Alex Baret
CS5780 Post-Lab 2

Lab 2 GitHub project:
https://github.com/alex-baret/CS5780/tree/master/Labs/LAB2

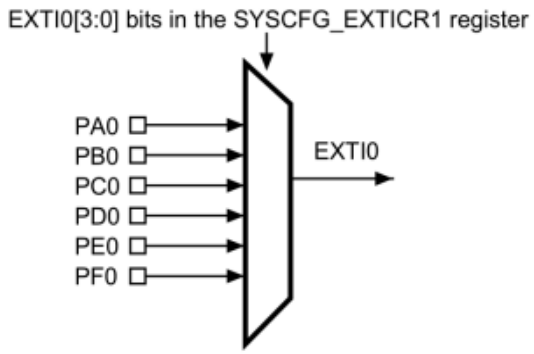
To find *main.c* within the GitHub repo, click the commit link -> click 'Browse files -> navigate to: Labs/LAB2/CS5780_LAB2_Alex_Baret/Core/Src/main.c

'At part 2.6, need to use logic analyzer' contains the state of main.c for checkoff 1
'Screenshot added, Checkoff Two complete' contains the state of main.c for checkoff 2

Post-Lab Questions

1. Why can't you use both pins PA0 and PC0 for external interrupts at the same time?

    - The SYSCFG pin multiplexers that the EXTI uses group external pins by their ordering within the GPIO peripherals. As a result of this grouping, PA0 and PC0 are both on the same multiplexer that defines the input to EXTI0. Since only a single pin from a group may be in use at a time, PA0 and PC0 can't be used for external interrupts at the same time.



2. What software priority level gives the highest priority? What level gives the lowest?

    - A software priority level of 0 gives the highest priority.
    - A software priority level of 3 gives the lowest priority.

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?

    - The NVIC has 32 bits reserved in its IPR registers for each interrupt (including non-implemented bits).
    - Only the uppermost two bits from those regions are implemented, giving four possible configurable priority levels (0-3)

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 pushing the Discovery board button and the LED change (interrupt handler start) measured by the logic analyzer was 198.6 microseconds.

5. Why do you need to clear status flag bits in peripherals when servicing their interrupts?

    - Status flag bits in peripherals need to be cleared when servicing their interrupts as they are used for pending interrupt requests so if the flag is not cleared the interrupt will repeat continuously because the request is never acknowledged as complete.