Digital Logic Circuits

4

Digital logic circuits are classified as:

(a) Combinational circuit

(b) Sequential circuit

Combinational Circuit	Sequential Circuit
1. Present output depends on present input only.	1. Present output depends on present input as well as previous output
2. No feedback is present.	2. Feedback is present.
3. No memory is present.	3. Due to feedback, memory is present
4. Example: Comparator, half adder, full adder, half subtractor, full-subtractor, multiplexer, demultiplexer,encoder, decoder.	4. Example: Flip-flop, counter, register.

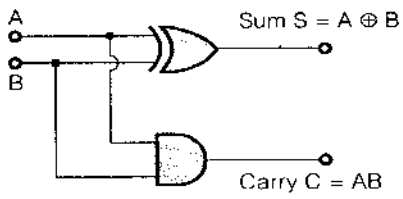
Combinational Circuits

(a) Arithmetic Circuit

Half Adder (H.A)

A logic circuit for the addition of two one-bit numbers is referred to as an "HALF ADDER (H.A)".

Symbol and Truth table:



Inputs		Outputs	
A	B	Sum(S)	Carry(C)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Logical expression:

Sum $\boxed{S = \bar{A}B + A\bar{B} = A \oplus B}$

Carry $\boxed{C = AB}$

Remember:

- Total number of NAND-gates required to implement half adder = 5
- Total number of NOR-gates required to implement half adder = 5
- To implement the half adder circuit by minimum number of logic gates (if we have all gates except EXOR and EXNOR) is "3".
- Total number of MUX required to implement half adder = 3.

Full Adder (F.A.)

It performs the arithmetic sum of the three input bits i.e. addend bit, augend bit and carry bit.

Logical expression:

Sum, $\boxed{S = A \oplus B \oplus C}$

Carry, $\boxed{C = AB + BC + CA = AB + C(A \oplus B)}$

Remember:

- A F.A. can be implemented by two H.A. and one OR-gate
- Total number of NAND-gate/NOR-gate required to implement a F.A is equals to "9".
- Total number of MUX required to implement full adder = 7.

Half Subtractor (H.S)

Logical expression:

Difference, $\boxed{D = \bar{A}B + A\bar{B} = A \oplus B}$

Borrow, $\boxed{B = \bar{A}B}$

Remember:

- Total number of NAND/NOR gates required to implement the H.S is equals to "5".
- Total number of MUX required to implement half subtractor = 3.

Full Subtractor (F.S)

It is a circuit which performs a subtraction between two bits taking into account that a '1' may have been borrowed by a lower significant stage.

Logical expression:

Difference, $\boxed{D = A \oplus B \oplus C}$

Borrow, $\boxed{B = \bar{A}B + \bar{A}C + BC = \bar{A}B + \overline{(A \oplus B)} \cdot C}$

Remember:

- A full subtractor can be implemented with two half subtractor and one OR Gate.
- Number of NAND/NOR gates required to implement the full subtractor is equal to '9'
- In parallel adder n full adder or {(n – 1) F.A. and 1 H.A.} or {(2n – 1) H.A. and (n – 1) or Gate} are required to add two n-bit numbers.
- Total number of MUX required to implement full subtractor = 7.

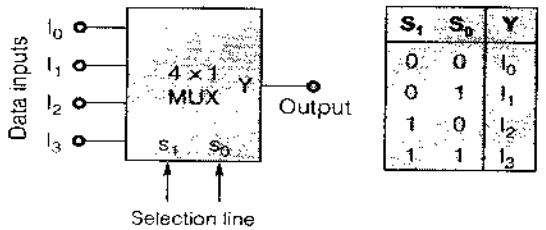
(b) Non Arithmetic Circuits

Multiplexer

Multiplexer is a combinational circuit which have many data input and single output depending on select or control input, one of the input line is transfer to the output, hence it is known as "many to one circuit" or "universal logic circuit" or "data selector circuit".

- The selection of a particular input line is controlled by a set of select lines.
- There are 2^n input lines where 'n' is the number select line.

4 : 1 MUX



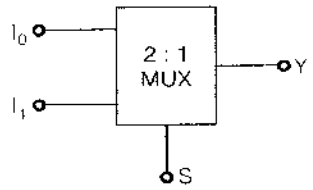
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

$$Y = \text{output} = \bar{S}_1 \bar{S}_0 I_0 + \bar{S}_1 S_0 I_1 + S_1 \bar{S}_0 I_2 + S_1 S_0 I_3$$

Note:

- The size of the MUX is specified by the 2^n of its input line and the single output line.
- MUX contains AND gate followed by an OR-gate.

2 : 1 MUX



$Y = \bar{S}I_0 + SI_1$

Higher order MUX using Lower order MUX:

Given MUX	To be Implemented MUX	Required number of MUX
2 : 1	4 : 1	3
4 : 1	16 : 1	4 + 1 = 5
4 : 1	64 : 1	16 + 4 + 1 = 21
8 : 1	64 : 1	8 + 1 = 9
8 : 1	256 : 1	32 + 4 + 1 = 37

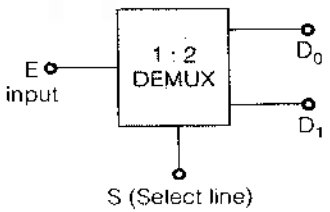
Remember:

- To implement 2^n : 1 MUX by using 2 : 1 MUX, the total number of 2 : 1 MUX required is $(2^n - 1)$.
- MUX is an Universal Logic gate.

Demultiplexer (DEMUX)

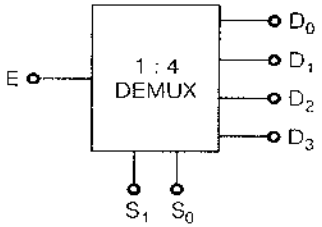
- It receives information on a single line and transmits it on one of 2^n possible output lines.
- A "Decoder" with an enable input can function as a "Demultiplexer".

1 : 2 Demux



$D_0 = \bar{S}E$ and $D_1 = SE$

1 : 4 DEMUX



Select lines		Outputs			
S_1	S_0	D_3	D_2	D_1	D_0
0	0	0	0	0	E
0	1	0	0	E	0
1	0	0	E	0	0
1	1	E	0	0	0

$D_0 = \bar{S}_1\bar{S}_0E, D_1 = \bar{S}_1S_0E, D_2 = S_1\bar{S}_0E, D_3 = S_1S_0E$

Remember:

- Other name of demultiplexer are "one to many circuit" or "data distributor circuit".
- Number of select line is $\log_2 n$ (where n is number of output line).

Higher Order DEMUX Using Lower Order DEMUX:

Given DEMUX	To be Implemented MUX	Required number of MUX
1 : 2	1 : 4	1 + 2 = 3
1 : 2	1 : 8	1 + 2 + 4 = 7
1 : 2	1 : 16	1 + 2 + 4 + 8 = 15
1 : 2	1 : 64	1 + 2 + 4 + 8 + 16 + 32 = 63
1 : 4	1 : 16	1 + 4 = 5

Decoder

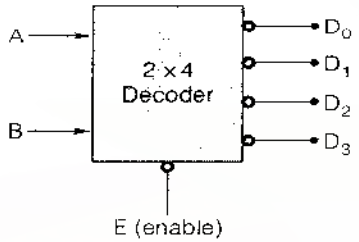
- Decoder is a combinational circuit that is use to convert binary to other codes such as
  (a) Binary to octal
  (b) Binary to decimal
  (c) Binary to hexadecimal
- In decoder, depending on binary data applied, one of the output line will become logic 1.
- A "Decoder" has many inputs and many outputs line.
- It is a combinational circuit that converts binary information from n input lines to a maximum 2^n unique output lines.
- If the n-bit decoded information has unused or don't care combinations, the decoder output will have less than 2^n outputs.

- Total number of output lines

$$m \leq 2^n$$

where n = Total number of input lines

2 × 4 Decoder



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

Note:

- 2 × 4 - Decoder may acts like 1 : 4 DE MUX and Vice-versa.
- Decoder and Demux circuits are almost same.
- Decoder contains AND-gates or NAND-gates.

Higher Order Decoder Using Lower Order Decoder:

Given Decoder	To be implemented Decoder	Required number of Decoder
2 : 4	4 : 16	1 + 4 = 5
2 : 4	3 : 8	2 + a NOT Gate
4 : 16	8 : 256	1 + 16 = 17

Encoder

- Encoder is used to convert other codes to binary such as
  (a) Octal to binary
  (b) Hexadecimal to binary
  (c) Decimal to binary
  (d) Decimal to BCD
- In encoder one of the input line is high and corresponding binary is available at the output.
- In priority encoder any number of inputs can be high but based on priority highest priority input corresponding binary is available at the output.

Code Converters

BCD to Excess-3 code

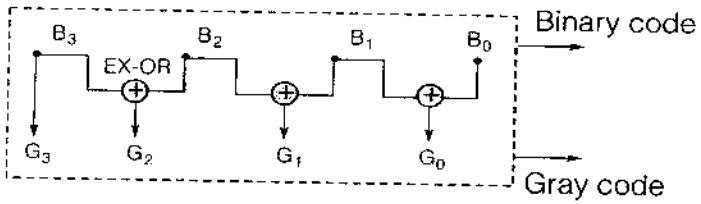
- Let input variables of **BCD code** is A, B, C, D and output variables of excess-3 is W, X, Y, Z.
- Required truth table: (BCD + 0011 = excess-3-code)

Inputs (BCD)				Output (Excess-3 code)			
A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

Minimized Boolean function:

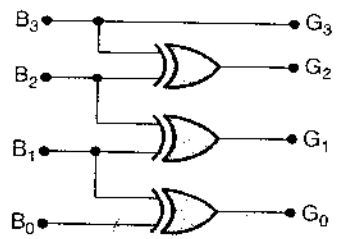
$$Z = \bar{D}$$
$$Y = C\bar{D} + \bar{C}D$$
$$X = \bar{B}C + \bar{B}D + B\bar{C}\bar{D}$$
$$W = A + BC + BD$$

Binary-to Gray Code Converter

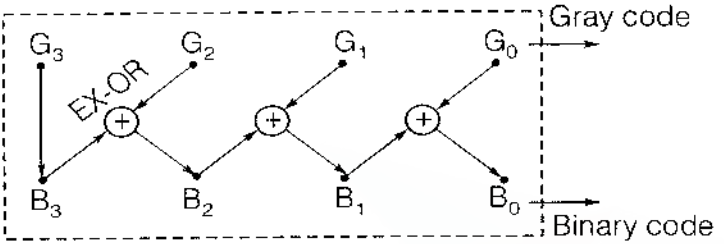


Equivalent logical gate diagram:

$$G_3 = B_3$$
$$G_2 = B_3 \oplus B_2$$
$$G_1 = B_2 \oplus B_1$$
$$G_0 = B_1 \oplus B_0$$

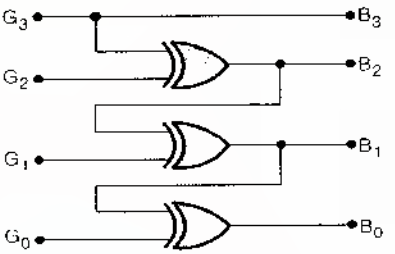


Gray to Binary Converter

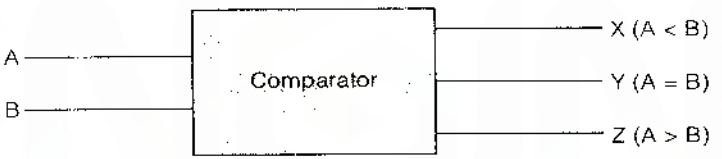


Equivalent Logical Gate Diagram:

$B_3 = G_3$
$B_2 = B_3 \oplus G_2$
$B_1 = B_2 \oplus G_1$
$B_0 = B_1 \oplus G_0$



Magnitude Comparator



Truth table:

A	B	X	Y	Z
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0
Output expression		$\overline{A}B$	$AB + \overline{A}\overline{B} = A \odot B$	$A\overline{B}$

■■■■