Boolean Logic

Sunday, October 25, 2020 2:21 PM

Basic digital logic gates perform logical operations of AND, OR, and NOT on binary numbers.
Information is stored in computer systems in binary form. A binary bit represents one of the two possible states, which are generally referred to as logic "1" and logic "0".
Includes instruction like AND, which only sets bits in the result that are set in both operands, OR, which only sets bits in the result that are set in at least one of the operands, and XOR, which only sets bits in the result that are set in one operand and not the other.
Boolean logic is a form of algebra where all values are either True or False. These values of true and false are used to test the conditions that selection and iteration are based around.

A *Boolean function* is a mathematical function that maps arguments to a value, where the allowable values of range (the function arguments) and domain (the function value) are just one of two values— *true* and *false* (or 0 and 1). The study of Boolean functions is known as *Boolean logic*.
Boolean functions.
To define any Boolean function, we need only to specify its value for each possible value of its inputs. The *not* function is a Boolean function of one variable The *and*, *or*, and *exclusive or* functions are familiar Boolean functions of two variables.
Notation. There are many competing notations for elementary Boolean functions. In this chapter, we primarily use the circuit-design notation.

	logic	Java boolean	Java bitwise	circuit design
NOT	$\neg x$	!x	~x	x ′
AND	$x \wedge y$	x && y	x & y	xy
OR	$x \vee y$	x || y	x | y	x+y
XOR	$x \oplus y$	x ^ y	x ^ y	$x \oplus y$

Truth tables is One way to define a Boolean function is to specify its value for each possible value of its arguments. We use a *truth table* to do so in an organized way. A truth table has one column for each variable, one row for each possible combination of variable values, and a column that specifies the value of the function for that combination.

NOT

x	x ′
0	1
1	0

AND

x	y	xy
0	0	0
0	1	0
1	0	0
1	1	1

OR

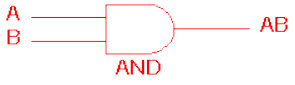
x	y	x+y
0	0	0
0	1	1
1	0	1
1	1	1

XOR

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

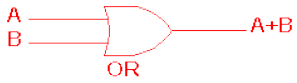
A truth table for a function of n variables has 2^n rows.

Boolean algebra is the building block of computing and has three basic operations
AND gate



2 Input AND gate		
A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

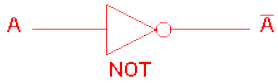
The AND gate is an electronic circuit that gives a **high** output (1) only if **all** its inputs are high. A dot (.) is used to show the AND operation i.e. A.B. Bear in mind that this dot is sometimes omitted i.e. AB
a statement of the form P AND Q is true only when both of its components are true, we conclude that 1 AND 1 should be 1, whereas all other cases should produce an output of 0,
OR gate



2 Input OR gate		
A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

The OR gate is an electronic circuit that gives a high output (1) if one or more of its inputs are high. A plus (+) is used to show the OR operation.

NOT gate



NOT gate	
A	$\overline{A}$
0	1
1	0

The NOT gate is an electronic circuit that produces an inverted version of the input at its output. It is also known as an *inverter*. If the input variable is A, the inverted output is known as NOT A. This is also shown as A', or A with a bar over the top, as shown at the outputs.

XOR Gate

An XOR gate (sometimes referred to by its extended name, Exclusive OR gate) is a digital logic gate with two or more inputs and one output that performs *exclusive disjunction*. The output of an XOR gate is **true** only when exactly one of its inputs is **true**. If both of an XOR gate's inputs are **false**, or if both of its inputs are **true**, then the output of the XOR gate is **false**.

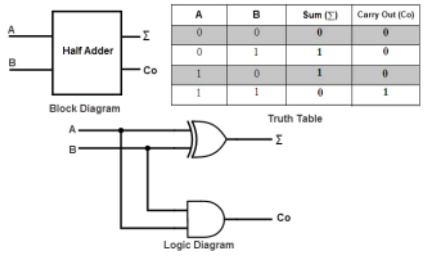
Inputs		Outputs
X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

Boolean algebra has three basic properties

1. Commutative
2. Associative
3. Distributive

Half Adder

A logic circuit block used for adding two one bit numbers or simply two bits is called as a half adder circuit. This circuit has two inputs which accept the two bits and two outputs, with one producing sum output and other produce carry output.



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

Truth Table

the expressions describing the below truth table are:

$$S = A \oplus B$$

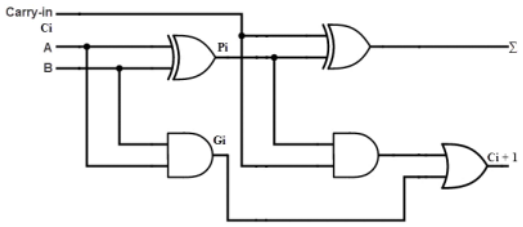
$$C_{out} = A \cdot B$$

Full Adder :

Full adder can be formed by combining two half adders and an OR gate as shown in above where output and carry-in of the first adder becomes the input to the second half adder that produce the total sum output.

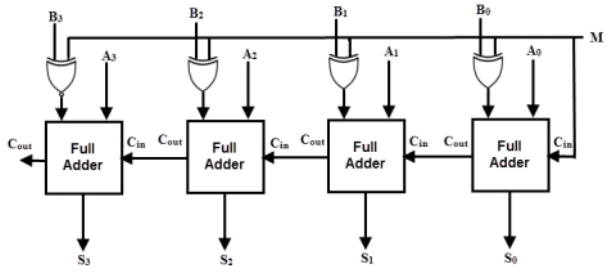
lower significant bit is called as the sum bit, whereas the higher significant bit is called as the carry bit. The logic circuits which are designed to perform the

addition of two binary numbers are called as binary adder circuits.



For performing the addition of binary numbers with more than one bit, more than one full adder is required depends on the number bits. Thus, a parallel adder is used for adding all bits of the two numbers simultaneously.
By connecting a number of full adders in parallel, n-bit parallel adder is constructed.
In the 4 bit adder, first block is a half-adder that has two inputs as AOBO and produces their sum SO and a carry bit C1. Next block should be full adder as there are three inputs applied to it. Hence this full adder produces their sum S1 and a carry C2. This will be followed by other two full adders and thus the final sum is C4S3S2S1SO.
The carry-out of the first adder becomes the carry-in for the second adder and so on
The circuit consists of 4 full adders since we are performing operation on 4-bit numbers.
Sum = (A XOR B) XOR C-in
Carry = (A OR B) OR (A XOR B) . C-in

4 Bit Binary Adder:

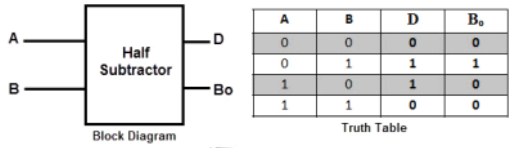


Binary Subtractor:

Binary Subtractor is a decision making circuit that subtracts two binary numbers from each otherto find the resulting difference between the two numbers. Unlike the Binary Adder which produces a SUM and a CARRY bit when two binary numbers are added together, the binary subtractor produces a DIFFERENCE, D by using a BORROW bit, B from the previous column

Half Subtractor :

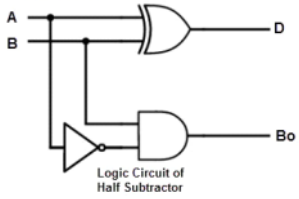
A half subtractor is a multiple output combinational logic network that does the subtraction of two bits of binary data. It has input variables and two output variables. Two inputs are corresponding to two input bits and two output variables corresponds to the difference bit and borrow bit.
The binary subtraction is also performed by the Ex-OR gate with additional circuitry to perform the borrow operation. Thus, a half subtractor is designed by an Ex-OR gate including AND gate with A input complemented before fed to the gate.



Block Diagram

A	B	D	B_o
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

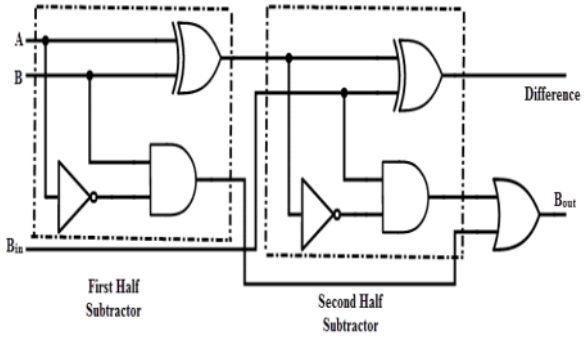
Truth Table



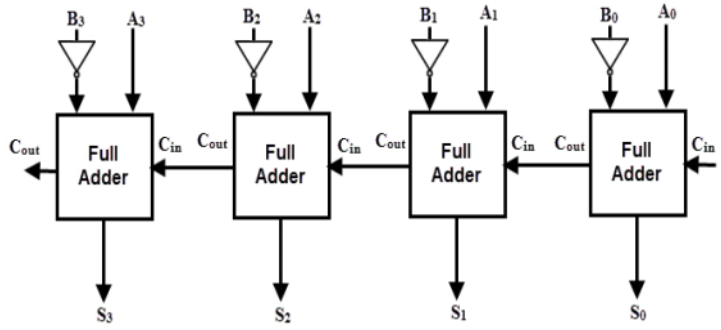
Logic Circuit of Half Subtractor

Full Subtractor :

It has three input terminals in which two terminals corresponds to the two bits to be subtracted (minuend A and subtrahend B), and a borrow bit Bi corresponds to the borrow operation. There are two outputs, one corresponds to the difference D output and other borrow output Bo as shown in figure along with truth table.
To perform the subtraction of binary numbers with more than one bit is performed through the parallel subtractors.



4 Bit Binary Subtractor :



Binary Adder & Subtractor

In Digital Circuits, A Binary Adder-Subtractor is one which is capable of both addition and subtraction of binary numbers in one circuit itself. The operation being performed depends upon the binary value the control signal holds.

We know that the subtraction of A by B is obtained by taking 2's complement of B and adding it to A. The 2's complement of B is obtained by taking 1's complement and adding 1 to the least significant pair of bits.

1's complement of B is obtained with the inverters (NOT gate) and a 1 can be added to the sum through the input carry.

When K= 1, the circuit is a subtractor and when K=0, the circuit becomes adder. The Ex-OR gate consists of two inputs to which one is connected to the B and other to input M. When M = 0, B Ex-OR of 0 produce B. Then full adders add the B with A with carry input zero and hence an addition operation is performed

When K = 1, B Ex-OR of 0 produce B complement and also carry input is 1. Hence the complemented B inputs are added to A and 1 is added through the input carry, nothing but a 2's complement operation. Therefore, the subtraction operation is performed.

1. For K = 0

The circuit is an adder, since we have B XOR 0 = B . The full-adders receive the value of B, the input carry C0 is 0, and the circuit performs the addition of A to B (i.e. A+B).

2. For K = 1

The circuit becomes a Subtractor, since we have B XOR 1 = B . The full-adders receive the value of B , the input carry C0 is 1. The B inputs are all complemented and a 1 is added through the input carry.

The circuit performs the addition of A to 2's complement of B