Name: _____ email: _____

Problem 1 (20%). Show each step of the pipeline machine (page 470 and 469) for the following instruction sequence: (note: Z is the zero flag from the ALU)

Assume $2=9, $4=8; Mem[12]=742; Mem[16]=1769. Treat the "nop" instruction as "add $0,$0,$0".

```
or    $2, $4, $4      #opcode   $rd, $rs, $rt
sw    $4, 8($4)       #opcode   $rt, offset($rs)
nop
nop
```

Clock	<IF/ID> <PC, IR>	<ID/EX> <WB,M,EX,PC,A,B,S,Rt,Rd>	<EX/MEM> <WB,M,PC,Z, ALU, B, R>	<MEM/WB> <WB,MDR,ALU,R>
0	<0,?>	<?,?,?,?,?,?,?,?,?>	<?,?,?,?,?,?,?>	<?,?,?,?,?>
1	**<4,"or $2,$4, $4">** observe that or $rd=$2,$rs=$rt=$4	<?,?,?,?,?,?,?,?,?>	<?,?,?,?,?,?,?>	<?,?,?,?,?>
2	**<8,"sw $4,8($4)">** observe that sw $rt=$4,8($rs=$4)	**<10, 000, 1100, 4, 8, 8, X, $4,$2>** observe that <wb=10, m=000, ex=1100, pc=4, fetch reg[$rs→$4]→8→A, fetch reg[$rt→$4]→8→B, S=00100 0000 100101=X, $rd=$4>	<?,?,?,?,?,?,?>	<?,?,?,?,?>
3	**<12, "nop">** observe that add $rd=0,$rs=0,$rt=0	**<0X, 001, X001, 8, 8, 8, 8, $4,$X>** observe that <wb=0X ,m=001, ex=X001, pc=8, reg[$rs→$4]→8→A, reg[$rt→$4]→8→B, S=8, $rt=$4, $rd=$X>	**<10, 000, X, 0, 8, 8, $2>** observe that <wb=10, m=000, **pc=X (don't care, not a branch),** ALU zero flag = Z=0, ALU=A or B = 8 + 8= 8, B=8, destination R = ($rt or $rd) = $2>	<?,?,?,?,?>
4	**<16, "nop">** observe that add $rd=0,$rs=0,$rt=0	**<10, 000, 1100, 12, 0, 0, 0, $0, $0>** observe that <wb=10, m=000, ex=1100, pc=12, $rs=$0=0=A, $rt=$0=0=B, S=0, $rt=$0, $rd=$0>	**<0X, 001, X, X or 0, 16, 8, X or $4>** observe that <wb=0X, m=001, **pc=X (don't care, not a branch),** ALU zero flag = Z = X or 0, ALU=A+offset = 8 + 8= 16, B=8, **R = ($rt or $rd) = X or $4>** **(don't care, no register writeback)** **Store to memory** Mem[ALU=16]=8	**<10, X, 8, $2>** **Register writeback to $4** **No memory fetch** Observe that <wb=10, mdr=X, ALU=8, R=$2>
5	**<20, "nop">** observe that add $rd=0,$rs=0,$rt=0	**<10, 000, 1100, 16, 0, 0, 0, $0, $0>** observe that <wb=10, m=000, ex=1100, pc=16, $rs=$0=0=A, $rt=$0=0=B, S=0, $rt=$0, $rd=$0>	**<10,000,X,0,0,0,$0>** observe that <wb=10, m=000, pc=X, Z=0, ALU=A+B=0+0=0, B=0, R=$0>	**<0X, X, X, X or $4>** **Write to Memory**

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

1	IF	Instruction fetch 1st part	3 ns
2	IS	Instruction fetch 2nd part	4 ns
3	ID	Register Read	2 ns; **Branch decision made here**
4	EX	ALU	3 ns
5	MEM	Data Access	8 ns,
6	WB	Register Write	2 ns

This computer has the following instructions:

Instruction	Operation
add $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 (16%) Fill in the following tables

Instruction	IF	IS	ID	EX	MEM	WB	Total Time	Multi-Cycles	Instruction Mix
add	**3**	**4**	**2**	**3**		**2**	**14**	**5**	50%
beq	**3**	**4**	**2**				**9**	**3**	25%
lw	**3**	**4**	**2**	**3**	**8**	**2**	**22**	**6**	17%
sw	**3**	**4**	**2**	**3**	**8**		**20**	**5**	8%

2b (5%) Fill the following table and show work.

Instruction	Clock frequency	CPI (no hazards)	MIPS (no hazards)
Single-cycle CPU	1/22ns **=45.5 MHz**	**=1**	45.5/1 **= 45.5 MIPS**
Multi-cycle CPU	1/8ns **=125 MHz**	5*50% + 3*25% + 6*17% + 5*8% **=4.67**	125/4.67 **=26.8 MIPS**
Pipeline CPU	1/8ns **=125 MHz**	(no hazards = no dependancies = no stalls) **=1**	125/1 **=125 MIPS**

The following parts refer to the pipelined machine only
For the following code: Assume no forwarding.

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,IS,ID,EX,M,WB) ~~and draw lines showing the forwarding.~~ **Note:** Branch decision is made in the ID stage.

Time	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
sub $3, $4, $5	**IF**	**IS**	**ID**	**EX**	**M**	**W**										
beq $1, $3, loop		**IF**	**IS**	**ID**	**ID**	**ID**	**EX**	**M**	**W**							
add $4, $4, $5			**IF**	**IS**	**IS**	**IS**	**ID**	**EX**	**M**	**W**						

2d (12%) Draw lines showing all the data dependencies in "Time" column.
and show the 6-stage pipeline sequence (IF, IS, ID, EX, M, WB) for the following code

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

2e (12%) Using forwarding, show the 6-stage pipeline sequence (IF, IS, ID, EX, M, WB)
and draw lines showing the forwarding.

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

2f (5%) Increase the number of pipeline stages to **7** stages. What stage would you split and why?

I would split the MEM stage into 2 stages (Mem1=4ns and Mem2=4ns) since is the slowest resource (8ns) of all the others.
Note: any other combination would not be optimal: Mem1=3ns and Mem2=5ns

Decreasing the worst case stage delay allows for the pipeline clock to increase.

This results in a faster MIPS.

2g (5%) What is the 7-stage pipeline clock? 4ns or 250 MHz and (assuming no hazards) MIPS= 250 MIPS

MIPS = Clock/CPI = 250 Mhz / 1CPI
Note: 250 MIPS is now faster than all cases in problem 2b!

Problem 3. Show all calculations for the following questions.
Assume an add takes 1 cycle if no dependency and if dependant then 3 clocks
Assume there is a 30% data dependency.
Assume a branch takes 2 cycle if true prediction and if false prediction then 7 clocks.
Assume that 15% of the branches are mispredicted.

3a (5%) What is the average add instruction time in clocks? __**1.6**___

1*70% + 3*30% = 1.6

3b (5%) What is the average branch instruction time in clocks? ___**2.75**___

2*85% + 7*15% = 2.75

Over the past three years, the *Goldman Sachs High Technology Group* has been the lead manager for initial public offerings (IPOs) of 129 technology companies. They (www.gs.com/hightech) would like you to have access to the following *extra credit which can be used for this and previous exams.*

For the following instruction sequence fill in the direct-mapped writeback data cache. The word size is 16 bits. Memory[0]=$4=0x742; Memory[42]=$3=0x1412; Memory[52]=0x1585; Memory[58]=0x1769;

GS1a) (5%) Fill in the miss cache column.

tag	index	byte offset	Instruction	Cache Miss?
111	01	0	lw $1, 58($0)	Yes
110	10	1	lbu $2, 53($0)	Yes
110	10	0	sw $3, 52($0)	No (already loaded by lbu $2,53($0))
000	00	0	lw $4, 0($0)	Yes
101	01	1	lbu $5, 43($0)	Yes
111	01	0	lw $6, 58($0)	Yes (flushed out by lbu $5,43($0))

GS1b) (10%) Show **all states** and **underline the final state** of the direct mapped data cache:

Index	Valid	Dirty	Tag	Data
00	N→Y	N	000	0x742
01	N→Y	N	111→101→111	0x1779→0x1412→0x1769
10	N→Y	N→Y	110	0x1585→0x1412
11	N	N		

GS2a. (10%) Assume 1536 bytes of real memory (0-511)(512-1023)(1024-1535); **LRU**, a page size of 512 bytes and no pages loaded in memory. Fill in the page fault columns. (Blank space implies No)

instruction	Page fault?	Flush which real page?	Write flushed page to disk?	Load what new virtual page	Load into what real page
lw $1, 58($0)	Yes			0 (=0..511)	0 (=0..511)
sw $7, 52($0)					
lw $5, 1412($0)	Yes			2 (=1024..1535)	1 (=512..1023)
lw $2, 742($0)	Yes			1 (=512..1023)	2 (=1024..1535)
sw $6, 1582($0)	Yes	0 (=0..511)	Yes	3 (=1526..2047)	0 (=0..511)
sb $1, 43($0)	Yes	1 (=512..1023)		0 (=0..511)	1 (=512..1023)
lbu $2, 1769($0)					

GS2b.. (5%) Fill out the TLB after execution of part GS2a. (hint: think of a fully associative cache)

Valid	Dirty	Virtual Page Tag	Physical Page Number
N→Y→N→ Y	N→Y→N→ N	0	X→0→X→1
N→Y→N	N→ N→N	2	X→1→X
N→Y	N→ N	1	X→2
N→Y	N→ N	3	X→0

Note: page 0 = 0..511, page 1 = 512..1023, page 2 = 1024..1535, page 3 = 1536..2047, …