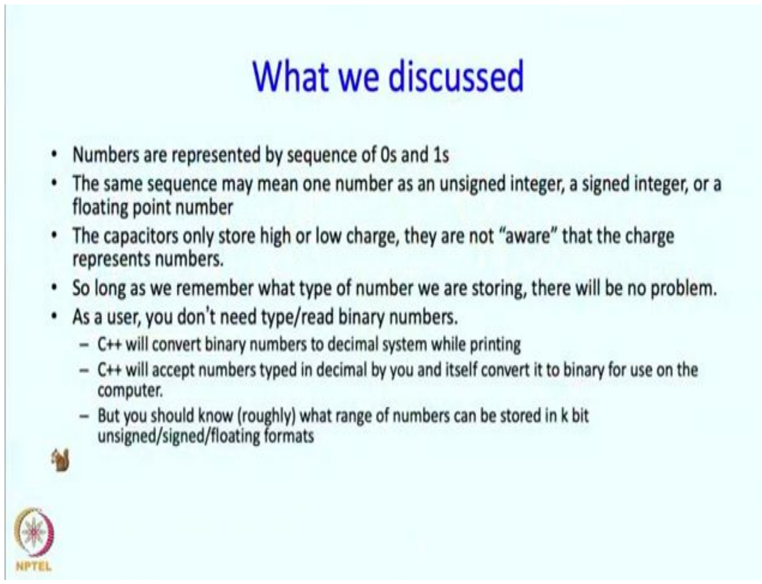
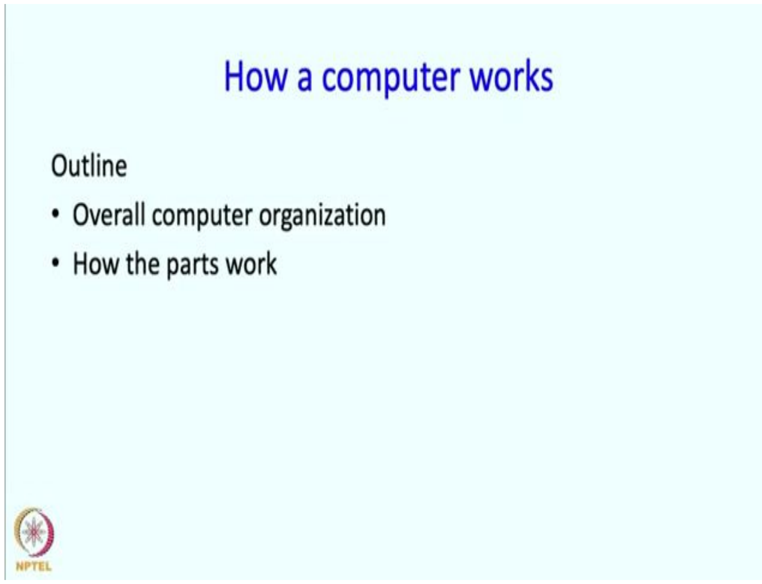
An Introduction to Programming through C++
Professor Abhiram G. Ranade
Department of Computer Science and Engineering
Indian Institute of Technology Bombay
Lecture No. 2 Part - 4
Problem Solving using Computer
Representing numbers on a computer

(Refer Slide Time: 0:36)



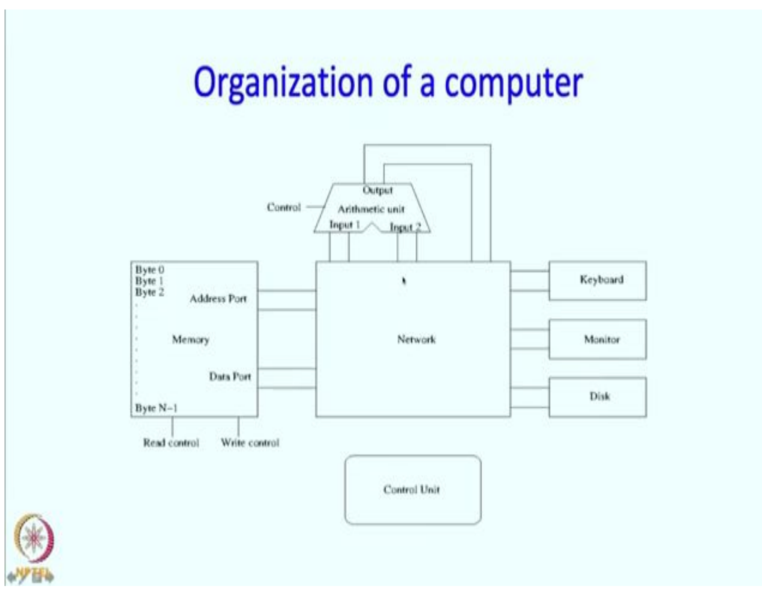
In the last segment we discussed how numbers are represented using sequences of 0's and 1's. So we discussed how integers are represented, with or without signs and how real numbers are represented. And we also said that you do not need to really worry about binary representation when you write a program because you are only expected to write something in decimal and the computer will convert it to binary for its internal processing. And likewise, when the computer prints or something, although it might be holding things in binary, it will print that out in the decimal, so that we can understand it very easily.

(Refer Slide Time: 1:13)



Now, I want to give you at a very high level how a computer works. So I am going to start by telling you the overall organization of a computer. And then I will talk about how the parts work.

(Refer Slide Time: 1:21)

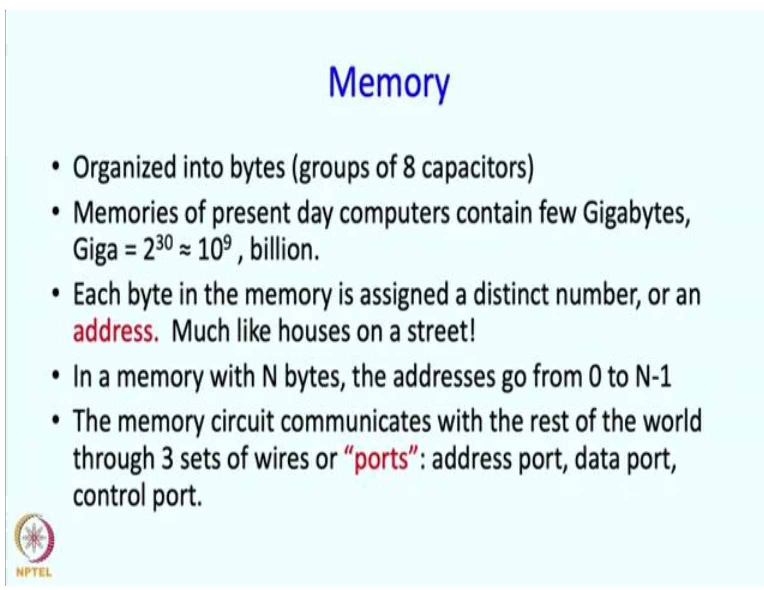


So here is what a computer looks in a very simplified form. So there are some important components. So, one important part is this memory. It has capacitors to store different bites, and it has something called an address port or a data port. And it has also controlled ports. So there can be a read control port in the write control port or there can be just a single control port which tells memory whether writing has to be done or reading has to be done.

Then there is a network, which connects these different components together. So perhaps the most interesting part of the computer is this so called arithmetic unit. So this receives, this can receive inputs and it has some control ports and based on what value is sent on the control ports, these numbers will get added, multiplied, subtracted or something like that and they will appear on the output port. And they can go back into the network and they can be sent back to the various components.

Then there can be a keyboard connected through the network and the values from the keyboard can go into the rest of the computer. There can be a monitored or a screen, there can also be a disc. And finally, perhaps the most complex part of this is the so called of control unit, which really controls all of these things.

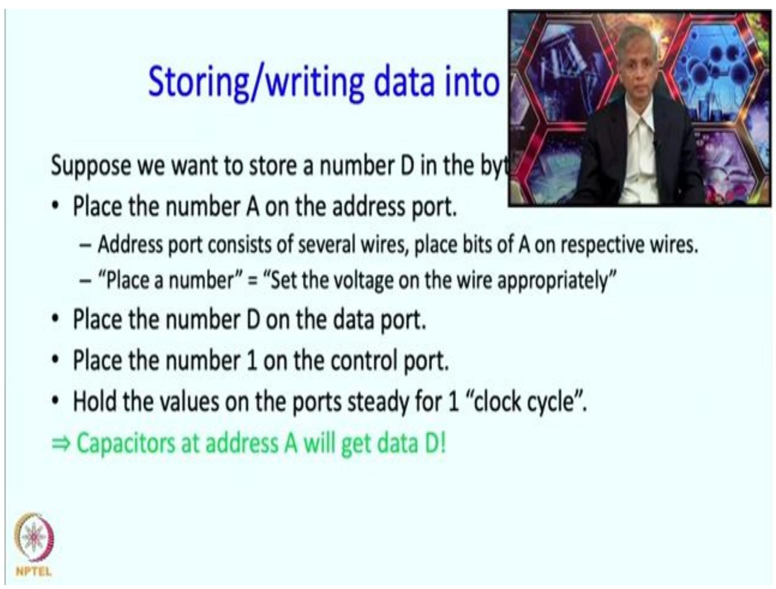
(Refer Slide Time: 2:51)



So then we talked a little bit about the memory. So typically memory is organized into bytes. So, groups of 8 capacitors. Okay. So modern memories will contain few gigabytes where, a Giga represents 2 raise to 30 or about 10 raise to 9 or about a billion. So a gigabyte is a billion bytes. Each byte in the memory is assigned a distinct number or a distinct address. So these are very much like house numbers on a street. So think off the capacitors as sitting on a long road andeach group of 8 capacitors has a number.

So in a memory with N bytes, the addresses go from 0 to N minus 1. And I can say that look now do something to house number 57 or rather I should say byte number 57. And those capacitors may get something written into them, or you may say look tell me what is in those capacitors? The memory, circuit communicates with the rest of the world using 3 sets of wires or ports. So these are the address port, data port and the control port. The control port could be a single wire or it could be multiple wires. So in the picture I had shown two wires the read port and the write port, but we can also have a single wire. So here I am going to talk about what happens if it is a single wire. These details are not important and we are just using those details just for making this picture a little bit more detailed and therefore more real.

(Refer Slide Time: 4:50)

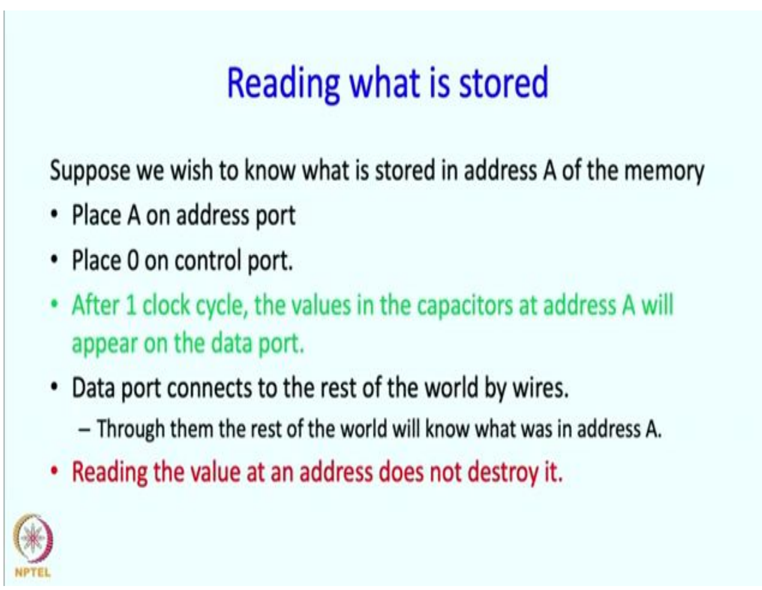


So how do you store or write data into memory? So, suppose I want to store a number D and D could be whatever, say 37 and I want to store it in the byte which has address A, so whatever 57, so in the 57th house or the 57th byte on this long road, I want to store the number 37 something like that. How do I do that? Well, on the address port, I am going to store the number A. So the address port consists of several wires, and basically when I say put the number A on the address port, I mean think of A as a binary number and place the bits of A on the respective wires. So that is what I am going to be in general, so if I say place a number, then I mean think of it as a binary number and place the corresponding voltages the corresponding wires.

In a similar manner I can place the number D on the data port. So you had that memory and it had those two ports and I have set the voltages on the corresponding wires at this point. Then I have this single wire control port suppose, and maybe I place the number 1 on it or if I had a two wire control port, I will place the number 1 on the right control port, it does not really matter, this is a minor detail. So now all these values I am going to hold steady for a period which is called a clock cycle. This is the time in which the circuits to their work. So at the end of that period, the capacitor at address A will get that data D. So something will happen because of which the circuits will work and the data will move. Those voltages will move from the data port to the capacitors themselves and not to any capacitors because, but the capacitors at address A. So that is what writing data into memory means.

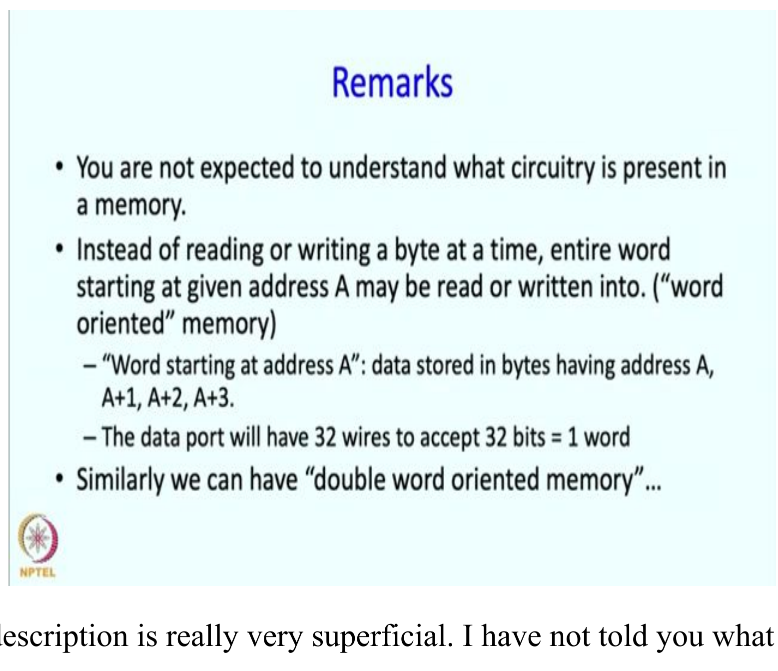
Now you may ask what is this clock cycle business? Well, that is something that is determined by the memory designer. So the memory designer will look at that circuit that he or she has designed and we will say that look this circuit requires 10 nanoseconds for it to do its work. And so the clock cycle will be 10 nanoseconds or something like whatever it is.

(Refer Slide Time: 7:35)



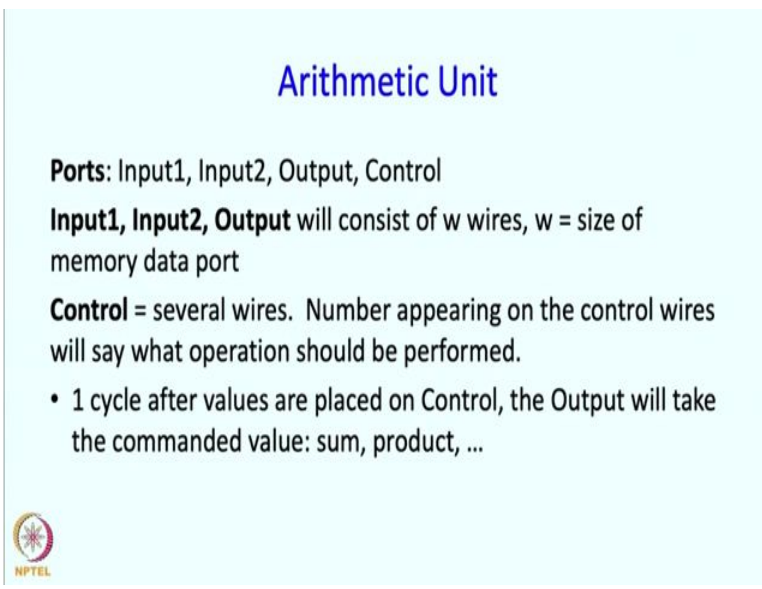
Now you do not just want to store data, but you also want to read what is stored. So suppose I want to know where is stored in accuracy of the memory. Again, what does that mean? So A could be 57 and if you think back about our picture where capacitors are sitting on this long road, groups of capacitors are sitting on this long road, I want to look at the 57th group. The A-eth group and figured out what is in those capacitors. How do I do that? So I am going to place A on the address port. Again, that means converting A to binary and storing the corresponding values on the wires in the address port. Place 0 on the control port, or if we had a two wire control port, we will place a one on the read control port, so again equivalent. And we will wait for one clock cycle, so again, the circuits will do their work and somehow they will copy the values in the capacitors at address A on to the wires in the data port. Now this copying is not destructive. That is the capacitors will still continue to hold those values. But, now those values will also appear on their data port. Now the data port connects to the rest of the word by wires and from there you can move those voltages or those that value wherever we want. So reading the value at any address does not destroy it. So even after this operation, address A will continue to hold the value that it was holding earlier.

(Refer Slide Time: 9:21)



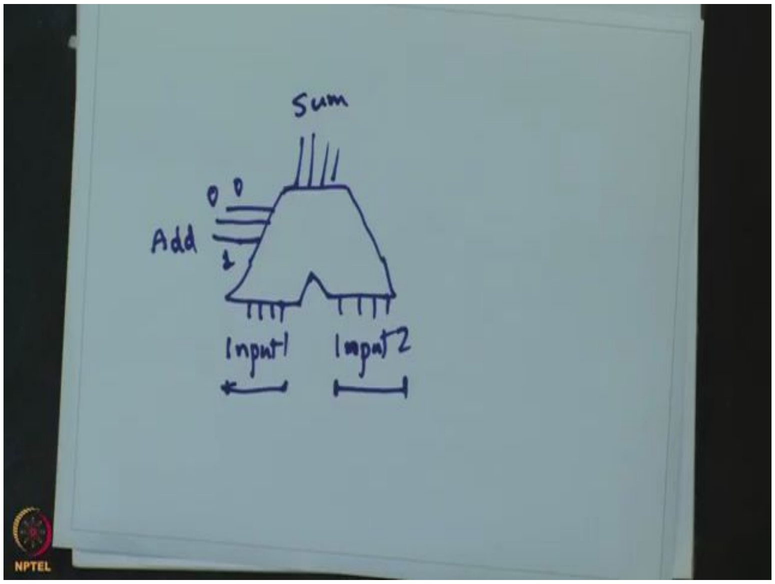
All right, so this description is really very superficial. I have not told you what the circuitry, what circuitry represents. I just said that their exist circuitry, which will do this movement and I have just told you that this box, this black box called memory has these 2 three openings through which it communicates with the rest of the world. And now this organization can be somewhat different. So again, this is all schematic. So I just want to tell you that this organization need not be exactly what I said earlier. So for example, instead of reading one byte at a time, you may read one word, so a word starting at address A maybe read or written into. In such a case, the memory is called the world oriented memory. Well, what does the phrase word starting at address A means? It simply means the data stored in bytes having address A, A plus 1, A plus 2, A plus 3. So remember that the byte is 8 bits, a word is, 32 bits. So 4 bytes make a word and so we have to regard, we have to consider the data stored in 4 bytes and the notion of words starting at address A is simply the 4 consecutive bytes at starting at address A. And typically in such a memory the data port will have 32 wires so that all the bits in that word can be brought out simultaneously. Similarly, we can have a double word oriented memory or maybe a half word oriented memory, whatever you want.

(Refer Slide Time: 11:12)



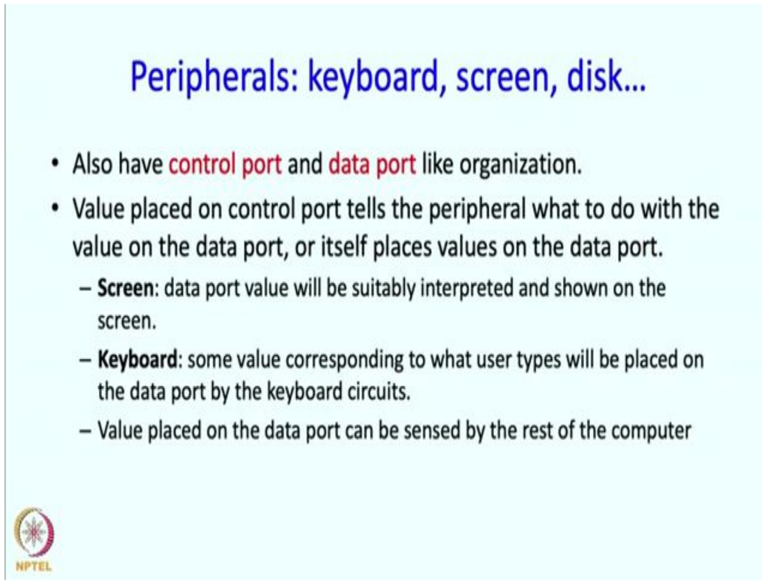
Next I am going to look at the arithmetic unit. So here the ports are input, output. So there may be there typically are 2 inputs and there is an output. And then there are, there is a controlled port. Input and outputs will consist of w wires, where w is typically the size of the memory data port. So if it is a word oriented memory, then this will typically be a word oriented ALU, but not necessary, this is just a typical organization. Control could be several wires. So the numbers appearing on the control wires will say what operation should be performed.

(Refer Slide Time: 12:04)



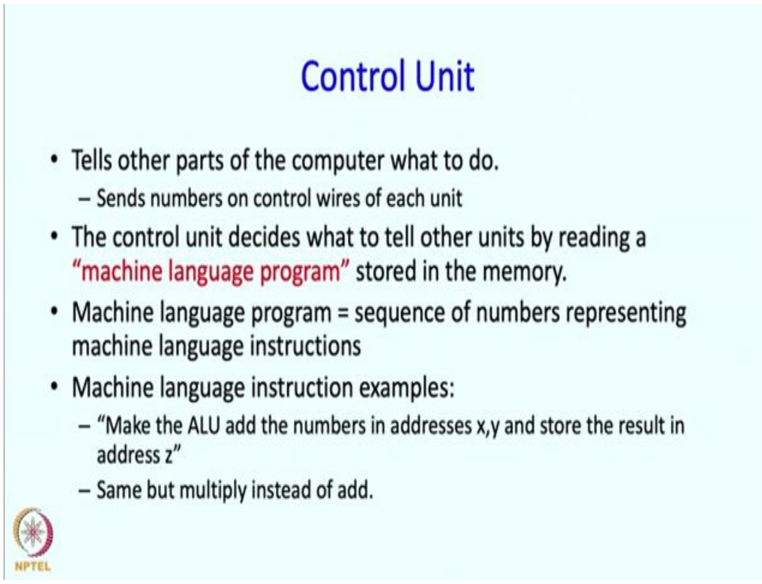
So, what happens over here? So, let me draw a picture. So typically the ALU is pictured in this funny fashion. So this is data port, input port. This is input 2 and this is the output port and there is a control port over here. So, if you want say something to be added up, what do you do? You place the value to be added on these wires, then you place the second value to be added on these wires. And then, there is this control port and it may have several wires. So there might be one wire which is the add wire. So you raise this wire to become one instead of 0 and everything else should of course be zeros that is going to command this arithmetic unit to take these two values and put out there sum over here. So that is how the arithmetic unit works. So arithmetic unit is, I guess sort of the heart of it because this is where the calculations are performed, but in principle it is fairly simple, the organization is fairly simple. So values come in, what is to be done with them should is fed through the control port and outputs go out. Of course, it has to contain circuits for doing multiplication, addition, division, whatever it is that you want to be done.

(Refer Slide Time: 13:54)



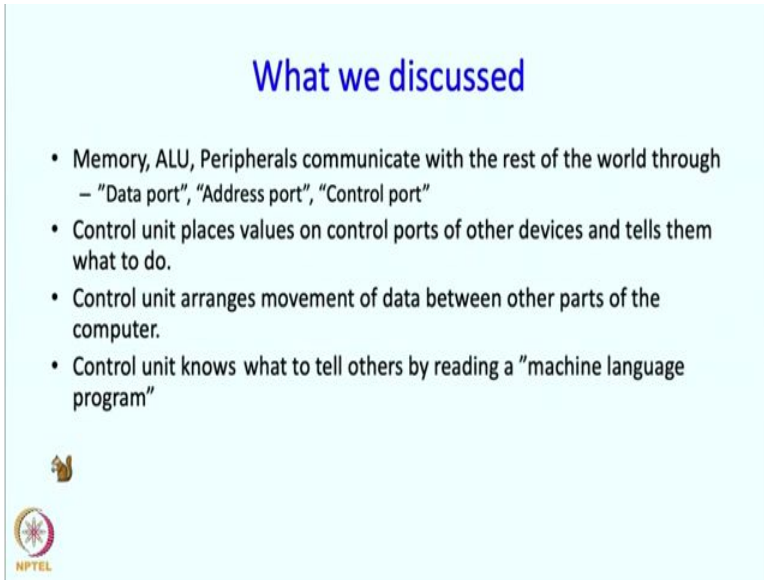
Next, there are the so called peripheral devices and which might include keyboards, screens and disks. So these also have a control port and the data port like organization. So I am going to describe this again in a very simplistic manner. We are going to place values on the control port and those really tell the peripheral to what is to be done with the value in the data port. Or, it may say, look this value that I am placing on the control port really indicates that you should place a certain value on the data port. So let me give an example. So if you take the screen as a peripheral, then the data port value is going to be suitably interpreted and shown on the screen. So the control port might say where that data value is going to be shown. That might be how a screen works. Or if it is a keyboard then you may say that the keyboard itself will place some value on the data port and the rest of the world will be able to read it. So your computer may be able to read it. So the control port of the keyboard may signal to the keyboard that now look now and I am waiting for you to send me something. So the value placed on the data port or keyboard can be sensed by the rest of the computer.

(Refer Slide Time: 15:15)



Then the final unit that I want to discuss is the so called control unit. So this control unit is what you might call the manager or the brains of this entire thing. This is the controller as the name says, it is going to tell the other parts what to do. How does it do that? Well, it sends numbers or commands on the control wires of each unit. So we said that the memory unit control wire has to become 1. So it is made 1 by the controller. Or the add control wire of the arithmetic unit has to become 1, who does it? The control unit, does it. The control unit decides what to tell other units by reading a machine language program stored in the memory. So of course the question arises, how does the control unit itself know? So for that there is the machine language program. A machine language program is a sequence of numbers representing machine language instructions and machine language instruction examples might be something like this. So there might be machine language instruction which says make the ALU add the numbers in addresses x and y and store the result in address z or maybe another instruction which says make the ALU. The ALU is an abbreviation for arithmetic and logic unit, which is the abbreviation which is sometimes used, but another abbreviation is also AU and which is just the arithmetic unit. So arithmetic and logic unit or ALU and AU arithmetic unit are abbreviations, which really mean the same thing for practical purposes. So you might have on machine language instruction which commands the AU or the ALU to do multiplication or do division or addition or whatever it is.

(Refer Slide Time: 17:28)



All right, so what have we discussed in this? So we have discussed that a computer has several parts like memory, ALU or arithmetic unit or AU, peripherals and these communicate with the rest of the world through data port, address port, control port, ports like this. Then there is a control unit, which tells other devices what to do by placing values on the control ports of those devices. The control unit arranges for data movement to happen between other parts of the computer, and the control unit knows what to tell others by reading a machine language program which is sitting in the memory of the computer.

So we will conclude this segment at this point.