Tutorial on Adder and Subtractor Logic Circuits

Digital Adder:

In digital electronics an adder is a logic circuit that implements addition of numbers. In many computers and other types of processors, adders are used to calculate addresses, similar operations and table indices in the arithmetic logic unit (ALU) and also in other parts of the processors. These can be built for many numerical representations like binary coded decimal or excess-3.

Adders are classified into two types:

1. Half adder

2. Full adder.

Half Adder-

The half adder circuit is required to add two input digits (for Ex. A and B) and generate a carry and sum. The half adder adds two binary digits called as augend and addend and produces two outputs as sum and carry (XOR is applied to both inputs to produce sum and AND gate is applied to both inputs to produce carry). It means half adder circuits can add only two digits in other words if we need to add more than 2 digits it will not work, so, it the limitation of an half adder electronic circuits. To resolve this problem a full adder circuit is required.

Application-

- The ALU of a computer uses half adder to compute the binary addition operation on two bits.
- Half adder is used to make full adder as a full adder requires 3 inputs, the third input being an input carry i.e. we will be able to cascade the carry bit from one adder to the other.
- Ripple carry adder is possible to create a logical circuit using multiple full adders to add N-bit numbers. Each full adder inputs a C (in), which is the C (out) of the previous adder. This kind of adder is called RIPPLE CARRY ADDER, since each carry bit "ripples" to the next full adder.

Full Adder-

A full adder logic circuit has three inputs A, B and C, which add the three input numbers and generate a carry and sum. The full adder adds 3 one bit numbers, where two can be referred to as operands and one can be referred to as bit carried in. And produces 2-bit output, and these can be referred to as output carry and sum. The difference between a half-adder and a full-adder is that the full-adder has three inputs and two outputs, whereas half adder has only two inputs and two outputs. When full-adder logic is designed, you string eight of them together to create a byte-wide adder and cascade the carry bit from one adder to the next. Figure given below show the logic circuits of Half adder and Full adder

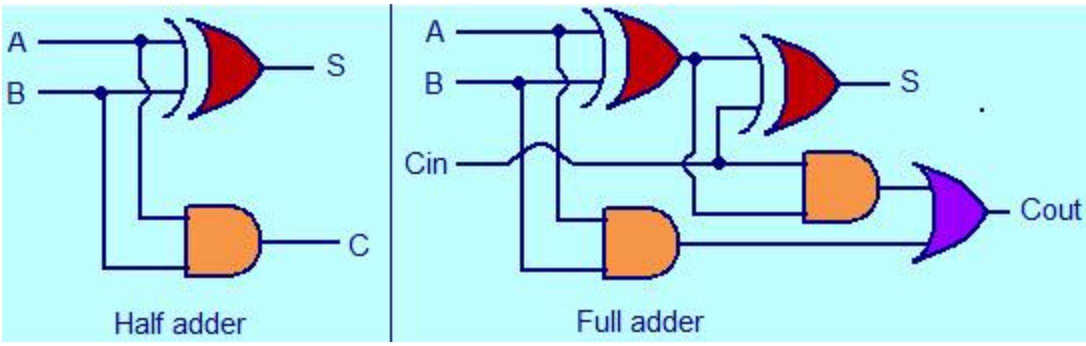
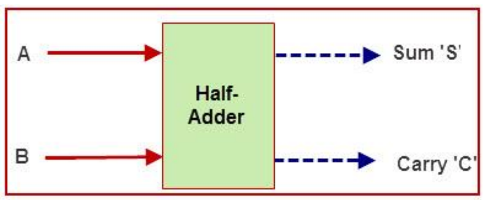


Fig. Half Adder and Full Adder Circuit

Symbol and Operation of Half Adder

Symbol of half adder circuit is shown in figure given below. And the operation can be stated as



You can design simple addition with the help of logic gates for half adder circuits.
Let's see an addition of single bits:
0+0 = 0
0+1 = 1
1+0 = 1
1+1 = 10
These are the least possible single-bit combinations. But the result for 1+1 is 10, the sum result must be re-written as a 2-bit output. Thus, the equations can be written as
0+0 = 00
0+1 = 01
1+0 = 01
1+1 = 10
The output '1'of '10' is carry-out. 'SUM' is the normal output and 'CARRY' is the carry-out.

Half Adder Truth Table

INPUTS		OUTPUTS	
A	B	SUM	CARRY
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

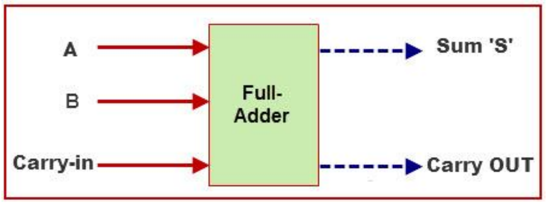
Half Adder Truth Table

Now it is clear that 1-bit adder can be easily implemented with the help of the XOR Gate for the output 'SUM' and an AND Gate for the 'Carry'. When we need to add, two 8-bit bytes together, we can be done with the help of full-adder logic. The half-adder is useful when you want to add one binary digit quantities. A way to develop two-binary digit adders would be to make a truth table and reduce it.

The simplest expression uses the exclusive OR function is:
Sum =A XOR B
Carry A AND B

Full Adder

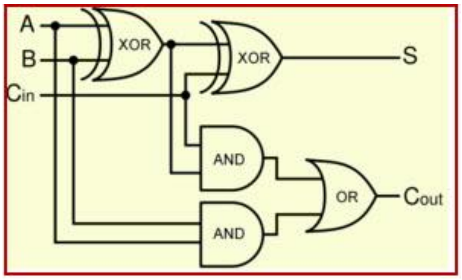
Full adder is difficult to implement than a half adder as it has three inputs. The first two inputs are A and B and the third input is an input carry as C-in. When full adder logic is designed, you string eight of them together to create a byte-wide adder and cascade the carry bit from one adder to the next. The output carry is designated as C OUT and the normal output is designated as S. Logic symbol of full adder circuit is shown below.



Truth Table of full Adder

INPUTS			OUTPUT	
A	B	C-IN	C-OUT	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

With the truth-table, the full adder logic can be implemented. You can see that the output S is an XOR between the input A and the half-adder, SUM output with B and C-IN inputs.
We take C-OUT will only be true if any of the two inputs out of the three are HIGH. So, we can implement a full adder circuit with the help of two half adder circuits. At first, half adder will be used to add A and B to produce a partial Sum and a second half adder logic can be used to add C-IN to the Sum produced by the first half adder to get the final S output.



If any of the half adder logic produces a carry, there will be an output carry. So, C OUT will be an OR function of the half-adder Carry outputs. Take a look at the implementation of the full adder circuit shown below.
The implementation of larger logic diagrams is possible with the above full adder logic a simpler symbol is mostly used to represent the operation. Given below is a simpler schematic representation of a one-bit full adder.

Sum = A XOR B NOT XOR Cin
Carry Out = A.B + Cin (A XOR B)

The relationship between the Full-Adder and the Half-Adder is half adder produces results and full adder uses half adder to produce some other result. Similarly, while the Full-Adder is of two Half-Adders, the Full-Adder is the actual block that we use to create the arithmetic circuits.

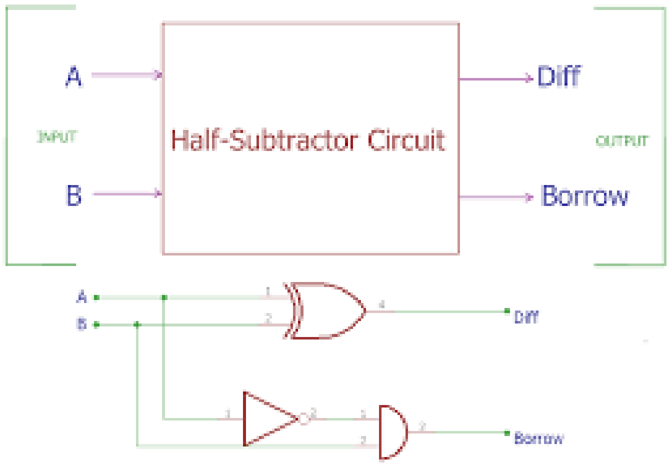
Digital Subtractor

In electronics, a **subtractor** can be designed using the same approach as that of an adder. The binary subtraction process is summarized below. As with an adder, in general case of calculations on multi-bit numbers, three bits are involved in performing the subtraction for each bit of the difference: the minuend (**A**), subtrahend (**B**), and a borrow in from the previous (less significant) bit order position ($\mathbf{B_{in}}$). The outputs are the difference bit ($\mathbf{D_{iff}}$) and borrow bit $\mathbf{B_{in}}$+1. The subtractor is best understood by considering that the subtrahend and both borrow bits have negative weights, whereas the **A** and **D** bits are positive.

Half Subtractor-

The half subtractor is a combinational circuit which is used to perform subtraction of two bits. It has two inputs, the minuend (**A**) and subtrahend (**B**) and two outputs the difference $\mathbf{D_{iff}}$ and borrows out $\mathbf{B_{out}}$. Borrows out signal is set when the subtractor needs to borrow from the next digit in a multi digit subtraction. That is, $\mathbf{B_{out} = 1}$ when $\mathbf{A} < \mathbf{B}$. Since **A** and **B** are bits, $\mathbf{B_{out} = 1}$ if and only if $\mathbf{A = 0}$ and $\mathbf{B = 1}$. An important point worth mentioning is that the half subtractor diagram aside implements **A - B** and not **B - A** since $\mathbf{B_{out}}$ on the diagram is given by $\mathbf{B_{out}}$ **= NOT A. B**

Figure shown below is depicted the symbol and circuit of a half subtractor.



Half-Subtractor Circuit

This is an important distinction to make since subtraction itself is not commutative, but the difference bit **D_{iff}** is calculated using an XOR gate which is commutative.

Truth Table

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

Logical Expression:

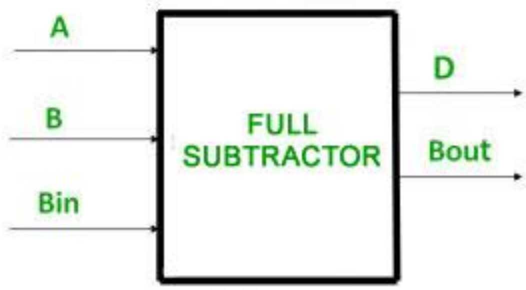
Difference (**D**) = **A** XOR B
Borrow = NOT **A.B**

Applications:

- Half subtractor is used to subtract the least significant column numbers. For subtraction of multi-digit numbers, it can be used for the LSB
- Half subtractor is used to reduce the force of audio or radio signals
- It can be used in amplifiers to reduce the sound distortion
- Half subtractor is used in ALU of processor
- It can be used to increase and decrease operators and also calculates the addresses

Full Subtractor

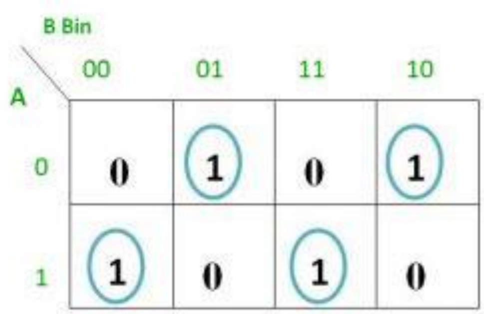
A full subtractor is a combinational circuit that performs subtraction of two bits, one is minuend and other is subtrahend, taking into account borrow of the previous adjacent lower minuend bit. This circuit **has three inputs and two outputs**. The three inputs A, B and Bin, denote the minuend, subtrahend, and previous borrow, respectively. The two outputs, D and Bout represent the difference and output borrows, respectively. Generally, the full subtractor is one of the most used and essential combinational logic circuits. It is a basic electronic device, used to perform subtraction of two binary numbers. Likewise, the full-subtractor uses binary digits like 0, 1 for the subtraction



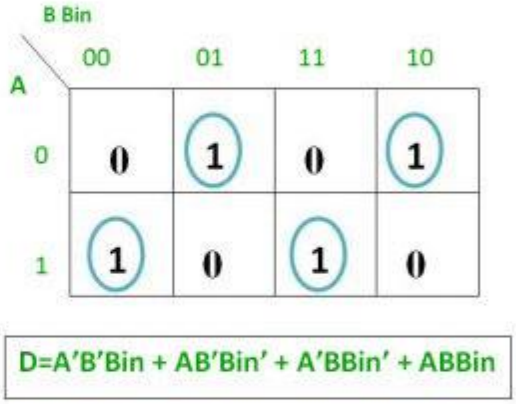
Truth Table

INPUT			OUTPUT	
A	B	Bin	D	Bout
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

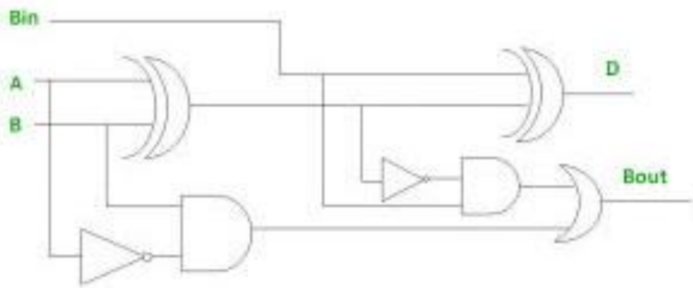
From above table we can draw the K-Map as shown for "difference" and "borrow".



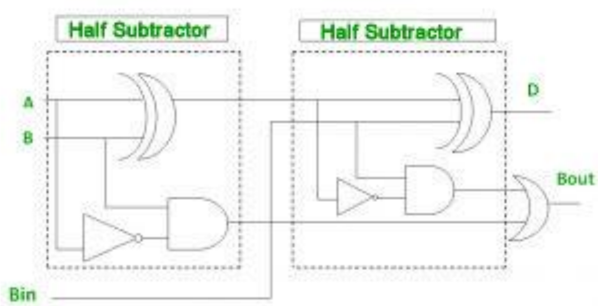
D=A'B'Bin + AB'Bin' + A'BBin' + ABBin



Logic Circuit for Full Subtractor –



Implementation of Full Subtractor using Half Subtractors –
2 Half Subtractors and an OR gate is required to implement a Full Subtractor.



Applications

- These are generally employed for ALU in computers to subtract as CPU and GPU for applications of graphics to decrease the circuit difficulty
- Subtractors are mostly used for performing arithmetical functions like subtraction in electronic calculators and digital devices

- These are also applicable for different microcontrollers for arithmetic subtraction, timers and program counter (PC)
- Subtractors are used in processors to compute tables, address, etc.

By using any full subtractor logic circuit, full subtractor using NAND gates and full subtractor using NOR gates can be implemented since both the NAND and NOR gates are treated as universal gates.