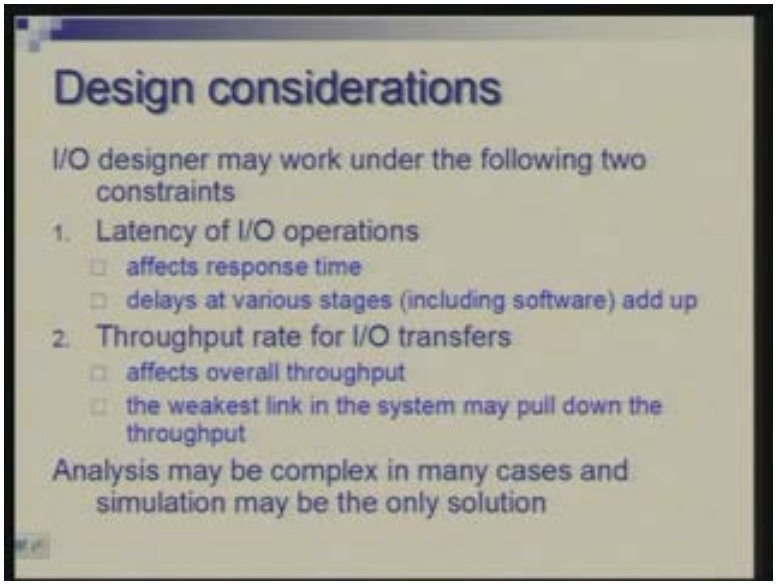
Computer Architecture
Prof. Anshul Kumar
Department of Computer Science and Engineering
Indian Institute of Technology, Delhi
Lecture - 37
Input/Output Subsystem: Designing I/O Systems

In the previous few lectures we have talked about individual aspects of I/O subsystem, talked about peripherals and the buses which interconnect them. We also have given some thought to how software looks at the whole operation. Today we will look at the overall system design issues and primarily the discussion will be in the form of examples where we will see that, given certain building blocks and certain requirements how we put them together to achieve certain performance objectives.

(Refer Slide Time: 1:34)



When you are designing the system you have to keep performance in mind. And as we discussed earlier the performance could focus on the latency aspect or the throughput aspects or a combination of these. The latency is important in terms of achieving certain response time. So when some event occurs outside the system as to respond and the response of I/O subsystem is very important in that case. So you have data which moves from external world into the system which gets processed and then maybe some response comes back. So, as the data moves through various stages all the delays get added up. The delay may be in terms of the rate at which the data gets generated at the peripheral, the way it crosses through controller, buses and adapters and finally it goes through processing and follows somewhat similar path.

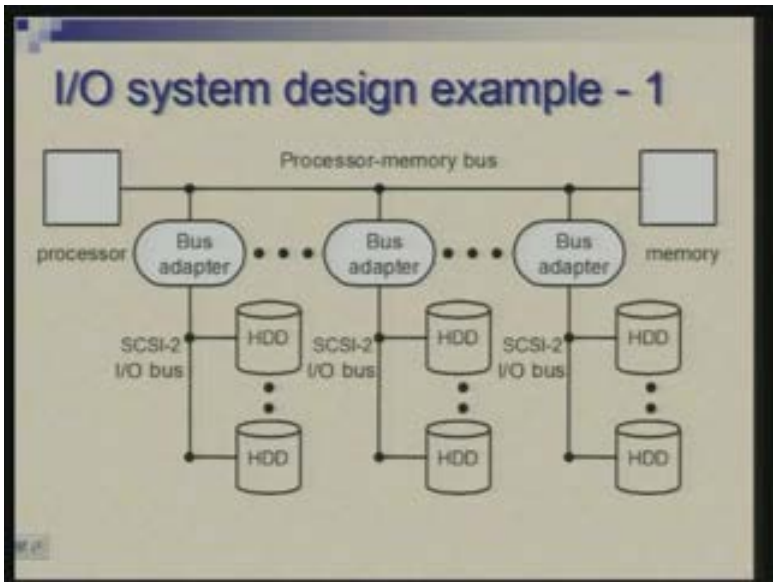
On the other hand, there is throughput issues where essentially the whole thing may actually like a pipeline and the slowest link in the whole system would basically ((dictate)) the throughput. So, as far as throughput is concerned it is comparatively easier

at least if you have to find the average. So you can image a steady state with data flowing and if you if you understand the capability at each stage about at what rate the data can flow then you can get a reasonable estimate of the throughput. But latency could often be a more complicated issue.

For example, the delay block of data and counters going through for example a bus would also depend upon how much load is being carried on the bus. So you need to look at complex interaction between various pieces of information which flow through various components of the system and at times analysis may not be very easy and the last resort may be that you try to simulate the entire behavior and then see in a very accurate manner what kind of latencies you can obtain.

So, in the example we will consider today, we will focus more on the throughput aspects. We will imagine steady state flow of data, like fluid flowing through various pipes we will try to see what is the throughput, what is the rate at which data can flow and that will give as idea of throughput.

(Refer Slide Time: 4:29)



So, first example we will consider is one where there is an application where huge amount of data has to be read out read from various disks or it could be the reverse that there is lots of data which have to be stored in the disk. So functionally it is a very simple application that you have a processor memory and lots of disks so we want to attain certain..... you want to maximize our throughput that is, as much data we can pull out of the disks for a given capacity of processor, memory, bus and other components in the system so we will try to achieve that.

So the architecture you see here has a processor memory bus on which certain number of adapters bus adapters are hooked, each provides an I/O bus so in this case we consider SCSI-2 I/O bus which is supporting multiple disks. So each can have one or more disks

and the disks read the data, send the blocks which traverse through this bus, goes through the adapter and ultimately go to the memory. The processor for each block of data which gets read there is some certain amount of processing involved from both points of view: to organize the I/O as well as to act upon the data so let us look at some numerical parameters associated with various components.

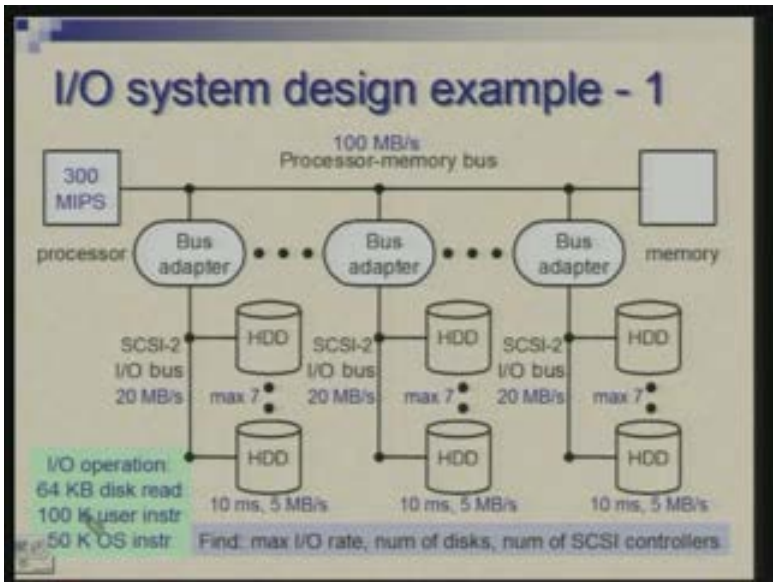
The processor has an instruction execution rate of 300 million instructions per second, so, that quantifies the capacity of processor to perform certain tasks. The bus which connects processor and memory is capable of handling a throughput of 100 megabytes per second. So data which comes from various sources is considered on this bus and when it is deposited in the memory, one has to keep in mind that there is a certain throughput limit here and how we arrive at bus throughput limit we have discussed. It depends upon the protocol, the size of block which is being transferred and so on. So suppose that analysis was carried out separately and one comes up with a figure of 100 megabytes performance second then these SCSI - 2 buses each has a capacity of 20 megabytes per second so these buses are carrying the data from the disk towards the memory.

Each SCSI controller is capable of handling up to seven disk drives and each disk drive requires 10 milliseconds to access that is seek time plus rotation latency plus controller delay and then rate at which data gets read out data gets transferred is 5 megabytes performance second.

Now what let us also define what each I/O operation is?
Each I I/O operation involves 64 kilobytes of data to be read from the disk. So let us say it has to be read from some contiguous area on the disk from a single track but every time you read a block of data it may be a different track. so every time we have to incur 10 milliseconds of delay before we start reading the next block and once we have incurred the 10 milliseconds delay then 64 kilobytes will come continuously at the rate of 5 megabytes per second.

Now as far as the SCSI bus is concerned, when one disk is busy in seeking or it is rotating to get to the position, other disk could be transferring the data, so seeking and transfer among different disks could overlap. Now each I/O operation of 64 kilobytes transfer requires some work on the part of the processor. So 100000 instructions are executed by the user program and 50000 instructions are executed by operating system. So imagine that each transfer takes place using DMA process which means that the processor will initially setup the process, it will instruct the DMA controller, we are not showing..... so let us say that all the SCSI control and DMA control is lumped within the bus adapters. So it has to initially instruct the DMA controller to read from a particular track and particular sector from a particular disk and it also supplies the address in the memory where things have to be deposited so that is the initiation path and when the transfer is over it needs to be informed with the help of an interrupt so again some instruction get executed. So on the whole OS goes through 50000 instructions all put together beginning and end and the data is processed by the CPU, it requires 100k instructions of the user program.

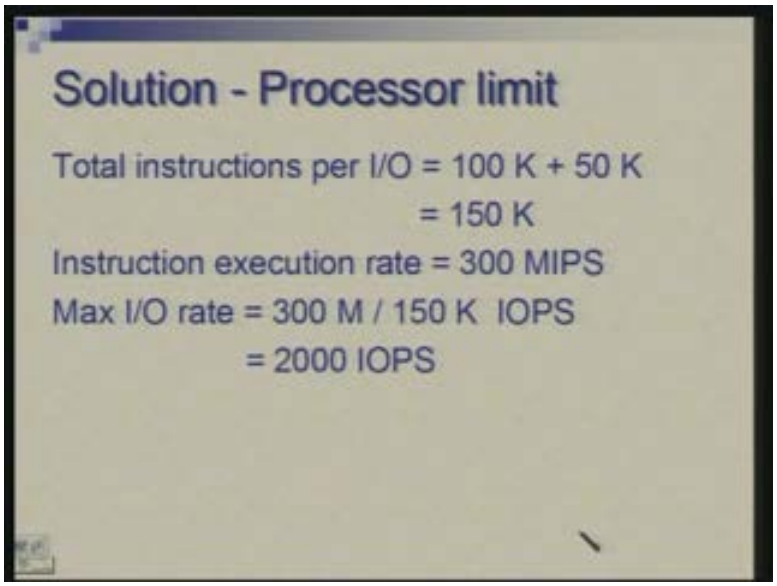
(Refer Slide Time: 11:15)



So now what is it we need to do? What is the problem?
Problem is to find out what is the maximum input output rate that this system can sustain. So the fixed components here are there is one single processor, one single memory and a single processor memory bus. The flexibility is in terms of putting as many disks as you can gainfully put in the system and depending upon the number of disks you have you will require certain number of SCSI buses and corresponding controllers. So the problem involves first finding out what is the maximum I/O rate. That means how many I/O operations as defined here can be carried out, at what rate can you do this and to achieve that how many disks and SCSI controllers are required.

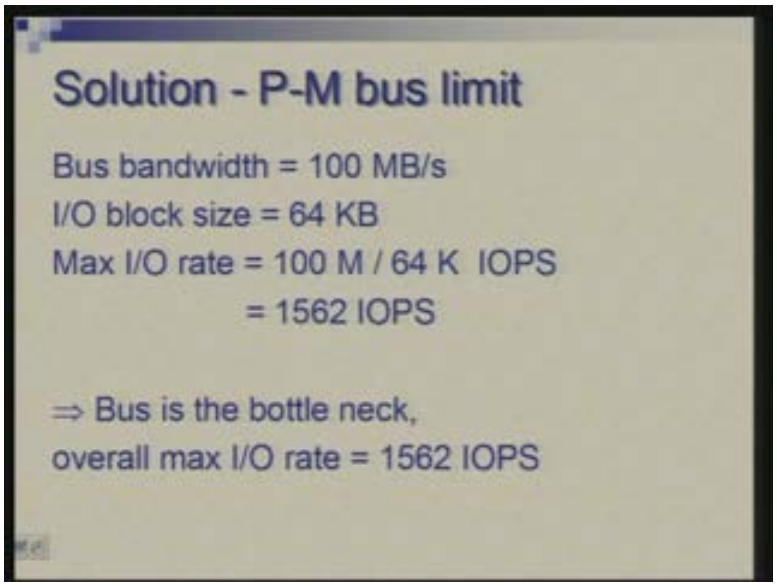
So now, to solve this first let us analyze what are the limits of individual components are and the limit will come from the fixed component that is the processor and the memory and corresponding bus.

(Refer Slide Time: 11:42)



So the processor as to execute 100 plus 50 that is 150000 instructions per I/O block and this execution happens at the rate of 300 milli 300 million instructions per second. So maximum I/O rate as far as processor is concerned is 300 million divided by 150k so many I/O operations per second is possible so this turns out to be 2000 I/O operations per second. So if processor if rest of the system was capable of matching this rate, that the processor is ready to do 2000 operations as defined here per second.

(Refer Slide Time: 12:33)

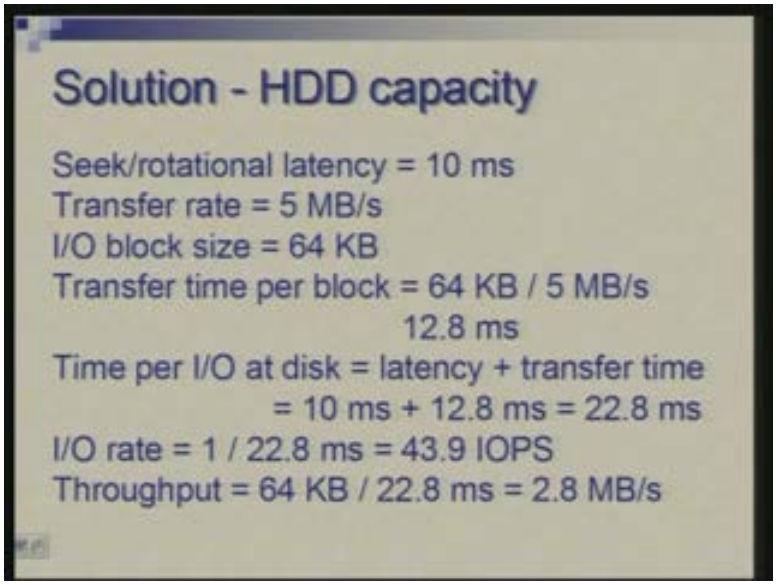


The next object is processor memory bus. So what is the limit as far as this is concerned. The bus has a bandwidth of 100 megabytes per second. Each I/O block it may come from

some disk and some SCSI controller, ultimately it will land up on the same bus and the bus will have to carry out transfer of 64 kilobytes. So now from these two figures you can figure out 100 million bytes per second is the capability, 64k bytes you want to transfer so basically you get 1562 I/O operations per second.

Now out of these two figures 2000 of CPU and 1562 of PM bus obviously this is a smaller figure and therefore that is the bottleneck. So processor has certain ((....13:23)) capacity and this will dictate the rate. So, our first transfer is that overall maximum I/O rate is 1562 I/O operations per second. Now the question is how do we achieve this? How many disks we need to put to do this?

(Refer Slide Time: 13:43)

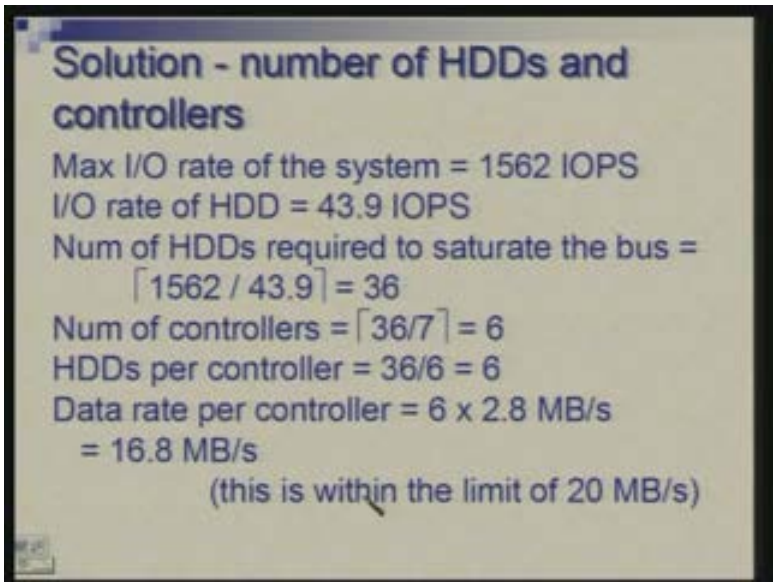


So let us look at one disk at a time. the seek plus rotational latency is 10 milliseconds and with 5 megabytes per second of transfer rate and 64 kilobytes of block size the transfer time would be 64 kilobytes divided by 5 megabytes per second it comes out to be 12.8 milliseconds. So now, for each I/O operation basically disk has to spend 10 milliseconds followed by 2.8 milliseconds and therefore on the whole each disk when it gets involved in I/O is busy for the sum of these two figures that is 10 milliseconds latency plus 12.8 milliseconds transfer time so a total of 22.8 milliseconds.

So, if disk was to be continuously asked to seek and then transfer seek and transfer, the rate at which this can churn out blocks is reciprocal of 22.8 milliseconds or 43.9 I/O operations per second. So the throughput the throughput which actually disk is sustaining is 64 kilobytes have been generated in a time of 22.8 milliseconds so it is although transfer rate was 5 megabytes per second effectively we are getting 2.8 megabytes per second is the demand each disk is placing on the SCSI controller so that much of data it is pumping on the SCSI bus and SCSI bus will concentrate from various disks and send further.

So now, to begin with let us we want to use this figure to get the number of disks.

(Refer Slide Time: 15:40)

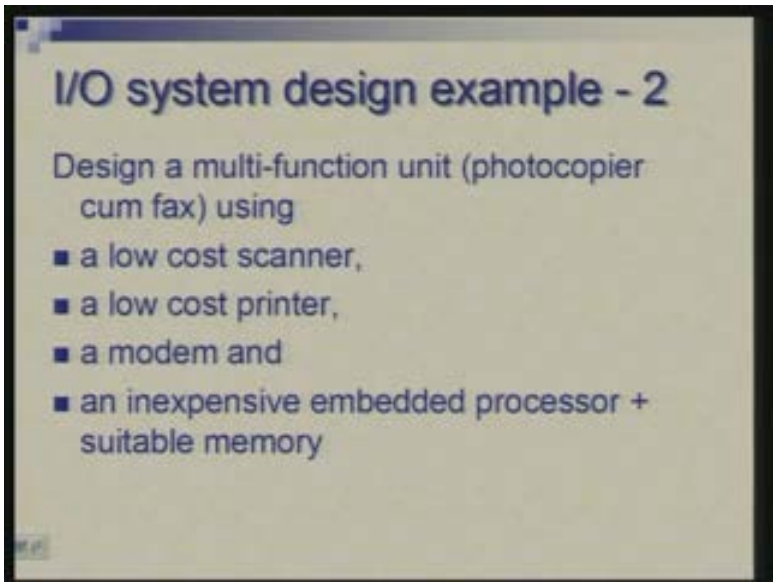


We have seen that maximum I/O rate of the system is 1562 I/O operations and I/O rate of the hard disk drive is 43.9 so the total number of disk drives which you require to saturate the PM bus is the ratio of these two 1562 divided by 43.9 so rounded off to integer you get 36 disks.

Now the question is of how do you distribute these 36 disks to various SCSI controllers? The number of controllers you require at least is 36 by 7 sealing of that because each controller is able to at most sustain at most control seven of these. so now this limit of 7 is from the point of view of its addressing capability. We also have to check whether the data generated by these disks will be handled by the bus of the SCSI controller. So the number of controller by this ratio comes out to be 6 and accordingly the number of drives which each controller is assigned, if you distribute them uniformly, it comes to be 6 so six 36 drives are grouped into six groups and each controller will see a throughput of six times 2.8, we have we have seen earlier that each disk is actually pumping data at the rate of 2.8 megabytes per second which is 16.8 megabytes and therefore this is within the limit of 20 megabytes of SCSI bus so therefore our answer which we have got here is fine that you need 36 disks and six controllers. So that that was one exercise where the parameters were number of disks and number of controllers and we were able to analyze and decide how this is figured out.

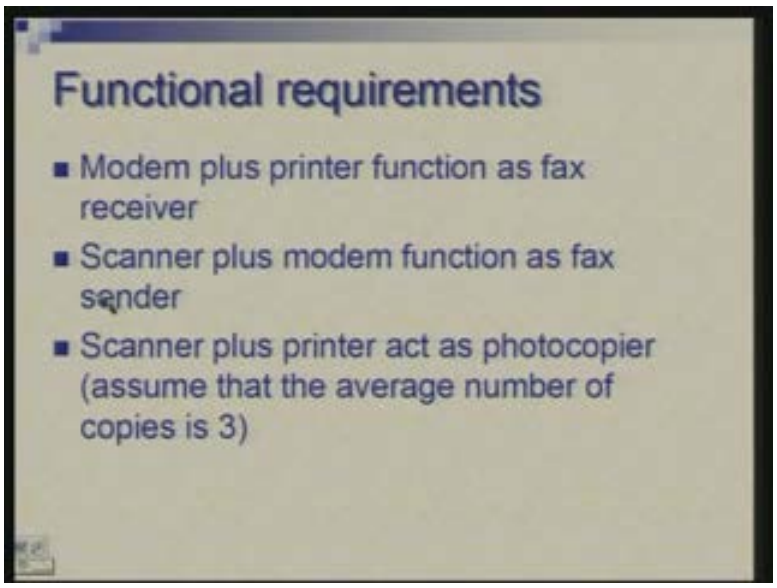
Before I go to next example is there any question?
So they are very simple calculations, very simple rules there are no complicated formula you just have to see how things are put together and there is lot of actually common sense which is behind this. Once you understand the operation of how things are happening rest is very straightforward.

(Refer Slide Time: 18:09)



The second problem we take is again design problem. What we are trying to do is we want to design a multi-functional unit, particularly something which can do photocopy and send and receive faxes and the way we want to do this is we take a low cost scanner, a low cost printer and a modem and connect them through some inexpensive processor and suitable memory. So put these peripheral devices together and we have certain performance objectives I will come to that in a moment; so functionally these will take care of this that is you can scan a document and fax, you can take an incoming fax and print it, you can also scan and print which effectively like a photocopy operation. So this is what I already said.

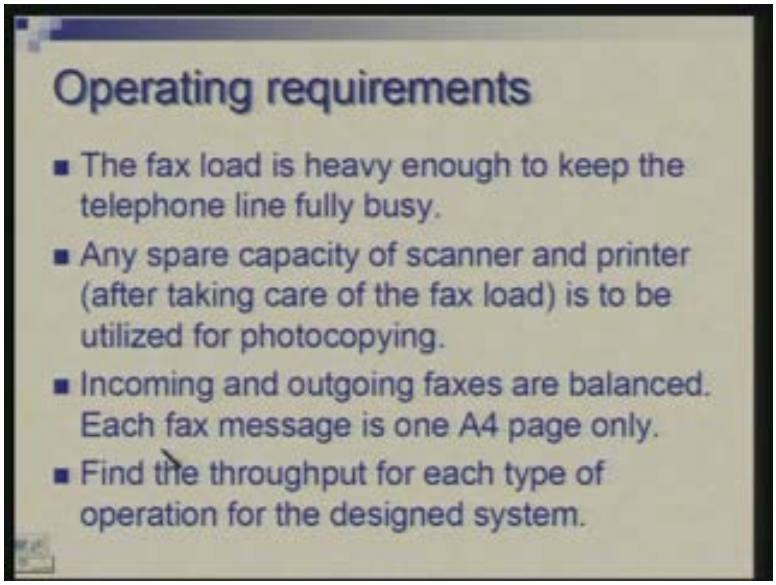
(Refer Slide Time: 19:13)



Modem plus printer function as fax receiver
Scanner plus modem function as fax sender and
Scanner plus printer acts as photocopier

So of course in between you have a processor and memory sitting in the background. Now we also assume for our further calculation that every photocopy operation may require certain number of copies that is you scan and print a couple of copies. Let us say, the average number of copies you print for every document you scan is 3.

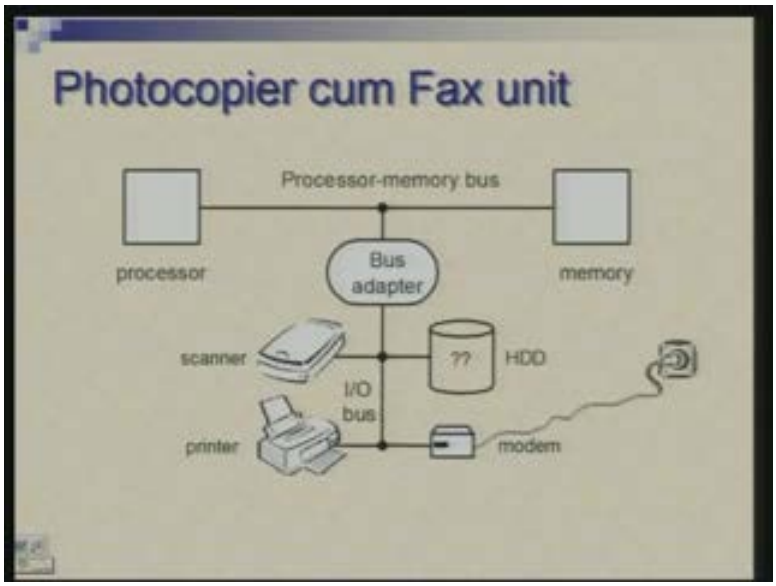
(Refer Slide Time: 19:50)



Then there are certain operating requirements. In what kind of scenario you want this to work. So let us say there is a heavy fax load so continuously fax are coming and going and we will work with the assumption that the telephone line to which this is connecting is continuously kept busy so let us say half the time it is..... incoming and outgoing faxes are balanced that means almost equal number of fax is coming and equal number has to be sent out and each fax message is a single page; we are not we do not want to worry about complication of sending multiple pages.

So just imagine that each..... we are only dealing with A4 size pages. So both printer and scanner have to work with that. And after satisfying the fax load whatever is the spare capacity of scanner and printer is to be utilized for photocopying and we want to get a feel of we want to know how many maximum at what rate we can do photocopying process. So we want to find eventually the throughput for each type of operation. That means what is the rate of fax message that would be sustained and what is the rate of photocopying that could be achieved.

(Refer Slide Time: 21:15)

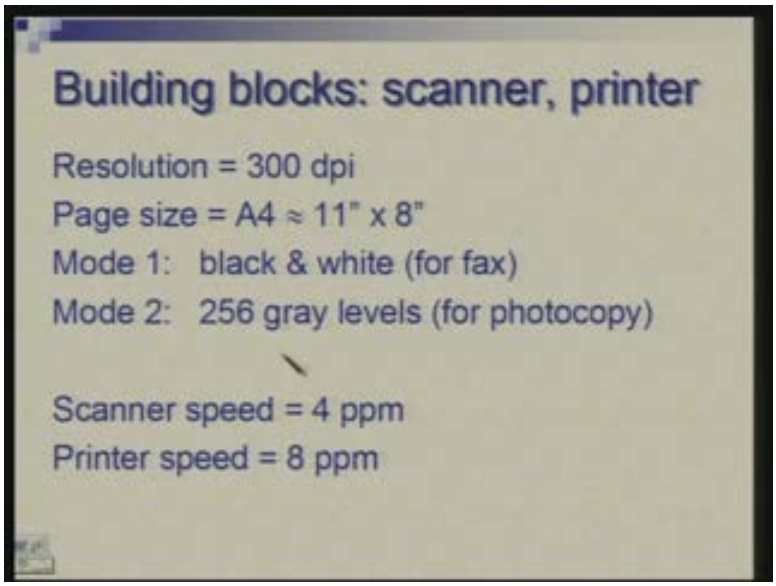


This is a this is a block diagram of the whole system we have in mind. We want to make it a low cost setup so there is processor memory bus and a single I/O bus. There is a bus adapter sitting here, a scanner, printer, modem which hooks to a telephone line and maybe we might need a disk drive so that is a question mark although it does not come directly from the specification.

So we want to build it as a sort of dedicated system and you know once it is designed and all assembled we want it to be just doing this work as a fax machine plus a photocopy. So whatever program is there could be possibly put in non-volatile memory and you can if disk is not required we will not put that, it will save may be little bit of money. So of course these am ignoring the details that these devices will eventually perhaps connect to different types of port or a bus.
For example, modem may go to serial port, these may connect to USB port or printer may go to parallel port and so on. So we will ignore that last point detail and assume that there is a single bus from which eventually all these interface will be hooked up.

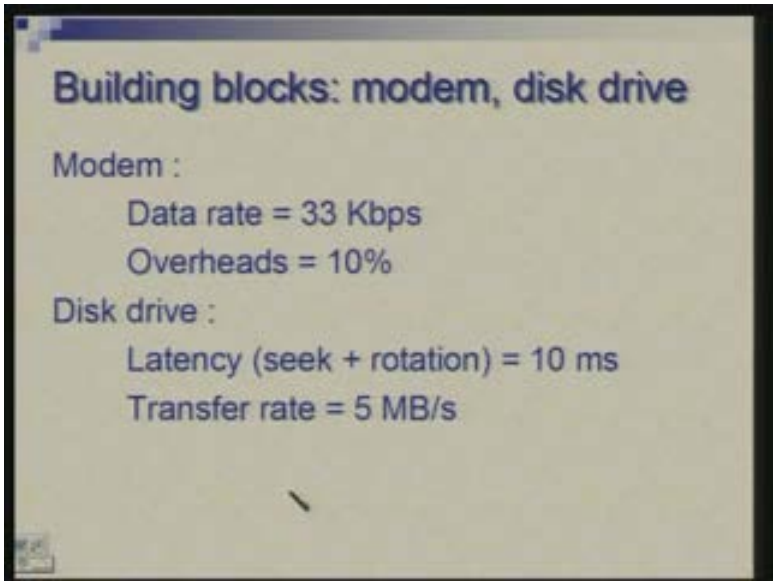
So now let us look at the building blocks which are available. So something will be sort of fixed and somewhere there will be choice and we will figure out what choices we make. So let us say printer and scanner are available to us; we will take a low or medium resolution print machines 300 dpi we do not need a 600 or 1200 dpi the page size fixed page size has to be handled and just to round it off let us say it is 11 inch by 8inch A4 size and it can both these can work in two modes: one mode is when you simply work with black and white so there are no gray levels, each pixel is either black or white. And in second mode you have 256 gray levels which you can represent with a byte.

(Refer Slide Time: 23:48)



So, for fax purpose we will use mode 1 and for photocopy purpose we will use mode 2. Let us say this scanner works at the speed of four pages per minute and printer works at the speed of eight pages per minute.

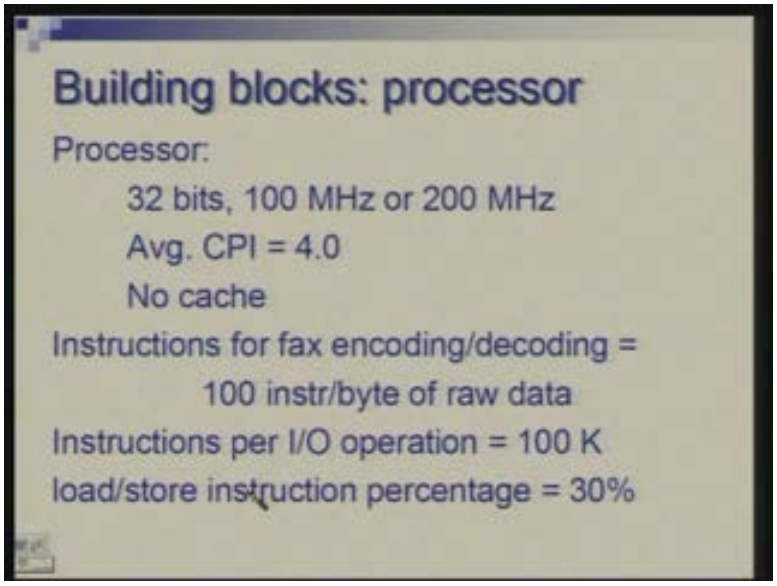
(Refer Slide Time: 24:30)



Continuing with the building blocks we have a modem available; we are not taking top of the line 56 Kbps modem, let us take 33 kilobits per second modem and it also let us say there is a 10 percent overhead that is whatever data is to be sent a few bits as part of the protocol has to be added by the modem and let us say it is 10 percent extra data which

gets added. There is a disk available which has a latency of 10 milliseconds and transfer time of 5 megabytes per second.

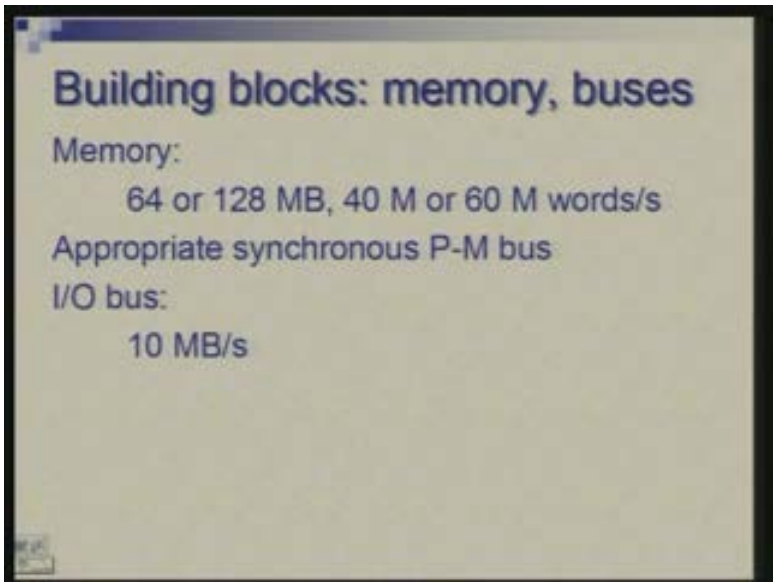
(Refer Slide Time: 24:50)



We want to use again a low cost processor which is available in two options 100 megahertz or 200 megahertz so it is not a gigahertz type of processor and it is not pipelined, it has a CPI of 4, it does not have cache so it directly accesses memory. The program which will be running on this to carry out this operation, the fax requires certain encoding and decoding so you have the scanned image of the page that needs to be encoded in certain form before you send and this encoding actually compresses the size of the data. Similarly the fax which is received has to be decoded and put back into a pixel matrix. So I will maybe I think that number is mentioned later. So we will assume that there is a compression or decompression of factor of 10 is to 1 which is happening and this process of encoding or decoding involves 100 instructions per byte of raw data.

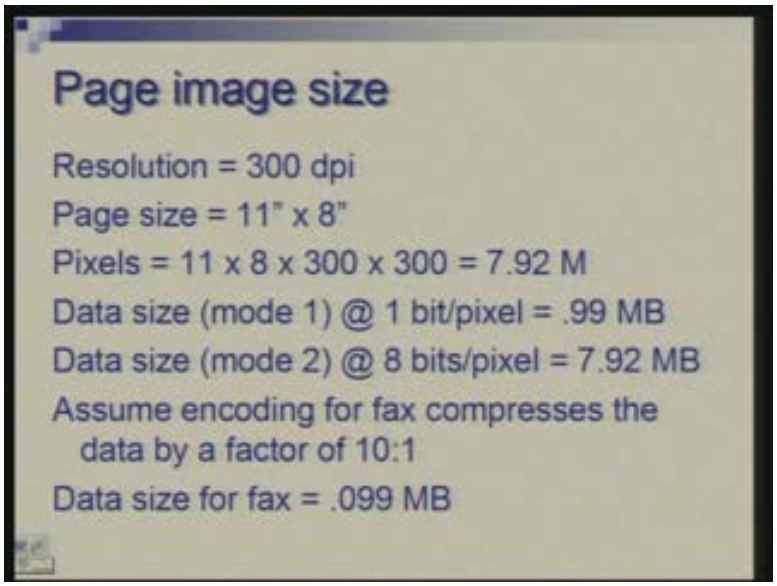
So let us say you have scanned the image, if the image is 10 kilobytes you will require 100 instructions to be executed per byte or if it is 1 megabyte accordingly. So the algorithm of encoding or decoding either way has to incur some computational load which is spelt out here; also each I/O operation I am assuming here a flat rate of 100k instructions per I/O operation. So, for example, sending a fax would involve scanning and sending to the modem. So each has to be setup in a DMA manner and I am assuming that each will require each will consume 100 instructions per processor and on the average the program would have let us say 30 percent of the instructions which are of load store type which will make additional access to the memory.

(Refer Slide Time: 27:06)



The memory is available in two sizes: 64 megabytes or 128 megabytes and two speeds 40 million words per second or 60 million words per second and there is an appropriate synchronous bus which will connect it to the processor, the I/O bus is to be used, there is no choice here, it is 10 megabytes per second. So now with all this we can proceed and try to put things together, see what kind of rates we get and what choices could be made for processor and memory.

(Refer Slide Time: 27:53)

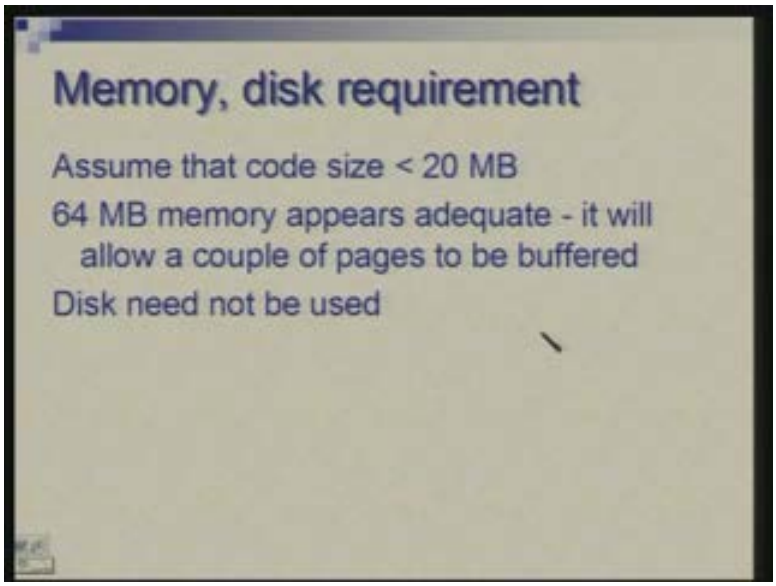


So let us first try to understand what information each page carries. We have seen that resolution of the page is 300 dpi and with page size of 11 inches by 8 inches we get total

number of pixels as 11 into 8 into 300 into 300 which is 7.92 million so those many pixels are there. Now depending upon the mode whether it is black and white mode or a gray scale mode you will require certain number of bits per pixel so mode 1 is 1 bit per pixel and mode 2 is 8bits per pixel. So you get, for 7.92 mega pixels you get 7.92 megabytes or 0.99 megabytes so these are bits and we have converted them into bytes.

Now the compression in fax and coding was 10 is to 1 so when you have to send or receive a page of these many megabytes actually what goes in the fax line will be one tenth of that so 0.099 megabytes.

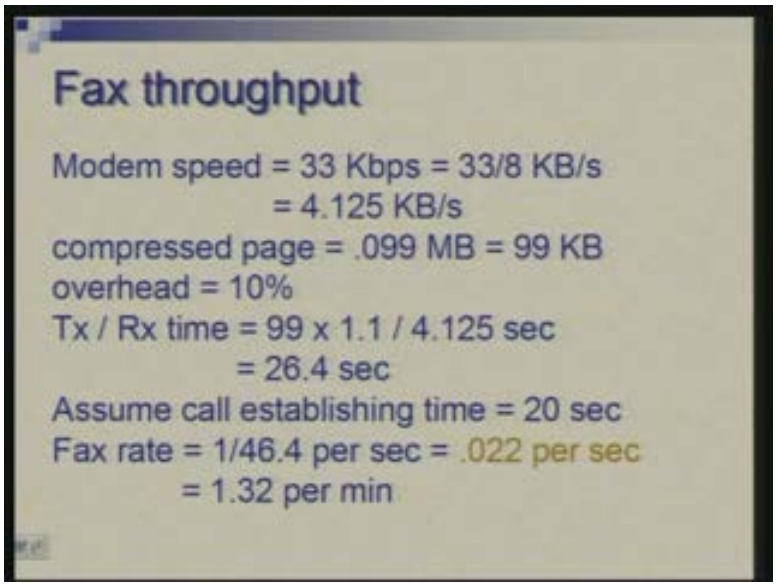
(Refer Slide Time: 29:07)



Now memory and disk size, let us have a look into that. So let us assume that the code or the software will take something less than 20 megabytes so 64 megabytes of memory will be adequate in the sense that we do not have to put 128, 64 megabytes will accommodate easily the program and it can also buffer several pages which has been scanned or which are waiting for printing so each of these requires something like 8 megabytes. So 64 megabytes gives us reasonable cushion for accommodating a couple of scan pages and since everything can be accommodated in memory we can eliminate the use of disk.

We can have the memory itself as a nonvolatile memory or we could have volatile memory augmented with flash memory because flash memory is slower so you will not like to execute program directly from flash. Instead of disk you can put a flash memory.

(Refer Slide Time: 30:21)

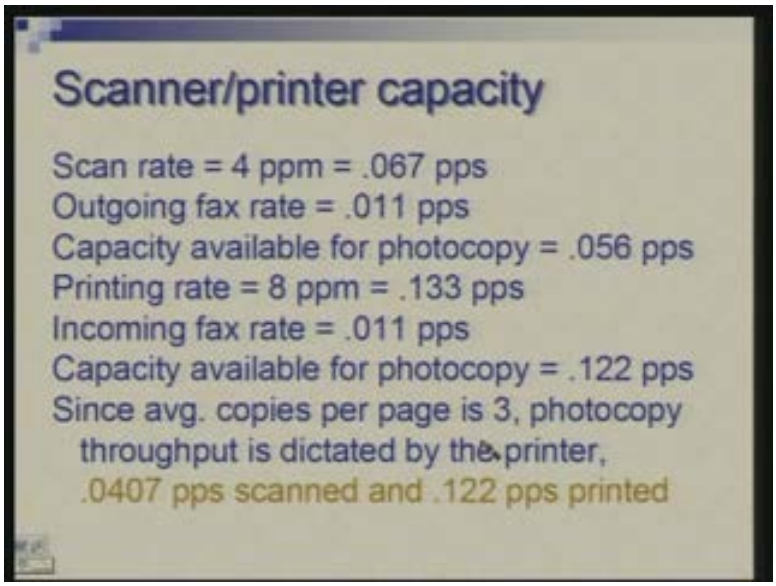


Now let us go to the fax process. So modem speed is 33 kilobytes per second or in terms of bytes you can divide by 8and get 4.125 kilobytes per second and the compressed page or encoded page is 0.099 megabytes which is same as 99 kilobytes and with 10 percent of overhead we can find out transmission or reception time. So 99 multiplied by 1.1 because it is overhead divided by the kilobytes per second rate we get 26.4 seconds so each page takes this much time to be sent or to be received plus now since this happens over telephone line there is time taken to establish a call; either there is an incoming call and modem will respond to that established connection or when it is to be sent number would be dialed and connection would be established. So let us say 20 seconds either way gets used up for this purpose for establishing a call.

So the two put together 26.4 and 20 is total time which is actually taken for which the modem and line are busy with 46.4 seconds and therefore fax rate the rate at which you can be handling faxes incoming and outgoing all together is reciprocal of this which is 0.022 per second or you can have a more convenient figure in per minute 1.32 per minute.

So basically this is one of the responses; this setup (Refer Slide Time: 32:18); here the things are actually limited more by the modem speed; the rate at which you can send fax and it is not so much influenced by the other components which are in the picture. So this is the rate which modem can sustain and rest of the system has to actually work with this.

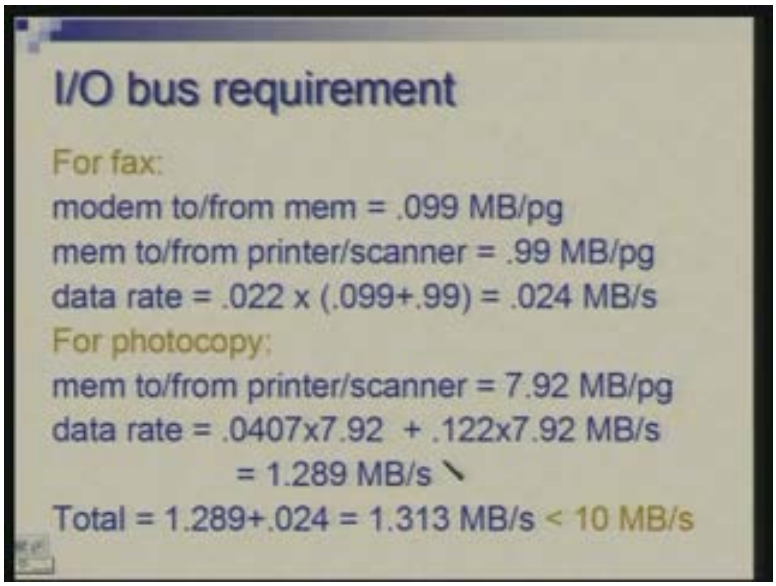
(Refer Slide Time: 32:44)



Now let us turn our attention to scanner and printer. So, scanning rate is 4 ppm or in terms of pages per second it is 0.067 so many pages per second, the outgoing fax rate 0.022, we divide into 0.011 incoming and 0.011 outgoing. So now the scanner has to match the outgoing faxes and therefore if so many faxes are going out, so many pages have to be scanned so the capacity of scanner available for photocopying is the difference of these two; it has a total capacity of 0.067 if you want to continuously scan pages so many pages per second could be scanned and out of that so many will be faxed out rest could be used for photocopying purpose.

On the other hand, printing rate is 8 ppm or 0.133 pps page per second, incoming fax rate is 0.011, the capacity available for photocopying is 0.122 pages per second. So these are spare capacity and how many photocopying operations we can do?
since If you recall I mentioned that for each page you scan you will print on the average three copies so printer has a slightly higher load and this figure is certainly less than three times that. So basically printer would dictate the photocopying rate and we say that we will be able to churn out so many photocopy pages per second and that means one third of that we need to scan on the average. So 0.0404 pages per second are scanned and 0.122 pages per second are printed in the photocopying process.

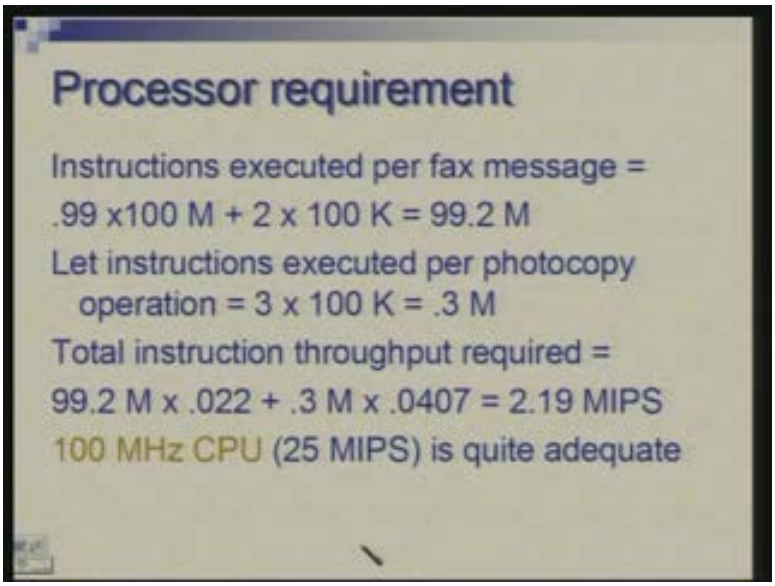
(Refer Slide Time: 34:50)



Now we go to the I/O bus requirement. so we have looked at each peripheral. we have looked at the modem, scanner and printer, we have gotten away with the disk so now I/O bus has to basically sustain the throughput which is being generated by these. so when you are receiving a fax you have a data coming from modem and it goes to the memory, it passes through I/O bus and similarly for outgoing fax it comes from memory and goes to the modem so it passes through the I/O bus. Similarly, from memory to printer or scanner to memory will happen at this rate, for each page per page this 0.099 megabytes per page this is the modem traffic, 0.99 megabytes per page is the printer scanner traffic. So the data rate which the I/O bus sees for the fax operation is 0.022 faxes per second multiplied by that means so many pages per second and these many megabytes per page. So the product gives you 0.024 megabytes per second so that is the demand placed on the I/O bus by the faxing operation.

For photocopy operation there is a traffic between memory to printer or scanner to memory which is 7.92 megabytes per page and the rate at which we are doing photocopy we are scanning 0.0407 pages per second that sends out this much of data and 0.122 pages per second we are printing so that again demands that much data so sum of this is 1.289 megabytes per second. So that is the total demand which is placed by photocopying operation when it is running to its peak on the I/O bus. So, sum of these two: 0.024 and 1.289 megabytes per second is 1.313 megabytes per second is the total I/O demand and the bus which has been provided to us is 10 megabytes per second so we are very comfortable with that and we can accept that solution.

(Refer Slide Time: 37:33)

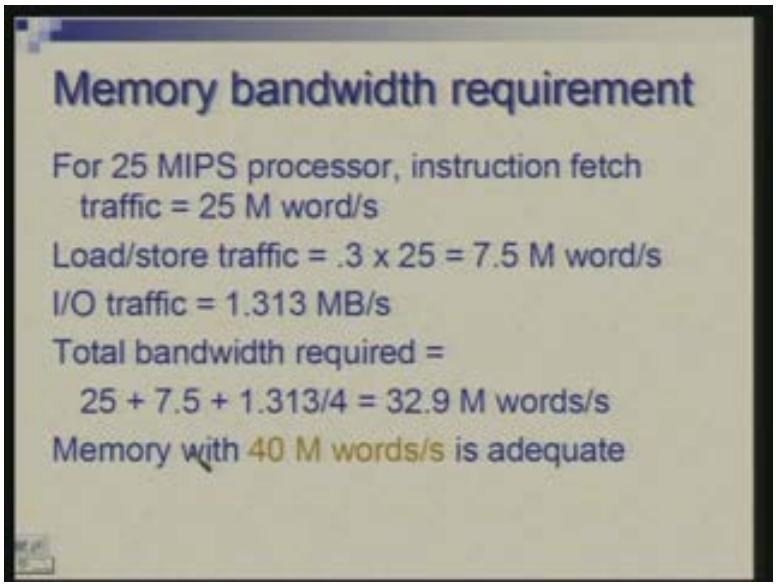


Now let us see how the processor is doing. The instruction which it needs to execute per fax message is..... I mentioned that there are 100 instructions per byte so 0.99 megabytes is the page data so this product gives you the number of instructions which will be required for either encoding or decoding a page and each fax in or fax out involves two operations so either reading from the modem or writing to the modem and on the other hand either scanning or printing so each will require 100 K instructions so a total of instructions which have to be executed is these many (Refer Slide Time: 38:28) for either encoding or decoding, these many for either sending or receiving so this turns out to be..... actually this is a very small component as you can see so bulk of the computational power is going in encoding or decoding, the I/O part is much smaller. So this is 99.2 million instructions per fax incoming or outgoing.

The instructions executed per photocopy: now, I have made a mistake here, it should have been 4. So, on the average you are reading you are doing one scan and you are doing three prints so it should have been 4 here so each requires 100 K instructions so a total of 0.04 M. The total instructions throughput required is this much plus 0.04 sorry this into the fax rate and this into the scan rate so together you will get 2.19 and this is any case is small so a small factor it will make a minor difference here so let us not worry about that.

The total instruction rate required on the part of processor is 2.19 million instructions per second and we talked of 100 megahertz or 200 megahertz CPU with 4cpi so basically 100 megahertz CPU will require..... we will actually deliver 25 MIPS which is very comfortable to handle that so we do not need to go for 200 megahertz processor, we can work with 100 megahertz processor; in fact if there was a 50 megahertz version available at a cheaper rate we would have picked that. So in any case, out of the choice given, this is quite adequate.

(Refer Slide Time: 40:33)



The memory bandwidth..... now we are not utilizing the processor to the full capacity but suppose after having seen that processor is under utilization maybe we might put in some other tasks on that. suppose processor was to run at full capacity it was not to stall or lack of no work or for lack of work; 25 MIPS would require 25 million instruction words to be fetched per second and load store we noticed earlier was..... 30 percent of the instructions are of load store type so additional demand would be 0.3 into 25 or 7.5 million words per second. So this is the traffic which the processor will generate for the memory; 25 million words per second and 7.5 words per second instruction traffic and data traffic, load store traffic.

The I/O traffic is also showing up on the memory bus which is very small, 1.313 megabytes per second. So all put together the memory bandwidth needs to be capable of handling this 25 plus 7.5 plus 1.313 divided by 4 to get million words and this was million bytes (Refer Slide Time: 42:03) so that is only 32.9 million words per second so we can pick up the memory which has out of the 40 and 60 two possibilities which were shown we can pick up the one which as 40 million per second capability so that is adequate for our purpose.

So basically, as an end result, you can see that we have we can get certain throughput rate for fax and photocopy operation and we are able to sustain that with the simpler of the processors and slower of the memories. So that is all as far as this example is concerned. If you have any questions; I hope these calculations are fairly straightforward for you; you only need to proceed in a certain manner and there is no fixed rule or procedure to solve such problems you know; each problem may be of its own kind and we all along we should understand what are the assumptions we are making in all these ((sc.....43:25))..... we are assuming that everything is in a steady state so there are no accumulations of data or nothing is getting starved for data so there are no congestions or

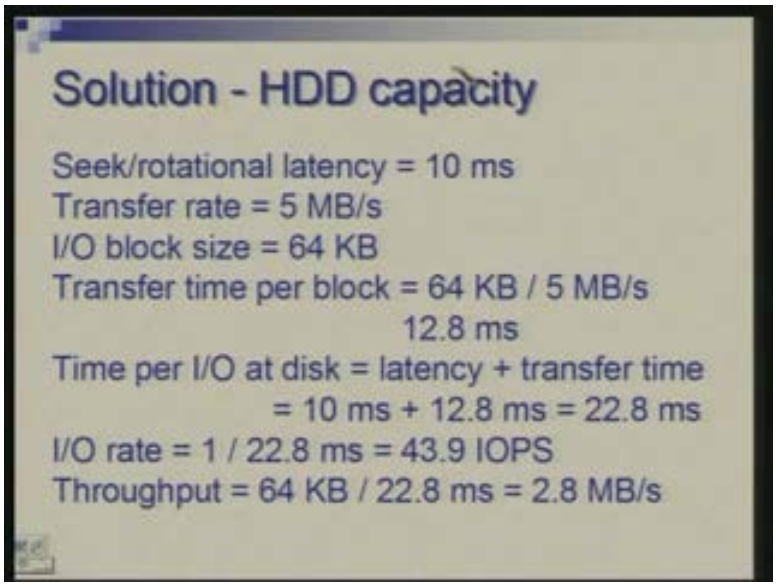
no starving, all traffic, all data traffic or instruction traffic is flowing smoothly and there is a steady state.

So, if you have multiple buses converging to one bus you simply add the traffic coming out of multiple devices which come on the same bus. So we have not analyzed here the delays; that is, once a fax comes how long it takes to print so that was the latency issue. Or from a time you press a scan button how long it takes to get three copies printed out, so that we have not analyzed, that is somewhat more complicated and often you need to use you may not be able to simply add the delays because as you are these operations are all interleaved; as you are receiving and sending faxes photocopying is taking place so these are interleaved and there may be sometime queuing delays which we have not worried about in bandwidth calculation but when you look at the latency you would need to..... for example, a page has been scanned and it needs to print it so how much is the delay between the two; when will the processor execute those instructions required; when will the bus be free; when will the printer be free so that analysis may require tools like queuing theory for example, or sometimes even if that is not adequate you would need to actually simulate the whole scenario and then study how much latency is involved. That is all for the moment.

Any questions?
[Conversation between student and Professor: (46:05)] yeah that is the rate at which disk is able to transfer data once you reach the appropriate point. Yeah, yeah so that is where things will get buffered in the controller.

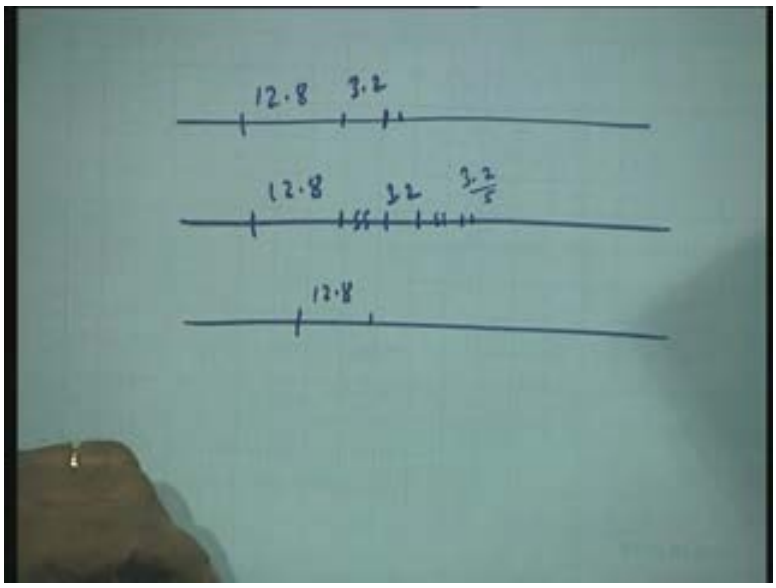
So let us say..... what you are saying is very realistic and it is going to happen; it may not be simultaneous but let us say one disk has started transferring maybe it is halfway through, another disk is ready. So now what will happen, it may be there will be some buffing at the disk and there will be buffering at the at the controller end. Also, we would need to understand how actually the two streams of data get put on the on the bus, do you allow them to be interleaved or not. So let us say they are not interleaved. So the data which is taking 1 milliseconds to come out of the..... as it is coming out of the disk the disk is active for 12.8 milliseconds but the rate at which it goes on the SCSI bus is 20 megabytes per second so let us say there is no interleaving of the blocks from two different sources and therefore the SCSI bus will be occupied with this much of data for how much time for 64 kilobits of data over let us see..... it will be one fourth of that actually so this is since this is 5 megabytes per second so some 3.2 milliseconds so basically what will happen is that if you are not allowing interleaving it will better to buffer this 64 kilobytes somewhere either at the disk controller end or at the SCSI controller end and then put it for 3.2 milliseconds on the bus.

(Refer Slide Time: 48:28)



Let me try to show you..... so one disk let us say spent 12.8 eight milliseconds and let us imagine that this data is going to some buffer and then over a period of 3.2 millisecond it is occupying a SCSI bus and over a shorter period how much was the memory bus I think 100 megabytes? Yeah, so one fifth of that, so a fraction of milliseconds it will be present on the processor memory bus. Now, in between here, although I am showing it contiguously but there could be gap here. So in actual practice we should be open to have a situation like this: 12.8 then some gap somewhere down the line 3.2 milliseconds it spends on the SCSI bus, somewhere later I do not know how much break it will spend 3.2 by 5 so many milliseconds on the P M bus.

(Refer Slide Time: 50:32)



So now each transaction you can imagine has these three parts and you have another one such thing coming from a different disk. So it may be its 12.8 was happening here (Refer Slide Time: 50:12) and its 3.2 may get just placed after that. So you have basically these streams coming from different disks, they get buffered and one after another they will be placed on the SCSI bus.

Similarly, the adapter which is sitting on the main, when it is put on the processor memory bus again they will be sequentialized put one after another. So let us say it gives a thin burst on P M bus. So suitable buffing at various ends will take care of this. Does it answer your question? Any other question? Okay thank you.