Computer Organization Quiz 2 – chap 2.

3/30 14:20 ~ 14:45

True or False (10 pts / problem)

1. (x) MIPS only offers the addition of two operands rather than more operands, such as the addition of three operands. The main advantage of this scheme is to follow the design principle: make common case fast.
2. (o) The difference between small Endian and big Endian is the sequence of storing the bytes of a word in the memory.
3. (x) The 32-bit MIPS CPU offers an immediate mode to the add instruction, where programmers can embed a constant of 32-bit in an add instruction. To have higher flexibility, the constant can be of 5 or 10 bit long.
4. (x) Consider the 2's complement signed integers of *n* bits. The number of 0 has two forms, one is +0 and the other is -0.
5. (o) Singed extension is to guarantee the integrity of translating a signed number of smaller number of bits to the same signed number in larger number of bits.
6. (o) The R-format of MIPS instructions contain 6 fields, including op code, rd, rs, rt, shift amount, and function code.
7. (o) In addition to obeying the design principle of making the common case fast, the I-format MIPS instructions follow the design principle that good design demands good compromises as well.
8. (x) As a function is invoked, OS allocates a memory space in the stack for storing the variables declared and used in this function. In MIPS, if variables *a* is declared in blocks *A*. In order to transfer the value of *a* to the caller, we can choose the transfer the value of *a* to the caller directly or transfer the memory address of *a* to the caller.
9. (o) For modern computer systems, a CPU often executes the processes of all users alternatively. In order to switch among all processes and execute each process correctly, a CPU needs to store its states, including the register contents, for the correct execution of a process.
10. (o) The dynamic linking can reduce the size of an executable file as compared to the static linking technique. The dynamic linking does not embed each library function used in a program into the executable binary file; on the contrary, the binary codes of invoked library functions will be loaded into memory only when they are invoked during the execution of a program.