Parul® University

Digital Electronics 203105201

Prof. Mitul M. Patel, Assistant Professor
Electronics & Communication Engineering

Parul®
University

DIGITAL LEARNING CONTENT

CHAPTER-3

COMBINATIONAL CIRCUITS

4.1: Introduction to Combinational circuits:

- Combinational circuit is a circuit in which we combine the different gates in the circuit, for example encoder, decoder, multiplexer and demultiplexer. Some of the characteristics of combinational circuits are following:

- The output of combinational circuit at any instant of time depends only on the levels present at input terminals.
- This circuit do not use any memory. The previous state of input does not have any effect on the present state of the circuit.
- It can have an n number of inputs and m number of outputs.

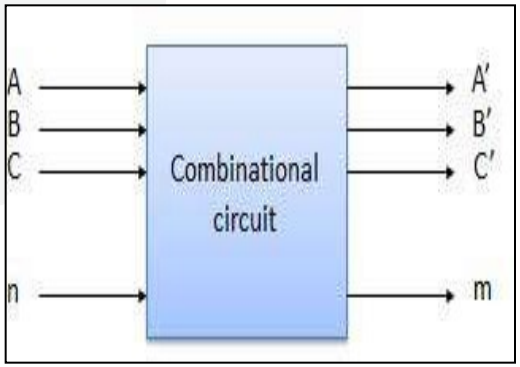


Image source : Google

4.2: Design Procedure

- The procedure involves the following steps:
- The problem is stated.
- The number of available input variables and required output variables is determined.
- The input and output variables are assigned letter symbols.
- The truth table that defines the required relationships between inputs and outputs is derived.
- The simplified Boolean function for each output is obtained.
- The logic diagram is drawn.

4.3: Adders

4.3.1: Half Adder:

- The half adder circuit is designed to add two single bit binary number A and B. It is the basic building block for addition of two single bit numbers. This circuit has two outputs carry and sum.
- S = A'B + AB'
- C=AB

Inputs		Outputs	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

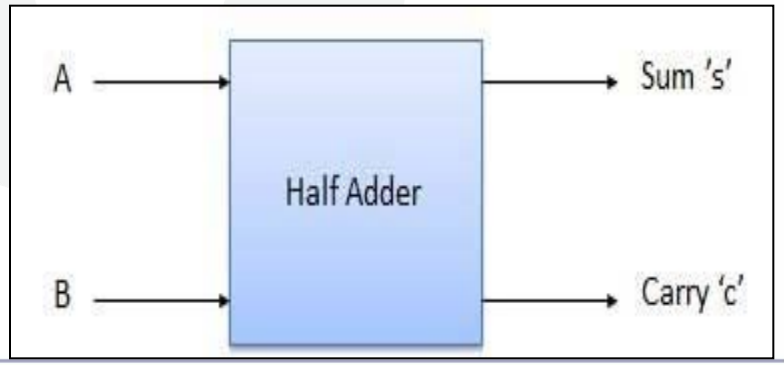


Image source : Google

4.3: Adders (Continue..)

4.3.1: Half Adder (Continue..):

- S = A'B + AB'
- C=AB
- Based on the above equations, the circuit can be designed as shown here.

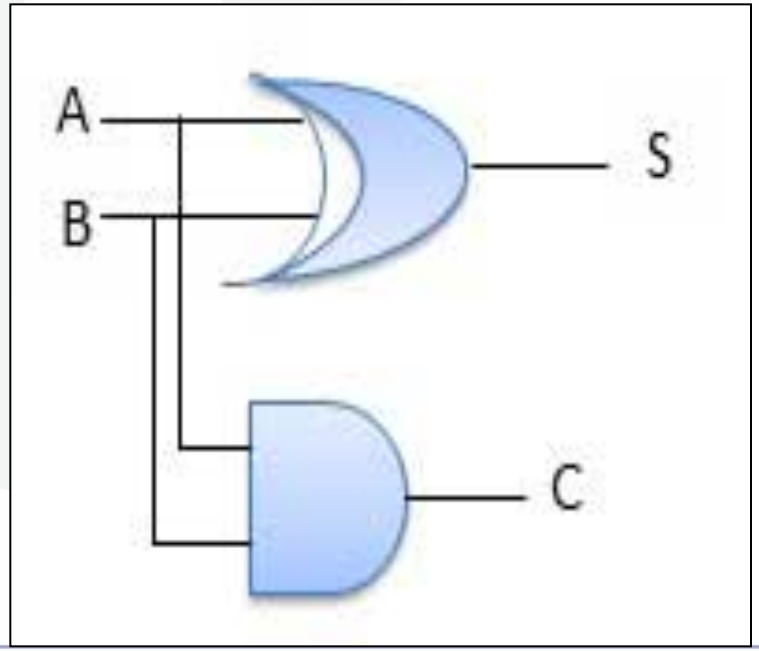


Image source : Google

4.3: Adders (Continue..)

4.3.2: Full Adder :

- Full adder overcomes the drawback of Half Adder circuit.
- It can add two one-bit numbers A and B, and carry C.
- The full adder is a three input and two output (Sum and Carry out) combinational circuit.

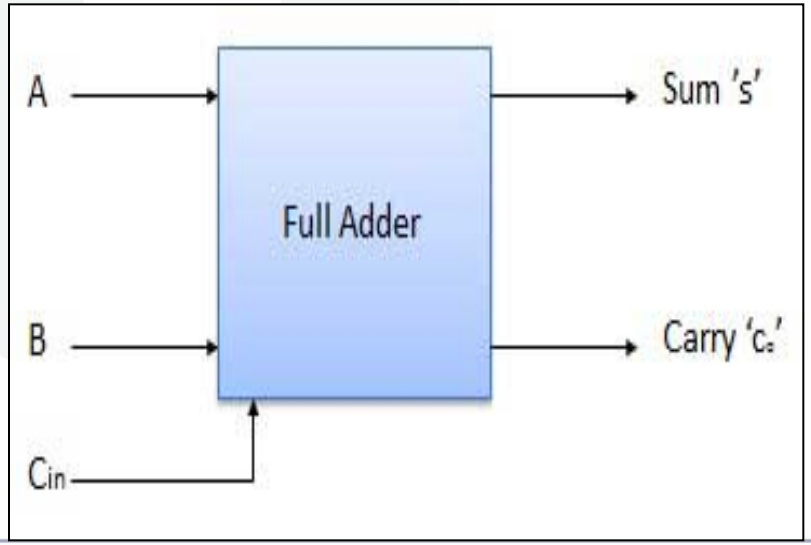


Image source : Google

4.3: Adders (Continue..)

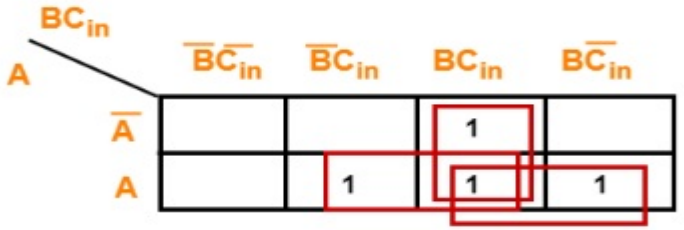
4.3.2: Full Adder (Continue):

Inputs			Outputs	
A	B	Cin	S	Cout
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

For S:

$$S = A \oplus B \oplus C_{in}$$

For C_{in}:



$$C_{out} = AB + BC_{in} + C_{in}A$$

Image source : Google

4.3: Adders (Continue..)

4.3.2: Full Adder (Continue):

•From the derived equations, circuit of full adder can be designed as shown in figure.

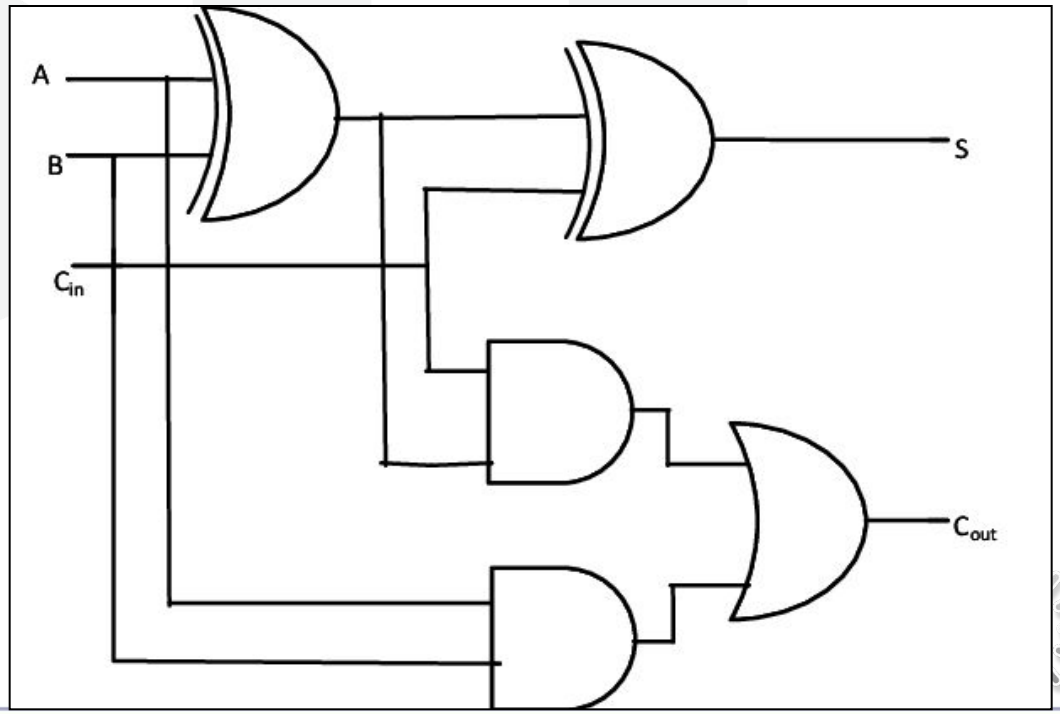


Image source : Google

4.4: Subtractors (Continue..)

4.4.1: Half Subtractor:

- A half-subtractor is a combinational circuit that subtracts two bits and produces their difference.
- It also has an output to specify if a 1 has been borrowed.

Inputs		Outputs	
A	B	Sum	Carry
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

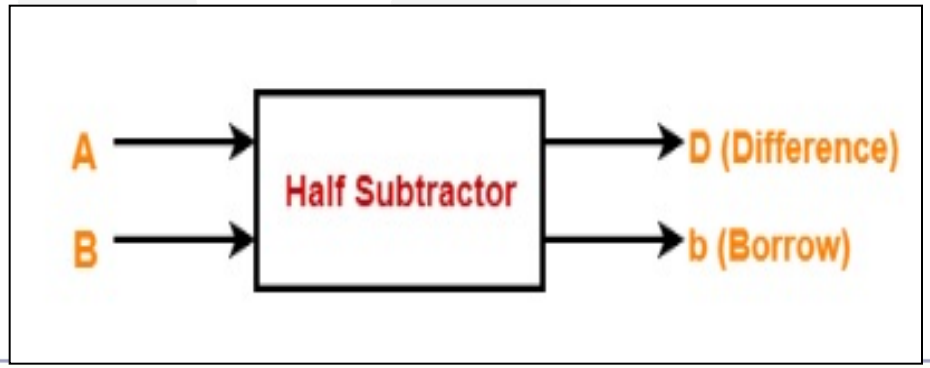


Image source : Google

4.4: Subtractors (Continue..)

4.4.1: Half Subtractor (Continue..):

- The simplified expressions for Sum and carry are as follows.
- D = A'B + AB'
- B=A'B
- From the derived equations, the circuit of half subtractor can be designed as shown in figure.

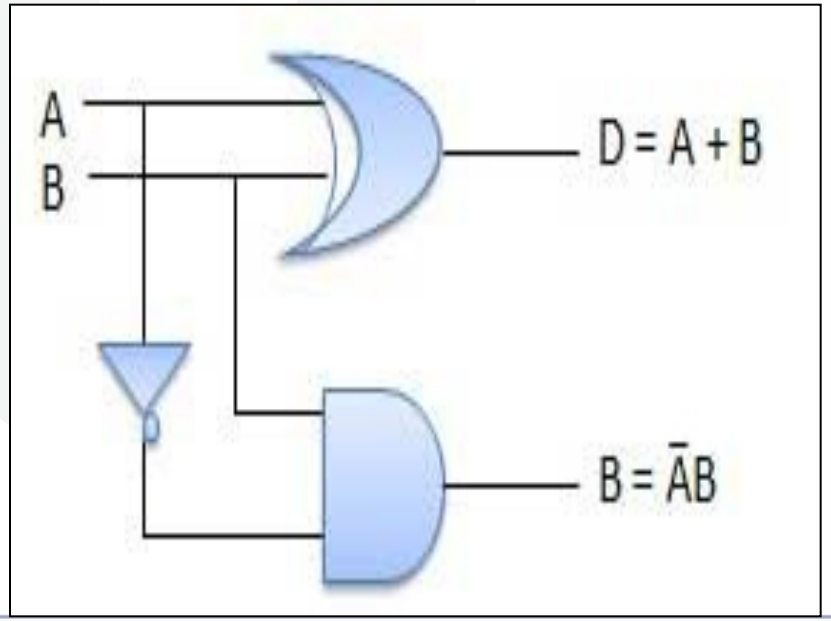


Image source : Google

4.4: Subtractors (Continue..)

4.4.2: Full Subtractor:

•Full subtractor overcomes the disadvantage of half-subtractor.

•A full-subtractor performs a subtraction between two bits, taking into account that a 1 may have been borrowed by a lower significant stage.

- It is a combinational circuit with three inputs A,B,C and two output D and B. A is the 'minuend', B is 'subtrahend', B is the 'borrow' produced by the previous stage, D is the difference output and C' is the borrow output.

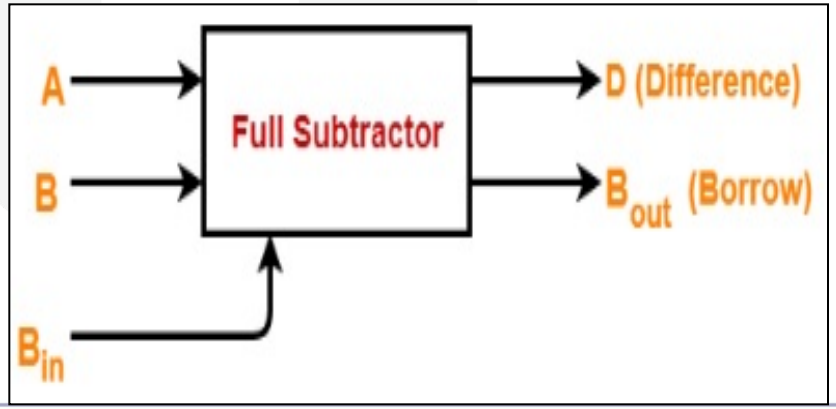


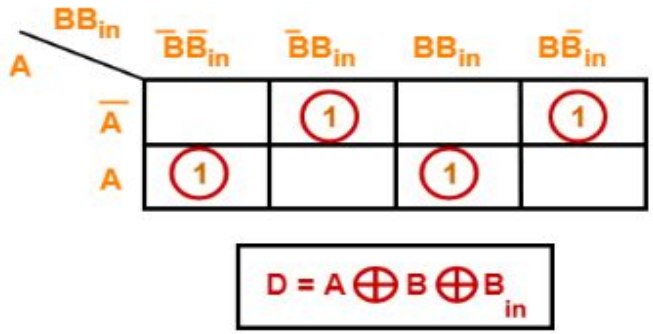
Image source : Google

4.4: Subtractors (Continue..)

4.4.2: Full Subtractor (Continue..):

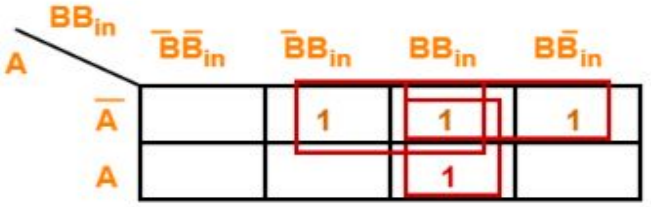
Inputs			Outputs	
A	B	Bin	D	Bout
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

For D:



$$D = A \oplus B \oplus B_{in}$$

For B_{in}:



$$B_{out} = \bar{A}B + (\bar{A} + B)B_{in}$$

Image source : Google

4.4: Subtractors (Continue..)

4.4.2: Full Subtractor (Continue..):

•From the expressions derived in previous slide, the circuit of full subtractor can be designed as shown here.

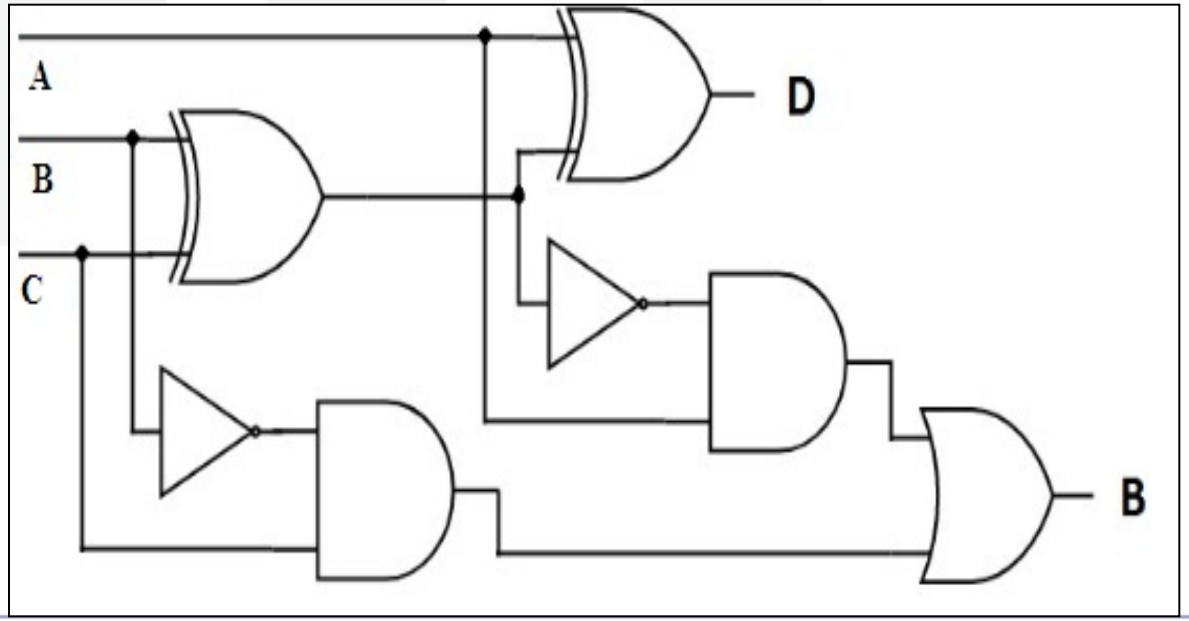


Image source : Google

4.5: N-Bit Parallel Adder

- To add two n-bit binary numbers we need to use the n-bit parallel adder. It uses a number of full adders in cascade. The carry output of the previous full adder is connected to carry input of the next full adder.
- 4-bit parallel adder is shown here.

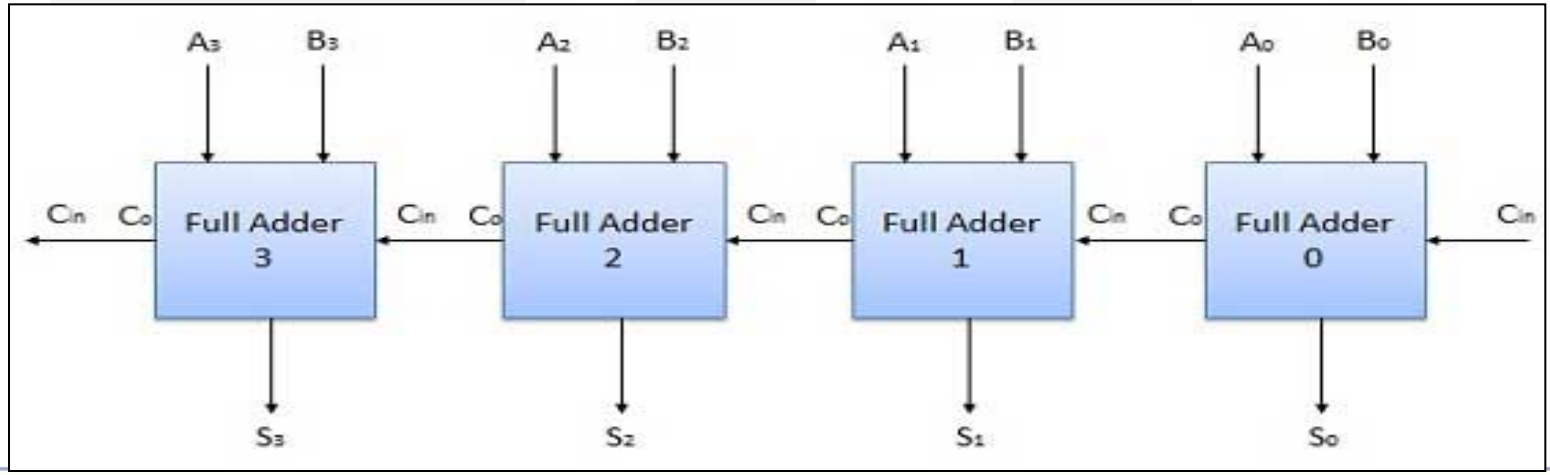


Image source : Google

4.6: N-Bit Parallel Subtractor

- To subtract one n-bit binary numbers from the other n-bit binary number, we need to use the n-bit parallel subtractor. It uses a number of full adders in cascade. But B is complemented then added to A.
- So, A + Complement of B = A-B
- 4-bit parallel subtractor is shown here.

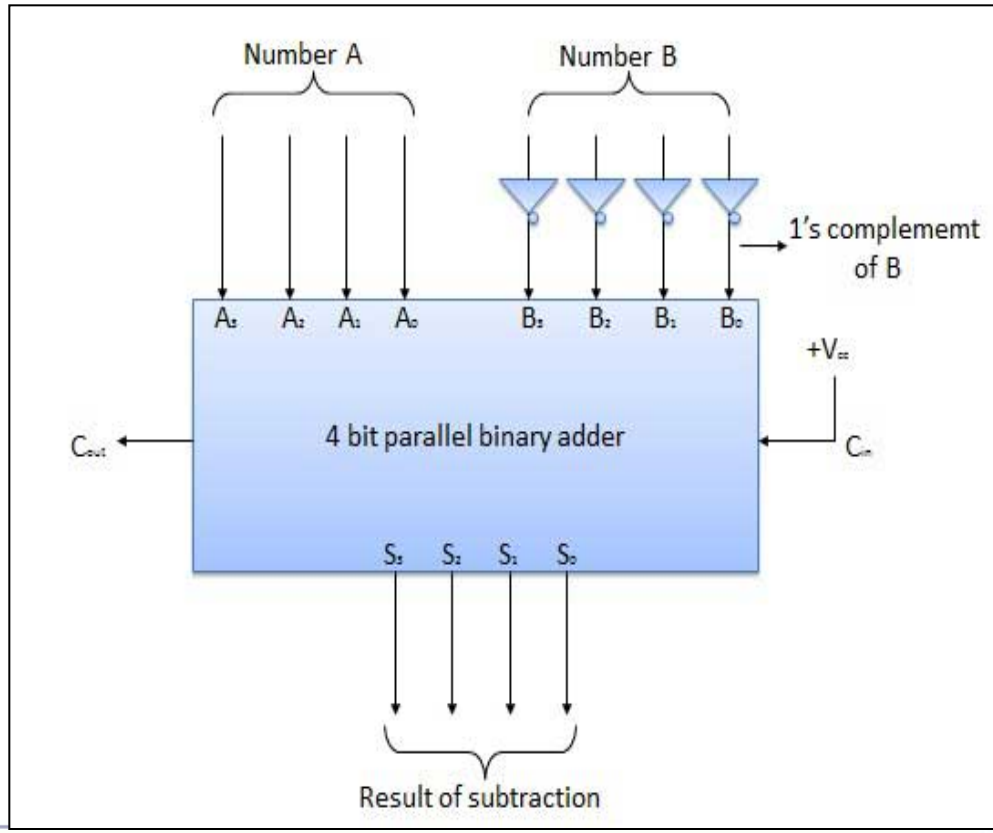


Image source : Google

4.7: Multiplexers

- A digital multiplexer 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 selection lines.
- 2" input lines □ n selection lines □ 1 output line

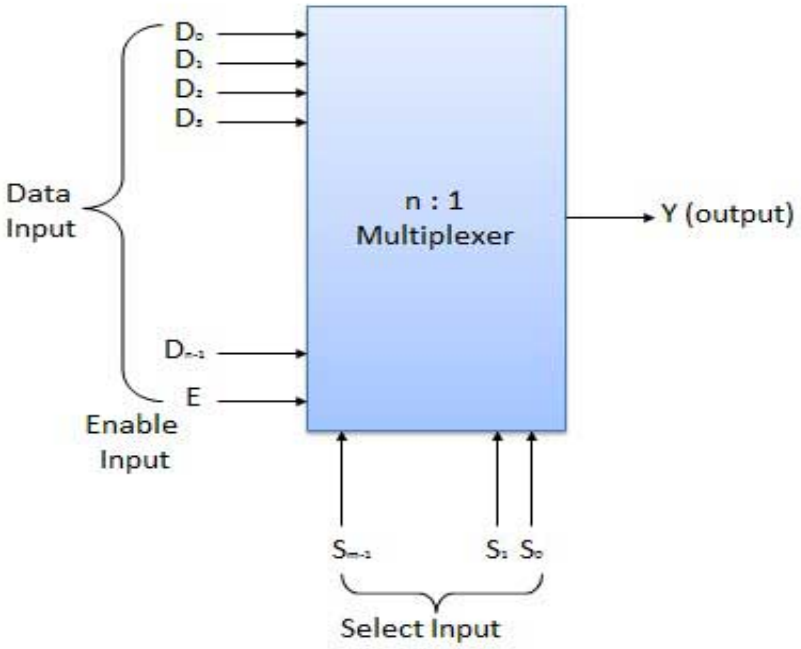


Image source : Google

4.7: Multiplexers (Continue..)

4 : 1 Multiplexer

- Truth table and block diagram of 4:1 multiplexer are shown here.
- From the truth table, the Boolean expression is derived as follows.
- $Y=S1'S0'I0+S1'S0I1+S1S0'I2+S1S0I3$

Selection lines		Output
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

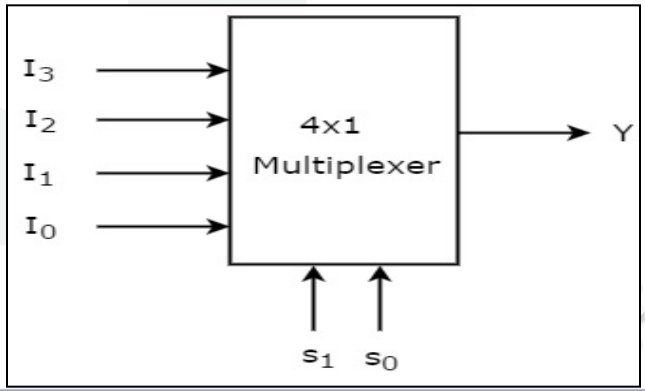


Image source : Google

4.8: De-Multiplexers

- A de-multiplexer performs the reverse operation of a multiplexer i.e. it receives one input and distributes it over several outputs.
- It has only one input, n outputs, m select input.
- De-multiplexer come in multiple variations:
  - 1 : 2 demultiplexer
  - 1 : 4 demultiplexer
  - 1 : 16 demultiplexer
  - 1 : 32 demultiplexer

4.8: De-Multiplexers

1:4 De-multiplexer

- 1x4 De-Multiplexer has one input I, two selection lines, s1 & s0 and four outputs Y3, Y2, Y1 &Y0.
- The block diagram or symbol of multiplexer is as shown here.

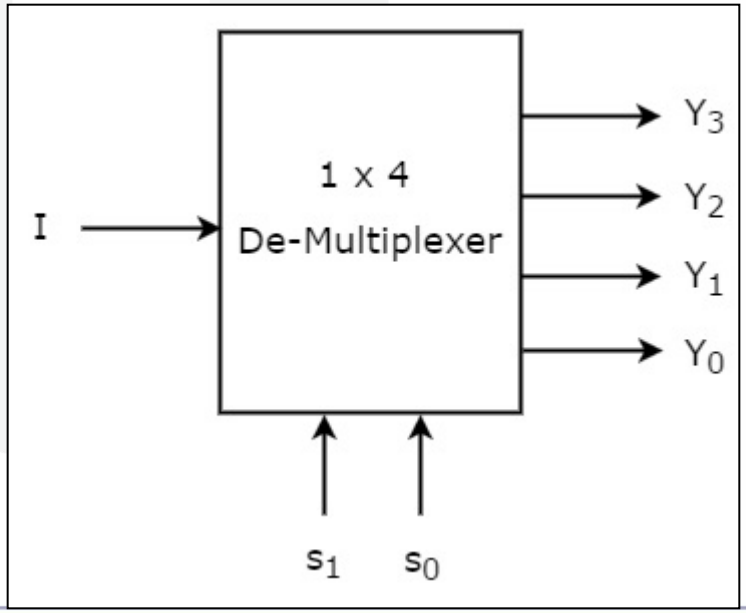


Image source : Google

4.8: De-Multiplexers (Continue..)

1:4 De-multiplexer (Continue..)

- From the above Truth table, we can directly write the Boolean functions for each output as
- Y3=s1s0I
- Y2=s1s0'I
- Y1=s1's0I
- Y0=s1's0'I

Selection Inputs		Outputs			
S_1	S_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	I
0	1	0	0	I	0
1	0	0	I	0	0
1	1	I	0	0	0

Image source : Google

4.8: De-Multiplexers (Continue..)

1:4 De-multiplexer (Continue..)

- Y3=s1s0I
- Y2=s1s0'I
- Y1=s1's0I
- Y0=s1's0'I

•From the above Boolean functions, we can design the circuit as shown here.

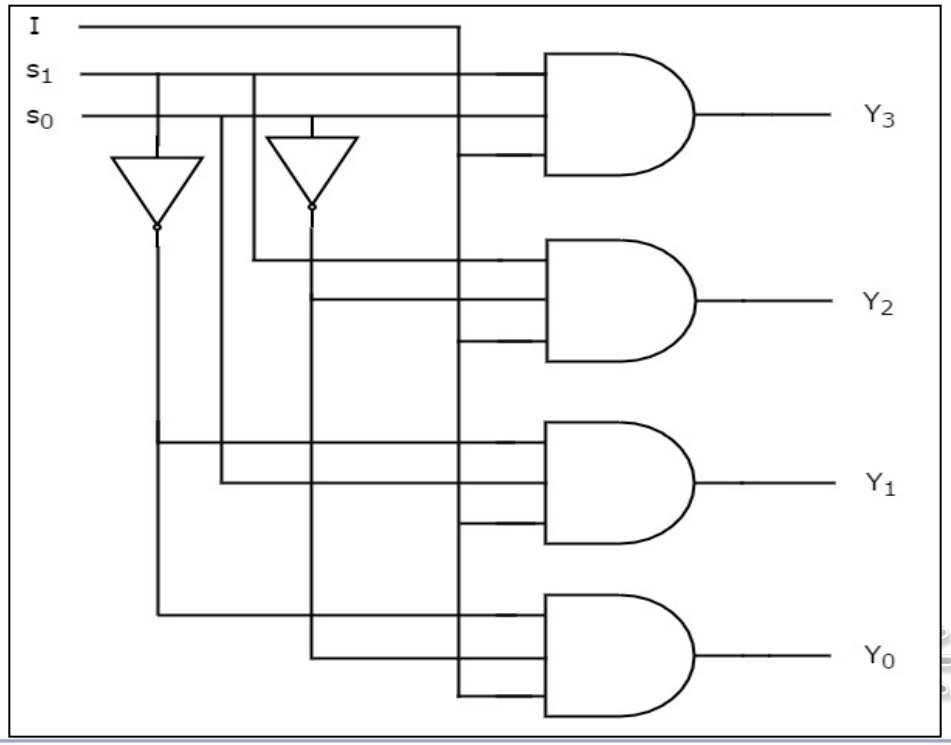


Image source : Google

4.9: Decoder

- A decoder is a combinational circuit that has n input and to a maximum m = 2n outputs. Decoder is identical to a de-multiplexer without any data input.

2 to 4 decoder

- The block diagram of 2 to 4 decoder is shown here.

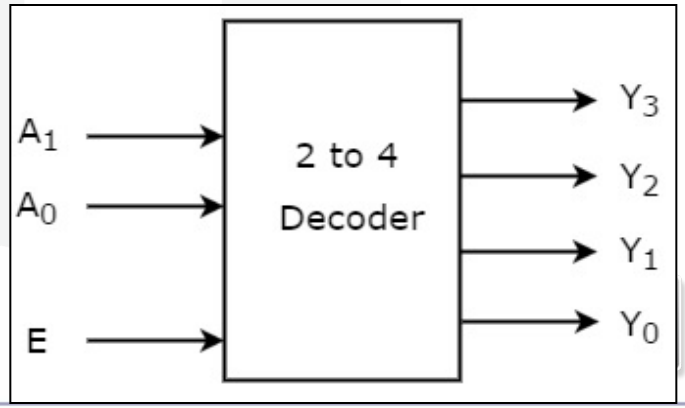


Image source : Google

4.9: Decoder (Continue..)

2 to 4 decoder (Continue..)

- The truth table of 2 to 4 decoder is shown here.
- From the truth table, the derived Boolean functions are:
  - Y3=E.A1.A0
  - Y2=E.A1.A0’
  - Y1=E.A1’.A0
  - Y0=E.A1’.A0’

Enable	Inputs		Outputs			
E	A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	x	X	0	0	0	0
1	0	X	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

Image source : Google

4.9: Decoder (Continue..)

2 to 4 decoder (Continue..)

- From the derived Boolean functions, the circuit of 2 to 4 decoder can be designed as shown here.
- Using similar method, 3 to 8 decoder can also be designed.

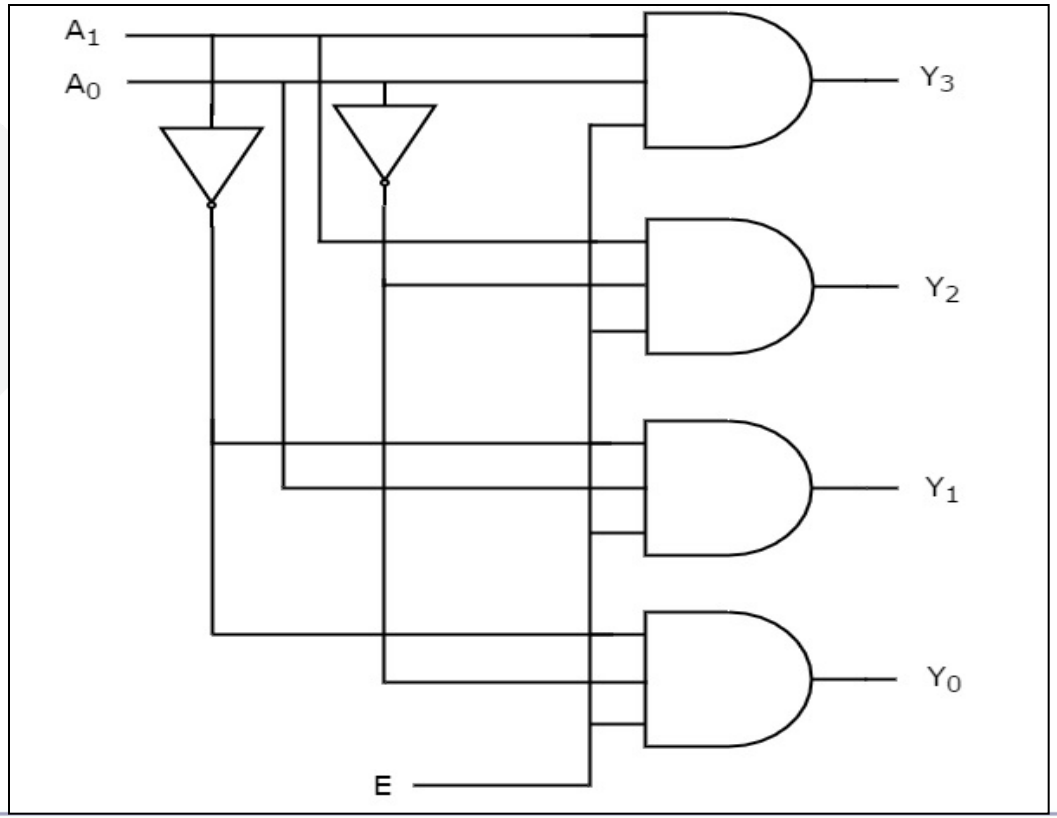


Image source : Google

4.10: Encoder

- An Encoder is a combinational circuit that performs the reverse operation of Decoder.
- It has maximum of 2n input lines and ‘n’ output lines. It will produce a binary code equivalent to the input, which is active High.

4 to 2 Encoder:

•The 4 to 2 Encoder has four inputs Y3, Y2, Y1 & Y0 and two outputs A1 & A0. The block diagram of 4 to 2 Encoder is shown here.

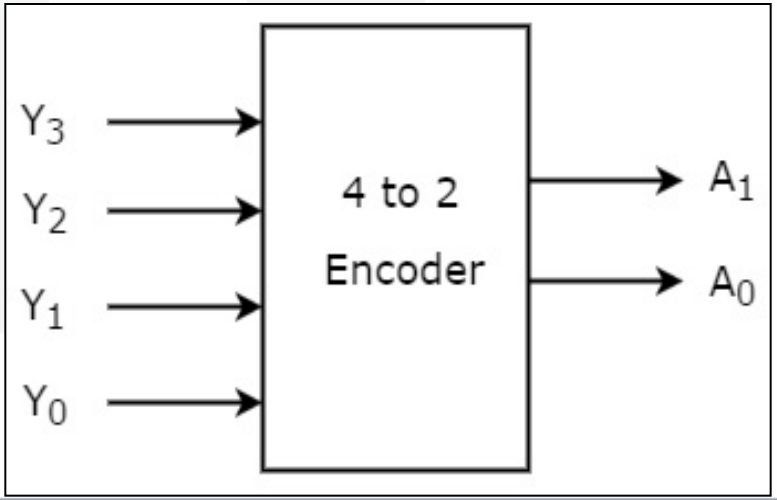


Image source : Google

4.10: Encoder (Continue..)

- The truth table of encoder is shown here.

Inputs				Outputs	
Y_3	Y_2	Y_1	Y_0	A_1	A_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

- The Boolean function for each output are as follows:

$A_1=Y_3+Y_2$

$A_0=Y_3+Y_1$

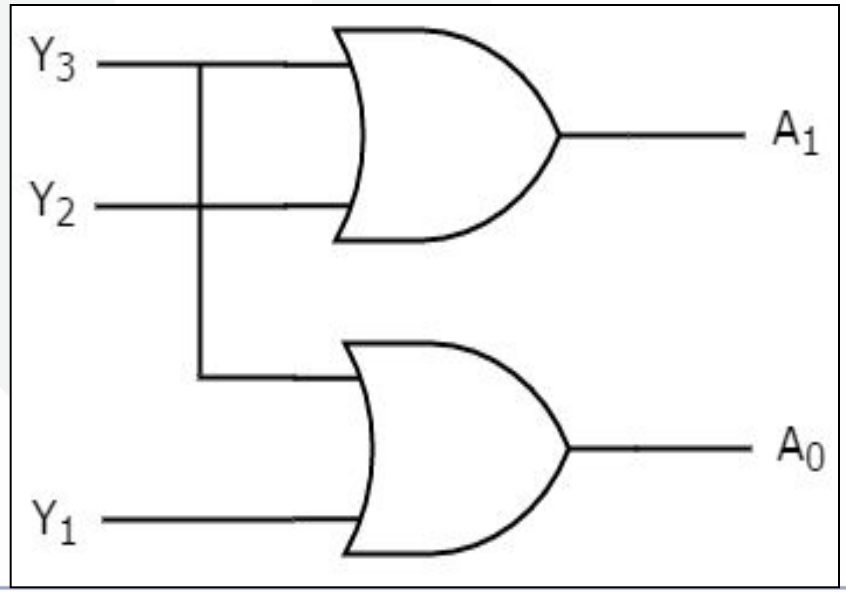


Image source : Google

4.11: Code-Converters

- The combinational circuits, which convert one code to other code are called as code converters.
- These circuits mainly consist of Logic gates.
- For example,

Binary code to Gray code converter

- It is the combinational circuit which is designed to converts a 4-bit binary code $B_3B_2B_1B_0$ into its equivalent Gray code $G_3G_2G_1G_0$.

4.11: Code-Converters (Continue..)

Binary code $B_2B_2B_1B_0$	Gray code $G_2G_2G_1G_0$
0000	0000
0001	0001
0010	0011
0011	0010
0100	0110
0101	0111
0110	0101
0111	0100
1000	1100
1001	1101
1010	1111
1011	1110
1100	1010
1101	1011
1110	1001
1111	1000

From Truth table, we can design Boolean expression using K-maps.

Image source : Google

4.11: Code-Converters (Continue..)

- From the truth table, k-map can be generated.
- From truth table and k-map, the simplified expression output digits are

$G_0=B_1'B_0+B_1B_0' \Rightarrow G_0=B_1 \oplus B_0$

G1=B1'B2+ B1B2'

G1=B1 ⊕B2

G2=B3'B2+ B3B2'

G2=B2 ⊕B3

And G3=B3

K-map for G_0

B_3B_2 \ B_1B_0	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	0	1	0	1
10	0	1	0	1

K-map for G_1

B_3B_2 \ B_1B_0	00	01	11	10
00	0	0	1	1
01	1	1	0	0
11	1	1	0	0
10	0	0	1	1

K-map for G_2

B_3B_2 \ B_1B_0	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	0	0	0	0
10	1	1	1	1

Image source : Google

4.11: Code-Converters (Continue..)

- From the above derived Boolean expressions, we can design the combinational circuit using logic gates as follows.

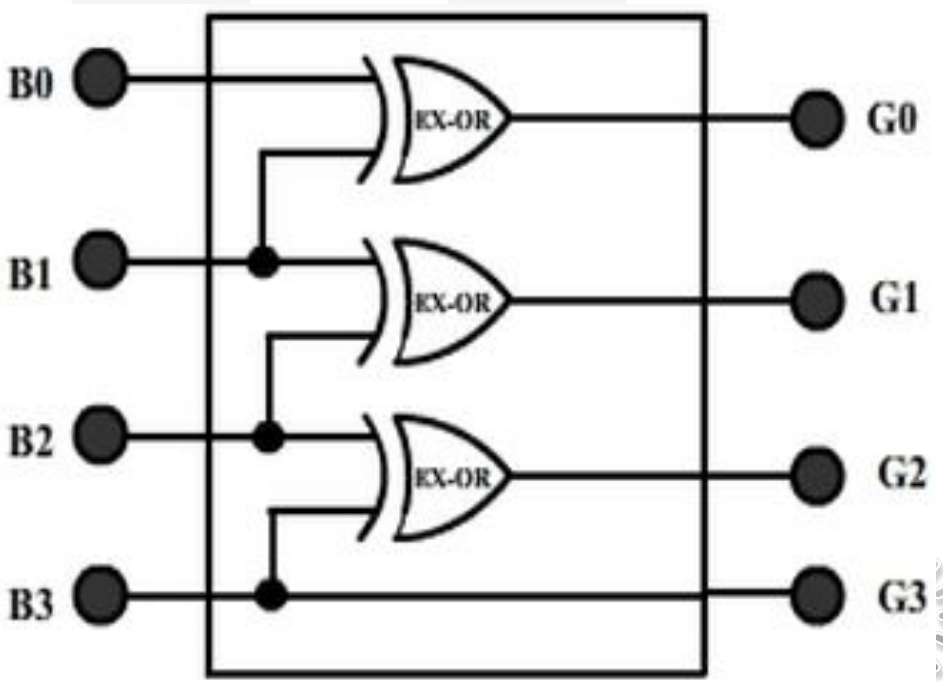
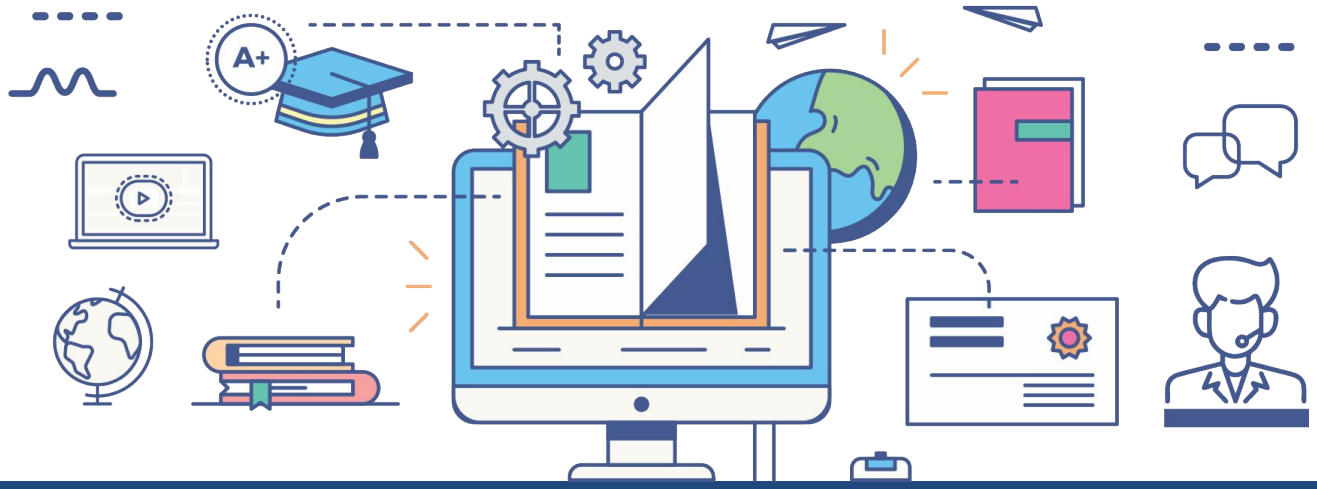


Image source : Google

DIGITAL LEARNING CONTENT



Parul® University

www.paruluniversity.ac.in