SNS COLLEGE OF TECHNOLOGY
(An Autonomous Institution)
COIMBATORE-35
DEPARTMENT OF ECE

Binary Adder

Binary Addition Circuits

Addition and Subtraction are two basic Arithmetic Operations that must be performed by any Digital Computer. If both these operations can be properly implemented, then Multiplication and Division tasks become easy (as multiplication is repeated addition and division is repeated subtraction).

Consider the operation of adding two binary numbers, which is one of the fundamental tasks performed by a digital computer. The four basic addition operations two single bit binary numbers are:

- 0 + 0 = 0
- 1 + 0 = 1
- 0 + 1 = 1
- 1 + 1 = (Carry)1 0

	1	1	0	0
	+1	+0	+1	+0
(carry)1	0	1	1	0

In the first three operations, each binary addition gives sum as one bit , i.e., either 0 or 1. But for the fourth addition operation (where the inputs are 1 and 1), the result consists

INPUT		OUTPUT	
A	**B**	**Sum**	**Carry**
0	0	0	0
0	1	**1**	0
1	0	**1**	0
1	1	0	**1**

of two binary digits. Here, the lower significant bit is called as the ‘Sum Bit’, while the higher significant bit is called as the ‘Carry Bit’.

For single bit additions, there may not be an issue. The problem may arise when we try to add binary numbers with more than one bit.

The logic circuits which are designed to perform the addition of two binary numbers are called as Binary Adder Circuits. Depending on how they handle the output of the ‘1+1’ addition, they are divided into:

- Half Adder
- Full Adder

Let us take a look at the binary addition performed by various adder circuits.

Half Adder

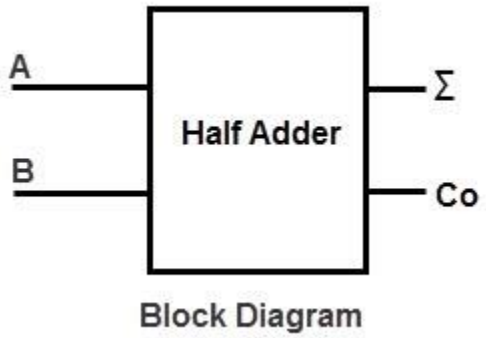
A logic circuit used for adding two 1-bit numbers or simply two bits is called as a Half Adder circuit. This circuit has two inputs and two outputs. The inputs are the two 1-bit binary numbers (known as Augend and Addend) and the outputs are Sum and Carry.

The following image shows the block diagram of Half Adder.

The truth table of the Half Adder is shown in the following table.

If we observe the ‘Sum’ values in the above truth table, it resembles an Ex-OR Gate. Similarly, the values for ‘Carry’ in the above truth table resembles an AND Gate.

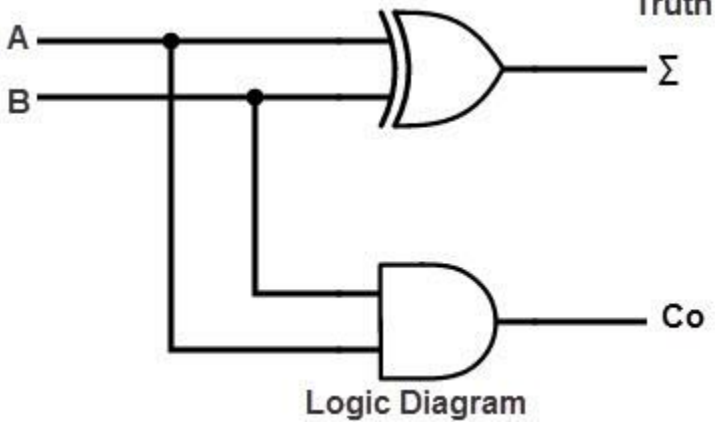
So, to properly implement a Half Adder, you need two Logic Gates: an XOR gate for 'Sum' Output and an AND gate for 'Carry' output. The following image shows the Logic Diagram of a Half Adder.



Block Diagram

A	B	Sum (∑)	Carry Out (Co)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Truth Table



Logic Diagram

In the above half adder circuit, inputs are labeled as A and B. The 'Sum' output is labeled as summation symbol (∑) and the Carry output is labeled with C_o.

Half adder is mainly used for addition of augend and addend of first order binary numbers i.e., 1-bit binary numbers. We cannot add binary numbers with more than one bit as the Half Adder cannot include the 'Carry' information from the previous sum.

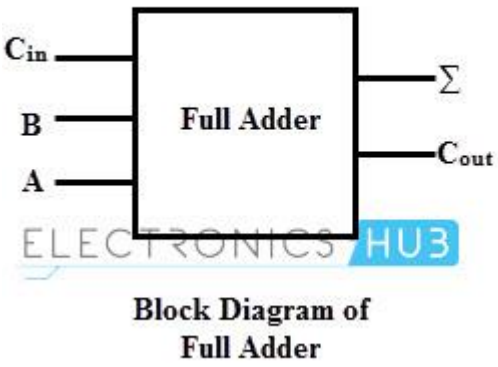
Due to this limitation, Half Adder is practically not used in many applications, especially in multi-digit addition. In such applications, carry of the previous digit addition must be added along with two bits; hence it is a three bit addition.

Full Adder

A Full Adder is a combinational logic circuit which performs addition on three bits and produces two outputs: a Sum and a Carry. As we have seen that the Half Adder cannot respond to three inputs and hence the full adder is used to add three digits at a time.

It consists of three inputs, of which two are input variables representing the two significant bits to be added, whereas the third input terminal is the carry from the previous addition. The two outputs are a Sum and Carry outputs.

The following image shows a block diagram of a Full Adder where the inputs are labelled as A, B and C_{IN}, while the outputs are labelled as $\sum$ and C_{OUT}.



Block Diagram of Full Adder

C_{in}	B	A	$\sum$	C_{out}
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**

Truth Table

Coming to the truth table, the following table shows the truth table of a Full Adder.

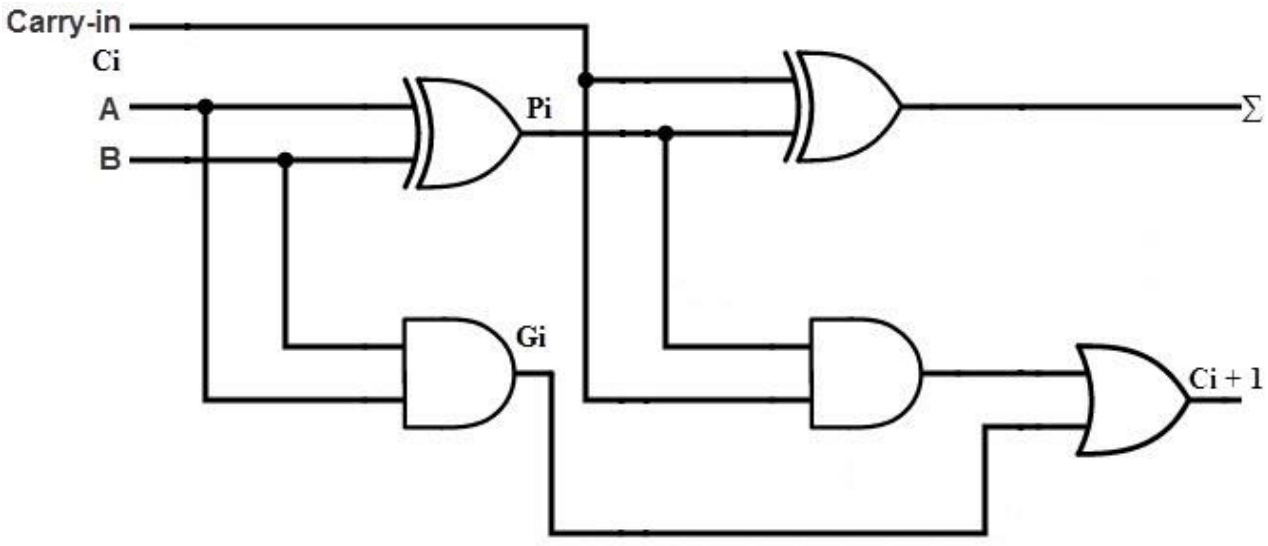
.

INPUT			**OUTPUT**	
A	**B**	**C_{IN}**	**Sum**	**C_{OUT}**
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**

From the above truth table, we can obtain the Boolean Expressions for both the Sum and Carry Outputs. Using those expressions, we can build the logic circuits for Full Adder. But by simplifying the equations further, we can derive at a point that a Full Adder can be easily implemented using two Half Adders and an OR Gate.

The following image shows a Full Adder Circuit implemented using two Half Adders and an OR Gate. Here, A and B are the main input bits, C_{IN} is the carry input, ∑ and C_{OUT} are the Sum and Carry Outputs respectively.

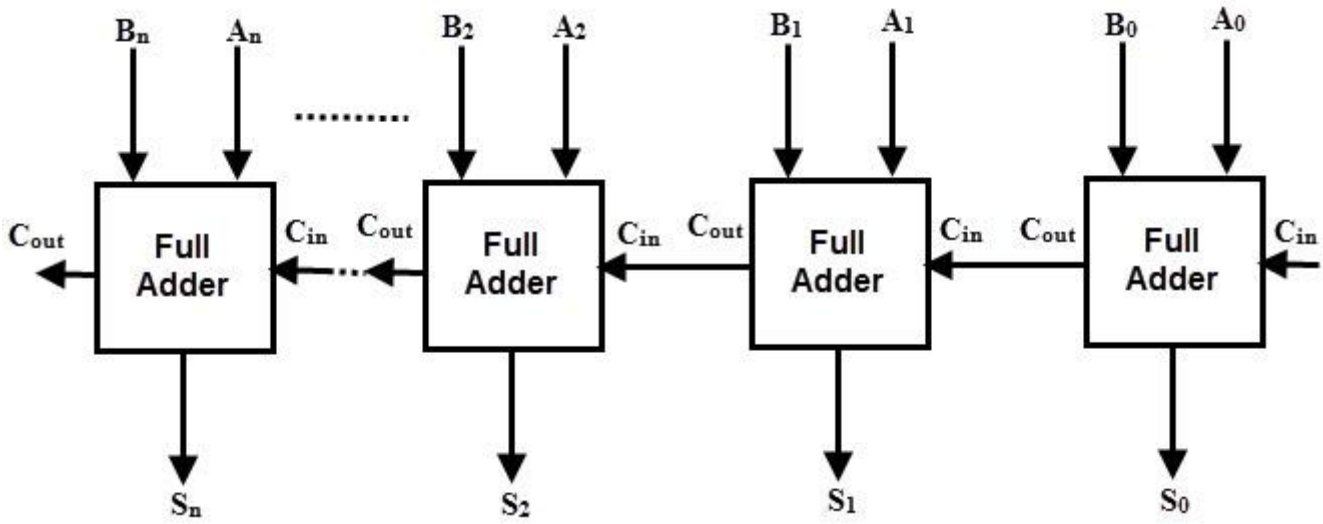


Parallel Binary Adders

A single Full Adder performs the addition of two one bit numbers and also the carry input. For performing the addition of binary numbers with more than one bit, more

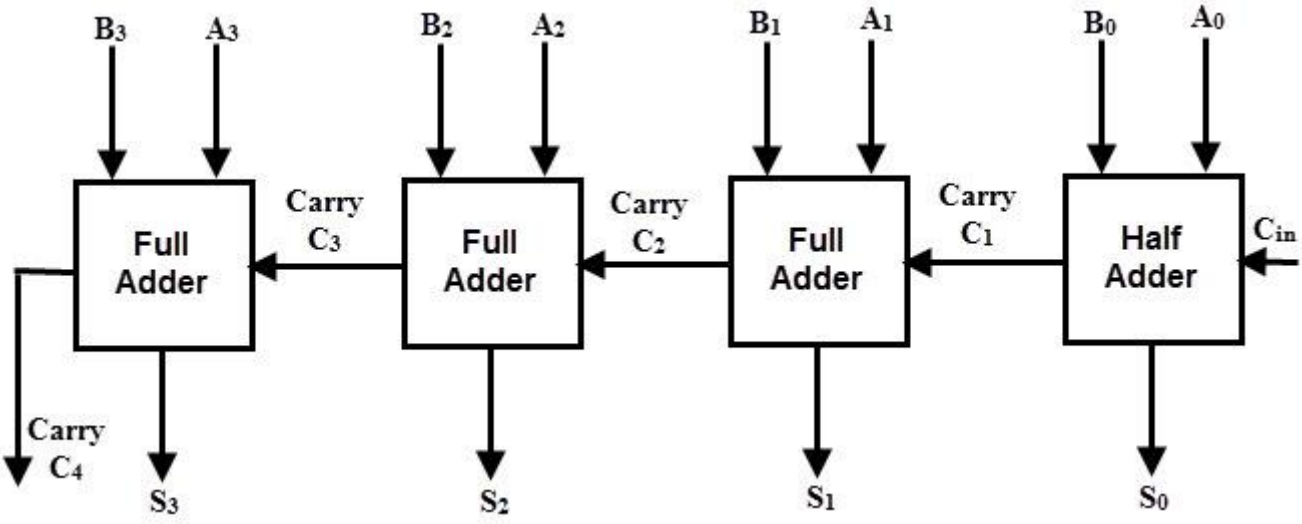
than one full adder is required and the number of Full Adders depends on the number bits. Thus, a Parallel Adder, is a combination of Multiple Full Adders and is used for adding all bits of the two numbers simultaneously.

By connecting 'n' number of full adders in parallel, an n-bit Parallel Adder can be constructed. From the below figure, it is to be noted that there is no carry at the least significant position, hence we can use either a half adder or make the carry input of full adder as zero at this position.



The following figure shows a Parallel 4-bit Binary Adder, which has three full adders and one half adder. The two binary numbers to be added are 'A_3 A_2 A_1 A_0' and 'B_3 B_2 B_1 B_0', which are applied to the corresponding inputs of the Full Adders. This parallel adder produces their result as 'C_4 S_3 S_2 S_1 S_0', where C_4 is the final carry.

y.



In the 4 bit adder, first block is a half-adder that has two inputs as A0 B0 and produces a sum S0 and a carry bit C1. The first block can also be a full adder and if so, then the input Carry C0 must be 0.

Next three blocks should be full adders, as there are three inputs applied to them (two main binary bits and a Carry bit from the previous stage).

Hence, the second block full adder produces a sum S1 and a carry C2. This will be followed by other two full adders and thus the final result is C4 S3 S2 S1 S0.

Commonly, the Full Adders are designed in dual in-line package integrated circuits. 74LS283 is a popular 4-bit full adder IC. Arithmetic and Logic Unit or ALU of an unit computer consist of these parallel adders to perform the addition of binary numbers.