1. (2 pts) What are the two main ways to define performance?

2. (2 pts) What is Amdahl’s law, in words?

3. (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?

4. (2 pts) What is a "balanced" pipeline?

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

6. (4 pts) Processor A requires 270 instructions to execute a given program, uses 1 cycles per instruction, and has a cycle time of 10 ns. Processor B requires 3 cycle per instruction, but only requires 100 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)

7. (6 pts) What are the 4 benchmark types we discussed in class? Do benchmark programs remain valid indefinitely? Why or why not?

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

9. (3 pts) Give 3 examples of dynamic branch predictors, in increasing order of accuracy.

10. (2 pts) What piece of hardware differentiates correlating predictors from other predictors?

11. (6 pts) In MIPS, branches and jumps are handled differently. Give 3 ways in which they differ.

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

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

How many branch delay slots will this new design require, assuming the branch condition is calculated and available at the end of E1? At the end of E2?

13. (12 pts) Here is a code sequence.

lw R2, 8(R10)

add R4, R2, R1

add R1, R2, R4

sw R2, 20(R0)

and R3, R5, R6

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 no 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 [00], AND [01], OR [10], NOT [11]

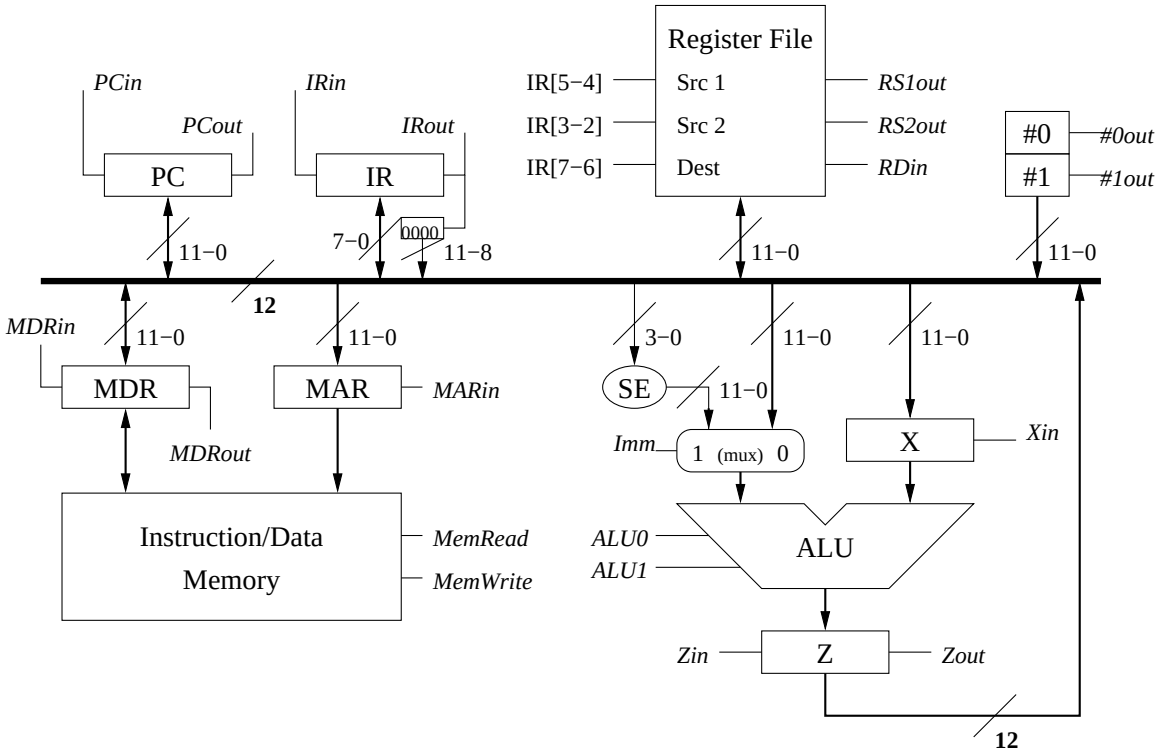
Here are some of the instructions that have been defined:

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.



14. (6) Fill in the microcode steps necessary to perform an instruction fetch.

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

15. (6) Now that you have done the instruction fetch, fill in the microcode steps necessary to perform the following instruction: SW R2, 5(R0)

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

16. (6 pts) Suppose we want to use microcode to provide the control signals for this machine. Assuming the longest instruction takes 9 cycles, sketch the simplest microcode configuration that can be used in this machine (how many entries, how wide is each entry, what does each entry contain, etc.)

    1. (1) How many entries would it have?

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

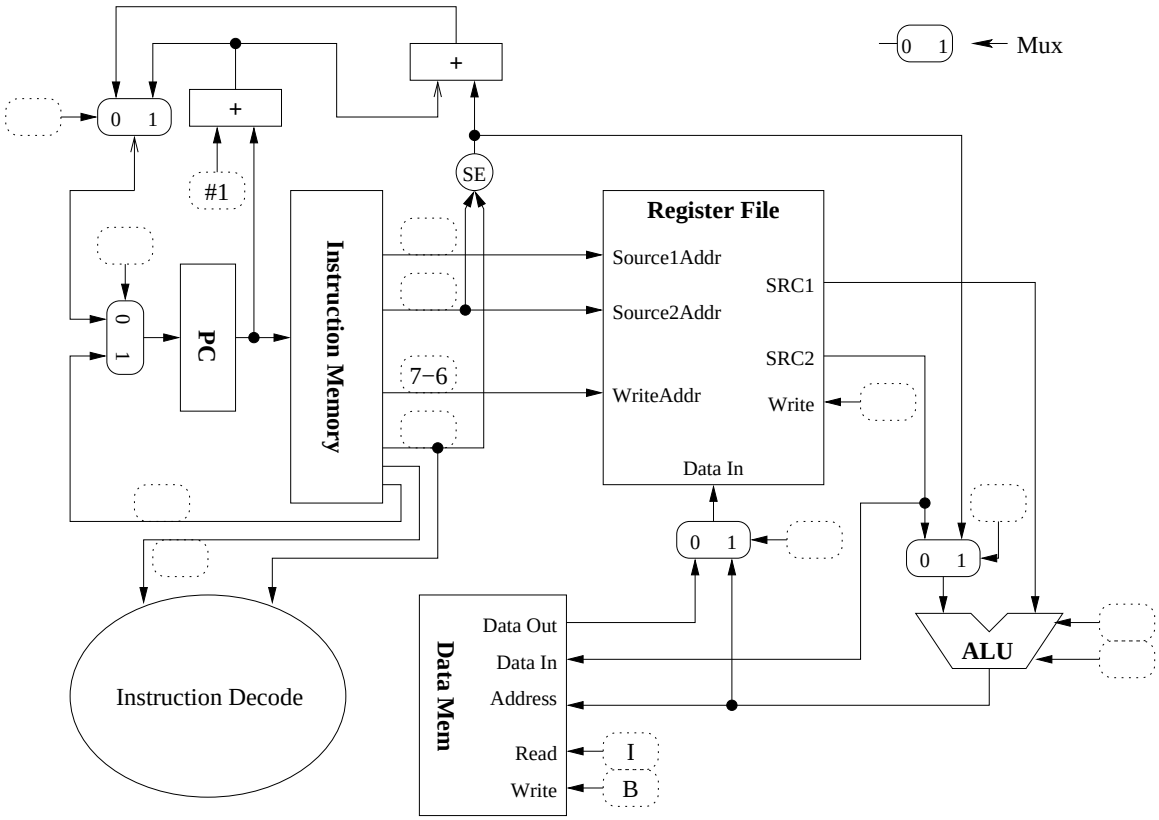
    3. (2) What would be contained in each memory location?

    4. (2) How would the memory locations be accessed? (In other words, what signals would you use to create the memory address?)

17. (6 pts) This multi-cycle 12-bit machine is very similar to the MIPS multi-cycle 32-bit design in the book. List 3 significant differences between the two designs (and obvious, trivial things like "one is 32 bits and one is 12" don't count.)

This is the same machine, redone so that it takes a single cycle to execute each instruction. Note that the control signals are now:

A) ALU0	C)ALU1	E) ALUSrc	G) Br	I) MemRead
B) MemWrite	D) PCin	F)RegWrite	H) WrAddr	J) WrSrc



18. (6) In the diagram above, make sure all the dashed boxes are filled. Several are already done for you. Use the corresponding letters instead of the actual names (F instead of RegWrite, for example).

19. (3) Now, write down the exact boolean equation for the MemRead signal.

20. (4 pts) We need a JAL instruction - let's say it uses register 0. Sketch the changes necessary on the above diagram, and/or describe in words what would need to change in this machine in order to make this work. (Don't forget about instruction set impacts ...)

21. (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

---

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