

1.

Programmed I/O:

In Programmed IO, the processor executes a program that gives it direct control of the I/O operation, including sensing device status, sending a read or write command, and transferring the data. When the processor issues a command to the I/O module, it must wait until the I/O operation is complete. The processor efficiency will be low since the processor works faster than the IO. When the processor encounters an instruction relating to I/O, it executes that instruction by issuing a command to the appropriate I/O module. The I/O module will perform the requested action and then set the appropriate bits in the I/O status register. The I/O module does not interrupt the processor. The processor need to periodically check the status of the I/O module until it finds that the operation is complete, this will reduce processor efficiency.

Interrupt-Driven I/O:

In Interrupt-driven I/O, the processor issues an *I/O command*, continues to execute other instructions, and is interrupted by the I/O module when the latter has completed its work. In this way Interrupt-driven I/O improves system performance compared to Programmed I/O. The working of Interrupt-driven I/O is as follows. The processor issues a READ command and continues other operations. the I/O module receives a READ command from the processor. The I/O module then proceeds to read data in from an associated peripheral. Once the data are in the module's data register, the module signals an interrupt to the processor over a control line. At the end of each instruction cycle, the processor checks for interrupts. When the interrupt from the I/O module occurs, the processor saves the context of the current program and processes the interrupt. The processor reads the word of data from the I/O module and stores it in memory. It then restores the context of the program it was working on and resumes execution. A disadvantage with Interrupt-driven and Programmed I/O is that the processor is responsible for extracting data from main memory for output and storing data in main memory for input.

DMA:

In DMA(Direct Memory Access) the I/O module and main memory exchange data directly, without processor involvement. DMA involves an additional module on the system bus. The DMA module behaves similar to the processor and takes control of the system from the processor. It needs to do this to transfer data to and from memory over the system bus. The DMA module use the bus only when the processor does not need it, or it must force the processor to suspend operation temporarily and this process is called cycle stealing. For a multiple-word I/O transfer, DMA is far more efficient than interrupt-driven or programmed I/O.

2.

There are two ways of mapping the I/O addresses to the processor address space.

Memory Mapped I/O:

In memory- mapped I/O, both memory locations and I/O devices uses the same address space. The processor treats the status and data registers of I/O modules as memory locations and uses the same machine instructions to access both memory and I/O devices. For example, with 10 address lines, a combined total of $2^{10} = 1024$ memory locations and

I/O addresses can be supported, in any combination. A single read line and a single write line are needed on the bus. The advantage of memory-mapped I/O is that the large set of instructions can be used, allowing more efficient programming. Disadvantage is that valuable memory address space is used up to address I/O devices.

Isolated I/O:

In Isolated I/O, the memory location address space is isolated from I/O device address space. The bus may be equipped with memory read and write, plus input and output command lines. The command line specifies whether the address refers to a memory location or an I/O device. The full range of addresses may be available for both memory locations and I/O devices. With 10 address lines, the system may now support both 1024 memory locations and 1024 I/O addresses. Because the address space for I/O is isolated from that for memory. Disadvantage of isolated I/O is there are only a few I/O instructions, as compared to memory-mapped I/O method.

3.

We are given that the DMA module is transferring characters to memory at the rate of 19200 bps.

Assuming that 1 char is 8 bits, we have $19200 \text{ bps} = 19200 / 8 = 2400 \text{ char/sec}$

Processor is fetching instructions at the rate of 4.5 MIPS = 4.5×10^6 instructions per sec

Also assuming that bus transfers are byte wide we can say that the fetch rate of the Processor after being affected by DMA module will be $4500000 - 2400 = 4497600$ instructions per second.

Percentage of processor slowdown will be $= \frac{2400}{4500000} * 100 = \mathbf{0.0533\%}$

4.

4.1)

We are given that processor can execute max at a rate of 4 MIPS = 4×10^6 instructions per sec.

I/O instructions are performed 10% of the time. Hence we will have

Maximum I/O instruction execution rate is $= 4 \times 10^6 \times 10\% = 400000$ instructions per second.

Given that one word transfer will require the processor to execute 4 instructions. So,

Maximum I/O data transfer rate is $= \frac{\text{Maximum I/O instruction execution rate}}{4}$

$$= 400000/4$$

$$= \mathbf{100000 \text{ words/sec}}$$

4.2)

Given that background processes runs 90% of the time and 3 cycles out of the 5 cycles are Memory cycles, so during background process DMA can use only 2 cycles and during I/O processes for the rest 10% DMA can use all the 5 cycles, so we will have

$$\begin{aligned}
\text{No of machine cycles per second} &= \text{MIPS} * \text{CPI} \\
&= (4 * 10^6)(0.9 * 2 + 0.1 * 5) \\
&= (4 * 10^6)(0.9 * 2 + 0.1 * 5) \\
&= 9200000 \text{ cycles/ second}
\end{aligned}$$

We are given that read/write requires 1 machine cycle hence

Data transfer rate = **9200000 words/second**

5.

There are 4 different ways by which processor can determine which device gave the interrupt.

Multiple interrupt lines:

Here multiple interrupt lines are provided between the processor and the I/O modules. It is impractical to dedicate more than a few bus lines or processor pins to interrupt lines so, even if multiple lines are used, it is likely that each line will have multiple I/O modules attached to it.

Software poll:

When an interrupt is detected by the processor, it will branch to an interrupt- service routine which polls each I/O module to determine which module caused the interrupt. The poll will be in the form of a separate command line. In this case, the processor will raise the TESTI/O and place the address of a particular I/O module on the address lines. The I/O module will respond positively if it set the interrupt. Each I/O module will have an addressable status register. The processor then reads the status register of each I/O module to identify the interrupting module. On identification, the processor branches to a device- service routine specific to that device. The disadvantage of the software poll is that it is time consuming.

Daisy chain:

All I/O modules share a common interrupt request line. The interrupt acknowledge line is daisy chained through the modules. When the processor senses an interrupt, it sends out an interrupt acknowledge. This signal propagates through a series of I/O modules until it gets to a requesting module. The requesting module typically responds by placing a word on the data lines. This word is referred to as a *vector* and is either the address of the I/O module or some other unique identifier. The processor uses the vector as a pointer to the appropriate device- service routine.

Bus arbitration:

The I/O module should gain control of the bus before it can raise the interrupt request line. So, only one module can raise the line at a time. once the processor detects the interrupt the processor responds on the interrupt acknowledge line. The requesting module will then place its vector on the data lines.

DMA module must use the bus only when the processor does not need it, or it must force the processor to suspend operation temporarily, this is called cycle stealing. When the bus is in the control of DMA, the processor can continue with other works which doesn't require the bus. It delegates the I/O operation to the DMA module. The DMA module transfers the entire block of data, one word at a time, directly to or from memory, without going through the processor. When the transfer is complete, the DMA module sends an interrupt signal to the processor. Thus, the processor is involved only at the beginning and end of the transfer. The processor may also be suspended. In that case, the processor is suspended just before it needs to use the bus. The DMA module then transfers one word and returns control to the processor, this is not an interrupt; the processor does not save a context and do something else. Rather, the processor pauses for one bus cycle.

6.

Firewire:

Firewire was developed as an alternative for the small computer system interface (SCSI) which to be used on smaller systems like personal computers and servers. The objective is meet the increasing demands for high I/O rates on the systems, while avoiding the bulky and expensive I/O channel technologies which are developed for mainframe and supercomputer systems. FireWire use a daisy- chain configuration. FireWire provides hot plugging, which makes it possible to connect and disconnect peripherals without having to power the computer system down or reconfigure the system. FireWire provides automatic configuration. In FireWire, there are no terminations, and the system automatically performs a configuration function to assign addresses. An important feature of the FireWire standard is that it specifies a set of three layers of protocols to standardize the way in which the host system interacts with the peripheral devices over the serial bus.

SCSI:

Small Computer System Interface is for connecting peripheral devices to small and medium sized computers. High-speed versions of SCSI is still popular for mass memory support on enterprise systems. The physical organization of SCSI is a shared bus. The bus will provide for parallel transmission rather than serial. Speeds range from 5 Mbps to 160 Mbps.

Thunderbolt:

Thunderbolt is one of the fastest, peripheral connection technology to become available for general-Purpose use. One Thunderbolt cable can manage the work which previously required of multiple cables. The technology combines the data, video, audio, and power into a single high-speed connection for peripherals such as hard drives, RAID arrays, video- capture boxes, and network interfaces. It provides up to 10 Gbps throughput in each direction and up to 10 watts of power to connected peripherals.

InfiniBand:

InfiniBand is an I/O specification which aims at the high-end server market. It describes an architecture and specifications for data flow among processors and intelligent I/O devices. InfiniBand will enable the servers, remote storage, and other network devices to be attached in a central fabric of switches and links.

PCI Express:

It is known as Peripheral Component Interconnect Express. The PCI is a popular high-bandwidth, processor-independent bus that can function as a mezzanine or peripheral bus.

Compared with other common bus specifications, PCI delivers better system performance for high-speed I/O subsystems. PCI has been widely adopted and is finding increasing use in personal computer, workstation, and server systems. Because the specification is in the public domain and is supported by a broad cross-section of the microprocessor and peripheral industry, PCI products built by different vendors are compatible.

SATA:

SATA is an interface for disk storage systems. It will provide the data rates up to 6 Gbps, with a maximum of 300 Mbps per device. SATA is widely used in desktop computers and embedded applications.

Ethernet:

Ethernet is a wired networking technology which are used in homes, offices, data centers, enterprises, and wide-area networks. Ethernet has moved from bus-based to switch based, and the data rate has periodically increased by an order of magnitude. With switch-based systems, there is a central switch, with all of the devices connected directly to the switch. Ethernet support the data rates up to 100 Gbps.

WiFi:

Wi-Fi is a wireless Internet access technology which are used in homes, offices, and public spaces. Wi-Fi in the enterprise has become an essential means of enhancing worker productivity and network effectiveness. And public Wi-Fi hotspots have expanded dramatically to provide free Internet access in most public places. As the technology of antennas, wireless transmission techniques, and wireless protocol design has evolved, the IEEE 802.11 committee has been able to introduce standards for new versions of Wi-Fi at ever-higher speeds.

