(1) [2] What is one of the simplest (and oldest) techniques for exploiting parallelism among instructions?

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

(3) [3] When we talk about the number of operands in an instruction (a 1-operand or a 2-operand, for example), what do we mean?

(4) [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?

(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) [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?

(7) [4] Why are there multiple dies per silicon wafer? Why not just fabricate one huge die per wafer?

(8) [3] Which is a better metric for comparing processors, energy or power? Why?

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

(12) [4] Why do most current chips feature multiple cores on the same die?

(13) [4] What are the 3 pipeline hazards? Which one can be solved by providing more resources?

(14) [6] In your first design of a 5-stage pipeline (F,D,E,M,W), F takes 24 time units, D takes 26, E takes 25, M takes 20, and W takes 5.

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?

(15) [4] Processor A requires 400 instructions to execute a given program, uses 3 cycles per instruction, and has a cycle time of 2 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)

(16) [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 40% faster (take 60% as long). Which should you do, and why?

(17) [6] You are responsible for designing a new embedded processor, and you have been told you are restricted to using a fixed 20 bit instruction size. To make this work you have decided to use a 2-operand instruction format, 128 registers, and support 64 instructions. Your boss just came in and said things have changed, and you can now use 24 bits - how would you change your instruction format? Be sure to explain your reasons.

(18) [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:

IF1 **IF2** **D/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?

[15] Here is a code sequence.

```
lw      R3, 8(R10)

add     R4, R2, R3

sub     R1, R4, R2

lw      R4, 20(R0)
```

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.

(19) [10] 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 ID stage of instruction i (note this machine has a 7 stage pipeline, RR=Register Read):

	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 EX stage (assume imprecise interrupts):

	1	2	3	4	5	6	7	8	9	10		
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			
i	IF	ID	ID/RR	EX	M1	M2	WB	<- M1 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?