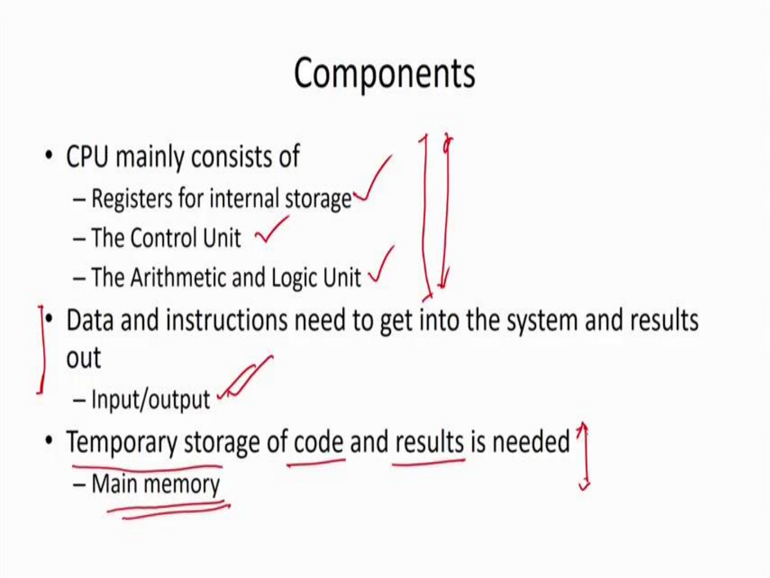
(Refer Slide Time: 29:05)



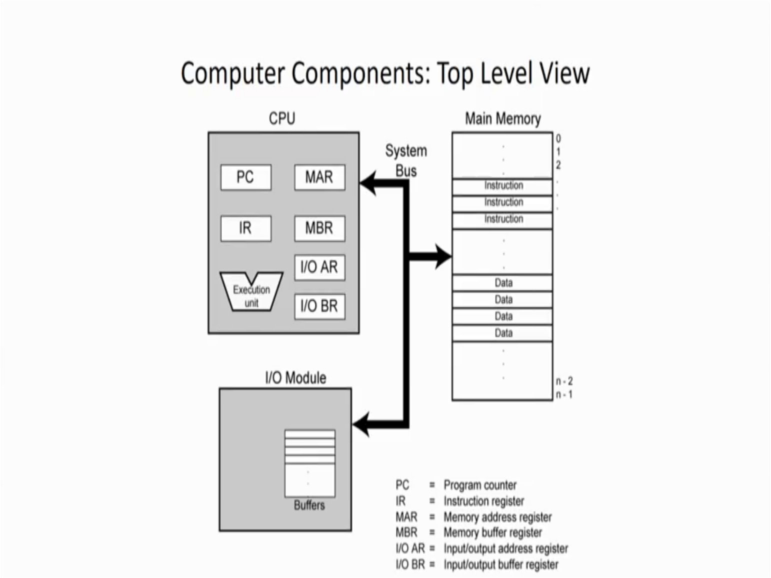
Components

- CPU mainly consists of
  - Registers for internal storage
  - The Control Unit
  - The Arithmetic and Logic Unit
- Data and instructions need to get into the system and results out
  - Input/output
- Temporary storage of code and results is needed
  - Main memory

So, now in a nutshell now we can say that what are the components that we are having inside the processor. So, CPU mainly consists of register for internal storage, it is having an control unit, and it has an arithmetic and logic unit. So, these are the three major components that we have inside a processor; and these components are connected through our interconnection network. Now, to work with this particular processor, what will happen we have to take bring the information inside that processor. So, for that somehow we need to bring this particular information and somehow we have to give the output to the users. So, for that we need this particular input output mechanism.

So, in a simple example, I can say that keyboard is my input device. So, through keyboard I can give the input to the processor and monitor is an output device to monitor I am going to get it. Secondly, and another way we need one more component which is known as a temporary storage for code and result and it is known as my main memory, because the computer works in Von Neumann stored program principle. So, you have to keep those information in the main memory. So, along with the processor we have input and output and we have main memory or storage unit.

(Refer Slide Time: 30:36)



Computer Components: Top Level View

PC = Program counter
IR = Instruction register
MAR = Memory address register
MBR = Memory buffer register
I/O AR = Input/output address register
I/O BR = Input/output buffer register

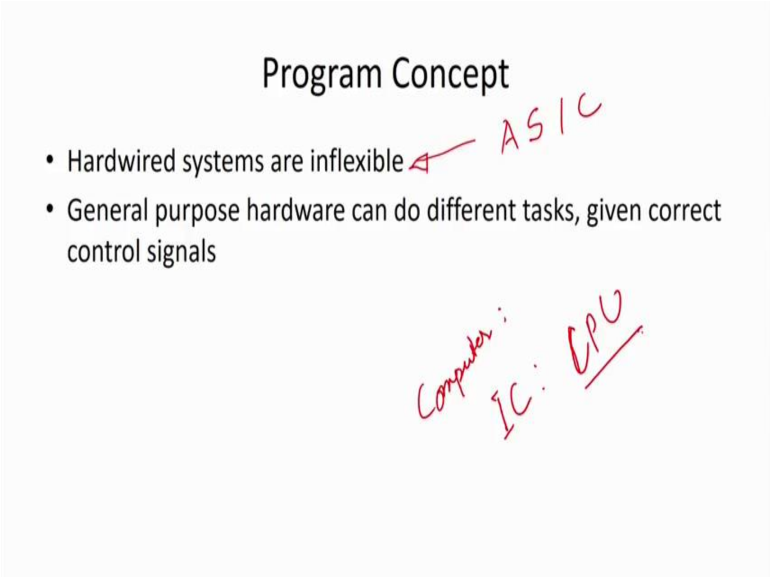
So, in this particular way now we can see the components of my computer it is a top level view. So, this is the processor CPU. So, CPU we are having some internal registers called already have talked about PC is nothing but program counter, IR is nothing but instruction register, MAR is your memory address register, MBR is your memory buffer register, along with that we are having two more additional register. We are talking about I/O AR that means I/O address register, and I/O BR I/O buffer register. So, these are the special purpose register along with that we are having some general purpose register also and this is the execution in basically ALU.

So, it can perform work then we have to bring the information from main memory. So, this memory will be connected to this particular processor we call this is the system bus. So, through bus we are going to connect this particular memory. Already I have talked about bus is nothing but the connecting wire inside the processor also we are having an internal bus, through that internal bus we are going to transfer information from one component to the other component. In case of your computer through the system bus we are going to transfer information from main memory to the processor.

Again that I/O module input output module will be connected to this particular processor through this particular system bus and all the input output device will be connected to this particular I/O module. So, we are going to discuss these things in detail in subsequent modules.

So, this is the top view of our computer and these are the components having processor, main memory and I/O module.

(Refer Slide Time: 32:17)



Program Concept

- Hardwired systems are inflexible ← ASIC
- General purpose hardware can do different tasks, given correct control signals

Computer :

IC : CPU

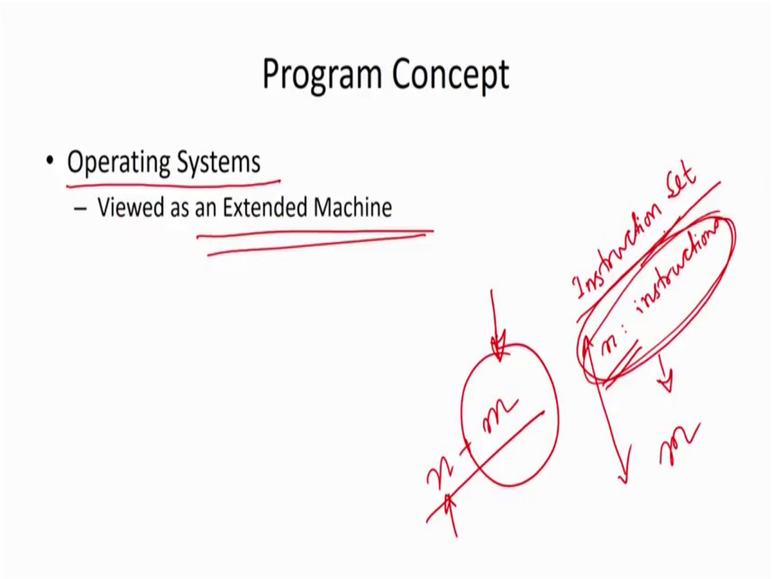
Now, what is a program concept? Now, why we say talk about a program what is a computer program? It is nothing but collection of instruction and we are going to perform those particular instruction. So, here we are talking about programming concept and why this programming concept is coming into picture. It says that just I have written it like that hardware system are inflexible but computer is also a hardware device. We build the computer with the hardware which is nothing but your electronics component, but in general we can say that hardware systems are inflexible.

Here I can give you a simple example if you talk about the television, ok TV is also an electronic device. So, we are having several components electronics components inside the television, but television is performing one particular task only. It receives the signals and according to the signal received, it is going to display it in the monitor. But along with that what will happen we are having some provision to change the channel, change the brightness or contrast all those things, but it is only going to perform that particular tasks only receive the signals and display into the monitor. And you can see, but again there is some provision to change the channels, program it with the help of different channel and like that but we cannot do any other work.

But if we are going to look for a general purpose devices then what will happen we must have a provision to configure it according to our requirement that means, we can program it according to our requirement. So, in that particular case we are going to say it is a general purpose hardware. So, computer can be treated as a general purpose hardware because we are having the programming facility and with the help of program we can carry out our tasks so that's why it says that in general hardware systems are inflexible.

So, if we are going to design a hardware circuit for to perform a specific task in that case we are going to say this is your ASIC - application specific integrated circuit. So, like that we are having ICs for our TV television. So, this is basically ASIC the application specific integrated circuit, but when we go for general purpose or when we go for computer then what will happen in that particular case we are having an IC, but this IC is the CPU or processor. So, this processor can be programmed according to our requirement. So, from that the concept of program is coming into picture.

(Refer Slide Time: 34:58)



Program Concept

- Operating Systems
  - Viewed as an Extended Machine

Instruction Set
m : instructions
n + m
m

So, when we are talking about the programming concept one of the issues or all of you know that we are having an operating system. So, when we are going to switch on our computer generally we say that we are booting up our computer. And when we say that we are booting our computer, we are booting it with the help of some operating system. We are having several operating system, and you know that basic one is your Windows, but you know about Unix you know about your Linux like that.

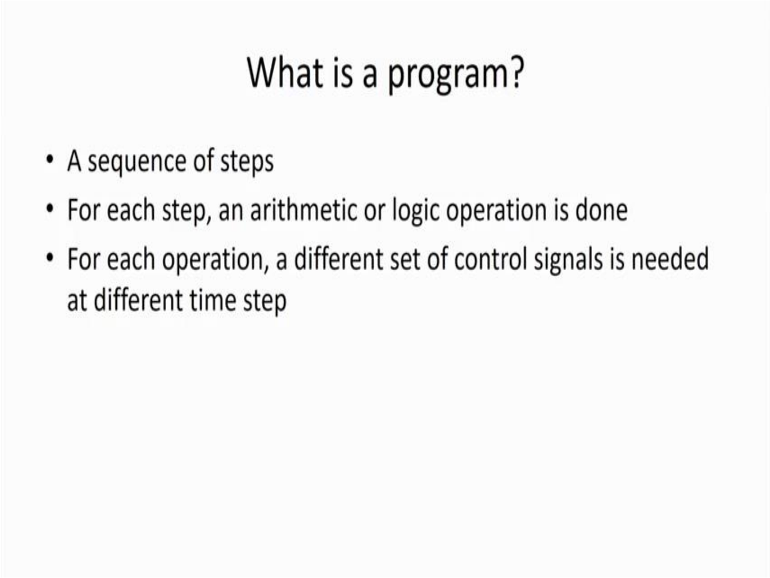
Now, what is an operating system? So, for this programming concept and for this particular your work when we are going to discuss about a computer organization and architecture here we can view this operating system as an extended machine. Why we are talking about that extended machine, because every computer or every processor is having a specific instruction set. So, it depends on the overall processor and we can say we are having n number of instruction. So, whatever we are going to do we are going to do everything with the help of those n instruction only. So, we have to write a program to perform any operation in that particular processor with the help of those particular n instruction only.

Now, one of the issue I can say that now whatever we have inside a memory say I am storing some information in the memory, I want to display those information to the monitor so that user can see it. So, now, when I am going to perform this particular task, taking the information from memory and displaying it into the monitor, I have to write a program. And when I am going to write a program then what will happen I have to know this particular instruction set and with the help of those instruction set, I am going to write a program to display information from memory to the monitor.

Now, what will happen you just see as a common users you have to write this particular program every time. And you have to know that complete instruction set. So, for that what will happen such type of routines, we are going to write in one go and we keep everything in one place and we say that this is that component of my operating system. So, when we boot then all those instruction will be in place, so that I can use it. So, now, that’s here I am saying we are going to view this operating system as an extended machine.

So, whatever n instruction we are having with the help of those instruction we can create another say m number of instruction to do different operation. So, total instruction I can say that n + m, this m instruction are basically some software written with the help of those n instruction, and they are integrated together and kept in the part of this particular operating system. So, this is why we are saying that opening system can be treated as an extended machine and what we are having basically in operating system. It is a collection of some programs and these programs are made of with the instruction of that particular processor. So, these are the programs concept that we have in computer.

(Refer Slide Time: 38:22)

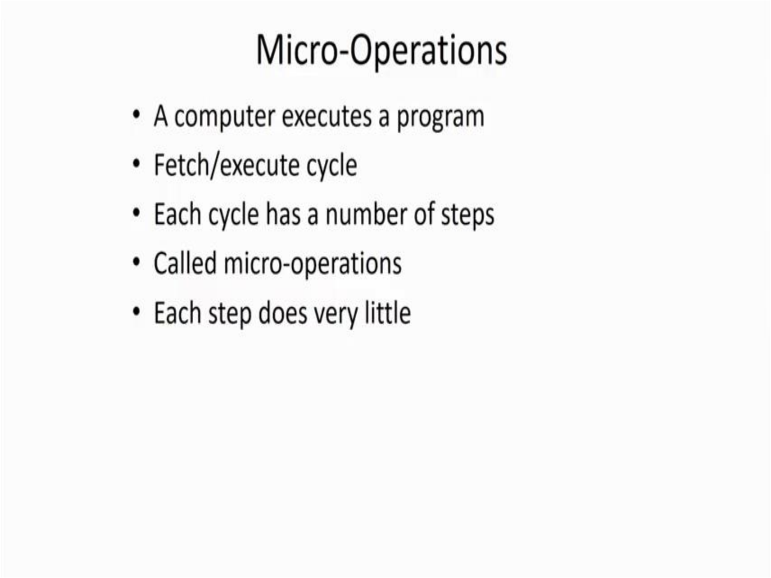


What is a program?

- A sequence of steps
- For each step, an arithmetic or logic operation is done
- For each operation, a different set of control signals is needed at different time step

Now, what is a program? Now, in a nutshell we can say that it is a sequence of steps. So, we are having sequence of step, we have to write in proper sequence. And when computer is going to execute this program then it is going to execute those particular step or instruction one by one. And for each step an arithmetic or logic operation is done because we are having the ALU, we are going to perform some operation with the help of this processing element. And for each operation a different set of control signal is needed at different times stamp, because already I have said that when I am going to perform an or execute an instruction, we have to perform some tasks. And to performance some tasks means we have to transfer information from one point to the other point, we have to perform some operation. So, control unit needs to generate those particular signals at appropriate time.

(Refer Slide Time: 39:19)

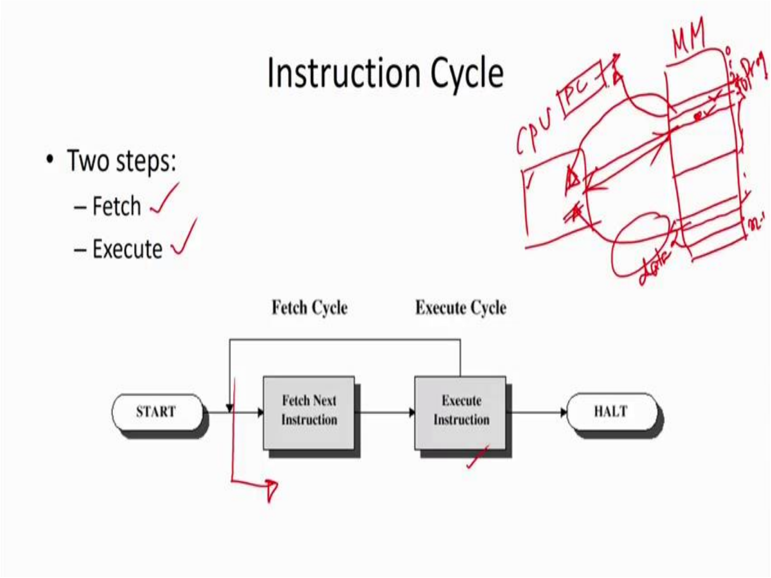


Micro-Operations

- A computer executes a program
- Fetch/execute cycle
- Each cycle has a number of steps
- Called micro-operations
- Each step does very little

Now, when we talk about an instruction in one go we cannot have the effect of the particular instruction, it is going to execute this particular instruction; and at some point of time, I have mentioned that instruction cycle can have two cycle one is fetch and second one is execute. So, we can say that two different tasks we are performing to complete that particular instruction. Now, in every cycle, we may have again several subtypes and combining all those particular subtask, we can say that we are completing the fetch cycle. Similarly, in execution cycle, we are going to perform different subtasks, after completing those particular subtasks, we are going to say that execution cycle is over. So, each cycle like fetch cycle and execute cycle has number of steps and all those steps are doing very little work and we are saying that these all little works are basically some micro operation.

(Refer Slide Time: 40:32)



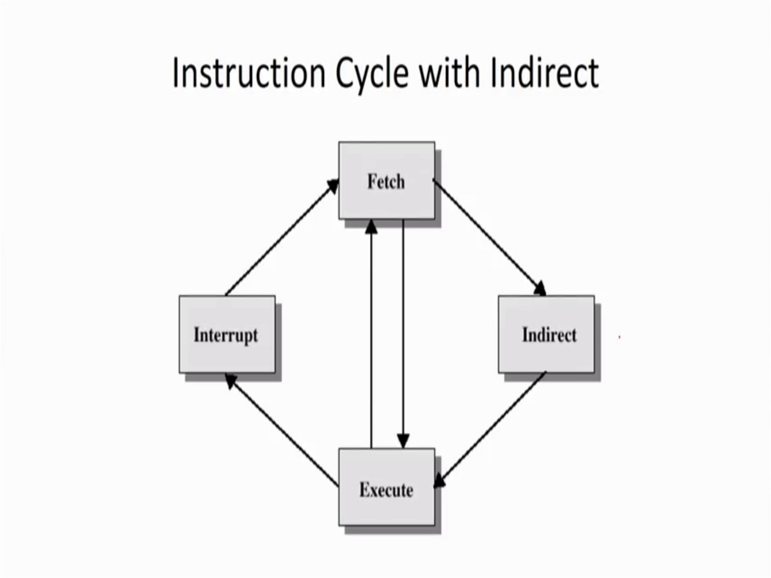
Instruction Cycle

- Two steps:
  - Fetch ✓
  - Execute ✓

So, after combining all those micro operation, we are going to effect of the operation. So, in this way we can see now here we are going to see how we are going to look for those particular micro operation. So, this is the instruction cycle in nutshell I can say that it is having two cycle, one is your fetch, second one is execute. So, what fetch means we are fetching the information, we are fetching the instruction then processor will be knowing what we need to do, and accordingly processor is going to carry out the job and we said this is the execution step.

So, basically where from we are getting it computer works on Von Neumann’s stored program principle that means, we are having this particular processor. And I am having this particular main memory, and it is connected through this particular system bus. So, in main memory we are having this particular program we are storing it. And somewhere we are having this particular data. So, when I am going to execute an instruction that instruction in this particular memory location, I have to bring it to the processor. So, this is called fetch. I am fetching the information. And after that once I fetch it now I know what to do accordingly I am going to execute it; and after completion of this instruction then I have to go for the next instruction because program is a collection of instruction. And we have to execute those instruction in sequence, so that’s why this cycle will repeat. So, now, this is the fetch and this is execute.

(Refer Slide Time: 41:54)



Another one we are having indirect cycle already I have mentioned that sometimes when I am fetching the instruction before execution I have to need the data. So, for getting the data I can go through this particular indirect cycle. So, in this particular case, I just say I am fetching the instruction to the processor. After getting the information that I need to perform some operation and processor now, now see that we have to act on some data, but these data are not available inside this particular processor. Then it will go to the indirect cycle and from to this indirect cycle since I know that my data is somewhere in this particular memory location it is going to bring this particular data inside the processors. So, this is basically indirect cycle. So, in indirect cycle we are going to bring the information inside a processor. Now, once my instruction is available once data are available inside the processor then processor can carry out those particular operation. So, this is indirect cycle.