

Name: _____ Email: _____

Problem 1. Show all calculations for the following questions.

Assume an add takes 2 cycle if no dependency and if dependant then 4 clocks

Assume there is a 25% data dependency.

Assume a branch takes 7 cycle if true prediction and if false prediction then 2 clocks.

Assume that 10% of the branches are mispredicted.

1a (5%) What is the average add instruction time in clocks?

1b (5%) What is the average branch instruction time in clocks?

Problem 2. Assume a simple **6 stage** pipeline with the following execution times

1	IF	Instruction fetch	5 ns
2	ID	Register Read 1st part	1 ns
3	IR	Register Read 2nd part	2 ns;
4	EX	ALU	4 ns Branch decision made here
5	MEM	Data Access	6 ns,
6	WB	Register Write	3 ns

This computer has the following instructions:

Instruction	Operation
sub \$rd, \$rs, \$rt	\$rd = \$rs - \$rt
beq \$rs, \$rt, disp16	pc = pc+2+(\$rs-\$rt=0?disp16:0)
lw \$rt, addr16(\$rs)	\$rt = Mem[addr16+\$rs]
sw \$rt, addr16(\$rs)	Mem[addr16+\$rs]=\$rt

2a (20%) Fill in the following tables

Instruction	IF	ID	IR	EX	MEM	WB	Total Time	Multi-Cycles	Instruction Mix
sub									30%
beq									10%
lw									40%
sw									20%

2b (15%) Fill the following table and show work.

Instruction	Clock frequency	CPI (no hazards)	MIPS (no hazards)
Single-cycle CPU			
Multi-cycle CPU			
Pipeline CPU			

The following parts refer to the pipelined machine only

2c (15%) For the following code: Assume no forwarding and no branch prediction.

Draw lines showing all the data dependencies and show the pipeline sequence (IF, ID, IR, EX, M, WB)

Note: Branch decision is made in the EX stage.

Time	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
lw \$3, 0(\$4)																
beq \$1, \$3, loop																
sub \$4, \$4, \$5																

Problem 3: The RISCEE5 computer is a multi-cycle architecture for the machine instructions of problem 1a. **Use X for Don't Care.** Assume parts a, b, c, and d are independent of each other. Assume the 8 bit memory system is smart and loads the proper 16 bits in the IR register in one memory read cycle. There is only a single **8 bit register** (i.e. accumulator, called A). The PC and alu are also eight bits wide.

Note: You can remove and keep the RISCEE5 diagrams from the exam.

Remember: anything AND with zero is always zero. Anything OR with one is one.

There is only one instruction format shown as follows:

Opcode	Data8 or Address8 field
8 bits	8 bits
15-8	7 - 0

Machine Instruction	Operation	Description
clear	A = 0	Clear accumulator
load addr8	A = Memory[addr8]	Load accumulator from memory
jptr addr8	PC = Memory[addr8]	Jump to memory pointer
sll3	A = A << 3	Shift Left Logical accumulator by three

3a (5%) Fill in the settings of the control lines needed for the “clear” instruction.

Clock Step	Mem Write	Mem Read	IorD	IR write	MDR write	P0	BZ	PC src	ALU op	ALU srcA	ALU srcB	Reg Write	Reg Dst
T ₁													
T ₂													
T ₃													
T ₄													
T ₅													

3b (10%) Fill in the settings of the control lines needed for “**load**” *from memory* instruction.

Clock Step	Mem Write	Mem Read	lorD	IR write	MDR write	P0	BZ	PC src	ALU op	ALU srcA	ALU srcB	Reg Write	Reg Dst
T ₁													
T ₂													
T ₃													
T ₄													
T ₅													
T ₆													
T ₇													

3c (10%) Fill in the settings of the control lines needed for “**jp_{tr}**” instruction.

Clock Step	Mem Write	Mem Read	lorD	IR write	MDR write	P0	BZ	PC src	ALU op	ALU srcA	ALU srcB	Reg Write	Reg Dst
T ₁													
T ₂													
T ₃													
T ₄													
T ₅													
T ₆													
T ₇													

3d (15%) Fill in the settings of the control lines needed for “**sl_{l3}**” instruction.

Clock Step	Mem Write	Mem Read	lorD	IR write	MDR write	P0	BZ	PC src	ALU op	ALU srcA	ALU srcB	Reg Write	Reg Dst
T ₁													
T ₂													
T ₃													
T ₄													
T ₅													
T ₆													
T ₇													
T ₈													
T ₉													
T ₁₀													

Extra Credit (25%): **For x1, x2, and x3: refer to Problem 2 for pipeline description.**

x1 (5%) Using **no** forwarding, Draw lines showing all the data dependencies in “Time” column. and show the 6-stage pipeline sequence (IF, ID, IR, EX, M, WB) for the following code

Time		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
sub	\$4, \$1, \$2																
sw	\$4, 4(\$2)																
sub	\$5, \$4, \$2																
lw	\$3, 8(\$5)																

x2 (5%) Using **no** forwarding, Draw lines showing all the data dependencies in “Time” column. (hint: “sw” instruction has a new register dependency) and show the 6-stage pipeline sequence (IF, ID, IR, EX, M, WB) for the following code

Time		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
sub	\$4, \$1, \$2																
sw	\$4, 4(\$4)																
sub	\$5, \$4, \$2																
lw	\$3, 8(\$5)																

x3 (5%) Using forwarding, Draw lines showing all the data dependencies in “Time” column. (hint: “sw” instruction), and show the 6-stage pipeline sequence (IF, ID, IR, EX, M, WB) for the following code and draw lines showing the forwarding.

Time		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
sub	\$4, \$1, \$2																
sw	\$4, 4(\$4)																
sub	\$5, \$4, \$2																
lw	\$3, 8(\$5)																

x4 (10%) Refer to Problem 3. Remove the datapath between ALU and the **PCSrc** mux selection number “1”. Fill in the settings of the control lines needed for the “**clear**” instruction.

Clock Step	Mem Write	Mem Read	lorD	IR write	MDR write	P0	BZ	PC src	ALU op	ALU srcA	ALU srcB	Reg Write	Reg Dst
T ₁													
T ₂													
T ₃													
T ₄													
T ₅													
T ₆													
T ₇													