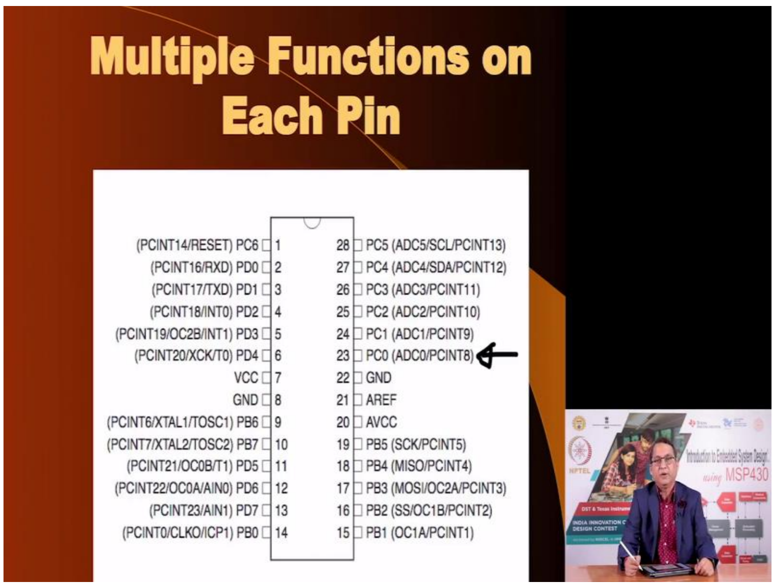
Introduction to Embedded System Design
Dhananjay V. Gadre
Netaji Subhas University of Technology, New Delhi
Lecture 6
Salient Feature of Modern Microcontrollers Continued

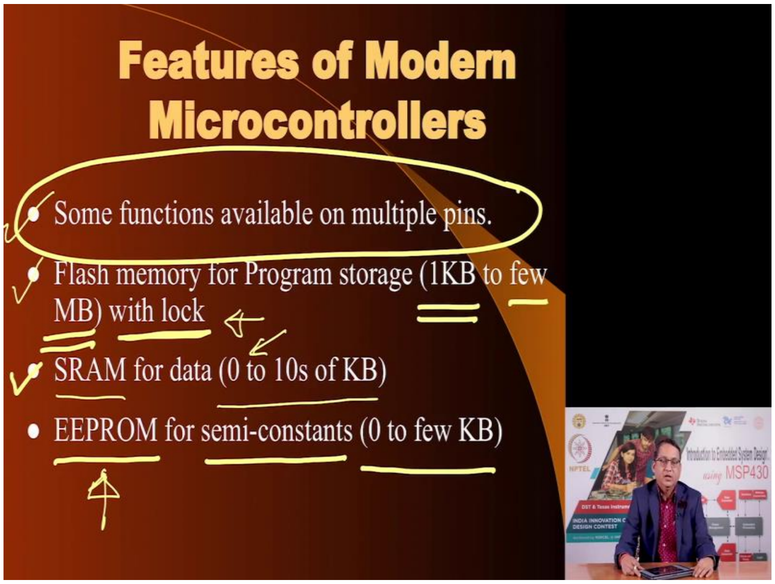
Hello, welcome back to the next part of the session. In the last session we discussed how modern microcontrollers offer multiple functions on any given pin. And this allows us to fit more functions within the limited number of pins available on a microcontroller. And it becomes the responsibility of the programmer that is the system designer to ensure that writing an appropriate program to select whatever functions they need for a given application are appropriately selected. Now, it will also happen that you need 2 functions and unfortunately on a given microcontroller, both those functions are mapped to the same pin.

(Refer Slide Time: 01:10)



I can take an example, let us look at this diagram here, and imagine that you do want to use pin number 23 because it offers you ADC channel 0, it also offers you interrupt on this pin called PCINT8, PCINT8 and you want both these functions at the same time. Now, the way they are mapped on a single pin, it will not be possible.

(Refer Slide Time: 01:50)



And therefore, you may either have to compromise by finding some other pin which has similar functionality, or if you were to use more recent more, you know, advanced microcontrollers you would find that in such microcontrollers given function may be available not only on 1 pin it may be available on multiple pins and so, you can judiciously use a function on a different pin so that you can continue to use the other function on that pin that you wanted to use. And so, this point and I will illustrate it by looking at the datasheet of such a microcontroller.

(Refer Slide Time: 02:25)

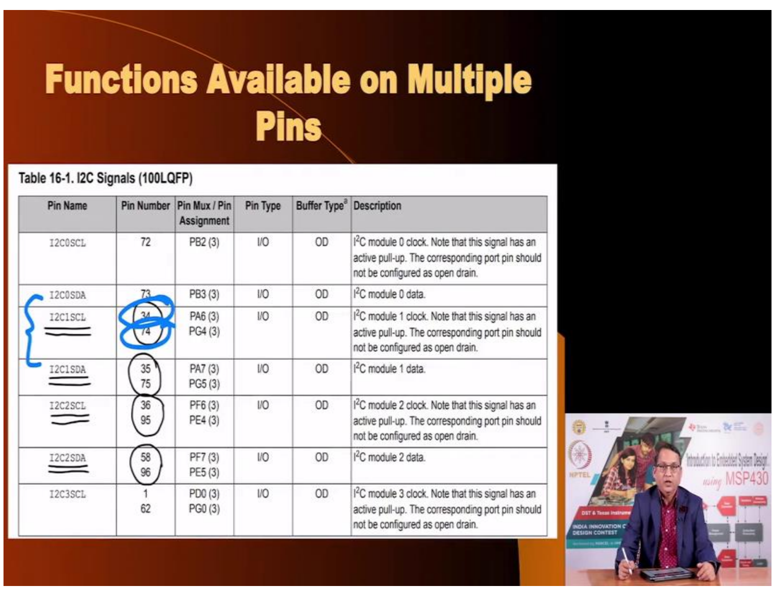


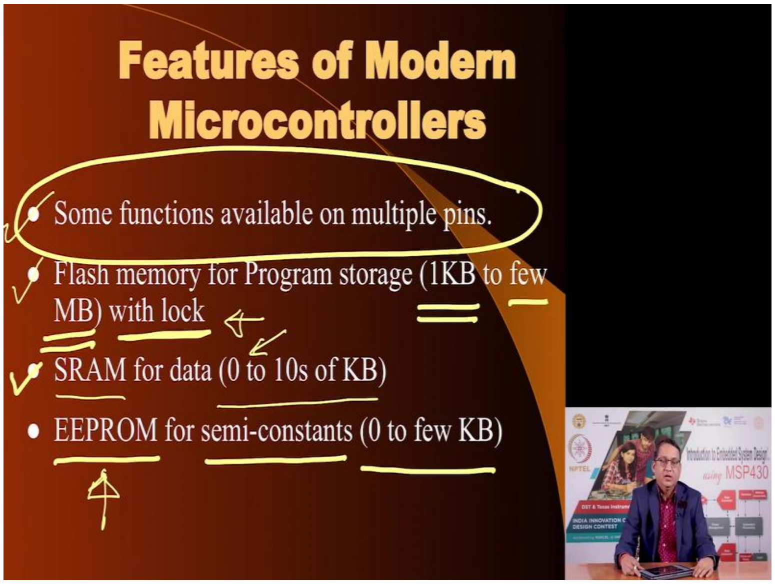
Table 16-1. I2C Signals (100LQFP)

Pin Name	Pin Number	Pin Mux / Pin Assignment	Pin Type	Buffer Type[a]	Description
I2C0SCL	72	PB2 (3)	I/O	OD	I²C module 0 clock. Note that this signal has an active pull-up. The corresponding port pin should not be configured as open drain.
I2C0SDA	73	PB3 (3)	I/O	OD	I²C module 0 data.
I2C1SCL	34 74	PA6 (3) PG4 (3)	I/O	OD	I²C module 1 clock. Note that this signal has an active pull-up. The corresponding port pin should not be configured as open drain.
I2C1SDA	35 75	PA7 (3) PG5 (3)	I/O	OD	I²C module 1 data.
I2C2SCL	36 95	PF6 (3) PE4 (3)	I/O	OD	I²C module 2 clock. Note that this signal has an active pull-up. The corresponding port pin should not be configured as open drain.
I2C2SDA	58 96	PF7 (3) PE5 (3)	I/O	OD	I²C module 2 data.
I2C3SCL	1 62	PD0 (3) PG0 (3)	I/O	OD	I²C module 3 clock. Note that this signal has an active pull-up. The corresponding port pin should not be configured as open drain.

This is the datasheet that I have for you to illustrate this point of how multiple or how functions could be available on multiple pins. Here, if you look at this pin, this function this is I square C module and this is the clock for the I square C module. It is available on 2 pins, pin number 34 and pin number 74 also.

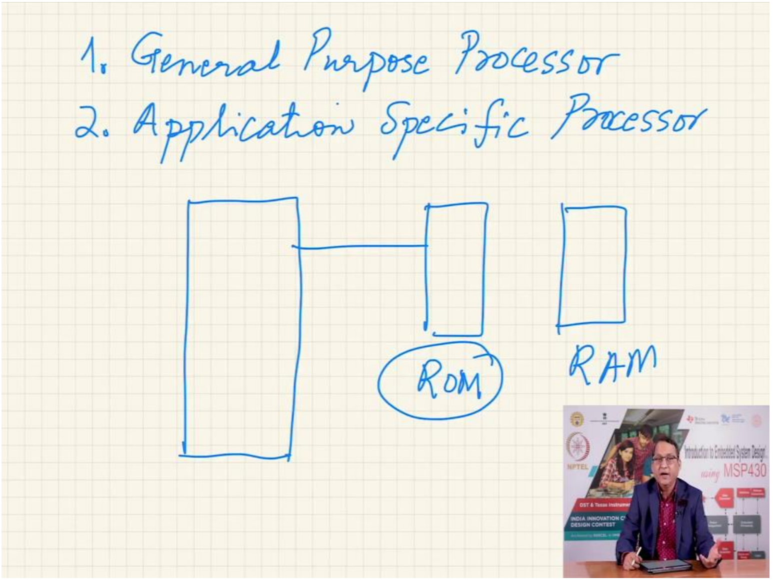
Now, if this pin happens to if on pin 34 happens to have another function you can forego that function and instead use pin number 34 to provide I square C clock, which in the earlier case where a given pin would have 2 or 3 functions and if those functions were not available on other pins, you would be stuck. And so, this is an important point which you could which you may consider while selecting an appropriate microcontroller.

(Refer Slide Time: 03:25)



Then you need a microcontroller to store your program. So, what is the kind of memories is available, so, it turns out that program memory of the in the range of 1 kilobyte to a few megabytes are available on contemporary microcontrollers. Obviously, you will choose a microcontroller with less amount of memory. When your application is small. It would also reduce the cost of your application cost of the microcontroller. And in case you need more memory you would choose an appropriately large amount of memory on a microcontroller that you want to consider.

(Refer Slide Time: 04:30)

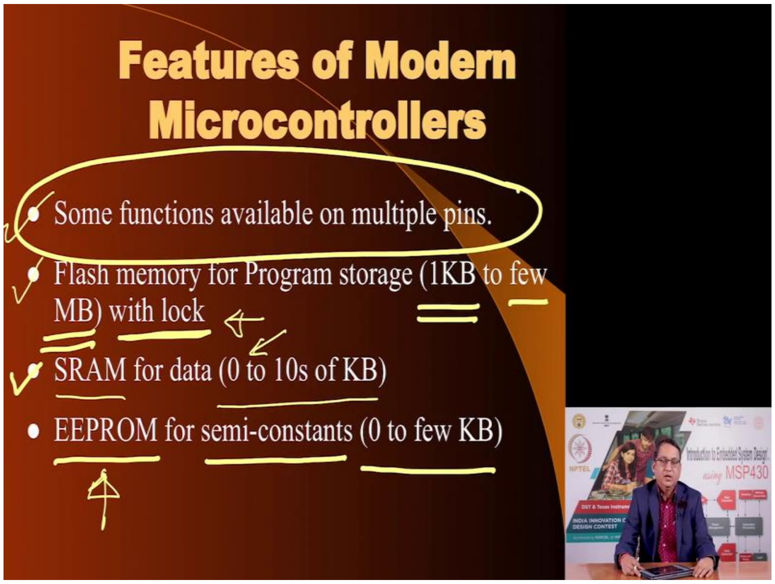


There is another feature that these have and this is called, what is the meaning of lock? Now, let me illustrate this point, especially in the context of the 3 methods of implementing a given application, 3 methods of implementing the embedded computer. And you would remember, you would recall that the first method was the General Purpose Approach. The second was the Application Specific Processor, General Purpose Processor Approach.

Now, in the case of General Purpose Processor, you would have the microprocessor here and you would connect it to a ROM of appropriate variety, maybe E square PROM and you would have RAM. Now, this ROM device consists your program which is the intellectual property of your product. Now, a competitor may find it very convenient to buy your product, remove this ROM from the circuit put it into a device that is commonly called EEPROM programmers.

They are also capable of reading the contents of EEPROM and they can replicate your system by copying the contents of your program. So, if your product is implemented using the General Purpose Processor Approach, it has the potential of being duplicated very easy.

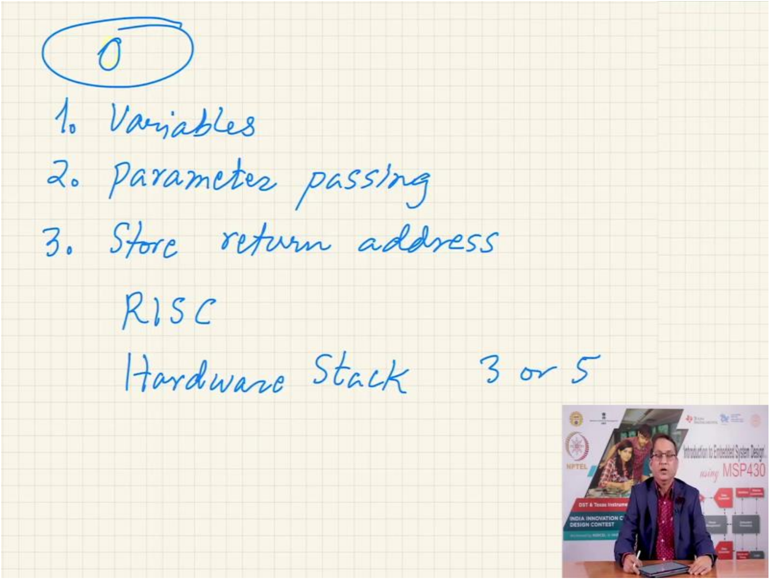
(Refer Slide Time: 06:10)

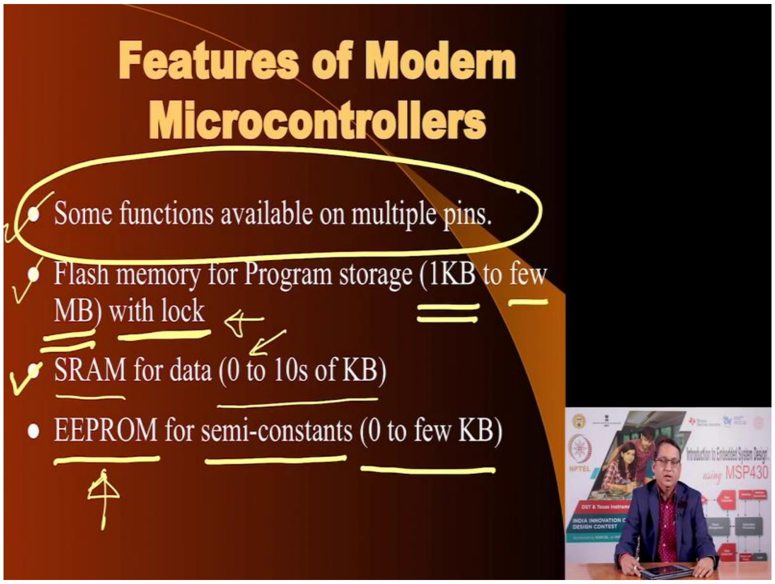


And therefore, a good better method is to use these microcontrollers with integrated program memories, specially which have a lock function, you can invoke this lock once your program development is complete, you can secure the contents of these memories by selecting the lock feature.

And thereafter, nobody will be able to read the contents of this memory by connecting it to the programming pins. And this is a very important feature to consider, especially when you are making a product. Then, these microcontrollers have SRAM which is used as data memory. And the value of these the amount of SRAM available on these microcontrollers ranges from anywhere from 0 kilobytes. That means no memory at all to 10s of kilobytes.

(Refer Slide Time: 07:15)





Now, there is a important question that comes to our mind, that if the amount of memory available on a microcontroller, data memory that is SRAM on the microcontroller is 0, is 0. That is no memory. How are we going to store our variables? 1, because the data memory is used for several purposes, 1 for storing variables, 2 for passing, passing parameters from the main program to the subroutine, and very importantly, to store return address, what is commonly called as a stack.

On the stack, you would store the return address of the program from which you would call a subroutine. So, before going on to the subroutine you would store that address to which you have to come back after the subroutine has been executed. And typically you would use data memory to store that. Now, the big question in front of us is, if we have selected a microcontroller in which there is no RAM, how are we going to perform these 3 functions?

And the answer is very simple. Typically, such microcontrollers which do not offer any RAM would usually be of the RISC category, RISC architecture and in RISC architecture, there is usually large number of internal registers, and therefore, you could use these registers for storing your variables. And you could use the same registers for passing the parameter to a subroutine. Of course, if your application is large that you cannot do with the number of registers available.

Clearly this is not an appropriate microcontroller for this application, you must find a microcontroller which has some or suitable amount of SRAM available, but for very small applications it is possible that you may get away without using the, without using any RAM, It still poses how do you store the return address? In such cases where the microcontroller does not have any SRAM instead of using RAM for the purpose of stack, such microcontroller offer what is called as hardware stack.

What is the hardware stack? Meaning there is a dedicated storage memory specified for the purpose of storing return addresses. Now, because if you have the regular microcontroller with some amount of RAM, the entire RAM could be used for storing return addresses when does the size of the stack grow? When you call a subroutine from within a sub routine from within a sub routine what is called as nested sub routines, when you have too many nested sub routines, they would increase the size of the stack as you return from sub routine, it would vacate the stack and therefore, the stack would stack size would reduce.

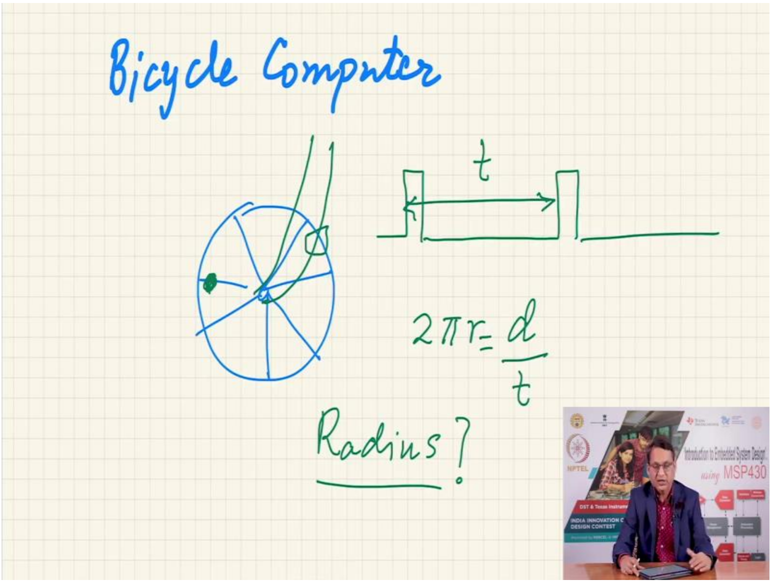
In the case of hardware stack, the storage is limited usually 3 or 5 return addresses and therefore you cannot have too many nested sub routines in such microcontrollers which do not have RAM and instead there is a dedicated hardware stack. In such applications, in such microcontrollers, you cannot involve too many you cannot write too many nested subroutines. So, this is a compromise that you would have to do.

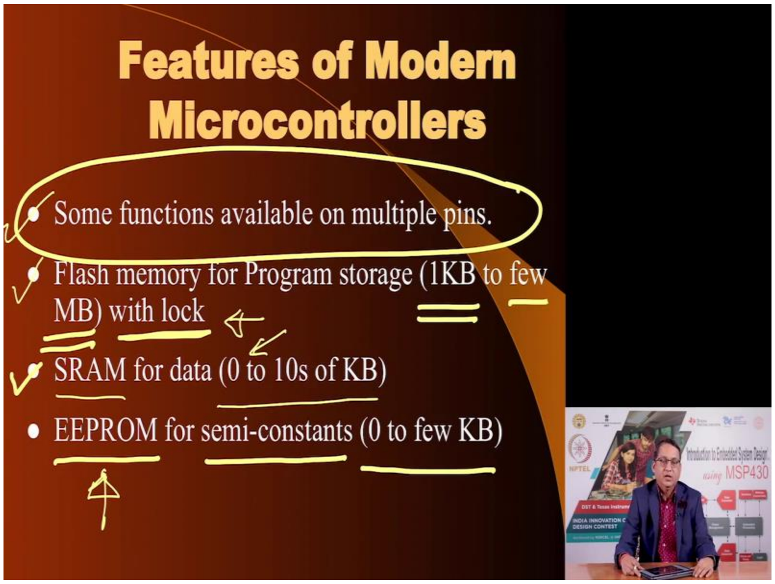
And the compromise is so that you can continue to utilize this inexpensive microcontroller for some cheap applications. But this is an important feature to keep in mind. And that is the meaning of SRAM with 0 amounts of memory. So, we have considered the availability of function on multiple pins, we have considered how much flash memory may be available in various microcontrollers.

We have considered how SRAM, the amount of SRAM available now let us come to another point and that is apart from flash memory, and SRAM. Another type of memory is usually available on microcontrollers and that is the E square PROM, E square PROM stands for Electrically Erasable Programmable Read Only Memory and I use a term which I have coined, I call it semi constants and the amount of E square PROM available could be anywhere from 0 kilobytes which means no E square PROM to a few kilobytes.

What is semi constant? You may not have normally heard this term, I specifically sort of created it to indicate constant a value which is constant most of the time, but once in a while, you may want to change it.

(Refer Slide Time: 12:55)





Can I give an example most certainly, imagine that you are a manufacturer of bicycles and modern bicycles? People like to have a computer on those bicycles, which are called bicycle computer. What does a bicycle computer do? Usually it will tell you the distance traveled the acceleration, the amount of time you took for a given journey.

Now, how would it, how would it measure all these parameters on the wheel of the bicycle, these are the axle and these are the spokes. And then from this you have probably the fork, which is holding the wheel at the center. On the fork on the on the spokes of the bicycle, you attach a small magnet and on the fork, you attach a Hall Effect sensor.

When the wheel rotates, this magnet passes in the vicinity of the Hall Effect sensor and the Hall Effect sensor indicates that by maybe a pulse, and so as the wheel rotates, you will keep on getting pulses the time between 2 pulses is the time that the bicycle took for 1 complete rotation of the wheel. And therefore, if I know this time, I also know that during this time I have traveled the bicycle has traveled 2 pi r, where r is the radius of the bicycle, I have traveled this much distance.

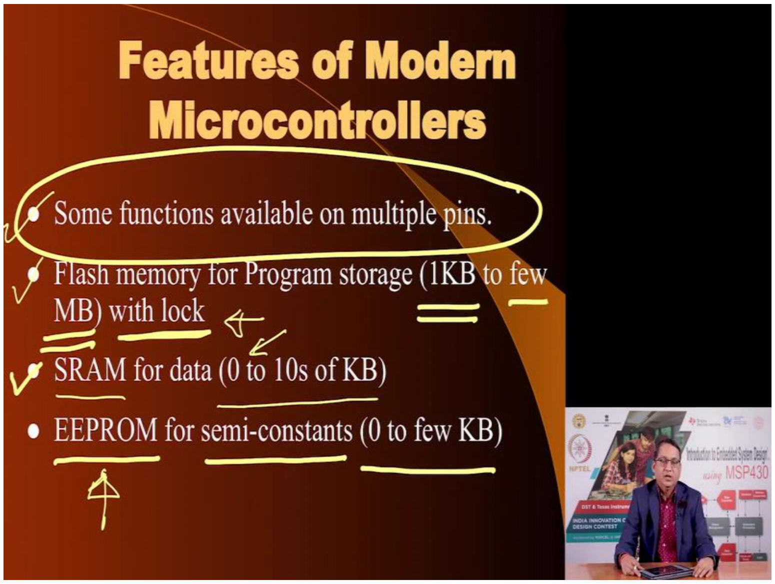
And so if I know the time and I know the distance d is equal to 2 pi r, if I divide it by time I get speed, and so on, and from that, I can calculate acceleration and so on. Now, if a man is typically a manufacturer of bicycles does not manufacture 1 size of bicycles, they manufacture several sizes, and the sizes are classified in terms of the wheel radius. And so would it mean that this manufacturer if he is making if they are making 10 different radius bicycles, that they would have to stop 10 different bicycle computers, that would be very impractical and it would lead to needless stocking of inventory.

What we should have is a mechanism so that the end user or maybe the dealer be just before making the sale of the bicycle, adjust the radius in the bicycle computer to reflect the radius of the bicycle and thereafter, when the user uses that computer on that particular bicycle, it computes the distance and velocities appropriately. Now, the problem is where should the value of the radius be stored?

If the radius is told on the flash memory, that is a program memory, it cannot be altered, if it is stored in the RAM, then every time this bicycle computer is switched off, it would be lost and that is where. So, I call this radius the value of the radius as a semi constant, most of the time it remains constant. But once in a while it may change in the case of this bicycle computer, if this computer is removed from this bicycle and has to be fitted on a bicycle with different radius, then this radius has to be changed.

Otherwise it would remain constant for the rest of the time that it is used on that bicycle. So, this is a semi constant. And where would you store it?

(Refer Slide Time: 16:21)

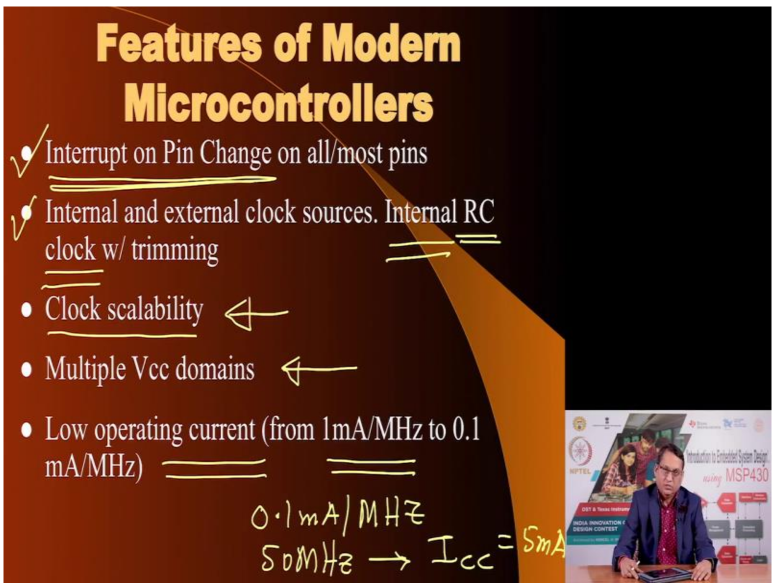


This is where we go back to our slide and say, the place we would store it is an E square PROM, because E square PROM is runtime, you can modify the contents of E square PROM at runtime, meaning as the application is running, you can change the value of the value stored in the E square PROM.

So, in our application, the value of the radius could be stored on an E square PROM. Through the menu, the user can select that they want to change the value of the radius. Maybe on their bicycle computer, there is an up and down switch, and you can adjust the value reflected on the display to reflect the correct radius of the bicycle.

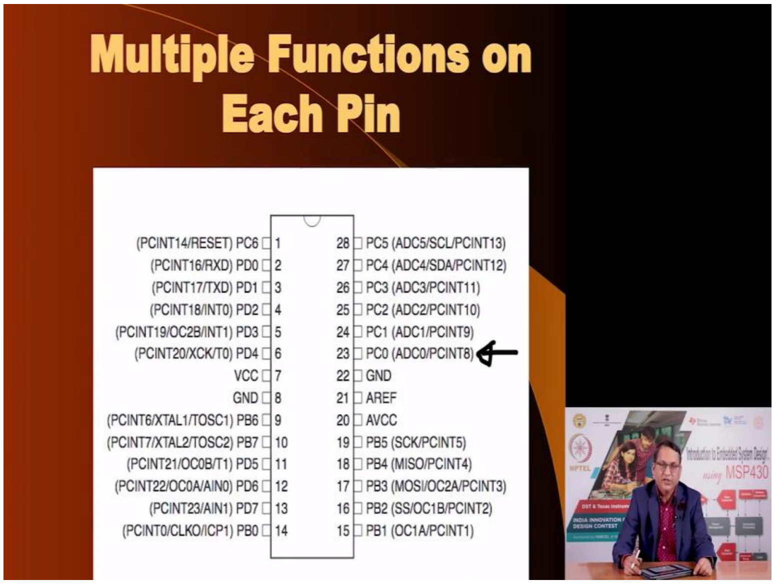
And then you press enter, it will update the value of the radius stored in the E square PROM and thereafter when you use the bicycle computer it would show the correct it will use the correct radius to compute the distance and velocity and so on and so forth. So, this is the meaning and function of E square PROM on modern microcontrollers. Let us see another the next slide.

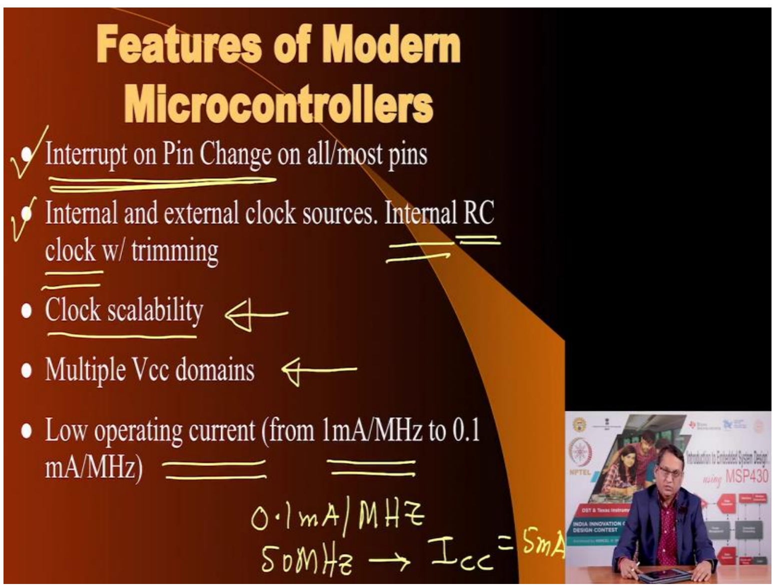
(Refer Slide Time: 17:27)



Now, on modern microcontrollers, you do not have to worry about the limited number of pins with the interrupt capability because now most of these microcontrollers offer almost all the pins with a feature called interrupt on pin change.

(Refer Slide Time: 17:48)



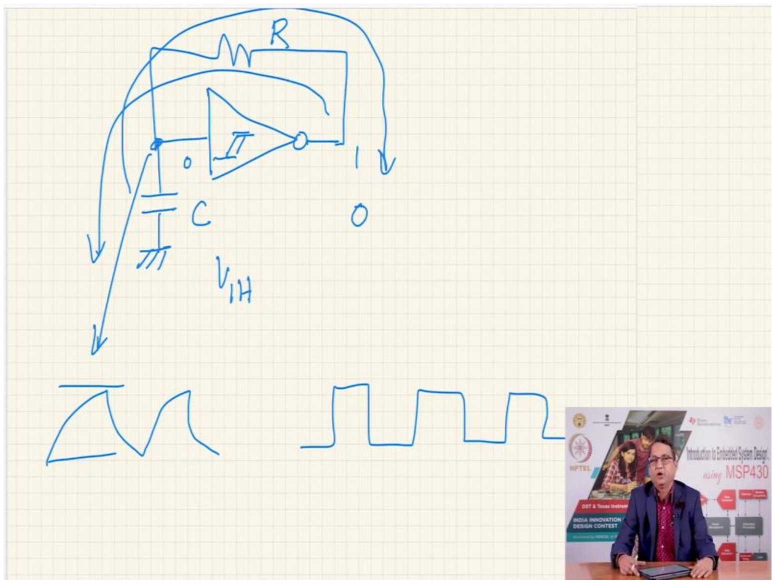


And I could show you the diagram of a microcontroller here. The pin diagram here if you see PCINT is actually pin change interrupt. That is almost every pin has an additional feature called pin change interrupt and therefore You could use these to take care of handling asynchronous events, if there are several asynchronous events, you could use such a microcontroller and such a feature on these microcontrollers. Now you need a clock to operate a microcontroller.

And the reason is that a microcontroller as an example of a logic circuit is of the type what we call as synchronous and it has a sequential synchronous. So, for that you need a clock signal. So, you have to derive the clock signal in a signal in some way. 1 of the methods is that you would use an internal source of clock internal oscillator and you connect an external component to determine the frequency of that clock.

That would increase the cost because that external component would take up 2 pins, maybe 3 pins, maybe 1 pin and it would be an additional cost. If you want to keep the cost down you would like the microcontroller to be able to generate a clock signal internally and such a feature is called internal and external clock sources. And usually the very convenient way of generating internal clock signal is by way of NRC oscillator you can create RC oscillator I can show you a simple example of a RC circuit to generate clock.

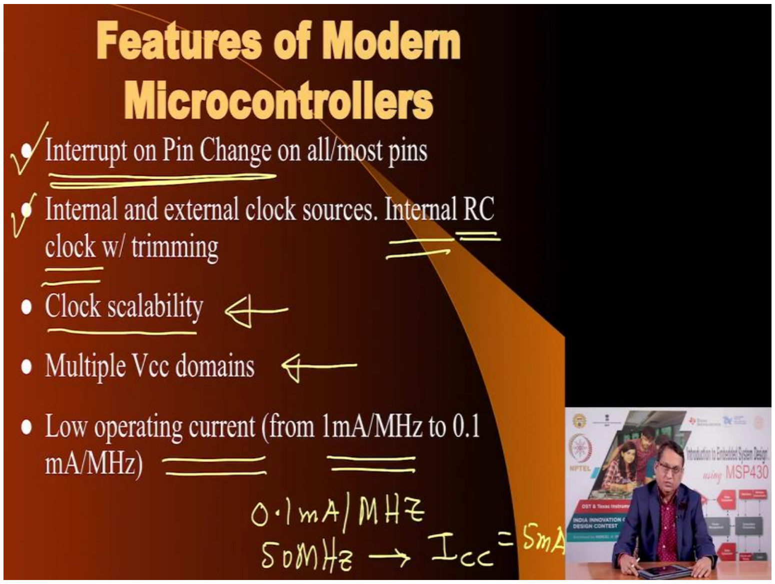
(Refer Slide Time: 19:31)



Here is my NOT gate and I will take a NOT gate of the type of Schmitt inverter, this is a Schmidt inverter, this is input and this is the output I connect capacitor here and I connect a resistor here. And we can consider let us say the input is 0, because this is an inverter this the output becomes 1. It charges the capacitor through this resistor using the time constant R and this capacitor C. As the capacitor gets charged and the voltage on it increases, goes beyond the Voh, Vih, Vih threshold, it turns the output low.

And now the capacitor discharges through this and so it leads to an oscillation if you see the waveform here between 2 thresholds, you would see the waveform on the capacitor going like this and the output would be a rectangular waveform and the frequency will be determined by the RNC and you see you can integrate all these features on the substrate of a microcontroller. The only problem is that the value of R and C are affected by temperature and voltage variations and so the clock frequency may not be very accurate. And so in such applications where not a very accurate clock is good enough.

(Refer Slide Time: 21:04)



You could use the feature of an internal RC oscillator, because it will keep the cost down. In case you want higher precision, then you could use the same internal oscillator but with external frequency determining components such as a quartz crystal to give you higher accuracy clocks. Now, another interesting feature that these microcontrollers offer is what is called as clock scalability. And in fact, MSP 430 offers this feature.

What is the meaning of clock scalability? Meaning that you can change the system clock the internal clock or external clock signal that we are we were just talking about, you could change the frequency of that clock at runtime. Normally, you would fix the frequency and you may not want to change it. But today, because our modern microcontrollers are CMOS based the power dissipation of a CMOS circuit is directly proportional to the frequency of operation and in case the microcontroller application deems it necessary that there is no available job to do.

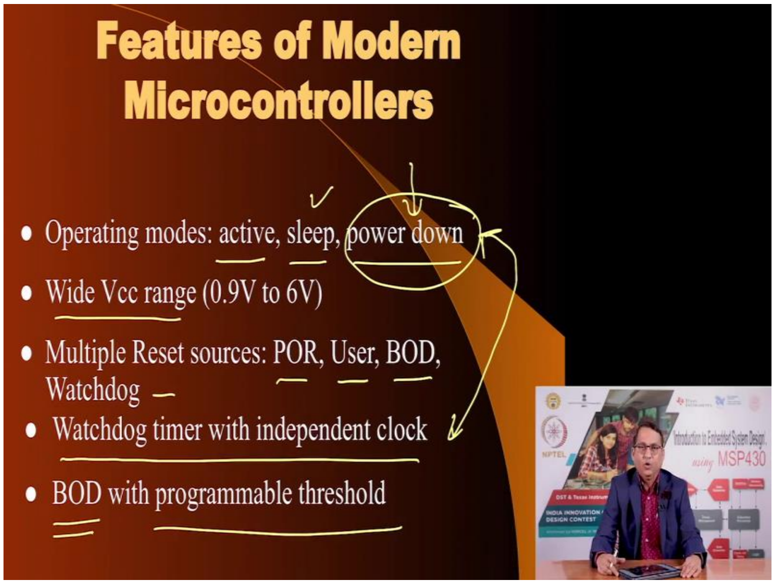
Why to waste power by continuing to clock at a higher frequency during that time it can reduce the participation by scaling down the clock and when some activities expected or some activity becomes available, it can clock increase the clock frequency to a higher value, so, as to be able to perform the program perform the tasks in a timely fashion. So, it has it should have the ability to change the clock frequency at runtime depending upon program logic having this feature will make a more batter efficient system.

And so, you as a designer when you are looking for various features you should see, read the datasheet of that microcontroller to find out whether clock scalability is an option or not. Also, modern microcontrollers operate at high a lot of great high frequencies in which case the power dissipation may increase and to control that, often times the chip is partitioned in multiple voltage domains, and therefore, you will find that your microcontroller requires 2 or 3 different supply voltages.

We will come to that feature when we talk about the ecosystem of a microcontroller. MSP 430 is the ones we are going to use is a simple microcontroller and it requires just 1 voltage supply. But many arm and advanced microcontrollers may require 2 or even 3 supply voltages and it becomes the responsibility of you the designer to provide those voltages. Modern microcontrollers operate at very low currents and typically these currents range in this these currents are in this range roughly 1 million ampere per megahertz to roughly 0.1 milli ampere per megahertz.

So, which means, if your microcontroller offers 0.1 milli ampere per megahertz kind of a feature and if you are going to clock it at 50 megahertz, then you can expect that the total supply current what we will call an ICC will be about 5 milli ampere this you can you will have to provide that the power supply should be capable of providing 5 milli amperes specifically to the microcontroller and this is a very low current. This is a great advancement that has happened.

(Refer Slide Time: 24:56)



Also to continue to optimize the available source of energy, battery modern microcontrollers operate in multiple modes of operation and they are invoked to ensure that batteries are used optimally and these modes are called active mode, sleep mode and power down mode. In active mode is the normal mode is the mode you would like the microcontroller to operate at when you want it to give the best performance, but in case the microcontroller system can find out that it is not being used, it could switch to the sleep mode of operation where the clock signal is cut off to certain functional units inside the microcontroller.

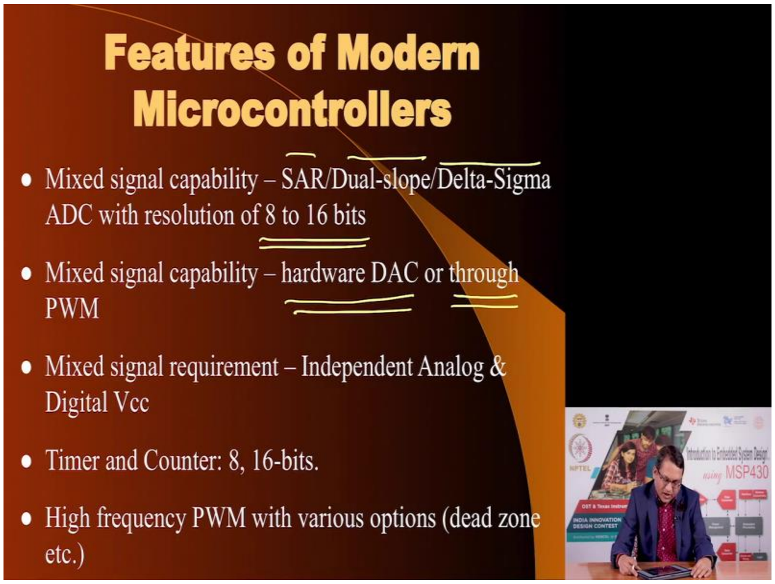
And when the clock is removed from a microcontroller functional function block it reduces the power consumption. So, you invoke sleep function so that power dissipation is reduced and in case there is no activity happening you can even go to what is called as a power down modes of operation. We will see this in the context of the microcontroller features when we discuss them. They operate modern microcontroller operate at using a wide power supply voltage ranges some between the range of 0.9 volts to 6 volts.

They offer or they have several sources of interrupts where they are called power on reset, user reset, brown out detector reset and watchdog reset and we will again consider these in detail when we look at the ecosystem of a microcontroller in detail. Watchdog timer is if it is available would be available with an independent clock. And the reason is when you consider it with this feature that you your microcontroller can operate in various modes of operation in case the microcontroller goes in power down mode, it would cut off the clock for the watchdog timer making it ineffective, and therefore you do not want that to happen.

So, usually watchdog timers come with an independent clock that will not be shut down in case you choose the power down mode. Also, the brown out detector reset has a programmable threshold brown out detector means whenever there is a low going voltage spike on the power supply, this low going voltage spike may potentially corrupt important registers inside the microcontroller.

And 1 way to ensure that the corrupted values do not doom the system, it does not block the system, 1 way to counter that is to reset the system completely. And you can set the threshold of voltage at which this feature this reset will take in that is the meaning of programmable threshold up on BOD reset sources.

(Refer Slide Time: 27:54)



Then, our model microcontrollers are great examples of what we call this mixed signal circuits. Mixed signals means it is not purely digital. It is not purely analog, but it is primarily a digital circuit with analog capabilities and therefore, it is classified as a mixed signal circuit. And for the mixed signal capability 1 important feature is that it should have an integrated analog to digital converter.

Analog to digital converters are of various types and what we have is successive approximation type of ADC or dual slope type of ADC or delta sigma type of ADC and the resolution of these

ADC range between 8 to 16 bits and therefore, when you are selecting a microcontroller, specifically from the point of view of analog performance, you should read the details in the datasheet to find out what kind of ADC it offers.

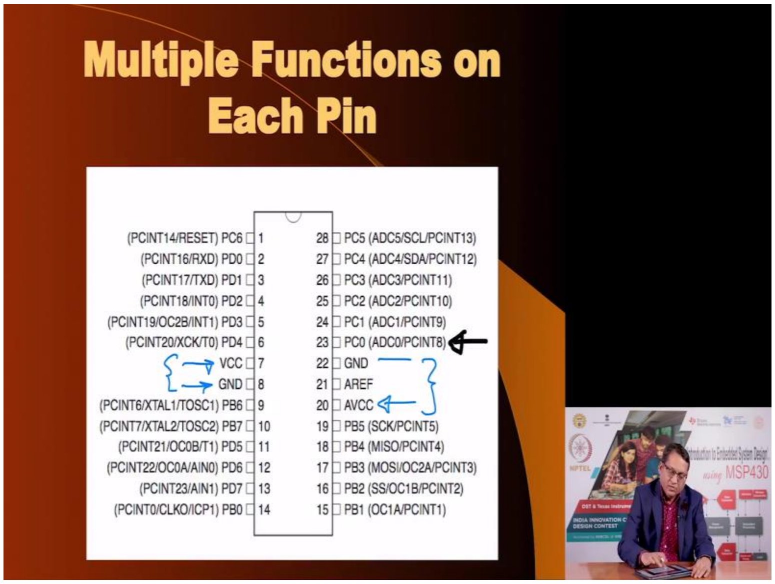
What is the resolution of this ADC and also the speed of this ADC because speed is heavily dependent on the architecture of the ADC meaning is it successive approximation or a dual slope or a delta sigma, dual slope ADCs are not that fast if you want high speed of conversion higher rate of conversion, then you may have to choose SCR successive approximation type of ADC or delta sigma type of ADC and this information is available in the datasheet.

We will consider this in detail, when we are looking at the way a microcontroller samples external signals in that context we will look at this feature again. Therefore, resolution and speed of ADC is an important consideration important feature to for you to verify before you select an appropriate microcontroller for your application. As part of the mix signal capability. A microcontroller may offer hardware DAC meaning it has an integrated digital to analog converter.

You simply load a number into it and the DAC will convert it into a corresponding proportional analog voltage. Or if it does not have a dedicated DAC, it would offer you that feature through what is called as pulse width modulation, we will go in detail of how to get DC voltages or analog voltages out of a pulse width modulated digital signal. As part of the mixed signal requirement, you would see that such microcontrollers have multiple power supply pins dedicated to supply power to the digital part of the circuit and some pins dedicated to supply the voltage to the analog part of the pins.

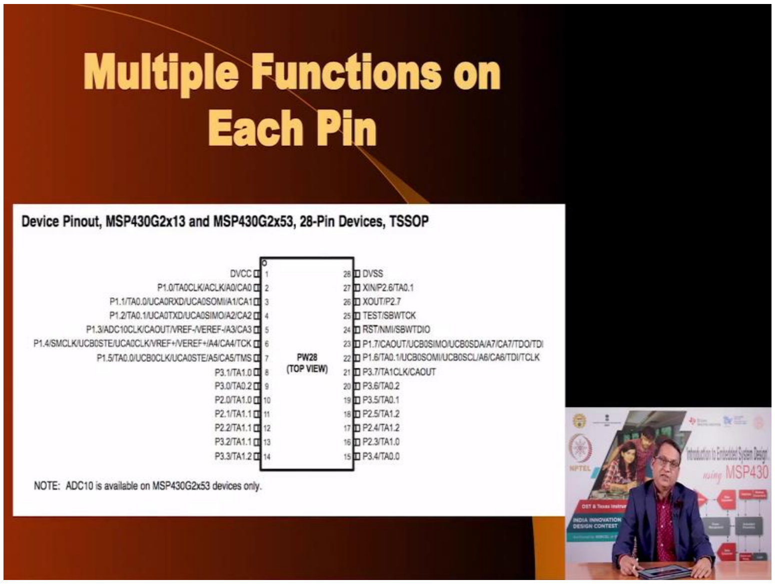
This is so that the digital power supply pins often the voltage there gets corrupted because of this switching action insider logic circuit and this may leak into the analog circuit. So, best way is to have any independent power supply pins. So, you have to be careful when you have multiple independent power supply pins to ensure that those power supplies are fed correctly and that the noise on logic the supply voltage for logic circuit does not leak into the power supply pins of analog circuits on those microcontrollers.

(Refer Slide Time: 31:32)



Here we can see go back and look at the pins of this AVR microcontroller. And you notice that it has a AVCC and VCC. This VCC means it is for the logic part of the circuit inside the microcontroller and AVCC means it is for the analog VCC specifically the ADC and the voltage reference source. And it also has 2 grounds although it does not explicitly mentioned this ground is paired with this VCC here, digital VCC and this ground is paired with the analog VCC.

(Refer Slide Time: 32:09)

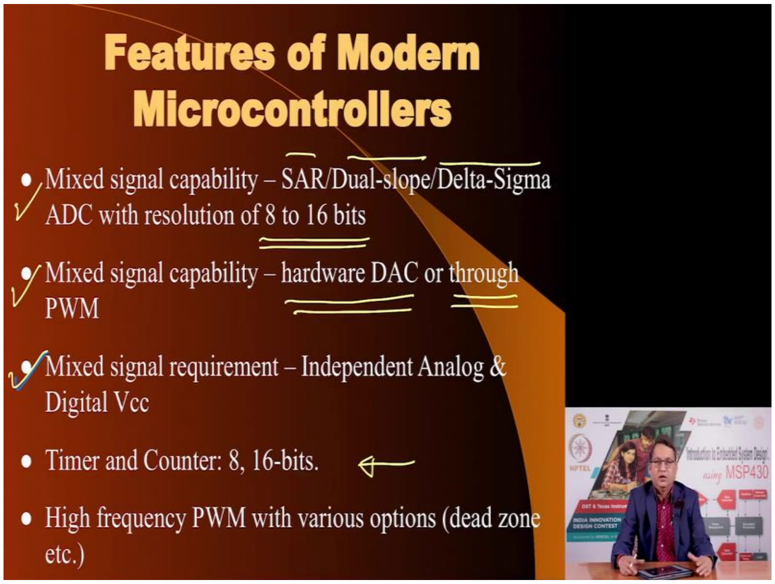


It turns out that on our MSP 430 there is no separate power supply pins for analog and digital in digital part of the circuits even though MSP 430 is also a mixed signal type of circuit, it has

ADCs but they have not deemed it necessary to separate the power supply pins. So, this is an important point to keep in mind when you are designing a circuit system using a microcontroller whether it has individual pins or independent pins.

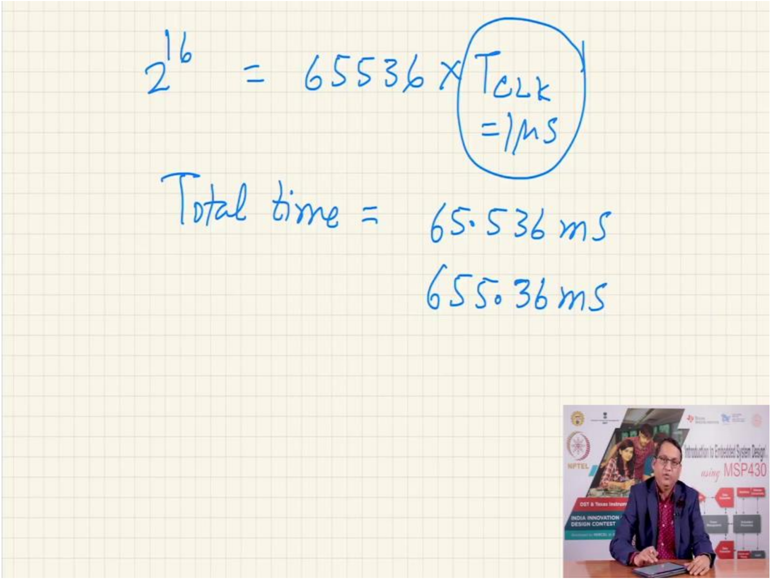
I mean, does it have a common pin or does it have independent pins for power supply for VCC for the analog part of the circuit and the digital part of the circuit.

(Refer Slide Time: 32:58)



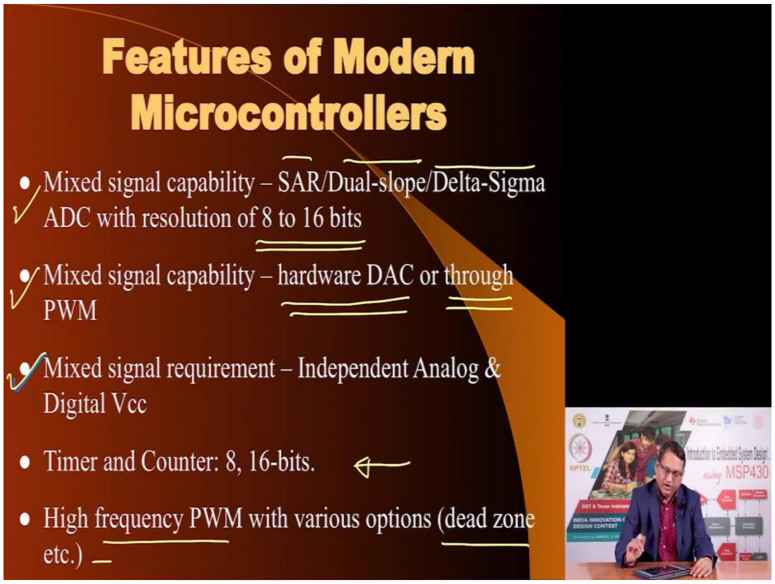
Then for measuring time and other events, the mod microcontrollers would offer you a timer as well as a counter and the resolution of these timers and counters range between 8 and 16 bits, which means it can count 2 raised to power 8, which is about 256 events or it can count 2 raised to power 16 events which is about 65,000 or so, and if you can have a clock signal for such 8 or 16 bit counters, it would be able to measure that much time that is that me explain.

(Refer Slide Time: 33:41)



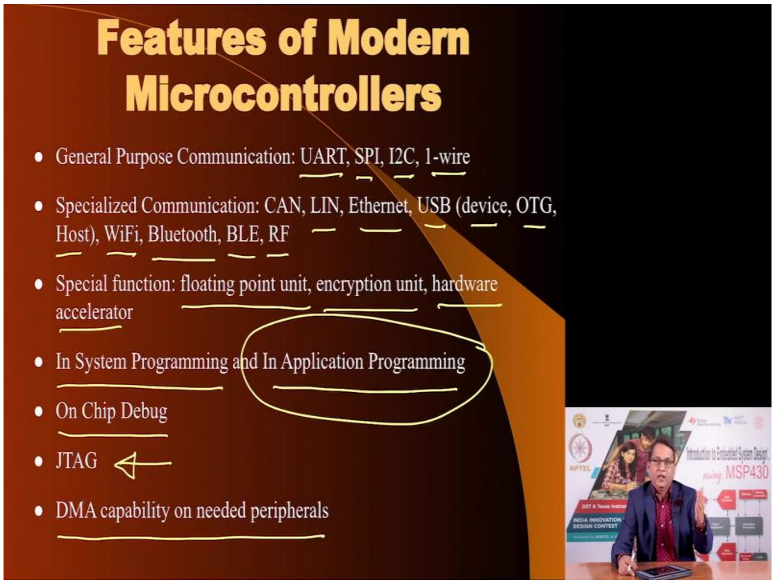
If I have a timer with 16 bits that means, it can count 2 raised to power 16 clock pulses, which is equal to 65536 clock pulses and if the clock period is TCLK, if it is a 1 microsecond, then the total time that the timer will be to measure for you is 65.536 milliseconds. If you want to measure more time, 1 way would be to reduce increase this period, which means reduce the frequency. And you if you divide it by 10 you will get 10 microseconds and then you can measure up to 655 milliseconds and so on. So, you can trade off the resolution with the dynamic range that is the maximum time that you can measure and that choice would be available to you what do you want to have.

(Refer Slide Time: 34:45)



It also offers high frequency pulse width modulation signals with various options. 1 of them is called dead zone. Dead zone allows you some rest time between 2 signals so that some specialized devices require it you should read the datasheet of your microcontroller to find out what are these options available to you.

(Refer Slide Time: 35:08)



Modern microcontrollers may need to communicate with the outside world through in your application and so, they offer general purpose communication protocols such as UART, SPI, Serial Parallel Interface, Serial Peripheral Interface, I square C which is inter IC communication and something called a 1 wire protocol. It may also offer some of them may also offer specialized communication protocols such as CAN Control Area Network, Lin, Ethernet, USB and if it is USB you have to find out whether what type of USB option is available.

Some of them are called device USB. There are other varieties called OTG like you have on your phone, mobile phone, smartphone, and on your desktop computer, the USB type is what is called as host USB. So, you must find out what is your application when you are when you want USB functionality, you want to know where what do you want to connect through this USB so that you can select the appropriate type of USB function.

Then it may offer you Wi-Fi 33, built in Wi-Fi, it may offer you regular Bluetooth or low energy Bluetooth and other RF communication protocols. Some microcontrollers may even offer specialized functions such as floating point unit or floating point ALU will perform floating point operations quite quickly. It may or they may also offer encryption units to encrypt the data before you save them.

And these are called hardware accelerators they may there may be hardware accelerator for other activities. Most modern microcontrollers will offer in System Programming. And some of them may also offer in application programming, our MSP 430 that we are going to use offers both these modes and in fact, our kit operates through the in application programming mode, we will be able to upload program while the system is still on.

And so, this is 1 of the modes of programming the microcontrollers. Modern microcontrollers also offer on chip debug capability which means while you have built your system you would if your system is not working properly you can invoke the on chip debug function to find out what is wrong by in inspecting the internal registers and memory locations.

Modern microcontrollers may offer JTAG which is a Joint Test Action Group protocol which allows you to invoke the on chip debug function and many high end microcontrollers may offer maybe using the DMA capability to speed up the communication between memory speed up the rate of transfer of data between memory and ports.

(Refer Slide Time: 38:23)

For developing applications, you need tools. And these are the application development tools, you need a development platform usually these days PC, whether a desktop or a laptop is used for application development, it must be able to run an assembler and a compiler. It should have, you should have simulator and debugger. You should also have evaluation boards so that you can test your small programs.

And you should have a mechanism to download code from this platform into the memory of your target microcontroller. And oftentimes, you may be required to emulate or debug your code. So, you should have an on chip debugger facility to access the on chip debugger through the JTAG interface.

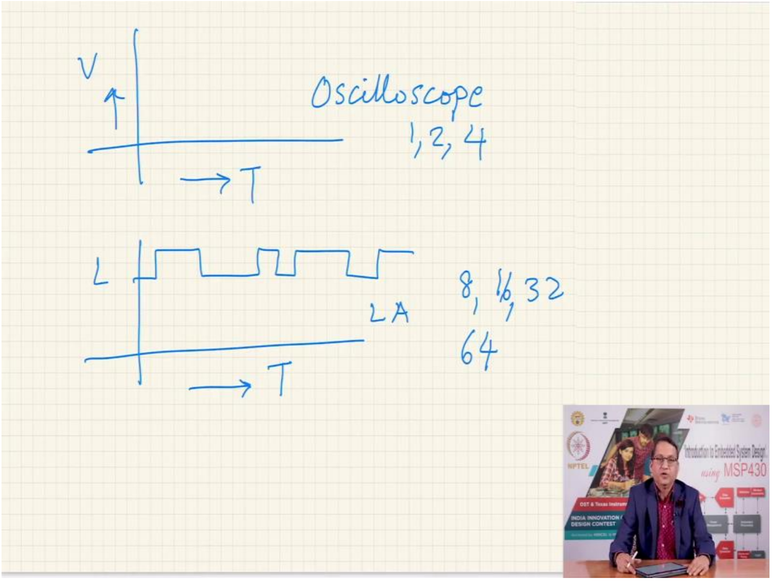
(Refer Slide Time: 38:57)

Apart from the application development tools, you may also be required some system development tools and for that, once you have finalized your requirements, you may want to create a circuit board. So, you need some circuit prototyping facility. You may also need schematic capture and PCB layout tools.

We, I mentioned that in the first lecture that it would be nice if you could download and use EagleCAD which is a PCB layout software, you should have circuit soldering facility maybe a soldering iron for small jobs and increasingly as the sizes of components are reducing, you may want a pick and place machine as well as a soldering oven for automatic soldering of these components.

You may need a microscope or a magnifier to magnify and inspect your fabricated circuit board to ensure that there are no shorts or no tracks are open. For electrical observations, you may need an oscilloscope or logic analyzer which is nothing but like an oscilloscope except you are able to look at multiple channel as a function of time.

(Refer Slide Time: 40:18)



So, an oscilloscope does this and oscilloscope allows you to look at time versus voltage plot of any waveform. This is your oscilloscope, on the other hand a logic analyzer allows you inspection as a function of time but of logic levels. So, you can look at signal and it will show you on this waveform like this is the way it. And unlike oscilloscope which usually is 1, 2, or 4 channels a logic analyzer this is oscilloscope and this is logic analyzer will give you 8 or 16 or 32 or 64 simultaneous signal observations capability.

(Refer Slide Time: 41:19)

So, logic analyzer is a very useful device to have. But increasingly because my microcontroller system is a mix system I need new instrument called Mixed Signal Oscilloscope. A Mixed Signal Oscilloscope is a oscilloscope which is a combination of normal oscilloscope and normal logic analyzer function integrated in a single instrument, it is called Mixed Signal Oscilloscope apart from that you would need a digital multimeter, LCR bridge is good tool to have, good instrument to have in your lab.

And variable bench power supply so that you can use that to power your system while testing. These are the system development tools. Now, we come to the end of this lecture. In the next lecture we will talk about, we will discuss various elements of microcontroller ecosystem, how to make sense of these elements and how to use them in our application with the ultimate goal of creating a successful embedded system design. Thank you very much, I will see you soon.