Very short answer questions. "True" and "False" are considered short answers.

(1) (2) What are the two biggest challenges facing MIMD processors?

(2) (1) Which type of predictor is more accurate, static or dynamic?

(3) (1) Is it possible to have a WAR hazard in a 9-stage MIPS-like pipeline?

(4) (1) The design of an instruction set has a definite impact on how pipelinable it is.

(5) (1) Give an example of an instruction set that is virtualizable.

(6) (1) Which type of operation is necessary in order to support synchronization?

(7) (1) What is a "name" hazard/dependence?

(8) (1) What is a "true" hazard/dependence?

(9) (1) Using a different mapping scheme will reduce which type of cache miss?

(10) (1) Which type of cache miss can be reduced by using longer lines?

(11) (1) Which type of cache miss can be reduced by using shorter lines?

(12) (2) What is coherence?

(13) (2) What is consistency?

(14) (2) What is the equation to calculate the average memory access time?

(15) (2) Give two techniques which will help reduce the Hit Time.

(16) (2) Give two techniques will help reduce the Miss Rate.

(17) (2) Give two techniques will help reduce the Miss Penalty.

(18) (2) There are two ways to define performance - what are they?

(19) (2) Predicting a direction is only half of the problem. What else is necessary in order to overcome control flow hazards?

(20) (2) Interrupts/Exceptions require the CPU to provide the OS with what two things?

(21) (2) What are the two main ways to insert a stall/bubble into the pipeline?

(22) (2) We talked about two techniques for providing out of order execution - Tomasulo's algorithm and Scoreboarding. What was the primary difference between them?

(23) (2) Clock rates have grown by a factor of 1000 while power consumed has only grown by a factor of 30. How was this accomplished?

(24) (2) In order to support out of order completion, the designer has to figure out how to deal with what two types of data hazards?

(25) (3) Why is it difficult to come up with good benchmarks for parallel processors?

(26) (3) What is the goal of the memory heirarchy? What two principles make it work?

(27) (2) What is Weak Scaling?

(28) (2) What is a "balanced" pipeline?

(29) (2) What does TLB stand for, and what is it/what does it do?

(30) (4) What are the three types of hazards? Which one can be eliminated by "throwing more money at the problem"?

(31) (4) What is the difference between static and dynamic scheduling? And why do we want to schedule code, anyway?

(32) (4) There is one particular type of branch that is particularly difficult for predictors to deal with. What is it, and what hardware structure is used to deal with this?

(33) (3) Assuming a 16-bit address and a 256-byte Direct Mapped cache with a linesize=8, show how an address is partitioned/interpreted by the cache.

(34) (3) Assuming a 16-bit address and a 160-byte 5-way SA cache with a linesize=4, show how an address is partitioned/interpreted by the cache.

(35) (3) Given a 1 Megabyte physical memory, a 24 bit Virtual address, and a page size of 2K bytes, write down the number of entries in the Page Table, and the width of each entry.

(36) (4) Given a 1 Megabyte physical memory, a 32 bit Virtual address, and a page size of 2K bytes, write down the number of entries in the Page Table, and the width of each entry. Is there a problem with this configuration? If so, how can you fix it?

(37) (5) The standard MIPS has a 5-stage pipeline, and uses a load delay slot. If the machine is redesigned to be a 9-stage pipeline, with the following stages:

F D1 D2 RR E1 E2 M1 M2 WB (where RR stands for Register Read)

How many load delay slots will this new design require, assuming the memory returns the value during M2? During M1?

(38) (4) The designer has the choice of using a virtually addressed cache or a physically addressed cache. Explain the difference, and give 1 disadvantage for each.

(39) (4) Supporting precise interrupts in machines that allow out of order completion is a challenge. Briefly explain why.

(40) (5) You have an instruction that is a fixed size of 24 bits, and you want to support 256 different operations. You also want to have 64 registers, and use a 3-operand instruction format. If it is possible to this, draw what an instruction would look like. If it is not possible, explain why, and show what you would do to fix the problem.

(41) (4) What is the definition of a basic block? Why is there a desire to create a bigger one?

(42) (4) Describe the difference between superscalar and VLIW processors.

(43) (3) Give two advantages to using a superscalar processor, and 1 advantage to using a VLIW.

(44) (6) Describe the difference between shared memory and message passing machines. Include the impact on design, cost, and programming model.

(45) (6) Understanding the hardware can influence how you write programs. Give 2 examples of how you might write software differently for a heavily pipelined machine verses a non-pipelined one.

(46) (6) In class, we talked about the cycle by cycle steps that occur on different interrupts. For example, here is what happens if there is an illegal operand interrupt generated by instruction i:

	1	2	3	4	5	6	7	8	9	
i	IF	ID	RR	EX	MEM	WB	<- Interrupt detected			
i+1		IF	ID	RR	EX	MEM	WB	<- Instruction Squashed		
i+2			IF	ID	RR	EX	MEM	WB	<- Trap Handler fetched	
i+3				IF	ID	RR	EX	MEM	WB	
i+4					IF	ID	RR	EX	MEM	WB

Fill out the following table if instruction i+1 experiences a fault in the EX stage (Overflow, for example):

	1	2	3	4	5	6	7	8	9	10	
i	IF	ID	RR	EX	MEM	WB					
i+1		IF	ID	RR	EX	MEM	WB				
i+2			IF	ID	RR	EX	MEM	WB			
i+3				IF	ID	RR	EX	MEM	WB		
i+4					IF	ID	RR	EX	MEM	WB	
i+5						IF	ID	RR	EX	MEM	WB

What happens in this case?

	1	2	3	4	5	6	7	8	9	10	
i	IF	ID	RR	EX	MEM	WB	<- Execution stage has overlflow				
i+1		IF	ID	RR	EX	MEM	WB	<- Inst Read causes Page Fault			
i+2			IF	ID	RR	EX	MEM	WB			
i+3				IF	ID	RR	EX	MEM	WB		
i+4					IF	ID	RR	EX	MEM	WB	
i+5						IF	ID	RR	EX	MEM	WB

(47) (2) What is the maximum number of exceptions that could happen at one time in the above machine?

In this question, we are going to wire up a 24-bit version of the machine used in the midterm.

The machine has 3 different instruction formats: R, I, and J.

R-type:

Opcode	rs	rt	rd
23-18	17-12	11-6	5-0

I-type:

Opcode	rs	rt	Immediate
23-18	17-12	11-6	5-0

J-type:

Opcode	Offset
23-18	17-0

The machine is byte-addressable. Immediates are sign-extended, and jumps are handled similar to what is done in the MIPS processor (they are combined with the PC, although no zero's are appended on the end).

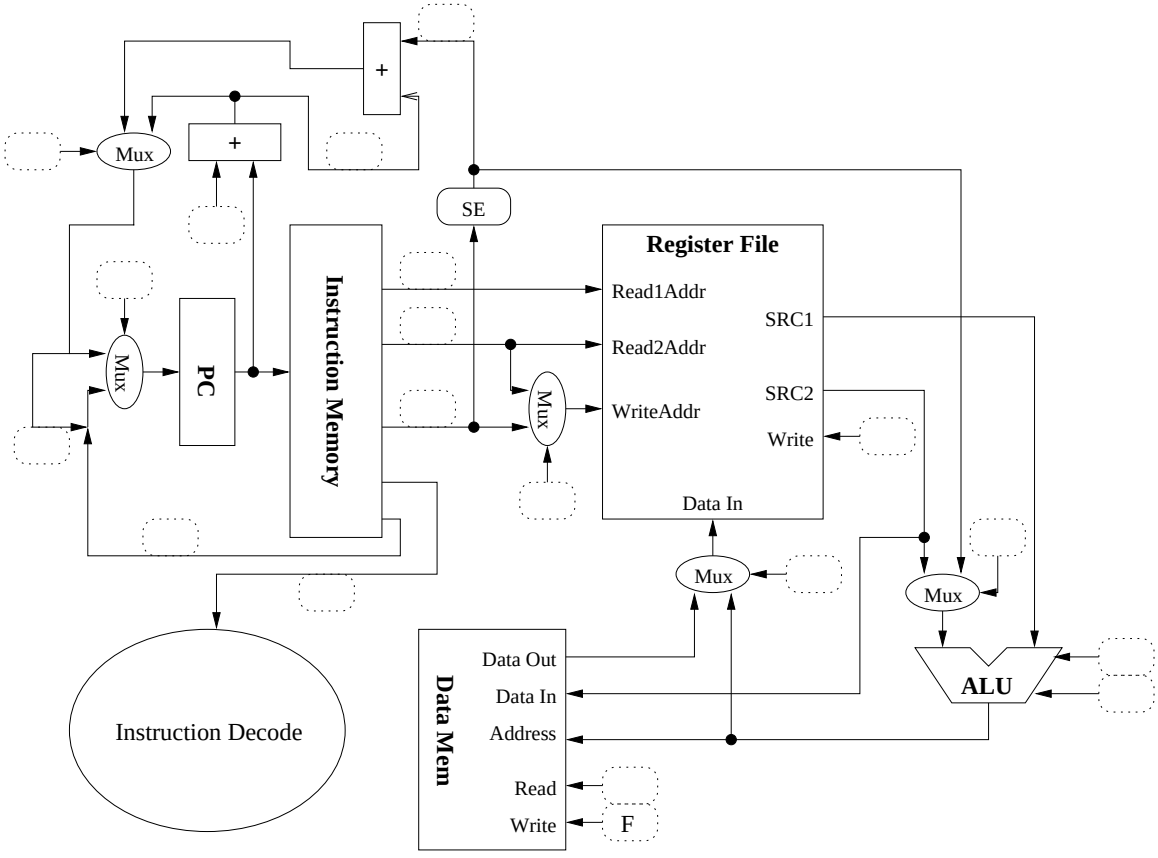
Here are some of the instructions that have been defined:

Name	**Opcode**	**Name**	**Opcode**	**Name**	**Opcode**
lw	010000	sw	001100	NOP	000000
beqz	100000	j	110000	NOT	000001
ADD	000010	ADD Imm	000011	SUB	000100
AND	000110	AND Imm	000111	XOR	001010
OR	001000	OR Imm	001001	XOR Imm	001011

Here are the 10 control signals.

A) PCIn	B) MemRead	C) ALUSource	D) ALU0	E) ALU1
F) MemWrite	G) RegWrite	H) Branch	I) WriteSrc	J) WrAddr

(48) (9) In the diagram below, make sure all the dashed boxes are filled. Two are already done for you. Use the letters from the previous page instead of the actual names (F instead of MemWrite, for example).



(49) (2) Write down the exact boolean equation for the Branch signal.

(50) (2) How many registers does this machine support?

(51) (5) We need a JAL instruction - let's say it uses register 9. Sketch the changes necessary on the above diagram. In addition, briefly describe in words what would need to change in this machine in order to make this work. (Don't forget about instruction set impacts ...)

(52) (8) Supporting multiple length instructions introduces new challenges to a pipelined machine. As we did in class, fill out the following table of execution, assuming the INT unit takes 1 cycle, the FP+ unit takes 2 cycles, and the FP* unit takes 4 cycles. Also assume there are separate FP and INT registers. The first one is done for you.

	1	2	3	4	5	6	7	8	9	10	11	
FP*	IF	ID	FP*	FP*	FP*	FP*	WB					<- No Problems
int		IF	ID	EX	MEM	___	___	___	___	___	___	<- _____
FP+			IF	___	___	___	___	___	___	___	___	<- _____
int				IF	___	___	___	___	___	___	___	<- _____
FP+					IF	___	___	___	___	___	___	<- _____
FP+						IF	___	___	___	___	___	<- _____
i+6							IF	___	___	___	___	<- _____