[2] The power consumed by a chip has increased over time, but the clock rate has increased at a far greater rate. How was this possible? How did designers keep the chip from melting?

[2] What is "leakage" current? If Vdd is lowered, what happens to the amount of leakage current?

[3] Area on the die used to be the most critical design constraint. That is no longer true - what is the most critical factor now? Why?

[2] Why is it so difficult for the processing elements on a CMOS-based chip to communicate with things that located off the chip?

[2] What do we mean when we say something is an N-operand machine?

[2] There are two main ways to define performance - what are they?

[3] What is a benchmark program? Do benchmark programs remain valid indefinitely? Why or why not?

[3] Why is it difficult to come up with good benchmarks for parallel processors?

[3] In your first design of a 5-stage pipeline (F,D,E,M,W) F takes 25 time units, D takes 22, E takes 50, M takes 35 and W takes 15. What will the clock rate be for this pipeline? Is it a balanced? If not, explain what you could do to fix it.

[3] An important program spends 80% of its time doing Floating Point operations, and 20% of its time doing integer arithmetic. By redesigning the hardware you can make the Floating Point unit 30% faster (take 70% as long), or you can make the Integer Unit two orders of magnitude faster (take 1% as long). Which should you do and why?

[3] Processor A requires 250 instructions to execute a given program, uses 4 cycles per instruction, and has a cycle time of 6 ns. Processor B requires 2 cycles per instruction, and requires 300 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?

[2] Which data hazard occurs when instructions are allowed to complete out of order? Which one occurs when instructions are allowed to issue out of order?

[1] Which is on average more accurate, static or dynamic branch prediction?

[1] One way to deal with control hazards is to predict if they should be taken or not. What else is needed in addition to a prediction?

[2] Give a one-word definition of coherence, and a one-word definition of consistency.

[3] Supporting precise interrupts in a machine that allows out of order completion is a challenge. Why do we care? Why is it important to support precise interrupts?

[4] What is the primary difference between superscalar and VLIW processors? Give two advantages to using a superscalar processor, and 1 advantage to using a VLIW.

[3] Speculation is a very useful technique for improving performance. However, it is not being used as extensively as it once was - why not?

[2] What is the definition of a basic block? Why is there a desire to create a bigger one?

[10] For each technique in the following table, indicate how (on average) the 3 terms of the AMAT equation are affected. The first one is done for you as an example.

Technique	Hit Time	Miss Rate	Miss Penalty
Decreasing Cache Size	**decreases**	**increases**	**unaffected**
Increasing Line/Block Size			
Increasing Associativity			
Compiler Optimizations			
Hardware Prefetching			
Compiler Controlled Prefetching			
Victim Cache			
Multilevel Cache			
Virtually Addressed Cache			
Non-blocking Cache			
Using Critical Word First			

[1] Using a different mapping scheme will reduce which type of cache miss?

[1] Which type of cache miss can be reduced by using longer lines?

[1] Which type of cache miss can be reduced by using shorter lines?

[2] Some instruction sets are easier to pipeline than others. Give one thing an instruction set should have in order to make it easier to pipeline.

[2] What are the two biggest challenges to obtaining a substantial decrease in response time when using a MIMD parallel processor?

[2] What makes one instruction set harder to write a virtual machine for than another one?

[3] Which is easier to write a program for, a shared memory machine or a message passing machine? Why?

[2] Which is more expensive to build - a shared memory machine, or a message passing machine? Why?

[3] Compilers have a hard time guaranteeing program correctness when doing static scheduling. Why is that? Give an example of a case where something might go wrong. (Hint - think Memory System)

[4] Give 2 examples of things you might avoid using or things you would do differently if you are writing software in C for a heavily pipelined machine, and performance (response time) was extremely important.

[8] The MIPS implementation we used in class has a 5-stage pipeline, writes to the register file during the first half of the 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 a 7-stage pipeline, with the following stages:

F 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 E1 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 E1) assuming the memory returns the value by the end of M1? M2?

[3] You are responsible for designing a new embedded processor, and for a variety of reasons you must use a fixed 24 bit instruction size. You would like to support 128 different opcodes, use a 3-operand instruction format, and have 64 registers. If it is possible to do this, draw what an instruction would look like. If it is not possible, explain why, and give at least 2 different ways to solve the problem.

[3] Assuming a 21-bit address and a 512-byte Direct Mapped cache with a linesize=8, show how an address is partitioned/interpreted by the cache.

[3] Assuming a 21-bit address and a 160-byte 5-way SA cache with a linesize=4, show how an address is partitioned/interpreted by the cache.

[2] Assuming a 21-bit address and a 178-byte FA cache with a linesize=2, show how an address is partitioned/interpreted by the cache.

[2] Given a 4 Megabyte physical memory, a 36 bit Virtual address, and a page size of 1K bytes, write down the number of entries in the Page Table, and the width of each entry.

[4] Given a 1 Megabyte physical memory, a 36 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?

[10] Here is a code sequence.

```
        sw      R1, 4(R10)

Label:  lw      R2, 0(R10)

        lw      R3, 8(R2)

        add     R4, R2, R3

        sw      R3, 20(R2)

        sw      R4, 24(R0)

        lw      R4, 4(R0)

        add     R5, R1, R3

        subi    R5, R5, 24
```

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

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

c) Reorder the code to remove as many stalls as possible (assuming there is forwarding logic). How many NOPs are left (if any)?

In this question, we are going to wire up a 12-bit processor. The machine is word-addressable, where a word is 12 bits. Immediates are sign extended, Offset is not. The machine has 3 different instruction formats: R, I, and J. Memory takes a single cycle to return a value.

R-type:	*(Arithmetic and logical:*				*rd = rs1 OP rs2)*
Opcode	rd	rs1	rs2	funct	
11-8	7-6	5-4	3-2	1-0	
I-type:	*(Arithmetic and logical:*				*rd = rs1 OP Immediate)*
	(Load:				*rd = mem[rs1+Immediate])*
	(Store:				*mem[rs1+Immediate]=rd)*
Opcode	rd	rs1	Immediate		
11-8	7-6	5-4	3-0		
J-type:	*(*				*PC = Offset)*
Opcode	Offset				
11-8	7-0				

The ALU can perform 6 functions, written this way: OP [ALU0 ALU1 ALU2]
Increment X [001], Add [010], SUB [011], AND [100], OR [101], NOT [110]

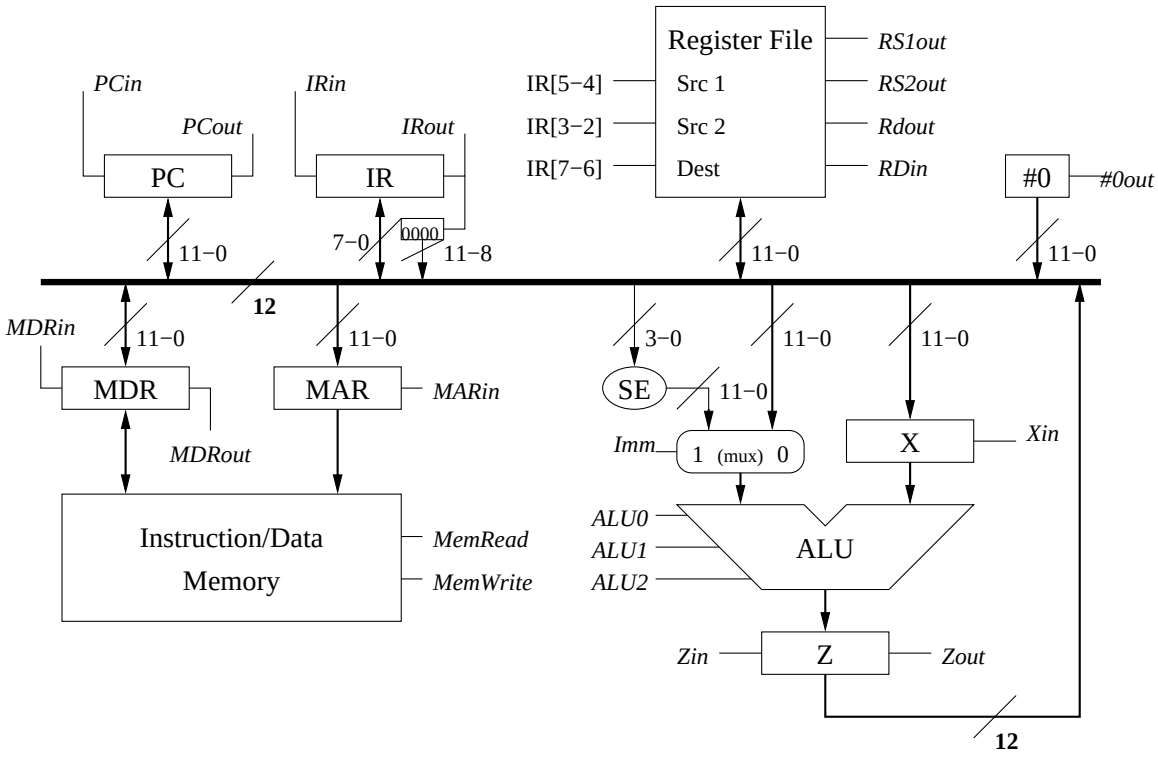
Here are some of the instructions that have been defined:

Name	**Opcode(Funct)**	**Name**	**Opcode(Funct)**	**Name**	**Opcode(Funct)**
NOP	0000(00)	lw	0001(xx)	sw	0011(xx)
NOT	1000(00)	beqz	0100(xx)	j	0101(xx)
AND	1000(01)	AND Imm	1001(xx)	ADD	1100(01)
OR	1000(10)	OR Imm	1010(xx)	ADD Imm	1101(xx)
XOR	1000(11)	XOR Imm	1011(xx)	SUB	1100(01)

Here are the 21 control signals.

PCin	PCout	IRin	IRout	MDRin	MDRout	MARin
Zin	Zout	RDin	RDout	MemRead	MmWrite	Imm
RS1out	RS2out	#0out	ALU0	ALU1	ALU2	Xin

Here is a diagram of the machine.



[6] Fill in the microcode steps necessary to perform an instruction fetch (incrementing the PC is considered part of fetch).

Step	IRin	PCin	MARin	MDRin	RDin	Xin	Zin	IRout	PCout	MDRout	Zout	RDout	RS1out	RS2out	#0out	ALU0	ALU1	ALU2	Mread	Mwrite	Imm
0																					
1																					
2																					
3																					
4																					
5																					

[2] Write down the binary bit pattern (the bit pattern that will be in the IR after you have done the instruction fetch) for the instruction: **SW R2, R1, 7**

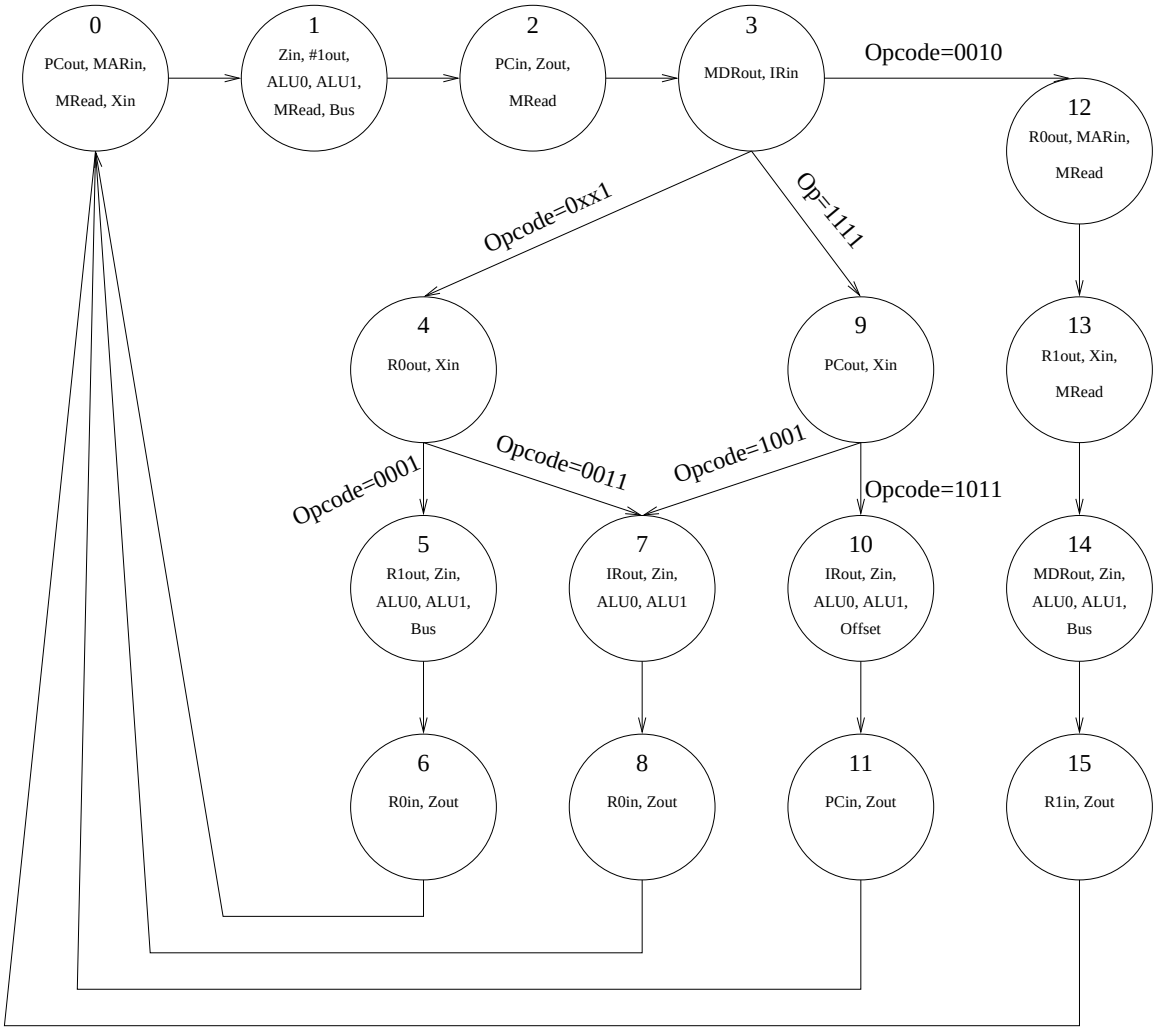
[6] Now that you have done the instruction fetch, fill in the microcode steps necessary to perform the following instruction: **SW R2, R1, 7**

Step	IRin	PCin	MARin	MDRin	RDin	Xin	Zin	IRout	PCout	MDRout	Zout	RDout	RS1out	RS2out	#0out	ALU0	ALU1	ALU2	Mread	Mwrite	Imm
0																					
1																					
2																					
3																					
4																					

[6] Find the basic blocks in this code (if there are any), and circle them.

```
label1:   lw    R2, 0(R10)
          lw    R3, 8(R1)
          add   R4, R3, R2
          sw    R4, 4(R10)
          bnez  R4, label3
label2:   sw    R4, 20(R4)
          sw    R4, 24(R0)
label3:   lw    R1, 4(R0)
          add   R5, R1, R4
          beq   R5, R6, label2
```

Here is the state diagram for a random machine:



[4] Assuming there are 4 state variables (Y3-Y0), that State0=!Y3*!Y2*!Y1*!Y0 (0000) and State15=Y3*Y2*Y1*Y0 (1111), write down the exact boolean equation for the **PCout** signal.

[2] Assuming the same situation as in the previous question, write down the exact boolean equation for **NextState10**.

[6] Suppose I have a 5-issue multithreaded machine, and there are 3 threads - A, B, and C. Assuming:

The number of independent instructions Thread A can find (in order): 2, then 3, then 0, then 2
The number of independent instructions Thread B can find (in order): 1, then 0, then 2, then 3
The number of independent instructions Thread C can find (in order): 2, then 1, then 3, then 0

Fill in the following table if coarse grained scheduling is being used.

Time	Slot1	Slot2	Slot3	Slot4	Slot5
0					
1					
2					
3					

Now fill in the following table assuming the use of fine-grained scheduling.

Time	Slot1	Slot2	Slot3	Slot4	Slot5
0					
1					
2					
3					

Now, repeat the process assuming simultaneous multithreading is being used.

Time	Slot1	Slot2	Slot3	Slot4	Slot5
0					
1					
2					
3					

[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 (note this machine has a 7 stage pipeline):

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

Fill out the following table if instruction i+1 experiences a fault in the M1 stage (Page Fault, for example):

	1	2	3	4	5	6	7	8	9	10			
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

What happens in this case?

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

What is the maximum number of exceptions that could happen at one time in the above machine? Why?