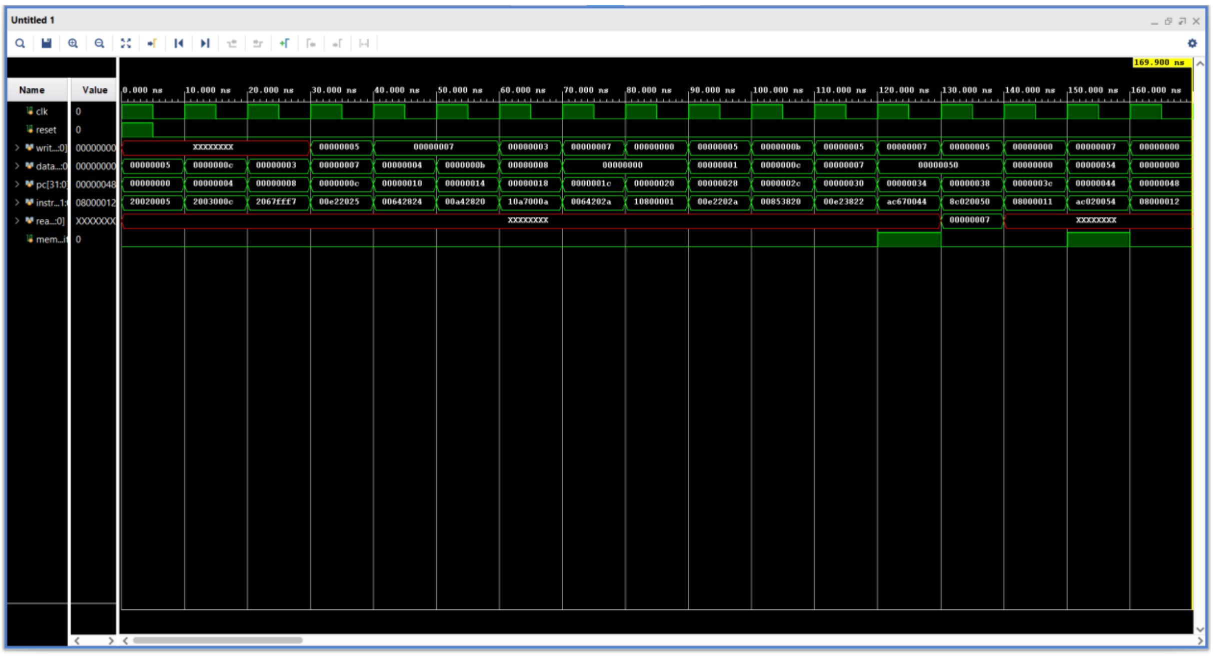
CS224
Lab No 4
Section 5
Alp Tuğrul Ağçalı
21801799

PART 1.A

Address	Machine Instruction	Assembly Code
0x00000000	0x20020005	addi $v0, $zero, 5
0x00000004	0x2003000c	addi $v1, $zero, 12
0x00000008	0x2067fff7	addi $a3, $v1, -9
0x0000000c	0x00e22025	or $a0, $a3, $v0
0x00000010	0x00642824	and $a1, $v1, $a0
0x00000014	0x00a42820	add $a1, $a1, $a0
0x00000018	0x10a7000a	beq $a1, $a3,0x000A
0x0000001c	0x0064202a	slt $a0 $v1 $a0
0x00000020	0x10800001	beq $a0, $zero, 0x0001
0x00000024	0x20050000	addi $a1, $zero, 0
0x00000028	0x00e2202a	slt $a0, $a3, $v0
0x0000002c	0x00853820	add $a3, $a0, $a1
0x00000030	0x00e23822	sub $a3, $a3, $v0
0x00000034	0xac670044	sw $a3, 44($v1)
0x00000038	0x8c020050	lw $v0, 50($zero)
0x0000003c	0x08000011	j 0x0000011
0x00000040	0x20020001	addi $v0, $zero, 1
0x00000044	0xac020054	sw $v0, 54($zero)
0x00000048	0x08000012	j 0x0000012

PART 1.D


Untitled 1
169.900 ns
Name
Value
0.000 ns
10.000 ns
20.000 ns
30.000 ns
40.000 ns
50.000 ns
60.000 ns
70.000 ns
80.000 ns
90.000 ns
100.000 ns
110.000 ns
120.000 ns
130.000 ns
140.000 ns
150.000 ns
160.000 ns
clk 0
reset 0
writ...:0] 00000000 XXXXXXXX 00000005 00000007 00000003 00000007 00000000 00000005 0000000b 00000005 00000007 00000005 00000000 00000007 00000000
data...:0 00000000 00000005 0000000c 00000003 00000007 00000004 0000000b 00000008 00000000 00000001 0000000c 00000007 00000050 00000000 00000054 00000000
pc[31:0] 00000048 00000000 00000004 00000008 0000000c 00000010 00000014 00000018 0000001c 00000020 00000028 0000002c 00000030 00000034 00000038 0000003c 00000044 00000048
instr...1: 08000012 20020005 2003000c 2067fff7 00e22025 00642824 00a42820 10a7000a 0064202a 10800001 00e2202a 00853820 00e23822 ac670044 8c020050 08000011 ac020054 08000012
rea...:0] XXXXXXXX XXXXXXXX 00000007 XXXXXXXX
mem...it 0

PART 1.E

i) In an R-type instruction what does writedata correspond to?
ii) Why is writedata undefined for some of the early instructions in the program?
iii) Why is readdata most of the time undefined?
iv) In an R-type instruction what does dataadr correspond to?
v) In which instructions memwrite becomes 1?

I) In R-type instructions write data is the data which comes from RD2 of RF and is going to go to MUX who choses wheter write-data or sign extended immediate goes to ALU.

II) Because they are not R type instructions or sw instructions. In these instructions write data is not needed because mux chooses the sign extended immediate. So, RF does not read the data in A2 port.(rt)

III) Becaue, read data is output of data memory and it is being used to read data which is going to be written to register by lw instruction.

IV) It equals to ALU result and does not go to the data Memory. It goes to MUX who choses the data which is going to be written in register A3(rd or rt).

V) sw.

PART 1.F

```
module alu(input  logic [31:0] a, b,
           input  logic [2:0]  alucont,
           output logic [31:0] result,
           output logic zero);

   always_comb
      case(alucont)
          3'b010: result = a + b;
          3'b110: result = a - b;
          3'b000: result = a & b;
          3'b001: result = a | b;
          3'b011: result = a << b;   // for pre part 1.F
          3'b111: result = (a < b) ? 1 : 0;
          default: result = {32{1'bx}};
      endcase

   assign zero = (result == 0 || result < 0) ? 1'b1 : 1'b0;
endmodule
```

PART 2.A

1. sraac

   IM[PC]

   RF[rs]<- RF[rs] >> RF[rt]

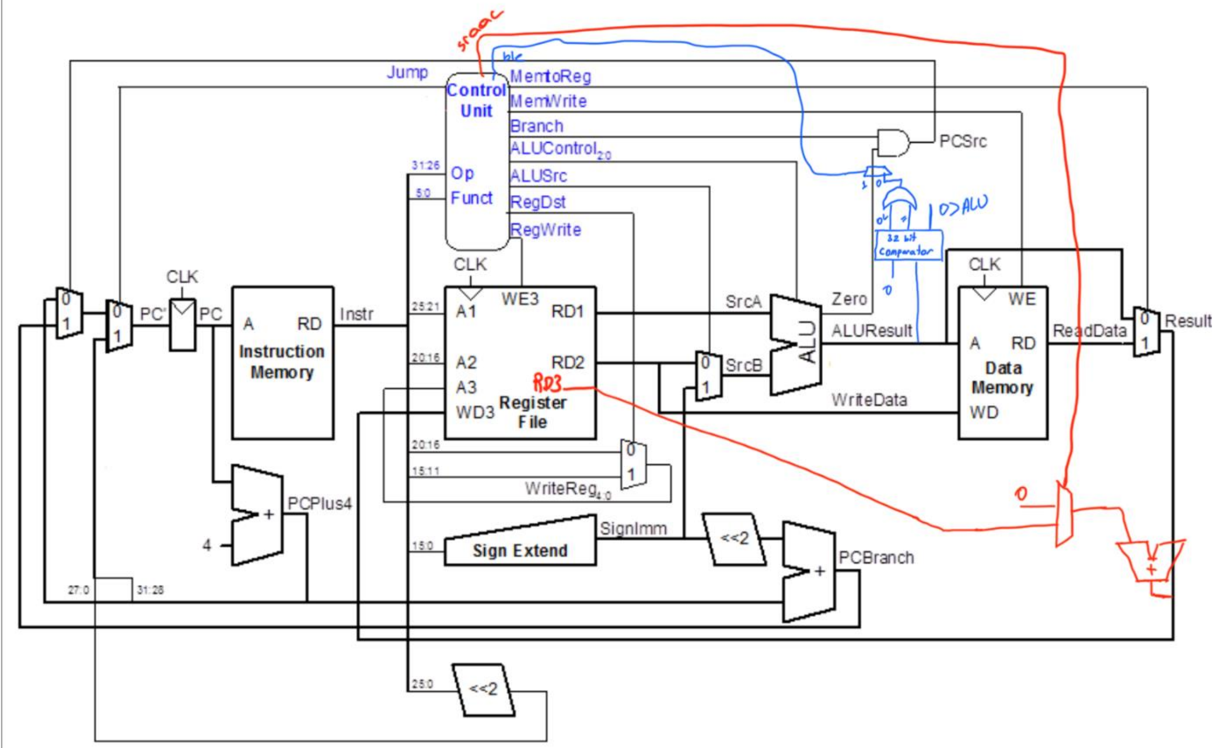
```
RF[rd]<- RF[rd] + RF[rs ]

PC <- PC+4
```

2. ble

```
IM[PC]
if( RF[rs] – RF(rt) == 0 | | ( RF[rs] – RF(rt) < 0)
        PC <- BTA
else
        PC <- PC + 4
```

PART 2.B



PART 2.C

Instruction	Opcode	RegWrite	RegDst	ALUSrc	Branch	MemWrite	MemToReg	ALUOp	Jump	sracc	Ble
R-type	000000	1	1	0	0	0	0	10	0	0	X
lw	100011	1	0	1	0	0	1	00	0	0	X
sw	101011	0	X	1	0	1	X	00	0	x	X
beq	000100	0	X	0	1	0	X	01	0	x	0
addi	001000	1	0	1	0	0	0	00	0	0	X
j	000010	0	X	X	X	0	X	XX	1	x	X
Sracc	000000	1	1	0	0	0	0	10	0	1	X
ble	000001	0	X	0	1	0	X	01	0	0	1

Table 1: Main Decoder for Original10

ALUOp	**Funct**	**ALUControl**
00	X	010 (add)
01	X	110 (subtract)
1X	100000 (add)	010 (add)
1X	100010 (sub)	110 (subtract)
1X	100100 (and)	000 (and)
1X	100101 (or)	001 (or)
1X	101010 (slt)	111 (set less than)
1X	000001(sracc)	011(shift right)

Table 2: ALU Decoder for Original10