COMPUTER SYSTEMS ORGANIZATION

Computer Arithmetic -- Spring 2010 -- IIIT-H -- Suresh Purini

Unsigned and Signed Integers

Signed or Unsigned Integers – Doesn't Matter! They still need to be represented as a sequence of bits.

- Consider sequences of w-bits. (Ex: 0101 is a 4-bit sequence)
- There are 2^w possible such sequences.
- It is up to us to interpret each of these sequences in the way we would like to.
- In other words we can associate decimal values to these n-bit strings according to our convenience.

Unsigned Integers

Unsigned Integer Interpretation for n-bit strings:

$$B2U_w : \{ 0, 1 \}^w \rightarrow \{ 0, \ldots , 2^w - 1 \}$$

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

Ex: $B2U_4(0101) = 5$, $B2U_4(1001) = 9$, …

- $U_{w\text{-}max} = B2U_w(111..1) = 2^w - 1$ (Max Integer Value)
- $U_{w\text{-}min} = B2U_w(000....0) = 0$ (Min Integer Value)

Signed Integers – Sign-Magnitude Representation

Sign-Magnitude Integer Interpretation for w-bit strings:

$$B2S_w : \{ 0, 1 \}^w \rightarrow \{ -2^{w-1}+1, \ldots 0, \ldots , 2^{w-1}-1\}$$

$$B2S_w(\vec{x}) \doteq (-1)^{x_{w-1}} \cdot \left(\sum_{i=0}^{w-2} x_i 2^i \right)$$

(w-1)th bit is the sign-bit.

Ex: $B2S_4(0101) = 5$, $B2S_4(1001) = -1$,

$B2S_4(0000) = 0$, $B2S_4(1000) = -0$

- $S_{w\text{-}max} = B2S_w(011..1) = 2^{w-1} - 1$ (Max Integer Value)
- $S_{w\text{-}min} = B2S_w(100....0) = -2^{w-1} + 1$ (Min Integer Value)

Signed Integers – One's Complement Representation

One's complement Integer Interpretation for w-bit strings:

$$B2O_w : \{ 0, 1 \}^w \rightarrow \{ -2^{w-1}+1, \ldots 0, \ldots , 2^{w-1}-1\}$$

$$B2O_w(\vec{x}) \doteq -x_{w-1}(2^{w-1}-1) + \sum_{i=0}^{w-2} x_i 2^i$$

Ex: $B2O_4(0101) = 5$, $B2S_4(1001) = -6$,

$B2O_4(0000) = 0$, $B2S_4(1111) = -0$

- $O_{w\text{-}max} = B2O_w(011..1) = 2^{w-1} - 1$ (Max Integer Value)
- $O_{w\text{-}min} = B2O_w(100....0) = -2^{w-1} + 1$ (Min Integer Value)

What is the Shortcut for converting Decimal to One's Complement and vice-versa?

Signed Integers – Two's Complement Representation

Two's complement Integer Interpretation for w-bit strings:

$$B2T_w : \{ 0, 1 \}^w \rightarrow \{ -2^{w-1}, \ldots 0, \ldots , 2^{w-1}-1\}$$

$$B2T_w(\vec{x}) \doteq -x_{w-1}2^{w-1} + \sum_{i=0}^{w-2} x_i 2^i$$

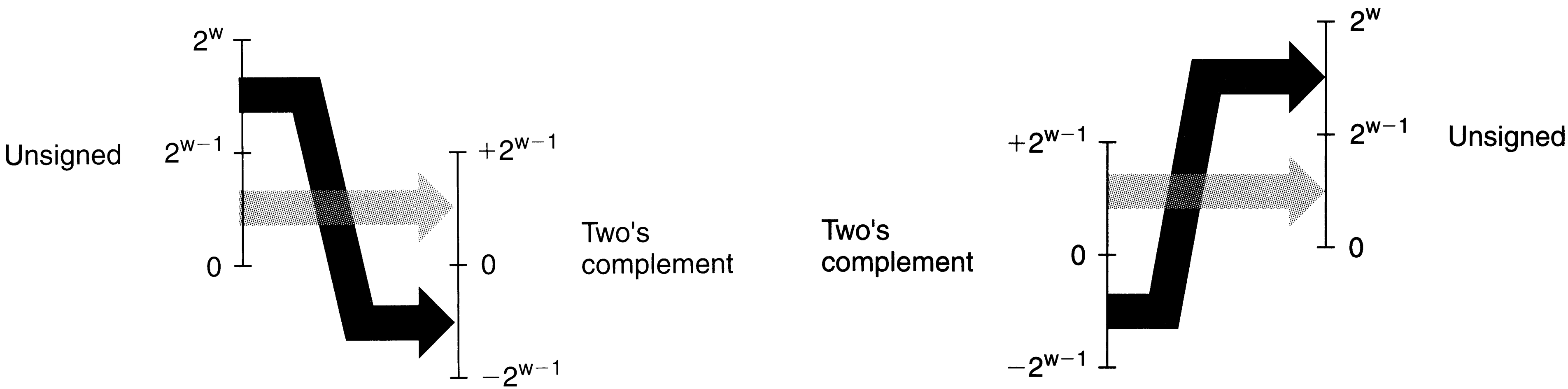
Ex: $B2T_4(0101) = 5$, $B2T_4(1001) = -7$,

$B2OT_4(0000) = 0$, $B2S_4(1111) = -1$

- $T_{w\text{-}max} = B2T_w(011..1) = 2^{w-1} - 1$ (Max Integer Value)
- $T_{w\text{-}min} = B2O_w(100\ldots0) = -2^{w-1}$ (Min Integer Value)

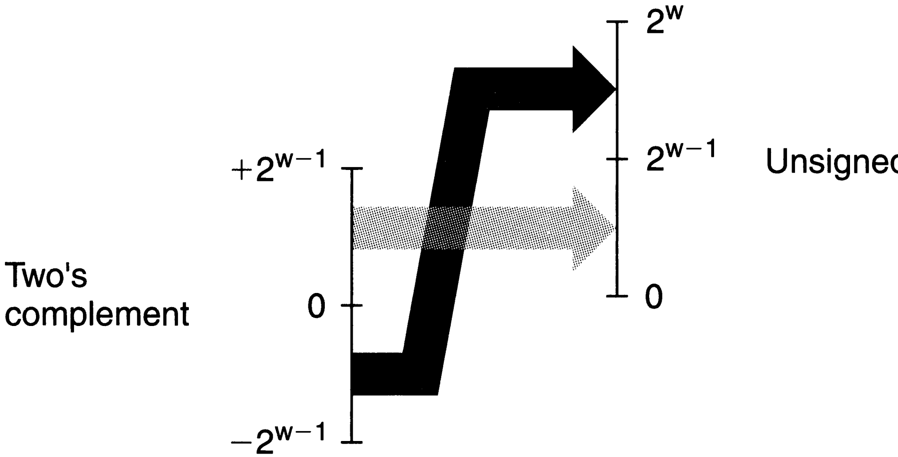
What is the Shortcut for converting Decimal to Two's Complement and vice-versa?

Unsigned and Two’s Complement Representations



Unsigned and Signed (2's Complement) Integer Addition

Row ID	Binary	Unsigned	2's Complement
R_0	000	0	0
R_1	001	1	1
R_2	010	2	2
R_3	011	3	3
R_4	100	4	-4
R_5	101	5	-3
R_6	110	6	-2
R_7	111	7	-1

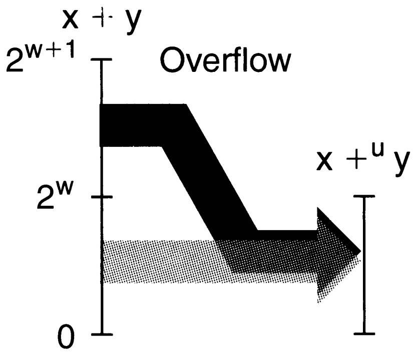


1. $R_2 + R_4 = R_6$
   1. 010 + 100 = 110
   2. 2 + 4 = 6
   3. 2 + (-4) = -2
2. $R_6 + R_7 = R_5$
   1. 110 + 111 = 1101
   2. 6 + 7 = 5 (13 mod 8)
   3. -2 + (-1) = -3
3. $R_2 + R_3 = R_5$
   1. 010 + 011 = 101
   2. 2 + 3 = 5
   3. 2 + 3 = -3

Overflow in Unsigned Integer Addition

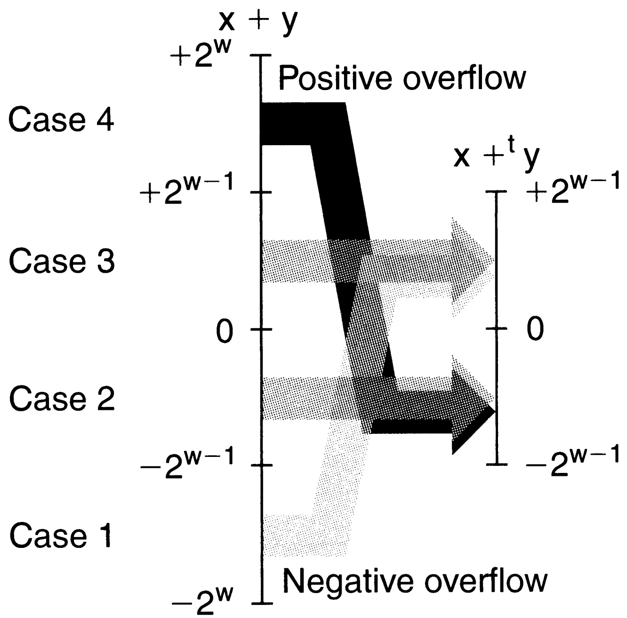
- W-bit Unsigned Integer Addition
  - Overflow: When a carry-bit onto $(w+1)^{th}$ bit position is generated.

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



Overflow in Signed (2's Complement) Integer Addition

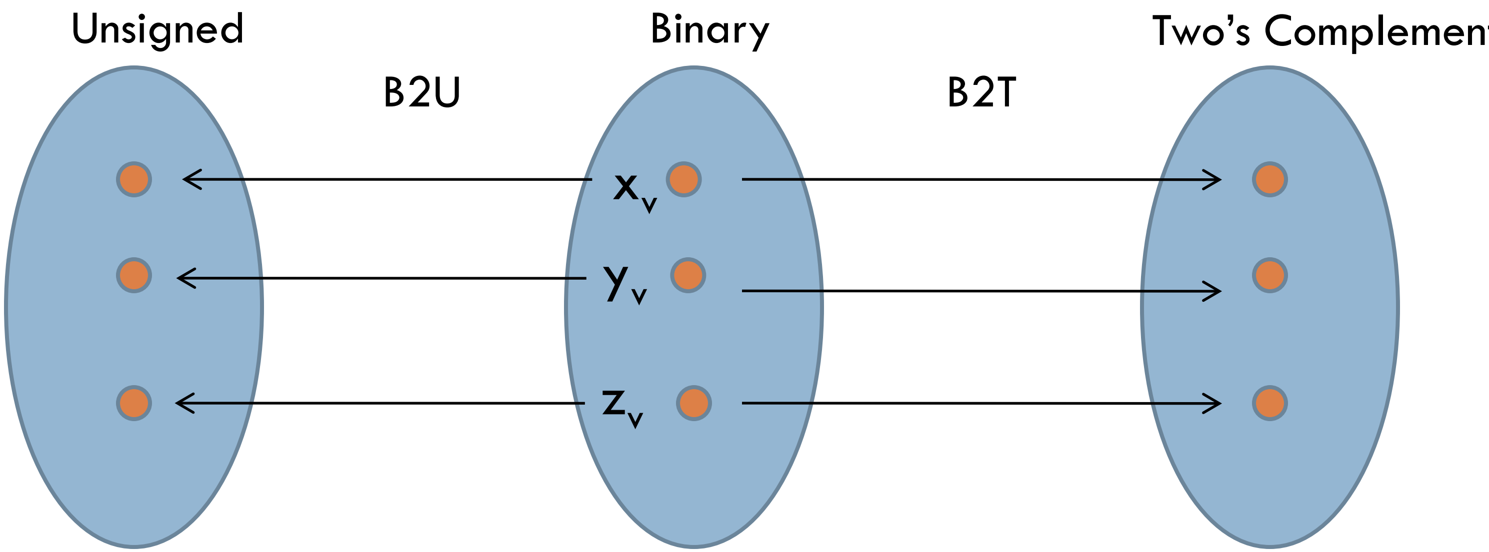
- W-bit signed addition
  - Overflow
    - Two positive integers are added and the result is negative
    - Two negative integers are added and the result is positive



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

Positive Overflow
Normal
Negative Overflow

Isomorphism Between Binary, Unsigned and Two's Complement Integer Arithmetic (modular)



$$x_v + y_v = z_v \pmod{2^w}$$

$$B2U_w(x_v) + B2U_w(y_v) = B2U_w(z_v) \pmod{2^w}$$

$$B2T_w(x_v) + B2T_w(y_v) = B2T_w(z_v) \pmod{2^w}$$

In all the above 3 cases, the mod operator is applied when the result doesn't lie within the set range (overflow)

Binary and One’s Complement Representations

Question: Does isomorphism exists between Binary and One’s Complement Representation of numbers?

Check this 4-bit example:

$$1100 + 1101 = 1001$$

$$-3 + (-2) = -5\ (1010)$$