Design of Mechatronic Systems
Professor Prasanna S. Gandhi
Department of Mechanical Engineering
Indian Institute of Technology, Bombay
Lecture 17
Microcontroller Architecture I

(Refer Slide Time: 00:11)

Lecture No. 17

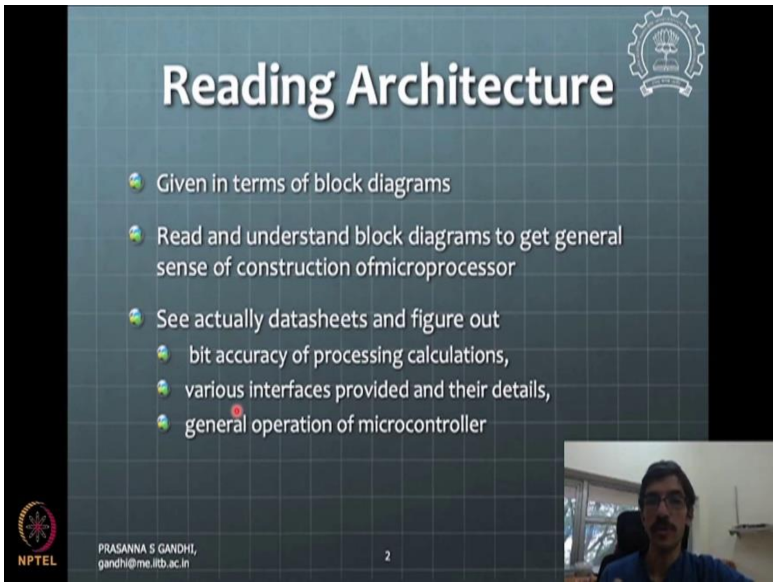
Microcontroller Architecture I

So, now for today's class we will start with a new topic "microcontroller architecture", and in that we will particularly look from the perspective of interfacing sensors and actuators and different kinds of microcontroller that we will need to know, so that we can choose appropriate ones for the application that we are looking for.

So, I am presuming for this class that you already have some exposure to the microcontroller programming and some kind of idea about how things go, and we will build on those ideas now, to kind of get to like more advanced versions of microcontrollers. And, one of them is Tiva board which is ARM Cortex M4 series microcontroller board that we will take up for programming in this class, and we will build some of the experimental understanding based on that syntax. So, now for a couple of classes we will look at this microcontroller stage programming details (some of the details), and I will post lot of material for that and you will get to kind of understand.

Once your microcontroller boards are there with you, maybe, we can run through to resolve some of the issues that you might be facing in one of the discussion classes. So, that is how our journey is going to be for next couple of lectures now, so let's begin.

(Refer Slide Time: 02:15)



Reading Architecture

- Given in terms of block diagrams
- Read and understand block diagrams to get general sense of construction ofmicroprocessor
- See actually datasheets and figure out
  - bit accuracy of processing calculations,
  - various interfaces provided and their details,
  - general operation of microcontroller

PRASANNA S GANDHI, gandhi@me.iitb.ac.in

2

So, see for a microcontroller, you need to be having the skill to look at the microcontroller architecture and understand some of the details of the architecture. Now, there can be different levels of this architecture, I will explain you what it means. So, usually there is a lot of information that will be given in terms of what today's microcontroller can do and cannot do, these are given in terms of block diagrams or some architectural understanding will be there.

So, we should have the ability to see these block diagrams and figure out what it means for us. So, we will look at some of these datasheets of the microcontroller and then figure out this exercise we will do as we go along with the different microcontrollers and some of the things that we discuss in today's class also.

So, typically, like bit accuracy of processing these calculations, so it is 8-bit microcontroller or 16-bit or 32-bit accuracy is there. Then, what are the interfaces are provided by this microcontroller? So, if you start off with one of the base “8085 microcontroller”, microprocessor actually, it was like the basic computer chip to begin with, when Pentium chip came, 586 kind of thing, it is, you can go back in history and see 8085 was one of the first computers that was developed, and it is still good kind of an academy understanding purpose, it is good to see how that microcontroller really works.

So, some of the basic understanding of how the commands are executed in micro processor or computer or microcontroller, there are similar kind of a base commands are executed. So, that all we will see based on the architecture in the block diagrams. So, I will post these datasheets of 8085, then some modern microcontrollers and our ARM Cortex, so you can kind of compare these data sheets and see and try to make sense of some of the things, some part of the thing that is there, we will try to go through some of the important aspects that we look for.

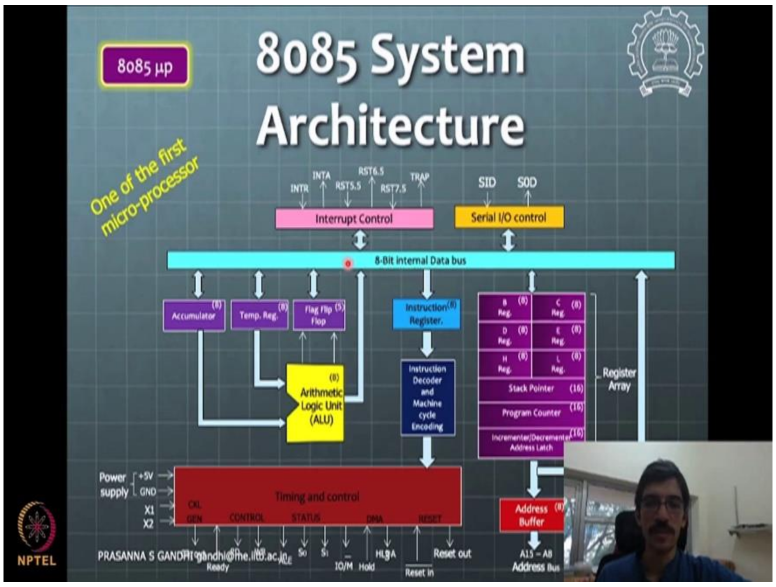
So, for example, these interfaces, in terms of interfaces, the 8085 will not have any interfaces to be provided, it will have just some kind of a ports that will be available as parallel ports and serial ports and something like that. And then, one has to have some additional peripheral chips interfaced with this microprocessor to make it microcontroller. So, that is a basic difference to make it a hardware interface for exchange of the data, not really microcontroller.

So, there is a difference between the word used as a microprocessor, and what uses a microcontroller. So, when you have a microprocessor, like you have Pentium chips and other computer chips are all your microprocessors, actually, and they typically will have to be interfaced with the external peripheral chips to talk to any hardware or environment or pin outs, somethings like that.

But microcontroller will have all those peripheral also will be a part of the chip. So, microcontroller chip will be very tiny kind of a chip in which everything is there, you have your ADC, DCA interface or name it whatever interfaces that microcontroller provides that are all already there on the chip itself, and only the pin outs terminals are out actually.

But microprocessor, you will have only the processing part on the chip and then all the control part or the interfaces peripheral devices, peripheral interfaces, basically, they will be a part of additional chips that are provided on board. So, the board will form entire control unit, but the chip by itself cannot form entire control unit. So, that is a difference between microprocessor and microcontroller and you should be aware about that.

(Refer Slide Time: 07:07)



So, let us move ahead this basic architecture of 8085 system, typically it is given in this kind of a form, this is a base level architecture at the elements which are there in the microcontroller. So, typically you will have this data bus indicated, , this represents like a bidirectional data bus, in some places there is a unidirectional data bus, it is not bi-directional.

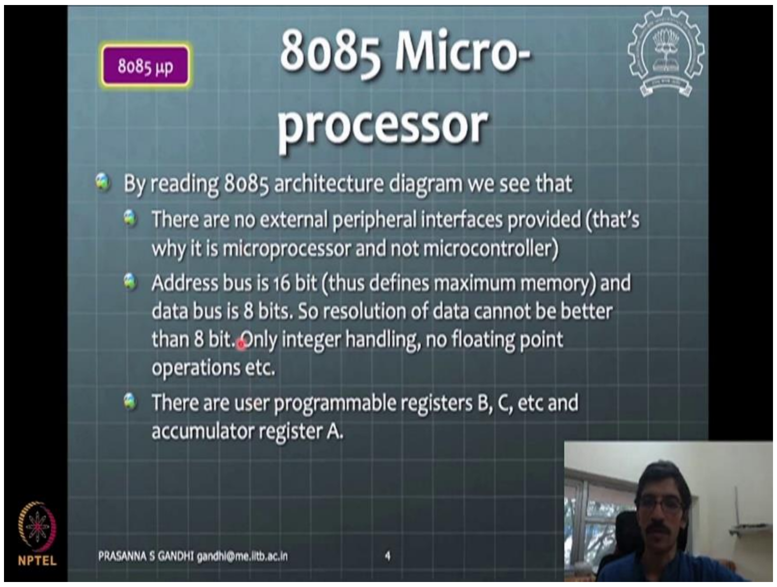
So, this arithmetic logic unit is small processing unit in which these registers can transfer data through this data bus, and the data from output of the value can be transferred to this another data bus. So, these are typically given block diagrams, one can read through to this block diagram, what are things this microcontroller has inside. So, this is not a microcontroller, in this case it is 8085 microprocessor chip.

And it has this typically internal register arrays will be there, so this 8-number indicated it is 8-bit register BCD some names will be given to the register and typically it will have a stack pointer program counter, these are some of the standard registers, it will be there in every microprocessor or microcontroller and of course there will be this timing and control unit.

So, this is an architecture block diagram, the basic level block diagram of 8085 nowadays a modern microcontroller you do not see this kind of board block diagrams. They will give like a more next layer of the block diagrams, in the sense, they may not give you all these registers, I will show you some of the block diagram, I mean you will get more clarity based on that.

So, when you say this is address bus, typically this data bus, if you see this 8-bit data bus is there, then address buffer is there, address bus is AD17 to A15 or address AD15 is like a 16-bit address bus is there for 8085 system of architecture, but these detailed specifications may not be there in the modern datasheets for a microcontroller, we will see some of the things there.

(Refer Slide Time: 09:47)

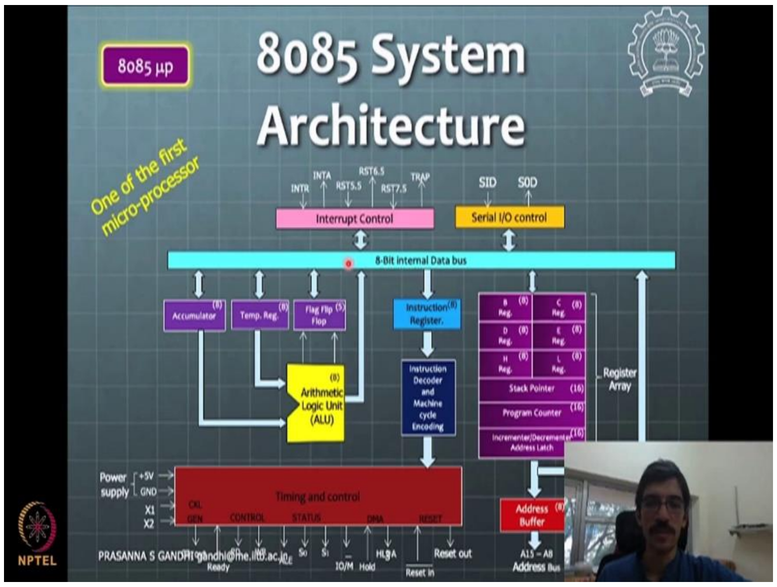


8085 µp

8085 Micro-processor

- By reading 8085 architecture diagram we see that
  - There are no external peripheral interfaces provided (that's why it is microprocessor and not microcontroller)
  - Address bus is 16 bit (thus defines maximum memory) and data bus is 8 bits. So resolution of data cannot be better than 8 bit. Only integer handling, no floating point operations etc.
  - There are user programmable registers B, C, etc and accumulator register A.

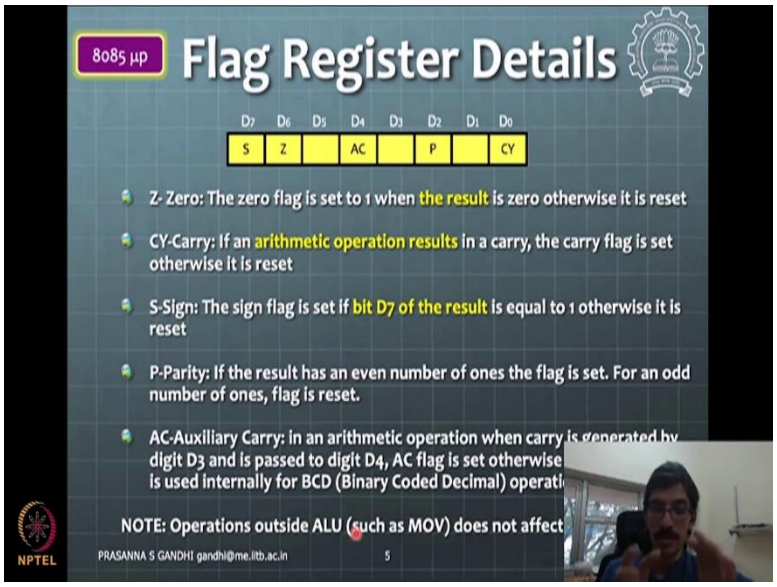
PRASANNA S GANDHI gandhi@me.iitb.ac.in 4



So, see that there are no external interfaces you can see in the 8085, so that is why, it is microprocessor and not microcontroller as we saw the definition. Then like that you can read all the things that I talked about on the last slide. There are many things you can make sense about and the instructions when they are given for such a microcontroller then they will be given in terms of these registers, so, there is a move instruction, contents of (say) move BC, so contents of register C are moved to register B, that some kind of a commands will be given, and their explanation will be given.

If, you do not know the structure of this architecture, then you will not make sense of those commands. So, if this diagram is not given and I am just giving this explanation of this command move B, see, it is very difficult to see, what is going to happen in the system. So, these architecture diagrams are quite important for many microcontroller systems.

(Refer Slide Time: 11:06)



8085 µp

Flag Register Details

D7	D6	D5	D4	D3	D2	D1	D0
S	Z		AC		P		CY

- Z- Zero: The zero flag is set to 1 when the result is zero otherwise it is reset
- CY-Carry: If an arithmetic operation results in a carry, the carry flag is set otherwise it is reset
- S-Sign: The sign flag is set if bit D7 of the result is equal to 1 otherwise it is reset
- P-Parity: If the result has an even number of ones the flag is set. For an odd number of ones, flag is reset.
- AC-Auxiliary Carry: In an arithmetic operation when carry is generated by digit D3 and is passed to digit D4, AC flag is set otherwise is used internally for BCD (Binary Coded Decimal) operati

NOTE: Operations outside ALU (such as MOV) does not affect

PRASANNA S GANDHI gandhi@me.iitb.ac.in 5

Typically, there will be these registers which are indicating some kind of operation done or not done. So, this is called a flag register, flag is something which you show when something is done or not done or something like that. So, this flag register also has this sense of notion, that it has some kind of a bits which are made 0 or 1 depending upon the operation that had happens. So, for example 0 flag will be set when some result of operation is 0, so this can be used for, further for looping operations.

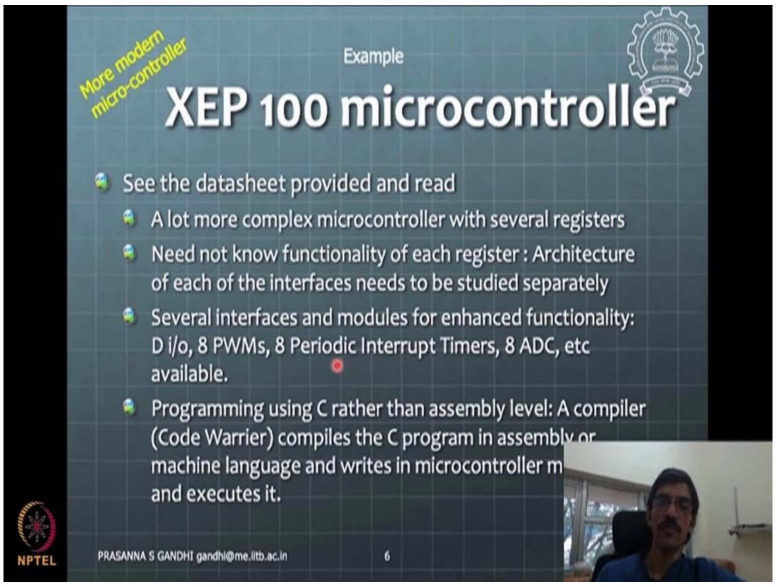
Say, for example if you have a variable which is decremented and when it becomes zero, this flag is automatically set and then one can read the flag register and take the control of the loop, okay I need to exit this loop after that flag zero happens there. So there is a like a base level that is what happens in the microcontroller or microprocessor. So, this flag register is used for many of these control commands that you typically see if while loops, so those kind of breaking or looping or coming out of the loop or conditional operation execution that all depend upon this flag register.

So, flag register plays a very important role in any microcontroller, it will have it, I mean you may not be able to see as a part of the architecture, it may not show that in modern microcontroller diagrams, but it is it is anyway there. So, like that there are many different things that you can read through and make sense, this is what happens like that.

So, see there is so much in other notes will be given operation outside ALU, such as this MOV command, move the contents of the register that will not affect the flag like that something will be given.

So, each command reference or whatever this instruction set it is called commands or commands of microcontroller they are typically called instruction sets. This instruction sets to give you the details of what that instruction does in the microcontroller. And typically, while giving that details, they will refer to a say okay the flag register is set, some bit in the flag register is set or nothing will change something like that will be given as part of the command instruction details.

(Refer Slide Time: 13:51)



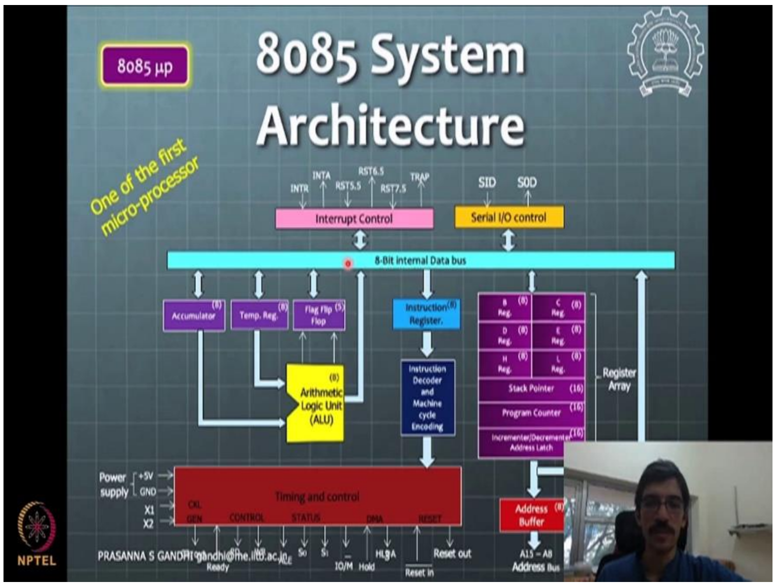
More modern micro-controller

Example

XEP 100 microcontroller

- See the datasheet provided and read
  - A lot more complex microcontroller with several registers
  - Need not know functionality of each register : Architecture of each of the interfaces needs to be studied separately
  - Several interfaces and modules for enhanced functionality: D i/o, 8 PWMs, 8 Periodic Interrupt Timers, 8 ADC, etc available.
  - Programming using C rather than assembly level: A compiler (Code Warrier) compiles the C program in assembly or machine language and writes in microcontroller m and executes it.

PRASANNA S GANDHI gandhi@me.iitb.ac.in 6

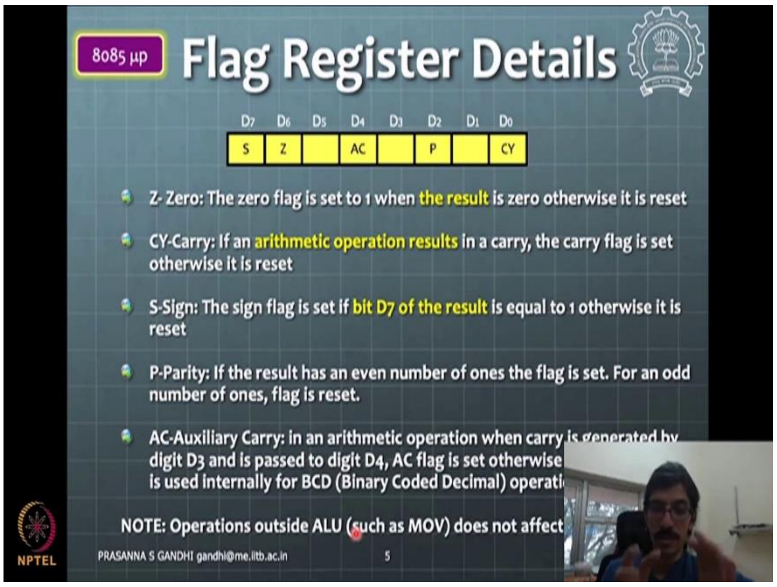


8085 µp

8085 System Architecture

One of the first micro-processor

PRASANNA S GANDHI gandhi@me.iitb.ac.in


8085 μp

Flag Register Details

D7	D6	D5	D4	D3	D2	D1	D0
S	Z		AC		P		CY

- Z- Zero: The zero flag is set to 1 when the result is zero otherwise it is reset
- CY-Carry: If an arithmetic operation results in a carry, the carry flag is set otherwise it is reset
- S-Sign: The sign flag is set if bit D7 of the result is equal to 1 otherwise it is reset
- P-Parity: If the result has an even number of ones the flag is set. For an odd number of ones, flag is reset.
- AC-Auxiliary Carry: In an arithmetic operation when carry is generated by digit D3 and is passed to digit D4, AC flag is set otherwise is used internally for BCD (Binary Coded Decimal) operati

NOTE: Operations outside ALU (such as MOV) does not affect

PRASANNA S GANDHI gandhi@me.iitb.ac.in 5

Now, let us come to take a more modern microcontroller, if you see the datasheet and I will share with you the datasheet. So, this if you see will be like lot more complex microcontroller as compared to 8085. There are several, I mean, many many different registers, in fact for each of the interface you will have a different architecture block diagram.

So, we know as a user should not, need not know functionality of each of the registers, we are not interested in that what we need to understand is how these things are and from programming perspective, how I can make use of this information that is given as a part of the architecture. That is what we are concerned with, we are not really interested in how this microcontroller is designed or how it is working inside, we are not interested in knowing how it is working inside.

Although, I mean, as a curiosity we may go into a little bit more details of some of the things, but I mean really it is not needed to know what or how things are working like that. So, something is given like if you move, if you adjust the contents of this register to this value then this will happen like that it will be given. So, you just believe that it will happen you do not need to know how it happens exactly.

There are a lot more details about architecture up to the register level needs to be known to really figure that out. So, we are not going to get into those details and that is not intention of this class. So, our intention is to see these block diagrams and figure out how can I program these to achieve, what I want to achieve with this microcontroller. So, that is what we are looking at all these diagrams from that perspective.

And generally, to know what are the limits that this microcontroller can be stretched to. Knowing the block diagrams one can say this cannot be possible with this microcontroller. This is absolutely given the block diagrams, reading the block diagram we can come to that conclusion this particular thing cannot be possible or this is up to this extent it can be possible. Those kinds of conclusions can be drawn very easily by looking at these architectural block diagram and some of the data sheets of this.

And in XEP 100 for example, there are several interfaces and modules for enhanced functionality, you have digital IOs, then PWMs, then periodic interrupt timers ADCs. These are now lot of modern microcontrollers, this will be definitely their kind-of interfaces. The digital IOs will be definitely there, PWMs will be definitely there. So, what we are talking about is now this series of microcontrollers, which are in the class of used as some mechatronics application. There is another class of these microcontrollers or microprocessors which are used for mobile platforms.

So, mobile or display applications. So, there are many different series of or families of microprocessors, the companies would develop catering to different different kind of needs of application. So, these are, so, XEP 100 have been designed for automobile controlling applications.

So, it will have the interfaces and other, these modules which are specifically needed for those kinds of applications, motion control applications. So, those we are, we will look at some of the microcontrollers or architectures of only those where we have need for this motion controller mechatronics kind of application. So, this is how these microcontrollers will be there.

And now if you see at 8085 these commands here are going to be of this kind what we saw this MOV command combined for example to move these, contents of this register. These are basic assembly language kind of programs. So, for example, here we have this MOV command that we mentioned here. So, these are more assembly language programs.

So, the microcontrollers like XEP your do not start writing those assembly level commands to do the programming, what you do is, you do these programming in C rather than at assembly level and compiler, typically there will be, the environment will be there, development environment, integrated development environment. So, that environment in this XEP for case we have this core warrior software environment, which will compile the C code and convert it into the assembly language and actually carry out the operation of dumping this assembly language code into memory of microcontroller to execute.

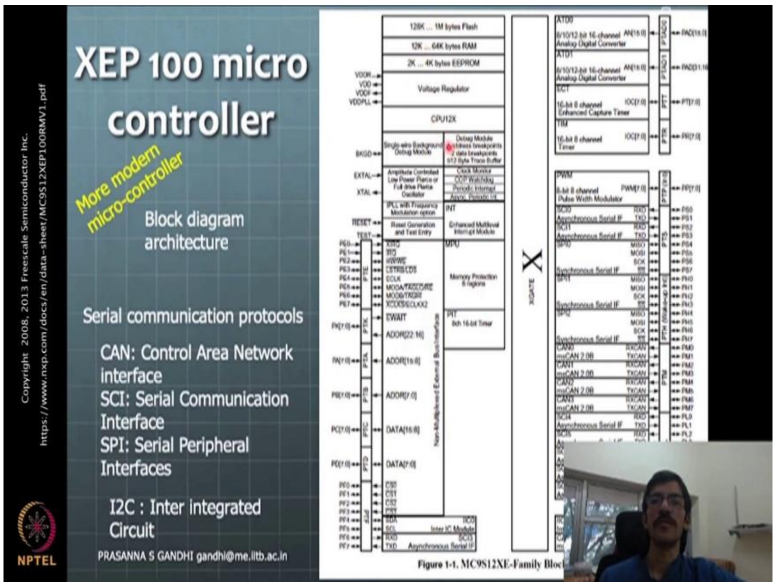
So, these all things are now done at a much higher level. So, a user does not need to know what are the opcodes are these assembly language commands that are used for executing your program. You just need to write your program in, as if you are writing in C and then your job is done.

So, we will get into programming aspects pretty soon, just hold on for a while. But this is important to know, because now the efficacy of what you are using in C program and what gets converted into assembly language lies with this code warrior or whatever Code Composer Studio in Tiva case. And so that compiler is determining although whatever you have written in C, how faithfully it is getting converted into code that microcontroller can understand or execute very easily.

Why I am saying that, is because in C or C++ there are same thing can be done many different ways, and not all of those ways would have been there in the compiler to compile and produce the output for the assembly level program. So, it may not be get compiled nicely depend upon compiler specifications, it may or may not get compiler it may not pop out also major errors, but you will wonder why I am, in C everything looks fine, but they still why it is not executing.

If that kind of situation comes it is probably time to change the logic that you are using in C to some simpler logic if you are using some complex logic, you can change it to some simple or you can try again and see. So, many times there are no unique or straightforward answers to some of these things, while it appears that I am doing everything right in C, but somehow it is not getting executed in microcontroller, that scenario is not really uncommon in microcontroller programming. We will come to the programming aspects some time soon.

(Refer Slide Time: 22:12)



XEP 100 micro controller

More modern micro-controller

Block diagram architecture

Serial communication protocols

CAN: Control Area Network interface

SCI: Serial Communication Interface

SPI: Serial Peripheral Interfaces

I2C : Inter integrated Circuit

PRASANNA S GANDHI gandhi@me.iitb.ac.in

Copyright 2008, 2013 Freescale Semiconductor Inc.
https://www.nxp.com/docs/en/data-sheet/MC9S12XEP100RMV1.pdf

NPTEL

So, this is a, you see this is a block diagram for architecture of the XEP 100 microcontroller and here there is a memory blocks that are written. So, this is a flash memory, RAM, EEPROM, so a lot of different kinds of memories are written. In voltage regulator, there are some pin Outs and pin Ins one can read through the datasheet to get understand more things, but one can see there are these some registers here like PTE, these are all seems input output registers here.

So, that can exchange the data in the digital format without outside environment. Then on this side you will have this some ATD, this is like an analog to digital converters, so some interface for ATD analog to digital conversion. So, like that you will find, see this is a PWM interface here, synchronous serial interfaces, scan interfaces, so these are different kinds of interfaces. So, CAN for example is control area network interface, this is like a serial communication protocol only, but only for short distance communication.

So, these are developed for specific application, if I say for example CAN interface is developed for the automobile kind of application. German, it come from Germany, German manufacturers they introduced this for the first time and they can make some standard, the way you have the serial interface you are at a standard now. Like that you will find lot of these interfaces that are produced provided by this XEP 100 microcontroller.

So, for example, if you want to say, look can I have 20 analog sensors interface with microcontroller, how do you answer this question? Think about this. So, 20 sensors I want to, which are producing analog output, I want to interface with this microcontroller, is it possible? By looking at this diagram you need to look at where you let give this analog input, we will give this analog input to the A to D converters here.

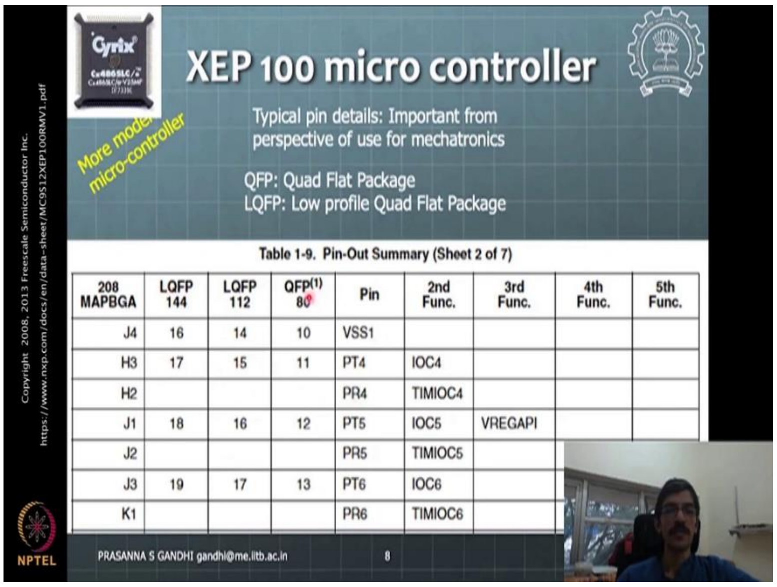
So, this is a 16 channel A to D conversion. So, this 16-bit thing is there and then there is 16 channel another A to D converter is there. So, these two are seems to be same, although they are given some terminology different here, you see the numbers here 16 to 0 channels are there, there are only 16 channels in this microcontroller.

So, what are these and what are these there will be some small difference between these two cases. So, one has to see in little more detail in there I think, but there 0 to 16 channels are there for this analog to digital converter. So, I cannot interface 20 sensors so I cannot use this microcontroller.

Now, those kinds of conclusions one can draw. So, if you get in this kind of confusion, these are seems like 32 channels, but then you need to look at little more details in the datasheet and you will get to know that, there are actually only 16 channels that are in the program. There one can see there are only 8 motors that I can run maximum through this XEP 100. So, there are 8 PWM channels to run my 8 motors.

Like, that one can conclude some of the things based on the architectural problems. And then one can go in more detailed block diagram of each of the interfaces, which I have not included here, I mean to kind of understand how that particular interface was, how, what are the different registers that are controlling that interface outputs. So, those details you will be given in that architectural diagram of each of the interfaces. And that is important to know from programming perspective. So, when we talk about programming, we will deal with this in little more details. So, let us move on from here.

(Refer Slide Time: 27:05)



XEP 100 micro controller

More modern micro-controller

Typical pin details: Important from perspective of use for mechatronics

QFP: Quad Flat Package
LQFP: Low profile Quad Flat Package

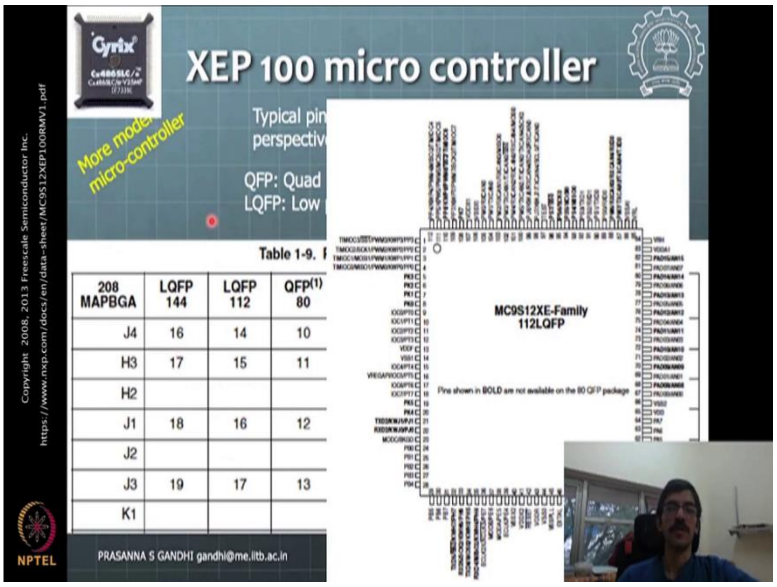
Table 1-9. Pin-Out Summary (Sheet 2 of 7)

208 MAPBGA	LQFP 144	LQFP 112	QFP(1) 80	Pin	2nd Func.	3rd Func.	4th Func.	5th Func.
J4	16	14	10	VSS1				
H3	17	15	11	PT4	IOC4			
H2				PR4	TIMIOC4			
J1	18	16	12	PT5	IOC5	VREGAPI		
J2				PR5	TIMIOC5			
J3	19	17	13	PT6	IOC6			
K1				PR6	TIMIOC6			

Copyright 2008, 2013 Freescale Semiconductor Inc.
https://www.nxp.com/docs/en/data-sheet/MC9S12XEP100RMV1.pdf

NPTEL

PRASANNA S GANDHI gandhi@me.iitb.ac.in 8



XEP 100 micro controller

More modern micro-controller

Typical pin
perspective

QFP: Quad
LQFP: Low

Table 1-9. P

208 MAPBGA	LQFP 144	LQFP 112	QFP(1) 80
J4	16	14	10
H3	17	15	11
H2			
J1	18	16	12
J2			
J3	19	17	13
K1			

Copyright 2008, 2013 Freescale Semiconductor Inc.
https://www.nxp.com/docs/en/data-sheet/MC9S12XEP100RMV1.pdf

NPTEL

PRASANNA S GANDHI gandhi@me.iitb.ac.in

So, this is another important aspect to consider that the pins may have different functioning. So, pin detail should be given, which pin is connected to, this pin number is connected to what, its function will be given, that this pin diagram details will be given in detail of each of the microcontroller. So, XEP 100, for example, so this is a pin VSS, so if you want to find out this, this is a supply voltage, there is no other function for this pin.

But for some of the other pins there are some functions this PT5 and some other functions here. So, how these functions, so the pin functions can change, one can program and change the pin functions. So that is what is important concept here to know, pin is not having a fixed function all the time, that is possibility in the new modern microcontrollers, see older microcontroller is if you go for there will be fixed pin functionality.

So, GPIO pin will just do only digital input output there is no other function that pin can do. But now, these modern microcontrollers will have this possibility of pins having multiple functions, and depending upon the package this is different kinds of packages, this low-profile Quad Flat Package or QFP that those packages are where this, how this chip is packaged, and these pins terminals are coming out is what it says.

So, this LQFP 144 means it has 144 pins that are coming out of this thing. So, this is like 112 LQFP, so 112 pins are there from these numbers from here to here, we will have 112 pins that are given by this package. So, each of the package some package may be missing some pin out, so you need to see okay what is the intention for you to program and then you can select appropriate package for the microcontroller.

So, this is about, little bit more about XEP 100. So, see we are not looking at all the entire information of XEP 100 and then moving on to next, it is impossible to do that, if you see the datasheet of XEP 100 it will be about 1000-page datasheet, you cannot expect yourself to go through the entire datasheet and there then only you are qualified to go for higher level that is not a way to understand my content, it will take humongous amount of time.

And then even then we will not be sure whether we will be able to be skilled enough to do what we want to do out of that microcontroller. So, the strategy or philosophy of using this information is what we need to understand here. So, that is why we want this, your background in the microcontroller domain, any microcontroller domain is okay, I did not say that you should have the syntax of some microcontroller, No, that is not requirement for this course. Because this philosophy of how do you look for what is important and that will come based on some small experience that you had before in MACLAB or your own microcontroller programming experience.

So, that experience is valuable, because it gives you some philosophical understanding of things. So, when you are now switching to new microcontroller, you know what to look for, or what not to look for, both ways like. So, that is where your experience would be valuable here. And we understand some part and then we say this part is not understood completely, but if we need that for some something we will get into more depth of that and come back to that, like that we should understand.

So, if you are able to understand its architectural diagrams well, lot of things will be immediately very clear. So, you can practice to go through some of these diagrams of XEP 100, I will just post these data sheets there or your 8085 of course is very simple, but XEP 100 some of the block diagrams you can go through and understand, oh, look I need to set some registers to get some meaningful operation done out of this microcontroller.

And this, all these, what commands I should look for, I can look at this register details of this microcontroller block diagrams and from there I can figure that out. So, you do not start reading the datasheet as if you are reading a book, No, the way to read datasheet is what do you what is your intention, what do you want to do. Say, I say okay my intention is to just program some visual output or input outputs. So, from that perspective I will look at only that interface which is digital input output interface for this microcontroller.

So, that is how you look for appropriate things in the appropriate microcontroller datasheet. And then some names that will be given for these interfaces also will be different. So, digital IOs, input output interface I say, but if you see in Tiva board it is given as general-purpose input outputs (GPIO) kind of terminology that is used. In XEP 100 I think there is some different terminology that have used, port integration module or some something like that is given PIT or something like that. So, those names also will be different.

So, you need to look for those names, look for your digital input outputs and then appropriately look for that particular interface and then you go into that detail. So, you should be open in some way to see that, this digital input output is I am searching for, but it is not there in this, this kind of a thing may happen. I see the entire content there is nothing written as digital input output and there is nothing written that GPIO in one of them or in microcontroller detachment, impossible, microcontroller should have some kind of a digital input of interface. So, then that, then its name you need to look at that might have been different.

So, this is how some things may differ from one family to other family and we need to hook on with the philosophy very firmly that philosophy is mostly same across multiple platforms and look for things that we want to do. So, we will have this opportunity to, you already have some whatever microcontroller you have gone before through, but we will have this opportunity to go through little more details and actually program this ARM controller Cortex series, M4 series controller.