COMPSCI 210 S1T 2005 Tutorial Five -------Data Representation
Aim for the tutorial:

In this tutorial, we will study examples for cover all knowledge of Data Representation. How to represent binaries, transform, and operate. What is Decimal, binary, octal, and hexadecimal Number representation? Transform octal number to decimal number, hexadecimal number to octal number and so on. Also we need know what different when we represent unsigned and signed number in binary. The 1's complement representation and the 2's complement representation, arithmetic with 1's complement and 2's complement.

1. Number represent in different base number:

Number	Decimal(base 10)	binary (base 2 in four bit)	octal(base8)	hexadecimal(base 16)
0	0	0000	0	0
1	1	0001	1	1
2	2	0010	2	2
3	3	0011	3	3
4	4	0100	4	4
5	5	0101	5	5
6	6	0110	6	6
7	7	0111	7	7
8	8	1000	//	8
9	9	1001	//	9
10	10	1010	//	A
11	11	1011	//	B
12	12	1100	//	C
13	13	1101	//	D
14	14	1110	//	E
15	15	1111	//	F
……	……	//	//	//

Number	Decimal(base 10)	binary (base 2 in 12 bit)	octal(base8)	hexadecimal(base 16)
1	1	0000 0000 0001	1	1
2	2	0000 0000 0010	2	2
4	4	0000 0000 0100	4	4
8	8	0000 0000 1000	10	8
16	16	0000 0001 0000	20	10
32	32	0000 0010 0000	40	20
64	64	0000 0100 0000	100	30
128	128	0000 1000 0000	200	80
256	256	0001 0000 0000	400	100
512	512	0010 0000 0000	1000	200
1024	1024	0100 0000 0000	2000	400
2048	2048	1000 0000 0000	4000	800

2. Transformation of the number:

The powers of ten are determined by the position relative to the decimal point. Using positional coefficients and weights we can express any weighted number System in the following generalized form:

$X = x_n w_n + x_{n-1} w_{n-1} + \ldots + x_{-1} w_{-1} + \ldots + x_{-m} w_{-m}$ (follow given condition in the lecture)

2.1 transform binary, octal, and hexadecimal to **decimal**

11001 (base 2) is:
$11001_2 = 1 \times 2^{**}4 + 1 \times 2^{**}3 + 0 \times 2^{**}2 + 0 \times 2^{**}1 + 1 \times 2^{**}0 = 25_{10}$

11361 (base 8) is:
$11361_8 = 1 \times 8^{**}4 + 1 \times 8^{**}3 + 3 \times 8^{**}2 + 6 \times 8^{**}1 + 1 \times 8^{**}0 = 4849_{10}$

11ACF (base 16) is:
$11ACF_{16} = 1 \times 16^{**}4 + 1 \times 16^{**}3 + 10 \times 16^{**}2 + 12 \times 16^{**}1 + 1 \times 16^{**}0 = 72399_{10}$

2.2 transform decimal, octal, and hexadecimal to **binary**

method: subtract largest power of 2 smaller than 51 until you reach 1:
$51_{10}=32+16+2+1=2^{**}5+2^{**}4+0+0+2^{*}1+2^{*}0= 110011_2$

Tip: Best way to transform from decimal to octal is to go via Binary Octal representation to binary representation

$345_8 = 3 \times 8^{**}2 + 4 \times 8^{**}1 + 5 \times 8^{**}0$
In Binary: 011 100 101 → 011100101_2

$38F_{16} = 3 \times 16^{**}2 + 8 \times 16^{**}1 + 15 \times 6^{**}0$
In Binary: 0011 1000 1111 → 001110001111_2

2.3 transform binary, decimal, and hexadecimal to **octal**

111001101011_2 = 111 001 101 011 = 7153_8

$51_{10}=32+16+2+1=2^{**}5+2^{**}4+0+0+2^{*}1+2^{*}0= 110011_2$ → ***0***11 0011_2 → 31_8（please group by 3its starting from lsb）

$2EF78_{16}$ = 0010 1110 1111 0111 1000_2 = ***0***00 101 110 111 101 111 000_2 = 567570_8

2.4 transform binary, octal, and decimal to **hexadecimal**

10101001110101011_2 = ***000***1 0101 0011 1010 1011_2 = $153AB_{16}$ （please group by 4 its starting from lsb）

37672_8 →011 111 110 111 010_2 → ***0***011 1111 1011 1010_2 → $3FBA_{16}$

1656_{10} = 1024+512+64+32+16+8 = 2**10+2**9+2**6+2**5+2**4+2**3 =11001111000_2 → 0110 0111 1000_2 → 678_{16}

3. Binary 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	//

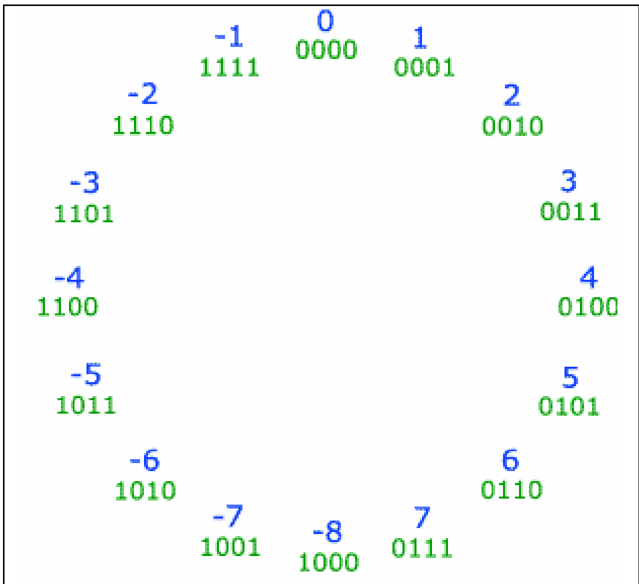
All the numbers you've looked at so far have been positive whole numbers. There is no equivalent in binary to the minus sign so other ways have been devised to represent negative values. The two most widely used are: **Sign magnitude and two's complement.**

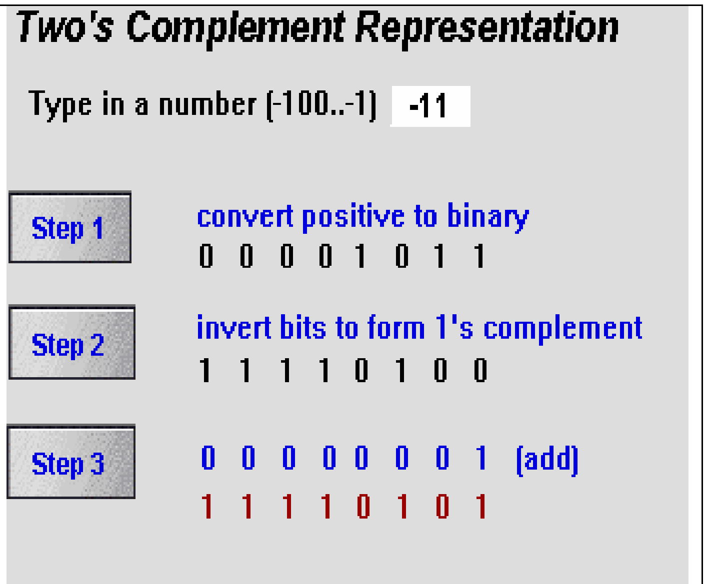
The rules for binary arithmetic are shown in the table below.

0 + 0	= 0
1 + 0	= 1
1 + 1	= 10
1 + 1 + 1	= 11

This means that a 4-bit number cannot represent any value greater than 7 because the 4th bit is used to indicate the sign. However the 4 bits can still represent 15 different values when the negative numbers are included. These values are:
-7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4 , 5, 6, 7
If there was no sign bit, 4 bits would represent the following 16 values:
0, 1, 2, 3, 4, 5 , 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

Decimal arithmetic uses different operations for addition and subtraction. Using two's complement, subtraction is carried out using the machine operation for addition. This system works for bit strings of any length. The examples given will use 8 or 4-bit strings for simplicity but a working PC is likely to use 32 or 64-bit strings. The first bit is again used to indicate a negative value but it also bears the position value.



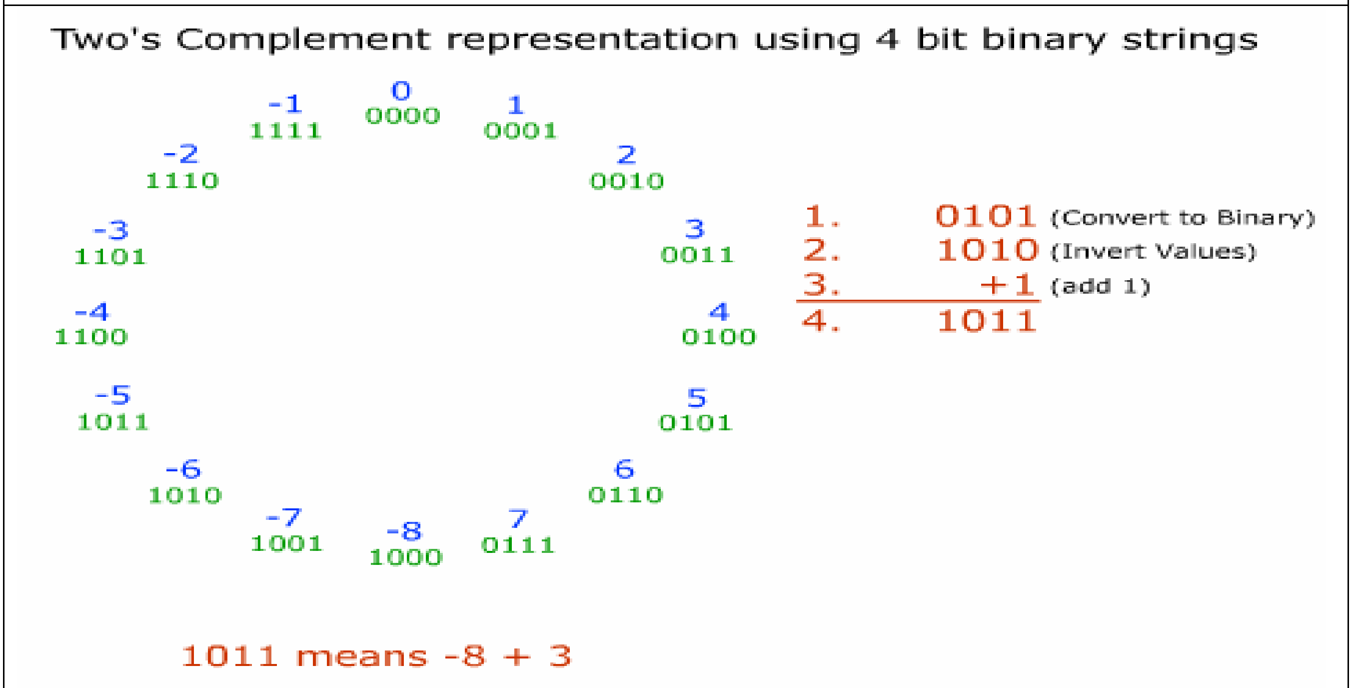


Converting a number to its negative form is a two stage process:

complement (invert) all the bits (the result at this stage is known as one's complement).

add 1.

If you're working with 8-bit strings all values must appear as 8 bits and the possible values will be in the range -128 to 127.



Example of Subtraction:

The processor treats subtraction like the addition of a negative number.

A - B = A + (-B)

All examples will be shown using 8 bit two's complement strings.

Subtraction is a 3 stage process:
invert the number to be subtracted;
add 1;
perform addition.

- Subtraction of 19 from 53
To evaluate 53 - 19

Step 1 19	0001 0011
invert	1110 1100
Step 2 add 1	0000 0001
	1110 1101
Step 3 add 53 and -19	0011 0101
	1110 1101
1	0010 0010

The bit on the left is overflow and is not included in the result. So the result of *53 - 19 = 0010 0010*.
You should always check the answer by converting it to decimal.
0010 0010 *32 + 2 = 34*
53 - 19 = 34

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
(+6) 0110	(-4) 1100
(+13) 1101	(-11) 0101

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

4. Exercise:

Question 1:
Can you work out why there are only 15 values when you use sign and magnitude but 16 when there is no sign bit?
Answer:
If you write out all the values for both representations, you'll see that there are two different versions of zero in sign and magnitude (*0 = 0000* and *-0 = 1000*). These mean exactly the same.

Question 2:
What is two's complement number for 1000 1010?

Answer:
1000 1010
= -128 +0+0+0+8+0+2+0
= -118

Question 3:
To convert 1101 (13_{10}) to its negative:

Answer:

.write the number in 8 bit format	0000 1101
invert the bits	1111 0010
add 1	0000 0001
	1111 0011

It is easy to check the answer:
1111 0011
= -128 +64 +32 +16 +0 +0 +2 +1
= -128 + 115
= -13

Q4 What is the decimal value of the unsigned 8-bit integer at location 0021

a. Hex value at location 0021 is 9B

b. Convert 9B to binary

9				B			
1	0	0	1	1	0	1	1

c. Convert binary to decimal

128	64	32	16	8	4	2	1	
1	0	0	1	1	0	1	1	128+16+8+2+1

d. Add up 128+16+8+2+1 = 155

Decimal value is 155

Q5 What is the octal value of the unsigned 8-bit integer at location 002B

a. Hex value at location 002B is CA

b. Convert to binary

C				A			
1	1	0	0	1	0	1	0

c. Convert binary to octal (groups of 3 bits from LSB end)

1	1	0	0	1	0	1	0
3		1			2		

Octal value is 312

Q6 What is result (in hexadecimal) of adding the 8-bit unsigned integers at locations 0058 and 005F

Hex value at 0058 is 54, Hex value at 005F is 04

a. Convert to 54 binary

5				4			
0	1	0	1	0	1	0	0

b.Convert binary to decimal

128	64	32	16	8	4	2	1
0	1	0	1	0	1	0	0

=64+16+4=84

c. Convert 04 to binary

0				4			
0	0	0	0	0	1	0	0

d. Convert binary to decimal

128	64	32	16	8	4	2	1
0	0	0	0	0	1	0	0

=4

d. Add decimal values 84+4= 88

e. Convert decimal 88 to binary

128	64	32	16	8	4	2	1
0	1	0	1	1	0	0	0

f. Convert binary to hex

128 128	64	32	16	8	4	2	1
0	1	0	1	1	0	0	0
5				8			

Hex value of sum is 58

Q7 What is result (in hexadecimal) of adding the decimal value 63 to the 8-bit unsigned integer at location 0047

Hex value at location 0047 is 28

a. Convert Hex 28 to binary

2				8			
0	0	1	0	1	0	0	0

b.Convert binary to decimal

128	64	32	16	8	4	2	1
0	0	1	0	1	0	0	0

= 32+8 = 40

c. Decimal addition 63+40 = 103

d. Convert decimal 103 to binary

128	64	32	16	8	4	2	1
0	1	1	0	0	1	1	1

e. Convert to binary to hex

0	1	1	0	0	1	1	1
6				7			

Hex value of sum is 67

In questions 5,6 and 7 signed 8 bit integers are represented in 2's complement form

Q8 What is the decimal value of the signed 8-bit integer at location 007D
Hex value at location 007D is 8C

a. Convert to binary

8				C			
1	0	0	0	1	1	0	0

b. Inspect the sign bit = 1, number is negative

c. Convert from 2's complement to normal binary

(i) Subtract 1

1	0	0	1	1	1	0	0	
0	0	0	0	0	0	0	1	-
1	0	0	0	1	0	1	1	

(ii)
Invert the bits, (turn 1 to 0 and 0 to 1)

1	0	0	0	1	0	1	1
0	1	1	1	0	1	0	0

d. Convert binary to decimal

128 128	64	32	16	8	4	2	1
0	1	1	1	0	1	0	0

=64+32+16+4=116
e. Add back the sign noted in b
Decimal value is –116

Q9 What is the result (in hexadecimal) of adding the 8-bit signed at locations 008B and 0091

Hex value at 008B is 9D, hex value at 0091 is 27
Working with 9D
a. Convert hex to binary

9				D			
1	0	0	1	1	1	0	1

b. Inspect the sign bit = 1, number is negative
c. Convert from 2's complement to normal binary
(i) Subtract 1

1	0	0	1	1	1	0	1	
0	0	0	0	0	0	0	1	-
1	0	0	1	1	1	0	0	

(ii) Invert the bits, (turn 1 to 0 and 0 to 1)

1	0	0	1	1	1	0	0
0	1	1	0	0	0	1	1

d. Convert binary to decimal

128	64	32	16	8	4	2	1
0	1	1	0	0	0	1	1

=64+32+2+1=99
e. Add back the sign noted in b
Decimal value is –99

Working with 27
f. Convert hex to binary

2				7			
0	0	1	0	0	1	1	1

g. Inspect the sign bit = 0, number is positive
h. Convert from normal binary (2's complement only for negative numbers)
i. Convert binary to decimal

128	64	32	16	8	4	2	1
0	0	1	0	0	1	1	1

=32+4+2+1=39
j. Result of decimal addition –99+39 = -60
k. Check sign of result (negative)
l. Convert decimal to binary, need 2's complement since it is negative

m. Convert 60 to binary

128	64	32	16	8	4	2	1
0	0	1	1	1	1	0	0

m. Convert to 2's complement

(i) Invert the bits, (turn 1 to 0 and 0 to 1)

0	0	1	1	1	1	0	0
1	1	0	0	0	0	1	1

(ii) Add 1

1	1	0	0	0	0	1	1	
0	0	0	0	0	0	0	1	+
1	1	0	0	0	1	0	0	

n. Convert binary to hex

1	1	0	0	0	1	0	0
C				4			

Hex value of result of addition is C4

Q10 What is signed 8-bit integer that results from adding the decimal value 63 to the 8-bit signed integer at location 00BF (show as hexadecimal)

Hex value at 00BF is 92

a. Convert hex to binary

9				2			
1	0	0	1	0	0	1	0

b. Inspect the sign bit = 1, number is negative

c. Convert from 2's complement to normal binary

(i) Subtract 1

1	0	0	1	0	0	1	0	
0	0	0	0	0	0	0	1	-
1	0	0	1	0	0	0	1	

(ii) Invert the bits, (turn 1 to 0 and 0 to 1)

1	0	0	1	0	0	0	1
0	1	1	0	1	1	1	0

d. Convert binary to decimal

128	64	32	16	8	4	2	1
0	1	1	0	1	1	1	0

=64+32+8+4+2=110

e. Add back the sign noted in b

Decimal value is –-110

f. Result of decimal addition –110+63=-47

g. Convert –47 to binary, note that it is negative to 2's complement is used

(i) convert 47 to binary

128 128	64	32	16	8	4	2	1
0	0	1	0	1	1	1	1

(ii) Invert the bits

0	0	1	0	1	1	1	1
1	1	0	1	0	0	0	0

(iii) Add 1

1	1	0	1	0	0	0	0

							1	+
1	1	0	1	0	0	0	1	

h. Convert binary to hex

1	1	0	1	0	0	0	1
D				1			

Hex result of addition is D1

5. Reference:

http://scholar.hw.ac.uk/site/computing/topic1.asp?outline=no
http://www-ee.eng.hawaii.edu/Courses/EE150/Book/chap1/subsection2.1.2.1.html