CS 1541
Homework 1 Answer Key | Total 20 points

Question 1: (5 points – 1(a) + 2(b) + 2(c))

Consider a MIPS program with the following mix of instructions

45%	R-type arithmetic instructions
25%	lw instructions
15%	sw instructions
15%	beq instructions

a. Assume that the instruction mix for this program is typical of all the programs that execute on a 2.5 GHz machine in which the R-type instructions execute in 5 cycles, the lw and sw instructions in 6 cycles and the beq instructions in 3 cycles. What is the average CPI (clock per instruction) for that machine?

CPI: (0.45)(5 cycles)+ (0.25)(6 cycles)+ (0.15)(6 cycles)+ (0.15)(3 cycles)
= 2.25+1.5+0.9+0.45 =5.1

b. Assume that it is possible to increase the frequency of execution to 2.7 GHz by changing the architecture such that beq executes in 4 cycles rather than 3 cycles. Would you recommend that change? Justify your answer.

New CPI: (0.45)(5 cycles)+ (0.25)(6 cycles)+ (0.15)(6 cycles)+ (0.15)(4 cycles)
= 2.25+1.5+0.9+0.6 = 5.25

At 2.7 GHz, the average time to execute an instruction is $\frac{5.25}{2.7}$ = 1.94 n.sec. Without the modification, the clock rate is 2.5 MHz and the average time to execute an instruction is $\frac{5.1}{2.5.}$ = 2.04 n.sec. Hence, this change leads to faster execution and should be recommended.

c. Assume that it is possible to modify the compiler such that the total number of executed instructions is increased by 5% but that the new instruction mix is changed to 50%, 20%, 15% and 15% for R-type, lw, sw and beq instructions, respectively. Would you recommend this modified compiler for any of the two machines (the 2.5 and 2.7 GHz machines)?

For the 2.5GHz machine,
CPI: (0.5)(5 cycles)+ (0.2)(6 cycles)+ (0.15)(6 cycles)+ (0.15)(3 cycles)
= 2.5+1.2+0.9+0.45 =5.05

The cycle time = $\frac{5.05}{2.5}$ = 2.02 n.sec
For the 2.7GHz machine,
CPI: (0.5)(5 cycles)+ (0.2)(6 cycles)+ (0.15)(6 cycles)+ (0.15)(4 cycles)
= 2.5+1.2+0.9+0.6 =5.20

The cycle time = $\frac{5.20}{2.7}$ = 1.92 n.sec

With a 5% reduction in the number of instructions (represented as x), both would be proportionally affected:

For 2.5 GHz machine,
Old exec. time: $(2.04\ x)$ n.sec
New exec. time: $2.02 * 1.05\ x = (2.12\ x)$ n.sec

For 2.7 GHz machine,
Old exec. time: $(1.94\ x)$ n.sec
New exec. time: $1.92 * 1.05\ x = (2.016\ x)$ n.sec

In both cases, the execution time is larger. Hence, I would not recommend the modified compiler for both machines

Question 2: (5 points – 1.5 lw, 1 add, 1 sub, 1.5 bne)

Determine the format for each instruction and the decimal values of each instruction field for the following program segment:

```
Loop: lw      $t1, 26 ($s1)
      add     $s2, $t1, $s2
      sub    $s1, $s1, $t2
      bne     $s1, $t3, Loop
```

Give also the machine code instructions (32 bits per instruction) for that program segment (follow the format given in slide 15 of the first lecture for each instruction). See Appendix A.10 of the textbook for op-codes of operands.

Loop: lw $t1, 26($s1)//$t1 = Mem[$s1 + 26], I-type format instruction

	op (6 bits)	rs (5 bits)	rt (5 bits)	immediate (16 bits)
lw	35	17	10	48
M. code	100011	10001	01010	0000000000110000

add $s2, $t1, $s2 //$s2 = $t1 + $s2, R-type format instruction

	op (6 bits)	rs (5 bits)	rt (5 bits)	rd (5 bits)	Shamt (5 bits)	Funct (6 bits)
add	0	10	9	9	0	32
M. code	000000	01010	01001	01001	00000	100000

sub $s1, $s1, $t2 //$s1 = $s1 + $t2, R-type format instruction

	op (6 bits)	rs (5 bits)	rt (5 bits)	rd (5 bits)	Shamt (5 bits)	Funct (6 bits)
add	0	17	18	17	0	32
M. code	000000	10001	10010	10001	00000	100000

bne $s1, $t3, Loop //if ($s1!=0) branch to 'Loop', I-type format instruction

	op (6 bits)	rs (5 bits)	rt (5 bits)	immediate (16 bits)
bne	5	17	0	-4 (offset to Loop)
M. code	000101	10001	00000	11111111 11111100 (2's complement)

Question 3: (5 points – 3(a) + 2(b))

In this question, we will augment the single cycle processor architecture that implements R-type, lw, sw and beq instructions to add a new type of instructions,

R-type-rm $r1, $r2, $r3

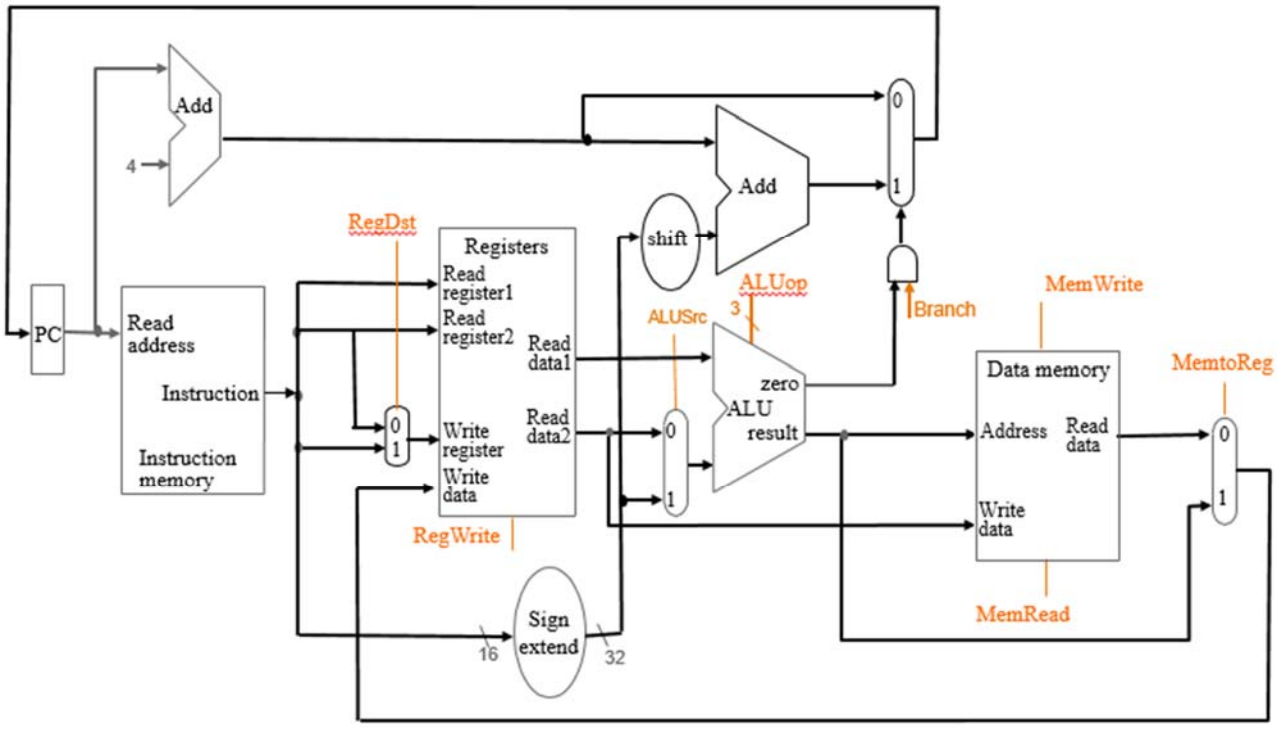
This instruction type is similar to the R-type instructions except that the second operand is found in the memory location whose address is found in $r3, which is different than the "R-type $r1, $r2, $r3" instructions, where the second operand is found in $r3. Note that in both the "R-type" and $R-type-rm" instructions, the first operand is found in $r2 and the result is stored in $r1. In other words the new instruction, "R-type-rm $r1, $r2, $r3", is equivalent to "lw $t1, 0 ($r3)" follwed by "R-type $r1, $r2, $t1".

a. Using this figure, modify the data paths to add the new instruction type. Produce a new figure, either by modifying the ppt file or just drawing the new data paths and muxes on top of print-outs of the figure. Your new figure should name any new control signals that you need to add (if any).
b. Specify the values of the control signals (the ones already in the figure plus the new ones that you may have to introduce) that are needed for the execution of the new instruction type. You do not have to specify the ALUop control signal since the figure uses only one ALUop signal as opposed to the two ALUop1 and ALUUop2 signals specified in Table 4.18 of the textbook.

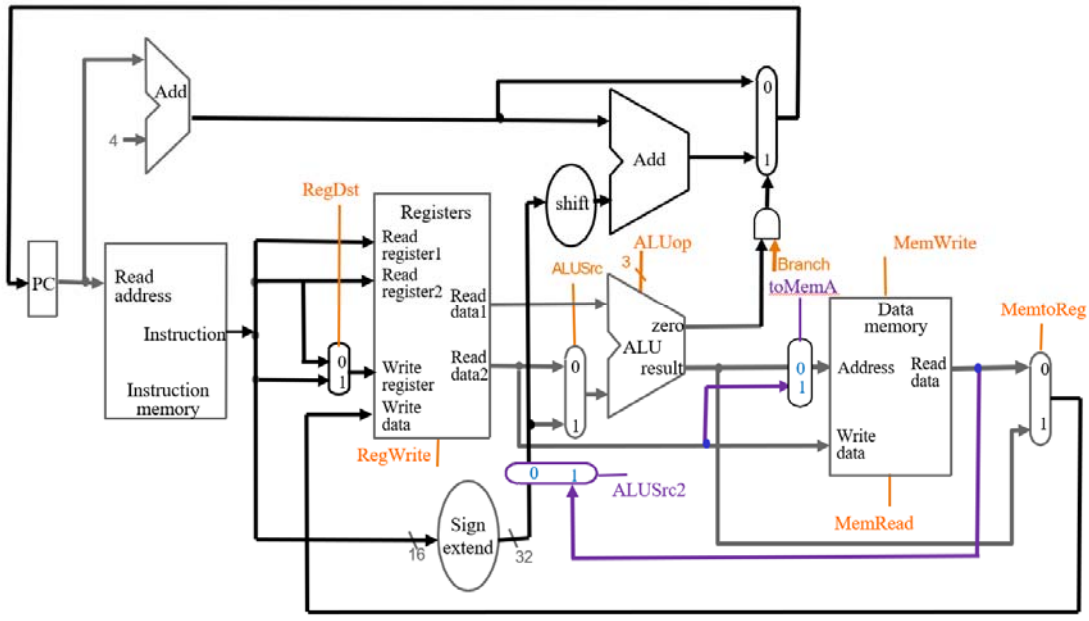
NOTE that the order of the registers in the assembly language instruction is different from the order in the machine (binary) instruction. Specifically, in a machine instruction, $r2 and $r3 are stored in bits 25-21 and 20-16, respectively, while $r1 is stored in bits 15-11.

CS 1541
Homework 1 Answer Key | Total 20 points

Original Figure:



Modified Figure (changes in purple):



CS 1541
Homework 1 Answer Key | Total 20 points

Control signals and values:

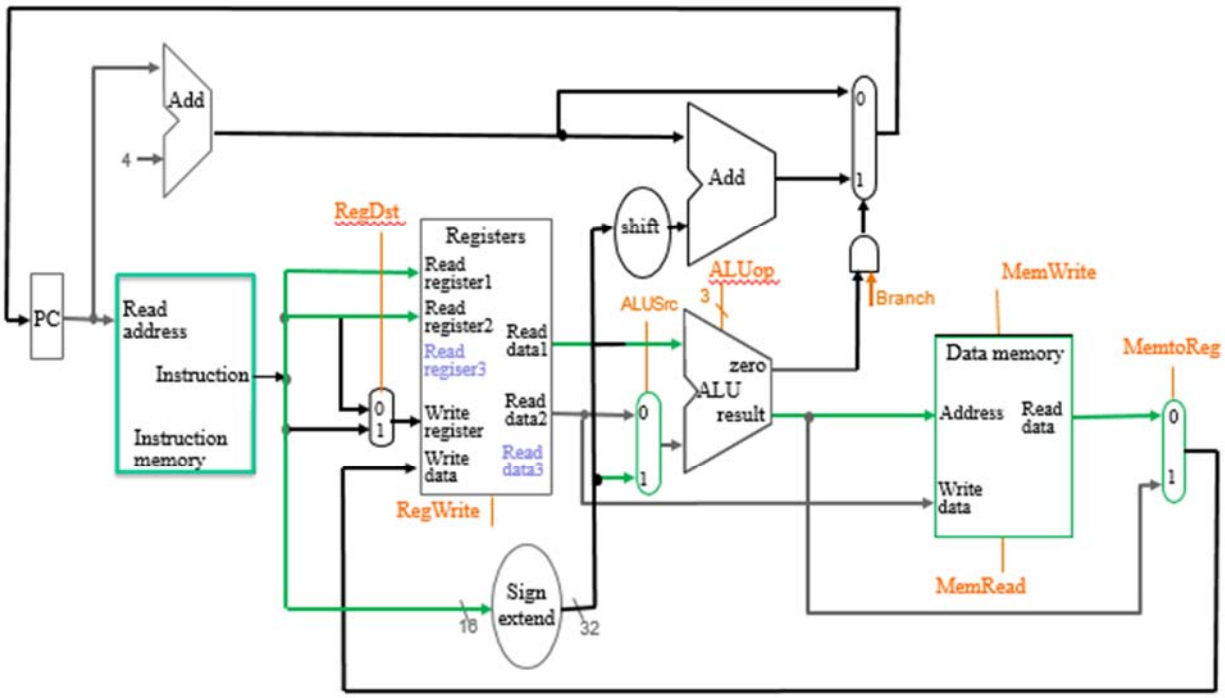
RegDst	Reg Write	ALUSrc	Branch	toMemA	toMemD	Mem Write	MemRead	MemToReg
-	0	0	0	1	0	1	0	-

Question 4: (5 points – 2(a) + 1(b) + 1(c) + 1(d))

In this question, we examine how latencies of individual components of the processor datapath affect the clock cycle time. For this question, assume the following latencies for logic blocks in the datapath and assume that blocks not listed have zero latencies:

I-Mem: 200ps	Add: 100ps	ALU: 180ps
Regs read: 95ps	Regs write: 95ps	D-Mem: 200ps
PC read: 10ps	Mux: 10ps	Sign-extend: 5ps

a. Given that the delay for the lw instruction is the longest among the R-type, lw, sw and beq instructions, the delay for lw determines the cycle time. Compute the delay for the lw instruction (and hence the cycle time) for the architecture shown in this figure.



There are 2 parallel paths, that for the PC fetch, and the lw instruction – the latter path is highlighted in green

PC path: PC read + Add + Mux = 10 + 100 + 10 = 120ps

lw instruction: PC read + I-mem + Regs read +ALU + D-mem + Mux(MemToReg) + Regs write = 10 + 200+ 95 + 180 + 200 + 10 + 95 = 790 ps

Hence, the delay for lw is 790 ps.

Note that the Sign Extend (5ps) + Mux(ALUSrc) path runs in parallel to the Reg reads and hence does not need to be added to the total delay

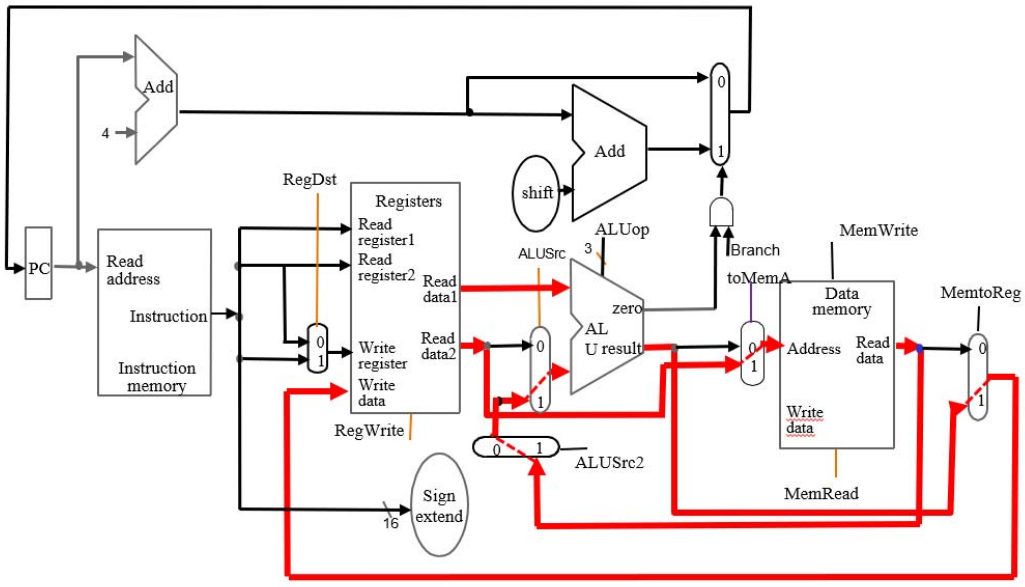
b. Compute the delays for the new instruction that you added in Question 3.

R-type-rm: PC read + I-mem + Regs read + Mux(toMemA) +D-Mem + Mux(ALUSrc2) + Mux(ALUSrc)+ALU + Mux(MemtoReg) + Reg write = 10 + 200 + 95 + 10 + 200 + 10 + +10 + 180 + 10 + 95 = 820 ps

c. How will the delays for the R-type, lw, sw and beq instructions change after the modifications that you made in Question 3 to add the R-type-m instruction?

Only 1 Mux has been added to the critical path before the data memory, and so the additional delay for the lw and sw instructions is 10ps. No additional delays are added to the R-type and beq instructions since the added muxes are not on the data paths for these instructions.

d. What is the cycle time for the modified architecture of Question 3?



Based on the answers to (a) and (c), we can recalculate the cycle time for the Rtype-rm instruction in the modified architecture as 820 ps