

Bitwise Operation

errantProgrammer

October 4, 2024

Contents

1	Bitwise Operation	3
1.1	AND &	3
1.1.1	Codigo Example	3
1.2	OR 	3
1.2.1	Codigo Example	3
1.3	XOR ^	3
1.3.1	Codigo Example	4
1.4	NOT ~	4
1.4.1	Codigo Example	4
1.5	Left Shift <<	4
1.5.1	Codigo Example	4
1.6	Right Shift >>	5
1.6.1	Codigo Example	5

1 Bitwise Operation

1.1 AND &

P	Q	P & Q
0	0	0
0	1	0
1	0	0
1	1	1

1.1.1 Codigo Example

```
6  int a = 7, b = 4;
7  int p = a & b;
8  cout <<"a & b = " << p << endl; // p = 4
```

$$\begin{array}{r} 111_2 \\ \& 100_2 \\ \hline 100_2 = 4 \end{array}$$

1.2 OR |

P	Q	P Q
0	0	0
0	1	1
1	0	1
1	1	1

1.2.1 Codigo Example

```
10  int c = 7, d = 4;
11  int q = c | d;
12  cout << "c | d = " << q << endl; // q = 7
13  // XOR
```

$$\begin{array}{r} 111_2 \\ | 100_2 \\ \hline 111_2 = 7 \end{array}$$

1.3 XOR ^

P	Q	P ^ Q
0	0	0
0	1	1
1	0	1
1	1	0

1.3.1 Codigo Example

```
6  int a = 7, b = 4;
7  int p = a & b;
8  cout <<"a & b = " << p << endl; // p = 4
```

$$\begin{array}{r} 111_2 \\ \wedge 100_2 \\ \hline 011_2 = 3 \end{array}$$

1.4 NOT ~

El not lo que hace nos brinda el complemento a 2, del número, no confundir con el negador lógico "!".

P	~P
0	1
1	0

1.4.1 Codigo Example

```
6  int a = 7, b = 4;
7  int p = a & b;
8  cout <<"a & b = " << p << endl; // p = 4
```

$$\begin{array}{r} \sim 1001_2 \\ \hline 0110_2 = 6 \end{array}$$

1.5 Left Shift <<

P	Q	P << Q
0	0	0
0	1	1
1	0	1
1	1	0

1.5.1 Codigo Example

```
6  int a = 7, b = 4;
7  int p = a & b;
8  cout <<"a & b = " << p << endl; // p = 4
```

$$\begin{array}{r} 111_2 \\ \ll 2 \\ \hline 011_2 = 3 \end{array}$$

<u>P</u>	<u>Q</u>	<u>P</u> \gg <u>Q</u>
0	0	0
0	1	1
1	0	1
1	1	0

1.6 Right Shift \gg

1.6.1 Codigo Example

```

6      int a = 7, b = 4;
7      int p = a & b;
8      cout <<"a & b = " << p << endl; // p = 4

```

$$\begin{array}{r}
111_2 \\
\gg 2 \\
\hline
011_2 = 3
\end{array}$$

2 Complemento a 2

El complemento a 2 de un n mer es: