GERMANNA
ACADEMIC CENTER
FOR EXCELLENCE

CS & IT Conversions

There are several number systems that you will use when working with computers. These include decimal, binary, octal, and hexadecimal. Knowing how to convert between these number systems is very important. Note: All the following number systems will explain only positive integers unless otherwise stated.

You can navigate to specific sections of this handout by clicking the links below.

Decimal (Base 10)

Decimal or base 10 is the number system that is normally used. The digits 0-9 are used to write all numbers. Reviewing how decimal works will make working in other number systems easier. In decimal, each place has a value or magnitude; for example, tens place and ones place. Each place is a power of 10. See Example 1. All number systems follow this same format.

Example 1: 25,198

Magnitude as Powers of Ten	10^4	10^3	10^2	10^1	10^0
Magnitude	10,000	1,000	100	10	1
	2	5	1	9	8

So the number 25,198 is the same as:

- $2 * 10^4 + 5 * 10^3 + 1 * 10^2 + 9 * 10^1 + 8 * 10^0$
- $2 * 10{,}000 + 5 * 1{,}000 + 1 * 100 + 9 * 10 + 8 * 1$
- $20{,}000 + 5{,}000 + 100 + 90 + 8$

Unsigned Binary (Base 2)

The difference between unsigned binary and decimal is that you have only two digits that are used to make up all binary numbers instead of ten. The two digits in binary are **0** and **1**. The place values are powers of two instead of powers of ten. Just as decimal has 10^0 at the right most place, binary has 2^0 at the right most place. Just as decimal has 10^1 one place left of 10^0, binary has 2^1 one place left of 2^0.

Example 2: Convert the unsigned binary number 101 to decimal.

Magnitude as Powers of 2	2^2	2^1	2^0
Magnitude	4	2	1
	1	0	1

The binary number 101 is the same as these values in decimal:

- $1 * 2^2 + 0 * 2^1 + 1 * 2^0$
- $1 * 4 + 0 * 2 + 1 * 1$
- $4 + 0 + 1$

The binary number 101 is **5** in decimal.

To convert from decimal to unsigned binary, again use the magnitude of each place value in binary. Start by writing out the magnitudes of binary until you write a value that is larger than the number you are trying to write. Then subtract the largest magnitude that is smaller than the number you are converting to decimal from the number you are converting to decimal. Place a one under that magnitude and repeat until you have 0. Then fill in any empty spaces with zeros.

Example 3: Convert 124 to unsigned binary.

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

124 is **1111100** in unsigned binary.

```
  124
- 64
  60
- 32
  28
- 16
  12
- 8
   4
- 4
   0
```

Example 4: Convert 65 to unsigned binary.

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

65 is **1000001** in unsigned binary.

```
  65
- 64
   1
- 1
   0
```

Octal (Base 8)

With octal you have 8 digits, the numbers 0-7, and the place values are the powers of 8.

Example 5: Convert 3415_8 to decimal.

(When you see a subscript as in this example, it tells you which number system the number is in. In this case 8 represents base 8 or octal.)

Magnitude as Powers of 8	8^3	8^2	8^1	8^0
Magnitude	512	64	8	1
	3	4	1	5

3415_8 is the same as these values in decimal:

- $3 * 8^3 + 4 * 8^2 + 1 * 8^1 + 5 * 8^0$
- $3 * 512 + 4 * 64 + 1 * 8 + 5 * 1$
- $1536 + 256 + 8 + 5$
- **1805**

To convert from decimal to octal, the steps are similar as converting from decimal to binary. First, write out the magnitudes of octal until you write a magnitude that is larger than the number you are converting. Next, divide the number you are converting by each magnitude. Write the integer part of the quotient under the magnitude. Then multiply the magnitude and the integer part of the quotient under it. Next, subtract the product from the number you are converting. Repeat this process until you have zero.

Example 6: Convert 197 to octal.

Step 1: 197/512 = 0 R 197

Magnitude	512	64	8	1
	0			

Step 2: 197/64 = 3 R 5
64 * 3 = 192

	197
-	192
	5

Magnitude	512	64	8	1
	0	3		

Step 3: 5/8 = 0 R 5
8 * 0 = 0

	5
-	0
	5

Magnitude	512	64	8	1
	0	3	0	

	5

Step 4: 5/1 = 5
1 * 5 = 5

-	5
	0

Magnitude	512	64	8	1
	0	3	0	5

So 197 is **305_8** in octal.

Hexadecimal (Base 16)

For number systems that use more than 10 digits to represent numbers, use the letters of the alphabet as the extra digits. In hexadecimal you have 16 digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F. A represents 10. B represents 11. C represents 12. D represents 13. E represents 14. F represents 15. The place values are the powers of 16.

Example 7: What is $2CA_{16}$ in decimal?

Magnitude as Powers of 16	16^2	16^1	16^0
Magnitude	256	16	1
	2	C	A

Remember that C is 12 and A is 10.
$2CA_{16}$ is the same as these values in decimal:

- $2 * 16^2 + C * 16^1 + A * 16^0$
- $2 * 16^2 + 12 * 16^1 + 10 * 16^0$
- $2 * 256 + 12 * 16 + 10 * 1$
- $512 + 192 + 10$
- **714**

Example 8: What is $1FF_{16}$ in decimal?

Magnitude as Powers of 16	16^2	16^1	16^0
Magnitude	256	16	1
	1	F	F

Remember that F is 15.
$1FF_{16}$ is the same as these values in decimal:

- $1 * 16^2 + F * 16^1 + F * 16^1$
- $1 * 16^2 + 15 * 16^1 + 15 * 16^1$
- $1 * 256 + 15 * 16 + 15 * 1$
- $256 + 240 + 15$
- **511**

To convert from decimal to hexadecimal, you will follow the same steps as with octal. First, write out the magnitudes of hexadecimal until you write a magnitude that is larger than the number you are converting. Next, divide the number you are converting by each magnitude. Write the integer part of the quotient under the magnitude. Then multiply the magnitude and the integer part of the quotient under it. Next, subtract the product from the number you are converting. Repeat this process until you have zero.

Example 9: Convert 2000 to hexadecimal.

Step 1: 2000/4096 = 0 R 2000

Magnitude	4096	256	16	1
	0			

Step 2: 2000/256 = 7 R 208
256 * 7 = 1792

	2000
-	1792
	208

Magnitude	4096	256	16	1
	0	7		

Step 3: 208/16 = 13
16 * 13 = 208

	208
-	208
	0

Remember 13 = D

Magnitude	4096	256	16	1
	0	7	D	

Step 4: 0/1 = 0
1 * 0 = 0

	0
-	0
	0

Magnitude	4096	256	16	1
	0	7	D	0

2000 is **$7D0_{16}$** in hexadecimal.

Numbers with Difference Bases

Unsigned binary is written with leading zeros if you are given a set number of bits (digits) to use. Also note that the largest number you can write in 4-bit unsigned binary is 15. You can not write 16 because it requires 5 bits and you may only use 4. 4-bit and 8-bit unsigned

binary is below as examples. When the number of bits are stated, you must use that number as with the 4-bit and 8-bit unsigned binary. See table below.

Numbers with Difference Bases					
Decimal (Base 10)	Unsigned Binary (Base 2)	Octal (Base 8)	Hexadecimal (Base 16)	4-bit Unsigned Binary	8-bit Unsigned Binary
0	0	0	0	0000	00000000
1	1	1	1	0001	00000001
2	10	2	2	0010	00000010
3	11	3	3	0011	00000011
4	100	4	4	0100	00000100
5	101	5	5	0101	00000101
6	110	6	6	0110	00000110
7	111	7	7	0111	00000111
8	1000	10	8	1000	00001000
9	1001	11	9	1001	00001001
10	1010	12	A	1010	00001010
11	1011	13	B	1011	00001011
12	1100	14	C	1100	00001100
13	1101	15	D	1101	00001101
14	1110	16	E	1110	00001110
15	1111	17	F	1111	00001111
16	10000	20	10	-	00010000
17	10001	21	11	-	00010001
18	10010	22	12	-	00010010
19	10011	23	13	-	00010011
20	10100	24	14	-	00010100
21	10101	25	15	-	00010101
22	10110	26	16	-	00010110
23	10111	27	17	-	00010111
24	11000	30	18	-	00011000
25	11001	31	19	-	00011001
26	11010	32	1A	-	00011010
27	11011	33	1B	-	00011011
28	11100	34	1C	-	00011100
29	11101	35	1D	-	00011101
30	11110	36	1E	-	00011110
31	11111	37	1F	-	00011111
32	100000	40	20	-	00100000

Signed Numbers in Binary

Signed binary works with both positive and negative numbers while unsigned binary works with only positive numbers. There are a number of ways to represent signed numbers in binary. One way is to use the leftmost bit as a sign bit. 0 is for positive. 1 is for negative. Then have the number follow. When this is done, you need to know how many bits to use. Note: This representation for signed numbers in binary it is not commonly used.

Example 10: Convert the decimal number 4 to 4-bit signed binary representation.

$4 = 100_2$. 4 is positive so the signed bit is 0.
0100 is 4 in 4-bit signed binary.

Example 11: Convert the decimal number -6 to 4-bit signed binary representation.

$6 = 110_2$. -6 is negative so the signed bit is 1.
1110 is -6 in 4-bit signed binary.

Two's Complement

The most common representation for signed binary numbers is two's complement. If you add a positive and a negative number in signed binary, their answer will not be correct. This is due to the way the numbers are represented. The reason two's complement is the most common is that if you add a positive and negative number together, this answer will be the correct value in two's complement. When working with two's complement, all the numbers must have the same number of bits (digits). Also, the left most bit is called the sign bit just like signed binary. It tells you if the number is positive or negative. 0 is for positive. 1 is for negative. Positive numbers in two's complement are the same as unsigned binary except that the leftmost bit (the sign bit) is 0. The only difference is that the left most bit represents sign and not a magnitude.

Example 12: Convert the decimal number 3 to 4-bit two's complement.

$3 = 11_2$. 3 is positive so the sign bit is 0.

±	4	2	1
0	0	1	1

0011 is 3 in 4-bit two's complement.

Example 13: Convert the decimal number 7 to 6-bit two's complement.

$7 = 111_2$. 7 is positive so the sign bit is 0.

±	16	8	4	2	1
0	0	0	1	1	1

000111 is 7 in 6-bit two's complement.

To write a negative number in two's complement is more difficult. The first step is to write the number as a positive number in two's complement. Then change all the bits. If it is 0, make it a 1. If it is a 1, make it a 0. The last step is to add one to the binary number. The sign bit should always be 1 for negative numbers. After the addition, the number is now in Two's Complement.

Example 14: What is -11 in 8-bit two's complement?

Step 1: 11 is 00001011_2 in 8-bit two's complement

Step 2: Changing the bits
00001011_2 becomes 11110100_2

Step 3:
$$\begin{array}{r} 11110100_2 \\ +\quad\quad 1_2 \\ \hline 11110101_2 \end{array}$$

So **11110101** is -11 in 8-bit two's complement.

Converting from two's complement to decimal follows the same general steps as converting from decimal to two's complement. First change the bits. Then add one to the number in binary. Remember not to change the sign bit. Then convert the number from binary to decimal. Lastly, change the sign of the decimal number.

Example 15: Convert 1110010 which is in two's complement to decimal.

Step 1: Changing the bits
1110010_2 becomes 0001101_2

Step 2:
$$\begin{array}{r} 0001101_2 \\ +\quad\quad 1_2 \\ \hline 0001110_2 \end{array}$$
Binary addition (Only 0s and 1s)
1 + 1 = 10

Step 3:

±	32	16	8	4	2	1
0	0	0	1	1	1	0

8 + 4 + 2 =14

Step 4: **-14** is 1110010 in 7-bit two's complement.

The following table allows you to see the differences between the various ways to represent numbers in binary using 4-bits. Also note that -8 can be represented in 4-bit two's complement but 8 can not be represented. 8 requires 5-bits in two's complement so it can not be written with 4-bits. Note the difference between Signed Binary and Two's Complement. With signed binary, the first bit is a sign bit and the number is the bits following the sign bit in unsigned binary. Two's complement is the complement of the number with one added to it.

4-bit Conversion Table

Decimal	Unsigned Binary	Signed Binary	Two's Complement
-8	-	-	1000
-7	-	1111	1001
-6	-	1110	1010
-5	-	1101	1011
-4	-	1100	1100
-3	-	1011	1101
-2	-	1010	1110
-1	-	1001	1111
-0	-	1000	-
0	0000	0000	0000
1	0001	0001	0001
2	0010	0010	0010
3	0011	0011	0011
4	0100	0100	0100
5	0101	0101	0101
6	0110	0110	0110
7	0111	0111	0111
8	1000	-	-
9	1001	-	-
10	1010	-	-
11	1011	-	-
12	1100	-	-
13	1101	-	-
14	1110	-	-
15	1111	-	-

Sample Problems

Convert these numbers to decimal.

1. 156_{16}
2. EA_{16}
3. F_{16}
4. $1D_{16}$
5. 25_{8}
6. 71_{8}
7. 20_{8}
8. 46_{8}
9. 10010_{2}
10. 1001_{2}
11. 11_{2}
12. 100_{2}

Convert these numbers to binary.

13. 54
14. 7
15. 64
16. 15

Convert these numbers to octal.

17. 8
18. 29
19. 123
20. 21

Convert these numbers to hexadecimal

21. 12
22. 41
23. 54
24. 17

Convert these numbers to 7-bit signed binary.

25. 35
26. 20
27. -48
28. -6

Convert the following 5-bit signed binary to decimal.

29. 10010
30. 01101
31. 00000
32. 11010

Write the following decimal values in 2's Complement using 8 bits.

33. 5
34. -1
35. -17
36. -20
37. 4
38. -3
39. -14
40. -16

Write the following Two's Complement in decimal.

41. 111011
42. 1101011
43. 11111
44. 1000
45. 011011

GERMANNA
ACADEMIC CENTER
FOR EXCELLENCE

Answers

1. 342
2. 234
3. 15
4. 29
5. 21
6. 57
7. 16
8. 38
9. 18
10. 9
11. 3
12. 4
13. 110110_2
14. 111_2
15. 1000000_2
16. 1111_2
17. 10_8
18. 35_8
19. 173_8
20. 25_8
21. C_{16}
22. 29_{16}
23. 36_{16}
24. 11_{16}
25. 0100011
26. 0010100
27. 1110000
28. 1000110
29. -2
30. 13
31. 0
32. -10
33. 00000101
34. 11111111
35. 11101111
36. 11101100
37. 00000100
38. 11111101
39. 11110010
40. 11110000
41. -5
42. -21
43. -1
44. -8
45. 27