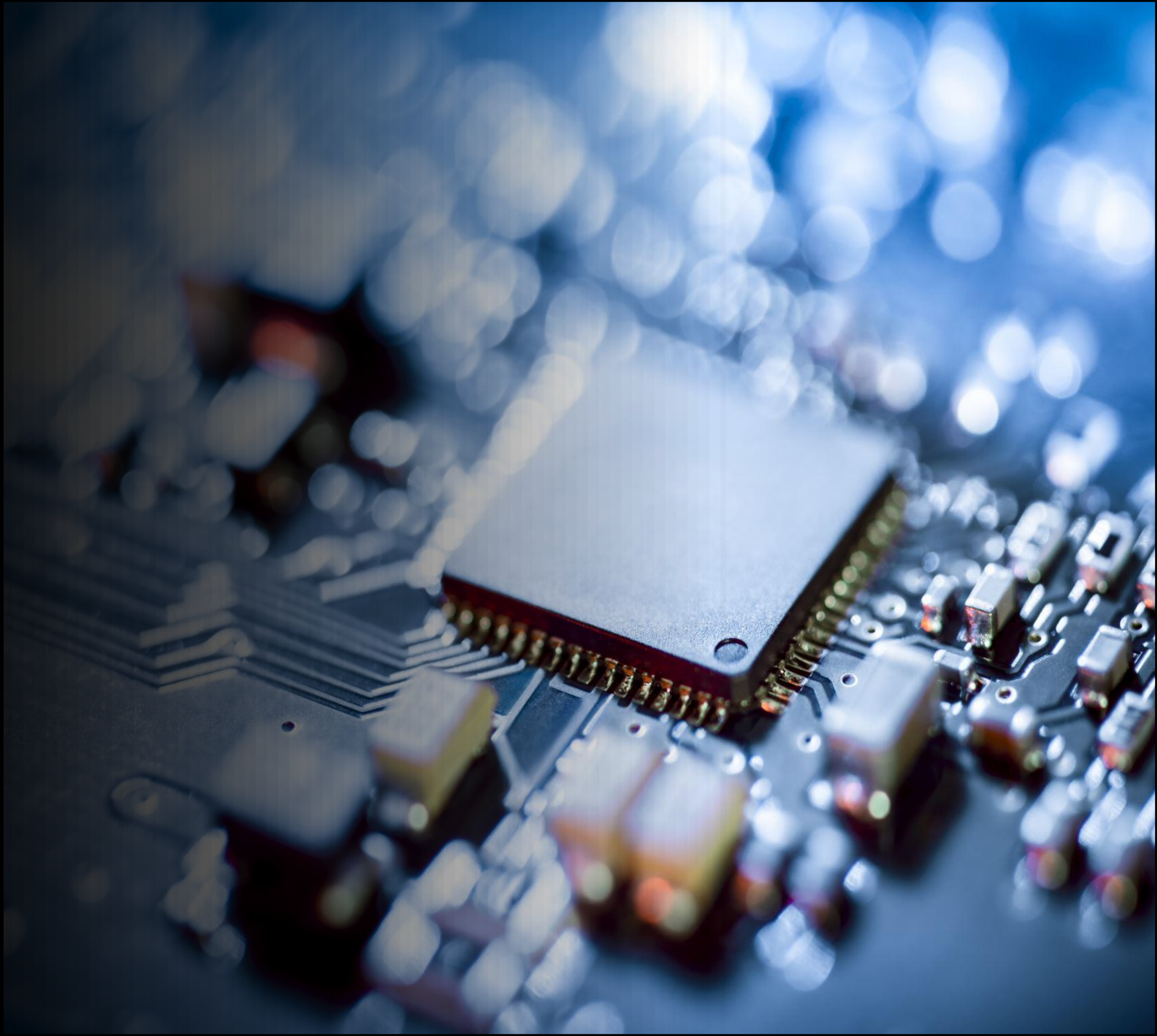
COMBINATIONAL CIRCUITS

DEEPA MATHEWS

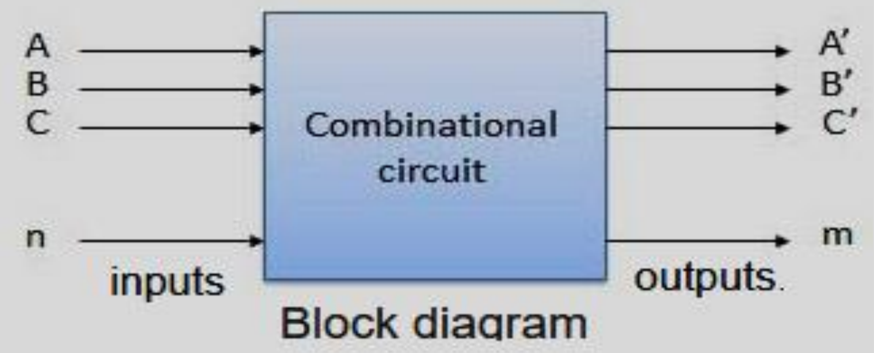


Combinational Circuits

- Unlike Sequential Logic Circuits whose outputs are dependant on both their present inputs and their previous output state, in a Combinational Logic Circuit, the output is dependant at all times on the combination of its present inputs.
- A **combinational circuit** comprises of logic gates whose outputs at any time are determined directly from the present combination of inputs without any regard to previous inputs.
- A combinational circuit is memoryless.
- The outputs of **Combinational Logic Circuits** are only determined by the logical function of their current input state, logic "0" or logic "1", at any given instant in time.

COMBINATIONAL CIRCUITS

- The basic components of a combinational circuit are: **input variables, logic gates, and output variables.**
- The logic gates accept signals from the inputs and generate signals to the outputs. This process transforms binary information from the given input data to a required output data.
- The 'n' input variables come from an external source whereas the 'm' output variables go to an external destination and the source or destination are storage registers.
- **If registers are included with the combinational gates, then the circuit becomes a sequential circuit.**



COMBINATIONAL CIRCUITS

Three main ways of specifying function of a combinational logic circuit are:

1. **Boolean Algebra** – This forms the algebraic expression showing the operation of the logic circuit for each input variable either True or False that results in a logic "1" output.

2. **Truth Table** – defines the function of a logic gate by providing a concise list that shows all the output states in tabular form for each possible combination of input variable that the gate could encounter.

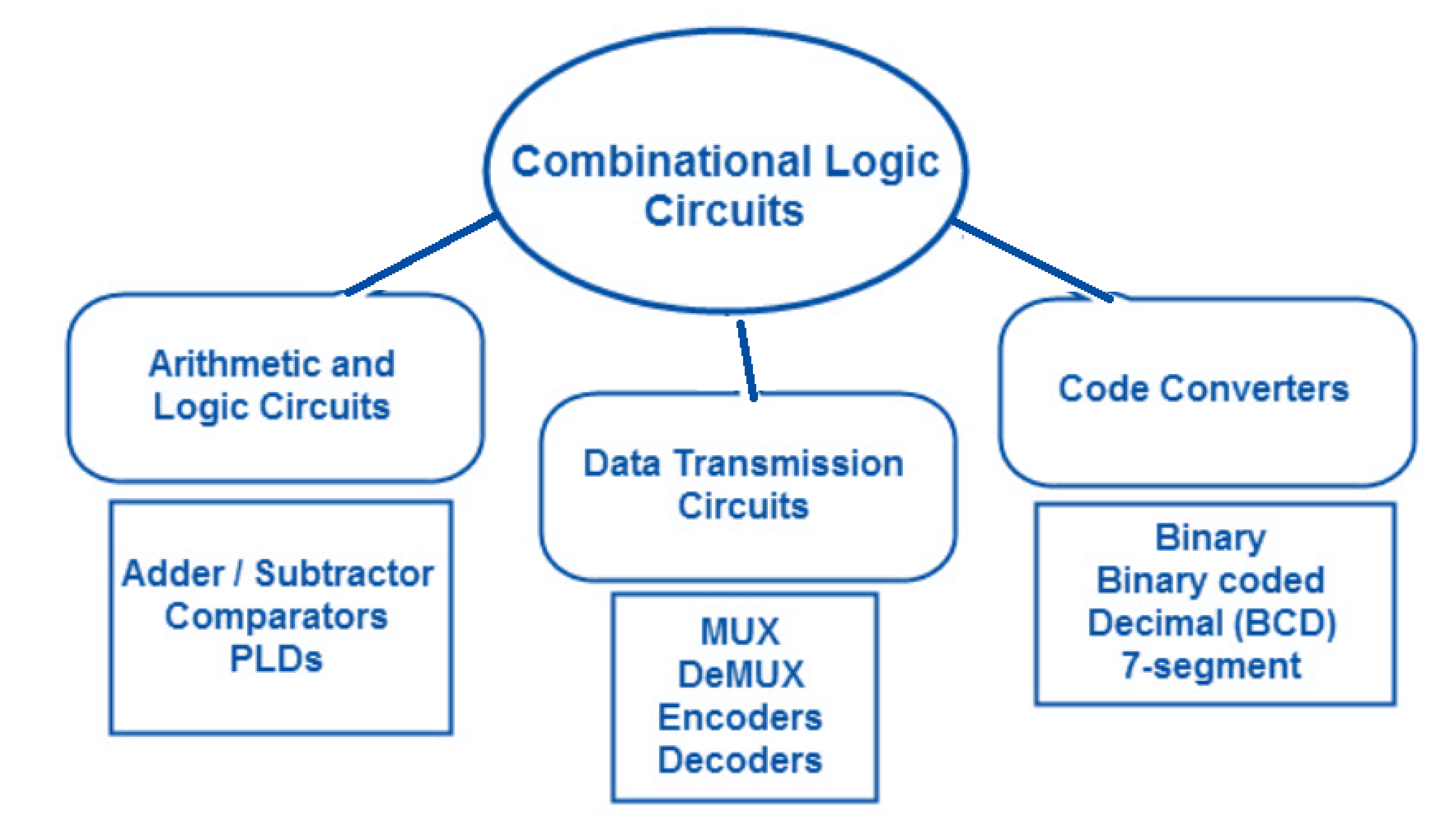
3. **Logic Diagram** – a graphical representation of a logic circuit that shows the wiring and connections of each individual logic gate, represented by a specific graphical symbol, that implements the logic circuit.

COMBINATIONAL CIRCUITS – DESIGN PROCEDURE

The **design procedure of a combinational circuit** involves the following steps:

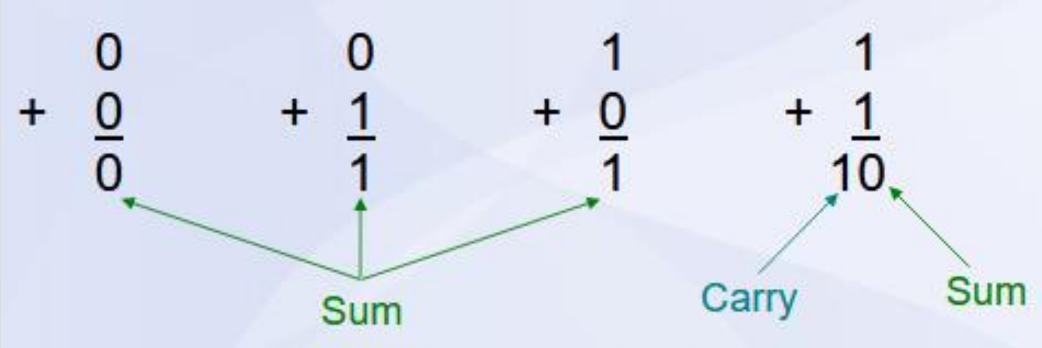
1. The problem is stated.
2. The total number of available input variables and required output variables is determined.
3. The truth table that defines the required relationships between inputs and outputs is derived.
4. The simplified Boolean function from each output is obtained.
5. Obtaining the logic diagram by the implementation of minimized Boolean expressions.

COMBINATIONAL CIRCUITS - CLASSIFICATION



ADDER

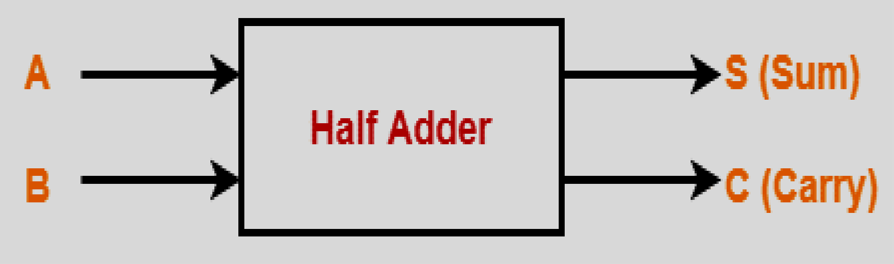
- The combinational circuit that performs the **addition** is called an adder circuit
- Adders are used in the arithmetic logic unit (ALU), and also in other parts of the processor, where they are used to calculate addresses, table indices, and similar operations.
- **Two Types - Half Adder and Full Adder**



Binary Addition

Half Adder

- Half Adder is a combinational logic circuit used for the purpose of adding **two** single bit numbers.



- The input variables –**addend & augend bits** are added together and produces the output variables - **sum and carry**.

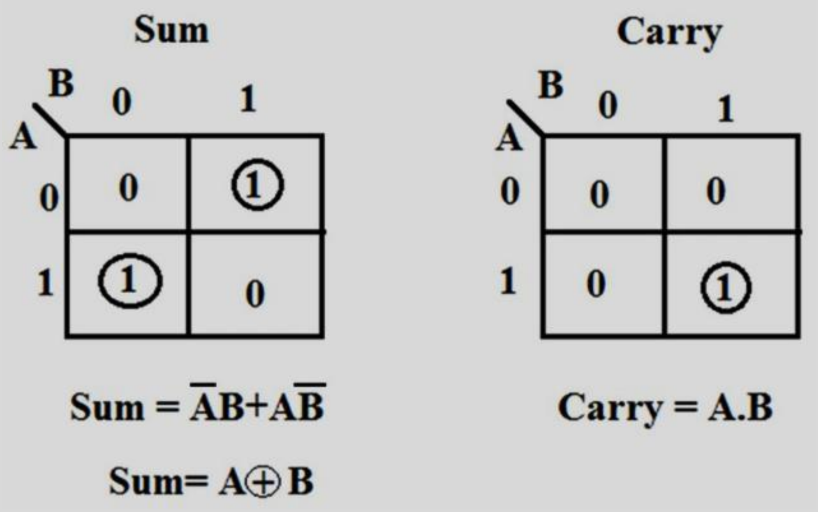
  Four possible operations:

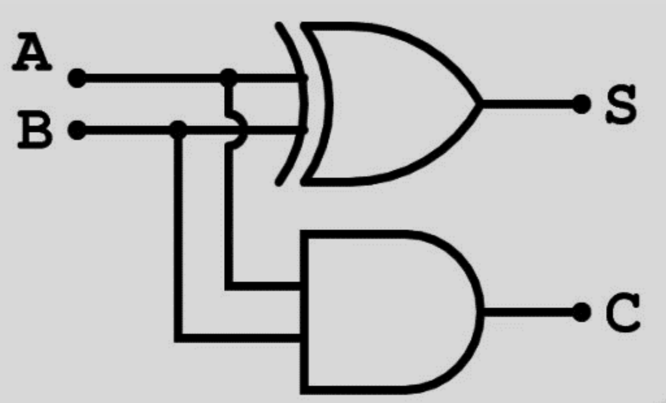
A	0	0	1	1
+ B	+ 0	+ 1	+ 0	+ 1
C S	0 0	0 1	0 1	1 0

Half Adder

Truth Table

Inputs		Outputs	
A	B	S	C-out
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



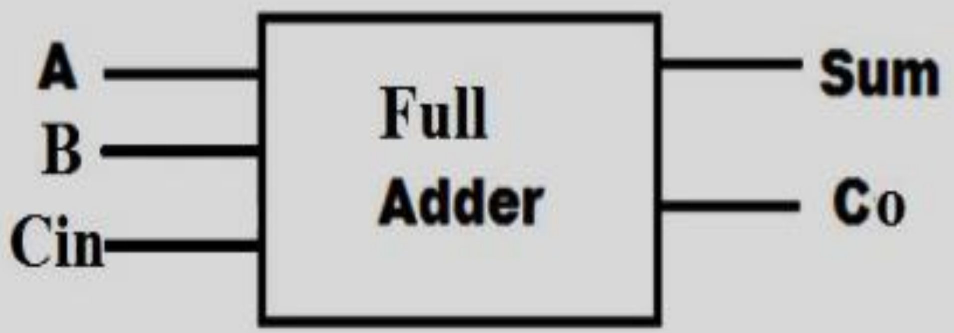


Limitation of Half Adder-

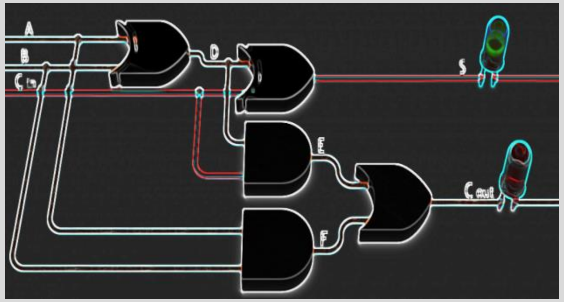
- Half adders have no scope of adding the carry bit resulting from the addition of previous bits. This is a major drawback of half adders.

Full Adder

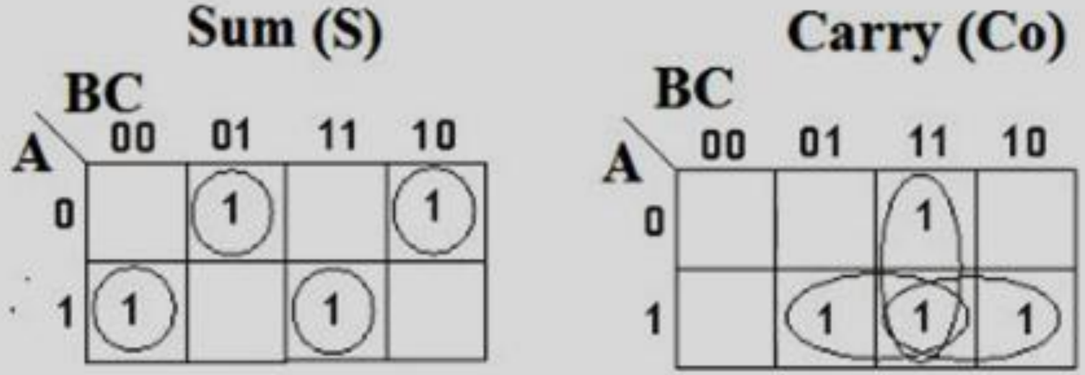
- Combinational circuit that is used for higher order bit addition.
- It adds three binary digits, among which two are the inputs (addend & augend), and one is the carry obtained from previous lower significant position.
- It generates two outputs – Sum and Carry

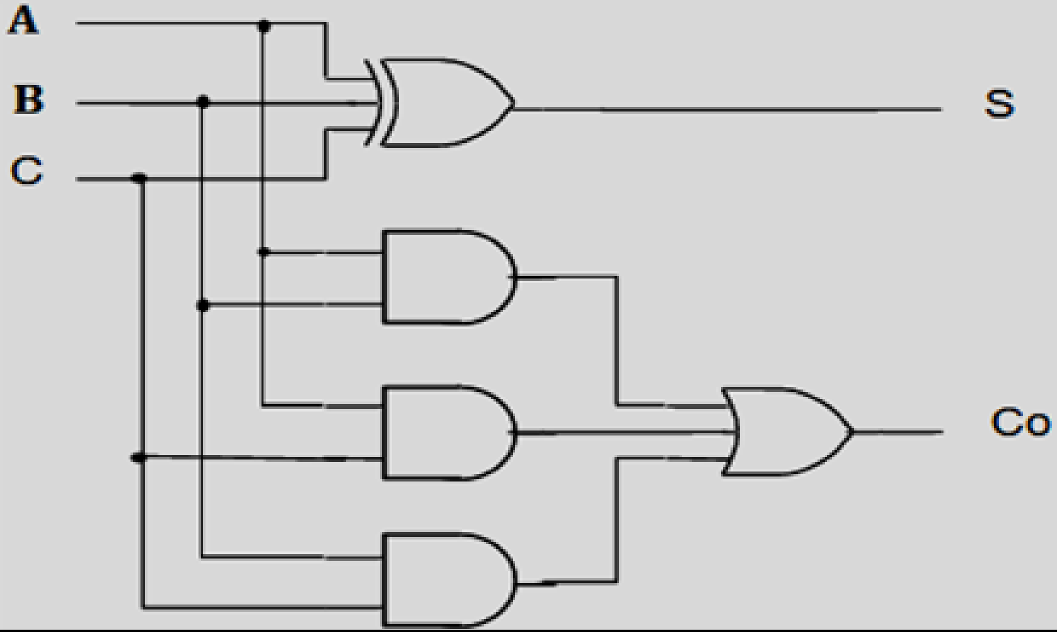


Full Adder



A	B	Cin	Sum	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





Carry, C_o = **AB + AC + BC (Using k-Map)**

Sum, S=AB'C'+A'B'C+A'BC'+ABC = $\mathbf{A \oplus B \oplus C}$

Implementation of Full Adder using Half Adder

A full adder can be designed using **two half adder** & **one OR gate**.

A	B	Cin	Sum	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

Sum, $S = A \oplus B \oplus C$

Carry, $C_o = A'BC + AB'C + ABC' + ABC$

$= A'BC + AB'C + AB(C'+C)$

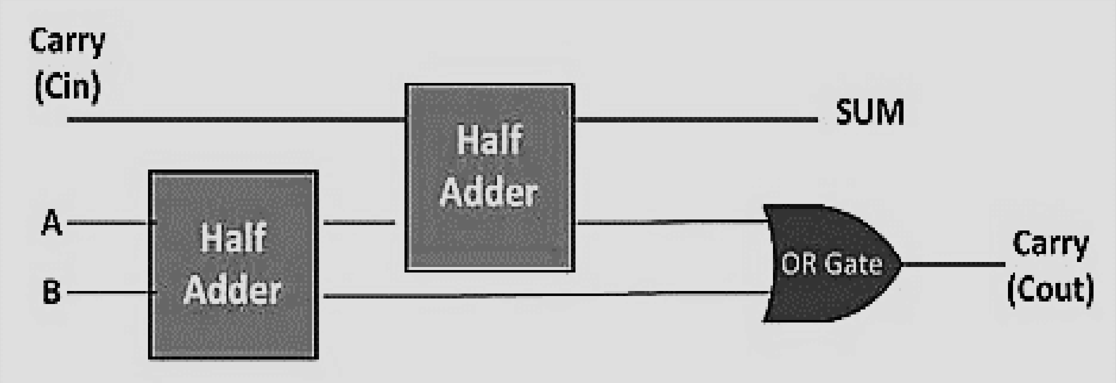
$= (A'B + AB')C + AB.1$

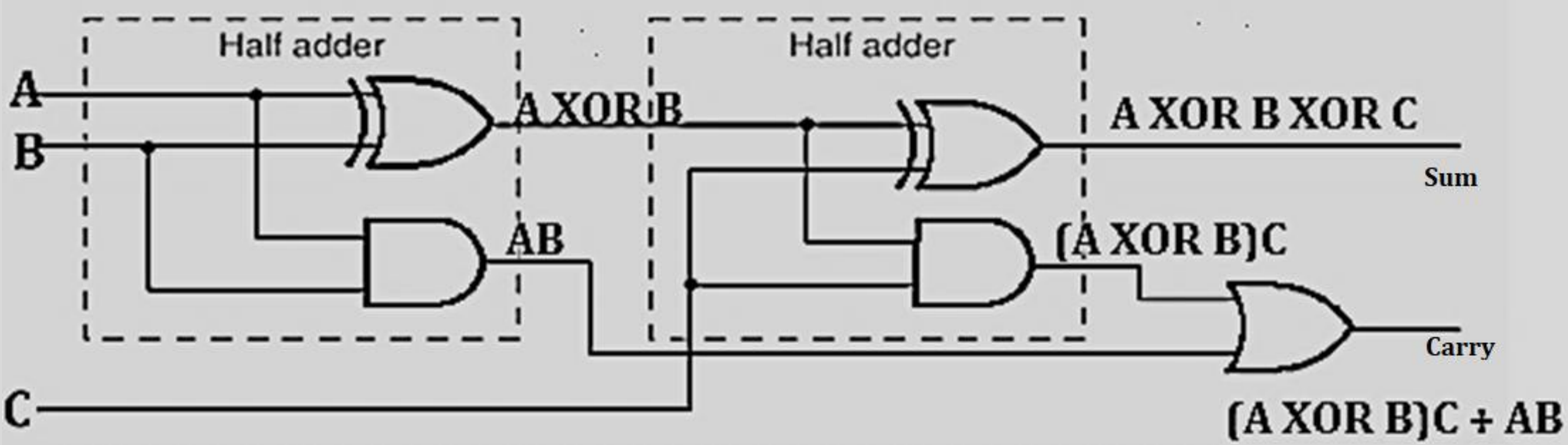
$= (A \oplus B)C + AB$

Implementation of Full Adder using Half Adder

Sum, S = $A \oplus B \oplus C$

Carry, $C_o = (A \oplus B)C + AB$





The **sum** is obtained as output from 2^{nd} Half Adder & OR gate will generate the **carry** obtained.

Thank You

DEEPA MATHEWS