Integer Representations

Computer Mathematics I

Jiraporn Pooksook
Department of Electrical and Computer Engineering
Naresuan University

Unsigned Encodings

When bit i in the binary representation has value 1, it contributes 2^i to the value.
For example for 4 bits, unsigned number represents from 0 to 15 in decimal notation.

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

$$\begin{aligned}
0000 &= 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 \\
&= 0 \\
1011 &= 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\
&= 11 \\
1111 &= 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\
&= 15
\end{aligned}$$

Unsigned Integer Addition

Addition of unsigned integers is the same as addition of decimal numbers.

- $0 + 0 = 0$
- $1 + 0 = 1$
- $0 + 1 = 1$
- $1 + 1 = 0$, carry 1

The problem is that computers have fixed sized types so we can't go on adding.

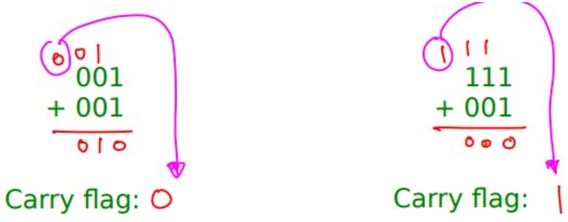

Figure: Retrieved from https://www.cl.cam.ac.uk/teaching/1415/CompFund/NumberSystemsAnnotated.pdf

Unsigned Integer Multiplication

- $101\ .\ 011 = ?$

$$
\begin{array}{rrrrrrl}
 & & & 1 & 0 & 1 & . \\
 & & & 0 & 1 & 1 & \\
\hline
 & & & 1 & 0 & 1 & \\
 & & 1 & 0 & 1 & & + \\
 & 0 & 0 & 0 & & & \\
\hline
0 & 0 & 1 & 1 & 1 & 1 & = 15
\end{array}
$$

- $011\ .\ 011 = 001001 = 9$

Negative Numbers: signed bit

If we represent signed numbers by defining the most significant bit as a sign bit then
0 is used for positive number
1 is used for negative number

$$
\begin{aligned}
0000 &= 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 \\
&= 0 \\
0111 &= 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\
&= 7 \\
1000 &= \text{-}\ [0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0] \\
&= \text{-}0 \\
1111 &= \text{-}\ [1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0] \\
&= \text{-}7
\end{aligned}
$$

If we do normal addition/subtraction, the result is not correct.

$$
\begin{aligned}
1101 + 0001 = 1110\ (\text{-}6) &\neq \text{-}5 + 1 = \text{-}4 \\
&\neq 13 + 1 = 14 \text{(unsigned representation)}
\end{aligned}
$$

Practice: signed-bit format

Convert -53 based-10 to based-2 in signed-bit format ($-53_{10} = ?_2$)

Answer:

base	2^5	2^4	2^3	2^2	2^1	2^0	
multiply	32	16	8	4	2	1	
binary	1	1	0	1	0	1	
sum	32	16	0	4	0	1	$= 53$

$53_{10} = 32+16+4+1 = 110101_2$
Then $-53 = 1\ 110101$

Negative Numbers: one's complement

If we represent signed numbers by defining the most significant bit as a sign bit and
for negative number, the other bits are flipped from 0 to 1 and 1 to 0 then

$$
\begin{aligned}
0000 &= 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 \\
&= 0 \\
0111 &= 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\
&= 7 \\
1111 &= \text{-}\ [0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0] \\
&= \text{-}0 \\
1100 &= \text{-}\ [0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0] \\
&= \text{-}3 \\
1000 &= \text{-}\ [1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0] \\
&= \text{-}7
\end{aligned}
$$

Practice: one's complement

Convert -53 based-10 to based-2 in one's complement format ($-53_{10} = ?_2$)

Answer:

base	2^5	2^4	2^3	2^2	2^1	2^0	
multiply	32	16	8	4	2	1	
binary	1	1	0	1	0	1	
sum	32	16	0	4	0	1	$= 53$

$53_{10} = 32+16+4+1 = 110101_2$
Convert bit 0 to 1 and 1 to 0
$110101 \rightarrow 001010$
Then $-53 = 1\ 001010$

Negative Numbers: two's complement

If we represent signed numbers by defining the most significant bit as a sign bit and
for negative number, the other bits are flipped from 0 to 1 and 1 to 0 and add 1 then

$0000 = 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = 0$
$0111 = 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 7$
$1111 = \text{-}\ [0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 + 0001] = \text{-}1$
$1100 = \text{-}\ [0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0001] = \text{-}4$

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

$1111 = -1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$

$= \text{-}8+4+2+1 = \text{-}1$

Two's complement

	12,345		−12,345		53,191	
Weight	Bit	Value	Bit	Value	Bit	Value
1	1	1	1	1	1	1
2	0	0	1	2	1	2
4	0	0	1	4	1	4
8	1	8	0	0	0	0
16	1	16	0	0	0	0
32	1	32	0	0	0	0
64	0	0	1	64	1	64
128	0	0	1	128	1	128
256	0	0	1	256	1	256
512	0	0	1	512	1	512
1,024	0	0	1	1,024	1	1,024
2,048	0	0	1	2,048	1	2,048
4,096	1	4,096	0	0	0	0
8,192	1	8,192	0	0	0	0
16,384	0	0	1	16,384	1	16,384
±32,768	0	0	1	−32,768	1	32,768
Total		12,345		−12,345		53,191

Figure 2.14 **Two's-complement representations of 12,345 and −12,345, and unsigned representation of 53,191.** Note that the latter two have identical bit representations.

Figure: Retrieved from Computer systems : a programmer's perspective / Randal E. Bryant, David R. O'Hallaron.-2nd ed.

Negative Numbers: two's complement

- Positive to negative: Invert all the bits and add 1

  0111 = 7 $\Rightarrow$ 1000 $\Rightarrow$ 1001 = -7
- Negative to positive: Invert all the bits and add 1

  1001 = -7 $\Rightarrow$ 0110 $\Rightarrow$ 0111 = 7

Practice: two's complement

Convert -53 based-10 to based-2 in two's complement format ($-53_{10} = ?_2$)

Answer:

base	2^5	2^4	2^3	2^2	2^1	2^0	
multiply	32	16	8	4	2	1	
binary	1	1	0	1	0	1	
sum	32	16	0	4	0	1	$= 53$

$53_{10} = 32+16+4+1 = 110101_2$

Convert bit 0 to 1 and 1 to 0

$110101 \rightarrow 001010$

Add 1

$001010 + 000001 = 001011$

Then $-53 = 1\ 001011$

Addition Two's complement

If we do normal addition, the result is not correct.

$$0100 + 0100 = 1000\ (\text{-}8) \neq 4 + 4 = +8$$
$$\neq 4 + 4 = 8(\text{unsigned representation})$$

So we introduce an overflow flag that indicates this problem.

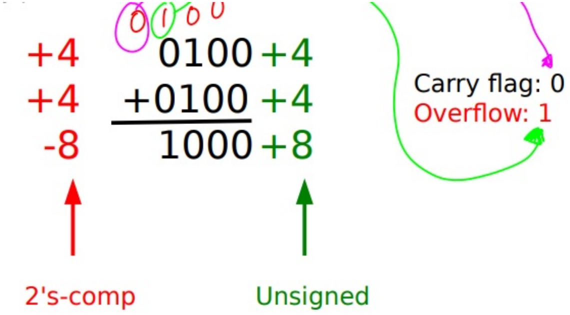

Figure: Retrieved from https://www.cl.cam.ac.uk/teaching/1415/CompFund/NumberSystemsAnnotated.pdf

Subtraction Two's complement

We just add two's complement.

$0100 - 0011 = 0100 + 1101 = 0001$

$1101 - 0100 = 1101 + 1100 = 1001$

Addition/Subtraction Two's complement

Hence, when adding/subtracting

- carry flag $\rightarrow$ overflow for unsigned integer
- overflow flag $\rightarrow$ overflow for signed integer
- If the sum of two positive numbers yields a negative result, the sum has overflowed.
- If the sum of two negative numbers yields a positive result, the sum has overflowed.
- Otherwise, the sum has not overflowed.

Two's complement Multiplication

We extend the integers to twice and just do the normal multiplication. Then take only last 8 digits.

- 1101(-3) . 0011(3) = 1111 1101 . 0000 0011
  = 10 1111 0111 = 1111 0111
  1111 0111 = -9 (two's complement)
- Some cases we don't need to extend bit.
  0011(3) . 0011(3) = 00 1001 = 9 (two's complement)

Mode	x		y		$x \cdot y$		Truncated $x \cdot y$	
Unsigned	5	[101]	3	[011]	15	[001111]	7	[111]
Two's comp.	−3	[101]	3	[011]	−9	[110111]	−1	[111]
Unsigned	4	[100]	7	[111]	28	[011100]	4	[100]
Two's comp.	−4	[100]	−1	[111]	4	[000100]	−4	[100]
Unsigned	3	[011]	3	[011]	9	[001001]	1	[001]
Two's comp.	3	[011]	3	[011]	9	[001001]	1	[001]

Figure 2.26 **Three-bit unsigned and two's-complement multiplication examples.** Although the bit-level representations of the full products may differ, those of the truncated products are identical.

Figure: Retrieved from Computer systems : a programmer's perspective / Randal E. Bryant, David R. O'Hallaron.-2nd ed.

Exercise

Question 1:
Convert 0110101 based-2 in two's complement format to based-10
($0110101_2 = ?_{10}$)

Question 2:
Convert 1001011 based-2 in two's complement format to based-10
($1001011_2 = ?_{10}$)

Question 3:
Convert 1001011 based-2 in one's complement format to based-10
($1001011_2 = ?_{10}$)

Question 4:
Convert 1001011 based-2 in signed bit format to based-10
($1001011_2 = ?_{10}$)

Exercise

Question 5:
Convert -25 based-10 to one's complement format

Question 6:
Convert -25 based-10 to two's complement format