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?

The location of the data on the I/O device, the starting location of the block of data in memory, the size of the block to be transferred and the direction of transfer, read (I/O → memory) or write (memory → I/O).

9.7 What is polling used for? What are the disadvantages of polling? What is a better way to perform the same job?

Polling is the continuous checking of various input devices in rotation at frequent intervals. It could tie up the CPU in an input loop which is not efficient use of CPU optimally. A wastage of CPU resources. A better way would be to introduce an interrupt which is a perfection situation for an interrupt routine to handle.

9.9 Consider the interface between a computer and a printer. For a typical printout, it is clearly impractical to send output data to the printer one byte or one word at a time (especially over a network!). Instead data to be printed is stored in a buffer at a known location in memory and transferred in blocks to memory in the printer. A controller in the printer then handles the actual printing from the printer's memory.

The printer's memory is not always sufficient to hold the entire print out data at one time. Printer problems, such as an "out of paper" condition, can also cause delays. Devise and describe, in as much detail as you can, an interrupt/DMA scheme that will assure that all documents will be successfully printed.

We can imagine that there is a data buffer in a device controller between the memory and the printer which can store a block of data. This controller cannot be disturbed by CPU. There will be no interrupt generated by the printer. If there is a printer problem, there will be an interrupt. After solving this problem, the printing will start again. After the document is successfully printed, CPU can process others again.

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?

There are many circumstances under which it is important to interrupt the normal flow of a program in the computer to react to special events such as an unexpected user command from the keyboard, a click of a mouse or touch of a finger on a screen, external input from a device requiring attention, an abnormal situation, such as a power failure, that requires immediate attention from the computer, an attempt to execute an illegal instruction, a request for service from a network controller, or the completion of an I/O task initiated by the program.

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.

4GHz = 4000000000Hz

$T = 1/4000000000 = 2.5 \times 10^{-10}s$

$t \text{ (one instruction)} = T \times 10 = 2.5 \times 10^{-9}s$

$\text{Amount of instructions} = 5/t = 2 \times 10^{9}$