



1.4.1 Bitwise Manipulation & Masks

Bitwise manipulation involves performing operations directly on the individual **bits** of a binary value.

Bitwise operations are commonly used to:

- Shift bits left or right.
- Set particular bits to 1.
- Clear particular bits to 0.
- Toggle bits.
- Extract specific bits from a value.

Bit Shifts

A **bit shift** moves every bit in a binary number to the left or right.

A shift by one to the left **multiplies** the binary number by 2.

$$\begin{array}{ccc} 00100100 & \rightarrow & 01001000 \\ 72 & \rightarrow & 144 \end{array}$$

A shift by one to the right **divides** the binary number by 2.

$$\begin{array}{ccc} 00110100 & \rightarrow & 00011010 \\ 104 & \rightarrow & 52 \end{array}$$

The number of positions shifted determines the multiplication or division.

Shift left 1 $\rightarrow \times 2$

Shift left 2 $\rightarrow \times 4$

Shift left 3 $\rightarrow \times 8$





Bitwise AND

AND compares corresponding bits in two binary values.

The result is 1 **only when both bits are 1**.

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

AND is particularly useful when **clearing or extracting bits**.

Bitwise OR

OR compares corresponding bits in two binary values.

The result is 1 when **at least one of the bits is 1**.

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

OR can be used to **set particular bits to 1**.

Bitwise XOR

XOR (exclusive OR) compares corresponding bits in two binary values.

The result is 1 **only when the two bits are different**.

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

XOR can be used to **toggle bits**.

