Sam Wimmer

Lab 2 5780

Prelab Questions:

1. What is the purpose of the NVIC peripheral?
    - The purpose of the NVIC peripheral is to enable and disable interrupts, recognizing the requests that are ready to be taken care of, cancel the pending requests and handling the interaction of multiple interrupts using priority.

2. What is the difference between interrupt tail-chaining and nesting?
    - The difference between these is that the tail-chaining is only a hardware ordering system that will do all the interrupts in order in which they occur which can result in heavier/larger interrupts to starve the smaller ones. Nesting is different because is both hardware and software and can let the more important interrupts to interrupt the smaller interrupts to execute interrupts by level of importance.

3. In what file are the CMSIS libraries that control the NVIC?
    - This is in the core_cm0.h file

4. What is the purpose of the EXTI peripheral?
    - The purpose is to allow non-peripheral sources to trigger interrupts.

5. What is the purpose of the SYSCFG pin multiplexers?
    - This peripheral controls the multiplexer; the importance of the multiplexer is that it is used to select the pins that are used to connect to the EXTI inputs.

6. What file has the defined names for interrupt numbers?
    - stm32f0xb.h – This file has the defined names for the interrupt numbers.
    -startup_stm32f072xb.s – This file has the names of the interrupt handlers.

7. What file has the Vector table implementation?

    - startup_stm32f072xb.s – the startup code and vector table are in this file.

Post Lab Questions:

1. Why can't you use both pins PA0 and PC0 for external interrupts at the same time?
   -These can't be used at the same time because all the PX0 pins are in the same multiplexor.
2. What software priority level gives the highest priority? What level gives the lowest?
   - The highest priority is 0 and the lowest priority is 3.
3. How many bits does the NVIC have reserved in its priority (IPR) registers for each interrupt (including non-implemented bits)? Which bits in the group are implemented?
   - There are 4 regions of 8 bits. Only the uppermost two bits of them are implemented.
4. What was the latency between pushing the Discovery board button and the LED change (interrupt handler start) that you measured with the logic analyzer? Make sure to include a screenshot in the post-lab submission.
   - The latency between the board and the LED change is 1.7 seconds as shown below.
5. Why do you need to clear status flag bits in peripherals when servicing their interrupts?

   - This bit needs to be set to show that the interrupt has been handled and that it is done and read to go back to the rest of the code or else it will continue to loop in the handler because if that bit isn't set it won't recognize that the interrupt has been handled.