Name _____ Student ID _____

CSc 252: Computer Organization

Spring 2025

Test 3: Mar. 6, 2025

Time: 30 min

DO NOT OPEN THE EXAM UNTIL INSTRUCTED TO DO SO

Please read this page and follow the directions before proceeding with the rest of the exam.

- To give all students the same amount of time to do the exam, please DO **NOT** OPEN THIS EXAM UNTIL INSTRUCTED TO DO SO.
- You are not allowed to use any external resources such as cellphones, notes, headphones, watch, neighbors, calculator, etc. If you have not done so yet, turn your cellphone off and place it in your backpack.
- The cellphone cannot be on you during the exam. If your cellphone is in your pocket, it will be considered cheating even if you are not looking at it (same for headphones and watches). We will collect your exam and ask you to leave.
- Place your final answers in the given boxes. You can show your work on any blank spaces.
- We recommend skimming the entire exam before completing any problems.
- Read carefully every question before answering and raise your hand if the question is unclear.
- DO NOT SPEAK TO ANYONE AT YOUR TABLE.

****** Good Luck! ☺ ******

Allowable MIPS Instructions

When writing MIPS assembly, the only instructions that you are allowed to use (so far) are:

- `add, addi, sub`
- `beq, bne, j`
- `slt, slti`
- `and, andi, or, ori, nor, nori, xor, xori`
- `sll, srl, sra`
- `lw, lh, lb, sw, sh, sb`
- `lui`
- `la`
- `syscall`

Name _____ Student ID _____

Write your name and student ID on all the exam pages for one extra credit point.

1. (3 points) Convert the following two's complement binary number to decimal: 0100 1101.

Place your final answers here.

77 or $2^6+2^3+2^2+2^0$

2. (1 point) Consider a single byte. If it is an unsigned integer, what are the minimum and maximum values that it can hold? Write your answer in decimal.

Place your final answers here.

Minimum Value	0
Maximum Value	255 or 2^8 - 1

(1 point) If it is a signed integer, what are the minimum and maximum values that it can hold?

Place your final answers here.

Minimum Value	-128 or -27
Maximum Value	127 or 27 - 1

3. (3 points) Give a snippet of MIPS code to assign the 32-bit value 0xc000_000c to the register $s0.

Place your final answer here.

A correct answer :

```
addi $s0, $zero, 0xC000
sll  $s0, $s0, 16
addi $s0, $s0, 0x000C
```

Another correct answer :

```
lui  $s0, 0xC000
ori  $s0, $s0, 0x000C
```

Name _____ Student ID _____

Mark your responses by **completely darkening the circle** with the correct answer. e.g., ●

4. (2 points) Which registers change after executing the MIPS instruction `jal fun1`

$ra and pc

5. (2 points) Which registers change after executing the MIPS instruction `j end`

pc

6. (2 points) Which registers change after executing the MIPS instruction `jr $ra`

pc

7. (10 points) Give the proper values for **bNegate** and **operation code** for each of the following instructions.
   You are not required to give the numbers (0,1,2,3); you may use the names of the operations (ADD, OR, etc).
   In case you've forgotten, I've given you a picture of the ALU, as well as the ALU element in the formula sheet.

	Place your final answers here.	
Instruction	**bNegate**	**Operation Code**
ADDI	0	ADD
SLTI	1	LESS
AND	0	AND
SUB	1	ADD

Name _____ Student ID _____

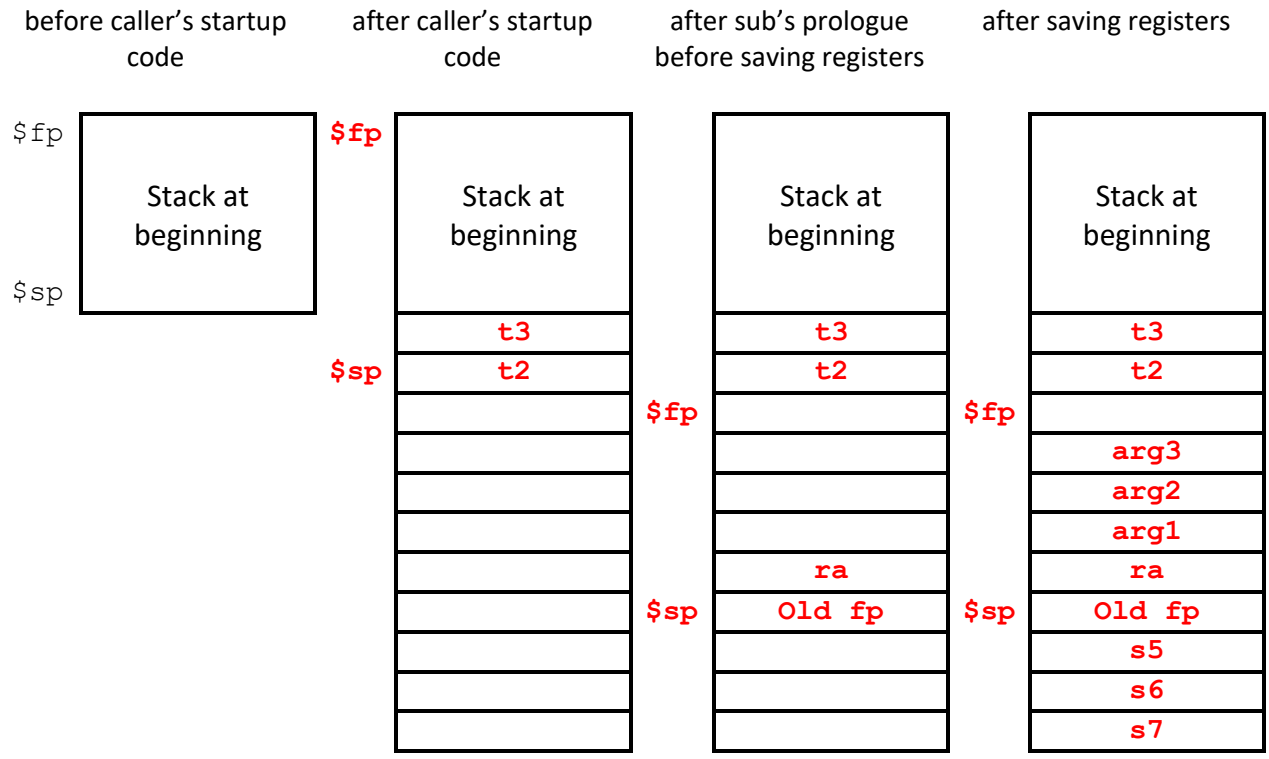
8. (10 points) In this problem, `caller()` calls `sub()`. The first column shows the state of the stack while `caller()` is running, just before its startup code. In the second column, show the state of the stack after the startup code in `caller()` has been completed but before the `jal` instruction. In the third column, show the state of the stack after `sub()` has run its function prologue but before it saved anything other than the prologue (not even `aX` registers). In the fourth column, show the state of the stack after `sub()` has saved all necessary registers.

Make sure to mark:

- The positions of `$fp, $sp`
- All values that have been written to stack. Use `arg1, arg2`, etc., for the various parameters.
- Store the arguments in the appropriate order.

Notes:

- When this problem begins, `caller()` is using (and wants to preserve) the registers `$t2, $t3, $s0, $s4, $s5`.
- `sub()` takes three parameters. It will need to store all of them on the stack.
- `sub()` will use the following registers in its code: `$t0, $t2, $t4, $s5, $s6, $s7`.



9. (11 points) Implement the following as a MIPS function.

```
int chooser(int a, int b, int c, int d) {
          if (a != b)
              return d - a;
          else
              return c + b;
}
```

Name _____ Student ID _____

Place your final answer here.

```
chooser:
    #the standard prologue
    addiu  $sp, $sp, -24
    sw     $fp, 0($sp)
    sw     $ra, 4($sp)
    addiu  $fp, $sp, 20


    beq  $a0, $a1, else # if a==b goto else
    sub  $v0, $a3, $a0  # v0 = d-a
    j    end

else:
    add  $v0, $a2, $a1  # v0 = c+b

end:
      #the standard epilogue
    lw     $ra, 4($sp)
    lw     $fp, 0($sp)
    addiu  $sp, $sp, 24
    jr     $ra
```