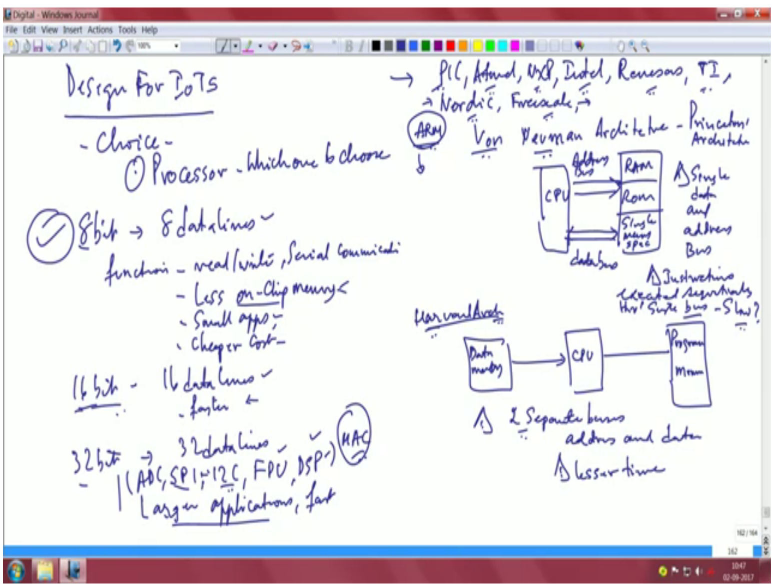
Design for Internet of Things
Prof. T V Prabhakar
Department of Electronic Systems Engineering
Indian Institute of Technology, Bangalore

Lecture - 29
Choice of Microcontrollers

So, now let us look at one important aspect when you talk about a course like design for internet of things, what is the way what is the method that you are going to use to select the processor for your application. You are going to make a piece of hardware, and the heart of the system the brain of the system indeed is the CPU right. So, you have to also have a particular way by which you choose this microcontroller or system on chip for your application. How much of it should be an emphasis on the application, and how much should it be on the merit of the architecture is a very important thing.

There is no simple answer to choice of a microcontroller. I have been trying to understand this in a very big perspective and I will point you to a text book where I found a few things with respect to choice of a microcontroller. But by enlarge it will be based on your own experience and maybe as we go along you will understand when you build systems that ultimately choice is can be made only based on very specific I would say requirement that you have. But nevertheless you should get a overview of how to choose otherwise how will you start anything right.

(Refer Slide Time: 01:44)



So, what I did was I just put down this particular chart for you here. If you look on the left side people talk about 8 bit 16 bit and 32 bit microcontroller for your operation, essentially you are talking about 8 data lines, 16 data lines and 32 data lines right. If you look at 8 bit controllers, they have less on chip memory basically and they are meant for really very specific small applications, cost is also extremely low you can do read write, you can do simple serial communication and so on.

But you cannot do anything more than that, for instance you cannot have an operating system running on top of such small microcontroller that would be very difficult to match. Then you have 16 bit microcontrollers they are definitely faster, they can have more on chip memory, and they can have lot more of both in terms of program memory as well as data memory, and they can do a few more things as well. People have attempted to port operating systems embedded operating systems for 16 bit controllers, but I think this my own impression is 16 bit microcontroller perhaps is not the way to go for your any embedded application.

Well I am only telling you I do not have a I would say very detailed answer for this, if you have you can put it this way if you have a sensor which is doing sensing application, periodically has to communicate this sensed data to another system, you just have a simple monolithic code block which just does a simple timer wakes up acquires data and you know transmits the data. You do not need anything more than an 8 bit microcontroller, but if you want to do anything more than more sophisticated than this, I would say between 16 bit and 32 bit microcontroller choose 32 bit microcontroller. Because you will be able to sort of future proof your design as things keep happening over the years.

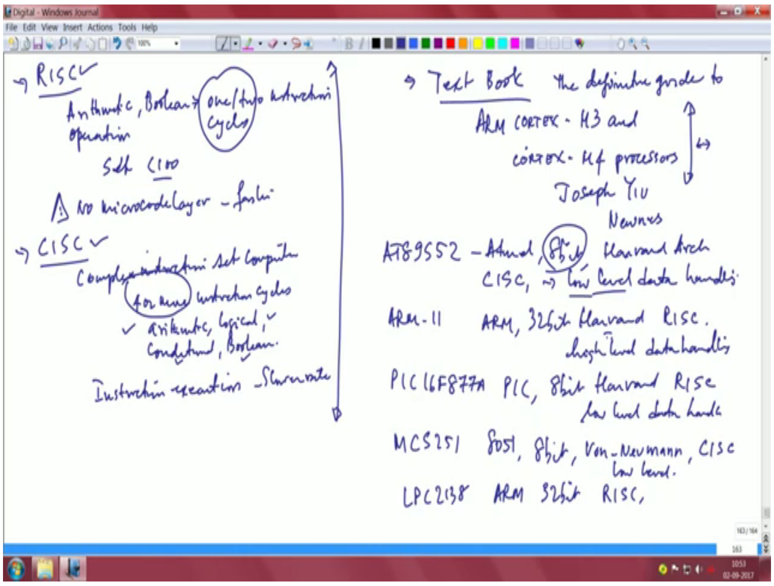
Let us say you are talking about a one decade lifetime consideration, then doing a 16 bit version and then moving to 32 bit is not the right way. Choose a very simple microcontroller if it is straightforward as I mentioned, and anything else you must go in for a 32 bit microcontroller. So, why is that? Well I can point you back to what I wrote here you can have digital buses support for digital buses like SPI multiple of them, you can have I2C and SPI both are digital buses for different sensors which are available. You can have you know 16 bit 24 bit ADC s which can be supported, then you can have floating point units inside DSP s inside them all this vast support for you know any large applications and fast application word I think 32 bit is the right way to go.

Not that you do not have SPI and I2C on 16 bit processors, but you will be very quickly limited. And DSP functionality for instance multiply and accumulate right MAC multiply and accumulate functionality will actually eliminate the need for you. To put a DSP processor if you are talking about any DSP application that you have in mind we will come to that, but let me tell you quickly what it actually means if you have audio application or you have image applications, you need DSP there is no second way of doing without a DSP because lot of application lot of these things were actually used MAC instructions right. These will have to be supported MAC will have to be supported therefore, and that is available only mostly in 32 bit microcontroller therefore, 8 choose 8 for very simple and straightaway go for 32 bit if you are looking at other applications.

And I can only tell you that in this world of microcontrollers I have wrote down here this is just a very small list of controllers that are available PIC, Atmel, NXP, Intel, Renissas TI, Nordic, rescale so many controllers company companies are available each one specializing in their own way. For instance if you take free scale they are very suited for automotive applications, and many of the controllers which essentially many of these controllers basically use arm as the bases. And this arm is basically a risk processor based system and that in turn is used in many variables mobile phones and so on right. So, many of them use arm in it is native way. Nordic semiconductor for instance uses a lot of arm core in it is in it is processor offerings then you have PIC you have Atmel and so on and so forth. Basically I am sure you would have studied all these in several parts of your educational career, but I just want you to I want to point out that two basic architectures are available for processors one is the Von Heuman architecture and the other is the Harward architecture I just put down a very simple picture which I picked up from somewhere on the internet and which is very very well returned part. I just do not I cannot point you to anything but I am sure if you Google you will find a lot of it.

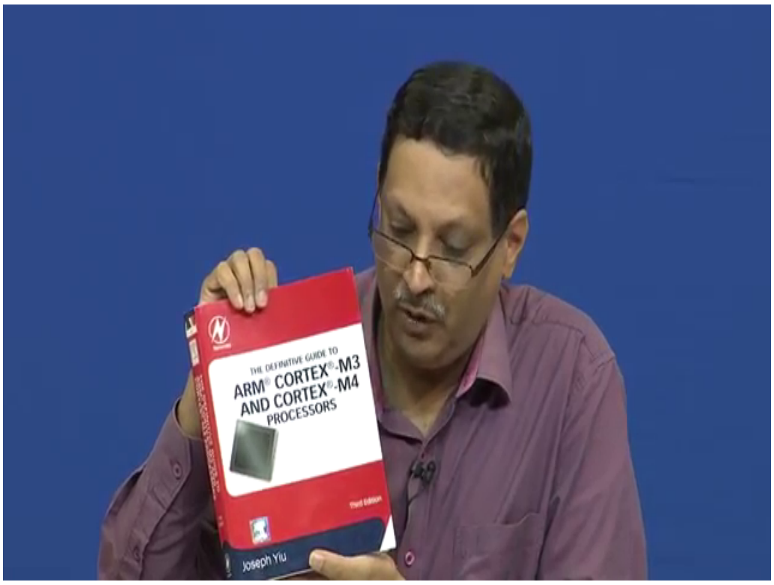
Essentially when you talk about von Heuman architecture also it is called the Prinston architecture basically you are talking about single data and address bus right you have a single address and data bus instructions are executed sequentially through the single bus and therefore, it might be a slow way of doing things because of it is sequential nature and a single bus. Whereas, Harvard architecture essentially you are talking of two separate buses; for address and data and hopefully that might actually give you a better performance and speed and so on. So, really you may have to also look at the processor that you buy whether it is really under the von Heuman architecture or the Harvard architecture so, that is already another thing.

(Refer Slide Time: 08:41)



Then people actually talk about a choice of controllers based on RISC or CISC I already introduced you the thing about risk, essentially you are talking about arithmetic, logical, conditional Boolean and all so many instructions which the processor has to execute. The question is whether it is to be done in one or two instruction cycles or whether does it do in multiple cycles more than one two instruction cycles. Well if you look at risk they can do it in one two instruction cycles and the instruction set itself is typically less than 100 whereas, CISC complex instruction set computer and this is reduced in section said computer, CISC you have 4 or more instruction cycle I am just giving you very general idea and I got very good understanding of the whole aspect of choice of microcontrollers from this very good textbook, which is write here with me at the moment and this book.

(Refer Slide Time: 09:47)

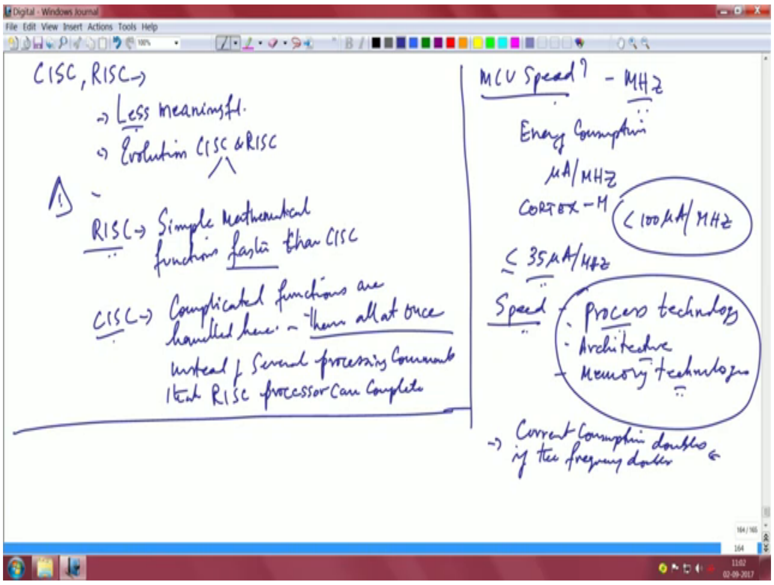


let me point you to this book is arm cortex M 3 and cortex M 2 processors Joseph Yu this is from published by (Refer Time: 09:53). So, this is the publisher. So, and I picked try to understand several things from this book to make a you know to give you a comprehensive idea of the controller. But before we go onto that let us see a summary of controllers that you know very well and what they are basic. This is just to give you an idea of what you might have purchased and what exactly is that whether it is an 8 bit or a 32 bit controller, and what it is corresponding architecture is.

If you look at AT 8 9 e s 52 this is basically from Atmal, this is from Atmal and it is 8 bit it is an 8 bit it follows the Harvard architecture that is already very good, and it is CISC based and it is purely meant for let us say low level data handling. Arm 11 is from arm right this is 32 bit it is also Harvard, but this is not CISK, but it is RISC, r i s c it is also meant for not for low level, but really for high level I would say high level data handling PIC I already mentioned PIC 16F87A this is come from the PIC it is 8 bit very popular microcontrollers Harvard just to give you a feel is also RISC and it is meant for see if you talk about 8 bit it has to be low level. So, it is indeed for low level data handling. I will take one from Intel MCS actually MCS 251, this is Intel 8051 based, this is the family of 8051 very very popular Intel family 8051, 8 bit and this for Neumann just to add a little more flavor on Neumann CISC based also meant for 8 bit.

So, it has to be low level. So, I will simply say low level then LPC 2138 also from arm this is arm based it is 32 bit and it is RISC based systems and therefore, it is meant for high level controller high level data handling application. So, essentially you will have to get into a little detail of understanding even before you choose whether you are in for Harvard or whether you are in for von Neumann and whether you are looking at RISC systems or are you looking at CISC systems, and basically this is an overview of what one should know as a very basic thing before you start. Now let us actually elaborate several things from this book and try to understand in a little more detail on how one can choose processors. So, let us move from there.

(Refer Slide Time: 14:20)



So, one of the things that occurs to occur here all these issues related to CISC RISC and so on in my own view these are less meaningful. I mean vendors will try to tell you many things why there m e a n i n g meaningful the less meaningful, because there is a continued evolution of both. So, I would say evolution of both CISC and RISC is happening. So, this is getting blurred. So, I would say do not go by what I wrote previously I just wrote it for academic purpose, but really if you want to. So, let me just point you here I wrote about RISC and CISC, but I also want to tell you that by the same token that CISC and RISC really are I would say it is becoming less and less meaningful, but it is then why did I write it right.

So, the next question will be why did we actually start the discussion on RISC and CISC at all. I will say RISC essentially is for handling simple math mathematical simple mathematical functions very importantly faster than faster than CISC processors faster than CISC, but CISC can do much more complicated functions faster. It can handle complicated functions are handled well here so, that indeed is the main thing it is not only about complicated functions being handled faster and it can do them all at once instead of instead of several processing commands that RISC systems RISC processor can complete. This is very very important they can do them all at once look how contrasting things are right you say RISC is can do one two instructions, but this is arithmetic Boolean operations it can do less than one two instructions, and this is complex instructions it takes more cycles but if you contrast it to mathematical functions RISC may not be the right choice, you may want to go in for. So, it again now will depend on what your application is.

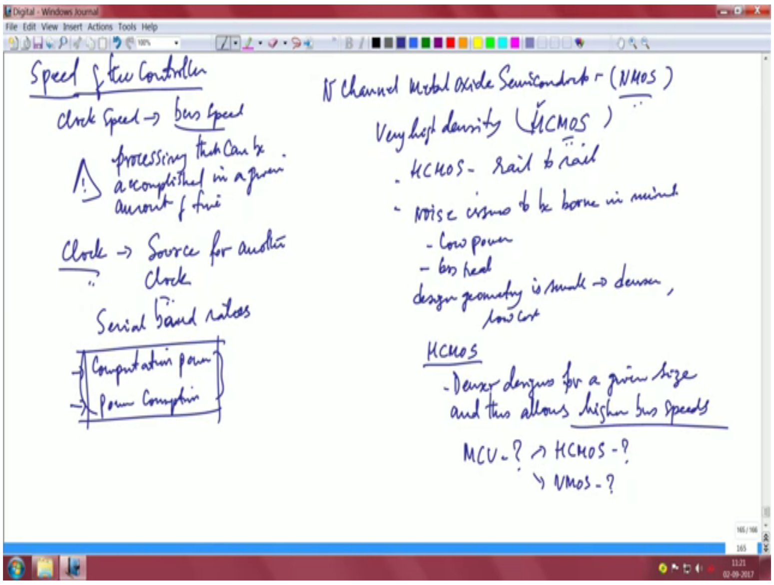
If your application is requires complicated functions even if risk is doing in one two instructions that we wrote here, instruction cycles sorry instruction cycles that is important if it is doing it even in one two instruction cycles, you may not want to choose them you may not want to choose a RISC processor. So, it is not so easy right. So, this is another nice thing about the about the RISC processors. Now there are other things when you actually talk about. So, this is about RISC and CISC direct comparison, then there are other things at about MCU speed. Big question everybody talks about speed and everybody talks about vendors I mean this many megahertz it can run. Well speed is directly got to do with energy consumption right consumption is indeed an issue and that is to do with MCU speed that is why vendors will give you a number so, many micro amps per megahertz, that is there already telling you that. If you look at arm this cortex I will write this in cap. So, that cortex M, there are basic claims that this should be less than 100 micro amps per megahertz. Less than 100 means how much that is where vendors will try to compete, they will try to optimize and you will get offerings of different from different vendors.

Using the same arm code for instance there are numbers which even tell you 35 micro amps per megahertz, people actually give these numbers some people say less than 30, less than 35. So, people say ah or less than or equal to or equal to 35 micro amps. So, basically less than 100 micro amps per megahertz is general each vendor will optimize and do a few things inside his chip and there are you will see datasheets which even give you 35 micro amps per megahertz, but really speed should be taken with a pinch of salt, this is you have to really take it very carefully. Because it has to do with several things it has to do with process technology, it has to do with the basic architecture of the system itself and it has to do a lot with the memory technologies as well.

So, if you just say so many micro amps per megahertz for energy and so many MCU speed measured in megahertz, it would not be the right way. Roughly speaking if these three are kept constant roughly speaking; we make an assumption roughly speaking if these 3 are kept constant process technology architecture and memory technology they are kept constant, then you can say that the current consumption doubles, if the frequency doubles. The you can you can say that, but that is very very rough very rough roughly speaking you can say that. So, you have to understand what this process technology is the architecture is and memory is.

So, let us see this is let me highlight you that single aspect of the process part, which will essentially allow you to understand the choice of the controller. So, let us finish of this speed of the controller, I mentioned to you about how the process technology actually comes into picture here. So, when you say speed you talk about clock speed, be careful you actually accurately you must be talking about bus speed you must actually you are referring to more accurately you should be called the bus speed. And this determines how much processing can be accomplished in a given amount of time. Processing that can be accomplished in a given amount of time.

(Refer Slide Time: 22:48)



I am sure you will appreciate the reason for en you said clock speed really the bus speed why because look at the architecture, it is very obvious right this is what will actually hold the final throughput from a CPU basic throughput will actually come from these buses. So, it is just insufficient if the clock here s running at a high speed, it is all about this bus these buses and the architecture of these processors which actually hold the key. So, I am not getting into detail, but I expect that this notes which I am preparing for you will actually give you the understanding of it. Some CPU s can have a narrow clock speed range whereas, others can operate down to even 0, because why clock speed it holds the key is because energy is tightly tied to the clock speed right so, that is another thing.

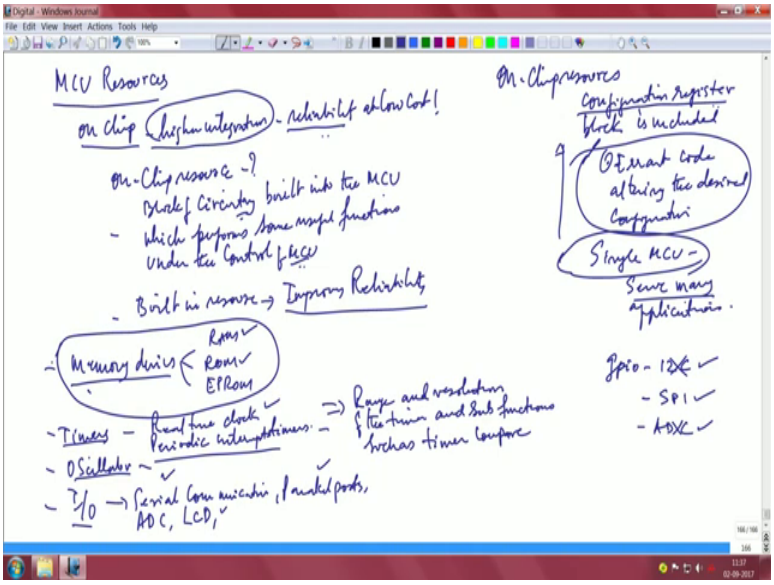
See this clock when you talk about clock of the CPU, this is a very you have to understand it in a lot more detail, because it can be the source for another clock. If you look at MSP 430 or any one of them you will go and see there are many many clocks. The basic clock can be used to generate other clocks that are very critical. So, for instance if you take the case of serial communication, serial baud rates b a u d baud rate you need clock here also right. So, that will basically come from one clock and it can generate another clock. So, in general what you can say is clock is important from a computational power it is required for computational power you quantify it in terms of computational power consumption all these two things essentially will sort of hold the key to the clock. Now I want to get into a little bit of detail on the process technology.

You can be talking about n channel metal oxide semiconductor which is nothing, but the M mass right which we know very well n channel metal oxide semiconductor this is one part. Then you have very high density HC MOS high density high density complementary metal oxide semiconductor. HC MOS what is the advantage of this is it can drive signals from rail to rail, HC MOS can drive from rail to rail unlike earlier CMOS processors hence this criteria can significantly affect. So, what will happen because of it is ability to drive rail to rail noise issues in system design should be born in mind HC MOS uses issues to be born.

So, they are less power hungry because of their basic construction of being high density C MOS semiconductor itself by nature it is less power. So, then you have less heat generation correct the design geometries are much smaller therefore, design geometry is smaller. So, as a result what will happen? You have higher density denser lot of packing when you do lower lot of packing, you get low cost right and all that. So, really if you now look processor itself you will have to actually get into what process technology went behind making of the processor.

So, you may have to bear this is mind. You cannot just say I will choose it based on the clock speed that is really the point we are trying to drive at. Just to summarize the memory technology HC MOS, just to summarize HC MOS not only uses less power and generates and therefore, generates less heat it allows higher bus bits it will not only allow for denser. So, I will say denser designs for a given size and thus allows higher bus speeds and if you say it is a denser design means it is also low cost. So, I would say you may have to look at MCU s what is the memory technology that is used? HC MOS or N MOS this will determine the memory technology process sorry not the memory it is the process technology.

(Refer Slide Time: 31:02)



Next issue that might come to the forefront for us would be related to what are the on chip MCU resources, this is also very very critical. Because we do not have time to do reliability in this course, but reliability parameter from the datasheets you have to pick if you have to choose a particular application for a particular application. Look at space grade, mill grade component why are mill grade components there? because of reliability right. So, whatever can be used for commercial applications cannot be used for space applications, cannot be used for strategic applications, military applications they cannot be used.

Because temperature range or which these components work can be different right and protecting from extreme temperatures when it is for a space application it is very important and there will be heat shields essentially protecting the components because you cannot manufacture for very very high temperature fluctuations. So, what they do is, when the system is being transitioned from here to out of space these extreme fluctuations can be avoided by enclosing them in heat shields and that heat shield is what you will have to worry about.

And after that you once it is in a particular environment known environment, this heat shield can be opened and components can start working. So, lot of things have to be born in mind before you choose anything, and the controller is the key here right. So, that is a very very important thing. So, let us put our attention a little bit on to MCU resources which are on chip I just wrote this quickly for you to understand. First of all if you want very high reliability you must do it at and you want it at very low cost you must look at higher integration on chip. So, what do you mean by an on chip resource? Essentially it is a piece of circuitry which is a slave of the MCU in other words the MCU is in it is control, MCU controls everything that is on chip there and essentially you are talking about improving reliability. What are these on chip resources? On chip resources could include memory timers, oscillators for clock generation, and input output lines input output pins. Essentially when you say memory devices you are talking about ram you are talking about rom, you are talking about e squared prom and so on.

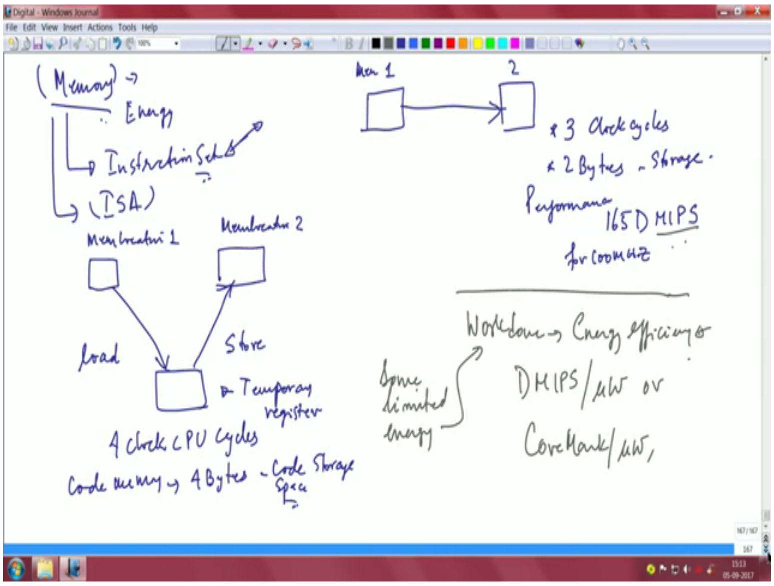
Timers when you say you talk about real time clock RTC which is integrated inside the MCU you can have an MCU without an RTC you can have an MCU with RTC. So, you can use for real time clock operations and you can have also have periodic time for other periodic interrupt timers in other words if you are doing sensing every let us say 5 minutes, you need a timer which will interrupt the CPU and say go and fetch data from the sensor right.

So, how do you do that? You do it with timers essentially and an interrupt is issued and which wakes up the system any other device which is interrupt driven can actually take this input and make available the data back to the CPU. Then IO you talk about serial communication we said 32 bit processors, which are full blown in terms of digital buses like I2C, SPI serial communication ports then debugging ports so many ports are available parallel ports ADC s, then LCD drivers, camera drivers so, many very special functions ports are available all of them will be available as pins, as just like input output general purpose input output pins.

So, you could basically use a configuration register and block register, register block which is included part of the CPU and you can configure different pins for different functionality on the fly and why do you want to do that? Well this is the idea you want to use a single MCU to serve many applications that is why you want to use this configuration register, which will allow you to configure the pins dynamically what can you configure. If you have a set of general purpose input output pins, you can move you can configure that for I2C operations, you can configure them for SPI operations you can configure for ADC operations and so on and so forth. But only thing you have to bear in mind when you do this kind of configuration register based systems is the errant code.

The code that you write if it is a buggy you will have erratic performance from the CPU right suddenly whatever you thought was configured for I2C refuses to work in it is way or a sensor, which is giving you an analogue signal refuses because the ADC configuration pin. The ADC that these pins are configured for ADC operations is gone and because of some arrant code. Therefore, it becomes important to concentrate on take care when you look at configuration of on chip resources.

(Refer Slide Time: 36:50)



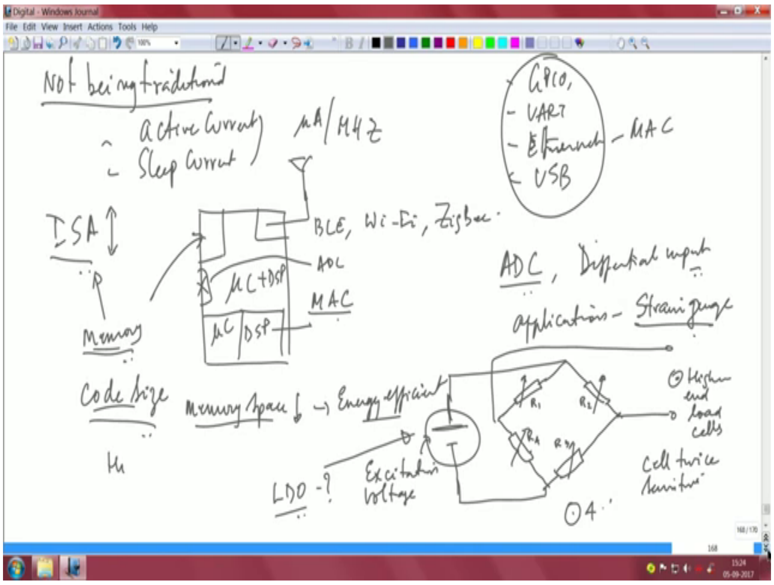
Now, this memory device is very very critical not just from cost perspective, but also from the perspective of the energy. So, we need to understand both. So, memory has to be understood in a nice manner. So, let me just put down a very simple picture for you. So, what does it mean? If you talk about memory you have to relate it to the instructions set somehow. This becomes very very critical the instruction set architecture or ISA is tightly tied to the memory itself and let me give you a very simple example of the importance of this. Supposing you have a memory location and you have another memory location two and this is one. Now let us say you want to do a memory to memory copy what is your objective memory to memory copy.

What would normally happen is you would do. So, I will remove this. So, that I keep it straightforward you do a load and you would a store right. So, you are pulling here and you are pushing it here, essentially this load store is nothing, but 4 clock cycles, 4 clock CPU cycles, this is just an example this is what is this? This is the temporary register. In terms of code memory you need 4 bytes of code storage that is the space nothing, but the space required. Ideally what do you want to do ideally what is there after all if you have one memory location one, mem location 1 and 2 you just want to move just this here. now some processors actually support this kind of operations as well and this is 3 clock cycles then 2 bytes only you need CPU clock cycles performance just compare performance of this system, you will be talking about a 165 D MIPS for 100 megahertz. So, you see another reason why when you do a CPU choice, you may have to keep this in mind if it is highly memory intensive operation in your application, then you maybe better of choosing a processor which supports three clock cycles just two bytes of code storage. So, you can see that program memory gets shorter by half actually right. So, this is storage there are processors which can do that.

So, keep this in mind as well and why what. So, what are we trying to say? The memory operation a memory on chip memory resource cannot be looked up in isolation, it is not about how much memory you have, but how you do it how you operate how do you manipulate that memory? You need an additional memory location temporary register, if you want to use if you do it this way with load and store instructions, you need more program memory whereas, you need less program memory here, you get better performance lower number of CPU clock cycles. So, memory is a very important parameter to be considered when you choose the CPU for your particular application. Keeping in mind the instruction set that is associated with the manipulations of these memory locations.

So, we can see this 165 (Refer Time: 41:54) stone MIPS million instructions per second you should just not look at it from the performance perspective, but you should also look at it in terms of the work done perspective. When you say work done you will again end up with energy efficiency. So, in other words people actually talk about D MIPS per microwatt or some people say core mark per microwatt. Core mark per microwatt essentially we are talking about the work done with the limited amount of energy. With some amount with the limited amount of energy with some energy renew, some energy on hand, some limited energy that you have what is the work done and therefore, what is the energy efficiency which is now measured in D MIPS per microwatt or core mark per microwatt. So, you are just not being very traditional and focusing just on not traditional I will say. So, let me take another sheet let us one not being traditional.

(Refer Slide Time: 43:28)



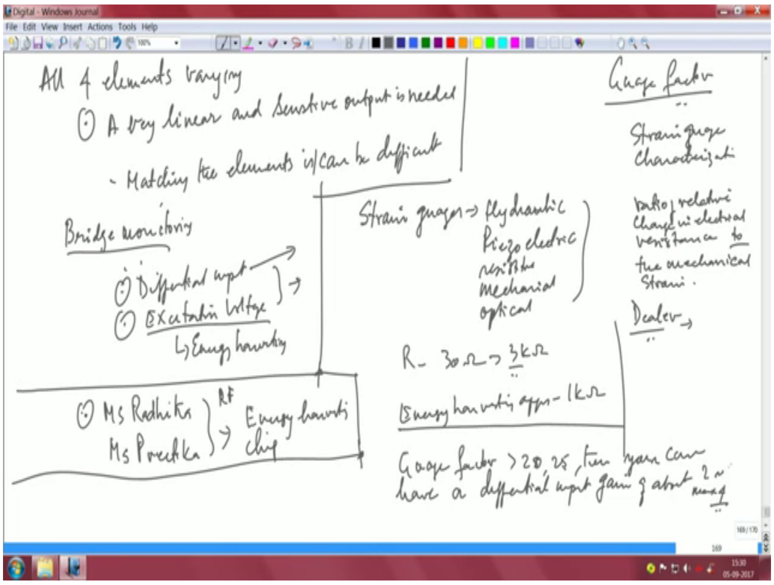
So, when you say you are not being traditional you are not just referring to or focusing on active current of the MCU and sleep currents right. You are not just focusing there, but indeed a lot more which essentially when you talk about these 2 I did mention to you talk about micro amps per megahertz see the difference, it is not just about per amps per micro micro amps per megahertz, but indeed it is a lot more in terms of the energy efficiency in terms of d MIPS per micro watt we did talk about the instruction set architecture and it is importance, but before we move on to focus on the instruction set and all that, we must also look at what does this microcontroller offer for you in terms of it is SOC capabilities.

Perhaps it offers you a BLE radio integrated perhaps offers you a wifi radio or even a Zigbee radio, this is as far as the communication is concerned all right and we also said whether it gives you capability of DSP and microcontroller microcontroller capability, is it DSP cum microcontroller with certain DSP functions which include multiply accumulate instructions, MAC instructions multiply and accumulate instructions or simple load store and these kind of instructions right. Of course, when we did talk about on chip resources we did talk about memory and it is now clear that instruction set architecture has a huge impact on the code size. If you have a very good instruction set architecture the code that you write will be small, the code memory will become small and if memory decreases if memory space decreases then the system is energy lot more energy efficient because you are not switching on that memory cells and in fact, it is quite a compact. So, everything is actually related memory space energy efficiency, code size, the instruction set architecture and so on. But however, having while getting to some details of these you know the choice of microcontroller getting influenced by these requirements, there are other important requirements as well. I will not go into details of you must choose the number of GPIO pins, there is something that is very standard right whether you need AUR, whether you need ethernet interface, whether ethernet MAC end 5 is available, MAC is available for you whether USB there, these are all some things that you will anyway check, but I think I want to draw your attention and I think it is lot more important to concentrate on what is the kind of ADC, that the system actually supports.

How many bit ADC is it and a lot more important thing is whether it will take differential input, whether it will accept a differential input. why is this important? Many applications many applications require which are quite similar to this strain gage applications right, strain gage applications actually are required differential inputs and how does one construct this simple strain gage kind of thing there are 4 arms right in other words there are 4 resistors and what do you do? You take output from here and here you apply sorry you apply the excitation voltage this is the excitation voltage. Now one of them or all of them one of them R1, R2, R3, R4 I simply call them I am I will draw one of them; obviously, it depends on let me put it this way. Now whether you want to make only one of them variable or whether you want all 4 of them to be variable or you want only a pair of them to be variable is a call that you will have to take. You can also have one you can also have two opposite pairs anything is possible, normally I will put it this way higher end.

So, let me take another no let me write it here. So, that it is good to see higher end load cells are commonly constructed this way. Higher end load cells load cell basically comprise of this strain gage right this is a strain gage and nothing but the load cell is nothing, but strain gages in a (Refer Time: 50:08) bridge configuration and higher end load cells are essentially constructed in this way, with the two strain gages connected oppositely, this makes basically the cell twice sensitive, but I can also tell you that you can also have all 4 ok.

(Refer Slide Time: 50:55)



Now in other words you can have all the 4 elements let me write in another sheet, all 4 elements are varying. This is done here is the key here this is done where a very linear and sensitive output is needed. So, hard part is matching the elements is difficult or can be difficult. So, it is sometimes rare to see you know all 4 elements varying particularly if you are talking about any low end application. But if it is indeed a very high end system you may have to actually make all 4 of them varying elements and what kind why are we talking this we are talking, this because if you are talking about bridge monitoring then typically all 4 elements are varying elements and why it is connected to microcontroller, because you will need to apply.

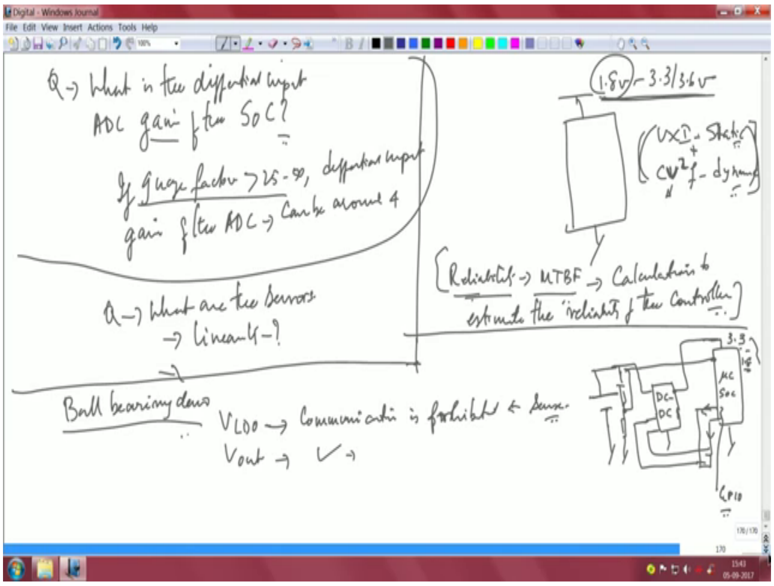
So, what you need you need differential input and you need excitation voltage bridge excitation voltage. Now go back and put everything that you know is this excitation voltage derived from energy harvesting. If so, you are in deep trouble because the energy that is harvested has to be stored into a capacitor, and you have to maintain the excitation voltage constant right. Every time you are making a measurement because this excitation voltage is varying all the time if this excitation voltage coming from this is varying all the time the output is going to also keep varying. So, you are doubling in a problem therefore, you must somehow ensure that this excitation voltage is kept constant during the time when you are making the measurement, which means it leads you to the requirement of an LDO. So, energy harvesting, LDO then the excitation voltage for the (Refer Time: 53:53) bridge all of them at the end of this course you realize are all connected and that is important this is one part. The other part which I wanted to impress upon you is with respect to the sense the strain gage sensors themselves. Now let me put down a few things, basically there are a variety of I have just taken one sensor as an example to highlight how you will choose a microcontroller this is just to give you a overall idea.

Basically if you take by the way all this work was done by two of my bright students miss Radhika and miss Preethika two bright students of my who worked on energy harvesting chip RF energy actually RF energy harvesting chip with an interface to a sensor like the strain gage. So, I am actually telling you from our experience on how to build how do you choose a microcontroller for a circuit, for a given application based on this requirement. So, let me come back here. So, if you look at strain gages you will see that there are many of them you will get hydraulic right hydraulic, you will find hydraulic piezoelectric, resistive right mechanical optical and so on and so forth.

Now the strain gages. So, basically it is a resistive element right you can have the strain gage R value varying from 30 ohms to about 3 k ohms it can be up to as high as 3 k ohm and higher the resistance value, good for the energy harvesting applications. For energy harvesting apps higher the R value better because the current consumption will be low will be minimal right. So, it is good to have a higher value. So, the in the system basically you will be connecting all the 4 which are all the 4 arms which are varying. So, that is indeed the way by which you are actually making a measurement.

The strain gage all the strain gages is a it is associated with a very important parameter and that parameter that you have to note indeed is the gage factor. People talk about gage factor which is an essential parameter for any strain gage characterization. What is this gage factor; we can look up Wikipedia, but let me tell you it is a ratio of relative change in electrical resistance to the mechanical strain. Please look up Wikipedia you will get understand this very well gage factor is. So, dealers essentially when you have when you go to a dealer to buy one strain gage, they will not tell you the strain gage this gage factor of the strain gage. So, you may actually have to perform an experiment and actually evaluate the gage factor, and you expect that the gage factor is linearly dependent on the strain that you measure, but whereas, the strain is directly proportional to the bridge sense voltage. So, all this will have to be born in mind. So, the key point is here, the gage factor if it is a high value let us say greater than 25 or 20 or 25, then you can have a differential input, input gain of about 2 up to a maximum of about 4, but if gage factors is low then your gain has to be increased. Question is whether what is the differential input ADC gain of the SOC? This is the question.

(Refer Slide Time: 60:03)



So, all the story ultimately boils down to the choice of the microcontroller will essentially mean if you are using a strain gage, and if the strain gage has a gage factor which is low then you must use a differential input amplifier op amp amplifier for instance, differential input ADC cum amplifier essentially it is because a strain gage gives you analog output right. Differential input analog ADC cum amplifier which is suitable from the which is possible from the microcontroller. So, summary if strain gage factor is greater than 25 or even close to 50, the differential input gain of the ADC gain essentially means it is also an amplifier right.

It is taking input it is amplifying and giving you can be around 4 or something like that. But if this is low you must put either an external you must use an external ADC or your controller choice will be dependent on whatever gives you the maximum gain that is the controller of your choice. So, you see now this is the big summary and you just cannot choose a controller, because it has a 8 bit ADC or a 12 bit ADC or a 16 bit ADC and so on, but you also have to keep in mind what are the sensors that you are going to connect. What are the sensors what do you want from these sensors, do you want them to maintain linearity and if so, how will you configure them will they take differential input what is a gain these parameters have to be born in mind to choose a microcontroller. Another important thing I mentioned to you about excitation voltage for the strain gage, excitation voltage across the strain gage it is also important that the microcontroller itself works over a range of input voltages for example; you can have a microcontroller which works from 1.8 volts up to about 3.3 or 3.6 volts. So, you should find out whether this microcontroller can actually work successfully over a range of voltages. Obviously, certain functionality will be compromised if you are working at very low voltages, when I say very low I mean in this order of 1.8 maybe certain blocks cannot work, but that still does not mean that you cannot do for example, some amount of compute maybe still possible. So, again it will depend on what is your application and really you have to focus a lot on the final application of interest before you actually choose the microcontroller.

There is a very important parameter which you will have to choose and that is with respect to it is reliability, you may have to look up the datasheets for the MTBF and use this MTBF in calculations to estimate the reliability of the controller. It is a topic by itself I would not like to go into detail, but you must note that reliability of the component is an important thing. Here we are not talking about any vendor in the irrespective of any vendor you must do this reliability calculation use the MTBF mean time between failures from the datasheet and estimate the reliability of the controller of your choice. So, this is another thing if you do the low reliable thing do not choose that you put a bar and say it should be so much reliable and only then I will actually use it in my applications. So, choice of microcontroller means what see the parameters that seem to pan out.

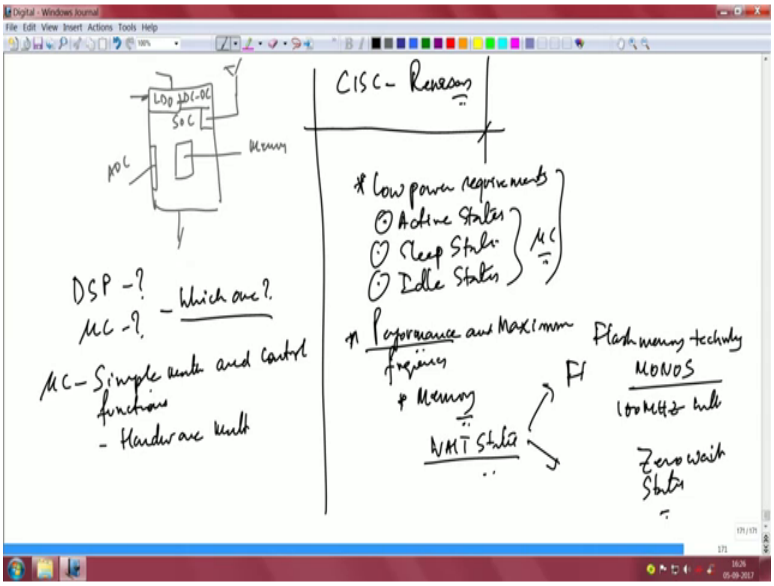
One is you are also talking about the voltage range you are talking about reliability, you are talking about ADC points, you were talking about instruction set architecture, which in turn relates to the on chip resources like memory right all of this comes like a capsule of requirements that you will have to satisfy in order to choose a microcontroller. So, I mentioned to you about the rail voltage, this rail voltage the operating voltage of the microcontroller and I said that it should go between range from 1.8 to let us say 3.3, 3.6 you can see again power becomes critical here right see when you talk about power we talk about overall power consumption as static power which is v into I this is static power and C V square f as the dynamic power and you add both of them you get the overall power. Clearly if V keeps going down the energy power consumption of the chip also is lower and lower is lower right. So, that is why lower voltages are attractive because from a power perspective both just not in terms of the static power consumption, but also during it is dynamic working time of course, f is a very important thing higher the frequency the power consumption increases as it can be seen here, but if you are able to offset it by lowering V because it changes as square it has a higher impact significant impact by reductioning the voltage.

So, this is what you have to bear in mind when you have to choose the controller. Apart from that this equation clearly indicates that while we do certain compromises on it is working recall we actually exploited a few things during the bearing demo ball bearing demo what did we do for example, in the ball bearing demo, we said unless the V out is available with VLDO we will do a few things VLDO and V out we will do a few things with VLDO for instance we would not do a communication is prohibited, but moment it shifts from VLDO to V out you do a communication and you do a few other things whereas, with VLDO you only do a sense if VLDO is there.

Clearly if you are able to sense the battery voltage right and to any of the analog pins this is the microcontroller SOC if you are able to sense and the reel itself the voltage operating voltage itself comes from the DCDC converter right and two programmable pins are available to you by which moment this voltage starts dropping which is sensed by the microcontroller the microcontroller SOC has the ability to program over the GPIO pins general purpose input output pins, configure a value into this DCDC converter which will reduce the voltage here from let us say it began with 3.3 and moment a signal is given a bit combination is put up here, it becomes 1.8. And how does who takes this decision of shifting from 1.8 to 3.3 it is the controller and how does the controller take? It is able to sense the battery voltage; obviously, battery voltage sensing is I have to make this circuit a little more accurate it is sense here and not like this ok.

So, this is important yeah I have to correct this this is coming from here and input perhaps goes here this is important put two resistors R 1 and R 2 put a simple resistor divider network sense it and use that for taking an intelligent decision. So, you should be able to write nice code nice software code which it is possible, but then the controller itself should be able to work in this range. 1.8 volts to 3.3 and that is therefore, that it is a very important requirement. Often this DCDC converter and even an l d o are also sometimes integrated inside the LDO and DCDC are integrated inside the SOC we need many times choice of whether you want t o use LDO or DCDC you can make with an external pin ok.

(Refer Slide Time: 72:28)



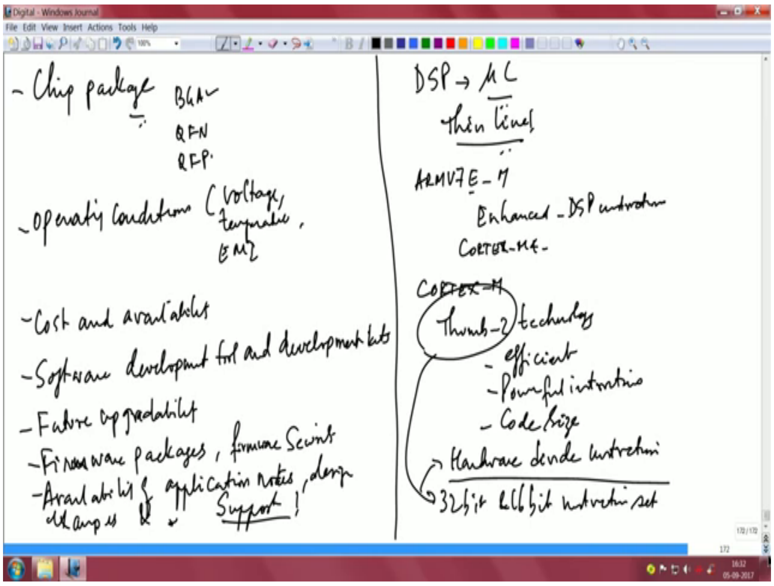
So, there is so, much of integration as I mentioned to you radio is there, ADC is there memory is there, memory I should show inside memory block is inside. So, this is very very important. Let me summarize this choice of microcontroller with so many other related things each and every point that we are going to say are very very important. Now as I mentioned to you I like this book I showed you this book last time right and I said arm cortex M 3 and cortex M 2 processors a it is a definitive guide. This is a good book and I enjoyed reading several chapters in this book, and I thought I will share a few things from this book to you which will help you to you know sort of cap everything into one nice capsule.

On choice of microcontroller I do not want to say what you should choose because there are so many people in the market, CISC if you take CISC processors Renisas is there, CISC Renisas that memory to memory copy which I showed you picture works beautifully with Renisas microcontroller, but how many of you are using it we do not know Ti. Ti has a range of microcontrollers free scale automotive markets they seem to have quite a few controllers then you have free scale Ti Intel, Intel is a big player right turbine into this microprocessor and microcontroller for 8051 is such a powerful popular architecture. So, it would be difficult Nordic, Nordic is another company. So, it is almost impossible for us to take a decision on what should be a the controller that you should choose whatever we have discussed till now are very important, and we will discuss a few more points and summarize everything in a nice way. Let me summarize first and then come backwards, what this book is saying you will have to look at following parameters to choose, low power requirements, many many things we have said about low power if you look at this lecture itself in the in it is form, you will see that energy efficiency power requirements energy harvesting capability, all of that seem to have occurred quite well there. Here very quickly you must look at the active states and the sleep states and the idle states that the microcontroller offers to you. You must study them in detail find out what is the current consumption purely from micro amps per megahertz specification perspective.

So, this is very important look up that. Performance and maximum frequency here you will be stuck in many ways I showed you that picture of load store where memory copy happened in two different ways, the RX MCU did it in just memory copy, and the other one put it to a temporary register and pulled it out umber of clock cycles are more it is doing a simple operation there is a low out there is a store operation involved. So, architecture plays a huge role in the whole process then we also mentioned about the fact that memory is a the technology is a process technology becomes important from a performance perspective then memory technology. See one thing is clear memory can pull down your performance drastically, why because if you have a slow memory which is which is embedded inside the controller and processor is executing it at very high speed you have to introduce for slow memories what are known as wait states you are stuck now.

There are two types of flash memories in the flash memory technology, there is a technology which allows you to run the to use the flash without the ah without introducing this one what is it without introducing wait states, which is called I am just trying to recall the name of the flash memory technology let me say effects calculation. So, MCU s that use MONOS flash technology can run at even at up to 100 megahertz without this is important without wait state; that means, that is with let me put it this way with 0 wait states. So, are you talking about bus speeds and clock speeds with memory access, with weight states or without weight states. So, performance is not so easy to actually quantify and this will differ from different MCU s. So, some MCU s may use simple flash because they want to cut down costs, and some may have the sophisticated mono flash technology. So, even at a lower clock speed performance will be comparable ok. So, keep that in mind what kind of flash technology that is being used.

(Refer Slide Time: 79:43)

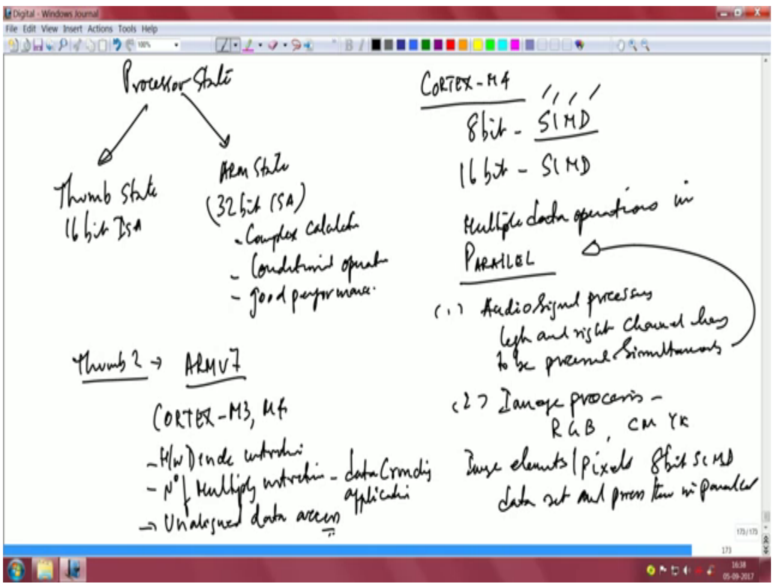


Book actually refers to is on the chip package, please do look up the chip package related information because whether it is available in what kind of packaging whether it is a BGA r is it a ball grid array package or whether it is a QFN or a QFP and what is the size of the package all that QFP so on are very very important. Operating conditions are important operating conditions are important what do you mean? Voltage temperature EMI and so on cost and availability software development tool and support for development kits right future upgradability firmware packages security firmware security availability of application notes, design examples and lastly of course, support ok.

All these things do come in we will have to be considered if they are if you have to choose a microcontroller, what I have to bore this bear these things in mind. There is always this problem that whether you want to choose between a controller by a simple microcontroller or whether you want to use a DSP, frankly this line is thinning down thin line now and currently it is a very thin line many things that a DSP can do is being already achieved by microcontroller very specific and very special instructions, which can do that. If you take example of arm the evolution of arm architecture V7E if you see this word E this letter E actually means enhanced is for enhanced and what do you mean by enhanced? It is an enhanced it has enhanced DSP instructions what a lovely thing already and particularly cortex M 2 has this these DSP enhanced instructions cortex M4 basically if you look at cortex m itself cortex M series of processors they support thumb 2 technology ISA which is nothing, but the instruction set architecture very efficient it is a very efficient technology powerful instructions and delivers a significant performance in terms of code size and all that. It also has divide instructions hardware divide instruction, look what is happening already right the thumb two technology essentially provide you a few if there is a support for thumb 2 means that you do have hardware divide instruction capability. So, you will be able to do both 32 bit and 16 bit instruction that is the key thing 16 bit instruction set it has both.

So, when you say thumb 2 you are talking about all these nice features which are coming in the instruction set architecture and; obviously, all these things essentially will lead to energy efficiency compact code size, and all features that we all the main features that we are really looking at. So, the support for thumb 2 technology and essentially means that you will be you basically have a smaller instruction set and instruction set is very compact very small and very compact basically you will talk about two processor states essentially. So, let me put down that, if you take arm cortex M which is cortex M 0 it has a very small an energy efficient it is a basically a very small and energy efficient processor, and it is easy to use quite like cortex M 3 or M 2 just let me put down this processor state ok.

(Refer Slide Time: 86:12)

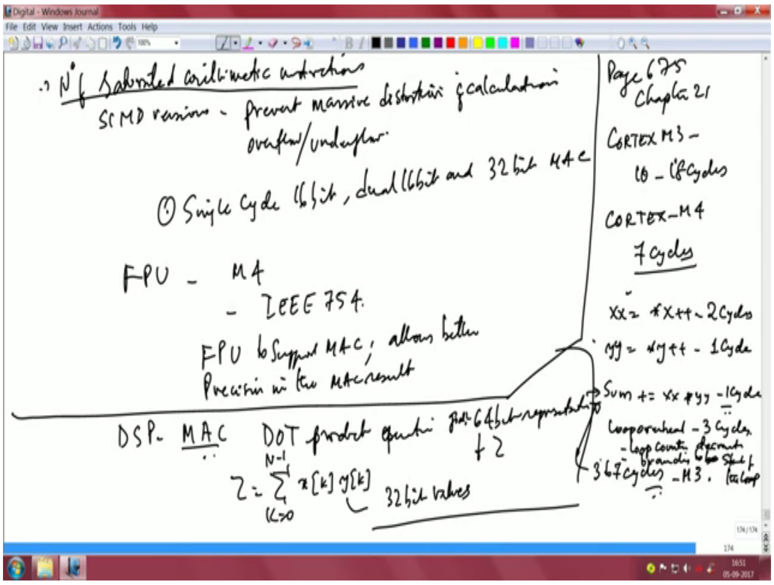


Essentially you are talking about thumb state and you are talking about the arm state. This arm state essentially is all about the 32 bit ISA and this thumb state is nothing, but the 16 bit ISA. So, here you can carry out very complex calculations, large conditional operations which include I would say large complex calculations means, large conditional large basically large conditional statements I would say conditional operations and of course, you get very good performance. Just to continue on the thumb two there are a number of new features both in if you take let me just take a example arm if you take arm architecture V 7 if you take you will get cortex in this you can buy you can source cortex M 3, M 4 all of them large number of new features and these include hardware divide I already mentioned multiply for data quenching operation.

So, I will say divide instructions I already mentioned this, but I will just write it for completeness divide hardware divide so, very important hardware divide then number of multiply correct instructions multiply instructions for data crunching for any data crunching applications look at them. What is also interesting is which is not which is normally available on very high end processors, there is something called an unaligned data access and this is also supported on cortex M 3 and M4. So, cortex M 4 for instance let me give you a very contrasting thing if you take cortex M 4 it has 8 bit SIMD it also has 16 bit SIMD what does it stand for? Single instruction multiple data why is all why are we talking about this? Because if your processor if it is if your processor is likely to do multiple data operations, if your processor is going to do multiple data operations in parallel this is important if it is doing things in parallel, then you will need instruction set support for SIMD could be 8 bit or 16 bit SIMD.

Let me give you very special very specific example look at just audio signal processing just look at audio signal processing what do you do here the left and right channel has to be processed simultaneously right this is what we mean, and for that you need these single instruction multiple data kind of instructions. Second application you can think about is image processing. If you are looking at RGB or you are looking at CMYK kind of elements of I mean image elements I will say image elements or image elements or pixels image elements slash pixels and you need to represent them you can use 8, but SIMD dataset and process them in parallel.

(Refer Slide Time: 91:39)



So, what happens is all RGB gets processed in one go just straightaway it gets processed in one go, it can also have a number of saturated arithmetic instructions including I mean the SIMD versions for preventing any massive distortion of calculation number of saturated arithmetic instructions including SIMD versions to support to ensure that you want to prevent massive distortions in calculation of calculation distortion of calculation right which results in overflow underflow and so on. So, you have to bear this also in mind. If you take some examples of this saturated arithmetic instructions one example I can give, you is single cycle 16 bit dual 16 bit and 32 bit multiply and accumulate.

Cortex has this instruction MAC instruction they can have many many such options multiplication of various combinations of upper and lower bits SIMD versions for 16 bit MAC single cycle is also possible in M 4 M 3 perhaps you need multiple cycles, but M 4 you can do it in single cycle what about FPU? FPU floating point unit what about FPU if you take the case of M 4 cortex M 4 it is compliant to the I triple E 754 standard single precision half precision you know all that is supported and FPU to support MAC right FPU 14 point unit to support multiply and accumulate means what does it mean? Allows better precision in the MAC result so, what we are trying to get to is that there is hardly any difference between the thin line between a normal microcontroller and a DSP is thinning down, and systems are getting very powerful one example is the cortex M 2 if you read it in detail you will understand kind of very sophisticated things that cortex M 2 can actually do arm cortex M 2 can actually do and just as an example to show that the it is a efficient yet you do not need a DSP per say for any of these multimedia related signal processing applications that you are looking at. This is in big summary about the choice of controllers that you have. Before I close I want to show you how simple in DSP if you have a digital signal processor if you have a MAC option multiply and accumulate option how simple it is to do a dot product to do a dot product operation basically. So, essentially this is what you want to do right k equal to 0, let me 0 to n minus 1 of this vector and this vector. So, these are the two vectors these are let us say 32 bit values and that you want to represent it goal is to represent 64 bit representation basically represent in. So, let me say 64 bit this is the goal. 64 bit representation of Z.

So, if you look at this example you will see that essentially it is just a few lines of code it is just a few lines of code, let me point you to that page from where I can actually put down the MAC. I will just point you to that in case you want to have a look you could do that and see how simple the instructions are. As I said I am pointing you to page number 6 7 5 of this book chapter on chapter 21 of this book. What he does is, the dot product basically it consists of a series of multiplications and additions right. So, we will leave that cortex M 3 of you take cortex M 3 if you take the inner loop of the dot product takes 10 to 18 cycles with the execution time being data dependent. 10 to 18 cycles is what cortex M 3 takes, but if you look at cortex M 4 the code is identical to that of the cortex M 3 you have 7 cycles and how is this 7 cycles split? You have this is 2 cycles then for y y this is 1 cycle then sum this is 1 cycle and loop over a is 3 cycles.

Essentially this is fetching data right this is fetching data from memory and incrementing the pointer fetching data and incrementing the pointer, then the next fetch this is the next fetch y y is the next fetch then there is a multiplication and addition correct and this is what this step is this is multiplication and addition right the main MAC part, which is required for the DSP operation which is now easily achieved in cortex M4, but takes more number of cycles in cortex M 3 being displayed here this is that part right.

Can which is amazing for cortex M 4 and this is where the cortex M 3 goes for a toss, heat can take anywhere from 3 to 7 cycles this is for M3. So, if you are trying to look by trying to buy a controller deciding on a controller look for this MAC operation which is it cannot be better than this just one cycle the loop itself will introduce an additional overhead this usually involves decrementing the loop counter. So, this is all about loop counter decrementing, then branching if any, branching and so, basically branching to beginning of the loop right.

You want to go to the start of the loop, all of that will require this additional 3 cycles. So, in essence this is all you need to worry about in terms of choice of a controller for an application.

Thank you very much.