Bit Shift

- Bit Shift is process of moving binary representation of a digit left or right.
- Type of shifts:
  - Logical Left Shift
  - Logical Right Shift
  - Arithematic Right Shift
  - Arithematic Left Shift

Logical Left Shift

- When shifting left
  - The most-significant bit is lost,
  - And a 0 bit is inserted on the other end.
- A single left shift multiplies a binary number by 2.
- The left shift operator is usually written as "<<" or same keyword is used in programming languages.
- Example:

```
1010 << 1  = 0101
Here each digit is shifted by 2 places, most-significant bit is lost
1001 << 2 = 0100
Here each digit shifted by 2 places, most-significant bit is lost, 0
```

```
MSB                        LSB
1   0   1   1   0   0   1   1   ←     0
⇙   ⇙   ⇙   ⇙   ⇙   ⇙   ⇙   ⇙
0   1   1   0   0   1   1   0
```

Logical Right Shift

- When shifting right with a logical right shift
  - The least-significant bit is lost
  - And a 00 is inserted on the other end.
- For positive numbers
  - A single logical right shift divides a number by 2, throwing out any remainders.
- Example:

```
1011 >>> 1  →  0101
1011 >>> 3  →  0001
```

Arithematic Right Shift

- When shifting right with an arithmetic right shift
  - The least-significant bit is lost
  - And the most-significant bit is copied.
- Example:

```
1011 >> 1  →  1101
1011 >> 3  →  1111 , in both cased most most-significant bit (here

0011 >> 1  →  0001
0011 >> 2  →  0000 , in both cased most most-significant bit (here
```

Arithematic vs Logical Right Shift

- If a number is encoded using two's complement then
  - An arithmetic right shift preserves the number's sign.
  - While a logical right shift makes the number positive.
- Example:

```
// Arithmetic shift
1011 >> 1  →  1101
    1011 is -5
    1101 is -3

// Logica shift
1111 >>> 1  →  0111
    1111 is -1
    0111 is  7
```

References

- https://bit-calculator.com/bit-shift-calculator
- https://open4tech.com/logical-vs-arithmetic-shift/