(1) [2] What are the two main ways to define performance?

(2) [2] What is one of the simplest (and oldest) techniques for exploiting parallelism among instructions?

(3) [2] As technological advances allow us to make both transistors and wires smaller and smaller, what happens to the power density?

(4) [2] When we talk about the number of operands in an instruction (a 1-operand or a 2-operand, for example), what do we mean?

(5) [4] In the last 20 years, clock rates have grown by a factor of 1000 while power consumed has only grown by a factor of 30. How was this accomplished? Why is this technique no longer effective?

(6) [4] It is difficult for the internal processing elements on a CMOS chip to cross the chip boundary and communicate with things that are on other chips. Explain why that is.

(7) [4] There are two different types of power consumption, static and dynamic.

a) Which one contributed the most to overall power consumption 20 years ago?

b) What is the situation now, and why has it changed?

(8) [4] Why are there multiple dies per silicon wafer? Why not just fabricate one huge die per wafer?

(9) [3] What is a benchmark program?

(10) [4] Is it true that as technology changes and chips use more and more transistors, benchmarks must change to adapt? Explain your answer.

(11) [4] Machines today use registers - often as many as they can. Give 2 advantages and 2 disadvantages to using registers.

(12) [4] A standard compiler optimization step is to do register allocation. However, there are times when register allocation is difficult to do while maintaining program correctness - explain why.

(13) [4] Why do most current chips feature multiple cores on the same die?

(14) [4] What are the 3 pipeline hazards? Which one can be solved by providing more resources?

(15) [6] In your first design of a 6-stage pipeline (F,D,E1,E2,M,W), F takes 27 time units, D takes 28, E1 takes 16, E2 takes 14, M takes 29, and W takes 28.

a) What will the clock cycle time be for this pipeline?

b) Is it a balanced pipeline? If not, explain what you could do to fix it. What would the cycle time be now?

(16) [4] Processor A requires 200 instructions to execute a given program, uses 4 cycles per instruction, and has a cycle time of 5 ns. Processor B requires 100 instructions to do the same program, and has a cycle time of 4 ns. How any cycles per instruction does Processor B need to use in order to give the same CPU time as Processor A? (Show your work)

(17) [4] An important program spends 70% of its time doing Integer operations, and 30% of its time doing floating point arithmetic. By redesigning the hardware you can either make the Floating Point unit 90% faster (take 10% as long), or the integer unit 30% faster (take 70% as long). Which should you do, and why?

(18) [6] You are responsible for designing a new embedded processor, and you have been told you are restricted to using a fixed 8 bit instruction size. To make this work you have decided to use a 1-operand instruction format, 16 registers, and support 16 instructions. Your boss just came in and said things have changed, and you now must use 16 bits - how would you change your instruction format? Be sure to explain why you are making the changes (what problem are you solving?)

(19) [10] The standard MIPS implementation has a 5-stage pipeline, writes to the register file during the first half of the D cycle and reads during the second half, and uses both a branch delay slot and a load delay slot. If the machine is redesigned to be an 8-stage pipeline, with the following stages:

F1 F2 D E1 E2 M1 M2 WB

a) Assuming this machine has a branch predictor and the branch condition is calculated by the end of the D stage, how big is the branch penalty (measured in cycles) when the prediction is incorrect? What if the branch condition is not calculated until the end of E2?

b) How many load delay slots would this machine need (assuming it has forwarding logic and you are forwarding to E2) if the memory returns the value by the end of M2? M1?

c) What type of data hazard does the above pipeline need to worry about?

d) If the above pipeline were modified to support out of order completion, what new data hazard would be introduced?

(20) [14] Here is a code sequence.

lw R1, 8(R10)

sub R2, R3, R1

add R1, R4, R3

and R3, R1, R4

a) Assuming a standard 5-stage pipeline that does not support hazard detection and does no forwarding,

a) Indicate all dependencies (draw lines/arrows between them, and write beside each line/arrow which hazard is involved).

b) Insert as many No Operations (NOPS) as required in order to ensure this code runs correctly. (Remember, writes to the register file occur on the first half of the cycle, and reads occur during the second half).

c) Circle the NOPs that can be removed if forwarding and hazard detection logic is implemented.

d) Is it possible to reorder (schedule) the code to remove any stalls? If so, show how you would do it. If not, explain why you can't.

(21) [9] 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 (note this machine has a 7 stage pipeline and supports imprecise interrupts. RR stands for Register Read):

	1	2	3	4	5	6	7	8	9	10	11	12	13
i	IF	ID	RR	EX	M1	M2	WB	<- Interrupt detected					
i+1		IF	ID	RR	EX	M1	M2	WB	<- Instruction Squashed				
i+2			IF	ID	RR	EX	M1	M2	WB	<- Trap Handler fetched			
i+3				IF	ID	RR	EX	M1	M2	WB			

Fill out the following table if instruction i+1 experiences a fault in the EX stage:

	1	2	3	4	5	6	7	8	9	10	11	12	13
i	IF	ID	RR	EX	M1	M2	WB						
i+1		IF	ID	RR	EX	M1	M2	WB					
i+2			IF	ID	RR	EX	M1	M2	WB				
i+3				IF	ID	RR	EX	M1	M2	WB			
i+4					IF	ID	RR	EX	M1	M2	WB		
i+5						IF	ID	RR	EX	M1	M2	WB	
i+6							IF	ID	RR	EX	M1	M2	WB

Assuming precise interrupts are being supported, what happens in this case?

	1	2	3	4	5	6	7	8	9	10	11	12	13
i	IF	ID	RR	EX	M1	M2	WB	<- EX stage has fault					
i+1		IF	ID	RR	EX	M1	M2	WB	<- Inst Decode has Illegal Instruction				
i+2			IF	ID	RR	EX	M1	M2	WB				
i+3				IF	ID	RR	EX	M1	M2	WB			
i+4					IF	ID	RR	EX	M1	M2	WB		
i+5						IF	ID	RR	EX	M1	M2	WB	
i+6							IF	ID	RR	EX	M1	M2	WB
i+7								IF	ID	RR	EX	M1	M2

What is the maximum number of exceptions that could happen simultaneously in the above machine? Why?