EECS 322 Test 4 sample Wednesday April 12, 2000

Name: ______________________________ Note: This exam is longer than the actual one

Problem 1. Show each step of the pipeline machine (page 470 and 469) for the following instruction sequence: Assume $1=9, $2=8; $3=5; $4=3; $5=2; Mem[12]=16. Treat the “nop” instruction as “add $0,$0,$0”.

lw $1, 4($2)
add $3, $4, $5
nop
nop

Clarification
lw C$rt→B, 4(C$rs→A)
add C$rd→R, C$rs→A,C$rt→B
add $rd=0, $rs=0, $rt=0
note: register 0 is always zero!

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,”lw $1,4($2)”>** observe that lw $rt=$1,4($rs=$2)	<?,?,?,?,?,?,?,?,?>	<?,?,?,?,?,?,?>	<?,?,?,?,?>
2	**<8, “add $3,$4,$5”>** observe that add $rd=$3,$rs=$4,$rt=$5	**<11, 010, 0001, 4, 8, 9, 4, $1,X>** observe that <wb=11,m=010,ex=0001,pc=4, $rs→$2→8→A, $rt→$1→9→B, S=4, $rt=$1, $rd=X>	<?,?,?,?,?,?,?>	<?,?,?,?,?>
3	**<12, “nop”>** observe that add $rd=0,$rs=0,$rt=0	**<10, 000, 1100, 8, 3, 2, X, $5, $3>** observe that <wb=10,m=000, ex=1100, pc=8, $rs=$4=3=A, $rt=$5=2=B, S=X, $rt=$5, $rd=$3>	**<11,010,20,0,12,9,$1>** observe that <wb=11, m=010, pc=pc+(S<<2)=4+(4<<2)=20, Z=0, ALU=S+$2=4+8=12, 9, $1>	<?,?,?,?,?>
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>	**<10,000,X,0,12,9,$3>** observe that <wb=10, m=000, pc=X, Z=0, ALU=A+B=3+2=5, B=2, R=$3>	**<11,16,12,$1>** =>Reg[$1]=16 **finally register 1 will contain the load memory word of 4 + $12** observe that <wb=11, mdr=Mem[ALU] =Mem[12]=16, ALU=12, R=$1>
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>	**<10,X,0,12,9,$3>** =>Reg[$3]=5 **finally register 3 will contain the add of $4 plus $5 in the writeback!** observe that <wb=10, mdr=X, ALU=5, R=$3>

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

1	IF	Instruction fetch	3 ns
2	ID	Register Read	1 ns
3	EX	Multiply	4 ns
4	EX2	ALU	2 ns
5	MEM	Data Access	3 ns
6	WB	Register Write	1 ns

This computer has the following instructions:

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

load & store requires alu to add addr16+$rs

(a) Fill in the following table

Instruction	IM	ID	EX	EX2	MEM	WB	Total Time	Multi-Cycles	Instruction Mix
add	3 ns	1 ns		2 ns		1 ns	7 ns	4	20%
lw	3 ns	1 ns		2 ns	3 ns	1 ns	10 ns	5	20%
sw	3 ns	1 ns		2 ns	3 ns		9 ns	4	20%
beq	3 ns	1 ns		2 ns			6 ns	3	20%
madd	3 ns	1 ns	4 ns	2 ns		1 ns	11 ns	5	20%

(b) Fill the following table

Instruction	Clock frequency	CPI	MIPS
Single-cycle CPU	1/11 ns= 90.9 MHz	1	90.9 Mhz/1=90.9 MIPS
multi-cycle CPU	1/4ns=250 Mhz	4.2	250 MHz/4.2 = 59.5 MIPS

The parts refer to pipelined machine only

(c) What is the instruction latency? (PAGE 522) with no dependancies, is the amount of time it is in the pipeline: 6

(d) What is the pipeline clock? slowest stage = 1/4ns = 250 MHz

(e) What is the pipelined speed up? page 440, non-pipelined (single)/pipelined = 11 ns/4ns = 2.75

(f) Without forwarding, fill in the hazard type (page 441-8) and condition (page 479), for the following instruction sequence:

Instruction	Hazard type
add $1, $2, $3	none
sub $4, $2, $1	data hazard
lw $5, data($6)	
beq $1, $5, loop2	two data hazards

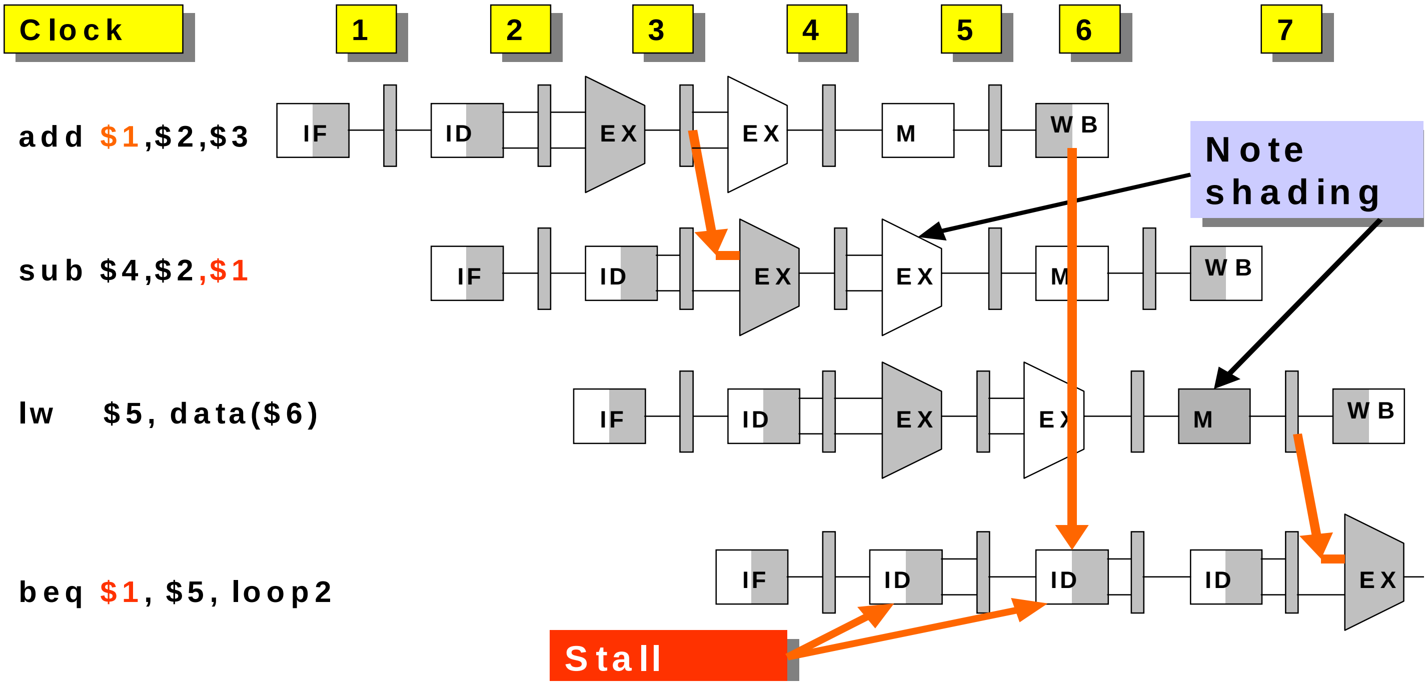
(g) **Without** forwarding, draw the pipeline sequence (like figure 6.4) for part f
(h) .Show all timing, including stalls.

Time	1	2	3	4	5	6	7	8	9	10	11	12
add $1, $2, $3	IF	ID	EX	EX2	MEM	WB						
sub $4, $2, $1		IF	ID	ID	ID	ID	EX	EX2	MEM	WB		
lw $5, data($6)			IF	ID	EX	EX2	MEM	WB				
beq $1, $5, loop2				IF	ID	ID	ID	ID	EX	EX2	MEM	WB

(i) Suppose the hazard detection unit does not work on the processor due to a bug in the design. Using the nop instruction, rewrite the code of part f which will give correct results (like page 478).

Time	1	2	3	4	5	6	7	8	9	10	11	12	…
add $1, $2, $3	IF	ID	EX	EX2	MEM	WB							
nop		IF	ID	EX	EX2	MEM	WB						
nop			IF	ID	EX	EX2	MEM	WB					
nop				IF	ID	EX	EX2	MEM	WB				
sub $4, $2, $1					IF	ID	EX	EX2	MEM	WB			
lw $5, data($6)						IF	ID	EX	EX2	MEM	WB		
nop							IF	ID	EX	EX2	MEM	WB	
nop								IF	ID	EX	EX2	MEM	WB
nop									IF	ID	EX	EX2	MEM
beq $1, $5, loop2										IF	ID	EX	EX2

(j) Using forwarding, draw graphical pipeline sequence (like figure 6.8) of part f. Show all shading.
(k) Show all timing including stalls.



(l) Using forwarding, reorder the code to minimize the stalling of part f.

lw	$5, data($6)	**move non-dependant load as early as possible!**
add	$1, $2, $3	
sub	$4, $2, $1	
beq	$1, $5, loop2	

(m) Given any 2 instruction combinations, what is the worst instruction sequence. (i.e. assume no forwarding or prediction). Draw the pipeline chart (like Figure 6.4). Determine the MIPS.

- In general data dependencies create the worst sequence. Here are two observations:
- The arithmetic or R-format instructions can be forwarded early because the alu result is early in the pipeline stage (i.e. EX or EX2 stages).
- The data result of the load is the worse than the R-format because the result can only come after the memory has been read (i.e. MEM stage).

Time	1	2	3	4	5	6	7	8	9	10	11	12
lw $1, 10($2)	IF	ID	EX	EX2	MEM	WB						
lw $2, 20($1)		IF	ID	ID	ID	ID	EX	EX2	MEM	WB		

Assume %100 percent dependancy: then instruction time is 1 + 3 stalls = 4 clocks

Then MIPS = clock speed/CPI = 250 MHz/4 = 62.5 MIPs

(n) Given any 2 instruction combinations, what is the worst instruction sequence using forwarding. Draw the pipeline chart (like Figure 6.4). Determine the MIPS.

Time	1	2	3	4	5	6	7	8	9	10	11	12
lw $1, 10($2)	IF	ID	EX	EX2	MEM	WB						
lw $2, 20($1)		IF	ID	ID	ID	EX	EX2	MEM	WB			

Assume %100 percent dependancy: then instruction time is 1 + 2 stalls = 3 clocks

Then MIPS = clock speed/CPI = 250 MHz/3 = 83.3 MIPs

(o) Using all the best possible instruction flow (i.e. no hazards, assume no forwarding or prediction) of the 5 instructions. Determine the MIPS.

That would be any set of instructions that are not data dependant (0% dependancy).

Then the instruction time = 1 clock

MIPS = Clock speed/CPI = 250 Mhz / 1 = 250 MIPs

(p) Show the worst possible instruction pipeline sequence of using only the 5 instructions. (i.e. assume no forwarding or prediction). Draw the pipeline chart (Figure 6.4). Determine the MIPS.

That would be that case of 100% dependency like part m. Same solution for MIPs as part m.

```
lw      $1, 10($2)
lw      $1, 10($1)
lw      $1, 10($1)
lw      $1, 10($1)
lw      $1, 10($1)
```