
**2021–2022 SPRING SEMESTER**

**CS224 – LAB 04**

# MIPS Single-Cycle Datapath and Controller

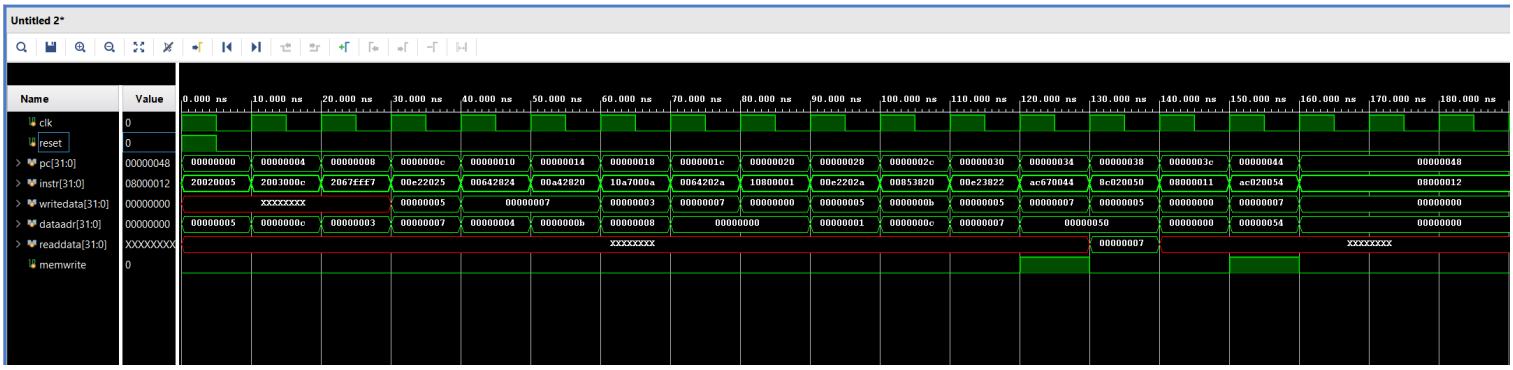
**NAME**: ABDULLAH RIAZ
**ID** : 22001296
**COURSE:**CS224
**SECTION:** 01
**DATE:** 03/04/2021

**Part1**

**a)**

| Location | Machine Instruction(hex) | Assembly Language Eqv. |
|---|---|---|
| 00 | 0x20020005 | addi $v0 $zero 0x0005 |
| 04 | 0x2003000c | addi $v1 $zero 0x000C |
| 08 | 0x2067fff7 | addi $a3 $v1 0xFFF7 |
| 0c | 0x00e22025 | or $a0 $a3 $v0 |
| 10 | 0x00642824 | and $a1 $v1 $a0 |
| 14 | 0x00a42820 | and $a1 $a1 $a0 |
| 18 | 0x10a7000a | beq $a1 $a3 0x000A |
| 1c | 0x0064202a | slt $a0 $v1 $a0 |
| 20 | 0x10800001 | beq $a0 $zero 0x0001 |
| 24 | 0x20050000 | addi $a1 $zero 0x0000 |
| 28 | 0x00e2202a | slt $a0 $a3 $v0 |
| 2c | 0x00853820 | add $a3 $a0 $a1 |
| 30 | 0x00e23822 | sub $a3 $a3 $v0 |
| 34 | 0xac670044 | sw $a3 0x0044 $v1 |
| 38 | 0x8c020050 | lw $v0 0x0050 $zero |
| 3c | 0x08000011 | j 0x0000011 |
| 40 | 0x20020001 | addi $v0 $zero 0x0001 |
| 44 | 0xac020054 | sw $v0 0x0054 $zero |
| 48 | 0x08000012 | j 0x0000012 |

**d)**




**e) i)** it corresponds to rd2 of the register file(R(rd) in Verilog)
**ii)** The early instructions are not r-type instructions therefore writedata is undefined in them
**iii)** lw is the only instruction that utilizes readdata and it is used only once in the instructions hence it is undefined most of the time
**iv)** It corresponds to Aluresult which is written to WD3 of the register file
**v)** Only sw instruction

**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'b011: result = a << b;
         3'b110: result = a - b;
         3'b000: result = a & b;
         3'b001: result = a | b;
         3'b111: result = (a < b) ? 1 : 0;
         default: result = {32{1'bx}};
      endcase

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

**Part 2**

**a)**

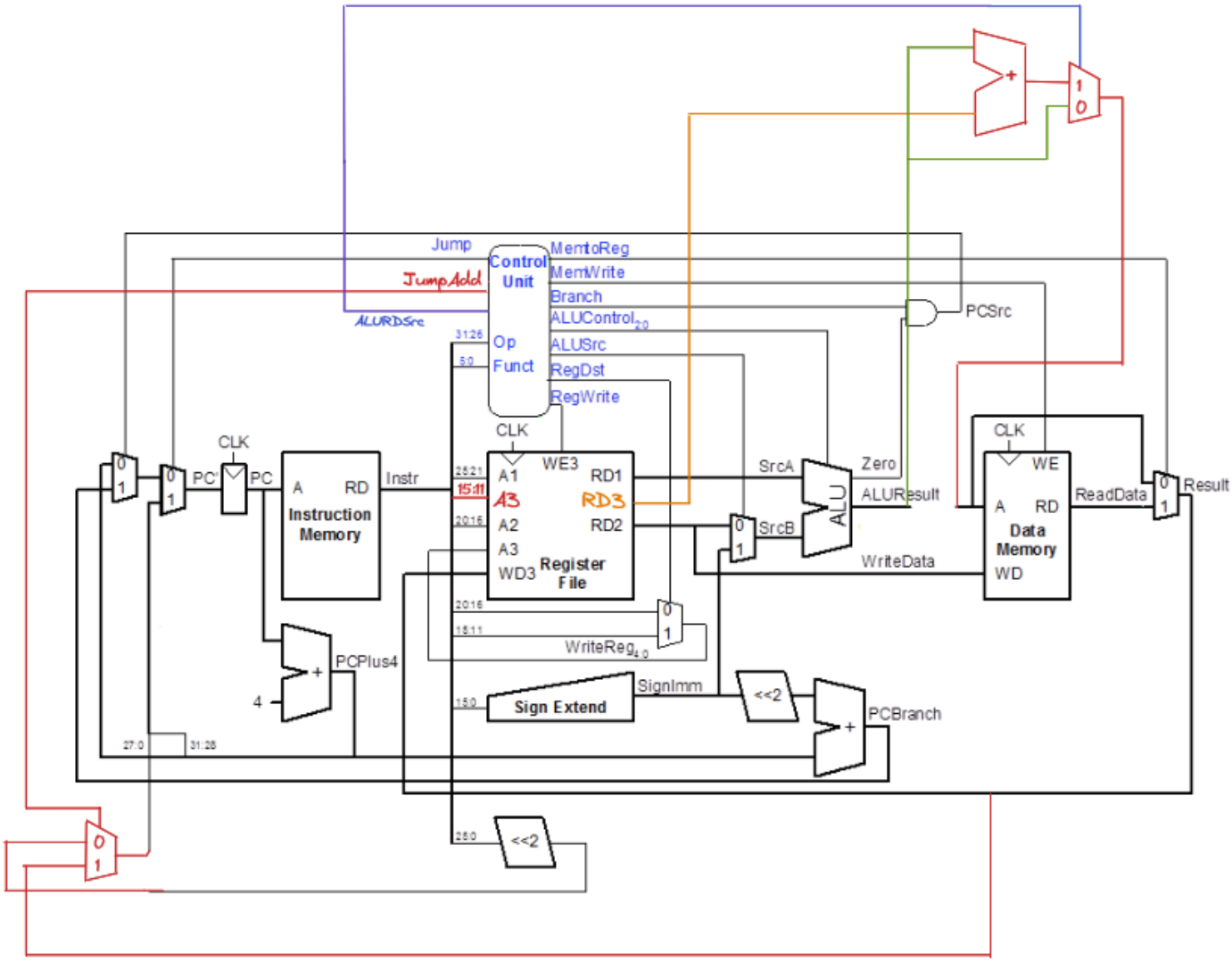
**sracc :**

IM[PC]
RF[rd] ← RF[rd] + (RF[rs] >> RF[rt])
PC ← PC + 4

**jm:**

IM[PC]
PC ← DM[RF[rs]] + SignExt(immed)]

**b)**



**c)**

| Instruction | Opcode | RegWrite | RegDst | ALUSrc | Branch | MemWrite | MemToReg | ALUOp | Jump | JumpAdd | ALURDSrc |
|---|---|---|---|---|---|---|---|---|---|---|---|
| R-type | 000000 | 1 | 1 | 0 | 0 | 0 | 0 | 10 | 0 | 0 | 0 |
| lw | 100011 | 1 | 0 | 1 | 0 | 0 | 1 | 00 | 0 | 0 | 0 |
| sw | 101011 | 0 | X | 1 | 0 | 1 | X | 00 | 0 | 0 | 0 |
| beq | 000100 | 0 | X | 0 | 1 | 0 | X | 01 | 0 | 0 | 0 |
| addi | 001000 | 1 | 0 | 1 | 0 | 0 | 0 | 00 | 0 | 0 | 0 |
| j | 000010 | 0 | X | X | X | 0 | X | XX | 1 | 0 | 0 |
| sracc | 000001 | 1 | 1 | 0 | 0 | 0 | 0 | 10 | 0 | 0 | 1 |
| jm | 100111 | 0 | X | 1 | X | 0 | 0 | 00 | 1 | 1 | 0 |

| 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 | 000010 (srl) | 100 |