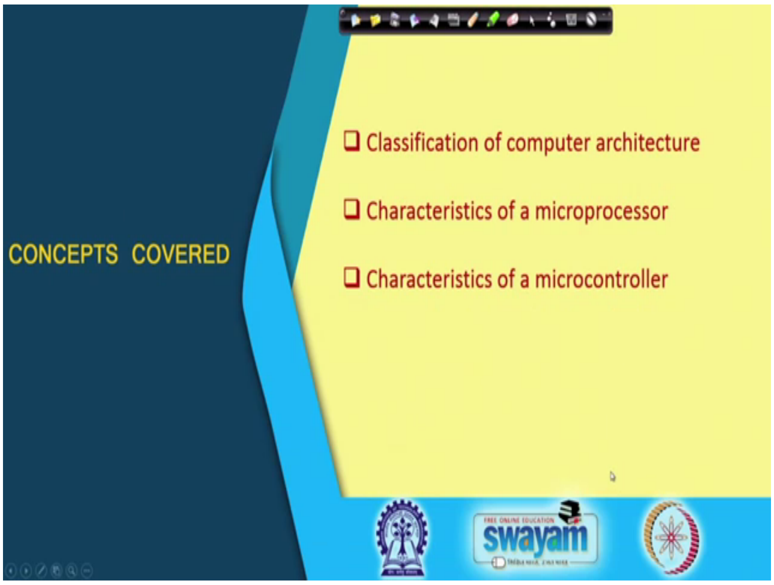
Embedded System Design with Arm
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 3
Microprocessors and Microcontrollers

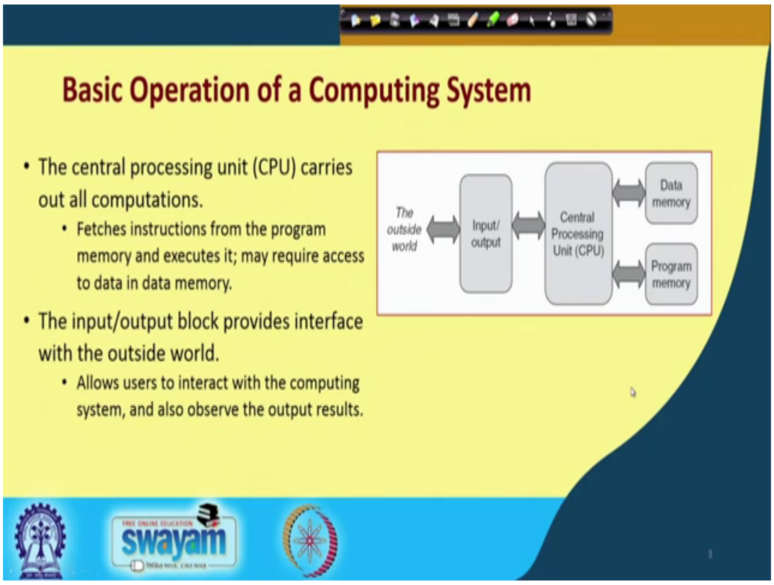
In this lecture we shall be talking about some basic concepts of microprocessors microcomputers and microcontrollers because, these are the brain or the processing power behind any embedded system that we see around us. So, it is always good to know what are the main differences between these things. So, the topic of the lectures you can see is Microprocessors and Microcontrollers.

(Refer Slide Time: 00:49)



So, these are the concepts that will be covered as part of this lecture. We shall be talking about some classification of computer architectures followed by the main characteristic features of microprocessors and microcontrollers ok.

(Refer Slide Time: 01:06)

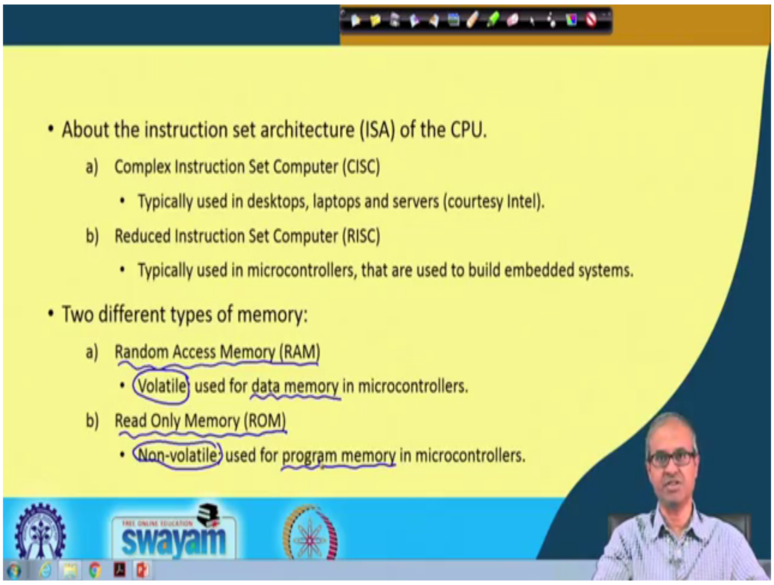


So, let us start with how a computer system works basically. Now, with respect to a computer system, so you may be knowing that there is something called Central Processing Unit or CPU that carries out all computations or calculations. So, you can see this schematic diagram, the CPU is sitting in the middle and there are some memory where program and the data are stored. So, what is CPU does? It fetches instructions from the program memory one by one, it fetches the instructions execute them and during execution it may require to transfer some data between the data memory, either reading a data or writing the data back into the data memory ok.

Now in addition to processing, the computer system often needs to interface or communicate with the so called outside world. The outside world is typically the user who is using a computer system for conventional systems like a desktop or a laptop right. So, the input output subsystem this allows users to interact with the computing system. Now, I have already mentioned when you talk about embedded systems here the way the processor interacts with the outside world.

The outside world may not always be a human being may not be a user, it may be environment, it senses some temperature pressure humidity some parameters of the environment, those will be the inputs. And similarly it can take some corrective action like it can turn on a heater turn on the compressor and so on these are the output devices ok.

(Refer Slide Time: 03:08)



Talking about the instruction set architectures this is one way in which you can classify computer systems. What kind of instructions the CPU is capable of executing depending upon the broad classification with respect to instruction sets we can classify computers as either a CISC architecture or a RISC architecture. CISC is the short form for Complex Instruction Set Computer. Now, the most common example of such CISC architecture are the Intel classes of processors which dominate the desktop laptop and server markets.

More than ninety percent of the processor inside are made by Intel or some clones of those made by Intel. These are called complex instruction set computers, the instruction can be fairly complex with lot of flexibilities and features.

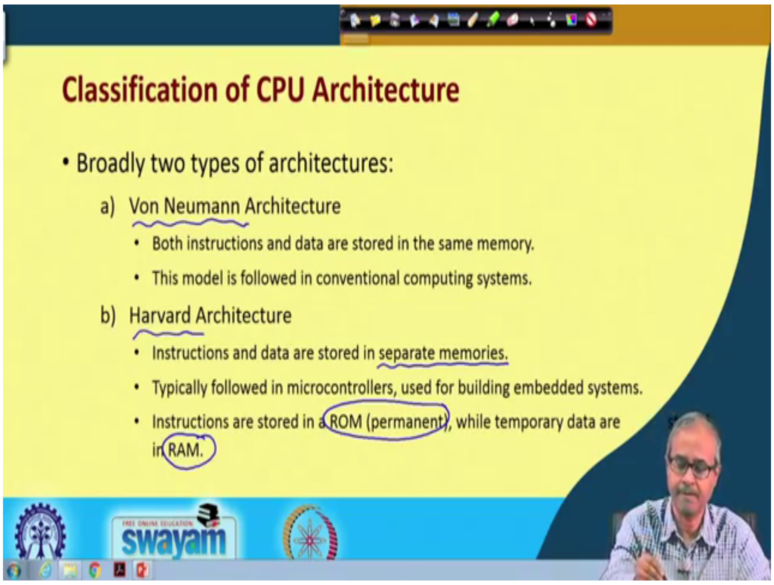
There is another philosophy called reduced instruction set computers. Here the instruction set is made very simple, because they are simple it is much easier to implement the computer system in hardware. So, in some sense it becomes more efficient with respect to execution of the instructions. So, most of the microcontrollers that we see today they are based on the RISC architecture because, of primarily this reason, they are easy to implement they are very simple ok. Now, not only microcontrollers, let me also tell you most of the modern processors at some level they are based on the risk philosophy of instruction execution because it makes the instruction execution unit much more efficient ok.

So, even the Intel processors I am talking about those are based on CISC internally this complex instructions are broken up into simpler instructions or micro instructions you can call, they look more like a RISC instruction set, which are then executed efficiently using a controller ok. Talking about the memory system broadly speaking there can be two kinds of memory; one which is called random access memory where you can both read and write and the other is read only memory when you store something permanently you can only read from their ok.

Random access memories are typically volatile, volatile means they retain the values which you are storing only during the time the power is switched on, as soon as you switch off the power the data gets lost. And these are used well here we are more interested in microcontrollers because they form the heart of embedded systems. So, for microcontroller this RAM’s are typically used to store data as the data memory, but when you talking about program memory the place where you are storing a program, this memory is typically non volatile; which means because it is a wrong read only memory, once you store it even if you switch off the power the program will not get destroyed.

So, in most microcontrollers there is an area of program memory which is implemented using ROM. But nowadays there are microcontrollers which use some alternate technologies let us say like flash memory, which is also non volatile where you can store some program, you could also change the program if you want, but if you switch off the power the program does not get destroyed fine.

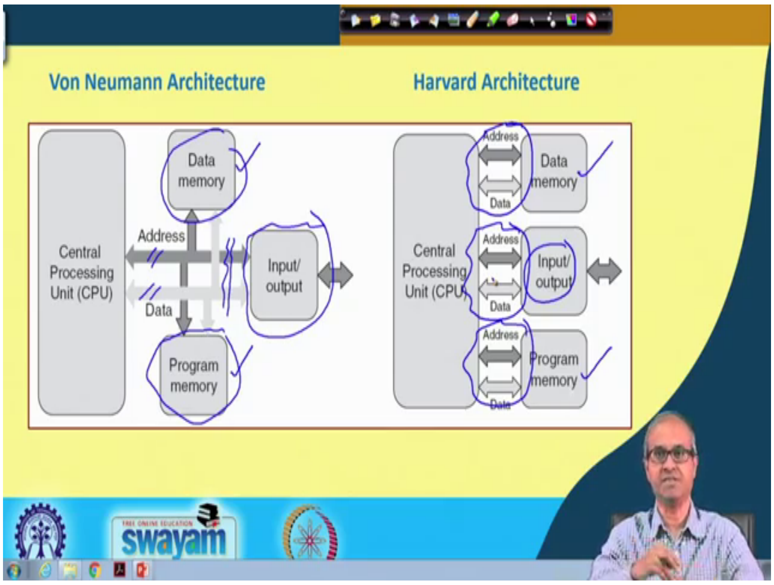
(Refer Slide Time: 07:28)



Broadly with respect to CPU architectures we are so far talking about instruction set architectures, now talking about how the CPU works there is a broad classification here also we talk about Von Neumann and Harvard class of Architectures. The basic difference is that in the Von Neumann Architecture we have a single memory both instructions and data they are stored in the same memory, most of the conventional computer systems that you see around us they are based on Von Neumann Architecture ok.

But in Harvard Architecture here conceptually there are 2 separate memories; in one of the memory you store the program, in another memory you store the data. Most of the microcontrollers they follow this principle they have separate memories right. So, instructions are typically stored as I said in a read only memory while data are stored in a random access memory, there are 2 different types of memories.

(Refer Slide Time: 08:47)



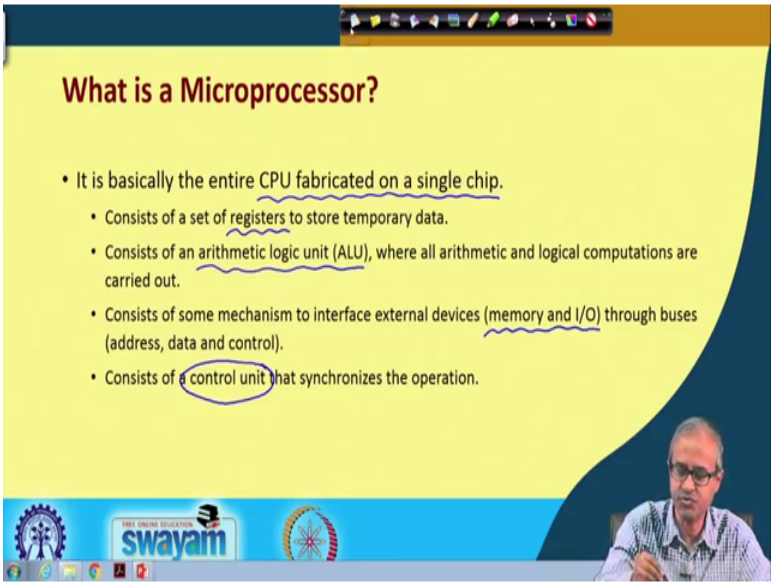
Pictorially for Von Neumann Harvard Architectures can be shown like this. In a Von Neumann Architecture as you can see here the data and program memory although you are showing them as separate, but CPU is accessing them over the same data bus using the same address.

So, with respect to CPU it appears that the memory is the same, but in practice we may store them separately in separate parts as this diagram shows program is here, data is here but with respect to the CPU the way the addresses are generated they are unified, same address and data lines are used to transfer program as well as data. And input output devices are also typically connected through the same bus. This is how typical Von Neumann Architectures look like. Of course, there are sophisticated architectures where there can be multiple buses for parallel transfer of data and so on.

But I am talking about the conventional concept ok. Now, in Harvard Architecture here you see there is separate data and program memories and address and data buses are entirely separate. So, while you are transferring program you are fetching a program in parallel you can also fetch or write some data into data memory. And in some architectures the input output devices are also connected via separate address and data buses So, this is the general form of Harvard Architectures.

the drawback is that you need so many buses, lot of input output pins to interface the external devices but the advantage is that you get on the average faster and parallel data transfer features.

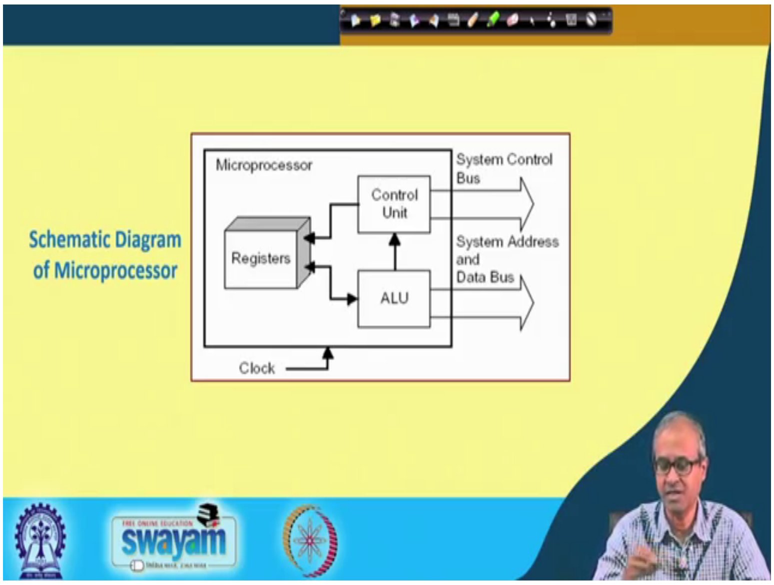
(Refer Slide Time: 10:56)



Let us now come to a microprocessor. What is a microprocessor? Well we have seen what is a CPU? Microprocessor is a general term which means a CPU fabricated within a single integrated circuit or IC chip. With the advancement in semiconductor technology earlier CPU circuits are very large and bulky they have started to become smaller and smaller and now they can fit inside a single chip. Nowadays for example, if we look at the Pentium CPU chip there are close to billions of basic components or transistors inside the chip, they are huge in size.

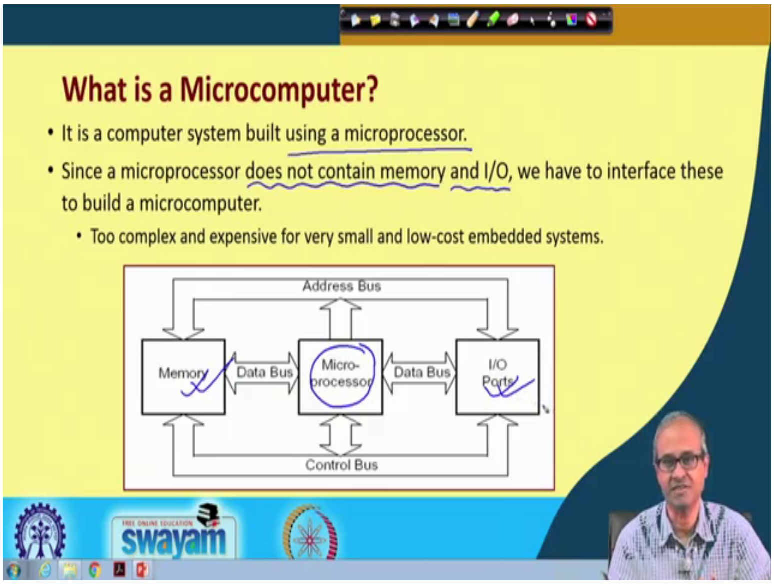
So, inside the microprocessor what are the things that are typically there; there are a set of registers, some are general purpose some are special purpose which are used for temporary storage of data. There is an ethic and logic unit where all the data processing are carried out both arithmetic operations and also logical operations and there is some mechanism for interfacing memorial IO devices. There are some external bus and control mechanism through which you can connect memory or IO devices with the microprocessor right. And of course, there is a control unit which can be considered as the brain of the system, which controls the entire operation of the microprocessor, it synchronizes all internal operations ok. This is what a microprocessor basically is.

(Refer Slide Time: 12:44)



So, schematically you can see these are the main components, you can see the registers here, you can see the arithmetic and logic unit, you could see the control unit. And externally there are some buses address and data bus and there can be control bus, address and data buses will be used to interface with external memory and IO devices. And the control bus will contain other signals like read, write, enable different kinds of interrupts and other things ok, some signals for interfacing slow memories, there are many such miscellaneous signals which are used for control purposes there will be part of the control bus ok.

(Refer Slide Time: 13:30)

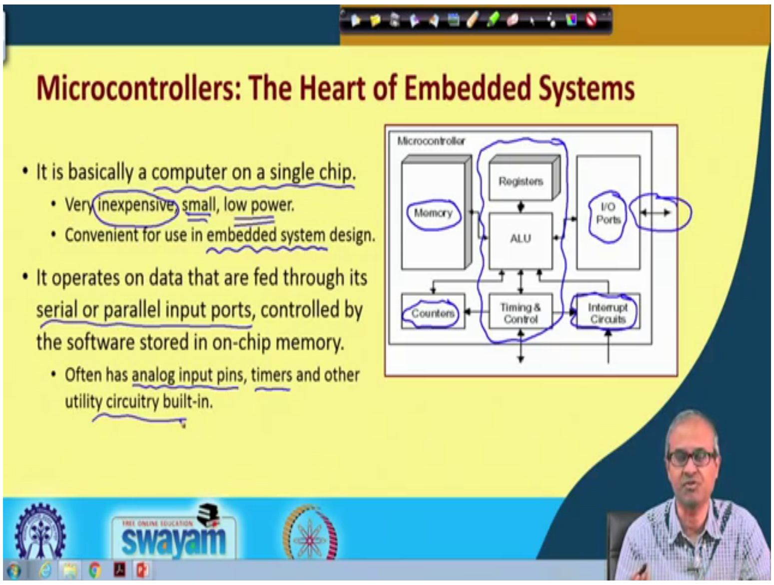


Now, starting from microprocessor let us now say next step what is a microcomputer. Microcomputer is a computer which is built around a microprocessor. You look at our desktop, you look at our laptop, there is a Pentium we talk about Intel I 3, I 5, I 7 class of processors inside our modern desktops and laptops. They are essentially micro processors and inside a desktop or a laptop it is not only the micro processors, there are memories, there are other devices, there is a disk drive there are some other interfaces to connect keyboard and mouse for example, printers that makes your entire computer. This is a micro computer what we call, it is a computer system built using a microprocessor ok.

Now, micro processor just said it contains basically some registers and arithmetic logic circuits and control it does not contain any memory or any facility for input output. So, to make a real computer system we have to interface all of these devices with the microprocessor right. But, the trouble with the micro computer is that since you have to interface so many things around a micro processor, the system becomes complex, bulky and also expensive. In addition it also consumes significantly higher power. So, it is fine if you use it for a high performance application like a desktop or a laptop, but not for embedded system application were ultra low power portability small size these features are quite essential.

So, a micro computer as you can see, this is built around a microprocessor, you have memory, you are IO ports and through this IO ports you can interface with several IO devices. This is what a microcomputer is.

(Refer Slide Time: 15:50)



Now, let us come to micro controllers which are the heart of embedded systems. What is a micro controller? Well microcomputer we have said, it is a computer system built around a micro processor. Now, what if the whole of the micro computer we can shrink and put inside a single chip, all the essential elements of a computer system I put inside a single chip. Of course, since I am putting everything inside one chip, I cannot have equivalent functionalities as compared to what I have in a desktop or laptop. For example, I cannot have four gigabytes of memory example ok.

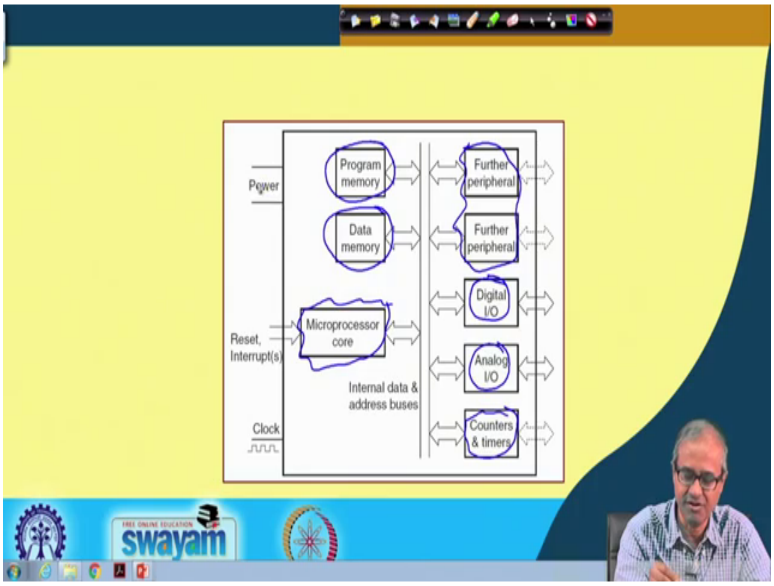
Since I am putting everything on a single chip, the total chip area has to be shared by processor memory or everything right. So, it is basically microcontroller a computer on a single chip because it is on a single chip it is relatively very low cost inexpensive. It is small it also consumes quite low power, very low power.

In fact and these are very widely used in embedded system design right. Now, if you look at this schematic of this micro controller what are the typical things which are present you will see that, well you have this CPU inside, here is your CPU registers ELU's and control. This constitutes a central processing unit. There is some memory of course, you cannot have very large memory some memory is there small in size. There can be some facility for connecting input output devices. There are some timers and counters which are required for many applications, there are interrupt circuits.

And there are many other facilities there can be some analog ports. Most of the signals that are coming from the outside world they are continuous or analog in nature they are not digital. So, if we have some kind of analog to digital converter built into the system into the chip it becomes very convenient to interface such devices. Typically all those facilities are provided inside the chip that ok. So, as you can see, microcontrollers typically operate on data that are fed through some input ports data coming from outside these are the ports and there is a program which is running stored in memory whatever calculation is going on, that is controlled by that program ok.

And as I have said typically they have analog pins, timers, counters and other utility circuitry. For example, you can have a pulse width modulation circuit which is very useful as we shall see later.

(Refer Slide Time: 18:59)



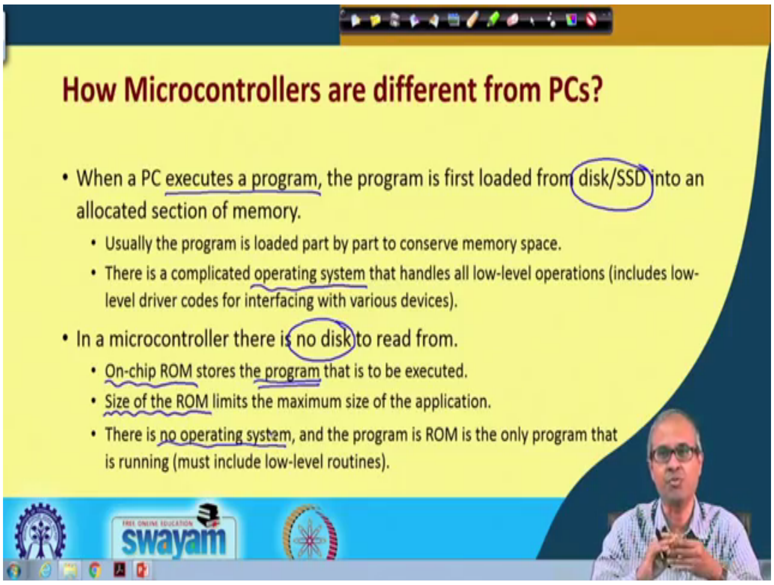
So, this is a slightly more detailed diagram of a microcontroller. So, you can see you have the basic microprocessor here, register ALU and the control, there is some program memory, a memory to store the program, there is a memory to store your data ram and there can be some input output pins. As you can see some of the input output can be digital, some may be analog there will be counters and timers and there can be sophisticated kind of interface also; like serial IO interfaces, pulse width modulated interfaces interrupt during IO and so on and so forth. So, all these facilities are provided in typical microcontrollers today, so that you can use it for almost any kind of embedded system applications ok. In addition you have the power supply reset interrupts clocks to the standard signals.

(Refer Slide Time: 20:07)

Microcontroller packaging and appearance you can see these are some typical microcontroller which are manufactured by a company called PIC, PIC microcontrollers. Now, in comparison you see one of the older micro processors Motorola 68000. This was a 48 pin chip, so large in size. So, in comparison the smallest PIC processor is so small, it is a fraction of the size of a coin as you can see.

So, this becomes very small so that in very small embedded system applications you can use these kind of processors very conveniently. This is the main advantage s talking about.

(Refer Slide Time: 20:53)



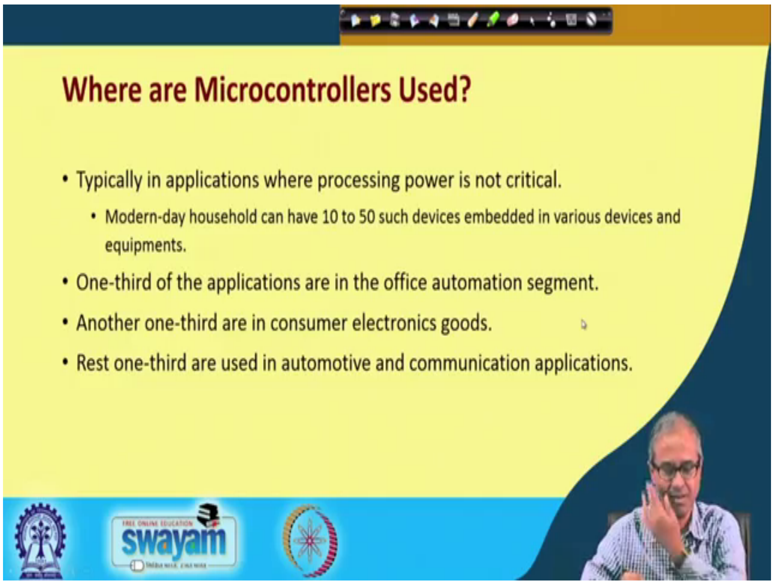
Now, microcontrollers are computers in their own right, PC's are also computers built around a microprocessor. So, what are the main differences, there are still some differences you see when a PC executes the program, from where does the problem come from? They are initially stored either in the hard disk or in the modern day systems we have something called solid state drive based on flash memory technology.

So, you can have hard disk or SSD, from there the program gets loaded into memory and from memory the instructions for executed one by one. This is how a typical or conventional computer system work right and there is an operating system you know of windows you know of Linux, they are fairly complicated program. They are responsible for handling all low level operations while the program is getting executed whenever there is an input output operation it is the operating system, the drivers therein who will be invoked or called it will take care of the IO operations. ok. But in a microcontroller it is a very small system.

So, you cannot afford to have a disk, there is no disk everything will be inside that microcontroller itself program whatever program is running. See in a computer system it is general you can load any program you want from the disk and run it but a microcontroller that is sitting inside an AC machine that will only run that program which is meant for controlling the ac machine right so, it is like that. So, in a typical microcontroller there is a small ROM inside the chip, this will be storing the program, fixed program, the program cannot be changed right and of course, the point to notice that the maximum size of the ROM that is provided there that limits the size and complexity of the program that you can run on the microcontroller ok.

And there is no operating system, whenever you reset or power on your system the program which is stored in the ROM that starts running, that is the only program there is no operating system. So, these are broadly the main differences between a microcontroller and a normal computing system like the PC.

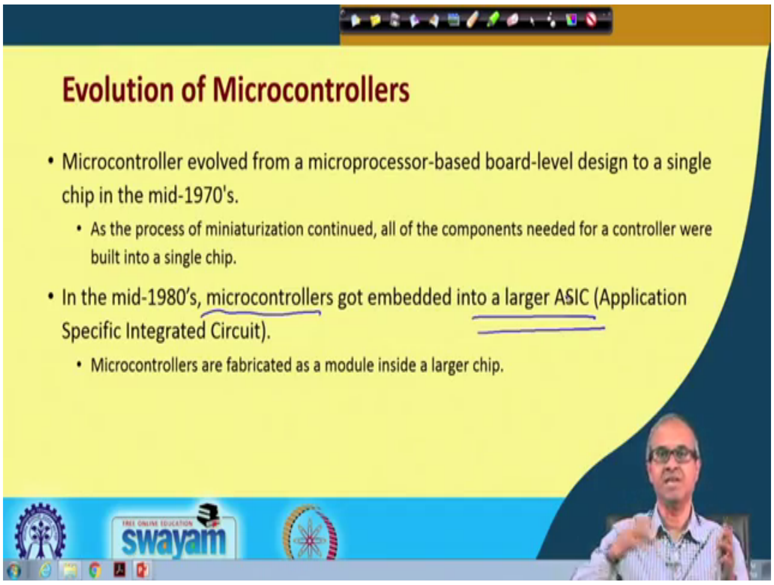
(Refer Slide Time: 23:35)



Microcontrollers are used in applications where processing power is not critical you do not need very high processing power rather you need low power, small size, low cost, these kind of attributes right. So, you typically find many such devices in modern day households, whatever equipment you purchase you deploy and install there will be some embedded micro controller inside it right.

So, there are many applications I already talked about this earlier in office automation segment consumer electronics, in automotive, in vehicles, in airplanes, in rockets missiles, everywhere communication you will find these devices everywhere.

(Refer Slide Time: 24:26)

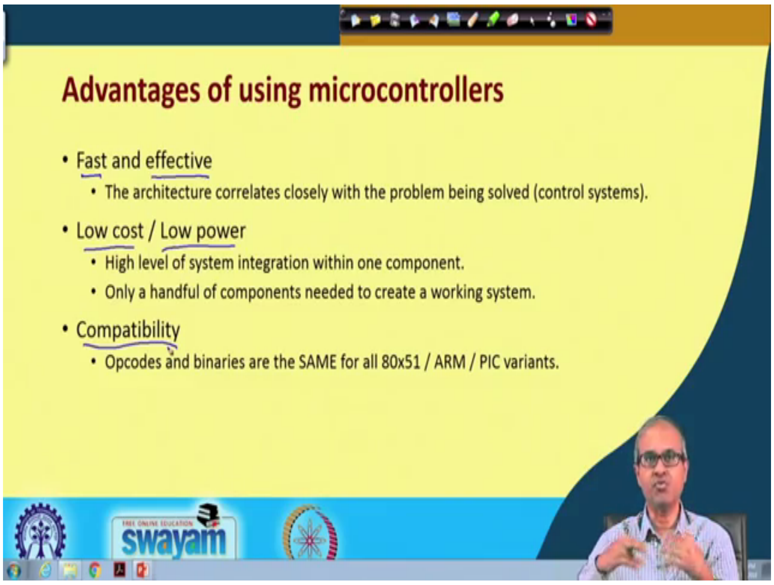


Now, talking about evolution since the 1970's the 1st generation of microcontroller started to evolve. There was one microcontroller which was developed by Intel, it was 8051, it is still being used, but talking about the power the flexibility and the innovativeness, when 8051 is rather a old kind of an architecture, it does not have too much flexibility right.

Starting from mid 80's and the process is continuing what we see now is that microcontrollers are all right, now they are getting embedded inside larger chips. These are called application specific integrated circuit. Like one of the microcontroller that we shall be demonstrating as part of this course, this microcontroller board which is from ST microelectronics STM, it has something called ARM Cortex M4 chip inside, now ARM Cortex M4 is not only the arm processor, not only a microcontroller but lot of other things all integrated in the same chip.

So, now you are going one step further, not only a microcontroller, but many other things you are putting or integrating inside the same chip, that is the present day trend.

(Refer Slide Time: 26:03)



So, to summarize the advantages of microcontrollers are they are fast, they are effective from the point of view of solving some particular application at hand; you are talking about the AC machine, the microcontroller inside is powerful enough to do the necessary processing there ok. It must be low cost because if it is of a higher cost it also increases the cost of the product, it must consume low power, so you know your electric bill should not take a hit right.

And compatibility; compatibility is very important. Typically you will find that this microcontrollers coming from certain companies or certain you can say did these are family of microcontrollers, PIC ARM these are some typical examples. Now, if you look at the microcontrollers across the family they are all instruction set compatible most of the instructions are compatible.

So, that once you develop a software for one member of the family when you move to a better processor, the same code can run or execute without change or with very little modification right. This is called compatibility it is very important with respect to maintainability. So, with this we come to the end of this lecture. In the next lecture from the next lecture we shall be starting some discussion on how we can make processing faster with particular regard to the ARM family of microcontrollers. We shall specifically see what are the features that are inside the arm family of microcontrollers and why they are different, why are they standing out, why people are using them so much so widely. So, this we shall be discussing in our subsequent lectures.

Thank you.