Bitwise Operators in C

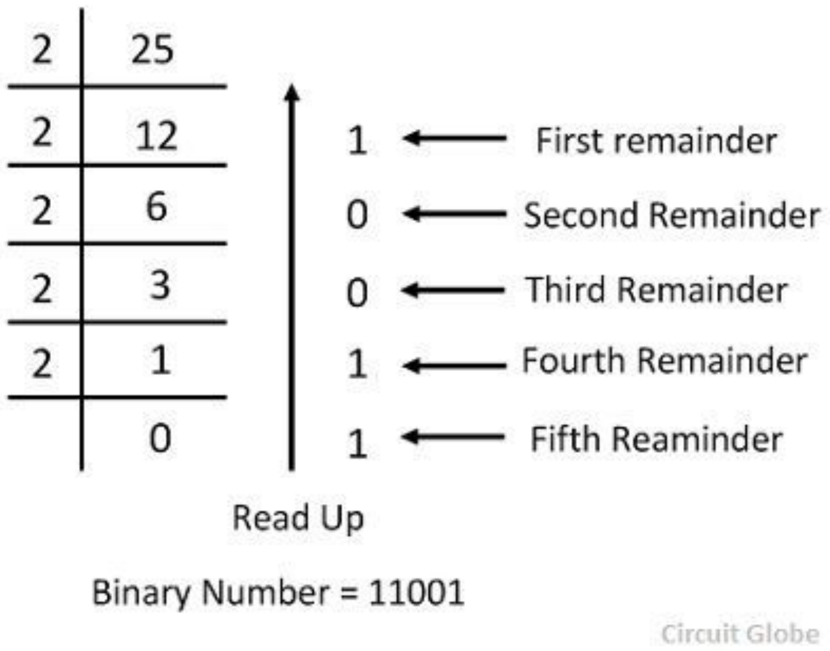
Dr Bhanu

Bitwise operators supported by C

Assume variable 'A' holds 60 and variable 'B' holds 13

Operator	Description	Example
&	Binary AND Operator copies a bit to the result if it exists in both operands.	(A & B) = 12, i.e., 0000 1100
|	Binary OR Operator copies a bit if it exists in either operand.	(A | B) = 61, i.e., 0011 1101
^	Binary XOR Operator copies the bit if it is set in one operand but not both.	(A ^ B) = 49, i.e., 0011 0001
~	Binary One's Complement Operator is unary and has the effect of 'flipping' bits.	(~A) = ~(60), i.e,. 1100 0011
<<	Binary Left Shift Operator. The left operands value is moved left by the number of bits specified by the right operand.	A << 2 = 240 i.e., 1111 0000
>>	Binary Right Shift Operator. The left operands value is moved right by the number of bits specified by the right operand.	A >> 2 = 15 i.e., 0000 1111

Decimal to Binary Conversion



2	25		
2	12	1	First remainder
2	6	0	Second Remainder
2	3	0	Third Remainder
2	1	1	Fourth Remainder
	0	1	Fifth Reaminder

Read Up

Binary Number = 11001

Dec to Bin conversion process

- Successively divide the number by 2
- Collect the reminders from bottom to top
- Continue until the quotient becomes 0

Decimal to Binary to Decimal

Power of 2	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Value	256	128	64	32	16	8	4	2	1
Binary Number				0	0	1	1	0	1
Binary Number			0	1	1	1	1	0	0

001101 = 8 + 4 + 1 = 13
0111100 = 32 + 16 + 8 + 4 = 60

Decimal to Binary

Power of 2	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Value	256	128	64	32	16	8	4	2	1
Binary Number									
Binary Number									

126? =

75? =

Binary to Decimal

Power of 2	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Value	256	128	64	32	16	8	4	2	1
Binary Number	0	0	0	0	0	0	0	0	0
Binary Number	0	0	0	0	0	0	0	0	0

0111001 =
1000101 =

Shifting operations - decimal

			6	5

Shift the number left by one digit → Multiply the number by 10^1 = 10

		6	5	0

← Shift left

Shift the original number left by two digits → Multiply the number by 10^2 = 100

	6	5	0	0

← Shift left

Note : Base of decimal system is 10

Left Shift Operation

Power of 2	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
Value	256	128	64	32	16	8	4	2	1	
Binary Number				0	0	1	1	0	1	
Shift left(1)			0	0	1	1	0	1	0	← Shift left
Shift left(2)		0	0	1	1	0	1	0	0	← Shift left

Shift left by one bit → Multiply by $2^{1.}$ → 13 became 26
Shift left by two bits → Multiply by 2^{2} → 13 became 52

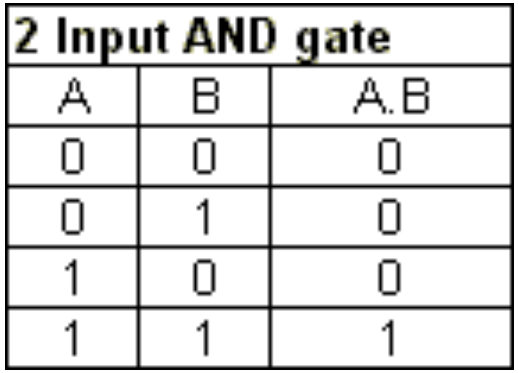
Note : Base of binary system is 2

Bitwise operators supported by C

Assume variable 'A' holds 60 and variable 'B' holds 13

Operator	Description	Example
&	Binary AND Operator copies a bit to the result if it exists in both operands.	(A & B) = 12, i.e., 0000 1100
|	Binary OR Operator copies a bit if it exists in either operand.	(A | B) = 61, i.e., 0011 1101
^	Binary XOR Operator copies the bit if it is set in one operand but not both.	(A ^ B) = 49, i.e., 0011 0001
~	Binary One's Complement Operator is unary and has the effect of 'flipping' bits.	(~A) = ~(60), i.e,. 1100 0011
<<	Binary Left Shift Operator. The left operands value is moved left by the number of bits specified by the right operand.	A << 2 = 240 i.e., 1111 0000
>>	Binary Right Shift Operator. The left operands value is moved right by the number of bits specified by the right operand.	A >> 2 = 15 i.e., 0000 1111

Bit-wise AND operation



2 Input AND gate		
A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Value	256	128	64	32	16	8	4	2	1
A		0	0	1	1	1	1	0	0
B		0	0	0	0	1	1	0	1
A & B		0	0	0	0	1	1	0	0

A = 60

B = 13

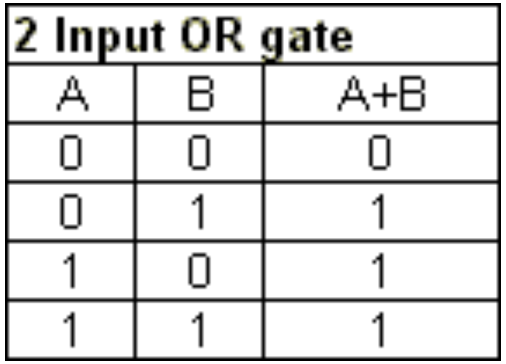
A&B = 12

Bitwise operators supported by C

Assume variable 'A' holds 60 and variable 'B' holds 13

Operator	Description	Example
&	Binary AND Operator copies a bit to the result if it exists in both operands.	(A & B) = 12, i.e., 0000 1100
|	Binary OR Operator copies a bit if it exists in either operand.	(A | B) = 61, i.e., 0011 1101
^	Binary XOR Operator copies the bit if it is set in one operand but not both.	(A ^ B) = 49, i.e., 0011 0001
~	Binary One's Complement Operator is unary and has the effect of 'flipping' bits.	(~A) = ~(60), i.e,. 1100 0011
<<	Binary Left Shift Operator. The left operands value is moved left by the number of bits specified by the right operand.	A << 2 = 240 i.e., 1111 0000
>>	Binary Right Shift Operator. The left operands value is moved right by the number of bits specified by the right operand.	A >> 2 = 15 i.e., 0000 1111

Bit-wise OR operation



2 Input OR gate		
A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Value	256	128	64	32	16	8	4	2	1
A		0	0	1	1	1	1	0	0
B		0	0	0	0	1	1	0	1
A | B		0	0	1	1	1	1	0	1

A = 60

B = 13

A|B = 61

Bitwise operators supported by C

Assume variable 'A' holds 60 and variable 'B' holds 13

Operator	Description	Example
&	Binary AND Operator copies a bit to the result if it exists in both operands.	(A & B) = 12, i.e., 0000 1100
|	Binary OR Operator copies a bit if it exists in either operand.	(A | B) = 61, i.e., 0011 1101
^	Binary XOR Operator copies the bit if it is set in one operand but not both.	(A ^ B) = 49, i.e., 0011 0001
~	Binary One's Complement Operator is unary and has the effect of 'flipping' bits.	(~A) = ~(60), i.e,. 1100 0011
<<	Binary Left Shift Operator. The left operands value is moved left by the number of bits specified by the right operand.	A << 2 = 240 i.e., 1111 0000
>>	Binary Right Shift Operator. The left operands value is moved right by the number of bits specified by the right operand.	A >> 2 = 15 i.e., 0000 1111

Bit-wise XOR operation

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

	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Value	256	128	64	32	16	8	4	2	1
A		0	0	1	1	1	1	0	0
B		0	0	0	0	1	1	0	1
A ^ B		0	0	1	1	0	0	0	1

A = 60

B = 13

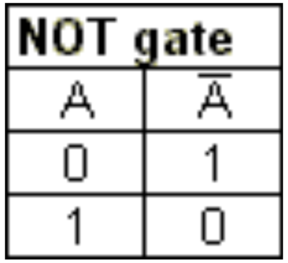
A^B = 49

Bitwise operators supported by C

Assume variable 'A' holds 60 and variable 'B' holds 13

Operator	Description	Example
&	Binary AND Operator copies a bit to the result if it exists in both operands.	(A & B) = 12, i.e., 0000 1100
|	Binary OR Operator copies a bit if it exists in either operand.	(A | B) = 61, i.e., 0011 1101
^	Binary XOR Operator copies the bit if it is set in one operand but not both.	(A ^ B) = 49, i.e., 0011 0001
~	Binary One's Complement Operator is unary and has the effect of 'flipping' bits.	(~A) = ~(60), i.e,. 1100 0011
<<	Binary Left Shift Operator. The left operands value is moved left by the number of bits specified by the right operand.	A << 2 = 240 i.e., 1111 0000
>>	Binary Right Shift Operator. The left operands value is moved right by the number of bits specified by the right operand.	A >> 2 = 15 i.e., 0000 1111

Bit-wise NOT operation



NOT gate	
A	$\overline{A}$
0	1
1	0

	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Value	256	128	64	32	16	8	4	2	1
A		0	0	1	1	1	1	0	0
~A		1	1	0	0	0	0	1	1

A = 60

~A= 195

Bitwise operators supported by C

Assume variable 'A' holds 60 and variable 'B' holds 13

Operator	Description	Example
&	Binary AND Operator copies a bit to the result if it exists in both operands.	(A & B) = 12, i.e., 0000 1100
|	Binary OR Operator copies a bit if it exists in either operand.	(A | B) = 61, i.e., 0011 1101
^	Binary XOR Operator copies the bit if it is set in one operand but not both.	(A ^ B) = 49, i.e., 0011 0001
~	Binary One's Complement Operator is unary and has the effect of 'flipping' bits.	(~A) = ~(60), i.e,. 1100 0011
<<	Binary Left Shift Operator. The left operands value is moved left by the number of bits specified by the right operand.	A << 2 = 240 i.e., 1111 0000
>>	Binary Right Shift Operator. The left operands value is moved right by the number of bits specified by the right operand.	A >> 2 = 15 i.e., 0000 1111

Left shift operation

	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Value	256	128	64	32	16	8	4	2	1
A		0	0	1	1	1	1	0	0
A << 2	0	1	1	1	1	0	0	0	0

A = 60

A<<2 = 240

Bitwise operators supported by C

Assume variable 'A' holds 60 and variable 'B' holds 13

Operator	Description	Example
&	Binary AND Operator copies a bit to the result if it exists in both operands.	(A & B) = 12, i.e., 0000 1100
|	Binary OR Operator copies a bit if it exists in either operand.	(A | B) = 61, i.e., 0011 1101
^	Binary XOR Operator copies the bit if it is set in one operand but not both.	(A ^ B) = 49, i.e., 0011 0001
~	Binary One's Complement Operator is unary and has the effect of 'flipping' bits.	(~A) = ~(60), i.e,. 1100 0011
<<	Binary Left Shift Operator. The left operands value is moved left by the number of bits specified by the right operand.	A << 2 = 240 i.e., 1111 0000
>>	Binary Right Shift Operator. The left operands value is moved right by the number of bits specified by the right operand.	A >> 2 = 15 i.e., 0000 1111

Right shift operation

	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Value	256	128	64	32	16	8	4	2	1
A		0	0	1	1	1	1	0	0
A >> 2				0	0	1	1	1	1

A = 60

A >> 2 = 15