Assessment 2.1

When interrupts are disabled on most devices, they remain pending until they can be processed when interrupts are again enabled. No further interrupts are allowed. The functioning of the devices themselves in temporarily halted. The functioning of the devices themselves is temporarily halted. But in real-time systems, the environment that generates the interrupts is often disassociated from the computer system. When interrupts are disabled on the computer system, the environment keeps on generating interrupts anyway. These interrupts can be lost.

a. Discuss the consequences of lost interrupts.

In a computer system, lost interruptions can have serious repercussions, particularly in real-time settings where timing is crucial. Important events or signals from outside sources could go unnoticed or unmanaged by the system while interrupts are recovered. Depending on the type of interruption, this may result in lost deadlines, corrupted data, or even system breakdowns. For instance, in a real-time control system, a missing interrupt could cause erroneous sensor readings, which would cause the system to make bad decisions or conduct erroneous actions. Lost interrupts in communication systems can result in data loss or transmission problems, affecting overall performance and dependability.

b. In a real-time system, is it better to lose occasional interrupts or to halt the system temporarily until interrupts are again enabled?

The decision to handle interrupts in a real-time system should be based on its unique needs and limitations. The system may temporarily turn off interruptions or allow them to be lost altogether before re-enabling them. It is acceptable to occasionally lose interruptions if the consequences are minor and do not compromise the system's overall functionality. However, suppose the interrupts indicate crucial commands or events that require immediate processing to maintain system integrity or safety. In that case, the system should be temporarily halted until interruptions can be successfully processed again. In specific scenarios, it might be more beneficial to sacrifice real-time responsiveness to temporarily uphold system correctness and reliability.