**Module 9: Interrupt Mechanisms**

1. Which of the following statements is an advantage of interrupt over polling? (There may be more than one correct answer.)

1. It saves the microcontroller’s time.
2. It is a more suitable processing structure for low-power applications.
3. It offers higher level of time precision.
4. It supports concurrent executions of service routines.

2. Which of the following statements is correct?

1. Interrupts are internal or external events that cause changes in program flow control outside a normal code sequence.
2. Exceptions are internal or external events that cause changes in program flow control outside a normal code sequence.
3. Exceptions are only triggered by external events that cause changes in program flow control outside a normal code sequence.
4. Interrupts are only triggered by internal events that cause changes in program flow control outside a normal code sequence.

3. What would happen if the Cortex-M0 processor was interrupted by a HardFault interrupt while it was executing an ISR by an NMI interrupt?

1. The HardFault interrupt would be serviced; this is commonly known as a nested exception.
2. The processor would terminate the NMI ISR and resume execution of the interrupted program thread.
3. The processor would rest itself to resolve the HardFault.
4. The HardFault interrupt would be ignored as it has a lower priority as compared to the NMI ISR.

4. Which of the following statements is correct? The Cortex-M0 architecture supports up to 32 external interrupts

1. each of which has a pre-defined and fixed priority level.
2. which have priority levels that are between -1 and -3.
3. each of which has a priority level that can be modified by the programmer.
4. each of which has a priority level higher than the priority level of a HardFault interrupt.

5. Which of the following statements best describes the behavior of a Cortex-M0 processor if it receives one NMI and two external interrupts at the same time?

1. The processor always executes the NMI ISR first.
2. The processor compares the priority levels of the two external interrupts and runs the higher priority exception handler.
3. The behavior of the processor cannot be predicted in this case, as it depends on the priority levels of the three interrupts.
4. None of the above.

6. Which of the following is not a task performed by the NVIC block?

1. It compares the priority between interrupt requests and the current priority level.
2. It handles nested interrupts automatically.
3. It temporarily stores interrupt service routines to be executed.
4. It generates signals that suspend the program being executed.

7. Why does the Armv6-M architecture use different register addresses to enable and disable interrupts?

1. To speed up the execution time of the exception handlers
2. To reduce the steps needed for enabling/disabling an interrupt
3. To avoid potential race conditions during nested interrupt operations
4. All of the above.

8. A Cortex-M0 processor is running a program with a priority level of 1 at time 0; it receives two interrupt requests in sequence: one at time 1 with a priority level of 2, and the second at time 2 with a priority level of -1. Which of the following statements better describes the actions the processor might take in such a case?

1. The processor will suspend the interrupt (2) until it finishes executing its current routine; it will then run the ISR routine of interrupt (2); following this, it will service the interrupt (-1) if it is still active.
2. The processor will ignore the interrupt (2) as it has very low priority, but it will suspend its current program immediately after it receives interrupt (-1) in order to run its corresponding ISR.
3. The processor will suspend interrupt (2) and continue executing its current routine until it receives interrupt (-1), at which point, it will immediately suspend its current program in order to run the appropriate ISR. After interrupt (-1) is serviced, the processor will go back to its original thread and finish it, and will then service interrupt (2).
4. The processor will execute the ISR of interrupt (2), then the ISR of interrupt (-1), and then go back to finish its main thread.

9. During the stacking process, which of the following registers will be pushed into the stack?

1. R0, R1, R2, R3
2. The link register (R14) and the program counter
3. The program status register(xPSR)
4. All of the above.