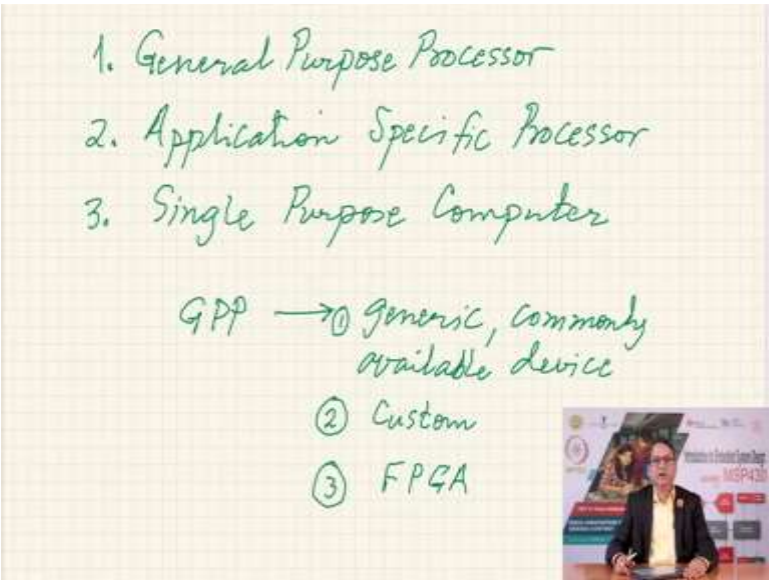
Introduction to Embedded System Design
Professor Dhananjay V. Gadre
Electronics and Communication Engineering
Netaji Subhas University of Technology
Badri Subudhi
Electrical Engineering Department
Indian Institute of Technology, Jammu
Lecture 03
Modular Approach to Embedded System Design

Hello, welcome to the next lecture in our course on introduction to Embedded System Design as you know, I am the Dhananjay Gadre and I will be conducting this session in the last session we discussed the definition of embedded systems the application areas and in how many various ways we can implement the embedded computer.

Before we start this lecture I want to follow-up on some of the issues we discussed in the last lecture namely if you recall in the last lecture I mentioned that there are 3 methods of implementing the embedded computer.

(Refer Slide Time: 01:10)



The first one is using a general purpose processor, the second is using an application specific processor and the third being single purpose computer. I did not discuss that what is the source of these 3 approaches meaning that if you wanted to implement the embedded computer using a general purpose processor where do you get a general purpose processor now it turns out that general purpose processor you can have readymade provided by some manufacturer and this would be called generic commonly available device.

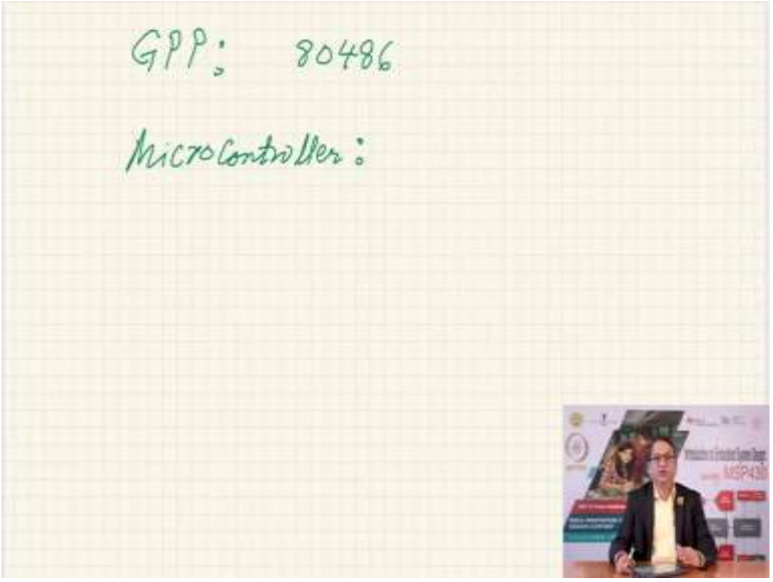
For example, you could take 8086 or you could take 68000 these are examples of general purpose processors this is one method. The second method could be that you could design your own you could design a processor of your choice using contemporary techniques and this would be a custom approach.

And the third is this existing companies could license you the code, the hardware description code for their processor and you could acquire those codes in whatever way and target that code into a programmable logic device such as FPGA these are the 3 methods.

If we want to implement our embedded computer using application specific processor then again, we can choose the first method that is select available microcontroller or a DSP or create our own custom application custom chip and the third you could take the hardware description code for that particular micro controller or DSP and target a programmable logic device.

Single purpose computer on the other hand because, there is no special chip available to do that there are usually two methods: one is that you design your own single purpose computer and then get it fabricated in a custom way or once you are designed it you could target the actual implementation in a field programmable gate array device. One more point I would like to mention here.

(Refer Slide Time: 04:35)

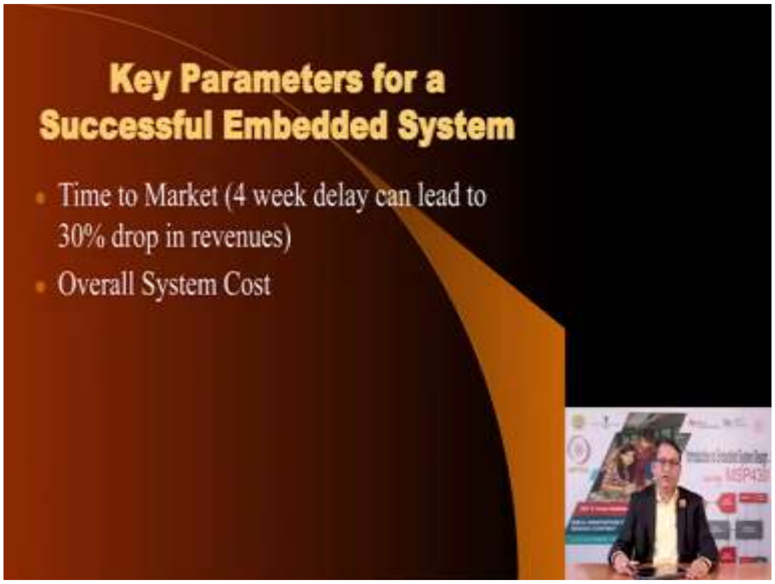


If you look at the general purpose processor scenario if I take an example of say 80486 as an example, there is usually not many varieties of 80486 microprocessor available there may be few based on maybe the speed performance of that microprocessor or maybe the packaging but varieties are limited.

On the other hand if you go to a microcontroller, which is an example of a application specific processor you would find because not one implementation may satisfy all the requirements that you would find that for a given family of microcontrollers usually there are hundreds of physical devices available each device offers various amounts of memories and ports and other peripheral features.

And once you are looking for your requirements, you could survey the available devices and select one which best fits your needs and so, that is the fundamental difference between the general purpose processor availability and the availability of microcontroller families. Now, we are ready to resume our current lecture. Now, we want, we have seen the ways we can implement the embedded computer but we want to look at what are the parameters by which we approach design of an embedded system.

(Refer Slide Time: 06:13)



And it turns out as a commercial implementation that there are two methods to matrix which we must keep in mind when we end our design the first one is called time to market meaning that once you decide that you want to create a certain product you must complete the design, task the fabrication task and launch your product quickly into the market.

Various surveys have shown that if there is a 4 week delay in the launch of a product it can lead to loss of market up to 30 percent drop in market you could experience. And this is very similar to an age old wisdom that they say early bird catches the worm so, time to market is an important consideration when you are designing an embedded application.

And the second consideration is that whatever application you are designing it must be cost effective, the overall system cost will dictate your success even if you can launch a product quickly if it is not affordable if people cannot afford it then your product will not be successful. So, both these matrix must be kept in mind both these points must be kept in mind when you as a professional embedded system designer start designing a product.

Now, we will see how our choices that we exercise are affected by these two matrixes. Now because we are doing this course with the assumption that using a microcontroller is a very popular method of implementing an embedded system so, we are going to see what are the various ways in which we can select the right microcontroller for our job in fact this topic we are going to revisit often times in subsequent lecture also.

(Refer Slide Time: 08:06)

I am going to discuss this idea of how to choose the microcontroller that is suitable for your application in different perspectives right now we are looking at it from the point of view of time to market and overall system cost. So, time to market being an important consideration when you select a particular microcontroller based on other considerations of whether it is suitable functionally or not question arises is the microcontroller architecture is easy to use?

Meaning the design team that is going to take up this design process are they able to understand the architecture can they understand it and quickly start implementing the design because, if they do not then it would impact the time to market metric. In that objective, one of the important considerations is to find out the microcontroller that you are choosing for your application does it support high level languages?

Now, in this age it may seem like a very odd question but, remember about 40 years ago when microcontroller started appearing the amount of memory available on chip for these microcontrollers was very limited and therefore, to make the best use of that microcontroller the preferred method of programming these devices was using the assembly language programs.

And assembly language programs for a same objective if you decide to write your application in assembly language program it would take more time compared to the time it would take if you wrote the same application in a high level language, why? Because the high level language allows you complex constructs with which to manipulate the data and therefore, you can quickly achieve your objective.

And so, if you even today if you are going to select a microcontroller you must find out whether the microcontroller has software tools which support the language of your choice. Usually these days people do not use assembly language program they prefer C or C++ and there are many other languages which are being used for programming microcontrollers in a for embedded applications.

And therefore, you as a designer must find out whether the language of your choice is supported through various software tools for the microcontroller that you want to use if not, it will impact time to market. Now, once you have decided to use a particular microcontroller and you start using it in your application.

What if, a few days into the development process you realize that your original estimates were all wrong and that this device does not have the resources in terms of maybe the memory or maybe in terms of the peripheral devices that it is not possible to convert your product using this particular microcontroller. Then, you must find out whether there are other compatible devices with more memory or different features that you may require.

And you want to retain that family meaning you do not want to find a completely new device. Because, then it would waste your development process and time that you have spent is now do you want to reuse it simply switch over to a new device and therefore, it is very important to know what are the what is the eco system of parts variety of parts that are available in a given family again this will be useful to meet the time to market objector.

Now, to develop an application using a microcontroller, it requires several development tools again we will see a list of such tools subsequently but, quick list of such devices to be able to program a microcontroller you need a compiler the compiler usually comes with an assembler. Once you have compiled your code you would like to test the validity of that code you would like to see whether the code works correctly or not without committing it into the microcontroller.

And so, you would like to have what is called as a debugger or simulator using which you will see the correctness of logic of your program. And after you have verified that you may want to test the code in real hardware, which is where you would require a evaluation kit and as you as I mentioned in the earlier lecture that as part of this course, you are going to receive MSP430 microcontroller evaluation kit, which will allow you to test programs in on the hardware for this microcontroller.

And similarly, when you are developing an application in a professional way you do not want to commit to making a printed circuit board right away you want to get a feel of the development process you want to see how your code is running on the hardware platform and therefore, having a readymade computer made by somebody is very useful this is where the role of evaluation kit comes into picture.

And apart from that you also need something called an emulator. And emulator is a mechanism to emulate the functionality emulate the behavior of that microcontroller in hardware and these days it is very easy to have that functionality in the olden times you would actually get a specialized device called in-circuit emulator it was an expensive piece of equipment.

But today almost all modern microcontrollers are equipped with the on-chip emulator on chip debugger, which allows you to emulate the function and verify whether, the program is working or not and if it is not working, what is the reason. So you need all these development tools. And in case things do not work out are there any support forums which can address your concerns or your difficulties or your problems.

Having a support forum maybe by the manufacturer or by the online community can be useful resource and therefore, given two devices which are suitable for your job one which has a better support which has better popularity probably should be chosen so that it helps you resolve different difficulties that you might face. Let us see more points.

(Refer Slide Time: 14:37)

Now, the earlier slide was a talking about time to market in this slide we are going to talk about the overall system cost. Now the overall system cost of your product is determined by several factor one of which is the cost of the microcontroller but, a microcontroller alone is insufficient to create an application usually it would require external components. These external components may be in the form of analog operational amplifiers or filters and so on and so, forth and therefore, you must consider the overall cost of not just the microcontroller but, also the external components.

Now you see in this slide if I say if one were to choose the piece of microcontroller which has programmable analog components on chip it would reduce the cost and would do further that it would also reduce the cost of the PCB.

So, the cost of the PCB is an important consideration in the overall system cost having less number of components because, you choose a microcontroller which has higher integration of the requirements that you need would lead to smaller chip count it would also increase the reliability of your product and it will lead to a reduced cost of printed circuit board thereby giving you a more economical product therefore, it can be more competitive.

Now, having designed product using a particular microcontroller often times it is seen that there is a demand in the market of a higher upgrade to that product. Now, if you are in that happy situation you would like to know is it possible to reuse the resources that you have accumulated for the development of that earlier version, could you use those resources and find microcontroller which has higher capabilities so as to fit more functionality in the new upgraded product.

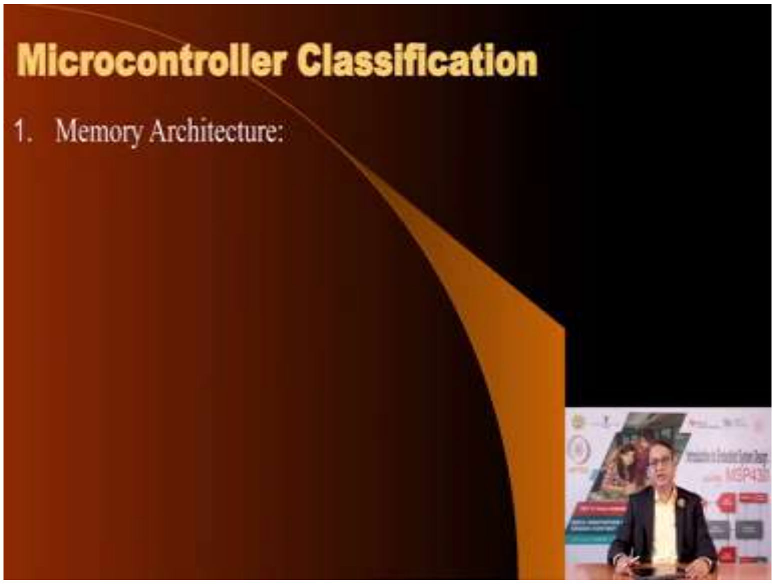
Having this knowledge or having the satisfaction of family, which has upgrade paths available would be beneficial in the long run. And usually the performance is improved because you can fit in more software features into the device. And then what you must also consider is the hidden cost of the development or the production what I mean by hidden cost? For example, suppose you are making your company is making 5 products and if these 5 products use 5 different integrated circuits, if it uses 5 different microcontrollers.

Then, the impact is that for maintaining a certain supply of your products you will have to create you will have to maintain an equal inventory of all those large inventory of all those 5 microcontroller chips that you are using for 5 products. On the other hand if those 5 products use common microcontroller the amount of inventory that you have to maintain to be able to offer you all your products get significantly reduced.

And therefore, having a smaller inventory leads to a better cost that you can offer to the customers and therefore, your product will be more competitive besides that you cannot be expected to stock infinite amount of, infinite quantities of these devices and therefore you must rely upon the manufacturers to supply those to you.

You must also consider how quickly can you acquire a new stock in case you need one. Is there waiting time? Is there turnaround time, does the manufacturer take too much time to fulfill your requirements? You must consider all these aspects because, eventually these are going to impact either time to market or the overall system cost. Now, since we have seen the development of the choices we make for the microcontrollers let us look at the classification of microcontrollers so, that we understand the overall ecosystem.

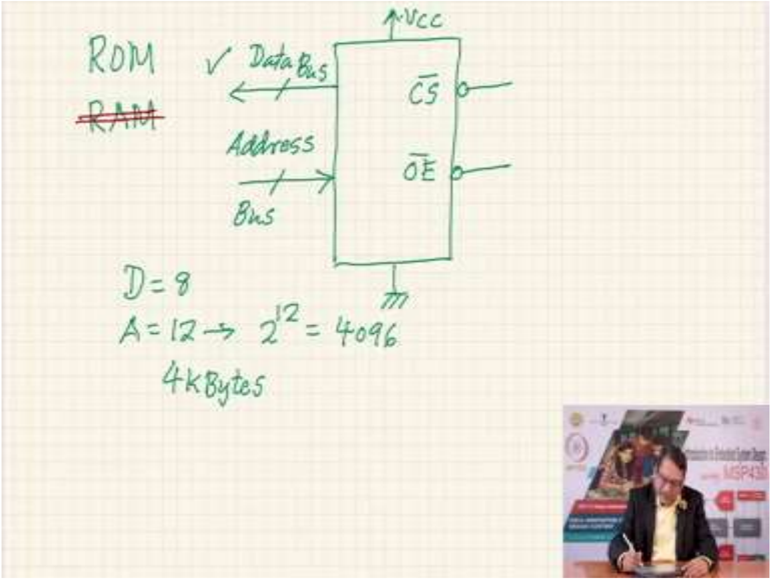
(Refer Slide Time: 18:50)



Microcontrollers and also microprocessors for that matter are classified at various levels. One of the methods of classification is the memory architecture. Memory architecture refers to the way the microprocessor or the microcontroller addresses the memory for its use and memory as we have discussed is required for two purposes. One is to store and execute the program and second is to store data which may be variables maybe stack and so on.

So, before we discuss the memory architecture, it may be impertinent to discuss what sort of memory devices exist and what are the interface is these memory devices offer, although for a microcontroller these memories would be integrated on the chip but, it helps to understand the way these memory devices work.

(Refer Slide Time: 19:51)



So, broadly we need two types of memories: one is the ROM and the other is RAM. A ROM device would have ofcourse all electronic devices need a power supply pin that goes without saying so it requires a VCC and the ground and right now, we are considering the ROM so I am going to cross this. The ROM would have data lines and because, you are going to read the contents of the ROM you do not have the freedom to write into a ROM at runtime meaning in your application.

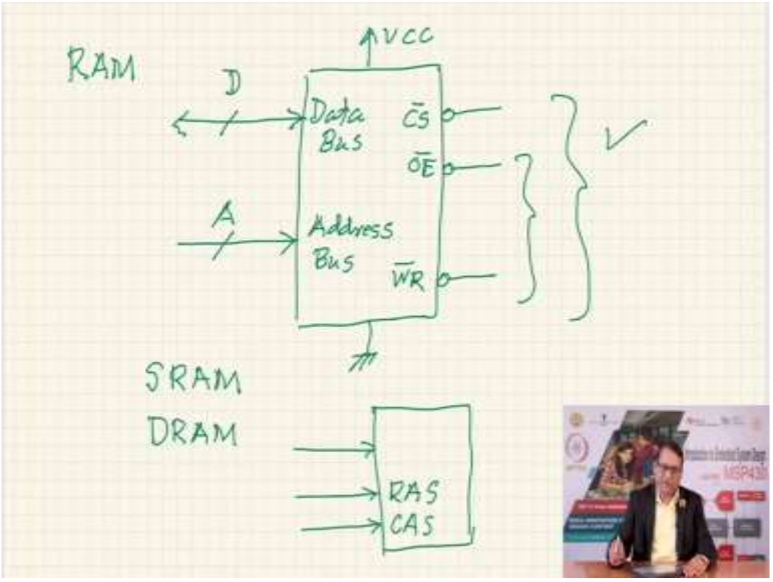
So, I have shown the direction of the data bus to be unidirectional these are data bus data bits then, to access the location you need to specify what address do you want to read the data from and for that you have the address bus. This is the data bus the number of bits on the address bus would indicate what is the capacity of this ROM device meaning how many locations there are the number of bits on the data bus will indicate that for each location how many bits are simultaneously being stored.

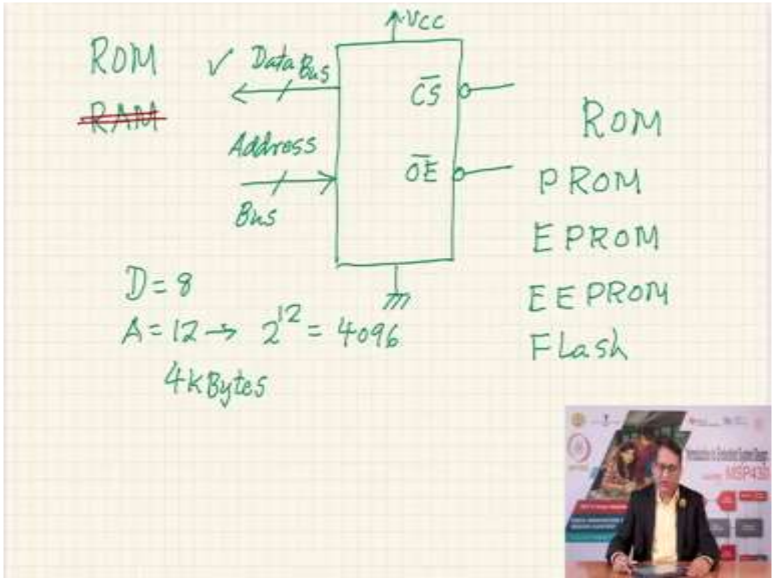
As an example, I could have a data bus D equal to 8 and A is equal to say 12 if that is the case that means this ROM device A 12 leads to 2 raise to power 12 locations, which is 4096 and therefore, this would be a 4KByte 8 bits mean a byte so this is a 4 Kilo bytes capacity ROM device. Apart from the data bus and the address bus it also requires a few control signals and these control signals are referred to as chip select.

And you see I have drawn a bar over that signal meaning chip select bar and to also indicate that the this is active low signal there is a bubble which means to select this chip out of perhaps several other chips that you would have connected on the microprocessor buses to enable this chip this pin must be held low and if you do not hold it low then, this chip will not interact with you will not provide you the data that you expect.

Apart from that it also has a pin called OE bar that stands for output enable bar. So, not only do you have to hold the chip select low you also need to hold OE bar low and then for the value of the bits provided on the address bus it will enable that location in the wrong device and the contents of that location would be available on the data bus this is how a ROM device works.

(Refer Slide Time: 23:16)





Similarly, the way the RAM works is for a RAM device we have almost identical configuration, except a couple of one more extra line so we have the VCC here, we have the ground. The data bus I am going to write it here now, because it is a RAM meaning you can read as well as write data. The data bus is by directional and again it may have certain number of bits. Let us call them let us call them D, number of bits.

The address is unidirectional address bus and this may have A number of address lines and it also requires control signals again a chip select and output enable signal this is till now, it is identical to a ROM device. But, now because you want to be able to write data into this so, you have an additional signal which I will call write bar and this is also active low signal.

The reason why control signals are often active low have to do with the fact that there are legacy issues that in 60s and 70s where the technology for logic devices was TTL, TTL preferred active low signals control signals and therefore even today many of the control signals on modern chips are also of active low variety.

These two signals exercise great caution must be exercised in issuing these signals that is you cannot turn on the output enable and the write bar signal at the same time because, if you did that the ram chip is going to be confused do you want to write data into it or do you want to read data from it and it will actually lead to a condition called bus contention.

Because, you might provide external data and since you have enabled the output enable signal the RAM would provide internal data and there might be a conflict of logic levels. And so, these two never should never be enabled at the same time but, the chip select has to be enabled together with either the output enable signal or the write bar signal. And where do these signals come from? These signals come from the control bus of the microprocessor or the microcontroller to which they are connected.

Now, let me go to the earlier slide here, we just functionally meant that we need a ROM device there are many varieties of ROM the first one is itself is ROM meaning you get a manufacturer to provide this chip, which is pre-stored at the factory level with the contents that you want then there is a type of device called P-ROM meaning programmable read only memory you get a blank chip and using a special programmer you load your required contents into this device.

But, thereafter you cannot use it if you want to reuse then there are a couple of varieties. One of the older ones was called E-PROM that is Erasable Programmable Read only Memory. This device could be raised with the help of ultraviolet light shine which was put onto the chip using a special window at the top of the chip and then you could erase it using this light and you could program it using a special programmer electrically.

And then, the fourth device you have these days is called EE-PROM where the erasing of the device can also be done electrically. So, this is electrically erasable programmable read only memory and a variety of E square PROM as it is popularly called is the type called flash memory, flash memory is type of E square PROM except it is faster to erase. And therefore, that is why the name flash you can perform that operation in a flash these are all examples of read only memory devices.

When we look at the RAM devices primarily you have two types of devices called SRAM and DRAM. SRAM stands for Static Random Access Memory and DRAM stands for Dynamic Ram. Static RAM is easier to use because, you simply provide the address you provide the control signals and data would come out or you could write data into it. DRAM on the other hand is more efficient because, it can it packs more data bits into the same area.

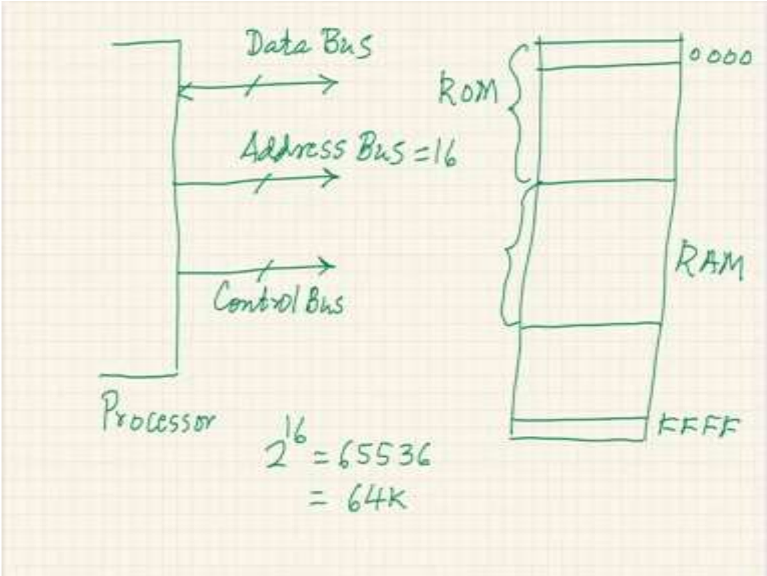
And but, it is little complicated to use it as it requires of frequent refreshing it also has a different method of providing the address, the number of address lines are less and to indicate that you have two signals called RAS and CAS. So, it is like the memories arranged in the form of a array and first you supply the row address then you supply the column address using the same address lines.

And so, the number of address lines address bits required to access this memory is reduced and you also need to frequently refresh the contents otherwise, it would lose the contents and therefore, useless as a ram and so, using it is slightly more complex but, because it is more efficient it is more dense modern applications often require often use DRAM chips. If the requirement is small then you might go with SRAM. And this functionally from a block diagrammatic point of view that here, is a device where you store read and write data this model is okay.

Now, let us go back to the actual presentation and see where this fits now, we are talking of memory architecture meaning how does the microprocessor connect to the memory devices and we certainly need two types of memory devices a ROM and a RAM the first memory architecture that we would like to discuss here is called Von-Neumann.

Von-Neumann refers to the way the microprocessor talks to the available memory and as I mentioned for any microprocessor for any microcontroller or any computer in general would require two types of memory ROM for the program and RAM for the data so, let us see how does a microprocessor communicate with memory under the Von-Neumann architecture.

(Refer Slide Time: 30:15)



So, if this is your processor and I am going to use the word processor little loosely it could be a microprocessor or it could be a microcontroller does not matter in a Von-Neumann architecture you would have a single data bus this is data bus you would have an address bus and outgoing pins for address and you would have suitable control signals to generate the chip select and output enable and write bar signals and so on. So, let us call this control bus now, depending upon the number of address bits address lines.

Let us say 16 address lines that means this processor can uniquely address 2 raised to power 16 locations and this is equal to 65536 locations and is commonly referred to as 64 kilobytes of memory, 64 kilo locations. Now, each location how many bits can it store will depend on the width of the data bus.

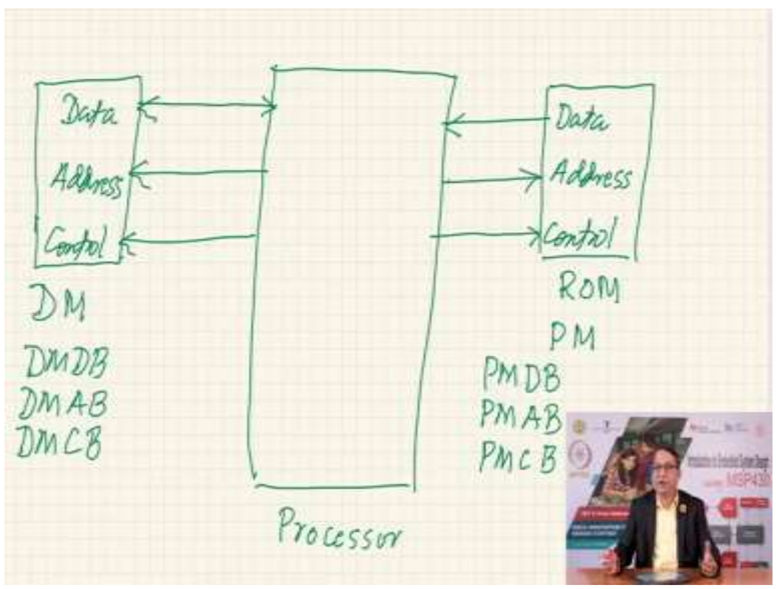
If it is 8 that means this processor can talk to 64 kilobytes of memory location so, let me show you that memory map. So it is going to start from first location and we will call this 0000 four nibbles to specify the address each nibbles is 4 bit so, 4 digits means there are 16 address lines and the last location will be f f f f that is all ones and therefore, this as 64000, 64k locations.

The microprocessor designer the system designer using this microprocessor must put suitable amount of ROM into this memory map and let us say the first few locations it decides to dedicate to ROM and then maybe from here, to certain a lower value it has RAM. so it is going to use a unified bus to read the program from the ROM and maybe read and write variables and store variables in the RAM.

What is the problem with this? What is the characteristics of this architecture? It is very simple. The number of pins required by the microprocessor to connect to memory devices is the lowest that you can find but, there is a little problem in this approach in terms of the number of activities that it can do. Because, there is only one common bus you can either read the memory contents or you can write to it. You cannot do both the things.

And when that need was felt that you wanted to improve the performance of your processors by including certain parallelism that is you would like to read the instruction while at the same time you may want to store the results from a previous instruction it called for certain modification of the memory architecture and so, a new architecture was devised and that was that is referred to today as Harvard architecture. In Harvard architecture what happens?

(Refer Slide Time: 33:50)



Now, instead of us unified memory bus we have two memory maps so, this is my processor I am, I put it in the center for a particular reason now, you can have type of memory devices with store the program we call them ROM and for that it would have data bus in this case the direction can only be this address lines, so this is data this is a dress and this is control signals.

Now, I call this the ROM or here, I am going to store program so, we commonly refer to this as program memory and therefore these buses which connect to a program memory are called PMDB that is Program Memory Data Bus, PMAB Program Memory Address Bus and PMCB Program Memory Control Bus you want parallelism in the activities that this processor does so, you have one more set of address map which allows you to connect to a data memory.

Now, data memory is could be RAM most often it is RAM but, it could also to be ROM because, you want to read some constants and so on. So, this is I will refer to it as data memory and again here you would require the data bus but, now they are by direction because, you want to be able to write data you have address like the other memory and you also have the control and these are this set of buses is referred to as DMDB, DMAB Data Memory Data Bus, Data Memory Address Bus and data memory control bus, what advantage it offers?

You can do two simultaneous operations, what are the, what is the disadvantage of this approach? You require perhaps you require twice as many pins on the physical microprocessor and so, it will lead to a bigger package. But, that is a trade of you as a designer must evaluate whether performance is more important to you or the size of the chip is more important to you and these two architectures are available.

(Refer Slide Time: 36:30)

This is one method of classifying microprocessors or microcontrollers that is does the microcontroller that you are choosing does it of does it offer Von-Neumann architecture for memory access or Harvard architecture? Then, at a different level of classification it is bit handling capacity that is how many data bits can it handle simultaneously? And what options we have? We have 4 bit data 8 bit data this is the size of the data bus.

And if we are looking at we are we are microcontroller is of the Harvard architecture then, it refers to the data bus size of the data memory not necessarily the program memory and so we are talking of 4 bit data bus 8 bit data bus 16 and so on and so forth. And then we have what is called as the instruction set architecture. I am going to continue with the description of the instruction set architecture in the next session and I will see you shortly. Thank you.