ARITHMETIC LOGIC UNIT OPERATIONS

LECTURER DALIA ADIL

1.Logical operations

- **Logical operations** in a CPU refer to a set of operations that manipulate binary values (0s and 1s) using logical functions. These operations are fundamental to decision-making processes in programming and data manipulation. The most common logical operations include:

1. **AND**: Produces a true (1) output only if both operands are true (1). For example, 1 AND 1=11 AND 1=1, but 1 AND 0=01 AND 0=0.
2. **OR**: Produces a true output if at least one of the operands is true. For example, 1 OR 0=11 OR 0=1.
3. **NOT**: A unary operation that inverts the value of its operand. For instance, NOT 1=0NOT1=0 and NOT 0=1NOT0=1.
4. **XOR (Exclusive OR)**: Outputs true only when the operands are different. For example, 1 XOR 0=11 XOR 0=1, but 1 XOR 1=01 XOR 1=0.
5. Logical operations are used in various applications, such as conditional statements, bit manipulation, and control flow in programming .

2. BITs_Shifting Operations

- **BITs_Shifting -operations** are those operations that are used for the serial transfer of information. These are also used in conjunction with arithmetic operation, logic operation, and other data-processing operations. There are three types of shift operations:

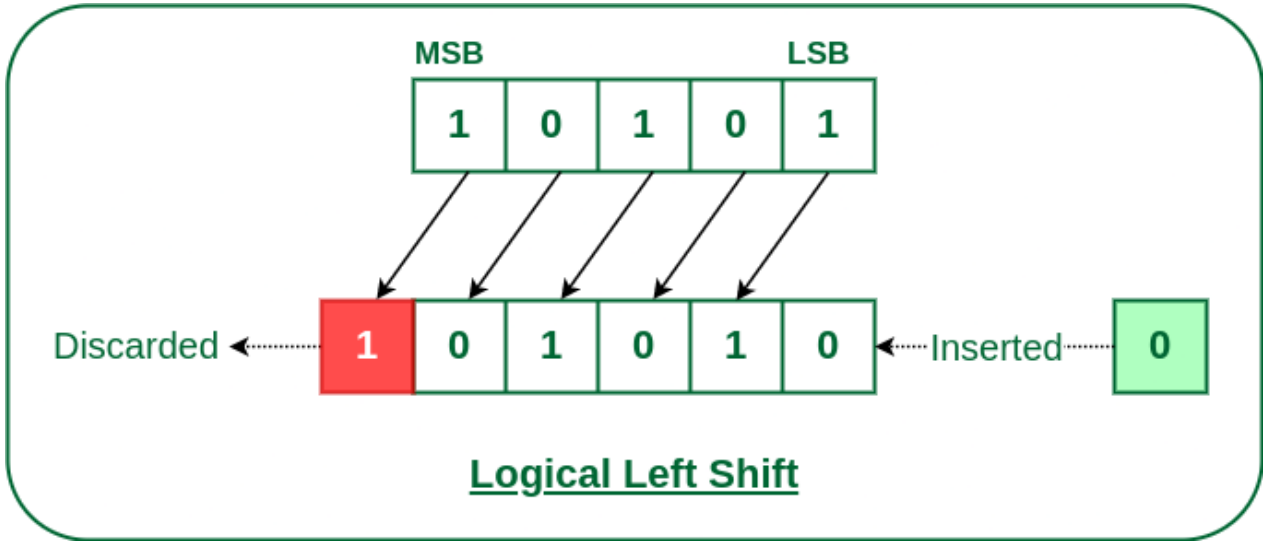
1.Logical Shift

- It transfers the 0 zero through the serial input. We use the symbols '**<<**' for the logical left shift and '**>>**' for the logical right shift.

a.Logical Left Shift

In this shift, one position moves each bit to the left one by one. The Empty least significant bit (LSB) is filled with zero (i.e, the serial input), and the most significant bit (MSB) is rejected.

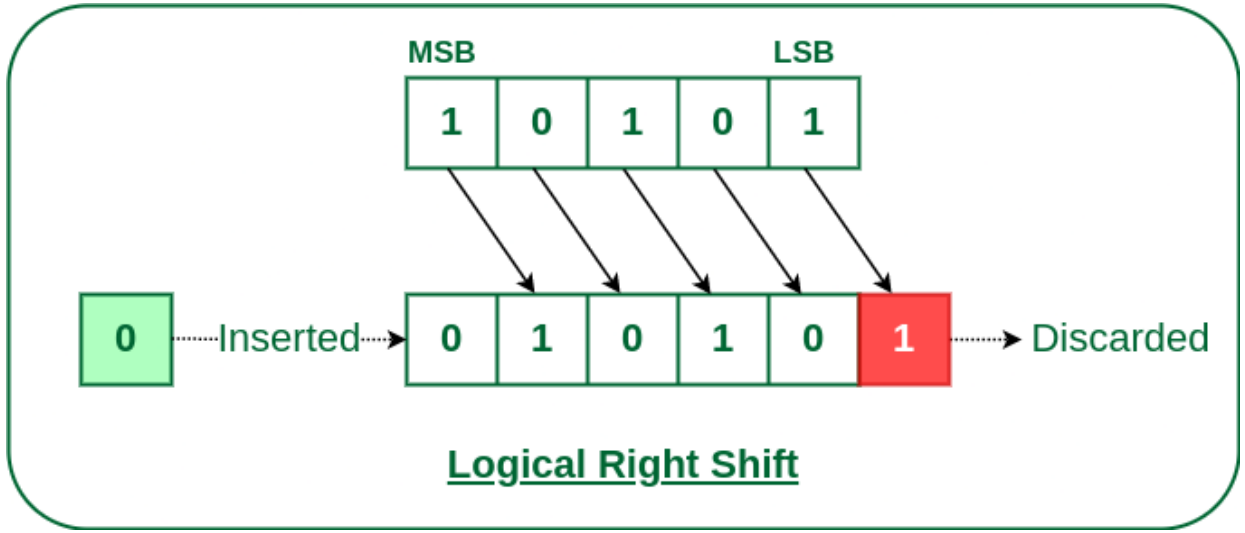
Note: Every time we shift a number towards the left by 1 bit it multiplies that number by 2.



b.Logical Right Shift

- *In this shift, each bit moves to the right one by one and the least significant bit(LSB) is rejected and the empty MSB is filled with zero.*

- **Note:** Every time we shift a number towards the right by 1 bit it divides that number by 2.



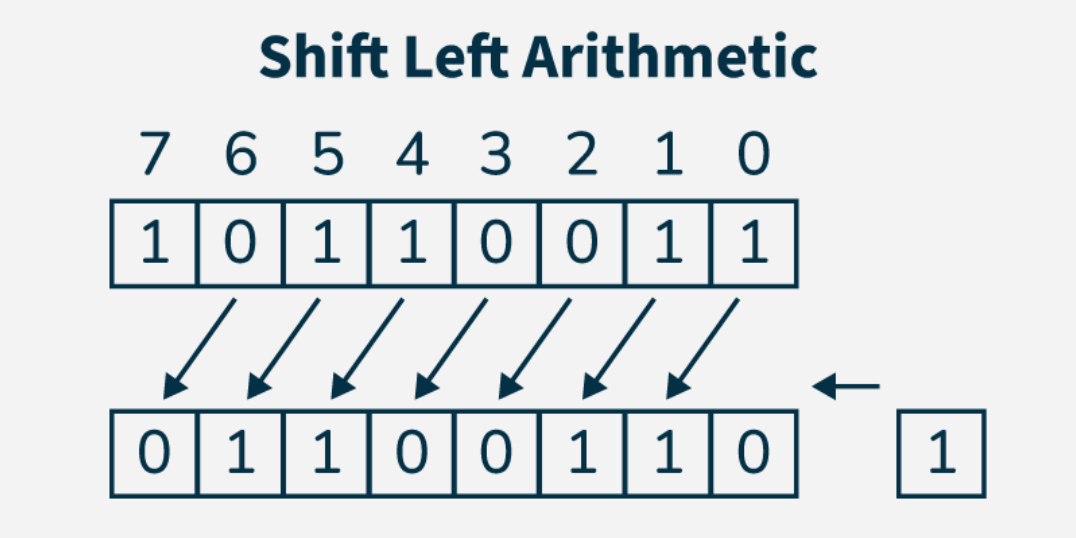
2. Arithmetic Shift

- The arithmetic shift operation moves the signed binary number either to the left or to the right position. Following are the two ways to perform the arithmetic shift.

2. Arithmetic Left Shift
3. Arithmetic Right Shift

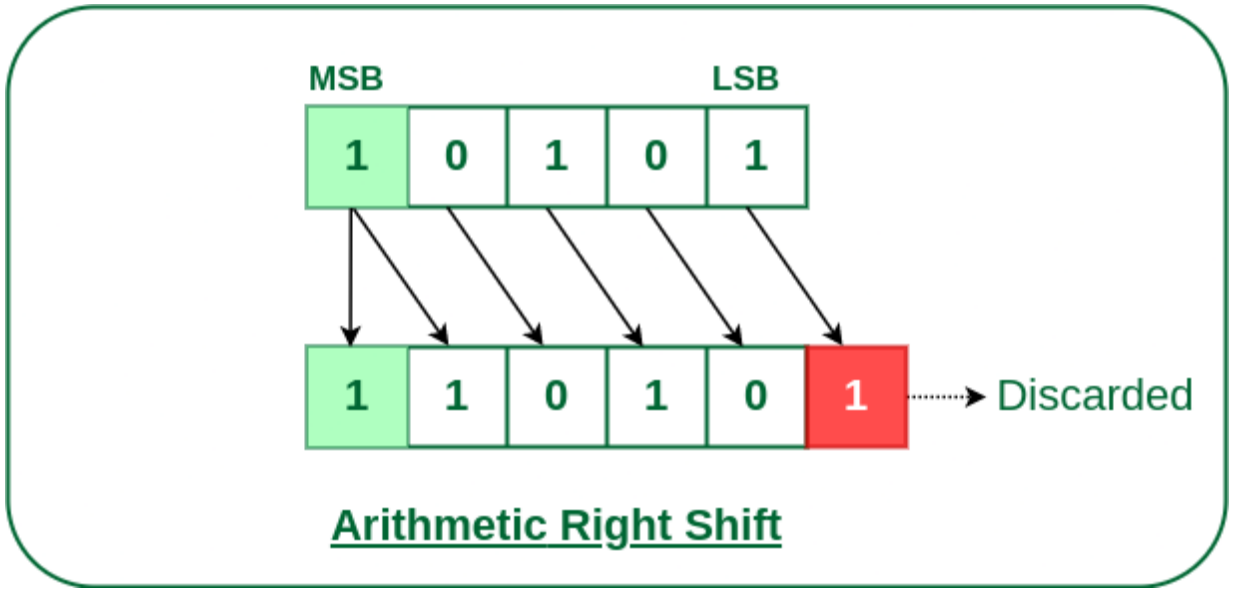
a. Arithmetic Left Shift

- *In this shift, each bit is moved to the left one by one. The empty least significant bit (LSB) is filled with zero and the most significant bit (MSB) is rejected. Same as the Left Logical Shift.*



b.Arithmetic Right Shift

- *In this shift, each bit is moved to the right one by one and the least significant(LSB) bit is rejected and the empty most significant bit(MSB) is filled with the value of the previous MSB*



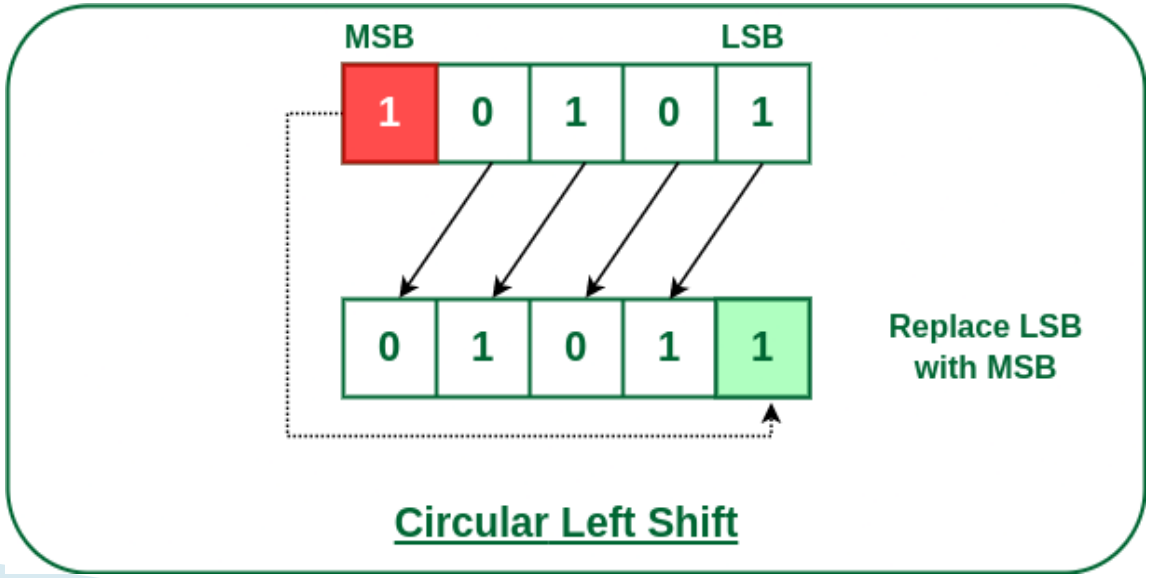
3.Circular Shift

- The circular shift circulates the bits in the sequence of the register around both ends without any loss of information.
  Following are the two ways to perform the circular shift.

2. Circular Shift Left
3. Circular Shift Right

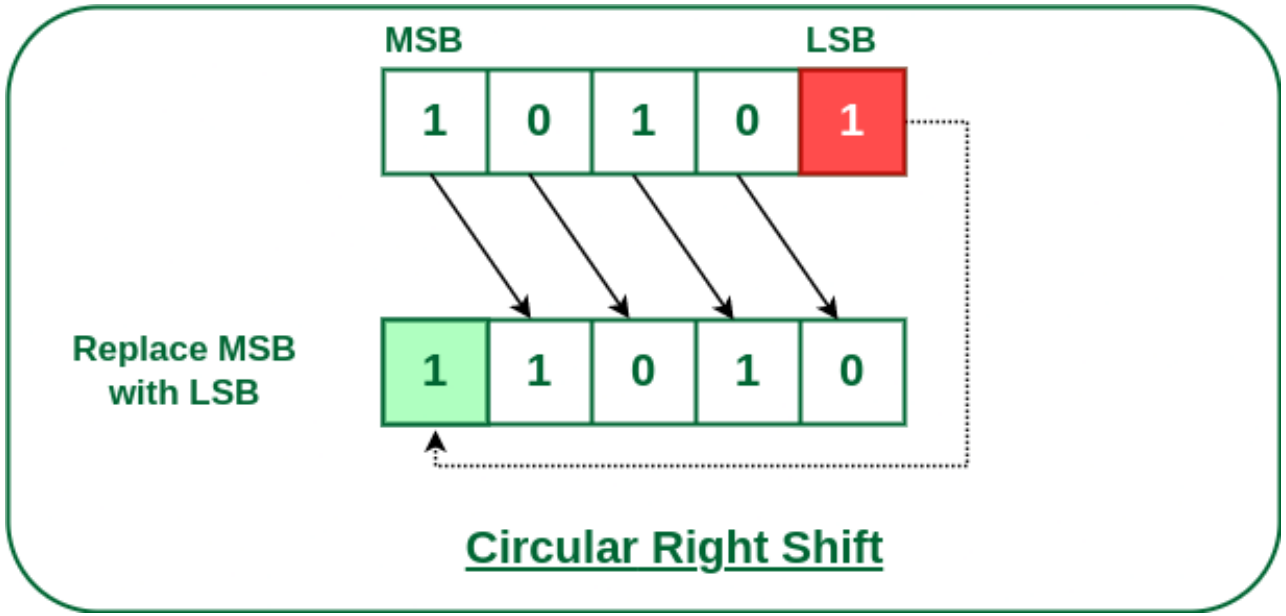
a. Circular Left Shift

- *In this shift operation each bit in the register is shifted to the left one by one. After shifting, the LSB becomes empty, so the value of the MSB is filled in there.*



b. Circular Right Shift

- *In this shift operation each bit in the register is shifted to the right one by one. After shifting, the MSB becomes empty, so the value of the LSB is filled in there.*



3. Arithmetic operations

- in a CPU are fundamental operations that perform mathematical calculations on numerical values. These operations are typically executed by the Arithmetic Logic Unit (ALU) within the CPU. The most common arithmetic operations include:
- **Addition**: Combines two numbers to produce their sum.
- **Subtraction**: Calculates the difference between two numbers.
- **Multiplication**: Finds the product of two numbers.
- **Division**: Divides one number by another to find the quotient.
- **Modulo**: Computes the remainder of a division operation.

complex arithmetic functions

- In addition to these basic operations, CPUs often support more **complex arithmetic functions**, such as:
- **Floating-point arithmetic**: For calculations involving real numbers, which includes operations like addition, subtraction, multiplication, and division of decimal numbers.
- **Integer overflow and underflow handling**: To manage situations where calculations exceed the maximum or minimum limits of data types.
- **Bitwise operations**: While primarily logical, they can also serve arithmetic purposes, especially in low-level programming and optimization.
- These operations are crucial for executing instructions in software applications, ranging from simple calculations to complex algorithms.