

## BITWISE MASKING (Q6)

- Every integer in C is represented by a sequence of 0 and 1
- Bitwise masking: Take an int and a "mask" (another int, whose bits you've set to 1 where you want to keep the original bits, and 0 where you want to clear them - and combining them with an `&` operator)

· Bitwise operators table:

Operator	Symbol	Name	Description	Example	Result
Shift-left	<<	Left shift	Shifts bits left by specified positions, filling with zeros	5 << 2 (101 << 2)	20 (10100)
Shift-right	>>	Right shift	Shifts bits right by specified positions	20 >> 2 (10100 >> 2)	5 (101)
Bitwise-OR		Bitwise OR	Returns 1 if either bit is 1	5   3 (101   011)	7 (111)
Bitwise-AND	&	Bitwise AND	Returns 1 only if both bits are 1	5 & 3 (101 & 011)	1 (001)
Bitwise-XOR	^	Bitwise XOR	Returns 1 if bits are different	5 ^ 3 (101 ^ 011)	6 (110)
Bitwise-complement	~	Bitwise NOT	Flips all bits (1 becomes 0, 0 becomes 1)	~5 (~101)	-6 (depends on system)

## XOR Properties:

$$a \wedge a = 0 \quad (\text{self-inverse})$$

$$a \wedge 0 = a \quad (\text{identity})$$

$$a \wedge b = b \wedge a \quad (\text{commutative})$$

$$a \wedge (b \wedge c) = (a \wedge b) \wedge c \quad (\text{associative})$$

## Shift property:

- Left shift by  $n$  positions ( $\ll n$ ) =  $\times$  by  $2^n$
- Right shift by  $n$  positions ( $\gg n$ ) =  $\div$   $2^n$  (for pos num)

### Shift-left ( $\ll$ )

A: 0110 1010

$A \ll 1 \rightarrow 1101 0100$

### Shift-right ( $\gg$ )

A: 0110 1010

$A \gg 1 \rightarrow 0011 0101$

### Bitwise OR ( $|$ )

A: 0110 1010

B: 0011 1100

$A | B \rightarrow 0111 1110$

### Bitwise AND ( $\&$ )

A: 0110 1010

B: 0011 1100

$A \& B \rightarrow 0010 1000$

### Bitwise XOR ( $\wedge$ )

A: 0110 1010

B: 0011 1100

$A \wedge B \rightarrow 0101 0110$

### Bitwise NOT ( $\sim$ )

A: 0110 1010

$\sim A \rightarrow 1001 0101$