Logic Tricks

Nail Tosun
Electric and Electronic Engineering Departmant, METU

2018
March

Binary Numbers

r's complement

General formula is following;

$$r^n - N$$

r is radix n is integer side length and N is the number that you complement.

Shortcut

Starting from the right to left;
Leave all zeros no change;
After first zero r-$digit_{afterzero}$
Then (r-1) - all digits.

r-1 complement

General formula

$$r^n - r^{-m} - N$$

Shortcut

Subtract all the digits with r-1

2's complement

Shortcut

Switch all the bits and add 1 or

Keep the same with all zero (right to left). After first non-zero term (don't change first non-zero term), toggle each bit.

1's complement

Shortcut

Just switch all the bits.

Signed magnitude convention

Signed Magnitude

First leftmost bit is sign bit other registers indicate magnitude.

Signed 1's complement

First leftmost bit is sign bit. If first leftmost bit is zero (indicates positive)
Other bits represents +number
If first leftmost bit is one (indicates negative)
Put a minus sign
Takes 1's complement.

Signed 2's complement

First leftmost bit is sign bit. If first leftmost bit is zero (indicates positive)
Other bits represents +number
If first leftmost bit is one (indicates negative)
Put a minus sign
Takes 2's complement. There is one representation for zero.

Boundaries

3 bit Signed 1's complement and magnitude methods

$$-3 \leq X \leq 3$$

2's complement method has extra one number(due to only has one zero.)

$$-4 \leq X \leq 3$$

Arithmetic addition in 2's complement

Take both number 2's complement.

Sum it up.

Fuck the carry bit there is no information here. Right of the carry bit is magnitude bit. Put your sign if it is 1 (indicates negative) take 2's complement to find its absolute value.

Overflow

Sign bits of the both operand are same and result sign is different. It indicates overflow.

Overflow can happen if both operands sign are same.

Extension of the bits

Copy the sign bit and paste left bits. 001 's extension the 6 bit is 000001; 100 's extension to 6 bit is 111100;

Excess 3

BCD+3

Self complementing

Parities

Finding the error in data transfer

Odd parity

Add one bit to ensure sum of all bits odd

Even parity

Add one bit to ensure sum of all bits even

Substraction with complentary

Substraction with r's complement

Operation of $(M)_r - (N)_r$ where M and N are unsigned integer.

Write N's r's complement and add with M;

$$M + r^n - N$$

if $M > N$ sum is produces carry bit. Ignore the carry bit. Sum is your result.

if $N > M$ sum is **don't produce carry bit** so take r's complement of the result.

Don't forget to put minus sign.

Substraction with r-1 complement

Take the ckarn operand r-1 complement.

Sum it up

If there is a **carry out** this indicates number is **positive**. add carry out to least significant bit.

If there is **no carry out**. It means the number is negative. Take ones complement