1. (3 pts) What are the three types of hazards pipelined machines must deal with?

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

3. (2 pts) What is Amdahl's law, in words?

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. (2 pts) What concept is at the heart of RISC processing?

7. (3) What is Register Renaming, and why is it important?

8. (4 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.

   (1) What is that distance measured in?

   (1) How is this accomplished?

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

9. (6 pts) What are the 4 benchmark types we discussed in class? Do benchmark programs remain valid indefinitely? Why or why not?

10. (4) Supporting precise interrupts in machines that allow out of order completion is a challenge. Briefly explain why.

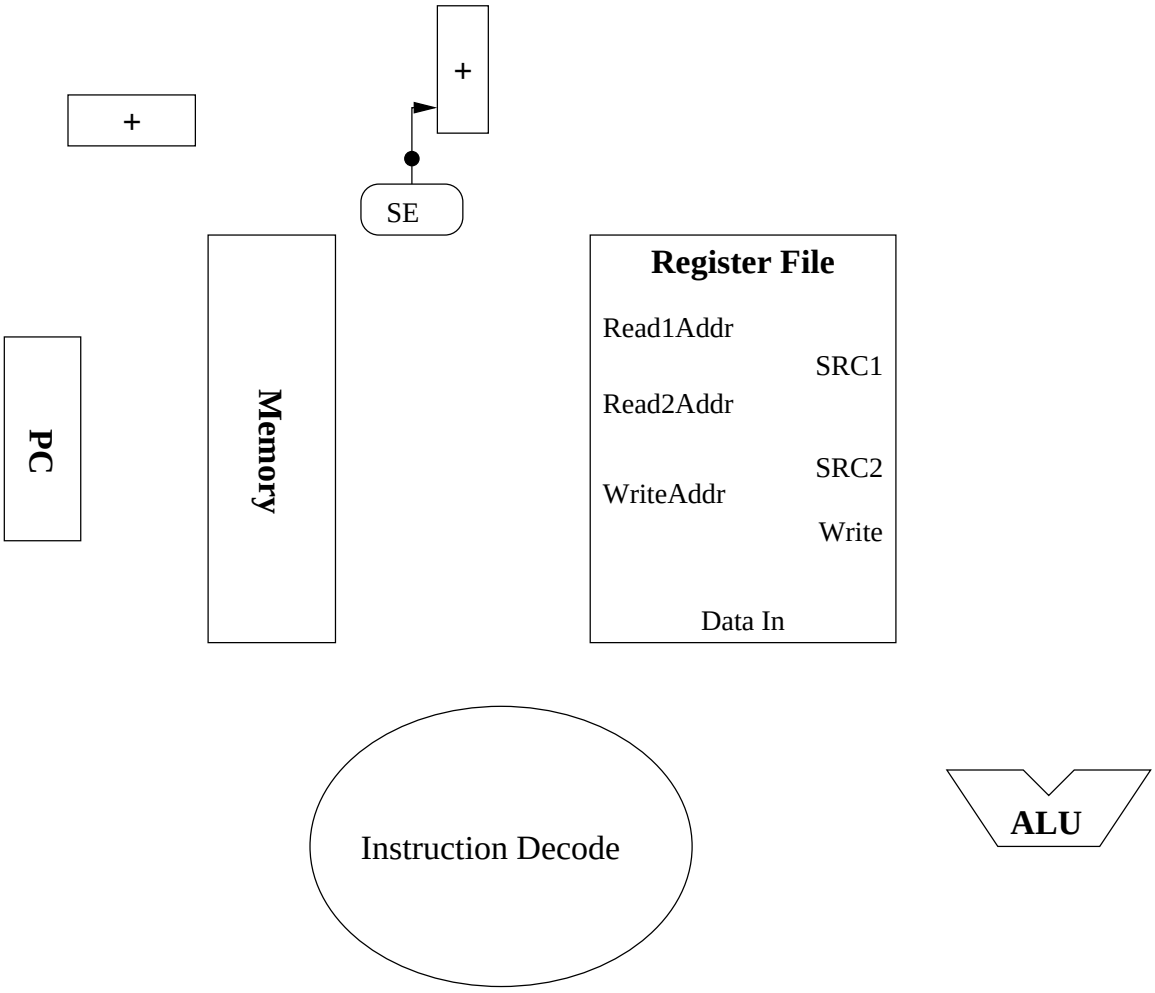
11. (4) Why are there multiple dies per silicon wafer? Why not just fabricate one huge die per wafer?

12. (3 pts) What is the advantage of using a fixed-size instruction?

13. (5 pts) What is a dispatch table? Where (and why) is it used?

14. (6) There are two circuit blocks we added when pipelining our processor, the forwarding logic block and the hazard detection block. Describe the problem the hazard detection logic block is designed to solve, and write down the equation it uses (approximately - it doesn't have to be exact, but it has to demonstrate that you know what is going on).

15. (11 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.



16. (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?)

17. (4) 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 during E1? During E2?

18. (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+1:

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| i | IF | ID | EX | MEM | WB | | | | |
| i+1 | | IF | ID | EX | MEM | WB | <- Interrupt detected | | |
| i+2 | | | IF | ID | EX | MEM | WB | <- Instruction Squashed | |
| i+3 | | | | IF | ID | EX | MEM | WB | <- Trap Handler fetched |
| i+4 | | | | | IF | ID | 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 | EX | MEM | WB | | | | | |
| i+1 | | IF | ID | EX | MEM | WB | | | | |
| i+2 | | | IF | ID | EX | MEM | WB | | | |
| i+3 | | | | IF | ID | EX | MEM | WB | | |
| i+4 | | | | | IF | ID | EX | MEM | WB | |
| i+5 | | | | | | IF | ID | EX | MEM | WB |

---

What happens in this case?

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| i | IF | ID | EX | MEM | WB | <- Execution stage has overlflow | | | | |
| i+1 | | IF | ID | EX | MEM | WB | <- Inst Read causes Page Fault | | | |
| i+2 | | | IF | ID | EX | MEM | WB | | | |
| i+3 | | | | IF | ID | EX | MEM | WB | | |
| i+4 | | | | | IF | ID | EX | MEM | WB | |
| i+5 | | | | | | IF | ID | EX | MEM | WB |

In this question, we are going to wire up a 14-bit version of the machine used in the midterm.

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

R-type:

| Opcode | rs | rt | rd | funct |
|---|---|---|---|---|
| 13-10 | 9-8 | 7-6 | 5-4 | 3-0 |

I-type:

| Opcode | rs | rt | Immediate |
|---|---|---|---|
| 13-10 | 9-8 | 7-6 | 5-0 |

J-type:

| Opcode | Offset |
|---|---|
| 13-10 | 9-0 |

The machine is word-addressable, where a word is 14 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.

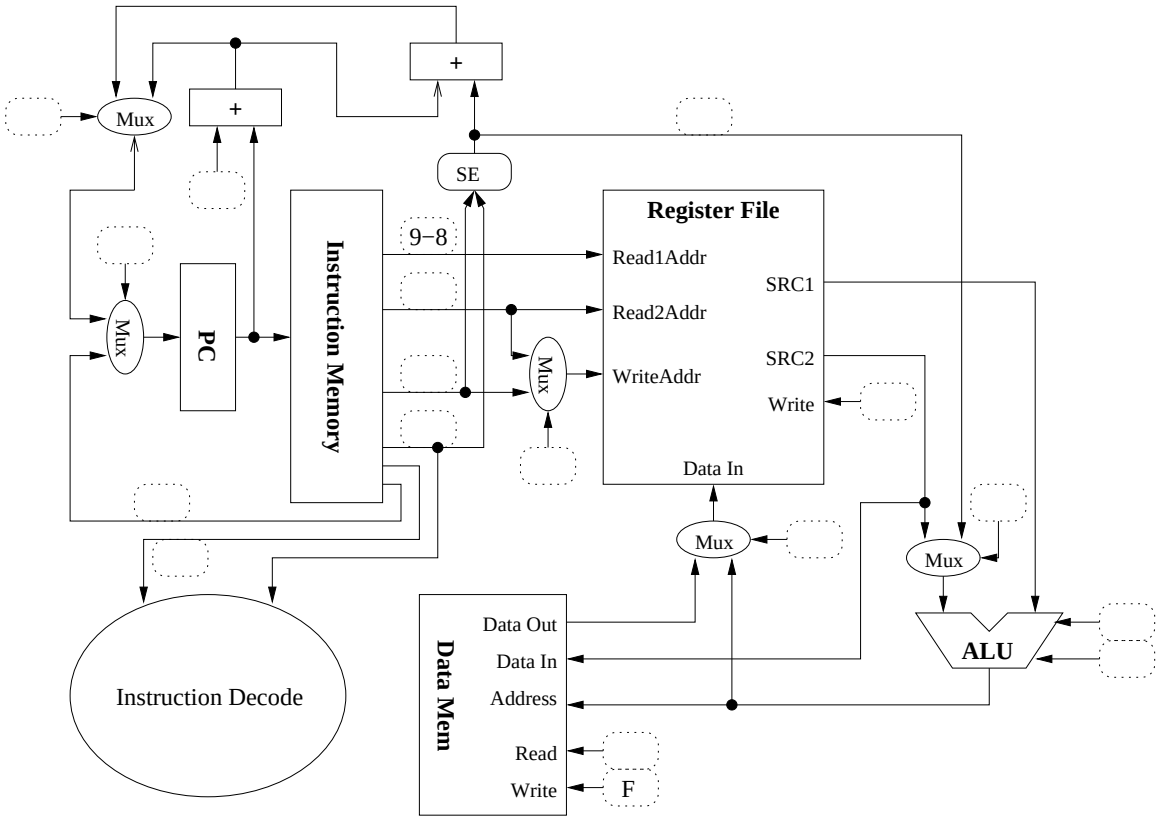
Here are some of the instructions that have been defined:

| **Name** | **Opcode(Funct)** | **Name** | **Opcode(Funct)** | **Name** | **Opcode(Funct)** |
|---|---|---|---|---|---|
| lw | 0010(xxxx) | sw | 0011(xxxx) | NOP | 0000(0000) |
| beqz | 0100(xxxx) | j | 0101(xxxx) | NOT | 1000(0011) |
| ADD | 1000(0000) | ADD Imm | 1001(xxxx) | SUB | 1000(0100) |
| AND | 1000(0001) | AND Imm | 1010(xxxx) | XOR | 1000(0101) |
| OR | 1000(0010) | OR Imm | 1011(xxxx) | XOR Imm | 1100(xxxx) |

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 |

19. (15) 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 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 ...)