EECS 373 F98 Sample Exam 2 Questions

1. What purpose(s) does an ABI serve?

2. What is a device driver? What purpose does it serve?

3. What is the minimum set of registers that must be saved and restored by an ISR that does not call any functions and does not allow nested interrupts? What additional registers must be saved and restored if the ISR calls other functions? What additional registers must be saved and restored if the ISR allows nested interrupts?

4. Assume you are using the MPC823 SIU interrupt controller to implement prioritized nested interrupts. The SIU interrupt controller gives you only a few basic capabilities: the ability to mask any interrupt (by setting its bit in the SIMASK register) and a register (the SIVEC register) that provides a code that identifies the highest-priority unmasked interrupt. List the exact steps that you must follow (what bits must be set/cleared in which registers and in which order) in each ISR to allow prioritized nesting, i.e., only higher-priority interrupts can interrupt an executing ISR.

   (Hint: the CIPC does a lot of this for you in hardware; for the SIU interrupt controller, you have to do the same things only in software. Read Section 16.15.2.3 of the data book carefully, and think about what the CISR does for you. (Hint:^{2} it's the same thing as the 68000/SPARC IPL discussed in lecture.) Note that the CIPC modifies the CISR automatically when you write the IACK bit in the CIVR, and that normally there is no need to modify the CIPC mask register (CIMR). Since the SIU controller has no equivalent to the CISR, you have to fake it by modifying SIMASK.)

5. Given four 4Kx8 SRAM memory chips (each having twelve address lines and eight data lines), build a 16 Kbyte memory module starting at address 0xCD000000. Each SRAM chip has a chip select input ($\overline{\text{CS}}$) that tells the chip it's being addressed, an output enable input ($\overline{\text{OE}}$) that drives the contents of the addressed byte onto the data pins (if $\overline{\text{CS}}$ is also asserted), and a write enable ($\overline{\text{WE}}$) input that takes the value on the data pins and latches it into the addressed byte (again, if $\overline{\text{CS}}$ is also asserted). Assume the processor you're using has a 32-bit bus with byte enable signals and a RD/$\overline{\text{WR}}$ signal. Assume whatever bus timing is convenient. Show how your design handles aligned word, halfword, and byte reads and writes.

6. The MPC823 handles unaligned memory accesses in hardware by converting them to multiple aligned bus transactions. (This is why they're slower than aligned accesses.) For each combination of low-order address bits and access size that results in an unaligned access, describe the set of aligned bus transactions that the MPC823 must perform.

7. You have a sensor on your bike that generates a digital pulse for every revolution of the front wheel. Describe how you would use this signal plus an MPC823 to calculate the speed of your bike in miles per hour. (Focus on the configuration of the 823 timer/counter(s) and a general description of the software. Hint: all your code should be in ISRs.)