Computer Organization and Structure

HW #3 - Suggested Solutions

1.

a 1.EA4B
2.F034

b 1.CFE3
2.8406

c 1.3380, 3380
2.47645, -14877

d 1.9662
2.6321

e 1.DE96
2.F29D

f

1.1011 1010 0111 1100
2.1010 1010 1101 1111
3. The attraction is that each hex digit contains one of 16 different characters (0–9, A–E). Since with 4 binary bits you can represent 16 different patterns, in hex each digit requires exactly 4 binary bits. And bytes are by definition 8 bits long, so two hex digits are all that are required to represent the contents of 1 byte.

2.

a 1. 50_{oct}*23_{oct}

	Action	Multiplier	Multiplicand	Product
0	Initial Vals	010 011	000 000 101 000	000 000 000 000
1	Prod = Prod + Mcand	010 011	000 000 101 000	000 000 101 000
	Lshift Mcand	010 011	000 001 010 000	000 000 101 000
	Rshift Mplier	001 001	000 001 010 000	000 000 101 000
2	Prod = Prod + Mcand	001 001	000 001 010 000	000 001 111 000
	Lshift Mcand	001 001	000 010 100 000	000 001 111 000
	Rshift Mplier	000 100	000 010 100 000	000 001 111 000
3	lsb = 0, no op	000 100	000 010 100 000	000 001 111 000
	Lshift Mcand	000 100	000 101 000 000	000 001 111 000
	Rshift Mplier	000 010	000 101 000 000	000 001 111 000
4	lsb = 0, no op	000 010	000 101 000 000	000 001 111 000
	Lshift Mcand	000 010	001 010 000 000	000 001 111 000
	Rshift Mplier	000 001	001 010 000 000	000 001 111 000
5	Prod = Prod + Mcand	000 001	001 010 000 000	001 011 111 000
	Lshift Mcand	000 001	010 100 000 000	001 011 111 000
	Rshift Mplier	000 000	010 100 000 000	001 011 111 000
6	lsb = 0, no op	000 000	001 010 000 000	001 011 111 000
	Lshift Mcand	000 000	101 000 000 000	001 011 111 000
	Rshift Mplier	000 000	101 000 000 000	001 011 111 000

2. 66_{oct}*04_{oct}

	Action	Multiplier	Multiplicand	Product
0	Initial Vals	000 100	000 000 110 110	000 000 000 000
1	lsb = 0, no op	000 100	000 000 110 110	000 000 000 000
	Lshift Mcand	000 100	000 001 101 100	000 000 000 000
	Rshift Mplier	000 010	000 001 101 100	000 000 000 000
2	lsb = 0, no op	000 010	000 001 101 100	000 000 000 000
	Lshift Mcand	000 010	000 011 011 000	000 000 000 000
	Rshift Mplier	000 001	000 011 011 000	000 000 000 000
3	Prod = Prod + Mcand	000 001	000 011 011 000	000 011 011 000
	Lshift Mcand	000 001	000 110 110 000	000 011 011 000
	Rshift Mplier	000 000	000 110 110 000	000 011 011 000
4	lsb = 0, no op	000 000	000 110 110 000	000 011 011 000
	Lshift Mcand	000 000	001 101 100 000	000 011 011 000
	Rshift Mplier	000 000	001 101 100 000	000 011 011 000
5	lsb = 0, no op	000 000	001 101 100 000	000 011 011 000
	Lshift Mcand	000 000	011 011 000 000	000 011 011 000
	Rshift Mplier	000 000	011 011 000 000	000 011 011 000
6	lsb = 0, no op	000 000	011 011 000 000	000 011 011 000
	Lshift Mcand	000 000	110 110 000 000	000 011 011 000
	Rshift Mplier	000 000	110 110 000 000	000 011 011 000

b.1. 50_{hex}*23_{hex}

Step	Action	Multiplicand	Product/Multiplier
0	Initial Vals	0101 0000	0000 0000 0010 0011
1	Prod = Prod + Mcand	0101 0000	0101 0000 0010 0011
	Rshift Product	0101 0000	0010 1000 0001 0001
2	Prod = Prod + Mcand	0101 0000	0111 1000 0001 0001
	Rshift Mplier	0101 0000	0011 1100 0000 1000
3	lsb = 0, no op	0101 0000	0011 1100 0000 1000
	Rshift Mplier	0101 0000	0001 1110 0000 0100
4	lsb = 0, no op	0101 0000	0001 1110 0000 0100
	Rshift Mplier	0101 0000	0000 1111 0000 0010
5	lsb = 0, no op	0101 0000	0000 1111 0000 0010
	Rshift Mplier	0101 0000	0000 0111 1000 0001
6	Prod = Prod + Mcand	0101 0000	0101 0111 1000 0001
	Rshift Mplier	0101 0000	0010 1011 1100 0000
7	lsb = 0, no op	0101 0000	0010 1011 1100 0000

	Rshift Mplier	0101 0000	0001 0101 1110 0000
8	lsb = 0, no op	0101 0000	0001 0101 1110 0000
	Rshift Mplier	0101 0000	0000 1010 1111 0000

2. $66_{hex} x 04_{hex}$

Step	Action	Multiplicand	Product/Multiplier
0	Initial Vals	0110 0110	0000 0000 0000 0100
1	lsb = 0, no op	0110 0110	0000 0000 0000 0100
	Rshift Mplier	0110 0110	0000 0000 0000 0010
2	lsb = 0, no op	0110 0110	0000 0000 0000 0010
	Rshift Mplier	0110 0110	0000 0000 0000 0001
3	Prod = Prod + Mcand	0110 0110	0110 0110 0000 0001
	Rshift Product	0110 0110	0011 0011 0000 0000
4	lsb = 0, no op	0110 0110	0011 0011 0000 0000
	Rshift Mplier	0110 0110	0001 1001 1000 0000
5	lsb = 0, no op	0110 0110	0001 1001 1000 0000
	Rshift Mplier	0110 0110	0000 1100 1100 0000
6	lsb = 0, no op	0110 0110	0000 1100 1100 0000
	Rshift Mplier	0110 0110	0000 0110 0110 0000
7	lsb = 0, no op	0110 0110	0000 0110 0110 0000
	Rshift Mplier	0110 0110	0000 0011 0011 0000
8	lsb = 0, no op	0110 0110	0000 0011 0011 0000
	Rshift Mplier	0110 0110	0000 0001 1001 1000

3.

a.1.614858756, 614858756

2.-1346437120, 2948530176

b.

1. 001001 00101 00110 0000000000000100

Syntax:	addiu $t, $s, imm
Encoding:	0010 01ss ssst tttt iiii iiii iiii iiii

=> **addiu $a2, $a1, 4**

2. 101011 11101 11111 0000000000000000

Syntax:	sw $t, offset($s)
Encoding:	1010 11ss ssst tttt iiii iiii iiii iiii

=> **sw $ra, 0($sp);**

c.1. 0|010 0100 1|010 0110 0000 0000 0000 0100_2
sign is positive
exponent = 010 0100 1_2=73
fraction = 010 0110 0000 0000 0000 01_2
x = (1+010 0110 0000 0000 0000 01_2)*$2^{(73-127)}$
= (1.296875476837158203125)*$2^{(-54)}$
2. 1|010 1111 1|011 1111 0000 0000 0000 0000_2
sign is negative
exponent = 010 1111 1_2=95
fraction = 011 1111 $_2$
x = -(1+011 1111_2)*$2^{(95-127)}$
= -(1.4921875)*$2^{(-32)}$

d.1. -1609.5_{10} = -11001001001.1_2
=-(1.10010010011_2)*2_{10}
Exponent = 10+127 = 137_{10} = 10001001_2
ans = 1 10001001 1001 0010 0110 0000 0000 000
2. -983.8125 = -1111010111.1101
= -(1.1110101111101_2)*2_9
Exponent = 9+127 = 136_{10} = 10001000_2
ans = 1 10001000 1110 1011 1110 1000 0000 000

e.1. -1609.5_{10} = -11001001001.1_2
=-(1.10010010011_2)*2_{10}
Exponent = 10+1023 = 1033_{10} = 100 0000 1001_2
ans = 1 100 0000 1001 1001 0010 0110 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
2. -983.8125 = -1111010111.1101
= -(1.1110101111101_2)*2_9
Exponent = 9+1023 = 1032_{10} = 100 0000 1000_2
ans = 1 100 0000 1000 1110 1011 1110 1000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

f.
IBM floating point numbers are represented by one bit for the sign (S), 7 bits for the exponent, and 24 bits for the fraction. The exponent is to the base 16 (not 2), and has a bias of 64.
1. -1609.5_{10} = -0110 0100 1001.1_2
=-649.8_{16}
= -(.0110 0100 1001 1_2)*16_3
Exponent = 3+64 = 67_{10} = 100 0011_2
ans = 1 100 0011 0110 0100 1001 1000 0000 0000
2. -983.8125 = -0011 1101 0111.1101
= -(.0011 1101 0111 1101_2)*16_3
Exponent = 3+64 = 67_{10} = 100 0011_2
ans = 1 100 0011 0011 1101 0111 1101 0000 0000

4.

S=0: Select A
S=1: Select B

A(input)	B(input)	S(input)	C(output)
X	1	1	1
1	X	0	1
X	0	1	0
0	X	0	0

A(input)	B(input)	S(input)	C(output)
1	1	1	1
1	1	0	1
1	0	1	0
1	0	0	1
0	1	1	1
0	1	0	0
0	0	1	0
0	0	0	0