Jong Park
CS 472 – Computer Architecture
Professor Lizhong Chen
HW1
April 17, 2019

Problem 1

Block	Delay (ps)
I-Mem	450
Add	100
Mux	20
ALU	150 + 130
Regs	250
D-Mem	550
Control blocks	100
Sign-extend	50

Notes:

- It's a single-cycle processor (One instruction per cycle).
- Addition of MUL to ALU => +130 ps latency & fewer instructions executed for MUL
- This is based on load instruction.
  - L_{total} = I-Mem + max(Reg, sign, control) +Mux + ALU + D-Mem + Mux + Reg (write)

1.1) What is the clock cycle time with and without this improvement? (10 pts)

Without improvement: 450 + 250 + 20 + 150 + 550 + 20 + 250 = **1690 ps**

With improvement: 450 + 250 + 20 + 280 + 550 + 20 + 250 = **1820 ps**

1.2) If this improvement can result in 15% fewer number of instructions executed as we no longer need to emulate the MUL instruction, what is the speedup achieved by adding this improvement? Does this increase or decrease the performance? (10 pts)

1690/(1820 * 0.85) = **1.0924**

This increases the performance by **9.24%**

1.3) If we want to achieve a speedup that is ≥ 1.3 by adding this MUL operation to ALU, what percentage of instructions should be reduced at least? (10 pts)

1690 ps / x = 1.3

1690 ps / 1.3 = 1300 ps

1-(1690/1300) ≈ **23.08% instructions reduction required**

Problem 2

Initial register values:

r2	r3	r4	r5	r6	r7	r8	r9	r10	r11	r31
2	0	1	6	2	4	12	7	3	8	0

```
loop1: addi   r11, r10, 4          # x5
       sw     r2, 16(r7)           # x5
       nor    r3, r2, r6           # x5
loop2: lw     r4, 0(r2)            # x5 x7
       lw     r8, 0(r3)            # x5 x7
       add    r5, r4, r8           # x5 x7
       nor    r6, r3, r5           # x5 x7
       sw     r6, 0(r2)            # x5 x7
       addi   r11, r11, -1         # x5 x7
       bne    r11, r31, loop2      # x5 x7
       addi   r9, r9, -1           # x5
       bne    r9, r2, loop1        # x5
```

2.1) Please list the breakdown percentage of executed instructions (e.g., add: 10%) (18 pts)

op	add	addi	bne	lw	sw	nor
Count/total	35 / 270	45 / 270	40 / 270	70 / 270	40 / 270	40 / 270
percentage	12.96%	16.67%	14.81%	25.93%	14.81%	14.81%

2.2) In what fraction of all cycles is the data memory used; in what fraction of all cycles is the instruction memory used? (7 pts)

D-Mem is used by lw and sw: 25.93% + 14.81% = **40.74%**

I-Mem is used by all operations: **100%**

2.3) In what fraction of all cycles is the input of the sign-extend circuit needed? What is this circuit doing in cycles in which its output is not needed? (10 pts)

Sign-extended: addi + bne + lw + sw = 12.96% + 14.81% + 25.93% + 14.81% = **72.22%**

Problem 3

Initial register values:

r0	r1	r2	r3	r4	r5	r6	r8	r11	r18
125	-219	-56	79	134	-2	15	99	141	211

Instruction: 000000 01000 00011 01011 00000 100100

R-type r8 r3 r11 shift:0 AND

AND r11, r3, r8 #

3.1) What are the outputs of the sign-extend and the “Shift left 2” unit (near the top of Figure 1) for this instruction word? (5 pts)

Sign-extend	Shift-left-2
000000 00000 00000 01011 00000 100100	000000 00000 00000 01011 00000 100100

3.2) What are the values of the ALU control unit's inputs for this instruction? (5 pts)

ALUOp	Instruction[5-0]
10	100100

3.3) What is the new PC address after this instruction is executed (assuming the current value of PC is pc)? Highlight the path through which this value is determined (5 pts)

New PC = PC + 4

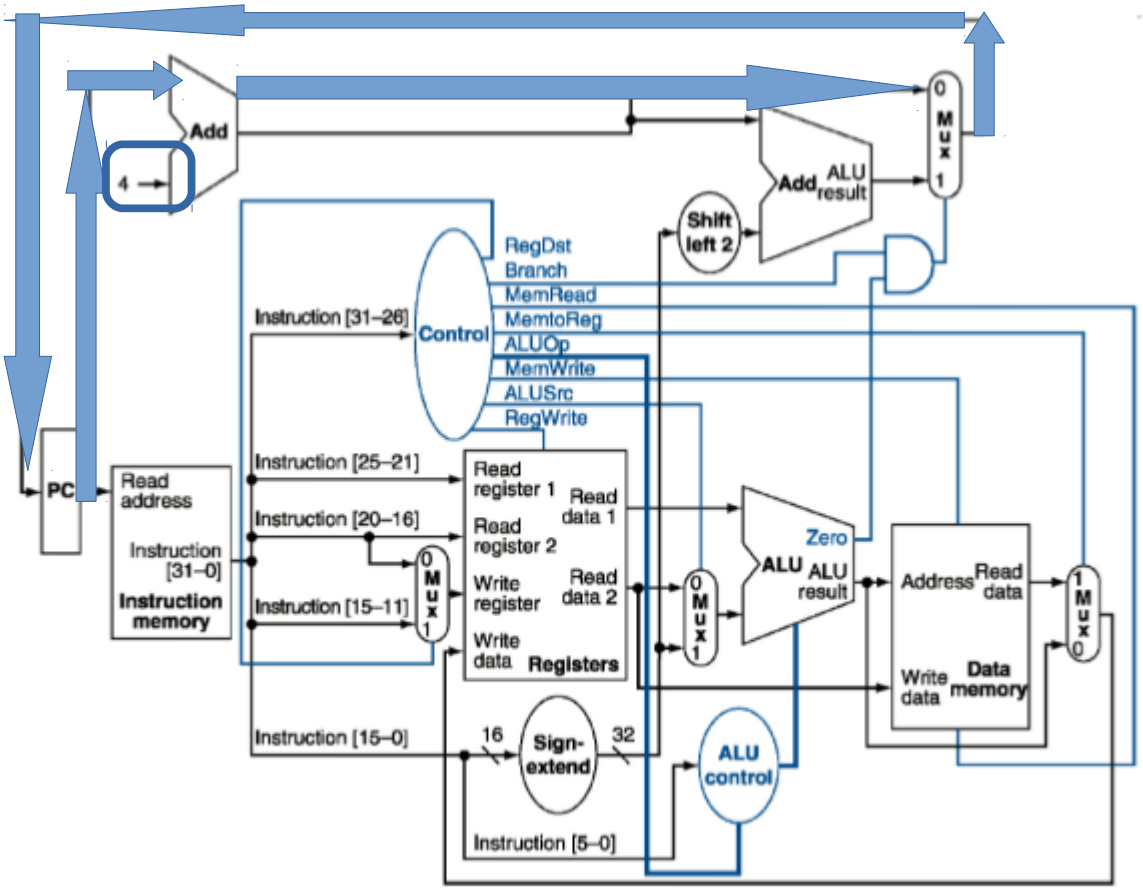

Figure 1. The basic implementation of the MIPS subset, including the necessary multiplexers and control lines

3.4) For each Mux, show the values of its data output during the execution of this instruction (5 pts)

Mux Output (Selection)	Write Register (1)	ALU (0)	Write Data (Register)(0)	PC (0)
Values	01011 (r11)	79	67 (= 99&79)	PC + 4

3.5) For the ALU and the two add units, what are their data input values? (5 pts)

Logic Unit	ALU		Add (PC+4)		Add (Branch)	
Input values	99	79	PC	4	PC +4	000000 00000 00000 01011 00000 100100

3.6) What are the values of all inputs for the "Registers" unit? (10 pts)

Register Type	Read 1	Read 2	Write Register	Write Data	RegWrite
Value	01000 (r8)	00011 (r3)	01011 (r11)	67	TRUE