EE282 Computer Architecture and Organization
Midterm Exam

October 25, 1999

(Total Time = 120 minutes, Total Points = 100)

Name: (please print) ____

In recognition of and in the spirit of the Stanford University Honor Code, I certify that I will neither give nor receive unpermitted aid on this exam.

Signature:____

This examination is open notes, open book. You may not, however, collaborate in any manner on this exam. You have two hours to complete the exam. Before starting, please make sure you have all 11 pages.

1	
2	
3	
4	
Total	

1) Short Answer Questions (30 points – 3 points each unless marked)

A) A program consist of 15 subroutines – each run once. You know the CPI for each subroutine and how many instructions are in each subroutine. Which mean should be used to compute the CPI for the entire program?

Weighted Arithmetic Mean

B) A well-encoded variable-length instruction set has better code density than a well-encoded fixed-length instruction set (true or false).

True

C) All accumulator machines must use self-modifying code to traverse a linked list (true or false).

False

D) Increasing a 5-stage DLX pipeline to a 6-stage pipeline by breaking the execute stage into two stages will add how many additional stall cycles to the following code:

```
ADD   R1,R2,R1
LW    R2, 0(R2)
ADD   R1,R2,R1
J     label
```

1 extra stall

5 stage pipeline:

```
F R A M W
  F R A M W
    F R – A M W
      F R – A M W
```

6 stage pipeline:

```
F R A A M W
  F R A A M W
    F R – - A A M W
      F R – - A A M W
```

E) What is the asymptotic prediction accuracy of a two-bit branch predictor on the repeating pattern TTNTTTTNTTTTNTT… (T=taken, N=not taken).

80% - eventually the predictor will hit the strongly-taken state after 4 T's – after that it will always predict T and be correct 80% of the time.

F) An out-of-order superscalar implementation of a RISC architecture can sometimes have a higher CPI than an in-order multiple-issue implementation with similar execution resources. (true or false)

True

G) (6 points)
Rewrite the following program from a 3-address DLX architecture using a 2-address version of the DLX architecture.

```
Example:        2 address DLX
                ADD   R8,R9 (R8 = R8 + R9)
```

Be sure that your program is exactly equivalent. (Assume R0 is always 0)

```
XOR    R3,R2,R1
ADD    R6,R5,R4
ADD    R7,R3,R6
```

For example:

```
XOR    R3,R3
XOR    R3,R2
XOR    R3,R1
XOR    R6,R6
XOR    R6,R5
ADD    R6,R4
XOR    R7,R7
XOR    R7,R3
ADD    R7,R6
```

H) Adding a faster floating-point unit to a machine will give a speedup of 3 on instructions that make up 75% of the runtime on the original machine. What is the overall speedup of the machine due to this modification.

Speedup = old execution time / new execution time
= 1/(.25 * 1 + .75 * 1/3)
= 1/.5
= 2

I) In order to reorder two store operations on an out-of-order superscalar processor, what two conditions must be met?

2 of the following:

The two stores cannot have overlapping addresses
There cannot be any loads or stores to overlapping addresses between the two stores
There cannot be any unrecoverable exceptions from the second store

2) Pipelining (25 points)

Consider the following 8-cycle pipelined processor that we will call SUPER-DLX.

IF1	Instruction fetch 1: send address to I-cache
IF2	Instruction fetch 2: receive instruction from I-cache
ID1	Instruction decode begins, branch or jump target computed
ID2	Read registers, branch condition computed
EX	ALU operations computed (address for load/store calculated)
MEM1	Memory 1: send address to D-cache
MEM2	Memory 2: read/write data to D-cache
WB	Write results back to register file

In SUPER-DLX, the writes to the register file occur in the first half of the cycle and the register reads occur in the second half of the cycle. You can assume that the IF and MEM stages are pipelined so that one instruction fetch (IF) and one data access (M) can begin each cycle.

A. (7 points) Assume that all possible useful bypass paths will be incorporated into this pipeline. List all of the bypass paths.

First look at your producers and consumers in the pipeline.
Producers: EX, MEM2
Consumers: EX, MEM2, ID1, ID2 (note: MEM1 is not a consumer because the address is calculated during the EX stage).

EX->EX
EX->ID2
EX->ID1

MEM1->EX
MEM1->ID2
MEM1->ID1

MEM2->MEM2
MEM2->EX
MEM2->ID2
MEM2->ID1

B. (6 points) List all forwarding paths that are exercised to execute the following code on the SUPER-DLX. How many stall cycles will occur, if any?

```
1.    LW    R2, 8(R4)
2.    ADD   R3, R2, R5
3.    ADD   R4, R1, R3
4.    SW    R4, 0(R6)
```

Forwarding Paths

1) MEM2->EX	**(LW to ADD)**
2) EX->EX	**(ADD to ADD)**
3) MEM2->MEM2	**(ADD to SW)**

(also accepted EX->EX for ADD to SW)

Number of Stall Cycles	**2**

There are 2 stalls between the LW and ADD. No other stalls occur.

C. (6 points) For the following piece of code how many stall cycles will occur from the first SUB to the instruction at END if the branch is taken? Assume that the branch target is always correctly predicted if the branch is taken. Answer the question for no prediction, predict not taken, and predict taken.

```
1.    SUB   R3, R7, R8
2.    BEZ   R3, END     // branch if R3 == 0
      ...
      ...
END:  SW    R3, 4(R9)
```

scheme	**# Stall Cycles**
no prediction	**4**
predict not taken	**4**
predict taken	**1**

Also accepted 3 for predict taken case since you were not explicitly told where the target prediction occurs.

For “no prediction” case there is one stall between the SUB and BEZ. Then 3 more stalls are incurred in waiting until the ID2 stage has finished executing. This brings total stalls to 4.

For “predict not taken” case there is one stall between the SUB and BEZ. Then the pipeline has to squish 3 instructions that were incorrectly fetched. This brings total stalls to 4.

For “predict taken” case there is one stall between the SUB and BEZ. Since we assume the target is predicted correctly in the IF1 stage (which is the normal place it would occur), we incur no other stalls. This brings our total stalls to 1. If you did not assume a prediction of target then you would know where to branch at the end of the ID1 stage and would then incur a total of 3 stalls.

D. (6 points) For each of the following instruction sequences, specify any forwarding path used and/or the number of stall cycles that are required on the SUPER-DLX.

Type	**Instruction Sequence**	**Forwarding Path**	**Stalls**

1	`ALU    Rx, -, -`
	`Store  Rx, 0(-)`
		MEM2_i => MEM2_i+1	**0**

2	`Load   Rx, 0(-)`
	`ALU    -, -, Rx`
		MEM2_i => EX_i+3	**2**

3	`Load   Rx, 0(-)`
	`Branch Rx, Ry, target`
		MEM2_i => ID2_i+4	**3**

4	`ALU    Rx, -, -`
	`Branch Rx, Ry, target`
		EX_i => ID2_i+2	**1**

5	`Load   Rx, 0(-)`
	`ALU    Ry, R0, R0`
	`ALU    -, -, Rx`
		MEM2_i => EX_i+3	**1**

6	`ALU    Rb, -, -`
	`Store  -, 0(Rb)`
		EX_i => EX_i+1	**0**

3) Branch Prediction (20 points)

Consider the following fragment of “C” code:

```
for(i=0;i<5;i++) {
   // block a
   if(i&1) { // if i is odd
      // block b
   }
}
```

and suppose that the “for” statement and the “if” statement are both realized with a BNZ instruction. Remember that in “C”, the loop test is performed prior to executing the loop body.

(a) (5 points) In the steady state, what is the prediction accuracy for each of the following predictors on these two branches. Fill in the blanks in the table

Predictor	Accuracy on “for”	Accuracy on “if”
Static, predicting taken for both the for and the if		
One-bit predictor		
Two-bit predictor		
Pattern-based predictor with a four-bit pattern		
Pattern-based predictor with a five-bit pattern		

Suppose that this machine has a 9-stage pipeline with the following stages

F1 – start the fetch, predict if the instr. is a branch, whether it is taken or not, and if taken, the target address.
F2 – complete the fetch
D – decode the instruction – know it’s a branch at the end of D, branch address available at the end of this stage
R – read the registers, resolve branch condition
A1 – start ALU operation
A2 – complete ALU operation
M1 – start memory operation
M2 – complete memory operation
W – write the result to registers

(b) (4 points) What is the penalty for a mispredicted branch?

(c) (4 points) What is the penalty when a branch is correctly predicted as taken, but the branch address is incorrectly predicted?

(d) (7 points) Assume the following

1. There are no stalls in this pipeline except for branch instructions.
2. Branch instructions account for 20% of all instructions
3. 30% of branch instructions are taken
4. Branches are correctly predicted 90% of the time
5. The target address for a taken branch is correctly predicted 50% of the time

What is the CPI for this machine?

4) Dynamic Scheduling (25 points)

A loop body has been naively compiled as:

```
      LOOP:
1.          LW    R4,0(R1)
2.          LD    F2,0(R2)
3.          LD    F4,0(R3)
4.          ADD   R4,R4,R3
5.          LD    F6,0(R4)
6.          MULD  F8,F6,F4
7.          MULD  F10,F8,F2
8.          ADDD  F8,F8,F2
9.          SUBI  R1,R1,#4
10.         SUBI  R2,R2,#4
11.         SUBI  R3,R3,#4
12.         SUB   R5,R1,R6
13.         BNEZ  R5,LOOP
```

The execution time of the operations (the time from which the instruction enters the first execute stage of the pipeline until the result is available for bypass) is as follows:

LW or LD (hit)	2 cycles	(Pipe is F-R-A-M-W)
LW or LD (miss)	7 cycles	
ADDD or SUBD	3 cycles	
MULD	4 cycles	(Pipe is F-R-A-A-A-A-W)

All other operations have a 1 cycle execution time. All execution units are fully pipelined and can start a new operation every cycle. Assume that all loads and stores hit in the data cache and that there are no instruction cache misses. The F, R, and W stages are 1 cycle for all operations. The number of A and M stages differ based on the operation and can be determined from the information above. An instruction is complete when it completes the W stage. The general-purpose register file is separate from the floating-point register file (one floating point register and one general-purpose register can be written each cycle). All possible forwarding paths exist. Nothing can issue unless a write port will be available when it reaches the write stage.

You may find the templates at the end of the exam useful in computing your answer. If you wish the graders to consider information on these templates it must be labeled clearly. Ambiguous information will be disregarded.

A. (7 points) How long does one iteration of this loop take to execute on a statically scheduled, in-order issue machine that can issue at most one instruction per cycle. Measure from when the first instruction starts (F stage) until all instructions complete the W stage. Do not modify the code.

Number of Cycles	**22**

B. (5 points) Now assume that your statically scheduled machine can fetch and issue up to 6 instructions per cycle. (There are enough register ports (read and write), memory ports, and execution units for any 6 instructions per cycle.) However, instructions still must issue *in order* – an instruction cannot issue (enter the A stage) before an earlier instruction. Now how many cycles does it take to execute the code?

Number of Cycles	**16**

C. (7 points) Manually reorder the code to give the best possible execution time on the single-issue, in-order machine from part A. Indicate the new ordering and give the execution time in cycles.

New Order	**1**	**3**	**4**	**5**	**2**	**6**	**9**	**10**	**11**	**7**	**8**	**12**	**13**
Number of Cycles	**17**												

D. (6 points) Now consider a dynamically-scheduled, out-of-order machine that can issue up to 2 instructions per cycle. Assume that the machine can fetch up to 15 instructions ahead and always executes the earliest instructions that are ready. The write stage can complete 2 instructions per cycle. How many cycles will it take to execute one iteration of this loop using the original code?

Number of Cycles	**16**

Name____________________

Instruction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
Part A																																	
1	F	R	A	M	W																												
2		F	R	A	M	W																											
3			F	R	A	M	W																										
4				F	R	A	M	W																									
5					F	R	A	M	W																								
6						F	R	-	A	A	A	A	W																				
7							F	R	-	-	-	-	A	A	A	A	W																
8								F	R	-	-	-	-	-	A	A	A	W															
9									F	R	-	-	-	-	-	A	M	W															
10										F	R	-	-	-	-	-	A	M	W														
11											F	R	-	-	-	-	-	A	M	W													
12												F	R	-	-	-	-	-	A	M	W												
13													F	R	-	-	-	-	-	A	M	W											
Part B																																	
1	F	R	A	M	W																												
2	F	R	A	M	W																												
3	F	R	A	M	W																												
4	F	R	-	-	A	M	W																										
5	F	R	-	-	-	A	M	W																									
6	F	R	-	-	-	-	-	A	A	A	A	W																					
7		F	R	-	-	-	-	-	-	-	-	A	A	A	A	W																	
8		F	R	-	-	-	-	-	-	-	-	A	A	A	W																		
9		F	R	-	-	-	-	-	-	-	-	A	M	W																			
10		F	R	-	-	-	-	-	-	-	-	A	M	W																			
11		F	R	-	-	-	-	-	-	-	-	A	M	W																			
12		F	R	-	-	-	-	-	-	-	-	-	A	M	W																		
13			F	R	-	-	-	-	-	-	-	-	-	A	M	W																	

Instruction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
Part C																																	
1	F	R	A	M	W																												
3		F	R	A	M	W																											
4			F	R	A	M	W																										
5				F	R	A	M	W																									
2					F	R	A	M	W																								
6						F	R	A	A	A	A	W																					
9							F	R	A	M	W																						
10								F	R	A	M	W																					
11									F	R	A	M	W																				
7										F	R	A	A	A	A	W																	
12											F	R	A	M	W																		
8												F	R	A	A	A	W																
13													F	R	A	M	W																
Part D																																	
1	F	R	A	M	W																												
2	F	R	A	M	W																												
3	F	R	-	A	M	W																											
4	F	R	-	-	A	M	W																										
5	F	R	-	-	-	A	M	W																									
6	F	R	-	-	-	-	-	A	A	A	A	W																					
7	F	R	-	-	-	-	-	-	-	-	-	A	A	A	A	W																	
8	F	R	-	-	-	-	-	-	-	-	-	A	A	A	W																		
9	F	R	-	A	M	W																											
10	F	R	-	-	A	M	W																										
11	F	R	-	-	-	A	M	W																									
12	F	R	-	-	-	-	A	M	W																								
13	F	R	-		-	-	-	A	M	W																							