

CompSci 210 Tutorial 2

1. Performing Arithmetic:

value	Sing Magnitude	Offset Binary	1's complement	2's complement
+7	0111	1111	0111	0111
+6	0110	1110	0110	0110
+5	0101	1101	0101	0101
+4	0100	1101	0100	0100
+3	0011	1100	0011	0011
+2	0010	1011	0010	0010
+1	0001	1010	0001	0001
0	0000	1000	0000	0000
-1	1001	0111	1110	1111
-2	1010	0110	1101	1110
-3	1011	0101	1100	1101
-4	1100	0100	1011	1100
-5	1101	0011	1010	1011
-6	1110	0010	1001	1010
-7	1111	0001	1000	1001
-8	//	0000	//	1000
-0	1000	//	1111	//

2. Overflow and underflow in computing:

Overflow and Underflow in addition:

- Adding two numbers with different signs can never produce an overflow or underflow.
- Adding two positive numbers produces an overflow if the sign of the result is negative.
- Adding two negative numbers produces an underflow if the sign of the result is positive.
- Note that in one case there is a carry out and in the other there is not

(+7) 0111 (-7) 1001
(+7) 0111 (-6) 1010
(+14) 1110 (-13) 0011

Overflow and Underflow in Subtraction:

- Subtracting two numbers with the same signs can never produce an overflow or underflow.
- Subtracting a negative number from a positive number produces an overflow
- If the sign of the result is negative.
- Subtracting a positive number from a negative number produces an underflow
- if the sign of the result is positive.

(+4) 0100 0100 -4 1100 1100
-(-5) -1011 0101 -(+5) -0101 1011
+9 1001 -9 0111

3. Example Binary Computation:

3.1) $3 - 2$ or $3 + (-2)$

Answer:

Sign Magnitude: $0011 - 0010 = 0001$

Offset Binary: $1011 + 0110 = (1) 0001$

1's Complement: $0011 + 1101 = (1) 0000 = 0001$

2's Complement: $0011 + 1110 = 0001$

3.2 Given the following bit pattern; 0101101_2

Work out the value of it based on the following assumption:

Unsigned 7-bit binary: (45)

7-bit sign magnitude (MSB is sign bit): (45)

7-bit 2's complement: (45)

XS-33 (Excess-K, $K = 33$): (12)

Unsigned fixed point (assume a 3 bit fraction, 0101.101): (5.625)

3.3 What is result for 17 Add 19 in binary?

And check is overflow or not in 5 bits(unsigned).

$$17_{10} = 10001_2$$

$$19_{10} = 10011_2$$

1 11 <--- Carry bits

(Showing sign bits) 010001

+ 010011

Discard extra bit -> 100100

That will be overflowing just use 5 bits binary, but not overflow in 6 bit binary.

3.4 What is result for -17 Add -19 in binary?

And check is overflow or not in 6 bits.

$$-17_{10} = 101111_2$$

$$-19_{10} = 101101_2$$

1 1111 <--- Carry bits

(Showing sign bits) 101111

+ 101101

Discard extra bit 1011100

FINAL ANSWER: $011100_2 = +28_{10}$

If we use 8 bits to represent the result, that will be $11011100(-36)$.

4. Exercise:

Question 1:

Cover following binary number to Unsigned 7-bit binary: 7-bit sign magnitude (MSB is sign bit): 7-bit 1's complement: 7-bit 2's complement: XS-13 (Excess-K, K = 13): Unsigned fixed point (assume a 3 bit fraction):

0110101

0001110

1101001

1000101

Question 2:

Computing following equations show the result in 16 bits 2's complement binary number:

$-112 + 63 =$

$78 - 13 =$

$-333 + 111 =$

$-123 - 14 =$

Question 3:

Answer the follow computing is overflow or not in 6 bit binary:

$001010 + 010100$

$010101 + 100010$

$110110 + 001111$

$110010 + 110011$

5. Solution:

1.

Unsigned 7-bit binary: (53)
7-bit sign magnitude (MSB is sign bit): (53)
7-bit 1's complement: (53)
7-bit 2's complement: (53)
XS-13 (Excess-K, K = 13): (40)
Unsigned fixed point (assume a 3 bit fraction, 0110.101): (6.625)

Unsigned 7-bit binary: (14)
7-bit sign magnitude (MSB is sign bit): (14)
7-bit 1's complement: (14)
7-bit 2's complement: (14)
XS-13 (Excess-K, K = 13): (1)
Unsigned fixed point (assume a 3 bit fraction, 0001.110): (1.75)

Unsigned 7-bit binary: (105)
7-bit sign magnitude (MSB is sign bit): (-41)
7-bit 1's complement: (-22)
7-bit 2's complement: (-23)
XS-13 (Excess-K, K = 13): (92)
Unsigned fixed point (assume a 3 bit fraction, 1101.001): (13.125)

Unsigned 7-bit binary: (69)
7-bit sign magnitude (MSB is sign bit): (-5)
7-bit 1's complement: (-58)
7-bit 2's complement: (-59)
XS-13 (Excess-K, K = 13): (56)
Unsigned fixed point (assume a 3 bit fraction, 1000.101): (8.625)

2.

$-112+63 = -49$ 1111 1111 1100 1111
78-13 = 65 0000 0000 0100 0001
 $-333+111 = -222$ 1111 1111 0010 0010
 $-123-14 = -137$ 1111 1111 0111 0111

3.

Answer: no
Answer: no
Answer: overflow
Answer: overflow