(1) [2] What are the two primary goals of a compiler (in order)?

(2) [3] What is one of the simplest (and oldest) techniques for exploiting parallelism among instructions?

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

(4) [4] 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] 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.

(6) [6] Clock rates have grown by almost 2 orders of magnitude compared to the increase in power consumption. How was this accomplished? Why is this technique no longer effective?

(7) [6] 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) [6] Benchmarks need to change over time. Why?

(9) [6] What are the two main ways to define performance? How do they differ? Give an example task for each.

(10) [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 reasons - explain why.

(11) [4] What are the 3 types of data hazards? Which one can be solved by using forwarding?

(12) [4] What are the 3 pipeline hazards? Which one can be solved by providing more resources?

(13) [6] In your first design of a 4-stage pipeline (F,D,E,M/W), F takes 22 time units, D takes 26, E takes 50, and M/W takes 30.

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?

(14) [4] Processor A requires 500 instructions to execute a given program, uses 2 cycles per instruction, and has a cycle time of 4 ns. Processor B requires 4 cycles per instruction, but only requires 200 instructions to do the same program. What must the cycle time of Processor B be in order to give the same CPU time as Processor A? (Show your work)

(15) [4] An important program spends 60% of its time doing Integer operations, and 40% of its time doing floating point arithmetic. By redesigning the hardware you can either make the Floating Point unit 80% faster (take 20% as long), or the integer unit 50% faster (take 50% as long). Which should you do, and why?

(16) [5] You are responsible for designing a new embedded processor, and for a variety of reasons you **must** use a fixed 20 bit instruction size. You would like to support 64 different operations, use a 3-operand instruction format, and have 32 registers. If it is possible to do 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.

(17) [10] The standard MIPS has a 5-stage pipeline, and uses a load delay slot. If the machine is redesigned to be a 7-stage pipeline, with the following stages:

IF D1 D2/RR E1 M1 M2 WB (where **RR** stands for Register Read)

a) Assuming the machine has bypass/forwarding logic, how many load delay slots will this new design require if the memory returns the value at the end of M2? At the end of M1?

b) How many branch delay slots will this new design require, assuming the branch condition is calculated and available at the end of E1?

c) As long as the compiler does not move any code around, does this machine have to worry about WAW hazards? Why or why not?

(18) [10] In an MOS device, there is a gate, drain, and source. Briefly explain how this device works. Drawing a picture is recommended.

(19) [9] We have talked about the cycle by cycle steps that occur on interrupts. For example, here is what happens if there is an illegal operand detected in the D stage of instruction i (note this machine has a 7 stage pipeline):

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

Fill out the following table if instruction i+1 experiences a fault in the M1 stage:

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

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