#HW, cec32x_HW_28_HW5_data_representations_and_programming, 140 points total.

HW 5. Data representations and mixed programming

Prob 5-1. (60 points total, 10 points each) Determine the result in both decimal and binary format and C and V flags of the following calculations for a 5-bit system using the C and V flag convention of an ARM processor:

```
1.  (-16) + (-16)
2.  14 - (-16)
3.  14 - 16
4.  15 - (-6)
5.  15 + (-6)
6.  14 + (-16)
```

Prob 5-2. R0 = 0xF, R1 = 0xFF, R2 = 0xFFF, and R4 = 0xFFFF. (These are all the uint values passed to the corresponding registers.)

Prob 5-2. (20 points total, 5 points each) Suppose we run the following statement: `function1(0xF, 0xFF, 0xFFF, 0xFFFF);`. What will be the values of registers R0, R1, R2, and R3 immediately after the program runs within function declared as `function1(uint32_t x, uint32_t y, uint32_t z, uint32_t w);` due to this call?

Prob 5-3. R0 = 15 and R1 = -1 or 0xFFFF_FFFF. The reason for R1 = 0xFFFF_FFFF is that we need to conver var2 (-1) from 8 bits to 32 bits (still -1 without changing the value).

Prob 5-3. (10 points total, 5 points each) Suppose we have defined two variables as `int8_t var1 = 0xF;` and `int8_t var2 = 0xFF;` and run the following statement: `function2(var1, var2);`. What will be the values of registers R0 and R1 immediately after the program runs within function `function2(int8_t x, int8_t y)` due to this call?

Prob 5-4. a. The return is in R0. b. The return is in R0 and R1 since two 32-bit words are needed.

Prob 5-4. (10 points total, 5 points each) Suppose we call a function that calculates the center of gravity of an aircraft using the Q15.16 format for the return. To which of the register(s) the value has to be loaded within the function so that that value can be taken by the caller? What if we use the Q31.32 format?

Prob 5-5. 3.25 * 2^16 = 3.25 * 4 * 2^14 = 13 * 2^14 (=0b0000_0000_0000_0011_0100_0000_0000_0000 no need to give the binary form since this is not specified).

Prob 5-5. (10 points total) Suppose the Q15.16 value returned from the above problem is 3.25. What will be the value in R0 before the function returns?

Prob 5-6. BX lr

Prob 5-6. (10 points total) Which is the instruction in the assembly function on page 8/9 of the class notes for Module 5 that stops the function and returns to the caller?

Prob 5-7. (20 points total) Write an improved C language version of the assembly function on page 8/9 of the class notes for Module 5. Note that the improvement is to return the number of bytes you copied from the source to the destination instead of using the global variable `count`. (Note that you still use the same two parameters for the function as before, and also you need to use the ending null character `'\0'` of the string pointed by `src` to stop the loop of the function.)

Soln for Prob 1: (1) unsigned; (2) signed.

(a) (1) $-16 \Rightarrow -16+32=16$

$\Rightarrow (-16)+(-16)=16+16=32 \notin [0,31] \Rightarrow \boxed{C=1}$

result $= 32-32 = \boxed{0_{10} = 0b00000}$

(2) $(-16)+(-16)=-32 \notin [-16,15] \Rightarrow \boxed{V=1}$

result $= -32+32 = \boxed{0_{10} = 0b00000}$

(b) (1) $-16 \Rightarrow -16+32=16$

$\Rightarrow 14-(-16) \Rightarrow 14-16=-2 \notin [0,31] \Rightarrow B=1 \Rightarrow \boxed{C=0}$

result $= -2+32 = \boxed{30_{10} = 0b11110}$

(2) $14-(-16)=14+16=30 \notin (-16,15] \Rightarrow \boxed{V=1}$

result $= 30-32 = \boxed{-2_{10}} = \boxed{0b11110}$

(C) (1) $14-16 \Rightarrow$ The same as (b) (1)

$\Rightarrow$ result $= \boxed{30_{10} = 0b11110}$, $\boxed{C=0}$

(2) $14-16=-2 \in [-16,15) \Rightarrow \boxed{V=0}$

$\boxed{\text{result} = -2_{10} = 0b11110}$

(d) (1) $-6 \Rightarrow -6+32=26$

$\Rightarrow 15-(-6) \Rightarrow 15-26=-11 \in [0,31) \Rightarrow B=1 \Rightarrow \boxed{C=0}$

result $= -11+32 = \boxed{21_{10} = 0b10101}$

(2) $15-(-6)=21 \in [-16,15) \Rightarrow \boxed{V=1}$

result $= 21-32 = \boxed{-11_{10} = 0b10101.}$

(e) (1) $-6 \Rightarrow -6+32=26$

$\Rightarrow 15+(-6) \Rightarrow 15+26=41 \notin (0,31] \Rightarrow \boxed{C=1}$

result $= 41-32 = \boxed{9_{10} = 0b01001}$

(2) $15+(-6)=9 \in [-16,15] \Rightarrow \boxed{V=0}$

result $= \boxed{9_{10} = 0b01001}$

(f) (1) $-16 \Rightarrow -16+32=16$

$\Rightarrow 14+(-16) \Rightarrow 14+16=30 \in [0,31] \Rightarrow \boxed{C=0}$

result $= \boxed{30_{10} = 0b11110}$

(2) $14+(-16)=-2 \in [-16,15] \Rightarrow \boxed{V=0}$

result $= \boxed{-2_{10} = 0b11110}$

Prob 5-7. (Note that number of bytes copied is not defined clearly, so when you copy N bytes plus the null character, you can return N or N+1. Both will be seen correct.

```
int copy_src_to_dst(uint8_t *src, uint8_t *dst) {
    int i = 0;
    do {
        uint8_t temp = *src++;
        *dst++ = temp;
        i++;
    } while (temp > 0);

    return i;
}
```

Other loops will also work. For example

```
int copy_src_to_dst(uint8_t *src, uint8_t *dst) {
    for (int i = 1; ; i++) {
        uint8_t temp = *src++;
        *dst++ = temp;
        if (temp == 0) {
            break;
        }
    }

    return i;
}
```