

My worst scores were -5 points. I had two of them and I corrected them both.

Question 13:

Your Answer:

1. And instruction like lui could fill in the entire register. As of now it is 32 so you can't fill in all the spots but if it were halved it could fill in all of the values of the register.
2. There would be more spills and you would have to save more stuff onto the stack pointer.

1. incorrect -- you seem to be conflating register size / register count and/or immediate value size... -5

2.

Misconception:

I for some reason got confused and I answered part a about the register size instead of the number of registers. For the second part I was thinking about the register amount being decreased.

Correction:

There would be less registers to program with. This would probably bleed into the temporary registers or the saved registers. An example is in a program with 32 registers if you were using all of the temporary and saved registers and you moved over to a mips processor with less you would have to save values on the stack pointer. This means if the programmer has lots of variables or values they are working with they will need to use the stack pointer more often to save values. For part a the positive would be that you would not need as many bits for specifying what register. For a I type instruction this allows you to have more bits towards forming an immediate value extending your reach in memory. For example if the registers dropped to 16 you would only need 4 bits for rs and rt giving you 2 more bits in the immediate part of the instruction.

Question 17:

Your Answer:

- a. You should use the immediate value as you only need two registers and you want to have an immediate value as well for the location of the stack. The r type does not have an immediate part and the j does not have room for registers you want to operate on.
- b. No we don't have any implementation for the stack pointer yet in our implementation for lab 2. We also don't have l type instructions implemented. We also do not have the \$ra set up yet.

b. --> -5 errrr...you have all of those things setup (\$sp is \$29 with no special hardware, \$ra is 31 with no special hardware needed for reading as described, and addi is one of the required instructions for lab2)

B.

Misconception:

I was thinking about the stack being different from the actual memory for some reason.

Correction:

Yes, we can from Lab2. We have all of the 32 general purpose registers implemented including the \$sp register. We also have the alu with immediate values so we can -4 off of a register value. We also had the memory implemented so we could store R[rt]. Overall, we have all the components implemented in lab2 to be able to implement that instruction.