COMPUTER ARCHITECTURE

LAB #2 Assignment April 21, 2018

(Hassan Elseoudy 3780 / Merna Adel 4168 / Bavli George 3750 / Mahmoud Mohamed 2718)

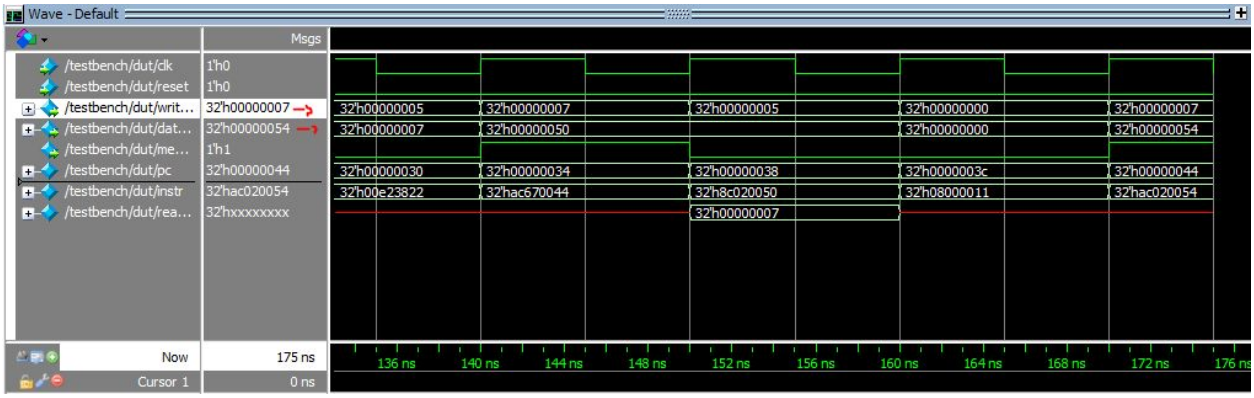
1. Please indicate how many hours you spent on this lab. This will not affect your grade (unless omitted), but will be helpful for calibrating the workload for next semester's labs.

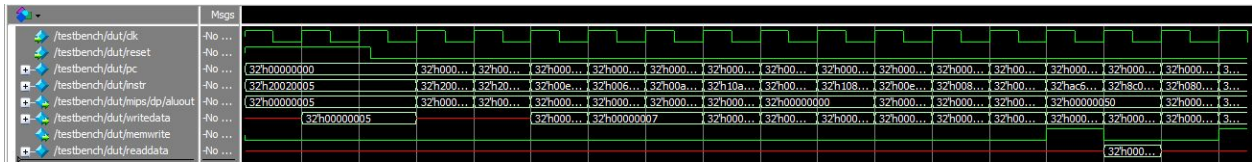
Answer : ~10 hours

2. A completed version of Table 1.

Note (X) -> not allowed in this instruction. Link

3. An image of the simulation waveforms showing correct operation of the processor. Does it write the correct value to address 84? Answer : yes





4. Marked up versions of the datapath schematic and decoder tables that adds the ori and bne instructions.

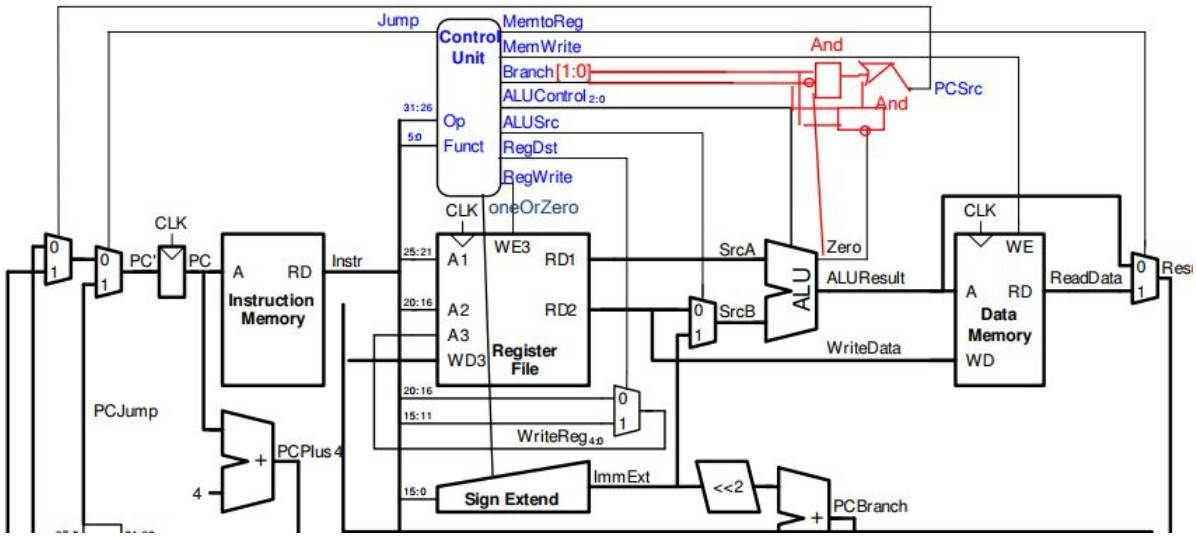
Extended functionality. Main Decoder:

Instruction	$Op_{5:0}$	RegWrite	RegDst	AluSrc	Branch	MemWrite	MemtoReg	$ALUOp_{1:0}$	Jump	One/Z		
R-type	000000	1	1	0	0 0	0	0	10	0	0		
lw	100011	1	0	1	0 0	0	1	00	0	0		
sw	101011	0	X	1	0 0	1	X	00	0	0		
beq	000100	0	X	0	1 0	0	X	01	0	0		
addi	001000	1	0	1	0 0	0	0	00	0	0		
j	000010	0	X	X	X X	0	X	XX	1	0		
ori	001101	1	0	1	0 0	0	0	11	0	1		
bne	000101	0	X	0	11	0	X	01	0	0		

Extended functionality. ALU Decoder:

$ALUOp_{1:0}$	Meaning
00	Add
01	Subtract
10	Look at `funct` field
11	ORI

```
assign pcsrc = (~branch[0] & branch[1] & zero) | (branch[0] & branch[1] & ~zero);
```



5. Your SystemVerilog code for your modified MIPS computer (including ori and bne functionality) with the changes highlighted and commented in the code.

Project Link : https://github.com/Hassan-Elseoudy/MIPS-SingleCycle

```
module signext( input logic [15:0] a,
        input logic oneOrZero,
        output logic [31:0] y);
    assign y = oneOrZero? {{16'b0},a} : {{16{a[15]}},a};
endmodule
```

```
6'b000000: controls <= 11'b11000000100; // RTYPE
6'b100011: controls <= 11'b10100010000; // LW
6'b101011: controls <= 11'b00100100000; // SW
6'b000100: controls <= 11'b00010000010; // BEQ
6'b001000: controls <= 11'b10100000000; // ADDI
6'b000010: controls <= 11'b00000001000; // J
6'b001101: controls <= 11'b10100000111; // ORI
6'b000101: controls <= 11'b00011000010;  // BNE
default:   controls <= 11'bxxxxxxxxxxx; // illegal op
```

```
case(aluop)
    2'b00: alucontrol <= 3'b010;    // add (for lw/sw/addi)
    2'b01: alucontrol <= 3'b110;    // sub (for beq/bne)
    2'b11: alucontrol <= 3'b001;    // or (for ori)
    default:
        case(funct)         // R-type instructions
            6'b100000: alucontrol <= 3'b010; // add
            6'b100010: alucontrol <= 3'b110; // sub
            6'b100100: alucontrol <= 3'b000; // and
            6'b100101: alucontrol <= 3'b001; // or
            6'b101010: alucontrol <= 3'b111; // slt
            default: alucontrol <= 3'bxxx;   // ???
```

6. The contents of your memfile2.dat containing your test2 machine language code

```
34088000
20098000
350a8001
11090005
0128582a
15600001
08000009
01485022
350800ff
016a5820
01484022
ad680052
```