Dordt College
Engineering 304, Microprocessor Interfacing
Problems on Interrupts

3.17 **Interrupt Density and Interval Constraints**. Consider a design which has four sources of interrupts, all of which can be active at the same time. The sources are identified by numbers 1, 2, 3, and 4, with 1 representing the highest priority source and 4 the lowest. The corresponding interrupt service routines take T_1=31, T_2=29, T_3=37, and T_4=43 µs, respectively. The minimum time between interrupts from the same source is TP_1=120, TP_2=200, TP_3=150, TP_4=1000 µs, respectively. For reliable operation, each source must be serviced before its next interrupt occurs.

a.) Will operation be reliable? Assume further interrupts are disabled for the duration of each interrupt service routine. Explain your answer.

b.) If the mainline routine is to be permitted to include critical regions, during which interrupts are temporarily turned off, then how long can these last and still maintain reliable operation?

c.) For case (a), given critical regions of no longer than 13 µs, then how much could T_1 be lengthened and still ensure reliable operation.

d.) If the T_{Pi} times were actually average times instead of minimum times between the interrupts, then for case (a), what percentage of CPU time is available to the mainline routine, on the average?

3.19 **Critical Regions**. It is of paramount importance to be able to disable interrupts to protect critical regions in our code. Consider, for example, the common technique for changing a bit on an output port while leaving the other bits unchanged. We can read the port (even though it is an output port) into an accumulator. Then we can force the selected bit (e.g., bit 5) to 1 with an OR instruction (leaving the other bits unchanged). Finally, we can write the result back out to the port.

a.) Assuming that there is an interrupt service routine which changes bit 3 on this same port when it is called, why should interrupts be disabled for the above operation which is intended to change bit five? To answer this, describe a scenario which leads to a malfunction.

b.) How often is such an error likely to recur if bit 3 is changed once per millisecond, with the three instructions taking 1 µs each, and if bit 5 is changed once every 10 ms or so? Assume that the two events are not synchronized to each other, but that at least one of them occurs in response to an unsynchronized, external event.

c.) Is there any benefit to be had in an instruction which sets (or clears or toggles) selected bits of a port *directly*? Explain. (Not all microcontrollers have such an instruction, but some do, including both the Motorola 68HC11 and the Intel 8096.)

These problems are from Peatman, John B, *Design With Microcontrollers*, McGraw Hill, 1988.