EECE.3170: Microprocessor Systems Design I

Fall 2016

Homework 1 Solution

1. *(50 points) Given each of the binary or hexadecimal number below, determine what the decimal value is if the number is (i) an unsigned integer, and (ii) a signed integer. Note that, in some cases, your answers for both will be the same.*

a. 01011000_2

Since MSB = 0, value is same whether unsigned or signed—figure out the significance of each position in which a bit = 1, and sum those values together.

01011000_2= 64 + 16 + 8 = **88**

b. 11001011_2

For an unsigned integer, we use the same method as in part (a)

11001011_2= 128 + 64 + 8 + 2 + 1 = **203**

For a signed integer, recognize that this value is negative; to find its magnitude, take the two's complement:

$-11001011_2 = 00110100_2 + 1 = 00110101_2$ = 32 + 16 + 4 + 1 = 53

Therefore, 11001011_2= **-53** when treated as a signed integer.

c. 0x93—recall that the leading 0x signifies the following value is in hexadecimal

For an unsigned integer, we don't really need to convert to binary; if you want to do so, 0x93 = 10010011_2. However, we can also just convert directly to decimal:

93_{16} = (9 x 16) + (3 x 1) = **147**

As a signed integer, note that this value is negative, since its MSB = 1. To find the magnitude, once again take the two's complement:

$-10010011_2 = 01101100_2 + 1 = 01101101_2$ = 64 + 32 + 8 + 4 + 1 = 109

Therefore, as a signed integer, 0x93 = **-109.**

d. 0x51A3

Since the most significant bit of this number is 0 (0x51A3 = $0101\ 0001\ 1010\ 0011_2$), it has the same value whether it is treated as a signed or unsigned integer. That value is:

$(5 \text{ x } 16^3) + (1 \text{ x } 16^2) + (10 \text{ x } 16^1) + (3 \text{ x } 16^0) =$
$(5 \text{ x } 4096) + (1 \text{ x } 256) + (10 \text{ x } 16) + (3 \text{ x } 1) = 20480 + 256 + 160 + 3 =$ **20899**

e. 0xDAB0

This number has different values when treated as signed or unsigned, since the MSB is 1 (0xDAB0 = $1101\ 1010\ 1011\ 0000_2$). As an unsigned integer:

$(13 \text{ x } 16^3) + (10 \text{ x } 16^2) + (11 \text{ x } 16^1) + (0 \text{ x } 16^0) =$
$(13 \text{ x } 4096) + (10 \text{ x } 256) + (11 \text{ x } 16) + (0 \text{ x } 1) =$
$53248 + 2560 + 176 =$ **55984**

As a signed integer, the magnitude is:

-0xDAB0h = $-1101\ 1010\ 1011\ 0000_2 = 0010\ 0101\ 0101\ 0000_2$ = 0x2550

I've shown the conversion back into hexadecimal because it might be slightly easier to figure out the decimal value of the magnitude using what we already know about converting a 16-bit value from hex to decimal:

$(2 \text{ x } 16^3) + (5 \text{ x } 16^2) + (5 \text{ x } 16^1) + (0 \text{ x } 16^0) =$
$(2 \text{ x } 4096) + (5 \text{ x } 256) + (5 \text{ x } 16) + 0 = 8192 + 1280 + 80 = 9552$

Therefore, 0xDAB0 = **-9552** as a signed integer.

2. *(50 points) Assume the contents of memory are shown below. All values are in hexadecimal. The table shows four bytes per line; the given address is the starting address of each line.*

   Each block in the table contains a single byte, with the low and high bytes per line indicated as shown. Each byte has its own address, so the byte at address 0x92220 is 0x89, address 0x92221 is 0xAE, address 0x92222 is 0xE1, and address 0x92223 is 0xF4.

   You should assume all multi-byte values are stored in little-endian format.

Address	Lo			Hi
0x92220	89	AE	E1	F4
0x92224	15	BA	FF	**70**
0x92228	**31**	**CE**	**EE**	**23**
0x9222C	**19**	**78**	**01**	06
0x92230	15	12	24	07
0x92234	B3	**A2**	**99**	**DA**
0x92238	44	20	C5	B6

For each address and amount of data listed, answer the following:

- *What data are stored at that address?*
- *Would an access to the given amount of data at that address be aligned?*
- *If the data represents a signed integer, what is the sign of that value?*

For example, given "Address: 0x92220, Data size: word," your response would be that the word at 0x92220 is 0xAE89, the access is aligned, and the data represents a negative integer.

Note: The key points to remember for this problem are:

- Little-endian data are stored with the least significant byte at the lowest address.
- An access is aligned if the address is divisible by the number of bytes being accessed.
- In signed formats, the integer is positive if the most significant bit is 0 and negative if that bit is 1.

a. *Address: 0x9222C, Data size: word*

The word at this address is **0x7819**, the access is aligned (since 0x9222C is divisible by 2), and the data represents a positive integer, since its most significant bit is 0.

b. *Address: 0x92235, Data size: byte*

The byte at this address is **0xA2**, the access is aligned (since every address is divisible by 1), and the data represents a negative integer, since its most significant bit is 1.

c. *Address: 0x9222B, Data size: double word*

The double word at this address is **0x01781923**, the access is not aligned (since 0x9222B is not divisible by 4), and the data represents a positive integer, since its most significant bit is 0.

(Note: in the diagram above, I've tried to color-code the answers, but this double-word contains the word accessed in part (a), so those two bytes are shown in red and the first and last bytes of the double-word are shown in green.)

d. Address: 0x92236, Data size: word

The word at this address is **0xAD99**, the access is aligned (since 0x92236 is divisible by 2), and the data represents a negative integer, since its most significant bit is 1.

e. Address: 0x92227, Data size: double word

The double word at this address is **0xEECE3170**, the access is not aligned (since 0x92227 is not divisible by 4), and the data represents a negative integer, since its most significant bit is 1.