CS 31 Homework 3: ARM64 Arithmetic

Due at the start of class Thursday, September 29, 2022

Your Name(s):

1. Assume the CPU is executing a program and the state of some of its registers is given in the table below. Show how the registers would be updated by the sequence of ARM64 instructions listed below by filling in the `Final Value` column. Show your work by listing the intermediate values of the registers.

Register	Current Value (show work here)	Final Value
x0	0	
x1	1	
x2	2	
x3	3	

Update the register state based on these instructions:

```
add  x0, x0, #20
add  x1, x0, x1
sub  x2, x2, x1
add  x2, x2, #3
sub  x3, x3, x2
add  x3, x3, x3
sub  x3, x3, #1
lsr  x1, x1, #4
and  x3, x3, #0xfe    /* this is tricky */
eor  x0, x0, x0       /* this is tricky */
orr  x2, x2, #0
/* think about these next two before answering */
mvn  x1, x1
add  x1, x1, #1
```

2. Assume the CPU is executing a function that has local variables `x`, `y`, and `z` allocated on the stack, and that `x` is allocated at the memory address that is `16` bytes from the address value stored in register `sp`, or `[sp, #16]`. Assume `y` is stored at `[sp, #8]`, and `z` is at `[sp]`.
For the assembly code and register values listed below:
(1) Show the values that will be stored in the registers and in memory when execution of these instructions is complete. If the value is unknown, write "?".
(2) Write a C code translation of the assembly code sequence. You may assume that `x`, `y`, and `z` have already been declared as `long` variables in the C code. You do not need to write the entire function, just the lines of C that might have generated the ARM64 instructions. Hint: our solution is 4 lines of C code.

```
                                        C Code Translation
                                        ------------------
mov  x0, #2
mov  x1, #3
str  x0, [sp, #16]
str  x1, [sp, #8]
add  x0, x0, x1
str  x0, [sp]
lsl  x1, x1, #1
str  x1, [sp, #8]
```

Memory Address	Final Value
0xff30	
0xff38	
0xff40	
0xff48	
0xff50	
0xff58	
0xff60	

Register	Initial Value	Final Value
x0	4	
x1	7	
sp	0xff48	