Digital Electronics
Moule-II

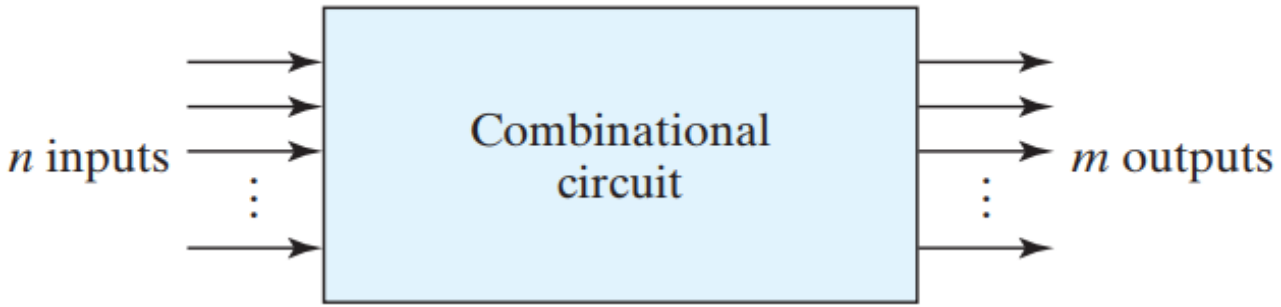
Ajith Kumar M M
Lecturer in electronics
MVGM, Government Polytechnic , vennikulam

Logic Circuits

- Logic circuits for digital systems may be combinational or sequential.
- A combinational circuit consists of logic gates whose outputs at any time are determined from only the present combination of inputs.
- A combinational circuit performs an operation that can be specified logically by a set of Boolean functions.
- Sequential circuits have storage elements in addition to logic gates.
- Their outputs are a function of the inputs and the state of the storage elements.
- Because the state of the storage elements is a function of previous inputs, the outputs of a sequential circuit depend not only on present values of inputs, but also on past inputs,

Combinational Circuits

- A combinational circuit consists of an interconnection of logic gates.
- Combinational logic gates react to the values of the signals at their inputs and produce the value of the output signal.
- Transforming binary information from the given input data to a required output data.



Half Adder

- Half Adder circuit needs two binary inputs and two binary outputs.
- The output variables produce the sum and carry.

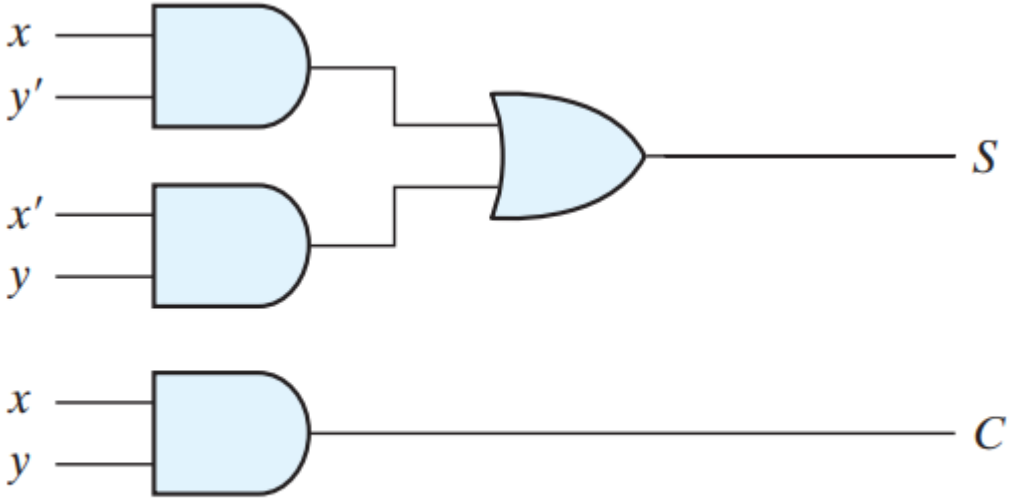
Half Adder

x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

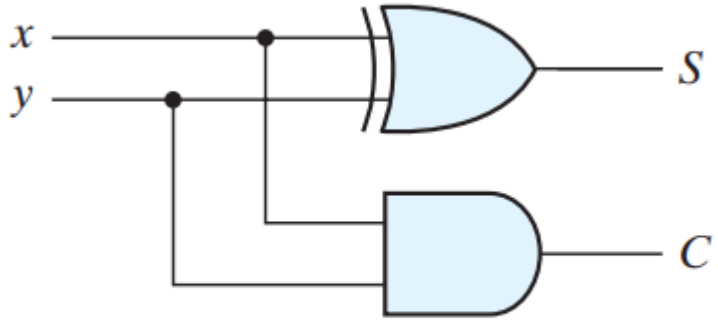
Half Adder

Half Adder

x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



(a) $S = xy' + x'y$
$C = xy$



(b) $S = x \oplus y$
$C = xy$

Full Adder

- Addition of n-bit binary numbers requires the use of a full adder,
- The process of addition proceeds on a bit-by-bit basis, right to left, beginning with the least significant bit.
- After the least significant bit, addition at each position adds not only the respective bits of the words, but must also consider a possible carry bit from addition at the previous position

1 1	carry
0 0 1 1 0 1 0	$= 26_{10}$
+0 0 0 1 1 0 0	$= 12_{10}$
0 1 0 0 1 1 0	$= 38_{10}$

Full Adder

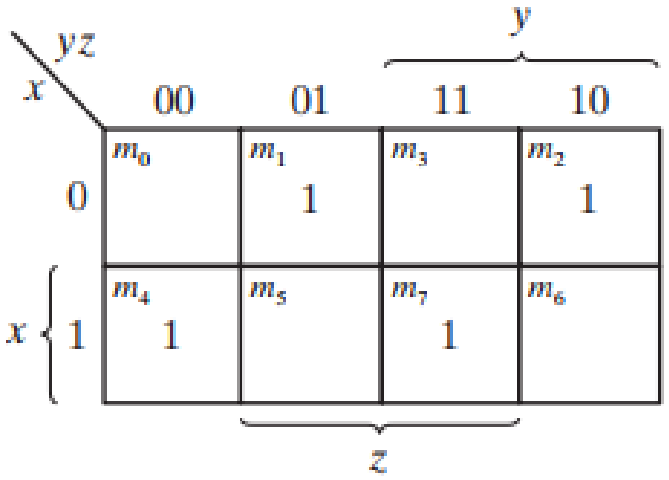
Full Adder

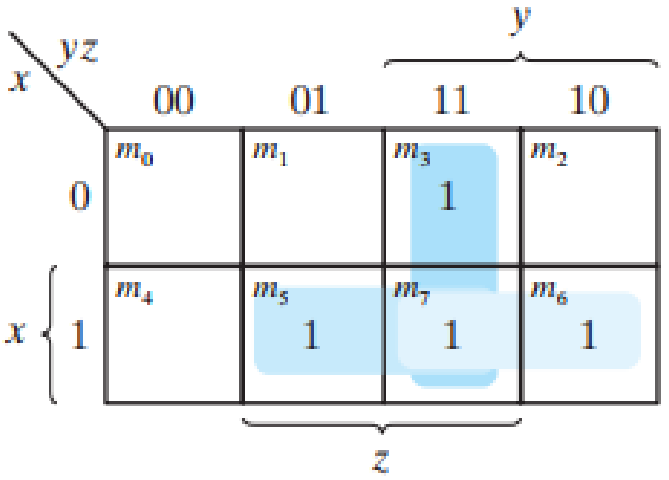
x	y	z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Full Adder

Full Adder

x	y	z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



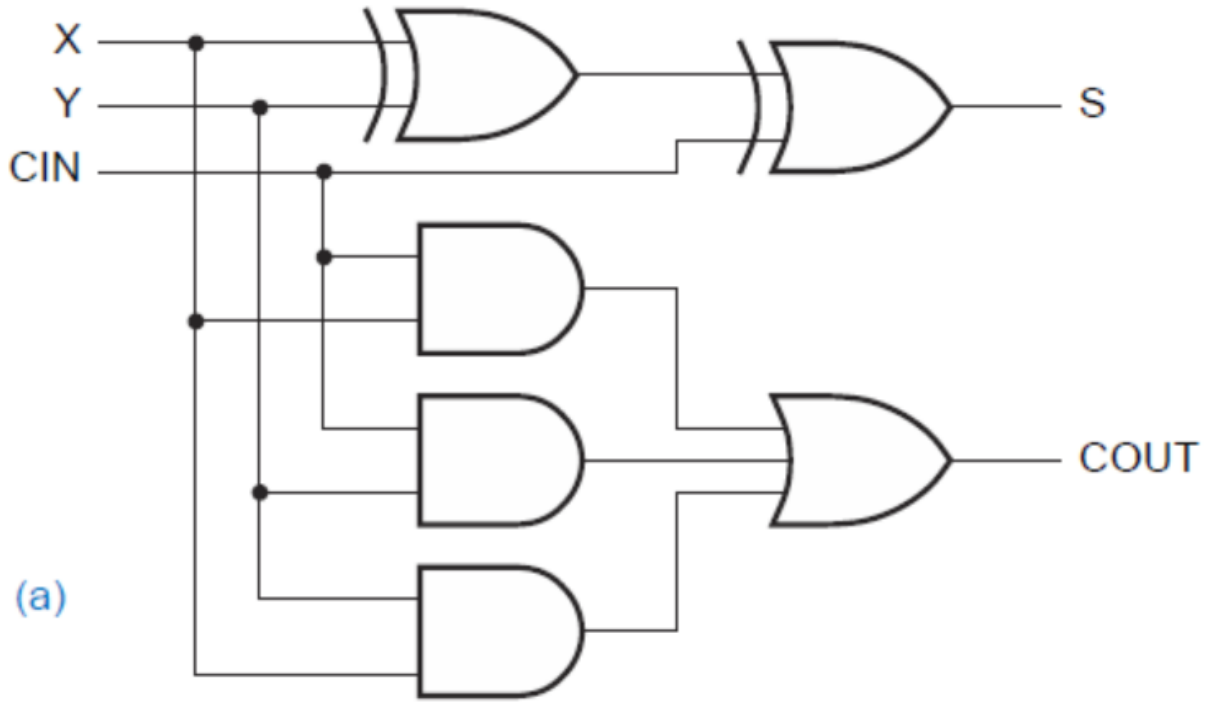


$$S = \bar{x}\bar{y}z + \bar{x}\,y\bar{z} + x\bar{y}\bar{z} + xyz$$

$$S = x \oplus y \oplus z$$

$$C = xy + yz + xz$$

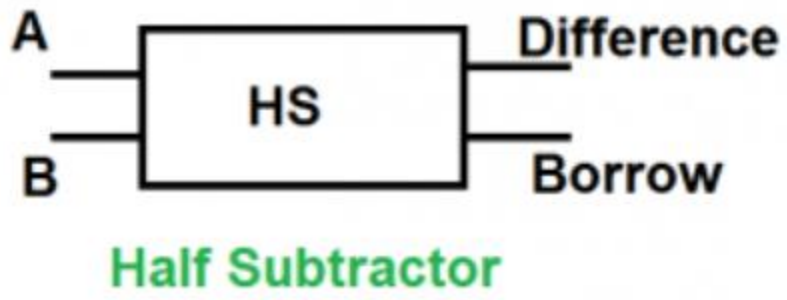
Full Adder



$$S = x \oplus y \oplus z$$

$$C = xy + yz + xz$$

Half Subtractor



Half Subtractor

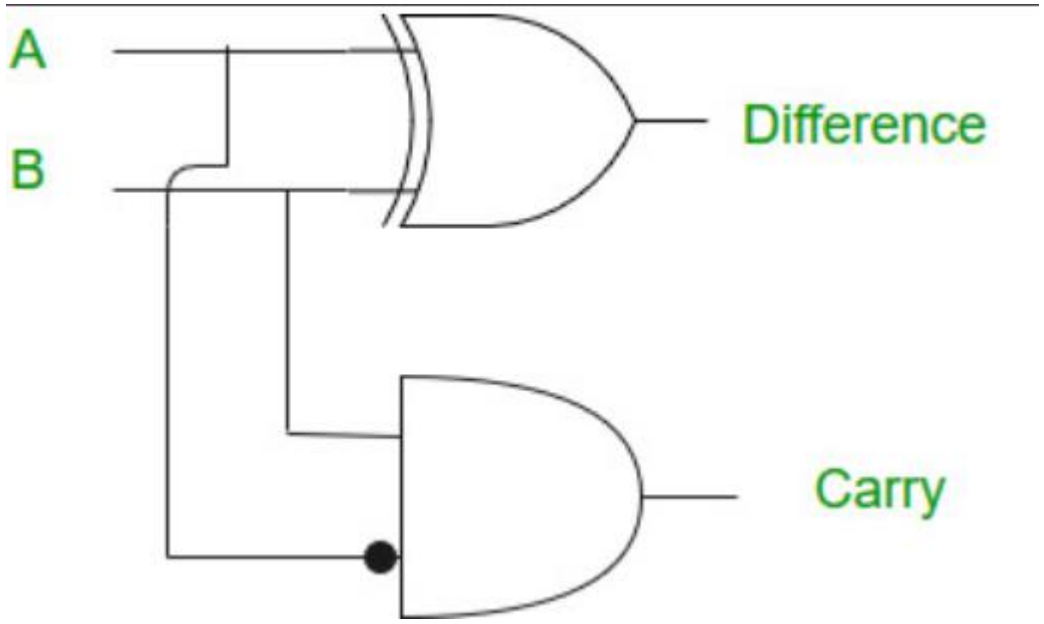
A	B	Diff	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$$Difference = A \oplus B$$

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

Half Subtractor

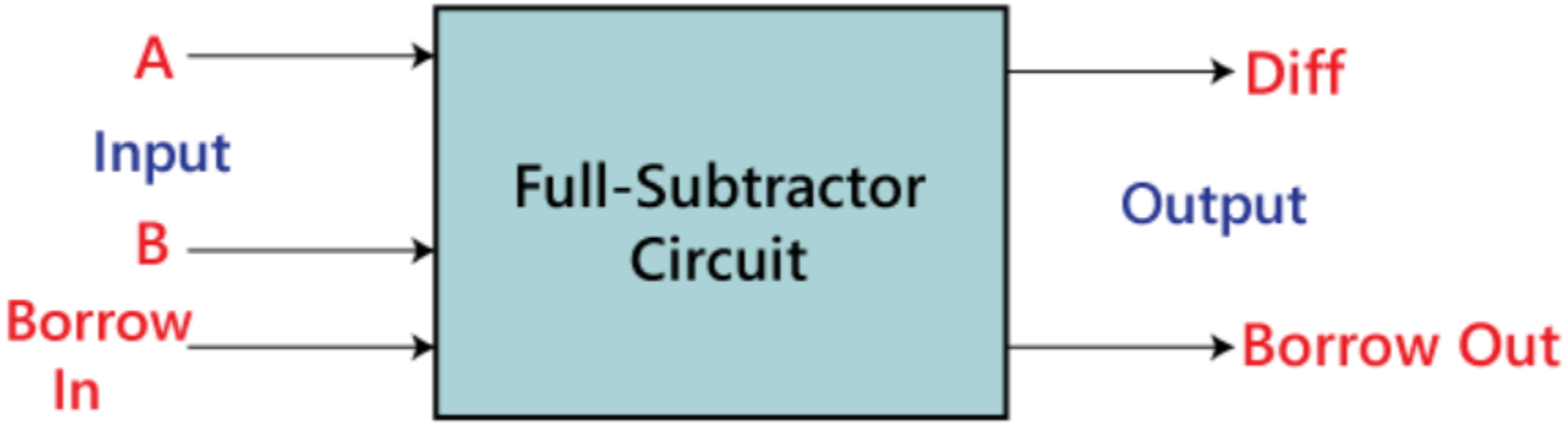
A	B	Diff	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0



$$Difference = A \oplus B$$

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

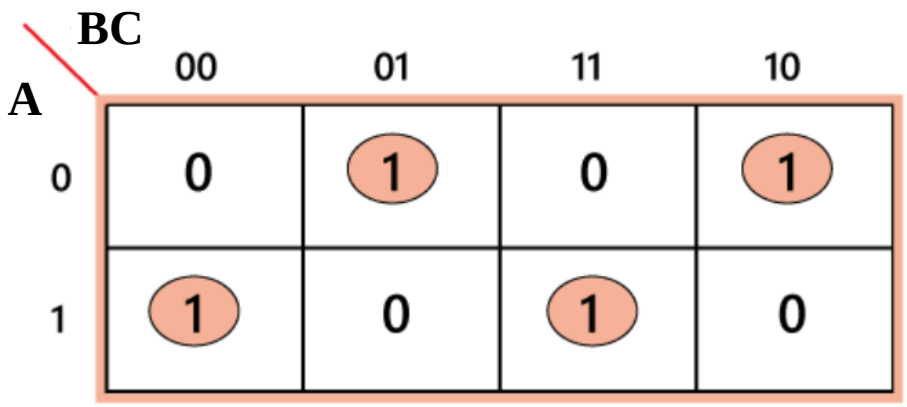
Full Subtractor



Full Subtractor

Inputs			Outputs	
A	B	$Borrow_{in}$	Diff	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Full Subtractor

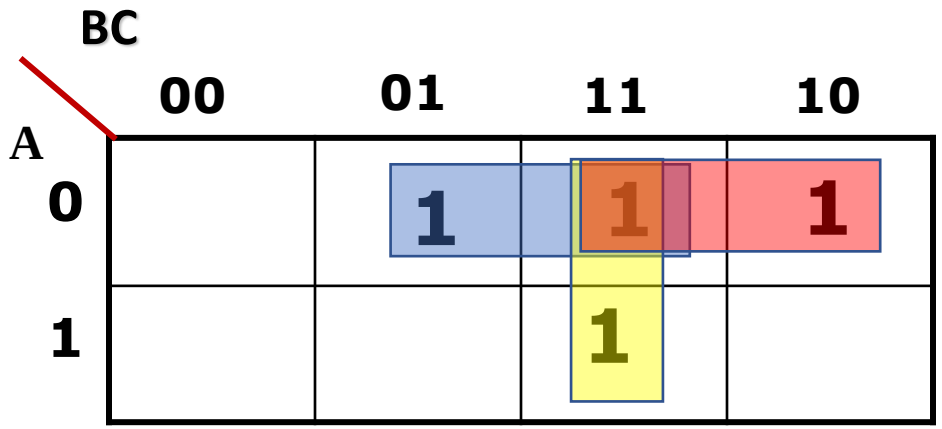


Inputs			Outputs	
A	B	$Borrow_{in}$	Diff	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$$Diff = A\bar{B}\bar{C} + \bar{A}\bar{B}C + ABC + \bar{A}B\bar{C}$$

$$Diff = A \oplus \mathrm{B} \oplus \mathrm{C}$$

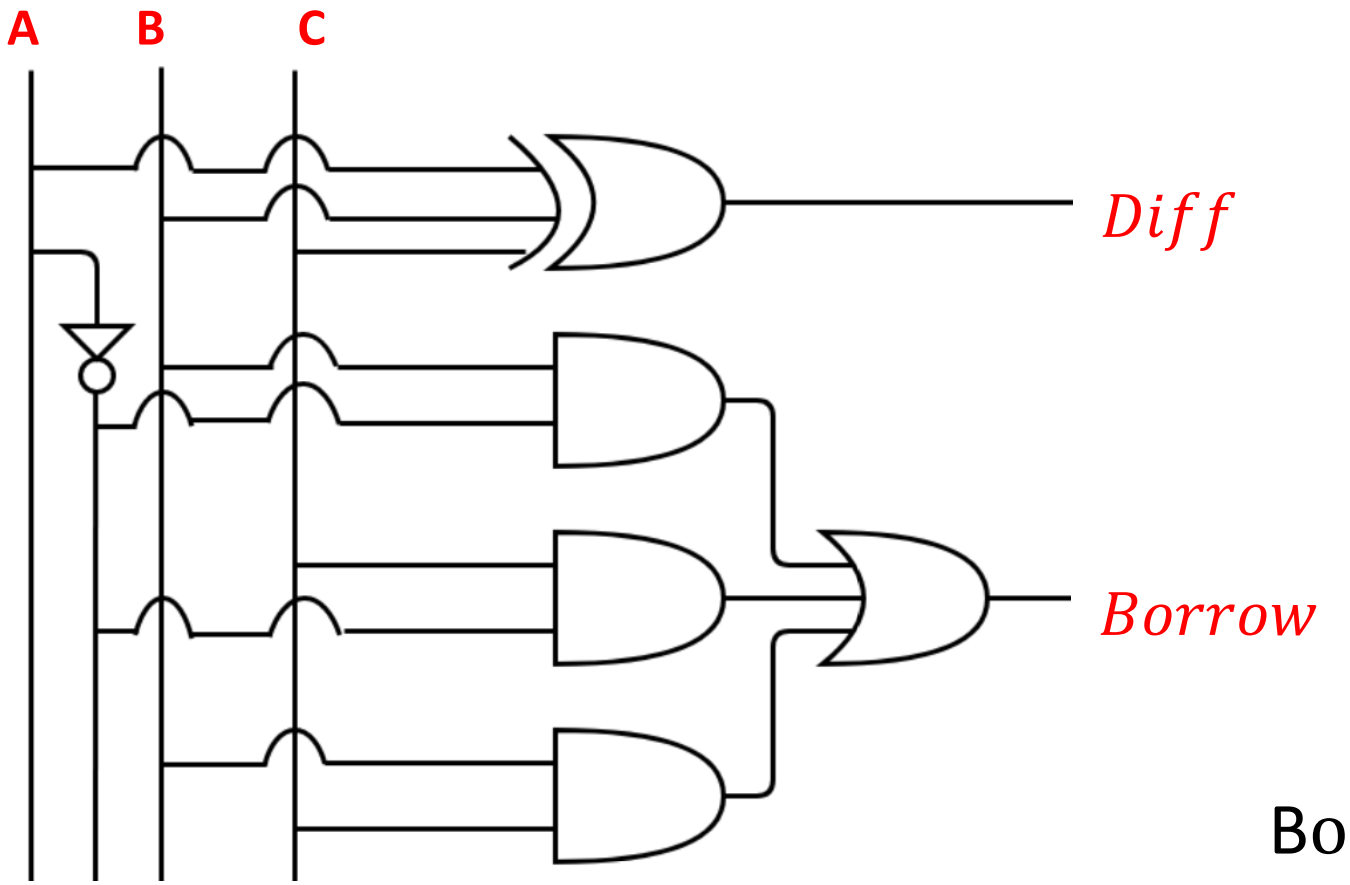
Full Subtractor



Inputs			Outputs	
A	B	$Borrow_{in}$	Diff	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$$\text{Borrow} = \bar{A}C + BC + \bar{A}B$$

Full Subtractor



$$Diff = A \oplus \mathrm{B} \oplus \mathrm{C}$$

$$\mathrm{Borrow} = \bar{A}C + BC + \bar{A}\mathrm{B}$$

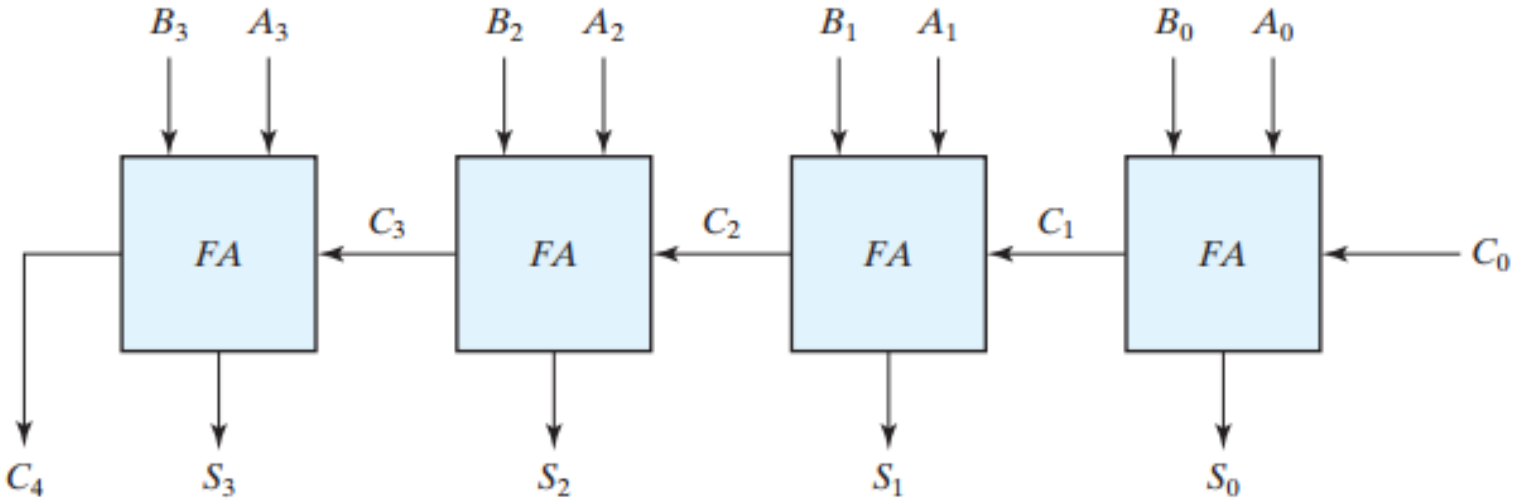
Parallel Adder

- A single full adder performs the addition of two one-bit numbers and an input carry.
- A **Parallel Adder** is a digital circuit capable of finding the arithmetic **sum** of two binary numbers that is **greater than one bit** in length

```
Carry     1   1 1 0
              1 0 1 1      +
              0 1 1 0
            ---------

              0 0 0 1
```

Four bit Parallel Adder



Carry 1 1 1 0

1 0 1 1 +

0 1 1 0

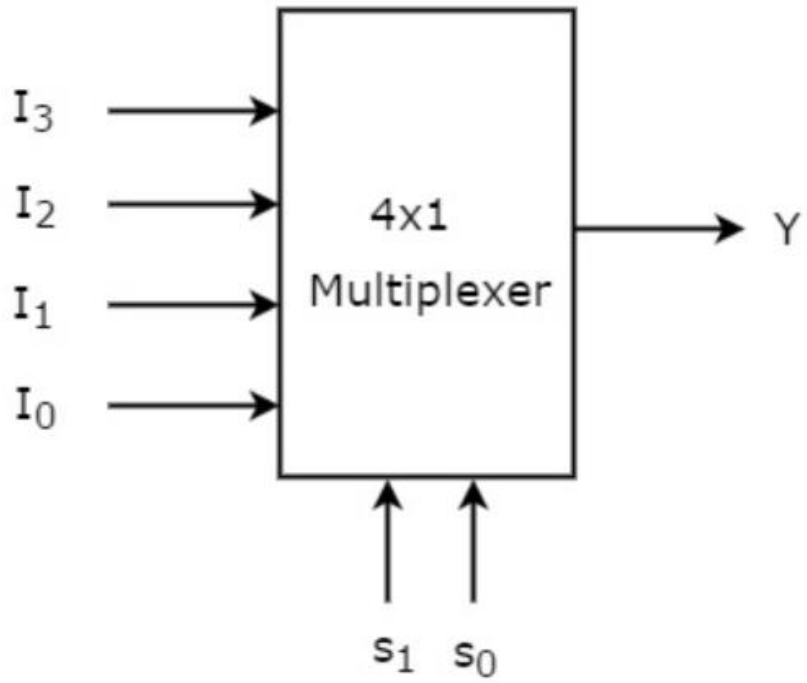
0 0 0 1

Multiplexer

- **Multiplexer** is a combinational circuit that has maximum of 2^n data inputs, ‘n’ selection lines and single output line.
- One of these data inputs will be connected to the output based on the values of selection lines.
- Since there are ‘n’ selection lines, there will be 2^n possible combinations of zeros and ones. So, each combination will select only one data input.
- Multiplexer is also called as **Mux**

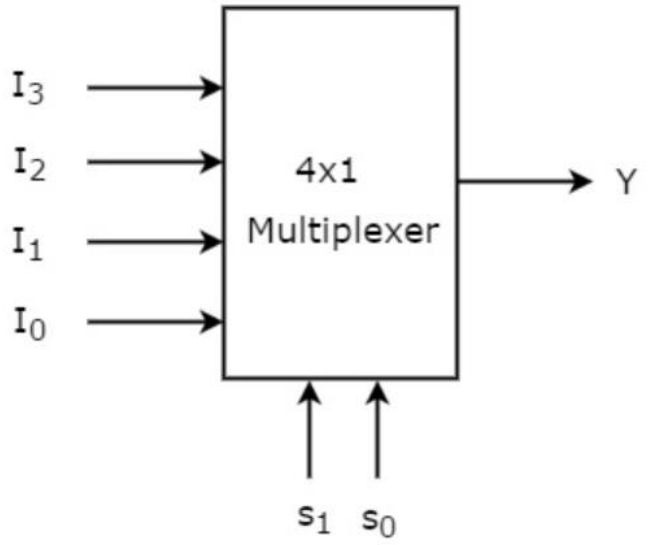
4x1 Multiplexer

- 4x1 Multiplexer has four data inputs I_3, I_2, I_1 & I_0, two selection lines s_1 & s_0 and one output Y.
- One of these 4 inputs will be connected to the output based on the combination of inputs present at these two selection lines.



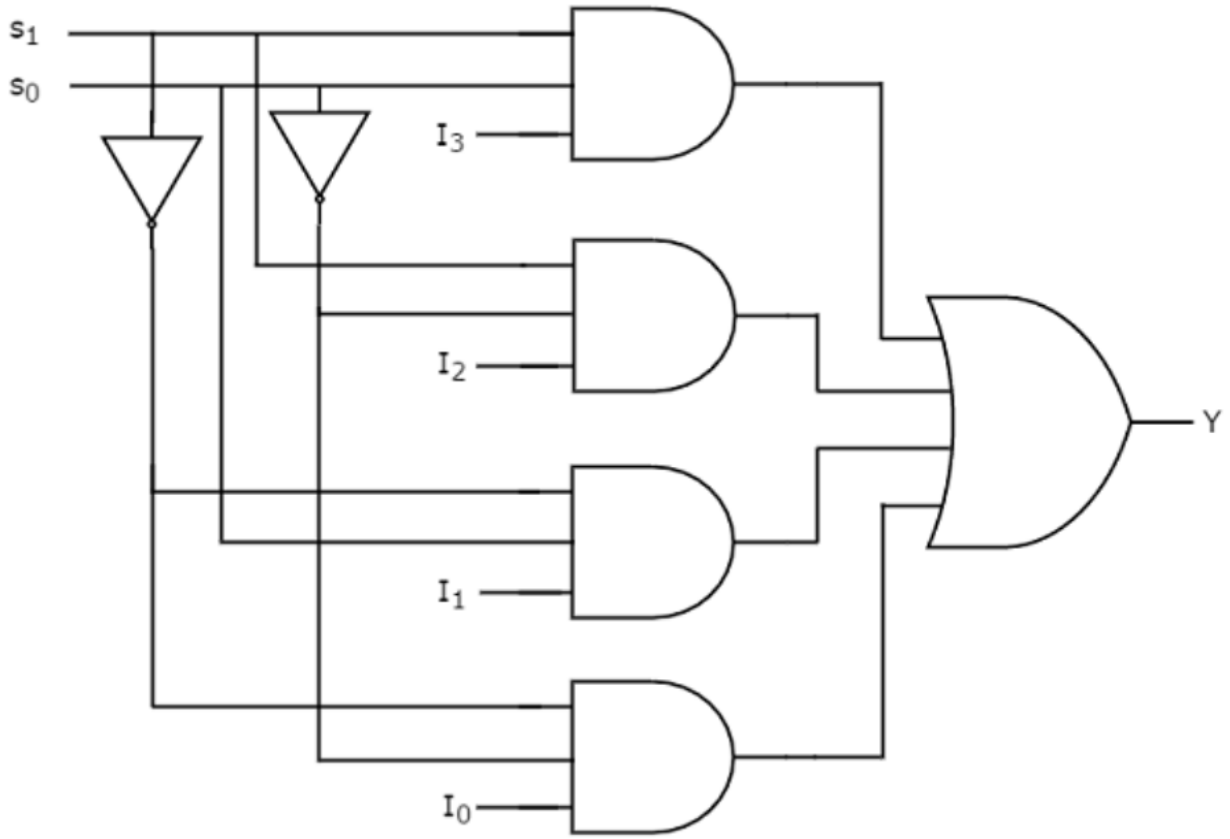
4x1 Multiplexer

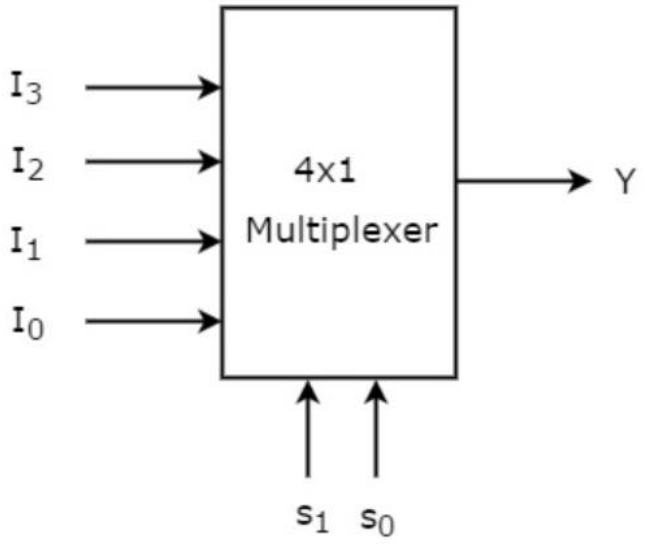
Selection Lines		Output
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3



$$Y = \overline{S_1}\,\overline{S_0} I_0 + \overline{S_1} S_0 I_1 + S_1 \overline{S_0} I_2 + S_1 S_0 I_3$$

4x1 Multiplexer

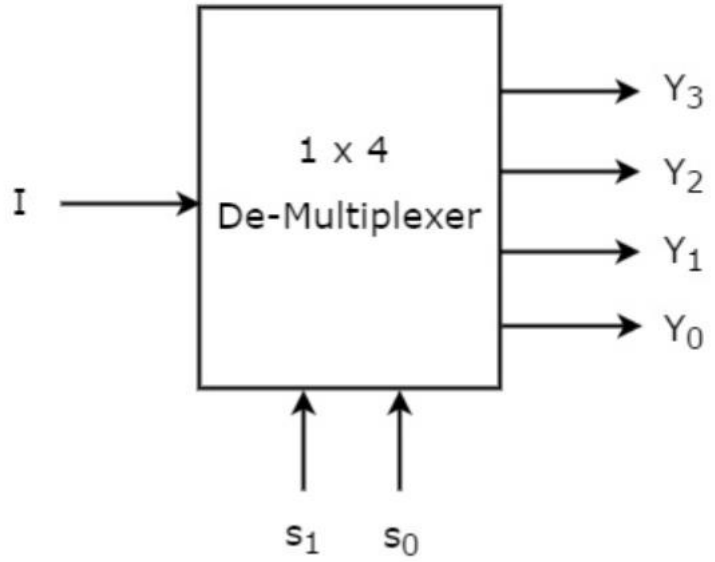




$$Y = \overline{S_1}\,\overline{S_0} I_0 + \overline{S_1} S_0 I_1 + S_1 \overline{S_0} I_2 + S_1 S_0 I_3$$

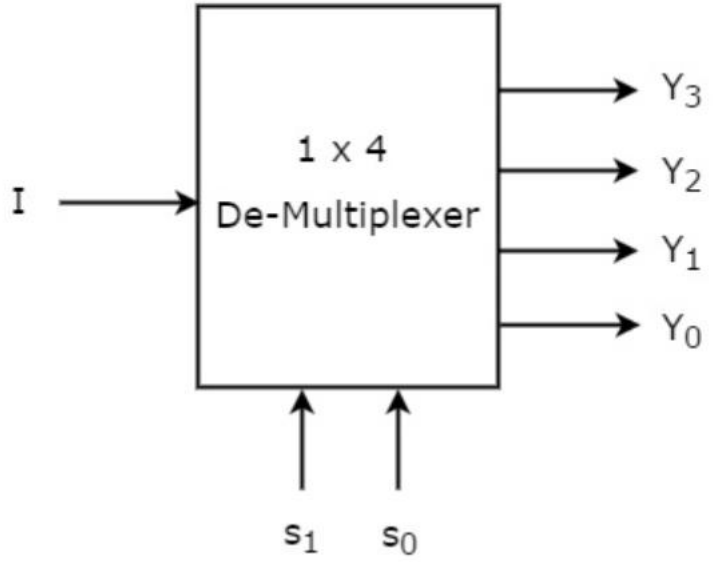
De-Multiplexer

- **De-Multiplexer** is a combinational circuit that performs the reverse operation of Multiplexer.
- It has single input, ‘n’ selection lines and maximum of 2^n outputs.
- The input will be connected to one of these outputs based on the values of selection line.
- De-Multiplexer is also called as **De-Mux**.



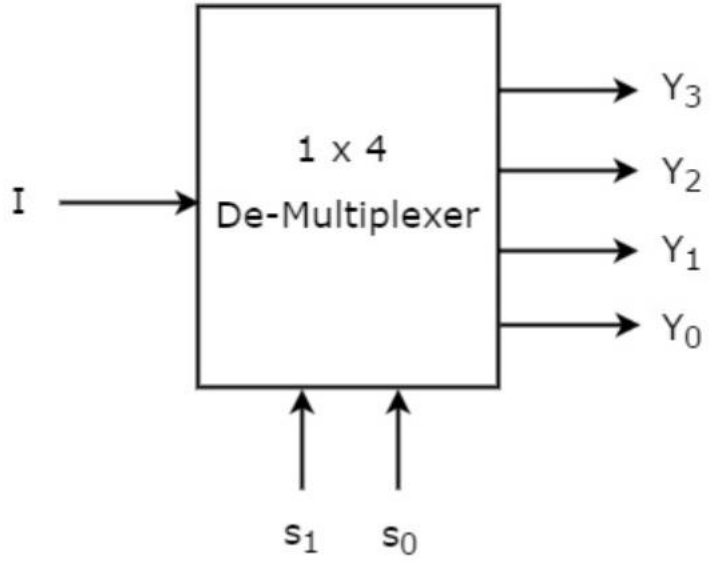
De-Multiplexer

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



De-Multiplexer

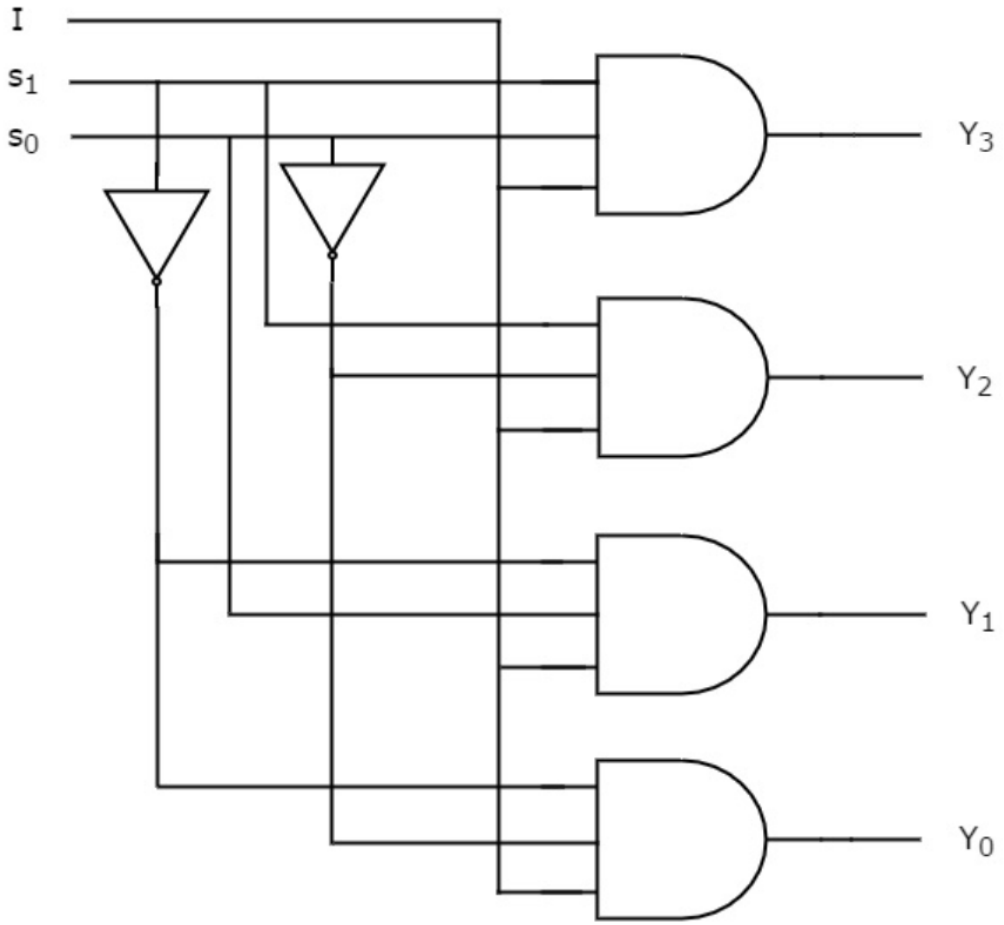
Selection Inputs		Outputs			
S_1	S_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	I
0	1	0	0	I	0
1	0	0	I	0	0
1	1	I	0	0	0



$$Y_0 = \overline{S_1}\,\overline{S_0}I \qquad Y_1 = \overline{S_1}S_0I$$

$$Y_2 = S_1\overline{S_0}I \qquad Y_3 = S_1S_0I$$

De-Multiplexer



$$Y_3 = S_1 S_0 I$$

$$Y_2 = S_1 \overline{S_0} I$$

$$Y_1 = \overline{S_1} S_0 I$$

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