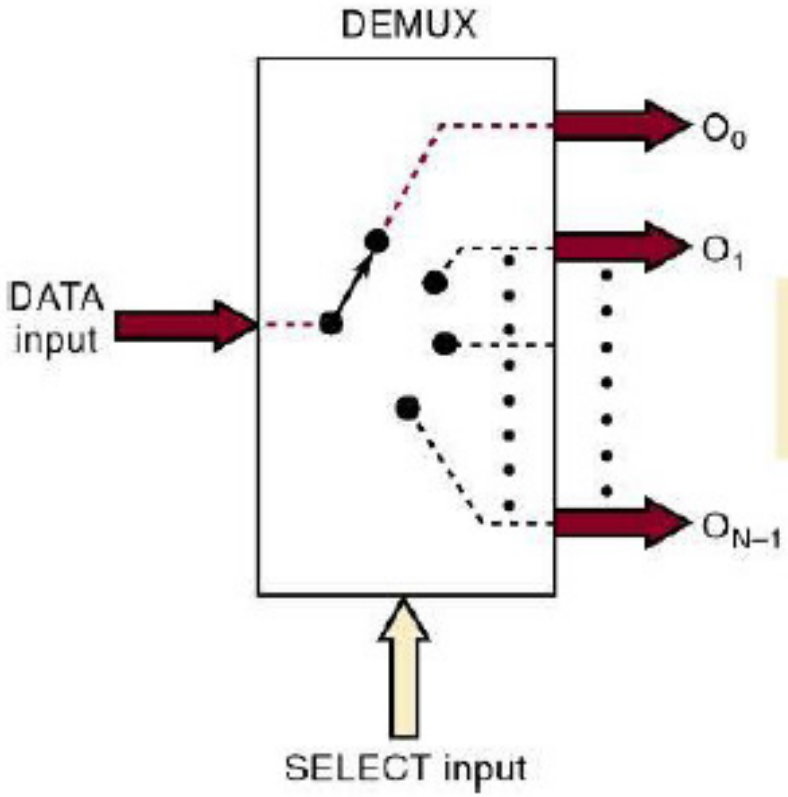
–Operation sequencing.

–Waveform/logic-function generation

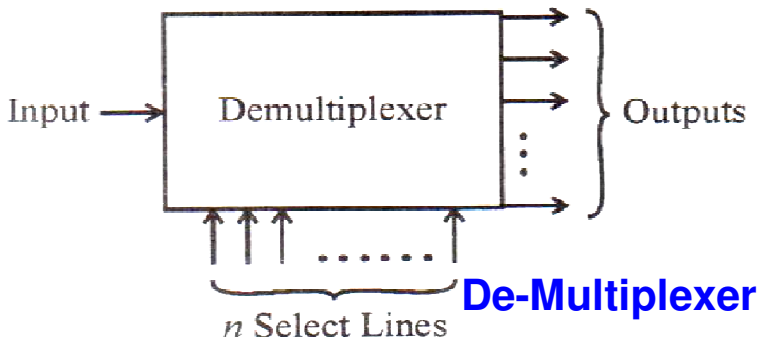


Demultiplexer

- A **demultiplexer (DEMUX)** takes a single input and distributes it over several outputs.
  - The select input code determines to which output the DATA input will be transmitted.

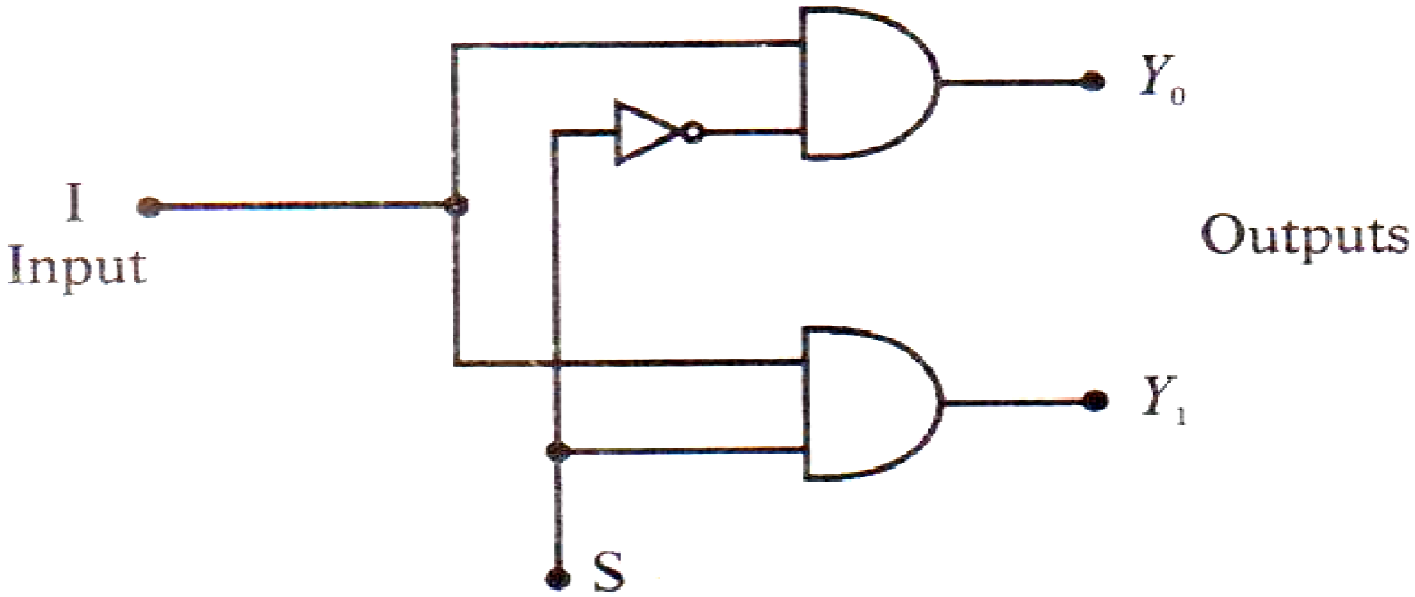


DATA input is transmitted to only one of the outputs as determined by select input code.



De-Multiplexer

One to two Multiplexer

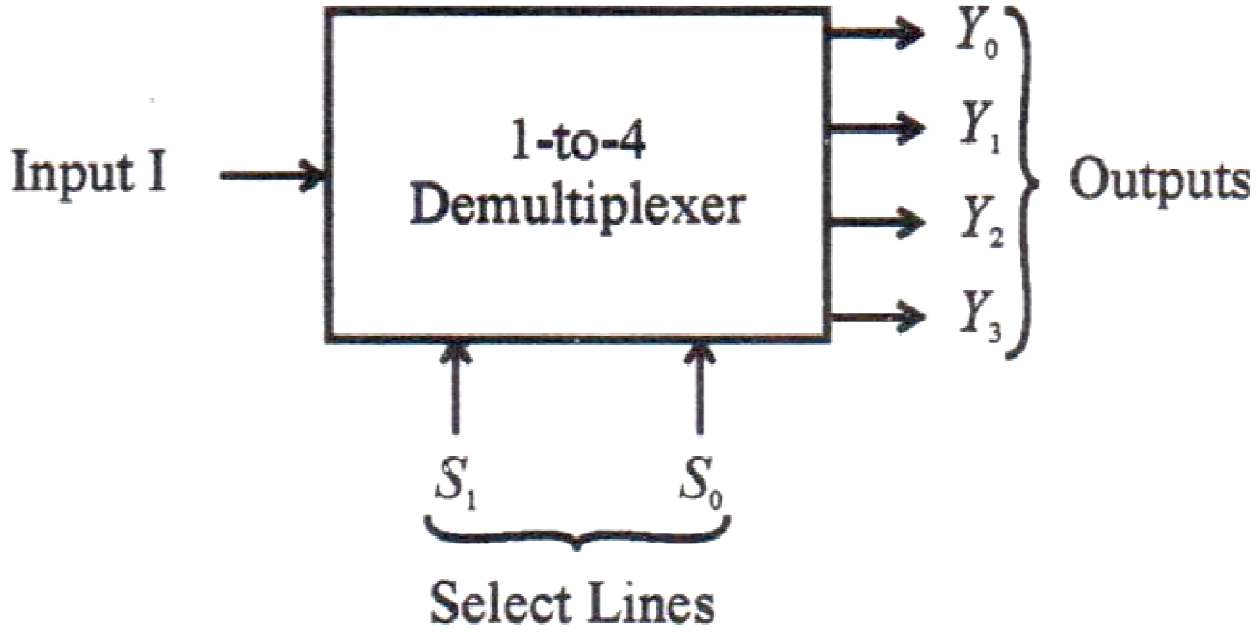


1-to-2 De-multiplexer

Truth Table
(1-to-2 Demultiplexer)

Select Line	Output	
S	Y_1	Y_0
0	0	1
1	1	0

One to Four Demultiplexer



Truth Table
(1-to4 Demultiplexer)

Select Line		Output			
S_1	S_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

From the truth table of 1-to-4 de-multiplexer we see that when

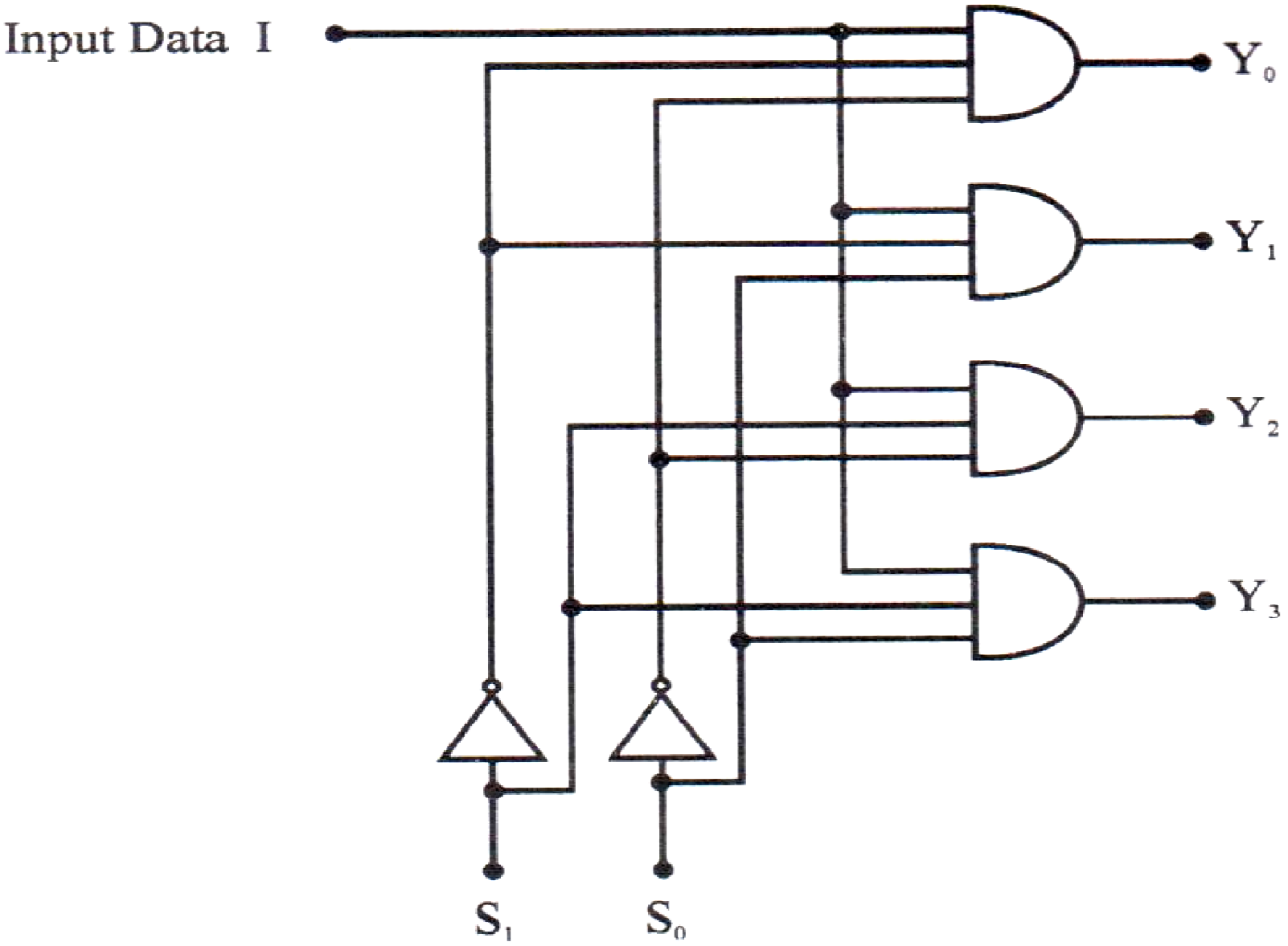
$S_1 = S_0 = 0$, then input data is present on Y_0 output.

$S_1 = 0$, $S_0 = 1$, then input data is present on Y_1 output.

$S_1 = 1$, $S_0 = 0$, then input data is present on Y_2 output.

$S_1 = S_0 = 1$, then input data is present on Y_3 output.

Combination Circuit for One to Four Demultiplexer

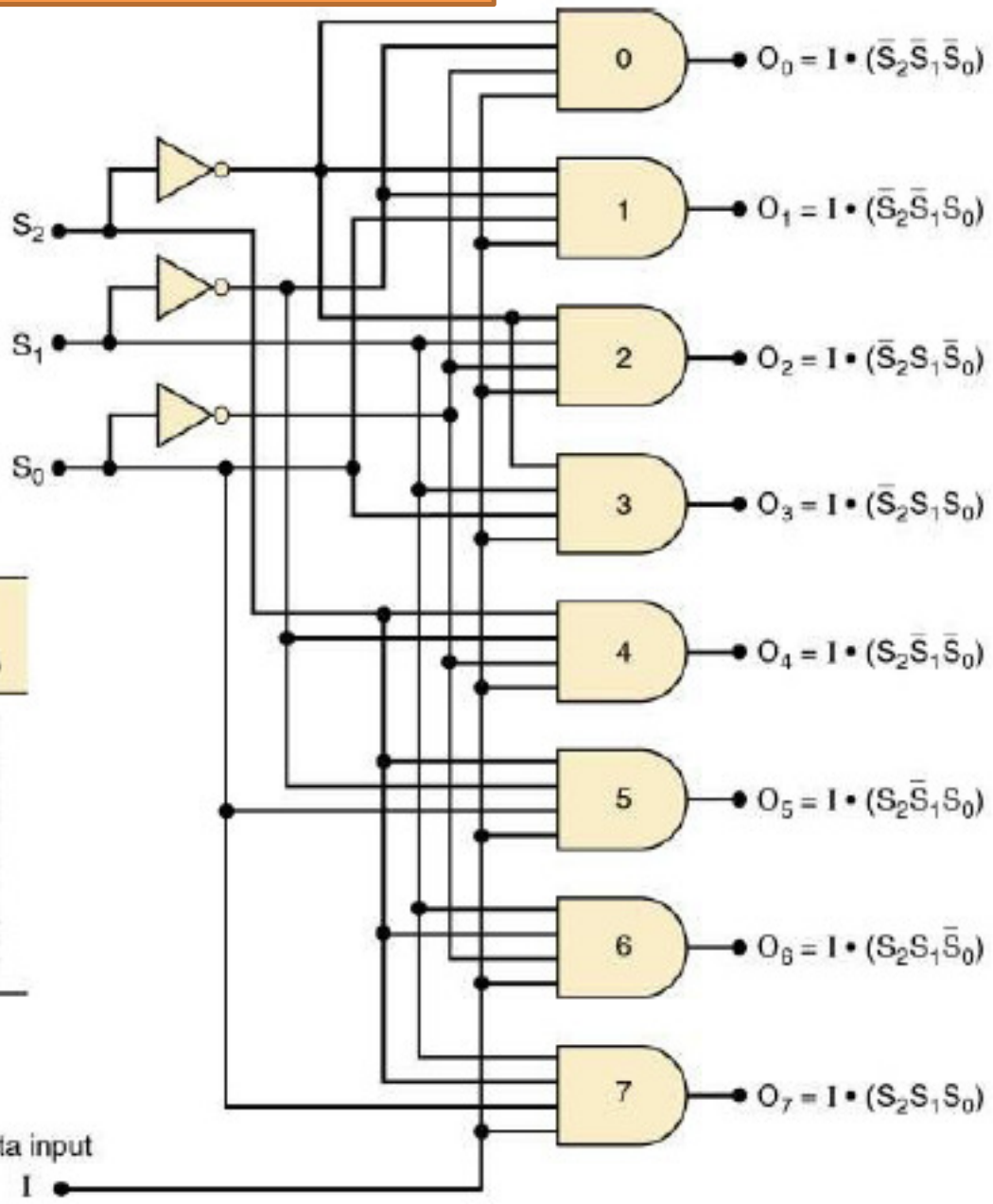


1 to 8 Line De-multiplexer

A 1 line to 8 line demultiplexer.

Select Code			Outputs							
S_2	S_1	S_0	O_7	O_6	O_5	O_4	O_3	O_2	O_1	O_0
0	0	0	0	0	0	0	0	0	0	I
0	0	1	0	0	0	0	0	0	I	0
0	1	0	0	0	0	0	0	I	0	0
0	1	1	0	0	0	0	I	0	0	0
1	0	0	0	0	0	I	0	0	0	0
1	0	1	0	0	I	0	0	0	0	0
1	1	0	0	I	0	0	0	0	0	0
1	1	1	I	0	0	0	0	0	0	0

Note: I is the data input



Block Diagram of 1-to 16 Demultiplexer