Computer Organization
Part – I
Prof. S. Raman
Department of Computer Science & Engineering
Indian Institute of Technology
Lecture – 1
Introduction to Computing

Good day everybody. In this series of lectures on computer organization, I propose to deal with quite a few fundamental aspects so that you would know what the basic elements in any computing system are and how they interact. Normally people mix up computer organization and computer architecture. I would also identify both these and then tell stress on what these are. Now let's start with the word computer.

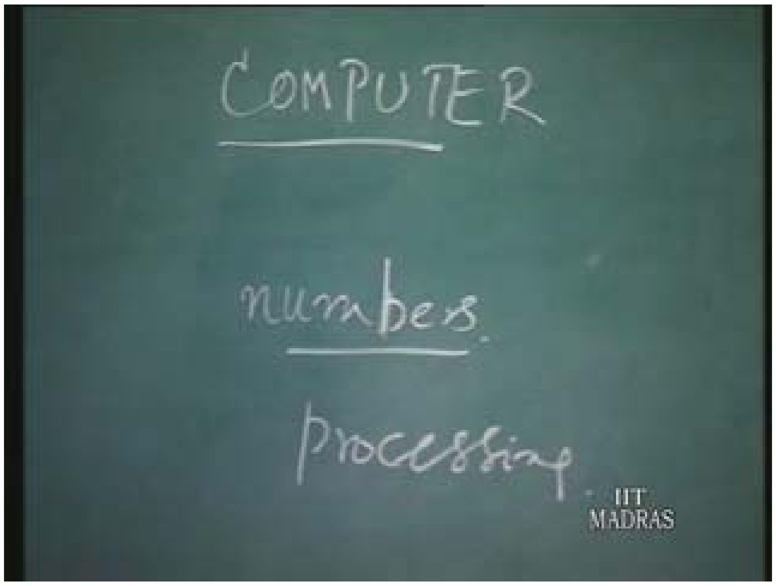
(Refer Slide Time 00:01:52)

Essentially, this has come about with the root compute, which means we have some numbers, and then there is some computation to be done on these numbers.

(Refer Slide Time 00:02:07)

This is how the whole thing started. But today, we also see that apart from processing these numbers that is, carrying out some kind of process the computer is also communicating. Computers are widely used for communication purposes.

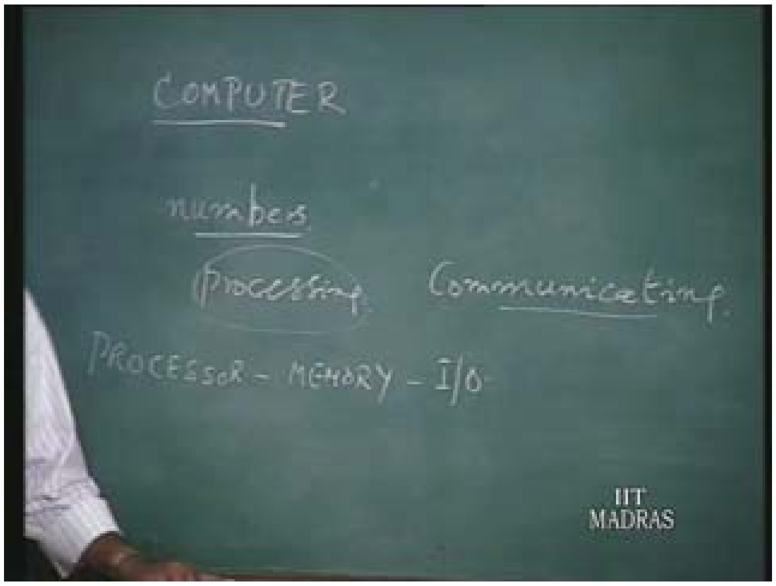
(Refer Slide Time 00:02:27)



This, in fact, has come about of late; so we will start with the original one; that is, processing, and that too with the processing of numbers.

We saw that computers are essentially used for processing and I said that's how the whole thing started, that is historically speaking. Basically any computing system will consist of a processor, for carrying out processing; then there is memory. So essentially, these two are the elements; that is, processor for carrying out the processing, and memory, essentially to store the numbers. In addition to these, we also have what we may call simply as I by O or input by output.

(Refer Slide Time 00:03:38)

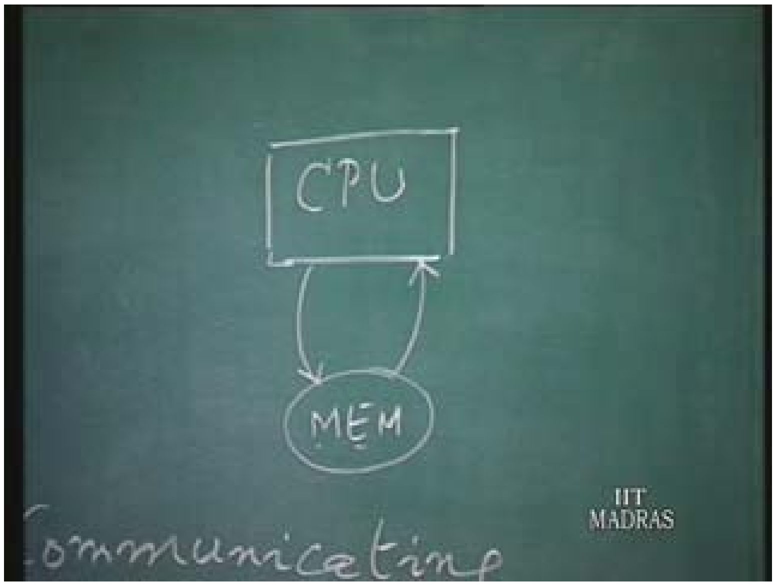


Later on, I will tell the relation between input/output as memory, because in fact I would say an input/output is an extended memory. What did I say about memory? I said input/output also will store the numbers, but in a different form. We will come to this later on; there is another aspect of input/output also. That is, in case we consider the computer system as a machine, then input/output helps man to interact with the system. That is another aspect of it; we will come back to this. I would say that the processor, which, in those days, was called a central processing unit or just CPU. It was called so mainly because processing in those days was centralized and hardware was in one place; that's how they got the word. CPU is the historical word, central processing unit, and the memories are the two things.

(Refer Slide Time 00:04:59)

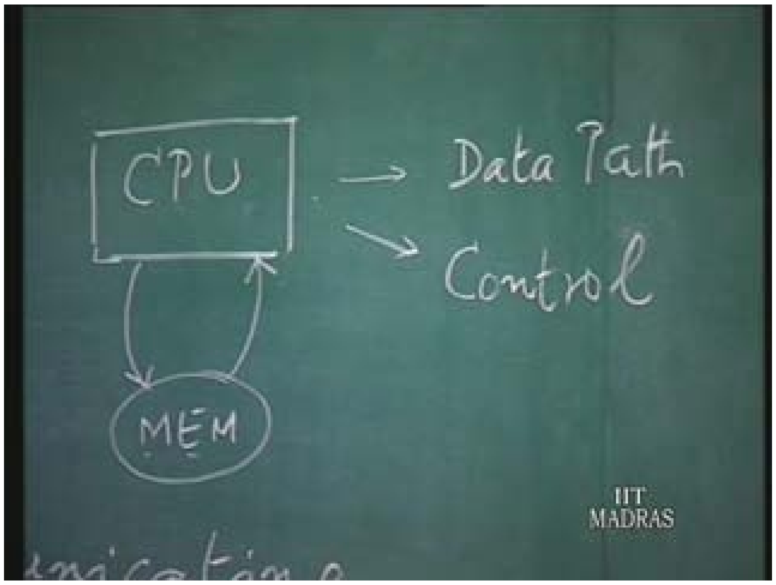
I will just briefly indicate to you – supposing we have this block, CPU or central processing unit, and then, let's say, the memory, let me just use a different symbol for this. Now it's like this: we have the data stored in the memory and the CPU is capable of processing. So what the processor will do is to keep addressing the memory; it will get the data from the memory, then it will carry out some processing.

(Refer Slide Time 00:05:37)



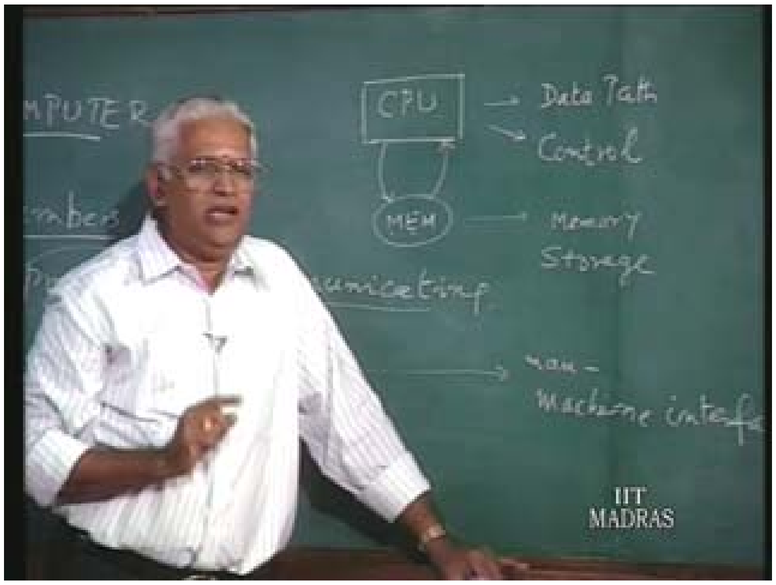
Essentially, the whole computer organization boils down to having two main units – one for processing and one for storage. Already I mentioned I by O or input/output; I would call it extended memory, in which case essentially what we have in any computing system is processor and memory. What does a particular processor consist of? Obviously, processor is the one, which is going to deal with the data and so the processor must be able to set up some kind of a path to handle the data. Data path setting – this is one aspect of the processor. Apart from this setting the path for the data to be processed, it should also be able to carry out the processing through a sequence of control segments.

(Refer Slide Time 00:06:49)



What we see here is that the processor will essentially consist of data path circuits and control circuits; and memory, as I said, is for storing. The one which would directly interact with the CPU path is still called memory. The one which would interact with the CPU, but not directly, is called the storage. Though I was using the word storage earlier to store the numbers in memory, it so happens that now the one which is closer to the CPU has come to be called memory; the one farther from CPU, but still effectively stores information or data, is called the storage. For instance, we talk about magnetic tapes, disk system, and such things in this storage; whereas semi-conductors, say RAMs, that is, random access memory and read only memory, will be coming under this particular category. For the moment, we will just define the I by O part also; we will deal with it a little later. I would just put it as I said earlier that essentially it is for man–machine interface.

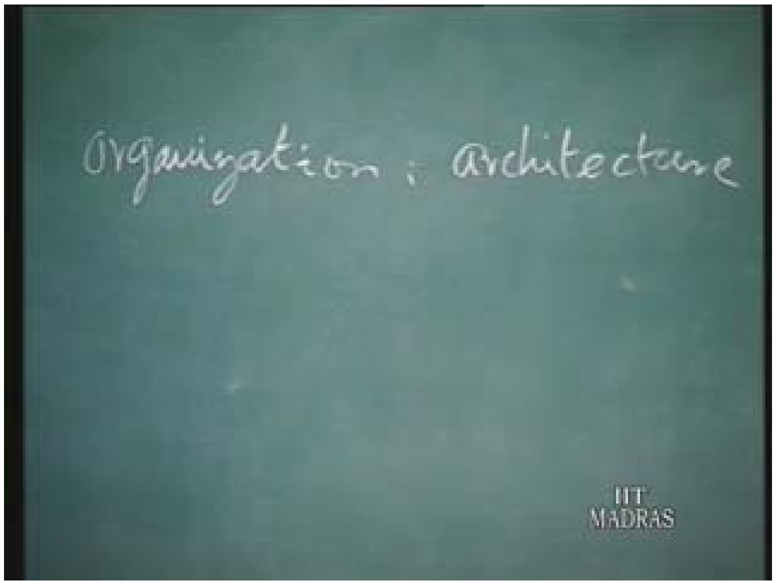
(Refer Slide Time 00:08:29)



Then there is another aspect to this. Depending on the kind of processor or the kind of computing system, the system may take the data from the memory, which means it is the data which had already been stored in the memory or it may take the data from the input by output system directly, which means from the external world, or we may say, the real world data. So we may have the data coming directly from the real world through the I by O devices, or we may have the data which has been previously stored in the memory or storage.

This way you can say that memory and I by O are not really different. That's why I said I by O can be considered as extended memory. Later we will see in detail that CPU deals with the memory in almost the same way in which the CPU will also deal with the I by O. This is one aspect of it. While starting the series, I said people generally mix up computer organization and architecture. In fact, in some places, many people do not even make any distinction between these. There is a slight difference. The organization – what does it deal with? The service, and the architecture part – there are two things. So I will take one and set the other against that.

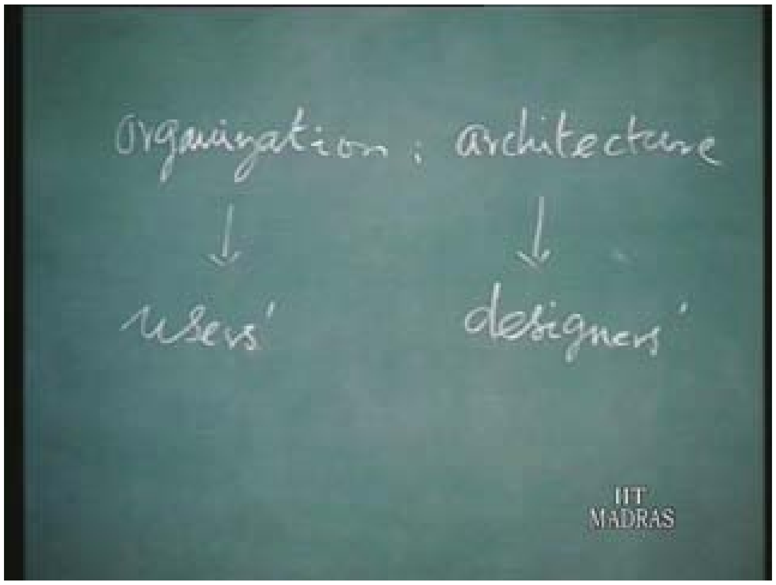
(Refer Slide Time 00:10:23)


Organization : architecture

IIT MADRAS

You can study the computer system either from the user's point of view or from the designer's point of view.

(Refer Slide Time 00:10:44)

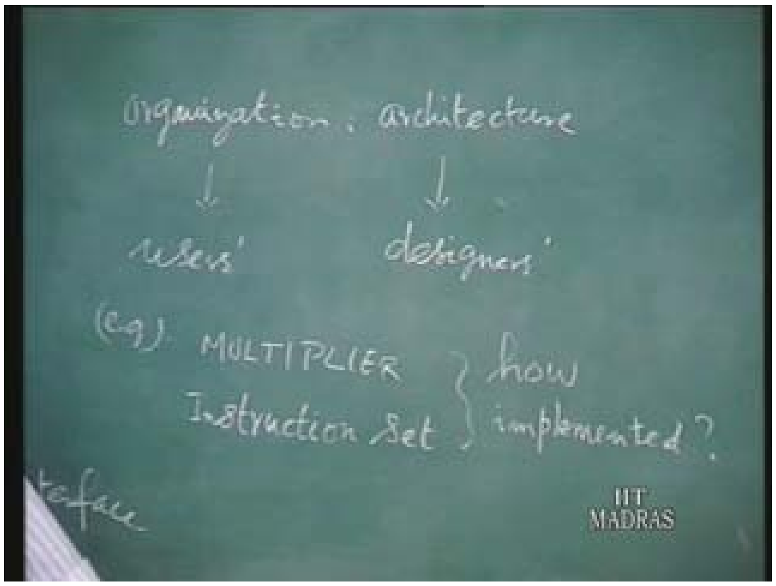

$$\begin{tikzcd}
\text{Organization} \arrow[d] & : & \text{architecture} \arrow[d] \\
\text{users'} & & \text{designers'}
\end{tikzcd}$$

IIT MADRAS

I think um these two different viewpoints really hold well with any system. For instance, we talk about something like a car driver and car mechanic – it is the same way here too.

So Computer Organization is a study of computer system from the user's point of view and Computer Architecture is the study of the same system from a designer's point of view. I will explain this further through an example. Let's take an example: suppose a computing system has a multiplier. From the organization point of view, it is enough if we know that the system has a multiplier. One need not know how exactly this multiplier is designed or implemented. In the same way, if you take a set of instructions, which we will just call an instruction set, it is enough if the user knows what exactly the set of instructions is in this particular one. We need not really know how this instruction is implemented. Whereas how these things are implemented is in fact the emphasis in the case of architectural studies. I hope now you got the point.

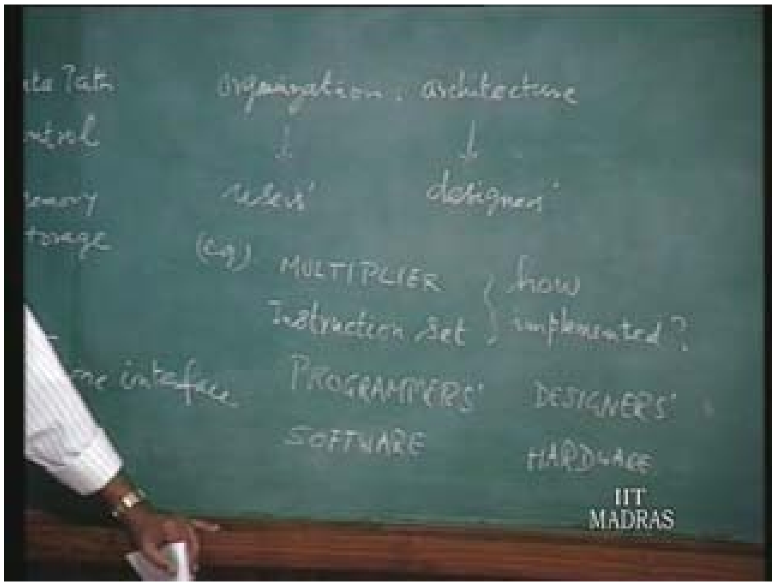
(Refer Slide Time 00:12:20)



How these are implemented is studied from the architectural point, that is, the designer point of view. For instance, everyone knows that a multiplier takes two numbers minimum, and then it gives out a product – that's good enough. How exactly it is implemented is in fact from the architectural point of view. So we may say that this is essentially when programmers are the users, whereas this particular one is for designers. So I can say that organization is a study of computer system from programmer's point of view, because the programmer must know whether a separate multiplication instruction is there or whether it has to be done through a set of instructions.

The architect of the system, that is, the designer must worry about how it is implemented, what is the algorithm of implementation, and depending on this, there can be different types of multiplier units. This man doesn't really bother about that. I may, without loss of generality, subsequently also say that essentially organization is a study of the system from software point of view, whereas architecture is a study of the system from hardware point of view, because it is being designed.

(Refer Slide Time 00:13:59)



Now you may ask what if you have someone whom you may call a software engineer – because he is also a designer he designs the software system, but essentially he does it from the user point of view. Similarly there can be different levels of designers; for instance, you can think of a system software designer or an application software designer. When we try to identify a programmer and designer as two different things, we must be very careful about what we are talking about.

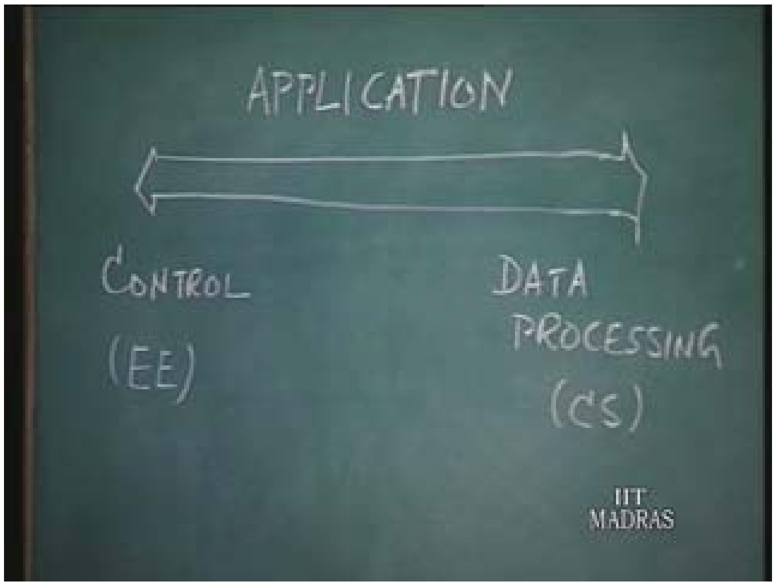
I hope you understand that essentially the study of the system organization is from user point of view, and architecture is from designer point of view. In other words, when we talk about architecture we would go more into the details of how the system is implemented, whereas here we will have a description, an overall description of the system. So from the organization point of view, we can summarize and say that we have a processor and memory. What are the types of processor and memory? We may not go into the details of processor or the memory. Having provided this particular introduction, I would now give further information – a slightly different point of view.
Because I said organization is essentially from user's point of view, the best thing is to start with what we may call as an application spectrum, because essentially any user is concerned with the application, that is, how and where a particular system is going to be used.

(Refer Slide Time 00:15:58)

APPLICATION
IIT
MADRAS

So what is this application spectrum? We may say broadly that at one end we have what we may call as control applications, which I would say a bit loosely as essentially an electrical engineer's view point. And at the other end, we can say again a little loosely as data processing end, and I must say that the focus of this particular thing is essentially a computer scientist's view point.

(Refer Slide Time 00:16:49)



I would say the same – you have electrical engineering, and, at the other end, you have computer science. Is it that the application spectrum gets really divided into these two ends rather mutually exclusively? No, it is not really so – there will be some applications for instance, which might need controlling in data processing, and data processing in control also. But we can broadly have these two classifications. Why do we have to really do that? Any control applications will be special purpose applications. You know precisely what the inputs for the system are, and you also know where exactly these outputs from the system will be used; whereas in the other end it is very general. So it's a general purpose. Here we really do not know what the purpose is; there will be a wide range of users, whereas here it will be a narrow range; even one specific user sometimes.

In the general purpose we really do not know; in other words, suppose we have software, since we know for what purpose this is being used, that software can be packed and put in a read only memory. I am talking about the program itself. The program can be put in the read only memory, whereas in this particular one, we do not know who the user is, so we must allow this particular user, whoever it may be – there will be different types of users – to load their program in RAM, that is, random access memory. That is, here we have a loadable memory. The program will be loaded by different users, here you have a program which is frozen and canned and put once for all, because we know what the purpose is. In other words, we may say this particular one is more hardware oriented.

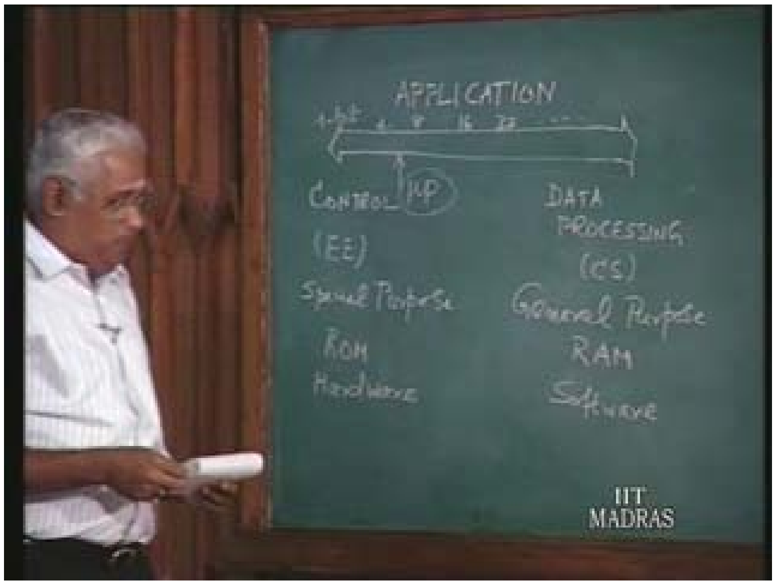
Even if you have a program, that is, you do have some software program but essentially it is hardware oriented, that program is used for some hardware application, whereas at this end we have this software application because we really do not know what the end use is. I hope you are able to see how we are able to distinguish between these two. Again in control applications, RAMs will be used, and in data processing applications, ROMs will be used. As I said earlier, they are not really mutually exclusive, but broadly we can classify these two.

So starting with application, this is how we have to really study a computing system, because depending on this particular one – whether they are data processing oriented, whether they are control oriented or in other way say whether it is electrical engineering oriented whether it is computer science oriented – all these issues will keep emerging and one will have to study them very closely.

And it is very interesting that these days almost everyone knows about computer; of course, they also know a microprocessor. Somewhere at this end of this entire spectrum, a microprocessor started. What exactly are its characteristics? There are many things we can talk about – suppose the width of the data that is being processed is, say, 1 bit at this end. The microprocessor's evolution started with 4 and then subsequently with 8 bit and then subsequently 16 and 32, 64, and so on, it keeps growing.

So I can really say that in the case of data processing, the data width is more; in the case of control application, generally, the data width is less and microprocessor started with 4 bit and evolved. Although microprocessor started with control application, essentially it might off lay almost all the processors, all the computer systems, especially desktop systems. They all use microprocessors, but they are all from data processing point of view. So this is a nice picture one should have in mind while talking about this.

(Refer Slide Time 00:21:16)

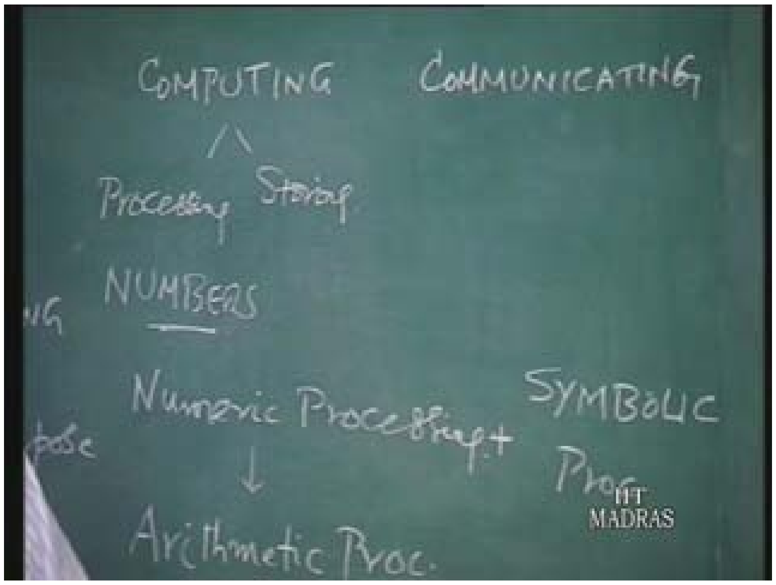


Apart from computing, there is another thing – that is communicating. I will touch upon this point a little later. Computing as I said essentially consists of processing and storing. We also said processing means essentially processing some data – that is, numbers. It all started with that, and of course, storing also was very much part of it. Later on, communicating came – what exactly is the history of communication? It started with line communication, say, telephones. Earlier, there was telegraphy and so on. Now we talk about data communications, in which computers also play a role. When we use computers for communication, again processing, storing are very much involved – it is not that it is entirely a different story altogether. I said computing started with the word compute – compute the numbers, then process, and then store. Now, more than processing, the present-day computers are used for storing and retrieval purposes, and that, in fact, is the starting point of information technology. It is not just computing – using numbers – it is storing some data and a set of data will convey some information to the actual users.

So from data, we progress further up to information; and when it comes to information, it is not really processing the data – it is storing the information and retrieving the information. So it is not only just processing but also storing; and so, in fact, the focus of issues has shifted from the CPU to memory or storage; that is, because of the advances in the semi-conductor technology. As I said earlier, when processors started growing, it was with control purpose; then they started growing towards data processing.

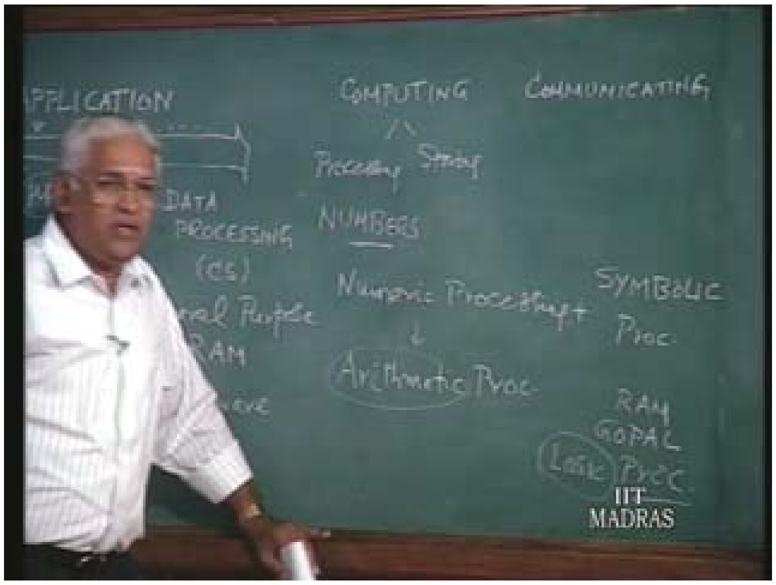
The advances were so much that simultaneously the technology advances in processing had also influenced the memory, which has been used for storing. So now, given that processors are quite powerful, it's no more a problem about storing – how much can be stored? Issues have gone in that particular direction. It all started with processing; that is, processing the numbers. So the main thing in processing is numeric processing: that is, what is the numeric that is processing the numbers? When we talk about numeric processing, essentially it is taking more than one number and then carrying out some arithmetic. So we may say any particular numeric processing will essentially involve arithmetic processing. Then subsequently, in addition to this particular numeric processing, we also have what is known as symbolic processing.

(Refer Slide Time 00:26:34)



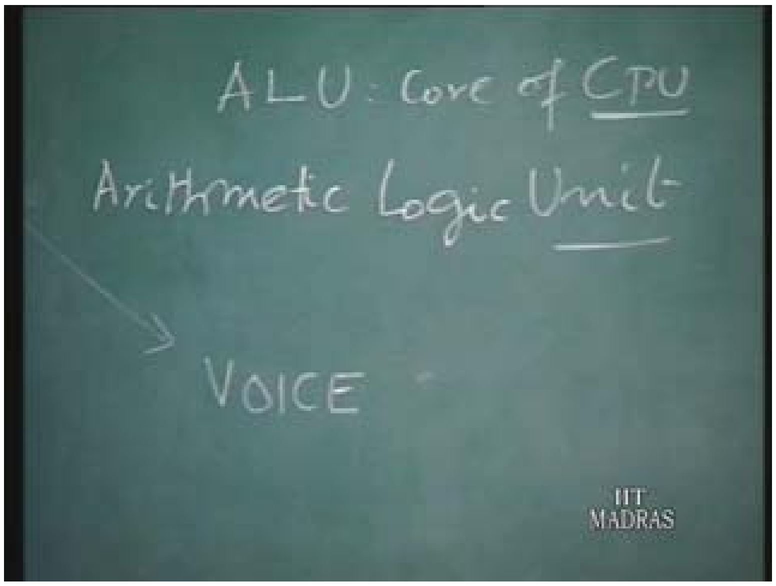
Now instead of numeric, now the processing is confined to some symbols. It’s not just numbers; actually the numbers represent some symbols. For instance, a name is a symbol – a string of characters, even a letter, a character a, b, c, d, whatever, a particular string such as Ram – that is one symbol. Anything else, for instance, another name Gopal – this is a symbol; a name; an address; these are all symbols. This particular symbolic processing needs what we may call as logic processes, mainly because this kind of processing does not involve numbers but one has to carry out some logic processing on these symbols. So essentially it is the arithmetic processing and the subsequent logic processing.

(Refer Slide Time 00:28:50)



These two have come together and the core of any specific CPU or processing is ALU. What do A and L stand for? A is for arithmetic; L for logic and U is for the unit. ALU is the core of CPU; that is, the essential thing of any processing unit. You can see that in a CPU, we are just seeing some detail of CPU essentially, that is the ALU, arithmetic and logic unit. It had come about because it all started with numeric processing; that is, arithmetic processing; and subsequently it evolved in to symbolic processing; that is, logic processing; and now essentially we have this particular ALU unit, which is the essential core of the processing unit. Let us not forget about the other aspect, which has come about, that is, communicating. I said earlier that communicating essentially started with telephone and so on; so we may say that in the case of telephone, voice communication is the main thing.

(Refer Slide Time 00:29:17)


ALU: Core of CPU

Arithmetic Logic Unit

VOICE

Now, after voice communication was introduced, over the same wires, even data, which were essentially used for processing computers, was being transmitted over the same set of copper wires and so on. Now, because of the advances in digitization, the same voice can be converted into some coded data and the coded data can also be used in this particular case. So you can see that earlier there was computing, and then the arithmetic logic processing, etc.

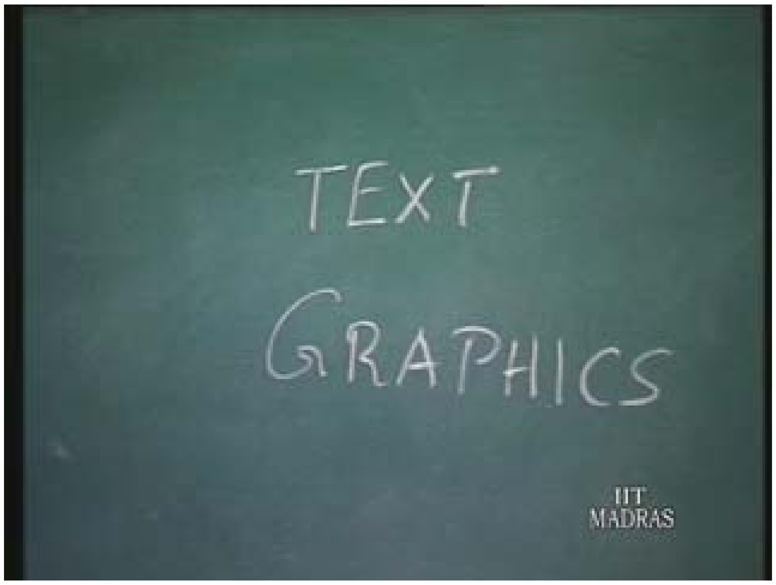
Now we are seeing that because computers have become cheaper, processors have become cheaper, and more and more memories are being used. Also, not just one computer but a set of computers are connected over a set of wires as in telephone-to-telephone communication. Similarly, we now have computer-to-computer communication, but what we are using is, say, if you are trying to use the same set of words which we use for telephone channels, essentially earlier with telephone equipments we were sending out the voice, but using the same, now data which computers are familiar with, can be used. But voice also can be sent over the same thing using coded data.

This is how things have been advancing. From computers, we have now come to network of computers, and these networks are in fact supported by the interconnecting communication lines. What has happened is that originally we started with numbers; and then when symbols came, you can see that some symbols such as these could be used – what exactly are these? These are nothing but what I may call as text, a part of text that you can see is a word.

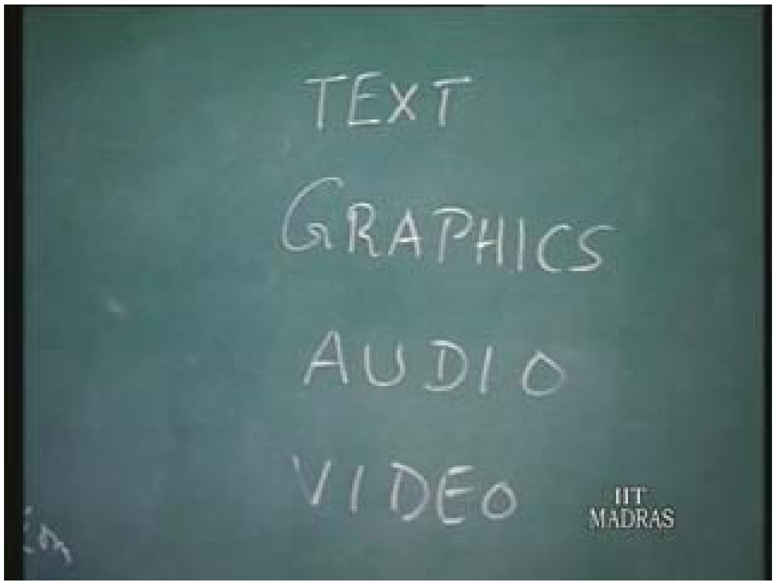
(Refer Slide Time 00:31:47)

So I can say that each one is a part of text. First it was numeric or number processing, then it had come out as text processing; after text, simple picture elements came about. So using these picture elements, you can depict good graphics capability. See how things have progressed – from numbers to words or part of text and after text some primitive elements of graphics – meanwhile, because communication also has become very much part of this, now we can add voice or, in other words, audio.

(Refer Slide Time 00:32:29)


TEXT
GRAPHICS
IIT MADRAS

(Refer Slide Time 00:33:12)


TEXT

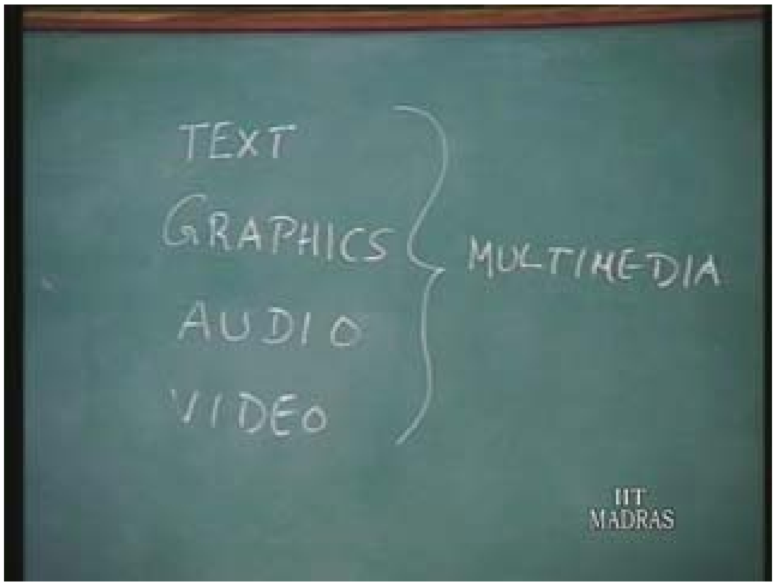
GRAPHICS

AUDIO

VIDEO

When talk about audio, the next thing, obviously, is video. Today's computers can support text, graphics, audio, and video. That is why we say the information or the data is coming in multiple forms and it can come in the form of text or graphics or audio or video – we talk about different types of media – and that is precisely what a multimedia computer is.

(Refer Slide Time 00:33:41)


TEXT

GRAPHICS

AUDIO

VIDEO

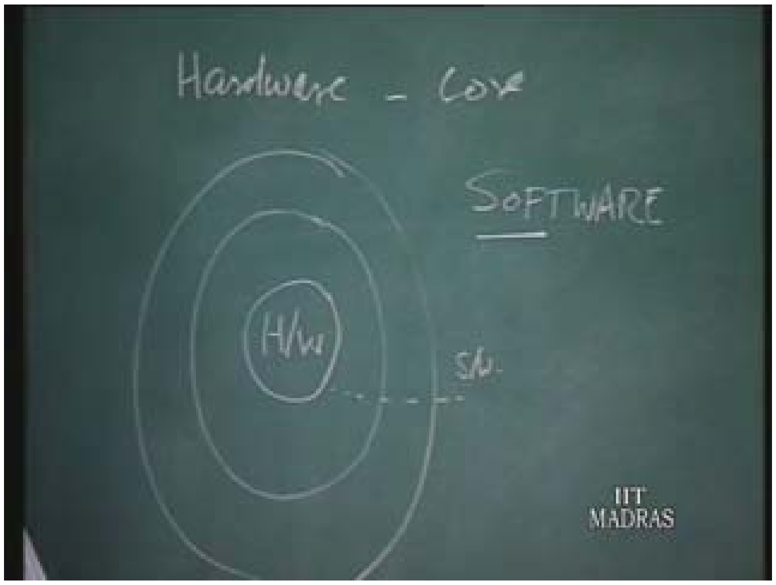
MULTIMEDIA

So from simple numeric processing, which is historical, today we have advanced to multimedia. This has been supported mainly because of these advances in semi-conductor technology. It first started with processors; subsequently memory, and later on, we will talk about the I/O part. Today, we know that the data can come in the form of these, which means the information is available to the user over different media. Compared with text or graphics, there are storage issues related to audio and video, mainly because essentially audio and video are all signals. These signals must be coded, and also when coded, the number of codes which have to be transmitted for reproducing this audio for replaying the video etc., take a lot of space. So compared with these, the storage issues have come up, and again, communication related issues are very much there. So storage issues and communication issues are there, whereas in the case of text, suppose we talk about some kind of a set of data in a data base – say employee's record, his name, address, the category of employee, etc. – all these things can be easily created. This is known as records in the text. You cannot do the same with audio and video, because these are generally called unstructured data.

You can bring in a structure for these, but in the case of audio and video, there is no particular structure. For instance given a particular audio, which generally is called clip, there is a specific duration, say 1 minute or half a minute, or whatever the duration. So if you want to describe how it is, a man can do that because he has very good information processing capability; but can a machine does it?

These are the issues that are generally called semantic issues. In the case of computer organization, we have a multimedia; we have to deal with all these issues also; it all depends on the type of system one is going to use. Now I will take up the next one – what is next? The next thing is in fact, going back to the beginning, organization is essentially from user point of view. But let us not forget that architecture is very much part of the whole show; in the sense that the hardware is the core of the entire system. This is the core, and when we talk about organization around this core, we will be talking about different layers and layers of software. All these things will be software. How did the very term software come in? That is interesting – we will take a look at that first, because hardware is really hard for every person or every user to understand.

(Refer Slide Time 00:37:30)



Given this hardware, you are going to soften this particular system by providing certain features or layers, so that one need not really be concerned with the hard aspects of this system. One need only be familiar with certain soft aspects of the system. Now, historically, things had grown in such a way that more and more layers of software had come about, mainly, again, because of the advances in technology. Initially it all started like this: a person using a computer must be familiar with the inner details of the processor or the system. In other words, there was not much distinction between organization and architecture. That is, a person dealing with hardware must know what we call as some machine details and if he is going to do the programming, he must know the codes related to that particular machine.

In other words, the program will consist of – you remember historically it consisted of all numbers – a string of numbers. All these numbers in this computer system are binary numbers; digital computers we have binary numbers. One talks about some kind of string of digits like this: that is, bit 0 or bit 1, and so on. This is possibly a machine code conveying some information; this part of the code will convey something and this part of the code will convey something else. Now, let us go back and recall – we said numbers are used for carrying out some arithmetic processing. So it is meaningful to understand it – this part let us say is the operand or the data; and this part refers to the operation. What is that operation? Generally it is an arithmetic operation. So this is a code related to operation, and this is either a code or that itself is operand; it all depends on how we talk about it. In other words, we have the data here: a data path and we have a code, which relates to operation and so generally this is called an opcode.
Now imagine a particular programmer must know these details at these very low levels, because it is at this low level that a particular active device, say, 0 means it is not conducting, 1 means it is conducting. That is, at the physical level of the machine, one must know what is going on. So this is very difficult to remember but that is hardware. So to soften this, one introduces other levels of codes. One goes out from machine code

to lesser and lesser or say lesser and lesser details; that is, we are trying to hide the hard information; meaning, instead of doing it this way, suppose this particular opcode corresponds to add and then this particular data some number; let us just say, for instance, A. If you are more familiar, let us say this particular thing refers to say A B. Now, instead of this code, the user can easily remember this, which means add A B. Possibly A and B will be added and the result will be the sum of this – that's what it is. So now, instead of this code, which is the machine code, if one can use this code, what is this code? What is specialty about this code?

This particular thing actually tells the user to remember. In other words, these are generally called mnemonic codes – mnemonic means aiding the memory. Mnemonic codes are aiding the opcodes – memory of the user, not the memory of the system. Mnemonic aiding the memory of the user so that the user can remember this better than this, which means, we have gone one layer away from the hard. Instead of a string of 0110, we have to have some layer, which helps the software developer or the user to remember, and when he comes out he doesn't have to remember 0110.

It is enough if we know it is add; and ADD for addition, say SUB for subtraction; and DIV for division and so on and so forth. It is easy to remember; one need not try very hard to remember these. Now, from machine code, we are going to the next level – these codes are still at a lower level. Why? That is because in each pattern like this, ADD A, B or whatever it is, we are only specifying one operation. At the next higher level, suppose we have an equation. We have gone from this to this or we will go to a higher level.

(Refer Slide Time 00:44:04)

There, suppose you can write something like this: X is equal to Y plus Z. Now, what is that we have here? We have, in fact, a function. So at this level, we are no more talking about just a code; we talking about a statement. It basically means ADD Y and Z and stores it as X. In fact, this is something more than this. So you can see from machine code

– I am just writing M by C for machine – we have gone to the next level, that is, mnemonic codes. But then, these mnemonic codes will be specific to a machine and generally that particular mnemonic code will be called an assembler code.

Now from the assembler code, we go to a higher level: generally it is called a high level language code. More than code, we may just call it a statement; a statement in a given language. So we go from the harder aspects to softer aspects in the sense that at this level, we can talk about different languages, for instance, FORTRAN is one. It is very famous and a very old one; PASCAL is another; BASIC is another; and so on. These are all high level languages. The user is going to write his program in this high level language. It is much better than doing a machine code level, because in the machine code level the user must know the inner details of the processor or the system and the moment we talk about inner details, it's going to vary from processor to processor, from system to system, because the CPU memory interaction can be different for different things. We also said that from architectural point of view, why should one be really bothered about the details of a car if one knows certain minimum things about car driving? Of course, if the car breaks down, he should know – its different matter altogether that you can always call a car mechanic and have this particular thing fixed.

So one need not know more details – one need know only the functional details of the machine he is going to use. We have considered assembler code in case of computers, and then from assembler code we have gone on to high level code or statement in a language. If one writes a program in this particular one, then one should have the appropriate code, another set of codes to translate from here to this – if one writes from here, similarly one must have another software to translate from this to this because, ultimately in any system, these are the codes which will be accepted by the hardware. The hardware knows only this. It is only the user who is trying to be away from this mainly because he is aided by the appropriate software, he is writing these codes. But inside, internally, one needs the machine code. So to write a program, one uses a high level language, but finally when the program is run, it is all a string of 0s and 1s. We are only saying that the user is hidden or the details of hardware are hidden from the user. So we need the necessary software or program which can translate from this to this, this to this, and this is how in fact the software has being going. That is precisely what it is.

So we may just conclude for today saying that we need to have the necessary program called translators, basically, to translate from one code to another code. Tomorrow we will continue and then see that the translators can be of different types, depending on the levels and so on. And this is how the software has been going on; that is, at this level it is a machine code, at another level it is an assembler code, at the third level it is a high level language code. But to see that the user may write this program for this he needs to have appropriate programs, which can translate this. That's what the translator is all about.