Bilkent University
Computer Science
CS224 - Computer Organization

Design Report
Lab # 4
Section 6

Osama Tanveer
21801147

17 April 2020

Part 1
b)

Location	Machine Instruction	Assembly Language Equivalent
00	0x20020005	addi $v0, $zero, 5
04	0x2003000c	addi $v1, $zero, 12
08	0x2067fff7	addi $a3, $v1, -9
0c	0x00e22025	or $a0, $a3, $v0
10	0x00642824	and $a1, $v1, $a0
14	0x00a42820	add $a1, $a1, $a0
18	0x10a7000a	beq $a1, $a3, 10
1c	0x0064202a	slt $a0, $v1, $a0
20	0x10800001	beq $a0, $zero, 1
24	0x20050000	addi $a1, $zero, 0
28	0x00e2202a	slt $a0, $a3, $v0
2c	0x00853820	add $a3, $a0, $a1
30	0x00e23822	sub $a3, $a3, $v0
34	0xac670044	sw $a3, 68 ($v1)
38	0x8c020050	lw $v0, 80 ($zero)
3c	0x08000011	j 0x0000011
40	0x20020001	addi $v0, $zero, 1
44	0xac020054	sw $v0, 84 ($zero)
48	0x08000012	j 0x0000012

c)

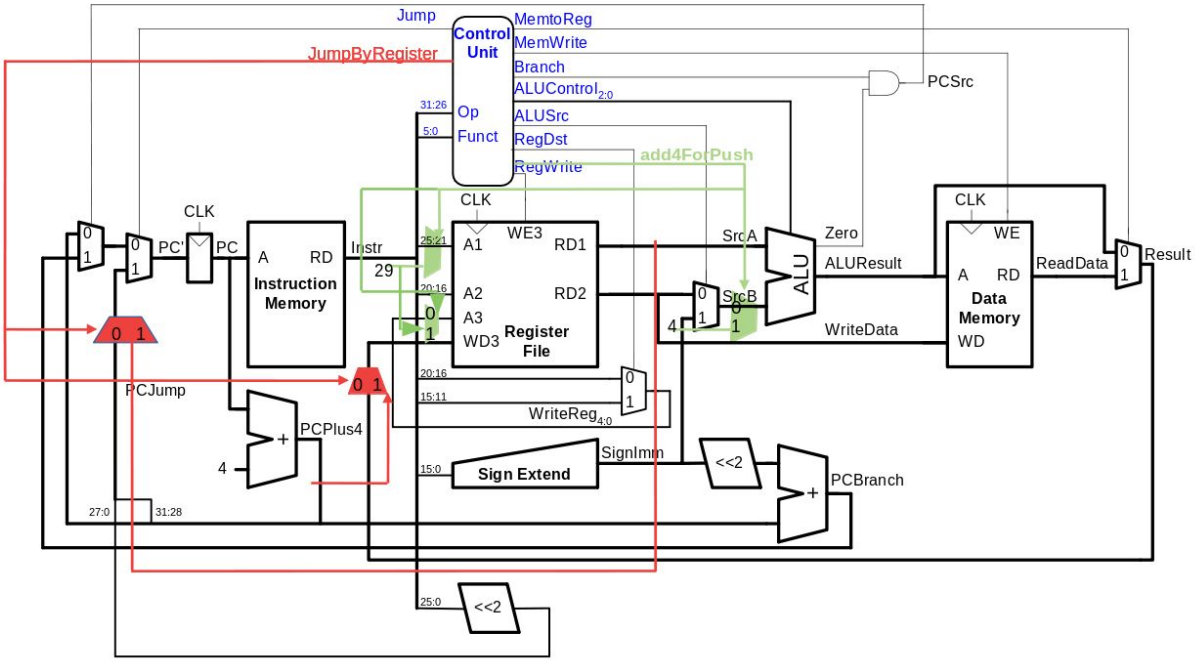
Full RTL for jalr
IM[PC]
RF[rd]← PC + 4
PC ← RF[rs]

Full RTL for push
IM[PC]
RF[sp] ←RF[sp] - 4
DM[sp] ←RF[rs]

d)

Jalr (brick red lines in datapath):
A new signal from the control unit is added, namely “JumpByRegiter”. To perform a jalr instruction, JumpByRegister should be set to 1 and in all other cases it should be set to 0. To change the datapath, 2 multiplexers are added. One of the multiplexers chooses between “PCJump” and the jump address to jump to according to the jalr instruction. The other multiplexer chooses between “Result” and “PCPlus4”.

Push (green lines in datapath):
For this instruction, add4ForPush is added. This signal acts as a select signal between “SrcB” and 4. This is used to subtract 4 from $sp, to store a new value on the stack. The rest of the instruction is handled by the datapath already present after the addition of jalr.



e)

Instruction	Opcode	RegWrite	RegDst	ALUSrc	Branch	MemWrite	MemtoReg	ALUOp	Jump	JumpBy Register	add4ForPush
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
jalr	000000	1	1	0	X	0	X	XX	1	1	0
push	000001	1	X	X	0	1	0	X1	0	0	1

f)

 # First 10 instructions to test the Orignal10
addi $v0, $zero, 5
addi $v1, $zero, 12
or $a0, $v0, $v1
and $a1, $v0, $v1
add $a2, $a1, $a0
beq $a1, $a2, 2
slt $a3, $v0, $v1
sub $t0, $v0, $v1
sw $a3, 68($v1)
lw $t7, 68($v1)

 # Setting the value of stack pointer at a preallocated position in the data memory
addi $sp, $zero, 63
push $v1 **# First push instruction**
push $v1 **# Second push instruction for confirmation of the validity of the implemented instruction**
jalr $v0, $a1 **# JALR instruction is at the end because it loops back to the 2nd instruction**