COA Assignment-13

Ans) Memory-mapped I/O (MMIO) and port-mapped I/O (PMIO) are two complementary methods of performing input/output between the CPU and peripheral devices in a computer. An alternative approach is using dedicated I/O processors - commonly known as channels on mainframe computers - that execute their own instructions.

Memory-mapped I/O -

Memory-mapped I/O uses the same address bus to address both memory and I/O devices - the memory and registers of the I/O devices are mapped to address values. So when an address is accessed by the CPU, it may refer to a portion of physical RAM, but it can also refer to memory of the I/O device. Thus, the CPU instructions used to access the memory can also be used for accessing devices. Each I/O device monitors the CPU's address bus and responds to any CPU access of an address signed to that device, connecting the data bus to the desired device's hardware register. To accomodate the I/O devices, areas of the addresses used by the CPU must be reserved for Ilo and must not be available for normal physical memory. The reservation might be temporary - the commodore 64 could bank switch between its I/O devices and regular memory or permanent.

Port mapped I/O -

Port-mapped I/O often uses a special class of CPU instructions specially for performing I/O. This is found on Intel microprocessors, with the IN and OUT instructions.

These instructions can read and write one to four bytes (outb, outw, outl) to an I/o device. I/o devices have a separate address space from general memory, either accomplished by an extra "I/o" pin on the CPU's physical interface, or an entire bus dedicated to I/o. Because the address space for I/o is isolated from that for main memory, this is sometimes referred to as isolated I/o.

2 Ans) I/o Device - It is due to the result of the I/o instructions that are written in the computer program. Each data item transfer is initiated by an instruction in the program. Usually transfer is from a CPU register and memory.

Interrupt - By using interrupt facility and special commands to inform the interface to issue an interrupt request signal whenever data is available from any device. In the meantime the CPU can proceed for any other program execution. The interface meanwhile keeps monitoring the device whenever it is determined that the device is ready for data transfer it initiates an interrupt request signal to the computer.

Direct Memory Access — The data transfer btw a fast storage media such as magnetic disk and memory unit is limited by the speed of the CPU. Thus we can allow the peripherals directly communicate with each other using the memory buses, removing the intervention of the CPU. The DMA controller takes over the buses to manage the transfer directly btw the I/o devices and the memory unit.