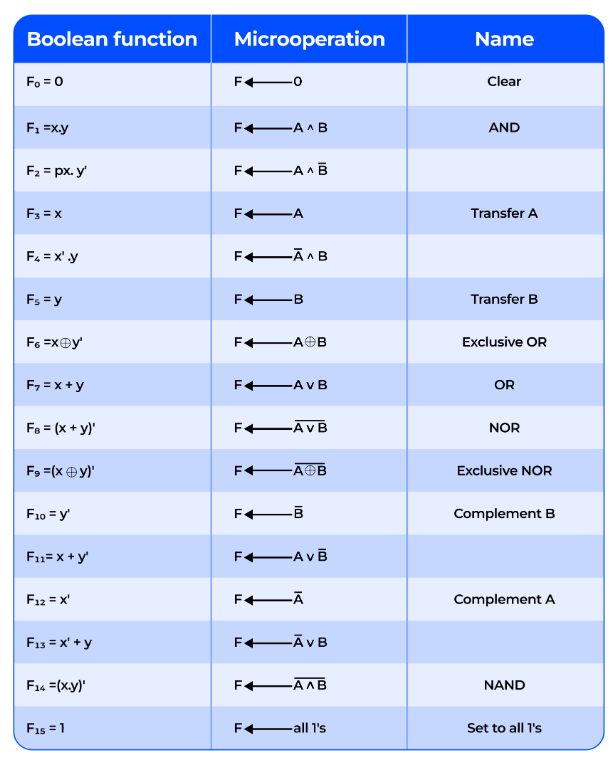
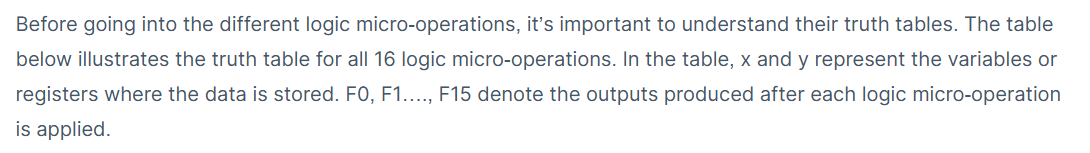
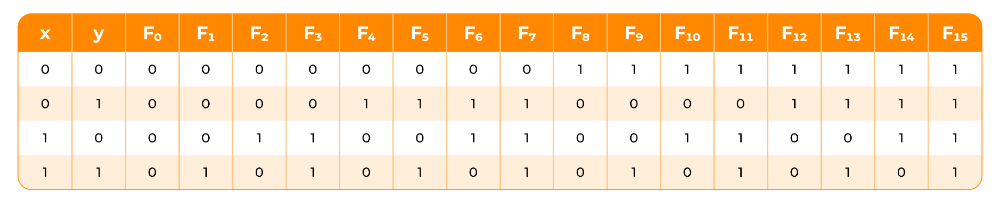
Logic Micro Operations

Logic micro-operations involve binary operations performed on registers containing strings of bits. Each bit within the register is treated individually and functions as an independent binary variable. This bitwise approach allows for a total of 16 distinct logic operations that can be performed:







Truth Table of 16 logic micro-operators

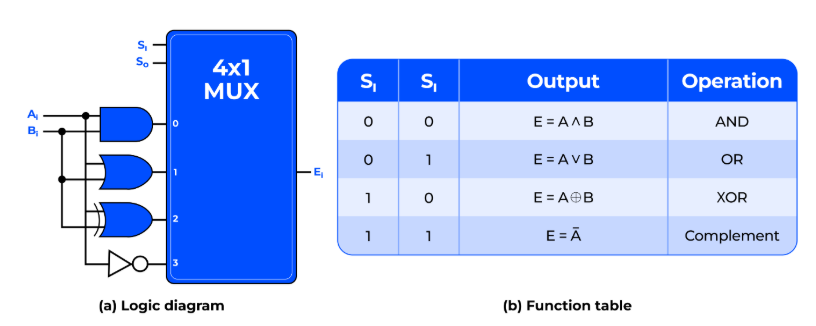
So, what does each logic micro-operation do? Let’s take a look:

1. **Clear:** This logic micro-operation sets all bits in the register to 0. Hence, it’s effectively resetting the data. It’s used to initialize or erase the contents of a register.
2. **AND:** The AND operation performs a bitwise AND between corresponding bits of two registers. The output bit is 1 only if both input bits are 1. Otherwise, it is 0.
3. **Transfer A:** Transfer A copies the contents of register A directly to the output without any changes.
4. **Transfer B:** The Transfer B operation copies the contents of register B directly to the output without any changes.
5. **Exclusive OR:** The Exclusive OR (XOR) performs a bitwise comparison between two registers. The output bit is 1 if the corresponding bits of the inputs are different, and it is 0 if they are the same.
6. **OR:** The OR performs a bitwise OR between corresponding bits of two registers. The output bit is 1 if at least one of the input bits is 1, otherwise it is 0.
7. **NOR:** This operation performs a bitwise OR operation followed by negation. The output bit is 1 only if both input bits are 0. Otherwise, the output is 0. It’s used to invert the result of an OR operation.
8. **Exclusive NOR:** The Exclusive NOR operation performs a bitwise comparison between two registers. It produces an output of 1 when the corresponding bits are the same and 0 when they are different.
9. **Complement B:** Complement B inverts all the bits in register B. It changes 1s to 0s and 0s to 1s. This is for performing negation or generating the complement of the data.
10. **Complement A:** The Complement A operation inverts all the bits in register A. It changes 1s to 0s and 0s to 1s.
11. **NAND:** The NAND logic micro-operation performs a bitwise AND operation between two registers and then negates the result. The output bit is 0 only if both input bits are 1. Otherwise, the output is 1. It’s the inverse of the AND operation.

**Set to all 1’s:** operation sets all the bits in a register to 1. This is used to initialize a register.

Implementing logic micro-operations in hardware involves using logic gates for each bit or pair of bits in the registers to carry out the desired logic function. Although there are 16 possible logic micro-operations, most computers rely on just four: **AND, OR, XOR (Exclusive OR), and NOT**.

**HARDWARE IMPLEMENTATION**



Logic Diagram of Hardware Implementation of Logic Circuit

The hardware implementation, as shown in the diagram, consists of a single stage that generates the four basic logic micro-operations. It uses four logic gates and a multiplexer. Each gate performs one of the required logic functions: AND, OR, XOR, or NOT.  Their outputs are fed into the multiplexer’s data inputs.

The multiplexer then uses two selection inputs, S1 and S0, to choose which data input to pass to the output. This configuration represents one stage, and for a circuit handling n bits, this setup would be replicated n times to accommodate all the bits.