LECTURE 2

DIGITAL ELECTRONICS

Logic Gates, Boolean Algebra, Combinational Circuits

Boolean Algebra

- A set of rules formulated by the English mathematician *George Boole* describe certain propositions whose outcome would be either *true or false*.

- With regard to digital logic, these rules are used to describe circuits whose state can be either, *1 (true) or 0 (false)*.

- In order to fully understand this, the relation between the an AND gate, OR gate and NOT gate operations should be appreciated.

- A number of rules can be derived from these relations as shown in the below table.

Boolean Laws

1.	Law of Identity	$A = A$ $\overline{A} = \overline{A}$
2.	Commutative Law	$A \cdot B = B \cdot A$ $A + B = B + A$
3.	Associative Law	$A \cdot (B \cdot C) = A \cdot B \cdot C$ $A + (B + C) = A + B + C$
4.	Idempotent Law	$A \cdot A = A$ $A + A = A$
5.	Double Negative Law	$\overline{\overline{A}} = A$
6.	Complementary Law	$A \cdot \overline{A} = 0$ $A + \overline{A} = 1$
7.	Law of Intersection	$A \cdot 1 = A$ $A \cdot 0 = 0$
8.	Law of Union	$A + 1 = 1$ $A + 0 = A$
9.	DeMorgan's Theorem	$\overline{AB} = \overline{A} + \overline{B}$ $\overline{A + B} = \overline{A}\,\overline{B}$
10.	Distributive Law	$A \cdot (B + C) = (A \cdot B) + (A \cdot C)$ $A + (BC) = (A + B) \cdot (A + C)$
11.	Law of Absorption	$A \cdot (A + B) = A$ $A + (AB) = A$
12.	Law of Common Identities	$A \cdot (\overline{A} + B) = AB$ $A + (\overline{A}B) = A + B$

Using the truth table:

- **Example 1**

solving algebraically

$$
\begin{aligned}
A + \overline{A}\,B &= A\,1 + \overline{A}\,B \\
&= A\,(1 + B) + \overline{A}\,B \\
&= A + A\,B + \overline{A}\,B \\
&= A + B\,(A + \overline{A}) \\
&= A + B
\end{aligned}
$$

A	B	$A + B$	$\overline{A}\,B$	$A + \overline{A}\,B$
0	0	**0**	0	**0**
0	1	**1**	1	**1**
1	0	**1**	0	**1**
1	1	**1**	0	**1**

Example 2

$$Z = (A + \overline{B} + \overline{C})(A + \overline{B}C)$$
$$Z = AA + A\overline{B}C + A\overline{B} + \overline{B}\overline{B}C + A\overline{C} + \overline{B}C\overline{C}$$
$$Z = A(1 + \overline{B}C + \overline{B} + \overline{C}) + \overline{B}C + \overline{B}C\overline{C}$$
$$Z = A + \overline{B}C$$

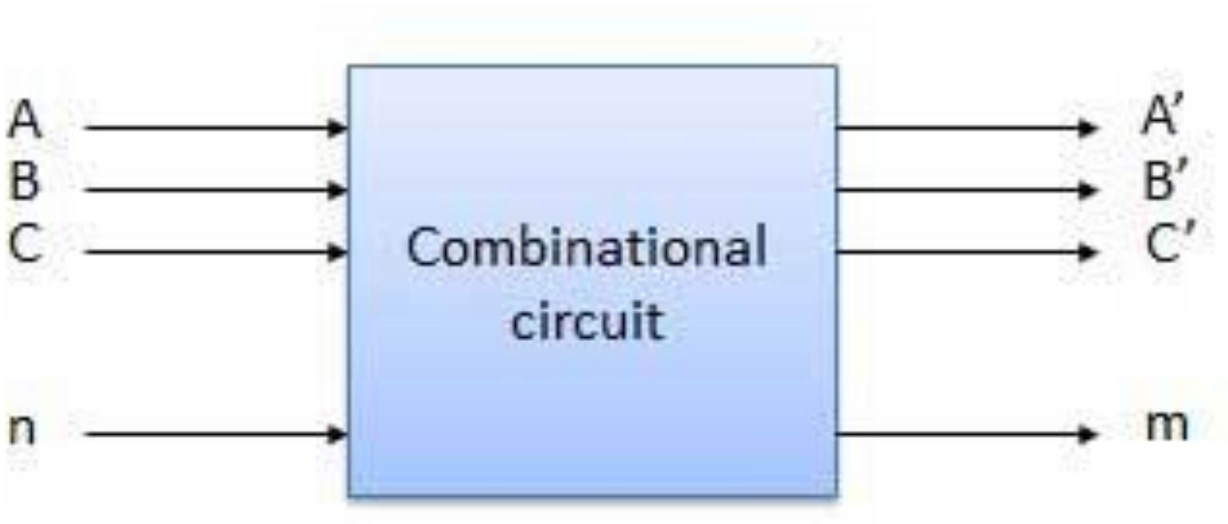
Example 3

$$A(\overline{A} + B) = A\overline{A} + AB$$
$$= 0 + AB$$
$$= AB$$

Combinational Circuits

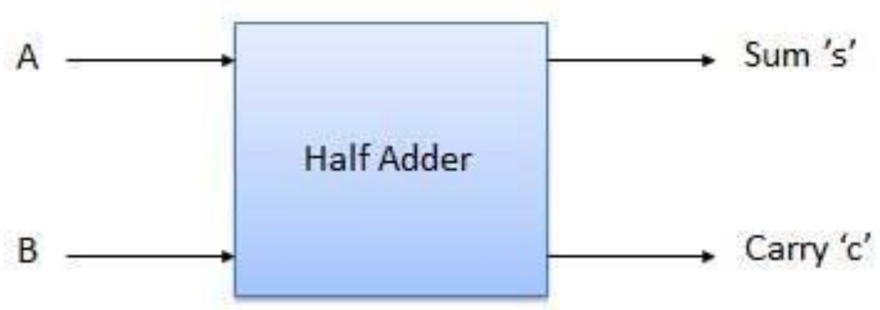
- Combinational circuit is circuit in which we combine the different gates in the circuit for example encoder, decoder, multiplexer and demultiplexer. Some of the characteristics of combinational circuits are following.
- The output of combinational circuit at any instant of time, depends only on the levels present at input terminals.
- The combinational circuit do not use any memory.
- The previous state of input does not have any effect on the present state of the circuit.
- A combinational circuit can have a n number of inputs and m number of outputs.

Block Diagram



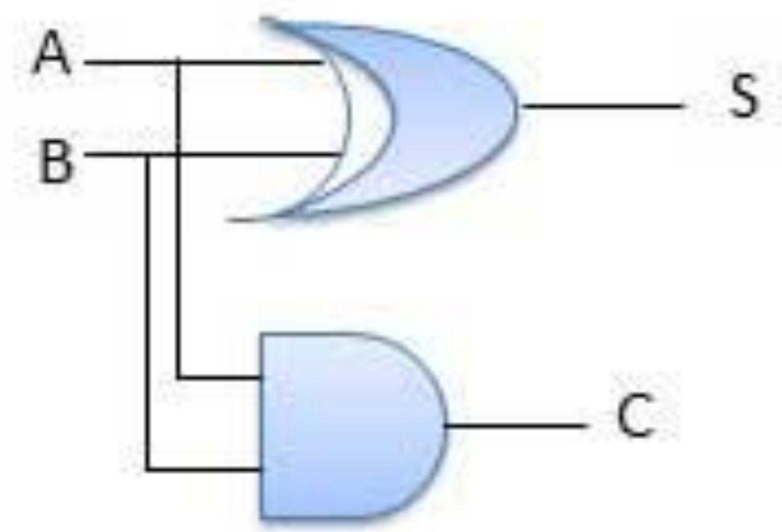
Half Adder

- Half adder is a combinational logic circuit with two input and two output.
- 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**.



Half Adder

Inputs		Output	
A	B	S	C
0	0	0	0
0	1	1	0
0	1	1	0
1	1	0	1

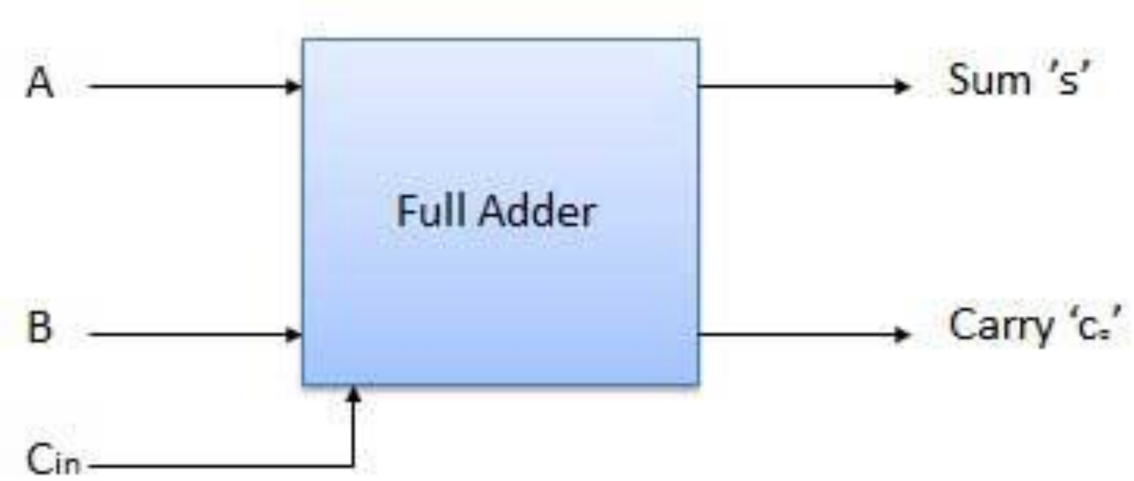


- **Full Adder**

Full adder is developed to overcome the drawback of Half Adder circuit.

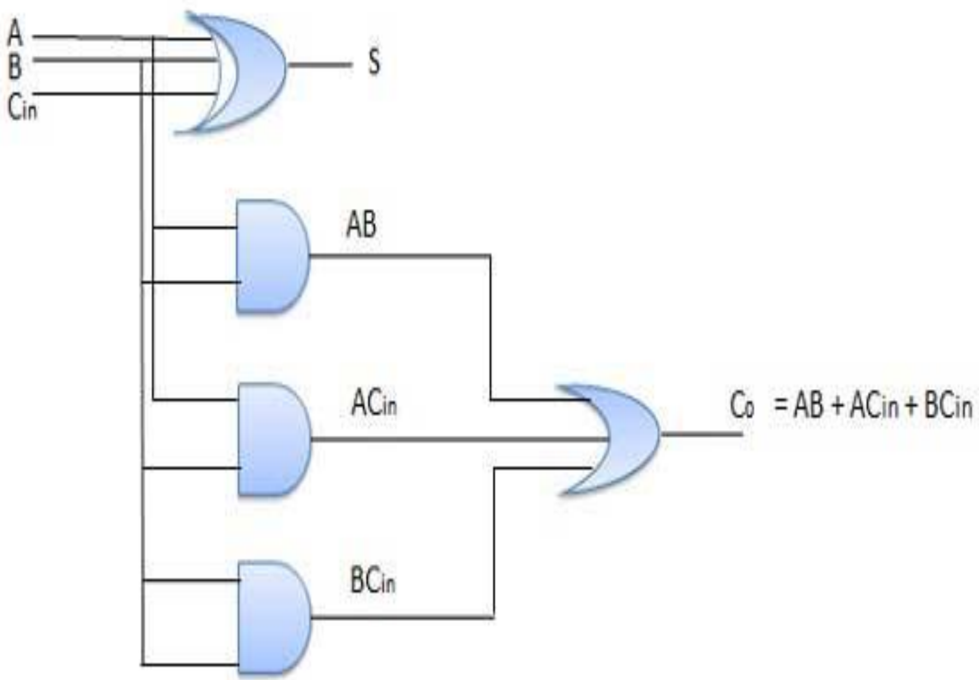
It can add two one-bit numbers A and B, and carry c.

The full adder is a three input and two output combinational circuit.



Full Adder

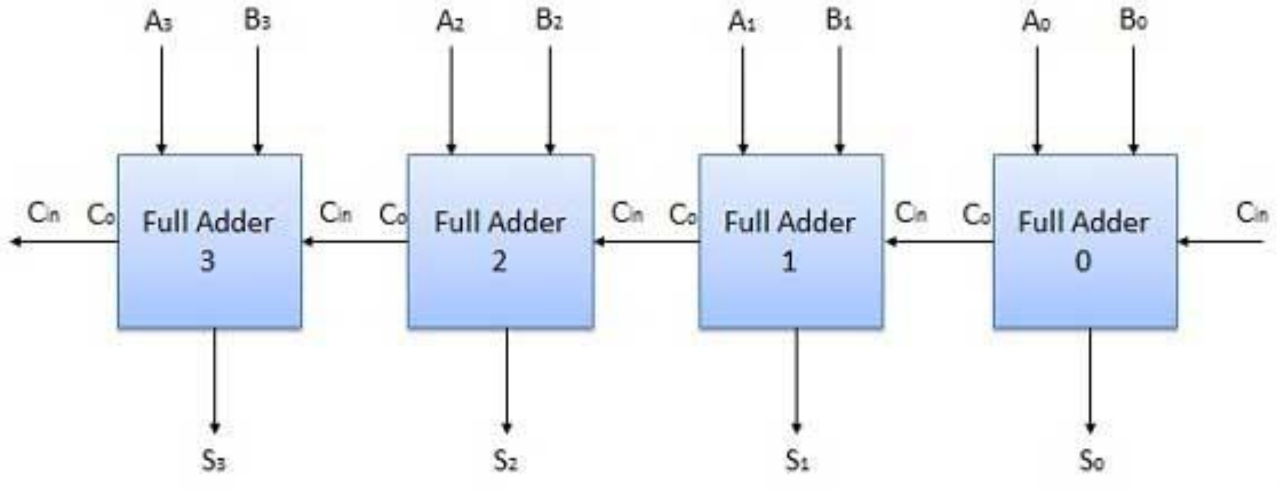
Inputs			Output	
A	B	C_{in}	S	C_o
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



- **N-Bit Parallel Adder**
- The Full Adder is capable of adding only two single digit binary number along with a carry input.
- But in practical we need to add binary numbers which are much longer than just one bit.
- 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

- In the block diagram, A_0 and B_0 represent the LSB of the four bit words A and B. Hence Full Adder-0 is the lowest stage.
- Hence its C_{in} has been permanently made 0. The rest of the connections are exactly same as those of n-bit parallel adder is shown in fig.
- The four bit parallel adder is a very common logic circuit.



- **N-Bit Parallel Subtractor**

- The subtraction can be carried out by taking the 1's or 2's complement of the number to be subtracted.
- For example we can perform the subtraction (A-B) by adding either 1's or 2's complement of B to A.
- That means we can use a binary adder to perform the binary subtraction.

- **4 Bit Parallel Subtractor**

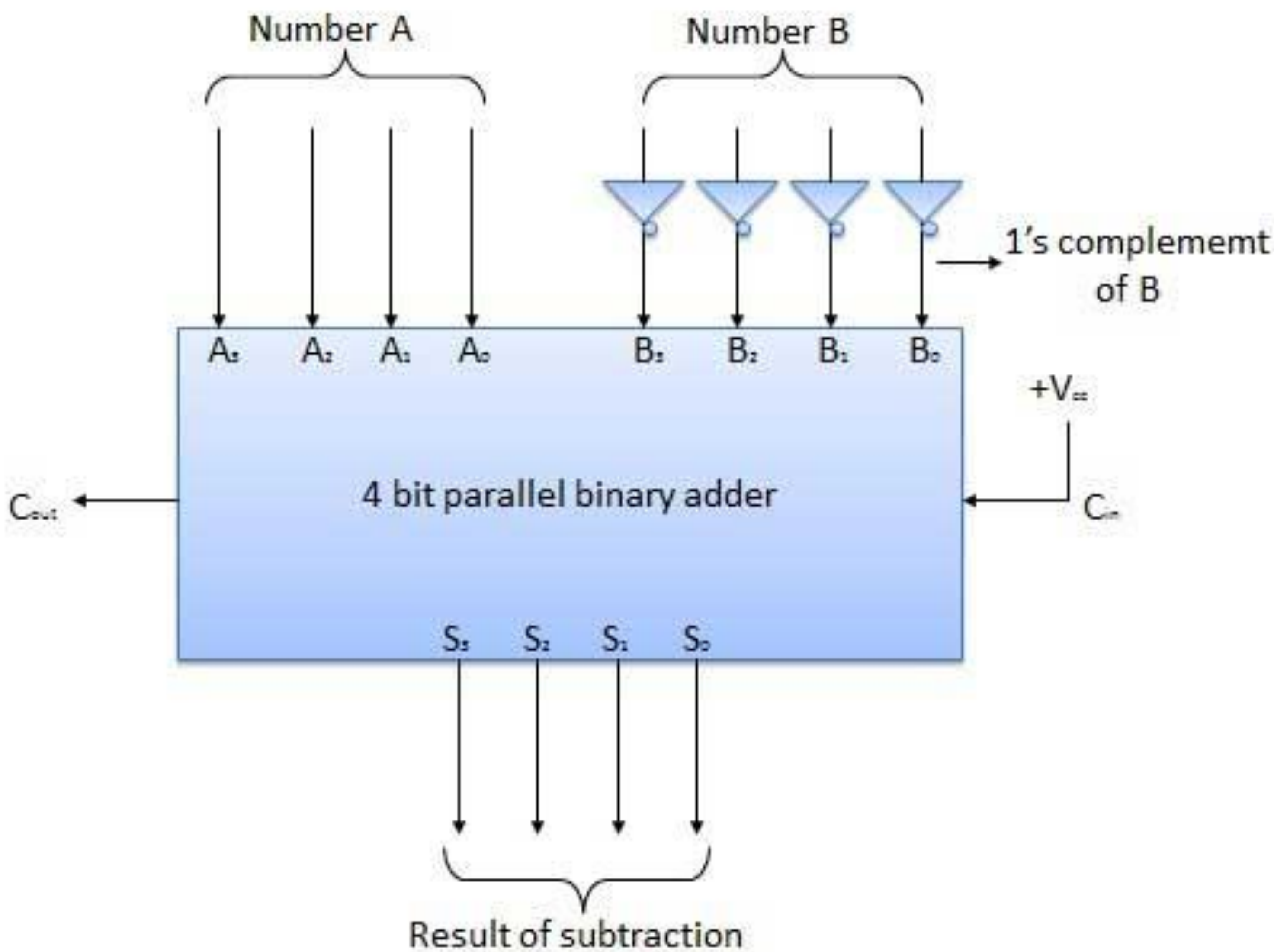
The number to be subtracted (B) is first passed through inverters to obtain its 1's complement.

The 4-bit adder then adds A and 2's complement of B to produce the subtraction.

S_3 S_2 S_1 S_0 represent the result of binary subtraction (A-B) and carry output C_{out} represents the polarity of the result.

If A > B then Cout =0 and the result of binary form (A-B) then C_{out} = 1 and the result is in the 2's complement form.

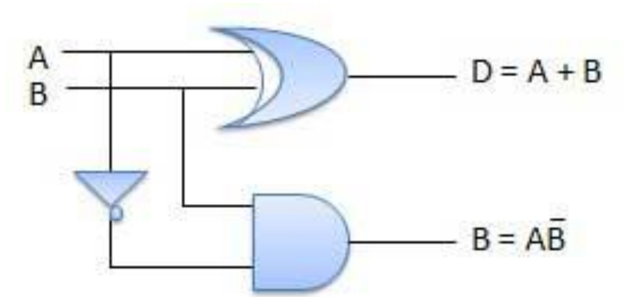
4 Bit Parallel SUBTRACTOR



- **Half Subtractors**

- Half subtractor is a combination circuit with two inputs and two outputs (difference and borrow).
- It produces the difference between the two binary bits at the input and also produces a output (Borrow) to indicate if a 1 has been borrowed.
- In the subtraction (A-B), A is called as Minuend bit and B is called as Subtrahend bit.

Inputs		Output	
A	B	(A – B)	Borrow
0	0	0	0
0	1	1	0
0	1	1	0
1	1	0	1



- **Full Subtractors**

- The disadvantage of a half subtractor is overcome by full subtractor.

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

Full Subtractors

Inputs			Output	
A	B	C	(A-B-C)	C’
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