EECS 314 Test 3 Thusday April 7, 2005

Name: _____ Email: _____

Problem 1. Assume a simple **6 stage** pipeline with the following execution times

1	IF	Instruction fetch	2 ns
2	ID	Register Read	3 ns; Read register $rs; **Branch decision made here**
3	IT	Register Read	1 ns; Read register $rt;
4	EX	ALU	9 ns;
5	MEM	Data Access	7 ns,
6	WB	Register Write	8 ns

This computer has the following instructions:

Instruction	Operation
clear $rd	$rd = 0
bz $rs, disp16	pc = pc+2+($rs=0?disp16:0)
lw $rt, addr16($rs)	$rt = Mem[addr16+$rs]
sll $rt, $rs,$rt	$rt = $rs << $rt

Alternate 2

8 10 2

1a (12%) Fill in the following tables

Instruction	IF	ID	IT	EX	MEM	WB	Total Time	Multi-Cycles	Instruction Mix
clear	2			9		8	19	3	25%
bz	2	3				8	5	2	25%
sll	2	3	1	9		8	23	5	25%
lw	2	3		9	7	8	29	5	25%

1b (9%) Fill the following table and show work.

Instruction	Clock frequency	CPI (no hazards)	MIPS (no hazards)
Single-cycle CPU	$\frac{1}{29} = 34.48$MHz	1	$\frac{34.48}{1} = 34.48$
Multi-cycle CPU	$1/9 = 111.1$MHz	① $(3+2+5+5)\cdot.25 = 3.75$ or ② $(2+2+5+5)\cdot.25 = 3.5$	$\frac{111.1}{3.75} = 29.626$ (or) $\frac{111.1}{3.5} = 31.743$
Pipeline CPU	$1/9 = 111.1$MHz	1	$\frac{111.1}{1} = 111.1$

1c (5%) Given two CPU designs, (a) CPU A has clock of 3.6Ghz and 1000 MIPS, (b) CPU B has clock of 2.5 Ghz and 1500 MIPS: Which is better and why?

① B has better MIPS Number

② Clock frequency is not a appropriate benchmark for determining the true performance of the machine

Problem 2: The RISCEE8 computer is a multi-cycle architecture for the following machine instructions. **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.

This computer has the following instructions:

Instruction	Operation
clear A	A = 0; pc=pc+2
Load A,offset8(A)	A = Mem[A+Offset8]; pc=pc+2

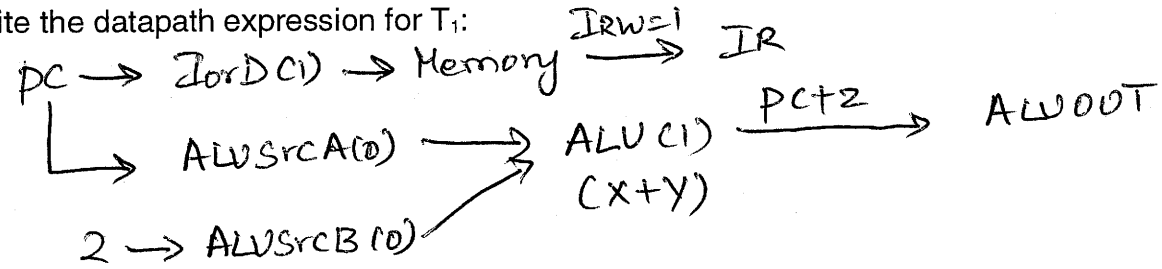
There is only one instruction format shown as follows:

Opcode	**Offset8 field**
8 bits	8 bits
15-8	7 - 0

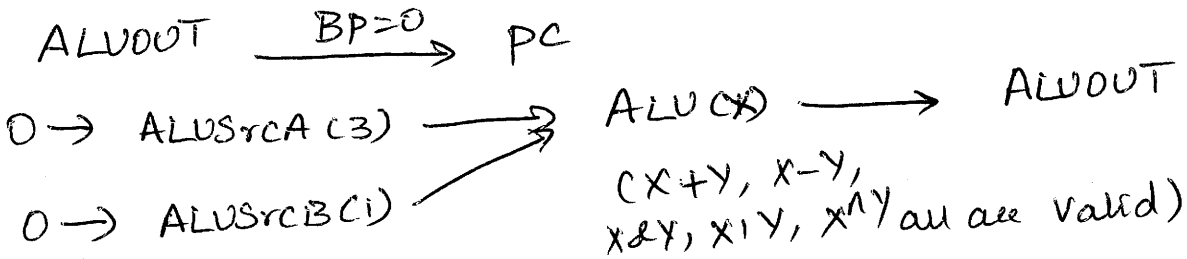
2a (10%) Fill in the settings of the control lines needed for the **"clear"** instruction.
(5% extra credit for using X=Don't Cares correctly)

Clock	MemW	MemR	IorD	IRW	BP	BZ	ALUsrcA	ALUsrcB	ALUop	RegW	RegDst
T_1	0	1	1	1	1	0	0	0	1	0	X
T_2	0	X	X	0	0	X	3	1	X	0	X
T_3	0	X	X	0	1	0	X	X	X	1	0
T_4											
T_5											

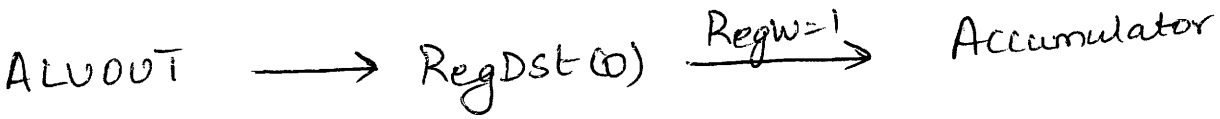
2b (5%) Write the datapath expression for T_1:



2c (5%) Write the datapath expression for T_2:



2d (5%) Write the datapath expression for T_3:



3a (10%) Fill in the settings of the control lines needed for the **"Load"** instruction from Problem 2. (5% extra credit for using X=Don't Cares correctly)

Clock	MemW	MemR	IorD	IRW	BP	BZ	ALUsrcA	ALUsrcB	ALUop	RegW	RegDst
T_1	0	1	1	1	1	0	0	0	1	0	X
T_2	0	X	X	0	0	X	2	3	1	0	X
T_3	0	1	0	0	1	0	X	X	X	0	X
T_4	0	X	X	0	1	0	X	X	X	1	1
T_5											
T_6											
T_7											

3b (5%) Write the datapath expression for T_2:

ALUOUT BP=0 PC

Offset8 → ALUSrcA(2) → ALU(1) → ALUOUT (X+Y)

Read A → ALUSrcB(3) → (A+Offset8)

3c (5%) Write the datapath expression for T_3:

ALUOUT → IorD(0) → Memory MemR=1 → Data → MDR

Data → MDR2

(Mem(A+Offset

3d (5%) Write the datapath expression for T_4:

MDR → RegDst(1) RegW=1 → Accumulator

(A ← Mem(A+Offset

4. Draw lines showing all the data dependencies in "Time" column. and show the 5-stage pipeline sequence (IF, ID, EX, M, WB) for the following code (Note: Can access instruction and data memory at the same time.)

4a (6%) Using **no** forwarding and **cannot** write and read the **same** register within a single clock cycle.

Time		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
sub	$4, $2, $1	IF	ID	EX	M	WB											
lw	$2, 3($1)		IF	ID	EX	M	WB										
sub	$9, $7, $8			IF	ID	EX	M	WB									

4b (6%) Using **no** forwarding and **cannot** write and read the registers within a single clock cycle.

Time		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
sub	$4, $2, $1	IF	ID	EX	M	WB											
lw	$4, 8($4)		IF	ID	ID	ID	ID	EX	M	WB							
sub	$5, $4, $2			IF	IF	IF	IF	ID	ID	ID	ID	EX	M	WB			

4c (6%) Using **no** forwarding and **can** write and read the registers within a single clock cycle.

Time		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
sub	$4, $2, $1	IF	ID	EX	M	WB											
lw	$4, 8($4)		IF	ID	ID	ID	EX	M	WB								
sub	$5, $4, $2			IF	IF	IF	ID	ID	ID	EX	M	WB					

4d (6%) Using forwarding and **can** write and read the registers within a single clock cycle.

Time		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
sub	$4, $2, $1	IF	ID	EX	M	WB											
lw	$4, 8($4)		IF	ID	EX	M	WB										
sub	$5, $4, $2			IF	ID	ID	EX	M	WB								

Extra Credit (5%): In the MIPS paper, what aspect about RISC allows the use of programmed logic arrays?

(1) Instructions are all one size, that results in less decoding logic.

(2) Instructions execute in one cycle.

RISCEE 8 Architecture