CS 2200 Homework 4

Fall 2018

Rules:

- Please print a copy of the assignment and hand write your answers. No electronic submissions are allowed. **Please print as one double-sided page. Do NOT staple multiple sheets together.** There will be a **40** point penalty if you do not.
- This is an individual assignment. You may discuss concepts but not the answers.
- Due Date: **26 September 2018 – 6:00 PM**. Bring your BuzzCard. Show up on time.

Name:_____ **GT Username:**_____ **Section:**_____

1. What is the IVT, what does it store, and where is it located?

2. Why is it necessary to disable interrupts in the INT macrostate when you enter the interrupt handler?

3. Consider the table below. You find some way to reduce the CPI of Type B instructions to 4, but have to increase the cycle time. What is the maximum increase (in %age) in the cycle time that makes this change beneficial?

Instruction type	Cycles	Frequency in program
A	2	30%
B	6	50%
C	5	20%

4. Complete the following interrupt handler by filling in the missing assembly code. Assume the processor has already placed PC in $k0, disabled interrupts, and set the PC ot the first instruction in int_handler.

   We've given you the main logic of the handler, which simply increments a counter in memory. **Your completed handler must allow nested interrupts, i.e., another interrupt should be permitted to come in while the handler is running.** Do not leave interrupts disabled longer than absolutely necessary. Only save the minimum of registers absolutely necessary.

   Here are the new instructions we added to LC-2200 for interrupt handling:
   - **EI** - enable the ability for interrupts to occur
   - **DI** - disable the ability for interrupts to occur
   - **RETI** - set PC = $k0 and enable interrupts

```
int_handler:
addi   $sp, $sp, _____
sw     _____, 0x0($sp)
_____

addi   _____, _____, -2
sw     _____, 0x1($sp)
sw     _____, 0x0($sp)

! main interrupt handler logic begins
lea    $t0, counter              ! load address of counter into $s0
lw     $t1, 0x0($t0)             ! load value of counter into $s1
addi   $t1, $t1, 1               ! add 1 to counter value
sw     $t1, 0($t0)               ! store new counter value in memory
! main interrupt handler logic ends

_____ $t0, 0x1($sp)
_____ $t1, 0x0($sp)
addi   _____, _____, 2


_____
lw     _____, _____($sp)
addi   $sp, _____, _____
_____

counter:   .fill 0x0             ! external counter
```