1. (2) In MIPS, what is the difference between and exception and an interrupt?

2. (2 pts) There are two different ways to measure performance. What are they?

3. (2 pts) Why doesn’t MIPS have a subtract immediate instruction?

4. (3 pts) Write down the 3-term CPU performance equation developed in class.

5. (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?

6. (2pts) The 32-bit quantity 0x12345678 is going to be stored in memory at location 0x100. If the memory is byte-addressable and uses a little-endian format, what does memory location 0x100 contain after the store occurs?

7. (6 pts) In MIPS, the jump instruction uses the distance (or Immediate) field to indicate the distance from the current PC that it wants to jump.

   (2) What is that distance measured in?

   (2) How is this accomplished?

   (2) What must be done if you need to jump further than that distance?

8. (6 pts) What are the 4 benchmark types we discussed in class? Do benchmark programs remain valid indefinitely? Why or why not?

In class we developed a 12-bit machine. In this question, we are going to wire the advanced version of it up.

The machine has 3 different instruction formats: R, I, and J.

R-type:

Opcode	rs	rt	rd	funct
11-9	8-7	6-5	4-3	2-0

I-type:

Opcode	rs	rt	Immediate
11-9	8-7	6-5	4-0

J-type:

Opcode	Offset
11-9	8-0

The machine is word-addressable, where a word is 12 bits. Immediates are sign-extended, and in a jump instruction, the jump is not relative to the current PC, but rather is treated as an absolute value.

There are 12 instructions:

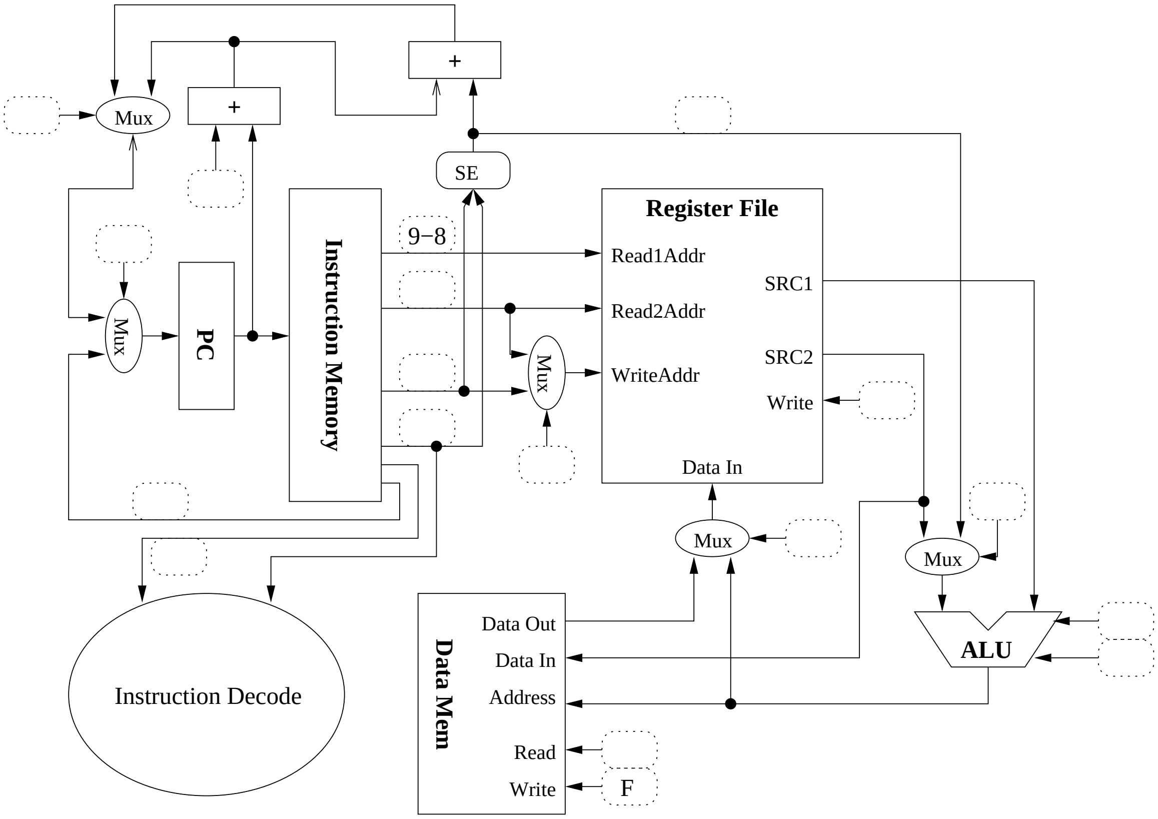
Name	**Opcode(Funct)**	**Name**	**Opcode(Funct)**	**Name**	**Opcode(Funct)**
lw	100(xxx)	sw	101(xxx)	NOP	011(000)
beqz	110(xxx)	j	111(xxx)	NOT	011(001)
ADD	011(000)	ADD Imm	000(xxx)		
AND	011(010)	AND Imm	001(xxx)		
OR	011(100)	OR Imm	010(xxx)		

Here are the 10 control signals.

A) ALU0 B) ALU1 C) ALUSrc D) Br E) MemRead

F) MemWrite G) PCin H) RegWrite I) WrAddr J) WrSrc

9. (35 pts) (32) In the diagram below, make sure all the dashed boxes are filled. Two are already done for you. Use the letters from the previous page instead of the actual names (F instead of MemWrite, for example).



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

10. (6 pts) We need a JAL instruction - let's say it uses register 3. 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 ...)

11. (13 pts) We can use boolean logic to create the various control signals. However, we can also use a simple memory (since as we discussed in 154A, a memory is essentially a truth table). If we were to use a memory, answer the following questions (It is OK to draw a figure to help convince me you know what you are doing):

(2) How many entries would it have?

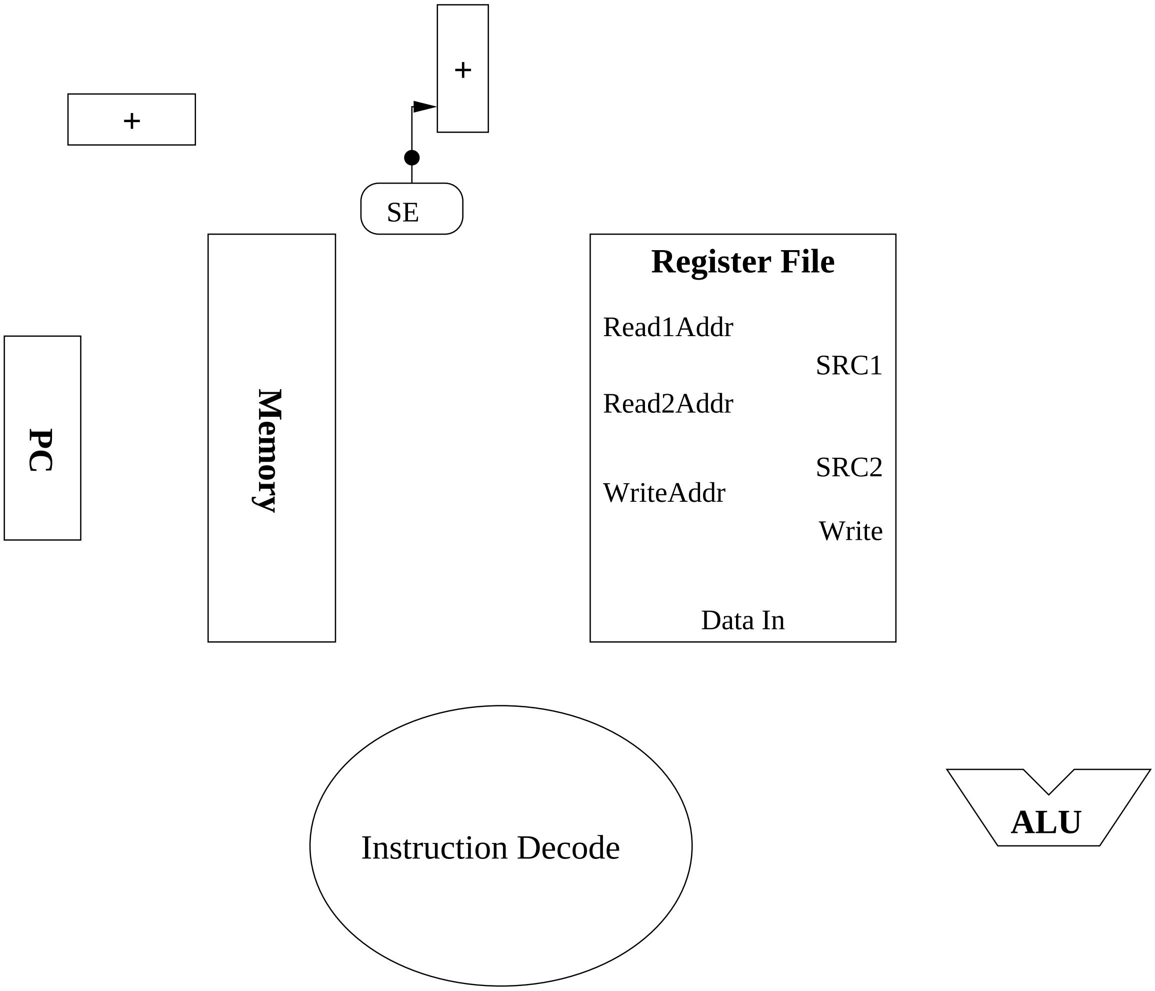
(2) How wide would each entry be?

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

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

(5) You can reduce the amount of memory necessary by 75% if you use a very small amount of combinational logic. Show/explain how that would be done.

12. (13 pts) There are a number of changes that must be made to our single cycle design in order to make it a multicycle CPU. What are the 5 new registers that must be added, and were do they go? (Sketch them in on the diagram below). Also, there are several new control signals - list 3 of them, and explain what they do.



13. (8 pts) We are going to use microcode to provide the control signals for this multicycle CPU. We will say there are 13 control signals, and we are going to have the longest instruction take 5 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. (2) How many entries would it have?

2. (2) 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?)