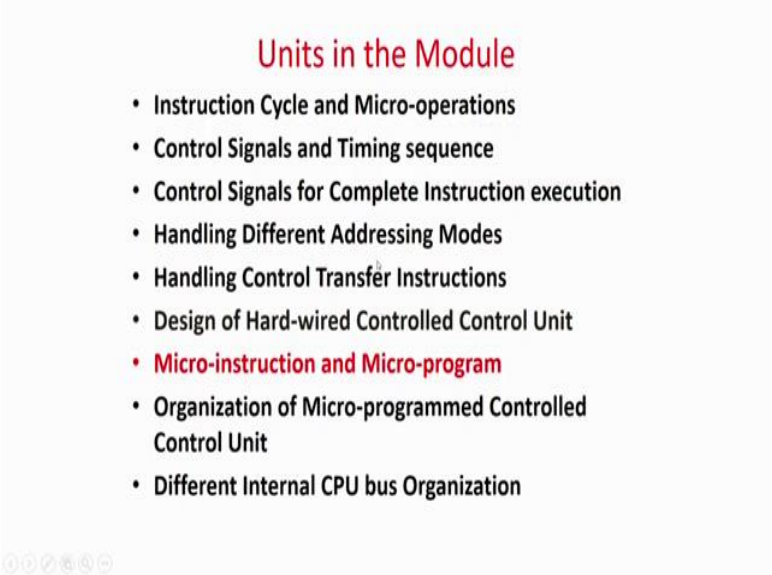
Computer Organization and Architecture: A Pedagogical Aspect

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

Lecture - 21
Microinstructions and Microprograms

Welcome to the 7th unit of the module we are discussing, that is on the control circuitry of the computer. So, this is 7th unit in which we are going to study about micro instructions and micro-program.

(Refer Slide Time: 00:42)



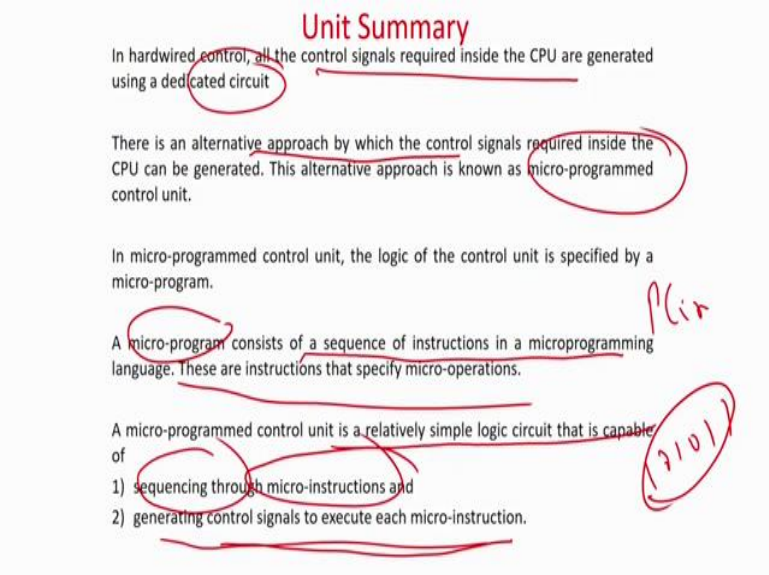
Basically, in the last unit we have seen that basically how to generate the control signals, if the hardware for this is exactly fabricated as a hard-coded non-modifiable circuit. Which you actually call as the hardwired control unit, that that is the sequence of micro instructions and the control signals to be generated corresponding to that can be generated using a hard-coded circuit, which is synthesized from its finite state machine control.

So, that part we have seen that the circuit we generates out of this is a non-flexible circuit, but it is very fast, but it is non-flexible and it cannot be changed. In fact, for a given sequence of micro instructions and the control signals correspondingly we generate a finite state machine, and then we synthesis the circuit out of it.

In the next units basically, that is on micro-program control, that is these 2 units. So, basically, we are going to study how the same thing that is generation of the control signals can be done in a more flexible way and in a terms of a program. So, it is not it's very similar to what we understand by a normal computer program, which we have already looking throughout these lectures. But instead of the macro instructions we will be using micro instructions, and that control we will actually call micro-program based control.

So, this actually unit is focused on the generating of the control signals using a micro-programmed control which is more flexible, you can change it, but of course it will be slower than a hardwired circuit.

(Refer Slide Time: 02:03)



So, what is the unit summary what we are going to look into that. Basically, if the control signals are generated from a dedicated circuit we call it as a hardwired control. And alternative approach is basically which we can generate such signals which are basically programmed into some kind of a memory. So, it is if it is a memory-based logic in which each of the memory cells or each of the memory word has the corresponding control signals to be generated, then that approach is actually called the micro-programmed control unit. Which is basically we are going to study in this unit.

So, basically a micro program consists of a sequence of instructions, and basically these instructions are nothing but which are the basically a micro-program corresponding to a basically a sequence of micro operations that is very well known, that a macro instruction has

some micro instructions which we have already discussed few lectures back. And corresponding each of the micro instructions or the micro operations basically we have some of the control signals, like for example, PC_{in} should be equal to 1 etcetera. So, those bits we can actually make them 1 and whichever are not required we make them 0, and store in one word of a micro-program control memory.

So, micro-program control memory is very similar to a normal memory, but we allocate it separately for the micro programming based control. So, we call it as a micro-program memory. So, if we can put these control signals explicitly in some memory locations, then when you access that word of the memory such the corresponding control signals will become 1 and 0 as required.

So, that is basically the idea of a micro-program based control. For each of the micro instructions and the control signals that has to be made 1 and 0 you rather write into 1 word of the memory. And then step by step you actually go from one location to another and so forth. So, basically a micro-program control unit is a simple logical circuit. In fact, it's a memory with some peripheral circuits, which actually will go through sequence of micro instructions. In fact, there is nothing but 01011 these are some of the control bits, or which have to be made 0 and 1 based on the instruction micro instruction. They are written in a memory and when you access that memory those values will be given as directly output as control signals.

And therefore, you have to you have to go in sequence that is one job of that logic, and then basically sequence I will tell you what is sequencing and control signal generation. It has 2 parts basically that is this control signals has to be generated in sequence.

So, control signal generation, generation of the control signal is very simple, that is just this values has to dumped from the memory and it has to go to the corresponding locations. So, in this case generating the signals is very straightforward, unlike in a hardwired based control where we have a finite state machine and the corresponding Moore or Mealy machine logic has to be implemented, here it is very straightforward. The values are already in the memory and you dump it out.

Sequencing actually slightly tricky, which is somewhat very easy in a finite state machine approach because there you have the flow of states which can take care very easily, but in this case, sequencing is slightly tricky, because unless until unless otherwise specified you will go from step 1 to step 2 to step 3 that is the sequential memory locations. Micro-program memory

locations like step 1, step 2 and step 3 and so, forth. But, whenever there is a jump instruction and you require also to check some flag conditions, which is simpler in finite state machine based approach.

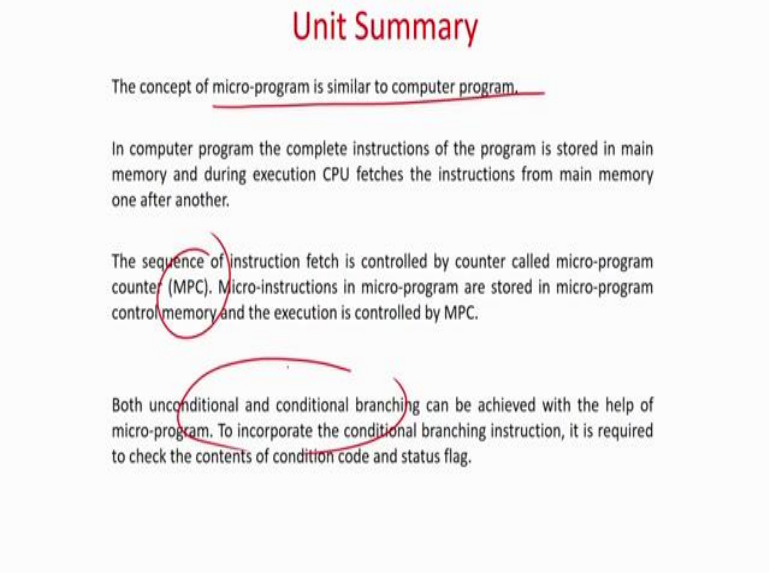
But here, we have to have separate arrangement. So, that just by looking at the control signals the that is the input output signals, then you have to also look at the flag signals, then you decide that whether the next location is the location which has to generate the corresponding signals, or you have to jump to some other location, where the control signals has to be generated accordingly.

So, basically micro-program based control is very simple, whatever signals you require you directly put them in the memory location. So, whenever you have to generate those control signals that memory word has to be fetched, but only sophistication is that you in normal case you will move through step 1, step 2, step 3 like that in the memory control memory sequentially. But whenever you have to jump you have to go to some other corresponding memory location. So, accordingly the program counter we call it actually here micro-program control counter, because the architecture is very similar to that of a normal program execution.

So, whenever you have to jump to others. Slightly more sophisticated program mechanisms has to be used which will take the values from the input output signals as well as the flag signals and take decision accordingly. So, there is a slight difficulty compared to the hardwired control.

But, simplicity here in terms of if you compare to hardwired control is that generating signals is very simple. There is no circuit explicitly required, required values are stored in memory locations, which are directly fetched, and the values given. That is what we are going to see in details.

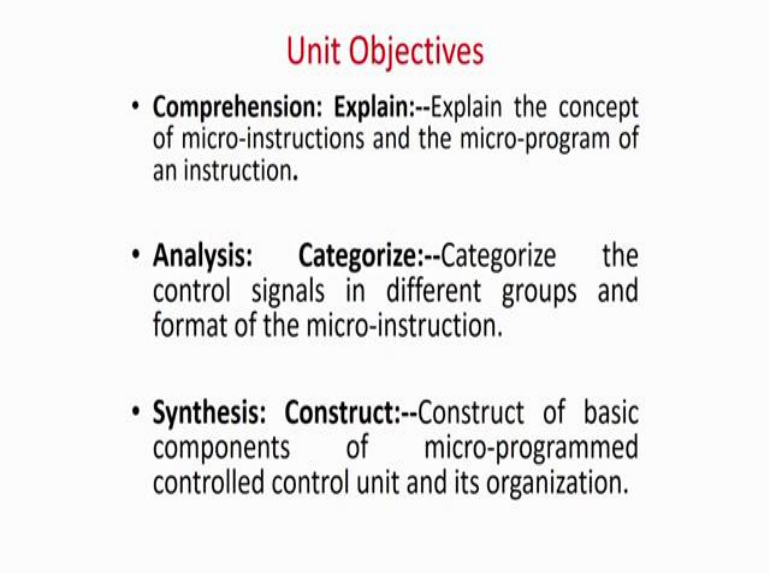
(Refer Slide Time: 06:45)



Basically, we can also say that micro-program is very simple to a computer program. In case of a computer program the whole macro instructions are stored in a memory, and you just actually fetch it one by another. And whenever you require a jump instruction, you jump that is actually controlled by the program counter. Here, also it is very simple very similar basically instead of a program counter we call it is a micro-program counter and also we have a explicit memory which will actually store this control signals corresponding to each of the micro instruction, and that is actually called the micro-program memory.

So, there is slight architecture is more or less similar, but before the term because, the flow of instructions which happens in a normal macro program is very similar in a micro program, but we use the word micro-program to differentiate basically. Similarly, it has more conditional and unconditional branching. So, all these are very similar. So, that is what we are going to look into this look in this unit.

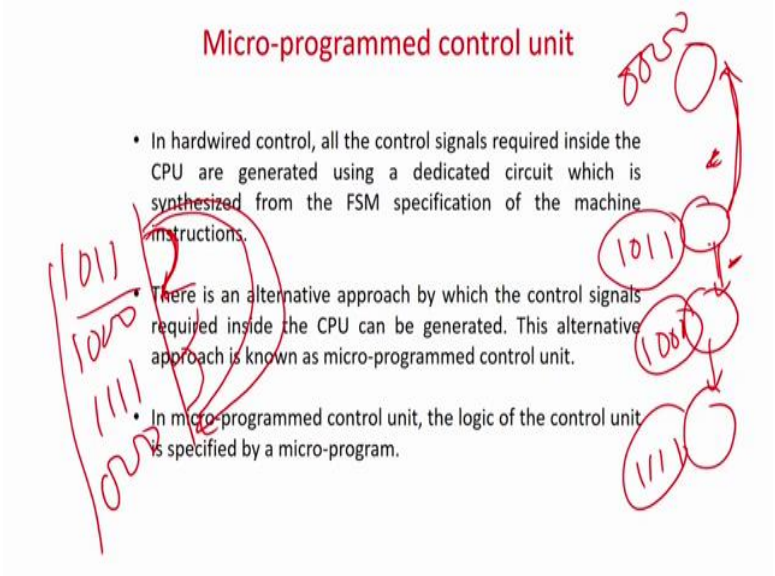
(Refer Slide Time: 07:37)



So, what are the basic objectives which we are going to fulfill after doing this lecture? So, it is a first is a comprehensive objective in which case you will be able to explain the concept of micro instructions, and the micro-program of an instruction that is given an instruction. You will be able to explain that what are the micro instructions corresponding to that. In fact, we have already learned it in the fewer some classes back, but in this case also you will be able to translate it into a micro-program.

Then, next is an analysis objective you will be able to categorize the control signals in different groups, and the format of micro instructions. That is given some instructions macro instructions you will be able to generate the micro instructions and the micro programs out of it. Synthesis there is a synthesis objective construct construction of or you will be able to construct, basic components of a micro-programmed control unit and it is organization. That is, you will be able to synthesize a whole micro-programmed control unit given a set of instructions. So, these are the basic objectives of this unit.

(Refer Slide Time: 08:34)



Let us go into the unit in details. So, what is a micro-programmed control units? So, already we have seen that, you know finite state machine approach or the hardwired based approach basically, we have some states which are synthesized as a hardware. So, you cannot change this and but it is quite fast that is the advantage, but problem is that for a given micro instruction sorry a macro instructions all sort of instructions this flow is very much fixed, that cannot be changed.

So, to have a more flexible approach we actually go for something called a micro-program based approach. So, let us assume that it generates some signals called 1011 may be the first signal is for program PC_{in}, second is for PC_{out}, this may be for MDR_{in} some control signal values, may be this state generates 1011, next state may be 1000, next state may be all 1's some arbitrary values it is generating.

So, in this case you just move from state 1 to state 3 and may be on some other condition check you can very easily implement, it can go to some other state where we have all 0's. So, we can have some condition check over here based on the inputs and the flags, but in case of micro control ah so in case of hardwired control, they are some it's a Moore or Mealy machine, there are hardwired circuits which will actually generate these values based on the state variables, or the state encoding process.

But, in case of micro-program we actually store these values called 1011, 1000, 1111 and all 0's is some memory location that we are calling as the micro-program memory. Then we go

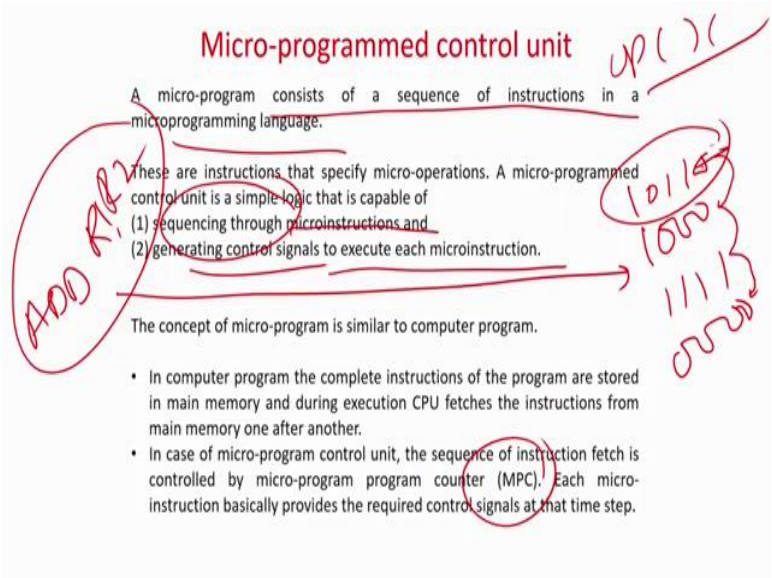
from this step to this step, this step to this step, this step to this step sequentially using a micro-program program counter.

So, in micro-programmed architecture if you store these values in a micro-program memory, where it is very simple just you go for one memory location to another fetch the values, and automatically control signals are generated. But, what is difficult in a micro-program approach compared to ah hardwired approach is there. For example, going from here to here or here may be depending on some input values or a flag values simple because, it's a hardwired circuit which is implemented.

But, in this case it's a memory which actually goes here. So, either you go from here to here, or you go from this location to this location, for that basically you should have some sophisticated arrangement. Which we will see which is not very straightforward, which we will have we have to have some more arrangement, so that depending on some kind of input values you decide whether this is the next instruction, or this is the next instruction.

So, for micro-programmed control unit when you are implementing a branch kind of an instruction, or where conditions has to be checked we have a slightly round around arrangement. But, it is straightforward in finite state machine, but overall micro-programmed is very flexible because, you can easily change the code if you like based on the requirements. But, the sequencing is slightly tricky which is more difficulty in case of ah fi sorry micro-program jump instructions based on some condition check is slightly sophisticated from that of finite state machine. So, basically that is what is a micro-programmed control unit.

(Refer Slide Time: 11:28)



So, basically as I told you the micro program consists of a sequence of instructions in a micro programming language that is correspond to a micro instruction ah micro instruction there are some micro instructions and basically, you have to sequence through the micro instructions and generate control signals to execute. So, that is a two jobs that is say for example, like as I told you say these are the signals which has to be generated. So, sequencing through the micro instruction is or generating the control signals means, you if you read this memory location. So, these values will be coming out from the memory micro-program memory. So, that will set the values accordingly.

So, generating a control signal for each micro instruction is simple, just fetch the memory location. Sequencing means, I generally sequence like this, but whenever there is some jump instruction you have to follow on you have to move from one instruction to another based on which is non-sequential based on some input variables then somehow I have to change the value of micro-program counter which will not be 1, but it will go to some other location.

So, in that case sequencing is slightly more sophisticated. Again, as I told you the micro-program concept is very similar to a normal program. So, the normal programs basically have some instruction which is fetched, decoded, it goes through an instruction register and it happens in that way, but same thing happens here, but yeah there like in that in a in a macro instruction, you have an opcode, then you have operands and so, forth.

But here, there is no questions of any opcode and operand basically. So, what happens I will just tell you one example may be say, we have the instruction called $ADD\ R1, R2$. So, this is a macro instruction. So, in that case you will have the when it goes to the instruction register, you have the opcode and you have the operands.

But in case of a micro in case of a micro micro-program, basically you have some bits and bytes. So in fact, these are also bits it into the in terms of bits. But here, we have opcode and operands and have separate meanings of different fields. But, in this case what happens generally there is one set of fields which will just correspond to the control signals directly. That is, you will directly feed these to different control positions like program counter in, ALU to be added etcetera.

There's a slight difference, but the control flow is very similar. You go from one instruction to another, then to another if required you jump. But, basically also it's the program counter which will take care of this, but that concept remains same. But here, you did not decode likewise that it is corresponding to add or store or something like that, directly control signals are generated in a micro-program.

But the flow architecture is same. So, therefore, we call it there is a micro-program PC that is micro-program program counter basically, and also this memory we call it as a micro-program memory to slightly differentiate. Flow is same but only the purpose is bit different. So, in case of a normal macro program you have $ADD, STORE, LOAD$ such that when instructions are executed you have an opcode and different operands. So, you have to use this opcode go to a address instruction decoder, and from there you will be able to generate some kind of control signals. But, in case of micro-program the control signals are actually basically are provided in the instructions itself, because macro instructions actually get translated into micro instructions, and they actually are encoded in terms of control signals ok.