1. (2 pts) What is a "balanced" pipeline?

2. (2 pts) What are the two main ways to define performance?

3. (2) What is the difference between fixed and hybrid instructions?

4. (2 pts) Clock rates have grown by a factor of 1000 while power consumed has only grown by a factor of 30. How was this accomplished?

5. (2 pts) When dealing with control hazards, a prediction is not enough - what else is necessary in order to eliminate stalls?

6. (2) Why are there multiple dies per silicon wafer? Why not just fabricate one huge die per wafer?

7. (2 pts) What piece of hardware differentiates correlating predictors from other predictors?

8. (3) What do we mean when we say something is an N-operand machine?

9. (3 pts) What is a benchmark program? Do benchmark programs remain valid indefinitely? Why or why not?

10. (3 pts) Give 3 examples of dynamic branch predictors, in increasing order of accuracy.

11. (3) The multicycle mips uses a single memory, but the pipelined mips has two. Why is that?

12. (3) What are the 3 hazards that pipelined machines have to deal with?

13. (3 pts) Processor A requires 500 instructions to execute a given program, uses 2 cycles per instruction, and has a cycle time of 9 ns. Processor B only requires 100 instructions to do the same program, and has a cycle time of 10 ns. How many cycles per instruction must Processor B have in order to give the same CPU time as Processor A? (Show your work)

14. (4) You are in charge of designing a new embedded machine, and you must use a fixed 10 bit instruction size (for a variety of reasons). You would like to support 16 different operations, use a 3-operand instruction format, and have 8 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.

15. (14 pts) Here is a code sequence.

lw R1, 0(R10)

lw R2, 4(R10)

add R3, R2, R1

sw R3, 20(R0)

lw R4, 4(R10)

add R5, R1, R4

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 cycles does it take now?

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.

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 4 functions, written this way: OP [ALU0 ALU1]
Add [11], AND [01], OR [10], NOT [00]

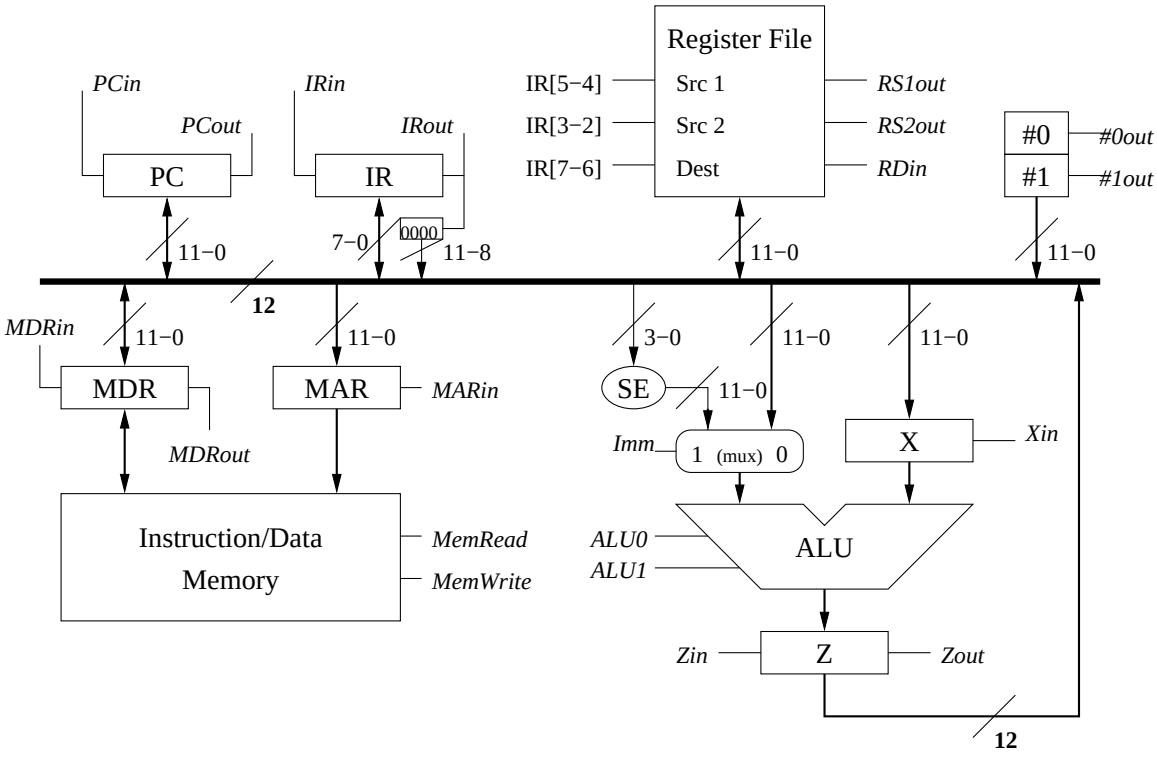
Here are some of the instructions that have been defined:

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	#1out	ALU0	ALU1	Xin

Here is a diagram of the machine.



16. (8) Fill in the microcode steps necessary to perform an instruction fetch.

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

17. (8) Now that you have done the instruction fetch, fill in the microcode steps necessary to perform the following instruction: SW R3, 9(R1)

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

18. (4 pts) Suppose we want to use microcode to provide the control signals for this machine. Assuming the longest instruction takes 8 cycles and the simplest microcode configuration,

a. (1) How many entries would the microcode memory have?

b. (1) How wide would each entry be?

c. (2) What would be contained in each memory location?

19. (9) The standard MIPS has a 5-stage pipeline, and uses a branch delay slot. If the machine is redesigned to be a 10-stage pipeline, with the following stages:

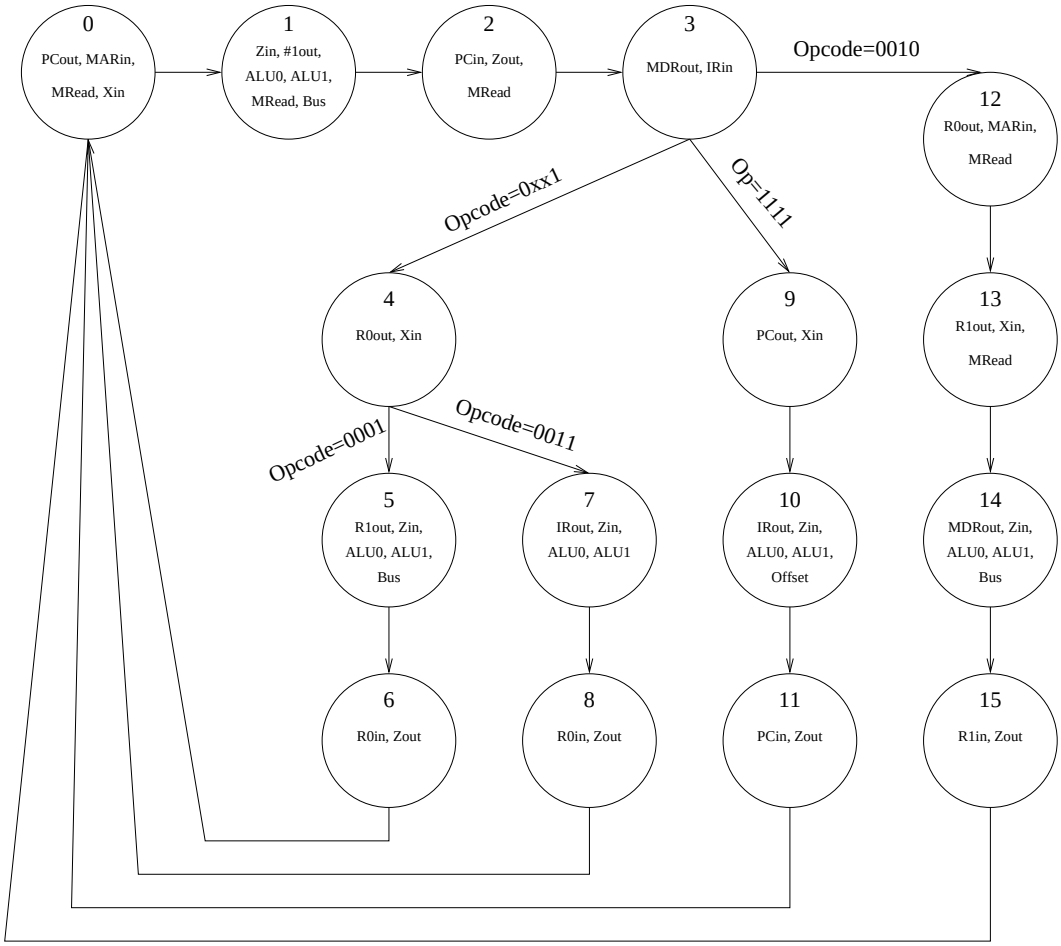
F D1 D2 RR E1 E2 M1 M2 M3 WB (where RR stands for Register Read)

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

b. How many branch delay slots if the branch condition is calculated and available at the end of E2?

c. How many load delay slots would this machine need (assuming it has forwarding logic)?

Here is the state diagram for a random machine:



20. (4) Assuming there are 4 state variables (Y3-Y0), that State0=!Y3*!Y2*!Y1*!Y0 (000) and State15=Y3*Y2*Y1*Y0 (1111), write down the exact boolean equation for the IRout signal.

21. (4) Assuming the same situation as in the previous question, write down the exact boolean equation for NextState7.

22. (4) If you were using a minimized microcode configuration for this machine, circle on the diagram where the dispatch roms points are.

23. (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 6 stage pipeline):

	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

---

24. (3) What is the maximum number of exceptions that could happen at one time in the above machine? Why?