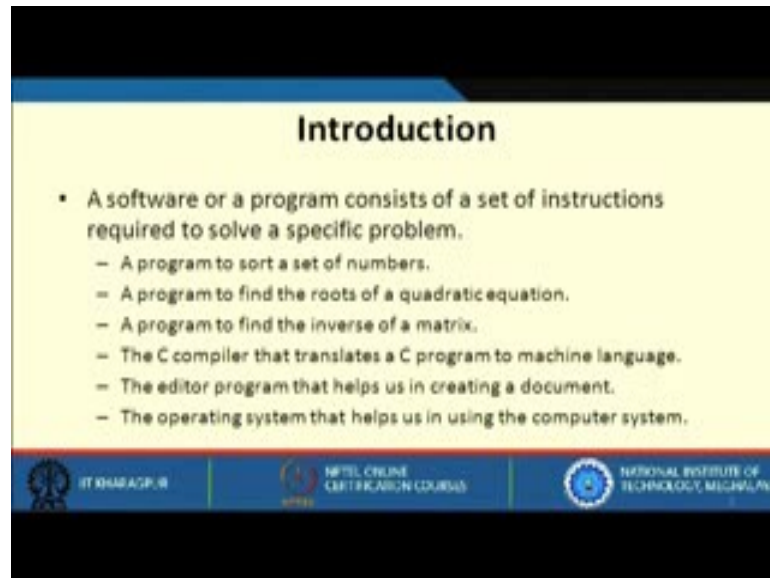


**Computer Architecture and Organization**  
**Prof. Kamalika Datta**  
**Department of Computer Science and Engineering**  
**National Institute of Technology, Meghalaya**

**Lecture - 04**  
**Software and Architecture Type**

(Refer Slide Time: 00:26)



**Introduction**

- A software or a program consists of a set of instructions required to solve a specific problem.
  - A program to sort a set of numbers.
  - A program to find the roots of a quadratic equation.
  - A program to find the inverse of a matrix.
  - The C compiler that translates a C program to machine language.
  - The editor program that helps us in creating a document.
  - The operating system that helps us in using the computer system.

IT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

Welcome to the fourth lecture software and architecture types. A software or a program consist of a set of instructions required to solve a specific problem. So, by that what we mean like a program to sort ten numbers, a program to add some numbers or a program to find out inverse of a matrix or you say a compiler a C compiler that converts your high level language into some machine language. All of these are a kinds of software. So, software consists of a set of instructions. A set of instructions are provided to perform certain task. And the operating system also is an software that helps us in using the computer system.

(Refer Slide Time: 01:30)

**Types of Programs**

- Broadly we can classify programs/software into two types:
  - a) **Application Software**
    - Which helps the user to solve a particular user-level problem.
    - May need system software for execution.
  - b) **System Software**
    - A collection of programs that helps the users to create, analyze and run their programs.

IT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | NATIONAL INSTITUTE OF TECHNOLOGY, BHUBANESWAR


So, what are the types of program that we have? Broadly we can classify the programs or you can say software into two types. First one is the application software which helps the user to solve a particular user level problem and it may require a system software for execution. Similarly, a system software is basically collection of many programs that helps the user to create analyze and run their programs. So, this is very important to know that we have two kinds of programs; one kind is application software, another kind is system software.

(Refer Slide Time: 02:26)

**(a) Application Software**

- Application software helps users solve particular problems.
- In most cases, application software resides on the computer's hard disk or removable storage media (DVD, USB drive, etc.).
- Typical examples:
  - Financial accounting package
  - Mathematical packages like MATLAB or MATHEMATICA
  - An app to book a cab
  - An app to monitor the health of a person

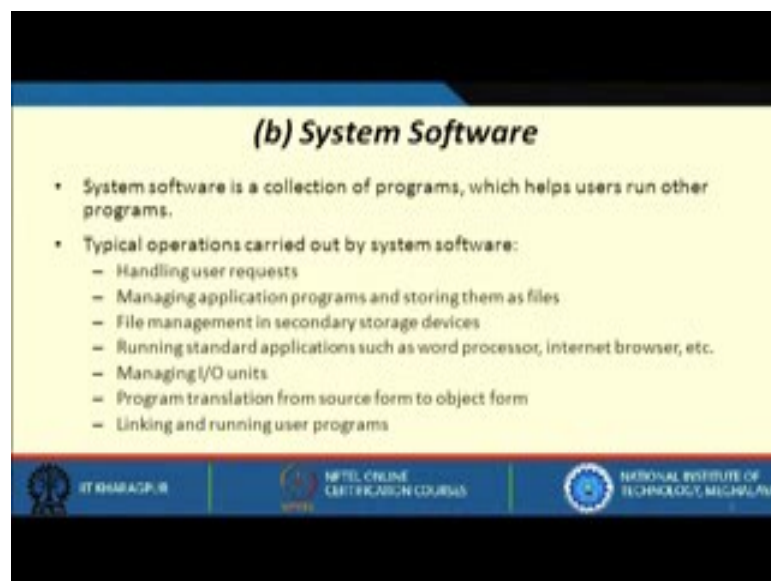
IT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | NATIONAL INSTITUTE OF TECHNOLOGY, BHUBANESWAR



Now, coming in detail of application software. This helps the user to solve a particular problem like what do you mean by a particular problem. Let us say we want to have a financial accounting package, I want to do some kind of financial accounting stuffs. So, for that I need a very specific software for that purpose, specific to that particular application. What is the application the application here is the financial accounting. In a similar fashion, a mathematical package like MATLAB, which is used to perform a particular kind of mathematical operations and various programs can be written using that. So, those are specific to some mathematical operation.

Similarly, you think of an app we have in our mobile phones to call a cab that is also an application, but what that application is doing that particular application is helping us to call a cab. In a same way, there can be various apps to monitor your health, there can be various apps to do various other functions. So, all these comes under application software.

(Refer Slide Time: 04:01)



**(b) System Software**

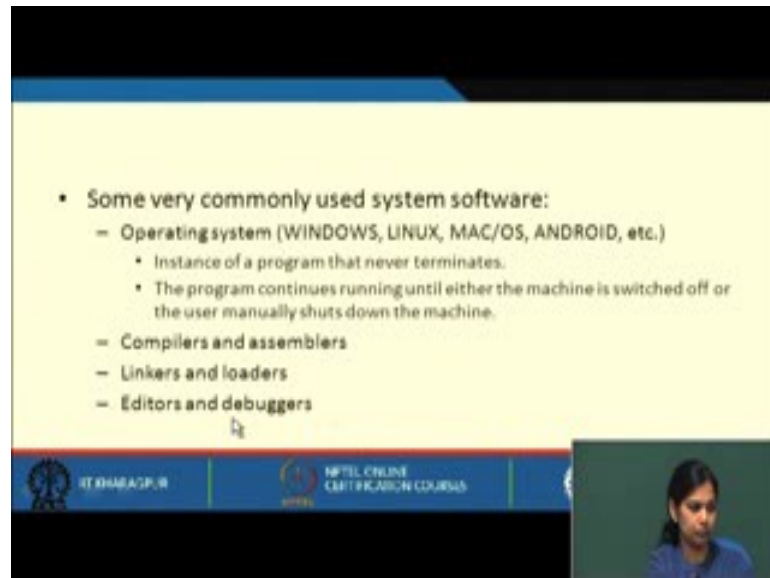
- System software is a collection of programs, which helps users run other programs.
- Typical operations carried out by system software:
  - Handling user requests
  - Managing application programs and storing them as files
  - File management in secondary storage devices
  - Running standard applications such as word processor, internet browser, etc.
  - Managing I/O units
  - Program translation from source form to object form
  - Linking and running user programs

IT KHARASIPUR | NPTEL ONLINE CERTIFICATION COURSES | NATIONAL INSTITUTE OF TECHNOLOGY, Bhubaneswar

Now, coming to system software. A system software is a collection of programs which helps user run other programs. So, a system software is also a collection of program, but it also helps other user to run a particular program. So, let us see some typical operations that are carried out by a system software. What it does, it handles user request, it manages application programs and storing them as files, it also does file management in secondary storage that is very important, running standard applications such as word

processor, internet browser, etc, managing input output unit, program translation from source form to object form, linking and running user program and many others.

(Refer Slide Time: 05:23)



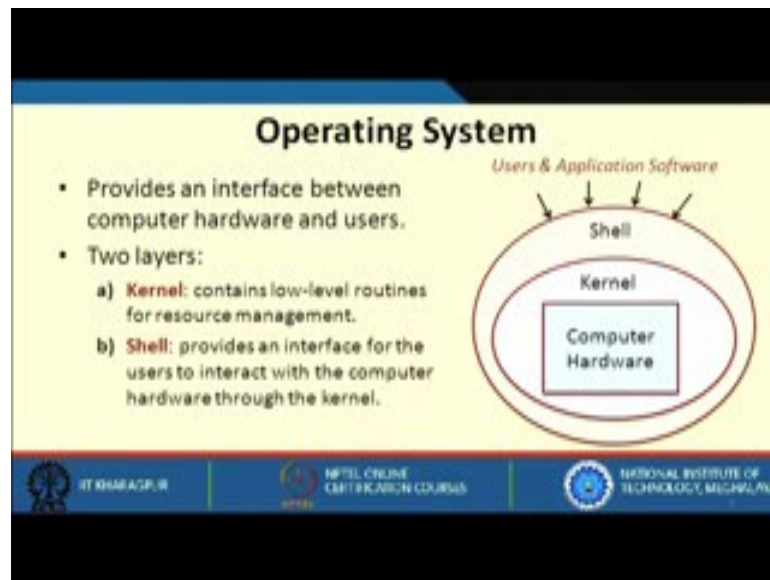
• Some very commonly used system software:

- Operating system (WINDOWS, LINUX, MAC/OS, ANDROID, etc.)
  - Instance of a program that never terminates.
  - The program continues running until either the machine is switched off or the user manually shuts down the machine.
- Compilers and assemblers
- Linkers and loaders
- Editors and debuggers

IT BHARATPUR | NPTEL ONLINE CERTIFICATION COURSES

So, now what are the commonly used system software that we all know is our windows machine, Linux, your MAC. So, various kind of operating system that are there consists of system software. And this program it continues running until you have switched off your machine or your machine is actually shut down. If your machine is shut down, the system software will automatically stop, but as long as your system is running those software will be running. You have compilers, you have assemblers, you have linkers and loaders, and also editors and debuggers.

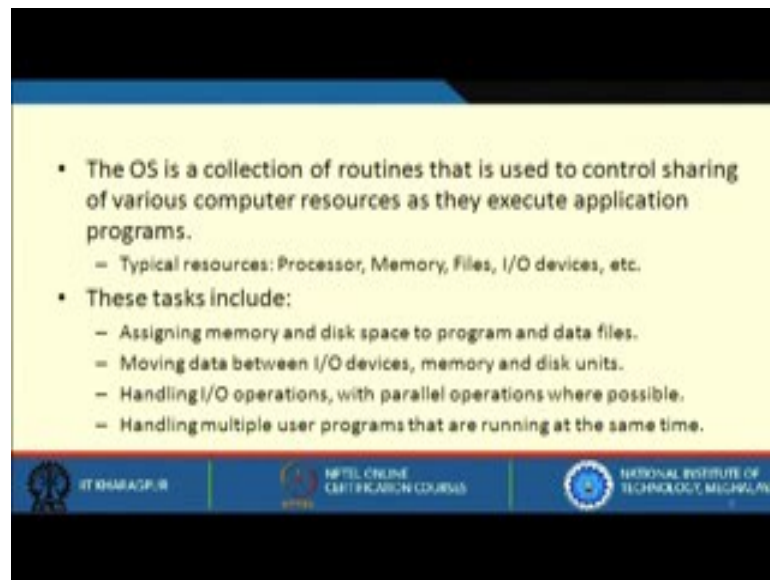
(Refer Slide Time: 06:15)



So, what is an operating system? It is a system software as I said and it provides an interface between the computer hardware and the user. So, the interface between your hardware and what is your hardware here your hardware is the processor, your hardware is the memory, your hardware is the input-output. So, how does the user actually interact with the hardware is through an operating system. So, operating system is sitting in between which helps in talking between the user and the hardware. The hardware and the user talk to each other through this operating system.

And there are two layers, kernel layer contains the low level routine. So, from this diagram, we can see that hardware sit in the bottom, and then we have a kernel which contains low level routines for resource management, and then we have a shell. Actually all users and application software cannot access this computer hardware directly. So, through this shell it provides an interface for the user to interact with the computer hardware through the kernel. So, kernel is sitting between shell and computer hardware; and between kernel and user, shell is sitting. So, shell is an interface for the user or the application software to interact with the computer hardware through this kernel.

(Refer Slide Time: 08:06)



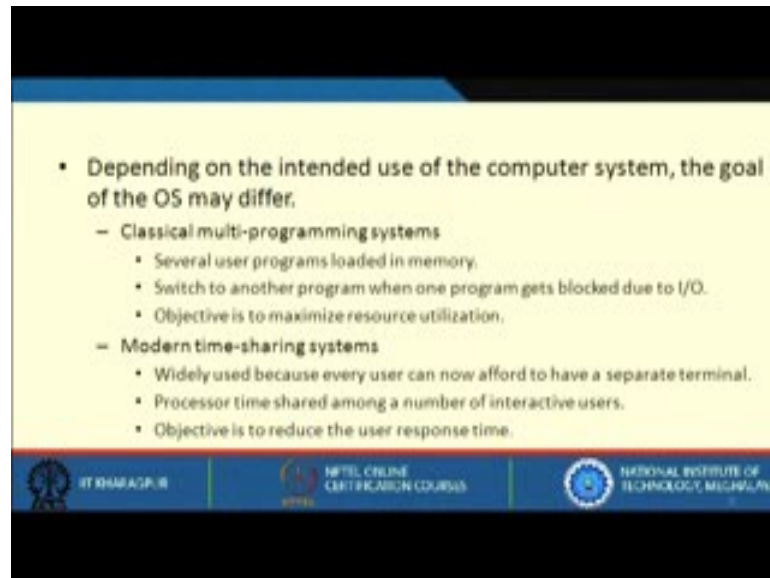
The slide features a yellow background with a blue header and footer. The main content is a bulleted list. The footer contains three logos: IIT Roorkee, NPTEL Online Certification Course, and National Institute of Technology, Meghalaya.

- The OS is a collection of routines that is used to control sharing of various computer resources as they execute application programs.
  - Typical resources: Processor, Memory, Files, I/O devices, etc.
- These tasks include:
  - Assigning memory and disk space to program and data files.
  - Moving data between I/O devices, memory and disk units.
  - Handling I/O operations, with parallel operations where possible.
  - Handling multiple user programs that are running at the same time.

Operating system is a collection of routines that are used to control sharing of various computer resources as they execute application programs. So, as you know that when we execute a program, some set of instructions get executed. So, for executing such instruction what we are doing we have to load those instruction into the memory, and then from memory it is brought into the processor and each time it is executed. When a processor is executing something the processor is executing only that particular instruction and then many other instructions can also come in other programs can also come in and be in the queue for execution.

So, it is the operating system task that how the various resources can be managed and allocated to processor. When the processor will be allocated to a particular process at what time, it is up to the operating system to decide upon. The task include assigning memory and disk space to programs and data files. Moving data between IO devices, memory and disk units. Handling input output operations with parallel operations were possible. And handling multiple user programs that are running at the same time. So, these are few tasks that are included. There are many more task that an OS performs.

(Refer Slide Time: 10:08)



• Depending on the intended use of the computer system, the goal of the OS may differ.

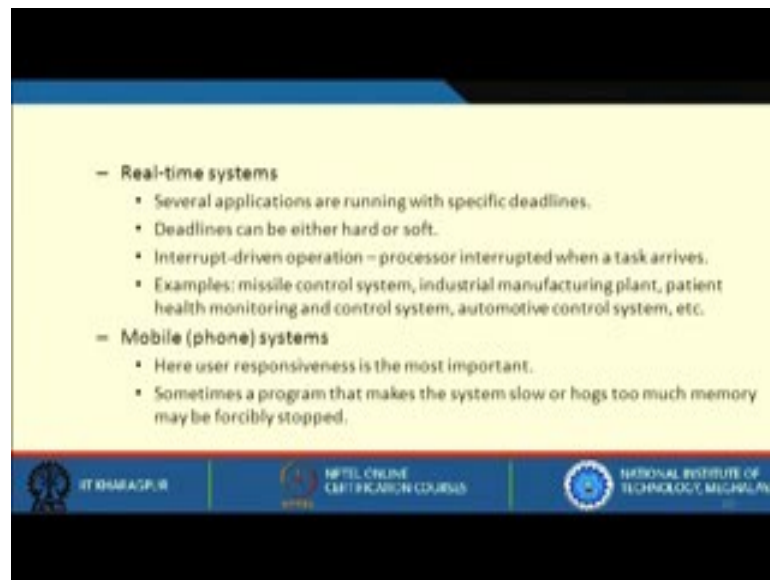
- Classical multi-programming systems
  - Several user programs loaded in memory.
  - Switch to another program when one program gets blocked due to I/O.
  - Objective is to maximize resource utilization.
- Modern time-sharing systems
  - Widely used because every user can now afford to have a separate terminal.
  - Processor time shared among a number of interactive users.
  - Objective is to reduce the user response time.

IT RANAPUR | NPTEL ONLINE CERTIFICATION COURSE | NATIONAL INSTITUTE OF TECHNOLOGY, BOMBAY

Now, depending on the intended use of computer system the goal of an operating system may differ like think of a classical multi programming system what happens there. There are several user programs loaded in memory and the OS can switch to another program when any other program is blocked for IO or any other purpose. So, let us say a program is running and at that point of the time that program needs some IO, an IO request has come for that particular program. So, the time it requires to handle that IO request, the processor can switch to another task and that another task can be assigned to the processor and run. So, the switching from one task to another is the main goal of classical multi programming system.

And what was the main objective, here the main objective was to maximize the resource utilization. So, the CPU must not sit idle; if it is doing some particular task and at a time that particular task requires some other resources for completion of that task then the processor has the flexibility to bring another task and execute that, and later when that particular task has completed its IO operation that task can get executed. Now, modern time sharing systems has some other properties. These systems are widely used because every user can now afford to have a separate terminal. Now, the processor time is shared among number of interactive users and here the main objective is to reduce the user response time. What is user response time, the user has requested for a task and by what time that particular request can be taken care of, so that is the user response time.

(Refer Slide Time: 12:19)



The slide contains the following text:

- Real-time systems
  - Several applications are running with specific deadlines.
  - Deadlines can be either hard or soft.
  - Interrupt-driven operation – processor interrupted when a task arrives.
  - Examples: missile control system, industrial manufacturing plant, patient health monitoring and control system, automotive control system, etc.
- Mobile (phone) systems
  - Here user responsiveness is the most important.
  - Sometimes a program that makes the system slow or hogs too much memory may be forcibly stopped.

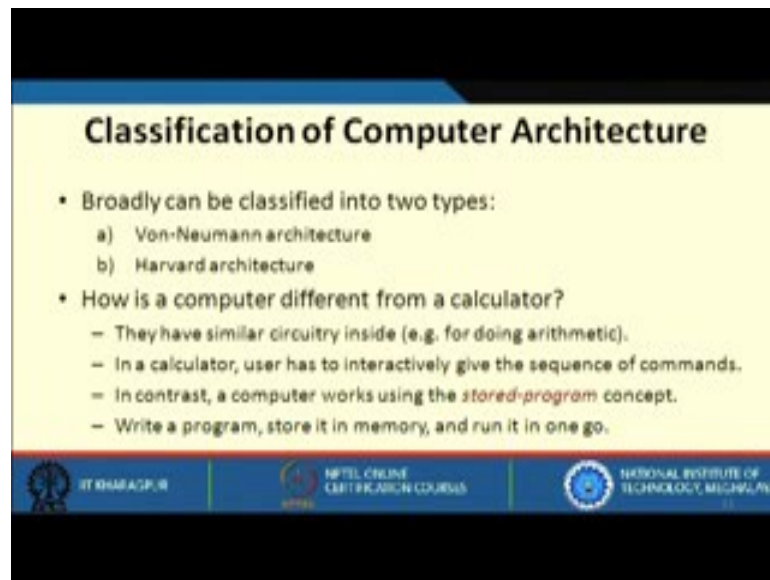
At the bottom of the slide, there are three logos: IIT Kharagpur, NPTEL ONLINE CERTIFICATION COURSES, and NATIONAL INSTITUTE OF TECHNOLOGY, MADRAS.

We have other kind of systems like real time systems. In real time systems there is a time constraint associated with it; and whenever there is a time constraint associated with it we can say that there is a specific deadline associated with the task. And these deadlines can be hard or soft. What do you mean by hard deadline? By hard deadline we mean that that particular task has to finish within that particular time. And by soft deadline that even if that deadline is not met, the system will not fail, but in a hard real time system if the deadline is not met the system may fail. Interrupt driven operations are also there where the processor is interrupted when a task arrives, this happens for some sporadic real time tasks, where there is no fixed time for task arrival. We do not know at what time a task will arrive. Some of the examples are missile control system, industrial manufacturing plant, patient health monitoring and control, and automotive control systems.

For mobile system the user responsiveness is most important. And sometime a program makes the system slow, let us say that we are running some programs in mobile. And in mobile we have limited memory, limited capability of certain things, we are putting everything in a very small space. So, if the computer is not able to handle it, what it does it stops those programs. So, it is forcibly stopped basically.



(Refer Slide Time: 14:27)



**Classification of Computer Architecture**

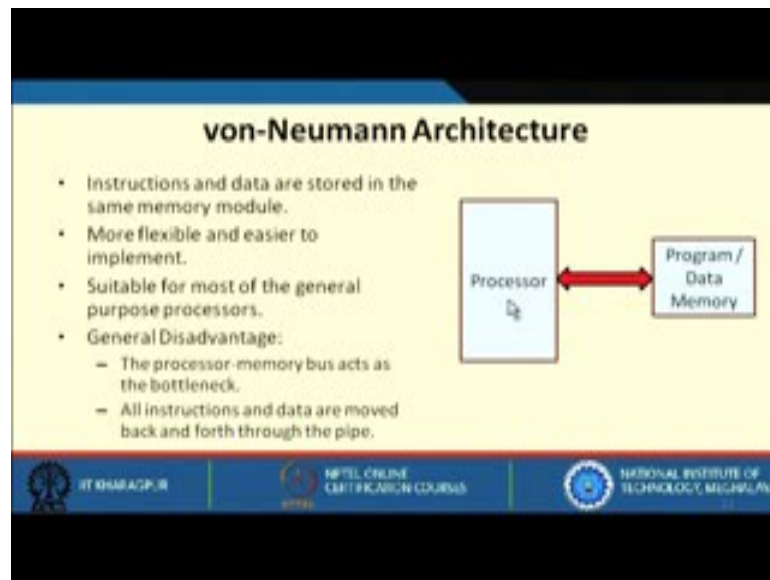
- Broadly can be classified into two types:
  - a) Von-Neumann architecture
  - b) Harvard architecture
- How is a computer different from a calculator?
  - They have similar circuitry inside (e.g. for doing arithmetic).
  - In a calculator, user has to interactively give the sequence of commands.
  - In contrast, a computer works using the *stored-program* concept.
  - Write a program, store it in memory, and run it in one go.

IT BHARATPUR | NPTEL ONLINE CERTIFICATION COURSES | NATIONAL INSTITUTE OF TECHNOLOGY, BHOPAL

Now, we will see the classification of computer architecture. Broadly, it can be classified into two types von-Neumann architecture and Harvard architecture. So, we will be coming into both these kind of architecture and both these kind of architecture are used in today's computing. How is a computer different from calculator? So, what a calculator does? it has got a circuitry it is adding something, it is dividing something, it is multiplying something and we are getting the result. It is a small device where it is battery operated and you can see the result in that small space.

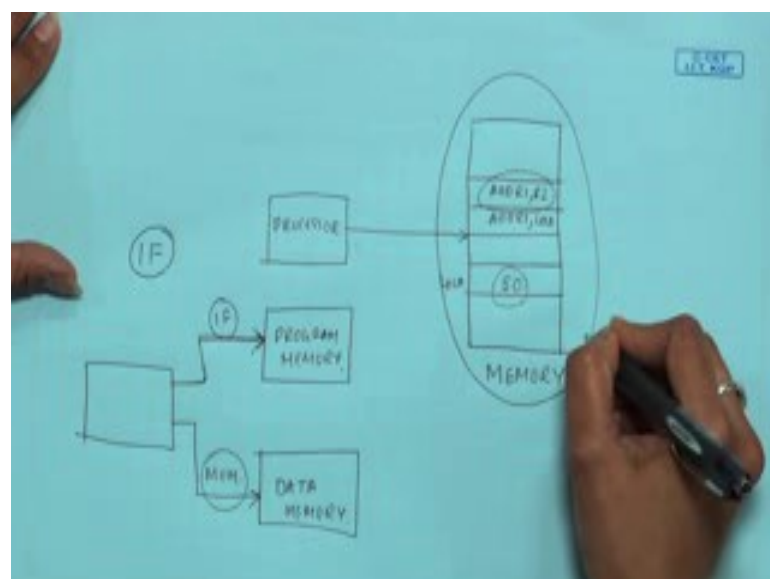
Now, is there any way that we can tell the calculator that I need to perform sequence of calculation one by one. But if you think of the old calculators which can only perform the task of adding 2 numbers or 10 numbers or 15 numbers. But in contrast to that if you see how it will be if we can load all the program into the memory and then I give the task to the computer that process all these instruction that I have stored into the memory one by one. So, this concept is known as stored program concept, where we load both the program and data into the memory and the programs are executed one by one and whatever data is required data are also read. So, we can write a program, store it in memory and we can run it in a go.

(Refer Slide Time: 16:25)



In von-Neumann architecture, both the instructions that is the program and the data are stored in the same memory module. So, this is the memory module and this is the processor both our program and our data are stored in same memory and it is very flexible and easier to implement. Suitable for most of the general purpose processor. But where is the bottleneck and what is the disadvantage? See we have loaded both the programs and data into the memory. If there is a way that I want to access both the program that is the instruction and the data at the same time, I cannot do that, why, because I have a single memory where both my program and data are stored.

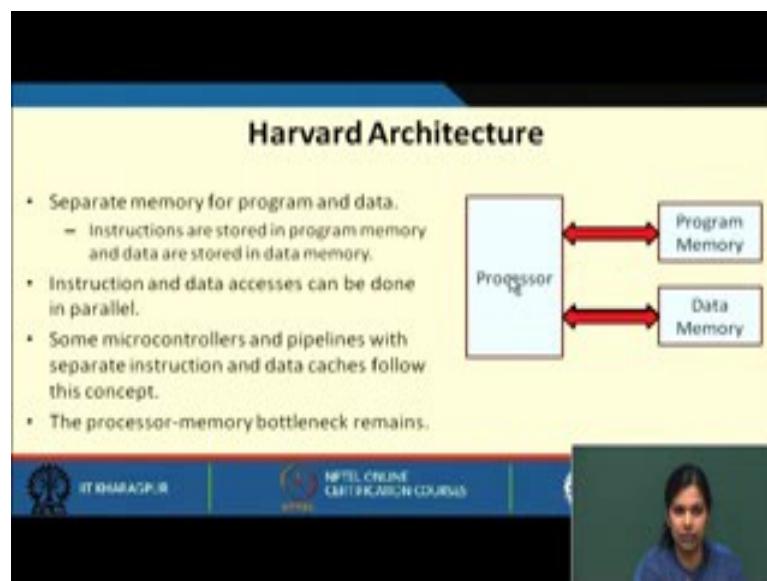
(Refer Slide Time: 17:25)



So, I am trying to say something like this. This is again my memory where in some location both my instruction as well as my data are stored; and the processor is connected to this. Now, when the processor is accessing instruction it cannot get the data at the same time, but what if we can get the instruction and the data at the same time, we will see that this feature is also required for reducing the processor and the memory speed gap.

So, we can see here this is one of the disadvantages; the processor memory bus acts as a bottleneck. So, at a time either instruction or data can be accessed. All instruction and data are moved back and forth through this pipe. So, this is the bottleneck, where processor has to wait for the programs as well as data. Now, what if we have a different program memory and a different data memory; then at the same time, we could also get the program that is the instruction, and at the same time we can get the data.

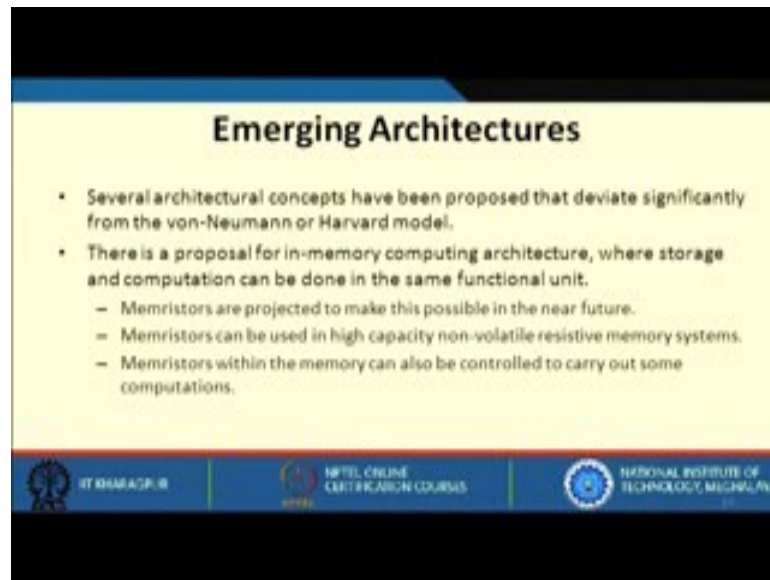
(Refer Slide Time: 19:19)



Let us see this kind of architecture is called Harvard architecture where we have separate memory for program and data. Instructions are stored in program memory and data are stored in data memory. So, we have a program memory and a separate data memory. Instruction and data access can be done in parallel; obviously, these are two different memories. So, the processor can access the program memory and the processor can also access the data memory at the same time. So, some of the microcontrollers and pipelines with separate instruction and data caches follow this concept. We will see what is

pipeline and we will also see that how separate instruction and data caches follow this approach. But here also this processor and memory bottleneck still remains. This is the processor; we will be accessing the memory, processor will be accessing the data, but multiple data cannot be brought in at the same time.

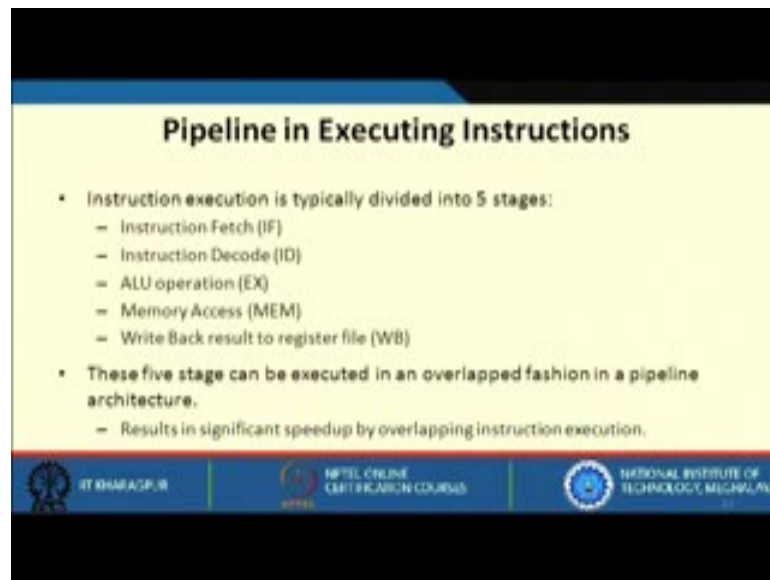
(Refer Slide Time: 22:44)



Now, people are talking about some emerging architectures as well. So, when they are talking beyond von-Neumann architecture. So, there is a proposal for in memory computing architecture where they say that both the storage and the computation can be done in the same functional unit. So, this is an emerging area, researchers are still working on it, looking into various aspects of it. It is projected that a circuit element called memristor; we have heard of resistor, capacitor, inductor. So, memristor is another circuit element which is projected to make it possible in near future. We have to wait for that, we cannot say at this point that of course, we will be coming with a non von-Neumann architecture in near future, but people are looking into it, people are thinking about it.

Memristors can also be used in high capacity non volatile resistive memory systems and can also be controlled to carry out some computation. So, memristor can be used as memory, memristor can also be used for logic computation. Because of these features we are saying that we can have some kind of in memory computing using memristor in near future.

(Refer Slide Time: 22:22)



**Pipeline in Executing Instructions**

- Instruction execution is typically divided into 5 stages:
  - Instruction Fetch (IF)
  - Instruction Decode (ID)
  - ALU operation (EX)
  - Memory Access (MEM)
  - Write Back result to register file (WB)
- These five stage can be executed in an overlapped fashion in a pipeline architecture.
  - Results in significant speedup by overlapping instruction execution.

IT BHARATPUR | NPTEL ONLINE CERTIFICATION COURSE | NATIONAL INSTITUTE OF TECHNOLOGY, BHARATPUR

Now, let us see that as I said that we have separate separate program memory and data memory. So, instructions are stored in instruction memory and data are stored in data memory. Pipeline is a concept that is used to speed up the execution of instructions, we will be looking into this in the later phase of this course, but just to give you a broad idea pipelining means overlap execution of instructions. An instruction execution is typically divided into five stages, the stages are; fetch the instruction, then we decode that instruction. After decoding that instruction, we execute that instruction. And after execution of that instruction, it may be required that we need to store the result into memory or we need to store the result into any of the processor register as well. So, that write back result to register file can also be done in the fifth stage.

So, an instruction execution cycle is basically divided into these five broad steps, instruction fetch, decode, execute, memory operation and write back. And these five stages can be executed in an overlapped fashion. We are never saying that we are doing parallel processing rather we are saying, we are doing some kind of overlapped execution of instructions. And this results in a significant speed up by overlapping instruction execution.

(Refer Slide Time: 24:27)

**Basic 5-stage Pipelining Diagram**

Instruction	Pipeline Stage						
1	IF	ID	EX	MEM	WB		
2		IF	ID	EX	MEM	WB	
3			IF	ID	EX	MEM	WB
4				IF	ID	EX	MEM
5					IF	ID	EX
Clock Cycle	1	2	3	4	5	6	7

IT BHARAGPUR | NPTEL ONLINE CERTIFICATION COURSE | NATIONAL INSTITUTE OF TECHNOLOGY, MICHAELBANK

Let us see how. So, these are the clock cycles and these are the five instructions that we are going to execute. First instruction is having fetch, decode, execute, memory operation, write back. So, once I have fetched an instruction it is now gone to the decode phase. Once I am decoding that first instruction, can I not fetch the next instruction. How, because decoding is performed in your processor and fetching is done from memory. So, we can overlap instruction decode and instruction fetch. So, when I am decoding the first instruction, the next instruction can be fetched.

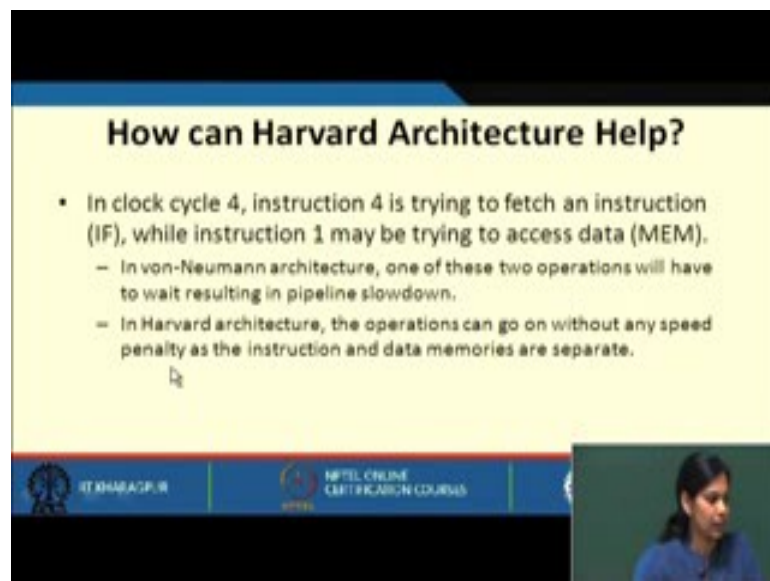
Similarly, when I am executing the first instruction, execution of that instruction happens in ALU. When I am executing this particular instruction then the next instruction can get decoded in the control unit. And parallelly the third instruction can be also fetched parallelly. So, we can see that fetching one instruction, decoding one instruction and as well as executing an instruction can all happen in parallel.

Let us move on with the next stage where we see that we are fetching an instruction - the fourth instruction; we are decoding the third instruction. We are executing the second instruction and we are doing some kind of memory operation for the first instruction. But you see if you are doing some kind of memory operation for the first instruction and you are fetching another instruction you cannot do this parallelly. Why because fetching can be performed in from the memory and some memory operation will also be done in the memory. But we can allow this to happen if we have Harvard kind of architecture.

Recall what I said in Harvard kind of architecture we have a program memory and you have a data memory. And now when you are fetching instruction, you are accessing the program memory. And when you are doing some memory operation that means, you are operating on data memory because what you are doing either you are writing something into after execution or even if you are reading something. So, both of this can happen in parallel if and only if you have such kind of architecture.

If you have an architecture where both the programs and the data are stored in single memory like this it would not have happened. So, this can only happen like fetching of an instruction and memory operation if you have Harvard kind of architecture. And what speed up we are gaining just see if these five instruction would take five cycles to execute then if one by one we want to execute it, but it would have been taken twenty five cycles, but now the first result is available after five and the next four result can be available after one by one. So, more four cycles will be required that is  $5 + 4 = 9$ , a total of 9 cycles will be required to get the result of all five instructions.

(Refer Slide Time: 28:57)



The slide is titled "How can Harvard Architecture Help?". It contains a bulleted list comparing von-Neumann and Harvard architectures. The von-Neumann architecture is shown to have a pipeline slowdown because one operation must wait for another. The Harvard architecture is shown to avoid this slowdown because instruction and data memories are separate. The slide also features logos for IIT KHARAGPUR and NPTEL ONLINE CERTIFICATION COURSES, and a small video inset of a woman in the bottom right corner.

**How can Harvard Architecture Help?**

- In clock cycle 4, instruction 4 is trying to fetch an instruction (IF), while instruction 1 may be trying to access data (MEM).
  - In von-Neumann architecture, one of these two operations will have to wait resulting in pipeline slowdown.
  - In Harvard architecture, the operations can go on without any speed penalty as the instruction and data memories are separate.

So, as I said how can this Harvard architecture actually help in clock cycle 4, as we have seen that instruction 4 was trying to fetch an instruction, while instruction 1 may be trying to access some data. In von-Neumann architecture one of these two operations will have to wait resulting in pipeline slowdown. But in Harvard architecture, what can

be done is both the operations can be done parallelly because we have separate data memory and separate program memory.

So, in this lecture, we have seen various software that are existing like application software, system software. The various kind of architecture that are existing, i.e. von-Neumann architecture, and the Harvard architecture, and how Harvard architecture actually helps in executing an instruction in a better and faster fashion.

Thank you.