Module -8

Multiplexers and Demultiplexers

Learning objectives

1. Get familiar with principles of Multiplexing and Demultiplexing
2. Study the logic diagram and truth tables of multiplexers
3. Understand the logic diagram and truth tables of demultiplexers
4. Explore the applications of multiplexers and demultiplexers

1. Introduction

There are many applications where many devices or system needs to access a common transmission media. Let us consider our cable TV system, in which multiple TV channels are received at home via a shared transmission media. Similarly, in computer networking, many computer needs to access a common printer or other shared hardware. In Smart instrumentation, a PC or microcontroller can acquire data gathered from many sensors through a single ADC. Our telephone exchanges utilize these multiplexers and demultiplexers extensively.

Multiplexer and demultiplexers are normally used for sharing resources. Multiplexers routes the data from many sources to one destination and demultiplexer redistributes data back from one source to many destinations.

In this module you will learn about the basic architecture of multiplexer and demultiplexers along with real life applications.

2. Principles of Multiplexing and Demultiplexing

Let us now begin with principles of multiplexing and demultiplexing. Let us consider a system with n-inputs. Multiplexing means sharing. It means many to one. A multiplexers (MUX) is a device that allows digital information from several inputs to be routed onto a single line for transmission as shown in Fig. 1.

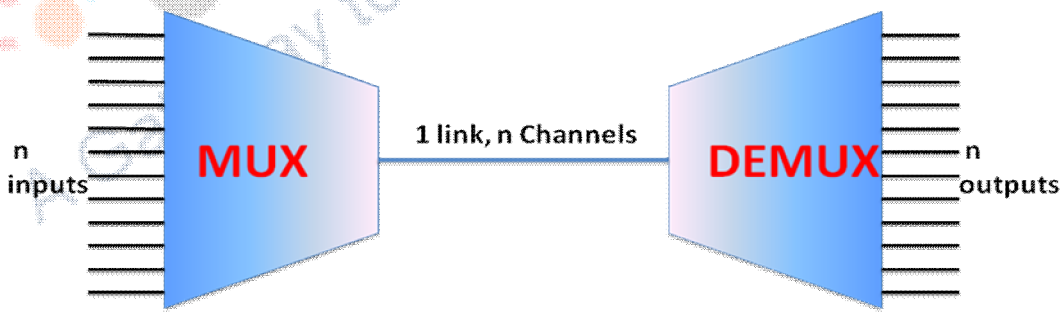


Figure 1: Principle of multiplexing and Demultiplexing

The multiplexed data is transmitted over a single line to a common destination. Demultiplexer accepts this data and redistributes it to the n outputs. Data select lines are used in many into one operation of multiplexer and one to many operation of demultiplexer. In short, multiplexers and

demultiplexers are combinational circuits designed to provide sharing of resources. The keyword is **sharing**.

3. Multiplexer

Multiplexer means many to one. A multiplexer (MUX) is a combinational circuit which is often used when the information from many sources must be transmitted over long distances and it is less expensive to multiplex data onto a single wire for transmission.

Multiplexer can be considered as multi-position or rotary switch as shown in fig. 2. There are n ï inputs and one output. The switch position is controlled by the selector lines. The select inputs decide which input is connected to the output.

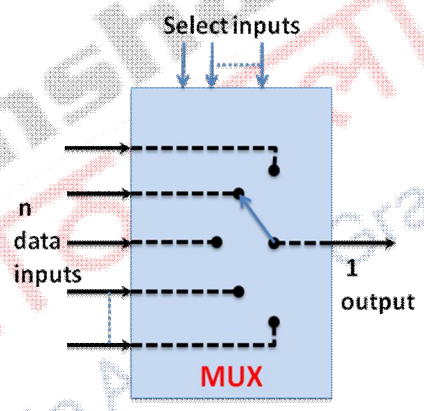


Figure 2: Multiplexer as multi-position or rotary switch

A multiplexer can be considered as semiconductor equivalent of multi-position switch. It is combinational logic circuit which has many input lines and a single output line. The basic operation is controlled by a selector lines that routes one of many input signals to the output. Fig.3 shows the logic symbol of general symbol of multiplexer.

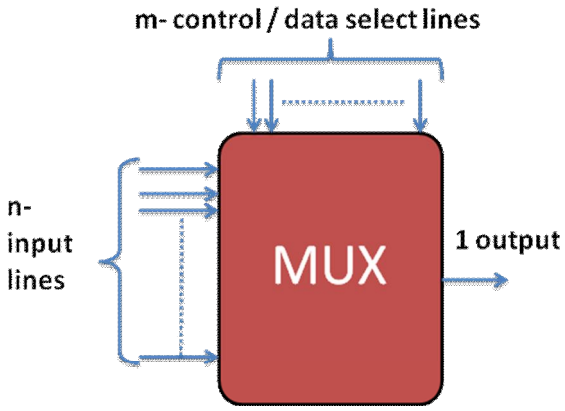


Figure 3: Logic Symbol for multiplexer

Multiplexer are also called as DATA Selector or router because it accepts several data inputs and allows only one of them to get through to the output at a time. The basic multiplexer has n input lines and single output line. It also has m i select or control lines. The relation between number of select lines and number of data inputs are

$$2^m = n$$

As multiplexer selects one out of many, it is often called as 2^m to 1 line converter.

3.1 Types of multiplexer

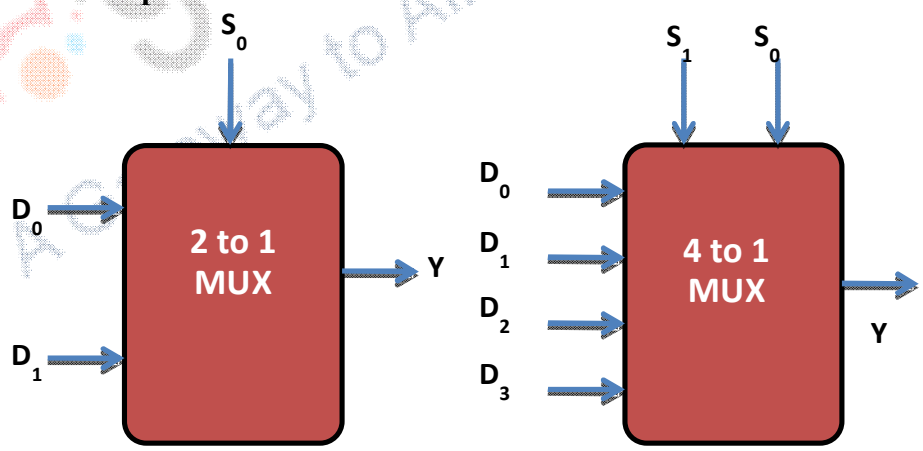


Figure-4(a) : Logic symbols of 2 to 1 and 4 to 1 multiplexers

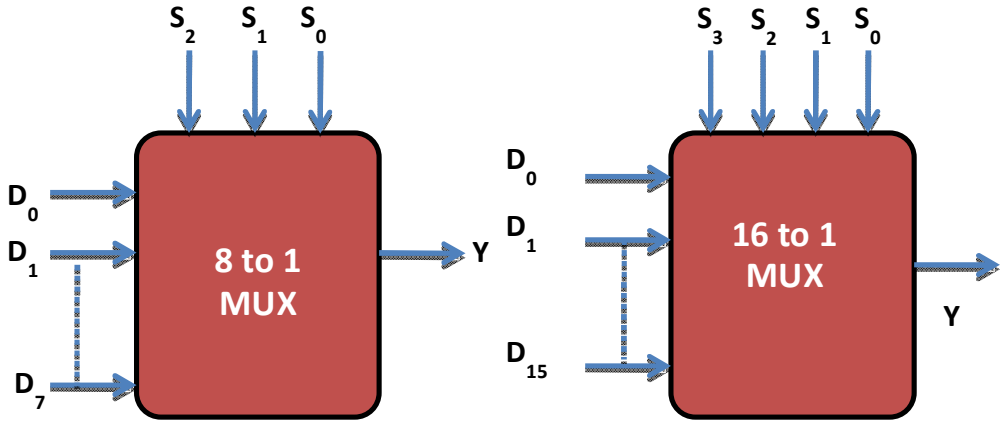
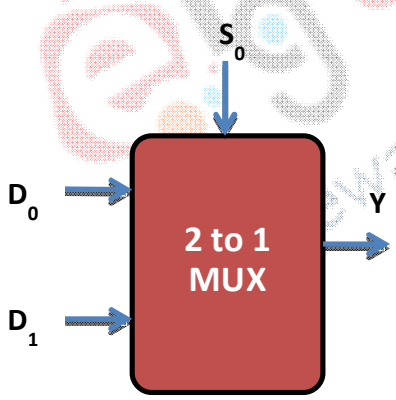


Figure 4(b) : Logic symbols of 8 to 1 and 16 to 1 multiplexers

Similarly we can extend the idea to 8 to 1 multiplexer and 16 to 1 multiplexer as shown in Fig. 4(b). For example 8 to 1 multiplexer with 8 inputs namely D0, D1,é D7, 3 select line S2, S1, So as the select line and Y as the single output. Similarly, the 16 to 1 multiplexer has 16 inputs D0, D1, é D15 , 4 select input S3,S2, S1 and S0 and Y as the output.

3.2 **A 2 to 1 multiplexer**



Select input S_0	Output Y
0	D_0
1	D_1

S_0	D_1	D_0	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Figure 5: Logic symbol, function table and Truth table of 2 to 1 Multiplexer

Figure 5 indicates the logic symbol, function table and truth table of 2 to 1 multiplexer. In this multiplexer, D0 and D1 are the data inputs, S0 is the select line and Y is the output of the

multiplexer. In a function table, Select line S0 is shown as the input and Y is the output. When S0 = 0 then the data D0 appears at output Y and when S0=1, the output Y receives the data D1.

Let us now prepare a detailed truth table for 2 to 1 mux. In this truth table S0, D1, D0 are the inputs and Y is the output. For S0=0, we have 4 possible combinations for the inputs 00,01,10 and 11. It is observed that Y always follows D0. For S0=1, we have similar 4 possible combinations for the inputs 00,01,10 and 11. It is observed that Y always follows D1.

The k-map required for this multiplexer is of three variables. Let us consider two rows for S0 and 4 columns for the data D1D0. Map the truth table into k-map by writing ó1ô to the appropriate minterm as shown in fig. 6.

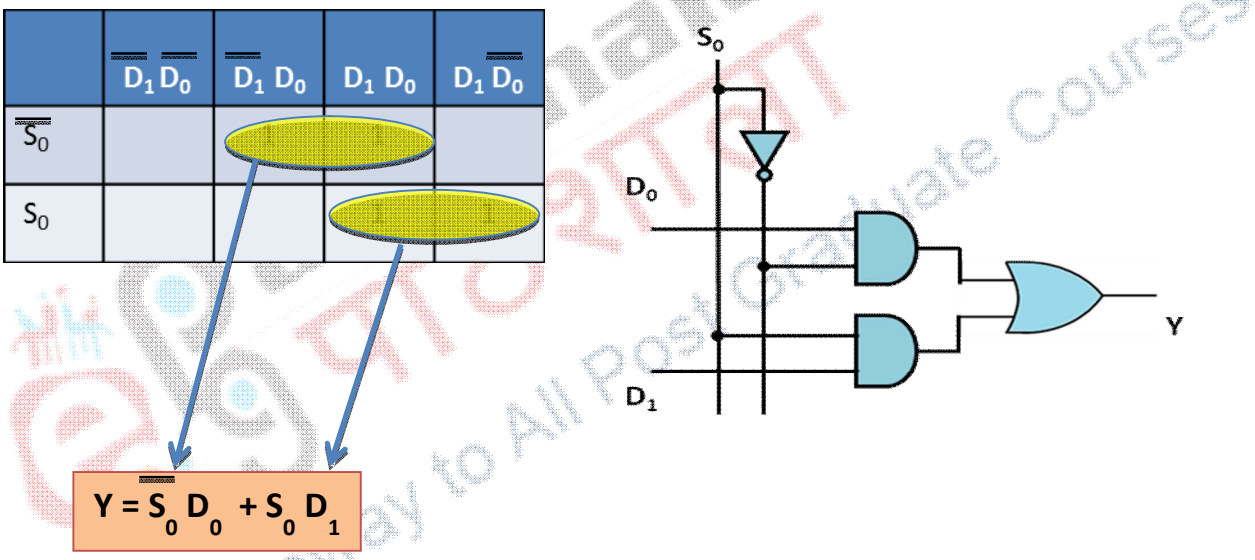


Figure 6: Simplification using K-map and Logic diagram of 2 to 1 Multiplexer

From the first pair we can eliminate the D1, as it is changing in the pair. Thus this pair contribute S0ðD0 in the output. Similarly, in the second group D0 gets eliminated as it is changing in the pair. Therefore, we get S0D1 as the second term in the Boolean expression.

The final Boolean expression is in sum of two product terms as shown here. This indicates either D0 or D1 will appear at the output decided by the value of Y.

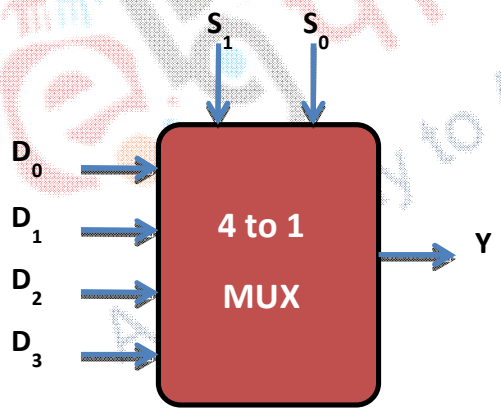
The 2 to 1 multiplexer can be implemented using two AND gates and one OR. The Not gate provides complemented and un-complemented form of S0 i.e. select line. The interconnections are made to AND gate to provide the necessary product term.

3.3 A 4 to 1 multiplexer

The second type of multiplexer is 4 to 1 multiplexer. The logic symbol (as shown in fig. 7) indicates that there are 4 data inputs namely D0, D1, D2,D3 and single output (Y) . For 4 data inputs there are two select lines namely S0 and S1. While preparing the function table we write S1 as MSB and S0 as LSB. The 2 select inputs provide 4 input combinations for selecting the proper data input at the output Y.

For 4 to 1 multiplexer, two bit binary code on select inputs (S1S0) allows the data from selected input (either from D0,D1,D2,D3) to pass to the output.

The output Y receives D0 only when S1=0 and S0=0. Similarly, output Y receives D1 only when S1S0=01. Output Y receives D2 only when S1S0=10. Output Y receives D3 only when S1S0=11.



Select inputs		Output
S_1	S_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

$$Y = S'_1 S'_0 D_0 + S'_1 S_0 D_1 + S_1 S'_0 D_2 + S_1 S_0 D_3$$

Figure 8: Logic symbol, Function table and Boolean expression for 4 to multiplexer

From the function table, the Boolean Expression can be written in SOP form. Each row of the function table provides the product term. Referring to the Boolean expression, it is possible to

draw the logic circuit consisting of a OR gate with 4 inputs. Each product term is represented by three input AND gate. One of the input of AND gate is the respective data input. The Select lines S1 and S0 along with inverter provide select input either in un-complemented or complemented form. Figure 8 indicates the logic diagram of 4 to 1 multiplexer. The data inputs and select lines are connected to the AND gates as per the requirement of the product term to generate the desired output.

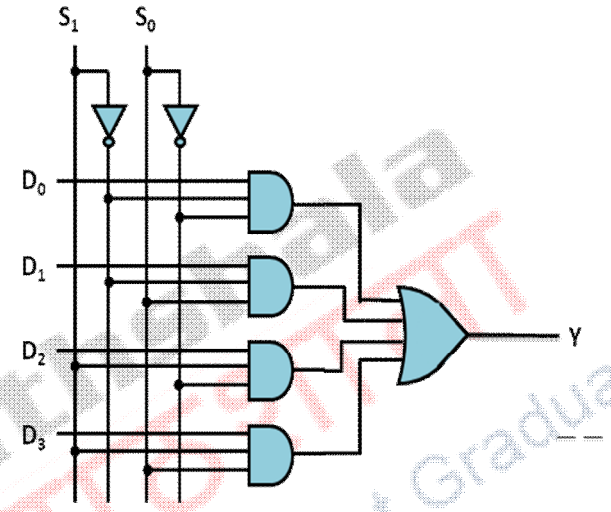


Figure 8: 4 to 1 multiplexer

The first AND gate receives D0, S1ô and S0ô as inputs. Similarly rest of the AND gates receives the appropriate inputs as per the product term.

As the data can be selected form any of the input lines, the multiplexer is also known as a **data selector**. In this fashion, it is possible to construct 8 to 1 multiplexer and 16 to 1 multiplexer. Note that the 8 to 1 multiplexer require three select inputs to select data from 8 inputs to the 1 output whereas the 16 to 1 multiplexer require 4 select lines.

3.4 Multiplexer ICs

IC number	Description	Output
74150	16 to 1 mux	Inverted input
74151	8 to 1 mux	Complementary outputs
74152	8 to 1 mux	Inverted input
74153	Dual 4 to 1 mux	Same as input
74157	Quad 2 to 1 mux	Same as input
74158	Quad 2 to 1 mux	Inverted input

Figure 9: Available multiplexer ICs in TTL family

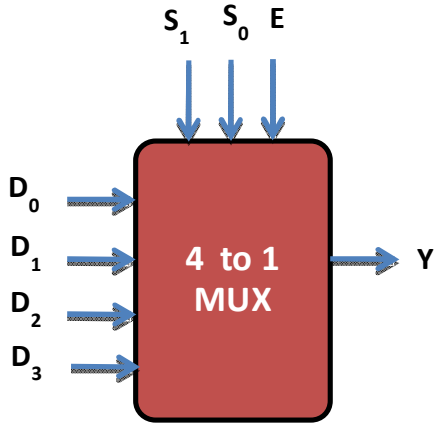
So far we have discussed construction of multiplexers using discreet logic gates. Commercially, multiplexers are available as MSI- IC format. The table indicates Multiplexer IC numbers for TTL logic family.

CMOS Multiplexer ICs are also popular amongst the Digital system designer because of low power consumption. Figure 10 indicates list of few CMOS multiplexer ICs.

IC number	Description
4051	8-channel analog multiplexer/ demultiplexer
4052	Dual 4-channel analog multiplexer/demultiplexer
4053	Triple 2-channel analog multiplexer/ demultiplexer
4067	16 channel mux / demux
4097	Differential 8-channel analog multiplexer/ demultiplexer

Figure 10: CMOS ICs for multiplexers

In many situations, an enable or gating input is added to the multiplexer. The multiplexer will be enables or operative only when the Enable input is active. In this case Enable is active high. When E=0, Multiplexer function is disabled. To enable multiplexer, E must be set to '1' as shown in fig. 11.

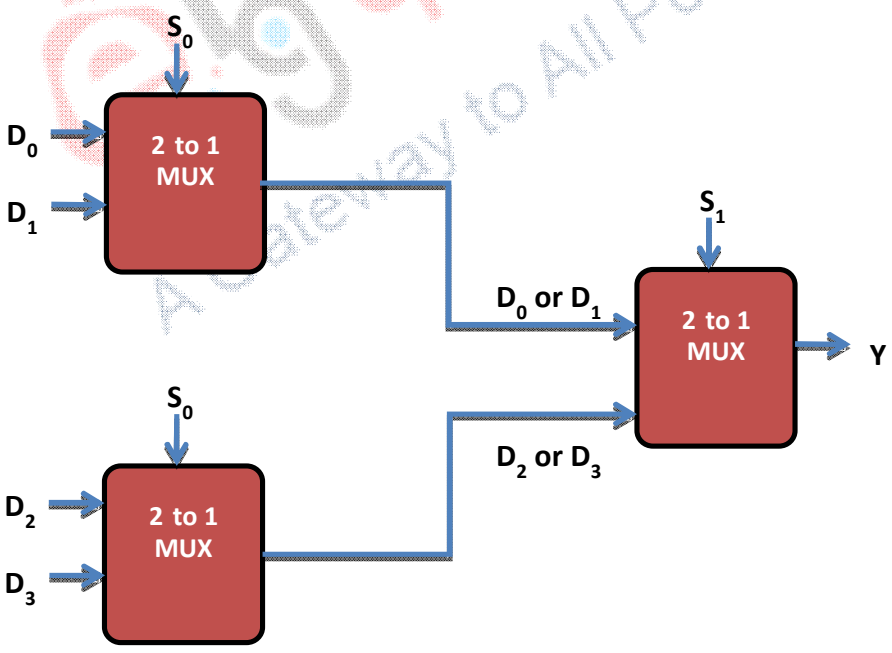


Select Inputs			Output
S_1	S_0	E	Y
X	X	0	X
0	0	1	D_0
0	1	1	D_1
1	0	1	D_2
1	1	1	D_3

Figure 11: Multiplexer with Enable input

3.5 Cascading multiplexer

As number of inputs to the multiplexers is limited say up to 16. To meet the larger input needs of multiplexers, smaller multiplexers can cascaded together for further expansion. This method of expansion of multiplexers is also known as tree multiplexing. Higher order multiplexers can be constructed by using lower order multiplexers as shown in Figure 12 and 13.



Select inputs		Output
S_1	S_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

Figure. 12: Construction of 4 to 1 multiplexer using two 2 to 1 multiplexer.

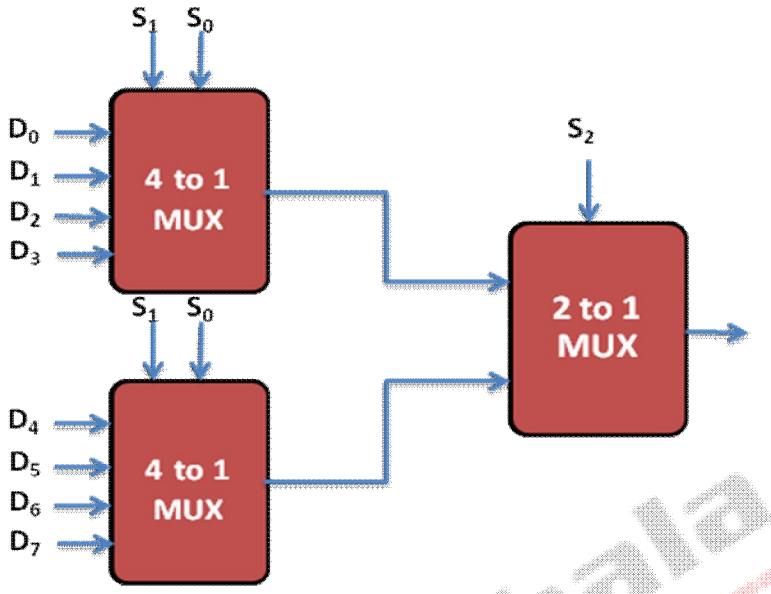


Figure 13 : Construction of 8 to 1 multiplexer

Select inputs			Output
S_2	S_1	S_0	Y
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3
1	0	0	D_4
1	0	1	D_5
1	1	0	D_6
1	1	1	D_7

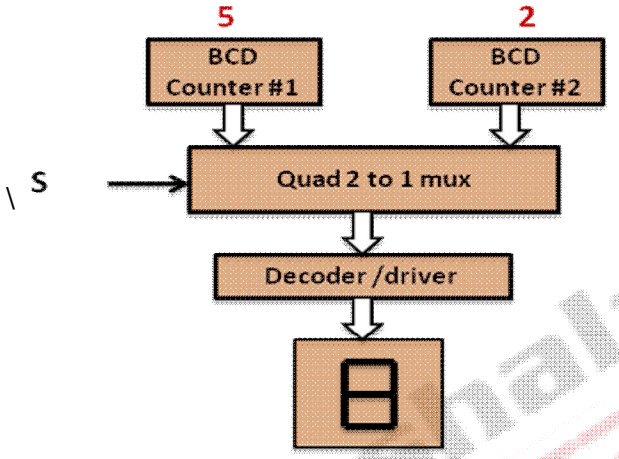
3.6 Applications of multiplexers

Multiplexer or data selectors are combinational circuits which transfer data from many sources to output under the control of data select lines. Multiplexer has many applications right from data routing, time division multiplexing, function generator to parallel to serial converter etc. A single multiplexer can replace several logic gates ICs, saving PCB area, interconnections, design efforts and cost. A list of popular applications is given below.

1. Data routing
2. Data bussing
3. Switch setting comparator
4. Multiplexer as a function generator
5. Parallel to serial converter
6. Cable TV signal distribution
7. Telephone network
8. Sharing printer /resources

1. Data Routing

Multiplexer can be used to rout digital data under the control of data select inputs. To display the contents of one of the two BCD counters on a common seven segment display is one of the important applications.



S	Display
0	Counter #1
1	Counter #2

Figure 14: Data routing

2. Data Bussing

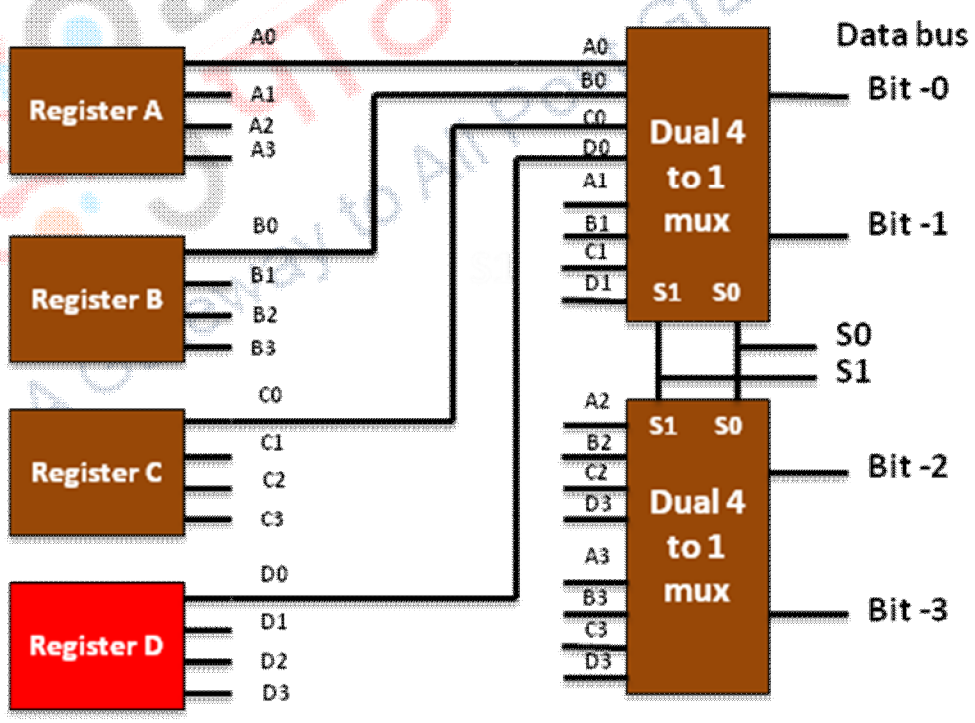


Figure 15: Design of bus structure

Multiplexers can be used to construct a data bus structure for microprocessors and microcontrollers. In this case multiplexers are used in such a manner that it accepts data from various multiple bit sources e,g, registers and connects the desired source to a data bus under the control of data selection inputs.

S1	S0	OUTPUT
0	0	Reg – A
0	1	Reg – B
1	0	Reg – C
1	1	Reg – D

Figure 16: Register selection logic

3. Switch setting comparator

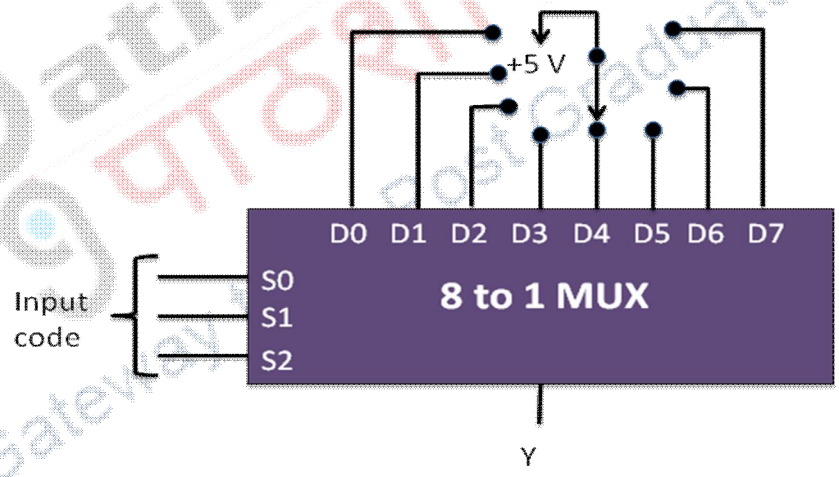


Figure 17: Switch setting comparator

Another application of multiplexer is switch setting comparator. A 3-bit code is used to represent one of eight possible switch positions and an output signal is generated to indicate equality. The fig indicates use of 8 t0 1 multiplexer as a switch setting comparator.

4. Multiplexer as function generator

For N- input variables , a total of 2^N different functions are obtainable . One can generate desired function using following procedure:

1. First write function in SOP form and identify the required multiplexer.
2. Assign appropriate input line to each product term . Use the decimal number corresponding to each term in the expression.
3. Connect these input lines to logic 1(+Vcc).
4. All other inputs which do not contribute to output function be connected to logic 0 level.
5. Apply the inputs to the select inputs of multiplexer.

Example: Let us consider implementation of the function

$$f(A,B,C) = \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}C$$

$$\sum m(1,3,5)$$

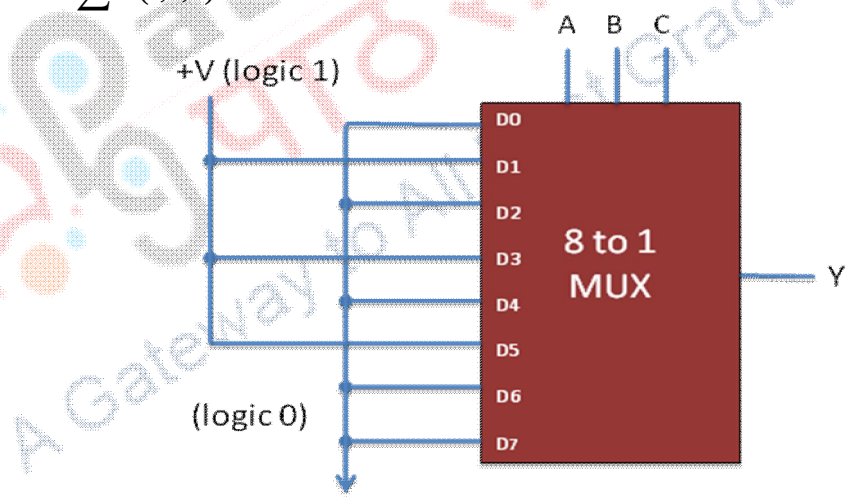


Figure 18: Multiplexer as a function generator

$$Sum = \sum m(1,2,4,7)$$

Example: Implement Full Adder using MUX:

A	B	C	SUM
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

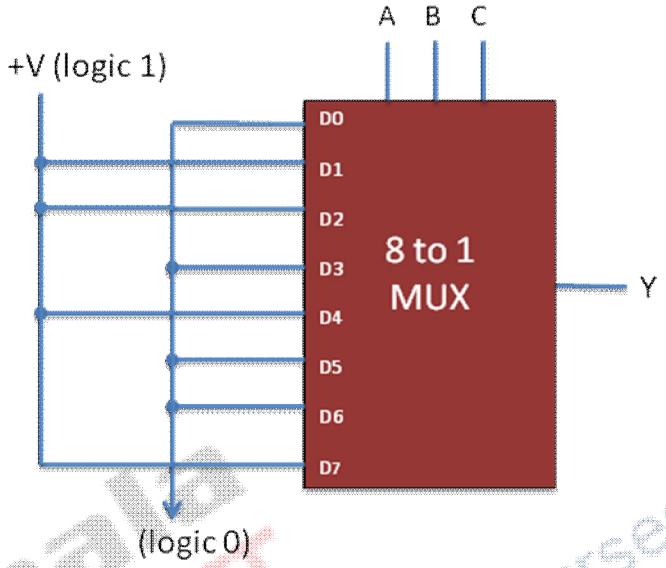


Figure 19: Full Adder using multiplexer

4. Demultiplexer

Demultiplexer has a single input and n output lines. Demultiplexer can be visualized as reverse multi-position switch. The select lines permit input data from single line to be switched to any one of the many output lines as shown in fig. 20.

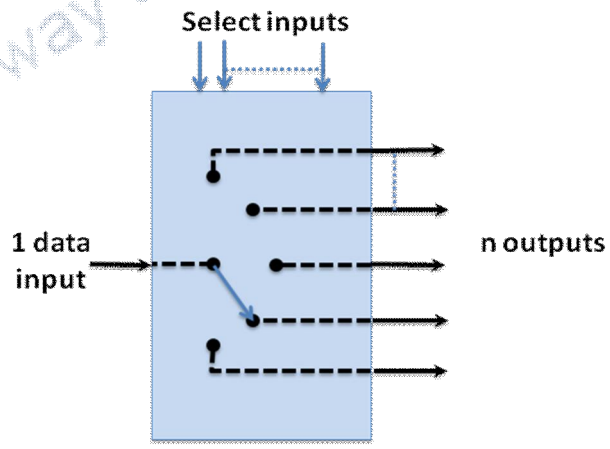


Figure 20: Multi-position switch as Demultiplexer

Demultiplex means one into many. A demultiplexer reverses the multiplexing operation. In other words, the demultiplexer takes one data input source and selectively distributes it to 1 of N output channels just like multi-position switch.

It also has ó**m**' select lines for selecting the desired output for the input data as shown in fig. 21. The mathematical relation between select lines and ó**n**' output are:

$$2^m = n$$

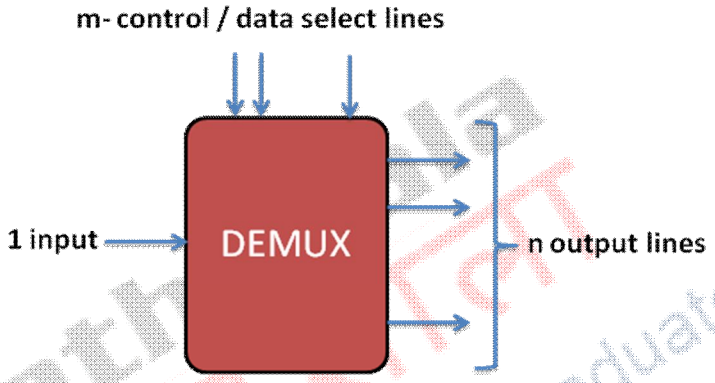
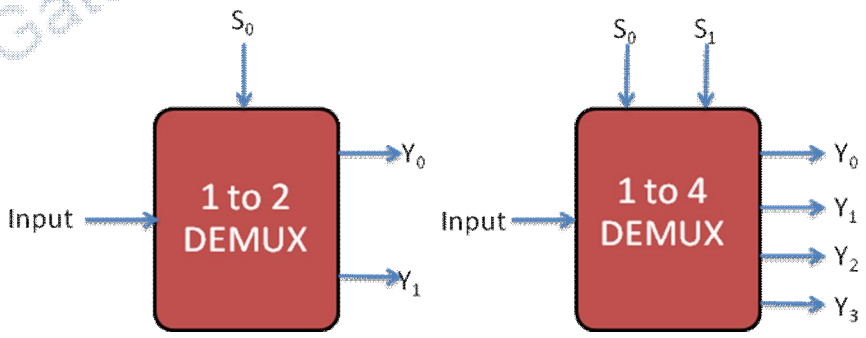


Figure 21: Logic symbol of basic demultiplexer

As a demultiplexer takes data from one input line and distributes over a 2^m output line, hence it is often referred to as 1 to 2^m **line converter**. There are four basic types demultiplexers: 1 to 2 demultiplexer, 1 to 4 demultiplexer, 1 to 8 demultiplexer and 1 to 16 demultiplexer as shown in fig. 22. Number of select lines decides this classification.



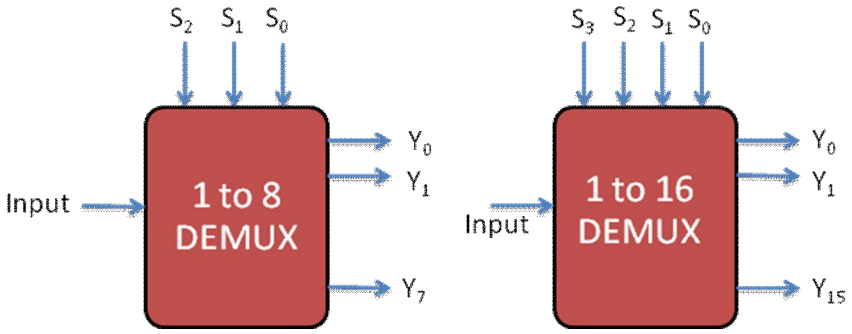
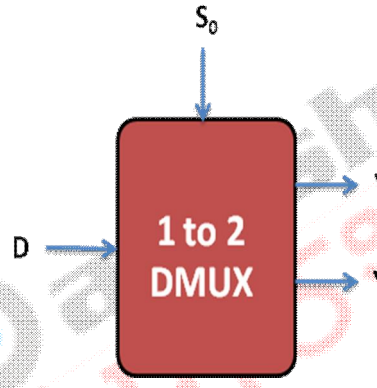


Fig. 22: Types of Demultiplexer

4.1 A 1to 2 demultiplexer



Select input	Outputs	
S0	Y0	Y1
0	D	0
1	0	D

Figure 23: Logic symbol and function table of a 1 to 2 demultiplexer

As shown in fig.22, in 1 to 2 demultiplexer, with S0=0 the Y0 output of demultiplexer receive the input data. Similarly when S0 becomes ó1ô, the Y1 output of demultiplexer receives the input data. Thus the Select or control line selects the desired output to which the input data is transferred or distributed. Hence, demultiplexer is also known as **data distributor**.

To distribute the input data D to Y0, the select input So should be 0 and Y1 will receive data input D when So=1. The Boolean expressions for the outputs are

$$Y_0 = \overline{S_0}\, D$$

$$Y_1 = S_0\, D$$

The implementation of 1 to 2 demultiplexer requires two 2 input AND gate and a NOT gate as shown in fig. 24. The product term for the output decides the interconnections between the gates data input and select lines.

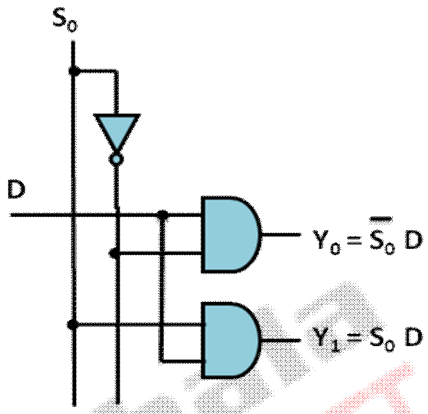
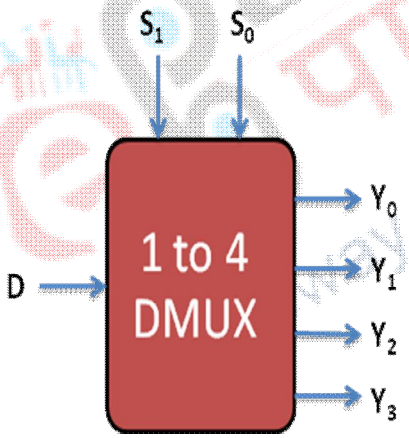


Figure 24: Logic diagram for 1 to 2 demultiplexer

4.2 A 1 to 4 demultiplexer



Inputs		Outputs			
S_1	S_0	Y_0	Y_1	Y_2	Y_3
0	0	D	0	0	0
0	1	0	D	0	0
1	0	0	0	D	0
1	1	0	0	0	D

Figure 25: Logic symbol and function table of a 1 to 4 demultiplexer

Fig. 25 indicates the logic symbol and function table of 1 to 4 demultiplexer. In 1 to 4 demultiplexer, the input data can be distributed to 1 of the 4 outputs. Selection of the output is decided by the binary word applied to the select lines. With S1S0=00, the Y0 output of demultiplexer receive the input data. For S1S0=01, the output Y1 receives the input data. With

S1S0=10, the input data is distributed to Y2 and when S1S0=11, the output Y3 receives the input data.

The Boolean expressions for the outputs are:

$$Y_0 = \overline{S_1}\,\overline{S_0}\,D$$

$$Y_1 = \overline{S_1}\,S_0\,D$$

$$Y_2 = S_1\,\overline{S_0}\,D$$

$$Y_3 = S_1\,S_0\,D$$

The implementation of 1 to42 demultiplexer requires four 3 input AND gates and two NOT gates as shown in fig. 26. The product term for the output decides the interconnections between the gates data input and select lines.

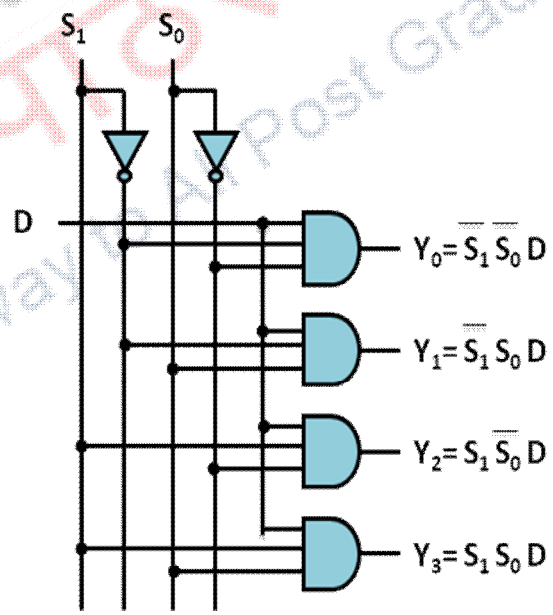


Figure 25: Logic diagram for 1 to 4 demultiplexer

4.3 Demultiplexer ICs

So far we have discussed construction of demultiplexers using discreet logic gates. Commercially,de multiplexers are available as MSI- IC format. The fig. 26 indicates a table of demultiplexer IC numbers for TTL logic family.

IC number	Description	Output
74138	1 to 8 demux/decoder	Inverted input
74139	Dual 1 to 4 demux/decoder	Inverted input
74155	Dual 1 to 4 demux/decoder	Complementary inputs
74156	Dual 1 to 4 demux/decoder	Open collecter ï complementary inputs
74154	1 to 16 demux/decoder	Same as input
74159	1 to 16 demux/decoder	Open collecter- same as input

CMOS ICs provide a combination of Multiplexer and Demultiplexer in a single chip IC. These are also popular amongst the Digital system designer because of low power consumption.

4.4 Applications of Demultiplexers

Digital demultiplexers are combinational devices controlled by a selector address that routes input data to one of many outputs of the demultiplexers. These can be used in following applications.

1. **Data demultiplexing**
2. **Clock demultiplexing**
3. **Memory addressing**
4. **Four phase clock generator**
5. **Function generation using DMUX**
6. **Switch encoding**
7. **Serial to parallel converter**

5. Summary

- A multiplexer is a logic circuit which routes data from many inputs to a single output and are available in 2 to 1, 4 to 1, 8 to 1 and 16 to 1 LSI forms. The multiplexer has 2^m input lines, m select lines and a single output.
- Multiplexers are commonly used in
    1) Data routing
    2) Data bussing
    3) Switch setting comparator
    4) Multiplexer as a function generator
    5) Parallel to serial converter
    6) Cable TV signal distribution
    7) Telephone network
    8) Sharing printer /resources
- The multiplexer is a universal logic circuit because it can generate any truth table.
- A demultiplexer is a logic circuit which takes data from one input line and distributes them to one of many output lines under the control of data select lines.
- Demultiplexer can be used as a decoder.
- Demultiplexers are commonly used in following applications.
    1. Data demultiplexing.
    2. Clock demultiplexing.
    3. Memory addressing.
    4. Four phase clock generator.
    5. Function generation using DMUX.
    6. Switch encoding.
    7. Serial to parallel converter.