Bitwise Operations – Chapter 2, For CS 2110

Shreyas Casturi

Contents

1 Chapter 2: Bitwise Operations

Now that we have an understanding of how to work with basic binary numbers, we can now learn about *logical operators* that work on bits/binary numbers, such as AND, OR, NOT. You may have seen these operators before.

We will deal with the concept of *floating-point numbers*, which carry more precision, but are conceptually harder to represent.

We will also learn to work with other numerical systems, such as octal and hexadecimal, with examples of conversion between these systems.

1.1 Logical Operators

Logical operators are done on bits, and are hence called *bitwise* operators.

We will denote a sub-sub-section to each major logical operator, usually showing two truth tables. Exercises will give you actual experience.

1.1.1 AND

A bitwise AND operation works like so, for given bits 0 and 1:

AND	0	1
0	0	0
1	0	1

A more general example is

A	B	(A AND B, AB, A & B)
0	0	0
0	1	0
1	0	0
1	1	1

1.1.2 OR

An OR operation works as so

OR	0	1
0	0	1
1	1	1

A general example is

A	B	(A OR B, A + B, $A \mid B$)
0	0	0
0	1	1
1	0	1
1	1	1

There is a difference between this: ||, and |. The latter is the bitwise OR operator, which does operations on bits, but the former is a conditional OR, used in evaluating statements, as seen in Java, C, and other languages.

1.1.3 NOT

A NOT operation doesn't require two bits, but rather one. A NOT operation negates/flips the present value.

So, we have

NOT	Result
0	1
1	0

A NOT can be represented as a " ".

1.1.4 XOR

An XOR operation is more interesting, and does require two bits/binary numbers. If the bits are the same, then we return 0, but if the bits aren't the same, we return 1.

We obtain

XOR	0	1
0	0	1
1	1	0

A general operation can be seen as

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

1.1.5 NAND

A NAND can be represented as a negation of the result of the AND operation. So, the result of A NAND B is equivalent to !(A & B).

NAND	0	1
0	1	1
1	1	0

Generally, this is written as

A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0

1.1.6 NOR

1.2 Bit Vectors

1.3 Hexadecmial

1.3.1 Converting from Binary to Hexadecimal, Vice Versa

1.4 Octal

1.4.1 Converting from Binary to Octal, Vice Versa

1.5 Floating Point Numbers