# CS224

Lab 4
Section 2
Arman Engin Sucu
21801777

# PART 1

a)

| Location | Machine Instruction | Assembly Language Equivalent |
| --- | --- | --- |
| 00 | 0x20020005 | addi $v0, $zero, 5 |
| 04 | 0x2003000c | addi $v1, $zero, 12 |
| 08 | 0x2067fff7 | addi $a3, $v1, -9 |
| 0c | 0x00e22025 | or $a0, $a3, $v0 |
| 10 | 0x00642824 | and $a1, $v1, $a0 |
| 14 | 0x00a42820 | add $a1, $a1, $a0 |
| 18 | 0x10a7000a | beq $a1, $a3, 10 |
| 1c | 0x0064202a | slt $a0, $v1, $a0 |
| 20 | 0x10800001 | beq $a0, $zero, 1 |
| 24 | 0x20050000 | addi $a1, $zero, 0 |
| 28 | 0x00e2202a | slt $a0, $a3, $v0 |
| 2c | 0x00853820 | add $a3, $a0, $a1 |
| 30 | 0x00e23822 | sub $a3, $a3, $v0 |
| 34 | 0xac670044 | sw $a3, 68($v1) |
| 38 | 0x8c020050 | lw $v0, 80($zero) |
| 3c | 0x08000010 | j 0x0000010 |
| 40 | 0x001f6020 | add $t4, $zero, $ra |
| 44 | 0x0c000012 | jal 0x0000012 |
| 48 | 0xac020054 | sw $v0, 84($zero) |
| 4c | 0x00039042 | srl $s2, $v1, 1 |
| 50 | 0x03E00008 | jr $ra |

e)
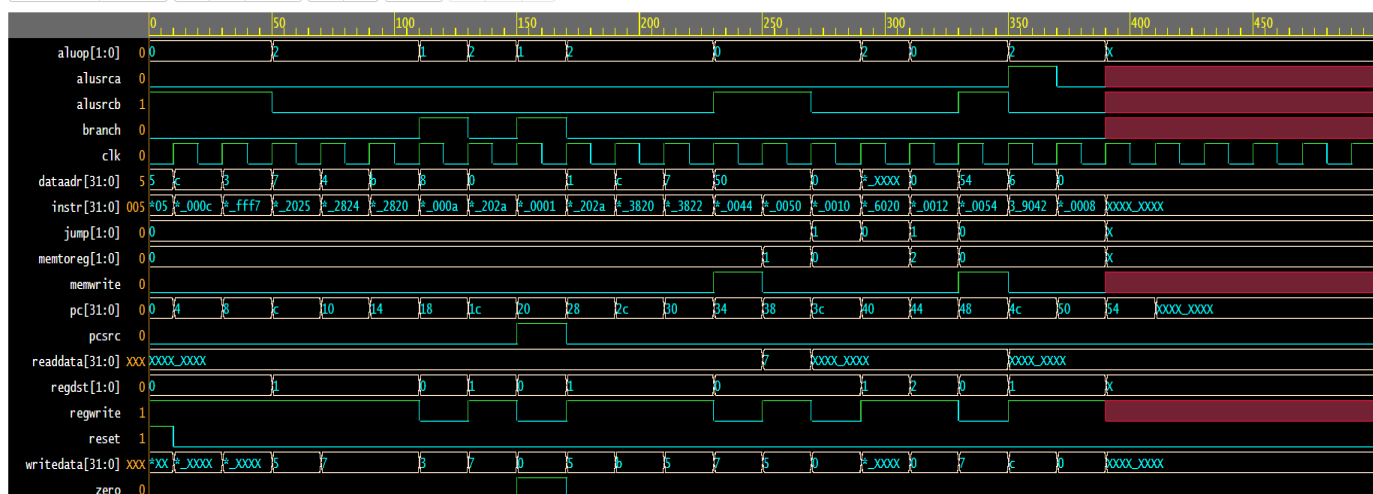
**f)**

i) It corresponds to data in the rt register

ii) Writedata undefined for some of the early instructions, since at the begining the value for registers can not be known. After writing datas to the registers, writedata(rt) value can be known.

iii) readdata only assigned for lw instruction, therefore, it is defined for only lw.

iv) It corresponds to aluot, however, for a R type instruciton dataadr corresponds to value that will be writen to rd register

v) dataaddress becomes undefined for add $t4, $zero, $ra, since $ra does not have any value in that time, therefore, aluout is undefined(dataadress).

**g)**

i) To support srlv additional changes are needed in ALU and imem.

ii) The module needs to be modified is ALUControl. I would modify it by changing $ALU_{[2:0]}$ to $ALU_{[3:0]}$, since to make ALU shift left there needs to be opened more room on ALU instructions.

# PART 2

## a)

subi: IM[PC]

RF[rt] ⬅ RF[rs] – SignExtended(immediate)

PC ⬅ PC + 4

## b)

subi instruction exactly the same with addi instruction except add instruction makes add in the alu, therefore, subi needs to make sub in the alu. This difference can be handled by ALUControl which is 110 for subtract.For those reasons no need any additional changes on the datapath. Moreover, the path fort he subi is showed in black:
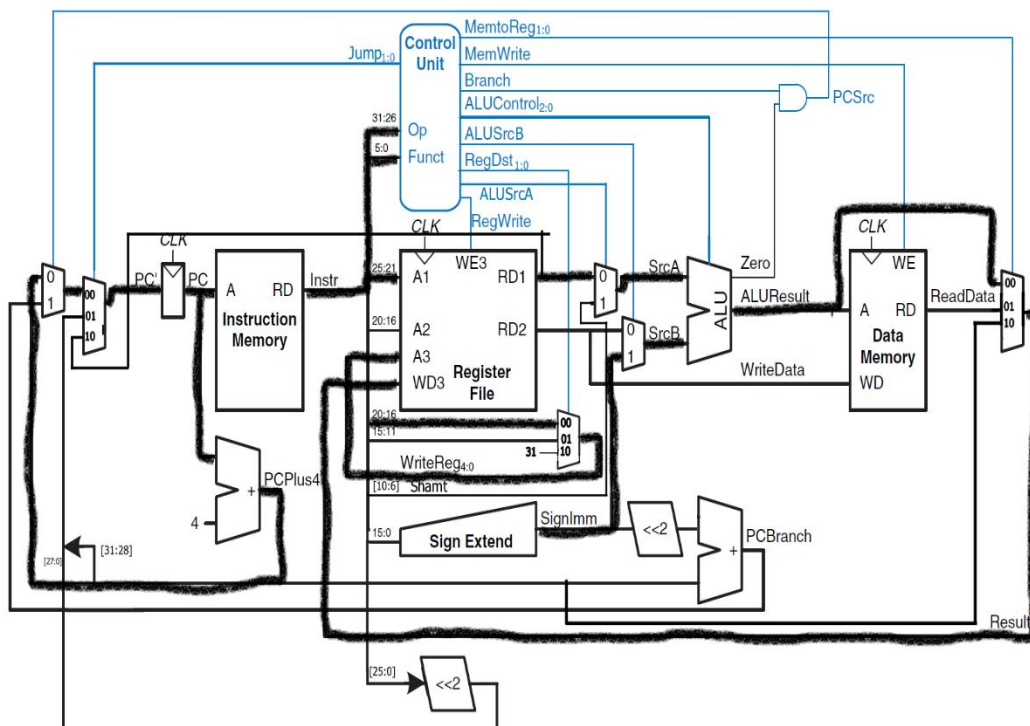


## c)

| Instruction | Opcode | RegWrite | RegDst | ALUSrcB | Branch | MemWrite | MemToReg | ALU Op | Jump |
|---|---|---|---|---|---|---|---|---|---|
| subi | 001100 | 1 | 00 | 1 | 0 | 0 | 00 | 01 | 0 |