CS224

Lab 4

Section 6

Deniz Semih Özal

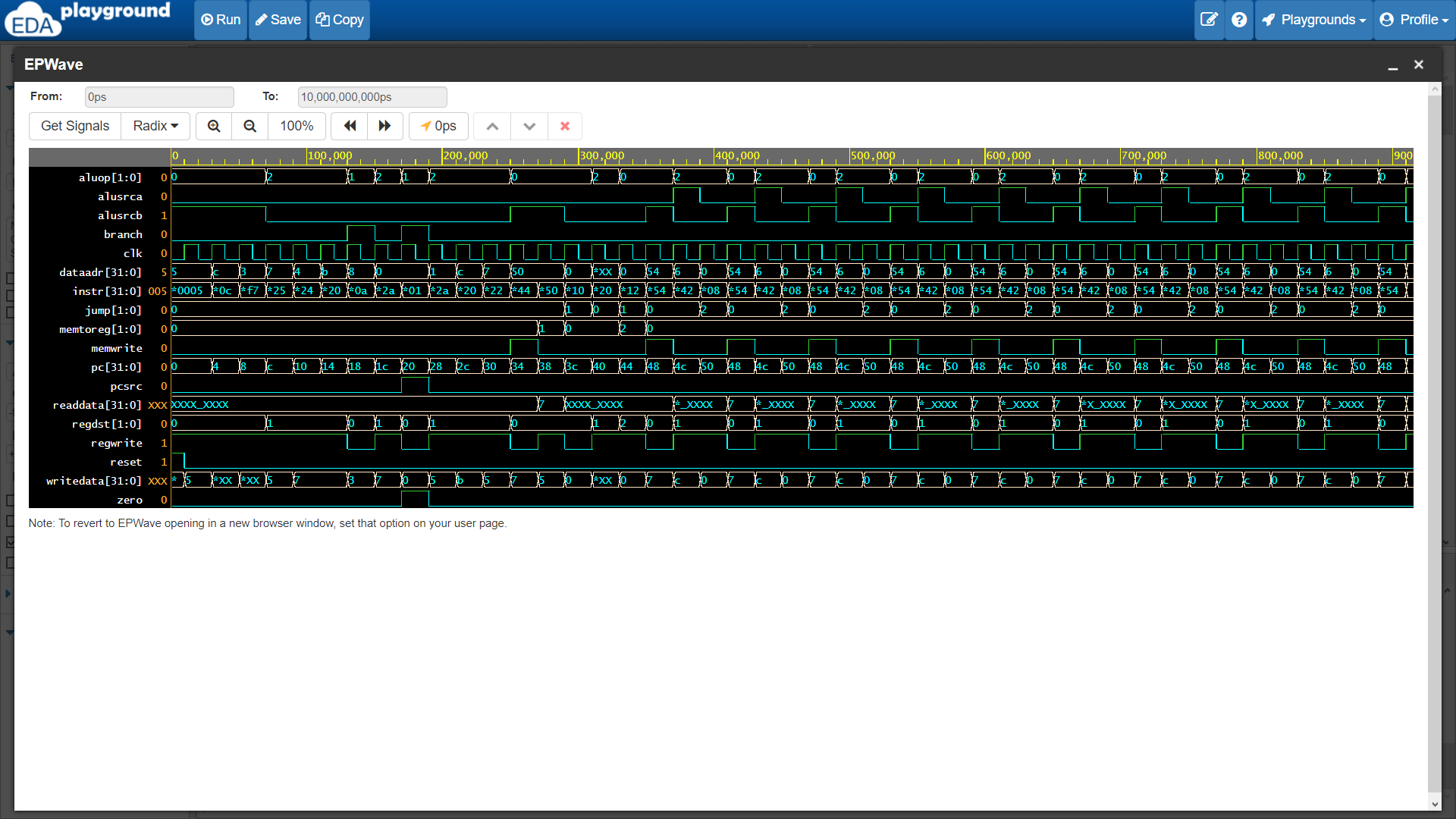
21802414

**PART 1**

A)Disassembled Version of Instructions

|  |  |  |
| --- | --- | --- |
| **Location** | **Instruction (Machine Code)** | **Assembly** |
| 0x00 | 0x20020005 | addi $v0, $0, 5 |
| 0x04 | 0x2003000c | addi $v1, $0, 12 |
| 0x08 | 0x2067fff7 | addi $a3, $v1, -9 |
| 0x0C | 0x00e22025 | or $a0, $a3, $v0 |
| 0x10 | 0x00642824 | and $a1, $v1, $a0 |
| 0x14 | 0x00a42820 | add $a1, $a1, $a0 |
| 0x18 | 0x10a7000a | beq $a1, $a3, 0xA |
| 0x1c | 0x0064202a | slt $a0, $v1, $a0 |
| 0x20 | 0x10800001 | beq $a0, $0, 0x1 |
| 0x24 | 0x20050000 | addi $a1, $0, 0 |
| 0x28 | 0x00e2202a | slt $a0, $a3, $v0 |
| 0x2c | 0x00853820 | add $a3, $a0, $a1 |
| 0x30 | 0x00e23822 | sub $a3, $a3, $v0 |
| 0x34 | 0xac670044 | sw $a3, 0x44($v1) |
| 0x38 | 0x8c020050 | lw $v0, 0x50($0) |
| 0x3c | 0x08000010 | j 0x10 |
| 0x40 | 0x001f6020 | add $t4, $0, $ra |
| 0x44 | 0x0c000012 | jal 0x12 |
| 0x48 | 0xac020054 | sw $v0, 0x54($0) |
| 0x4c | 0x00039042 | srl $s2, $v1, 1 |
| 0x50 | 0x03E00008 | jr $ra |

E) Waveform of Top Level MIPS



**F) Questions**

i) In an R-type instruction writedata corresponds to the result of ALU, and since 00 goes to the register aluout is the answer.

ii) ALU control sends signals that are corresponding to the function and its default state is 3’bxxx.

iii)Because there is no lw instruction until 0x38 so data in memory is not read until that time.

iv) Again it corresponds to AlU result, so aluout is the answer.

V) During jump instruction AlU and register file is not used so that aluout will be undefined, thus dataaddress will be undefined.

**G)** i) We need to connect Shamt into the register file and read corresponding data in shamt section, then proceed with the ALU part.

ii) To support sll instruction, a function and ALU control singal for sll could be added since we need to change ALU decoder.

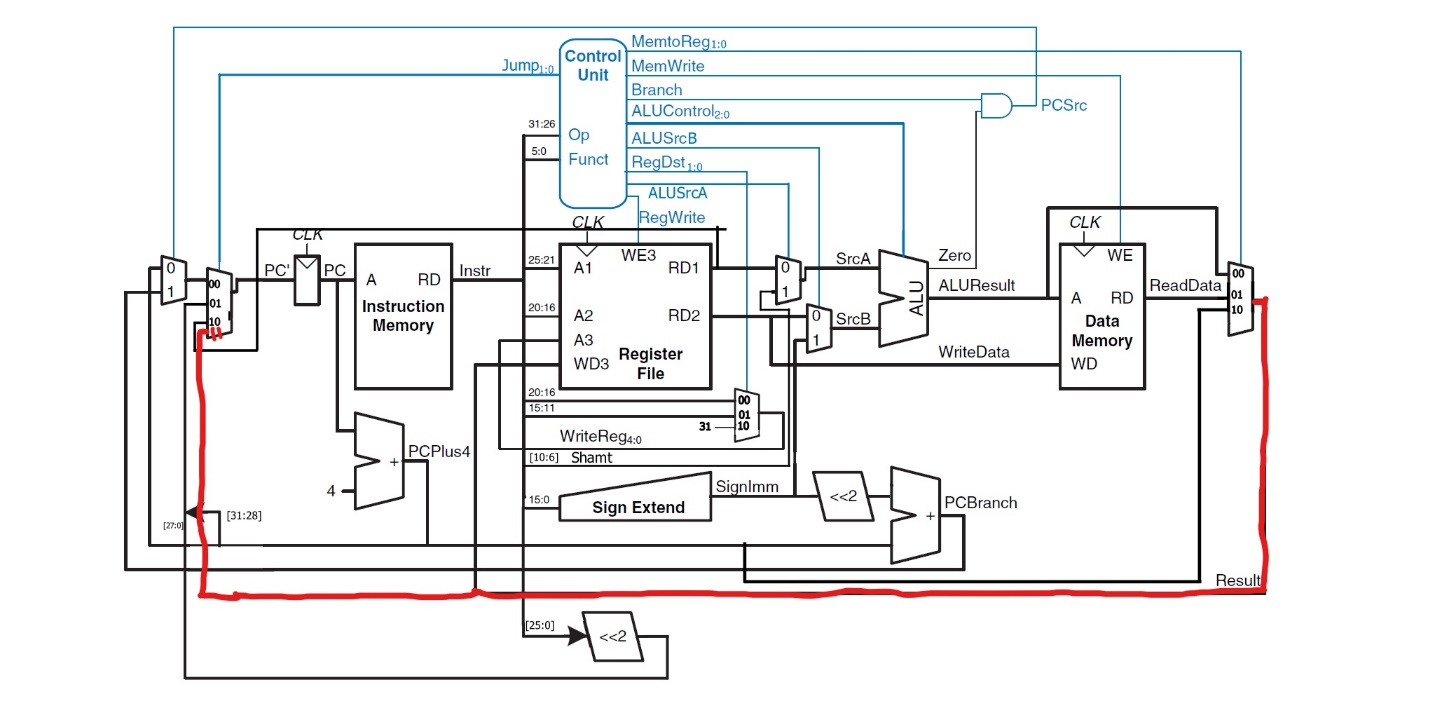
**Part 2**

a) RTL Expression

IM[PC]

PC <- RF[rs] + Signext(immed)

b) Final Datapath



c) Final Main Decoder Table

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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 |
| jm | 000111 | 1 | 00 | 0 | 1 | X | 0 | 00 | 00 | 11 |
|  |  |  |  |  |  |  |  |  |  |  |