Paper 7, TDC Part-3
Chapter– 3, Number Systems and Codes
Electronics
Lecture - 5

By:

Mayank Mausam

Assistant Professor (Guest Faculty)

Department of Electronics

L.S. College, BRA Bihar University,

Muzaffarpur, Bihar

- **<u>2'S Complement Arithmetic : -</u>**

To perform different types of arithmetic operation in digital system we need to design different digital circuits for each type of operation like addition, subtraction, multiplication and division. Designing so many digital circuits will make the system complex, costly and bulky. To get rid of this if we design such a circuit that can perform different operation by providing specific input.

So if it is possible to design such a circuit that can perform subtraction using digital circuit for addition.

Number Systems and Codes

Then the design of arithmetic circuit is cheaper and less bulky.

In 2’s complement the subtraction is performed by adding the 2’s complement representation of the subtrahend while addition of signed positive number is performed by the same method of addition.

Addition/Subtraction in 2’s Complement

Addition/Subtraction of signed binary numbers can be performed conveniently using 2’s complement representation of both the operands. So the method is used to perform addition/subtraction operation in digital systems.

of addition.

Addition/Subtraction in 2’s Complement

Addition/Subtraction of signed binary numbers can be performed conveniently using 2’s complement representation of both the operands. So the method is used to perform addition/subtraction operation in digital systems.

Binary Subtraction in 2’s Complement

Subtraction in binary system is performed in following steps-

a) Add the 2’s complement of the subtrahend to the minuend.

b) When the minuend is greater than the subtrahend, a carry is generated at the end of addition, discard the carry and the result is given by the remaining bits which is positive.

c) When the minuend is smaller than the subtrahend, no carry is generated at the end of addition, the answer is negative and is in 2’s complement.

Let’s see few example.

Number Systems and Codes

Examples on 2's complement based Subtraction, using 8-bit representation,

(i) 64 − 32

Ans. 2's complement representation of +64 =
0 1 0 0 0 0 0 0

2's complement representation of −32 =
$\cancel{0}\,\cancel{0}\,1\,0\,0\,0\,0\,0$

Now,

64 − 32 = 64 + (−32)

$$
\begin{array}{lrl}
 & & ^{1} \\
64 \rightarrow & & 0\;1\;0\;0\;0\;0\;0\;0 \\
+(-32) \rightarrow & (+) & 1\;1\;1\;0\;0\;0\;0\;0 \\
\hline
32 & & 1\;0\;0\;1\;0\;0\;0\;0\;0
\end{array}
$$

Number Systems and Codes

64 → 0 1 0 0 0 0 0 0
+(−32) → (+) 1 1 1 0 0 0 0 0

32   1 0 0 1 0 0 0 0 0

└ Carry. ⟶ Discard it.

Result is 0 0 1 0 0 0 0 0 → (+32) in 2's complement representation also

(ii) 32 − 64

Soln.. 2's complement representation of +32 =
00100000

2's complement representation of −64 =
11000000

Number Systems and Codes

Now, $32 - 64 \Rightarrow 32 + (-64)$

$$\begin{array}{r} 32 \\ +(-64) \\ \hline -32 \end{array} \rightarrow \begin{array}{r} 00100000 \\ +11000000 \\ \hline 11100000 \end{array}$$

So if it's negative

Result is $11100000 \rightarrow (-32)$ in 2's complement representation also.

(iii) $32 - (-64)$

Soln. 32 in 2's complement representation = 00100000

+64 in 2's complement = 01000000

Number Systems and Codes

$$\begin{array}{rcrcr} 32 & \Rightarrow & 32 & \rightarrow & 00100000 \\ -(-64) & \Rightarrow & +64 & \rightarrow & +01000000 \\ \hline & & 96 & & 01100000 \end{array}$$

$\hookrightarrow$ +ve

Result is 01100000 → (96) in 2's complement.

(iv) −64 −32

$$\begin{array}{rcr} -64 & \rightarrow & 11000000 \\ +(-32) & \rightarrow & +11100000 \\ \hline -96 & & 1\,10100000 \end{array}$$

$\hookrightarrow$ Discarding carry.

Result is 10100000 → (−96)

Number Systems and Codes

Note: - a) When the 2 operands are of the opposite sign, the result is to be obtained by the rule of subtraction using 2’s complement.

b) When the 2 operands are of the same sign, the sign bit of the result is to be compared with the sign bit of the operands. “If the sign bits are same”, then the result is correct and is in 2’s complement form.

“If the sign bits are not same”, the result can’t be accommodated using eight bits and the result is to be interpreted suitably. The result in this case will consist of nine bits (carry and eight bits), and the carry bit will give the sign of the number.

Octal Number System

As clear from the name, the octal number system uses eight digits/symbols- 0,1,2,3,4,5,6 and 7 to represent numbers. So the number system with eight as radix/base is Octal Number System.

It is also a positional system and may has two parts: integer and fractional, set apart by radix point.

Example:- $(1732.505)_8$, $(776.21)_8$, $(6617)_8$, $(0.207)_8$ etc

Conversion of Octal number system into other number system

- **Octal-to-Decimal:-**

Number Systems and Codes

Octal number can be converted into its equivalent decimal number using weights assigned to each octal digit position.

Example- Convert following Octal number to its equivalent decimal number.

(a) $(7761)_8$ (b) $(0.517)_8$

(c) $(102.44)_8$

Solution (a) $(7761)_8 \to 7\times 8^3 + 7\times 8^2 + 6\times 8^1 + 1\times 8^0$

$= 3584 + 448 + 48 + 1$

$= (4045)_{10}$

(b) $(0.517)_8 = 0\times 8^0 + 5\times 8^{-1} + 1\times 8^{-2} + 7\times 8^{-3}$

$= 0 + 0.625 + 0.0147 + 0.0137$

$= (0.6534)_{10}$

Number Systems and Codes

Example. Convert following Octal number to its equivalent decimal number.

(a) $(7761)_8$ (b) $(0.517)_8$

(c) $(102.44)_8$

Solution (a) $(7761)_8 \rightarrow 7 \times 8^3 + 7 \times 8^2 + 6 \times 8^1 + 1 \times 8^0$

$= 3584 + 448 + 48 + 1$

$= (4045)_{10}$

(b) $(0.517)_8 = 0 \times 8^0 + 5 \times 8^{-1} + 1 \times 8^{-2} + 7 \times 8^{-3}$

$= 0 + 0.625 + 0.0147 + 0.0137$

$= (0.6534)_{10}$

Number Systems and Codes

(b) $(0.517)_8 = 0\times8^0 + 5\times8^{-1} + 1\times8^{-2} + 7\times8^{-3}$

$= 0 + 0.625 + 0.0147 + 0.0137$

$= (0.6534)_{10}$

(c) $(102.44)_8 = 1\times8^2 + 0\times8^1 + 2\times8^0 + 4\times8^{-1} + 4\times8^{-2}$

$= 64 + 0 + 2 + 0.5 + 0.0625 = (66.5625)_{10}$

Decimal-to-Octal Conversion:-

The procedure for conversion of decimal number is same as that of binary to decimal. Here 8 is used in place of 2 for division.

Number Systems and Codes

(Q1) Convert following ~~Octal~~ decimal number to Octal number. Upto four digit after decimal

(a) $(903)_{10}$ (b) $(0.715)_{10}$
(c) $(133.27)_{10}$

Solution:

		Remainder	
8	903		
8	112	7	→ LSB
8	14	0	
8	1	6	
	0	1	→ MSB

So, $(903)_{10} = (1607)_8$

(b) $(0.715)_{10} = (\,?\,)_8$ Upto 4 digit

Number Systems and Codes

(b) $(0.715)_{10} = (?)_8$ Upto 4 digit

$0.715 \times 8 = 5.720 \rightarrow 5$

$0.72 \times 8 = 5.66 \rightarrow 5$

$0.66 \times 8 = 5.28 \rightarrow 5$

$0.28 \times 8 = 2.24 \rightarrow 2$

$(0.715)_{10} = (0.5552)_8$

(c) To find octal equivalent of $(133.27)_{10}$ we seperatly find octal equivalent of Integer part first i.e. 133 the we find the octal equivalent of fractional part i.e. '.27'.

Number Systems and Codes

			Remainder	
8	133			
8	16	→	5	→ LSB
8	2	→	0	
	0	→	2	→ MSB

Now decimal part,

$0.27 \times 8 \rightarrow 2.16 \rightarrow 2$

$0.16 \times 8 \rightarrow 1.28 \rightarrow 1$

$0.28 \times 8 \rightarrow 2.24 \rightarrow 2$

$0.24 \times 8 \rightarrow 1.92 \rightarrow 1$

So, $(133.27)_{10} = (205.2121)_8$

Number Systems and Codes

Octal-To-Binary Number :-

A octal number can be converted to it's equivalent binary number by replacing each octal digit by 3-bit equivalent binary number. Table below gives the binary equivalent of 0 - 7 octal symbol/digit.

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

Soln:- (a) $(700)_8 = (\overbrace{111}^{7}\ \overbrace{000}^{0}\ \overbrace{000}^{0})_2$

(b) $(5142)_8 = (\overbrace{101}^{5}\ \overbrace{001}^{1}\ \overbrace{100}^{4}\ \overbrace{010}^{2})_2$

Example:- Convert following Octal numbers to it equivalent binary number

(a) $(0.712)_8$ (b) $(63.02)_8$

Soln:- (a) $(0.712)_8 = \cancel{(0.00}\ (0.\overbrace{111}^{7}\ \overbrace{001}^{1}\ \overbrace{010}^{2})_2$

(b) $(63.02)_8 = (110\ 011.000\ 010)_2$

Number Systems and Codes

Let's verify the process with the help of conversion ~~to~~ into Decimal system.

Binary equivalent of $(56)_8 = (101\,110)_2$

Decimal equivalent of $(56)_8 = (5 \times 8^1 + 6 \times 8^0)_{10}$

$= (40 + 6)_{10} = (46)_{10}$

Decimal equivalent of $(101110)_2 =$

$= (1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0)_{10}$

$= (32 + 0 + 8 + 4 + 2 + 0)_{10}$

$= (46)_{10}$

~~So~~, $(56)_8 = (101\,110)_2 = (46)_{10}$

So we ~~c~~while converting Octal to binary we replace each Octal digit by equivalent 3-bit binary number.

Number Systems and Codes

Binary-to-Octal Number System :-

Binary numbers can be converted into equivalent octal number by making ~~of 3~~ ~~bits~~ groups of three bits. For integer part grouping ~~of~~ of bits is done from LSB to MSB while for fractional part, the grouping of bits are done from ~~the~~ the binary point. Then replace each group of 3 bits by it's equivalent Octal symbol/digit.

Example Convert following Binary Numbers to its equivalent Octal Numbers.

(a) 110101100 (b) 1001101

(c) 0.110111011 (d) 0.11011

(d) 11011.0101

Direction of grouping.

Solution (a) $(110\ 101\ 100)_2 = (654)_8$

MSB ↓6 ↓5 ↓4 ← LSB

(b) $(1\ 001\ 101)_2 =$

↓1 ↓5

Here after grouping we are left with one

Number Systems and Codes

bit for which there is no more bits to make group. So here we add 2 zero ('0') towards MSB to make grouping. By adding zeros the value of number will remain unchanged.,
So after two zeros,

$(\overbrace{001}^{}\ \overbrace{001}^{}\ \overbrace{101}^{})_2 = (115)_8$

$\quad\ \ 1 \qquad 1 \qquad 5$

In case of fractional part zeros are added towards right to form the group. Then again the number will remain unchanged. This can be seen in example discussed later.

→ Direction for grouping

(c) $(0.\overbrace{110}^{}\,\overbrace{111}^{}\,\overbrace{011}^{})_2 \rightarrow (0.673)_8$

(d) $(0.\overbrace{110}^{}\,11)_2$ here one more bit is

required to make group so we add ~~zero~~ required number of zero to the number to make group. So number after adding zero.

Number Systems and Codes

bit for which there is no more bits to make group. So here we add 2 zero ('0') towards MSB to make grouping. By adding zeros the value of number will remaining unchanged.,

So after two zeros,

$(001\ 001\ 101)_2 = (115)_8$

In case of fractional part zeros are added towards right to form the group. Then again the number will remain unchanged. This can be seen in example discussed later.

Direction for grouping

(c) $(0.110\ 111\ 011)_2 \rightarrow (0.673)_8$

(d) $(0.110\ 11)_2$ here one more bit is required to make group so we add ~~zero~~ required number of zero to the number to make group. So number after adding zero.

$$(0.\overbrace{110}\,\overbrace{110})_2 = (0.66)_8$$

(d) $(\overbrace{11}\,\overbrace{011}.\overbrace{0101})_2 \Rightarrow$

After adding required zero to integer part and fractional part.

$$(\overbrace{011}\,\overbrace{011}.\overbrace{010}\,\overbrace{100})_2 = (33.24)_8$$

Thank You