Series no. 2 solution

Exercise 1

1. Ranges of 8-bit and 16-bit signed integers:

	Sign-Magnitude	**1's Complement**	**2's Complement**
n-bit	$[-(2^{(n-1)}-1), +(2^{(n-1)}-1)]$	$[-(2^{(n-1)}-1), +(2^{(n-1)}-1)]$	$[-(2^{(n-1)}), +(2^{(n-1)}-1)]$
8-bit (1 byte)	$[-(2^7-1), +(2^7-1)]$ = [-127, +127]	$[-(2^7-1), +(2^7-1)]$ = [-127, +127]	$[-(2^7), +(2^7-1)]$ = [-128, +127]
16-bit (2 bytes)	$[-(2^{15}-1), +(2^{15}-1)]$ = [-32767, +32767]	$[-(2^{15}-1), +(2^{15}-1)]$ = [-32767, +32767]	$[-(2^{15}), +(2^{15}-1)]$ = [-32768, +32767]

2.

- $(512)_{10}$ =- $(1000000000)_2$ →This value cannot be encoded using 1 byte; a minimum of 2 bytes is necessary for encoding.

- $(512)_{10}$= - $(0000001000000000)_2$ = (1000001000000000) in Signed Magnitude representation.

3.

Decimal	Sign-Magnitude	1's Complement	2's Complement
+27	00011011	00011011	00011011
-45	10101101	11010010	11010011
-128	/	/	10000000
-117	**11110101**	10001010	10001011
+117	**01110101**	01110101	01110101
-86	11010110	**10101001**	10101010
+41	00101001	**00101001**	00101001
-14	10001110	11110001	**11110010**
+114	01110010	01110010	**01110010**

- **Note:** The value (-128) cannot be encoded in Signed Magnitude and 1's Complement because it exceeds the coding ranges illustrated in the first question of the exercise; it requires more than 8 bits for proper encoding.

Exercise 2

1.

$(10000111)_{SM} = -\ (00000111)_2$

$= -\ (0000000000000111)_2$

$= (1000000000000111)_{SM}$

2.

a. When read as an unsigned integer:

$(C0)_{16} = (1100\ 0000)_2 = (192)_{10}$

b. When read as a signed integer: It is represented differently depending on whether we assume the representation in Sign/Magnitude, in 1's Complement, or in 2's Complement:

- Sign/Magnitude:

  $(C0)_{16\ SM} = (1100\ 0000)_{2\ SM} = -\ (0100\ 0000)_2 = -\ (64)_{10}$

- 1's Complement:

  $(C0)_{16\ 1C} = (1100\ 0000)_{2\ 1C} = -\ (0011\ 1111)_2 = -\ (63)_{10}$

- 2's Complement:

  $(C0)_{16\ TC} = (1100\ 0000)_{2\ TC} = -\ (0100\ 0000)_2 = -\ (64)_{10}$

3.

- $(8000)_{16\ TC} = (1000000000000000)_{2\ TC} = -\ (1000000000000000)_2 = -\ (32768)_{10}$
- $(00FF)_{16\ TC} = (0000000011111111)_{2\ TC} = +(0000000011111111)_2 = +(255)_{10}$

Exercise 3

a. **$(107)_{10} - (67)_{10} = (?)_{TC}$**

$(107)_{10} - (67)_{10} = (107)_{10} + (-67)_{10}$

$(107)_{10} = (01101011)_2 = (01101011)_{TC}$

$(-67)_{10} = -\ (01000011)_2 = (10111101)_{TC}$

$$\begin{array}{r} {}^{1}\ {}^{1}0^{1}1^{1}1^{1}0^{1}1^{1}0^{1}1\ 1 \\ +\quad 1\ 0\ 1\ 1\ 1\ 1\ 0\ 1 \\ \hline = (1|\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0)_{TC} \end{array}$$

carry

- A negative and positive added together cannot overflow because the sum is between the addends. Since both of the addends fit within the allowable range of numbers, and their sum is between them, it must fit as well →The result is correct →No Overflow.
- Adding in decimal equivalent:

  $(00101000)_{TC} = +\ (00101000)_2 = +\ (40)_{10} = (107)_{10} - (67)_{10}$

b. **$(-106)_{10} - (5)_{10} = (?)_{TC}$**

$(-106)_{10} - (5)_{10} = (-106)_{10} + (-5)_{10}$

$(-106)_{10} = -\ (01101010)_2 = (10010110)_{TC}$

$(-5)_{10} = -\ (00000101)_2 = (11111011)_{TC}$

$^{1\,1}1\,^{1}0\,^{1}0\,^{1}1\,^{1}0\,^{1}1\;1\;0$

$+\quad 1\;1\;\;1\;1\;1\;\;0\;1\;1$

$= (1\,|\,1\;0\;\;0\;1\;0\;\;0\;0\;1)_{TC}$

carry

- We added two negative numbers and obtained a negative number (the sign bit is 1) →The result is correct →No Overflow.
- Adding in decimal equivalent:

  $(10010001)_{TC} = -\,(01101111)_2 = (-111)_{10} = (-106)_{10} - (5)_{10}$

c. $\mathbf{(111)_{10} + (25)_{10} = (?)_{TC}}$

$(111)_{10} = (01101111)_2 = (01101111)_{TC}$

$(25)_{10} = (00011001)_2 = (00011001)_{TC}$

$^{1}0\,^{1}1\,^{1}1\,^{1}0\,^{1}1\,^{1}1\,^{1}1\;1$

$+\quad 0\;\;0\;0\;\;1\;1\;\;0\;0\;1$

$=\quad (1\;\;0\;0\;\;0\;1\;\;0\;0\;0)_{TC}$

- We added two positive numbers and obtained a negative number (the sign bit is 1) →The result is incorrect → Overflow.
- ➢ This is because the result (**136**) does not fall within the authorized range with the given number of bits **[-128,+127])**.
- Adding in decimal equivalent:

  $(10001000)_{TC} = -(01111000)_2 = (-120)_{10}$

  $(111)_{10} + (25)_{10} = (136)_{10}$

d. $\mathbf{(-126)_{10} - (85)_{10} = (?)_{TC}}$

$(-126)_{10} - (85)_{10} = (-126) + (-85)_{10}$

$(-126)_{10} = -\,(01111110)_2 = (10000010)_{TC}$

$(-85)_{10} = -\,(01010101)_2 = (10101011)_{TC}$

$^{1\,0}1\;0\;0\;0\;0\,^{1}0\;1\;0$

$+\quad 1\;0\;1\;0\;1\;0\;1\;1$

$= (1\,|\;0\;0\;1\;0\;1\;1\;0\;1)_{TC}$

carry

- We added two negative numbers and obtained a positive number (the sign bit is 0) →The result is incorrect → Overflow.
- ➢ This is because the result (**-211**) does not fall within the authorized range with the given number of bits **[-128,+127])**.
- Adding in decimal equivalent:

  $(-126)_{10} - (85)_{10} = -\,(211)_{10}$

  $(00101101)_{TC} = -(00101101)_2 = +(45)_{10}$

Exercise 4

1.

a. $(+45)_{10} = +(101101)_2 = +(1.01101)_2 \times 2^{+5}$

Mantissa M= $(01101)_2$

Sign S= 0 (positive)

True exponent e = 5

Biased exponent E = e + Bias= 5 + 127 = 132=$(10000100)_2$

The IEEE 754 single-precision floating-point format is:

0	1	0	0	0	0	1	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

S (1 bit) **E (8bits)** **M (23 bits)**

In binary: N = 01000010001101000000000000000000

In Hexadecimal: N = 42340000

b. $(+13.5)_{10} = +(1101.1)_2 = +(1.1011)_2 \times 2^{+3}$

Mantissa M = $(1011)_2$

Sign S = 0 (positive)

True exponent e = 3

Biased exponent E = e + Bias= 3 + 127 = 130= $(10000010)_2$

The IEEE 754 single-precision floating-point format is:

0	1	0	0	0	0	0	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

S (1 bit) **E (8bits)** **M (23 bits)**

In binary: N = 01000001010110000000000000000000

In hexadecimal: N = 41580000

c. $(-160.75)_{10} = -(10100000.11)_2 = -(1.010000011)_2 \times 2^{+7}$

Mantissa M = $(010000011)_2$

Sign S = 1 (negative)

True exponent e = 7

Biased exponent E = e + Bias = 7 + 127 = 134= $(10000110)_2$

The IEEE 754 single-precision floating-point format is:

1	1	0	0	0	0	1	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

S (1 bit) **E (8bits)** **M (23 bits)**

In binary: N = 11000011001000000100000000000000

In hexadecimal: N = C3204000

d. $(-32.625)_{10} = -(100000.101)_2 = -(1.00000101)_2 \times 2^{+5}$

Mantissa $M = (00000101)_2$

Sign S = 1 (negative)

True exponent e = 5

Biased exponent $E = e + 127 = 5 + 127 = 132 = (10000100)_2$

The IEEE 754 single-precision floating-point format is:

1	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

S (1 bit) **E (8bits)** **M (23 bits)**

In binary: N = 11000010000000101000000000000000

In hexadecimal: N = C2028000

2.

a. $(17BE0000)_{16} = (00010111101111100000000000000000)$

S (1bit) **E (8 bits)** **M (23 bits)**

Sign S = 0 (positive)

Mantissa $M = (011111)_2$

Exponent $E = (00101111)_2 = 47$

E-Bias= 47-127= -80

Normalized form: $\mathbf{(-1)^S \times (1.M)_2 \times 2^{E-Bias}}$

$(17BE0000)_{16} = +(1.011111)_2 \times 2^{-80} = +(1011111)_2 \times 2^{-6} \times 2^{-80} = (+95) \times 2^{-86}$

b. $(C3F00000)_{16} = (11000011111100000000000000000000)$

S (1 bit) **E (8bits)** **M (23 bits)**

Sign S = 1 (negative)

Mantissa $M = (111)_2$

Exponent $E = (10000111)_2 = 135$

E-Bias = 135-127 = + 8

Normalized form: $\mathbf{(-1)^S \times (1.M)_2 \times 2^{E-Bias}}$

$(C3F00000)_{16} = -(1.111)_2 \times 2^{+8} = -(1111)_2 \times 2^{-3} \times 2^{+8} = -(15)_{10} \times 2^{+5}$

Exercise 5

1.

a. $(87)_{10} = (1010111)_2 = (10000111)_{BCD} = (1111100)_{GR} = (10111010)_{XS3}$

b. $(153)_{10} = (10011001)_2 = (000101010011)_{BCD} = (101010011)_{BCD} = (11010101)_{GR} = (010010000110)_{XS3} = (10010000110)_{XS3}$

c. $(637)_8 = (415)_{10} = (110011111)_2 = (10000010101)_{BCD} = (101010000)_{GR} = (11101001000)_{XS3}$

d. $(BC8)_{16}$ = $(101111001000)_2$ = $(3016)_{10}$ = $(11000000010110)_{BCD}$ = $(111000101100)_{GR}$ = $(110001101001001)_{XS3}$

e. $(1101001)_{BCD}$ = $(69)_{10}$ = $(1000101)_2$ = $(1100111)_{GR}$ = $(10011100)_{XS3}$

f. $(100011000)_{BCD}$ = $(118)_{10}$ = $(1110110)_2$ = $(1001101)_{GR}$ = $(10001001011)_{XS3}$

g. $(1011001011)_{GR}$ = $(1101110010)_2$ = $(882)_{10}$ = $(100010000010)_{BCD}$ = $(101110110101)_{XS3}$

h. $(100010010011)_{GR}$ = $(111100011101)_2$ = $(3869)_{10}$ = $(11100001101001)_{BCD}$ = $(1101011100111100)_{XS3}$

i. $(11001010)_{XS3}$ = $(97)_{10}$ = $(1100001)_2$ = $(1010001)_{GR}$= $(10010111)_{BCD}$

j. $(110001101011)_{XS3}$ = $(938)_{10}$ = $(1110101010)_2$ = $(1001111111)_{GR}$= $(100100111000)_{BCD}$

2.

Explaining the procedure: Start by representing the number 13 in gray code, then, calculate the succeeding numbers in the sequence using the following rule for transitioning from one number to the next:

- If the count of 1s is even, reverse the last digit (the least significant digit).
- If the count of 1s is odd, reverse the digit positioned to the left of the rightmost 1.

For example:

$(13)_{10}$ = $(1101)_2$ = $(1011)_{GR}$, since the number of 1s is odd:

$(14)_{10}$ = $(1001)_{GR}$

Decimal	Gray Code
13	01011
14	01001
15	01000
16	11000
17	11001
18	11011
19	11010
20	11110
21	11111
22	11101
23	11100
24	10100
25	10101