**Date: 19-04-2021**

**Branch: CSE (III Year) and EE (III Year)**

**Topic:** Arithmetic Micro-Operations, Logic Micro-Operations,

Shift Micro-Operations, Arithmetic logical shift unit.

**Time: 11:00 AM -12:00 AM**

**Arithmetic Micro-operations in Registers**

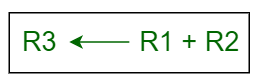
We can perform arithmetic operations on the numeric data which is stored inside the registers.

**Example :**

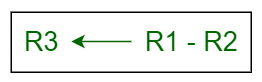
R3 <- R1 + R2

The value in register R1 is added to the value in the register R2 and then the sum is transferred into register R3. Similarily, other arithmetic micro-operations are performed on the registers.

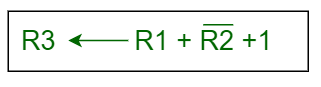
* **Addition –**  
  In addition micro-operation, the value in register R1 is added to the value in the register R2 and then the sum is transferred into register R3.

[](https://media.geeksforgeeks.org/wp-content/uploads/20200709233907/add10.png)

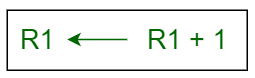
* **Substraction –**  
  In subtraction micro-operation, the contents of register R2 are subtracted from contents of the register R1, and then the result is transferred into R3.

[](https://media.geeksforgeeks.org/wp-content/uploads/20200709233913/sub11.png)

There is another way of doing the subtraction. In this, 2’s complement of R2 is added to R1, which is equivalent to **R1 – R2**, and then the result is transferred into register R3.

[](https://media.geeksforgeeks.org/wp-content/uploads/20200709233917/sub21.png)

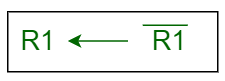
* **Increment –**  
  In Increment micro-operation, the value inside the R1 register is increased by 1.

[](https://media.geeksforgeeks.org/wp-content/uploads/20200709233911/incr.png)

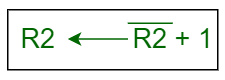
* **Decrement –**  
  In Decrement micro-operation, the value inside the R1 register is decreased by 1.

[](https://media.geeksforgeeks.org/wp-content/uploads/20200709233909/decr.png)

* **1’s Complement –**  
  In this micro-operation, the complement of the value inside the register R1 is taken.

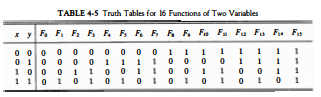
[](https://media.geeksforgeeks.org/wp-content/uploads/20200709234034/ones_comp.png)

* **2’s Complement –**  
  In this micro-operation, the complement of the value inside the register R2 is taken and then 1 is added to the value and then the final result is transferred into the register R2. This process is also called Negation. It is equivalent to **-R2**.

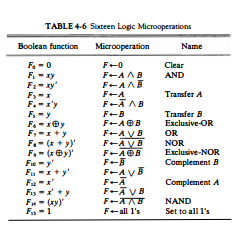
[](https://media.geeksforgeeks.org/wp-content/uploads/20200709234035/twos_comp.png)

[**Logic Micro operations**](https://padakuu.com/article/499-logic-microoperations)

There are 16 different logic operations that can be performed with two binary variables. They can be determined from all possible truth tables obtained with two binary variables as shown in Table 4-5. In this table, each of the 16 columns F0 through F15 represents a truth table of one possible Boolean function for the two variables x and y. Note that the functions are determined from the 16 binary combinations that can be assigned to F.



The 16 Boolean functions of two variables x and y are expressed in algebraic form in the first column of Table 4-6. The 16 logic microoperations are derived from these functions by replacing variable x by the binary content of register A and variable y by the binary content of register B. It is important to realize that the Boolean functions listed in the first column of Table 4-6 represent a relationship between two binary variables x and y. The logic microoperations listed in the second column represent a relationship between the binary content of two registers A and B. Each bit of the register is treated as a binary variable and the microoperation is performed on the string of bits stored in the registers.



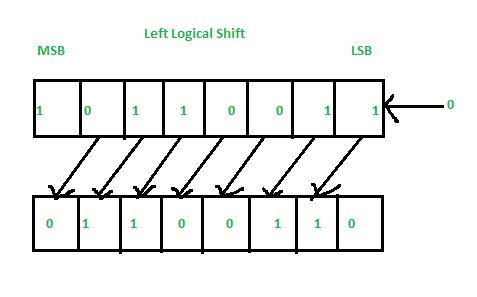
**Shift Micro-Operations**

Shift micro-operations are those micro-operations that are used for serial transfer of information. These are also used in conjunction with arithmetic micro-operation, logic micro-operation, and other data-processing operations.

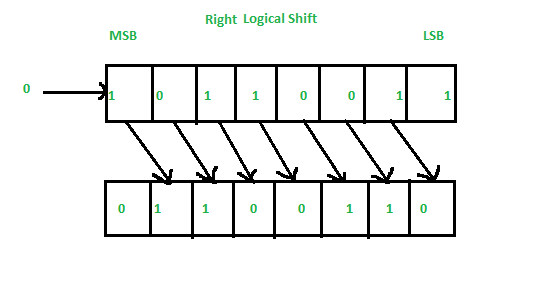
There are three types of shifts micro-operations:

**1. Logical :**  
It transfers the 0 zero through the serial input. We use the symbols shl for logical shift-left and shr for shift-right.

1. **Logical Shift Left –**  
   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.

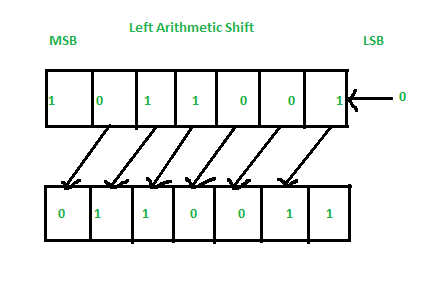


1. **Right Logical Shift –**  
   In this one position moves each bit to the right one by one and the least significant bit(LSB) is rejected and the empty MSB is filled with zero.

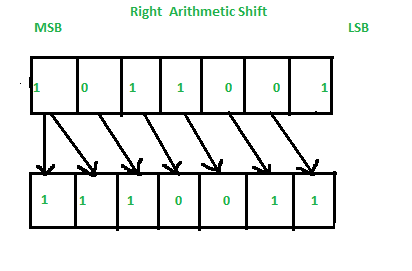


**2. Arithmetic :**  
This micro-operation shifts a signed binary number to the left or to the right position. In an arithmetic shift-left, it multiplies a signed binary number by 2 and In an arithmetic shift-right, it divides the number by 2.

1. **Left Arithmetic Shift –**  
   In this one position moves each bit 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.

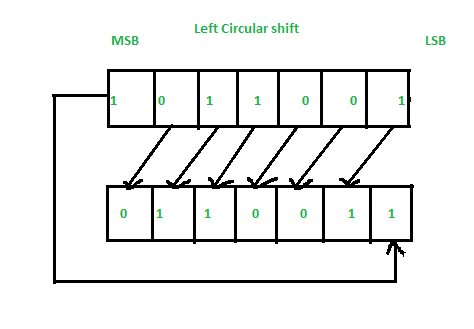


1. **Right Arithmetic Shift –**  
   In this one position moves each bit to the right one by one and the least significant bit is rejected and the empty MSB is filled with the value of the previous MSB.

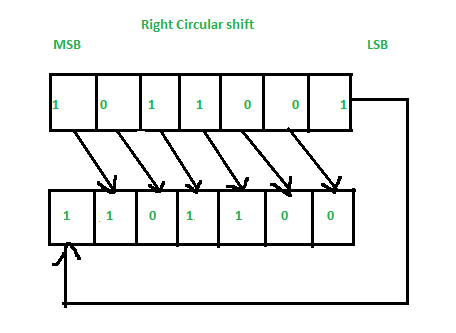


**3. Circular :**  
The circular shift circulates the bits in the sequence of the register around the both ends without any loss of information.

1. **Left Circular Shift –**



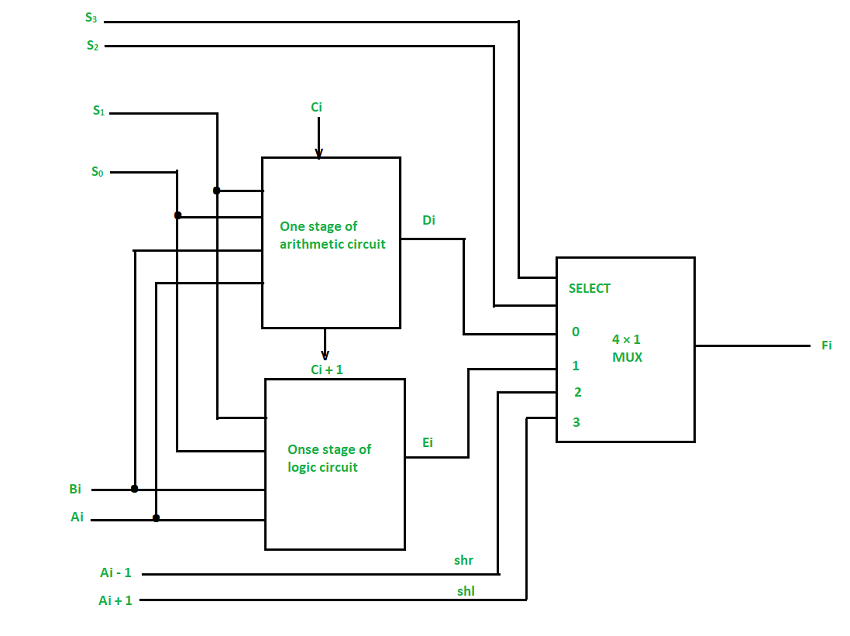
1. **Right Circular Shift –**



**Arithmetic Logic Shift Unit in Computer Architecture**

**Arithmetic Logic Shift Unit (ALSU)** is a member of the [Arithmetic Logic Unit (ALU)](https://www.geeksforgeeks.org/introduction-of-alu-and-data-path/) in a computer system. It is a digital circuit that performs logical, arithmetic, and shift operations. Rather than having individual registers calculating the micro operations directly, the computer deploys a number of storage registers which is connected to a common operational unit known as an arithmetic logic unit or ALU.

Now, to implement the micro operation, the contents of specified registers are allocated in the inputs of the common Arithmetic Logic Unit. The Arithmetic Logic Unit performs an operation that leads as a result and gets transferred to a destination register. Arithmetic Logic Unit may be a combinatory circuit in order that the complete register transfer operation from the supply registers through the ALU and into the destination register is performed throughout one clock pulse amount. Sometimes, the shift micro operations are performed in a separate unit, but sometimes it is made as a part of full ALU.



***One stage of ALSU***

We can combine and make one ALU with common selection variables by adding arithmetic, logic, and shift circuits. We can see the, One stage of an arithmetic logic shift unit in the diagram below. Some particular micro operations are selected through the inputs S1 and S0.

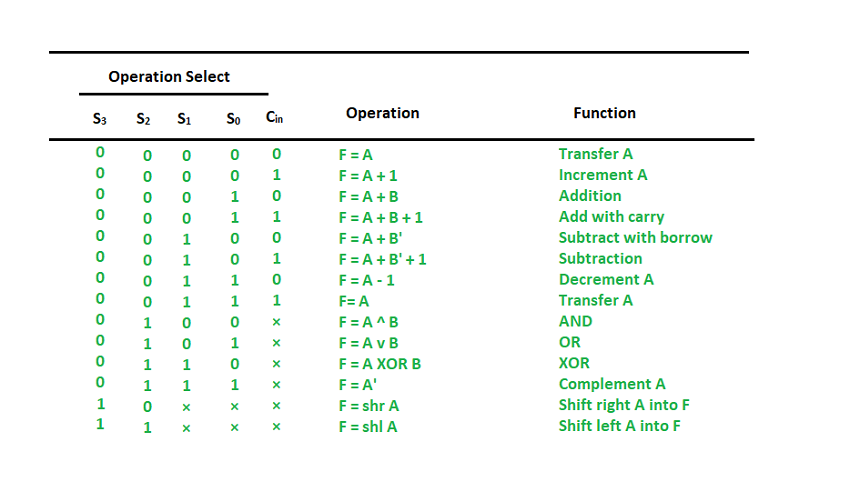
4 x 1 multiplexer at the output chooses between associate arithmetic output between Ei and a logic output in Hi. The data in the multiplexer are selected through inputs S3 and S2 and the other two data inputs to the multiplexer obtain the inputs Ai – 1 for the ***shr*** operation and Ai + 1 for the ***shl*** operation.

**Note:** The output carry Ci + 1 of a specified arithmetic stage must be attached to the input carry Ci of the next stage in the sequence.

The circuit whose one stage is given in the below diagram provides 8 arithmetic operations, 4 logic operations, and 2 shift operations, and Each operation is selected by the 5 variables S3, S2, S1, S0, and Cin.

The below table shows the 14 operations perform by the Arithmetic Logic Unit:

1. The first 8 are arithmetic operations which are selected by S3 S2 = 00
2. The next 4 are logic operations which are selected by S3 S2 = 01
3. The last two are shift operations which are selected by S3 S2 = 10 & 11



***Function table of ALSU***