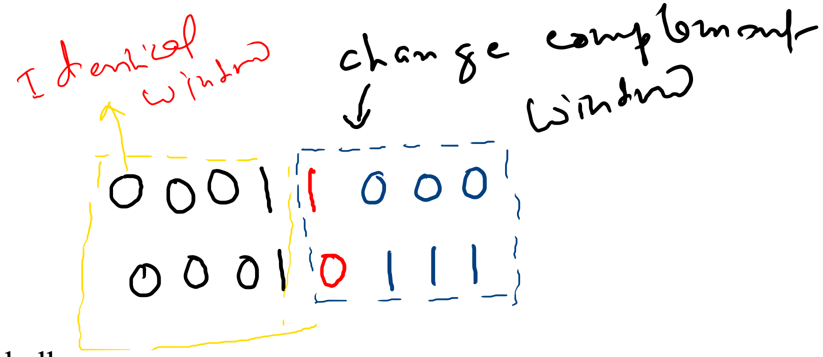
BIT LEVEL DIFFERENCE IN x and (x-1)

Example: (8 bit system)

2^4 2^3 2^2 2^1 2^0

Even: $x = +24$ 0001 1000

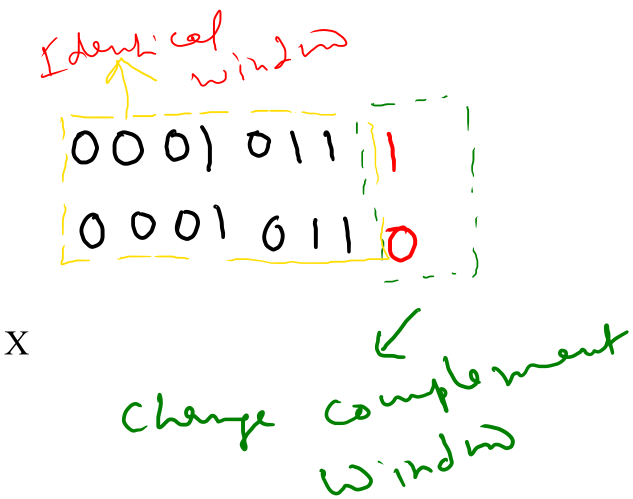
$x-1 = 23$ 0001 0111



Bit representation of (X-1) for even X : The rightmost 1 bit of X is set to ZERO and all the trailing zero bits are set to 1.

odd: $x = +23$ 0001 0111

$x-1 = 22$ 0001 0110



Bit representation of (X-1) for odd X : The rightmost 1 bit of X is set to ZERO. Since X is odd number so rightmost 1 bit is the first bit.

Q1: How to preserve identical window?
If we apply bitwise & OR bitwise | operation between X and X-1.

Q2: How to preserve the change window ?
If we apply 1's complement to X then change window of X-1 will be preserved and vice versa.
e.g ~X&(X-1) : change window of X-1 will be preserved
X&(~(X-1)) : change window of X will be preserved

Q3: How to get all 1s or all 0s in change window ?
-- For all 1's we need to apply
-- bitwise inclusive | operation between X and X-1. Identical window is intact.
---bitwise exclusive ^ operation between X and X-1. Identical window becomes zero.

-- For all 0's we need to apply bitwise & operation between X and X-1

Bit level difference in X AND -X i.e. 2's complement of X

$x = 24$, $\rightarrow$ 0 0 0 1 | 1 0 0 0

$-x = -24$ $\rightarrow$ 1 1 1 0 | 1 0 0 0

Identical window

2's complement of $x = 24$

complement change window

$-x = 1\text{'s complement} + 1$

i.e. $-x = (\sim x + 1)$

applying De Morgan's law using complement

$\sim -x = \sim(\sim x + 1)$

$\Rightarrow$ $\boxed{\sim -x = (x - 1)}$ $\rightarrow$ so, $(x-1)$ can be replaced with complement of $(-x)$

complement of $(-x)$

i.e. 2's complement of x

Manipulating Rightmost Bits

A. X &(X-1) : Use this formula to turn off the rightmost 1-bit of X.

--Even X contains the trailing ZEROs that is off itself.

--Odd X doesnot contain any trailing zero bit.

$x \& (x-1)$ can be written as

$x \& (\sim -x)$

Application : To find whether the given X is power of 2. Or the given X is 0.

eg: $x = 16\ (2^4)$ 0 0 0 1 0 0 0 0

& $x-1 = 15$ 0 0 0 0 1 1 1 1

0 0 0 0 0 0 0 0 0

IF X &(X-1)==0 : THEN X is power of 2

B. X &(X+1) : Similarly, Use this formula :

--to turn off all the trailing 1 bits of X, if none then produces X itself. (e.g. X= 10100111 result = 10100000)

--and to turn off the rightmost 1-bit of (X+1)

--Even (X+1) contains the trailing ZEROs i.e. off in itself.

--Odd (X+1) doesnot contain any trailing zero bit.

--Similarly, this formula can be used to test

1. **if an unsigned integer X is of the form (2^n - 1)**
2. **if X is 0**
3. **if X contains all bits as 1**

Manipulating Rightmost Bits

1. x & (x – 1) and its opposite x | (x + 1)

x & (x – 1): use to turn off the rightmost 1-bit in a word, producing 0 if none (e.g., input: 01011000 output: 01010000).
This can be used to determine if an unsigned integer is a power of 2 or is 0: apply the formula followed by a 0-test on the result.

x | (x + 1) : use to turn on the rightmost 0-bit in a word, producing all 1's if none (e.g., input: 10100111 output:10101111).

2. x & (x + 1) and its opposite x | (x– 1)

x & (x + 1) : use to turn off the trailing 1's in a word, producing x if none (e.g., input: 10100111 output:10100000)
This can be used to determine if an unsigned integer is of the form 2n– 1, 0, or all 1's: apply the formula followed by a 0-test on the result.

x | (x– 1) : use to turn on the trailing 0's in a word, producing x if none (e.g., input:10101000 output: 10101111)

Manipulating Rightmost Bits

3. ~x & (x + 1) and its opposite ~x | (x – 1)

~x & (x + 1) : use to create a word with a single 1-bit at the position of the rightmost 0-bit in x, producing 0 if none (e.g. input:10100111   output:00001000)

~x | (x – 1) : use to create a word with a single 0-bit at the position of the rightmost 1-bit in x, producing all 1's if none (e.g. input:10101000 output: 11110111).

4. ~x | (x + 1) : use to create a word with 0's at the positions of the trailing 1's in x, and 0's elsewhere, producing all 1's if none (e.g.input:10100111 output: 11111000)

5. x & (−x) :use to isolate the rightmost 1-bit, producing 0 if none (e.g. inupt:01011000 output:00001000)

6. x ^ (x - 1) : use to create a word with 1's at the positions of the rightmost 1-bit and the trailing 0's in x, producing all 1's if no 1-bit, and the integer 1 if no trailing 0's (e.g. input: 01011000   output: 00001111)

7. x ^ (x + 1) :use to create a word with 1's at the positions of the rightmost 0-bit and the trailing 1's in x, producing all 1's if no 0-bit, and the integer 1 if no trailing 1's (e.g. input: 01010111 output: 00001111)

De Morgan's Laws Extended

The logical identities known as De Morgan's laws can be thought of as distributing, or "multiplying in," the ***not*** sign. This idea can be extended to apply to the expressions of this section, and a few more, as shown here. (The first two are De Morgan's laws.)

$$\neg(x \,\&\, y) = \neg x \mid \neg y$$

$$\neg(x \mid y) = \neg x \,\&\, \neg y$$

$$\neg(x + 1) = \neg x - 1$$

$$\neg(x - 1) = \neg x + 1$$

$$\neg{-x} = x - 1$$

$$\neg(x \oplus y) = \neg x \oplus y = x \equiv y$$

$$\neg(x \equiv y) = \neg x \equiv y = x \oplus y$$

$$\neg(x + y) = \neg x - y$$

$$\neg(x - y) = \neg x + y$$

Formulae use to get trailing ZEROs with the help of leading ZEROs

Use one of the following formulas to create a word with 1's at the positions of the trailing 0's in x, and 0's elsewhere, producing 0 if none (e.g. input: 01011000 output: 00000111):

Solution 1: using X and X-1

x = 0101 1000

x-1 = 0101 0111

complement change window

Identical required

aim is to preserve 1's at the position of trailing zero's

Strategy: Just apply complement to x and then use bitwise & operator between ~x and x-1

Sol^n 1: $\boxed{\sim x \, \& \, (x-1)}$ → eq 1

Formulae use to get trailing ZEROs with the help of leading ZEROs

Use one of the following formulas to create a word with 1's at the positions of the trailing 0's in x, and 0's elsewhere, producing 0 if none (e.g. input: 01011000 output: 00000111):

Solution 2: using X and 2's complement of X i.e. -X

$x = 0101\,1000$

$-x = 1010\,1000$ → Identical

Change window

strategy 1: apply bitwise & to preserve the identical window. now substract 1 to get the desired output

$$\boxed{(x \,\&\, -x) - 1} \rightarrow eq2$$

strategy 2: apply bitwise '|' to get all 1's in complement change window, now take the complement to get the desired output

$$\boxed{\sim(x \mid -x)} \rightarrow eq3$$

Figure out formulae to determine if a nonnegative integer is of the form 2^j − 2^k for some j ≥ k≥ 0

eg. $2^j = 2^7$; $2^k = 2^3$

$$2^j - 2^k = 2^7 - 2^3$$

$$
\begin{array}{rcl}
 & 2^5 = & 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \\
- & 2^3 = & 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0 \\
\hline
 & 2^j - 2^k = & 0\ 1\ 1\ 1\ 0\ 0\ 0\ 0
\end{array}
$$

OBSERVATION : If any number is in the form of (2^j - 2^k) then the set bits will be contiguous and all other bits will be zero.

Question : What to invent to figureout if a given positive integer is in the form of (2^j - 2^k) ?

Solution : Need to invent formulae to turn off the rightmost contiguous string of 1's (e.g., 01011100 ==> 01000000). Then apply the formula followed by a 0-test on the result.

Figure out formulae to determine if a nonnegative integer is of the form 2^j − 2^k for some j ≥ k≥ 0

Agenda: To figureout formulae to turn off the rightmost contiguous string of 1's (e.g., 01011100 ==> 01000000)

Some assumed y

$x = 0\,1\,0\,1\,1\,1\,0\,0$

$\;\;\;\;\;0\,0\,0\,0\,0\,1\,0\,0$ → add 1 at the position of rightmost 1 of x.

Intermediate result $= 0\,1\,1\,0\,0\,0\,0\,0$

We need to make this 1 to zero to get the desired output

If we apply bitwise '&' between x and Intermediate result we can get desired output.

Formula $= \boxed{(x + y) \,\&\, x}$

y must be a number which has 1 at the rightmost set bit of x and other bits are zero.

Figure out formulae to determine if a nonnegative integer is of the form 2^j − 2^k for some j ≥ k≥ 0

Agenda: To figureout formulae to turn off the rightmost contiguous string of 1's (e.g., 01011100 ==> 01000000)

Formula = $(x+y) \& x$

We need a Y of same size of X which has single set bit at the postion of rightmost 1 bit of X

Example : If X =01011100 then required Y=00000100

Q How to find y?

Strategy 1: using x and $(x-1)$

Strategy 2: using x and $(-x)$

We need a Y of same size of X which has single set bit at the postion of rightmost 1 bit of X
Example : If X =01011100 then required Y=00000100

Strategy 1: Using X and X-1

$x = 01011100$

$x-1 = 01011011$

change complement window

Identical window

Two possibilites

A: If we apply complement to x and then apply bitwise '&'

$\sim x = 10100011$

&

$x-1 = 01011011$

00000011

If we add 1 to this we can set desired output. 00000100

$$\boxed{y = (\sim x \,\&\, (x-1)) + 1}$$

B: $\boxed{y = x \,\&\, \sim(x-1)}$

$(x-1)$ is $\sim(-x)$

$\sim(x-1) = \sim(\sim -x)$

$= -x$

$$\boxed{y = x \,\&\, -x}$$

We need a Y of same size of X which has single set bit at the postion of rightmost 1 bit of X

Example : If X =01011100 then required Y=00000100

Strategy 2: Using X and -X

$x = 01011100$

$-x = 10100100$

Identical window

complement

$$\boxed{y = x \,\&\, (-x)}$$

Figure out formulae to determine if a nonnegative integer is of the form 2^j − 2^k for some j ≥ k≥ 0

Agenda: To figureout formulae to turn off the rightmost contiguous string of 1's (e.g., 01011100 ==> 01000000)

Formula = $(x + y) \& x$

We can replace y with

(I) $y = (\sim x \& (x-1)) + 1$

(II) $y = x \& (-x)$

replace y in formula:

$$\boxed{\text{Formula 1}: (x + ((\sim x \& (x-1)) + 1)) \& x}$$

$$\boxed{\text{Formula 2}: (x + (x \& -x)) \& x}$$

Figure out formulae to determine if a nonnegative integer is of the form 2^j − 2^k for some j ≥ k≥ 0

Agenda: To figureout formulae to turn off the rightmost contiguous string of 1's (e.g., 01011100 ==> 01000000)

Identical

$x = 01011100$ → change complement window

$x-1 = 01011011$

Formula using bitwise |

Formula 3: $((x \mid (x-1))+1) \& x$

FORMULAE VARIANTS

FORMULA 1: (X + (~X&(X-1)+1)) & X

FORMULA2: (X +(X&-X)) & X

FORMULA3:((X|(X-1))+1)&X

Figure out formulae to determine if a nonnegative integer is of the form $2^j - 2^k$ for some $j \geq k \geq 0$

Agenda: To figureout formulae to turn off the rightmost contiguous string of 1's (e.g., 01011100 ==> 01000000)

FORMULAE VARIANTS

FORMULA 1: (X + (~X&(X-1)+1)) & X

FORMULA2: (X +(X&-X)) & X

FORMULA3:((X|(X-1))+1)&X