Paper 7, TDC Part-3
Chapter– 3, Number Systems and Codes
Electronics
Lecture - 7

By:

Mayank Mausam

Assistant Professor (Guest Faculty)

Department of Electronics

L.S. College, BRA Bihar University,

Muzaffarpur, Bihar

Number Systems and Codes

Hexadecimal Arithmetic: -

Arithmetic operations process with hexadecimal numbers are similar to the process of arithmetic operations with binary or octal or decimal systems.

In a digital circuit it is easier to enter the information using hexadecimal system, while these information are handled in the form of binary system. So the arithmetic operations are performed by the digital circuits on the hexadecimal number, first by converting the hexadecimal numbers to binary numbers. Hexadecimal numbers are converted to their equivalent binary numbers using hexadecimal to binary converter.

Hexadecimal Addition: -

Directly with hexadecimal numbers –

Following rules are used when adding two hexadecimal numbers: -

a) If the addition of two hex digits is F_{16} (15_{10}) or less, write down the corresponding hexadecimal digit.

b) If the addition of two hex digits is greater than F_{16} (15_{10}), bring down the amount of the sum that exceeds 16_{10} and a carry to the next column.

Number Systems and Codes

Example:- Add the following hex numbers :->

(a) $(30)_{16} + (41)_{16}$ (b) $(78)_{16} + (47)_{16}$

(c) $(7B)_{16} + (22)_{16}$ (d) $(89)_{16} + (78)_{16}$

(e) $(8A)_{16} + (97)_{16}$ (f) $(AC)_{16} + (36)_{16}$

(g) $(FF)_{16} + (CC)_{16}$ (h) $(D6)_{16} + (79)_{16}$

Soln:-> (a)

$$\begin{array}{r} (30)_{16} \\ +\ (41)_{16} \\ \hline (75)_{16} \end{array}$$

(b)

$$\begin{array}{r} 78_{16} \\ +\ 47_{16} \\ \hline BF_{16} \end{array}$$

$8 + 7 = 15_{10}$
$= F_{16}$

$7 + 4 = 11_{10}$
$= B_{16}$

Number Systems and Codes

(c)

$$\begin{array}{r} 7B_{16} \\ +\ 22_{16} \\ \hline 9D_{16} \end{array}$$

$B_{16} \Rightarrow 11_{10} + 2_{10} = 13_{10} = D_{16}$

(d)

$$\begin{array}{r} 1 \leftarrow \text{Carry} \\ 89_{16} \\ +\ 78_{16} \\ \hline 101_{16} \end{array}$$

$9 + 8 = 17_{10} - 16_{10} = 1_{10}$

$= 1_{16}$ with a carry 1,

$8 + 7 + 1(\text{Carry}) = 16_{10} - 16_{10} = 0_{10}$

$= 0_{16}$ with a carry 1

$(89)_{16} + (78)_{16} = (101)_{16}$

Number Systems and Codes

(e)

$$\begin{array}{r} {}^{1} \\ 8A_{16} \\ +\ 97_{16} \\ \hline 121_{16} \end{array}$$

$A_{16} \rightarrow 10_{10} + 7_{10} = 17_{10} - 16_{10} = 1_{10}$

$= 1_{16}$ with a carry 1

$8 + 9 + 1(\text{carry}) = 18_{10} - 16_{10} = 2_{10}$

$= 2_{16}$ with a carry 1,

(f)

$$\begin{array}{r} {}^{1} \\ AC_{16} \\ +\ 36_{16} \\ \hline E2_{16} \end{array}$$

$C_{16} \rightarrow 12_{10} + 6_{10} = 18_{10} - 16_{10} = 2_{10}$

$= 2_{16}$ with a carry 1.

$A \rightarrow 10_{10} + 3_{10} + 1_{10}(\text{carry}) = 14_{10} = E_{16}$

Number Systems and Codes

(g)
$$\begin{array}{r} {}^{1}\quad \\ FF_{16} \\ +\,CC_{16} \\ \hline CB_{16} \end{array}$$

$F_{16} \to 15_{10}$ & $C_{16} \to 12_{10}$

$15_{10} + 12_{10} = 27_{10} - 16_{10} = 11_{10}$

$= B_{16}$ with a carry 1.

Similarly $F_{16} + C_{16} + 1_{16} = C_{16}$ with a carry 1.

$F_{16} \to 15_{10}$ & $C_{16} \to 12_{10}$, $1_{16} = 1_{10}$

$15_{10} + 12_{10} + 1_{10} = 28_{10} - 16_{10} = 12_{10} =$

$= C_{16}$ with a carry 1

Number Systems and Codes

(h)
$$\begin{array}{r} {}^{1}\ \ \\ D6_{16} \\ +\ 79_{16} \\ \hline (14F)_{16} \end{array}$$

$6_{10} + 9_{10} = 15_{10} \rightarrow F_{16}$

$D_{16} \rightarrow 13_{10} + 7_{10} = 20_{10} - 16_{10}$

$= 4_{10}$ with a carry 1

However

Addition of hexadecimal numbers in digital circuit or system will be performed using 2' complement complement method by ~~taking~~ converting the hex number to its equivalent 2's complement representation. At the end providing the result in hex number system.

Number Systems and Codes

Example Add $(D6)_{16}$ & $(79)_{16}$ using 2's complement method.

$$
\begin{array}{lcl}
 & & \overset{1\,1\,1}{} \\
(D6)_{16} & \rightarrow & 1101\ 0110 \\
+(79)_{16} & \rightarrow & 0111\ 1001 \\
\hline
(14F)_{16} & \rightarrow & \underbrace{1}_{1}\underbrace{0100}_{4}\ \underbrace{1111}_{F}
\end{array}
$$

$1 \quad 4 \quad F \rightarrow (14F)_{16}$

Subtraction in hexadecimal system

Subtraction in hexadecimal system is just like binary ~~or~~ decimal system.

When we borrow '1' from higher position

Number Systems and Codes

number then in hex system $(16)_{10}$ has been borrowed by smaller place number. Let's see some example.

Example :- Subtract (a) 79 from D6
(b) D6 from 79

Soln. (a)

$$\begin{array}{r} D6 \\ -\ 79 \\ \hline (5D)_{16} \end{array}$$

Since 6 is smaller than 9 so 1 will be borrowed from D that means 6 will become now $(6+16)_{10} = (22)_{10} - (9)_{10}$

$= (13)_{10} \rightarrow (D)_{16}$

Number Systems and Codes

$= (13)_{10} \rightarrow (D)_{16}$

Now as 1 is borrowed from D so D is now one less i.e. $(D-1)_{16} = (C)_{16}$

and $(C-7)_{16} = (12-7)_{10} = (5)_{10} = (5)_{16}$

(5)

$$\begin{array}{r} 7\,9 \\ -\,D\,6 \\ \hline -(5D)_{16} \end{array}$$

Process will be ~~done~~ same just like subtraction done in decimal system

Subtraction of Hexadecimal numbers using 2's complement :→

In 2's complement method we ~~conv~~ write hexadecimal numbers in 2's complement ~~repr~~

representation.

Example → ~~Subtract~~ (a) 79 from D6
(b) D6 from 79

Soln :→ (a) $(D6)_{16}$
$- (79)_{16}$
$(5D)_{16}$

2's complement of D6 is →
1101 0110
~~0011~~

2's complement repⁿ of 79 is →
0111 1001

So, 2's complement repⁿ. of −79 is →
1000 0111

Now, for ~~addⁿ~~ subtraction using 2's complement is done by adding 2's complement representation

Number Systems and Codes

is done by adding 2's complement representation

$$
\begin{array}{lcr}
(D6)_{16} & \xrightarrow{2's} & 1101\ 0110 \\
-(79)_{16} & \xrightarrow{2's} & +\ 1000\ 0111 \\
\hline
(5D)_{16} & & (1)\,\underbrace{0101}_{5}\ \underbrace{1101}_{D}
\end{array}
$$

Carry → Discard

So we can see that the result is true

Soln. (b)

$$
\begin{array}{lcrl}
79_{16} & \xrightarrow{2's} & 0111\ 1001 & \\
-\ D6_{16} & \xrightarrow{2's} & +\ 0010\ 1010 & -\ (2\text{'s of } -D6 \\
\hline
-\ 5D_{16} & & 1010\ 0011 &
\end{array}
$$

2's complement of result.

2's complement of 1010 0011 = 0101 1101

$\underbrace{0101}_{5}\ \underbrace{1101}_{D}$

that is 1010 0011 is (−5D) in 2's complement.

Addition and subtraction of hex numbers can be done by 16's complement method too.

Binary Codes: -

Data in digital systems are processed in the binary format. There are various binary codes used to represent data. Different codes have been designed for various purpose. Some codes are used for error detection and error correction.

The same binary number represents different value in different codes, depending on the codes used.

For example the binary number 1000001 represents 65 (decimal) in straight binary, 41 (decimal) in Binary Coded Decimal (BCD) and alphabet ‘A’ in ASCII code.

Straight Binary Codes: -

This code represent numbers using straight binary form. This code is nothing but the binary number system that is discussed in previous lectures.

It is a weighted code since a weight is assigned to every position.

Example- 0, 1, 10, 11, 11101.

Various arithmetic operation in digital system is performed in this straight binary codes.

Straight Binary Codes: -

This code represent numbers using straight binary form. This code is nothing but the binary number system that is discussed in previous lectures.

It is a weighted code since a weight is assigned to every position.

Example- 0, 1, 10, 11, 11101.

Various arithmetic operation in digital system is performed in this straight binary codes.

Binary Coded Decimal (BCD): -

In this code, decimal digits 0 to 9 are represented by equivalent 4 bit binary numbers. A decimal number in BCD is written by writing 4-bit binary number for each digit appearing in the decimal number.

Example- (a) BCD representation of 5094 is 0101000010010100 where 0101 represents 5, 0000 for 0, 1001 for 9 and 0100 for 4.

(b) 721 in BCD is written as 011100100001 where 0111 represents 7, 0010 for 2 and 0001.

So in BCD more number of bits are required to represents a decimal number than straight binary code.

This code is also known as 8-4-2-1 code. 8,4,2 and 1 are the weights of the 4 bits of the binary code of each decimal digit.

So this code is also a weighted code. Arithmetic operation in this code is done in little different way.

Excess-3 Code: -

This is another form of BCD. In this code each decimal digits 0 to 9 is coded by adding decimal 3 to decimal digit and then writing 4 bits equivalent binary number.

Example: Decimal 5 in excess-3 code is written by writing binary equivalent after adding 3 to 5 i.e. binary for 8- “1000”. So in excess-3 code decimal 5 is written as “1000” .

Number Systems and Codes

Decimal Number	Straight Binary Code	BCD	Excess-3 Code
0	0000	0000	0011
1	0001	0001	0100
2	0010	0010	0101
3	0011	0011	0110
4	0100	0100	0111
5	0101	0101	1000
6	0110	0110	1001
7	0111	0111	1010
8	1000	1000	1011
9	1001	1001	1100
Table to represent Decimal digit 0 to 9 in different Codes			

Number Systems and Codes

Excess-3 is not a weighted code. It is a self complementing code, that means 1’s complement of the coded number yields 9’s complement of the number itself.

Example: Excess-3 code for decimal 6 is 1001, its 1’s complement is 0110 which is excess-3 code for decimal 3, which is 9’s complement of 6.

This self complementing property of excess-3 code helps considerably in performing subtraction operation in digital systems.

Thank You