INTERRUPT DRIVEN I/O, DMA AND DISKS

Interrupt-driven I/O:

The problem with the programmed I/O is that the processor has to wait a long time for the input/output module of concern to be ready for either reception or transmission of more data. The processor, while waiting, must repeatedly interrogate the status of the Input/ Output module. As a result the level of performance of entire system is degraded.
An alternative approach for this is interrupt driven Input / Output. The processor issue an Input/Output command to a module and then go on to do some other useful work. The input/ Output module will then interrupt the processor to request service, when it is ready to exchange data with the processor. The processor then executes the data transfer as before and then resumes its former processing. Interrupt-driven input/output still consumes a lot of time because every data has to pass with processor.

DMA:

The previous ways of I/O suffer from two inherent drawbacks.

1. The I/O transfer rate is limited by the speed with which the processor can test and service a device.
2. The processor is tied up in managing an I/O transfer;a number of instructions must be executed for each I/O transfer.

When large volumes of data are to be moved, a more efficient technique is required:Direct memory access. The DMA function can be performed by a separate module on the system bus, or it can be incorporated into an I/O module. In either case , the technique works as follow.
When the processor wishes to read or write a block of data, it issues a command to the DMA module by sending the following information.

- Whether a read or write is requested.
- The address of the I/O devices.
- Starting location in memory to read from or write to.
- The number of words to be read or written.

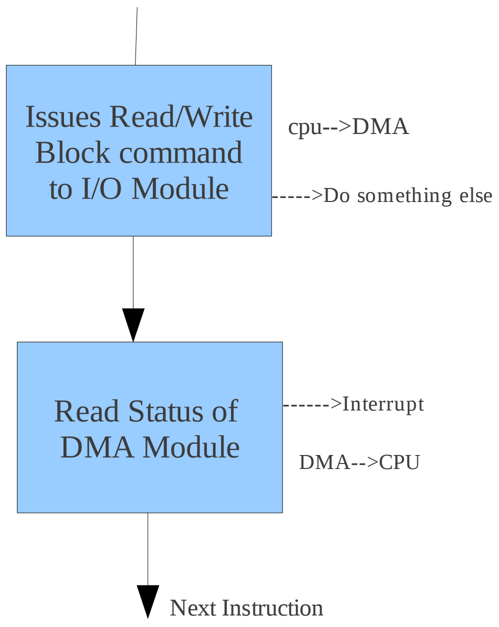


Fig:DMA

The processor then continues with other work. It has delegated this I/O operation to the DMA module, and that module will take care of it. The DMA module transfers the entire block of data, one word at at 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 at the end of the transfer.
In programmed I/O cpu takes care of whether the device is ready or not. Data may be lost. Whereas in Interrupt-driven I/O, device itself inform the cpu by generating an interrupt signal. if the data rate of the i/o is too fast. Data may be lost. In this case cpu most be cut off, since cpu is too slow for the particular device. the initial state is too fast.
it is meaningful to allow the device to put the data directly to the memory. This is called DMA.
dma controller will take over the task of cpu. Cpu is general purpose but the dma controller is specific purpose.

A DMA module controls the exchange of data between main memory and an I/O module. The processor sends a request for
the transfer of a block of data to the DMA module and is interrupted only after the entire block has been transferred.

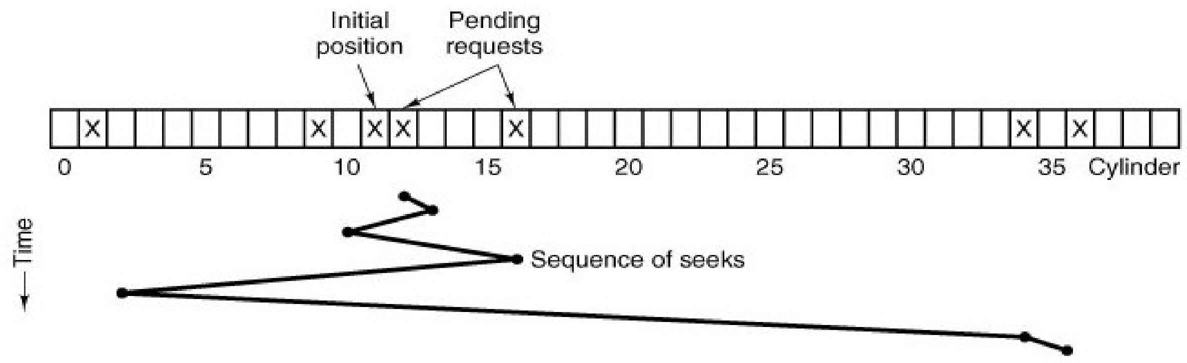
Disks:

All real disks are organized into cylinders, each one containing as many tracks as there are heads stacked vertically. The tracks are divided into sectors, with the number of sectors around the circumference typically being 8 to 32 on floppy disks, and up to several hundred on some hard disks. The simplest designs have the same number of sectors on each track.
Disk Arm Scheduling Algorithms:

1. First come First server FCFS
2. Shortest Seek First Shortest Seek First (SSF) disk scheduling algorithm.
3. The elevator algorithm for scheduling disk requests.

Consider a disk with 40 cylinders. A request comes in to read a block on cylinder 11. While the seek to cylinder 11 is in progress, new requests come in for cylinders 1, 36, 16, 34, 9, and 12, in that order.

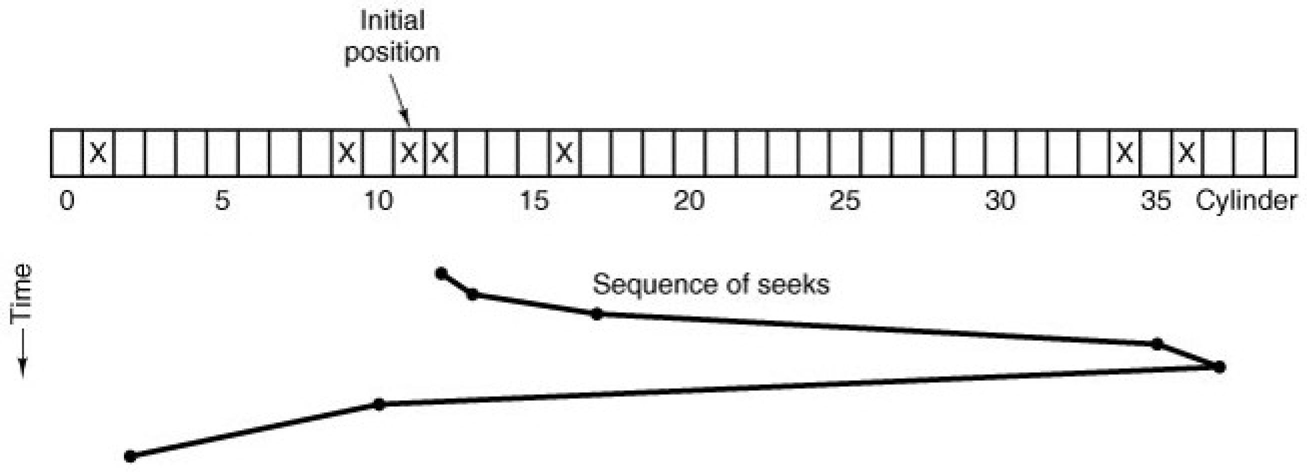
Shortest Seek First (SSF) disk scheduling algorithm.
Pick the request closet to the head.



Total no of Cylinder movement: 61

The elevator algorithm for scheduling disk requests:
Keep moving in same direction until there are no more outstanding requests in that direction, then they switch direction. The elevator algorithm requires software to maintain 1 bit, the current direction bit. UP or DOWN. If it is UP the arm is moved to the next highest pending request and if it is DOWN if it moved to the next lowest pending request if any.

the elevator algorithm using the same seven requests as shown above, assuming the direction bit was initially UP. The order in which the cylinders are serviced is 12, 16, 34, 36, 9, and 1, which yields arm motions of 1, 4, 18, 2, 27, and 8, for a total of 60 cylinders.



Source : http://dayaramb.files.wordpress.com/2012/02/operating-system-pu.pdf