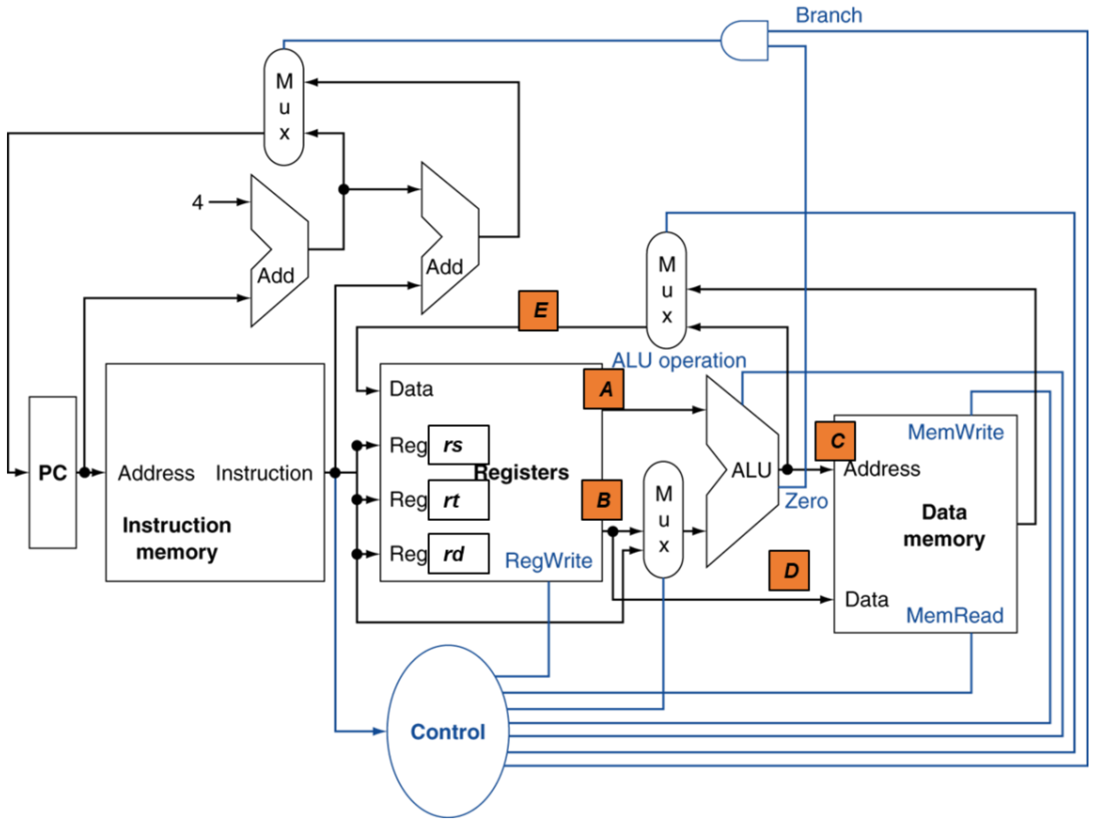
Homework 2- EECS 388: Embedded Systems, Spring 2024

Note: The answer must be submitted as a "typed" MS Word/PDF document.
Total points: 2 (total 100 points in the course)

Question 1 (1.5 point): Based on the datapath of a processor shown in the schematic, write the state/values (in hex) of the marked datapath signals (A to E) after each of the instructions on the table executes. Assume the following initial values for the registers:

    - $t0 = 0x0000_FFFF,
    - $t2 = 0x0000_00FF,
    - $t3 = 0x0000_0FFF,
    - $s1 = 0x0000_000F,
    - $s2 = 0x0000_0F0F.
- Consider the initial value of all other signal and memory location is zero unless changed by earlier instructions.
- These instructions in the table are presented as a sequence (i.e., the later instructions should consider the changes to register, and memory location caused by earlier instructions)
- If a signal is not needed for a specific instruction, you can skip that and mark as "not used" in the table.

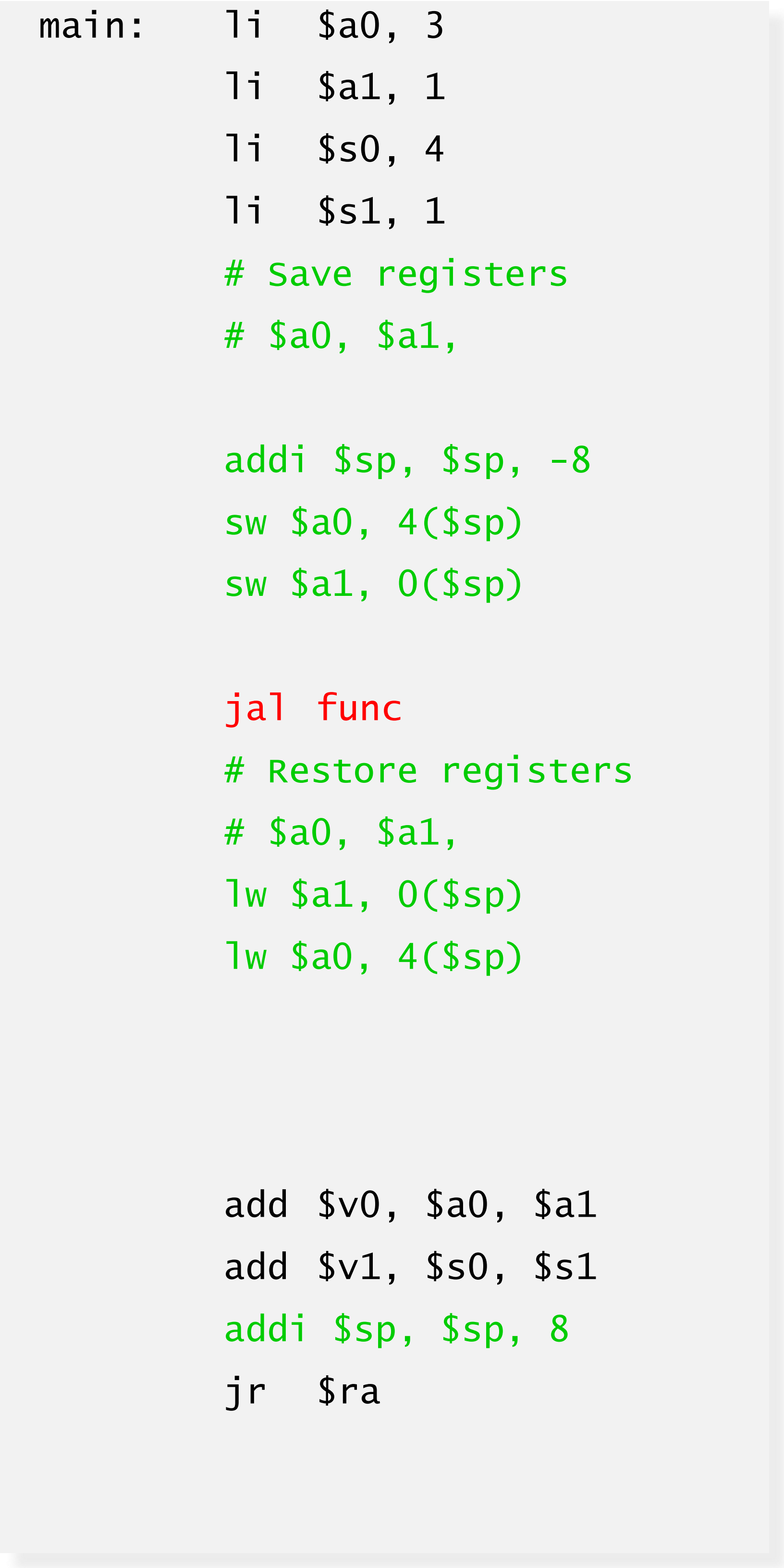


Homework 2- EECS 388: Embedded Systems, Spring 2024

Instruction	A	B	C	D	E
sub $t0, $t2, $s1	Value of $t2 or 0x0000_00FF	Value of $s1 or 0x0000_000F	**Not used**	**Not used**	0x0000_00F0
sw $t3, 0x4($s2)	Value of $s2 Or 0x0000_0F0F	**Not used**	0x0000_0F13	Value of $t3 or 0x0000_0FFF	**Not used**
lw $t2, 0x4($s2)	Value of $s2 Or 0x0000_0F0F	**Not used**	0x0000_0F13	**Not used**	Value of $t2 or 0x0000_00FF

Homework 2- EECS 388: Embedded Systems, Spring 2024

Question 2 (0.5 point): For the following code below, write down the MIPS assembly code to save and restore the registers using stack for register $a0, $a1, $ra, $s0, $s2.



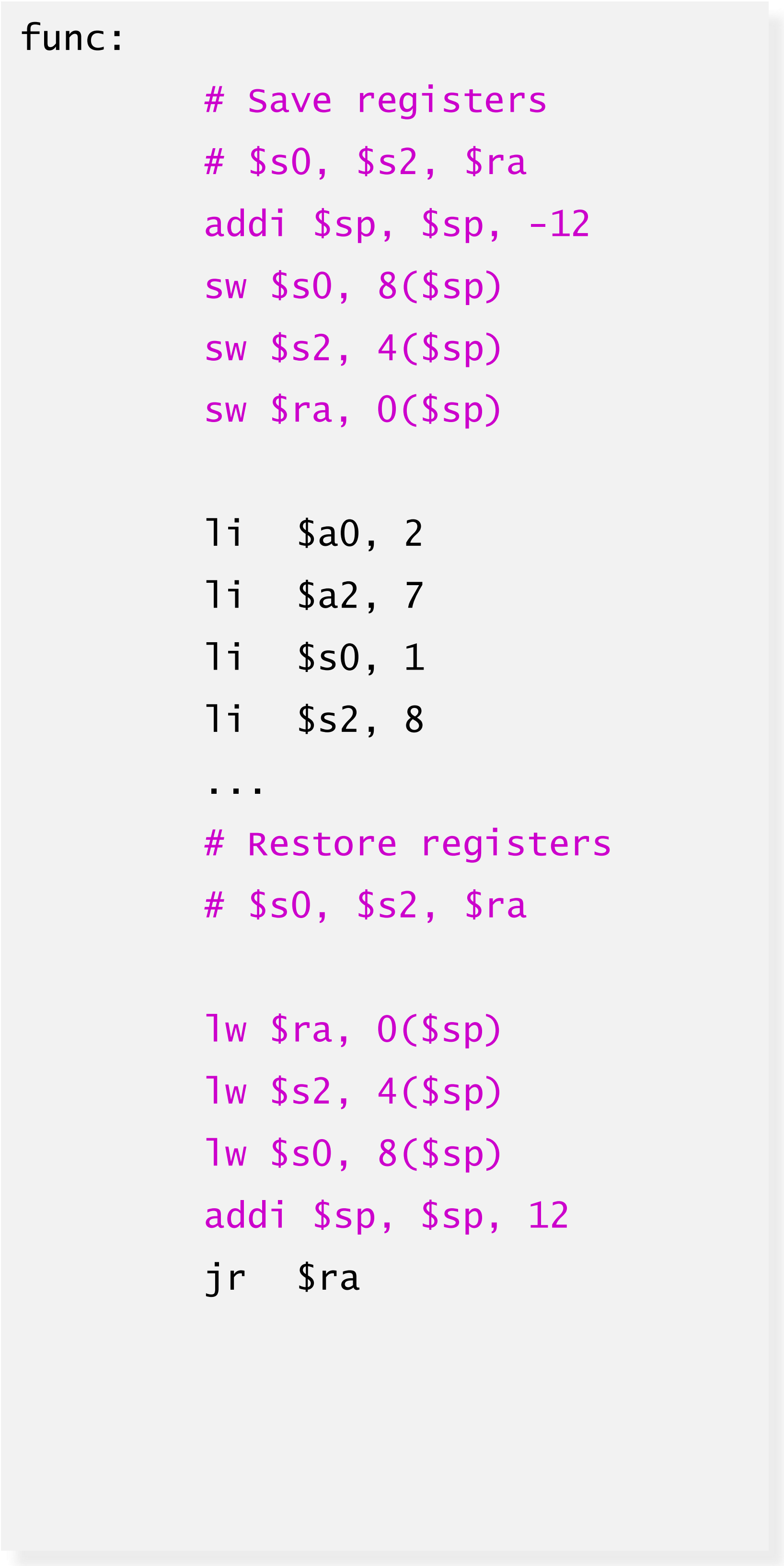
```
main:    li   $a0, 3
         li   $a1, 1
         li   $s0, 4
         li   $s1, 1
         # Save registers
         # $a0, $a1,

         addi $sp, $sp, -8
         sw $a0, 4($sp)
         sw $a1, 0($sp)

         jal  func
         # Restore registers
         # $a0, $a1,
         lw $a1, 0($sp)
         lw $a0, 4($sp)



         add  $v0, $a0, $a1
         add  $v1, $s0, $s1
         addi $sp, $sp, 8
         jr   $ra
```



```
func:
        # Save registers
        # $s0, $s2, $ra
        addi $sp, $sp, -12
        sw $s0, 8($sp)
        sw $s2, 4($sp)
        sw $ra, 0($sp)

        li  $a0, 2
        li  $a2, 7
        li  $s0, 1
        li  $s2, 8
        ...
        # Restore registers
        # $s0, $s2, $ra

        lw $ra, 0($sp)
        lw $s2, 4($sp)
        lw $s0, 8($sp)
        addi $sp, $sp, 12
        jr  $ra
```