Computer Organization and Architecture: A Pedagogical Aspect

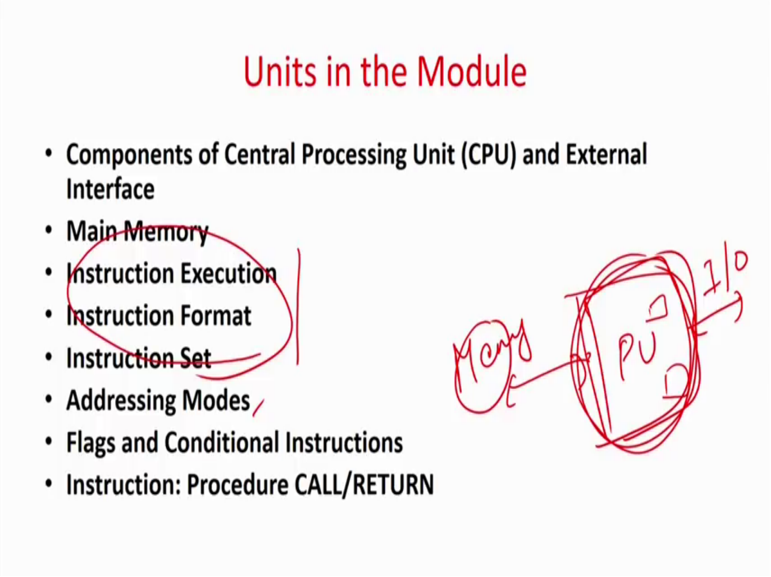
Prof. Jatindra Kr. Deka
Dr. Santosh Biswas
Dr. Arnab Sarkar
Department of Computer Science & Engineering
Indian Institute of Technology, Guwahati

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

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 will first start with a module and then will have the units, but before that will also see what are the objectives of each of the module and then will see what is the objectives of which of the units and then will go for discussion on the elaborate units and then will see some question answers so. In fact, this whole course is an 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 intern to a CPU.

(Refer Slide Time: 01:34)



Units in the Module

- Components of Central Processing Unit (CPU) and External Interface
- Main Memory
- Instruction Execution
- Instruction Format
- Instruction Set
- Addressing Modes
- Flags and Conditional Instructions
- Instruction: Procedure CALL/RETURN

In fact, if you look at a computer organisation and architecture perspective which 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 there actually the units of the module. So, the first module on this the main part of the computer organisation of 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 up 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 subsequent units. But will be mainly looking at the you the memory which will be the memory access techniques or how it is actually interacting with your central processing unit on that aspect. So, you will just look at the memory on abstract basis in this 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 detailed later. Main emphasis of this units of this modules will be an instruction set, instruction format and how instructions are executed, that is given of C code or given any high language, high level language code it is a 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.

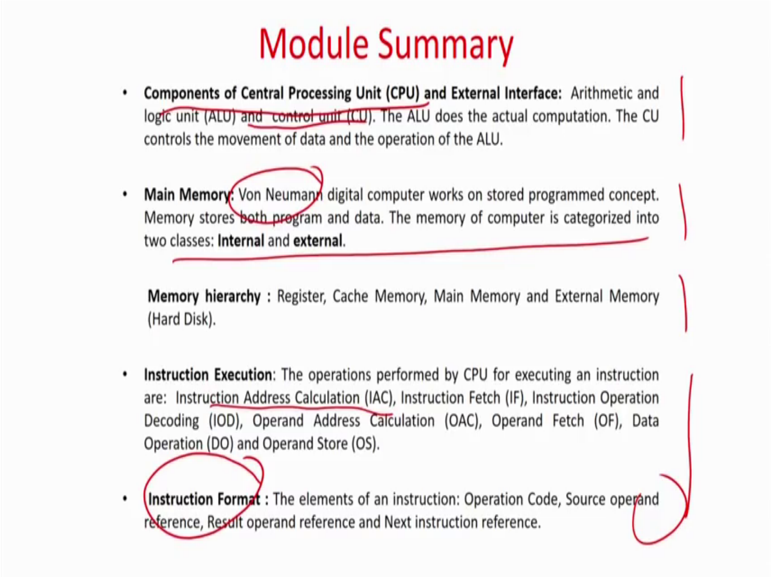
So, our main focus will be how such a high level code is basically translated or once you got a machine language or high level code translated into an instruction which is understand by your central processing unit and how actually it executes and what are the instructions, how they look like. For example, if you have a equal to B plus C is a high level language, how you 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 this 3 modules.

Then we will go for something called addressing modes that how you can have, how you can address or how a instruction address or how a in other words how a instruction executes on different type of data. Are 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 are 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 looking look over there and finally, in the last two units we are going to covered up certain instructions which actually required, which are not a very procedure 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 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 a 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 will try to show how basically a code is executed or how a code can be written, which exucally which exactly executes on the central processing unit of your processor.

(Refer Slide Time: 05:49)



Module Summary

- **Components of Central Processing Unit (CPU) and External Interface:** Arithmetic and logic unit (ALU) and control unit (CU). The ALU does the actual computation. The CU controls the movement of data and the operation of the ALU.
- **Main Memory:** Von Neumann digital computer works on stored programmed concept. Memory stores both program and data. The memory of computer is categorized into two classes: **Internal** and **external**.

  Memory hierarchy : Register, Cache Memory, Main Memory and External Memory (Hard Disk).
- **Instruction Execution**: The operations performed by CPU for executing an instruction are: Instruction Address Calculation (IAC), Instruction Fetch (IF), Instruction Operation Decoding (IOD), Operand Address Calculation (OAC), Operand Fetch (OF), Data Operation (DO) and Operand Store (OS).
- **Instruction Format :** The elements of an instruction: Operation Code, Source operand reference, Result operand reference and Next instruction reference.

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

So, first will study on this, next will next will look at what is a main memory, will give the very basic idea of a memory in this module because the other modules will be dedicated to memory. Here 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 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 access 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.

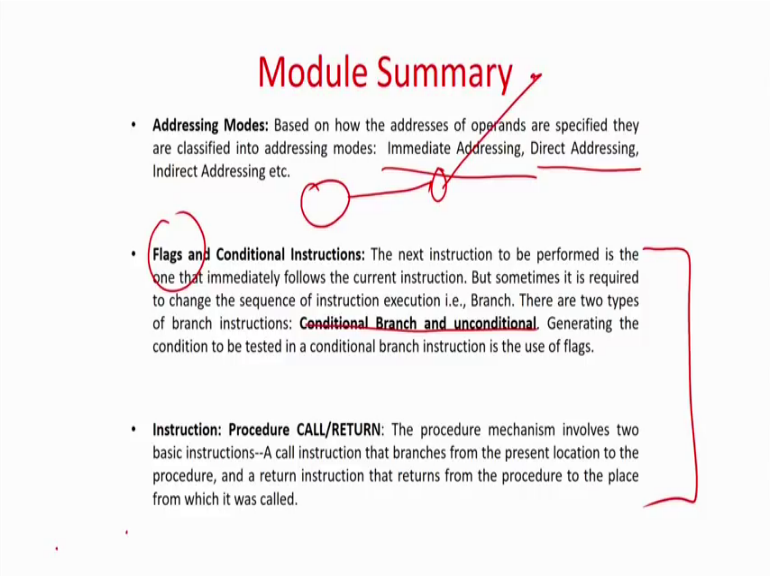
So, this unit, this module will give you a very brief overview of what is the main memory, how it is organised and why memory is required to execute your instructions.

Then will talk about the memory hierarchy in a very broad right, I mean a more will be detailed will be taught when will cover to the when will cover the units on modules on memory architecture. Main emphasize emphasis of this module is an 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 place to from the memory then actually you fetch the, what are the operation 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 off code which will tell what to do whether it has addition multiplication subtraction store then will have to tell for on what will operate, where is the data? The data is in the instruction itself the data is to be fetch from the memory; data has to be fetch from the register so will discuss about the different instruction formats.

(Refer Slide Time: 08:36)



Module Summary

- **Addressing Modes:** Based on how the addresses of operands are specified they are classified into addressing modes: Immediate Addressing, Direct Addressing, Indirect Addressing etc.
- **Flags and Conditional Instructions:** The next instruction to be performed is the one that immediately follows the current instruction. But sometimes it is required to change the sequence of instruction execution i.e., Branch. There are two types of branch instructions: **Conditional Branch and unconditional**. Generating the condition to be tested in a conditional branch instruction is the use of flags.
- **Instruction: Procedure CALL/RETURN**: The procedure mechanism involves two basic instructions--A call instruction that branches from the present location to the procedure, and a return instruction that returns from the procedure to the place from which it was called.

And then 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 where the data can directly access 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.

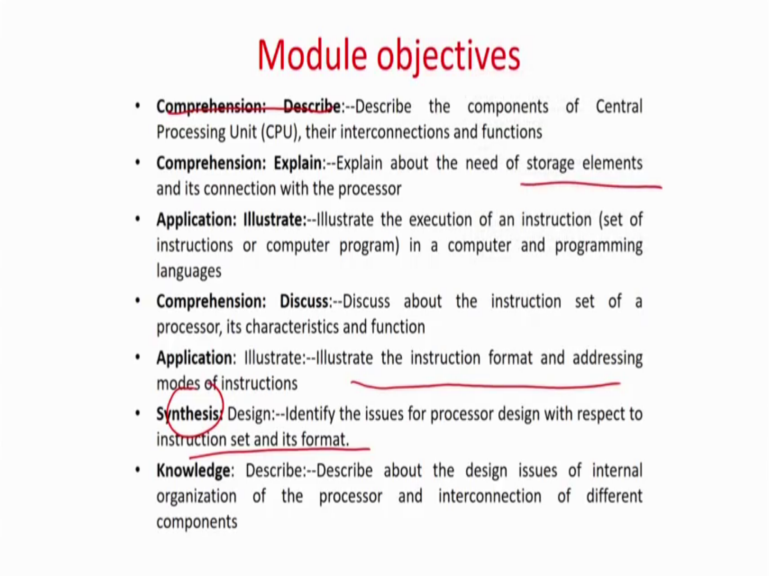
So, main emphasise of this module as I told you is basic, how the instruction actually executes and what are the different instruction types and what are the instruction formats. So, will cover of that and the last two part again what are the conditional instructions because most of the instructions 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, infact there are 2 type of instructions branching instruction conditional, unconditional, conditional means it will depend on some values or 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 something called flag variables or some flag registers which gets set or reset get 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 execute 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 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 register 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 pollen procedure.

(Refer Slide Time: 10:58)



Module objectives

- **Comprehension: Describe**:--Describe the components of Central Processing Unit (CPU), their interconnections and functions
- **Comprehension: Explain**:--Explain about the need of storage elements and its connection with the processor
- **Application: Illustrate:**--Illustrate the execution of an instruction (set of instructions or computer program) in a computer and programming languages
- **Comprehension: Discuss**:--Discuss about the instruction set of a processor, its characteristics and function
- **Application**: Illustrate:--Illustrate the instruction format and addressing modes of instructions
- **Synthesis:** Design:--Identify the issues for processor design with respect to instruction set and its format.
- **Knowledge**: Describe:--Describe about the design issues of internal organization of the processor and interconnection of different components

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

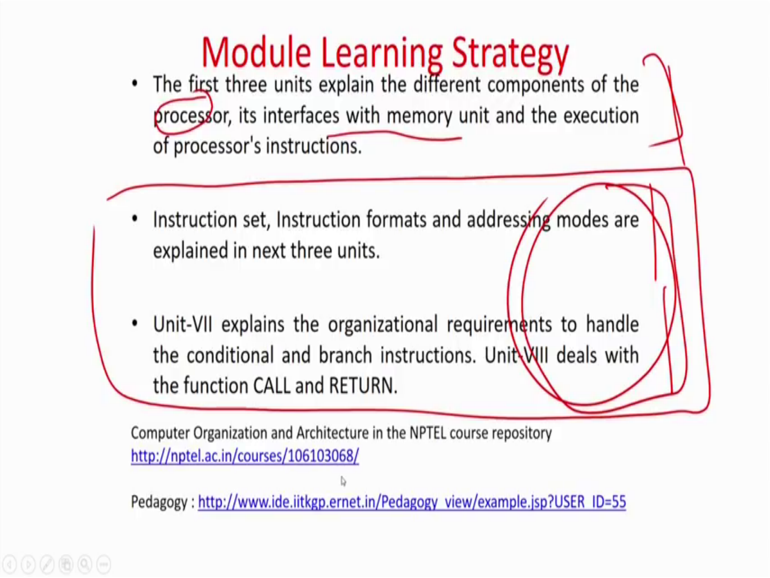
Then next is an application based on what you have learn, you will be able to illustrate the execution of an instruction in a computer and programming languages. So, that if I give you a simple programming like you have to add 3 numbers then you will be able to illustrate with an example how such a program 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 it characteristics and what are the functions. That is given a processor you will be able to tell what will be the instruction set is 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 it 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 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 purpose also will be able to describe. So, these are the basic objective 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 precise quantify that after doing this module you will be able to achieve this objective.

(Refer Slide Time: 13:33)



Module Learning Strategy

- The first three units explain the different components of the processor, its interfaces with memory unit and the execution of processor's instructions.
- Instruction set, Instruction formats and addressing modes are explained in next three units.
- Unit-VII explains the organizational requirements to handle the conditional and branch instructions. Unit-VIII deals with the function CALL and RETURN.

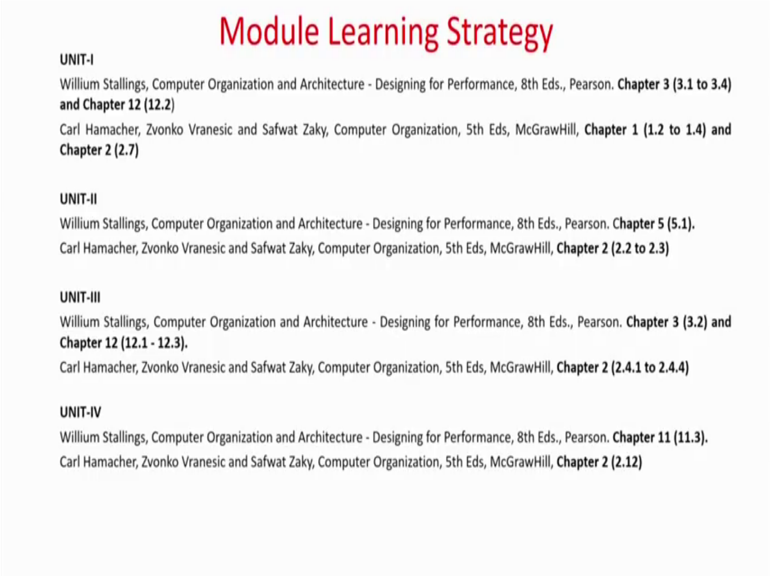
Computer Organization and Architecture in the NPTEL course repository
http://nptel.ac.in/courses/106103068/

Pedagogy : http://www.ide.iitkgp.ernet.in/Pedagogy_view/example.jsp?USER_ID=55

Then we will be very shortly tell you how to learn and how to go about these module, how to able to learn it effectively. So, the first 3 modules explain the components of processes is interfaces with memory unit and execution of process 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 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, these basically these are the main emphasize of this module where we tell about different instructions, formats, how it executes, some complex instructions like jam and procedure, but to know that you have to understand the different broad architecture in a very brief manner of the processor, this a central processing unit how it interconnect 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, MPTEL web course id is there.

(Refer Slide Time: 18:53)



Module Learning Strategy

UNIT-I

Willium Stallings, Computer Organization and Architecture - Designing for Performance, 8th Eds., Pearson. **Chapter 3 (3.1 to 3.4) and Chapter 12 (12.2)**

Carl Hamacher, Zvonko Vranesic and Safwat Zaky, Computer Organization, 5th Eds, McGrawHill, **Chapter 1 (1.2 to 1.4) and Chapter 2 (2.7)**

UNIT-II

Willium Stallings, Computer Organization and Architecture - Designing for Performance, 8th Eds., Pearson. **Chapter 5 (5.1).**

Carl Hamacher, Zvonko Vranesic and Safwat Zaky, Computer Organization, 5th Eds, McGrawHill, **Chapter 2 (2.2 to 2.3)**

UNIT-III

Willium Stallings, Computer Organization and Architecture - Designing for Performance, 8th Eds., Pearson. **Chapter 3 (3.2) and Chapter 12 (12.1 - 12.3).**

Carl Hamacher, Zvonko Vranesic and Safwat Zaky, Computer Organization, 5th Eds, McGrawHill, **Chapter 2 (2.4.1 to 2.4.4)**

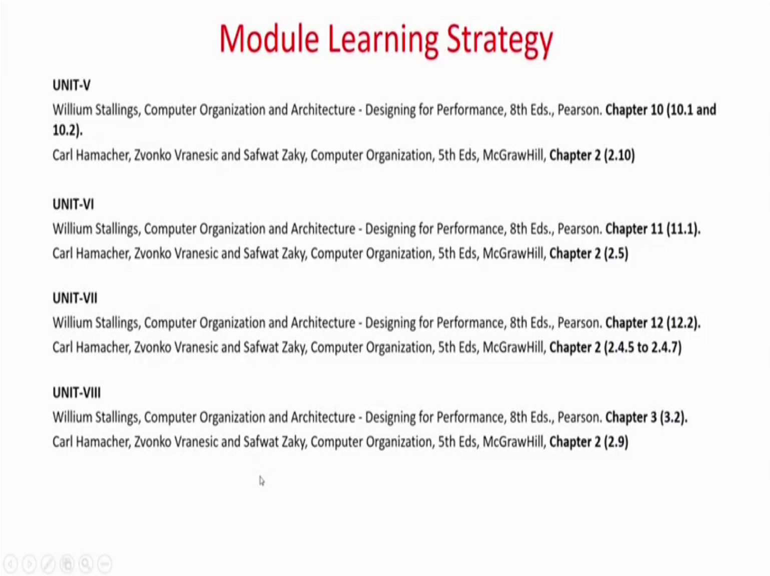
UNIT-IV

Willium Stallings, Computer Organization and Architecture - Designing for Performance, 8th Eds., Pearson. **Chapter 11 (11.3).**

Carl Hamacher, Zvonko Vranesic and Safwat Zaky, Computer Organization, 5th Eds, McGrawHill, **Chapter 2 (2.12)**

Apart from that this slides please keep record we are giving references of the standard Williams Stallings book on computer organisation architecture and hamactures 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)



Module Learning Strategy

UNIT-V

Willium Stallings, Computer Organization and Architecture - Designing for Performance, 8th Eds., Pearson. **Chapter 10 (10.1 and 10.2).**

Carl Hamacher, Zvonko Vranesic and Safwat Zaky, Computer Organization, 5th Eds, McGrawHill, **Chapter 2 (2.10)**

UNIT-VI

Willium Stallings, Computer Organization and Architecture - Designing for Performance, 8th Eds., Pearson. **Chapter 11 (11.1).**

Carl Hamacher, Zvonko Vranesic and Safwat Zaky, Computer Organization, 5th Eds, McGrawHill, **Chapter 2 (2.5)**

UNIT-VII

Willium Stallings, Computer Organization and Architecture - Designing for Performance, 8th Eds., Pearson. **Chapter 12 (12.2).**

Carl Hamacher, Zvonko Vranesic and Safwat Zaky, Computer Organization, 5th Eds, McGrawHill, **Chapter 2 (2.4.5 to 2.4.7)**

UNIT-VIII

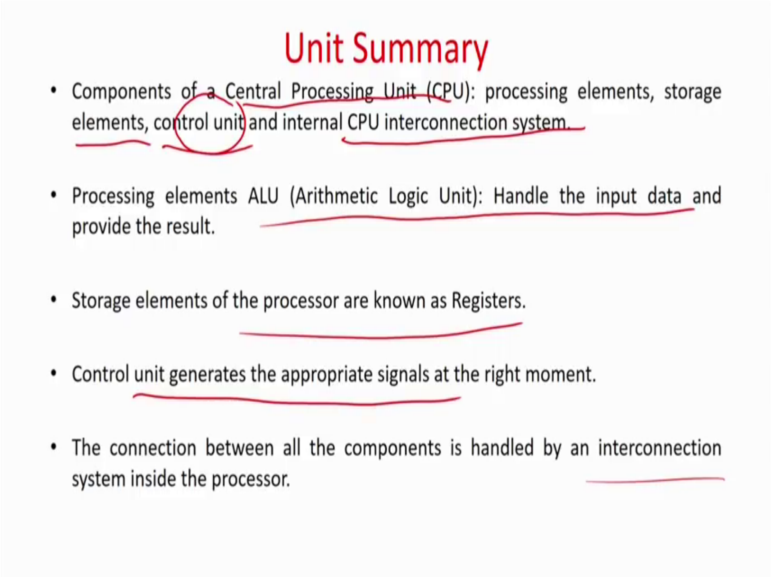
Willium Stallings, Computer Organization and Architecture - Designing for Performance, 8th Eds., Pearson. **Chapter 3 (3.2).**

Carl Hamacher, Zvonko Vranesic and Safwat Zaky, Computer Organization, 5th Eds, McGrawHill, **Chapter 2 (2.9)**

So, after you listen to the lectures your expected to go through this relevant pin pointed references and then you can start solving some of the problems which will be discuss at the end of the module as well as end of the units. But the basic module strategies is that the first 3 unit has to be learn in a overview man 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 I have built a background for this module which is on addressing mode instruction set, an 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 it external interface, as I told you this unit will tee deal with central processing unit and external interface 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 unit which have already discussed.

(Refer Slide Time: 16:25)



Unit Summary

- Components of a Central Processing Unit (CPU): processing elements, storage elements, control unit and internal CPU interconnection system.
- Processing elements ALU (Arithmetic Logic Unit): Handle the input data and provide the result.
- Storage elements of the processor are known as Registers.
- Control unit generates the appropriate signals at the right moment.
- The connection between all the components is handled by an interconnection system inside the processor.

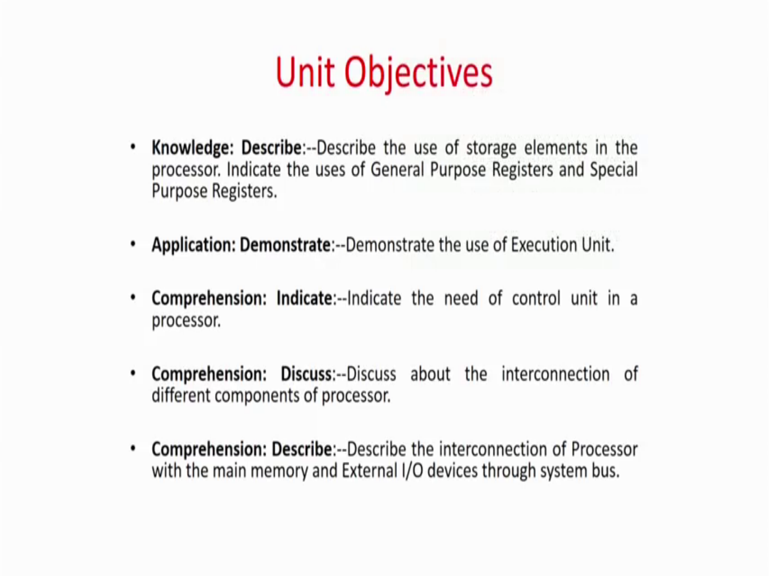
So, now we will go to the unit summary and then will go ahead. So, this unit will specifically focus on what are the units of a central processing unit, the storage elements that is memory there is control unit and interconnection system. So, processing unit will do all your computation, storage element is a memory because you are using a von Neumann architecture and how to control that is when data is 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 generate control signals at the right movement 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 will do it in a very broad manner because the idea of this module is to study how a 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)



Unit Objectives

- **Knowledge: Describe**:--Describe the use of storage elements in the processor. Indicate the uses of General Purpose Registers and Special Purpose Registers.
- **Application: Demonstrate**:--Demonstrate the use of Execution Unit.
- **Comprehension: Indicate**:--Indicate the need of control unit in a processor.
- **Comprehension: Discuss**:--Discuss about the interconnection of different components of processor.
- **Comprehension: Describe**:--Describe the interconnection of Processor with the main memory and External I/O devices through system bus.

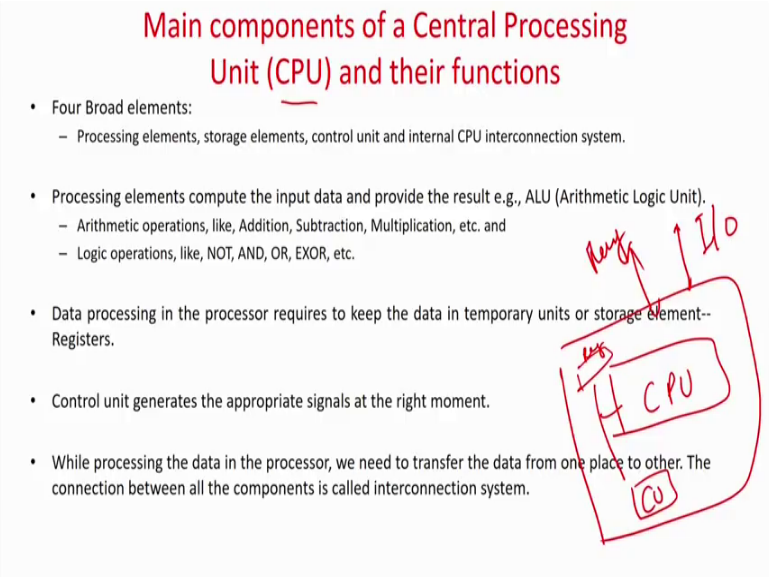
So, you will be so if you look at this Morley the, mostly the objectives are knowledge base, application base and comprehension base that is they will more or less on recall or knowledge kind of a analysis objective. In this case this is not more on this is not mainly focussed some kind of a design or implementation object it is moistly based on developing a knowledge and comprehension, for 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 register.

Why of this memory in basically the registers are require, 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 small loop how basically an execution unit basically looks at it. Then we will indicate you will able to ident indicate the need of a control unit why a control unit is needed.

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

So, as again repeating this unit basically is more of a recall type of an 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, let us start the unit.

(Refer Slide Time: 19:14)



Main components of a Central Processing Unit (CPU) and their functions

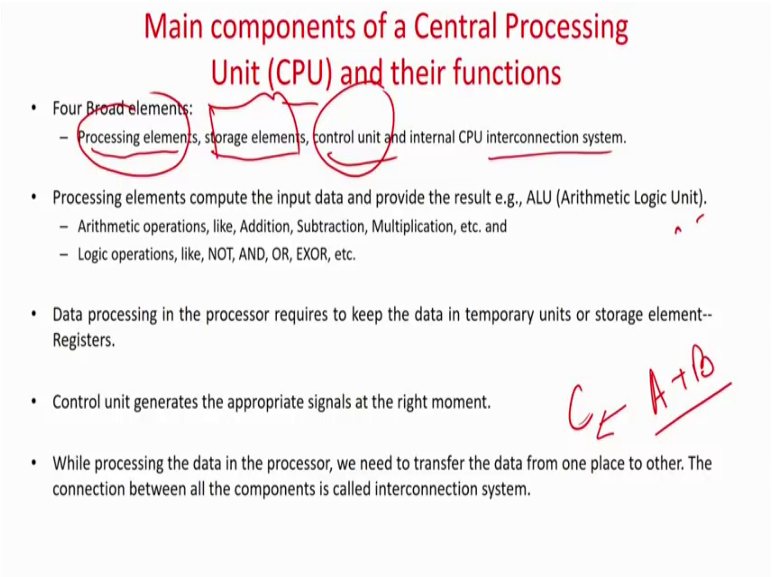
- Four Broad elements:
  - Processing elements, storage elements, control unit and internal CPU interconnection system.
- Processing elements compute the input data and provide the result e.g., ALU (Arithmetic Logic Unit).
  - Arithmetic operations, like, Addition, Subtraction, Multiplication, etc. and
  - Logic operations, like, NOT, AND, OR, EXOR, etc.
- Data processing in the processor requires to keep the data in temporary units or storage element-- Registers.
- Control unit generates the appropriate signals at the right moment.
- While processing the data in the processor, we need to transfer the data from one place to other. The connection between all the components is called interconnection system.

So, what are the main components of a central functioning unit or central processing unit we generally called 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 the this is central processing unit and generally will have a controller unit. I will take a more little figures and some memory, memory banks basically which I actually called registers, which are internal to central processing unit and the, but again all these things is 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 is an external is a ram or it is your we called the main memory, but when you have talking about the central fun 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 will whenever you need to make a differentiation will tell you either 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 processing element, one is a storage element, control unit and interconnection system, what is the processing unit?

(Refer Slide Time: 20:29)



Main components of a Central Processing Unit (CPU) and their functions

- Four Broad elements:
  - Processing elements, storage elements, control unit and internal CPU interconnection system.
- Processing elements compute the input data and provide the result e.g., ALU (Arithmetic Logic Unit).
  - Arithmetic operations, like, Addition, Subtraction, Multiplication, etc. and
  - Logic operations, like, NOT, AND, OR, EXOR, etc.
- Data processing in the processor requires to keep the data in temporary units or storage element-- Registers.
- Control unit generates the appropriate signals at the right moment.
- While processing the data in the processor, we need to transfer the data from one place to other. The connection between all the components is called interconnection system.

C ← A+B

See I take an example that, I have to add A plus 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 are, is 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 is 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 jam etcetera and then also actually the processing element are also responsible for storing the data, fetching the data like you have to bring the data A you have 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 will be more specific for a time being it is shift to assume that the central processing unit will operate only on 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 code and code assuming then the value of A and B has to first brought into the storage elements that is the register’s 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 of signal for those are generate done by a unit which is called the control unit and of course, you require an interconnection system because data has to be broad 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.

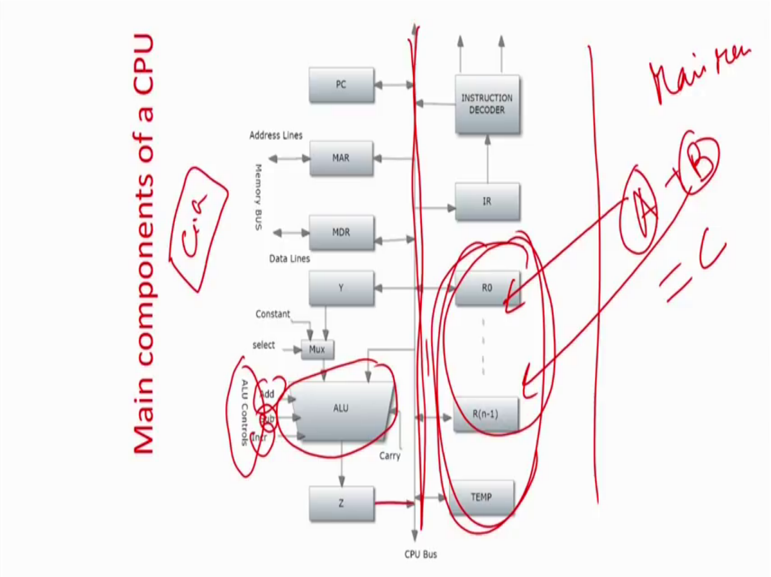
So, another set of interconnections and finally, the register will write the value of memory location C. So, another set of interconnections. So, that is why there is a big set of interconnection among the internals.

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 means registers and they generally tell that we are writing some temporary variables over here. So, why do 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 plus 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 valued when that data is return to the memory location C. So, therefore, and some computation like if you want to add A plus B plus C plus D.

So, generally you add A plus B stored in a register then you bring C, then you add the temporary value of A plus B with C, then you put A plus B plus C another register 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 generates 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.

(Refer Slide Time: 25:13)



Main components of a CPU

So, this is the basic set of elements which is in a CPU and their functions. So, till now I am telling you. So, much stories about different components of a CPU, register, ALU and so many things, but without looking at a picture it seem just like a story without a movie, just like a I am telling was story without showing the photographs. So, this is actually a picture of broad picture of a main components of a central processing unit. So, how it looks? So, as I told you there is something called the arithmetic and logic unit; so this part is responsible for performing all type of a arithmetic and logic operations, but as I as I told you lot of controls are require because we have an adder here, subtractor here, multiplier here, but not.

So, you see like there are some control signals add, subtract, implement. So, this control values will come from a controller that when I have to do what, then as I told you what there are many other components will come out slowly.

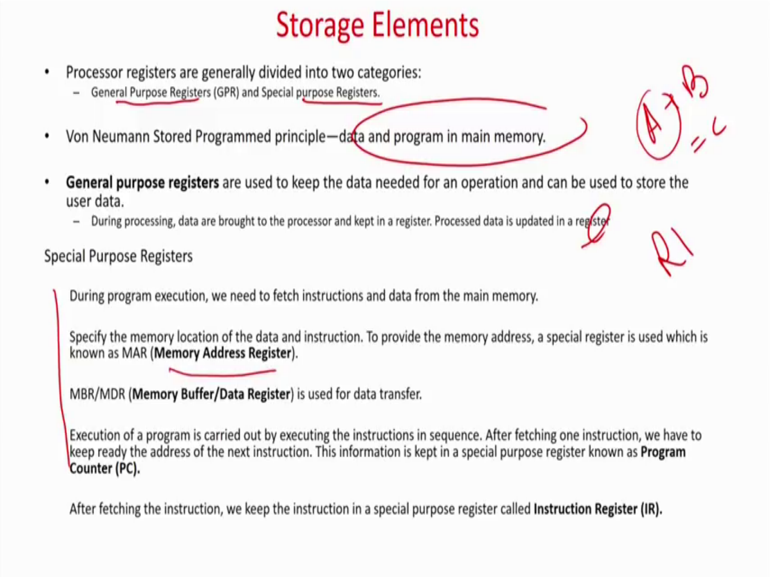
So, this is the register bank that is R 0 to R n 10th, be something we do not require right now in this discussion. So, the registers there are n registers over here that is whenever you want to take some as I told you A plus B equal to C. So, this A value will be is loaded to register R 0, B may be register this one they are main memory which is external to the CPU and the registers are nothing, but same memory location you can assume, but at a inside the processor. So, that will has to be loaded here because the processor generally the central processing unit will (Refer Time: 26:39) of the internal of it and directly not talk with the main memory.

So, this is the register bank this is the CPU controller is here I am not drawing it. So, this is the controller, it will send all the control signals when to do what and in fact you can see all the connections of the interconnecting buses. So, this is a main where it is connecting the ALU to all the registers to other registers as well, as was as I was seeing that 2 type of registers generalize general purpose register and specialize registers, R 0 to R n minus 1 mainly which I have talking about you load the data from the memory etcetera are general purpose registers.

Which is open to the programmers to use right and there are some specialised instruction some special register like I R that is a instruction register, program count will come to it memory address register, memory data register, they are specialize the registers you have to access, but you cannot directly use as your choice like the general purpose registers, they have a special meaning and they are use for a special purpose.

So, the registers you can see are general purpose, special purpose you have a ALU and you have controller to generate the signals. So, this is, this is the quit familiar diagram if you look at computer architecture any standard book or any standard lecture will tell you about the basic design. So, whenever somebody ask you to draw CPU basically the ALU, the buses, the general purpose registers the special registers are actually drawn, to show basically different components and basic what are the jobs of each of the block.

(Refer Slide Time: 28:02)



Storage Elements

- Processor registers are generally divided into two categories:
  - General Purpose Registers (GPR) and Special purpose Registers.
- Von Neumann Stored Programmed principle—data and program in main memory.
- **General purpose registers** are used to keep the data needed for an operation and can be used to store the user data.
  - During processing, data are brought to the processor and kept in a register. Processed data is updated in a register

A + B = C

R1

Special Purpose Registers

During program execution, we need to fetch instructions and data from the main memory.

Specify the memory location of the data and instruction. To provide the memory address, a special register is used which is known as MAR (**Memory Address Register**).

MBR/MDR (**Memory Buffer/Data Register**) is used for data transfer.

Execution of a program is carried out by executing the instructions in sequence. After fetching one instruction, we have to keep ready the address of the next instruction. This information is kept in a special purpose register known as **Program Counter (PC).**

After fetching the instruction, we keep the instruction in a special purpose register called **Instruction Register (IR).**

So, as I told you now will start looking at the storage elements, now will elaborate on each of the components 4 components we have seen. So, first general start with the ALU. So, as I told you ALU basically has lots of adders, multiplier, subtractors internally. So, mainly if you have the concept of digital design and which has also discuss in the basic module on which are the prerequisite and which are start from professor Deka that you have learn or you have revive basically some of the concept of digital design. So, ALU design mainly the logic and arithmetic units mainly consist of adder multiplier, subtractor which are from the digital design principles.

Now, let us look at the registers that is your memory or the storage elements in the CPU perspective. So, as I told you, you have seen some of the registers like R 0 to R 1 where you generally keep your daily to daily variables like A, B, C, D. So, they are general purpose register, allow with that you observe and remember the names say something called I R that is the instruction register, say something called the program counter say something called the MAR memory address register and MDR memory data register. So, they are use from some specialized functions, you cannot directly access them they are use some specific job they have to be access like indirect manner and in fact you never require to use them while executing any code.

So, let us now go about it. So, that 2 regi, 2 type of storage elements general purpose and special purpose. So, as I told you we are using Von Neumann architecture. So, both data and program are in the memory. So, memory concept as to be very important by a understanding a CPU operation and our case as you have dealing with the central processing unit, our data or our memory or our program everything is stored in a memory and internal to that one is a register.

So, as I told you general purpose registers are used to keep data needed for operation and our user data basically. So, generally data are brought from the memory to the processor keep your register they are accessed abided by the CPU or by the arithmetic logic unit and then again they are written back to the memory which is the main memory we are talking about.

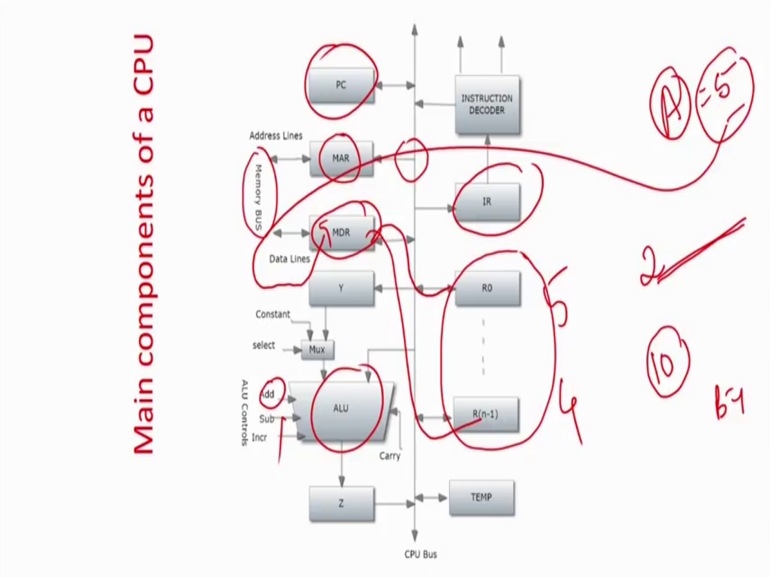
But that is simple, but generally there is something called as I told you lots of special purpose register. So, will see one by one, there is something called main memory address register. So, there is a memory, as I told you if you are doing A plus B equal to C. So, A is nothing, but a memory location. So, if you have done assembly level programming. So, basically we light load say R on1e register 1 from memory location A. So, you have to fetch the value of A to register R 1, then you need to know what is the value of memory location A.

So, now I have to load the value of memory location A to the memory then the memory will fetch the data from memory location A and dumb it in R 1. So, some somewhere I should specify where is the address. So, that is the special purpose address register which is called the memory address register that is memory address register. Though your arithmetic and logic unit will generate or sometime the instruction decoder will generate the address from where the value of the memory has to be fetched. So, this actually is the memory address register.

So, whenever you are saying A equal B plus C or whatever the memory location address is generated in the or kept in the memory address register who is actually goes to the memory busm will come what is the memory bus basically the address is stored for the memory address register this one will talk to the main memory. Bring the data from that address to memory buffer register and then write to the temporary register. So, memory address register you specify where is the address of the data that has to be fetched.

Next is memory buffer or memory data register, now we have given the address the memory has read the value from the address and you have to dumped the data, then where it is dumped? Basically regular special purpose register called memory data register. So, say memory location A has been read A was having the value of 5.

(Refer Slide Time: 31:53)



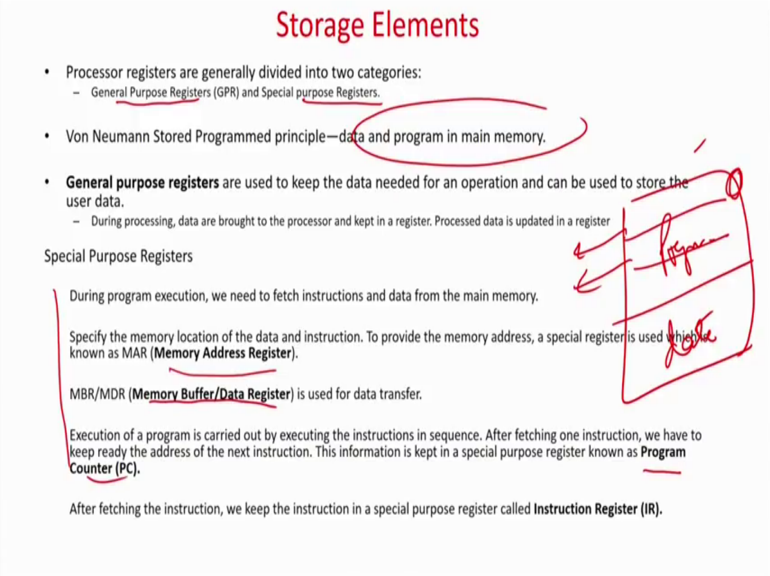
Main components of a CPU

Let us I show you. So, now, the memory will get the value of a memory location A, let us be 5 and it will where it will give it to the processor, how it will give it to the CPU. So, it will actually dumped the value of a is something called a memory data register, memory data register will temporarily hold the value of 5 then bia the system bus it will be give you to R 1 or any general purpose register. So, this MDR memory data register which is special purpose register to store the value of the temporarily the memory location that has been read.

Another important general purpose register is the program counter that was you have written in this point a program basically execute in steps. For memory location all the instructions are stored in the memory as well as data is also stored in the memory because is Von Neumann architecture. So, you first take the value of memory location say 1, execute it memory location 2, instruction 3, 4.

So, whatever are the instruction in memory location 1, then 2, then 3 you keep on sequentially executing it and how do you know what is the next instruction to be executed the value or apart basically if you look at it a part of the memory is your draw it like this, a part of the memory is data and a part of the memory is program.

(Refer Slide Time: 33:02)



Storage Elements

- Processor registers are generally divided into two categories:
  - General Purpose Registers (GPR) and Special purpose Registers.
- Von Neumann Stored Programmed principle—data and program in main memory.
- **General purpose registers** are used to keep the data needed for an operation and can be used to store the user data.
  - During processing, data are brought to the processor and kept in a register. Processed data is updated in a register

Special Purpose Registers

During program execution, we need to fetch instructions and data from the main memory.

Specify the memory location of the data and instruction. To provide the memory address, a special register is used which is known as MAR (**Memory Address Register**).

MBR/MDR (**Memory Buffer/Data Register**) is used for data transfer.

Execution of a program is carried out by executing the instructions in sequence. After fetching one instruction, we have to keep ready the address of the next instruction. This information is kept in a special purpose register known as **Program Counter (PC).**

After fetching the instruction, we keep the instruction in a special purpose register called **Instruction Register (IR).**

So, first instruction, the first location it is executed second instruction is executed and so forth.

But then who knows that what is the address location of this instruction, actually that is a special memory special register which is called the program counter which is PC, it will remember. So, it will first start with the memory location one, then that instruction executed PC is automatically incremented.

Now, it is 2, the second memory location instruction is brought and execute it and where it is brought it is brought to a special instruction called special register called instruction register, I R is an instruction register. So, whenever you execute an instruction the value of PC is the address of the instruction that has to be executed. So, PC will give the value to memory address register, memory address register will ask the memory to give the data it will give the data in the memory data register.

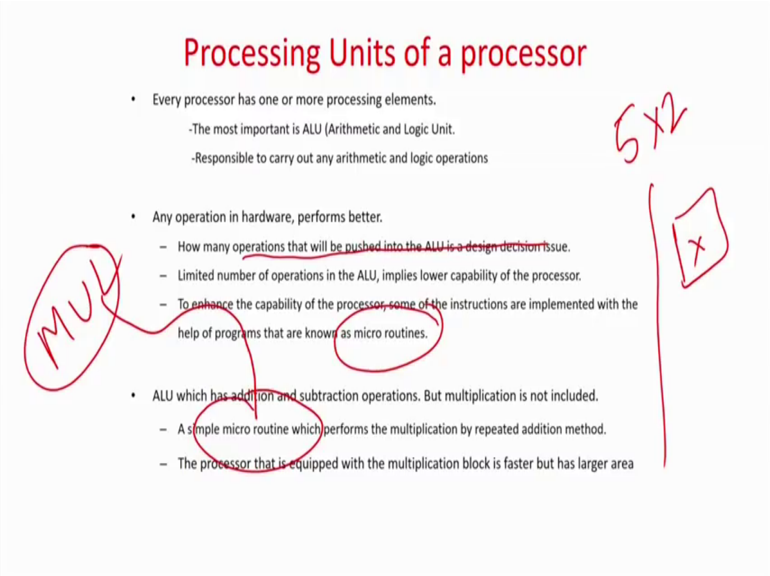
If is an instruction the instruction register will take the instruction decode it and execute it, but the PC is always auto implemented, first instruction second instruction third instruction. So, it keep remembering and when a jump instruction comes then the PC will not incremented by 1, it will low point to the loop instruction location which is to be jump see for example, these after second memory location instruction, you have to execute the instruction which is the memory location number 10 then after executing the second instruction PC will not be 3 is a jump. So, the PC will have the value of now 10.

So, basically program counter is also specialised instru specialised registers which hold the value of the next instruction to be executed, as well as instruction register is another specialised register which is actually holds the instruction and for the decoder to decode what the instruction has to do.

So, as I told you apart from general purpose register, there are lot of specialise the registers which have some special job to do and which is not accessible to the user for the programming and many others are there. We are just giving an broad overview that what are the storage elements registers 2 types general purpose and specialised, let us go ahead.

Then as I tolding, telling you that next very important of the or you can should not say which is more important to the other we are talking about the ALU.

(Refer Slide Time: 35:17)



Processing Units of a processor

- Every processor has one or more processing elements.
  - -The most important is ALU (Arithmetic and Logic Unit.
  - -Responsible to carry out any arithmetic and logic operations
- Any operation in hardware, performs better.
  - How many operations that will be pushed into the ALU is a design decision issue.
  - Limited number of operations in the ALU, implies lower capability of the processor.
  - To enhance the capability of the processor, some of the instructions are implemented with the help of programs that are known as micro routines.
- ALU which has addition and subtraction operations. But multiplication is not included.
  - A simple micro routine which performs the multiplication by repeated addition method.
  - The processor that is equipped with the multiplication block is faster but has larger area

So, ALU as I was say is generally responsible for computation. So, there is one very important thing, as I told you ALU should have a arithmetic and logic unit that is mandatory which is carrying out the logical operations as well as arithmetic operations. But there is very one important thing we have to understand there is a trade off, I would like my ALU to be extremely super fast I would like to have adder, multiplier, subtractor, graphic processing as well as may be what not I want a oracle ALU you give an instruction immediately it is executed you can do that, but at will actually make the size of the ALU extremely large and very expensive for the cost of the processor.

So, here we have to make a decision that how many operations will be pushed in the ALU is a decision issue that whether you will your ALU will do only add and subtract or it can also do add, subtract, multiply and divide because you know at the multiplication can be implemented by multiple number of additions. Similarly division can also be implemented by combination of addition and subtraction and even subtraction can be implemented in a twos compliment arithmetic by only addition.

So, now it depends on how much hardware you give it to the processor. So, if your processor has a multiplier, hardware multiplier then you give 2 numbers immediately it will be multiplied and you are going to get the answer. But if I say that I do not have a multiplier unit, I only have a adder and subtractor then easier processing unit not able to multiply no, processing unit can do anything that is possible by algorithm.

So, that is again a theoretical prospective I do not want to go, but if your processor has an adder, subtractor and some registers then it can be shown that any algorithm or any problem that can be solved by an algorithm can be executed by the simple processor, but anyway that is two of the theoretical angle.

So, I do not have a multiplier, but still my processing unit can do a multiplication that is you have to write a small routine which is actually called micro routine, a small routine will be there for the time being you can understand the small program which will say it is 2 into 5. So, what it will do? It will add 2 5 times so, but now your processor or your arithmetic logic unit is able to do 5 into 2 but it will take more time because you are going to add 5 2 times or 2 5 times.

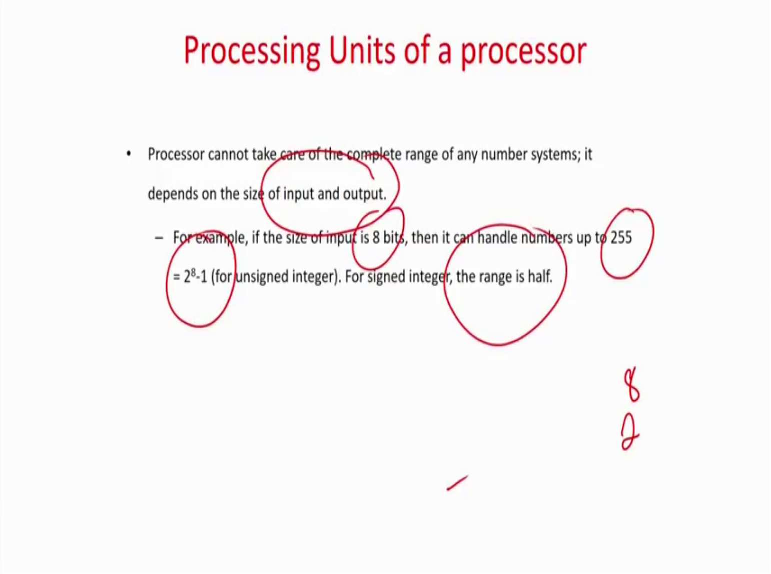
So, is a code will execute to implement that, but if I have a dedicated multiplier unit any type of multiplier you can think about add a multiplier or whatever nature, then you did not execute any microcode you keep 5 and 2 immediately it will multiplied and it will give the answer, but in that case the area of the microprocessor of sorry, area of the processing unit or arithmetic and logic unit will be larger. So, there is always a trade off that what to put in the software and what to put in the hardware.

Software means you have to write micro routines like as I told you multiplication can be done by addition or you want to put a dedicated hardware to execute it in that case ALU will be more complex it will faster, but it will be more expensive too. So, therefore, there is lot of trade off in between this. So, there is actually sometimes called micro routine that you will have the instruction, multiply A cross B, but actually multiply will be internally represented by a micro routine which will break it up into smaller instructions and actually it is implementing multiplication by multiple additions.

So, brought idea of ALU is very simple, you design it for addition multiplication subtraction using digital design fundamentals, but how much you put and how much you put in a software like a micro routine depends on the decision of design. So, in the next you modules and lectures will see such type of designs, in fact the next module will handle those things that if you have to design a processor and you do not have a multiply unit then how do write a micro routine which will execute it.

So, as again very important thing, what is the number size you can represent in processor of course, is not infinite we always do the something called 32 bit machine, 64 bit machine, you heard the term.

(Refer Slide Time: 39:01)



Processing Units of a processor

- Processor cannot take care of the complete range of any number systems; it depends on the size of input and output.
  - For example, if the size of input is 8 bits, then it can handle numbers up to 255 = 2^8-1 (for unsigned integer). For signed integer, the range is half.

8
2

Exactly it depends on the range of numbers it actually depends on the input and output size of the processor, arithmetic logic unit say for example, if the input size is 8 bits then the number it can handle is 2, 255 that is 2 to the power 8 for unsign number, if you are using a sign number then one bit will be deserved for sign or other way. So, the range is high. So, in other words a processor is not is pos, but again you can use that is I told you that you can use multiplication, in multiple type of additions you do it similarly you can represent the very large number also, but not directly.

If your, if you have a input of 8 bits you have to truncate it, you have to go for the first 8 bit LSB, then next LS set of 8 bit LSB and so on. But there is it will going on a iterates manner, initially you cannot have everything and directly go 8 bit input means 2 to the power 255 un sign and half of that in sign can be directly used for computation, larger than that you have go initiation first LSB 8 then next then next and so forth.

So, in fact that what is I am going to say is that processor can do a lot of things, in fact everything which is possible by a algorithm, but what is directly possibly what is the hardware implementing it, like 8 bit processor like directly to 8 bit numbers can be multiplied or whatever, but if it the 16, 16 bit numbers then you have to go it in phase first 8 LSB and next and so forth.

So, either you go in a iteration and take a more longer time or you have a more expensive ALU, with a more hardwaring is a design choice. Then as I told you we are coming to the third component which is called the interconnections and controller design is more complex, I will just give you the idea, but not going to elaboration because controller signals are assume that it will come. So, in fact let us not go interconnection directly.

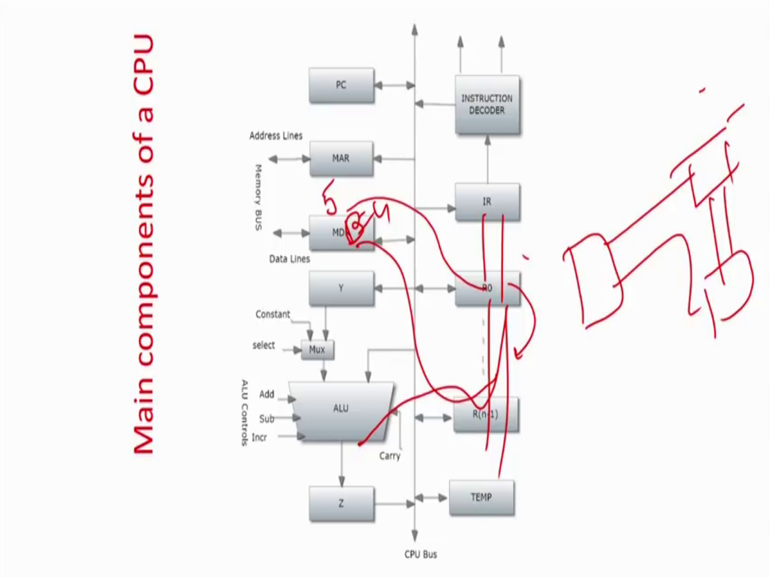
So, we can go for first and very basic idea of what is basically the control signals, say for example, and why do not require elaborate at this position is something like that. Say I want to add A plus B, the value of A has to be first j first the program counter PC has to tell that instruction has to be executed, then the value of a has the A has to be put to the memory address register the address a memory address register will get the value to the address memory, memory will get the value of variable A it will dump into the memory data register say and it will go to a general purpose register from R 0 to R 1 similarly you have to repeat treat for b. So, 1 one register will have the value of 5 that is A and may be you can assume that B is in the value of 4. So, another register will have it.

Now, you have to add it. So, a plus 5 plus 4 will happen and this it will be stored in another temporary register may be R 3 and again the value of R 3 will be return to the memory data register and now the memory address register will have the value of C and the from the data. Now, from the memory data register which is now 9 will be return back to the value of C, in fact memory data register both (Refer Time: 41:59) value when it coming from the memory as well as when it is going from the central processing unit to the memory it is a buffer.

So, lot of control signals are required over here like for example, when the memory data register is writing the value of A to memory register 0, a connection has to made like this. When the value of B is being read from the memory data register are connection has made like this. So, lot of multiplexing signals has to be there, when you have add it you have to made this line 1 and all others are 0 because the ALU responsible for addition subtraction multiplication everything. So, I assume that there is a block which is a controller, which will generates signals on appropriate time, how to design the controller is a big job that we are going to see later.

So, therefore, you assume that there is a oracle sitting which will generate all the signals has and when required. So, that is the basically the idea of a controller. So, I at this position point is not wise to go more depth on the design of controller because it will involve 4 to 5 units totally dedicated on that. Now, basically you are going to go for the interconnections as I told you I cannot put one very simple way you can tell that, I can have a matrix side of an arrangement which is the very best like this one I can because you see as I told you. So, there is a single memory data register.

(Refer Slide Time: 43:15)


Main components of a CPU

It will get the value of a first then will get the value of B that is equal to 4, then the value of B will be fed to R 0, the value of B will be fed to R some value. So, what I can do? I can say that this is memory data register and these are all the registers.

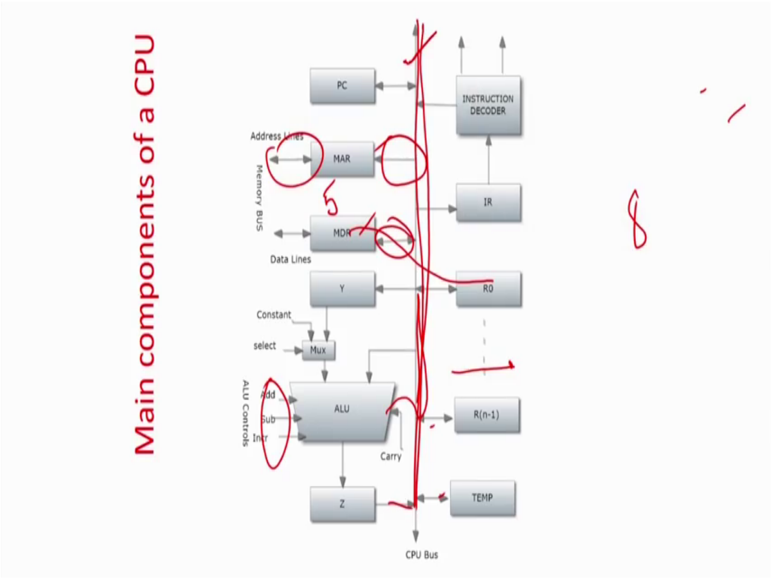
So, I can make a connection like this, every point we have 3 point because value of R 0 may also be loaded to R 3. So, in between these also lot of interconnections. So, everything may be interconnected, ALU may be able one to write it to directly to.

So, I can make a pure matrix connection, that any point of time any 2 guys can be connected R one can be connected to R 5, R 1 can be connected to ALU, R 1 can be connected to memory data register. In fact there it will have a very nice arrangement because everybody can talk to everybody in at any point of time, but you have to understand that it will make a very cube curve type or a very cerotic type of type of a interconnection network.

In fact then you random in to a situation that nothing can be debug or nothing can be gained out of it because your interconnecting network or it will matrix interconnecting matrix will more expensive to implement then the whole compro whole CPU itself. So, that is not going to wide decision.

So, what build to do, people implement a bus structure that is a single wire is there or single set of a wires are there for different purpose and whenever somebody has to use as wire they have to use it on a mutual exclusive basis that is; obviously, simultaneous access cannot be there. But the cost is very reasonable because say whenever say for example, the m d R is supplying data to R 0 at the same time ALU may not be able to use.

(Refer Slide Time: 44:43)

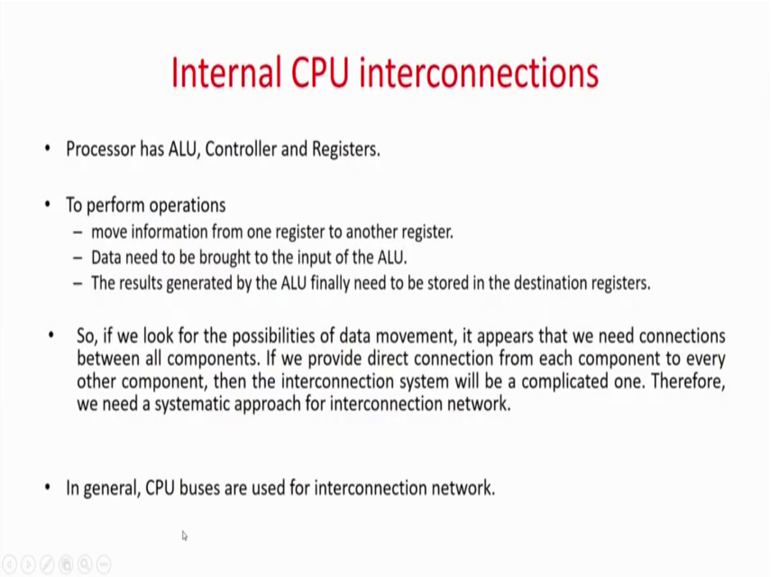

Main components of a CPU

This bus to write to another register it has to wait till the memory data register has transfer the value R 0 then only ALU can do it. So, it is a, the whole organization is such a fashion that they are all a on a mutual exclusive basis, this where is centralized. So, either any one of them can use it at 1 time. So, other guys has to wait for it. So, it is a time taken is slightly higher, but it is a more feasible implementation and more cheaper implementation and as you have talking about not a very advance processor. So, more advanced processor has replicate the number of buses, there is 2 bus organization 3 bus organization and so forth.

So, generally we do not call it where, when will be talking in terms of computer architecture will call it bus because we know that when a talking about 8 bit input. So, 8 lines are coming out. So, this is not a single line actually it is a 8 line, 8 bit bus. So, you make a sign like this and it implies that it has a 8 bit bus, 8 internal lines are going on it.

So, you should use the term bus and not to wire, but again the most important concept here is that they are all in sheared basis. So, when this if I call it has a data bus. So, only appear of registers will transfer the data through a ALU then it will be first transferred from the register to the may be ALU from ALU to register then when one transfer is going on the other all guys has to wait.

(Refer Slide Time: 46:07)



Internal CPU interconnections

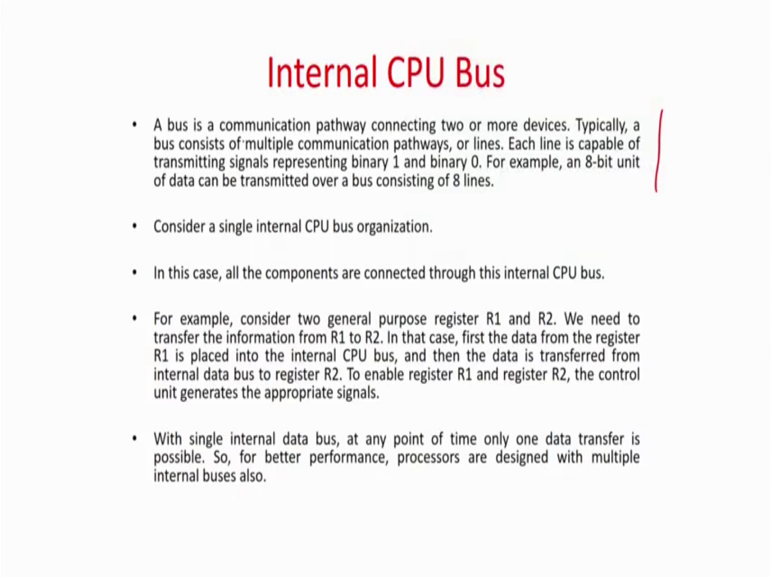
- Processor has ALU, Controller and Registers.
- To perform operations
  - move information from one register to another register.
  - Data need to be brought to the input of the ALU.
  - The results generated by the ALU finally need to be stored in the destination registers.
- So, if we look for the possibilities of data movement, it appears that we need connections between all components. If we provide direct connection from each component to every other component, then the interconnection system will be a complicated one. Therefore, we need a systematic approach for interconnection network.
- In general, CPU buses are used for interconnection network.

So, that is the basic concept of a single bus processor that is we generally call it CPU interconnection. So, a processor is ALU controller and registers and they have to interconnect among them. So, what are the interconnection operations what it requires data need to be brought from the input to the ALU, very important that is you have to get the data from the memory to the register and then to the ALU then 1 register may have to send data to the other. Result generated by the ALU permanently has to be stored to the register and then has to be return to the memory so lot of there are transfer will happen.

So, if you have components as I told if you have interconnection with all components very nice, but in fact large number of number some arrangement and more cost. So, the interconnection cost will be more higher than the cost of the components itself.

So, therefore, we use a bus architecture, bus means number of lines will be more than 1, it is a 8 bit architect bus then will be 8 bit data bus. So, you call the use bus so in fact addition. So, you can see bus, this is typical definition of a bus, bus is a communication pathway between 2 or more devices.

(Refer Slide Time: 47:05)



Internal CPU Bus

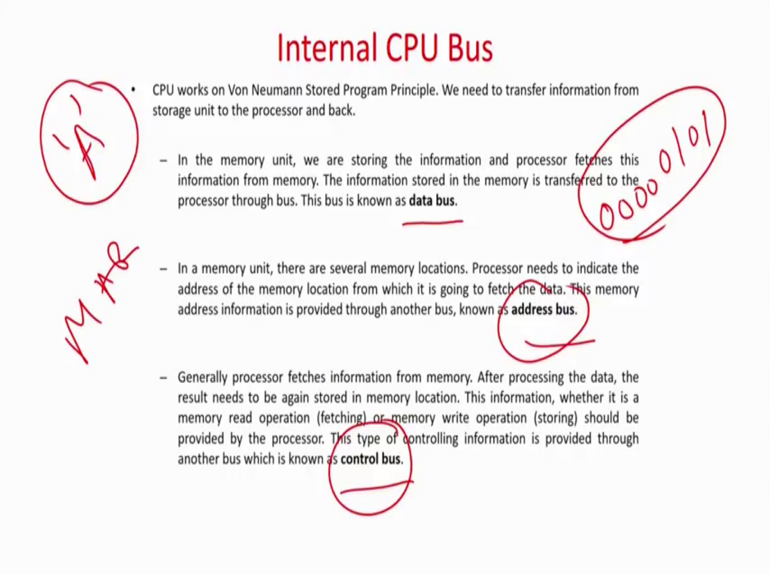
- A bus is a communication pathway connecting two or more devices. Typically, a bus consists of multiple communication pathways, or lines. Each line is capable of transmitting signals representing binary 1 and binary 0. For example, an 8-bit unit of data can be transmitted over a bus consisting of 8 lines.
- Consider a single internal CPU bus organization.
- In this case, all the components are connected through this internal CPU bus.
- For example, consider two general purpose register R1 and R2. We need to transfer the information from R1 to R2. In that case, first the data from the register R1 is placed into the internal CPU bus, and then the data is transferred from internal data bus to register R2. To enable register R1 and register R2, the control unit generates the appropriate signals.
- With single internal data bus, at any point of time only one data transfer is possible. So, for better performance, processors are designed with multiple internal buses also.

Typically bus consist of multiple communication path way or lines for example, is a 8 bit bus means it can directly transmit 8 bit data, generally we are talking about or throughout the 2 modules will talking of a single bus architecture because that was a long time for long large number of a design of processing elements or processors.

Single bus was standard and everything has to be connected to the single bus architecture, then I told you it is a multiplexing arrangement because you have to wait when the other 2 guys are transferring for example, if 2 general purpose register R 1 and R 2 need to transform data to from R 1 to R 2. In this case first the data from register R 1 is placed to the internal CPU bus and then the data is transferred from the internal CPU bus to the register 2, then you have some control signals to do that, but when the transfer is being going on other guys cannot do anything they have to wait. So, R 1 will give the data to a some register, the control signal will be appropriate reset and then through the data bus it will go to R 2 other guys are just sitting ideal they cannot do any operation with the bus.

So, with the single bus at any point of time only 1 data transfer is possible. So, we better performance you can increase more number of bus, the best of already told you fully matrix operations. So, more have been any type of hardware more you put on hardware more benefit you are going to get.

(Refer Slide Time: 48:21)



Internal CPU Bus

- CPU works on Von Neumann Stored Program Principle. We need to transfer information from storage unit to the processor and back.
  - In the memory unit, we are storing the information and processor fetches this information from memory. The information stored in the memory is transferred to the processor through bus. This bus is known as **data bus.**
  - In a memory unit, there are several memory locations. Processor needs to indicate the address of the memory location from which it is going to fetch the data. This memory address information is provided through another bus, known as **address bus.**
  - Generally processor fetches information from memory. After processing the data, the result needs to be again stored in memory location. This information, whether it is a memory read operation (fetching) or memory write operation (storing) should be provided by the processor. This type of controlling information is provided through another bus which is known as **control bus.**

Then 3 very important things we have to tell you, there is something in a bus when you talk about single bus or multiple bus, a bus has basically 3 sub parts data bus, address bus and control bus. So, what is a data bus? As I told you, you want to transfer some very varying of a variable from register R 1 to R 2. So, a bus which will do this operation is called the data bus it will take the numbers 5 that is may be 1 0 sorry, 0 1 0 1 it will take it from register 1 and it will transmit the data to register B, then a bus which will doing it in fact is a 8 bit bus. So, 4 LSB is will be 0 so the bus which will doing is actually called a data bus. Now, as I told you sometimes you have to get the value of some variable from the memory, like add A 1, A 2 or A comma B. So, the value of a that is the memory location has to be sent through the register sorry memory by the memory data memory address register MAR.

So, the value of A that is the sorry the value of a will be fetched, but what is the memory location, the memory location of A that is the address of the A has to be send to memory address register now which bus will send it that is actually called the address bus. So, address bus will always consists of the address of the register. So, address of the memory locations, In fact data bus and address bus are 2 separate buses.

So, data bus and address bus are not multiplex, some if some data transfer is going in the data bus the address bus can freely send some new data which has to be fetched from the main memory the address can be send by that and. Secondly, as I told you the third type of unit called the control bus.

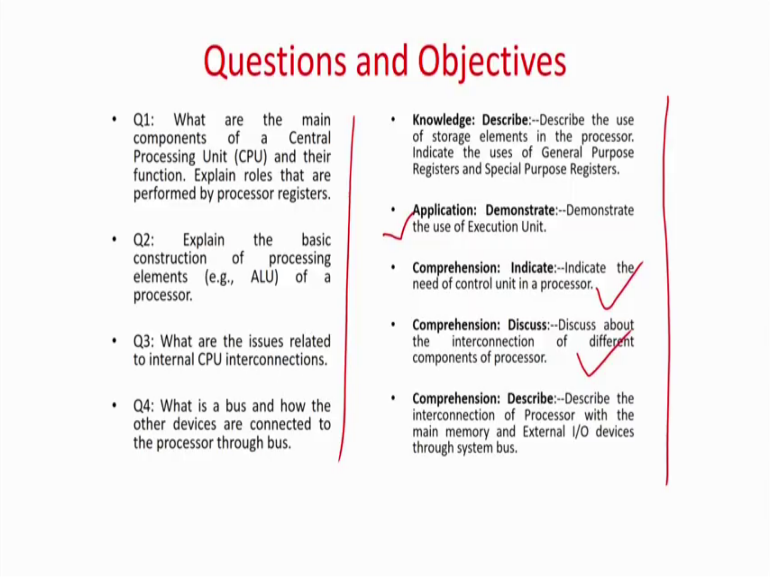
So, which is generating the signal as I told you when to add sub signals has to be change to the ALU, A A R 1 has to be connected to R 2 simultaneously all other guys cannot use the address bus address sorry data bus. So, some internal signalling has to be done. So, a separate set of wires actually do it for you that is the control bus. So, this 3 buses will totally make a system bus of the internal CPU bus. So, there is only one internal CPU bus in single bus organisation, but the address bus data bus and control bus are segregated and they can be use simultaneously there is no multiplexor, multiplexing in between them right.

If I show you this for the last time so I can tell you that this is the data bus, all this connections are data buses because they are transferring the data these are the control bus, add subtract registers are control signals which is generated like and this bus memory sorry memory address register. So, this will be the value which will be generated which is the address bus. So, it will send the value of the memory location which has to be fetched.

So, overall we are at this first lecture on this unit we are telling you the basic definition what is the basic internal meanings and we are keeping many things is very abstract manner. So, basically we are giving when idea of what is a data bus, what is a address bus, what is a control bus, but lot of intricacies are there, like in fact how the address bus can be multiplex and when the some address bus is bring the work, how we can utilise the data bus through other operation. So, we are coming the few future units.

As I already told you 3 buses and again the multiple buses can be used to enhance the performance of the procedure un performance of the program. So, with this we come to the end of this unit. So, this unit basically talked about the different units of the different parts of a central processing unit, the interconnection the registers the arithmetic logic unit and the control, in a very broad manner which you can give an idea what is the internally looks like it.

(Refer Slide Time: 51:57)



Questions and Objectives

- Q1: What are the main components of a Central Processing Unit (CPU) and their function. Explain roles that are performed by processor registers.
- Q2: Explain the basic construction of processing elements (e.g., ALU) of a processor.
- Q3: What are the issues related to internal CPU interconnections.
- Q4: What is a bus and how the other devices are connected to the processor through bus.

- **Knowledge: Describe**:--Describe the use of storage elements in the processor. Indicate the uses of General Purpose Registers and Special Purpose Registers.
- **Application: Demonstrate**:--Demonstrate the use of Execution Unit.
- **Comprehension: Indicate**:--Indicate the need of control unit in a processor.
- **Comprehension: Discuss**:--Discuss about the interconnection of different components of processor.
- **Comprehension: Describe**:--Describe the interconnection of Processor with the main memory and External I/O devices through system bus.

Now, as again as you have telling about pedagogical aspect. So, we have told that there are certain objectives like describe the user processor, demonstrate the use of execution unit and so forth. So, these are the objective you have claimed that a person should be able to do when he has completed this unit well.

So, again we put some 4 questions over here, like what are the main components of a central processing unit and their function, explain the role that are performed by the processor registers. So, once you have done the lecture I think in the very few first few after first few slides I have told you about the answers of that, again the second question explain the basic construction of a processing element ALU of a processor. As I told you it consists of digital design elements like adder, multiplier, subtractor then you have to decide whether you want to do addition, sorry multiplication by addition or you want have a hardware unit. So, you have broadly discuss about what is the basic construction of a ALU not the details, but the basic how it what, what are the issues and how it think about the design based on trade of between accur trade of between what power it has and what is the area and what is the cost. Similarly what are the issues related to CPU interconnections, like as I told you single bus. So, you have to wait if you are using a matrix kind of a connection it can be again may recovery high and arrangement, but what are the issues means how it is interconnected because there is a multiplex arrangement.

So, simultaneously their transfer cannot be done so that already has been discussed. What is a bus and how other design are connected to the bus that is also very simple, you have done an illustrate in a pictorial fashion of how the ALU the IR the general purpose registers are connected. So, these are the 4 questions which you are expected to answer after going through the module and you have already discussed it, but then the you are also you appreciate that if you are say for example, if you are able to, if you are able to des explain the basic construction of a arithmetic and logic unit then you are able to demonstrate of satisfies this objective seeing that demonstrate the usage of an execution unit. So, if you are able to explain the basic construction of an ALU you are actually successfully bidding of this objective.

Similarly, if you are say that what are the main components of a central function unit and what are the roles. So, if you are able to tell what are the main functions of the units you will be able to indicate the need of a control unit, similarly you will able to discuss about the interconnections that is if you can tell what are the main components then and what are the functionality then you can easily say that why are controller is needed and how a interconnections are there.

So, in fact if you look at the questions and if you look at the objectives answering all the questions will actually need all the objectives and again if you can appreciate whether knowledge and mainly comprehension base objectives. So, the questions are also of that nature, that you have describe you have discuss there is no more no design in this and in fact this module is expected to give a very brief an abstract overview of a central processing unit. So, you mostly, now in the unit we will now start dealing with how a memory external memory is organized. So, in that case will go slightly in depth of that aspect then will try to go how a program is executed.

So, in that case the objectives start becoming more of a design and more of analysis and application based. So, the now it is knowledge based. So, it will be more on simulation based objectives most on design base. So, that once you do all those units you will able to designs of that. So, we that we come to the end of the first unit of this module and next will start looking at the memory units of in a overview label and then you slowly go to the in depth of how the instructions are executed in a processor, which is a main role of this module.

Thank you.