Chapter 5: Bit-String Flicking

WMC CS Club

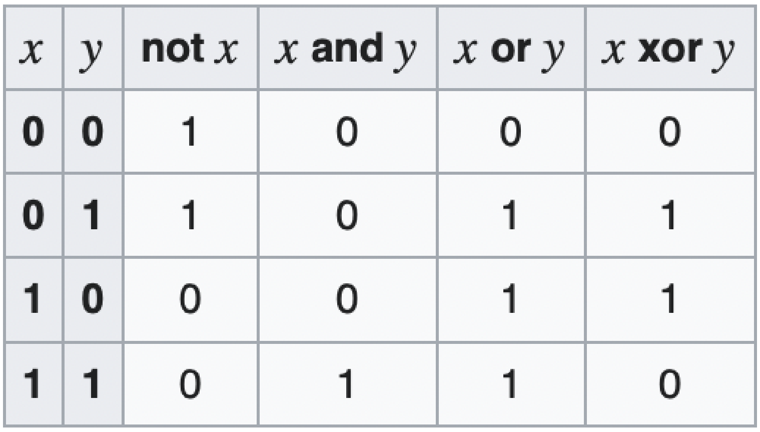
Jan 6, 2022

Bit manipulation is all about binary operations. Recall that there are only two states in the binary world: 0 or 1, on or off, rain or not rain, . . .

§1 Bitwise Operators

The logical operators are NOT ($\sim$ or $\neg$), AND (&), OR ($|$), and XOR ($\oplus$).

- NOT is a unary operator that performs logical negation on each bit. Bits that are 0 become 1, and those that are 1 become 0. For example: $\sim$**101110** has a value of **010001**.
- AND is a binary operator that performs the logical AND of each bit in each of its operands. The AND of two values is 1 only if both values are 1. For example, **1011011 and 011001** has a value of **001001**. The AND function is often used to isolate the value of a bit in a bit-string or to clear the value of a bit in a bit-string.
- OR is a binary operator that performs the logical OR of each bit in each of its operands. The OR of two values is 1 only if one or both values are 1. For example, **1011011 or 0011001** has a value of **1011011**. The OR function is often use to force the value of a bit in a bit-string to be 1, if it isn't already.
- XOR is a binary operator that performs the logical XOR of each bit in each of its operands. The XOR of two values is 1 if the values are different and 0 if they are the same. For example, **1011011 xor 011001** = **110010**. The XOR function is often used to change the value of a particular bit.



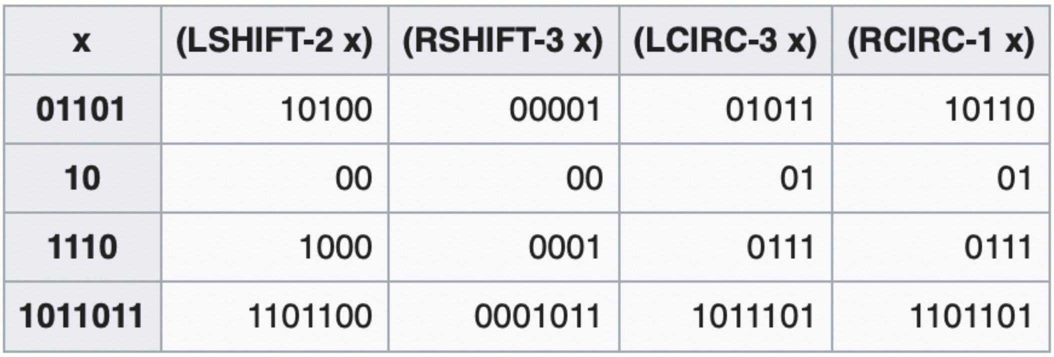
x	y	not x	x and y	x or y	x xor y
0	**0**	1	0	0	0
0	**1**	1	0	1	1
1	**0**	0	0	1	1
1	**1**	0	1	1	0

§2 Shift Operators

Logical shifts (LSHIFT-x and RSHIFT-x) “ripple” the bit-string x positions in the indicated direction, either to the left or to the right. Bits shifted out are lost; zeros are shifted in at the other end.

Circulates (RCIRC-x and LCIRC-x) “ripple” the bit string x positions in the indicated direction. As each bit is shifted out one end, it is shifted in at the other end. The effect of this is that the bits remain in the same order on the other side of the string.

The size of a bit-string does not change with shifts, or circulates. If any bit strings are initially of different lengths, all shorter ones are padded with zeros in the left bits until all strings are of the same length.



x	(LSHIFT-2 x)	(RSHIFT-3 x)	(LCIRC-3 x)	(RCIRC-1 x)
01101	10100	00001	01011	10110
10	00	00	01	01
1110	1000	0001	0111	0111
1011011	1101100	0001011	1011101	1101101

> **Remark 2.1.** The shortcuts:
> LSHIFT-n: Erase the first n characters, add padding to the back.
> RSHIFT-n: Erase the last n characters, add padding to the front.
> LCIRC-n: Move n characters from the front to the back.
> RCIRC-n: Move n characters from the back to the front.

> **Remark 2.2.** The order of precedence (from highest to lowest) is: NOT; SHIFT and CIRC; AND; XOR; and finally, OR. In other words, all unary operators are performed on a single operator first. Operators with equal precedence are evaluated left to right; all unary operators bind from right to left.

§3 Examples and Practice Problems

Example 3.1

Evaluate: (NOT(000101 AND 101001 OR NOT 000101)).

Example 3.2

Evaluate: (101110 AND NOT 110110 OR (LSHIFT-3 101010)).

Example 3.3

Solve for X (5-bit string): (LCIRC-2 X) OR (RSHIFT-2 01010) = (NOT 00000) AND X.

Practice Problems

1. Evaluate: (010111 XOR NOT 101010).
2. Evaluate: (LCIRC-2 01101) OR (RSHIFT-1 11111)
3. Evaluate the expression: (RSHIFT-1 (LCIRC-4 (RCIRC-2 01101))).
4. Evaluate the following: (RCIRC-2 (LSHIFT-1 (LCIRC-1 (RSHIFT-2 (NOT 10100))))).
5. Evaluate: (R-CIRC-2003((LCIRC-2002(LCIRC-2001(NOT NOT (NOT NOT 10011)))))).
6. List all possible values of x (5 bits long) that solve the equation: (LSHIFT-1 (10110 XOR (RCIRC-3 x) AND 11011)) = 01100.

Answer Key

1. 000010
2. 11111
3. 01010
4. 00010
5. 10011
6. 00000, 00001, 00100 and 00101.