CSE2312-001 (Fall 2021)
Homework #6

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 target date to complete problems 1-2 of this homework set is November 20, 2021. The target date to complete problems 3-5 of this homework set is November 30, 2021

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

1. Write assembly functions that implement the following C functions:

   a. float sumF32(const float x[], uint32_t count)
      // returns the sum of the elements in an array (x) containing count entries

   b. double prodF64(const double x[], uint32_t count)
      // returns the product of the elements in an array (x) containing count entries

   c. double dotpF64(const double x[], const double y[],uint32_t count)
      // returns the dot product of two arrays (x and y) containing count entries

   d. float maxF32(const float x[], uint32_t count)
      // returns the maximum value in the array (x) containing count entries

2. For the following code, calculate the number of instruction cycles required to execute the following code, using the simplified pipeline timing rules in class, including the time to call this function with BL bro8 and the time to return from the function with BX LR. You can assume that the pipeline is full before the BL bro8 instruction is executed.

```
bro8:                       Clocks    Iterations
    MOV R1, R0              1           1          1x1     = 1
    MOV R0, #0              1           1          1x1     = 1
    MOV R2, #0x00000080     1           1          1x1     = 1
    MOV R3, #0x00000001     1           1          1x1     = 1
bro8_loop:
    TST R1, R2              1           8          1x8     = 8
    ORRNE R0, R3            1           8          1x8     = 8
    MOVS R2, R2, LSR #1     1           8          1x8     = 8
    MOV R3, R3, LSL #1      1           8          1x8     = 8
    BNE bro8_loop           3 or 1     (7)+(1) (7x3)+(1x1) = 22
    BX LR                   2           1          1x2     = 2
                                                 4+32+22+2 = 60
```

Clock cycles: _____60clock cycles

If the clock rate is 2 GHz, what is the execution time in nanoseconds? _____30 ns

The only thing that effects execution path is:
BNE bro8_loop
which depends on the status of R2
R2 is modified by:
MOVS R2, R2, LSR #1

So... converting R2 to binary from Hexadecimal looks like:
0x0 0 0 0 0 0 8 0
0000 0000 0000 0000 0000 0000 1000 0000

It will take 8 LSR before the only bit is shifted out of the register R2.
This means the loop will be taken 8 times because the branch condition will be true 7 times and on last time the condition will be false

Clock cycles = 60

2 GHz means 2 cycles/nanosecond so

$$\frac{60 \text{ cycles}}{} \times \frac{\text{nanoseconds}}{2 \text{ cycles}} = 30 \text{ nanoseconds}$$

3. Assume SP = 0x20001034 before the following instructions are executed:

```
Address     Instruction
10000000:       BL fn
          fn:

10001000:       MOV R0, #8192
10001004:       MOV R1, #0x10000000
10001008:       MOV R2, #0x7400
1000100C:       PUSH {R0, R1, R2, LR}
          loop:
10001010:       B loop
```

After this program enters the endless loop:

What is the value of the SP? _____

Assuming the processor uses little-endian convention, what is the value of the following memory locations (place X in the blank if there is not enough information):

Address	8-bit Data
0x2000103B	_____
0x2000103A	_____
0x20001039	_____
0x20001038	_____
0x20001037	_____
0x20001036	_____
0x20001035	_____
0x20001034	_____
0x20001033	_____
0x20001032	_____
0x20001031	_____
0x20001030	_____
0x2000102F	_____
0x2000102E	_____
0x2000102D	_____
0x2000102C	_____
0x2000102B	_____
0x2000102A	_____
0x20001029	_____
0x20001028	_____
0x20001027	_____
0x20001026	_____
0x20001025	_____
0x20001024	_____

4. Explain the concept of memory virtualization, including the concept of paging and fragmentation. Also explain the role of virtualization in memory protection between running processes (“programs”).

5. Explain the concept of cache, including the principle of locality. Explain how this can speed up memory accesses.