CSE2312 (Fall 2022) Homework #1

Notes: All numbers are in base-10 unless otherwise noted.
If part of a problem is not solvable, explain why in the answer area.
The 0x number prefix indicates the number is base-16 as in the C language.

The target date to complete this homework set is June 11, 2022.

This homework set will not be graded, but please solve all the problems to prepare for the quizzes and exams.

1. Convert the following numbers between bases:

a. 10111011 (base-2) = ____187____ (base-10)

b. 10111011 (base-2) = _____0xBB__ (base-16)

c. 0x5249 = 5249 (base-16) = ___0101 0010 0100 1001____(base-2)

d. 0x5249 = 5249 (base-16) = ____21065_____(base-10)

e. 16383 = __0011 1111 1111 1111_____ (base-2)

f. 4095 = __0x0FFF_____ (base-16)

2. What is the range of the following C99 variable types assuming the processor uses two's compliment arithmetic for signed number representation?

a. uint8_t ____0____ to ____255__(2^8- 1)____

b. uint16_t _____0_____ to _65535_(2^{16}- 1)_

c. uint32_t ___0__ to _4,264,967,295_(2^{32}- 1)__

d. int8_t ____-128_-(2^{8-1})__ to ____127_(2^{8-1}- 1)____

e. int16_t __-32,768_-(2^{16-1})___ to ____32,767_(2^{16-1} - 1)_

f. int32_t -2,147,483,648_-(2^{32-1}) to _2,147,483,647_(2^{32-1} - 1)

3. Write the binary representation of the C99 variables given below.

Example: for uint8_t x = 13, the answer would be answer is: 0000 1101 (base-2)

a. uint8_t x = 27; 00011011

b. uint8_t x = 122; 0111 1010

c. uint8_t x = 215; 1101 0111

d. uint8_t x = 40; 0010 1000

e. int8_t x = -40; 1101 1000

f. int8_t x = -103; 1001 1001

g. int8_t x = 103; 0110 0111

h. uint16_t x = 13000; 0011 0010 1100 1000

i. int16_t x = 13000; 0011 0010 1100 1000

j. int16_t x = -13000; 1100 1101 0011 1000

k. uint32_t x = 262144; 0000 0000 0000 0100 0000 0000 0000 0000

l. int32_t x = -50; 1111 1111 1111 1111 1111 1111 1100 1110

m. int32_t x = 50; 0000 0000 0000 0000 0000 0000 0011 0010 (Leading zeros NOT required unless describing register contents, which this is not)

4. Write the status of the Carry (C), Zero (Z), Negative/Sign (N), and Overflow (V) flags after an 8-bit ALU performs an ADD operation on the following 8-bit arguments (a and b):

Hint: Remember that the ALU just sees bits and does not know if the numbers represent signed or unsigned numbers.

a. uint8_t a = 91, uint8_t b = 23 C = 0, Z = 0, V = 0, N = 0

b. uint8_t a = 102, uint8_t b = 3 C = 0, Z = 0, V = 0, N = 0

c. int8_t a = 32, int8_t b = -22 C = 1, Z = 0, V = 0, N = 0

d. int8_t a = -32, int8_t b = 22 C = 0, Z = 0, V = 0, N = 1

e. int8_t a = 100, int8_t b = -100 C = 1, Z = 1, V = 0, N = 0

f. int8_t a = -130, int8_t b = 100 No Solution, -130 exceeds range of int8_t

g. int8_t a = -32, int8_t b = 72 C = 1, Z = 0, V = 0, N = 0

5. Assuming an 8-bit ALU, show the status of the Zero (Z) and Sign (S) flags and the result after each operation. Note the bases carefully.

a. arg1 = 33, arg2 = 2; result = arg1 OR arg2 0010 0011 (0x35), Z = 0 , S = 0

b. arg1 = 0x23, arg2 = 0x14; result = arg1 OR arg2 0011 0111 (0x37), Z = 0 , S = 0

c. arg1 = 0x2C, arg2 = 0x78; result = arg1 AND arg2 0010 1000 (0x28), Z = 0 , S = 0

d. arg1 = 0xA5; result = NOT arg1 0101 1010 (0x5A), Z = 0, S = 0

e. arg1 = 29; result = NEG arg1 1110 0011 Z = 0, S = 1