Problem INT-1 (4 parts)  

**Interrupts**

**Part A:** Give a specific example of each of the following exceptions.

A **synchronous** exception: arithmetic overflow, heap overflow, page fault, SWI, divide by 0

An **asynchronous** exception: I/O interrupt, timer interrupt, DMA request

**Part B:** Where does an exception (interrupt) handler obtain storage? What impact does this have on the access functions for this storage?

The exception handler uses the stack. The Push operation must allocate space by adjusting the SP before using the space. Otherwise, an interrupt might overwrite the program's data before the SP is adjusted. A similar constraint exists during stack popping.

**Part C:** Explain the difference between a synchronous and asynchronous exceptions. Give an example of each type.

**Synchronous** interrupts are initiated by the CPU to handle exceptions in program execution or memory accesses (e.g., divide by 0 or page faults). Asynchronous interrupts occur independent of the CPU's clock and are not a function of the current state of the program or memory (e.g., mouse/keyboard interrupts).

**Part D:** Explain the difference between the IRQ and NMI inputs on most processors. Which input is most often used? Why is the other input required?

**Interrupt Request Line (IRQ) inputs** are used by hardware devices to send maskable interrupt signals to the CPU (which the CPU can temporarily ignore).

**Nonmaskable interrupts (NMI)** cannot be ignored by the CPU; the CPU must handle them immediately. IRQS are most common, but NMI signals are needed to handle critical problems such as low battery level and overheating.