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

Control Unit
Lecture – 16
Instruction Cycle and Micro-operations

Welcome to the next module on control unit. So, in the course in the last module we have basically seen that what is the basic architecture of a computing system? Like we have a control unit, then on the other side we have the memory, we may have some peripherals and then on a very theoretical level we have seen what are the component inside like a memory block, then we have also seen what are the registers in the control unit, then we have seen an ALU, how they are connected and also we have seen in a very theoretical notion and how a code get executed.

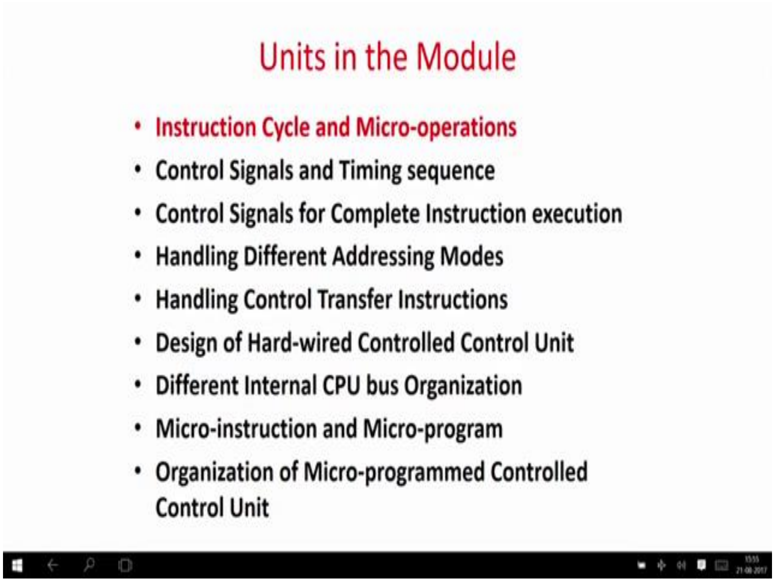
But you might have thought that such a complicated situation that you instruction is taken, then just the next instruction is taken after that it is decoded, fetched and so many intermediate steps happen.

And they also happen in the hardware where everything is in terms of bits. So, then means; that means, there should be lot of control signals or there should be lot of control intricacies which happens, because which helps for a code to execute in such an integrated environment that is on a basic computing system.

So, the last module was on the basic architecture of that system. In the next module that is in the control unit what we are going to see we are going to look at basically how those, how those units.

Like for example, your registers, your memory, your CPU, your busses, how they are basically interconnected and what are the signals and what are the exact signal flows that is required and how the signals are generated. So that a code executes in a current fashion. So, in this module we will be mainly concentrating on how a control unit is generated? What are the required signals? How the signals are generated? And how the sequence of signals actually maintains a proper flow of the code execution? So, that is the basic idea of this module.

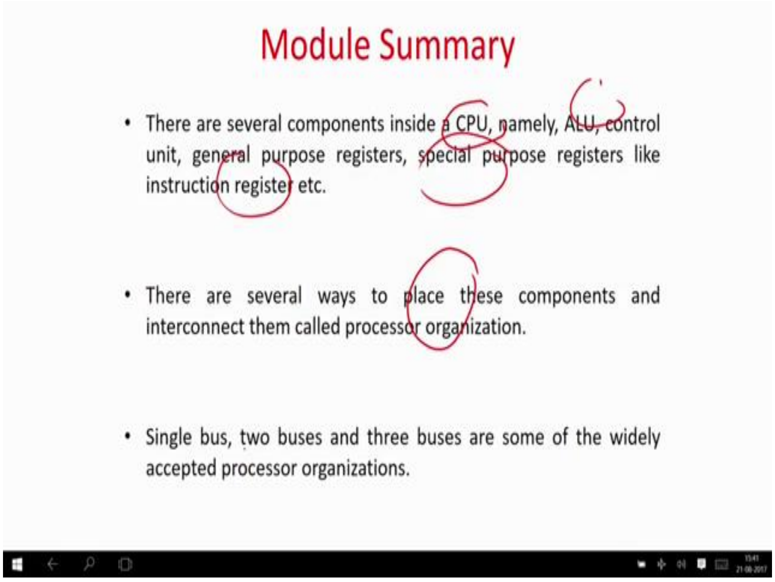
(Refer Slide Time: 02:08)



So, in this module basically we will be making mainly looking at the instruction cycle and the micro operations inside that, then we will be making mainly looking at control signals and timing sequence and so forth. You can go through all the different units in this module. So, this will give you a feel that basically this module focuses on how what are the instructions? How they are divided to micro instructions? And how automatically some control signals are generated, which actually are required for the control flow of the data in the registers or CPU busses etcetera and how basically what are the different ways are actually generating the signals in the control unit.

So, a code executes in a code and the interrupts and if you have for example: jump instruction. So, how basically a code executes and what are the required control signals for this generation and how actually timing is Organizer and what is the timing issues etcetera, which we will be learning in different units of the module. So, the details of the modules are available in this slide.

(Refer Slide Time: 03:08)



So, as we are going by a pedagogical aspect. So, let us discuss on the module summary. That what is going to be covered in this module so in fact, as we already discussed in the last module that a basically a computing element has CPU, ALU, special purpose registers, instruction registers etcetera.

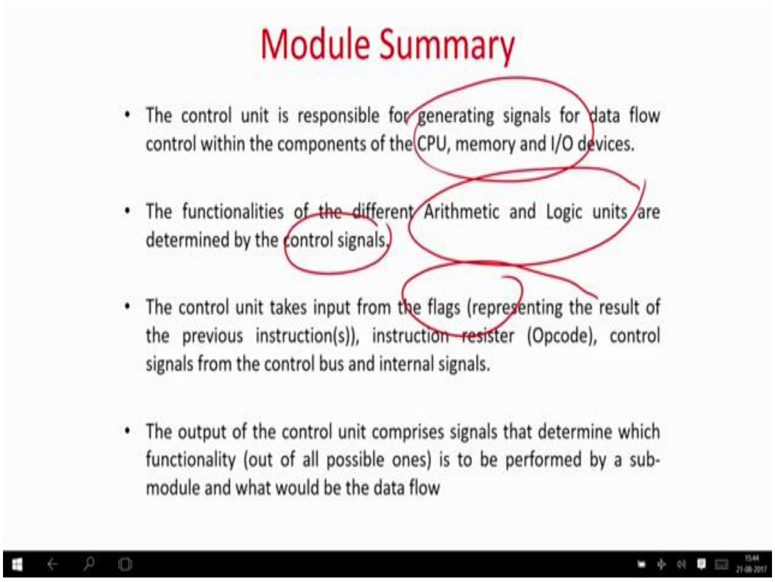
And basically there are several ways to interconnect it that, whether you have a single bus architecture, whether you have double bus architecture, whether you have three bus architecture, whether you have some local memory for a local CPU etcetera.

So, we will study in this module how basically different ways of connecting the components like as I told you one bus, two bus and three bus organization. So, we will have a look at this look at the idea of such interconnects of such units like register s, CPU etcetera and then we will see for different such configurations how are signal generated? And how actually the flow of code moves?

Because it is very obvious that if you have three bus system the execution of code will be faster compare to a single bus system, because in if your single bus system is there then there has to be lot of multiplexing because there is a one resource which is a bus and it is only dependent on for all data transfers.

But for example, if you have a three bus system then actually some bus can be dedicated for register to register flow, some bus can be used for memory to register flow and so forth. So, therefore, the control signals will also be different in the latter case compare to the formal case if there is a three bus system the code execution will be faster because we have now three busses to solve the problem and so forth. So, we will be looking at how control signals are generated and in different processor organizing; organization configurations.

(Refer Slide Time: 04:40)



Then basically as I told you whatever your code moves and how a code sequence a code is sequenced or if I say that you fetch a code or fetch a instruction or you fetch a data that mean some signals has to be generated. Like we have already seen that if you are looking for a memory, you should generate whether it's a read write signal, we have to generate some values for the memory buffer register and then the memory buffer register has already all the all the data is already given to the memory buffer register from the memory, you should generate a signal saying that the data is now ready and it has been sent to the buffer you can read it.

For example you want to write something to the memory, then you write it into the memory buffer register and then you have to wait for some time all this handshaking signals. For example, if there is a 0 flag. So, the 0 flag is set; that means, the control signal will be generated. So, all such different signals how they are generated and how they actually involve in the flow of code or flow of code execution in the control unit will be studied.

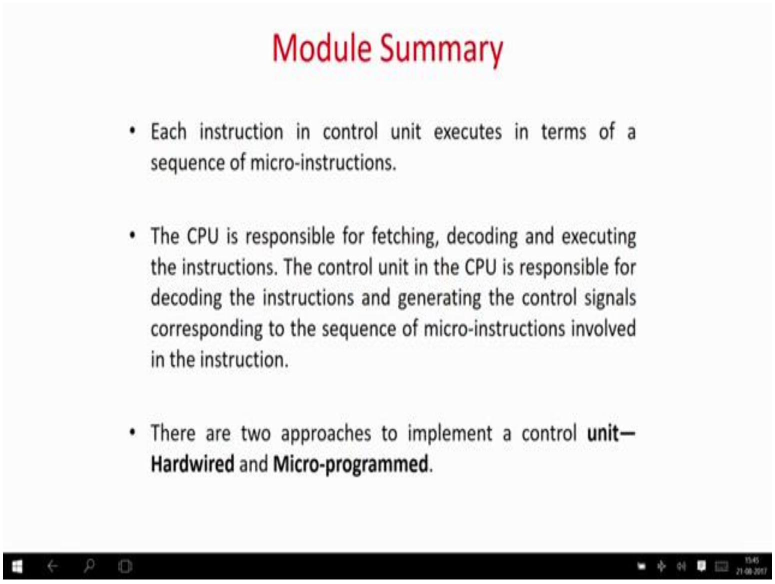
Then we will actually see different functional arithmetic logic units in terms of different control signals. In other words here in this module if you compare to the previous module we will look at the same thing, but in the last module we assumed that data is fetched, then if there is a control instruction and the flag registers are set accordingly program counter will change, but here we will deal with how the check actually what are the control signals that result in the check and how those control signals are generated?

Like for example, as I told you if there is some flags. So, flags are set and reset they generate some control signals. How those control signals will be utilized and how this how you can design a circuit, which will read the control signals from the wire and modify the program counter.

Now, we will be looking in a much more hardware related issues than a superficial level which we last looked in the last module. Like for example, the output of the control units determine the signals which actually are subdued to another function module. Like for example, these are peripheral device. So, the peripheral device will interrupt your system.

So, if it is interrupted what type of control signals will be generated? So, all the in a nutshell we look at different type of control signals how they are generated and we look it for different architectures like single bus, multiple double bus and three bus architecture.

(Refer Slide Time: 07:01)



Then basically another very important thing is that whenever how they whenever we will be talking about the control signals, we have to first understand that like a very simple instruction, which we have already seen in the last module. Like ADD A, B or ADD A, 3030 hex that is memory location. So, in fact, there is something called a macro instruction, but a macro instruction will also involve some kind of micro instructions. Like when we say ADD A, 3030 hex; that means, first there should be a micro instruction which will be involving to read the data from the memory location 3030 to the memory to the accumulator or in this case register A or whatever may be instruction like.

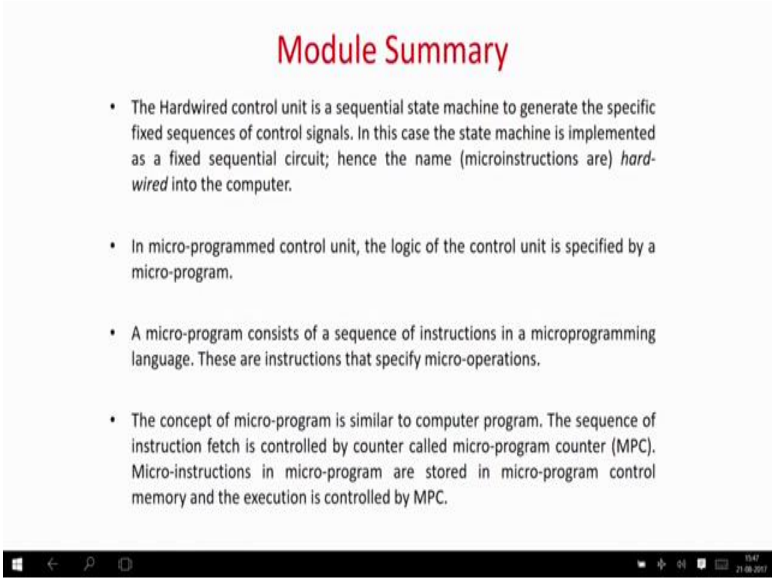
So, in fact, that will involve some kind of micro operations or micro instructions like first we have to give the address of 3030 to the memory address register, then you have to wait for certain amount of time then the memory will generate a signal that data has been given to the memory buffer register from 3030. So, that is a control signal which has to be read from the memory saying that data from 3030 location has been given to the memory buffer register. Now your instruction register or your accumulator will read the data from the memory buffer register.

So, the small small steps like then; obviously, after instruction is executed you have to increment the value of PC. So, a larger instruction like ADD A, 3030 involves several micro instructions. As I told you incrementing the PC, writing the value of the address in the memory address register, reading from the memory buffer register after the signal the memory gives a signal that I have written the data.

So, all this small small instructions we call as micro instructions and we assume that each micro instruction can be operated at one unit of time. So, basically in this module a substantial part will cover that given a macro instruction or otherwise an instruction; like load, ADD, subtract, multiply, what are the micro instructions involved in it how the how what are the different micro instructions for a macro instructions?

How they are sequenced? How they can be optimized? And how basically we can write for a given code? How can we write it in terms of micro instructions or how basically in fact, you might have observed that I was telling you that for each microinstruction some kind of control signals are generated? Then we will see that basically for a given code what are the micro instructions and then we will study basically how we can generate those micro instructions?

(Refer Slide Time: 09:27)



So, micro instructions basically there are two approach one is called the hardwired approach and one is called the micro programmed approach; which basically generates the control signals depending on your micro instructions.

So, in this module we will be looking at depth in both the ways. So, in a hardwired control based micro instruction or control signal generation means; we will have a final state machine, which is a hard coded machine, which is implemented in the hardware and based on your micro instructions or the micro instructions corresponded to the main instruction, the your flow will move in this sequential machine, because we know that it is a sequential operation. So, a sequential state machine will move and each state will generate the control signals. And we called it as a hardwired business, because it is fixed to the hardware and you cannot do any changes there.

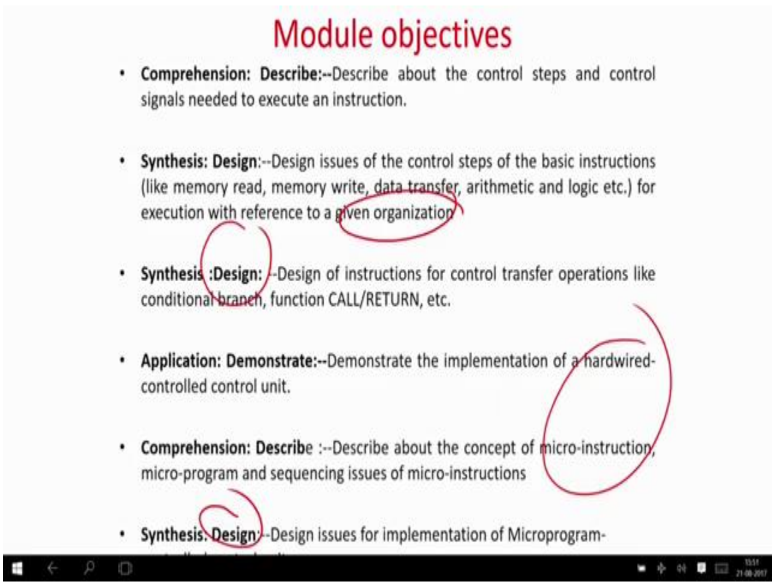
But of course, a more flexible version of this is called microprogrammed environment. So, in micro programmed environment, you can assume that basically the same thing like a normal code involved, but in this case your micro instructions would not be hardwired like a hardwired control, but you will have a another simplified set of codes for each macro code or in fact, for a given instruction which I am for the timing calling macro instructions, it will involve some kind of micro instruction.

So, micro instructions will be very similar to a normal set of instructions and, but instead of a program counter, which is involved for the macro instructions here there will be something called the MPC. Who is called the micro programmed control instructions construction unit?

Main memory it will be a separate memory for this micro instructions or in other words in case of micro programmed based unit you can assume it that for each macro instruction there is a simple code corresponding to the micro instructions and it will also execute in sequence like the macro instructions, but instead of large instructions like ADD A, B here we will have small small instructions like; load memory address register from the instructions register.

Load the value from the memory to the memory buffer register and read the signal; that means, now the instructions are at the micro level, which can be executed at each at each time unit, but you can think it like macro ins macro instruction 1 2 3 4 each macro instruction will lead to some micro instructions. It will also be stored in a microprogrammed memory it will also have movement like 1 2 3 4 and instead of the program counter here we will have the micro program environment. So, it is more flexible. So, more details will come up when we deal with the absolute exact modules will reach through.

(Refer Slide Time: 11:45)



So, now if you look at it, what are the objectives of this module? So, the objective of the module first it is a comprehensive objective you will be able to describe about the control steps and control signals needed to execute an instruction this is one of the most important part of this module. That will give an instruction sometimes I will call it macro instruction to differentiate it from the micro instructions. So, if I give you an instruction like I say that load accumulator 3030. So, you will be able to tell me what are the exact control steps and what is the exact control sequence required to do it; then this is a synthesis objective. So, synthesis objective says that design issues of control steps of the basic instructions like read memory for execution with reference to a given organization.

That means you will able to design a system or a computing control unit system, with given to a given organization; that means, if it is a single bus or multiple bus, then you will be able to design the control steps required you can also design the micro instructions and the control signals required to be generated it can be a micro programmed control, it can be hardwired control.

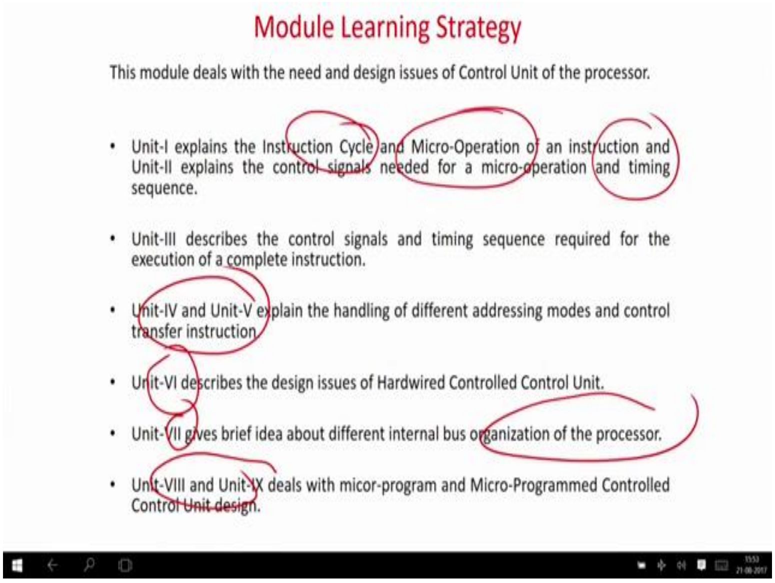
So, you will be able to design such a system based on a given organization. Of course, you will be also able to design that is another synthesis objective design instruction for control operations like branch function call etcetera. So, in fact, I mean the 2 design here we have separated out the 2 in I mean synthesis objectives.

One is for the normal arithmetic, data transfer operations and another set for branch instruction, interrupt instruction and call return instructions.

Then of course, 2 important things you will be able to design this control signal based on hardwire based control unit and micro program based control unit that is the hardware business as well as software business. And the last instruction is again a design instruction. So, you design issues for implementation of the micro program as well as your hardwired control unit you will be able to design both, compare among them both and find out which is the more optimized implementation at any point of time.

So, this is actually the module main module objectives.

(Refer Slide Time: 13:49)



Now, look at the module learning strategy. So, basically unit 1 and unit 2 will be basics of this module, where we will study instruction cycle and the micro operations of an unit. Basically in first unit we will be learning that what is an instruction cycle that we know fetch, decode, execute sometimes you have an interrupt and so forth and what of the micro instructions available for them.

And the second one basically we will deal with timing sequence, that exactly what is the time cycles and more integrated details of the control signals will be studied over there.

So, first unit will tell what are the signals? The second unit will tell what are the exactly timing sequences in terms of timing diagrams? Unit 3 will describe the control signals I mean timing sequence required for a complete execution of a continuous instruction. In 1 or 2 will be building the basics and in third unit we will take a large code and we will take an organization like a single bus multiple bus and we will see how it goes?

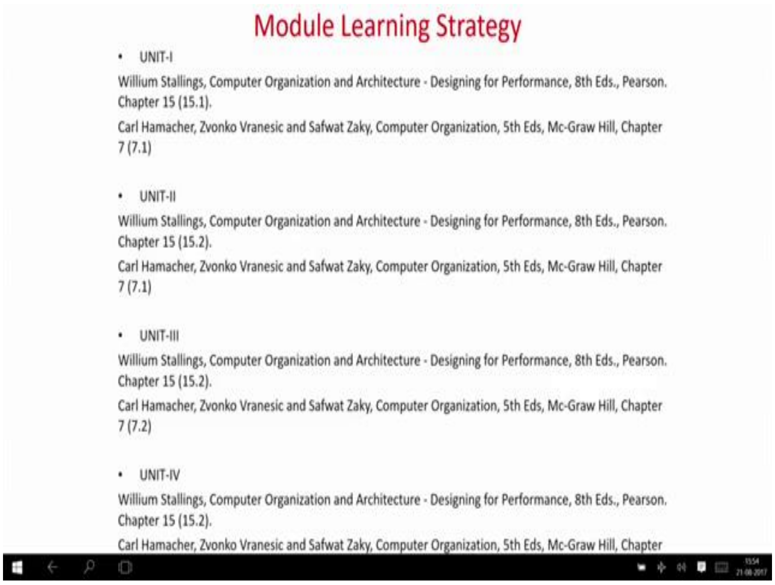
Instruction 4 and 5 will take different addressing modes and we will study the same things. Like we have already seen different type of instruction modes are available oh sorry different type of addressing modes are available like direct, indirect, base, displacement.

So, we will see how the control signal generation changes when whenever there are different types of addressing modes. Instruction 6 will describe about the hardwired control unit. So, now, in the instruction 6 we will deal with, but if you are given a set of instructions, if you want to make a hardwired control unit how to design that? Inst unit 7 we will tell you about the different type of bus architecture, basically as I was telling that we will study with in depth of different type of single bus multiple bus and we will study for all of them how to design the control set instructions? And last 2 units will basically deal on deal with how to design micro programmed control units for different bus organizations.

Basically that is what is the idea. So, initially we will start with very basic instruction set what are the micro instructions for that, then we will look at the timing sequence then we will see if for different set of instructions or different addressing modes how they change, then we will give you an idea of hardwired control based design and then we will also give you an idea for how to design a micro programmed control based unit for generating different type of signals. And control signals in a control unit and also we will look at different architectures in terms of different bus system buses single and multiple.

So, basically I am not going to read out the slide you can look at it we are describing here basically what are the, what are different units?

(Refer Slide Time: 16:19)



And from which book you can study this basically we are referring William Stallings book and Hamacher’s book. So, exactly which unit which topic and which module you have to read like for example, in unit one which chapter you have to read or all the detailed down in these slides basically.