Lab 1
Number Systems

Objectives

- Students will come to know about various number systems.
- They will be able to perform conversions between number systems.
- They will be able to perform basic arithmetic operations on binary number systems.
- They will be able to perform addition in hexadecimal and octal number systems.
- They will be able to represent singed numbers.
- They will be able to represent BCD numbers and perform their addition.

NUMBER SYSTEM

Number systems are the technique to represent numbers in the computer system architecture, every value that you are saving or getting into/from computer memory has a defined number system.

Computer architecture supports following number systems.

- **Binary number system**
- **Octal number system**
- **Decimal number system**
- **Hexadecimal (hex) number system**

BINARY NUMBER SYSTEM

A Binary number system has only two digits that are **0 and 1**. Every number (value) represents with 0 and 1 in this number system. The base of binary number system is 2, because it has only two digits.

OCTAL NUMBER SYSTEM

Octal number system has only eight (8) digits from **0 to 7**. Every number (value) represents with 0,1,2,3,4,5,6 and 7 in this number system. The base of octal number system is 8, because it has only 8 digits.

DECIMAL NUMBER SYSTEM

Decimal number system has only ten (10) digits from **0 to 9**. Every number (value) represents with 0,1,2,3,4,5,6, 7,8 and 9 in this number system. The base of decimal number system is 10, because it has only 10 digits.

HEXADECIMAL NUMBER SYSTEM

A Hexadecimal number system has sixteen (16) alphanumeric values from **0 to 9** and **A to F**. Every number (value) represents with 0,1,2,3,4,5,6, 7,8,9,A,B,C,D,E and F in this number system. The base of hexadecimal number system is 16, because it has 16 alphanumeric values. Here A is 10, B is 11, C is 12, D is 14, E is 15 and F is 16.

Number system	Base(Radix)	Used digits	Example
Binary	2	0,1	$(11110000)_2$
Octal	8	0,1,2,3,4,5,6,7	$(360)_8$
Decimal	10	0,1,2,3,4,5,6,7,8,9	$(240)_{10}$
Hexadecimal	16	0,1,2,3,4,5,6,7,8,9, A,B,C,D,E,F	$(F0)_{16}$

CONVERSIONSDECIMAL TO OTHER

1. DECIMAL TO BINARY

Decimal Number System to Other Base

To convert Number system from **Decimal Number System** to **Any Other Base** is quite easy; you have to follow just two steps:

A) Divide the Number (Decimal Number) by the base of target base system (in which you want to convert the number: Binary (2), octal (8) and Hexadecimal (16)).

B) Write the remainder from step 1 as a Least Signification Bit (LSB) to Step last as a Most Significant Bit (MSB).

Decimal to Binary Conversion				Result
Decimal Number is : **$(12345)_{10}$**				
2	12345	1	**LSB**	Binary Number is **$(11000000111001)_2$**
2	6172	0		
2	3086	0		
2	1543	1		
2	771	1		
2	385	1		
2	192	0		
2	96	0		
2	48	0		
2	24	0		
2	12	0		
2	6	0		
2	3	1		
	1	1	**MSB**	

2. DECIMAL TO OCTAL

Decimal to Octal Conversion				Result
Decimal Number is : **$(12345)_{10}$**				
8	12345	1	LSB	Octal Number is **$(30071)_8$**
8	1543	7		
8	192	0		
8	24	0		
	3	3	MSB	

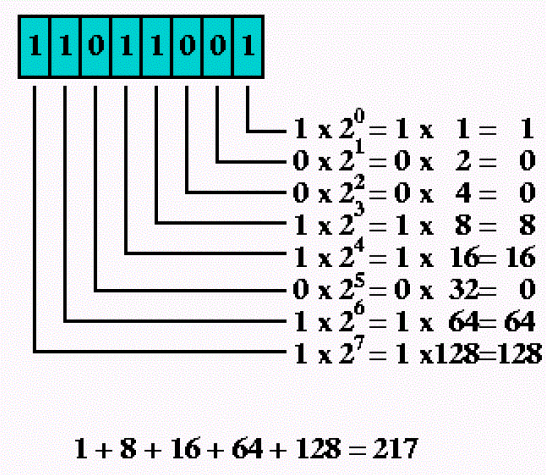
3. DECIMAL TO HEXADECIMAL

Decimal to Hexadecimal Conversion	Result
Example 1 Decimal Number is : **$(12345)_{10}$** 16 | 12345 → 9 LSB 16 | 771 → 3 16 | 48 → 0 3 → 3 MSB	Hexadecimal Number is **$(3039)_{16}$**
Example 2 Decimal Number is : **$(725)_{10}$** 16 | 725 → 5 LSB 16 | 45 → 13 (D) 2 → 2 MSB	Hexadecimal Number is **$(2D5)_{16}$** **Convert** **10, 11, 12, 13, 14, 15** **to its equivalent...** **A, B, C, D, E, F**

BINARY TO OTHER

Multiply the digit with 2(with place value exponent). Eventually add all the multiplication becomes the Decimal number.

1. BINARY TO DECIMAL



2. BINARY TO OCTAL

An easy way to convert from binary to octal is to group binary digits into sets of three, starting with the least significant (rightmost) digits.

Binary: 11100101 =	11 100 101	
	011 100 101	Pad the most significant digits with zeros if necessary to complete a group of three.

Then, look up each group in a table:

Binary:	000	001	010	011	100	101	110	111

Octal:	0	1	2	3	4	5	6	7

Binary =	011	100	101	
Octal =	3	4	5	= 345 oct

3. BINARY TO HEXADECIMAL

An equally easy way to convert from binary to hexadecimal is to group binary digits into sets of four, starting with the least significant (rightmost) digits.

Binary: 11100101 = 1110 0101

Then, look up each group in a table:

Binary:	0000	0001	0010	0011	0100	0101	0110	0111
Hexadecimal:	0	1	2	3	4	5	6	7

Binary:	1000	1001	1010	1011	1100	1101	1110	1111
Hexadecimal:	8	9	A	B	C	D	E	F

Binary =	1110	0101	
Hexadecimal =	E	5	= E5 hex

OCTAL TO OTHER

1. OCTAL TO BINARY

Converting from octal to binary is as easy as converting from binary to octal. Simply look up each octal digit to obtain the equivalent group of three binary digits.

Octal:	0	1	2	3	4	5	6	7
Binary:	000	001	010	011	100	101	110	111

Octal =	3	4	5	
Binary =	011	100	101	= 011100101 binary

2. OCTAL TO HEXADECIMAL

When converting from octal to hexadecimal, it is often easier to first convert the octal number into binary and then from binary into hexadecimal. For example, to convert 345 octal into hex:

(from the previous example)

Octal =	3	4	5	
Binary =	011	100	101	= 011100101 binary

Drop any leading zeros or pad with leading zeros to get groups of four binary digits (bits):
Binary 011100101 = 1110 0101

Then, look up the groups in a table to convert to hexadecimal digits.

Binary:	0000	0001	0010	0011	0100	0101	0110	0111
Hexadecimal:	0	1	2	3	4	5	6	7

Binary:	1000	1001	1010	1011	1100	1101	1110	1111
Hexadecimal:	8	9	A	B	C	D	E	F

Binary =	1110	0101	
Hexadecimal =	E	5	= E5 hex

Therefore, through a two-step conversion process, octal 345 equals binary 011100101 equals hexadecimal E5.

3. OCTAL TO DECIMAL

The conversion can also be performed in the conventional mathematical way, by showing each digit place as an increasing power of 8.

345 octal = $(3 * 8^2) + (4 * 8^1) + (5 * 8^0)$ = (3 * 64) + (4 * 8) + (5 * 1) = 229 decimal

HEXADECIMAL TO OTHER

1. HEXADECIMAL TO BINARY

Converting from hexadecimal to binary is as easy as converting from binary to hexadecimal. Simply look up each hexadecimal digit to obtain the equivalent group of four binary digits.

Hexadecimal:	0	1	2	3	4	5	6	7
Binary:	0000	0001	0010	0011	0100	0101	0110	0111
Hexadecimal:	8	9	A	B	C	D	E	F
Binary:	1000	1001	1010	1011	1100	1101	1110	1111

Hexadecimal =	A	2	D	E	
Binary =	1010	0010	1101	1110	= 1010001011011110 binary

2. HEXADECIMAL TO OCTAL

To Convert a hexadecimal number to octal octal number, we need to first convert the hexadecimal number into binary number and then from binary to Octal. Simply look up each hexadecimal digit to obtain the equivalent group of four binary digits.

Hexadecimal:	0	1	2	3	4	5	6	7
Binary:	0000	0001	0010	0011	0100	0101	0110	0111
Hexadecimal:	8	9	A	B	C	D	E	F
Binary:	1000	1001	1010	1011	1100	1101	1110	1111

Hexadecimal =	A	2	D	E			
Binary =	1010	0010	1101	1110	= 1010001011011110 binary		
Re-grouping	001	010	001	011	011	110	
Octal	1	2	1	3	3	6	=121336 Octal

Binary Arithmetic

Rules of Binary Addition

- 0 + 0 = 0
- 0 + 1 = 1
- 1 + 0 = 1
- 1 + 1 = 0, and carry 1 to the next more significant bit

For example,

00011010 + 00001100 = 00100110

			1	*1*					*Carries*
	0	0	0	1	1	0	1	0	= $26_{(\text{base } 10)}$
+	0	0	0	0	1	1	0	0	= $12_{(\text{base } 10)}$
	0	0	1	0	0	1	1	0	= $38_{(\text{base } 10)}$

00010011 + 00111110 = 01010001

		1	*1*	*1*	*1*	*1*			*carries*
	0	0	0	1	0	0	1	1	= $19_{(\text{base } 10)}$
+	0	0	1	1	1	1	1	0	= $62_{(\text{base } 10)}$
	0	1	0	1	0	0	0	1	= $81_{(\text{base } 10)}$

Rules of Binary Multiplication

- 0 x 0 = 0
- 0 x 1 = 0
- 1 x 0 = 0
- 1 x 1 = 1, and no carry or borrow bits

For example,

00101001 × 00000110 = 11110110

		0	0	1	0	1	0	0	1 = $41_{(\text{base } 10)}$
	×	0	0	0	0	0	1	1	0 = $6_{(\text{base } 10)}$
		0	0	0	0	0	0	0	0
	0	0	1	0	1	0	0	1	
0	0	1	0	1	0	0	1		
0	0	1	1	1	1	0	1	1	0 = $246_{(\text{base } 10)}$

Binary Division

Binary division is the repeated process of subtraction, just as in decimal division.

For example,

00101010 ÷ 00000110 = 1 1 1 = $7_{(\text{base } 10)}$

```
00000111
                 ______________________
1  1  0 )  0  0  1  10  1  0  1  0   =  42(base 10)
               -     1  1  0         =   6(base 10)
               ___________

                     1                  borrows
                  1  0  10  1
               -     1  1   0
                  ___________

                        1  1  0
                  -     1  1  0
                     __________

                              0
```

```
10000111 ÷ 00000101 =
00011011
                       1  1  0  1  1   =  27(base 10)
           ________________________
1  0  1 )  1  0  0  10  0  1  1  1     =  135(base 10)
        -  1  0  1                     =  5(base 10)
        ___________

              1  1  10
           -  1  0  1
           ___________

                 1  1
           -        0
           _________

                 1  1  1
           -     1  0  1
           ___________

                    1  0  1
              -     1  0  1
              ____________

                          0
```

Example − Division

101010 / 000110 = 000111

```
            1 1 1        = 7₁₀
         ___________
000110 ) 1 10 1 0 1 0    = 42₁₀
         - 1 1 0         = 6₁₀
         _______
          1 10 0 1
          - 1 1 0
          ________
             1 1 0
            - 1 1 0
            _______
                 0
```

Hexadecimal Addition

Following hexadecimal addition table will help you greatly to handle Hexadecimal addition.

+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10
2	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11
3	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12
4	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13
5	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14
6	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15
7	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16
8	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17
9	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18
A	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19
B	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A
C	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B
D	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C
E	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
F	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E

X

Sum

Y

To use this table, simply follow the directions used in this example – Add A_{16} and 5_{16}. Locate A in the X column then locate the 5 in the Y column. The point in 'sum' area where these two columns intersect is the sum of two numbers.

$A_{16} + 5_{16} = F_{16}$

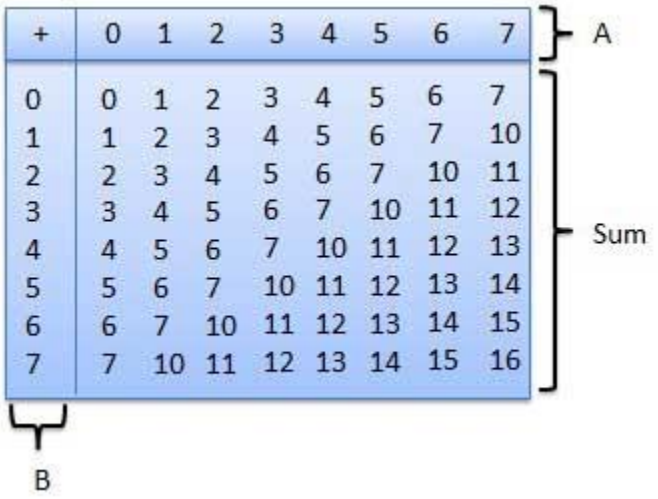
Example

$4A6_{16} + 1B3_{16} = 659_{16}$

1	carry
4 A 6	$= 1190_{10}$
+ 1 B 3	$= 435_{10}$
6 5 9	$= 1625_{10}$

Octal Addition

Following octal addition table will help you to handle octal addition.



To use this table, simply follow the directions used in this example: Add 6_8 and 5_8. Locate 6 in the A column then locate the 5 in the B column. The point in 'sum' area where these two columns intersect is the 'sum' of two numbers.

$6_8 + 5_8 = 13_8$.

Example

$456_8 + 123_8 = 601_8$

1 1	carry
4 5 6	$= 302_{10}$
+1 2 3	$= 83_{10}$
6 0 1	$= 385_{10}$

Signed Numbers

We can make the binary numbers into the following two groups − **Unsigned numbers** and **Signed numbers**.

Unsigned Numbers

Unsigned numbers contain only magnitude of the number. They don't have any sign. That means all unsigned binary numbers are positive. As in decimal number system, the placing of positive sign in front of the number is optional for representing positive numbers. Therefore, all positive numbers including zero can be treated as unsigned numbers if positive sign is not assigned in front of the number.

Signed Numbers

Signed numbers contain both sign and magnitude of the number. Generally, the sign is placed in front of number. So, we have to consider the positive sign for positive numbers and negative sign for negative numbers. Therefore, all numbers can be treated as signed numbers if the corresponding sign is assigned in front of the number.

If sign bit is zero, which indicates the binary number is positive. Similarly, if sign bit is one, which indicates the binary number is negative.

Representation of Un-Signed Binary Numbers

The bits present in the un-signed binary number holds the **magnitude** of a number. That means, if the un-signed binary number contains **'N'** bits, then all **N** bits represent the magnitude of the number, since it doesn't have any sign bit.

Example

Consider the **decimal number 108**. The binary equivalent of this number is **1101100**. This is the representation of unsigned binary number.

$$108108_{10} = 11011001101100_2$$

It is having 7 bits. These 7 bits represent the magnitude of the number 108.

Representation of Signed Binary Numbers

The Most Significant Bit MSBMSB of signed binary numbers is used to indicate the sign of the numbers. Hence, it is also called as **sign bit**. The positive sign is represented by placing '0' in the sign bit. Similarly, the negative sign is represented by placing '1' in the sign bit.
If the signed binary number contains 'N' bits, then N−1N−1 bits only represent the magnitude of the number since one bit MSBMSB is reserved for representing sign of the number.

There are three **types of representations** for signed binary numbers

- Sign-Magnitude form
- 1's complement form
- 2's complement form

Representation of a positive number in all these 3 forms is same. But, only the representation of negative number will differ in each form.

Example

Consider the **positive decimal number +108**. The binary equivalent of magnitude of this number is 1101100. These 7 bits represent the magnitude of the number 108. Since it is positive number, consider the sign bit as zero, which is placed on left most side of magnitude.

$$+108+108_{10} = 0110110001101100_2$$

Therefore, the **signed binary representation** of positive decimal number +108 is **01101100**. So, the same representation is valid in sign-magnitude form, 1's complement form and 2's complement form for positive decimal number +108.

Sign-Magnitude form

In sign-magnitude form, the MSB is used for representing **sign** of the number and the remaining bits represent the **magnitude** of the number. So, just include sign bit at the left most side of unsigned binary number. This representation is similar to the signed decimal numbers representation.

Example

Consider the **negative decimal number -108**. The magnitude of this number is 108. We know the unsigned binary representation of 108 is 1101100. It is having 7 bits. All these bits represent the magnitude.

Since the given number is negative, consider the sign bit as one, which is placed on left most side of magnitude.

$$-108-108_{10} = 1110110011101100_2$$

Therefore, the sign-magnitude representation of -108 is **11101100**.

1's complement form

The 1's complement of a number is obtained by **complementing all the bits** of signed binary number. So, 1's complement of positive number gives a negative number. Similarly, 1's complement of negative number gives a positive number.

That means, if you perform two times 1's complement of a binary number including sign bit, then you will get the original signed binary number.

Example

Consider the **negative decimal number -108**. The magnitude of this number is 108. We know the signed binary representation of 108 is 01101100.

It is having 8 bits. The MSB of this number is zero, which indicates positive number. Complement of zero is one and vice-versa. So, replace zeros by ones and ones by zeros in order to get the negative number.

$$-108-108_{10} = 1001001110010011_2$$

Therefore, the **1's complement of** 108108_{10} is 1001001110010011_2.

2's complement form

The 2's complement of a binary number is obtained by **adding one to the 1's complement** of signed binary number. So, 2's complement of positive number gives a negative number. Similarly, 2's complement of negative number gives a positive number.

That means, if you perform two times 2's complement of a binary number including sign bit, then you will get the original signed binary number.

Example

Consider the **negative decimal number -108**.

We know the 1's complement of $(\mathbf{108})_{10}$ is $(\mathbf{10010011})_2$

2's compliment of 108108_{10} *= 1's compliment of* 108108_{10} *+ 1.*

= 10010011 + 1

= 10010100

Therefore, the **2's complement of** 108108_{10} is 1001010010010100_{2}.

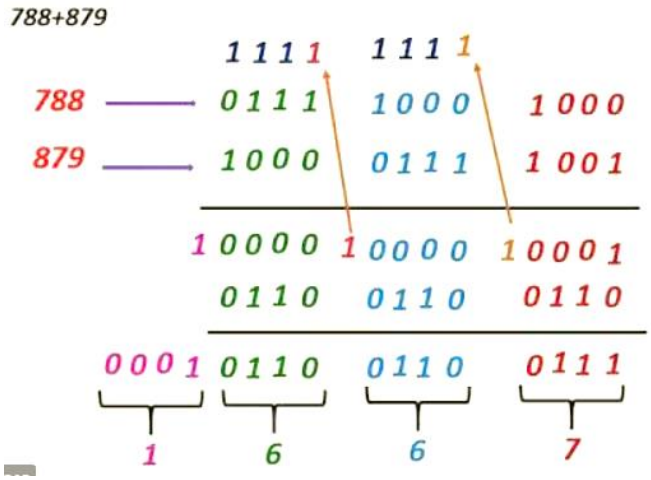
Binary Coded Decimal (BCD) Numbers

Binary-Coded Decimal Code BCD is a class of binary encodings of decimal numbers where each decimal digit is represented by a fixed number of bits, four bits. For example, decimal 396 is represented in BCD with 12 bits as 0011 1001 0110. A decimal number in BCD is the same as its equivalent binary number only when the number is between 0 and 9. A BCD number greater than 10 looks different from its equivalent binary number, even though both contain 1's and 0's. Moreover, the binary combinations 1010 through 1111 are not used and have no meaning in BCD.

4-bit Binary	Hexadecimal	BCD	Decimal Equivalent
0000	0	0000	0
0001	1	0001	1
0010	2	0010	2
0011	3	0011	3
0100	4	0100	4
0101	5	0101	5
0110	6	0110	6
0111	7	0111	7
1000	8	1000	8
1001	9	1001	9
1010	A		10
1011	B		11
1100	C		12
1101	D		13
1110	E		14
1111	F		15

BCD adders

A BCD number is added like a 4-bit binary number for each corresponding digit. Since BCD has 4 bits with the largest number being 9; and the largest 4-bit binary number is equivalent to 15, there is a difference of 6 between the binary and the BCD adder. If a sum exceeds 9, 6 is further added to it and carry is propagated to the next digit as shown below.



Lab Exercise

Task-1

Convert the basis of the following numbers. Use direct method where applicable.

a) (10101010)2 = ( )10
b) (11100011)2 = ( )8
c) (00111011)2 = ( )16
d) (3671)8 = ( )10
e) (7452)8 = ( )16
f) (1B32A)16 = ( )10
g) (7E4F2)16 = ( )2
h) (854125)10 = ( )2
i) (71342)10 = ( )8

Task-2

Perform the following arithmetic operations on numbers.

a) $(10101010)_2 + (10011011)_2$
b) $(11101001)_2 \times (101)_2$
c) $(11100011)_2 / (11)_2$
d) $(BCE)_{16} + (241)_{16}$
e) $(437)_8 + (151)_8$

Task-3

Sum the following numbers using binary number system. (Represent numbers in 2's complement form.)

a) 75 + (-35)
b) 100 + (-15)
c) (-24) + (-40)

Task-4

Add the following BCD numbers. (Show intermediate working.)

a) $(198)_{BCD} + (433)_{BCD}$
b) $(291)_{BCD} + (474)_{BCD}$