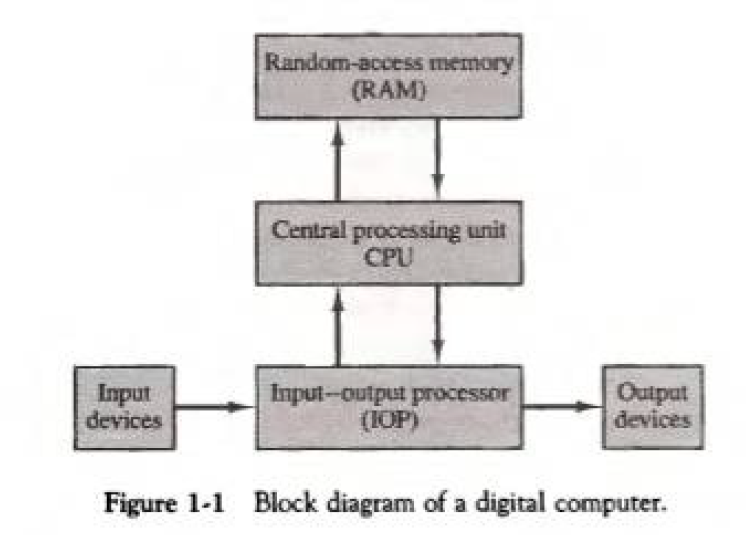
* 1. **Digital Circuits**
* Digital Computers function more reliably if only two states are used because of the physical restrictions of components and because human logic tends to be binary.
* A computer is divided into two functional entities –
* **Hardware:** Consists of all electronic components and all electromechanical devices that comprise the physical entity of the device.
* **Software:** Consists of instructions and data that the computer manipulates to perform various data-processing tasks.
* **Program:** A sequence of instructions for the computer.
* **Database:** Data that are manipulated by the program.
* **Operating System:** The programs included in the system software package. Its function is to compensate the differences that exist between user needs and the capability of the hardware. System software of a computer consists of a collection of programs whose purpose is to make more effective use of the computer.

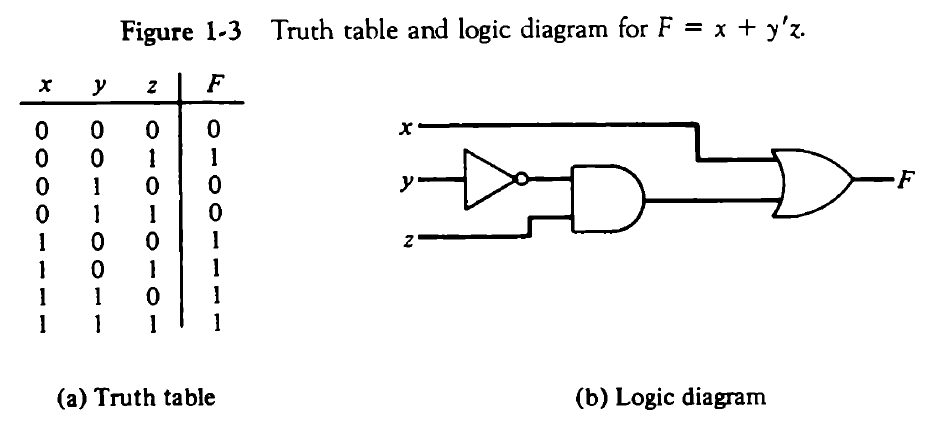


* **Computer Organisation:** Concerned with the way the hardware components operate and the way they are connected to form the computer system.
* **Computer Design:** Concerned with the design of the hardware system.
* **Computer Implementation:** Concerned with the determination of what hardware should be used and how the parts should be connected.
* **Computer Architecture:** Concerned with the structure and behaviour of the computer as seen by the user. Includes information formats, instruction sets and techniques for addressing memory.
  1. **Logic Gates**
* **Signals:** Physical quantities that represent binary information in digital computers, e.g. electrical signals like voltages.
* **Gates:** Blocks of hardware that produce signals of binary 1 or 0 when input logic requirements are satisfied.
* **Truth Tables:** Tabular form to represent the input-output relationship of the binary variables for each gate.

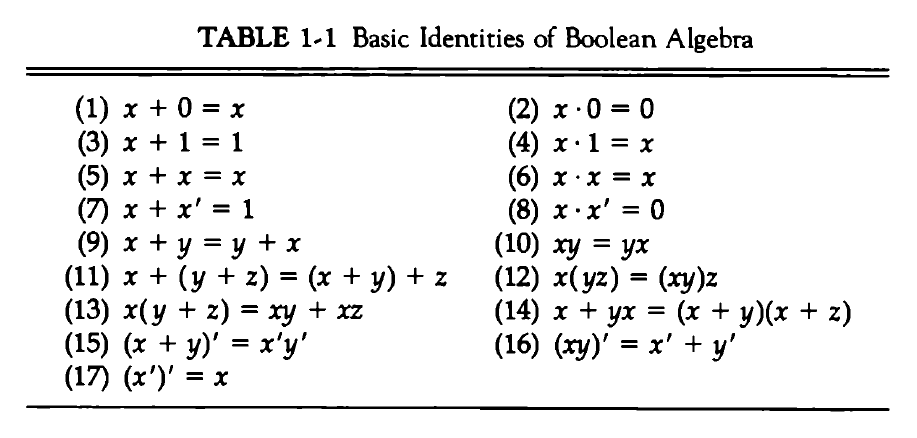
A table of maths with numbers and symbols

Description automatically generated with medium confidence

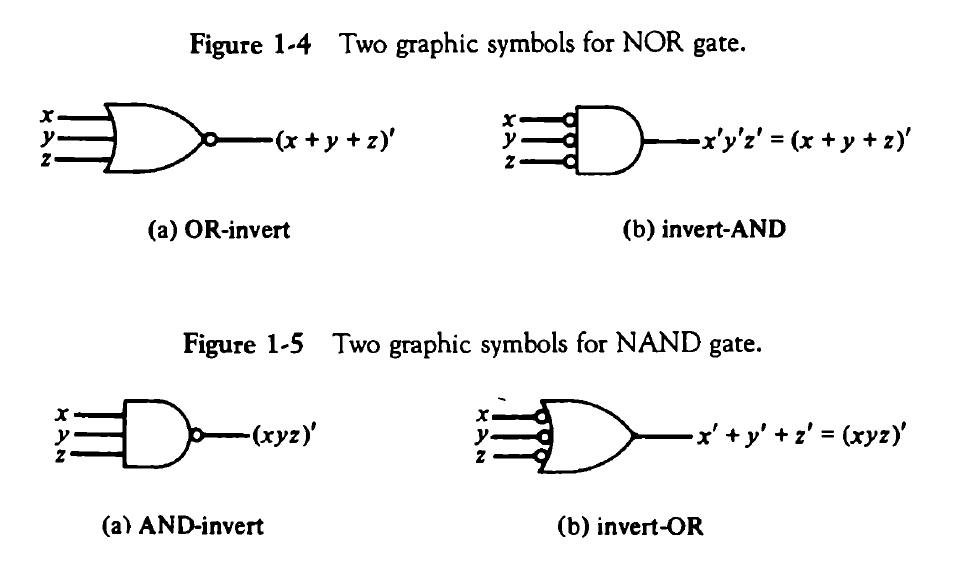
* 1. **Boolean Algebra**
* **Boolean Algebra:** Algebra that deals with binary variables and logic operations.



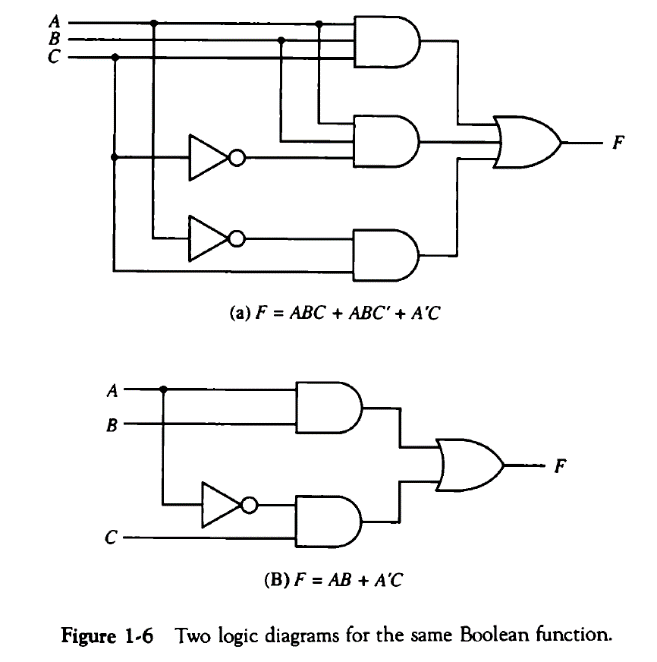
* The basic identities of Boolean algebra are used throughout and make calculations much easier.



* NOR and NAND gates have two different symbols for their logic diagram and this is made obvious by the D’ Morgan Laws which are stated by (15) and (16) in the above diagram.

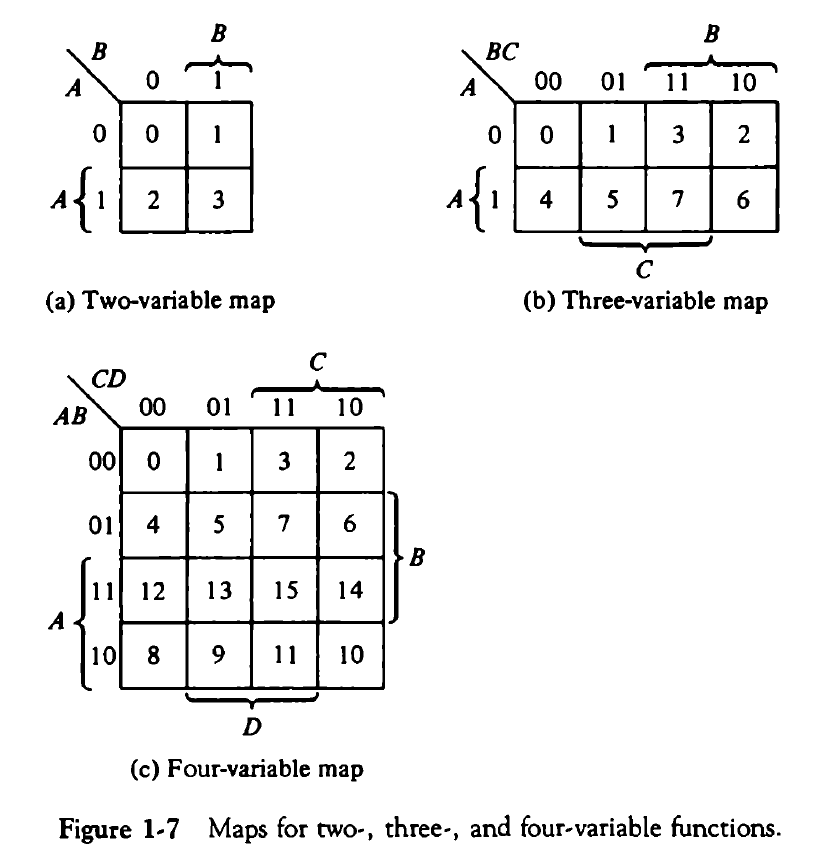


* The following diagram shows two logic circuits, one before simplifying the logical expression and one after simplifying it.



**1.4 Map Simplification**

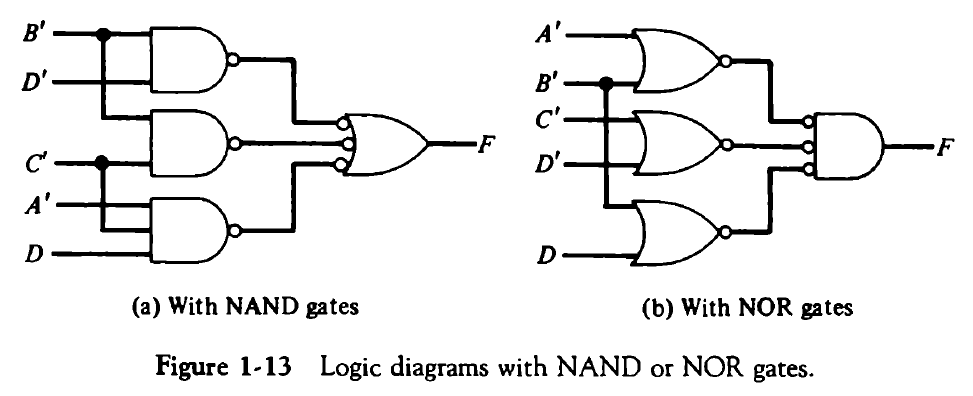
* **Minterm:** Each combination of variables in a truth table. A function of n variables will have 2n minterms.
* **Karnaugh Map (or K-Map):** Provides a simple, straightforward procedure for simplifying Boolean expressions.



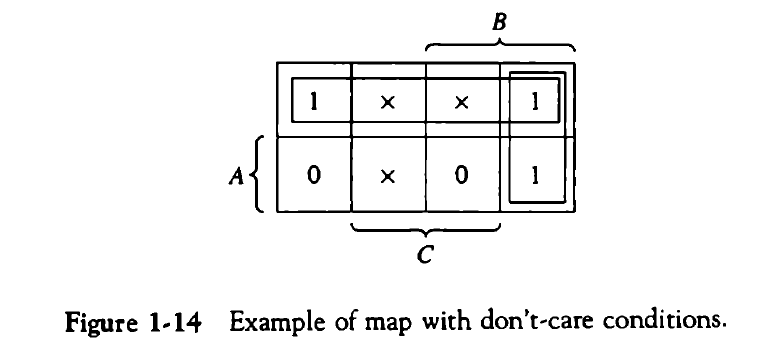
* **Sum-of-products** expression can be implemented with NAND gates.
* **Product-of-sums** expression can be implemented with NOR gates.

A diagram of a product

Description automatically generated

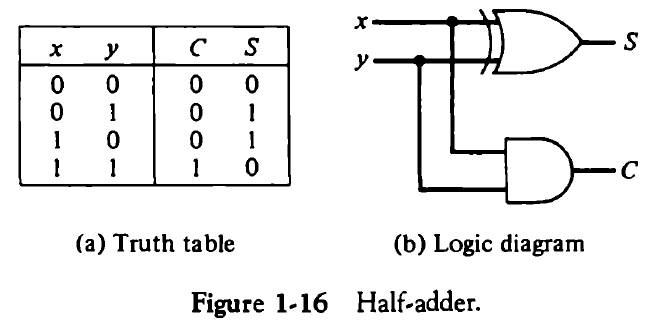


* **Don’t Care Conditions:** There are conditions when it does not matter whether the function produces a 0 or 1 for a given minterm.
* We say we *don’t care* what the output of the function is going to be.
* Don’t care minterms are marked with an X.
* Don’t care conditions are used for simplifying expressions.



**1.5 Combinational Circuits**

* **Combinational Circuit:** A connected arrangement of logic gates with a set of inputs and outputs.
* The binary values of the outputs are a combination of the binary combination of inputs.
* The n binary input variables come from an internal source.
* The m binary output variables go to an external destination.
* **Half-Adder:** A combinational circuit that performs the arithmetic addition of two bits is called half-adder.
* Input variables are called the augend and addend bits.
* Output variables are called the sum and carry bits.

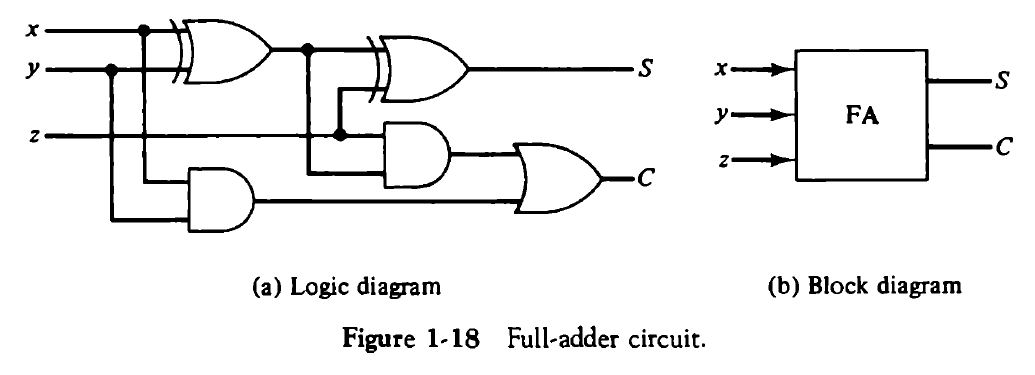


* Boolean functions for the two addends can be obtained as follows –

S = x’y + xy’ = x Ꚛ y

C = xy

* **Full-Adder:** A combinational circuit that forms the arithmetic sum of three input bits.
* The input bits are called augend, addend, and carry.
* The output bits are called sum and carry.



A table with numbers and symbols

Description automatically generated

* Boolean functions for a full-adder can be obtained as follows –

S = x Ꚛ y Ꚛ z

C = xy + (x Ꚛ y)z