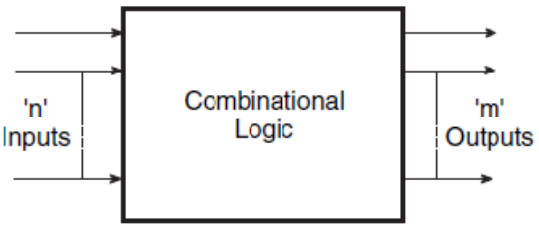
Digital Electronics

Combinational Logic Functions

Binary Adders & Subtractors

Digital logic circuits can be classified as either combinational or sequential circuits. A combinational circuit is one where the output at any time depends only on the present combination of inputs at that point of time with total disregard to the past state of the inputs. The logic gate is the most basic building block of combinational logic. The logical function performed by a combinational circuit is fully defined by a set of Boolean expressions. The other category of logic circuits, called sequential logic circuits, comprises both logic gates and memory elements such as flip-flops. Owing to the presence of memory elements, the output in a sequential circuit depends upon not only the present but also the past state of inputs.



Block Diagram of Combinational Logic Circuit

Above figure shows the block schematic representation of a generalized combinational circuit having n input variables and m output variables or simply outputs. Since the number of input variables is n, there are 2n possible combinations of bits at the input. Each output can be expressed in terms of input

variables by a Boolean expression, with the result that the generalized system can be expressed by m Boolean expressions.

General Design procedure for combinational logic circuits:

The design of combinational circuits starts from verbal outline of the problem and ends in a logic circuit diagram, or a set of Boolean functions from which the logic diagram can be easily obtained.

The different steps involved in the design of a combinational logic circuit are as follows:

1. Statement of the problem.

2. Identification of input and output variables.

3. Expressing the relationship between the input and output variables.

4. Construction of a truth table to meet input–output requirements.

5. Writing Boolean expressions for various output variables in terms of input variables.

6. Minimization of Boolean expressions.

7. Implementation of minimized Boolean expressions.

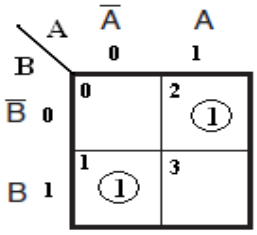
Design and applications of Binary adders:

Half Adder:

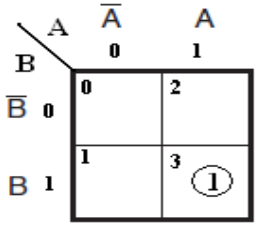
A half adder is a combinational logic circuit that performs the arithmetic addition of two bits. Such a circuit thus has two inputs that represent the two bits to be added and two outputs, with one producing the SUM output and the other producing the CARRY.

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

For Sum S:



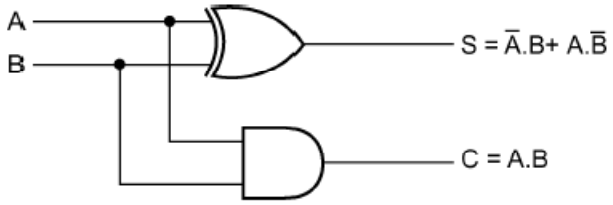
For Carry C:



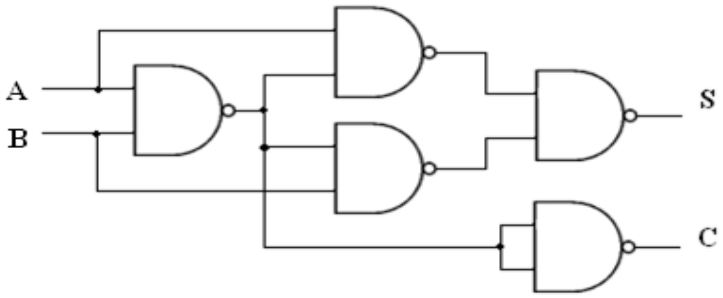
The Boolean expressions for the SUM and CARRY outputs are given by the equations

$$S = A'B + AB' = A \oplus \mathrm{B}$$

$$C = A \cdot B$$



Half Adder using NAND gates:



Full Adder:

A full adder circuit is an arithmetic circuit block that can be used to add three bits to produce a SUM and a CARRY output. Two of the input variables X and Y represent the two significant bits to be added and the third input Z represents the carry from the previous lower significant position.

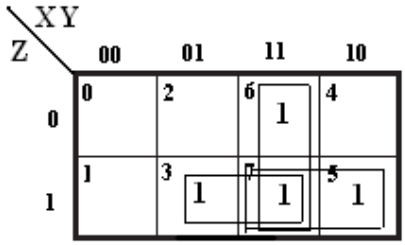
X	Y	Z	S	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$S = \sum m(1, 2, 4, 7)$$

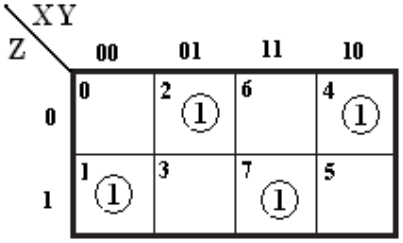
$$C = \sum m(3, 5, 6, 7)$$

Full adder using SOP Expressions:

For Sum S:

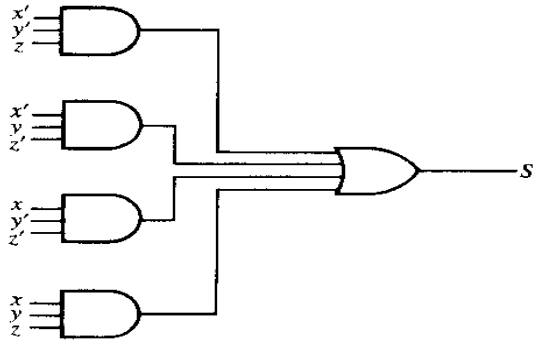


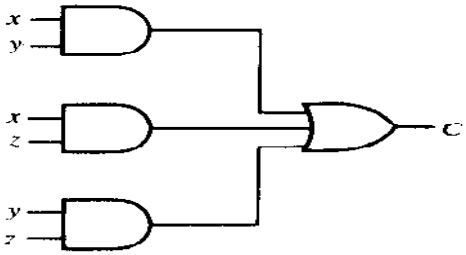
For Carry C:



$$S = X'Y'Z + X'YZ' + XY'Z' + XYZ$$

$$C = XY + YZ + XZ$$



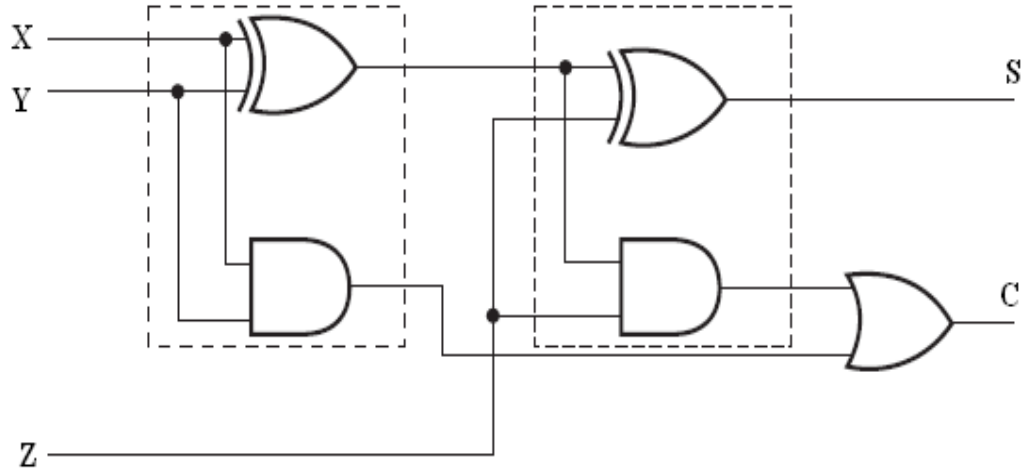


Full adder with two Half adders:

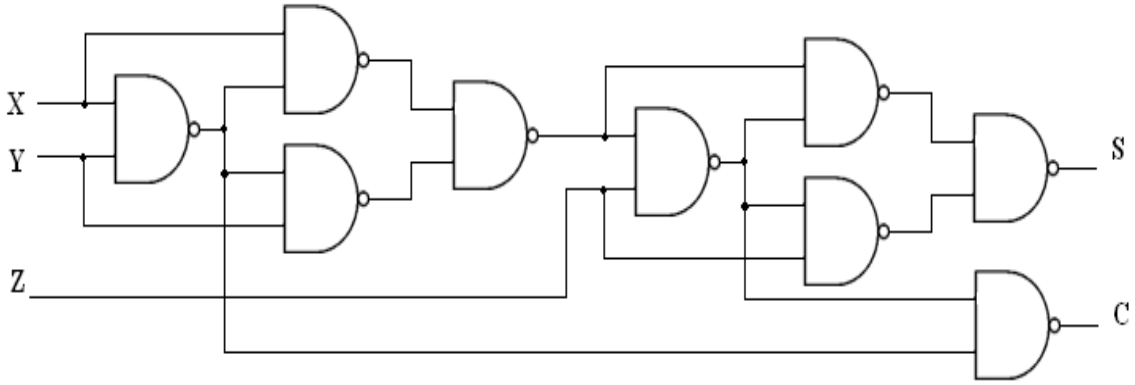
$$S = X'Y'Z + X'YZ' + XY'Z' + XYZ = Z'(X'Y + XY') + Z(XY + X'Y')$$
$$= X \oplus \mathrm{Y} \oplus \mathrm{Z}$$

From the truth table,

$$C = X'YZ + XY'Z + XYZ' + XYZ = Z(X'Y + XY') + XY(Z + Z')$$
$$= XY + Z(X \oplus \mathrm{Y})$$



Full adder using NAND gates:



Subtractors:

In the method of subtraction, each subtrahend bit of the number is subtracted from its corresponding minuend bit to form a difference bit. If the minuend bit is smaller than the subtrahend bit, a 1 is borrowed from the next significant position.

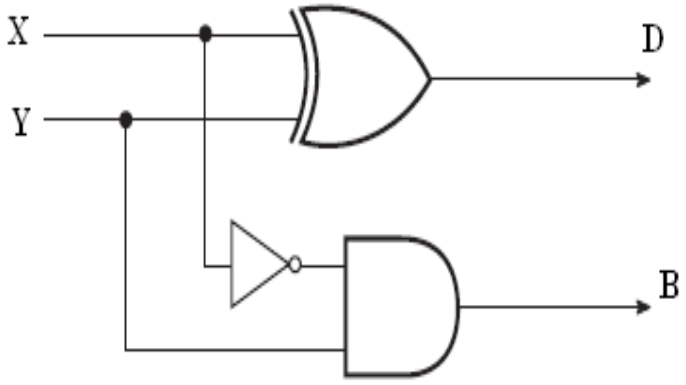
Half-Subtractor:

A *Half-Subtractor* is a combinational circuit that can be used to subtract one binary digit from another to produce a DIFFERENCE output and a BORROW output. The BORROW output here specifies whether a '1' has been borrowed to perform the subtraction.

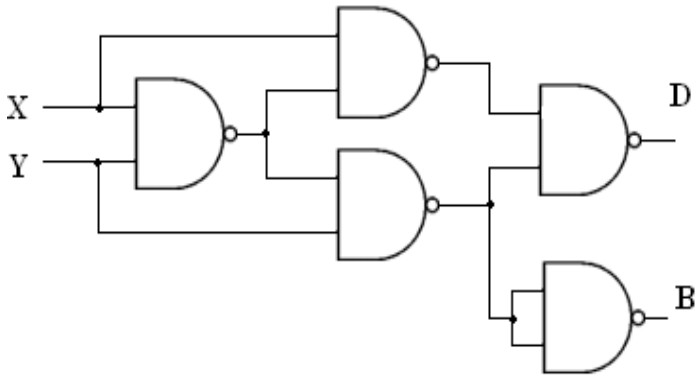
X	Y	D	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$$D = X'Y + XY' = X \oplus Y$$

$$B = X'Y$$



Half-Subtractor using NAND gates:



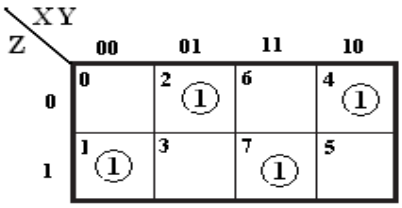
Full-Subtractor:

A full subtractor performs subtraction operation on two bits, a minuend and a subtrahend, and also takes into consideration whether a '1' has already been borrowed by the previous adjacent lower minuend bit or not. This circuit has three inputs and two outputs. The three inputs, X, Y and Z denote the minuend, subtrahend, and previous borrow, respectively. The two outputs, D and B represent the difference and output borrow, respectively.

X	Y	Z	D	B
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

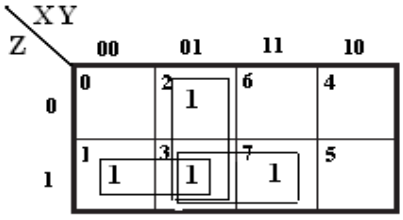
$D = \sum m(1, 2, 4, 7)$ $B = \sum m\,(1, 2, 3, 7)$

For Difference D:



$D = X'Y'Z + X'YZ' + XY'Z' + XYZ$

For Borrow B:



$B = X'Y + X'Z + YZ$

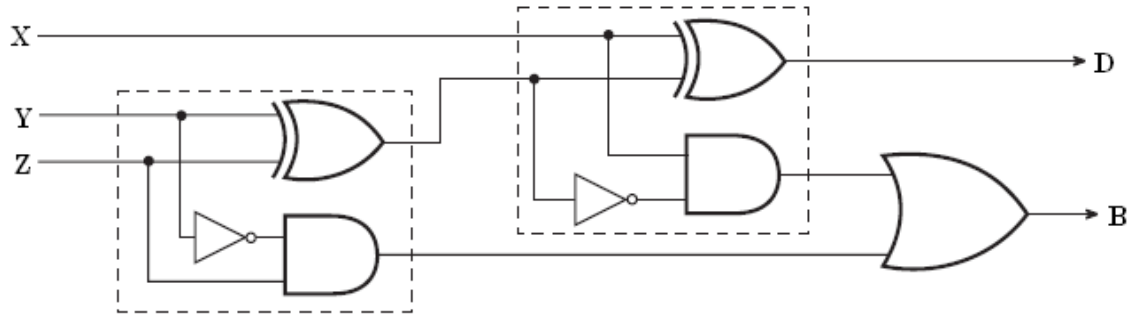
Full-Subtractor using two Half-Subtractors:

$D = X'Y'Z + X'YZ' + XY'Z' + XYZ$

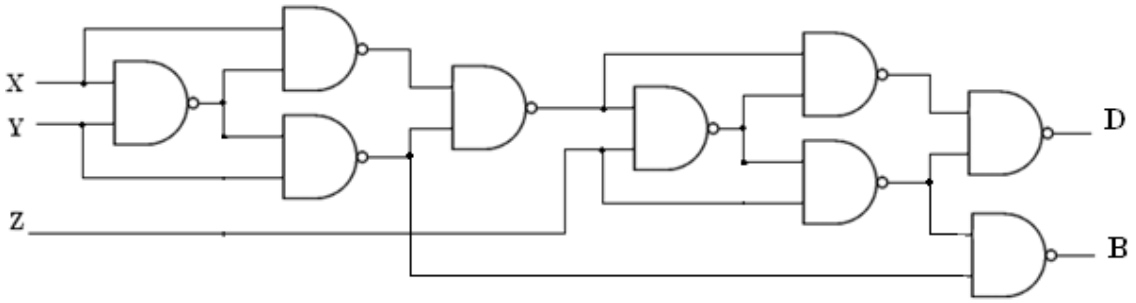
$= Z'(X'Y + XY') + Z(XY + X'Y') = X \oplus \mathrm{Y} \oplus \mathrm{Z}$

From truth table,

$B = X'Y'Z + X'YZ' + X'YZ + XYZ = X'Y + Z(X \oplus \mathrm{Y})'$

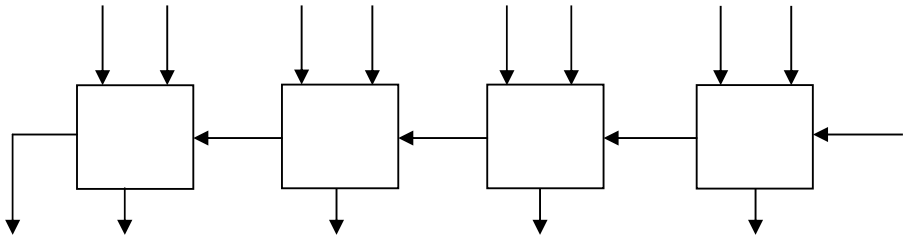


Full-Subtractor using NAND gates:



Binary Parallel Adder:

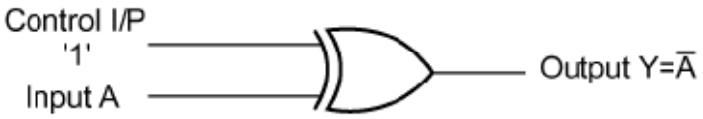
A binary adder is a digital circuit that produces the arithmetic sum of two binary numbers. It can be constructed with full adders connected in cascade, with the output carry from each full adder connected to the input carry of the next full adder in the chain.

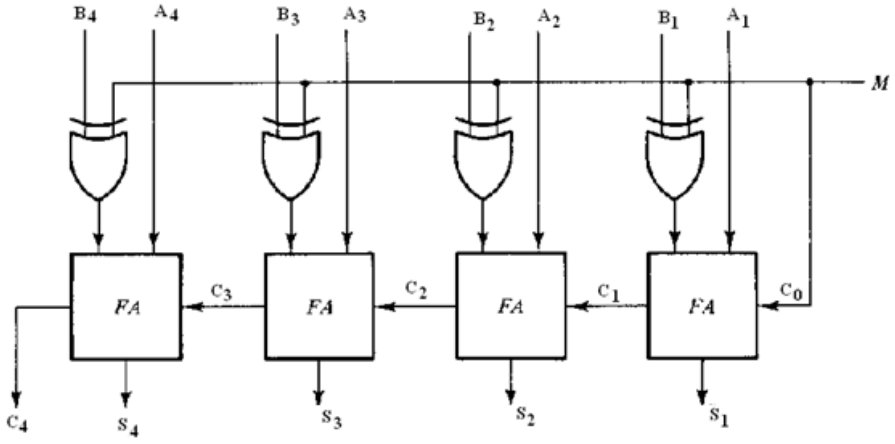


Binary Adder/Subtractor:

A Binary Adder/Subtractor is used to perform both addition and subtraction using a single circuit. Subtraction of two binary numbers can be accomplished by adding

2's complement of the subtrahend to the minuend and disregarding the final carry, if any. Full adders can be used to perform subtraction provided we have the necessary additional hardware to generate 2's complement of the subtrahend and disregard the final carry or overflow. For addition, the addend bits must be added with augend bits and for subtraction, 2's complement of B is added with A. Hence Ex-OR gate is used as controlled inverter to achieve this operation.





The mode input M determines the operation. With $M = 0$, the circuit acts as an adder and the outputs provide sum of the two numbers. When $M = 1$, the circuit acts as a subtractor and the output bits provide the difference of the two inputs.