DIGITAL ELECTRONICS

UNIT - I

Number Systems

* Review of Number Systems:

- In general, in any number system there is an ordered set of symbols known as digits with rules defined for performing arithmetic operations like addition, multiplication, etc.
- A collection of these digits makes a number which in general has two parts - integer and fractional, set apart by a radix point (·), that is

Mixed no.

$$(N)_b = \underbrace{d_{n-1}\, d_{n-2} \cdots d_i \cdots d_1\, d_0}_{\text{Integer portion}} \underset{\substack{\uparrow \\ \text{Radix Point}}}{\cdot} \underbrace{d_{-1}\, d_{-2} \cdots d_{-f} \cdots d_{-m}}_{\text{Fractional Portion}}$$

where $N \rightsquigarrow$ a number

$b \rightsquigarrow$ radix or base of the number system

$n \rightsquigarrow$ no. of digits in integer portion

$m \rightsquigarrow$ no. of digits in fractional portion

$d_{n-1} \rightsquigarrow$ Most significant bit (MSB)

$d_{-m} \rightsquigarrow$ least significant bit (LSB)

and $0 \leq (d_i \text{ or } d_{-f}) \leq b-1$

1. Decimal Number System :

- Decimal system Contains ten unieue symbols 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9.
- Base or radix is ten.

Radix → indicates the no. of unieue digits.

- The value of a decimal number is the sum of the products of the digits of that number with their respective column weights.

EX : ① 7,392 $= 7 \times 1000 + 3 \times 100 + 9 \times 10 + 2$

$= 7 \times 10^3 + 3 \times 10^2 + 9 \times 10^1 + 2 \times 10^0$

② 9256.26 $= 9 \times 1000 + 2 \times 100 + 5 \times 10 + 6 + 2 \times 10^{-1} + 6 \times 10^{-2}$

$= 9 \times 10^3 + 2 \times 10^2 + 5 \times 10 + 6 \times 10^0 + 2 \times 10^{-1} + 6 \times 10^{-2}$

* 9's and 10's complements :

- Subtraction of decimal numbers can be accomplished by 9's and 10's complement methods.

9's complement :

↳ 9's complement of a decimal no. is obtained by subtracting each digit of that decimal no from 9.

EX : Find 9's complement of following decimal nos

(a) 3465 (b) 782.54 (c) 4526.075

Sol: (a)

$$\begin{array}{r} 9999 \\ -3465 \\ \hline 6534 \ \checkmark \\ \hline \end{array}$$

(b)

$$\begin{array}{r} 999.99 \\ -782.54 \\ \hline 217.45 \ \checkmark \\ \hline \end{array}$$

(c)

$$\begin{array}{r} 9999.999 \\ -4526.075 \\ \hline 5473.924 \ \checkmark \\ \hline \end{array}$$

10's complement:

↳ 10's complement of a decimal no. is obtained by adding a 1 to its 9's complement

Ex: Find 10's complement of the following decimal nos.

(a) 4069 (b) 1056.074

Sol: (a)

$$\begin{array}{r} 9999 \\ -4069 \\ \hline 5930 \\ +1 \\ \hline 5931 \ \checkmark \\ \hline \end{array}$$

(b)

$$\begin{array}{r} 9999.999 \\ 1056.074 \\ \hline 8943.925 \\ +1 \\ \hline 8943.926 \ \checkmark \\ \hline \end{array}$$

Scanned with CamScanner

9's complement Method of Subtraction:

Step1: Obtain 9's complement of the subtrahend.

Step2: Add ~~the~~ it to the minuend. Call this no. as intermediate result

Step3: ↳ If there is a carry, it indicates that the answer is positive. Add the carry to LSB of this result to get the answer. This is called end around carry.

Step 4: If there is no carry, it indicates that the answer is negative and the intermediate result is its 9's complement. Take 9's complement of this result & place a negative sign in front to get the answer

Ex: Subtract the following nos. using the 9's complement method.

(a) 745·81 − 436·62 (b) 436·62 − 745·81 .

Sol: (a)

$$\begin{array}{r} 745.81 \\ -\,436.62 \\ \hline 309.19 \end{array} \Rightarrow \begin{array}{rl} 745.81 & \\ +\,563.37 & (9\text{'s complement of } 436.62) \\ \hline \textcircled{1}\,309.18 & (\text{Intermediate result}) \\ \hookrightarrow +\quad 1 & (\text{End around carry}) \\ \hline 309.19 & (\text{Answer}) \end{array}$$

(b) $\begin{array}{r} 436.62 \\ -745.81 \\ \hline -309.19 \\ \hline \end{array} \Rightarrow \begin{array}{r} 436.62 \\ +254.18 \\ \hline 690.80 \end{array}$

$+254.18$ (9's complement of 745.81)

690.80 (Intermediate resut with no carry)

$\downarrow$ 9's complement

-309.19 (Answer)

10's complement Method of Subtraction:

Step 1: obtain 10's complement of the subtrahend.

Step 2: Add it to the minuend.

Step 3: If there is a carry, Ignore it The presence of the carry indicates that the answer is positive The result obtained is itself the answer

Step 4: If there is no carry, it indicates that the answer is negative & the result obtained is its 10's complement Take 10's complement of the result & place a negative sign infront to get the answer.

EX: Subtract the following numbers using the 10's complement method.

(a) 2928.54 − 416.73 (b) 416.73 − 2928.54

Sol: (a)

$$\begin{array}{r} 2928.54 \\ -0416.73 \\ \hline 2511.81 \\ \hline \end{array} \Rightarrow \begin{array}{r} 2928.54 \\ +9583.27 \\ \hline \textcircled{1}\,2511.81 \end{array}$$

(10's complement of 416.73)

(Ignore carry)

↓

Answer

(b)

$$\begin{array}{r} 0416.73 \\ -2928.54 \\ \hline -2511.81 \\ \hline \end{array} \Rightarrow \begin{array}{r} 0416.73 \\ +7071.46 \\ \hline 7488.19 \end{array}$$

(10's complement of 2928.54)

(No carry)

↓ Take 10's complement

− 2511.81 Answer

2. Binary Number system :

- A binary no. consists of a sequence of bits, each of which is either a 0 or a 1.
- Base or radix is two.
- The binary number system is used in digital computers because the switching circuits used in these computers use two-state devices such as transistors, diodes etc
- A transistor can be OFF or ON, a switch can be OPEN or CLOSED, a diode can be OFF or ON. etc. These devices have to exist in one of the two possible two states. ie; 0 and 1

→ Binary to Decimal Conversion:

Examples:

① Convert 10101_2 to decimal.

Sol: $\underset{2^4}{1}\ \underset{2^3}{0}\ \underset{2^2}{1}\ \underset{2^1}{0}\ \underset{2^0}{1} = (1\times2^4)+(0\times2^3)+(1\times2^2)+(0\times2^1)+(1\times2^0)$

$= 16+0+4+0+1$

$= 21_{10}$

② Convert 11011.101_2 to decimal

Sol: $\underset{2^4}{1}\ \underset{2^3}{1}\ \underset{2^2}{0}\ \underset{2^1}{1}\ \underset{2^0}{1}\ .\ \underset{2^{-1}}{1}\ \underset{2^{-2}}{0}\ \underset{2^{-3}}{1}$

$= (1\times2^4)+(1\times2^3)+(0\times2^2)+(1\times2^1)+(1\times2^0)+(1\times2^{-1})+(0\times2^{-2})$
$+(1\times2^{-3})$

$= 16+8+0+2+1+0.5+0+0.125$

$= 27625_{10}$

③ Convert $\underset{2^6}{1}\underset{2^5}{0}\underset{2^4}{0}\underset{2^3}{1}\underset{2^2}{0}\underset{2^1}{1}\underset{2^0}{1}{}_2$ to decimal

Sol: $(1\times2^6)+(0\times2^5)+(0\times2^4)+(1\times2^3)+(0\times2^2)+(1\times2^1)+(1\times2^0)$

$= 64+0+0+8+0+2+1$

$= 75_{10}$

→ Decimal to Binary Conversion:

Method 1: Sum-of-Weigtts

Ex: ① Convert 163_{10} to binary using sum-of-weights method.

Sol: The largest decimal no., which is a power of 2, not exceeding 163 is 128.

$$128 = 2^7 = 10000000_2$$

The remainder is $163 - 128 = 35$

→ The largest no., which is a power of 2, not exceeding 35 is 32

$$32 = 2^5 = 100000_2$$

The remainder is $35 - 32 = 3$

→ The largest no., which is a power of 2, not exceeding 3 is 2

$$2 = 2^1 = 10_2$$

The remainder is $3 - 2 = 1 \rightarrow 2^0 \quad 1_2$

$$\therefore 163_{10} = 100000000_2 + 100000_2 + 10_2 + 1_2$$

$$= 10100011_2$$

Repeat the above process on the successive remainders till you get a 0 remainder

Method 2: Double - ~~ba~~ dabble method.

(5)

Ex: Convert
① 163_{10} to binary

Dividend
↑
divisor ← $2)\ 163\ (81$ → Quotient
162
1 remainder

Successive division

		Remainder
2	163	1 LSB
2	81	1
2	40	0
2	20	0
2	10	0
2	5	1
2	2	0
2	1	1 ↑ MSB
	0	

Successive division by 2 till quotient becomes zero.

∴ Therefore

$163_{10} = (10100011)_2$

More Examples:

② Convert 163.875_{10} to binary

Method 1: Integer part $163_{10} = 10100011_2$

Fraction part 0.875_{10}

→ The largest fraction, which is a power of 2, not exceeding 0.875 is 0.5

$0.5 = 2^{-1} = 0.100_2$

The remainder is $0.875 - 0.5 = 0.375$

→ The largest fraction, which is a power of 2, not exceeding 0.375 is 0.25

$$0.25 = 2^{-2} = 0.01_2$$

The remainder is $0.375 - 0.25 = 0.125$

→ The largest fraction, which is a power of 2, not exceeding 0.125 is 0.125 itself

$$0.125 = 2^{-3} = 0.001_2$$

$$\therefore \quad 0.875_{10} = 0.100_2 + 0.01_2 + 0.001_2$$

$$= 0.111_2$$

The final result is $163.875_{10} = 10100011.111_2$

Method 2:

Integer part $163_{10} = 10100011$

Fraction part 0.875_{10}

Given fraction		0.875
Multiply 0.875 by 2	MSB ↓	1.75
" 0.75 "		1.5
" 0.5 "	LSB	1

$(0.875)_{10} = (0.111)_2$

Successive multiplicati by 2

or till the desired accuracy is obtained.

$\therefore \quad 163.875_{10} = 10100011.111_2$

Decimal fraction are successively multiplied by 2, till the fraction Part of the Product is 0

③ Convert decimal 41 to binary

Sol: Successive division

		Remainder
2	41	
2	20	1 LSB
2	10	0
2	5	0
2	2	1
2	1	0 ↑
	0	1
		MSB

$$\boxed{(41)_{10} = (101001)_2}$$

④ Convert $(0.6875)_{10}$ to binary

Sol:

Given fraction.		0.6875
Multiply 0.6875 by 2	LSB ↓	1.375
" 0.375 "		0.75
" 0.75 "		1.5
" 0.5 "	MSB	1.0

$(0.6875)_{10} = (0.1011)_2$

* Binary Addition:

The rules for binary addition are

0+0 = 0 ; 0+1 = 1 ; 1+0 = 1 ; 1+1 = 10

ie; 0 with a carry 1

EX: Add the binary nos. 1101.101 and 111.011

Sol:

```
      1101.101
 +     111.011
     1 111  11
  ------------
     10101.000
```

* Binary subtraction:

The rules for binary subtraction are

$0-0=0$; $1-1=0$; $1-0=1$; $0-1=1$, with a borrow of 1.

EX: Subtract 111.111_2 from 1010.01_2

Sol:

```
     1010.010
      111.111
      11   11
  -----------
      011.011
```

* Binary Multiplication:

The rules for binary multiplication are

$0\times0=0$; $1\times1=1$; $1\times0=0$; $0\times1=0$

3. Octal Number System:

- Octal number system was extensively used by early minicomputers.
- It is a positional weighted system.
- Its base or radix is 8.
- It has 8 independent symbols 0,1,2,3,4,5,6 and 7.
- $\because 8 = 2^3$, every 3-bit group of binary can be represented by an octal digit
- when dealing with large binary numbers of many bits, it is convenient and more efficient for us to write the numbers in octal rather than binary.

Octal to Binary Conversion:

→ Just replace each octal digit by its 3-bit binary equivalent

Ex: ① Convert 367.52_8 to binary.

Sol:

3	6	7	.	5	2
011	110	111		101	010

Result ⇒ 011110111.101010_2

Ex: ② Convert 723.143_8 to binary

Sol: 111 010 011 . 001 100 011

Result ⇒ 111010011.001100011_2

Binary to Octal Conversion:

→ For given binary number, make groups of 3 bits each. on either side of the binary point

→ Replace each 3-bit binary group by the equivalent octal digit

EX: ① Convert 110101.101010_2 to octal

Sol: 110 101 . 101 010

6 5 5 2 Result ⇒ 65.52_8

EX: ② Convert 101011111001.0111_2 to octal

Sol: 010 101 111 001 . 0118 100

2 5 7 1 . 3 4

Result ⇒ 2571.34_8

Octal to Decimal Conversion:

→ Multiply each digit in the octal no. by the weight of its position and add all the product terms.

EX: ① Convert 4057.06_8 to decimal

Sol: $\underset{8^3\,8^2\,8^1\,8^0\;8^{-1}\,8^{-2}}{4057.06_8} = 4\times8^3+0\times8^2+5\times8^1+7\times8^0+0\times8^{-1}+6\times8^{-2}$

$= 2048+0+40+7+0+0.0937$

$= 2095.0937_{10}$

Ex: ② Convert 5497_8 to decimal

Sol: $\underset{8^3\ 8^2\ 8^1\ 8^0}{5\,4\,9\,7} = 5\times8^3+4\times8^2+9\times8+7\times8^0$

$=$

Decimal to Octal Conversion:

⇝ convert integer and fraction parts separately.

⇝ To convert given decimal no.

- ↳ Divide the given no. by 8 till quotient is 0.
- ↳ last remainder is MSD.

⇝ To convert given decimal fraction,

- ↳ Successively multiply it by 8 till product is 0. or till required accuracy is obtained
- ↳ The first integer from the top is MSD.

Ex: ① Convert 378.93_{10} to octal

Sol:

Integer conversion

Successive division	Remainders
$8\,\big|\,378$	
$8\,\big|\,47$	2
$8\,\big|\,5$	7
0	↑ 5

$\therefore 378_{10} = 572_8$

Fraction Conversion: successive multiplication

0.93×8	↓	7.44
0.44×8		3.52
0.52×8		4.16
0.16×8		1.28

$\therefore 0.93_{10} = 0.7341_8$

Final result $\Rightarrow 378.93_{10} = 572.7341_8$.

EX: ② Convert 5497_{10} to binary.

Sol: Conversion of large decimals to ~~de~~ binary and large binary nos to decimal can be conveniently and Quickly performed via octal

$5497_{10} \longrightarrow$ OCTAL $\longrightarrow$ BINARY

Successive Division		Remainders
8	5497	1
8	687	7
8	85	5
8	10	2
8	1	↑ 1
	0	

$\therefore 5497_{10} = 12571_8 = 001010101111001_2$

001 010 101 111 001

Ex: ③ Convert 101111010001_2 to decimal.

Sol:

Binary → OCTAL → DECIMAL

101 111 010 001 $\quad 5721_8$

5 7 2 1 $\quad \downarrow$

$$8 \cdot \; 5\times8^3+7\times8^2+2\times8^1+1\times8^0$$

$$= 2560+448+16+1$$

$$= 3025_{10}$$

4. Hexadecimal Number System:

- It is a positional weighted system.
- It has 16 unique symbols 0,1,2,3,4,5,6,7,8,9, 10, A,B,C,D,E and F
- Base or Radix is 16.
- A 4-bit group is called a nibble.
- Since computer words come in 8 bits, 16 bits, 32 bits and so on, ie; multiples of 4 bits, they can be easily represented in hexadecimal
- This no. system is particularly useful for human communications with computers.
- This is most commonly used no. system in computer literature. It is used in large and small computers

Hexadecimal to Binary Conversion:

→ ~~To~~ Replace each hex digit by its 4-bit binary group.

EX: ① Convert $4BAC_{16}$ to binary

Sol:

4 B A C

0100 1011 1010 1100

Result ⇒ 0100101110101100_2

EX: ② Convert $3A9E.B0D_{16}$ to binary.

Sol:

0011 1010 1001 1110 . 1011 0000 1101

Result ⇒ $0011101010011110.101100001101_2$

Binary to Hexadecimal Conversion:

→ Make groups of 4 bits each on either side of the binary point and replace each 4-bit group by the equivalent hexadecimal digit

EX: ① Convert 1011011011_2 to hexadecimal

Sol:

10 1101 1011

↓ ↓ ↓

2 D B

Result ⇒ $2DB_{16}$

EX: ② Convert 01011111011.011111_2 to hexadecimal

Sol:

$$\begin{matrix} 0010 & 1111 & 1011 & . & 0111 & 1100 \\ \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow \\ 2 & F & B & & 7 & C \end{matrix}$$

Result ⇒ $2FB.7C_{16}$

Hexadecimal to Decimal Conversion:

→ Multiply each digit in the hex number by its position weight and add all those product terms.

EX: ① Convert $5C7_{16}$ to decimal

Sol: $\underset{16^2}{5}\ \underset{16^1}{C}\ \underset{16^0}{7}$ $= (5\times16^2) + (12\times16^1) + (7\times16^0)$

$= 1280 + 192 + 7$

$= 1479_{10}$

EX: ② Convert $A0F9.0EB_{16}$ to decimal

Sol: $\overset{16^3}{A}\overset{16^2}{0}\overset{16^1}{F}\overset{16^0}{9}.\overset{16^{-1}}{0}\overset{16^{-2}}{E}\overset{16^{-3}}{B}$

~~16 16² 16¹ 16⁰~~

$= (10\times16^3) + (0\times16^2) + (15\times16^1) + (9\times16^0) + (0\times16^{-1}) + (14\times16^{-2})$
$+ (11\times16^{-3})$

$= 41209.0572_{10}$

Decimal to Hexadecimal Conversion:

- Convert integer and fraction parts separately.
- To convert a decimal integer no.
  - Divide the given no. by 16 till quotient is 0.
  - last remainder is MSD.
- To convert a decimal fraction
  - successively multiply it by 16, till product is 0 or till required accuracy is obtained
  - The first integer from the top is MSD.

Hex dabble method

Ex. ① Convert 2598.675_{10} to hex.

Sol:

Integer part

	Successive division	Remainder Decimal	Hex
16	2598		
16	162	6	A26
16	10	2	
	0	↑ 10	

$\therefore 2598_{10} = A26_{16}$

Fraction part

Multiply by 16	Remainder Decimal	Hex
0.675×16	10.8	A
0.8×16	12.8	c
0.8×16	↓ 12.8	c
0.8×16	12.8	c

$\therefore 0.675_{10} = 0.Accc_{16}$

$\therefore$ Result is $A26.Accc_{16}$

Ex: ② Convert 49056_{10} to binary

Sol: Conversion of very large decimals to binary and Very large binary nos to decimal is very much simplified if it is done via the hex route

$$49056_{10} \rightarrow \text{HEX} \rightarrow \text{BINARY}$$

Successive Division		Remainder		
		Decimal	Hex	Binary group
16	49056	0	0	0000
16	3066	10	A	1010
16	191	15	F	1111
16	11	↑ 11	B	1010
	0			

$\therefore 49056_{10} = 1010111110100000_2$

Ex: ③ Convert 1011011101101110_2 to decimal

Sol: Binary → Hex → decimal

1011	0111	0110	1110
B	7	6	E

$$\overset{16^3}{B}\overset{16^2}{7}\overset{16^1}{6}\overset{16^0}{E}_{16}$$

↓

$$(11 \times 16^3) + (7 \times 16^2) + (6 \times 16^1)$$
$$+ (14 \times 16^0)$$
$$= 46958_{10}$$

Hexadecimal to Octal Conversion:

→ First convert hexadecimal no. to binary and then binary no. to octal HEX → BINARY → OCTAL

Ex: ① Convert $B9F.AE_{16}$ to octal

Sol: Binary : 1011 1001 1111 . 1010 1110
group of 4 bits

→ group of 3bits : 101 110 011 111 . 101 011 10

octal → 5 6 3 7 . 5 3 4

Result ⇒ 5637.534_8

Octal to Hexadecimal Conversion:

OCTAL → BINARY → HEX.

Ex: Convert 756.603_8 to hex

Sol:
Group of 3 bits } : 111 101 110 . 110 000 011

Group of 4 bits : 0001 1110 1110 . 1100 0001 1000

Hex : 1 E E . C 1 8

Result ⇒ $1EE.C18_{16}$

* Convert the following nos. with the given radix to decimal and then to binary.

(a) 4433_5 (b) 1199_{12} (c) 5654_7 (d) 1221_3

Sol: (a) $\overset{5^3}{4}\overset{5^2}{4}\overset{5^1}{3}\overset{5^0}{3}_5 = 4\times5^3+4\times5^2+3\times5^1+3\times5^0$

$= 618_{10}$

2	618	
2	309	— 0
2	154	— 1
2	77	— 0
2	38	— 1
2	19	— 0
2	9	— 1
2	4	— 1
2	2	— 0
2	1	— 0
	0	— 1 ↑

$618_{10} = \boxed{1001101010_2}$

Answer

(b) $\overset{12^3}{1}\overset{12^2}{1}\overset{12^1}{9}\overset{12^0}{9}_{12} = 1\times12^3+1\times12^2+9\times12^1+9\times12^0$

$= 1989_{10}$

If decimal no. large, it can be first converted to octal|hex and then to binary

		Decimal.	Hex
16	1989		
16	124	— 5	5
16	7	— 12 ↑	C
		— 7	7

$1989_{10} =$ ~~5C7~~ $_{16}$ $7C5_{16}$

~~0101~~ $\boxed{011111000101_2}$

Answer

(c) $\underset{7^3}{5}\underset{7^2}{6}\underset{7^1}{5}\underset{7^0}{4}_{7} = 5\times7^3+6\times7^2+5\times7^1+4\times7^0$

$= 2048_{10}$ Decimal → Hex → Binary

$$\begin{array}{r|l}
16 & 2048 \\ \hline
16 & 128 \;-\; 0 \\ \hline
16 & 8 \;-\; 0 \\ \hline
 & 0 \;-\; 8 \uparrow
\end{array}$$

$2048_{10} = 800_{16}$

$1000\;0000\;0000_2$ Answer

(d) $\underset{3^3}{1}\underset{3^2}{2}\underset{3^1}{2}\underset{3^0}{1}_{3} = 1\times3^3+2\times3^2+2\times3^1+1\times3^0$

$= 52_{10}$

$$\begin{array}{r|l}
2 & 52 \\ \hline
2 & 26 \;-0 \\ \hline
2 & 13 \;-0 \\ \hline
2 & 6 \;-1 \\ \hline
2 & 3 \;-0 \\ \hline
2 & 1 \;-0 \\ \hline
 & 0 \;-1 \uparrow
\end{array}$$

$= 110100_2$ Answer

* Given that $16_{10} = 100_b$, find the value of b.

Sol: Convert $\underset{b^2 b^1 b^0}{100}b$ to decimal

$1\times b^2+0\times b^1+0\times b^0 = 16 \Rightarrow b^2 = 16 \Rightarrow \boxed{b=4}$

* Given that $292_{10} = 1204$ in some no. system, find the base of that system.

Sol: $292_{10} = 1204_b = 1\times b^3+2\times b^2+0\times b^1+4\times b^0 = b^3+2b^2+4$

By trial & error, $\boxed{b=6}$

* Binary Arithmetic:

- Binary Addition
- Binary Subtraction
- Binary Multiplication
- Binary Division

Binary Addition:

Rules : 0+0 = 0 ; 0+1 = 1 ; 1+0 = 1 ; 1+1 = 10
↓
result 0
carry 1

1+1+1 = 11
↳ result 1
carry 1

EX : ① Add the binary nos. 1011 and 0110

$$\begin{array}{rrrrr} & 1 & 0 & 1 & 1 \\ + & 0 & 1 & 1 & 0 \\ & 1 & 1 & & \\ \hline 1 & 0 & 0 & 0 & 1 \end{array} \text{ Answer} \qquad \begin{array}{r} \longrightarrow 11 \\ \longrightarrow 6 \\ \hline 17 \end{array}$$

NOTE: Do not ignore Carry

EX: ② Add the binary nos. 1101.101 and 111.011

$$\begin{array}{r} 1101.101 \\ + \;\; .111.011 \\ 1\;\; 111\;\; 11\;\; \\ \hline 10101.000 \end{array} \text{ Answer}$$

Binary Subtraction:

Rules: 0 − 0 = 0 ; 1 − 0 = 1 ; 1 − 1 = 0 ;
0 − 1 = 1 with a borrow/

EX: ① Subtract $(0011)_2$ from $(1100)_2$

$\;\;1\;\overset{0}{\cancel{1}}\;\overset{1}{\cancel{10}}\;\overset{1}{10}\;\to\;12$

$-\;0\;0\;1\;1\;\to\;3$

$\;\;1\;0\;0\;1$ ✓ $\;\;9$

EX: ②

$\;\;\overset{0}{\cancel{1}}\;10\;1\;1\;0\;1$

$-\;1\;0\;1\;0\;0$

$\;\;0\;1\;1\;0\;0\;1$ ✓

~~③ $\;{}^{0}1\;\overset{0}{\cancel{1}}\;0\;0\;\overset{0}{\cancel{1}}\;0\;1\;1$~~

~~$-\;1\;0\;1\;0\;1\;1\;0$~~

~~$\;\;1$~~

~~$\;\;0\;1\;0\;1\;1\;0\;1\;0\;1$~~

③ $\;1\;\overset{10}{\cancel{1}}\;\overset{1}{\cancel{10}}\;10\;\overset{0}{\cancel{1}}\;10\;1\;1 \to 11001011$

$-\;1\;0\;1\;0\;1\;1\;0$

$\;\;1\;1\;1\;0\;1\;0\;1$ ✓

④ 1010.010

$\;\;\cancel{1}\;\overset{1}{10}\;\overset{10}{\cancel{1}}\;\overset{1}{10}\;.\;\overset{1}{10}\;\overset{10}{\cancel{1}}\;\overset{2}{10}$

$-\;1\;1\;1\;.\;1\;1\;1$

$\;\;0\;0\;1\;0\;.\;0\;1\;1$ ✓

⑤ $\;\overset{0}{\cancel{1}}\;\overset{1}{10}\;\overset{1}{10}\;\overset{1}{10}\;10\;1$

$-\;1\;1\;1\;1$

$\;\;0\;1\;0\;0\;1\;0$ ✓

* Complements of Numbers:

- Complements are used in digital Computers to simplify the subtraction operation and for logical manipulation.
- Simplifying operations leads to simpler, less expensive circuits to implement the operations.
- There are 2 types of Complements for each base-r system
  - ② radix Complement
  - ① diminished radix Complement

① Diminished radix complement:

For a Given a no. N with base r having n digits,

(r-1)'s complement of N is its diminished radix complement

→ It is defined as $(r^n - 1) - N$.

EX: Decimal No

$n = 4$ → diminished radix Complement is defined as $(10^4 - 1) - N$

$9999 - N$

EX: → Find diminished radix complement for (a) 546700_{10} (b) 12398

Sol: (a) 9's complement ⇒ $(10^6 - 1) - 546700$

$999999 - 546700$

$= 453299$

→ ~~Find~~ (b) 9's Complement ⇒ $(10^5-1)-12398$

$= 99999 - 12398$.

$= 87601$.

For binary nos.

$r=2$, $r-1=1$ ∴ 1's complement is diminished radix complement

↓

defined as $(2^n-1)-N$

n → no. of digits

N → given no.

EX: If $n=4$, we have $(2^4-1) = 15 = 1111_2$

Thus, 1's complement of a binary no. is obtained by subtracting binary digits from 1.

1's complement is formed by changing 1's to 0's and 0's to 1's.

EX: Find diminished radix complement for the given binary sequence

(a) 1011000 (b) 0101101

Sol: (a) $n=7$, $(2^7-1)-N$

1111111 − 1011000

= 0100111

(b) $n=7$, $(2^7-1)-N$

1111111 − 0101101

= 1010010.

Diminished Radix Complement

(r−1)'s complement

↳ For octal nos. → 7's complement

↳ subtract each octal digit from 7

↳ For Hexadecimal nos. → 15's Complement

↳ subtract each hex digit from 15.

② Radix Complement:

↳ r's complement

→ For a given n-digit no. N in base r, it is defined as → $r^n - N$ for $N \neq 0$

↘ 0 for $N = 0$

→ r's complement is defined as (r−1)'s Complement +1. ie; $[(r^n - 1) - N] + 1$

For decimal no.

↳ radix complement is 10's complement

ie, 9's complement + 1

For Binary no.

↳ radix complement is 2's complement

ie; ~~2~~ 1's Complement + 1

Subtraction with complements:

- 10's Complement
- 9's Complement
- 2's Complement
- 1's complement

1's Complement method of Subtraction:

Step 1: Obtain 1's Complement of the subtrahend.

Step 2: Add it to the minuend.

Step 3: If there is a carry, bring the carry around and add it to LSB. This is called end around carry. Look at sign bit (MSB).

~~Step 4:~~ If this is a "0", result is positive and is in true binary form.

Step 4: If MSB is a 1 (or) whether there is no carry, the result is negative and take ~~the~~ its 1's complement to (get the magnitude in binary) obtain final resut

Ex: ⑥ Subtract the following binary nos. using 1's complement method

(a) 0 0011001 — 00001110

(b) 00001110 — 00011001

Sol: (a) : Minuend → 0 0 0 1 1 0 0 1

1's comple subtrahend → 1 1 1 1 0 0 0 1

\+ 1 1 1   1

① 0 0 0 0 1 0 1 0

→ 1 (end around carry)

\+

0 0 0 0 1 0 1 1

MSB   LSB

MSB = 0

∴ Result is positive & is in pure binary

Answer: 00001011 $+11_{10}$

(b) Minuend → 0 0 0 0 1 1 1 0

Subtrahend → 1 1 1 0 0 1 1 0 (1's complement)

\+ 1 1 1

1 1 1 1 0 1 0 0 (No carry)

MSB   LSB

No carry

MSB = 1

↓ 1's complement

So result is negative

Answer: 0 0 0 0 1 0 1 1 -11_{10}

2's Complement method of Subtraction:

Step1: Obtain 2's complement of the subtrahend.

Step2: Add it to the minuend

Step3: If there is a carry, Ignore it Look at sign bit (MSB). If this is a "0", result is positive and is in true binary form

Step4: If MSB is a 1 (or) if there is no carry, the result is negative and take its 2's complement to (find its magnitude in binary) obtain final result .

EX: Given two binary nos. X = 1010100 and Y = 1000011, perform the subtraction (a) X − Y and (b) Y − X by using 2's complements.

Sol: (a)

Minuend → 1 0 1 0 1 0 0

Subtrahend → 0 1 1 1 1 0 1 (2's complement of Y)

+ 1 1 1 1

Ignore ① 0 0 1 0 0 0 1

MSB LSB

MSB = 0 → So result is positive

0010001 Answer

(b) Minuend → 1 0 0 0 0 1 1

Subtrahend → 0 1 0 1 1 0 0 (2's complement of X)

\+

1 1 0 1 1 1 1

MSB LSB

0101011

1 1 1

0101100

No carry

MSB=1 → So, result is negative

2's complement

0 0 1 0 0 0 1 Answer

EX : Subtract 00011101 from 00101110 using 2's complement

Y X

Sol:

X

Minuend → 0 0 1 0 1 1 1 0

Subtrahend → 1 1 1 1 0 0 1 1 (2's complement of Y)

\+ 1 11 111

1 0 0 1 0 0 0 0 1

Ignore MSB LSB

MSB=0 ↳ result is positive ∴ 00100001 Answer

* Signed Binary Numbers:

- Unsigned numbers represents positive integers (including zero).
- ~~Signed~~ In ordinary arithmetic, a negative no. is indicated by a minus sign & a positive no. by a plus sign.
- Because of hardware limitations, computers must represent everything with binary digits.
- There are two ways of representating
- For Unsigned nos. ⟶ left most bit is MSB of the no.

For signed nos. ⟶ leftmost bit ~~is~~ represents the sign & rest of the bits represent the no.

If sign bit is 0 ⟶ positive
1 ⟶ negative

Ex: ① +9 (Unsigned binary) ⟶ 01001

↳ signed binary ⟶ $\underline{0}1001$.

sign bit

② −9 ⟶ signed binary ⟶ $\underline{1}1001$

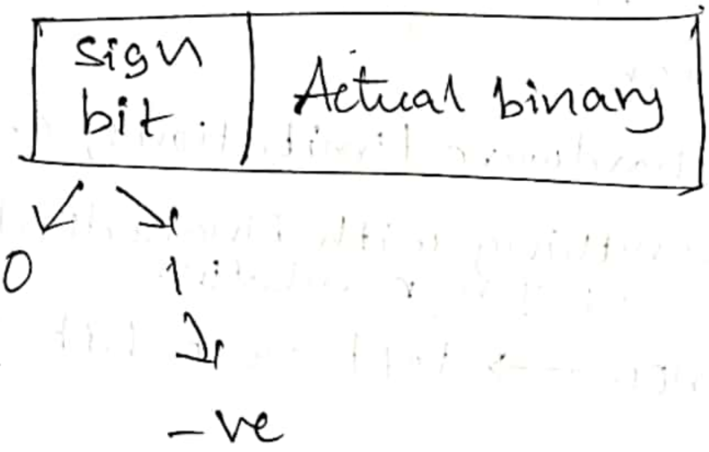
Representation of signed numbers :

① signed-magnitude form

② complement form — 1's / 2's

①

Syntax :



unsigned ⇝ 9 ⇝ 1001

signed ⇝ +9 ⇝ 01001

⇝ -9 ⇝ 11001

Ex : ① Represent -26 in signed-magnitude form

Sol: $(26)_{10} \rightarrow (\quad)_2$

↳ $(11010)_2$

2	26	
2	13	— 0
2	6	— 1
2	3	— 0
2	1	— 1
	0	— 1 ↑

Signed-magnitude representation of -26 is $\underset{\text{sign bit}}{\underline{1}}$ 11010

+26 → $\underset{\text{sign bit}}{\underline{0}}$ 11010

Ex: Represent −26 in Complement form

Sol: $(26)_{10} = (11010)_2$ SMR → −26
↳ 1 11010
sign bit

⇒ 1's Complement form of −26 } → 1 00101
sign bit

(sign bit as it is, take 1's complement of actual binary)

⇒ 2's Complement form of −26 } → 1 00110
sign bit

(sign bit as it is, obtain 2's complement of actual binary)

Ex: ② Represent +26 in all forms

Sol: • Binary of 26 → 11010

SMR of +26 → 011010

1's compl. of +26 → ~~011010~~ ~~000101~~ 011010

2's Compl. of +26 → ~~011010~~ ~~000110~~ 011010

Ex: Find 2's complement of given signed no.

1 00110.

Sol: $\underset{\text{sign bit}}{\underline{1}}$ 00110

1's Compl → 1 11001

$+$ 1

2's comple → $\underset{\text{sign bit}}{\underline{1}}$ $\underset{2^4\,2^3\,2^2\,2^1\,2^0}{11010}$ → -26 Answer

NOTE: 1's complement of (1's complement of x) = x

2's complement of (2's complement of x) = x

Ex: Express -45 in 8-bit 2's complement form

Sol: $(45)_{10} = (\,?\,)_2$

2 | 45
2 | 22 — 1
2 | 11 — 0
2 | 5 — 1
2 | 2 — 1
2 | 1 — 0
0 — 1 ↑

$(101101)_2$

SMR of $+45$ = 00101101

SMR of -45 = 10101101 (sign bit ↓ on the first 1)

1's comple. → 11010010

$+1$

2's comple → 11010011 Answer

Arithmetic addition:

For Negative nos. → Take 2's complement

Case (1): Both positive

EX: ① Add $(6)_{10}$ and $(13)_{10}$ using 8-bit.

$$\begin{array}{rcl} +6 & \longrightarrow & 00000110 \\ +13 & \longrightarrow & 00001101 \\ \hline +19 & \longrightarrow & 00010011 \\ \hline \end{array}$$

Carry bit need to be discarded

② ~~Add~~

Case (2): Smaller Negative

EX: ① Add $(13)_{10}$ and $(-6)_{10}$ using 8-bit

Sol:

$$\begin{array}{rcl} 13 & \longrightarrow & 00001101 \\ -6 & \xrightarrow{2\text{'s compl}} & 11111010 \\ \hline +7 & \longrightarrow & \boxed{1}\,00000111 \\ \end{array}$$

Ignore carry

carry

sign bit

Answer

$(+6)_{10} = 00000110$

$(-6)_{10} = 10000110 \xrightarrow{1\text{'s}} 11111001 \xrightarrow[+1]{2\text{'s}} 11111010$

② Add $(46)_{10}$ and $(-14)_{10}$ using 8-bit

Sol:

2 | 46
2 | 23 – 0
2 | 11 – 1
2 | 5 – 1
2 | 2 – 1
2 | 1 – 0
0 – 1 ↑

$(46)_{10} = (101110)_2$

~~$(-14)_{10} =$~~

$+46_{10} = 00101110$

SMR

↳ $-14_{10} = 10001110$

sign bit

$(14)_{10} = 00001110$

1's comple. → 11110001

+1

2's comple. → 11110010

+46 → 00101110

–14 → 11110010

+32 → 1) 00100000 Answer

carry ignore

sign bit

Case (3): Greater Negative

Ex: ① Add $(-13)_{10}$ and $(6)_{10}$ using 8-bit

Sol: $(13)_{10}$ → 00001101

SMR → -13_{10} → 10001101

↳ 1's comple → 11110010

↳ 2's comple → 11110011

+6 → 00000110

–13 → 11110011

–7 → 11111001

sign bit

No carry

Take 2's comple.

10000111 Answer

Case(4): Both negative

Ex: ① Add −25 to −14 using 8-bit

Sol:
⇒

2 | 25
2 | 12 − 1
2 | 6 − 0
2 | 3 − 0
2 | 1 − 1
0 − 1 ↑

$25_{10} = 11001_2$

$25_{10} \rightarrow$ 8-bit $\rightarrow$ 00011001

$+25_{10} \rightarrow$ 00011001
↙ sign bit

SMR $-25_{10} \rightarrow \underline{1}0011001$
sign bit

-25_{10} <
1's complement → 11100110
2's complement → 11100111

⇒ 14_{10} = 00001110

SMR -14_{10} = $\underline{1}$0001110
sign bit

1's complement → 11110001
2's complement → 11110010

$$
\begin{array}{rcl}
 & & 1\;1\;\;\;\;\;\;1\;1\;\;\;\; \\
-25 & \xrightarrow{2's} & 1\;1\;1\;0\;0\;1\;1\;1 \\
-14 & \xrightarrow{2's} & 1\;1\;1\;1\;0\;0\;1\;0 \\
\hline
-39 & \longrightarrow & 1\;1\;1\;0\;1\;1\;0\;0\;1
\end{array}
$$

↙ Carry ignore   sign bit

↓ 2's complement

$\boxed{1\;0\;1\;0\;0\;1\;1\;1}$ Answer

sign bit ← 1; 2^5 2^4 2^3 2^2 2^1 2^0

② Add −75 to +26 using 8-bit 2's complement arithmetic.

Sol:

2 | 26
2 | 13 − 0
2 | 6 − 1
2 | 3 − 0
2 | 1 − 1
0 − 1 ↑

$26_{10} = 11010$

$(+26)_{10}$ → 8-bit

00011010

$(75)_{10}$ → Hex → Binary

16 | 75
16 | 4 − 11 B
0 − 4 ↑ 4

4B → 01001011

$+75_{10}$ → 01001011

SMR → -75_{10} → 11001011
sign bit

1's complement → 10110100

2's complement → 10110101

$$
\begin{array}{rl}
-75 \xrightarrow{2's} & 1\,0\,1\,1\,0\,1\,0\,1 \\
+26 \rightarrow & 0\,0\,0\,1\,1\,0\,1\,0 \\
\hline
-49 \rightarrow & 1\;\underbrace{1\,0\,0\,1\,1\,1\,1}
\end{array}
$$

sign bit

↓ No carry

Take 2's complement of actual binary

1 0 1 1 0 0 0 1

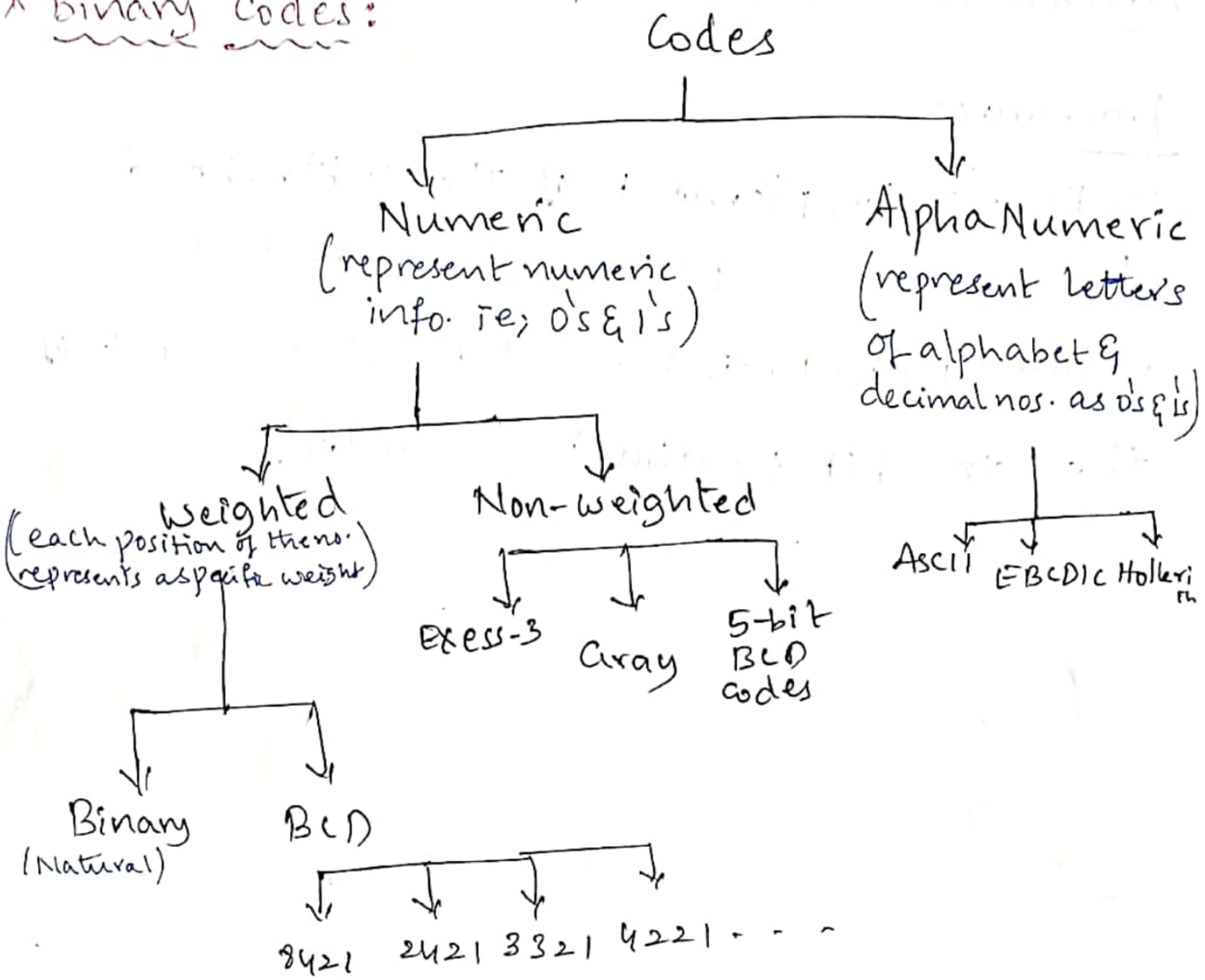
2^6 2^5 2^4 2^3 2^2 2^1 2^0 Answer

Arithmetic Subtraction:

Procedure:

Take 2's complement of the subtrahend (including sign bit) and add it to the minuend (including sign bit). A carry out of the sign-bit position is discarded.

* Binary Codes:



→ Natural Binary code:

- An n-bit binary code is a group of n bits that assumes upto 2^n distinct combinations of 1's and 0's, with each combination representing one element of the set that is being coded.

- A set of 4 elements, requires a two-bit code
- " 8 " " a 3-bit code
- " 16 " " " 4 "

→ BCD Code:

- A BCD code is one, in which the digits of a decimal no are encoded- one at a time-into groups of four binary digits.
- These Codes Combine the features of decimal and binary nos.

Code word: It is defined as the seluence of binary digits which represents a decimal digit

- BCD is a weighted code
- Widelyused BCD code is 8421.

Decimal symbol	BCD digit
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

NOTE: Binary combinations from 1010 to 1111 are not used & have no meaning in BCD.

Advantage of BCD code:

① Using BCD code, ~~conversion~~ its easy to convert to and from decimal.

② less efficient than the pure binary, in the sense that it reluires more bits.

For ex, 14 → 1110 (Pure binary)
→ 0001 0100 (BCD)

Disadvantage of BCD code:

① Arithmetic operations are more complex than they are in pure binary

② The rules of binary addition & subtraction do not apply to the entire 8421 number. but only to the individual 4-bit groups.

BCD Addition:

Step 1: Individually add the corresponding digits of the decimal nos. expressed in 4-bit binary groups starting from LSD.

Step 2: If there is a carry out of one group to the next group, it is added to the next group

(or) If the sum term is an illegal code then

add 6_{10} (0110) to the sum term of that group

Step 3: If there is no carry, and if the sum term is not an illegal code, no correction is needed.

EX: ① Perform the following decimal additions in the 8421 Code

(a) 25 + 13 (b) 679.6 + 536.8

Sol: (a)

BCD

25 ⟶ 0010 0101

+13 ⟶ 0001 0011

38 ⟶ 0011 1000 (No carry & no illegal code)

3 8

(b)

BCD

679.6 ⟶ 0110 0111 1001 . 0110

+ 536.8 ⟶ 0101 0011 0110 . 1000

1216 . 4 ⟶ 1011 1010 1111 . 1110 (No carry, but illegal 0110 codes)

\+ 0110 0110 . 0110

(Propage carry) ①0 0 0 1 ①0 0 0 0 0①0 1 0 1 · ①0 1 0 0

+1 +1 +1 +1

0001 0010 0001 0110 . 0100

1 2 1 6 . 4 ✓

(c) 4 ⟶ 0100

+5 ⟶ 0101

9 1001

(d) 4 ⟶ 0100

+8 ⟶ 1000

12 1100 (illegal code)

\+ 0110

1 0010

+1

0001 0010

1 2

(e) 8 → 1000
+9 → 1001
27   10001 ~~carry~~ (illegal code)
+0110
10110
1   7 ✓

(f) 184+576

184 → 0001 1000 0100
+576 → 0101 0111 0110
760   0110 1111 1010
0110 0110
0110 ①0101 ①0000 00
+1 ↲   +1 ↲
0111 0110 0000
7   6   0

BCD subtraction:

Step1: Subtract the digits of each 4-bit group of the subtrahend from the corresponding 4-bit group of the minuend in binary starting from the LSD.

Step2: If there is a borrow from the next group, then subtract 6_{10}(110) from the difference term of this group.

<u>Step</u> 3: If there is no borrow from the next higher group then no Correction is required

EX: perform the following decimal subtractions in the 8421 BCD code.

(a) 38 − 15 (b) 206.7 − 147.8

Sol:

(a)

38 → 0011 1000 (38 in BCD)

− 15 → 0001 0101 (15 in BCD)

23 — 0110 0011 (No borrow)

2 3 ✓

(b)

206.7 → 0 0 ~~1~~10 1010 1010 10 ~~1~~~~1~~10 · 10 1 1 1

− 147.8 → 0 0 0 1 0 1 0 0 0 1 1 1 · 1 0 0 0

58.9

9

0 0 0 0 ~~1~~0 1 1 ~~0~~1 1 0 · 1 1 1 1

− 0 1 1 0 − 0 1 1 0 ·− 0 1 1 0

0 0 0 0 0 1 0 1 1 0 0 0 · 1 0 0 1

5 8 · 9 ✓

BCD subtraction using 9's complement Method:

① Perform the following decimal subtractions in BCD by 9's Complement method.

(a) 305.5 − 168.8 (b) 679.6 − 885.9

Sol: (a)

$$\begin{array}{r} 305.5 \\ -168.8 \\ \hline 136.7 \end{array} \quad \longrightarrow \quad \begin{array}{rl} 305.5 & \\ +831.1 & (9\text{'s complement}) \\ \hline \boxed{1}\ 136.6 & \\ \hookrightarrow +1 & (\text{End around carry}) \\ \hline 136.7 & \end{array}$$

$$\begin{array}{rrrrrrl} 305.5_{10} \rightarrow & 0011 & 0000 & 0101 & \cdot & 0101 & \\ +831.1_{10} \rightarrow & +1000 & 0011 & 0001 & \cdot & 0001 & \\ & & & & & 1 & \\ \hline & 1011 & 0011 & 0110 & \cdot & 0110 & (\text{No carry, } 1011 \text{ illegal code}) \\ & +0110 & & & & & \\ & 11\ \ & & & & & \\ \hline & \boxed{1}\,0001 & 0011 & 0110 & \cdot & 0110 & \\ & \hookrightarrow & & & & +1 & \\ \hline & 0001 & 0011 & 0110 & \cdot & 0111 & \\ & 1 & 3 & 6 & \cdot & 7 & \end{array}$$

BCD Subtraction using 9's complement Method:

Step 1: Take 9's Complement of the subtrahend

Step 2: Add it to the minuend using BCD addition

Step 3: If the result is invalid BCD, then Correct by adding 6.

Step 4: Shift the carry to next bits

Step 5: If end ~~a~~round carry is generated then add it to the result

If there is no carry, it indicate the answer is -ve, & result is its 9's Complement. Take 9's complement again ~~and~~ to get actual result

BCD Subtraction using 10's complement Method:

Step 1: Take 10's complement of subtrahend.

Step 2: Add it to minuend using BCD addition

Step 3: If the result is invalid BCD, then correct it by adding 6

Step 4: shift the carry to nex bits

Step 5: If end around carry is present, discard it If not say result is -ve, & is in 10's complement form. Take 10's complement again to get actual result

(b)

$$\begin{array}{r} 679.6 \\ -885.9 \\ \hline -206.3 \end{array} \quad \begin{array}{l} \longrightarrow \\ \longrightarrow \\ \end{array} \quad \begin{array}{r} 679.6 \\ 114.0 \\ \hline 793.6 \end{array} \text{ (9's complement)}$$

$$\downarrow \text{ 9's complement}$$

$$-206.3$$

$$\begin{array}{rl} 679.6 \longrightarrow & 0110 \; 0111 \; 1001 \, . \, 0110 \\ 114.0 \longrightarrow & 0001 \; 0001 \; 0100 \, . \, 0000 \\ & \; {}^{1}{}^{1}{}^{1} \\ \hline & 0111 \; 1000 \; 1101 \, . \, 0110 \\ & +0110 \\ \hline & 0111 \; 1000 \; ①0011 \, . \, 0110 \\ & 1 \hookleftarrow \\ \hline & 0111 \; 1001 \; 0011 \, . \, 0110 \\ & 7 9 3 . 6 \end{array}$$

There is no carry & therefore result is −ve and is in its 9's compliment form. The 9's complement of 793.6 is 206.3. So the result is −206.3

BCD Subtraction using 10's Complement Method:

Ex: Perform the following decimal subtractions in BCD by 10's complement method

(a) 342.7 − 108.9 (b) 206.4 − 507.6

Sol: (a)

$$\begin{array}{r} 342.7 \\ -108.9 \\ \hline 233.8 \end{array} \quad \begin{array}{l} \rightarrow 342.7 \\ \rightarrow +891.1 \text{ (10's complement)} \\ \hline ①233.8 \text{ (Ignore carry)} \end{array}$$

$$\begin{array}{rl} 342.7 \rightarrow & 0011\ \ 0100\ \ 0010 \cdot 0111 \\ +891.1 \rightarrow & 1000\ \ 1001\ \ 0001 \cdot 0001 \\ + & \\ \hline & 1 \cdot 0111\ \ 1101 . 0011 \cdot 1000 \\ & +0110 + 0110 \\ \hline & ①0001\ ①0011\ \ 0011 \cdot 1000 \\ + & 1 \quad\quad +1 \\ \hline & ①\ 0010\ \ 0011\ \ 0011 \cdot 1000 \\ & \quad\ \ 2 \quad\quad 3 \quad\quad 3 \cdot 8 \ \checkmark \end{array}$$

Ignore carry

(b)

$$\begin{array}{r} 206.4 \\ -507.6 \\ \hline -301.2 \\ \hline \end{array} \quad \begin{array}{l} \rightarrow 206.4 \\ \rightarrow +492.4 \text{ (10's complement)} \\ \hline 698.8 \text{ (No carry)} \\ \quad \downarrow \text{10's} \\ -301.2 \ \checkmark \end{array}$$

$$\begin{array}{rl} 206.4 \rightarrow & 0010\ 0000\ 0110 . 0100 \\ +492.4 \rightarrow & 0100\ 1001\ 0010 . 0100 \\ & 11\ \ \ 1 \\ \hline & 0110\ 1001\ 1000 . 1000 \text{ (No carry, No illegal)} \\ & \cancel{+0110} \\ & \ \ 6\ \ \ \ \ \ \ 9\ \ \ \ \ \ \ 8\ \ .\ \ 8 \\ \hline & \cancel{0110\ 1111\ 1000 . 1000} \end{array}$$

There is no carry & result is -ve and is in its 10's complement form. The 10's Complement of 698.8 is 301.2 So the result is -301.2

Excess-3 code:

- Un weighted code. Also called as XS-3
- Each coded combination is obtained from the corresponding binary value plus 3.
- It's an example of self-complementing code

↓

If the codeword of 9's complement of N ie, 9-N can be obtained from the code word of N by interchanging all the 0's and 1's.

Decimal Digit	Excess-3
0	0011
1	0100
2	0101
3	0110
4	0111
5	1000
6	1001
7	1010
8	1011
9	1100

2421 Code :

- Example of ~~un~~ self-complementing code
- It is a weighted code

Decimal Digit	2421
0	0000
1	0001
2	0010
3	0011
4	0100
5	1011
6	1100
7	1101
8	1110
9	1111

Gray code :

- A/D Converter → Analog to digital
- Non-weighted code, Not suitable for arithmetic operations