Clarifications

NUMBER BASE

1010_2 56_{16} 0.5_{16}

15_{10} $\underline{10_{10}}$ $\underline{10_2}$

$$B2U_w(\vec{x}) = \sum_{i=0}^{w-1} \underline{x_i} \cdot \underline{2}^i$$

$$B2U_{\underline{4}}([1001]) = 1\times 2^3 + 0\times 2^2 + 0\times 2^1 + 1\times 2^0$$

3 2 1 0

$w-1$

$$= 8 + 1 = 9$$

$\underline{\text{II}}$ $\underline{\text{2's complement encoding}}$ (for signed numbers)

$$B2T_{\underline{w}}(\vec{x}) = -x_{w-1}2^{w-1} + \sum_{i=0}^{w-2} x_i 2^i$$

$$B2T_4([1001]) = -1\times 2^3 + 0\times 2^2 + 0\times 2^1 + 1\times 2^0$$

$\underline{3}|\underline{2\,1\,0}$

$$= -8 + 1 = -7$$

$\underline{1}\ 1\ 0\ 1\ 1$

$W = 2$

$0\ 0 \rightarrow 0$

$0\ 1 \rightarrow 1$

$1\ 0 \rightarrow -2 \quad (-1 \times 2^1 + 0 \times 2^0)$

$1\ 1 \rightarrow -1 \quad (-1 \times 2^1 + 1 \times 2^0)$

$W = 3$

$0\ 0\ 0 \rightarrow 0$

$0\ 0\ 1 \rightarrow 1$

$0\ 1\ 0 \rightarrow 2$

$0\ 1\ 1 \rightarrow 3$

$1\ 0\ 0 \rightarrow -4$

$1\ 0\ 1 \rightarrow -3$

$1\ 1\ 0 \rightarrow -2$

$1\ 1\ 1 \rightarrow -1$

For a bit vector of length w, the values range from

-2^{w-1} to $2^{w-1} - 1$

$W = 3$

-2^2 to $2^2 - 1$

-4 to 3.

$W = 5$

$1\,1\,1\,1\,1 \rightarrow -1$

MACROS → defined in limits.h (32-bit systems)

UINT_MAX → 4,294,967,295.

→ 0x FF FF FF FF

INT_MAX → 2,147,483,647.

→ 0x 7F FF FF FF.

→ 0 31 1's

INT_MIN → -2,147,483,648.

→ 0x 80 00 00 00

→ 1 31 0's.

1000 0 0 0 0 0 0 ... 00

Conversion from signed → unsigned

unsigned int u = 4,294,967,295u;
(or)
= 0xFFFFFFFF;

int s = u;

printf ("u = %u, s = %d", u, s);

u = 4,294 ... s = -1

If signed & unsigned appear in an operation, signed → unsigned

$-1 < 0u$ 4 billion < 0u

⇒ True   No. False

Sign Extension (for signed numbers).

→ Operation of increasing the number of bits of a binary number.

→ while preserving the sign   00000001

$0110_2 \rightarrow 6$

$0000\ 0110_2 \rightarrow 6.$

$1001 \rightarrow -7 \quad (-8+0+0+1)$

$1111\ 1001 \rightarrow -1\times 2^7 + (1\times 2^6 + 1\times 2^5 + 1\times 2^4 + 1\times 2^3 + 0 + 0 + 1\times 2^0)$

$\rightarrow -7$

Scanned by CamScanner

Truncate

0000

int x = 53191; => 0x 00 00 CF C7

short sx = x; => 0x CF C7

1100 1111 1100 0111

=> -12345

int y = sx // 0x FF FF CF C7

=> // -12345

INTEGER ARITHMETIC

1) Unsigned Addition

```
 1100
+ 100
-----
 1000
```

w=4

W = 4

```
  1 1 1
  1 1 1 1
  1 1 1 1
-----------
[1] 1 1 1 0
```

word size inflation

$$x \underset{w}{\overset{u}{+}} y = \begin{cases} x+y, & x+y < 2^w \\ x+y-2^w, & 2^w \le x+y \le 2^{w+1} \end{cases}$$

unsigned

w bits

Max val for a w-bit vector $\rightarrow 2^w - 1$

$4 \rightarrow 2^4 - 1$

$\rightarrow 15$ //

$1001_2 \rightarrow 9_{10}$

$1100_2 \rightarrow 12_{10}$

$\boxed{1}0101_2 \rightarrow 21$

discard

$\rightarrow 0101_2 = 5_{10}$

$x+y-2^w = 21 - 2^4$

$= 21 - 16$

$= 5$ //

---

int x = 3 bill //

int y = 3 billion

int z = x + y ;

2) Two's complement addition

Will discuss in detail in next class

Range of w-bit vector $\quad -2^{w-1} \le x, y \le 2^{w-1} - 1$

$w = 3$

$8 \le x, y \le 7$

Range of sum $\quad -2^{w} \le x+y \le 2^{w} - 2$

$$x +^{t}_{w} y = \begin{cases} x+y-2^{w}, & 2^{w-1} \le x+y \\ x+y, & -2^{w-1} \le x+y < 2^{w-1} \\ x+y+2^{w}, & x+y < -2^{w-1} \end{cases}$$

$= 2^{w-1} - 1 + 2^{w-1} - 2$

$= 2(2^{w-1}) - 2$

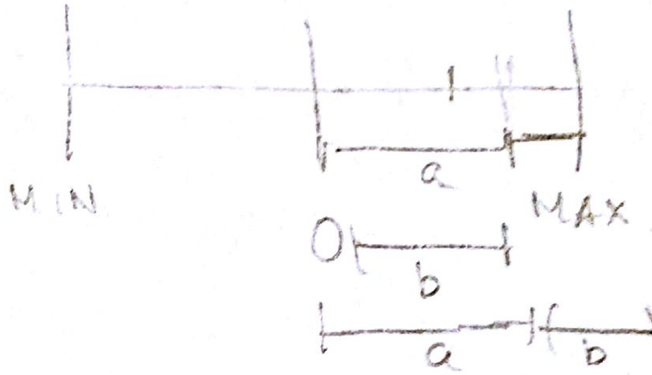
$= \boxed{2^{w} - 2}$

How to test for overflow?

For unsigned $\rightarrow (a, b)$

```
does Overflow (unsigned a, unsigned b)
{
    return ((a+b) < a);
}
```

For **signed (a, b)**




```
does_Overflow (a, b)
{
    if (a > 0)
    {
        return (b > (INT_MAX - a))
    }
    else { // negative

        ______________________ ;

    }
}
```

Multiplying constants

```
int x = 1;
x * 14;
↑
```

Compiler will optimize this // Since 14 is a constant

```
      1 1 1
    x 1 1 1
    -------
      1 1 1
    1 1 1 _
  1 1 1 _ _
  ---------
          1
  ---------
```

Scanned by CamScanner

$x = 3$; // 0011_2.

$x << 1$; // $0110_2 \to 6$

$x << k$

$\downarrow$

$x * 2^k$

$x = 1$

$x * 14 \to (2^3 + 2^2 + 2^1)$

$8 + 4 + 2$

$= x * (2^3 + 2^2 + 2^1) = 14.$

$= (x * 2^3) + (x * 2^2) + (x * 2^1)$

$x = 1$

$= x << 3 + x << 2 + x << 1$

$[0001] << 3 = 1000$

$[0001] << 2 = 0100$

$[0001] << 1 = 0010$

$1110_2 \Rightarrow 14$

$14 \to (2^4 - 2^1)$

$<< 4$ ( )

$<< 1$ (−) ( )