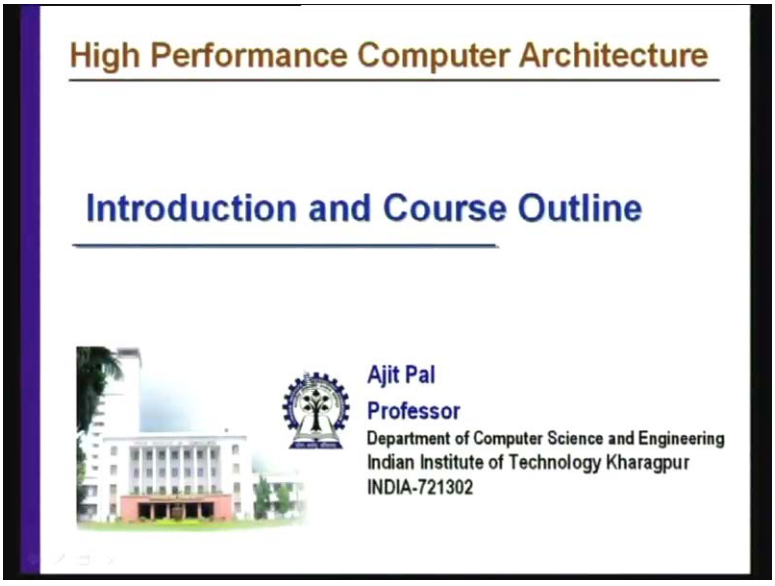
High Performance Computer Architecture
Prof. Ajit Pal
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

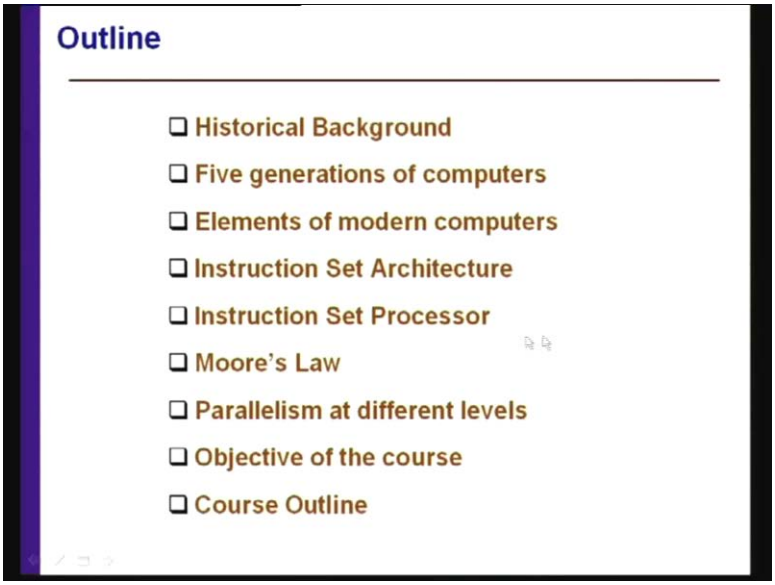
Lecture - 1
Introduction and Course Outline

(Refer Slide Time: 00:57)



Dear viewers, I welcome you to the lecture series on high performance computer architecture. In 40 lectures, I shall try to provide an over view of computer architecture at different levels and various aspects of advanced computer architecture. And today we shall start with the first lecture and the title of today's lecture is introduction and course outline.

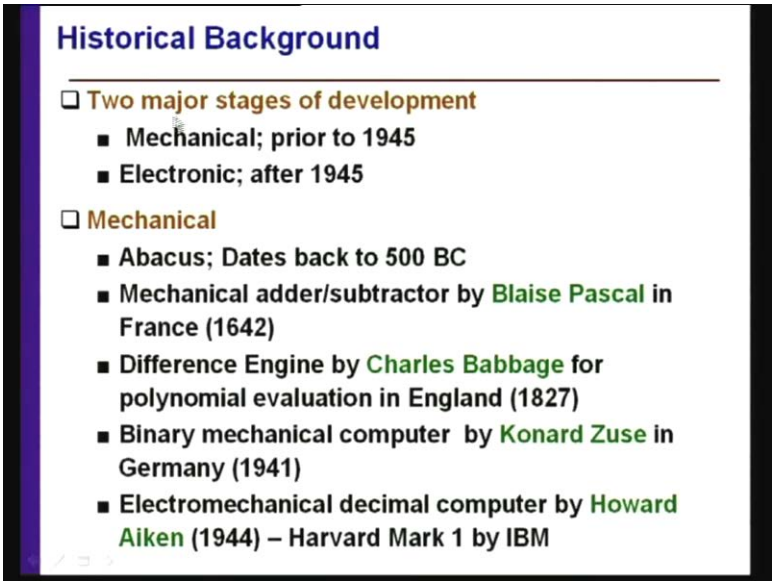
(Refer Slide Time: 01:10)



So, in this lecture I shall try to give an introduction of the course and given outline of the different topics that I shall cover in this course and here is the outline of today's lecture. First I shall give a historical background because, today we find a, an array of computers or powerful array of computers surrounding us, performing different purposes and providing the need of our daily life. And this does, this did not happen in a single day and it has taken many years, may be 100 years to reach this stage and how gradually we have arrived at this stage, I shall give a kind of evolution that has taken place to reach this particular stage. And in this respect I shall discuss about five generations of computers and then, I shall talk about the elements of modern computers, what are the different components that are, that builds a modern computer.

Then, I shall introduce to you the instruction set, architecture which is essentially a programmer's view of the processor and then instruction set processor which essentially represents the way, the processor is realized. Then, I shall talk about some related topics like Moore's law which has helped in the, in the progress of the computers and the Moore's law is the, is the force behind this gradual evolution of computers. And then I shall discuss about parallelism at different levels, we shall see to reach higher performance parallelism is the, parallelism is the key idea and how parallelism have been incorporated at different levels, that I shall discuss. And finally at the end, I shall give an objective of the course and an outline of the course.

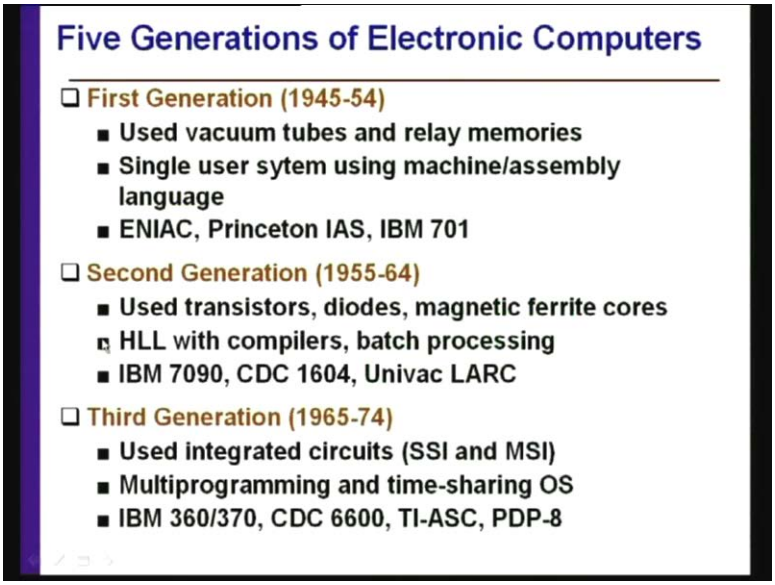
(Refer Slide Time: 03:19)



If you look at the, the history of computers as I mentioned the history encompasses more than 100 years. So, computing equipments are available may be for 100's of years and the, the, the period can be divided into two categories. First one is the mechanical era, that means prior 1945 and then after 1945, we can say that electronic era. So, in the mechanical era the computers were built by using mechanical components like, I mean ((Refer time: 04:03)) and other things. For example abacus, which was invent developed in china that dates back to 500 BC, that was use for the purpose of I mean calculation of numbers.

Then mechanical adder, subtractor machine that was built by Blaise Pascal, Blaise Pascal in France in 1942. This again it belongs to this mechanical era because, no electronic component was used in building it. Similarly difference engine that was built by Charles Babbage for polynomial evaluation, evaluation that was developed in England in 1927 then, binary mechanical computer was develop by Konard Zuse in Germany in 1941. Then, electro mechanical decimal computer was developed by Howard Aiken in 1944 in Harvard and then mark 1 by IBM, okay. So, this is the mechanical era.

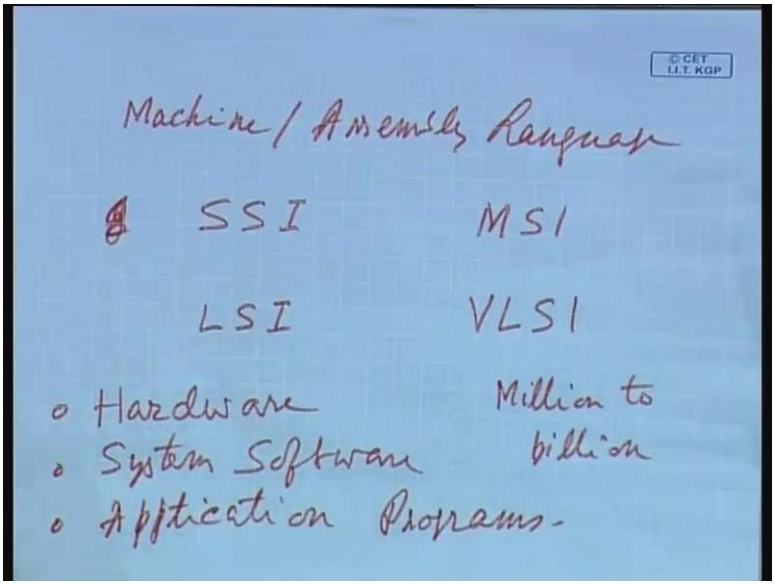
(Refer Slide Time: 05:15)



Coming to the electronic era, we can again divide in five generation, the first generation was before the invention of transistor obviously, before the mention of transistors, the electronic building block or component that was used is vacuum tubes. And that also used relay memories so, what is the vacuum tube? Vacuum tube was, is a kind of a tubes, small tubes with a filament which emits electron and by controlling the flow of electron the, the controlling flow of current you can actually realize the two states on and off and that is how you can realize a digital computers. So, vacuum tubes were used and relay elements which are essentially switches, which are used to store information for, where used as memory.

And this computers obviously, when used realized is vacuum tubes there very large in size. So, they were usually the, this computers were actually occupying big rooms and obviously distributing lots of power and computing power, although, the computing power was relatively much smaller in comparison to today's standard. So, maybe we are 1000 times slower and lesser computing power than present the PC that we use now a day. Then, these computers were essentially single user system. So, single users systems mean that, operating system that was used can provide, I mean can allow only one user to use the computer. And so far as the programming language is concerned, it was in a very, I mean primitive stage that means, you can use only machine or assembly language.

(Refer Slide Time: 07:43)



So, these computers were single users systems and the programming language that was available was machine or assembly language. Obviously, whenever you write a program in machine an as a machine language or assembly language it is very painful, machine or assembly language and to write a program, I mean programming was a big, big challenge. So, these computers were not very user friendly you can say and examples of this type of computers are ENIAC, Princeton IAS and IBM 701. So, these computers actually the first generation of computers were primarily developed, because of the requirement of department of defense of USA. So, they wanted to calculate the trajectory of a cell that is launched from the worships and where it will fall like that.

So, to do that calculation you need a computer. So, these computers were primarily developed for that purpose so in 40's. Then, came the second generation computer in between 1955 and 64. So, in these decade computers were realized by using transistors so, transistors were invented and which was used, I mean for the realization of the computers. So, transistors diodes and so far is memory is concerned magnetic ferrite course were used to store information. Then high level languages were used with compilers that means, now you can write program in high level language and so compilers were developed.

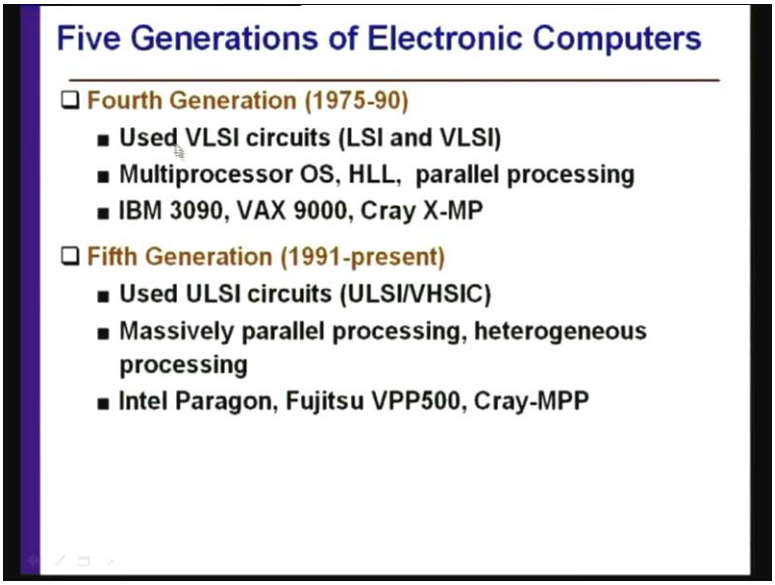
So, that was a very big step so far as the users are concerned and also it allowed batch processing that means, earlier the computers were developed for single users. Now you can do batch

processing that means, a particular computer can be used one after the other by different users. So, the computers like IBM 7090, CDC, 1604, Univac LARC these are the examples of the second generation computers, built using transistors and diodes. Then came the invention of integrated circuits so, by integrated circuits we mean you can put more than one electronic component like, transistors, diode and so on, on a single silicon wafer.

So, depending on the number of active devices that you can put on single silicon wafers, you can categorize into different types known as small scale integration SSI, small scale integration. Where, you can put a may be 1 to 10 activity devices then MSI where, you can put 10 to 100's of devices like transistors and diodes. Then, LSI large scale integration where you can put 1000's of devices. And now a days we are in the era of VLSI where, you can put millions of transistors or active devices may be say may be million to billion, million to billion of transistors you can put on a single IC.

So, these are the because of the evolution of integrated circuits has led to the development of powerful third generation computers, using S, but in this third generation only SSI and MSI devices are used. And so far as the operating systems are concerned, multi programming and time sharing OS were used and so a single computer can be used by a large number of users and all of them will be having the feeling that they are the sole users of the computer. So, in this category was IBM 360/ 370 computers so, this IBM 360/370 main frames computers became very popular and widely used throughout the world and in addition to that you have got CDC 6600, Texas Instruments ASC and PDP 8. These are different third generation computers which are developed by different manufactures and widely used.

(Refer Slide Time: 12:28)

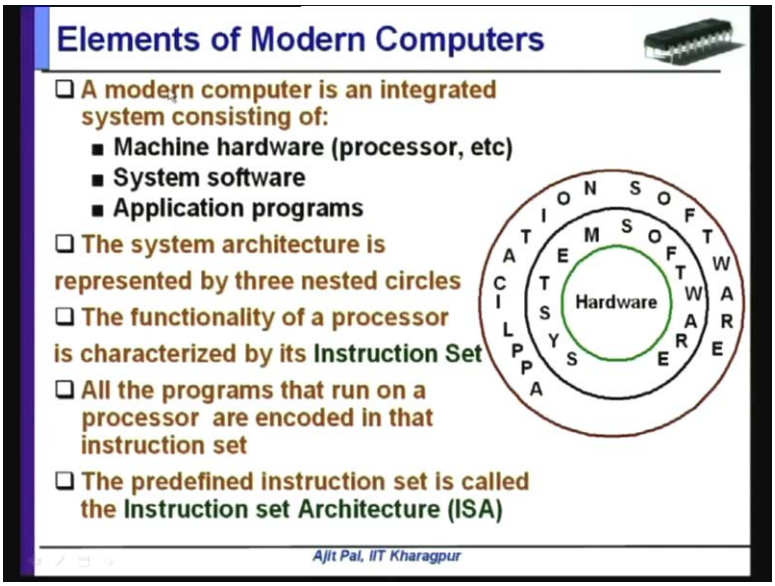


Then came we fourth generation as I said with the advancement of VLSI technology, this fourth generation computers use VLSI circuits and VLSI circuits and as a consequence, they were very powerful. And so far as operating system is concerned, multiprocessor operating system are used that means, these fourth generation computers it was possible to realize multi processor chips, a single IC can contain multiple processors. So, multiple operating system were developed various high level languages were flourished and parallel processing was, was very popular using this fourth generation computers, That means it was possible to have parallel processing, I shall discuss about what are the different types of parallel processing possible in the course of this lecture.

And the various computers based on this fourth generation are where IBM 3090, VAX 9000, Cray X-MPP and then come came the fifth generation which are, which are started may be from 1991 to present day. So, these use ULSI circuits, ultra large scale integrated circuits and or very high I mean SIC. So, these are the different types of very high scale integrated circuits were used in realizing fifth generation computers and to realize massively parallel processing. And earlier so far as the multiprocessor, parallel processing were concerned all the processors were homogeneous. That means same type of processors were used in those in those parallel processors, but in the fifth generation it was possible to have heterogeneous processor.

That means, one can perform I mean a graphic processing, one can perform integer processing and so on. So, different types of heterogeneous processors were combined to realize computer systems and there are the fifth generation computers, examples of this fifth generation computers are Intel paragon, Fujitsu VPP500, Cray MPP. So, these are some representative examples, but obviously this list is not complete, there are many more fifth generation computers.

(Refer Slide Time: 15:19)

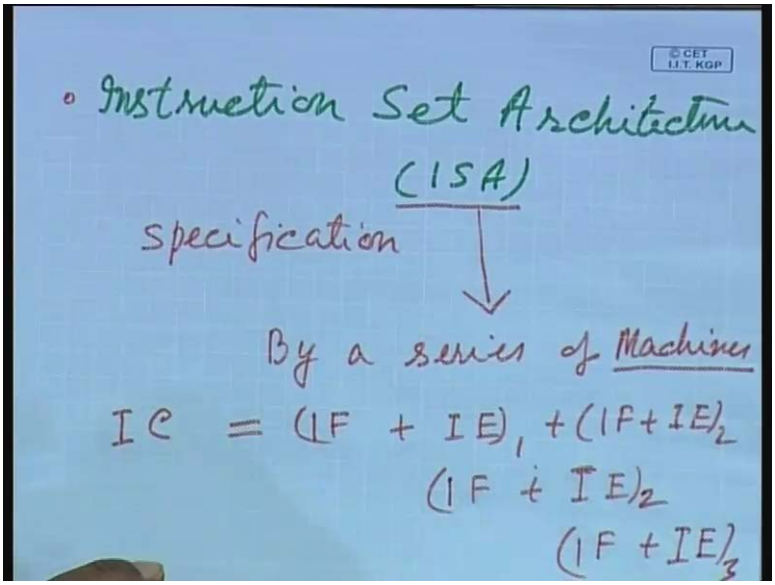


Coming to the elements of modern computers, you can see a modern computer is an integrated system consisting of machine hardware, system software and application programs. So, three very important components that makes a computer workable, useable to useable or number one is hardware, you require hardware. Second is your system soft ware and third is your application programs, various application programs and this can be represented with the help of this nested circles. So, as you can see hardware is at the core or center of these three circles. So, this hardware is being, is the interface between the application software and hardware is the system software. System software is essentially the operating system and different types of operating systems have evolved over the years and these system software or operating systems allows application software to run on this hard ware.

So, and the actually to the system software or the user or programmer the functionality of the processor is characterized by instruction set. So, a processor can execute a set of instructions and

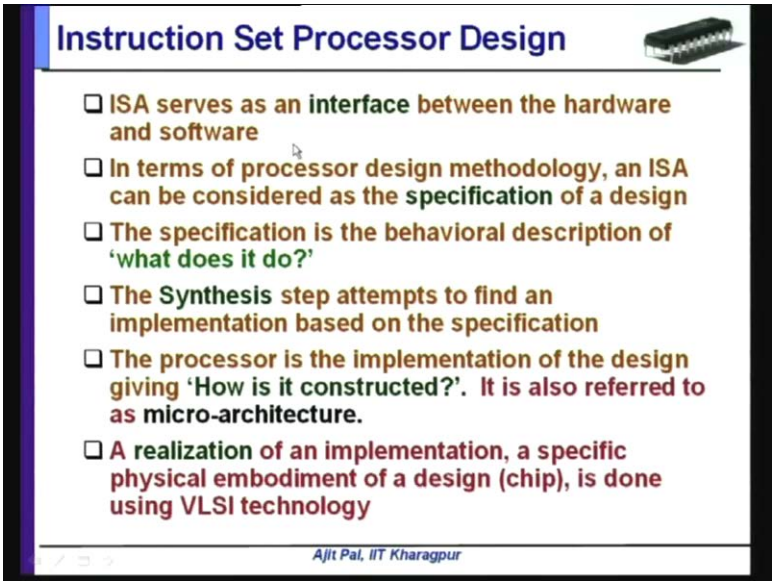
which can be used to write a program. So, program can be considered as a sequence of instructions. So, you can pick up instructions from the instructions set and realize a program. So, a programmer's view of the processor is as essentially the instruction set and that is why it is called instruction set architecture.

(Refer Slide Time: 17:32)



So, instruction set architecture or ISA in short, plays a very important role and it, it is a kind of abstraction and all, all programmers can look at it. So, this predefined instruction set is called instruction set architecture.

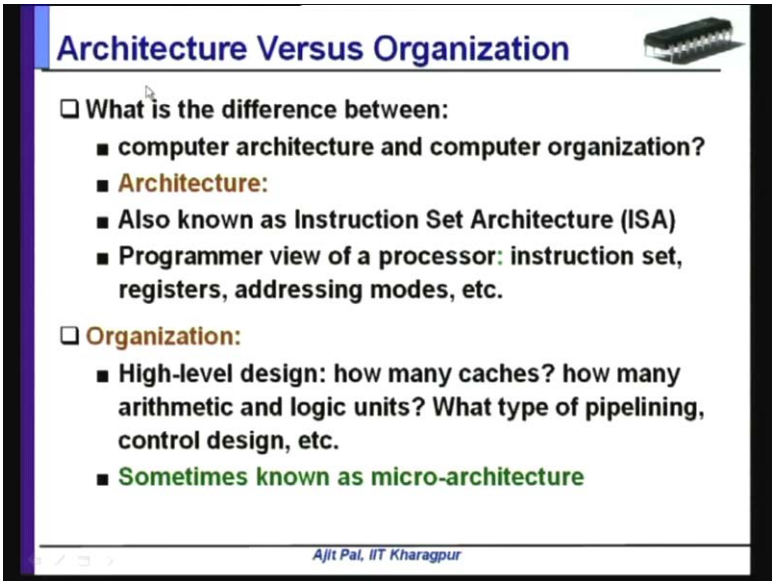
(Refer Slide Time: 18:00)



Now, this ISA or instruction set architecture serves as an interface between the hardware and software as I have told. So, here you have got the hardware and here you have got the different types of software like system, software application software and this instruction set architecture serves as an interface between the hardware and software. So, in terms of processor design methodology an ISA can be considered as the specification of a design. That means, so whenever you go for designing a system you have to provide specification. So, this specification this ISA, instruction set architecture can be considered as the specifications. And this instruction set architecture is, I mean to realize the instruction set architecture, you implement a, or synthesis a hardware.

So, the specification is the behavioral description, what does it do, what the processor can do? Then the synthesis step attempts to find an implementation based on the specification and then comes the processor, which is the implementation of the design and how is it constructed? That means instruction processor design concerns with how a processor is constructed. So, it is also referred to as a micro architecture so, a realization of an implementation, a specific physical embodiment of a design is done using VLSI technology now days. So, I in this course we shall discuss about the way this is, how it is being done, but before that as we shall see later, first we shall introduce instruction set architecture.

(Refer Slide Time: 19:56)

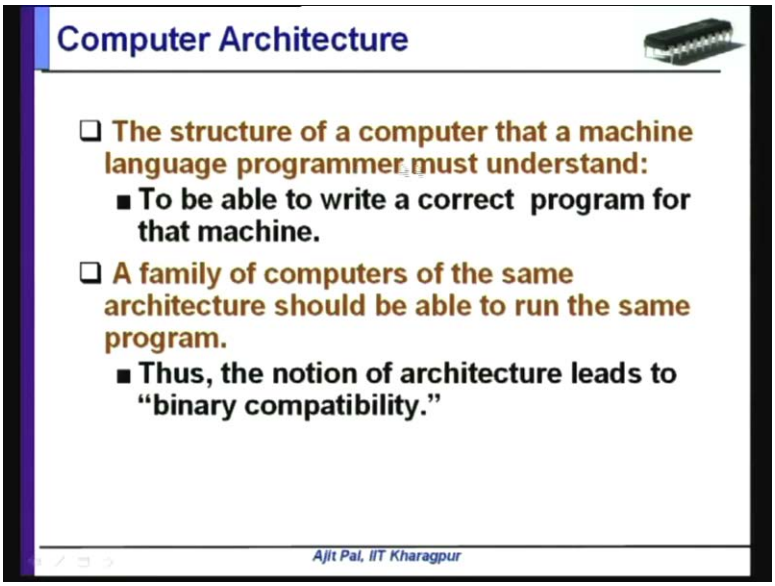


I will, before I proceed further it is essential to distinguish between these two terms, which you will encounter quite often may. One is architecture another is organization so, what is the difference between the two? So, a computer architecture and computer organization here as you can see, architecture is a short form not instruction set architecture. So, it is also known as instruction set architecture, as I mentioned this is a programmer's view of a processor. So, that means it, it, it specifies what are the instruction and different instruction that you can perform that means, data transfer, data manipulation like that. So, addition, subtraction, multiplication, division and then move instruction to transfer data between processor and memory processor and IO load, store like that.

And then, it will also provide a give an idea about different types of resistors, which you can use for storing information temporarily, while executing a program. So, resistors provide intermediate, I mean storage and then, the, you have you can have various addressing modes. Addressing modes allows you to access operands in various ways from resistors and memory and also from IO. So, this is architecture is essentially a programmer's view of the processor, on the other hand as I mentioned organization represents the high level design that means, the way the processor is implemented, how many caches memory it has got? How many arithmetic and logic units it has got?

What type of pipelining, control pipelining is being used? What type of control design is being done, whether it is hard ware control unit or whether it is a micro program control unit, whether the processor is single cycle, multi cycle, pipeline like that. So, these are all are decided as part of the organizations. So, this is also called micro architecture. So, this is sometimes known as micro architecture.

(Refer Slide Time: 22:27)

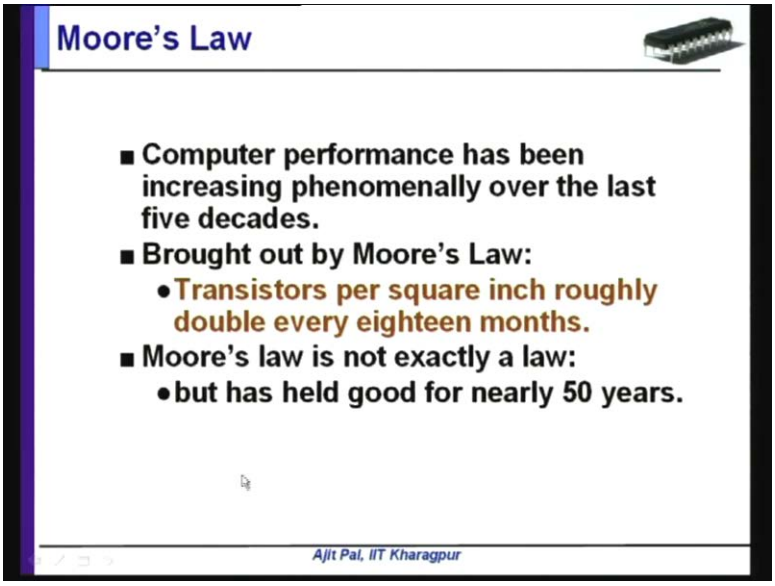


So, the structure of a computer that a machine language programmer must understand, that is your architecture to be able to write a correct program for that machine and a family of computers of the same architecture should be able to run the same program. So, here as you can see same instruction set architecture can be realized by a series of, series of machines or processors. Then in the realized, by using view in different ways one can use can be realized using transistor circuit, another can be by integrated VLSI circuits and they can the implementation can be done in different ways. But as long as they execute the same instruction set architecture, then we can say that there is a kind of binary compatibility that means, a program written for one machine can be run on another machine.

So, this is a very important concept and this binary compatibility plays a very important role, whenever we go for designing computers and we go from one generation to next generation of processors. So, you have to, I mean you have to take into consideration the binary compatibility

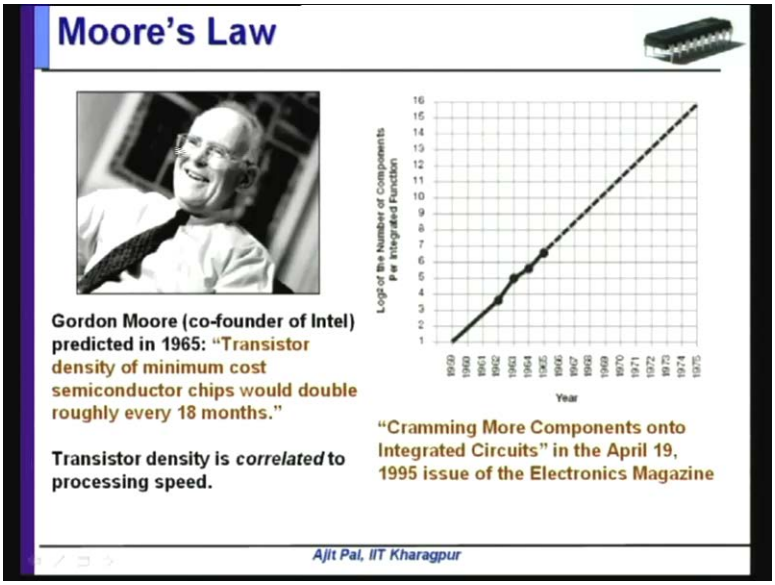
whenever you realize the next generation processor so that, the programs develop for the earlier generation can be used, can be, cannot be I mean need not be thrown away, need not be wasted and can be used.

(Refer slide Time: 24:18)



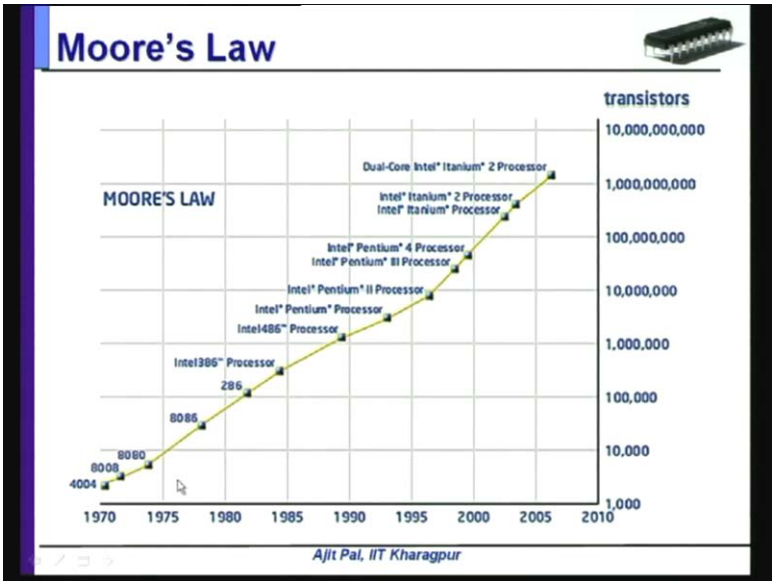
So, now coming to one very important ah concept that is your Moore's law, Gordon Moore he was one of the founder of Intel, he proposed a law based on some his observation. So, he compute what have, what have, what has been found that the computer performance has been increasing phenomenally over the last five decades. And this enhancement, this performance improvement is an outcome of you can say Moore's law. So, this was brought about by Moore's law, what is that? Moore's, Moore's law states that transistors per square inch roughly double, doubles every 18 months. So, it is not Moore's law is not exactly a law, but this particular rule you can say transistors per square inch roughly doubles every 18 months that has been hold good for nearly 15 years.

(Refer Slide Time: 25:29)



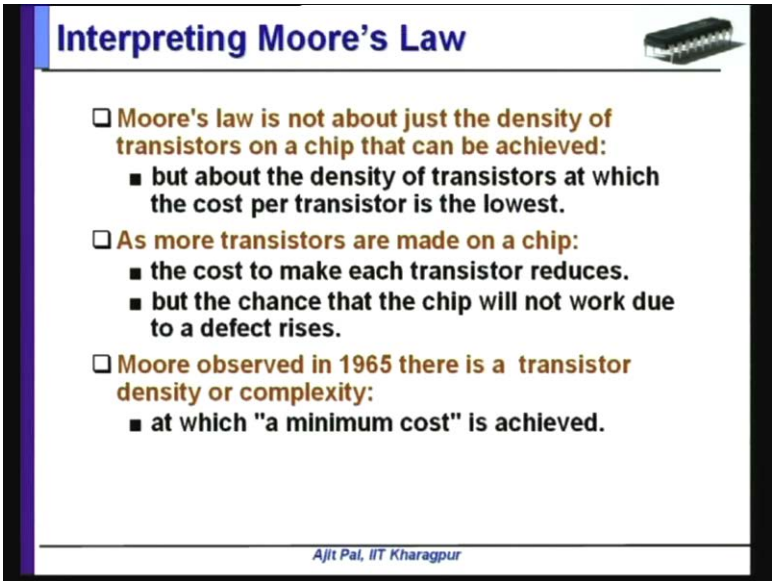
And this is the, this is Gordon Moore, each one of the co-founders of Intel and this is what he stated back in 19654 in his famous paper, he wrote it for electronic magazine, electronics magazine. So, he was asked to write an article predicting the future of electronic circuits. So, the title of the article was cramming more components onto integrated circuits and that was published in April 19, 1995 the issue of electronic magazine. And in that article, he predicted that transistor density of minimum cost semi conductor chips, would double roughly at every 18 months and obviously transistor density is correlated to processing speed, as we shall see.

(Refer Slide Time: 26:26)



So, this, this shows the, this shows how Moore's law has remained valued for, for a large number of about 50 years. You can see here only the Intel processors are being shown and on this side, on the y axis you have got the number of transistors and on x axis we have got the number of the years. So, as you can see as the, as you go from 1970 to 2010 the number of transistors is reaching from 1000 to several millions. Several millions of transistors are used to realize processor so, starting from simple Intel 4004, which was obviously having few 1000's of, I mean transistors to present their dual core and multi core processors, requiring multibillion transistors may be tens of billions of transistors.

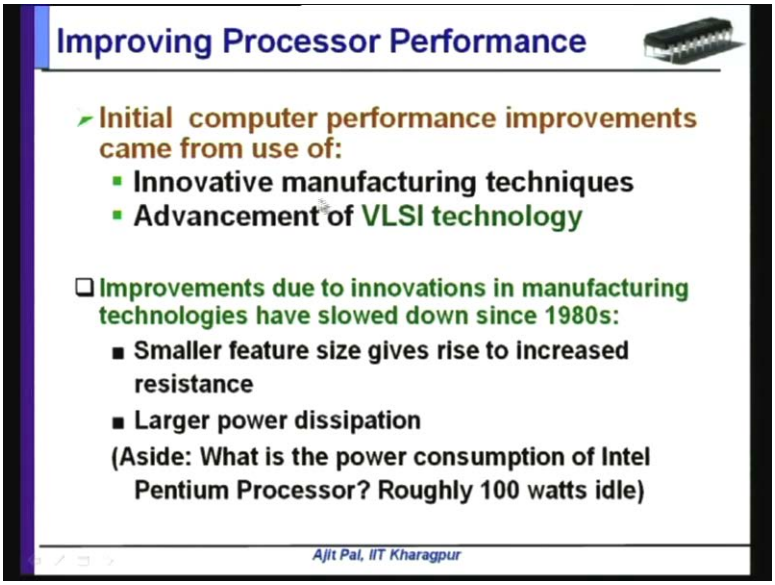
(Refer Slide Time: 27:34)



So, this shows the, the how Moore's law, Moore's law has really influenced the growth of computers. And Moore's law is not about just the density of transistors on a chip that can be achieved, but about density of transistors at which the cost per transistor is the lowest. That means, it not only says about how many transistors you can for fabricate, but it also says how economically you can do the fabrication. That means, that means you have to realize I mean transits IC's with more number of transistors in a economic way, cost of effective way. So, as more transistors are made on a chip, the cost to make each transistor reduces, but the chance that the chip will not work due to a defect rises.

Of course this, this problem is there and that is you have to take of it by liable processing technology and Moore observed in 1965, there is a transistor density or complexity at which a minimum cost is achieved. So, based on I mean a minimum where, the transistor density or complexity at which is a minimum cost is achieved, he proposed this law which has become famous.

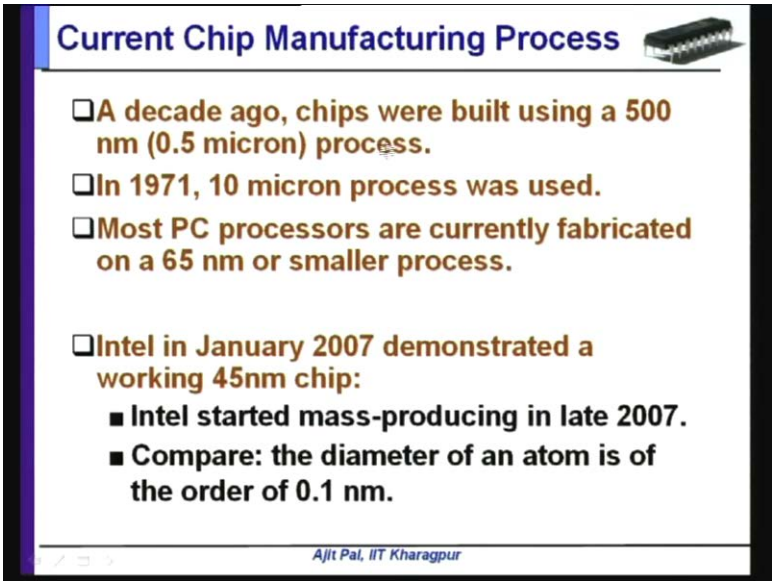
(Refer Slide Time: 29:01)



So, we can say that the initial computer performance improvements came from use of innovative manufacturing techniques, advancement of VLSI technology and which actually based on Moore's law you can say. And improvements due to innovations in manufacturing technologies have slowed down since 1980's of course the, the grow, the rate at which it was growing has slowed down since 1980's. So, smaller feature size gives rise to increased resistance, these are the two reasons. Number one is smaller feature size gives rise to increased resistance, this is one problem.

Second is larger power dissipation so, as we shall see the Pentium processor generates about 100's watts of power dissipation. So, 100 watts of power dissipation from an IC is a very large power dissipation. So, this power has to be dissipated with the help of suitable packaging and pooling technique and so as you put more and more transistors the, the power dissipation increases, the cost of packaging and cooling increases, this is one parameter which is, which limits the increase in number of transistors on a chip.

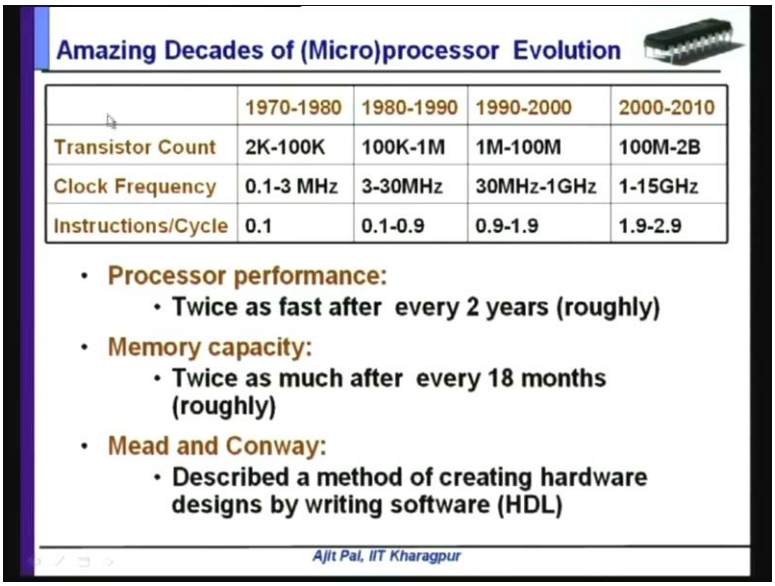
(Refer Slide Time: 30:28)



So, a decade, as I mention a decade ago chips was built using 500 nanometer technology. So, 0.5 micron technology you can say and in 1971, 10 micron process was used and most processors are used are currently fabricated on 65 nanometer or smaller process. So, now a days because of the advancement of VLSI technology, thanks to Moore's law which has been followed the, it has progressively, I mean the dimension has reduced gradually from 500 nanometer to. So, to sub-micron technology now as you can see now a days we use 65 nanometer or smaller process, may be 45 or 33 nanometer.

So, Intel in January 2007 demonstrated a working 45 nanometer chip and Intel started mass producing in late 2007, based on this 45 nanometer chip. So, you can it is very easy to pronouns this 45 nanometer, 33 nanometer and like that, but you think about the diameter of an atom. So, diameter of an atom is of the order of 0.1 nanometer so, we can see that we are very close to the diameter of an atom. So, the precession at which the VLSI technology VLSI fabrication is taking place now days is not far away from the, the diameter of an atom.

(Refer Slide Time: 32:04)



So, this particular table gives you the amazing decades of micro processor evolution. So, as you can see from 1970 to 1980 the transistor count was 2 k to 2000 k and clock frequency was 0.1 to 3 mega hertz and number of instructions per cycle was 0.1. That means you require 10 cycles to execute a instruction and in 1980 to 1990 the number of transistors increase from 1000 k to 1 million and the speed increases from 3 to 30 mega hertz and number of instructions per cycle also reduced. So, you can execute more number of instructions in a 0.1 that means, you can more or less execute one instruction per cycle.

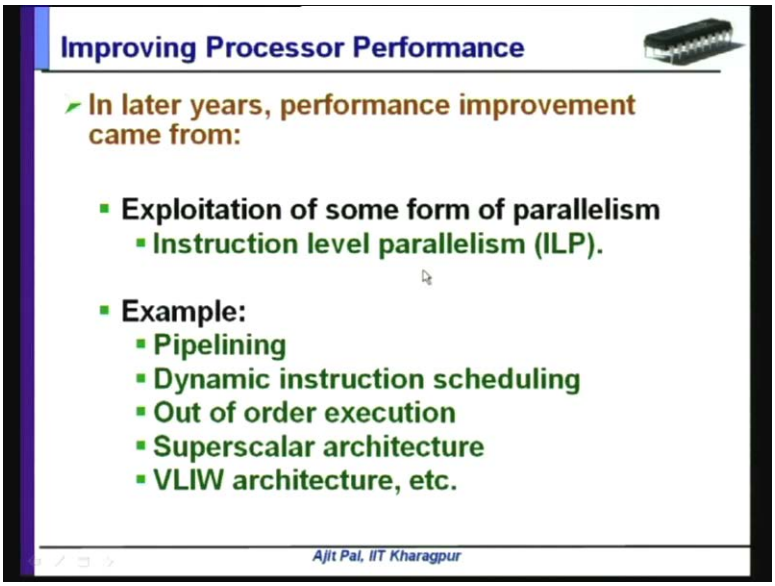
And then in 1990 to 2000 the number of transistor increase from 1 million to 100 million and the speed increase from 30 megahertz to 1 giga hertz and their number of instructions per cycle changed and reached 0.9 to 1.9. So, you may be asking how is it possible to execute more than one instruction per cycle, later on we shall discuss about that and that can be done by using super scalar architecture. And later on we shall discuss a detail about it then, came the 2000 to 2010, the present day technology where you can have 100 million to 2 billion transistors on a single chip and speed is from 1 to 15 giga hertz. So, 1 giga hertz means 10 to the power 9 hertz so, you can see the speed at which it will work and then, the number of instruction per cycle is again between 1.9 to 2.9 and it can be still more.

So, the processor performance has become twice as a fast after every 2 years and memory capacity has increased twice as much after every 18 months, roughly based on following Moore's law. And I should mention about another name Mead and Conway actually they

described a method of increasing hardware design, designs by writing software. So, whenever the number of transistors in a processor increases, it is not possible to do the design manually. So, we will require automated technique.

So, Gordon Moore developed this, develop this hardware description language by which you can, you can automate the design of the processor. That means, whenever you are going for designing processors with millions of transistors, you have to use cad tools, computer aided design tools and Mead and Conway was the founder of that. I mean they proposed the, this method for creating hardware designs by writing software. So, by writing software you can design hardware, that important step.

(Refer Slide Time: 35:28)



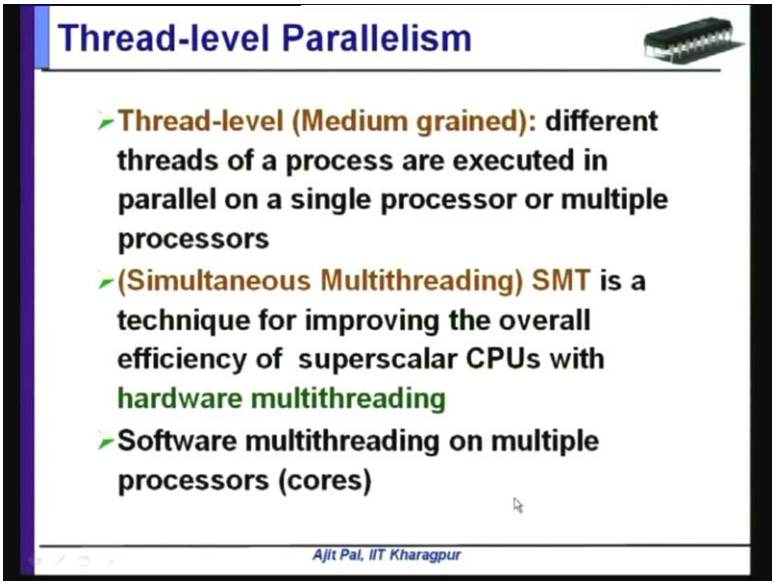
Question arises, how can we improve performance? So, initially as we have seen the improvement occurred because of the advancement of VLSI technology, but in subsequent years other thing the performance improvements came from exploitation of some form of parallelism. So, some form of parallelism where used in, in these, in these processors first is instruction level parallelism, what do you mean by instruction level parallelism? Instruction level parallelism means you know normally the execution of instruction take place serially, serially that means you execute, you fetch an instruction then, execute. So, you can say that instruction say cycle can be divided broadly in to two parts, instruction fetch, last instruction execution.

So, first you perform fetch one instruction then, you execute it then you fetch another instruction, this is for instruction 1. Then you fetch another instruction and then, execute another instruction that second instruction. So, in this way it goes on serially, there is no parallelism, but subsequently techniques were developed for instruction level parallelism and a pipelining is the first technique which, which used instruction level parallelism. In pipelining you will see the instruction later on we shall discuss in detail, we shall see the instructions can be executed in overlap manner. That means, say that instruction fetch when your execute performing instruction fetch of instruction one, instruction execution of instruction one.

You can perform instruction fetch of instruction 2 and again instruction execution of, when an instruction execution of instruction 2 is going on then, you can perform instruction fetch of instruction 2 and in this way it goes on. So, later on I shall discuss it in more details and also there were other techniques which are used, which is known as dynamic instruction scheduling, in which a multiple instructions can be scheduled dynamically with the help of hardware. And particularly, where ever you have got multiple execution units and then out of order execution can be performed and then super scalar processor.

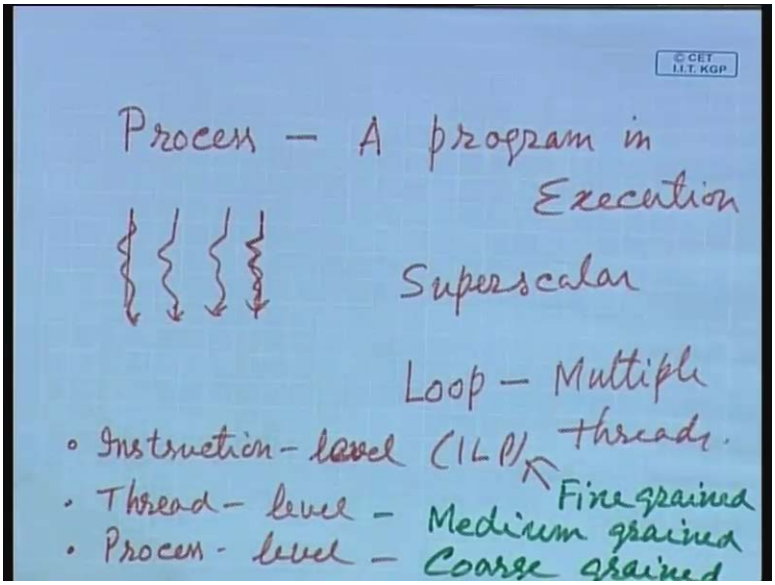
Where, you have got multiple processing elements within a single processor then, you can have VLIW architecture which, which, which can use super scalar architecture. So, the, the compiler will, will pack several instructions in a, I mean several instruction in a single instruction and which can be faced an executed serially. So, these are the this, this instruction level parallelism I shall discuss in detail in my lecture.

(Refer Slide Time: 38:48)



Then there is another level of parallelism, second level of parallelism that is your thread level parallelism. So, thread level parallelism we can say it is medium grained and different threads of a process are executed in parallel on a single processor or multiple processors.

(Refer Slide Time: 39:09)



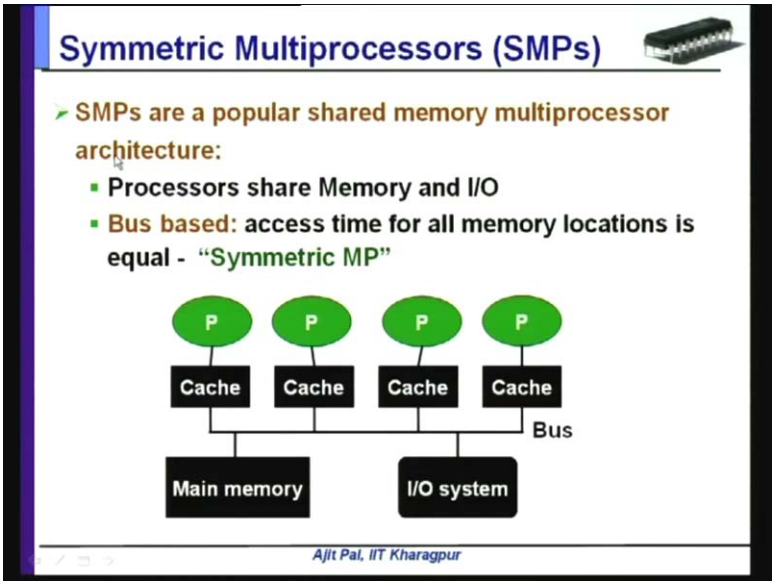
As you know you are familiar with what is process, process is nothing but a program in execution. Then a single process can be, can have multiple threads you can say, multiple threads

which can be executed in parallel. If you have got multiple processing unit for example, in a super scalar architecture you have got multiple processing elements. So, different threads can be executed in different processing elements. So, for example you are performing a loop program, in a loop program different iterations can be considered as different threads and which can be executed on different processing elements in a single processor.

So, a loop can have multiple threads so, this thread level parallelism is another medium grained compared to instruction level parallelism, which is fined grain. So, you can say, you can categorize in three types, first one is instruction level parallelism, instruction level parallelism or in short ILP. This is your fine grained, fine grained the second is your thread level parallelism and which is medium grained and third type is as we shall see later, process level, process level parallelism we and which is coarse grained you can say. So, these three thread levels of types of parallelism can be used and this simultaneous multi threading is a technique for improving the overall efficiency of supers scalar CPU's using hardware multi-threading.

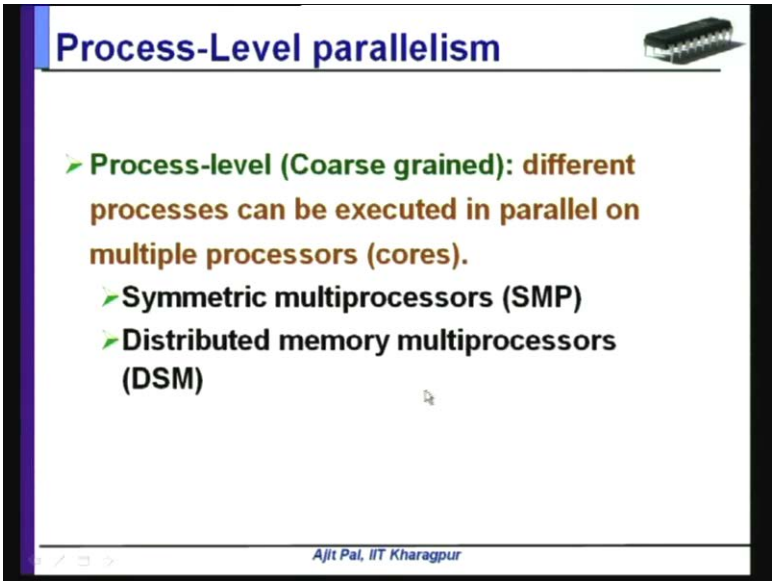
That means, in a single processor where you have got multiple functional unit of that is present in a super scalar processor, you can have this hardware multiple threading. That means, this thread level parallelism is exploited with the help of hardware and which is known as SMT or simultaneous multi-threading. And then of course, you can have software level multithreading on multiple processors or cores multi-threading, software multi-threading that can be done on multiple processors or cores.

(Refer Slide Time: 42:34)



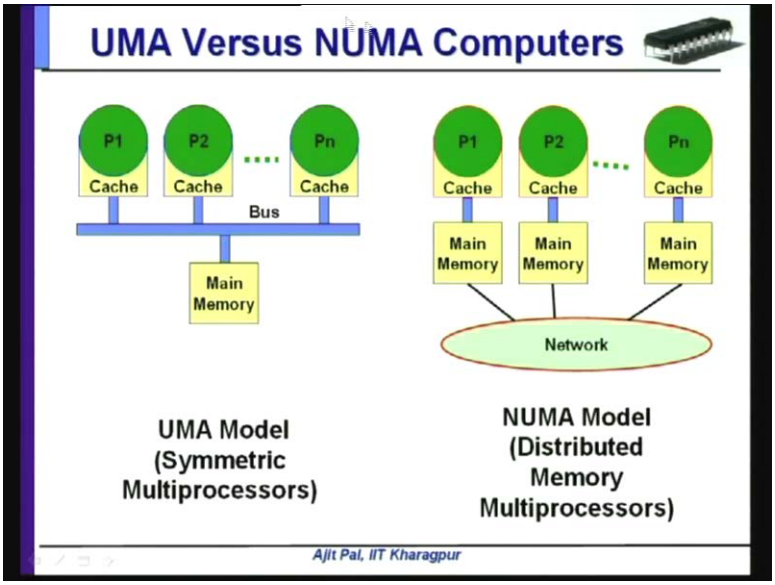
So, this is your symmetric multi processor SMP's this is very popular shared memory multiprocessor architecture. So, here as you can see you have got multiple processors, multiple processors each processor is having a private cache. Each of this processor is having a private cache and all of them are connected through a bus to main memory and IO and this is. So, this, this memory main memory and IO are shared and sharing is done through a common bus and it is called symmetric multi-processor. The reason for that is, each of these processors will take the same time to access main memory as well as Io. So, it is so, the access time for main memory and IO devices is symmetric for all the processor or uniform.

(Refer Slide Time: 43:35)



So, it is also called uniform processor architecture and process level as I mentioned process level or coarse grained where, you have got different processors that can be executed in parallel on multiple processors. And you can use symmetric multi-processors as I have already shown, also you can have distributed memory multi processors as you see in this diagram.

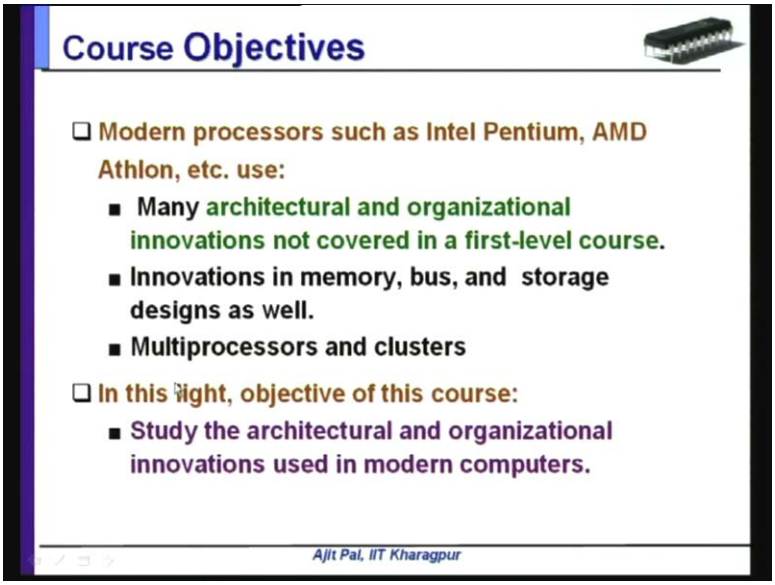
(Refer Slide Time: 44:00)



So, this particular model which I have already shown here each processor is has got a private cache and there is a shared bus through which main memory or shared memories accessed. So, this is called uniform memory access because, each of them can access it in a uniform manner and it is also called symmetric multi-processors as I have told. Then, you have got non uniform memory access, here as you can see the, the, there the memory is distributed, main memory is distributed and when, when this processor is accessing this memory, obviously access time will be smaller. When this processor, processor two is trying to access the main memory attach to processor 1 obviously, it has to do it through this network or called inter connection network.

So, whenever it is accessing through this inter connection network, the access time will be longer and also this access time can be variable depending on the, the, on the type of network being used and availability of the network. So, this is your distributed memory multi processor where, you have got multiple processors and the different memories attached to different processor are accessed through a inter connection network or it can be a local area network.

(Refer Slide Time: 45:36)



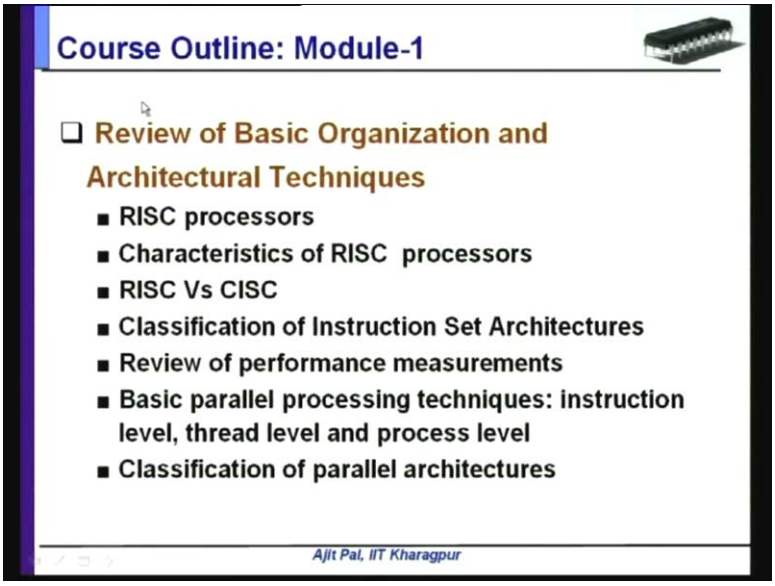
Now, what is the objective of this course? So, I have discussed broadly given an idea about the different types of processors, that you can have and also we have discuss about the different types of parallelism that is possible, instruction level parallelism, thread level parallelism, process level parallelism and different types of architecture. Now what is the objective of this,

this advanced computer architecture course? We shall see that, modern processors such as Intel, Pentium, AMD Athlon etcetera are used many architectural and organizational innovations that has not been covered in the first level course. So, each and every student who are attending this course must have attended a first level course on computer architecture and organization.

So, in that first level course these advanced topics were not in covered. So, which have and it is very essential for them, for the computer science students to learn this details of this processors. So, particularly the various innovations that have been used in implementing these processors, then innovations in memory, bus and storage designs as well. So, we shall see, we shall discuss about hierarchical memory organizations, the way, the performance gap between memory and processor can be reached.

Then, we shall also see multi processors, how multi processors can be realized and clusters can be implemented and in this way you can say the objective of this course is to study the architectural and organizational innovations used in modern computers. So, in a single sentence we can state the objective of this course, study the architectural and organizational innovations used in modern computers.

(Refer Slide Time: 47:49)

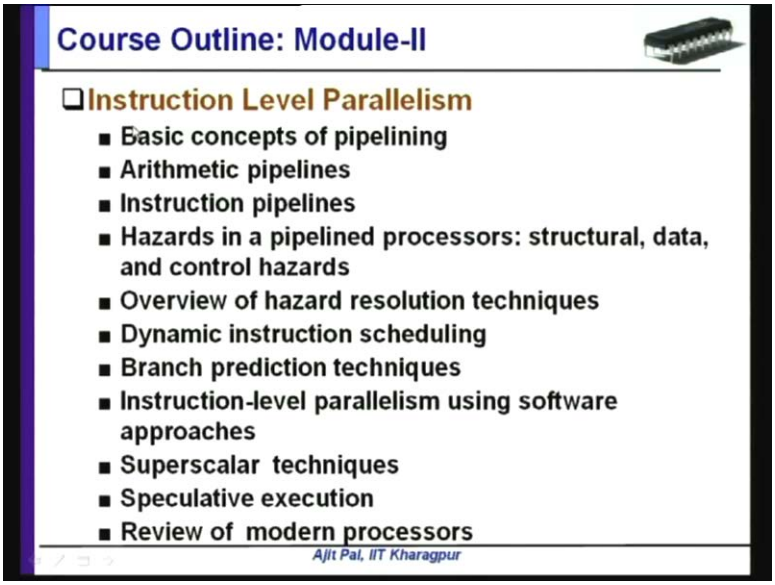


Now let me give you an outline of the course that I shall discuss and the course has been divided into several modules, 5 modules. So, in module one I shall present the review of the basic

organization and architectural techniques. So fundamentals, the fundamentals of different I mean processors like RISC processor, what is RISC processor, what is CISC processor and what are the characteristics of RISC processor and what are the differences between in RISC and CISC processor. And then, the classification of instruction set architecture, these we shall discuss in this particular module and also we shall discuss about the, the way, the performance of processors are measured.

So, it is very whenever you say high performance, question naturally arises how do you really measure the performance. So, we shall discuss about the technique by which the performance can be specified and performance can be measured and we shall review these performance measurement techniques. And then, I shall discuss about the basic parallel processing techniques, as I have already told instruction level, thread level and process level and I shall also discuss classification of parallel architecture, in this module 1, the various parallel processor architecture that is possible.

(Refer Slide Time: 49:30)



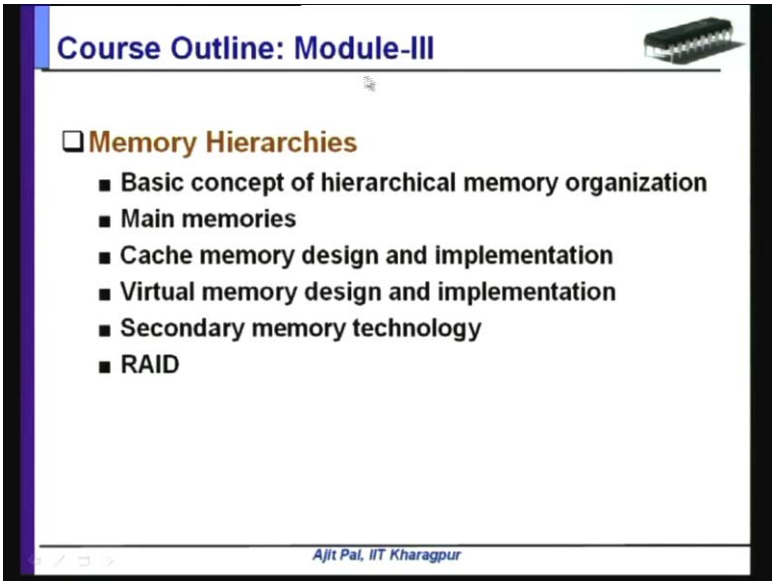
Then coming to module 2, I shall focus on instruction level parallelism and as I mentioned the first approach, which exploits instruction level parallelism is pipelining. So, basic concept of pipelining will be introduced and based on that basic concept we shall discuss about the, the arithmetic pipelines. How different arithmetic operations like fourteen point addition, then

integer multiplication can be performed pipeline, that I shall discuss. But most important is the instruction pipelines, which are used in all modern processor. So, I shall discuss in more details about instruction pipelining and particularly whenever you go for instruction pipelining, I will find that there are different types of hazards, that is present in a pipelines processors.

That means, whenever you do instruction pipelining you want to utilize each and, each and every cycle processor, cycle. But unfortunately because of various types of dependences like data dependences, control dependence and structural dependence you will find that it is not possible to avoid these hazards. And I shall discuss about the three different types of hazards, that is your structural hazard, data hazard and control hazard and also we shall discuss various hazards resolution techniques that can be used. Then, as I mentioned I shall discusses about dynamic instruction scheduling, that can be that is important in the context of your super scalar architecture and also branch prediction technique, which is related to control hazards.

How we can predict branch and we can minimize the effect of control hazards then, we shall discusses about instruction level parallelism using software approaches and as I, we will discuss about super scalar techniques, speculative executions and highlight how these various techniques have been implemented in modern processors.

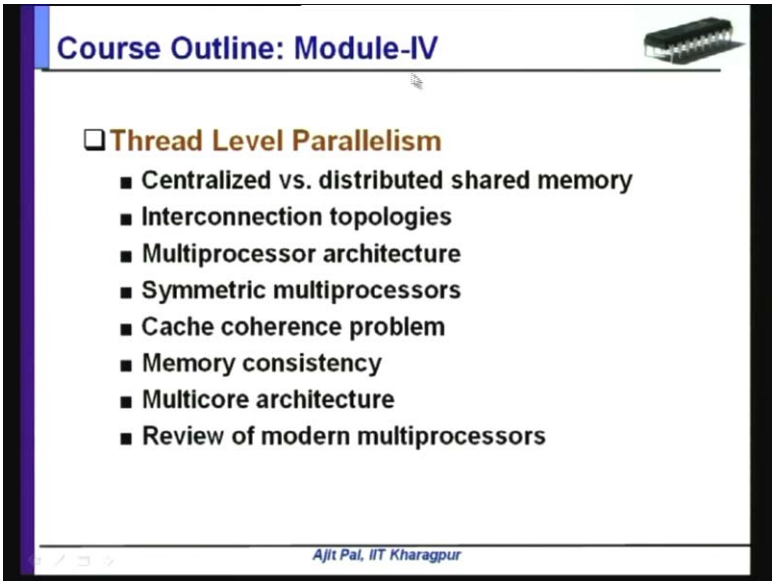
(Refer Slide Time: 52:06)



So, I shall review some of the modern processors, coming to module three, I shall discuss about memory hierarchies as I, as I mentioned the speed of processor is increasing. And later on we shall see the speed of memory is not increasing at the same rate. So, how do you bridge the gap? So, to bridge the performance gap one important approach that is being used is known as hierarchical memory organization where, memory is organized in a hierarchical manner, in terms of speed. So, process the memory which is very close to the processor is known as cache memory then, you have got main memory. Then the third, third type of memory is secondary memory.

So, I shall discuss about these different types of memories like, main memories, cache memory design and implementation, virtual memory design and implementation. Then, secondary memory technology and also I shall discus about RAID, which is used in the context of secondary memory, redundant array of independent discs, that I shall discuss and how it is used to improve reliability as well as performance.

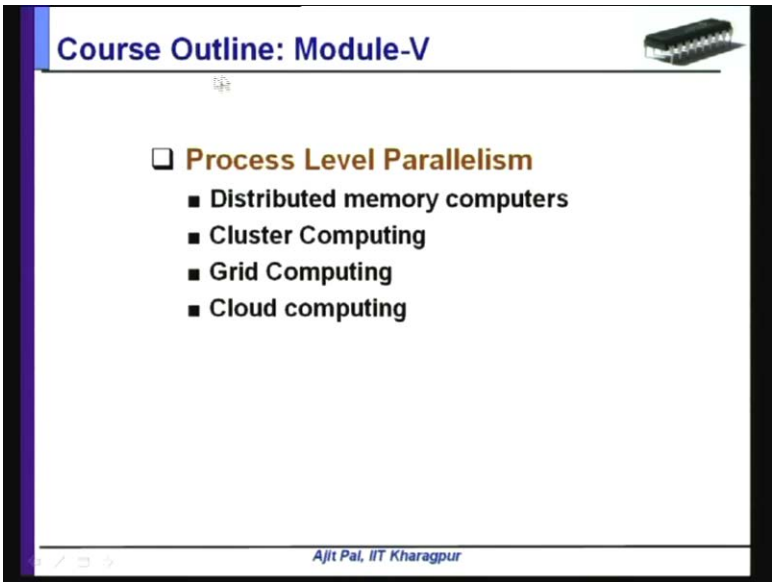
(Refer Slide Time: 53:28)



Coming to module 4, I shall discuss about thread level parallelism and in this context, we shall discuss centralized verses distributed shared memory architecture. Then various interconnection topologies, multi processor architectures and then symmetric multi processor and in this context there will be a problem known as cache coherence problem. Because, whenever you have got say

private cache and shared memory that will lead to cache coherence problem, leading to some inconsistency in the information that is stored in a memory. And how it is over come we shall discuss and then multi core architecture is an, is essentially an extension of multi processors in which the different processors are implemented on a single chip. And I shall discuss about modern multiprocessor, give a review of modern multiprocessors.

(Refer Slide Time: 54:25)



Coming to the 5th module where process level parallelism will be considered, I shall discus about distributed memory computers, different type, and different alternatives possibilities. And three different types of computing's which are increasingly becoming popular, one is known as cluster computing, grid computing and cloud computing. That I shall cover in this process level parallelism. So, with this we have come to the end of this lecture and in the next lecture we shall start with instruction set architecture.

Thank you.