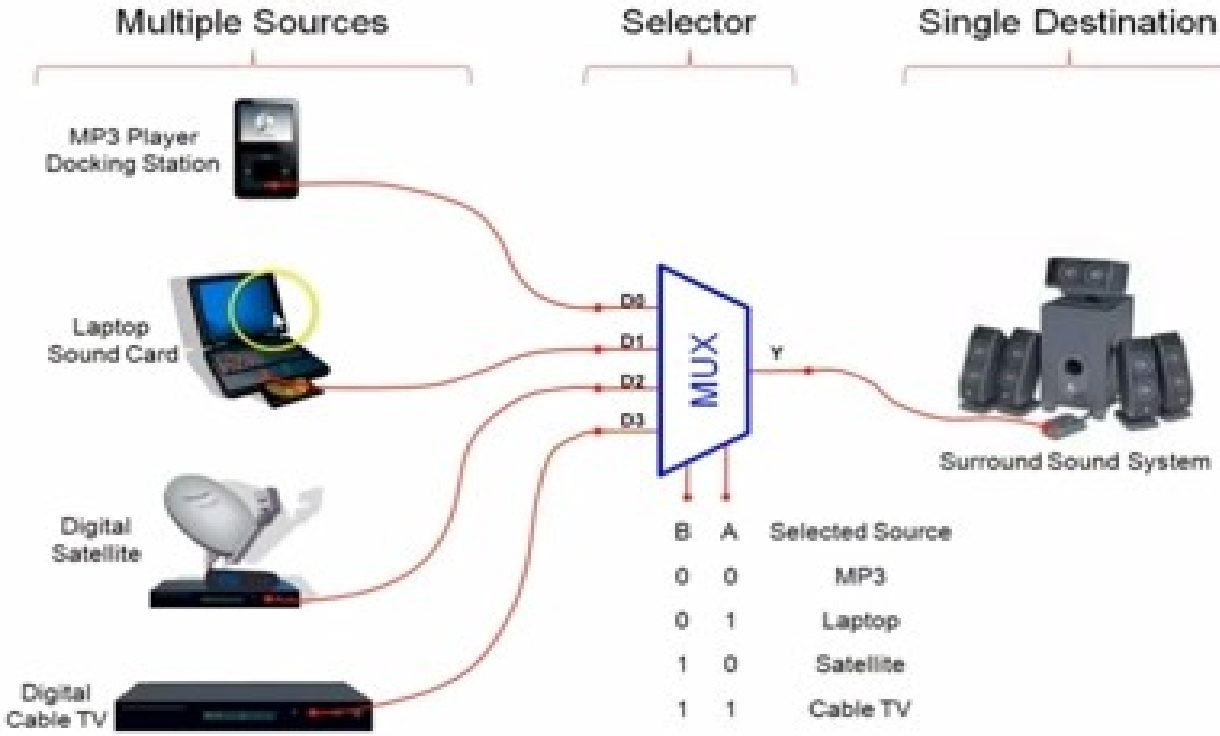
M U LT I P L E X E R S

A multiplexer is a combinational circuit that selects binary information from one of many input lines and directs it to a single output line.

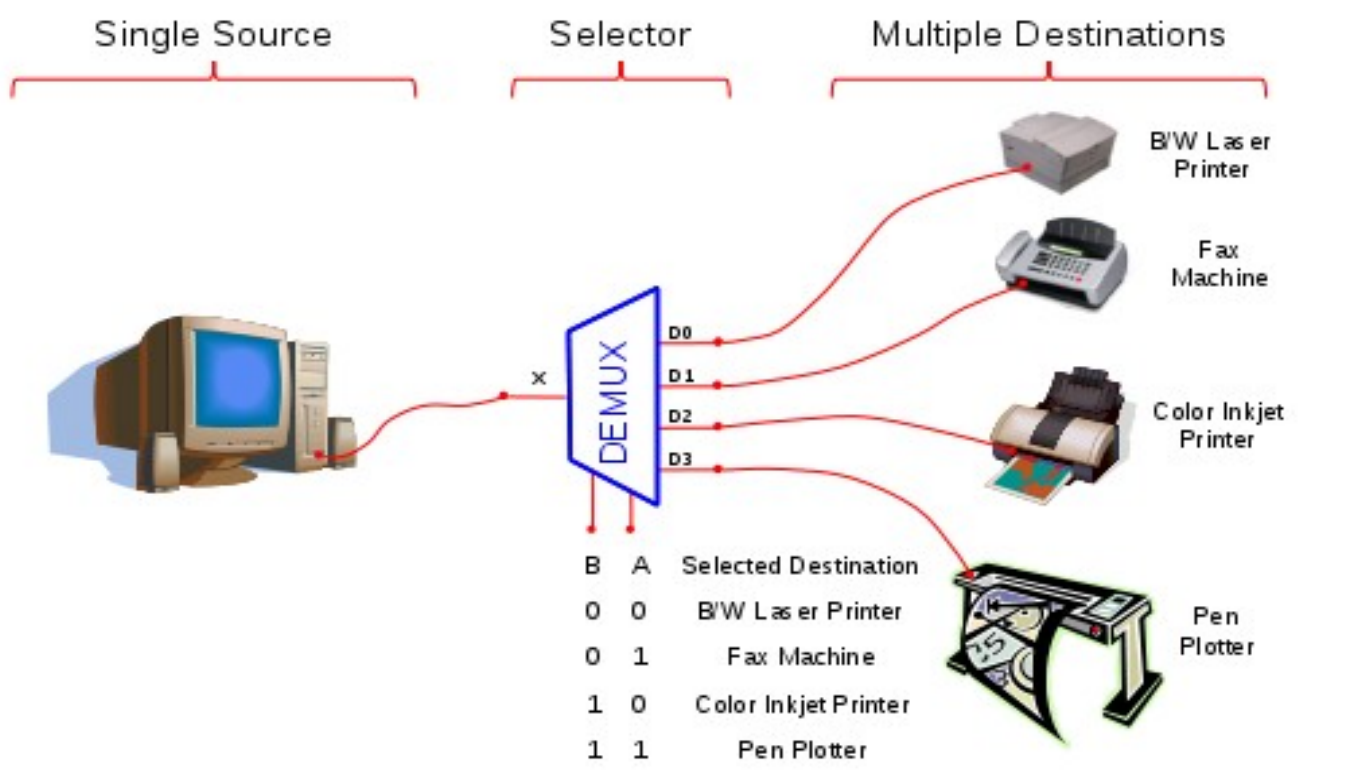
The selection of a particular input line is controlled by a set of selection lines.

Normally, there are 2^n input lines and n selection lines whose bit combinations determine which input is selected.

Typical Application of a MUX

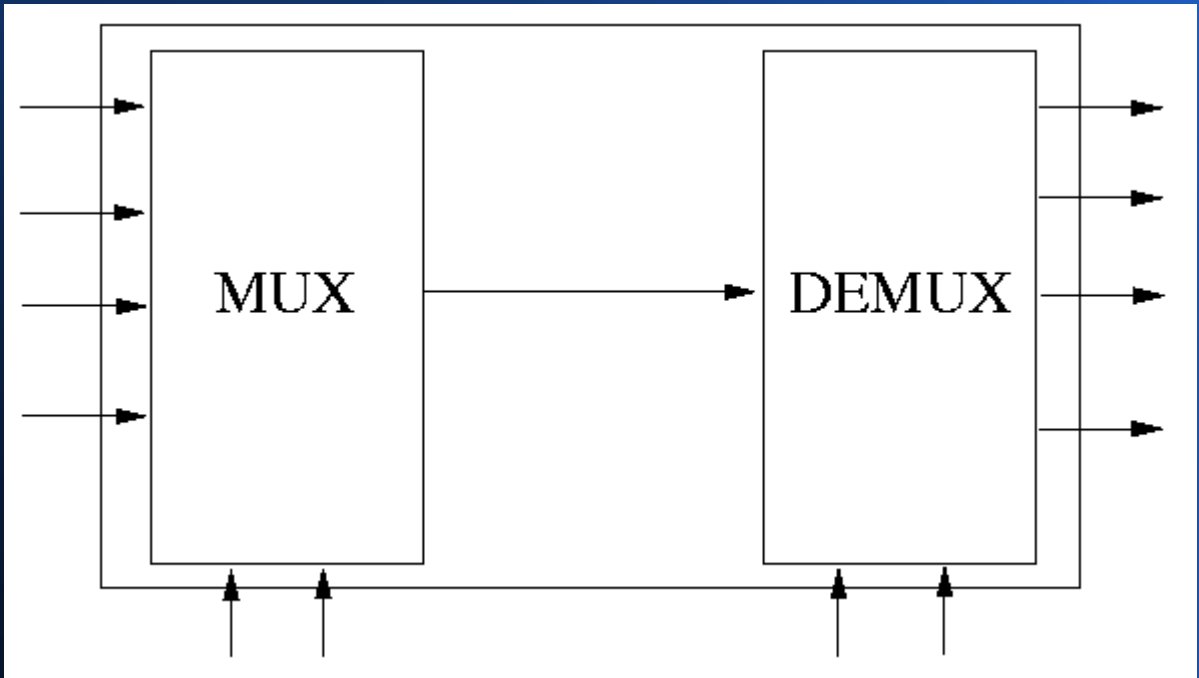


Typical Application of a DEMUX



One use for multiplexers is cost saving by connecting a multiplexer and a demultiplexer together over a single channel (by connecting the multiplexer's single output to the demultiplexer's single input).

In this case, the cost of implementing separate channels for each data source is higher than the cost and inconvenience of providing the multiplexing/demultiplexing functions.



Note that the MUX-DEMUX pair works only as a simple routing device. It doesn't manipulate the data signal that is being sent through it.

At any given instant of time, only one input channel can send data to only one output channel.

A MUX-DEMUX pair is different from an ENCODER-DECODER pair in that the latter modulates the data signal so that an "n" bit data is sent through a channel with bits less than "n" and then on reaching the receiver side the decoder once again demodulates the data signal into an "n" bit signal

The MUX-DEMUX pair just sends the data as it is without manipulating any of its parameters.

Time-division multiplexing (TDM) is a type of digital (or rarely analog) multiplexing in which two or more bit streams or signals are transferred apparently simultaneously as sub-channels in one communication channel, but are physically taking turns on the channel.

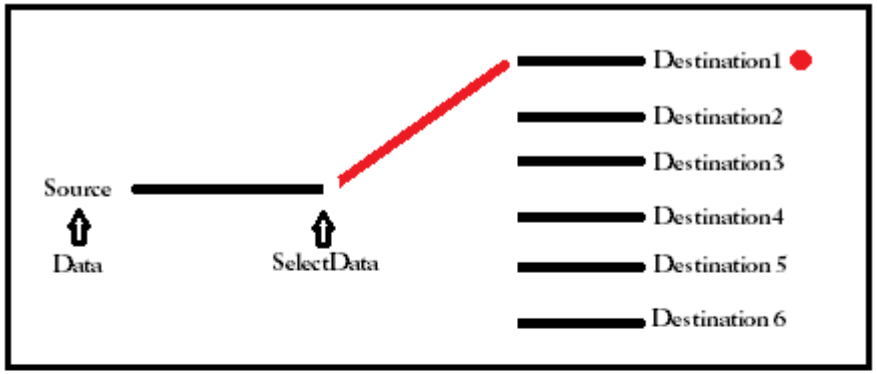


Figure1

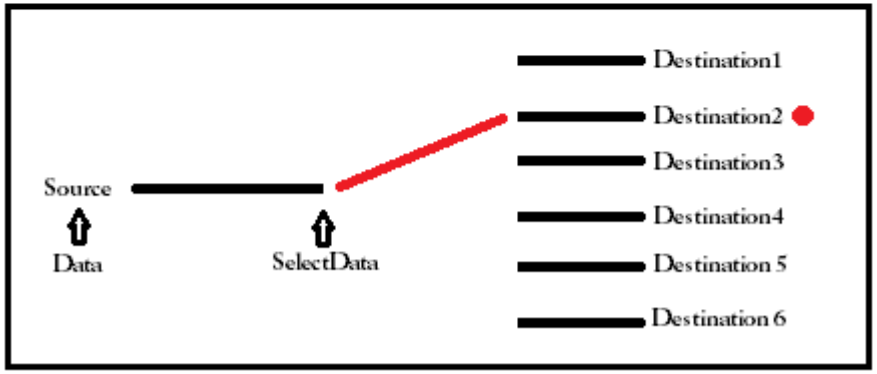
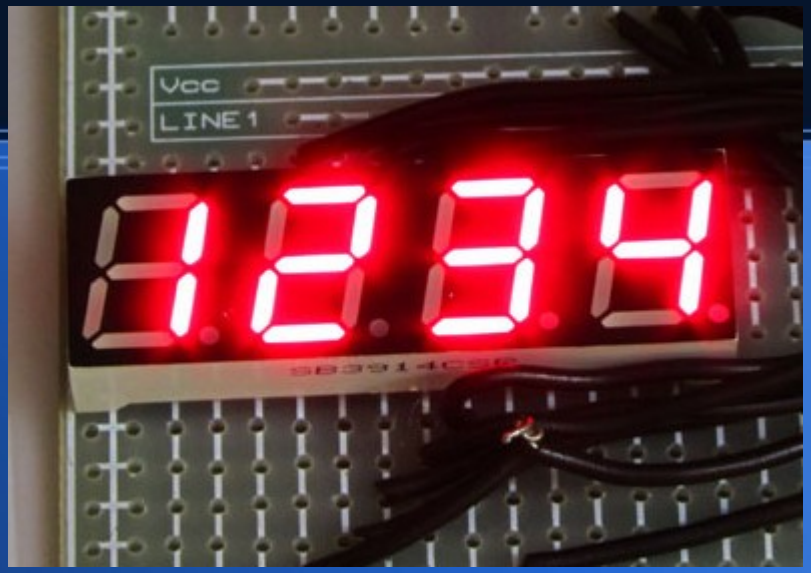


Figure2

7 Segment Display Multiplexing Technique

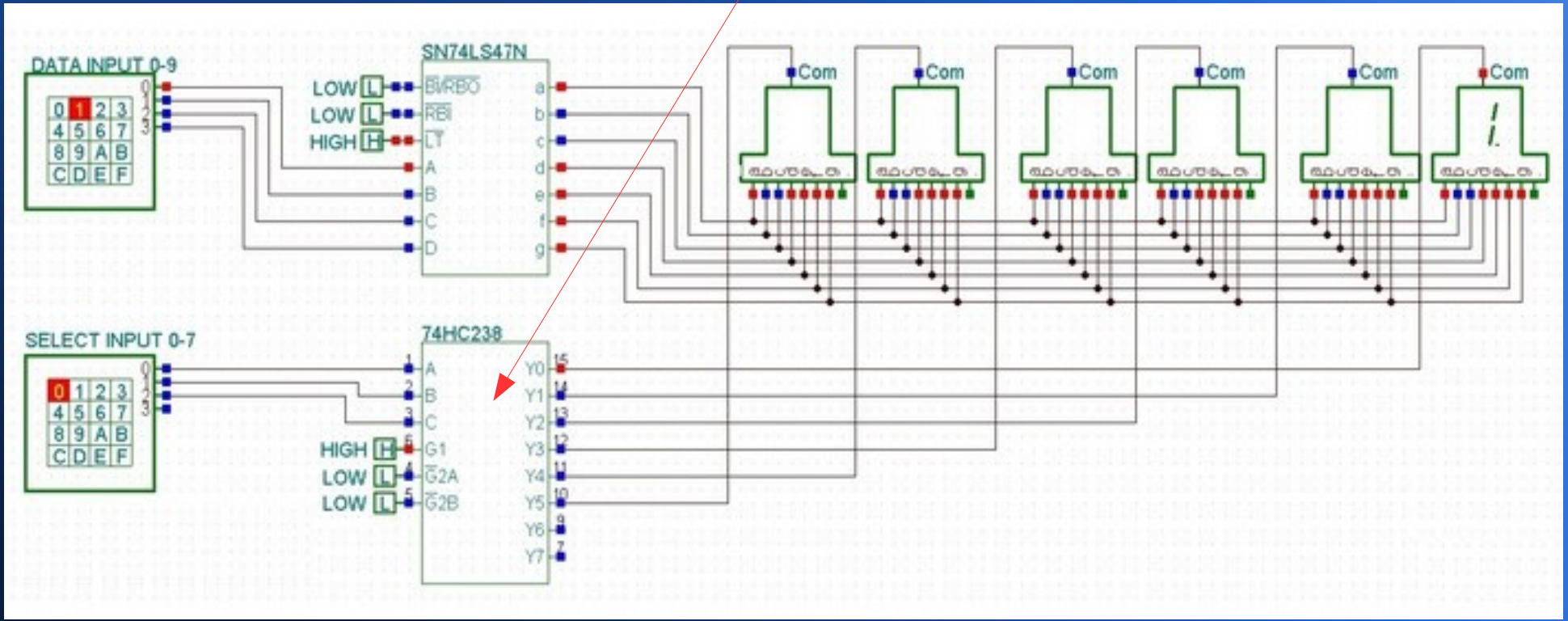


Multiplexing is used to save I/O pins

Each display is turned ON at a rate above 100 times per second and it will appear that all the displays are turned ON at the same time due to POV(Persistence of vision).

As each display is turned ON, the appropriate information must be delivered to it so that it will give the correct reading.

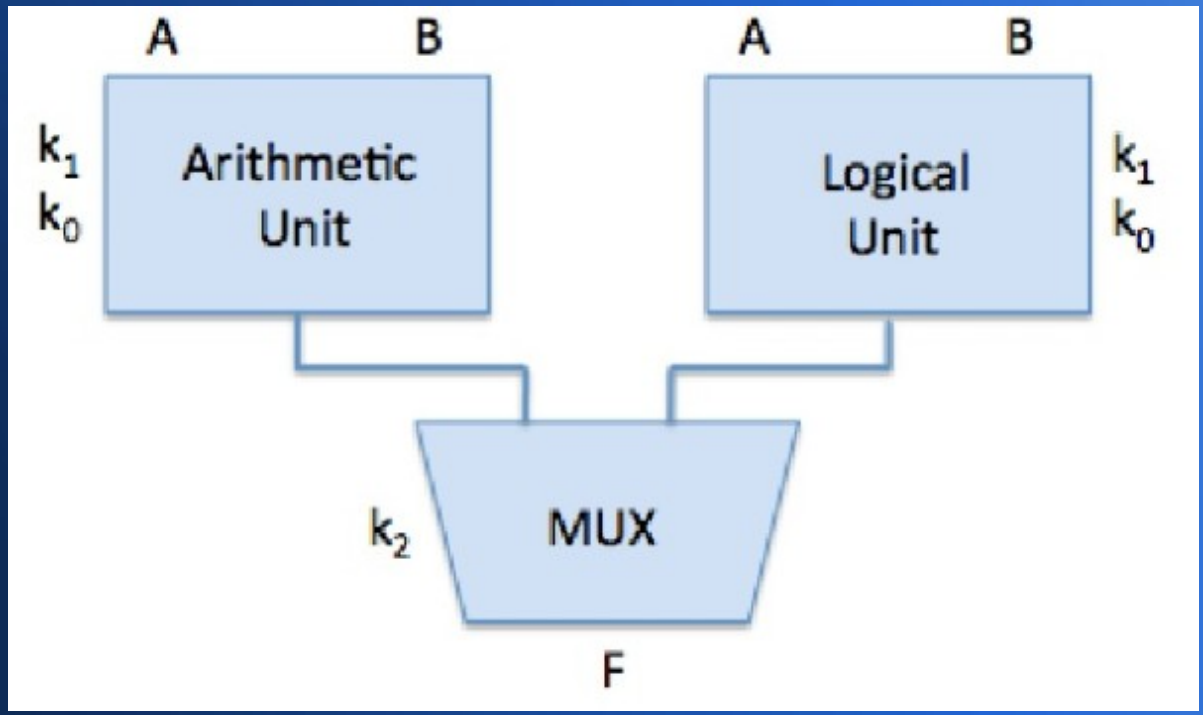
74HC238 3 to 8 line Decoder/Demultiplexer



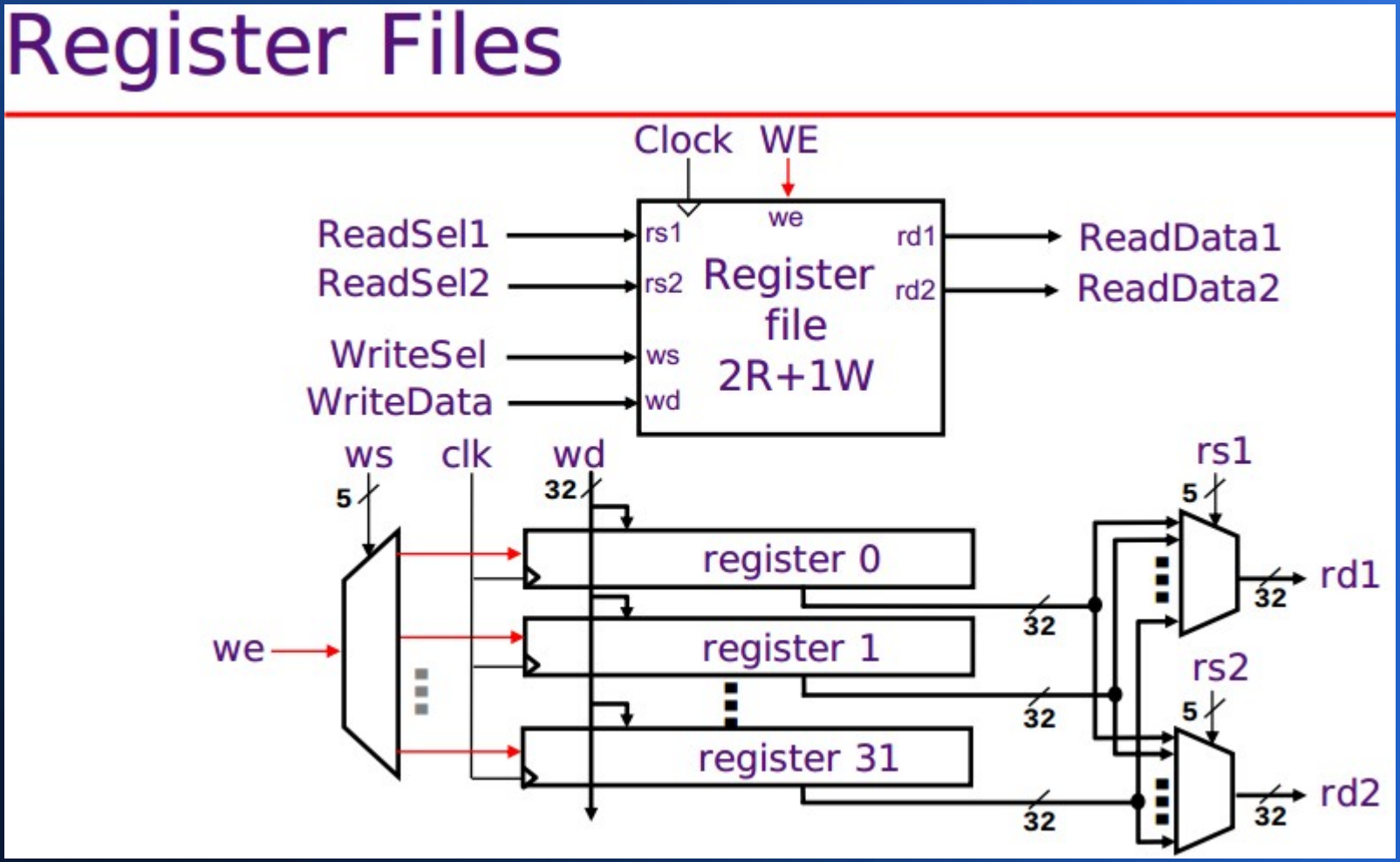
A n-bit ALU is a component which performs an operation on two n-bit inputs, A (an-1...a1 a0) and B (bn-1...b1 b0), and produces an n-bit output F.

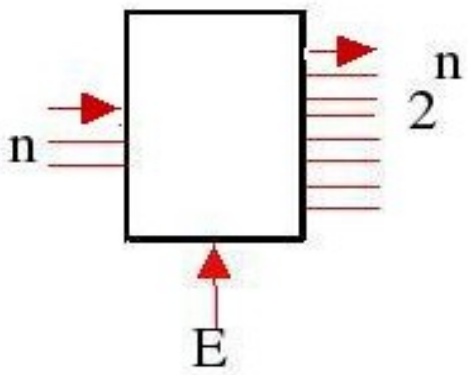
The ALU also accepts control inputs which specify which operation we should perform on A and B. We will have three control inputs, k2, k1, and k0, allowing us to encode and select from eight (four arithmetic and four logical) operations. The table below shows a typical set of operations that could be implemented by the ALU

k_1k_0	F, when $k_2 = 0$	F, when $k_2 = 1$
0 0	A PLUS B	A AND B
0 1	A MINUS B	A OR B
1 0	A PLUS 1	A'
1 1	-B	A XOR B

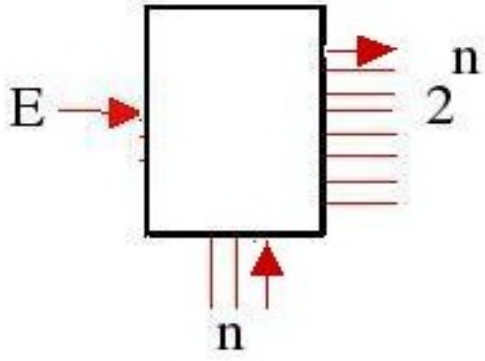


This register file makes it possible to simultaneously read from two registers and write into one register.

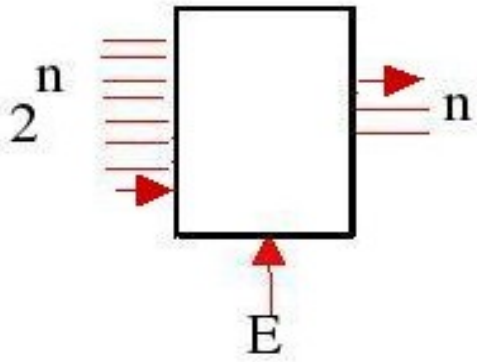




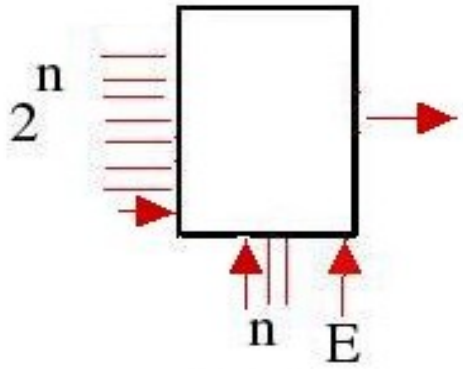
decoder



demultiplexer



encoder



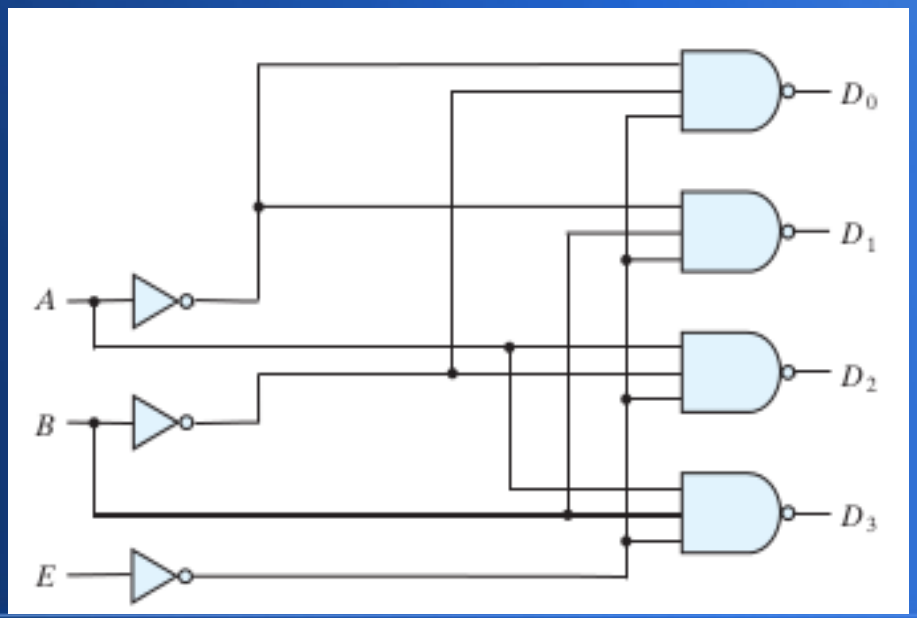
multiplexer

Two-to-four-line decoder with enable input

Some decoders are constructed with NAND gates. Since a NAND gate produces the AND operation with an inverted output, it becomes more economical to generate the decoder minterms in their complemented form.

Furthermore, decoders include one or more enable inputs to control the circuit operation.

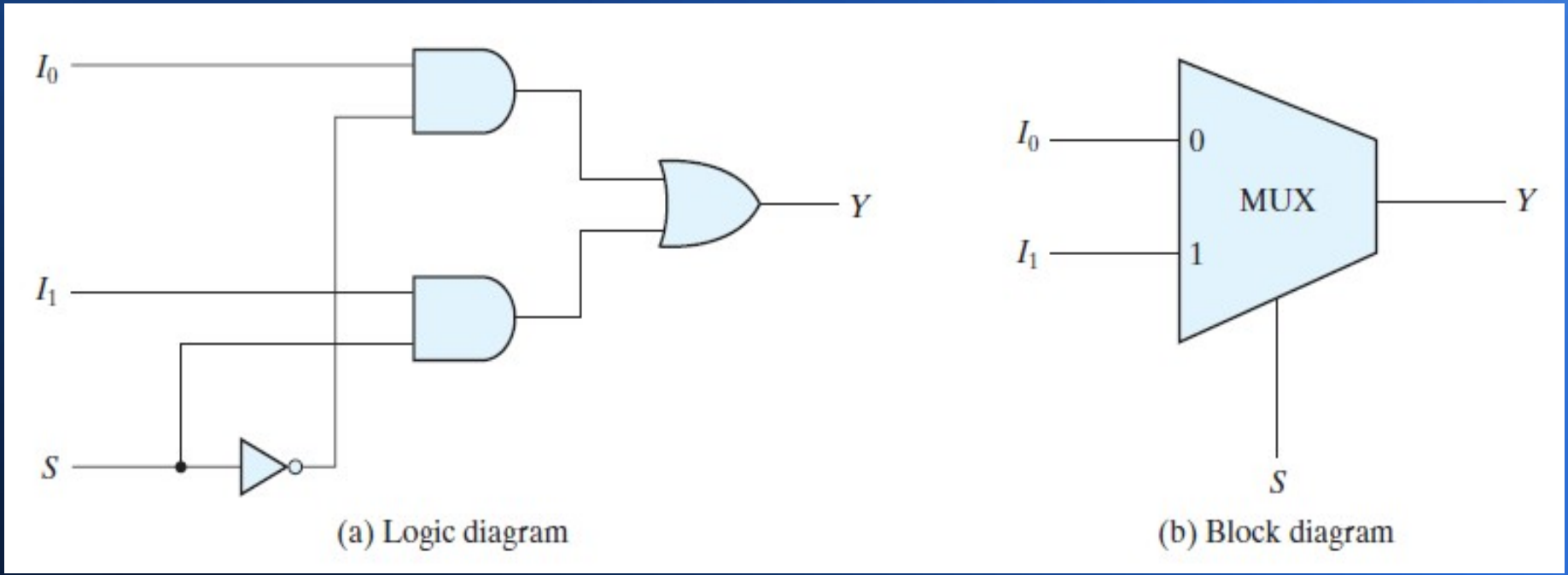
E	A	B	D_0	D_1	D_2	D_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0



An electronic multiplexer can be considered as a multiple-input, single-output switch, and a demultiplexer as a single-input, multiple-output switch.

The schematic symbol for a multiplexer is an isosceles trapezoid with the longer parallel side containing the input pins and the short parallel side containing the output pin

Two-to-one-line multiplexer



A two-to-one-line multiplexer connects one of two 1-bit sources to a common destination.

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 I0 has a path to the output.

When S = 1, the lower AND gate is enabled and I1 has a path to the output.

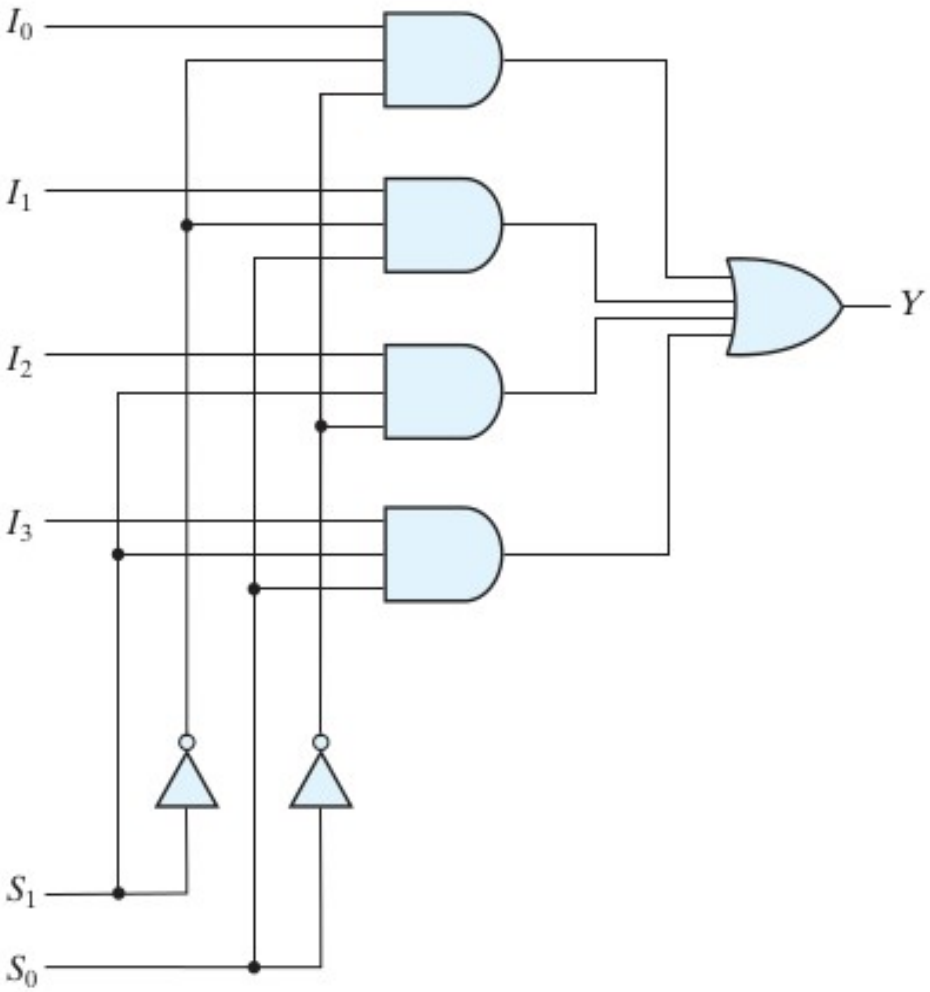
The multiplexer acts like an electronic switch that selects one of two sources.

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

It suggests visually how a selected one of multiple data sources is directed into a single destination.

The multiplexer is often labeled “MUX” in block diagrams

Four-to-one-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

The AND gates and inverters in the multiplexer resemble a decoder circuit, and indeed, they decode the selection input lines.

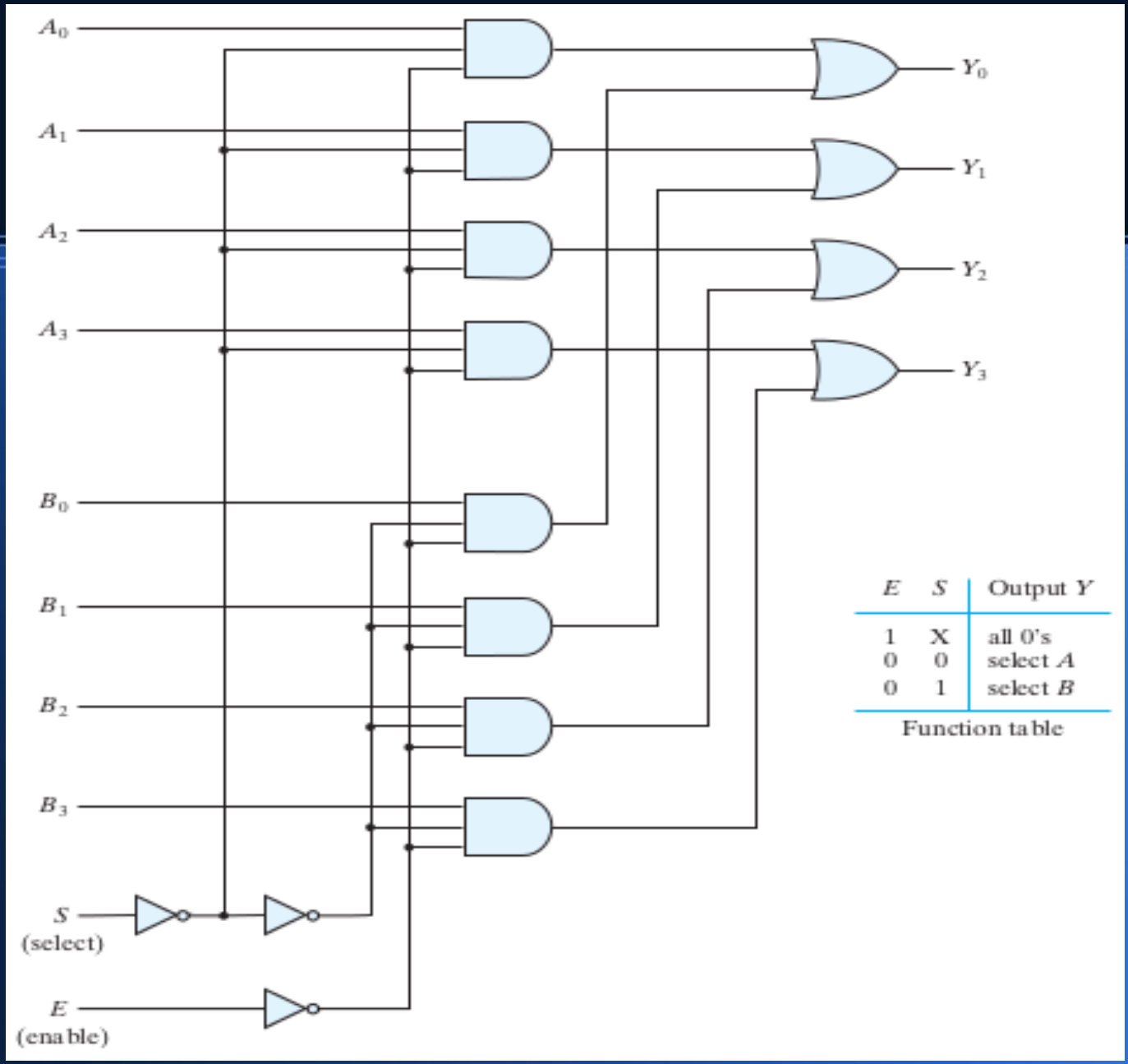
In general, a 2^n-to-1-line multiplexer is constructed from an n-to-2^n decoder by adding 2^n input lines to it, one to each AND gate.

The outputs of the AND gates are applied to a single OR gate.

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.

Quadruple two-to-one-line multiplexer



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

Function table

Multiplexer circuits can be combined with common selection inputs to provide multiple-bit selection logic. Although the circuit contains four 2-to-1-line multiplexers, we are more likely to view it as a circuit that selects one of two 4-bit sets of data lines

The circuit has four multiplexers, each capable of selecting one of two input lines.

Output Y0 can be selected to come from either input A0 or input B0.

Similarly,output Y1 may have the value of A1 or B1, and so on.

Input selection line S selects one of the lines in each of the four multiplexers.

The enable input E must be active (i.e., asserted) for normal operation.

As shown in the function table, the unit is enabled when E = 0.

Then, if S = 0, the four A inputs have a path to the four outputs.

If, by contrast, S = 1, the four B inputs are applied to the outputs.

The outputs have all 0’s when E = 1, regardless of the value of S

Boolean Function Implementation

We have shown that a decoder can be used to implement Boolean functions by employing external OR gates.

An examination of the logic diagram of a multiplexer reveals that it is essentially a decoder that includes the OR gate within the unit.

Minterms of a function are generated in a multiplexer by the circuit associated with the selection inputs.

The individual minterms can be selected by the data inputs, Thereby providing a method of implementing a Boolean function of n variables with a multiplexer that has n selection inputs and 2^n data inputs, one for each minterm

A more efficient method for implementing a Boolean function of n variables with a multiplexer that has n - 1 selection inputs.

The first n - 1 variables of the function are connected to the selection inputs of the multiplexer.

The remaining single variable of the function is used for the data inputs.

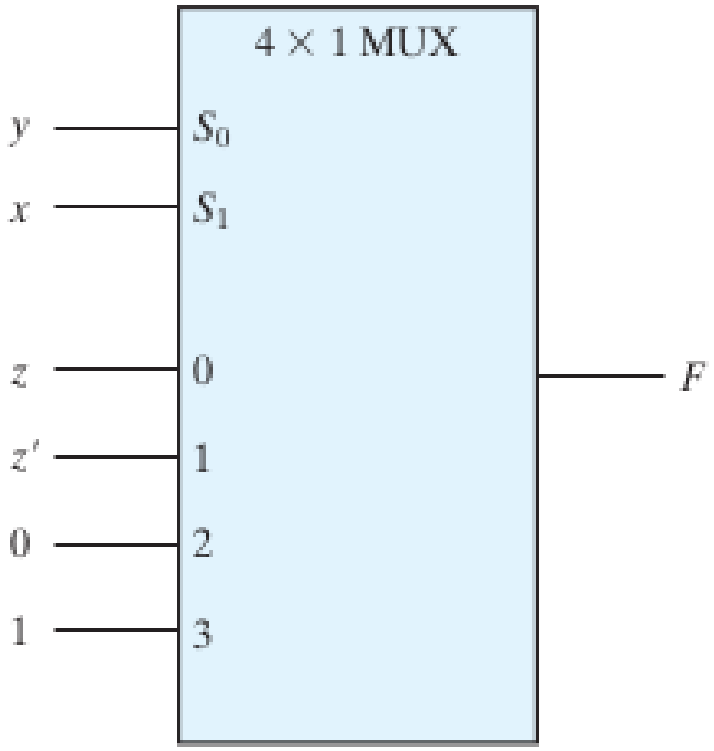
If the single variable is denoted by z, each data input of the multiplexer will be z, z', 1, or 0

Implementing a Boolean function with a multiplexer

$F(x, y, z) = \Sigma(1, 2, 6, 7)$

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

The two variables x and y are applied to the selection lines in that order; x is connected to the S1 input and y to the S0 input.

The values for the data input lines are determined from the truth table of the function.

When xy = 00, output F is equal to z because F = 0 when z = 0 and F = 1 when z = 1.

This requires that variable z be applied to data input 0.

The operation of the multiplexer is such that when xy = 00, data input 0 has a path to the output, and that makes F equal to z.

In a similar fashion, we can determine the required input to data lines 1, 2, and 3 from the value of F when xy = 01, 10, and 11, respectively

The general procedure for implementing any Boolean function of n variables with a multiplexer with n - 1 selection inputs and 2^(n - 1) data inputs follows from the previous example.

To begin with, Boolean function is listed in a truth table. Then first n - 1 variables in the table are applied to the selection inputs of the multiplexer.

For each combination of the selection variables, we evaluate the output as a function of the last variable.

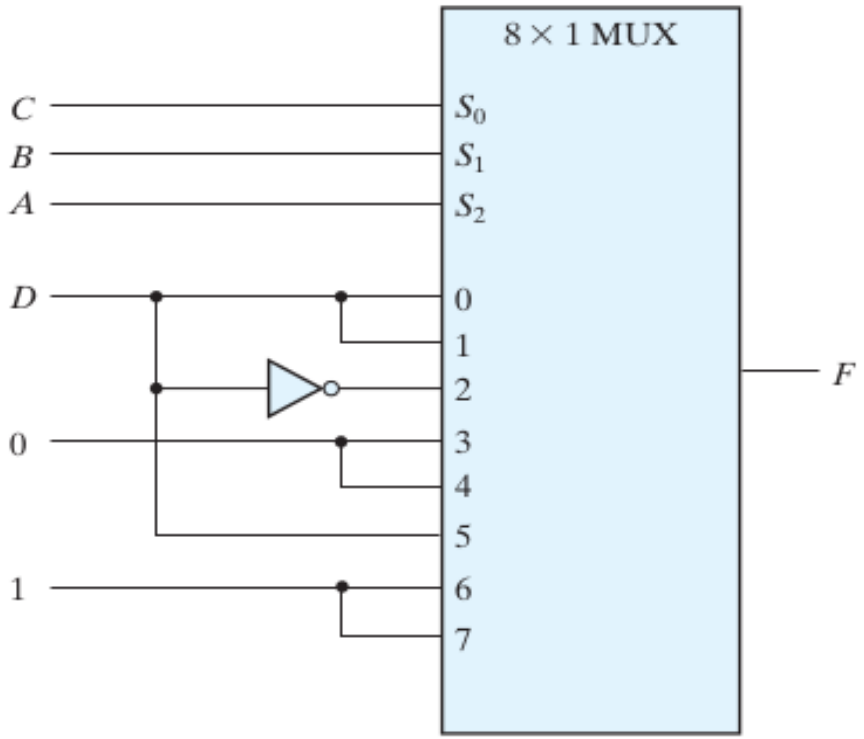
This function can be 0, 1, the variable, or the complement of the variable.

These values are then applied to the data inputs in the proper order

Implementing a four-input function with a multiplexer

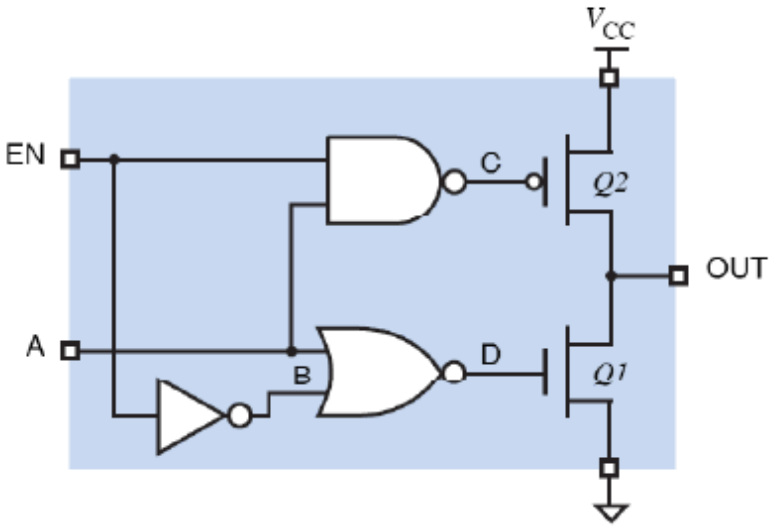
$$F(A, B, C, D) = \Sigma(1, 3, 4, 11, 12, 13, 14, 15)$$

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	

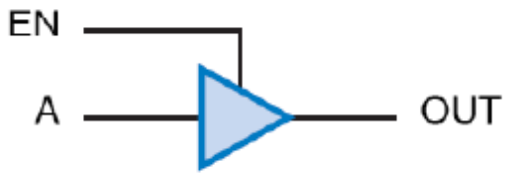


Three-state gates

Three-state gates may perform any conventional logic, such as AND or NAND. However, the one most commonly used is the buffer gate



EN	A	B	C	D	*Q1*	*Q2*	OUT
L	L	H	H	L	off	off	Hi-Z
L	H	H	H	L	off	off	Hi-Z
H	L	L	H	H	on	off	L
H	H	L	L	L	off	on	H



Three-state gates

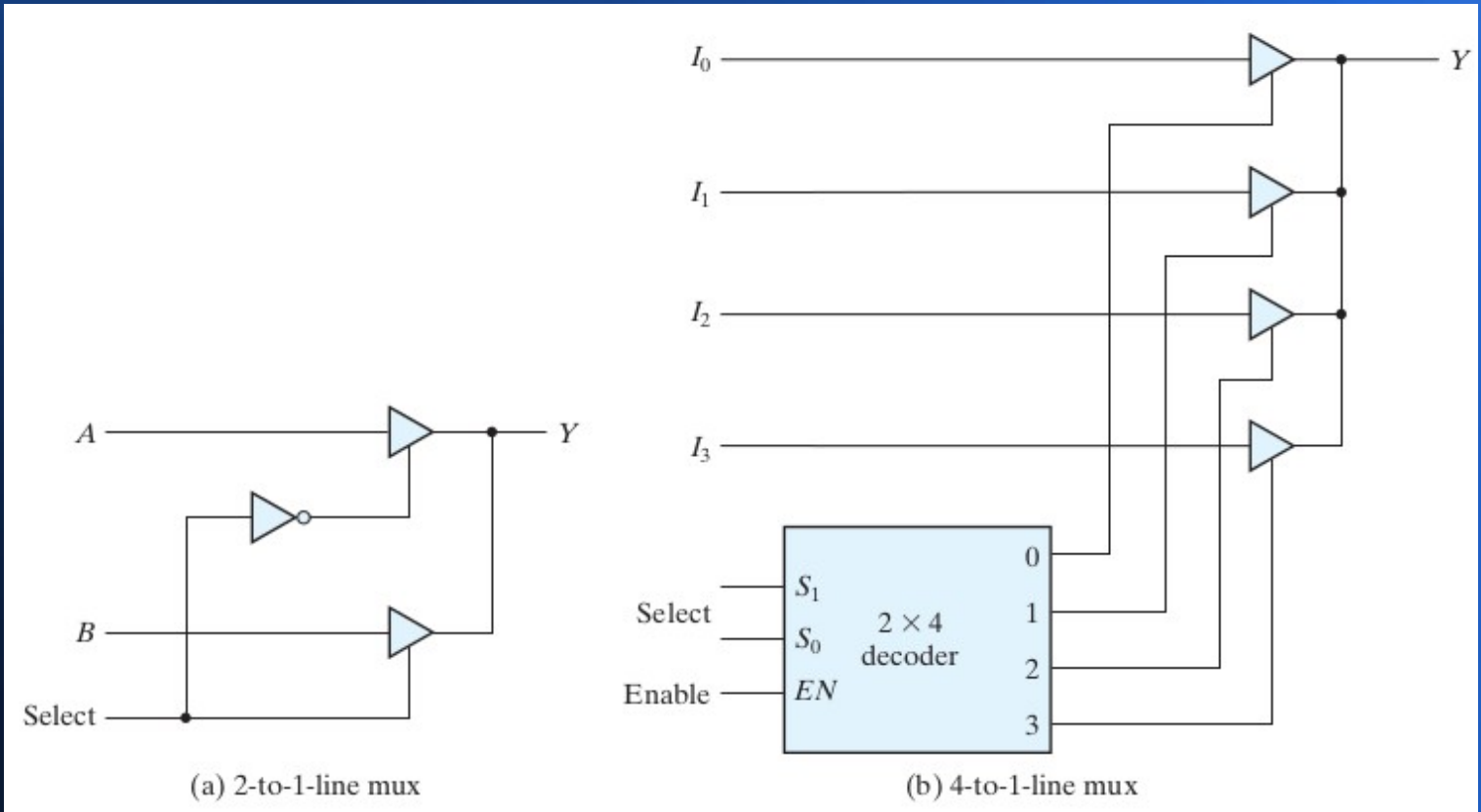
Two of the states are signals equivalent to logic 1 and logic 0 as in a conventional gate.

The third state is a high-impedance state in which
(1) the logic behaves like an open circuit, which means that the output appears to be disconnected,
(2) the circuit has no logic significance, and
(3) the circuit connected to the output of the three-state gate is not affected by the inputs to the gate.

If more than one device is electrically connected, putting an output into the Hi-Z state is often used to prevent short circuits, or one device driving high (logical 1) against another device driving low (logical 0).

Three-state buffers can also be used to implement efficient multiplexers, especially those with large numbers of inputs.

Multiplexers with three-state gates



The construction of multiplexers with three-state buffers.

The construction of a two-to-one-line multiplexer with 2 three-state buffers and an inverter:-

The two outputs are connected together to form a single output line. (Note that this type of connection cannot be made with gates that do not have three-state outputs.)

When the select input is 0, the upper buffer is enabled by its control input and the lower buffer is disabled.

Output Y is then equal to input A. When the select input is 1, the lower buffer is enabled and Y is equal to B

The construction of a four-to-one-line multiplexer

The outputs of 4 three-state buffers are connected together to form a single output line.

The control inputs to the buffers determine which one of the four normal inputs I0 through I3 will be connected to the output line.

No more than one buffer may be in the active state at any given time.

The connected buffers must be controlled so that only 1 three state buffer has access to the output while all other buffers are maintained in a high impedance state using the decoder.