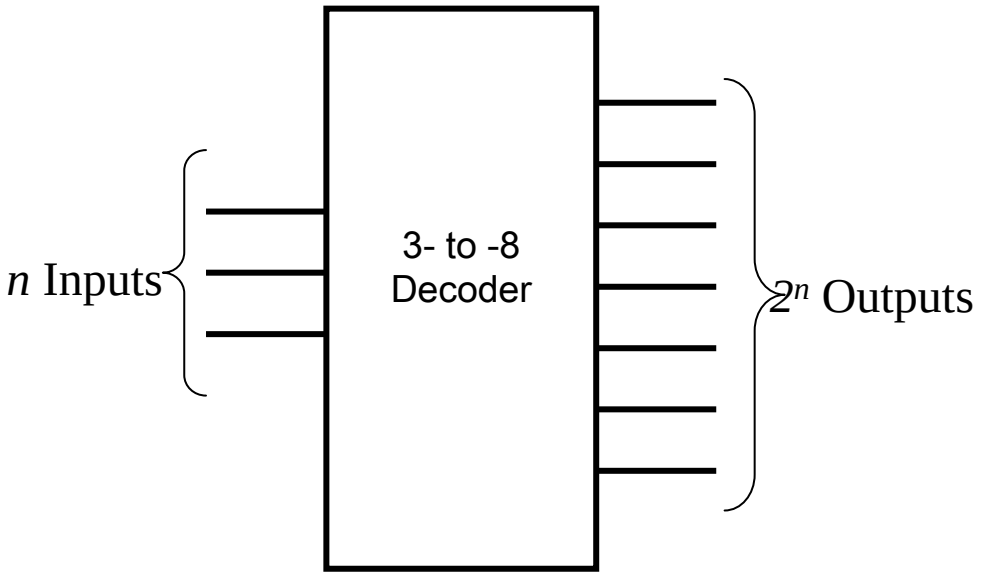
Combinational Logic Implementation Using Decoders, Encoders & Multiplexers

Combinational Circuits

- The outputs are a function of the present set of inputs only
- The inside of a combinational circuit is made of logic gates
- Combinational logic circuits are important components of digital systems
- Each output can be thought of as a function of all the inputs – if there are *m* outputs and *n* inputs then there are *m* Boolean functions, one describing each output

Decoders

- *A* decoder is a combinational logic circuit that converts binary information from *n* input lines to a maximum of 2^n unique outputs
- Also called the *n*-to-*m* line decoders for example:
  - 2-to-4 line decoder
  - 3-to-8 line decoder



Purpose: to generate the 2^n minterms of n input lines

: Typical applications Binary to Octal Converter, BCD-to-7 segment display decoder

Inner Structure of Decoder

Generally: The inner structure of Decoders is composed of an array of AND or NAND gates that generate the required minterm.

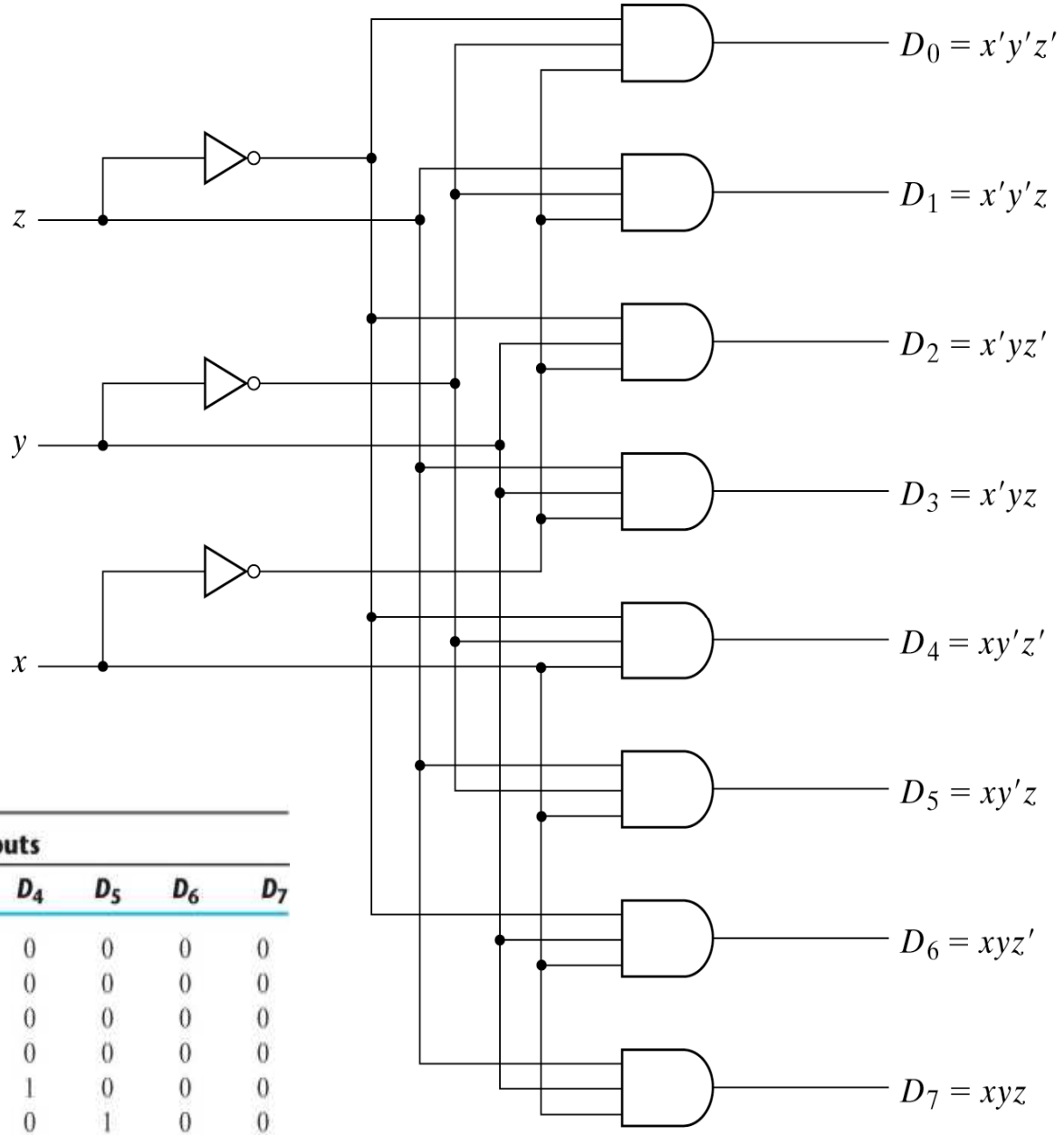
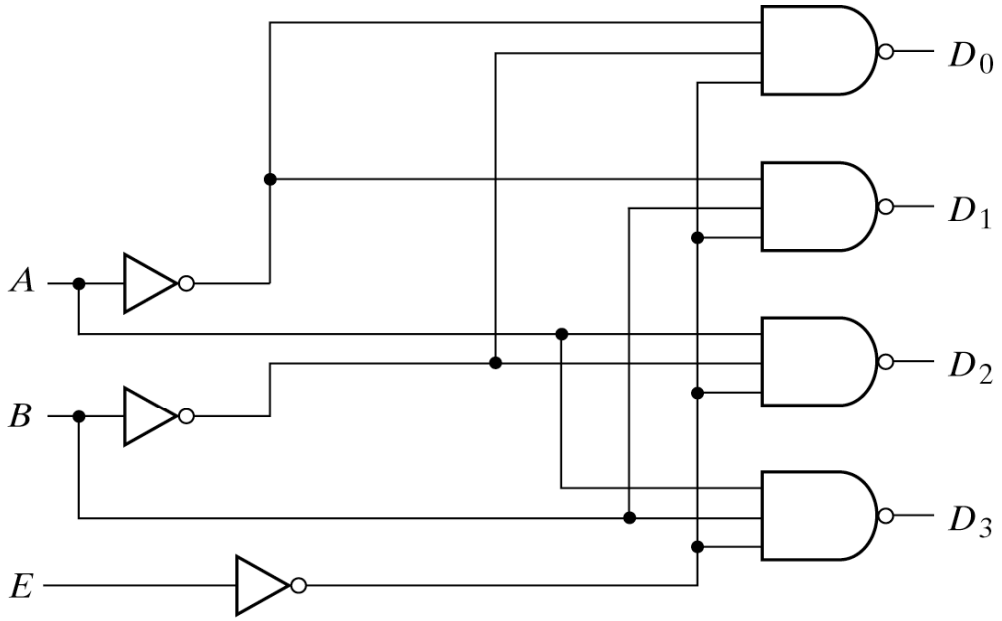


Fig. 4-18 3-to-8-Line Decoder

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

NAND Decoder with Enable Line



(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

(b) Truth table

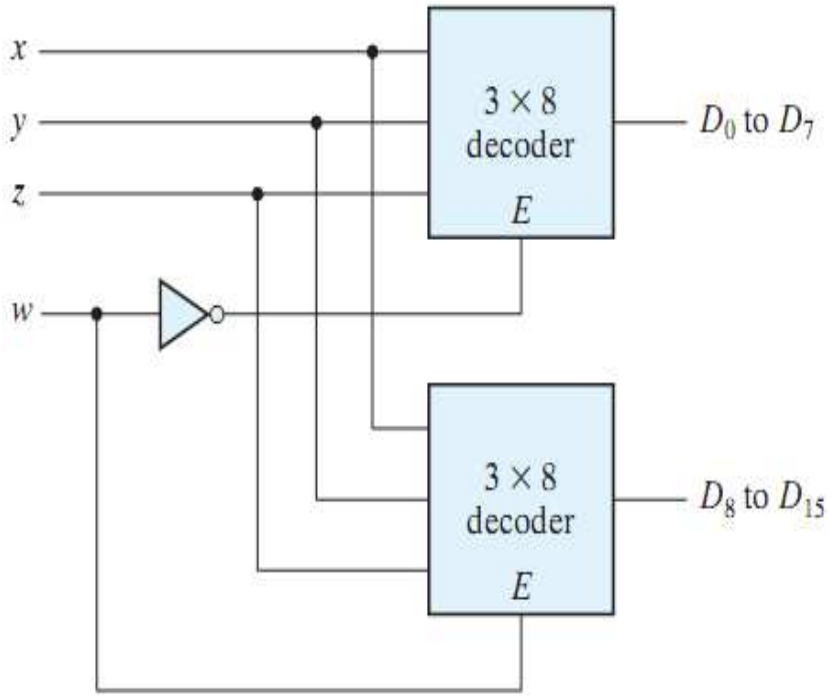
Fig. 4-19 2-to-4-Line Decoder with Enable Input

One or more input control line may be used to control the operation of decoder

Decoders with enable lines can work as de-multiplexer where data is provided through the enable lines and the input variables are used to select specific output channel

Decoder Expansion

- Smaller Decoders with enable lines can be used to build bigger Decoders.
- In the figure two 3-8 decoders are used to build a 4-16 decoder.
- Input w is used as enable line, when w=0, the upper decoder is enabled so, outputs D_0 to D_7 are available, while D_8- to- D_{15} are all zeros. When w=1 the operation is reversed and the lower decoder is enabled while all the outputs of the upper encoder are zeros



4x16 Decoder constructed with two 3x8 Decoders

Combinational Circuit Implementation Using Decoder

- **Since decoders produce 2^n minterms of n- input variables, an external OR gates can be used to form a logical function in SOP form**
- **To implement a given function with decoder and external gates:**

  - express the function as canonical SOP

  - select decoder that has no of inputs equat to the number of input variables in the given functions,

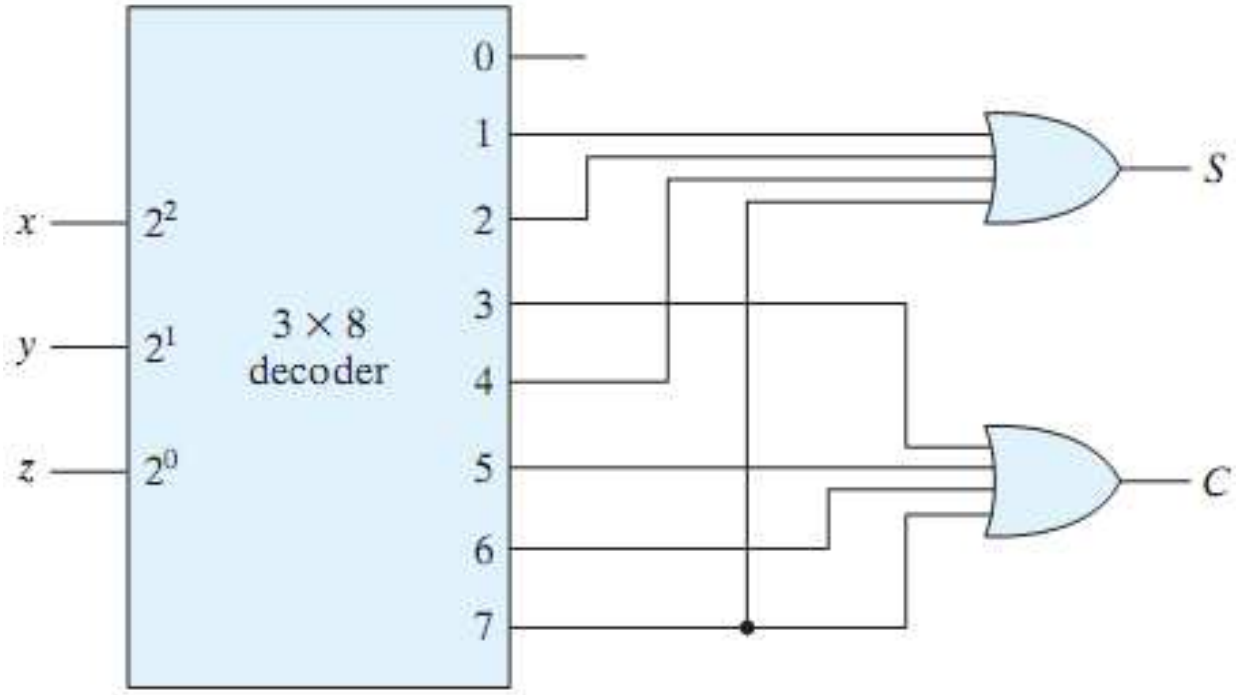
  - select the proper external gate
- **E.g. Implement full adder circuit whose outputs are given as:**

  S (x,y,z)=Σ (1, 2, 4, 7)

  C (x,y,z)=Σ (3, 5, 6, 7)

With a suitable decoder and external gates

Combinational Circuit Implementation Using Decoder



Implement full adder circuit whose outputs are given as:

S (x,y,z)=Σ (1, 2, 4, 7)

C (x,y,z)=Σ (3, 5, 6, 7)

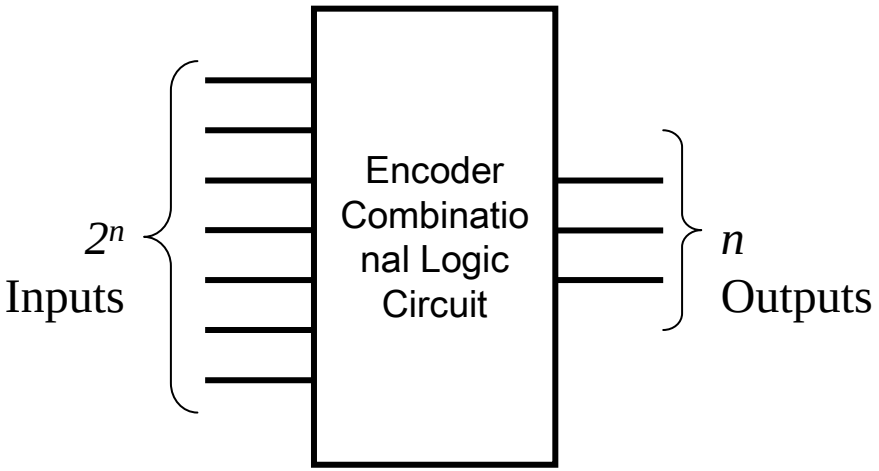
With a suitable decoder and external gates

Combinational Circuit Implementation Using Decoder

- If the number of minterms > $2^n/2$ then express function as F' and use NOR gate in the external gate to obtain the function F.
- If NAND gates are used to construct the decoder, then the external gate must be NAND gate (instead of OR gate)

Encoders

- Performs the inverse operation of a decoder
- Has 2^n or fewer input lines and n output lines
- The output generates the binary code corresponding to the input value



$z = D_1 + D_3 + D_5 + D_7$

$y = D_2 + D_3 + D_6 + D_7$

$x = D_4 + D_5 + D_6 + D_7$

Limitation:

1- only one of the input is allowed to be 1

2- when all inputs are zeros, the output is zero but this situation is the same as input D0=1!!

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

Priority Encoder

- It is an encoder circuit that include the priority function.
- The operation of the priority encoder is such that if two or more inputs are equal to 1 at the same time, the input having the highest priority will take precedence.

Truth Table of a Priority Encoder

Inputs				Outputs		
D_0	D_1	D_2	D_3	x	y	V
0	0	0	0	X	X	0
1	0	0	0	0	0	1
X	1	0	0	0	1	1
X	X	1	0	1	0	1
X	X	X	1	1	1	1

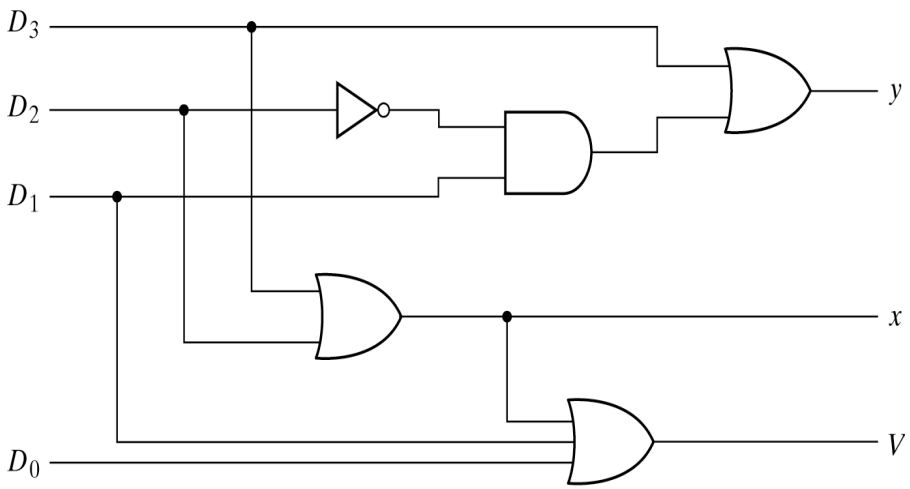


Fig. 4-23 4-Input Priority Encoder

$$x = D_3 + D_2$$

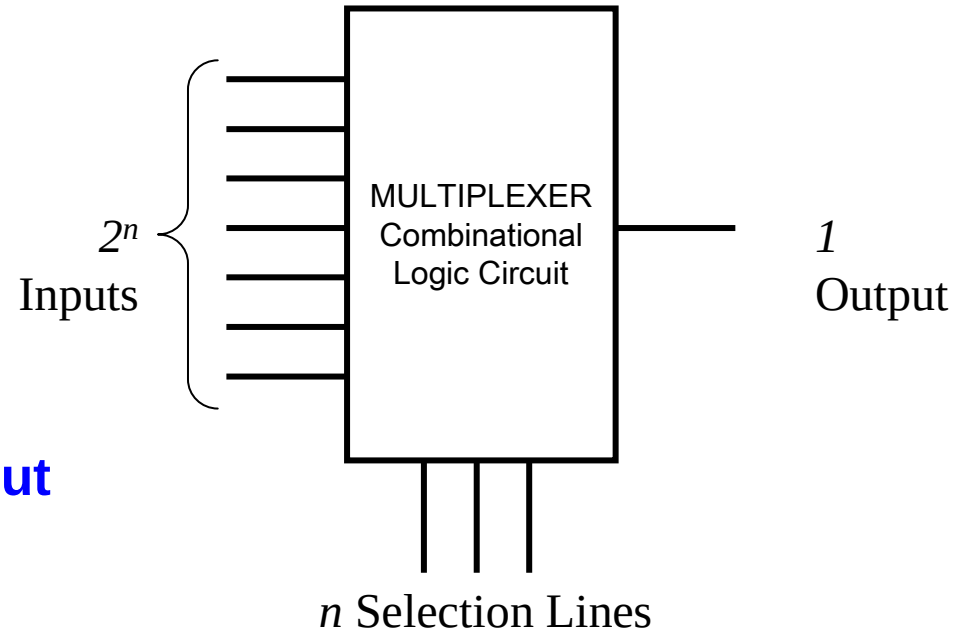
$$y = D_3 + D_1\overline{D}_2$$

$$V = D_3 + D_2 + D_1 + D_0$$

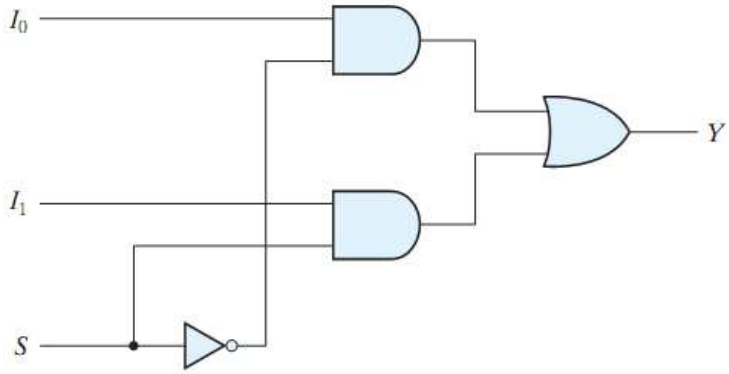
V: is the valid bit indicator that is set to 1 when one or more inputs are equal to 1. If all inputs are zero, there is no valid input and V=0

Multiplexers

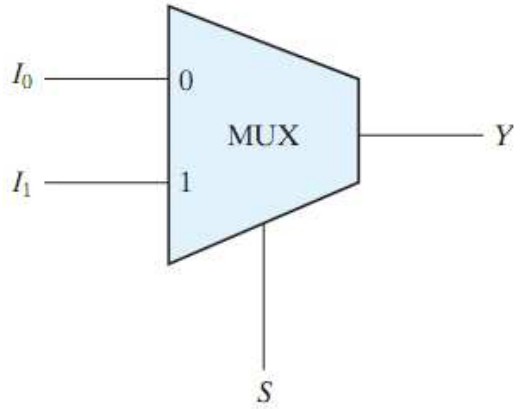
- A multiplexer or Data Selector is a combinational circuit that selects binary information from one of many input lines and directs the information to a single output line



The selection of a particular input line is controlled by a set of selection lines. For 2^n data line we have n selection lines

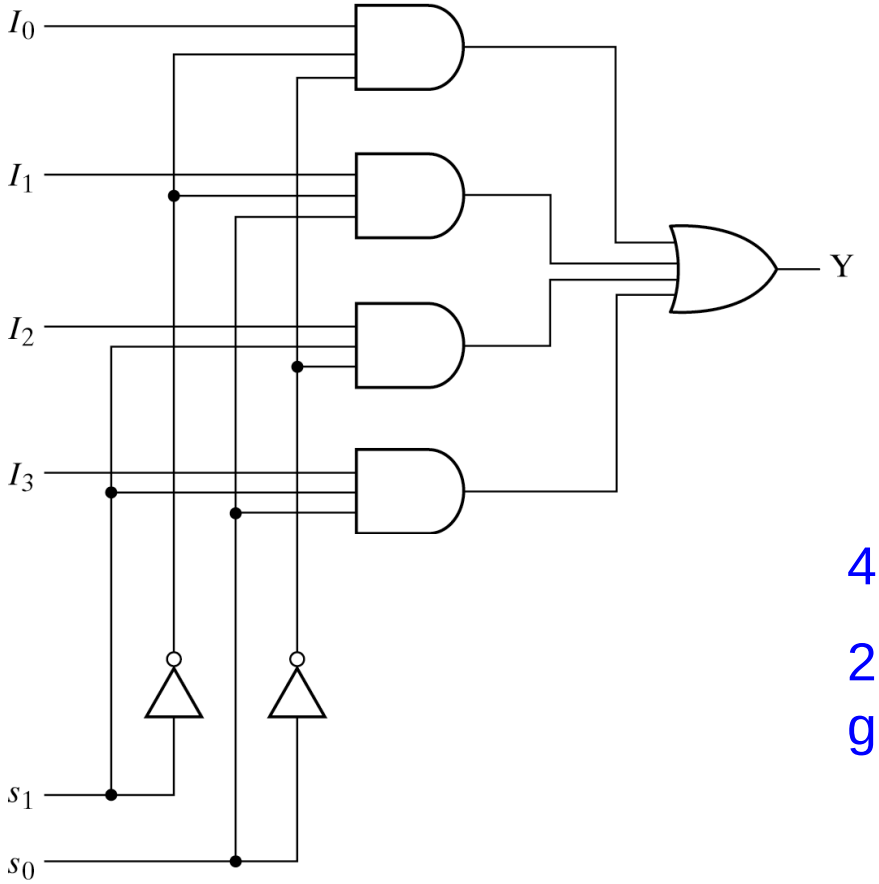


(a) Logic diagram



(b) Block diagram

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

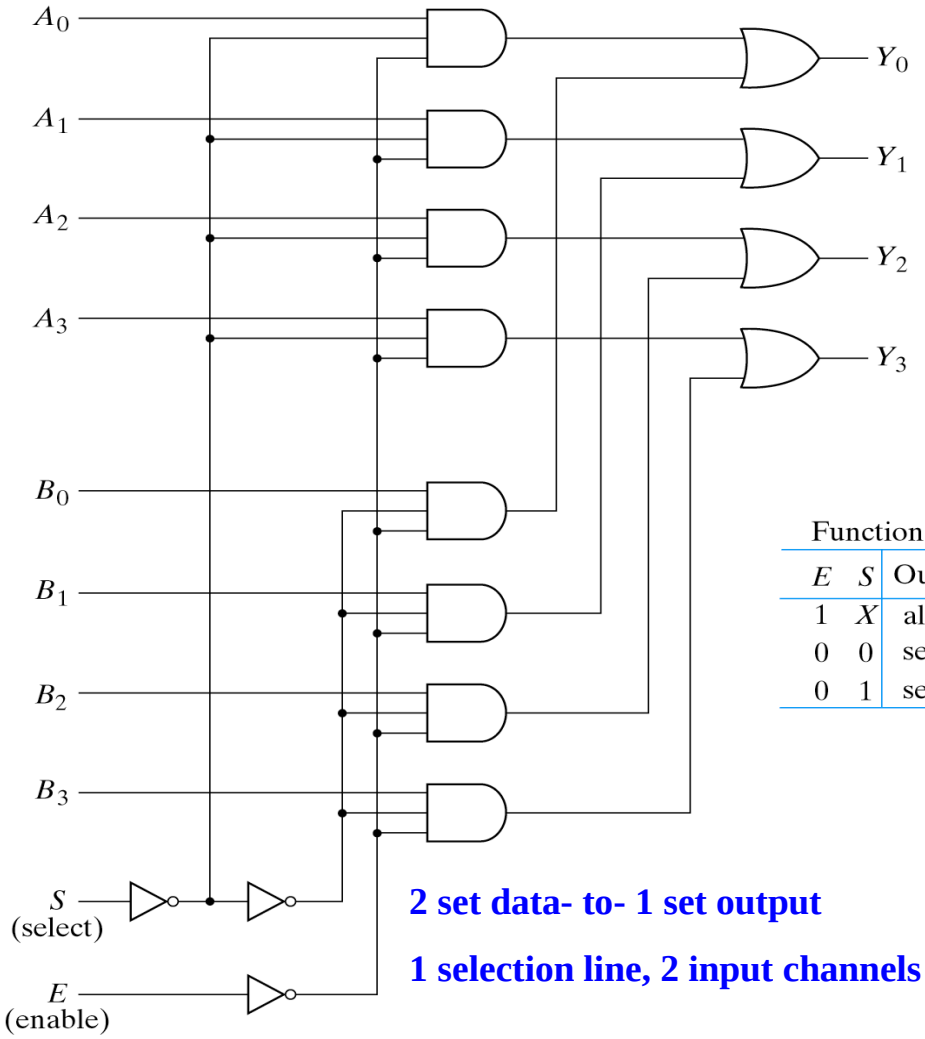
4- Input data lines

2-selection lines to select one AND gate and directs its data to output

Fig. 4-25 4-to-1-Line Multiplexer

Building Parallel Multiplexer

- A MUX can have an enable input line to control the flow of data so, if it is enabled, it will behaves as normal MUX, if not all outputs are zero.
- Multiplexer blocks can be combined in parallel with common selection and enable lines to perform selection on multi-bit quantities



Function table

E	S	Output Y
1	X	all 0's
0	0	select A
0	1	select B

Fig. 4-26 Quadruple 2-to-1-Line Multiplexer

Implementing Boolean Functions Using MUX

- The logic diagram of a MUX reveals that it is essentially a decoder that includes the OR gate with the unit.
- The minterms are generated by the selection line and the selection among the minterm is achieved by the data input lines
- Any Boolean function of *n*-variables can be implemented using a MUX with *n-1* selection lines
- (n-1) input variables of the function will be connected to the selection lines and the n-th (assume =Z) input variable is evaluated according to the value of the minterms of the function. The evaluated values are connected to the data lines, so each data input can be either Z, Z',0, or 1.

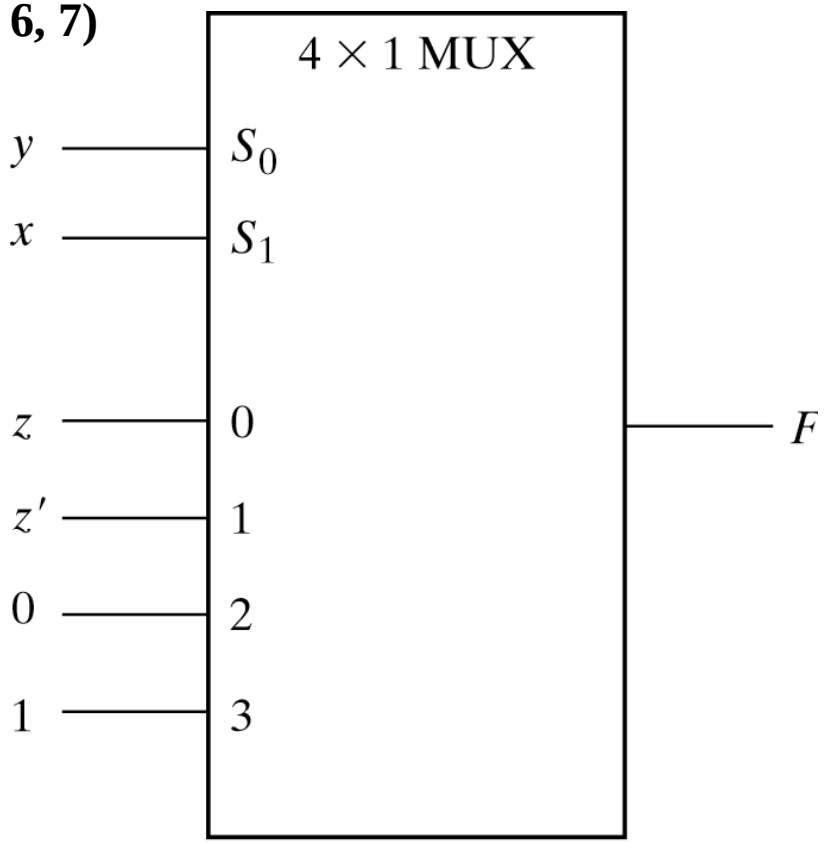
Example of Implementing Boolean Functions Using MUX

Implement the function F(x, y, z)=m (1, 2, 6, 7)

x , and y should be connected with the same order to S_1 and S_0 respectively

x	y	z	F	
0	0	0	0	$F = z$
0	0	1	1	
0	1	0	1	$F = z'$
0	1	1	0	
1	0	0	0	$F = 0$
1	0	1	0	
1	1	0	1	$F = 1$
1	1	1	1	

(a) Truth table



(b) Multiplexer implementation

Fig. 4-27 Implementing a Boolean Function with a Multiplexer

Example of Implementing Boolean Functions Using MUX

Implement the function F(A, B,C,D)=m(1, 3, 4, 11, 12, 13, 14, 15) with 8-to-1 MUX

A	B	C	D	F	
0	0	0	0	0	$F = D$
0	0	0	1	1	
0	0	1	0	0	$F = D$
0	0	1	1	1	
0	1	0	0	1	$F = D'$
0	1	0	1	0	
0	1	1	0	0	$F = 0$
0	1	1	1	0	
1	0	0	0	0	$F = 0$
1	0	0	1	0	
1	0	1	0	0	$F = D$
1	0	1	1	1	
1	1	0	0	1	$F = 1$
1	1	0	1	1	
1	1	1	0	1	$F = 1$
1	1	1	1	1	

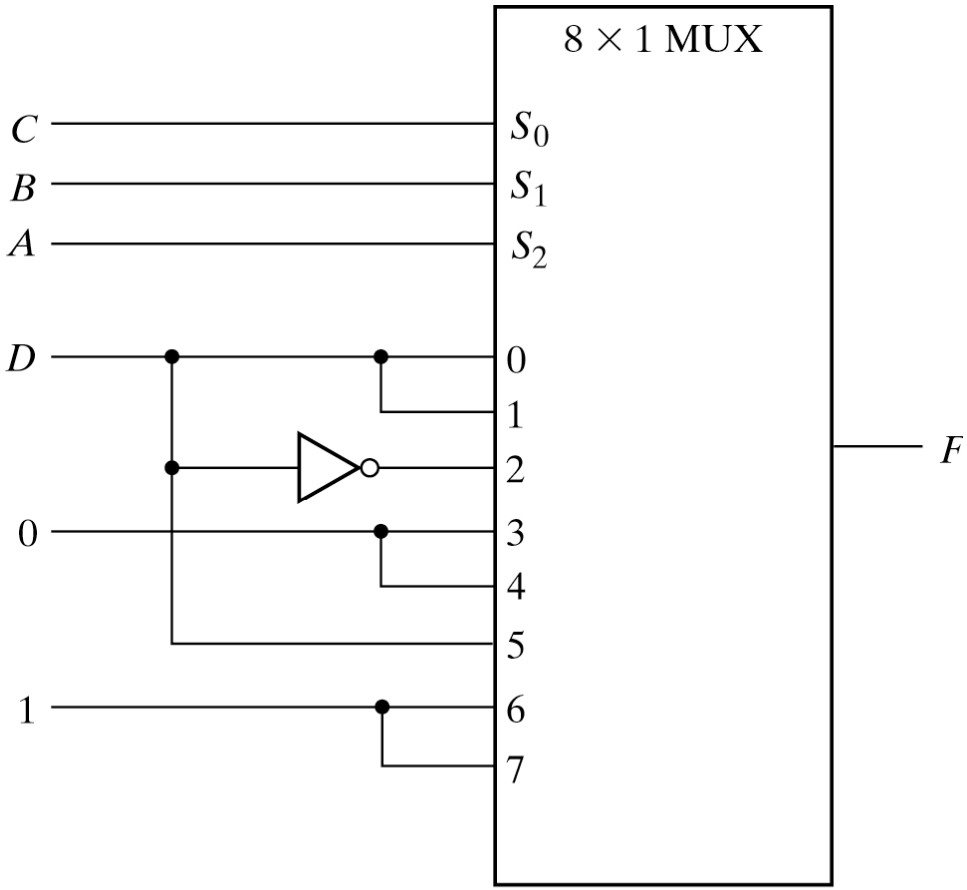
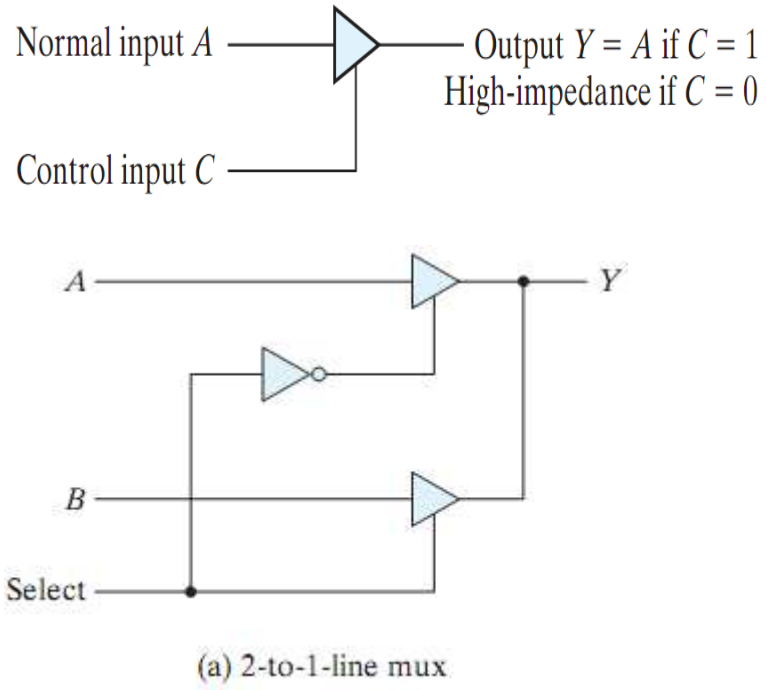


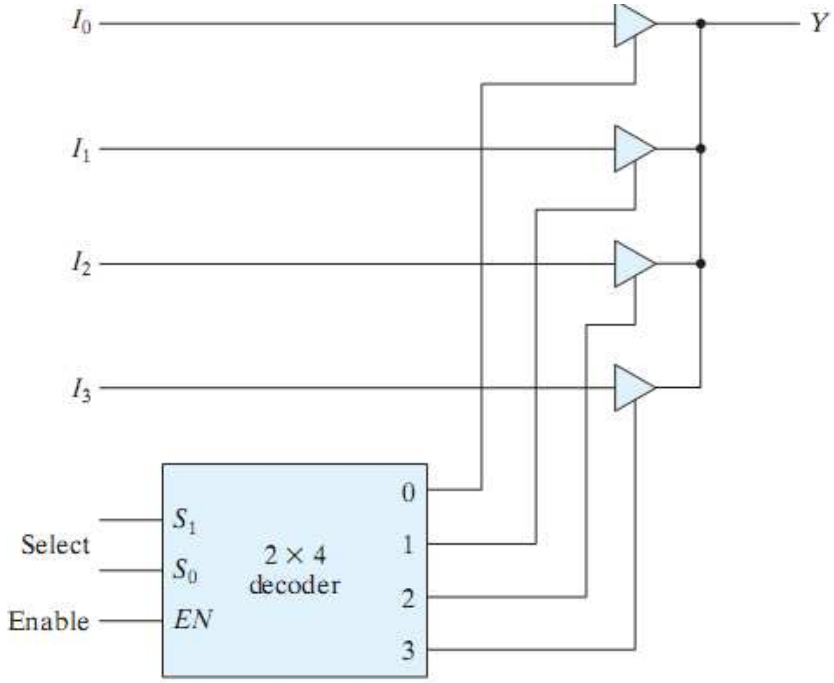
Fig. 4-28 Implementing a 4-Input Function with a Multiplexer

MUX with Three State Gates

Three state gates are digital circuits that exhibit three states. Two of the states are signals equivalent to logic 0 and logic 1 as in the conventional gate. The third state is a high impedance state, in which the logic behaves like an open circuit. It is possible to connect the output of three state gates to common line without causing loading effect.



(a) 2-to-1-line mux



(b) 4-to-1-line mux

Demultiplexers

- Performs the inverse operation of a multiplexer
- A combinational circuit that receives input from a single line and transmits it to one of 2^n possible output lines
- The selection of the specific output is controlled by the bit combination of n selection lines