ELEC 371 Interrupt Mini-Lab

- Review pseudocode specifications for an interrupt-based program that is intended to provide an initial experience in basic use the original timer interface in the DE0 Computer (which is also included in the new QUEENS Computer).
- Based on the pseudocode specifications, prepare a complete assembly-language program for the Nios II by introducing code in the appropriate places of a template assembly-language source file.
- To aid you in preparation of the program, consult page 4 of the Lab 1 description for references to appropriate parts of vendor-provided technical documentation where information on interrupt-based programming is available.
- *Students are directed to follow both general and interrupt-specific programming guidelines provided by the instructor.* The vendor-provided documentation illustrates certain technical details in its sample code, but you should closely follow the pseudocode specifications from the instructor and pursue the development of all assembly-language code in this course as described by the instructor, including a strong emphasis on modularity.
- Use the on-line simulation tools to test the execution of your completed program and ensure that the expected behavior (LED blinking on and off) is achieved. Because it is a simulation, the rate of blinking will certainly not be exact, and it may even vary from computer to computer. Nonetheless, if it blinks as expected, testing the same program on the hardware will have behavior with a precise rate of blinking for the LED.
- The *absolute minimum requirement* is to complete the steps above to acquire an awareness of timer functionality and programming. As stated in lectures, however, students will be expected to prepare code that handles multiple interrupt sources, as reflected in slides 24 to 26 of the Chapter 3 material. Therefore, the *actual expectation* is that the remaining steps below are also pursued as part of this independent learning exercise.
- Part 3 of Lab 1 was the first interrupt-based example involving pushbutton functionality. The preceding steps above have now separately explored timer interrupt capability. The final part of this independent learning exercise is prepare a new assembly-language program in a file combined.s that involves both pushbutton and timer interrupts.
- One important change for the combined program is to use different LEDs for the different interrupt sources. Therefore, when you copy your timer-related code into the new combined.s file, modify the timer interrupt handling to toggle bit 1 of the LEDs instead of bit 0. Let the pushbutton interrupt handling toggle LED bit 0 as done originally for Lab 1.

- Another important change for the combined program is to properly test for the different interrupt sources within the interrupt-service routine. For reference purposes, consult slides 24 to 26 in the Chapter 3 material, but also note that the instructor has intentionally included two interrupt sources in the first question of Tutorial 3, hence that tutorial discussion is highly relevant for this independent learning exercise.
- The initialization subroutine must now prepare for both interrupt sources in the I/O interfaces as well as in the ienable register.
- It is your choice as to whether to have dedicated subroutines for separate handling of the pushbutton and timer sources (again, see Chapter 3 material and Tutorial 3 for guidance), or whether to have a longer interrupt-service routine with all of the code for both sources contained within it.
- Furthermore, if you choose to use dedicated subroutines as indicated above, it is also your choice on whether to clear the interrupt source in the interrupt-service routine before calling the dedicated subroutine to perform the task(s) for servicing the interrupt, or whether to clear the interrupt source within the dedicated subroutine. The instructor has certainly emphasized modularity, but the primary consideration for this combined example is the proper handling of multiple interrupt sources.
- Either way, do not forget to clear the interrupt source.
- Test your combined.s with the simulator and confirm that the LED controlled by the timer interrupt blinks continuously whereas the LED controlled by the pushbutton only toggles in response to pressing/releasing the pushbutton (which in the simulator is clicking twice on the box that represents the pushbutton).