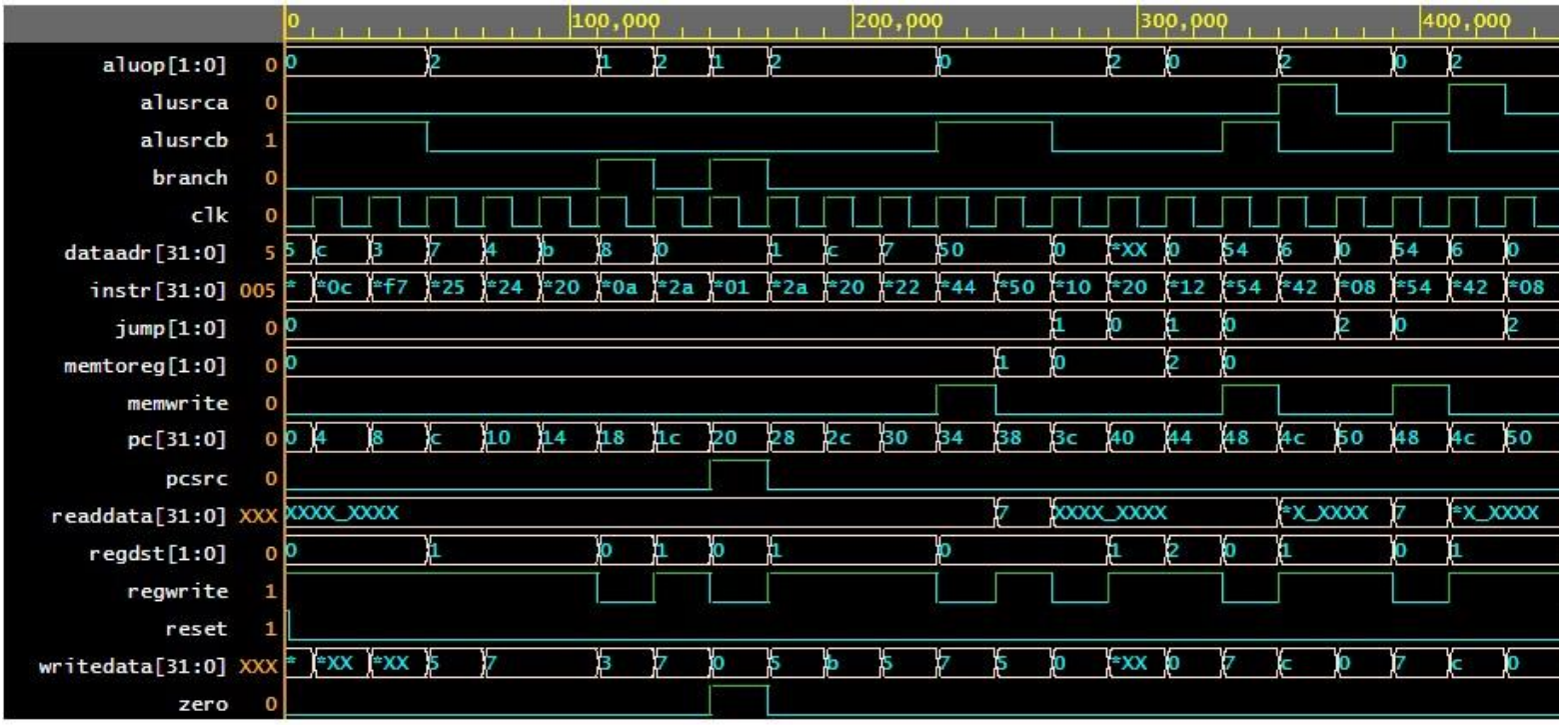
CS224
Lab 4
Preliminary Report
Section-1

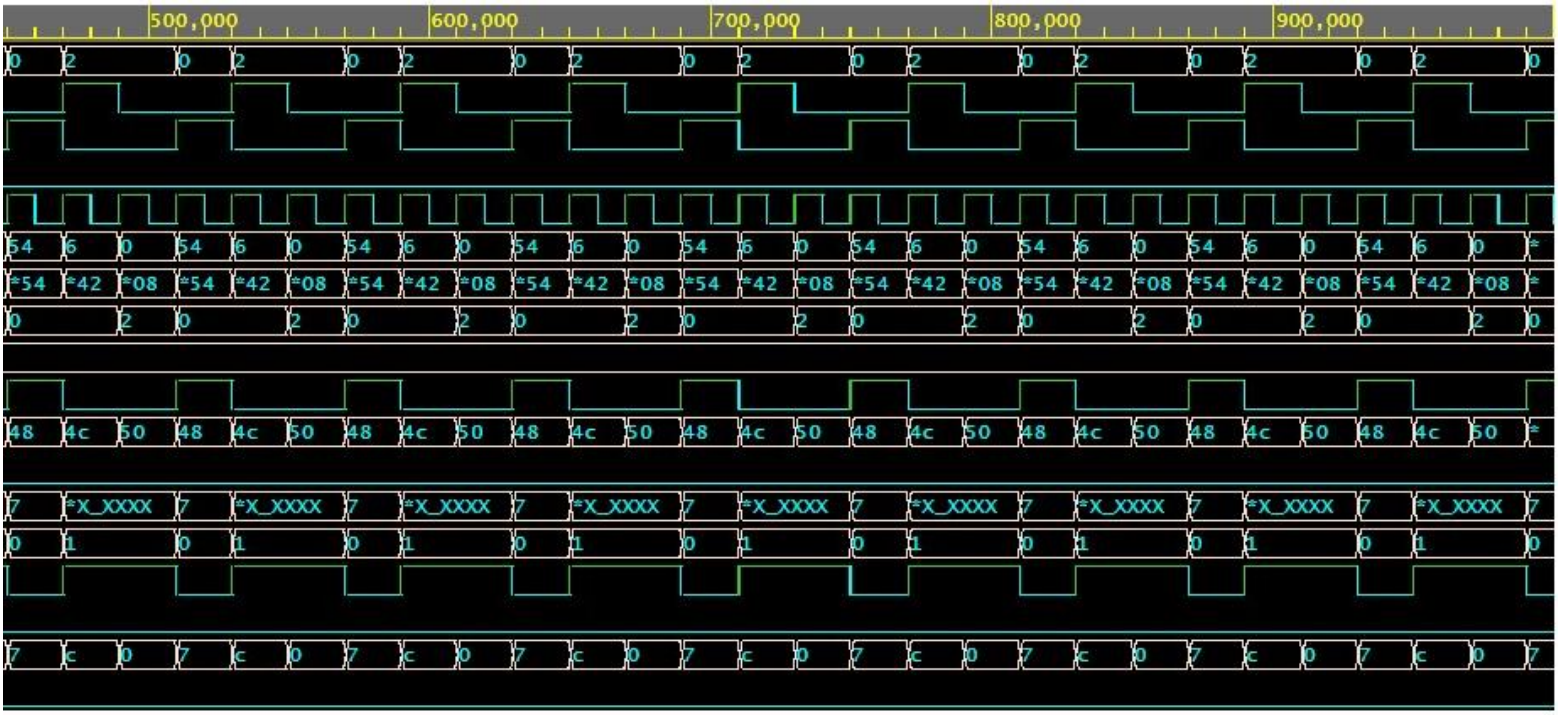
Ata Seren
21901575

Part 1.a

Location of Instruction (hex)	Machine Instruction (hex)	Assembly Language Equivalent
00	20020005	addi $v0, $0, 5
04	2003000c	addi $v1, $0, 12
08	2067fff7	addi $a3, $v1, 65527
0c	00e22025	or $a0, $a3, $v0
10	00642824	and $a1, $v1, $a0
14	00a42820	add $a1, $a1, $a0
18	10a7000a	beq $a1, $a3, 10
1c	0064202a	slt $a0, $v1, $a0
20	10800001	beq $a0, $0, 1
24	20050000	addi $a1, $0, 0
28	00e2202a	slt $a0, $a3, $v0
2c	00853820	add $a3, $a0, $a1
30	00e23822	sub $a3, $a3, $v0
34	ac670044	sw $a3, 68($v1)
38	8c020050	lw $v0, 80($0)
3c	08000010	j 3
40	001f6020	add $t4, $0, $ra
44	0c000012	jal 18
48	ac020054	sw $v0, 84($0)
4c	00039042	srl $s2, $v1, 1
50	03e00008	jr $ra

Part 1.e (I separated my waveform to make it visible)





Part 1.f

i) Writedata corresponds to value in destination register (rd in instruction).

ii) If destination register isn't used and saved a data by an instruction before, writedata becomes undefined in waveform.

iii) Readdata is mostly undefined because no data is read from memory to store or load in undefined parts.

iv) It corresponds to computational result of ALU.

v) Because instruction before the point dataaddress become undefined is J and since there are no data used in j, no value saved in dataaddress.

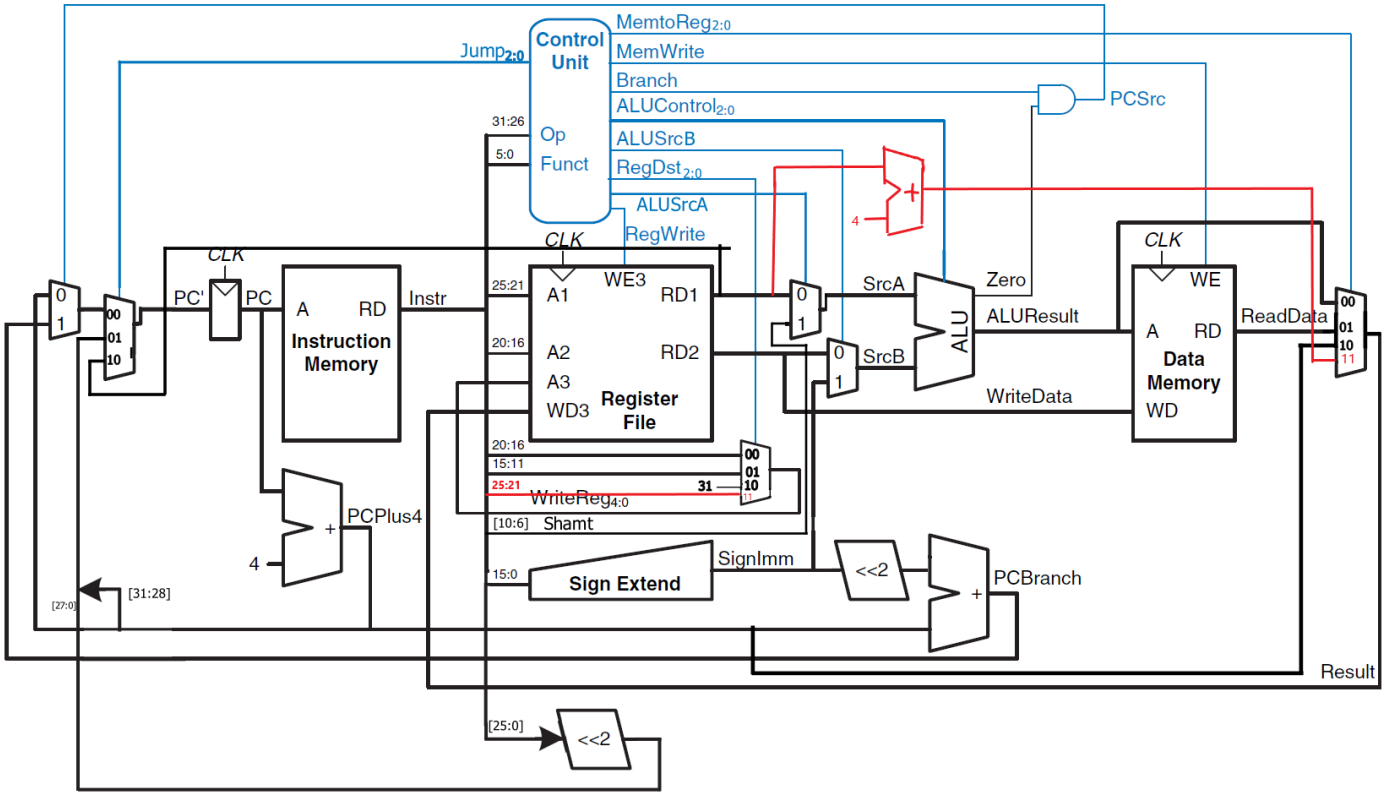
Part 1.g

i) I would have to change aludec module and create a new case in it which gets opcode 000110 which is different than others. Also, I would have to change alu module to get variable value.

ii) I need to change aludec module and alu module to create a new case. Then, in the case of alu module, I need to do opposite of srl, shifting left instead of right.

Part 2

```
sw+: IM[PC]
     DM[RF[rs]+SignExt(Imm16)] <= RF[rt]
     RF[rs] <= RF[rs]+4
     PC <= PC + 4
```



Instruction	Opcode	RegWrite	RegDst	ALUSrcA	ALUSrcB	Branch	MemWrite	MemToReg	ALUOp	Jump
R-type	000000	1	01	0	0	0	0	00	10	00
srl	000000	1	01	1	0	0	0	00	10	00
lw	100011	1	00	0	1	0	0	01	00	00
sw	101011	0	X	0	1	0	1	XX	00	00
beq	000100	0	X	0	0	1	0	01	01	00
addi	001000	1	00	0	1	0	0	00	00	00
j	000010	0	X	X	X	X	0	XX	XX	01
jal	000011	1	10	X	X	X	0	10	XX	01
jr	000000	1	01	0	0	0	0	00	10	10
sw+	001100	1	11	0	1	0	1	11	00	00