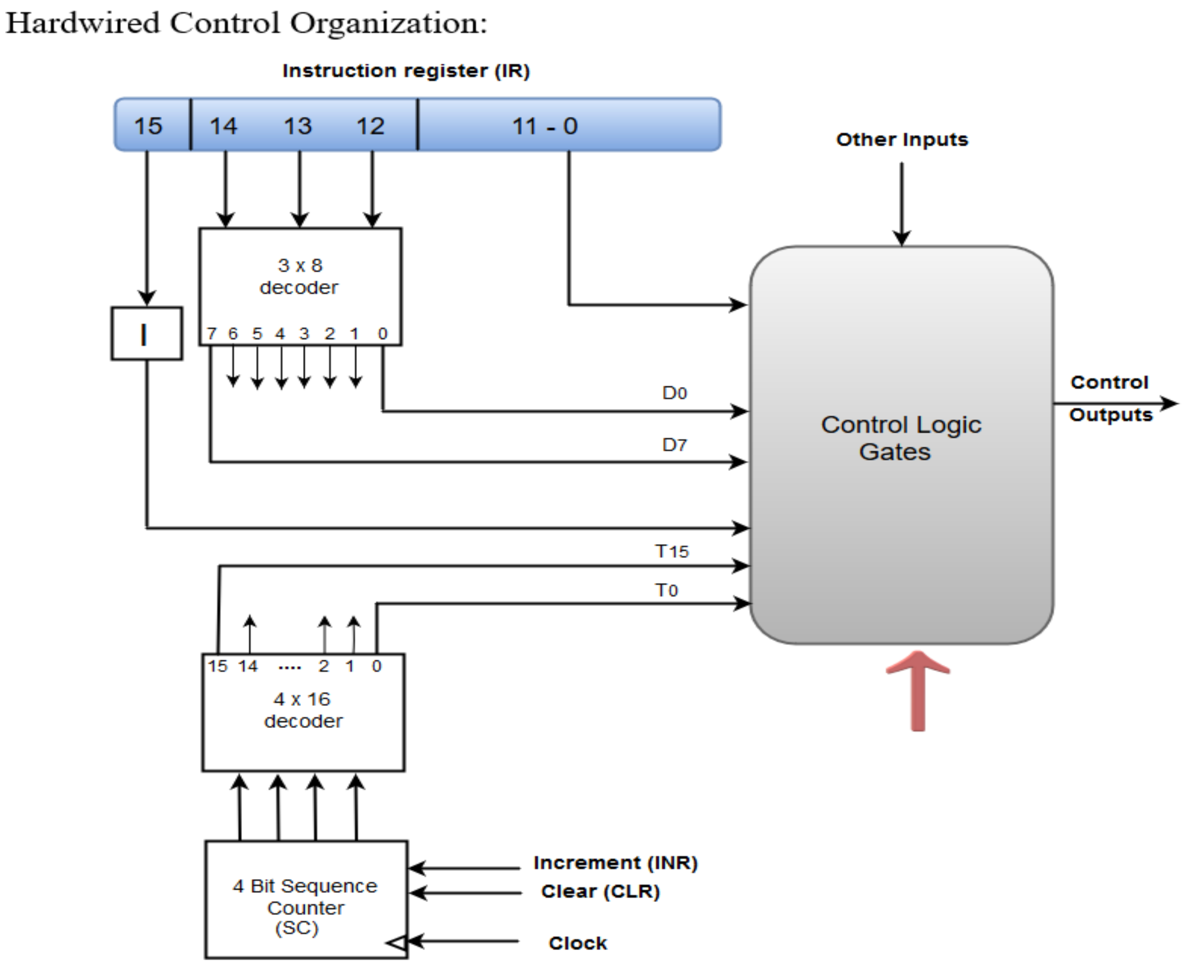
Control Logic Gates

The Control Logic Gate for a basic computer is same as the one used in Hard wired Control organization.

The block diagram is also similar to the Control Logic Gate used in the Hard wired Control organization.



Inputs for the Control Logic Circuit

- The input for the Control Logic circuit comes from the two decoders, I flip-flop and bits 0 through 11 of IR.
- The other inputs to the Control Logic are AC (bits 0 through 15), DR (bits 0 through 15), and the value of the seven flip-flops.

Outputs of the Control Logic Circuit

- The control of the inputs of the nine registers
- The control of the read and write inputs of memory
- To set, clear, or complement the flip-flops

- S2, S1, and SO to select a register for the bus
- The control of the AC adder and logic circuit.

Digital Computers

A Digital computer can be considered as a digital system that performs various computational tasks.

The first electronic digital computer was developed in the late 1940s and was used primarily for numerical computations.

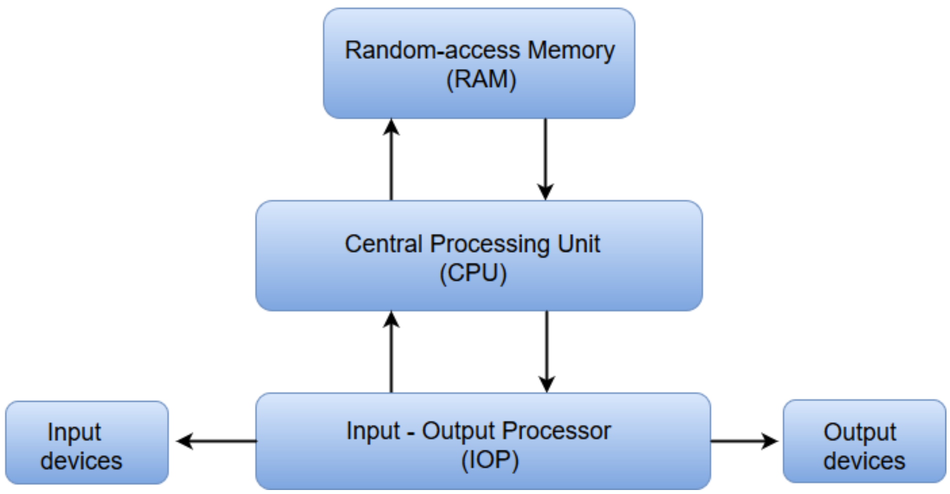
By convention, the digital computers use the binary number system, which has two digits: 0 and 1. A binary digit is called a bit.

A computer system is subdivided into two functional entities: Hardware and Software.

The hardware consists of all the electronic components and electromechanical devices that comprise the physical entity of the device.

The software of the computer consists of the instructions and data that the computer manipulates to perform various data-processing tasks.

Block diagram of a digital computer:



- The Central Processing Unit (CPU) contains an arithmetic and logic unit for manipulating data, a number of registers for storing data, and a control circuit for fetching and executing instructions.
- The memory unit of a digital computer contains storage for instructions and data.
- The Random Access Memory (RAM) for real-time processing of the data.
- The Input-Output devices for generating inputs from the user and displaying the final results to the user.
- The Input-Output devices connected to the computer include the keyboard, mouse, terminals, magnetic disk drives, and other communication devices.

Logic Gates

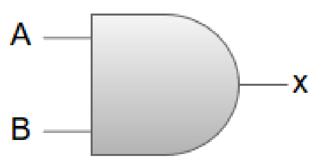
- The logic gates are the main structural part of a digital system.

- Logic Gates are a block of hardware that produces signals of binary 1 or 0 when input logic requirements are satisfied.
- Each gate has a distinct graphic symbol, and its operation can be described by means of algebraic expressions.
- The seven basic logic gates includes: AND, OR, XOR, NOT, NAND, NOR, and XNOR.
- The relationship between the input-output binary variables for each gate can be represented in tabular form by a truth table.
- Each gate has one or two binary input variables designated by A and B and one binary output variable designated by x.

AND Gate

The AND gate is an electronic circuit which gives a high output only if all its inputs are high. The AND operation is represented by a dot (.) sign.

AND Gate:



Algebraic Function: $x = AB$

Truth Table:

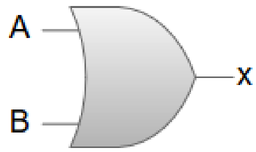
A	B	x
0	0	0
0	1	0
1	0	0
1	1	1

OR Gate

The OR gate is an electronic circuit which gives a high output if one or more of its inputs are high. The operation performed by an OR gate is represented by a plus (+) sign.

OR Gate:

Truth Table:

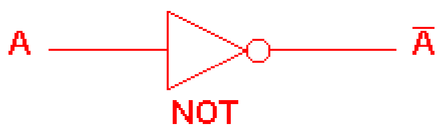


Algebraic Function: $x = A + B$

A	B	x
0	0	0
0	1	1
1	0	1
1	1	1

NOT Gate

The NOT gate is an electronic circuit which produces an inverted version of the input at its output. It is also known as an **Inverter**.



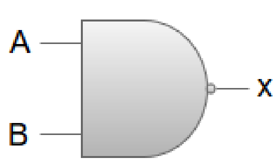
NOT gate	
A	$\overline{A}$
0	1
1	0

NAND Gate

The NOT-AND (NAND) gate which is equal to an AND gate followed by a NOT gate. The NAND gate gives a high output if any of the inputs are low. The NAND gate is represented by a AND gate with a small circle on the output. The small circle represents inversion.

NAND Gate:

Truth Table:



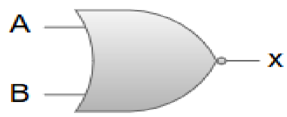
Algebraic Function: $x = (AB)'$

A	B	x
0	0	1
0	1	1
1	0	1
1	1	0

NOR Gate

The NOT-OR (NOR) gate which is equal to an OR gate followed by a NOT gate. The NOR gate gives a low output if any of the inputs are high. The NOR gate is represented by an OR gate with a small circle on the output. The small circle represents inversion.

NOR Gate:



Algebraic Function: $x = (A+B)'$

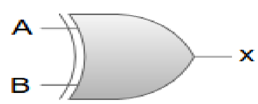
Truth Table:

A	B	x
0	0	1
0	1	0
1	0	0
1	1	0

Exclusive-OR/ XOR Gate

The 'Exclusive-OR' gate is a circuit which will give a high output if one of its inputs is high but not both of them. The XOR operation is represented by an encircled plus sign.

XOR Gate:



Algebraic Function: $x = A \oplus B$

or

$x = A'B + AB'$

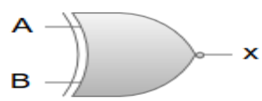
Truth Table:

A	B	x
0	0	0
0	1	1
1	0	1
1	1	0

EXCLUSIVE-NOR/Equivalence Gate

The 'Exclusive-NOR' gate is a circuit that does the inverse operation to the XOR gate. It will give a low output if one of its inputs is high but not both of them. The small circle represents inversion.

Exclusive-NOR Gate:



Algebraic Function: $x = (A \oplus B)'$

or

$x = A'B' + AB$

Truth Table:

A	B	x
0	0	1
0	1	0
1	0	0
1	1	1

Boolean Algebra

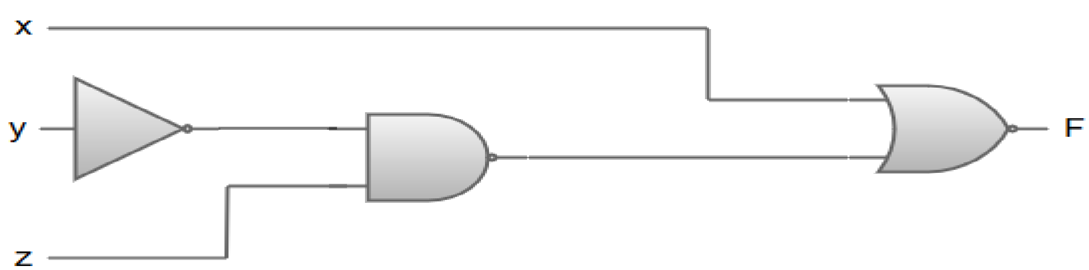
Boolean algebra can be considered as an algebra that deals with binary variables and logic operations. Boolean algebraic variables are designated by letters such as A, B, x, and y. The basic operations performed are AND, OR, and complement.

The Boolean algebraic functions are mostly expressed with binary variables, logic operation symbols, parentheses, and equal sign. For a given value of variables, the Boolean function can be either 1 or 0. For instance, consider the Boolean function:

$F = x + y'z$

The logic diagram for the Boolean function $F = x + y'z$ can be represented as:

$F = x + y'z$



- The Boolean function $F = x + y'z$ is transformed from an algebraic expression into a logic diagram composed of AND, OR, and inverter gates.
- Inverter at input 'y' generates its complement y'.
- There is an AND gate for the term y'z, and an OR gate is used to combine the two terms (x and y'z).
- The variables of the function are taken to be the inputs of the circuit, and the variable symbol of the function is taken as the output of the circuit.

The truth table for the Boolean function F = x + y'z can be represented as:

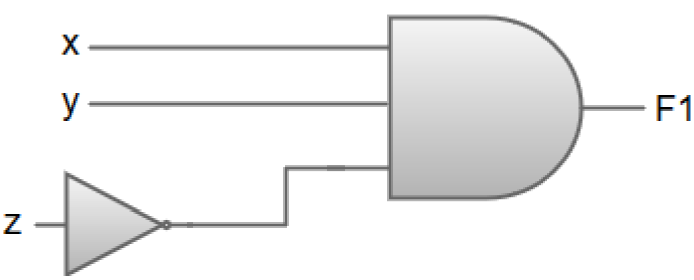
F = x + y'z

x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Examples of Boolean algebra simplifications using logic gates

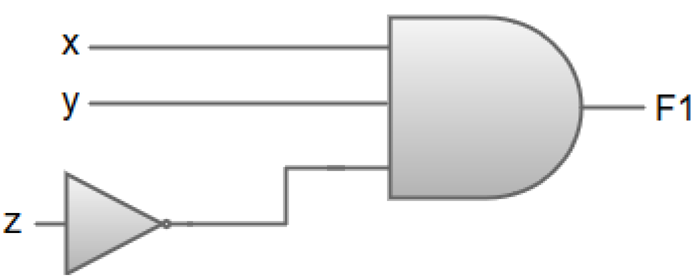
In this section, we will look at some of the examples of Boolean algebra simplification using Logic gates.

1. F1 = xyz'



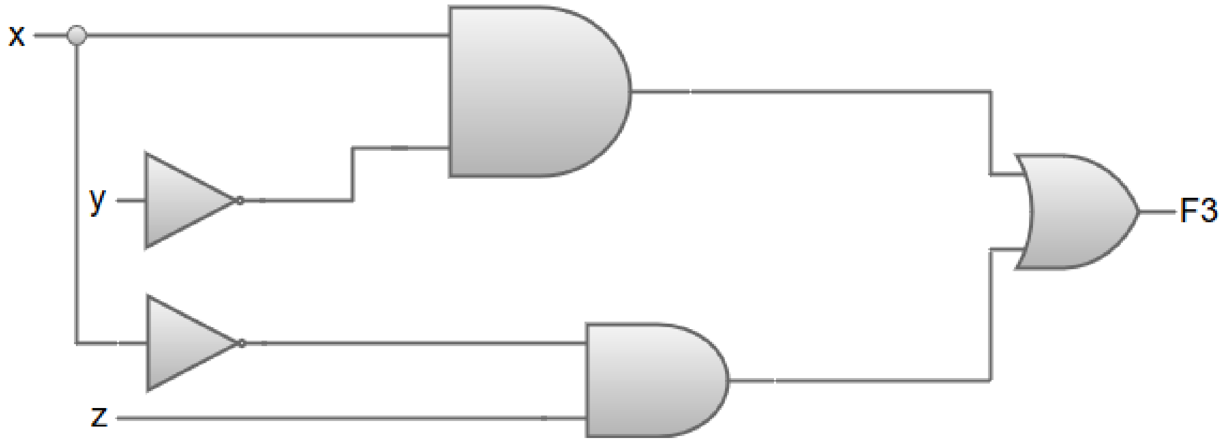
2. F2 = x + y'z

1. F1 = xyz'

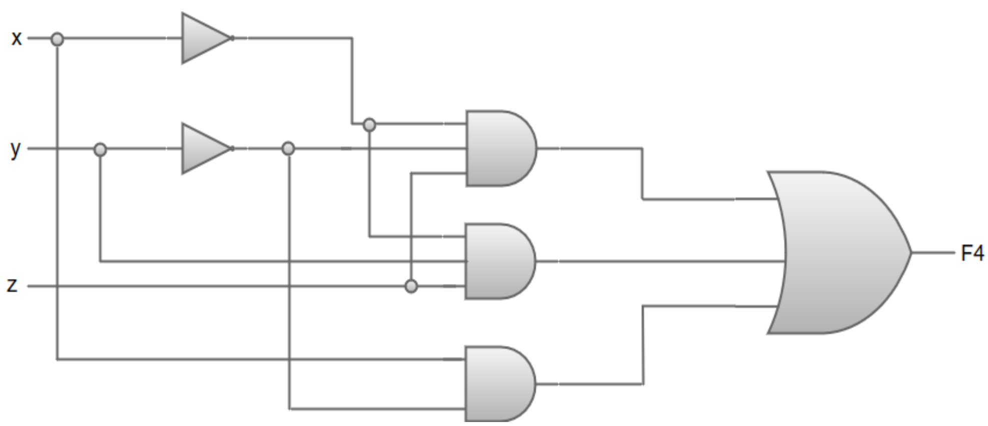


2. F2 = x + y'z

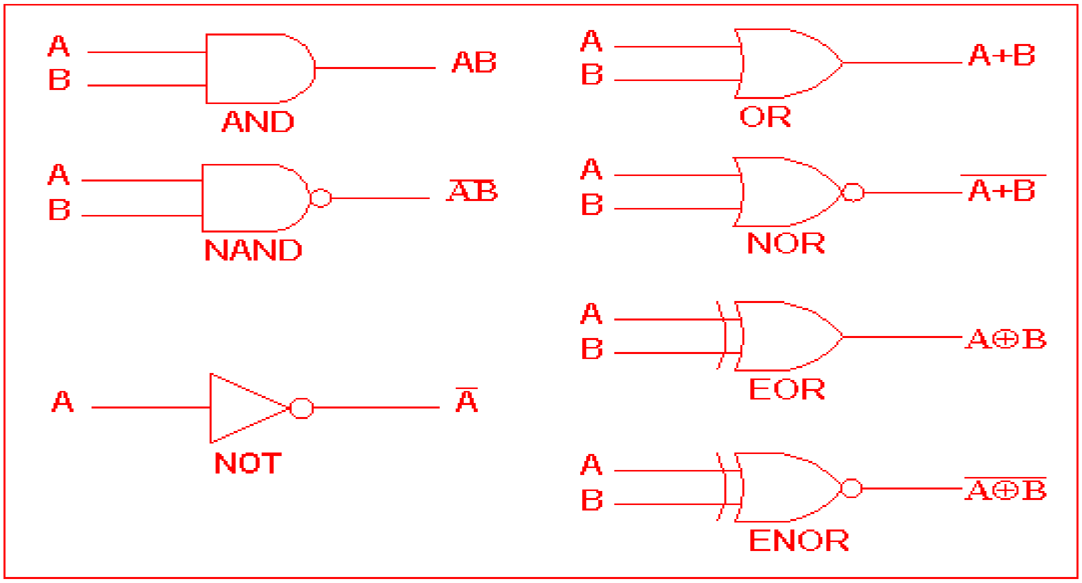
3. F3 = xy' + x'z



1. F4 = x'y'z + x'yz + xy'



Logic gate Symbols



INPUTS		OUTPUTS					
A	B	AND	NAND	OR	NOR	EXOR	EXNOR
0	0	0	1	0	1	0	1
0	1	0	1	1	0	1	0
1	0	0	1	1	0	1	0
1	1	1	0	1	0	0	1

Logic gates representation using the Truth table

A NAND gate can be used as a NOT gate using either of the following wiring configurations.

(You can check this out using a truth table.)

Laws of Boolean Algebra

The basic Laws of Boolean Algebra can be stated as follows:

- Commutative Law states that the interchanging of the order of operands in a Boolean equation does not change its result. For example:
    1. OR operator → A + B = B + A
    2. AND operator → A * B = B * A
- Associative Law of multiplication states that the AND operation are done on two or more than two variables. For example:
  A * (B * C) = (A * B) * C
- Distributive Law states that the multiplication of two variables and adding the result with a variable will result in the same value as multiplication of addition of the variable with individual variables. For example:
  A + BC = (A + B) (A + C).

- Annulment law:
  A.0 = 0
  A + 1 = 1
- Identity law:
  A.1 = A
  A + 0 = A
- Idempotent law:
  A + A = A
  A.A = A
- Complement law:
  A + A' = 1
  A.A'= 0
- Double negation law:
  ((A)')' = A
- Absorption law:
  A.(A+B) = A
  A + AB = A

De Morgan's Law is also known as De Morgan's theorem, works depending on the concept of Duality. Duality states that interchanging the operators and variables in a function, such as replacing 0 with 1 and 1 with 0, AND operator with OR operator and OR operator with AND operator.

De Morgan stated 2 theorems, which will help us in solving the algebraic problems in digital electronics. The De Morgan's statements are:

1. "The negation of a conjunction is the disjunction of the negations", which means that the complement of the product of 2 variables is equal to the sum of the compliments of individual variables. For example, (A.B)' = A' + B'.
2. "The negation of disjunction is the conjunction of the negations", which means that compliment of the sum of two variables is equal to the product of the complement of each variable. For example, (A + B)' = A'B'.