Review Questions - External Interrupts

Week 3

Review Questions - EXTI

1. Which System Exceptions have priorities that are preset?
   - The following System Exceptions have fixed, pre-assigned priorities:
     - Reset (priority -3)
     - Non-Maskable Interrupt (NMI) (priority -2),
     - Hard Fault (priority -1).

2. How do we set the System Exception priorities that are user-programmable?
   - System Exception priorities are set through the System Handler Priority registers – SYSPRI1, SYSPRI2 and SYSPRI3.

Review Questions - EXTI

3. What is the difference between edge-triggered and level-sensitive interrupts?
   - Level Interrupts:
     - A level-triggered interrupt is triggered as long as the interrupt signal is held active.
     - A peripheral can thus assert the interrupt signal until it no longer needs servicing. This can potentially hold up system resources unnecessarily.
   - Edge Interrupts:
     - An edge-triggered interrupt is sampled synchronously on the rising edge of the processor clock.
     - Peripheral requesting interrupt must asset interrupt for at least one clock cycle.

4. “Interrupts are asynchronous”. What do you understand by this statement?
   - It means that interrupts are events that can occur at any time, independent of clock phases.
   - Interrupts can be triggered at any time during program execution.

Review Questions - EXTI

5. What is the Interrupt Number for GPIO port C? What is the corresponding address in the Interrupt Vector Table that contains the pointer to the ISR?
   - GPIO Port C interrupt number = 2.
   - Address in Vector table = 64 + (2 × 4) = 72 = 0x48.

6. If there are multiple interrupts occurring at the same time, how are these presented to the ARM CPU?
   - When multiple GPIO interrupts occur, a single interrupt output is sent to the NVIC for the entire GPIO port. The NVIC prioritizes and routes ALL interrupts to the processor.

Review Questions - EXTI

7. How many levels of interrupt priority are supported by the NVIC? Explain how the number of levels comes about.
   - A total of 8 interrupt priority levels are supported. Interrupt priority is defined through a 3-bit field in the Interrupt Priority (PRI0 to PRI34) registers.

8. How many GPIO Interrupt registers are there?
   - There is a total of 7 GPIO interrupt registers:
     - Interrupt Sense register **(GPIOIS)**
     - Interrupt Both Edges register **(GPIOIBE)**
     - Interrupt Event register **(GPIOIEV)**
     - Interrupt Mask register **(GPIOIM)**
     - Mask Interrupt Status register **(GPIOMIS)**.
     - Raw Interrupt Status register **(GPIORIS)**.
     - Interrupt Clear register **(GPIOICR)**.

Review Questions - EXTI

9. Which register do we read in order to identify the source of a GPIO interrupt?
   - We can read either the Raw Interrupt Status (GPIORIS) or the Masked Interrupt Status (GPIOMIS).
   - Reading the GPIOMIS register shows interrupt conditions that are allowed to be passed to the NVIC (*interrupts that are not masked*).
   - GPIORIS register indicates that a GPIO pin meets the conditions for an interrupt, but it does not necessarily meant the interrupt has been sent to the NVIC.
   - Thus, the GPIORIS register will indicate status of the GPIO pin even if interrupt is not enabled.

10. Which register would clear an interrupt?
    - Use the Interrupt Clear (GPIOICR) register.
    - For edge-triggered interrupts, writing a '1' to the IC bit in the GPIOICR register clears the corresponding bit in the GPIORIS and GPIOMIS registers.
    - If the interrupt is a level-triggered, the IC bit in this register has no effect.

Review Questions - EXTI

11. Why do we need to clear an interrupt?
    - We need to clear an interrupt in the ISR so that the next interrupt can be triggered.

12. Why should an ISR be kept to execute as short a time as possible?
    - The time spent in an ISR should be kept as short as possible so as not to hold up system resources or cause other interrupt events to be held-up or missed.

13. What does this CMSIS function do?

```
NVIC_SetPriority (GPIOC_IRQn, 5);
```

    - The function sets the interrupt priority for GPIO Port C to priority level 5.