signed and unsigned

- We can use signed and unsigned with only interger datatypes (char, short, int, long).
- We can not use signed and unsigned with float, double, derived and user defined datatypes.

unsigned

- all bits are used to store value
- eg char
  - size of char - 1 byte (8 bits)
  - All 8 bits are used to store value
  - unsigned char var = 5; (0000 0101)

signed

- MSB is used to store sign of the value
  - MSB = 1 --> negative
  - MSB = 0 --> positive
- eg char
  - size of char = 1 bytes (8 bits)
  - 1 bit(MSB) is used to store sign of value
  - remaining 7 bits are used to store actual value
  - signed char var = -5;
- All negative numbers are stored in 2's complement form

Bitwise operators

- used to manipulate bits of variables
- 6 bitwise operators are present
- &, |, ~, ^, <<, >>

Bitwise AND (&)

```
A   B   Y
0   0   0
0   1   0
1   0   0
1   1   1
```

- when all inputs are 1 then output is 1
- when any one of the input is 0 then output is 0

Bitwise OR (|)

```
A   B   Y
0   0   0
0   1   1
1   0   1
1   1   1
```

- when all inputs are 0 then output is 0
- when any one of the input is 1 then output is 1

Bitwise NOT (Complement) (~) (unary)

```
A   Y
0   1
1   0
```

- always get output complement of input

Bitwise XOR (^)

```
A   B   Y
0   0   0
0   1   1
1   0   1
1   1   0
```

- when all inputs are similar then output is 0
- when inputs are different then output is 1

Bitwise Left shift (<<)

- bits of variable are shifted by given number towords left
- and those many 0s are added from right side

Bitwise right shift (>>)

- unsigned
  - bits of variable are shifted by given number towords right
  - and those many 0s are added from left side
- signed
  - bits of variable are shifted by given number towords right
  - and MSB is added into those many bits from left side

Endianness (architecture)

- decides how multi byte data is stored inside memory
- There sre two types of endianness
    - Little endian
    - Big endian

Little endian

- lower byte is stored into lower address
- eg short sh = 0x4142 if(sh is created at address 100)
    - Lower byte 42 is stored at lower address 100
    - Higher byte 41 is stored at higher address 101

Big endian

- higher byte is stored into lower address
- eg short sh = 0x4142 if(sh is created at address 100)
    - Lower byte 42 is stored at higher address 101
    - Higher byte 41 is stored at lower address 100

Assignment Questions

1. check number is divisible by 4.
2. find next divisible by 4 for given number.
3. find previous divisible by 4 for given number.
4. check nth bit of register
5. set nth bit of register
6. clear nth bit of register
7. toggle nth bit of register
8. set bits 12 to 15 of register
9. clear bits 17 to 20 of register
10. read value from bit 19 to 24 of register
11. write value on bit 8 to 15 of register
12. wait while bit 4 of register is 0
13. wait while bit 4 of register is 1