**Question: 9.5 Suppose you wish to send a block of data to a tape drive for storage using DMA. What information must be sent to the tape controller before the DMA transfer can take place?**

**Answer:** The information that must be sent to the tape controller prior to the DMA transfer consists of 5 parts:

1. Where the data is stored or to be stored in memory.
2. Where the data is located or to be located on the tape.
3. How much data is to be transferred?
4. What is to be the direction, from tape to memory or from the memory to tape.
5. When exactly the transfer has to be started.

In order to send out these instructions to the tape controller, Programmed Output Instructions are used.

**Question: 9.5 What is polling used for? What are the disadvantages of polling? What is a better way to perform the same job?**

**Answer:** Polling is the process where the computer or controlling device waits for an external device to check for its readiness or state, often with low-level hardware. For example, when a printer is connected via a parallel port, the computer waits until the printer has received the next character. These processes can be as minute as only reading one bit. This is sometimes used synonymously with busy-wait polling. In this situation, when an I/O operation is required, the computer does nothing other than check the status of the I/O device until it is ready, at which point the device is accessed. In other words, the computer waits until the device is ready. Polling also refers to the situation where a device is repeatedly checked for readiness, and if it is not, the computer returns to a different task. Although not as wasteful of CPU cycles as busy waiting, this is generally not as efficient as the alternative to polling, interrupt-driven I/O. The disadvantage is the overhead required to use polling. A device must be polled frequently enough to assure that data held by the I/O device awaiting transfer is not lost. If there are a large number of devices to be polled, much of the CPU's time is wasted doing polling instead of other, more useful processing. The use of interrupts is a better way of managing I/O requests.

**Question 9.12 In general, what purpose does an interrupt serve? Stated another way, suppose there were no interrupts provided in a computer. What capabilities would be lost?**

**Answer:** All in all, an interrupt serves to enable outer access to the CPU to advise the CPU of outside occasions that require consideration or activity by the CPU. Without interrupts, the CPU would be required to play a functioning and constant job in looking for outer occasions, or hazard losing data that may be essential to the framework. Polling would need to be utilized for this reason. The expense of polling is CPU overhead, since each polling activity requires the utilization of an output instruction adhered to by an input instruction. The extra CPU weight can be extreme under certain conditions, for instance when the quantity of surveyed gadgets is enormous, and the polling must be done every now and again. The general impact is a decrease in framework execution.

**Chapter 9 Calculation Exercise** “If my CPU runs at 4.0GHz, and on average takes 10 clock cycles to complete an instruction, how many instructions will be completed in the time it takes to type” MY CPU IS RUNNING NOW"? Assume it takes 5 seconds to type the message. Show your work and how you arrived at the solution”.

**Answer:** 4.0 G (cycles/sec) \*(1 instruction/10 cycles) \* (5 seconds) = 2,000,000,000 Instructions