Computer Organization and Architecture: A Pedagogical Aspect

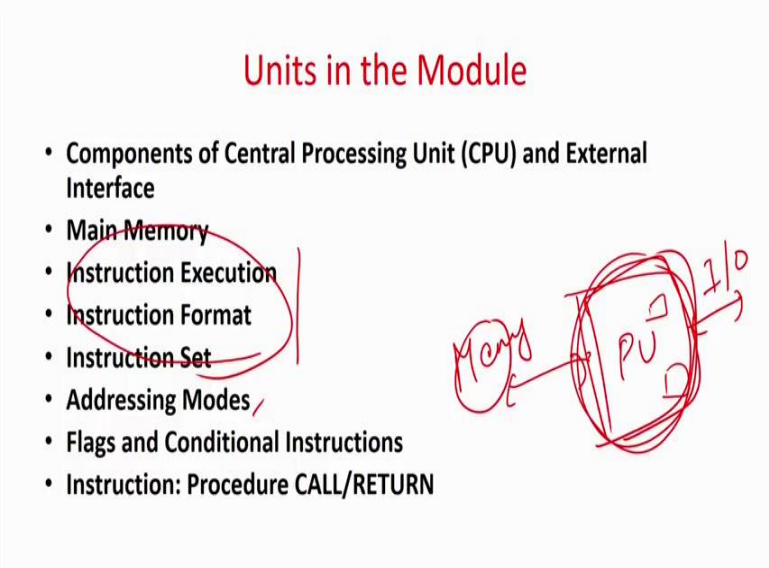
Prof. Jatindra Kr. Deka
Dr. Santosh Biswas
Dr. Arnab Sarkar
Department of Computer Science & Engineering
Indian Institute of Technology, Guwahati

Addressing Modes, Instruction Set and Instruction Execution Flow
Lecture – 07
Components of Central Processing Unit (CPU) and External Interface

Hello students, welcome to the next module which is on addressing mode instruction set and instruction execution flow. So, as you know this course is on computer organisation and architecture and will be are teaching in some pedagogical aspect. So, as already we have discussed in the last few lectures by professor Deka, that we will first start with a module and then will have the units, but before that we will also see what are the objectives of each of the module and then we will see what is the objectives of which of the units and then we will go for discussion on the elaborate units and then will see some question answers. So in fact, this whole course is on organisation, but from a pedagogical aspect.

So, this is this next module, which is on addressing mode, instruction set and instruction execution flow and I am doctor Santosh Biswas from department of computer science of IIT Guwahati and I will be covering many of the modules on the arithmetic and logic operations, memory organisations which are in internal to a CPU.

(Refer Slide Time: 01:34)



In fact, if you look at a computer organisation and architecture perspective we generally have the processing units at one side, we have the memory on one side and if you considered as a third part that should of the IO units. So, in fact I will be mainly looking at the central or the heart of the computer organisation and architecture that is the, that is the process, processing part.

So, the first unit on that part will actually give you an idea on the several aspects which is discussed in this slide that is they are actually the units of the module. So, the first module on this the main part of the computer organisation or the processing part of the computer architecture, will actually comprise of the central processing unit and then will have we look at the main memory which is also a part of the not actually as a part of the hard disks or the exact main memory which will be there will be more details in the subsequent units. But we will be mainly looking at the memory which will be the memory access techniques or how it is actually interacting with your central processing unit on that aspect. So, we will just look at the memory on a very abstract basis in these units, in terms like how the processor will actually talk to the memory and how it works.

We are not going into the details of the memory organisation in this particular units or modules, which will be dealt later. Main emphasis of this units of this modules will be instruction set, instruction format and how instructions are executed, that is given a C code or given any high language, high level language code it is converted into assembly language code or machine language, machine language code. Then actually it is executed by the processing unit of your computer or computer or your central processing unit and in that way.

So, our main focus will be how such a high level code is basically translated or once you got a machine language or high level code translated into an instruction which is understand by your central processing unit and how actually it executes and what are the instructions, how they look like. For example, if we have A = B + C is a high level language, how we can express? It is a language which is understood by the central processing unit and if such a code or if a code is written in that language that is in terms of instructions how it is executed by your central processing unit will be covered actually by these three modules.

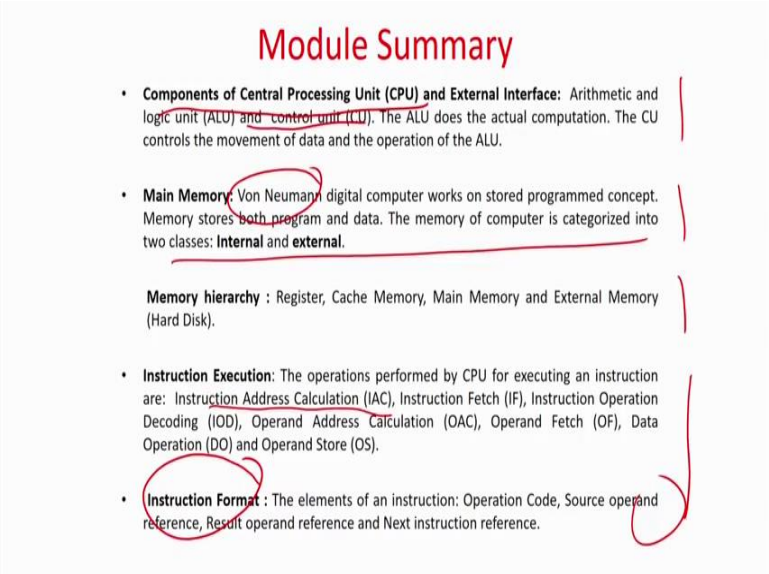
Then we will go for something called addressing modes that how you can have, how you can address or how an instruction address or how a in other words how an instruction executes on different type of data. A data can be in the memory, are which is may be a main memory which

is the outside processing unit, a data can be also inside a very small called a cache memory which is inside the processing unit or data can also be in a register so all this things will cover up. So, how a how an instruction can take care of all the data which is available in different parts or different aspects of the memory.

So, on that terms how can you classify what are the different type of instructions. So, we look over there and finally, in the last two units we are going to cover certain instructions which actually required, which are not a very procedural way of executing the code like for example, we can think that instruction 1 then instruction 2 and so forth.

But sometimes when you want to execute a jump instruction or whether you want you want to execute a function and then return from the function. So, specialised type of instructions are required for which flags and conditional instructions and call procedure and return, these are the two modules which will focus on such type of an instruction by which you can implement jump conditional, conditions of your code and also how can how can you implement a function or a procedure. So, that is the basic idea of this module and the units as divided into the form which is stored in the slide, which in which basically in a very broad term we will try to show how basically a code is executed or how a code can be written, which exactly which exactly executes on the central processing unit of your processor.

(Refer Slide Time: 05:49)



So, as I told you the whole course is on pedagogical aspect. So, already have discussed a brief on the units which will be covered in to the module. So, module summary will be something

like this; we will first study the components of the central processing unit and the CPU and the external interface. In this case basically we will try to cover up the arithmetic and logic unit and the control unit of a processor that is the central part of the processor as I told you generally have a mathematic, arithmetic and logic unit and there is a control unit.

So, first we will study on this, next we will next will look at what is a main memory, we will give the very basic idea of a memory in this module because the other modules will be dedicated to memory. Here we will just give the idea that what is a memory, what is stored in the memory and has already discussed in the previous module about Von Neumann architecture. So, you are all now adopting the Von Neumann architecture. So, our data as well as code is both stored in the memory.

So, before we can even understand how a code or how a machine language code exist in the processor, we need to know the basic idea of a main memory what is an internal memory, what is an external memory, how data is organised over there and how it is basically accessed by the processor because in case of Von Neumann architecture the pro for the processor to run both the instruction as well as the data has to be taken from the memory. So, this unit, this module will give you a very brief overview of what is the main memory, how it is organised and why memory is required to execute your instructions.

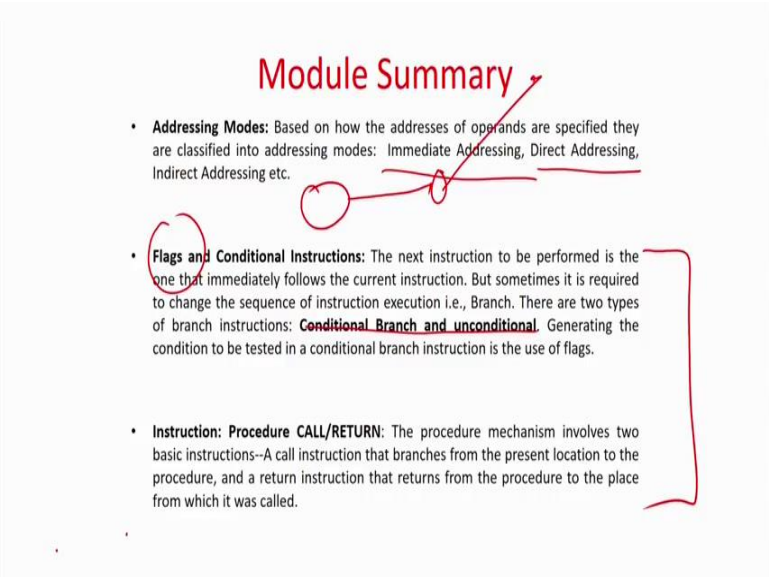
Then we will talk about the memory hierarchy in a very broad light, I mean more will be detailed will be taught when we will cover to the when we will cover the units on modules on memory architecture. Main emphasize emphasis of this module is on instruction execution, instruction formats and instruction set that is if you want to execute an instruction what are the different phases. So, in fact actually it first actually calculate for example, it has to execute certain operation of an add.

So, first it has to find out the instruction address calculation that is means, when the instruction is loaded into the memory that you want to execute. So, the instruction address has to be calculated, then the instruction is fetched from the memory then actually you fetch the, what are the operation it is want to do that has to be decoded and then you have to find out on what operands it want to operate, that we have to find out then you have to bring the operands from the memory execute and store it.

So, next will be have a detailed look on how instructions are executed, then we look at how a instruction basically looks like because it should have an op code which will tell what to do

whether it is addition, multiplication, subtraction, store then we will have to tell for on what will operate, where is the data? The data is in the instruction itself the data has to be fetched from the memory; data has to be fetched from the register so we will discuss about the different instruction formats.

(Refer Slide Time: 08:36)



And then we will again look at on elaborate mode on what are the different modes of addressing like where is the data? Is the data directly in the instruction, whether the data can be directly accessed from the address which is given in the instruction or whether it is an indirect mode that is a in terms, you can think in terms of pointer of a C language that the address given in the instruction is corresponding to one memory location and in that memory location will point to some other where the data will be there.

So, main emphasis of this module as I told you is basic, how the instruction actually executes and what are the different instruction types and what are the instruction formats. So, we will cover of that and the last two part again what are the conditional instructions because most of the instructions basically work in steps, first instruction, second instruction, third instruction and so forth, but generally if you want to execute a conditional instruction then you have to go for a jump.

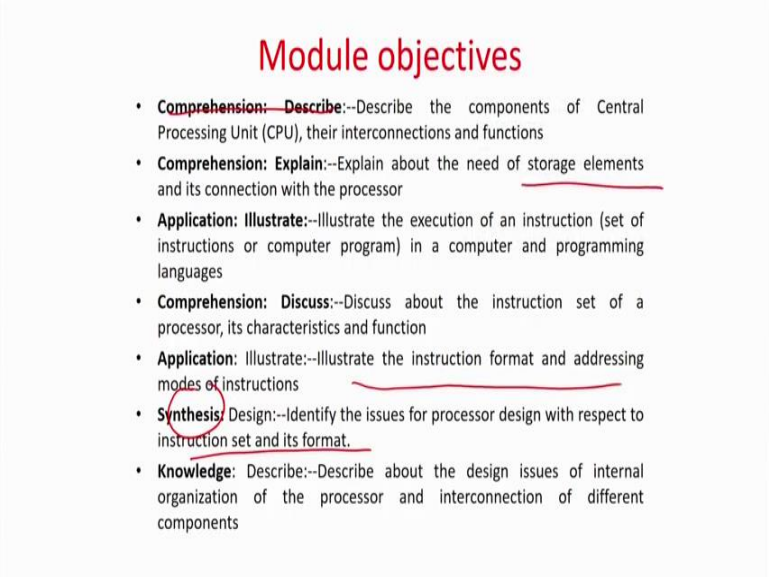
So, there are actually jump or branching instruction, in fact there are 2 type of instructions branching instruction conditional and unconditional. Conditional means it will depend on some values of some variables and unconditional means you have to just jump with respect to

another. So, one unit will be dedicated on how such instructions are executed, it depends on what how such instructions what are the format, how it looks like and how they are controlled like for example, if I say a conditional branch it will depend on something.

So, actually there are something called flag variables or some flag registers which gets set or reset by some operations and then your conditional instruction will check the flag and jump if the flag is true or sometimes the if it is false it will keep on executing in the sequential way, unconditional branches do not have anything of that nature the just a jump. So, one unit will be dedicated for that and finally, one unit will dedicated for how to write a procedure and how to jump to the procedure after you complete it, again you come back to from where you have started.

So, this is unit will be dedicated to that. So, when you as you all know when you start an instruction for a procedure you have to back up whatever was the value of the present values of the memory, registers etcetera. So, that when you come back after executing the procedure you have to again regain those values and start the operation. So, one unit will be this one, on the instruction following procedure.

(Refer Slide Time: 10:58)



Now, again as this is a pedagogical method, we have to tell you that what are the precise module objectives that after you complete this module all the units of the module what you are going to achieve. So, first is you will be is a comprehension objective that is comprehension objective describe, you will be able to precise the components of a central processing unit their

interconnection and functions. Secondly, you will be able to explain the need of storage elements and its connection with the processor. Storage element in fact external memory, registers different types of registers, some of the registers is a storage which is inbuilt in the processor. So, why such registers are required you will be able to explain properly.

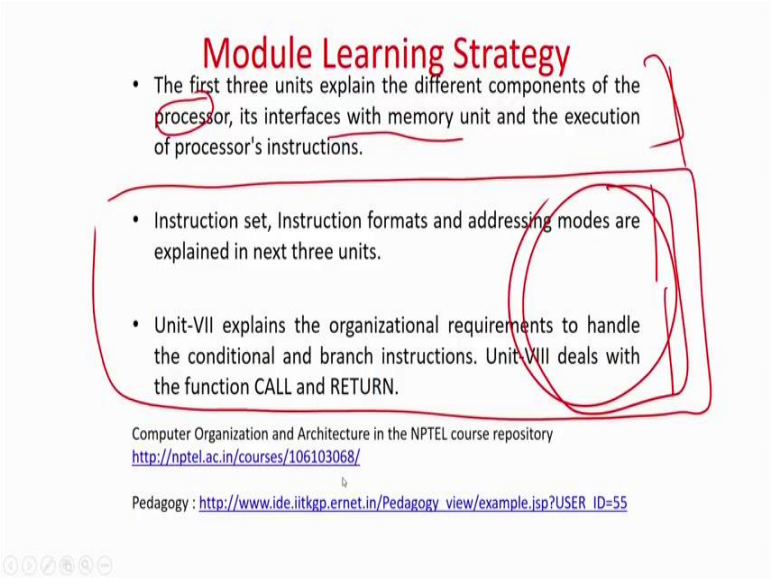
Then next is application based on what you have learned, you will be able to illustrate the execution of an instruction in a computer and programming languages. So, that if I give you a simple programming like you have to add 3 numbers then you will be able to illustrate with an example how such a program which is written in instruction set of a processor will be executed.

Comp next objective is comprehension, you will be able to discuss about the instruction set of a processor its characteristics and what are the functions. That is given a processor you will be able to tell what will be the instruction set it supports, basically what are the characteristics what is the functionality etcetera. Also as an application of this you will be able to illustrate the instruction format and addressing mode that is what are the different parts of an instruction? What is its format, how it addresses the operands and where it stores the values? So, you will be able to illustrate with different examples.

Next important objective is synthesis, that you will be based on all the knowledge you will be able to design that is you will be able to identify the processor design with respect to instruction set and its format. That is you will be quite easy for you to design an identify the issues, that is how you can design a processor if the instruction set and its formats are given, what are the issues how you can make it efficient etcetera. Finally, final objective is you will be able to describe about the design issues of internal organization of the processor, interconnection of different components, that is your central processing unit is not only about your arithmetic and logic operation and all the instructions, they have to be connected to the memory they have to be connected to some I/O devices etcetera.

So, how internally they will be connected. So, that part also you will be able to describe. So, these are the basic objectives of this module, once you do all the units and solve the problems you will be able to precise achieve this module objectives. As again as this a pedagogical based architecture so, we are going to precisely quantify that after doing this module you will be able to achieve this objective.

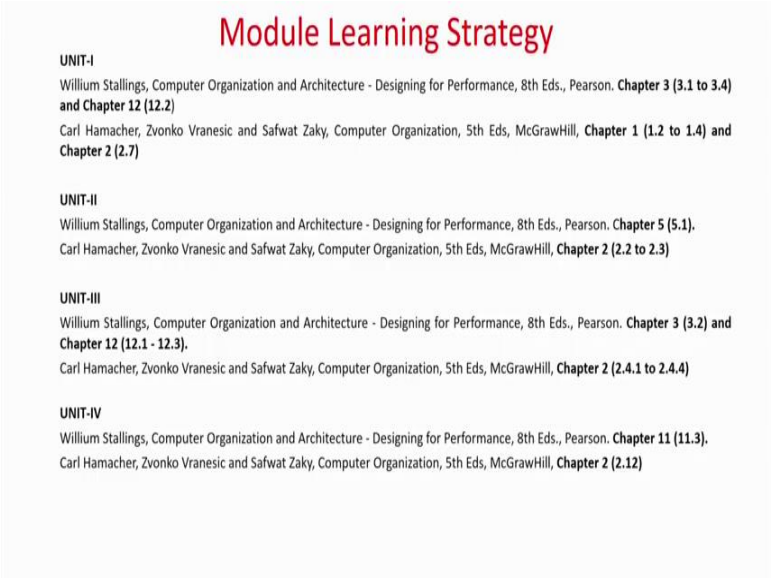
(Refer Slide Time: 13:33)



Then we will be very shortly tell you how to learn and how to go about this module, how to able to learn it effectively. So, the first 3 modules explain the components of processors, its interfaces with memory unit and execution of processor instructions. So, as I told you this will be only a very overview because dedicated modules will be given for the processor design, interfaces and memory. But it will give you an idea that given a because without telling how a processor is interconnected to a memory, what is the broad architecture of a memory I cannot tell you how a code basically executes in a memory. The main idea of this module is instruction set formats, addressing modes which are described in 3 units and then how to handle branch instructions and call and return instructions.

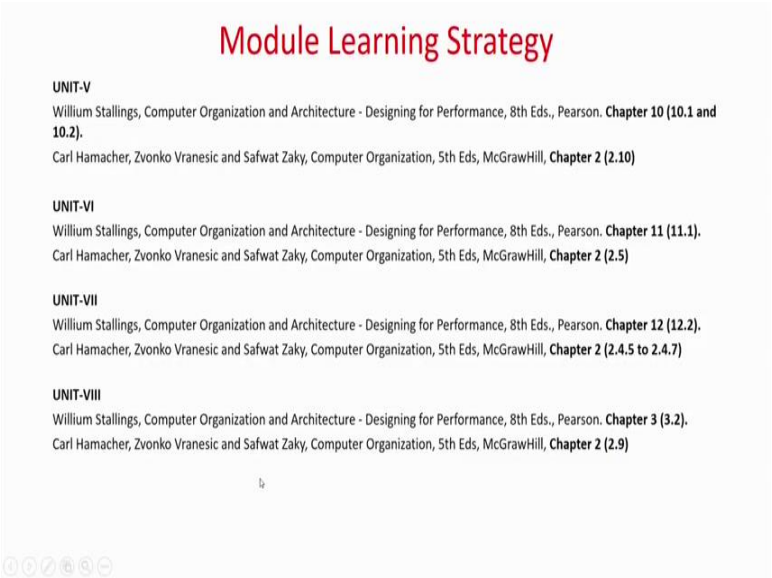
So, this basically these are the main emphasis of this module where you will tell about different instructions, formats, how it executes, some complex instructions like jump and procedure, but to know that you have to understand the different broad architecture in a very brief manner about the processor, this a central processing unit, how it interconnects and broad idea of a memory. So, mainly we have to read in details on this one and some overview of this is enough. So, 2 important links are, we have we have been given a link on pedagogical methods which will tell you about this and one NPTEL video sorry, NPTEL web course id is there.

(Refer Slide Time: 18:53)



Apart from that this slides please keep record we are giving references of the standard Williams Stallings book on computer organisation architecture and Hamacher's book which like unit 1, unit 2, unit 3, unit 4 and so on from which chapters, which page number all details are given over here.

(Refer Slide Time: 15:06)

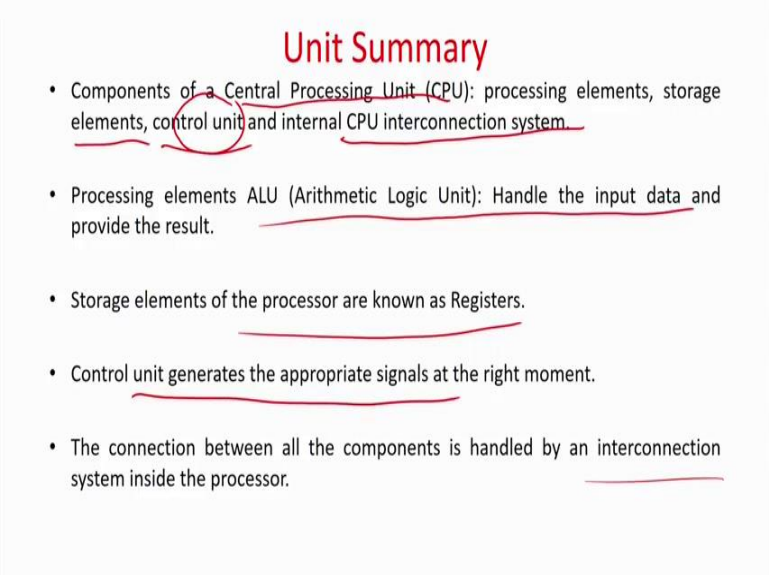


So, after you listen to the lectures you are expected to go through these relevant pinpointed references and then you can start solving some of the problems which will be discussed at the end of the module as well as end of the units. But the basic module strategy is that the first 3

units has to be learned in a overview manner and the other 6 lectures has to be done that is in next 2 points has to be understood in depth because that is the whole idea of this unit module sorry.

So, with this we have built enough background for this module which is on addressing mode, instruction set, and instruction execution flow. Now what we are going to do? We are going to start the first unit of this module and the unit is component of a central processing unit CPU and its external interface, as I told you this unit will deal with central processing unit and external interfaces in a very broad manner because our main job is to understand how a code executes in a processor. So, but this one is the prerequisite to know about that know about how a code executes. So, again we have all seen what was our module what was module summary, now which module is consisting of some units which have already discussed.

(Refer Slide Time: 16:25)



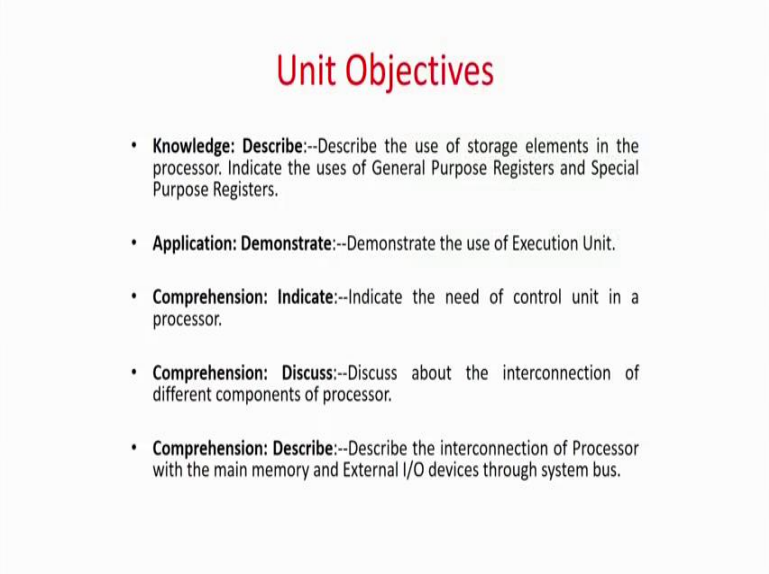
So, now we will go to the unit summary and then will go ahead. So, this unit will specifically focuses on what are the units of a central processing unit, the storage elements that is memory there is control unit and interconnection system. So, processing unit will do all your computation, storage element is a memory because you are using a von Neumann architecture and how to control that is when data will flow from memory, when to processor that will be controlled by a unit which is called a control unit and to connect everything there is a interconnection system or a bus system.

Then we will deal with the processing element that is the arithmetic and logic unit, then we will study about the storage element which is mainly the registers because unless I tell you what is a main memory I cannot define what is a register. Basically register is something which is a memory which is inbuilt to the processor; main memory is a memory which is external to the processor.

Then we will see the idea of control unit, how it generates control signals at the right moment to control the data flow and finally, we will also see about the interconnection networks because we have certain elements like processing, storage, control unit how they are interconnected among this one. So, this is the summary of the module and we will do it in a very broad manner because the idea of this module is to study how an instruction basically executes.

But this is the prerequisite, then what is the objective of this unit. So, this unit is again I told you is a broad overview to understand how the code executes.

(Refer Slide Time: 17:46)



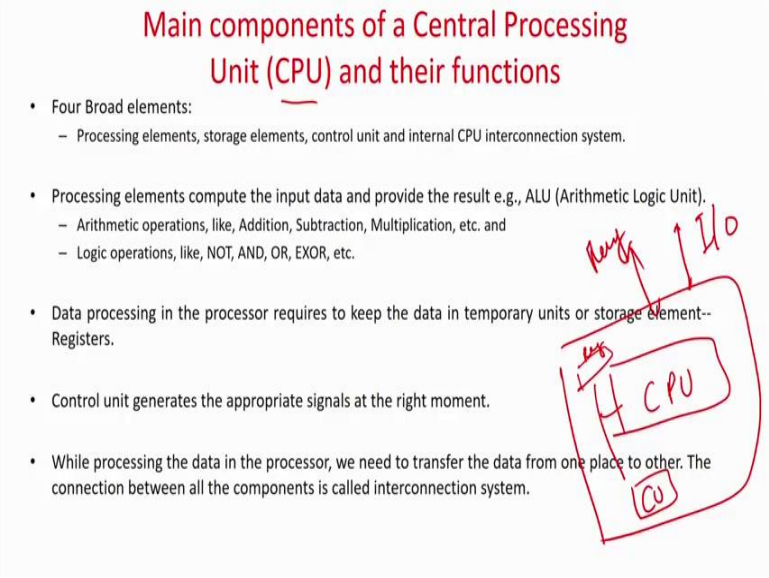
So, you will be to the, mostly the objectives are knowledge base, application base and comprehension base that is they will more or less on recall or knowledge kind of a analysis objective. In this case this is not more on this is not mainly focussed some kind of a design or implementation object, it is mostly based on developing a knowledge and comprehension. The first objective is you will be able to describe the use of storage elements in the processor, what are the uses of different registers like a general purpose register and a special purpose registers.

Why are this memory basically the registers are required, then you will be able to demonstrate the use of an execution unit that is a small demonstration like if I have to add 2 numbers, if I had to run a small loop how basically an execution unit basically looks at it. Then we will indicate you will able to indicate the need of a control unit why a control unit is needed.

Then you will able to describe about interconnection of different units of the processor, why it is required how they are interconnected that we will be able to discuss and finally, we will be able to describe about the interconnection of processors through a bus, they are not basically done through wires why it is a bus what is the modulation those things you will be able to describe after you complete this unit.

So, as again repeating this unit basically is more of a recall type of a unit where you have to where your knowledge will be build and not much of application because application build based for this module will be on instruction execution in a processor. So now let us start the unit.

(Refer Slide Time: 19:14)



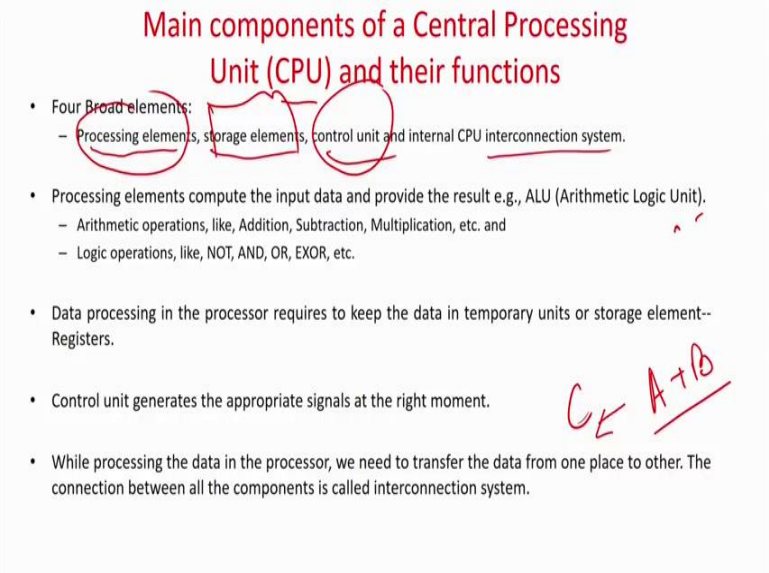
So, what are the main components of a central functioning unit or central processing unit we generally call as the CPU and their functions? So, basically as I told you so generally this is your CPU, internal you have lot of stuff generally if I draw bigger box. So, you have your I/O modules here, you have your memory here that is your main memory, inside this central processing unit and generally we will have a controller unit. I will take a more detailed figures and some memory, memory banks basically which I actually call registers, which are internal

to the central processing unit and the, but again all these things has to be connected by a bus. So that is what is the basic architecture.

So, again as a differential when we talk of a memory it's an external it's a RAM or its your we call the main memory, but when we are talking about the central processing unit it is the register, they are for the time being they are more or less similar, but one is external and one is internal. So, for this we will whenever we need to make a differentiation we will tell you otherwise we will call it by memory, but for this unit you have to think that it is nothing, but a register.

So, broadly 4 different type of elements are there, one is a processing element, one is a storage element, control unit and interconnection system, what is the processing unit?

(Refer Slide Time: 20:29)



See I take an example that, I have to add A + B, this is what I have to do. So, processing is nothing in this case, but is the addition, the not only addition you have to take the value variable A, what is the value of A what is the value of B and you have to store it in C. So, the processing is lots of steps here, it's not only the addition processing is you have to add it before that you have to bring the variables A and bring the variable value B and store it in C. So, bringing the values adding it, storing it, everything actually done by something called a processing element which is your arithmetic and logic unit.

So, arithmetic means add, logic unit means some logic operations like jump etcetera and then also actually the processing elements are also responsible for storing the data, fetching the data like you have to bring the data A, you have to bring the data B from the main memory because the data and everything is stored in the main memory and actually you have to store back the result in memory location C.

Then actually what is the storage element, whenever we talk of a processor generally the data is in the main memory and whenever you want to operate in that processor then of the central processing unit it has to be done on the registers because, I am whenever we will go to more depth we will be more specific for a time being it is safe to assume that the central processing unit will operate only on the data which is in the register. So, you have to bring if A and B are 2 main memory locations as I told you main memory is external to the central processing unit. So, whenever you have to bring the data for variable A and variable B you generally bring it to the processor register.

As I told you the very simple boundaries main memory outside the CPU, registers memory locations, memory only, internal to the CPU. So, CPU does not directly interact with the memory, for the time being quote unquote assuming then the value of A and B has to be first brought into the storage elements that is the registers in this case and then you do the computation, the processing element and then you again store back the result from the register to the memory location C.

So, now what is the control unit doing, control unit is generating the control signals at appropriate time, what do you mean by that? See for example, I do not have a adder asset I have a arithmetic and logic unit. So, it has to be configured that whether I want to add, whether I want to subtract, whether I want to multiply. So, different signals has to be generated that now I want to add, now I want to fetch the value of A to a register.

So, different modes of configurations or the signals for those are generally done by a unit which is called the control unit and of course, you require an interconnection system because data has to be brought from the main memory to the register or connection from main memory to the register. From the register you have go to the ALU arithmetic and logic unit where you add, it the ALU will again write back to the data to another register. So, another set of interconnections and finally, the register will write the value of memory location C. So, another set of interconnections. So, that is why there is a big set of interconnection among the internals.

So, now as I told you the processing element compute input data and provide the result that is the basic arithmetic and logic unit. So, they lot of arithmetic operations like addition, multiplication, subtraction, logical operations this way. Data processing in the processor requires to keep the data in temporary units or storage elements called registers. So, this has to be remembered, that when I tell about the registers or when I tell about this as memory in internal to the CPU basically it is nothing but they are something called a registers.

So, registers are as I told you is the memory location or memory when you think about a internal processing system or internal inter the CPU, but generally the name is registers and they generally tell that we are writing some temporary variables over here. So, why do we call it temporary variables it is very important that they call it temporary variables because say that I want to get the value of A to the register, I get the value of B to another register, add it and then put it to C, but A + B the value will be stored in the register and they say that it is a temporary value. Because the final value of C is valid when that data is return to the memory location C. So, therefore, and some computation like if you want to add A + B + C + D.

So, generally you add A + B store in a register then you bring C, then you add the temporary value of A + B with C, then you put A + B + C another register to get the value of D and so forth. So, all the intermediate results are stored in a in the registers therefore, they call it temporary data and as I told you the control signals will tell what to do when. So, mainly there are lot of multiplexing arrangements and all those stuff so the control signals will generate for that and as I told you when processing data after the so many data movement will be there so you require a connection between the components which is called the interconnection of stuff.