Homework #6 — Due 11/30/98

1. What problems might occur if one interrupt is allowed to interrupt the interrupt handler of another interrupt?

   If the first interrupt handler is processing data, and it is interrupted before it has time to finish, it is possible that the second interrupt may take so long to process that the data may be lost (e.g., if new data comes along in the meantime).

   (Note — most people incorrectly answered this question by telling me that the first interrupt would get "lost" in some way. However, looking back at the text I see that this topic was not discussed there, or even in my written notes, only in what I said in class on 11/16, so I won't take off any credit for this answer, although it's technically not correct.)

Homework #6 — Due 11/30/98

2. Compare the following three methods for loading immediate values into a register: (i) the MOVE instruction discussed in class, (ii) the SPARC MOV instruction, and the Chapter 9 Simple Machine's SETLO instruction. (This question counts double.)

   The MOVE instruction discussed in Lecture 16 is a real instruction that loads an immediate value (of unspecified size) into a register.

   The SPARC MOV instruction (see Lecture 24) has similar functionality, but (i) it is a synthetic instruction, and (ii) it loads 13-bit immediate values.

   The S.M.'s SETLO instruction (see Lecture 28) is a real instruction that loads an 8-bit immediate value into a register.

Homework #6 — Due 11/30/98

3. In the datapath of the Chapter 9 Simple Machine, why does the PC value need to go to both Bus 2 and Bus 3?

   During an instruction fetch, the PC value must to go via Bus 3 to the MAR, so that the instruction at that address can be fetched.

   During the execution of a branch instruction, the PC value must go via Bus 2 to the ALU, so that it can be added to the branch offset to produce the new PC value.

Homework #6 — Due 11/30/98

4. How does the µPC and µIR compare to the "real" PC and IR?

   All CPUs have a PC and IR; only microcoded CPUs have a µPC and µIR.

   The PC holds the address of the next instruction in the main memory; the µPC hold the address of the next micro-instruction in the control store.

   The IR holds the current assembly language instruction being executed; the µIR hold the current micro-instruction being executed.