Unit – 3

Session - 10

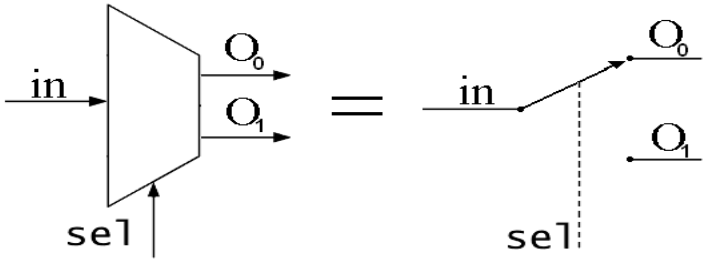
Data Processing Circuits

Objectives

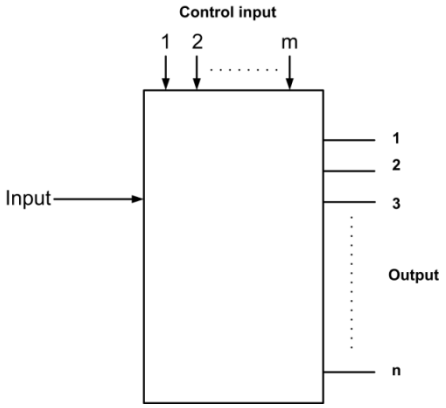
- Design of demultiplexers and demultiplexer trees
- Understanding decoders
- Combinational logic design using decoders
- Understanding seven-segment decoders and encoders

Demultiplexers

Demultiplex means one into many. A demultiplexer (DEMUX) is a combinational logic circuit with one input and many outputs. By applying control signals, we can steer the input signal to one of the output lines. The block diagram of a simple 1-to-2 demultiplexer block diagram and its switch equivalent circuit is as shown below:



Consider the block diagram of the general demultiplexer as shown below:



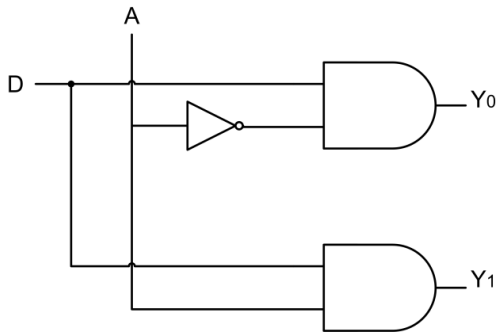
It has one input signal, 'm' control input or select signals, and 'n' output signals (n <= 2^m).

Design of 1-to-2 Demultiplexer

The truth table of 1-to-2 demultiplexer is as shown below:

Data input	Select input	Outputs	
D	A	Y_0	Y_1
0	0	0	0
1	0	1	0
0	1	0	0
1	1	0	1

The output expressions are Y_0 = D.A' and Y_1 = A.D and the logic circuit realization is as shown below:



Demultiplexer ICs

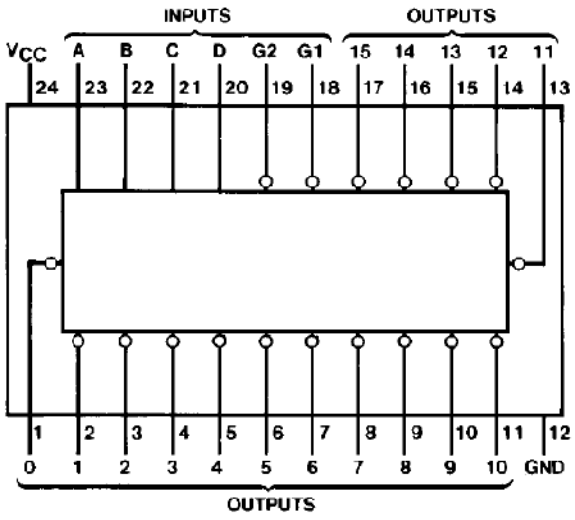
The popular demultiplexer TTL ICs are listed below. They can also be used as decoders.

IC No.	DEMUX Type	Decoder Type
74154	1-to-16	4-to-16
74138	1-to-8	3-to-8
74155	1-to-4	2-to-4

The TTL IC 74154

The 74154 is a 1-to-16 demultiplexer / decoder. G1 is used as data input. G2 is used as strobe input and must be low to activate the IC. Data is inverted at the input and again on any output. With double inversion, data passes through the circuit unchanged.

The pin out diagram is as shown:



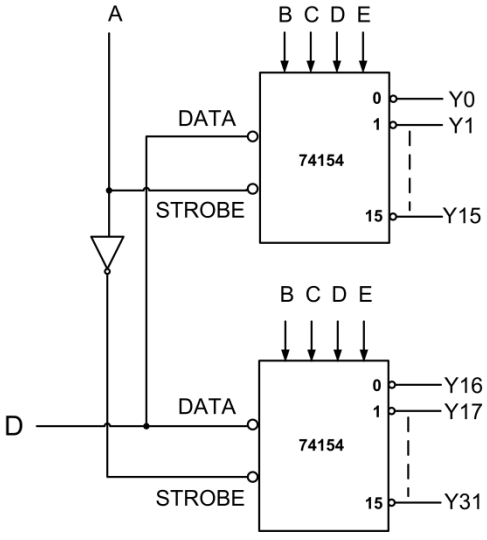
Demultiplexer Tree

Example 1:

Realize 1-to-32 demultiplexer using two 1-to-16 demultiplexers

Solution:

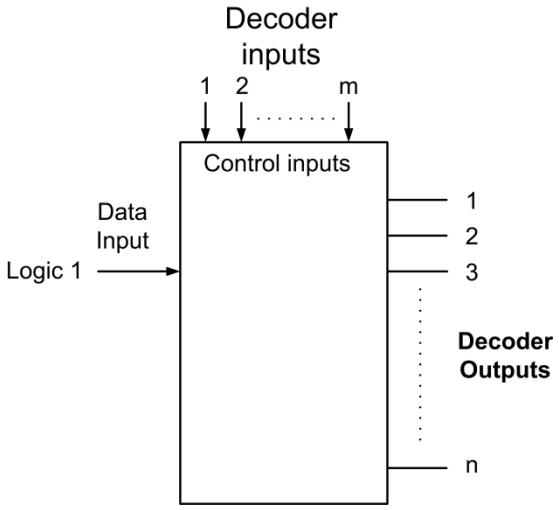
1-to-32 demux has 5 select variables A, B,C, D, E. We use two units of 1-to-16 demultiplexers to obtain 32 outputs. If A = 0, the top IC is chosen, and if A = 1, the bottom IC is chosen. Depending on value of BCDE, data is directed to one of the outputs. The 1-to-32 demultiplexer is realized as shown below:



Decoder

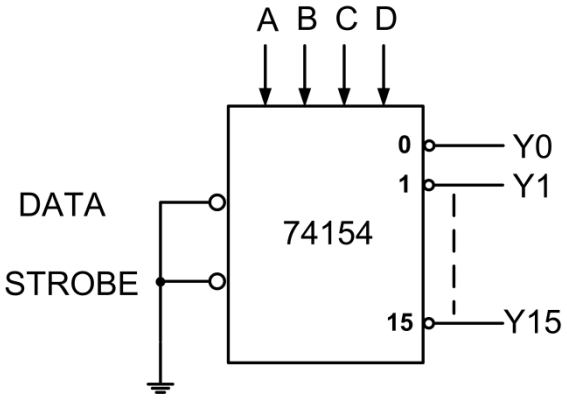
Demultiplexer can be used as a decoder. The data input of the demultiplexer is always connected to logic 1. The decoder inputs are given to the control or select lines. The inputs are decoded by activating one of the output lines. 1-to-n demultiplexer converted in to 1-of-n decoder. Only one of the 'n' output lines will be high.

The block diagram of 1-of-n decoder is as shown below:



1-of-16 Decoder IC

74154 IC can be used as 1-of-16 decoder. 1-of-16 decoder is also called 4 line-to-16 line decoder. The block diagram is as shown below:



Decoder Applications

Decoders can be used in combination logic circuit design.

Example 1 :

Show how using a 3-to-8 decoder and multi-input OR gates following Boolean expressions can be realized simultaneously:

f1(A, B, C) = Σ m(0, 4, 6)

f2(A, B, C) = Σ m(0, 5)

f3(A, B, C) = Σ m(1, 2, 3, 7)

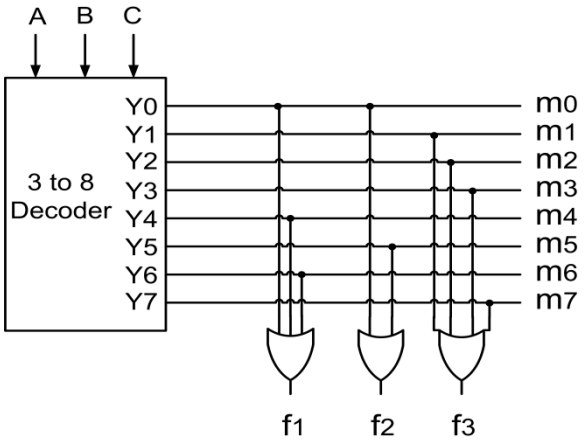
Solution:

f1 = Σ m(0, 4, 6) = m0 + m4 + m6

f2 = Σ m(0, 5) = m0 + m5

f3 = Σ m(1, 2, 3, 7) = m1 + m2 + m3 + m7

We use OR gates to add the minterms. The Boolean expressions are realized as shown below:

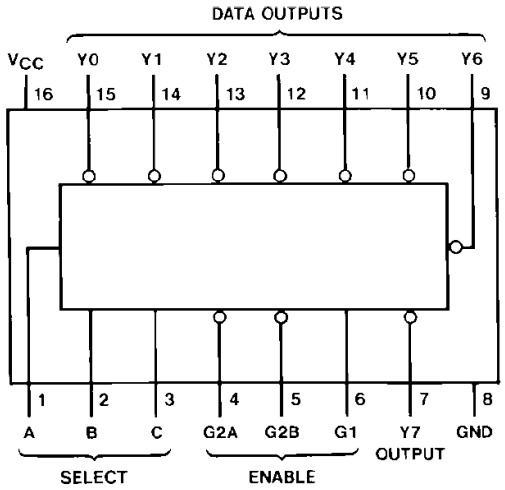


Example 2:

Realize a full adder using 3-to-8 decoder IC 74138 and NAND gates.

Solution:

The pin out diagram of the 3-to-8 decoder IC 74138 is as shown:



The truth table of full adder is written as shown below:

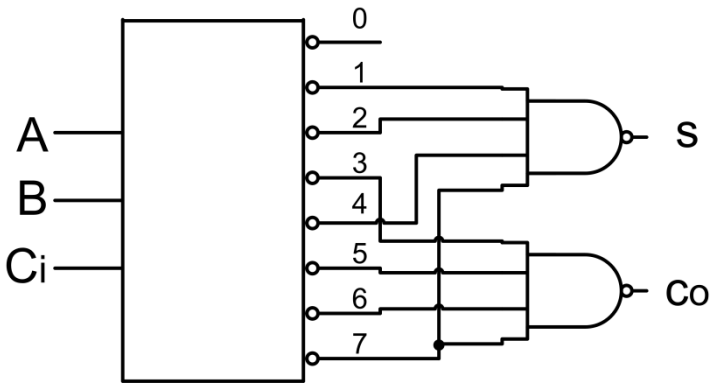
Inputs			Outputs	
A	B	Ci	S	Co
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

The output expressions in SOP form:

$S = \Sigma\, m(1, 2, 4, 7)$

$Co = \Sigma\, m(3, 5, 6, 7)$

Full adder is realized as shown:



BCD-to Decimal Decoders

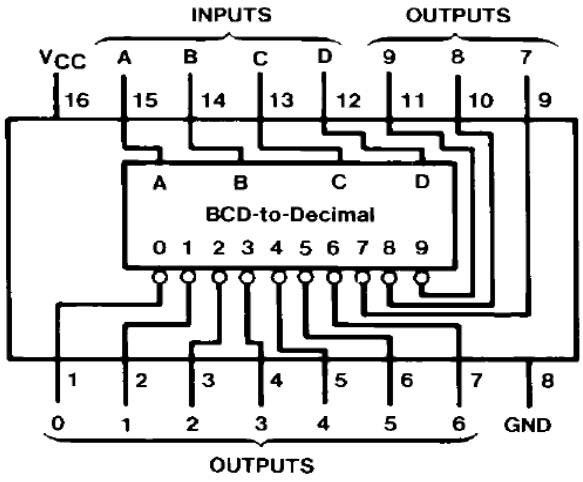
BCD is an abbreviation for binary-coded decimal. The BCD code expresses each digit in a decimal number by its nibble equivalent.

Example: Decimal number 429 in its BCD form is

4	2	9
0100	**0010**	**1001**

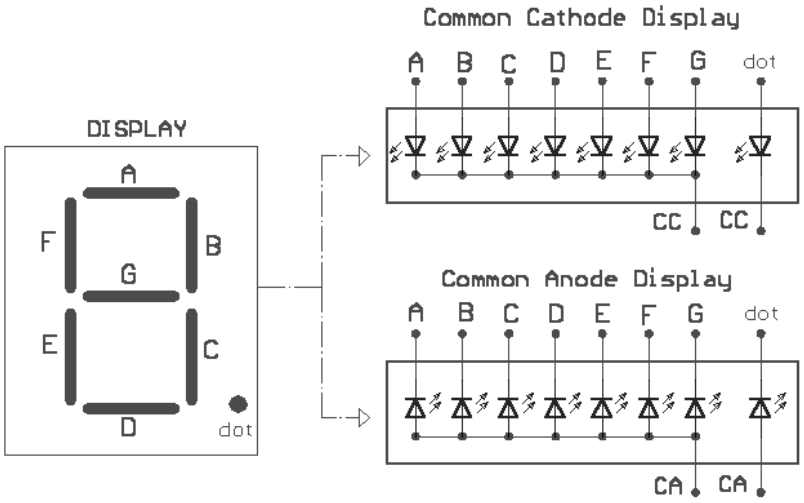
The 7445 IC

The 7445 is a BCD to Decimal Decoder / Driver IC. It is also called 1-of-10 decoder and its pin out diagram is as shown below:



Seven-Segment Decoders

A seven-segment decoder-driver is used to drive a seven-segment indicator. A seven-segment indicator has 7 Light-emitting Diodes labeled a – g as shown:



It may be the common-anode or the common-cathode type. By forward-biasing different LED's, we can display the decimal digits 0 -9.

The 7446 IC

The 7446 is a TTL IC decoder-driver that can be used to drive a common-anode seven-segment indicator. It requires external current-limiting resistors.

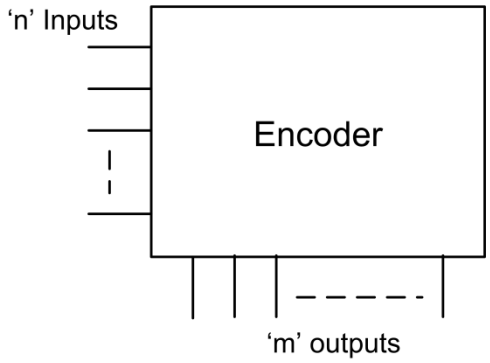
The 7448 IC

The 7446 is a TTL IC decoder-driver that can be used to drive a common-cathode seven-segment indicator. It has its own current-limiting resistors on the chip.

Encoder

Encoding is the process of converting familiar numbers or symbols into some code format. An encoder is a digital circuit that receives digits (decimal, octal, etc.) or alphabets or special symbols and converts them into their respective binary codes. It has 'n' input lines, only one of which is active. The active input is converted to a coded binary output with 'm' bits. Basically it is a combinational logic circuit.

Encoder performs operation reverse to that of a decoder. The block diagram is as shown:



Octal-to-Binary Encoder

It has 8 inputs, corresponding to 8 octal digits. It converts the selected octal digit into 3-bit binary output. The truth table is as shown:

Inputs – Octal Digits								Binary Outputs		
I0	I1	I2	I3	I4	I5	I6	I7	B_2	B_1	B_0
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

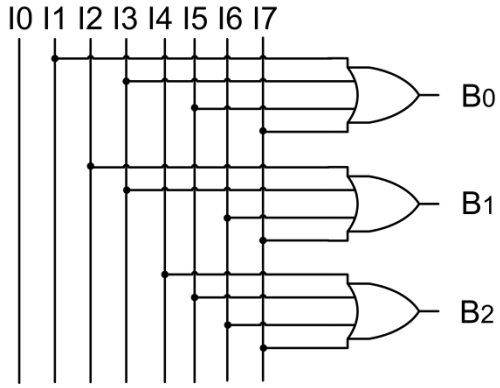
The output expressions are:

$B_0 = I1 + I3 + I5 + I7$

$B_1 = I2 + I3 + I6 + I7$

$B_2 = I4 + I5 + I6 + I7$

The octal-to-binary encoder is realized as shown:



Decimal-to-BCD Encoder

It has 10 inputs, corresponding to 10 decimal digits. It converts the selected decimal digit into 4-bit BCD output.

The 74147 IC

The 74147 TTL IC is a decimal-to-BCD encoder. It is a priority encoder because it gives priority to the highest-order input.

Design of Priority Encoder

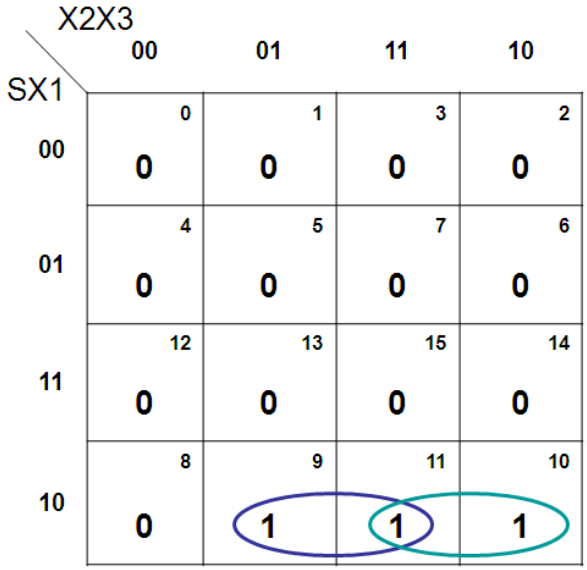
Example:

Design a priority encoder, the truth table of which is shown below. The order of priority for three inputs is X1>X2>X3. However, if the encoder is not enabled by S or all the inputs are inactive the output AB = 00.

Inputs				Outputs	
S	X1	X2	X3	A	B
0	X	X	X	0	0
1	1	X	X	0	1
1	0	1	X	1	0
1	0	0	1	1	1
1	0	0	0	0	0

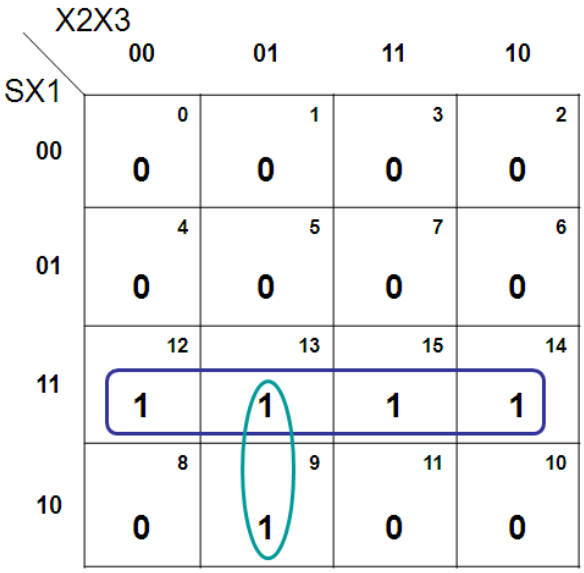
Simplification using K-map:

For output A:



We obtain A = SX1'X3' + SX1'X2.

For output B:



We obtain B = SX1 + SX2'X3.

The output expressions are realized using logic gates.

Questions

1. Define decoder. Draw logic diagram of 3:8 decoder with enable input.

2. Design a circuit that realizes the following two functions using a decoder and logic gates:

   F1(A, B) = ∑m(0, 3) and F2(A, B) = ∑m(1, 2)

2. Define encoder. Design decimal-to-BCD encoder?

4. What is a priority encoder?