

## Assembler and Circuits

Due: March 12, 2018

**QUESTION 1: ASSEMBLER PROGRAMMING**

Download the MARS emulator. The link was provided in the course slides. Select File/New to allow you to input the following program:

```
# Hello World in MIPS Assembly

        .data          # Data segment
msg:     .asciiz "Hello World!\n"

        .text          # Code segment

main:

        la $a0, msg     # Load the address of msg into register $a0
        li $v0, 4       # Tell the OS to use library 4 (print a string)
        syscall         # Call the library function

        li $v0, 10      # Tell the OS to use library 10 (exit program)
        syscall         # Call the library function
```

Then, press Run/Assemble to assemble your program and see syntax errors. When you have no syntax errors, then press Run/Go to see the program run.

Submit a screenshot of the execution of your program on the MARS emulator displaying the output.

**QUESTION 2: ASSEMBLER PROGRAMMING**

Write a MIPS program using the MARS emulator. The program calculates the number of days someone has been alive given their age, assuming every year is exactly 365 days, no leap years. This program must compute the number of days using the ADD instruction and a loop, it cannot use the multiply instruction.

Assume in the .DATA section is a label called YEARS that is .WORD and a label called DAYS which is also a .WORD. Assign a number to YEARS in your source file. Your program uses the number you placed in YEARS, computes the number of days and saves the result in DAYS. The program then terminates.

This program does not output anything on the screen. The user will have to look in memory to see if the computation was carried out correctly.

Submit a screenshot of the execution of your program on the MARS emulator displaying the result.

**QUESTION 3: CIRCUIT DRAWING**

Assume a circuit that has a bus and connected to this bus is the following: two nibble registers R1 and R2, an 8-bit instruction register IR, and a simple control unit CU. IR has a 2-bit op-code and two 3-bit parameters called S and D. The IR register looks like this: IR = OP:S:D, the concatenation of the op-code and S and D parameters into a single register called IR. The op-code 00 means MOVE. The op-code 11 mean ISSAME. The syntax of the MOVE command is as follows:

MOVE DESTINATION\_REGISTER SOURCE\_REGISTER                      Example: MOVE R1 R2

This causes the contents of source register to be copied into the destination register. The previous contents of the destination register are overwritten.

ISSAME REGISTER UNSIGNED\_INTEGER                      Example: ISSAME R1 5

If the register and the integer are the same then nothing happens otherwise the register is assigned the value of the integer. The previous value is overwritten.

Construct the above circuit in Logisim. You will need to determine an optimal bus. You will need to determine how the control unit will execute the two instruction.

Your circuit will need to have: a bus, a control unit, IR, R1, R2.

You can **only** use the following machines in your solution: wires, AND, OR, NOT, XOR, Flip-flop, Clock. You can build black boxes. To facilitate your (and the TA) testing provide input and output pins to initialize and see the contents of R1, R2 and the IR.

**WHAT TO HAND IN**

A screenshot for question 1.

A screenshot for questions 2.

Your logisim circuit file for question 3. If you have multiple files then ZIP them into a single file.

**HOW IT WILL BE GRADED**

Question 1: 5 points

Question 2: 10 points

Question 3: 15 points

Total points = 30

Each question is graded proportionally to how correct your solution is compared with the official solution.