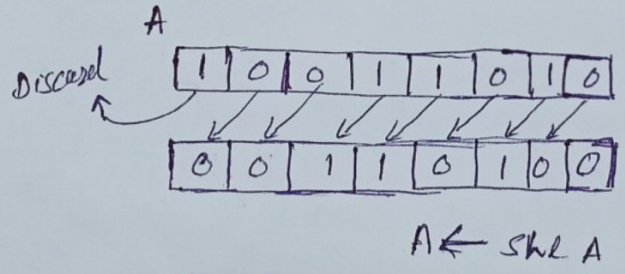
Shift Micro-operation :-

Shift is a useful operation, which can be used for serial transfer of data. Shift operations can also be used along with other operations. For example, for implementing a multiply operation, arithmetic micro-operation (addition) can be used along with the shift operation. The shift operation may result in shifting the contents of a register to the left or right. Depending on what bit enters the register and where the shift-out bit goes, the shifts are classified in three types. These are:

(i) logical (ii) Circular (iii) Arithmetic.

(i) Logical shift :- Logical shift operation is of two types (a) Left shift (b) Right shift.

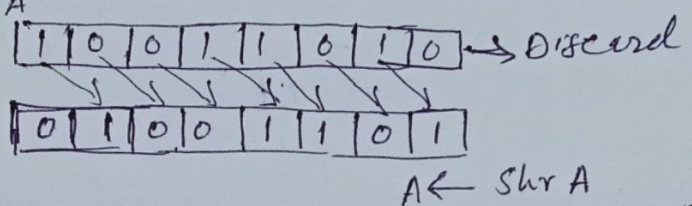
(a) logical left shift :- Each bit of the register is shifted towards left. For example,



A ← Shl A

Initially the content of register A was 10011010 after, shift (left) operation, the content of register A become, ⊙0110100.

(b) logical Right shift :- Each bit of the register is shifted towards right. For example

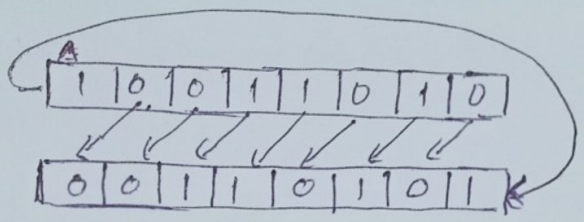


A ← Shr A

The content of register A after right shift operation becomes ⊙1001101.

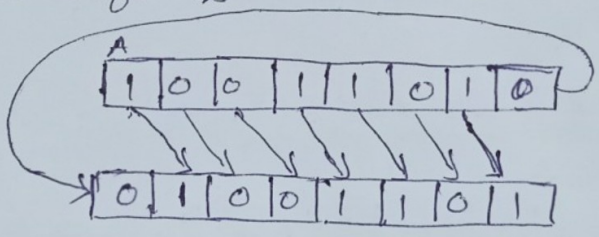
(ii) Circular shift micro-operations.

In this shift operation, the MSB bit of the register are virtually connected to the LSB bit of the same register. So, in this case of shift operation (left or right), the data doesnot get lost. Instead the data moves in a circular fashion. This can also be circular left shift and circular right shift. For example,



A ← cil A

This is the example of circular left shift. Example of circular right shift is



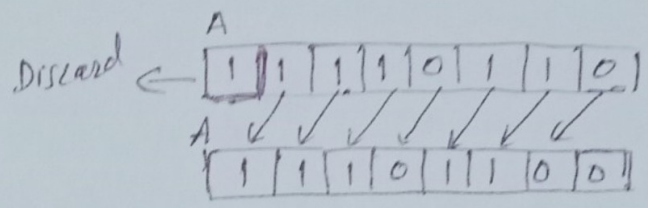
A ← cir A

The content of register A after circular left shift is 00110101 and after circular right shift, the content is 01001101.

(iii) Arithmetic Shift:- Arithmetic shift is a micro-operation that shifts a signed binary number to the left or right. An arithmetic shift left multiplies a signed binary number by 2 and arithmetic shift right divides the number by 2. But, as the division or multiplication by 2, the sign of a number should not be changed, therefore, arithmetic shift must leave the sign bit unchanged.

Example of arithmetic shift-left is as

Let A be an 8-bit register, such that



Decimal eqv of A = -10

Hence content of A after left-shift = 11101100

Since Sign bit is 1, number is negative,

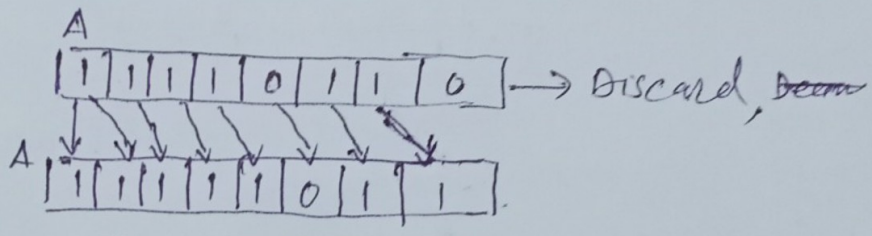
taking 2's complement of 11101100 as

```
  00010011
         1
----------
  00010100
```

∴ Content of A after left-shift is -20

Hence, arithmetic left-shift multiplies a signed binary number by 2.

Arithmetic shift-right is as



Decimal eqv. of signed binary no.

= -10 (before shifting)

After right-shift operation, the content of

A = 11111011

Since Sign bit is 1, the number is -ve

2's complement of 11111011

```
  00000100
         1
----------
  00000101   → 4+1=5
```

∴ content of after right-shift operation is -5.

Hence, arithmetic right-shift divides a signed binary number by 2.