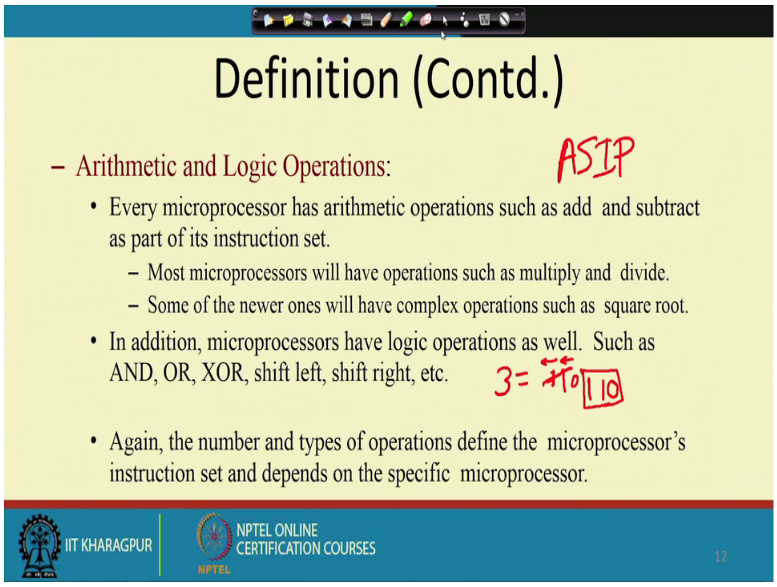
Microprocessors and Microcontrollers

Prof. Santanu Chattopadhyay
Department of E & EC Engineering
Indian Institute of Technology, Kharagpur

Lecture – 08
8085 Microprocessors (Contd.)

So, if we continue with other part of the term that we had in the definition of microprocessor, another thing that we have is the arithmetic and logic operations.

(Refer Slide Time: 00:21)



Definition (Contd.)

– Arithmetic and Logic Operations: ASIP

- Every microprocessor has arithmetic operations such as add and subtract as part of its instruction set.
  - Most microprocessors will have operations such as multiply and divide.
  - Some of the newer ones will have complex operations such as square root.
- In addition, microprocessors have logic operations as well. Such as AND, OR, XOR, shift left, shift right, etc. $3 = 110$ $\boxed{110}$
- Again, the number and types of operations define the microprocessor's instruction set and depends on the specific microprocessor.

So, this is the basic thing like any processor that is designed is supposed to do some operation on the data and what are the operations that can be done, this is decided by the processor design team. So, some processor which is very simple, they are we may be satisfied by in doing say basic operations like addition, subtraction. So, some processors may support multiplication for example, if we look into 8 0 8 5 it does not support multiplication, but if you go to other processors like 8 0 8 6 or say when 8 0 5 1, so, those processors, you will find that multiplication is supported there.

Division is supported by some of the processors. So, that way what are the operations that we are going to support, that is decided by the designer of the processor and if you are supporting more operations means the circuitry associated circuitry will become more complex. So, the cost of the system will go up. So, it is complexity will increase.

So, every microprocessor has arithmetic operations such as add, subtract as part of it is instruction set. So, normally addition, subtraction these are very common most microprocessors will have operations like multiply and divide, but it is not always true some of the newer ones will have complex operation such as square root.

So, you can and if you are designing a dedicated processor for some application, which are known as the ASIP design; if you are having these ASIP designs which stands for Application Specific Integrated Processors. So, for the ASIP design, we have got special instructions. For example, we can say that we have got the one filtering operation that is you are getting the data samples from the outside world and in a signal processing application, we are doing some filtering and all that.

So, there this we entire filtering operation may be a single comment in the response to which the processor executes the filtering routine and accordingly the filtering operation is done. So, they will come under this ASIP design, but those are very complex. If we do not go into that even then this we can have the basic instructions or basic operations by all these microprocessors.

So, apart from this arithmetic operation, there are some logic operations like AND, OR, XOR, then there are some shifting operations a shift left, shift right like that. So, shift left shift right, these are they have got several application. So, one particular application is many a time we need to multiply a number by 2 or powers of 2. So, if you multiply a number by 2; that means, you are actually doing a left shift of the whole thing by 2. So, like say if I have the number say 3; 3 in binary notation it is 1 1.

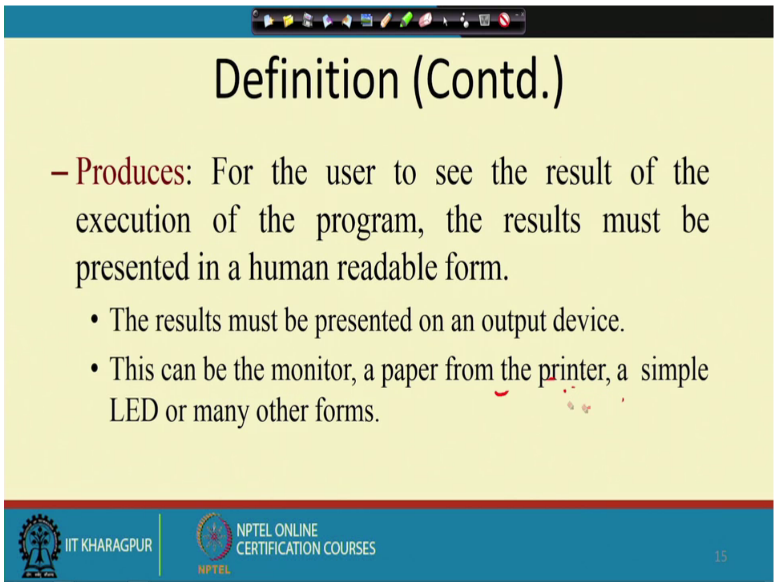
Now, if you are doing a left shift then what will happen, this bit will become 0. So, these ones are shifted. So, this one is shifted there, the next one is also shifted there and then new 0 gets introduced at this point. So, you get the new pattern as 1 1 0. And, if you look into this value, the binary value of this the decimal value of this is 6. So, that you see that it is nothing, but multiplication by 2, that we shift left is a multiplication by 2. Similarly, shift right; this 6, if you shift right by 1 bit position you will get back 3, so that is, division by 3.

So, we have got shift left and shift right for doing the operation. So, that is one type of application of the shift instructional shifts type of operation. Another application that we can have is many times we need to mask out certain bits from them from a pattern. So, there also to prepare this mask this shift instruction will be utilized. So, we will see that later when we go into that type of application.

So, the number and types of operations define the microprocessors instruction set and depends on the specific microprocessor. So, what are the operations that you can do and on what type of data. For example, can we have a microprocessor that performs say the string concatenation operation on some string data? So, does it support string type data and if it supports string type data can it do string concatenation. So, if I design a microprocessor for that purpose for doing this string operation then this string concatenation is definitely one important operation.

So, naturally in that case we will be introducing that instruction into the microprocessor basic operation set. So, this basic instruction set this. So, the instruction set like if you are trying to learn about a new microprocessor, if you learn about it is instruction set and the data type that it can handle.

(Refer Slide Time: 05:47)



Definition (Contd.)

- Produces: For the user to see the result of the execution of the program, the results must be presented in a human readable form.
    - The results must be presented on an output device.
    - This can be the monitor, a paper from the printer, a simple LED or many other forms.

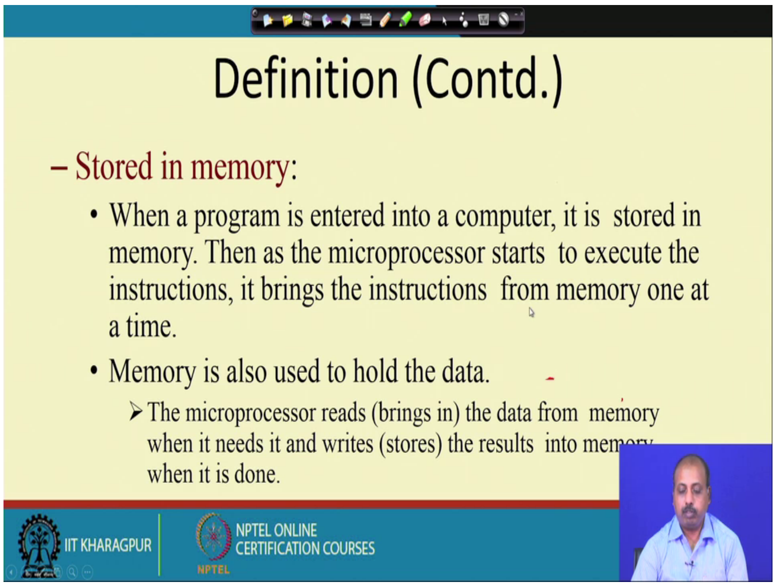
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NPTEL
15

So, if we are going into the definition further say the procedures are the produces the data that is produced by the microprocessor; they are actually for the user to see the result on the execution of a program. The result must be presented in a human readable form.

So, we should be able to understand the value. So, if the value is stored in the memory then a general user will not be able to see that. As a result, there should be some output device on to which the result should be given. So, that result is given in the normally we have got output device like say LEDs, we can have say some simple printer or it can be say a some LCD display. So, we can have some type of display on to which the result will be shown.

So, that way we can have these results are produced by the microprocessor and the produced result should be readable by the human being.

(Refer Slide Time: 06:53)



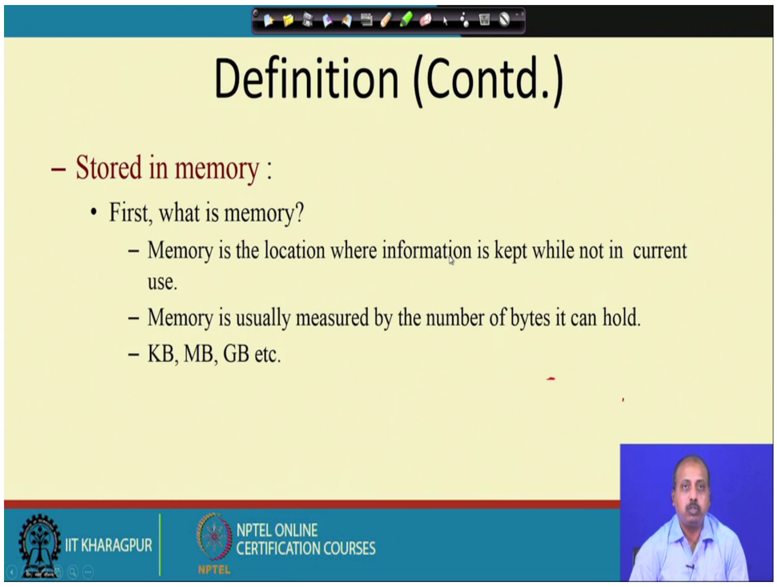
Definition (Contd.)

– Stored in memory:

- When a program is entered into a computer, it is stored in memory. Then as the microprocessor starts to execute the instructions, it brings the instructions from memory one at a time.
- Memory is also used to hold the data.
  - ➢ The microprocessor reads (brings in) the data from memory when it needs it and writes (stores) the results into memory when it is done.

Another important thing that we have not considered is the stored in memory.

(Refer Slide Time: 07:03).



Definition (Contd.)

- Stored in memory :
  - First, what is memory?
    - Memory is the location where information is kept while not in current use.
    - Memory is usually measured by the number of bytes it can hold.
    - KB, MB, GB etc.

So, what we store in memory. We have already seen like what is memory, so, we can just quickly recapitulate. So, this is the location from where the information is kept while not in current use. What do I mean by not in current use? Not in current used by the processor. So, processor is at present not using it. Processor at present is using the data which is inside the processor inside the microprocessor.

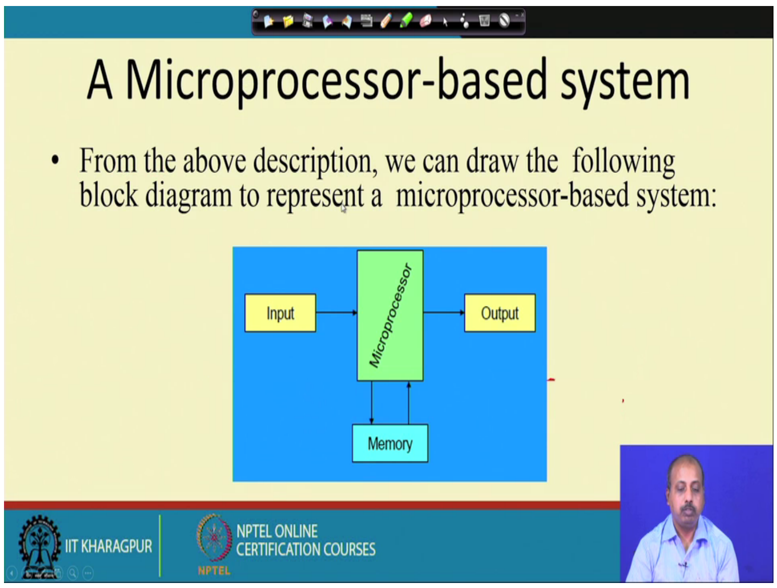
So, it may be in some CPU register or it may be that data or the instruction that has been faced by the processor from the memory, it is in the instruction register or it may be some internal register or it may be some operation that it is doing in the ALU. So, ALU temporary registers are holding those values. So, that way, those are the data that are in use and at the same time there will be other data which are not in use, so, they will be used in future.

So, they are actually stored in the memory and we know that they can hold number of bytes in it; kilobyte, megabyte, gigabyte etcetera. They are the different sizes of the memory that we have. So, what do you mean by stored in memory? When a program is entered into the computer it is stored in the memory, because you cannot store the program in the processor, because processor has got only a few registers. So, if we use that for storing the program then where are we going to store the variable and the data part of it.

So, what is done is that the program is stored in the memory and as I said that microprocessor, any microprocessor when it starts executing, it is actually getting the instruction. It is getting the instruction from the memory. So, it brings the instruction from memory one at a time. So, first it being the first instruction executes it then it plays the next instruction execution that way it goes on. Apart from this program, memory also holds the data. In the part of it is execution if the microprocessor finds that it needs to read some data from some memory.

So, for example, it may be that the instruction that it is executing is adding the content of 2 memory locations and storing the result in a third memory location. So, for executing this instruction the processor will need to get to call the contents of the 2 memory locations from where the data should be taken. They are there to be added and finally, the value should be stored in the in the destination memory location. So, that way it has to have this memory also walks as the place where you store the data.

(Refer Slide Time: 09:59)



A Microprocessor-based system

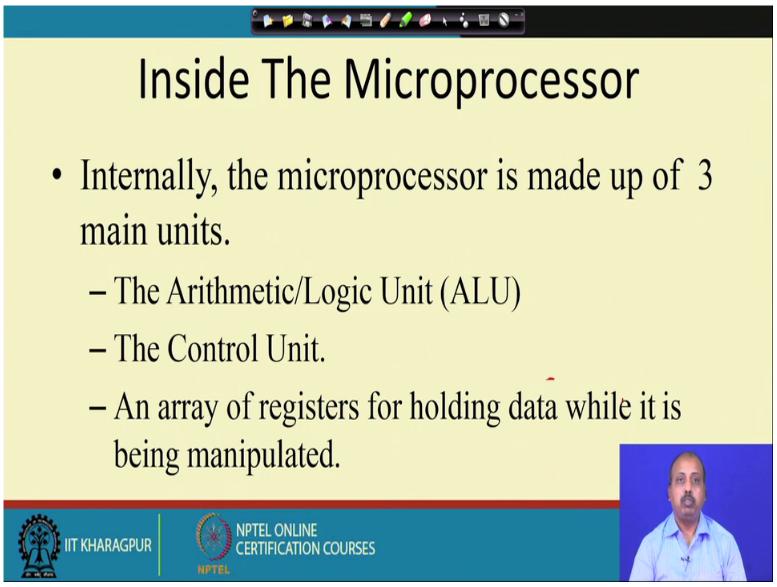
- From the above description, we can draw the following block diagram to represent a microprocessor-based system:

And finally, so, this produces part we have already discussed. So, it is actually showing the result to the user. So, overall diagram, overall picture is like this; so, if this is the microprocessor, it has got with it some input devices by which the input data can be input program data can be taken and it is producing some output which is going to the output device and in this particular diagram the way it is represented, it is somehow the program has been loaded into the memory. Though normally, we have got some facility by which we can load this memory by some other means maybe by some separate computer or by burning the EPROM part of it with the program whatever.

Somehow, the program has been loaded into the memory and this microprocessor it asks the memory to get the next instruction. It gives the next instruction; it starts executing and in the process it may need to access the input device for say getting some different value from the environment or it may need to access the memory to get some operand if it is we were working with some variables doing some operation on that and finally, whatever result is computed if it is required that it will be flashed to the users then they will be put onto the output device.

So, this is the block diagram that represents the microprocessor-based system.

(Refer Slide Time: 11:23)



Inside The Microprocessor

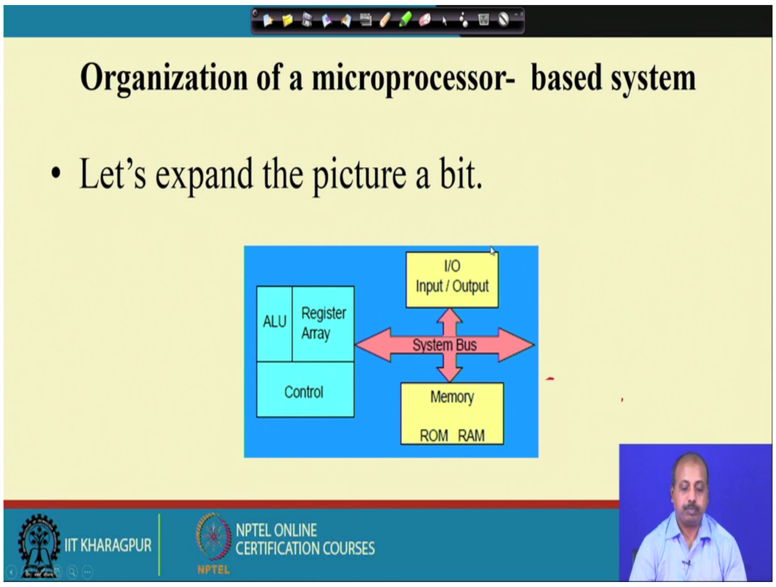
- Internally, the microprocessor is made up of 3 main units.
  - The Arithmetic/Logic Unit (ALU)
  - The Control Unit.
  - An array of registers for holding data while it is being manipulated.

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NPTEL

Now, if we look inside the microprocessor in our architecture discussion architecture portion of our lectures. So, we have seen that any processor is made up of 3 main units; same is true for the microprocessors. The one of them is the ALU, Arithmetic Logic Unit, another one is the Control Unit that controls the operation of all other parts of the microprocessor and we have an array of registers for holding data while it is being manipulated.

So, we have got this ALU, Control Unit and the registers. Now, these registers may be of different types that we will see slowly.

(Refer Slide Time: 12:06)



Organization of a microprocessor- based system

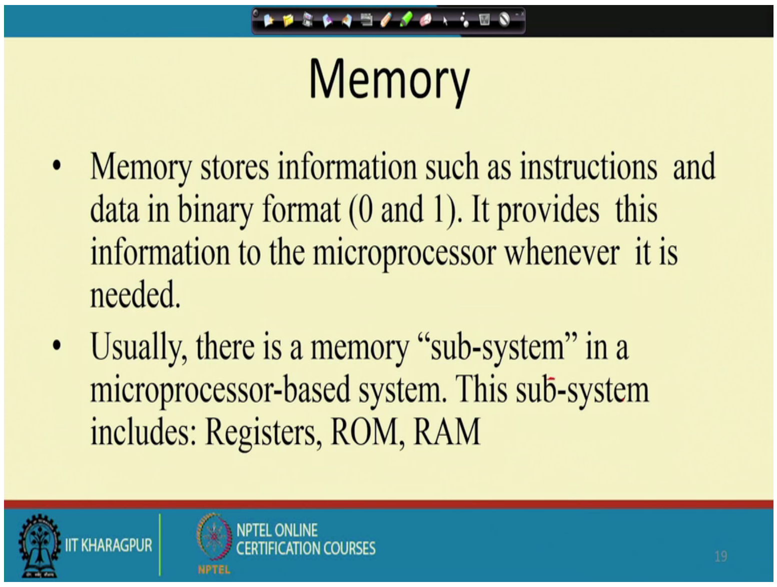
- Let's expand the picture a bit.

So, if we expand the previous diagram a bit then you will see that we have got a situation where this processor. So, it is divided into ALU, register array and control. So, there is a system bus on which this processor is connected this memory is connected I/O devices are connected. So, they are all connected over the system bus.

So, you can assume that as if we have a bus. Bus is nothing, but a collection of wires, a set of parallel wires running in the chip, from this it may be on a board. If I consider this entire system as if as a PCB, these are some copper lines running. So, from this copper line, we have got these individual chips hanging. So, each of this processor I/O device memory, they may be a single chip or a collection of chips and they are all connected to the system bus. So, they are hanging from the system bus.

So, whenever this processor, the system bus consists of say address lines, data lines and control lines coming out of the processor and they are connected to the memory and I/O in some as it is required. So, there are decoders and all that, they are not shown here, but at the block level, this is the situation.

(Refer Slide Time: 13:29)



Memory

- Memory stores information such as instructions and data in binary format (0 and 1). It provides this information to the microprocessor whenever it is needed.
- Usually, there is a memory "sub-system" in a microprocessor-based system. This sub-system includes: Registers, ROM, RAM

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NPTEL
19

So, if we look into the memory in more detail. So, memory stores information such as instructions and data in binary format and it provides this information to the microprocessor whenever it is needed. It stores the instructions and data they are stored in the memory. So, as I said that the processor, the microprocessor when it is reset it will be programmed to access a particular memory location. So, this programming is fixed by the designer of the microprocessor. There is a fixed address of the memory at which it will access and it is expected that the fast valid instruction is there in that memory location.

So, the processor will get that instruction it will start executing it. So, if it needs data it will again access the memory. In this type of design, in this type of memory organization, what is happening is that we have got the same memory containing both program and data. So, program and data, you cannot distinguish between them like if by mistake I have a program in which I sort, say 10 numbers and these programs is loaded from say memory location 1000 onwards and that data set is there from location 1500 onwards. This 10 data set they are stored from 1500 onwards and the memories, the program is stored from 1000 onwards.

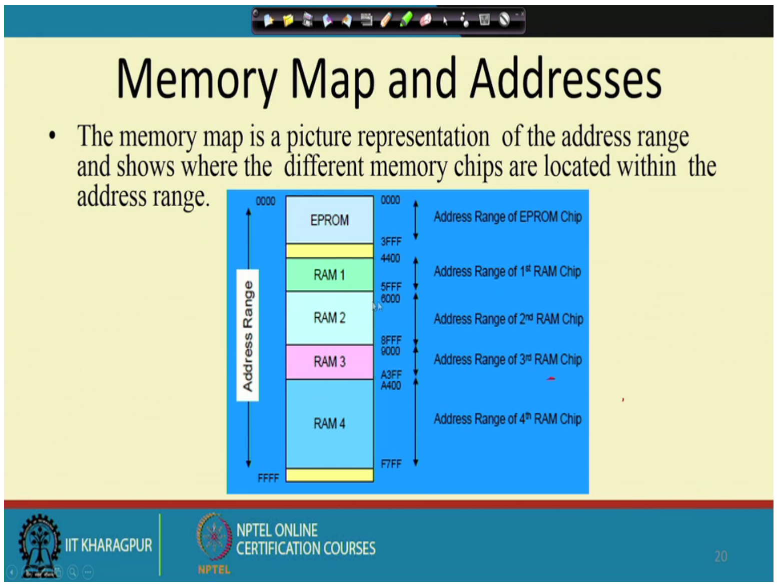
Now, if by mistake the processor is stored that the program is actually from 1500 where I have actually stored the numbers to be sorted. So, if I do like this then it will try to interpret that data as instruction and it will try to do execute that particular program whatever be it is meaning. So, that is the garbage, but the processor has got no other options, it will do that way only. In this type of organization, this type of processor organization what is happening is that the processor is cannot distinguish between the program part and the data part and the program and data they are part of the same memory. So, this type of organization where program and data are part of the same memory, they are called von Neumann organization and the program and data are put into the same memory.

So, you can think about a better architecture which is known as Harvard architecture where this program part and the data part they are kept in totally separate memory chips. So, program part will never mix with data part, as a result the processor cannot do this mistake of taking a data as a program statement or program instruction. So, that type of architecture is known as Harvard architecture. So, the processors that are designed now, they are mostly based on Harvard architecture because of this thing. But, we can have both the types; both von Neumann and Harvard are there, but that is the basic difference.

So, in a computer system, memory is a subsystem; because subsystem means for the whole system memory is a part of it and memory is often a sizable amount of the system. It includes the registers, ROM and RAM. So, you see that it is a subsystem. It is a part of a system, but that system may not be a monolithic one. If you look into the components of the memory, we say that it is registers; these registers such as CPU register.

So, they are actually residing in the CPU or the process within the processor. On the other hand this ROM and RAM, they are typically outside the processor. So, we have got these registers ROM and RAM, they are different part, they may be distributed in the system in various places, but they are essentially part of the same memory, because the registers can also be made to store some value that way.

(Refer Slide Time: 17:34)



Memory Map and Addresses

- The memory map is a picture representation of the address range and shows where the different memory chips are located within the address range.

0000
EPROM
RAM 1
RAM 2
RAM 3
RAM 4
FFFF
Address Range
0000
3FFF
4400
5FFF
6000
8FFF
9000
A3FF
A400
F7FF
Address Range of EPROM Chip
Address Range of 1st RAM Chip
Address Range of 2nd RAM Chip
Address Range of 3rd RAM Chip
Address Range of 4th RAM Chip
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
20

So, this memory map, this tells like how this memory is distributed for a particular design. For example; suppose, I have got a total address range for a processor from 0000 to FFFF, that is, total 64 k that is the address space for the processor and out of that the from address 0000 to 3FFF up in this range we are put one EPROM. So, this is the there I can have some basic programs to be executed. So, that is kept in the EPROM. Then 3FFF to 4400, this part is not used. This address range is not used, but this RAM 1 will be selected if the generated address is in the range of 4400 to 5FFF. So, that is the address range of the first ram chip.

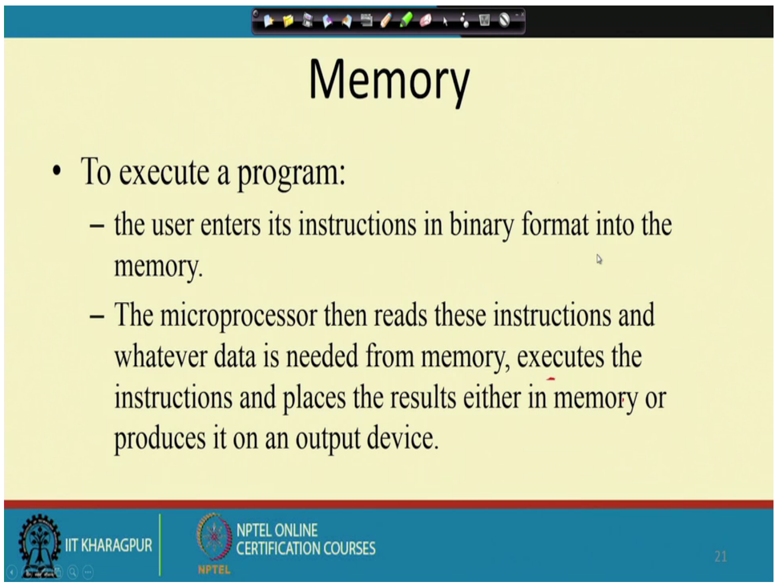
So, this part any address generated between these 2 that is 4400 to 43FF, so, in that range whatever address is generated, there is no memory chip here. So, those should be detected by the operating system and told that there is no such valid chip there so there are address error something like that. So, the RAM 1 is from 4400 to 5FFF, RAM 2 is from 6000 to 8FFF. So, that way we have got 3 – 4 RAM chips in this system. So, they are of different capacity capacities and they are distributed over these address ranges.

So, this is called the memory map of the system. So, if you are working with the system microprocessor based system, you should know what is the address space, what is the address map and of course, there are many other things that you need to know, but this is the first vital information. So, if you are trying to load your program on to this memory you should not try to load it from address say 3FFF or you should not try to load sorry you should not try to load it from 4000 onwards because that space there is no chip.

And there similarly you should not try to load the program from F800 onwards because in this range also there is no chip. So, rest of the places I have got RAM and since we would have we cannot change the content of EPROM. We cannot write on to it until and unless it is accepting the programming part, you cannot update this EPROM part. So, you should not try to write on to the EPROM part, you should only read.

So, once this memory map is known, operating system can introduce those checks and it can introduce the error checks and all that.

(Refer Slide Time: 20:26)



Memory

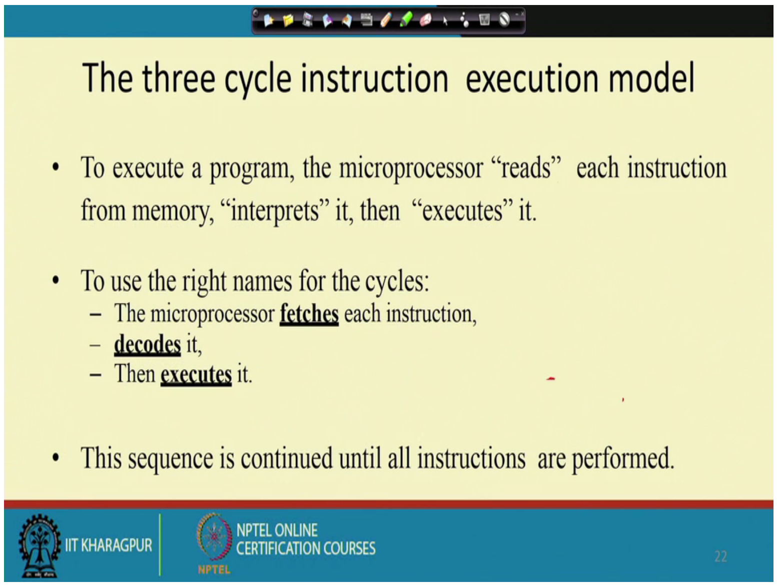
- To execute a program:
  - the user enters its instructions in binary format into the memory.
  - The microprocessor then reads these instructions and whatever data is needed from memory, executes the instructions and places the results either in memory or produces it on an output device.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES 21

So, to execute a program, the user will enter instructions in binary format into the memory. Apparently, it seems a bit awkward like how the user will enter the program in a binary format we never do that. So, we write our program in some high level language and those high level language programs. So, they are converted into this binary level which is known as a machine level by using some compilers, but ultimately as far as the some execution of the program is concerned, we have to load the program in binary format into the memory. So, that has to be done and the typical system, that is done by means of the operating system tools like loader and all that. That is a different part different course in which you can learn about it, the system software utilities.

So, once the program has been loaded into the memory then the microprocessor will be able to read these instructions and whatever data is needed from the memory execute those instructions and place the results in memory or it can produce the result on an output device. After executing the instruction, the result that is produced is again another variable value. So, that value may be stored in some register. It may be stored in the memory or there are some input and output instructions by which we can read the values from some input device or we can write the values to some output device. So, in and out, these instructions are there which can be utilized for that.

(Refer Slide Time: 21:59)



The three cycle instruction execution model

- To execute a program, the microprocessor "reads" each instruction from memory, "interprets" it, then "executes" it.
- To use the right names for the cycles:
  - The microprocessor **fetches** each instruction,
  - **decodes** it,
  - Then **executes** it.
- This sequence is continued until all instructions are performed.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

22

So, there are 3 instruction execution models, 3 cycles in which an instruction is executed; they are known as fetch, decode and execute. How an instruction will be executed? First, the instruction will be read from the memory. So, the microprocessor first reads the instruction, then it interprets it and finally, it executes it. So, how to use this? We should give some proper name to these 3 operations. When it is reading the instruction, this is known as the fetching of instruction.

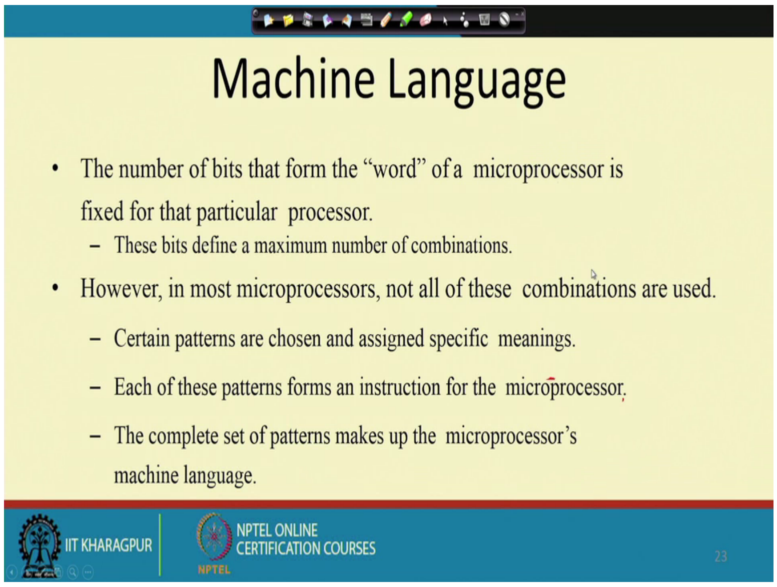
So, as if the processor is fetching the instruction from the memory. Once the instruction has been fetched then the instruction has to be understood like what the instruction means. So, that is called the decode stage and after the decode stage has been done then the processor knows like what is the operation to do and in that case we have seen in our architecture lectures that there was a decoder and the decoder generates a sequence of control signals to be execute to be to be activated, so that, the operation is done by the functional modules that we have inside the processor. So, that way I can execute those instructions whether that the instruction gets executed by means of that activation of control signals in sequence. That is the execute phase.

So, we have got fetch phase, we have got decode phase, we have got execute phase and once. So, after this work like I have fetched one instruction decoded it, executed it, then what? Then again the same thing fetch the next instruction, decode it, execute it, then again the processor will phase the next instruction decode it execute it till it comes to a statement an instruction which asks the processor to halt.

So, normally all the processors they have got 1 halt instruction when this halt instruction is executed then the processor will stop doing these fetch decode execute cycles anymore and it will wait in that state until and unless an interrupt comes from the outside world. So, that the processor is put back to it is original mode of execution. So, that is the 3 cycle instruction model of any processor. So, this is very commonly known as fetch, decode, and execute cycle.

So, even advanced processors, they will follow this particular model. So, maybe in advanced processors, this execute stage is very complex. That way it is divided into number of sub stages, this decode stage is complex it is divided a number of sub stages and fetch part it is doing some sort of overlapping, in the sense, that when you are executing the current instruction at that time, you can fetch the next instruction. So, those ways there can be an overlap between these fetch, decode and execute cycles over the instructions. So, this is this is known as instruction pipelining through the processor.

(Refer Slide Time: 25:15)



Machine Language

- The number of bits that form the "word" of a microprocessor is fixed for that particular processor.
  - These bits define a maximum number of combinations.
- However, in most microprocessors, not all of these combinations are used.
  - Certain patterns are chosen and assigned specific meanings.
  - Each of these patterns forms an instruction for the microprocessor.
  - The complete set of patterns makes up the microprocessor's machine language.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

23

So, machine language; this is the language understood by the machine. So, like we understand English, when we are writing a program, we understand the language C. So, when I am writing a program in C that is basically good for another person who knows the language C to understand what I want to compute. But, as I said, that microprocessors or processors, they will never understand this English like language statements. So, they will understand only zeros and ones. So, I should be able to express my program in terms of those zeros and ones. So, this is actually the machine language program.

So, the number of bits that form the word of a microprocessor is fixed for a particular processor. So, that way, if it is fixed, if I say that the word size is say 8 bit, that will tell me what is the maximum number of combinations that I can have. If the word size is say 8 bit then the total alternatives that I have is 2 power 8 equal to 256. The processor can use at most 256 different meaning of the word that is faced from the memory. So, I can say that I can have 256 different instructions for this processor.

If a processor has got 16 bit word then I have got more flexibility in the instruction set design where there can be 2 power 16 or 64 k different types of instructions that are possible. But, that is a huge number and in most of the cases microprocessors will not use all these combinations, so, processor designs even for an 8 bit word. So, it may not be using all the 256 combinations. So, each of these patterns will be an instruction for the microprocessor and the complete set of patterns will make up the microprocessors machine language.

So, this is that the complete set of patterns that are possible. So, if you out of say 256 if a processor support a 100 different instructions, that information that there are 100 different instruction and which combinations mean what. For example, if there may be an addition operation this addition operation is given a code. So, that what is the code? So, that a code means and one particular 8 bit pattern. What is that pattern? So, that way if you note down all these 100 different patterns. So, they are actually the machine language for the microprocessor.