Chap 04

P4 – 1.

a. NOT (99)16

(99)16 = 0 1 1 0 0 0 1 1

=> NOT (99)16 = 0 0 0 1 1 1 0 0

b. NOT (FF)16

(FF)16 = 1 1 1 1 1 1 1 1

=> NOT (FF)16 = 0 0 0 0 0 0 0 0

c. NOT (00)16

(00)16 = 0 0 0 0 0 0 0

=> NOT (00)16 = 1 1 1 1 1 1 1 1

d. NOT (01)16

(01)16 = 0 0 0 0 0 0 1

=> NOT (00)16 = 1 1 1 1 1 1 1 0

P4 – 2.

a. (99)16 AND (99)16

(99)16 = 0 1 1 0 0 0 1 1

(99)16 = 0 1 1 0 0 0 1 1

=> (99)16 AND (99)16 = 0 1 1 0 0 0 1 1

b. (99)16 AND (00)16

(99)16 = 0 1 1 0 0 0 1 1

(00)16 = 0 0 0 0 0 0 0 0

=> (99)16 AND (00)16 = 0 0 0 0 0 0 0 0

c. (99)16 AND (FF)16

(99)16 = 0 1 1 0 0 0 1 1

(FF)16 = 1 1 1 1 1 1 1 1

=> (99)16 AND (FF)16 = 0 1 1 0 0 0 1 1

d. (FF)16 AND (FF)16

(FF)16 = 1 1 1 1 1 1 1 1

(FF)16 = 1 1 1 1 1 1 1 1

=> (FF)16 AND (FF)16 = 1 1 1 1 1 1 1 1

P4 – 3.

a. (99)16 OR (99)16

(99)16 = 0 1 1 0 0 0 1 1

(99)16 = 0 1 1 0 0 0 1 1

=> (99)16 OR (99)16 = 0 1 1 0 0 0 1 1

b. (99)16 OR (00)16

(99)16 = 0 1 1 0 0 0 1 1

(00)16 = 0 0 0 0 0 0 0 0

=> (99)16 OR (00)16 = 0 1 1 0 0 0 1 1

c. (99)16 OR (FF)16

(99)16 = 0 1 1 0 0 0 1 1

(FF)16 = 1 1 1 1 1 1 1 1

=> (99)16 OR (FF)16 = 1 1 1 1 1 1 1 1

d. (FF)16 OR (FF)16

(FF)16 = 1 1 1 1 1 1 1 1

(FF)16 = 1 1 1 1 1 1 1 1

=> (FF)16 OR (FF)16 = 1 1 1 1 1 1 1 1

P4 – 4.

a. NOT [(99)16 OR (99)16)]

(99)16 OR (99)16 = 0 1 1 0 0 0 1 1

=> NOT [(99)16 OR (99)16)] = 1 0 0 1 1 1 0 0

b. (99)16 OR [NOT (00)16]

NOT (00)16 = 1 1 1 1 1 1 1 1

(99)16 = 0 1 1 0 0 0 1 1

=> (99)16 OR [NOT (00)16] = 1 1 1 1 1 1 1 1

c. [(99)16 AND (33)16)] OR [(00)16 AND (FF)16]

(99)16 = 0 1 1 0 0 0 1 1

(33)16 = 0 0 1 0 0 0 0 1

=> (99)16 AND (33)16 = 0 0 1 0 0 0 0 1

(00)16 AND (FF)16 = 0 0 0 0 0 0 0 0

=> [(99)16 AND (33)16)] OR [(00)16 AND (FF)16] = 0 0 1 0 0 0 0 1

d. (99)16 OR (33)16 AND [(00)16 OR (FF)16]

(00)16 = 0 0 0 0 0 0 0 0

(FF)16 = 1 1 1 1 1 1 1 1

=> [(00)16 OR (FF)16] = 1 1 1 1 1 1 1 1

(33)16 = 0 0 1 0 0 0 0 1

=> (33)16 AND [(00)16 OR (FF)16] = 0 0 1 0 0 0 0 1

(99)16 = 0 1 1 0 0 0 1 1

=> (99)16 OR (33)16 AND [(00)16 OR (FF)16] = 0 1 1 0 0 0 1 1

P4 – 5.

Bits pattern: 0 0 1 1 0 0 1 1

Mask: 0 0 0 0 1 1 1 1

=> (AND) = 0 0 0 0 0 0 1 1

P4 – 6.

Bits pattern: 0 0 1 1 0 0 1 1

Mask: 0 0 0 0 1 1 1 1

=> (OR) = 0 0 1 1 1 1 1 1

P4 – 7.

Bits pattern: 0 0 1 1 0 0 1 1

Mask: 1 1 1 0 0 0 1 1

=> (XOR) = 1 1 0 1 0 0 0 0

P4 – 8.

Bits pattern: 0 0 1 1 0 0 1 1

Mask: 0 0 0 1 1 1 0 0

=>(AND) = 0 0 0 1 0 0 0 0

P4 – 9.

Bits pattern: 00010000(the pattern is an integer in two’s complement format.)

00010000 Arithmetic Right 00001000

00001000 Arithmetic Right 00000100

P4 – 10.

Bits pattern: 0010 (the pattern is an integer in two’s complement format.)

00000010 Arithmetic Left 00000100

00000100 Arithmetic Left 00001000

00001000 Arithmetic Left 00010000

P4 – 11.

h g f e d c b a Original

0 0 h g f e d c One right shift

0 0 0 0 h g f e Two right shifts

0 0 0 0 0 0 1 1 Mask

0 0 0 0 0 0 f e Result

P4 – 12.

a. 19 + 23

19 = (00010011)2

23 = (00010111)2

1 1 1 1 Carry

0 0 0 1 0 0 1 1 19

+ 0 0 0 1 0 1 1 1 23

= 0 0 1 0 1 0 1 0 R

=> (+19) + (+23) = (+42)

b. 19 – 23

19 = (00010011)2

23 = (00010111)2

1 1 Carry

0 0 0 1 0 0 1 1 19

+ 1 1 1 0 1 0 0 1 (23 + 1)

= 1 1 1 1 1 1 0 0 R

=> (+19) + (-23) = (-4)

c. −19 + 23

19 = (00010011)2

23 = (00010111)2

1 1 1 1 1 1 1 Carry

1 1 1 0 1 1 0 1 (19 + 1)

+ 0 0 0 1 0 1 1 1 23

= 0 0 0 0 0 1 0 0 R

=> (-19) + (+23) = (+4)

d. −19 – 23

19 = (00010011)2

23 = (00010111)2

1 1 1 1 1

1 1 1 0 1 1 0 1 (19 + 1)

+ 1 1 1 0 1 0 0 1 (23 + 1)

= 1 1 0 1 0 1 1 0 R

=> (-19) + (-23) = (-46)

P4 – 13.

a. 161 + 1023

161 = (0000000010100001)2

1023 = (0000001111111111)2

1 1 1 1 1 1 1 1 1 1 Carry

0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 1 161

+ 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1023

= 0 0 0 0 0 1 0 0 1 0 1 0 0 0 0 0 R

=> (+161) + (+1023) = (+1184)

b. 161 – 1023

161 = (0000000010100001)2

1023 = (0000001111111111)2

1 Carry

0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 1 161

+ 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 (1023 + 1)

= 1 1 1 1 1 1 0 0 1 0 1 0 0 0 1 0 R

=> (+161) + (-1023) = (-862)

c. −161 + 1023

161 = (0000000010100001)2

1023 = (0000001111111111)2

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 Carry

1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 (161 + 1)

+ 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1023

= 0 0 0 0 0 0 1 1 0 1 0 1 1 1 1 0 R

=> - (+161) + (+1023) = +862

d. −161 − 1023

161 = (0000000010100001)2

1023 = (0000001111111111)2

1 1 1 1 1 1 1 1 1 1 1 Carry

1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 (161 + 1)

+ 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 (1023 + 1)

= 1 1 1 1 1 0 1 1 0 1 1 0 0 0 0 0 R

=> (-161) + (-1023) = (-1184)

P4 – 14.

a. 11000010 + 00111111

1 1 1 1 1 1 1

1 1 0 0 0 0 1 0

+ 0 0 1 1 1 1 1 1

= 0 0 0 0 0 0 0 1

=> (-62) + (+63) = (+1)

b. 00000010 + 00111111

1 1 1 1 1

0 0 0 0 0 0 1 0

+ 0 0 1 1 1 1 1 1

= 01000001

=> (+2) + (+63) = (+65)

c. 11000010 + 11111111

1 1 1 1 1 1 1

1 1 0 0 0 0 1 0

+ 1 1 1 1 1 1 1 1

= 1 1 0 0 0 0 0 1

=> (-62) + (-1) = (-63)

d. 00000010 + 11111111

1 1 1 1 1 1 1

0 0 0 0 0 0 1 0

+ 1 1 1 1 1 1 1 1

= 0 0 0 0 0 0 0 1

=> (+2) + (-1) = (+1)

P4 – 15.

a. 32 + 105

There has been an overflow

b. 32 – 105

There was no overflow

c. −32 + 105

There was no overflow

d. −32 – 105

There has been an overflow

P4 – 16.

a. (012A)16 + (0E27)16

(012A)16 = 0000000100101010

(0E27)16 = 0000111000100111

1 1 1 1

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

+ 0 0 0 0 1 1 1 0 0 0 1 0 0 1 1 1

= 0 0 0 0 1 1 1 1 0 1 0 1 0 0 0 1

=> (012A)16 + (0E27)16 = (F51)16

b. (712A)16 + (9E00)16

(712A)16 = 0111000100101010

(9E00)16 = 1001111000000000

1 1 1 1

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

+ 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0

= 0 0 0 0 1 1 1 1 0 0 1 0 1 0 1 0

=> The result is not valid because of overflow

c. (8011)16 + (0001)16

(8011)16 = 1000000000010001

(0001)16 = 0000000000000001

1

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

+ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

= 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0

=> (8011)16 + (0001)16 = (8012)16

d. (E12A)16 + (9E27)16

(E12A)16 = 1110000100101010

(9E27)16 = 1001111000100111

1 1 1 1 1

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

+ 1 0 0 1 1 1 1 0 0 0 1 0 0 1 1 1

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

=> (E12A)16 + (9E27)16 = (7F51)16

P4 – 17.

a. 19 + 23

19 = 00010011

23 = 00010111

0 0 0 1 0 0 1 1

+ 0 0 0 1 0 1 1 1

= 0 0 1 0 1 0 1 0

=> 19 + 23 = 42

b. 19 – 23

19 = 00010011

-23 = 10010111

0 0 0 1 0 0 1 1

+ 1 0 0 1 0 1 1 1

= 1 0 1 0 1 0 1 0

=> 19 – 23 = -42

c. −19 + 23

-19 = 10010011

23 = 00010111

1 0 0 1 0 0 1 1

+ 0 0 0 1 0 1 1 1

= 1 0 1 0 1 0 1 0

=> −19 + 23 = (-42)

d. −19 – 23

-19 = 10010011

-23 = 10010111

1 0 0 1 0 0 1 1

+ 1 0 0 1 0 1 1 1

= 0 0 1 0 1 0 1 0

=> −19 – 23 = (+42)

P4 – 18.

a. 34.75 + 23.125

34.75 = 1.000101100 x 2^5

23.125 = 1.0111001 x 2^4

1 . 0 0 0 1 0 1 1 0 0 x 2^5

+ 0 . 1 0 1 1 1 0 0 1 0 x 2^5

= 1 . 1 1 0 0 1 1 1 1 0 x 2^5

=> 0 10000100 11001111000000….

b. −12.625 + 451.00

−12.625 = 1.10010100x2^3

451.00 = 1.11000011x2^8

0 . 0 0 0 0 1 1 0 0 1 0 1 x 2^8

+ 1 . 1 1 0 0 0 0 1 1 0 0 0 x 2^8

= 1 . 1 1 0 0 1 1 1 1 1 0 1 x2^8

=> 0 10000111 110011111010000…

c. 33.1875 − 0.4375

33.1875 = 1.000010011x2^5

− 0.4375 = 1.1100x2^-2

1 . 0 0 0 0 1 0 0 1 1 0 0 x 2^5

+ 0 . 0 0 0 0 0 0 1 1 1 0 0 x 2^5

1 . 0 0 0 0 1 1 0 1 0 0 0 x 2^5

=> 0 10000100 0000110100…

d. −344.3125 − 123.5625

−344.3125 = 1.010110000101x2^8

−123.5625 = 1.1110111001x2^6

1 . 0 1 0 1 1 0 0 0 0 1 0 1

+ 0 . 0 1 1 1 1 0 1 1 1 0 0 1

= 1 . 1 1 0 1 0 0 1 1 1 1 1 0 x 2^8

=> 1 10000111 1101001111100000….

P4 – 19.

a. Overflow can occur because the magnitude of the result is greater than the

magnitude of each number and could fall out of the presentable range.

b. Overflow does not occur because the magnitude of the result is smaller than

one of the numbers; the result is in the presentable range.

c. When we subtract a positive integer from a negative integer, the magnitudes of

the numbers are added. This is the negative version of case a. Overflow can

occur.

d. When we subtract two negative numbers, the magnitudes are subtracted from

each other. This is the negative version of case b. Overflow does not occur.

P4 – 20.

The result will be the maximum number of bits allowed.

P4 – 21.

The result will be 0