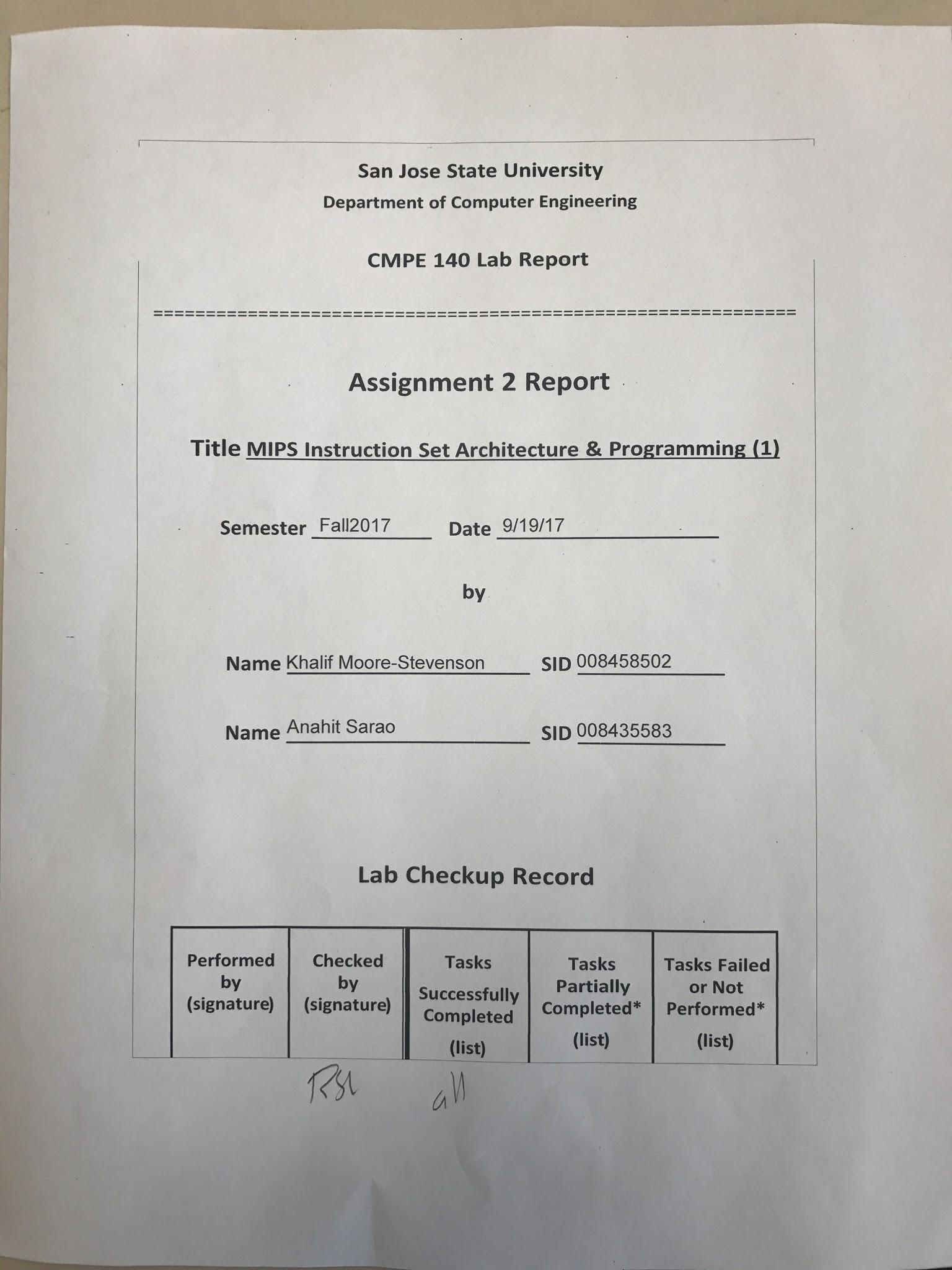
****

***Design Discussion***

The source code provided for this lab was assembled using MIPS assembler. The objective was to run each instruction using the single step function. While each instruction was executed the output was traced as which registers were being utilized including the generated machine code. The source code and output test results table are provided below.

The list of tasks were fully completed in lab:

* Assemble and execute code with MIPS assembler.
* Fill out machine code and register table.

*Source Code*

# Assembly Description Address Machine

main: addi $2, $0, 5 # initialize $2 = 5 0 20020005

addi $3, $0, 12 # initialize $3 = 12 4 2003000c

addi $7, $3, -9 # initialize $7 = 3 8 2067fff7

or $4, $7, $2 # $4 <= 3 or 5 = 7 c 00e22025

and $5, $3, $4 # $5 <= 12 and 7 = 4 10 00642824

add $5, $5, $4 # $5 = 4 + 7 = 11 14 00a42820

beq $5, $7, end # shouldn't be taken 18 10a7000a

slt $4, $3, $4 # $4 = 12 < 7 = 0 1c 0064202a

beq $4, $0, around # should be taken 20 10800001

addi $5, $0, 0 # shouldn't execute 24 20050000

around: slt $4, $7, $2 # $4 = 3 < 5 = 1 28 00e2202a

add $7, $4, $5 # $7 = 1 + 11 = 12 2c 00853820

sub $7, $7, $2 # $7 = 12 - 5 = 7 30 00e23822

sw $7, 68($3) # [80] = 7 34 ac670044

lw $2, 80($0) # $2 = [80] = 7 38 8c020050

j end # should be taken 3c 08000011

addi $2, $0, 1 # shouldn't execute 40 20020001

end: sw $2, 84($0) # write adr 84 = 7 44 ac020054

j main # go back to beginning 48 08000000

*Recorded Test Results*

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Adr | Expected Machine Code | Actual Machine Code | PC | Registers | | | | | Memory Content | |
| $v0 | $v1 | $a0 | $a1 | $a3 | [80] | [84] |
| 00 | 20020005 | 20020005 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 |
| 04 | 2003000c | 2003000c | 4 | 5 | c | 0 | 0 | 0 | 0 | 0 |
| 08 | 2067fff7 | 2067ffff | 8 | 5 | c | 0 | 0 | 3 | 0 | 0 |
| 0c | 00e22025 | 00e22025 | c | 5 | c | 7 | 0 | 3 | 0 | 0 |
| 10 | 00642824 | 00642824 | 10 | 5 | c | 7 | 4 | 3 | 0 | 0 |
| 14 | 00a42820 | 00a42820 | 14 | 5 | c | 7 | b | 3 | 0 | 0 |
| 18 | 10a7000a | 10e5000a | 18 | 5 | c | 7 | b | 3 | 0 | 0 |
| 1c | 0064202a | 0064202a | 1c | 5 | c | 0 | b | 3 | 0 | 0 |
| 20 | 10800001 | 10040001 | 20 | 5 | c | 0 | b | 3 | 0 | 0 |
| 24 | 20050000 | 20050000 |  |  |  |  |  |  |  |  |
| 28 | 00e2202a | 00e2202a | 28 | 5 | c | 1 | b | 3 | 0 | 0 |
| 2c | 00853820 | 00853820 | 2c | 5 | c | 1 | b | c | 0 | 0 |
| 30 | 00e23822 | 00e23822 | 30 | 5 | c | 1 | b | 7 | 0 | 0 |
| 34 | Ac670044 | Ac670044 | 34 | 5 | c | 1 | b | 7 | 7 | 0 |
| 38 | 8c020050 | 8c020050 | 38 | 7 | c | 1 | b | 7 | 7 | 0 |
| 3c | 08000011 | 08000011 | 3c | 7 | c | 1 | b | 7 | 7 | 0 |
| 40 | 20020001 | 20020001 |  |  |  |  |  |  |  |  |
| 44 | Ac020054 | Ac020054 | 44 | 7 | c | 1 | b | 7 | 7 | 7 |
| 48 | 08000000 | 08000000 | 48 | 7 | c | 1 | b | 7 | 7 | 7 |

***Conclusion***

In conclusion the code execution resulted in successful executions. Figures in the appendix show code execution including the registers being used including the data segment for each memory address.

***Appendix***

