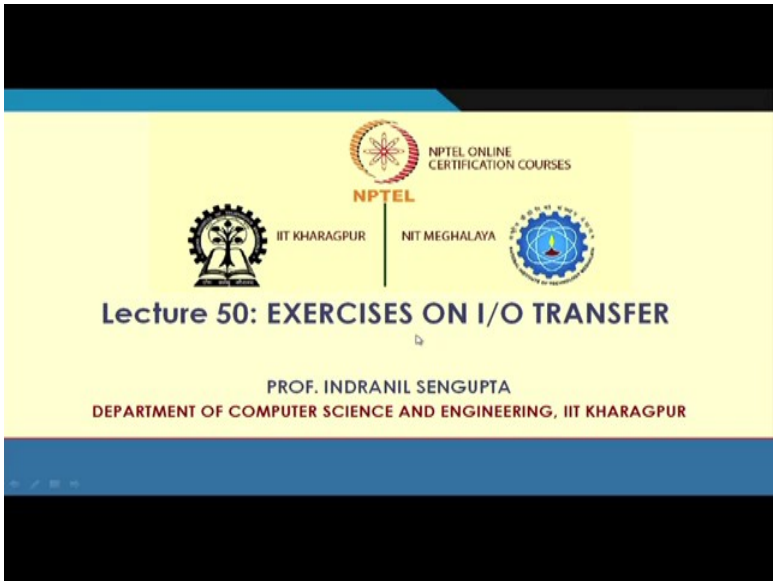
Computer Architecture and Organization
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture No – 50
Exercises On IO Transfer

In this lecture we shall not be learning anything new, I shall not be teaching you anything new, rather I shall be taking some exercises on input output transfer and try to work them out with you. I think if you work out some exercises, you will have a better understanding regarding some of the concepts which we have already discussed during the last few lectures.

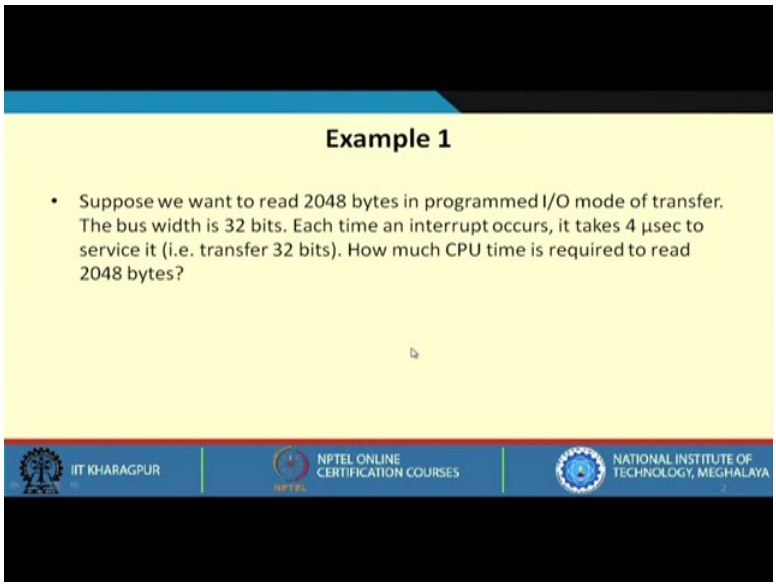
(Refer Slide Time: 00:45)



So, this lecture is titled exercises on IO transfer.

Let us look at the exercises one by one and let us try to solve them.

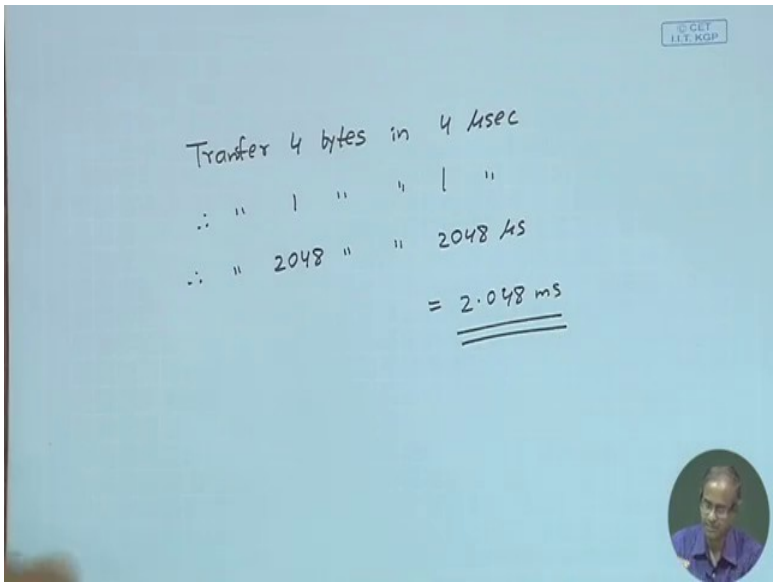
(Refer Slide Time: 00:54)



Just look at the first example. This says that we want to read 2048 bytes in programmed IO mode of transfer, the bus width is 32 bits. This means that we can transfer 4 bytes at a time, 32 bits each time. When an interrupt occurs, it takes 4 microsecond to service it; service means one data transfer of 32 bits.

So, to transfer 4 bytes, I need 4 microsecond. The question is how much CPU time is required to read all 2048 bytes.

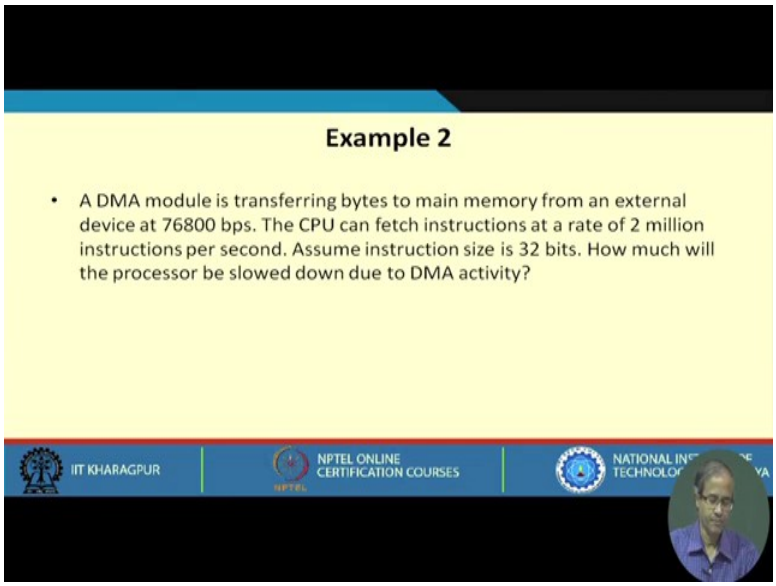
(Refer Slide Time: 02:03)



So, we are transferring 4 bytes in 4 microseconds. Therefore, we will transfer 1 byte in 1 microsecond. Now, our problem is to transfer 2048 bytes. So, it will be 2048 microsecond or 2 milliseconds.

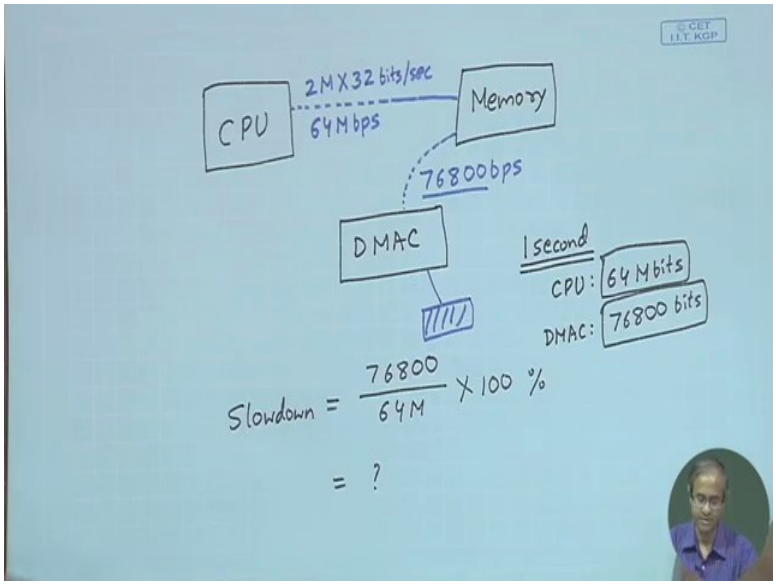
Let us move on to next problem.

(Refer Slide Time: 02:52)



Here there is a DMA module. The DMA module is transferring bytes to main memory from an external device, where the data transfer rate is given. Let us show pictorially whatever we are reading out.

(Refer Slide Time: 03:24)



In this schematic, there are three part is involved i.e CPU, memory and DMA controller.

Now, the problem statements says that the device is trying to transfer bytes to main memory. And the speed is specified 76800 bits per second.

Now, the problem says again that the CPU can fetch instructions at a rate of 2 million instructions per second. You assume that instruction size is 32 bits. So, what does this mean? CPU can fetch instructions at the rate of 2 million instructions per second, where each instruction is 32 bits. So, here if you look at CPU memory connection, now here we are saying that 2 million instructions, which means 2 million x 32 bits per instruction, so many bits per second.

So, these many bits per second is what IO is trying to transfer. CPU memory connection actually can go up to 2 x 32 = 64 Mbps. Now, the question is how much will the processor be slowed down due to DMA activity.

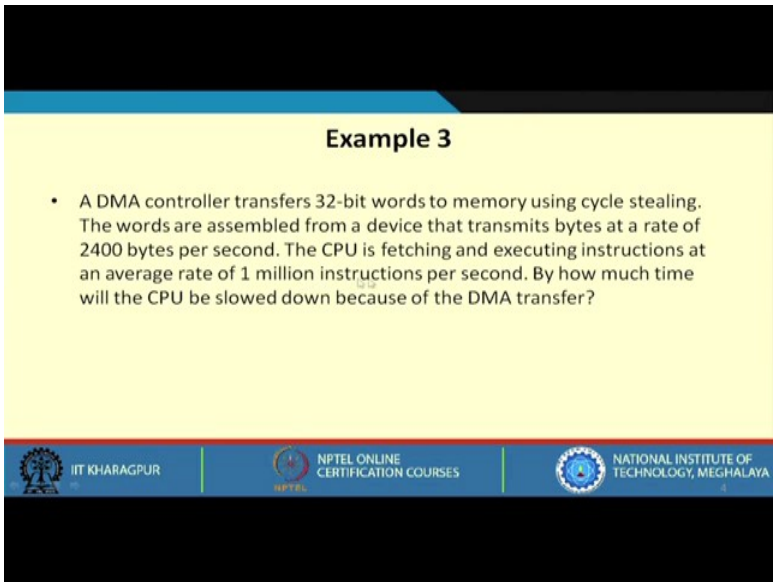
Let us look at a one second window. Within a one second window, the CPU normally was transferring 64 megabits of data and the DMA controller is transferring data at is 76800 bits per second.

Out of this total time, these many bits will be taken up by DMA controller. You see this CPU could have transferred so many bits, but out of that, these many bits are stolen by DMA controller. The slow down you can calculate like this. So, how much CPU time getting reduced divide by the total 64m. So, this will be the fraction. If you multiply it by 100, you will be getting a percentage.

The calculation is shown.

Let us move on to the third problem.

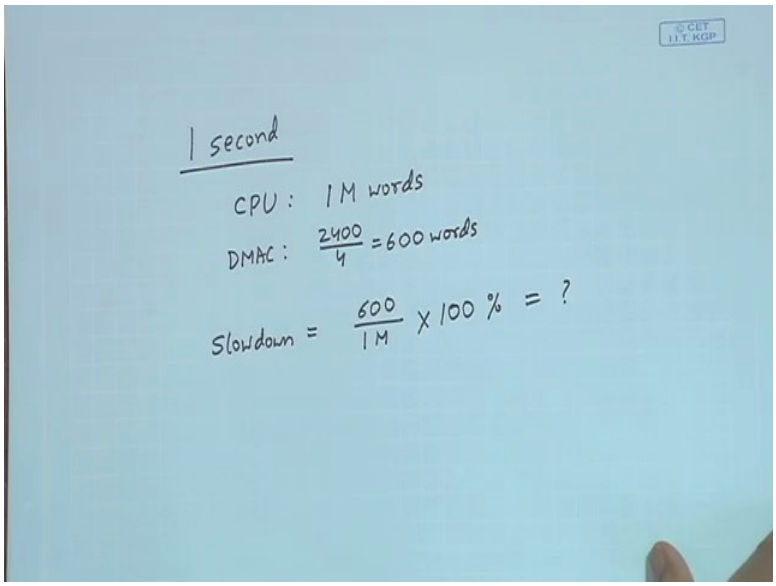
(Refer Slide Time: 08:10)



The third problem is a similar one using DMA controller. This says DMA controller transfers 32 bit words to memory using cycle stealing and the device from where the data transferred, they transfer bytes at a rate of 2400 bytes per second. So, this is exactly similar to the previous problem.

So, the DMA controller is transferring 32 bit words. The words are coming at the speed 2400 bytes per second and the CPU is fetching and executing instructions, one million instruction, let say one instruction is one word. So, one million words, but the only thing we will have to do is that here the number of bytes is 2400. It means it will be 600 words.

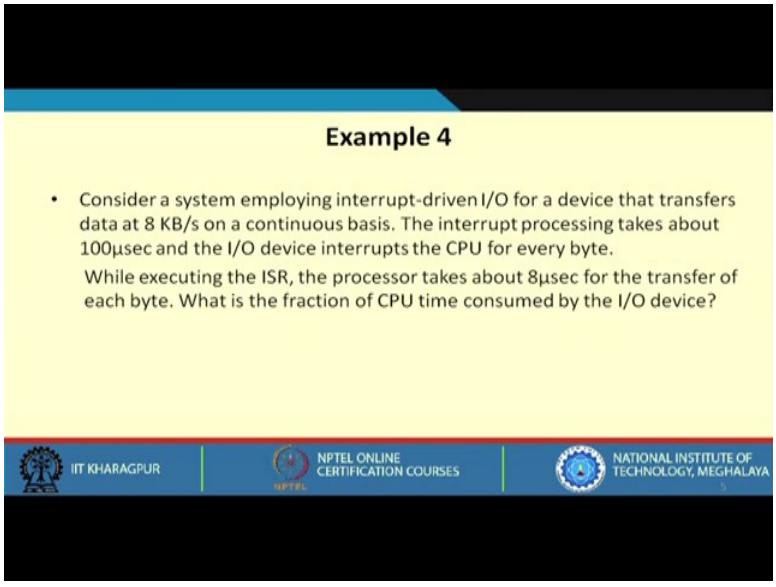
(Refer Slide Time : 09:21)



The CPU is transmitting one million instruction per second, an instruction is one word. So, CPU will transmit 1 Mwords. DMAC is transferring 2400 bytes, 4 bytes at a time. It will be 2400 divide by 4 = 600 words.

So, slowdown will be given by 600 divided by 1M. If you multiply by 100, you get a percentage. So, this you can calculate how much it is. So, it will be a small fraction.

(Refer Slide Time : 10:44)



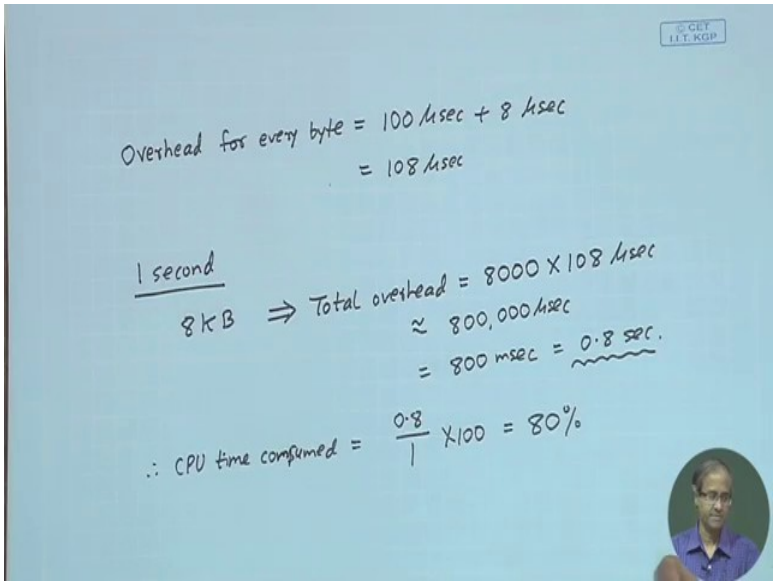
Here, we have a problem of interrupts. There is a system that employs interrupt driven IO for a device that transfers data at 8 kilobytes per second on a continuous basis. The

device is continually transferring data and the interrupt processing takes 100 microseconds. For every interrupt, the CPU will be spending 100 microseconds as overhead for saving status, restoring status, jumping to ISR, everything taken together.

Another thing mentioned here is that IO device will interrupt the CPU for every byte. So, there are two things. There is a device which is transferring data at 8 kilobytes per second and for every byte the device will interrupt the CPU; and for the interrupt, the overhead is 100 microseconds per interrupt processing, and 8 microsecond for actual transfer of each byte. So, it will be 108,.

So, you can easily calculate here the overhead.

(Refer Slide Time : 12:15)



For every byte transfer, the overhead will be 100 + 8 = 108 microsecond. Now, the device is transmitting data at the rate of 8 kilobytes per second on a continuous basis. So, the question is what is the fraction of CPU time consumed by IO device?

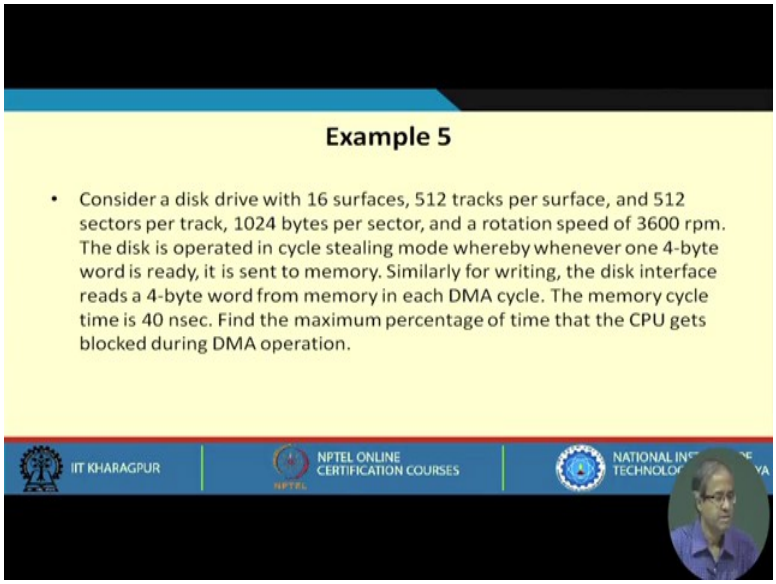
Let us take the one second window again. In one second, the total data that is being sent is 8 kilobytes. Total overhead will be how much? This 8 k means let say let call it 8000 for simplicity of calculation. Actually it will be 8192 i.e. 8000 into 100 and 8, so many microseconds. So, approximately assuming this to be 100, it will be 8 lakh microsecond which means 800 microseconds which means 0.8 seconds,.

So, you see out of 1 second, 0.8 second is spent as overhead. So, you are left with only 0.2 second for useful work. CPU time consumed in this example due to IO overhead consumed will be 0.8 divide by the total 1 second. If you multiply by 100, you will be getting the percentage which is 80%.

So, you see about 80% of CPU time is getting wasted here. This is a characteristic of interrupt driven I/O, where if the data rate is very fast you have to do interrupt processing very frequently, the overwrite can be very high.

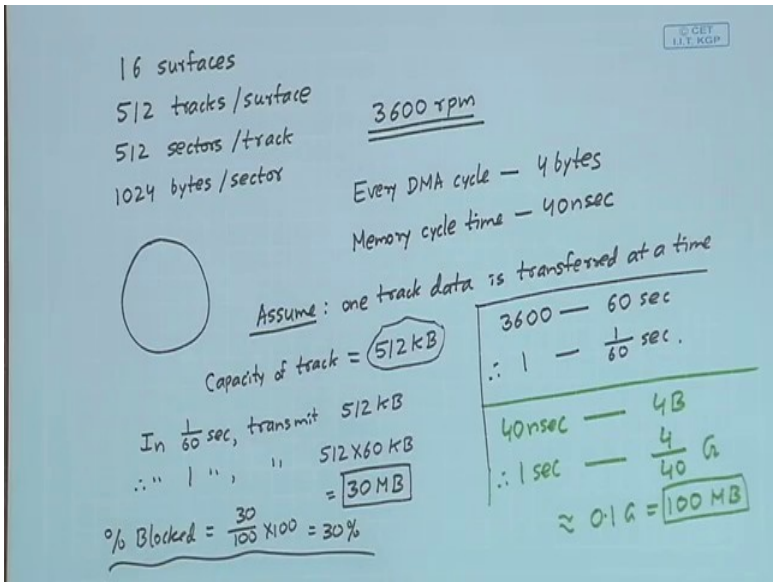
Let us move on to the next problem.

(Refer Slide Time : 14:50)



This is a problem that involves a disk and DMA. We have a disk drive with 16 surfaces. That means 8 platters, and there are 512 tracks per surface.

(Refer Slide Time : 15:23)



There is 512 sectors per track and 1024 bytes per sector. These are some data which are specified and of course, it is specified that the rotation speed is 3600 rpm.

The disk is operated in cycle stealing mode, whereby whenever 4 byte word is ready, it is sent to memory. Similarly, for writing the disk interface reads 4 byte word from memory in each DMA cycle. So, what it actually says is that in every DMA cycle, we transfer 4 bytes of data. Another thing is mentioned that the memory cycle time is 40 nanoseconds. Memory cycle time is actually the memory access time.

The question is what is the maximum percentage of time the CPU gets blocked during DMA operation? Well, we do not need all these data. Just you assume that data from one track can be transferred at a time. Let us assume although they are 16 surfaces, we assume that one track data is transferred at a time.

Let us see what is the capacity of a track. In a track, there are 512 sectors and each sector is 1 kilobyte. So, it will be 512 kilobytes. This is the capacity of a track and let us see what is the rotational speed. It is 3600 rpm. So, how much time it takes for one rotation? So, to make a calculation like this 3600 rotations in 1 minute, let say we do it in seconds, in 60 seconds. Therefore, one rotation in 1 by 60 seconds.

So, when the disk is rotating once, this much data will be transmitted. So, what will be the data rate? It is 512 kilobyte in 1/60 second. So, you can say in 1/60 second, you

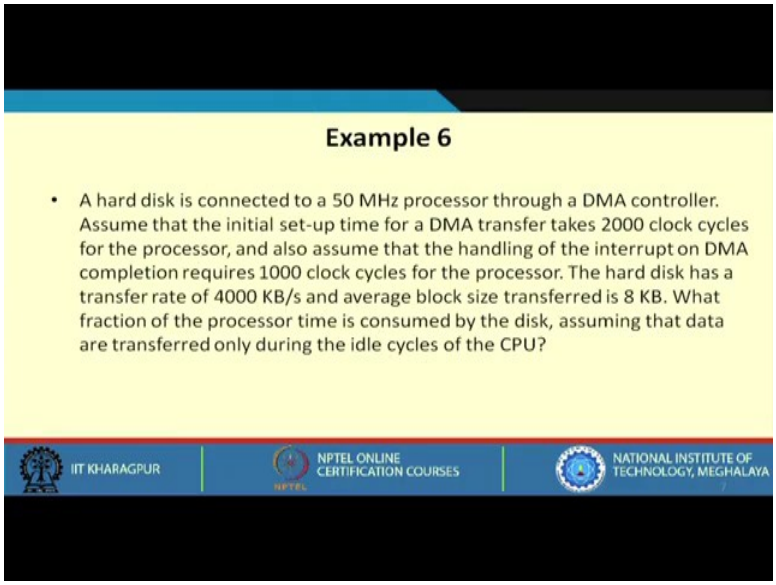
transmit 512 kilobytes of data. Therefore, in one second you will be transmitting 512 x 60 KB. So, in one second, you are transmitting 30 megabytes.

Let us now look at the memory, memory cycle time is 40 nanosecond. What does this mean? In 40 nanosecond, we can read or write one word. So, let us say 4 bytes word, because each DMA cycle, it can read/write 4 bytes. We say that in that 40 nanosecond, you can transfer 4 bytes.

So, let us see in one second how much data can memory support. It is 4 / 40 nano = 0.1G = 100MB. The memory can support 100 megabytes of data transfer and your disk is sending data at 30 megabytes per second. So, you can now just calculate what percentage it will get blocked out. It comes to 30%.

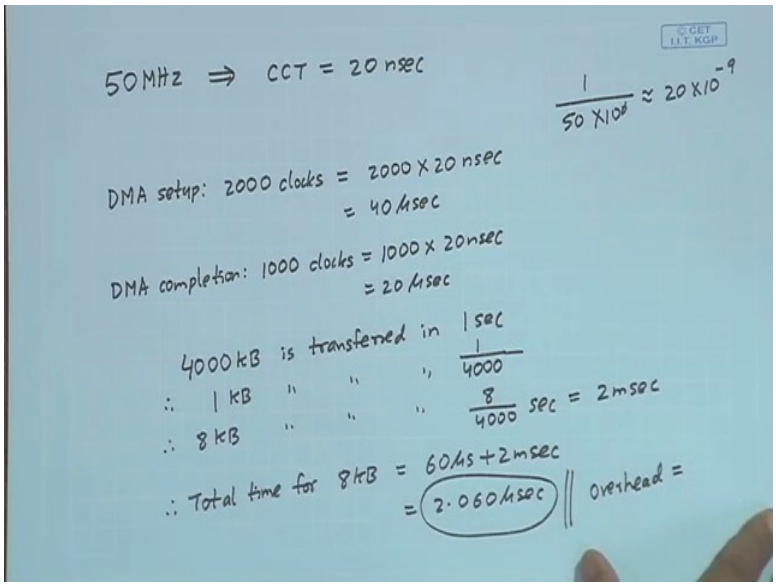
Let us move on to the next problem.

(Refer Slide Time : 21:50)



This problem says this hard disk again connected to 50 MHz processor.

(Refer Slide Time : 22:07)



50 MHz means CCT will be 20 nanoseconds.

The initial setup time for DMA transfer takes 2000 clock cycles, and handling of interrupts on DMA completion requires 1000 clock cycles. What does this mean? We have already calculated clock cycle time. It is 2000 x 20 nanoseconds which means 40 microseconds, and DMA completion requires 1000 clocks. So, again it will be 1000 x 20 nanoseconds = 20 microseconds.

The hard disk has a transfer rate of 4000 KBps and the average block size transferred is 8 KB. What fraction of the processor time is consumed by the disk? For DMA setup and DMA completion, you have the overheads. You can see for initialization, you need 40 microseconds. At the end, you need 20 microseconds, but after that you are transferring data whose block size is 8 kilobytes, where the transfer rate is 4000 kilobytes per second. So, you can calculate like this 4000 kilobytes is transferred in one second.

Now, you calculate one block of 8 KB which is mentioned; so how much time will take for that block to get transferred. Average block size is 8 KB. So, therefore, 1 KB is transferred in 1 / 4000 K.

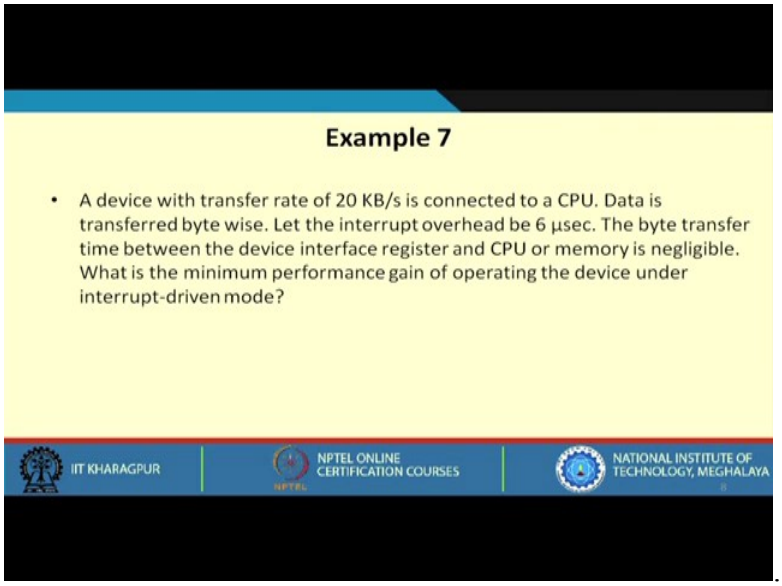
The total time for transferring 8 KB comes to 2.060 microseconds.

The question is what fraction of the processor time is consumed by the disk? Assume that the data are transferred only during the idle cycles of CPU. So, you can actually find out

total 8 KB is being transmitted this much amount of time. I will leave it an exercise for you. you can calculate the overhead like that.

You see in this way we have calculated that how much time will be taking for the block transfer of 8 KB. Now, that block transfer is doned, it is assume that it is done in cycle stealing mode and CPU is not disturbed, and this is the total time that is taken for that.

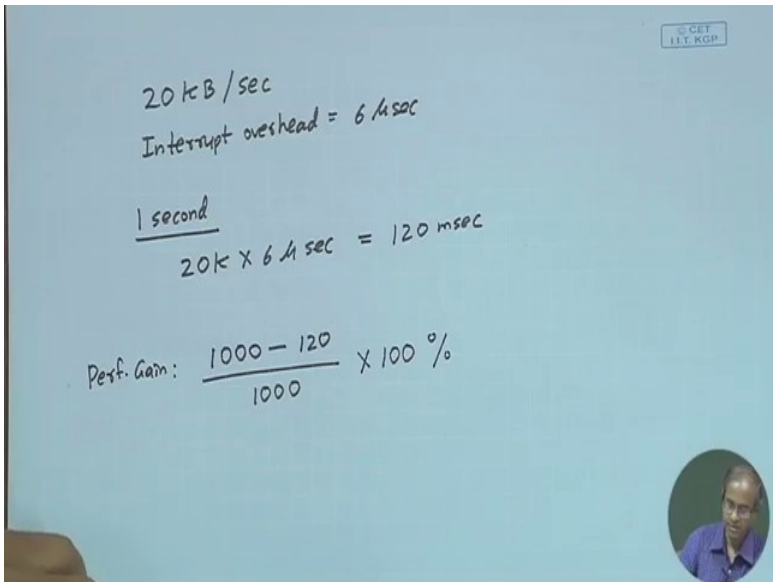
(Refer Slide Time : 27:55)



This says a device with transfer rate of 20 kilobytes per second is connected to CPU. Data is transferred bytewise, interrupt overhead is assumed to be 6 microsecond. The byte transfer time between device interface register and CPU or memory is negligible. What is the minimum performance gain of operating the device under interrupt driven mode.

You see data is transferred byte wise. You have a transfer rate of 20 kilobytes per second.

(Refer Slide Time : 28:34)



It is said that your interrupt overhead is 6 microsecond, and you are transferring byte wise. So, for every byte there will be an interrupt. You look at a 1 second window again. So, in 1 second window, total 20 kilobytes will be transmitted and for every byte there will be an overhead of 6 microsecond. So, the total overhead will be 20 k x 6 microsecond. So, this will become 120 milliseconds.

So, the question is what is the minimum performance gain? Initially the time was 1 second, total time was 1000. Out of that 120 is being wasted as overhead, divided by 1000. But if you do not use interrupt driven, your entire time would have been wasted. So, this will be your performance gain. If you multiply it by 100, you will get the resulting percentage.

In this lecture we worked out some problems on IO interfacing. Most of the problems are of the standard of GATE examination that you may be aware of. In the next lectures, we shall be starting our discussions on some common IO bus standards like USB for example. What are the silent features there, how they are used, what are the peak speeds, advantages and so on and so forth.

Thank you.