Introduction to Embedded System Design

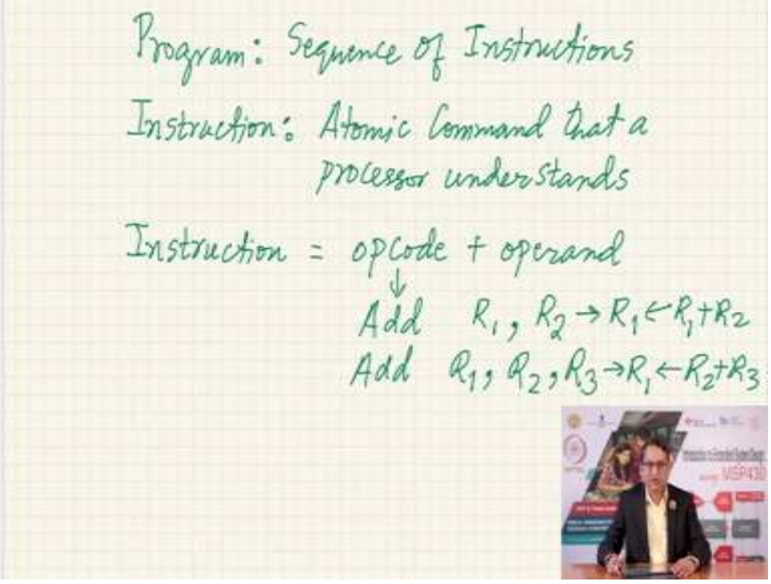
Professor Dhananjay V. Gadre
Electronics and Communication Engineering
Netaji Subhas University of Technology
Badri Subudhi
Electrical Engineering Department
Indian Institute of Technology, Jammu

Lecture 04

Modular Approach to Embedded System Design (Continued)

Welcome back. So we are going to resume our discussions and we are going to consider another classification of microcontrollers or microprocessor. And this one is called instruction set architecture. As the term suggests the classification, the types of microprocessors are often classified in the way the instructions are designed for a given microprocessor. Now to understand that let us first of all describe what are instructions and all that.

(Refer Slide Time: 01:04)



Program: Sequence of Instructions

Instruction: Atomic Command that a processor understands

Instruction = opcode + operand

↓

Add $R_1, R_2 \rightarrow R_1 \leftarrow R_1+R_2$

Add $R_1, R_2, R_3 \rightarrow R_1 \leftarrow R_2+R_3$

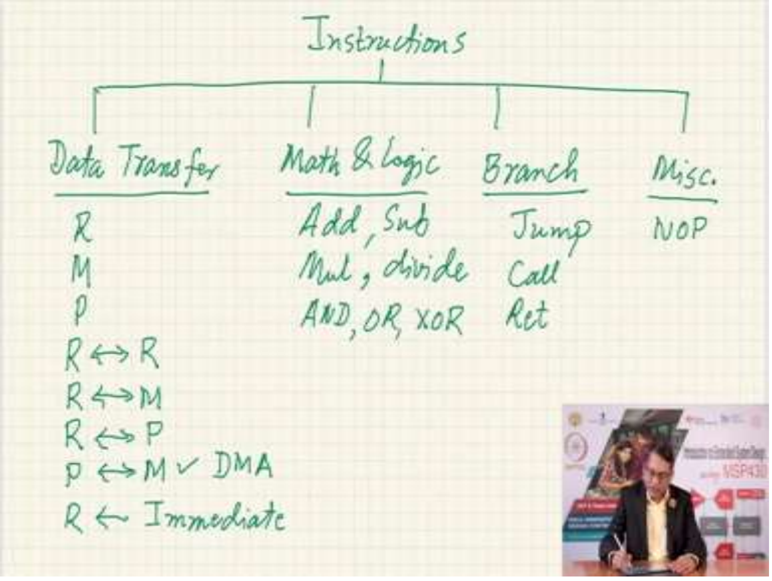
Now, what are you going to write in the memory of the microprocessor is a program. And what is a program? A program is a sequence of instructions. Often times when I asked this the meaning of a program people mentions it is a program is a set of instructions. That is wrong because set can be rearranged and it would not be the same program. So the correct definition is a program is a sequence of instructions.

Therefore, what is an instruction? An instruction is the lowest level atomic command that a processor understands. What does an instruction consist of? It consists of 2 things. The first part of the instruction is called the op-code. This indicates the basic operation that the instruction is going to perform. And the second part is the operand meaning what is it going to work on.

For example, an op-code could be to perform an addition and so the op-code refers to add. Operand is what is it going to add? Is it going to add 2 numbers? How are the 2 numbers specified? Are the 2 numbers part of are stored in 2 registers or the format could be Add R1, R2, R3. If it is of the first type, it means that R1 gets the value of R1 plus R2. And in the second type, it could mean that R1 gets the value of R2 plus R3.

Whether it is of the first type or second type would have a lot of bearing on the size of the instruction. But needless to say, the operation to be performed is described as the op-code. And the operation to be performed on what numbers is the operand part of the instruction. And this is where things can take large variety lead to different types of instruction set architecture. And before we go to that, let me also classify the instructions.

(Refer Slide Time: 04:01)



The types of instructions broadly can be classified in 4 types of instructions. The first instruction refers to what is called as Data Transfer Instructions. As the name suggests, these instructions deal with moving data from one storage location to another storage location in a microprocessor or in a computer system.

What are these locations? These locations are, could be registers. They could be memory and they could be ports. And so, you would want to move data between these entities. So instructions that move data between two registers would be available. Instructions that move data between registers and memory location would also be available.

Instructions that move data between the register and port would also be available. And if we look at other combinations, you may expect that instructions to move data between memory location and a port may also be available. But if you consider this act, this requirement. Memory is outside the CPU; the port is outside the CPU. Why would you want to involve a microprocessor or the CPU in this transfer?

It is only going to slow it down because the microprocessor is not adding any value to this activity. It will have to do this transfer in 2 steps. One it will have to first read the content of the port and then read the content of the port into a register and from the register it may move the content to a memory location.

And therefore, this is a slow process and usually there are no instructions to perform this operation. But this is a very desirable operation and therefore, most of the time this is available as a hardware feature of the processor and it is called Direct Memory Access. Apart from moving data between these entities, sometimes you would like to initialize the register with the immediate value or a literal value.

For example, you want to initialize a register with some known constant or something like that. So, you would have instructions to support this kind of transfer. Then the other type of instruction would be for what is called as Math and Logic. And this would include instructions such as add, subtract, maybe multiply, divide, bit operations such as AND. So, you are going to AND the bits of two operands or you may want to OR or other similar logical operations.

Then the third type of instructions are what are called as branch. Branch instructions are those instructions which change the sequence of instructions being executed. Normally, the microprocessor or the microcontroller fetches the contents of a memory location. After it executes it, completes it, it goes to the next sequential memory location and so on and so forth.

But once in a while, you may want to alter this sequence. You may want to jump to an arbitrary location. Why? Because maybe at that arbitrary location you have stored a subroutine that you would like to execute. So, the instructions of such nature where the control of the microprocessor is altered from the current location to a unspecified or a different location which is not necessarily sequential such instructions are called branch instructions.

What are the types of instructions? So, you would like to jump to a arbitrary location, a particular location. You may want to jump it without any condition or you may want to involve a condition. Maybe there are things like flags in the microprocessor; so you would like to jump if the if a certain flag is set or you may want to jump to a location if this flag is reset.

Similarly, you may want to call a subroutine. So, that instruction is for call. Again, call could be unconditional or it could be based on certain flags. Similarly, for call corresponding to call once you have to jump to that subroutine address, after you have finished executing that subroutine, you would want to resume and come back to the main program from where you had called this subroutine.

So, you would have instructions for return. And again, return instructions could be unconditional or they could also be conditional. Apart from this any other instruction that you may have would be in this fourth category and I call them miscellaneous instructions. And one very common instruction that is available across all microprocessor platforms is called NOP that is no operation.

It is usually to if I want to use the word waste time or to spend time. There could be an instruction to change the operating mode of the processor. Maybe you want to switch to a low operating mode or it may want to modify certain flags for say, interrupts it may want to enable disable interrupts.

So, instructions of such nature as you see they do not fall in the first 3 categories would be classified under the miscellaneous categories. Now, these instructions irrespective of the processor architecture, these are the 4 types of basic instructions that any processor has.

(Refer Slide Time: 10:24)

Addressing Modes

opcode + operand

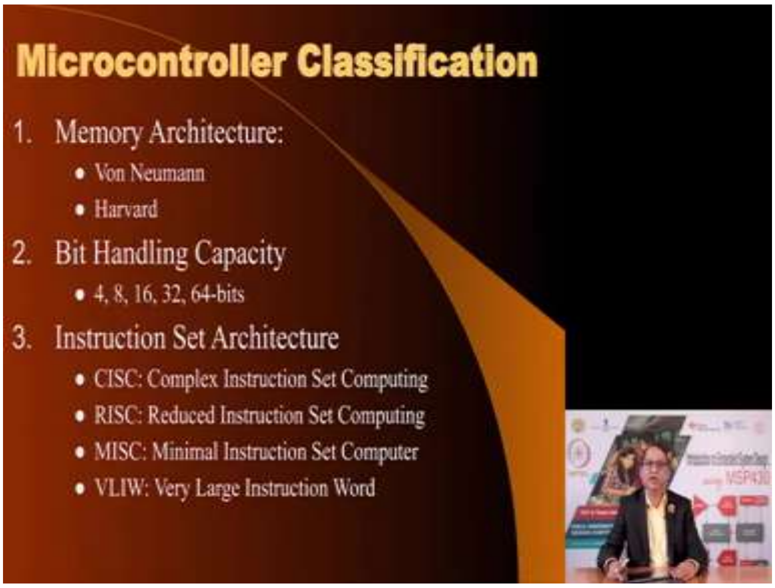
Now, another concept that is built into these instructions is called addressing modes. We are going to talk in more detail about addressing modes but let me use this opportunity to give you a brief idea because it is required to explain the concept of instruction set architecture. Addressing mode refers to the way in which foreign instruction which has a given op-code.

How is the operand specified? Meaning if I want to add 2 numbers, those 2 numbers could be available in 2 registers or 2 numbers could be such that one number is available in the register and the other number is in the memory location. Or one number could be in the register and the other number is immediate number. So, these modify or these offer more variety in the basic instruction like add.

Similarly, you want to fetch, you want to transfer data between two entities. Do you want to specify the address of the location? How do you specify the address of that location? Is it in another register? Or is it in a memory location? Are you specifying the address of that memory location completely as part of the instructions or you are saying no use the contents of another register to go into the memory based on the contents of; so, you are using a register as a pointer or you are adding an offset to that pointer.

To the basic data transfer instruction, you can create more varieties by the help of this concept called addressing modes. And because of which the total number of instructions that are available in a given architecture microprocessor can increase or decrease. And traditionally, the number of instructions that were available and now the number of instructions means not just the basic operations but the variety is available with each basic operation.

(Refer Slide Time: 12:26)



Microcontroller Classification

1. Memory Architecture:
   - Von Neumann
   - Harvard
2. Bit Handling Capacity
   - 4, 8, 16, 32, 64-bits
3. Instruction Set Architecture
   - CISC: Complex Instruction Set Computing
   - RISC: Reduced Instruction Set Computing
   - MISC: Minimal Instruction Set Computer
   - VLIW: Very Large Instruction Word

If you keep on adding more and more variety, it would be called it would be classified as a complex instruction set computing architecture. Earlier architectures were of this category, of this type CISC. And then people figured that large number of instructions do not necessarily improve the performance of the processor.

It is the same thing as there may be thousands of words in the English language but most of the time you only use a very small subset of those words to convey whatever you want to say most of the time and that argument led to the development of a new architecture. And this was called Reduced Instruction Set Architecture or Reduced Instruction Set Computing.

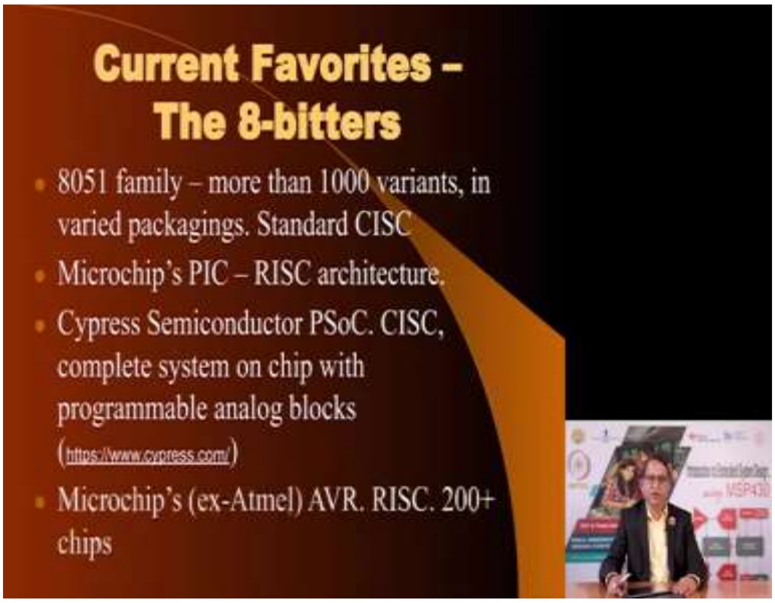
And in this architecture the number of varieties available for any given basic type of instruction was severely curtailed. And it is not that processor designers can arbitrarily choose to have more variety or less variety. It has to be supported by the architecture, the hardware of the microprocessor, in terms of the available registers and the way memory is accessed and so on. But, to put on record, this is another popular architecture.

The third architecture was when this Reduced Instruction Set Architecture came into being, the designers of CISC type of processors started borrowing ideas from RISC architecture and vice versa. RISC architecture developers started borrowing ideas from CISC architecture. And that probably angered certain purists and they said, no, we do not want to increase the number of instructions and they brought out a new architecture which is called MISC: Minimal Instruction Set Computer.

Unfortunately the number of actual devices available in this family probably one or two only, this idea died a quiet death and one largely only hears about RISC or CISC. But there is one more architecture and that is called Very Large Instruction Word meaning that in a single instruction word, you have several instructions so that it can improve performance by parallelizing the execution. So, these are common instruction set architectures.

Having gone through this, it would be now imperative to see what is available in the market. Since we are looking at microcontrollers, we must know what are commonly available microcontrollers. And based on these previous knowledge we can see whether a particular microcontroller is whether it is of CISC type or RISC type and so on so forth. So, I am going to class list some of the microcontroller, available microcontrollers based on the number of data bits.

(Refer Slide Time: 15:29)



Current Favorites – The 8-bitters

- 8051 family – more than 1000 variants, in varied packagings. Standard CISC
- Microchip's PIC – RISC architecture.
- Cypress Semiconductor PSoC. CISC, complete system on chip with programmable analog blocks (https://www.cypress.com/)
- Microchip's (ex-Atmel) AVR. RISC. 200+ chips

So, this is a list for some popular 8-bit microcontrollers. One of the most popular and one of the oldest microcontroller families is the 8051 from Intel. Although, currently Intel does not manufacture any microcontrollers in this family. It has outsourced; it has licensed the technology to other manufacturers. And more than 100 manufacturers worldwide manufacture integrated circuits microcontrollers based on this architecture.

And there are more than 1000 types of microcontrollers in this family. It is of CISC nature. Then we have a company called Microchip. It has a microcontroller family called PIC. That is a RISC architecture. It is also one of the very popular microcontrollers. Then we have Cypress Semiconductor. You may recall me talking about the system on-chip paradigm and they produce a microcontroller family called PSoC.

This is our CISC architecture, programmable system on-chip. And then microchip which acquired a company called Atmel which produced a microcontroller family called AVR which has more than 200 chips in its portfolio. It is of a RISC architecture. And the very popular Arduino platform uses AVR microcontrollers. So, these are top 8-bit microcontroller devices.

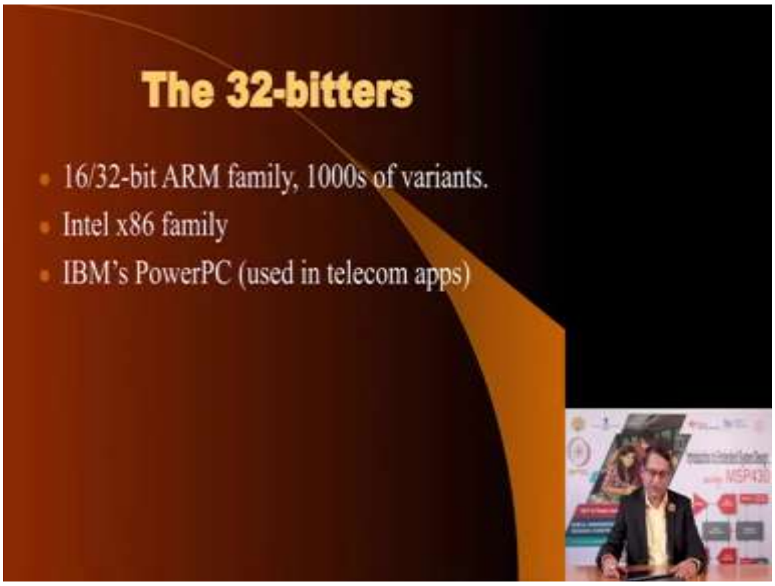
(Refer Slide Time: 16:55)

The 16-bitters

- Texas Instruments MSP430
- Microchip PIC24
- STMicroelectronics ST10
- NXP HC12 (Legacy), HC16

From 8-bit, people would migrate perhaps to 16-bit devices and the available microcontrollers are our very own Texas Instruments MSP430 which we are going to use in this course. It is a 16-bit architecture. Then we have Microchip has a 16-bit family called PIC24. It has ST10 which is a STMicroelectronics offering and NXP which was previously Motorola and then Freescale has a HC12 and HC16 microcontrollers.

(Refer Slide Time: 17:29)



The 32-bitters

- 16/32-bit ARM family, 1000s of variants.
- Intel x86 family
- IBM's PowerPC (used in telecom apps)

From 16, you could have 24-bit architectures. But there are not very many examples and usually people go to 32-bit devices. And we have 16-bit or 32-bit ARM microcontrollers. ARM is one of the largest microcontroller designers and they have 1000 of variants. Incidentally, ARM does not manufacture the actual silicon products. It licenses the technology to third-party vendors who then integrate their own ideas of peripherals and create physical devices.

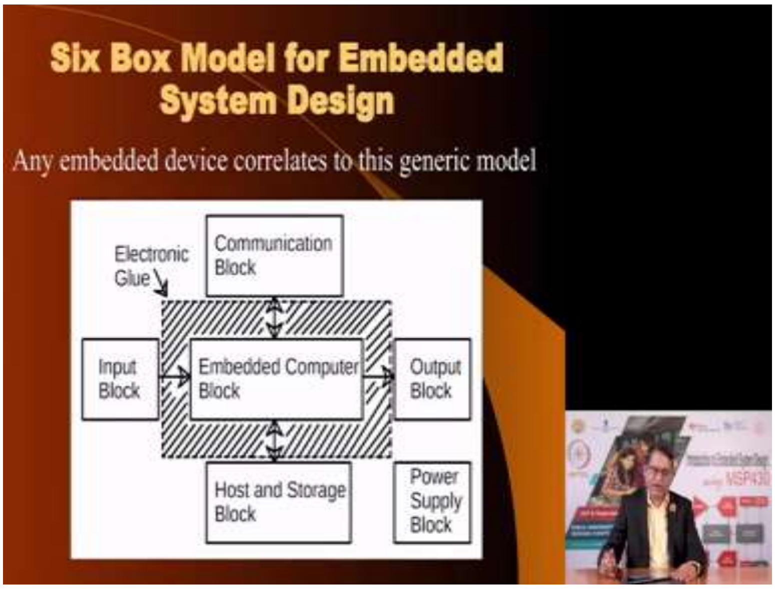
In fact, Texas Instruments make ARM varieties of microcontrollers called TIVA. Many other companies do that. And we also have the Intel's x86 family and IBM's PowerPC. It is heavily used in telecom applications and earlier Apple computers used to use PowerPC. But more than 10-12 years ago, they switched over to the Intel family and then they became very-very popular and successful. But these are the 32-bit, popular microcontroller architectures.

(Refer Slide Time: 18:38)

Modular Approach to an Embedded System Design

Having seen the available varieties in microcontrollers, it is now time to look at a modular approach to designing an embedded system. This is an approach that we have developed here at NSUT in the past few years. As the name suggests that it is possible to look at any embedded system design in the by-way off few modules. So, let us look at this.

(Refer Slide Time: 19:07)



Six Box Model for Embedded System Design

Any embedded device correlates to this generic model

So, we have created what we call as a 6-box model for embedded system design. Any embedded system application could be partitioned in 6 boxes. Each box has a certain characteristic. Let us go through these boxes. You have an input block or input box which is which consists of devices through which a user would interact with the, would provide inputs to the embedded system. These are also devices through which the embedded system would acquire information regarding the environment, maybe sensors and other devices.

The inputs provided by the input block are passed on to the microcontroller that is the embedded computer block. Then we will consider what types of embedded computers? We already seen how they are implemented. The embedded computer acquires the information from the input block, processes this information and produces an output. Processes this information by way of executing a program.

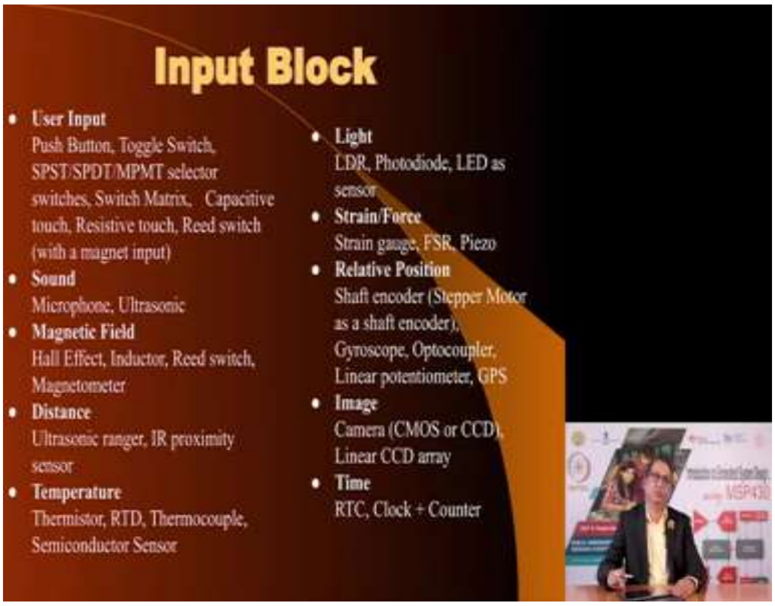
So, the embedded computer block has all the elements to execute that program meaning various types of memory devices. And then produces an output which may be for the user (humans) or it could be to actuate certain controls that are described in the output block. Most of the embedded systems applications would require at least these 3 blocks and the power supply block. Because without power supply, it is not possible to run any electronic circuit.

It may also certain embedded applications may require to communicate with the outside world and so you have a communication block. And sometimes it may require to store data locally, beyond the capacity of the microcontroller so you would have maybe a hard disk or a serial storage device. It may also want to visualize that data in the form of connecting to a host. And so, the sixth block is a host and storage block.

And all these individual blocks are connected tight to the microcontroller or with each other using binding glue that I call as electronic glue. And these are basically electronic circuits which are specific to a particular application. And so we have not classified it as a box but it is dependent on a given application.

Now, what we are going to do is we are going to dive deep into each of these blocks and see what are the available options? How do you when you have a particular requirement, what all options you have?

(Refer Slide Time: 21:48)



Input Block

- **User Input**
  Push Button, Toggle Switch, SPST/SPDT/MPMT selector switches, Switch Matrix, Capacitive touch, Resistive touch, Reed switch (with a magnet input)
- **Sound**
  Microphone, Ultrasonic
- **Magnetic Field**
  Hall Effect, Inductor, Reed switch, Magnetometer
- **Distance**
  Ultrasonic ranger, IR proximity sensor
- **Temperature**
  Thermistor, RTD, Thermocouple, Semiconductor Sensor
- **Light**
  LDR, Photodiode, LED as sensor
- **Strain/Force**
  Strain gauge, FSR, Piezo
- **Relative Position**
  Shaft encoder (Stepper Motor as a shaft encoder), Gyroscope, Optocoupler, Linear potentiometer, GPS
- **Image**
  Camera (CMOS or CCD), Linear CCD array
- **Time**
  RTC, Clock + Counter

So, the input block consists of could be of user inputs for the user to provide input to the embedded system. And these are the various options: you may have a push button or toggle switch which is on and off. You could have switch with multiple settings like you have on a fan speed control: low, medium, high, and so on. These are called SPST or SPDT, or multi pole multi throw switches.

You could have a matrix of switches. You could have a mechanism to interact with the system using a capacitive touch like I am doing with my pen here. You could have a resistive touch which used to be available on older mobile phones. You could also use a Reed switch to interact with the system using a magnet. These are the various options of interacting with the embedded system for the user.

Apart from that, you may want to record or you may want to capture sound. And for sound you could use a microphone or an ultrasonic sensor to sense sound which is beyond the 20 kilohertz range. You may also want to sense the magnetic field in some certain applications and the options are you could use the Hall Effect.

Yesterday, in the last, one of the previous lectures I showed you this levitating doll project. It used a Hall Effect sensor. You could also use an inductor if the especially the magnetic field is of a varying nature and inductor is a good sensor to sense the magnetic field. You could also use a reed switch. It is a binary sensor meaning it will sense either the magnetic field is present or it is absent. And you could also use a magnetometer.

You could, you may want to read the distance to certain object. So, for that you may use the ultrasonic ranger, you could use infrared proximity sensors. You may want to read the temperature of the environment and for that many-many options are available. You could use a thermistor, you could use an RTD device, you could use a thermocouple or you could also use a semiconductor sensor.

In fact, MSP430 microcontroller has on-chip semiconductor-based temperature sensor, so that you could also sense the temperature of the chip. You may want to input information regarding ambient light and for that you could use LDR. And as you know, an LDR is part of our inventory. We could use a photodiode. It turns out that even LED which is otherwise an output device that is with which you can generate light, a LED can also be used to sense light.

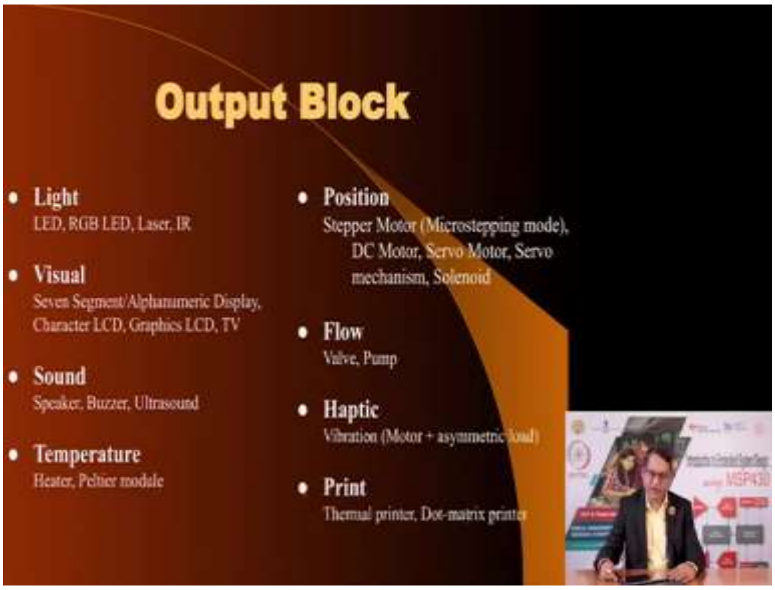
Then you may want to input strain and or force information and for that you have a strain gauge or FSR that is force sensitive resistor. You could use a piezo-sensitive device, piezo sensor measure strain and force. You may want to measure the relative position with respect to certain object and for that you may use a shaft encoder which is usually used when you own your modern instruments like an oscilloscope.

When you are changing the voltage setting or the time setting, the knob that you are moving is actually connected to a shaft encoder. Or in your car stereo when you are changing the stations with the help of a tuning knob, a varying knob, that knob is actually connected to a shaft encoder. You may you could also use a gyroscope or you could use an opto-coupler. You could use linear potentiometers.

Potentiometers which have knob where you move it up and down. And ofcourse, you could use a GPS which is a Global Positioning System. You may want to input image of an input and this image is basically light arranged in either an array or in a straight line. And for that you can use a camera sensor or a linear CCD array. CCD stands for Charge Coupled Device. It is a type of image sensor.

And most importantly, you may want to measure time and for that you could use some internal mechanism to count time or you could use a specialized device called real time clock for keeping track of time. These are all the; not all these is a big list of input devices that would go into the input block.

(Refer Slide Time: 26:19)



Output Block

- **Light**
  LED, RGB LED, Laser, IR
- **Visual**
  Seven Segment/Alphanumeric Display, Character LCD, Graphics LCD, TV
- **Sound**
  Speaker, Buzzer, Ultrasound
- **Temperature**
  Heater, Peltier module
- **Position**
  Stepper Motor (Microstepping mode), DC Motor, Servo Motor, Servo mechanism, Solenoid
- **Flow**
  Valve, Pump
- **Haptic**
  Vibration (Motor + asymmetric load)
- **Print**
  Thermal printer, Dot-matrix printer

Okay apart from the input block we have the output block and these are this is the block through which the embedded system outputs information to the user or to the environment. And you may want to create light or you may want to control light. For that you have an LED, you may have a RGB LED, that is led with red, green and blue together. You may have you may want to control a laser or you may want to control the infrared LED like you would do in a TV remote.

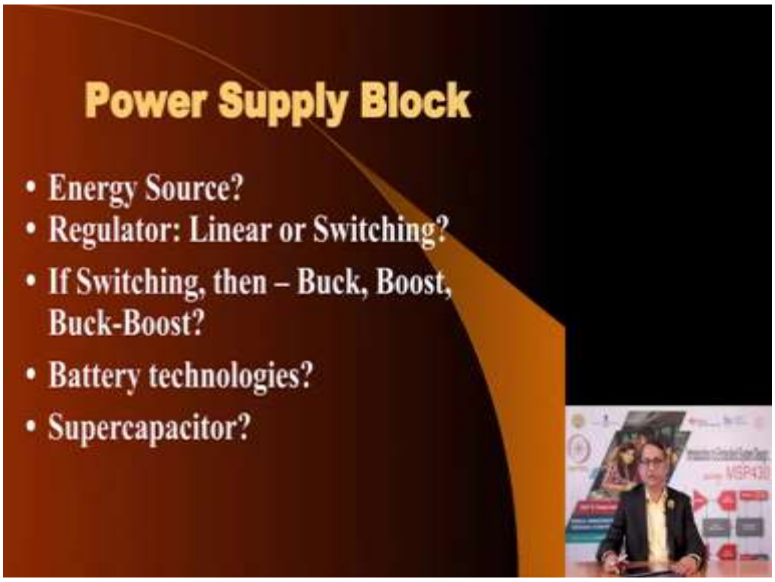
You may want to provide visual information and for that you could use a 7-segment display and we you know that we have a 7-segment display in our experiments inventory, components inventory. You may also have alphanumeric display. You may use a LCD and we have an LCD in our inventory. Or you could use graphics LCD, such as you have on mobile phones or you may want to output information on a television.

You may want to generate sound. And for that you would use a speaker or a buzzer or ultra sound generator so that it generates frequencies in the ultrasound domain. You may want to control the temperature and for that you have largely two devices. One is a heater. Resistor is a good way to generate heat or you could have a Peltier module. The advantage of a Peltier module is that you can heat or cool the environment or at least one side of the surface of the Peltier module.

Then you have, you may want to control the position and for that you would use various types of motors such as a stepper motor, DC motor, a servo motor and so on. You could also use a solenoid where you can move the contacts on and off. You may want to control the flow of liquids and gases. And for that you would use a valve or a pump.

You may want to provide haptic feedback to the user. This haptic feedback happens on your mobile phone whenever you input device, it vibrates a little bit. That is a haptic feedback. And for that, you use a vibration device which consists of a motor and asymmetric load. And you may want to output information in a printed form for which you may have a thermal printer or a dot matrix printer. These are some of the output block components.

(Refer Slide Time: 28:38)



Power Supply Block

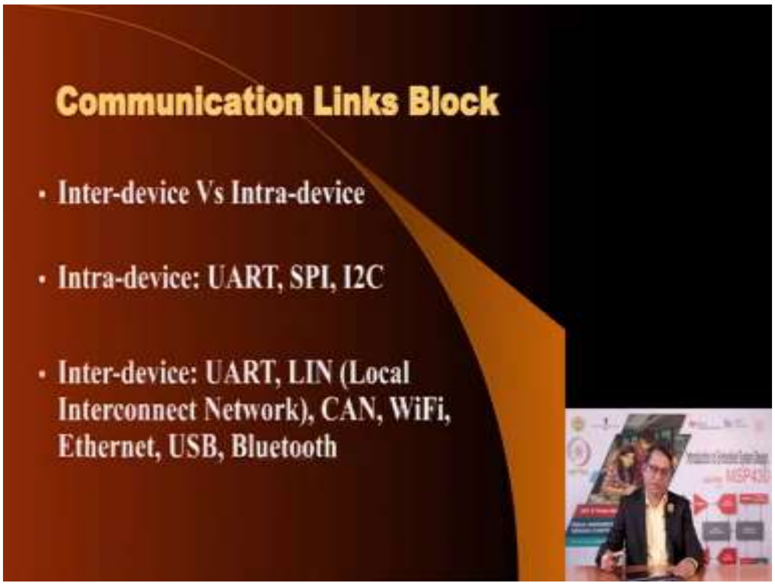
- Energy Source?
- Regulator: Linear or Switching?
- If Switching, then – Buck, Boost, Buck-Boost?
- Battery technologies?
- Supercapacitor?

Then you have the power supply block. For power supply, the first important thing to consider is what is the actual source of energy. Are you getting it through the wall outlet, 220 volt or 110 volts as the case maybe. The energy from the source of energy may be sporadic. It may not be stable and so you may want to, you have to consider a regulator. Regulators are of two types: linear or switching regulators.

Then, if you are going to use a switching regulator, what type of regulator? There are 3 broad types called buck, boost and buck-boost Are you going to use, are you going to need batteries in your application? Because the input sources sporadic and you want to perform your system wants to operate during those times where the input energy may not be available. And therefore, you create a battery backup.

Or are you going to use a super capacitor to store energy during times when the main source of energy is not available? All these considerations go into the design of the power supply block.

(Refer Slide Time: 29:45)



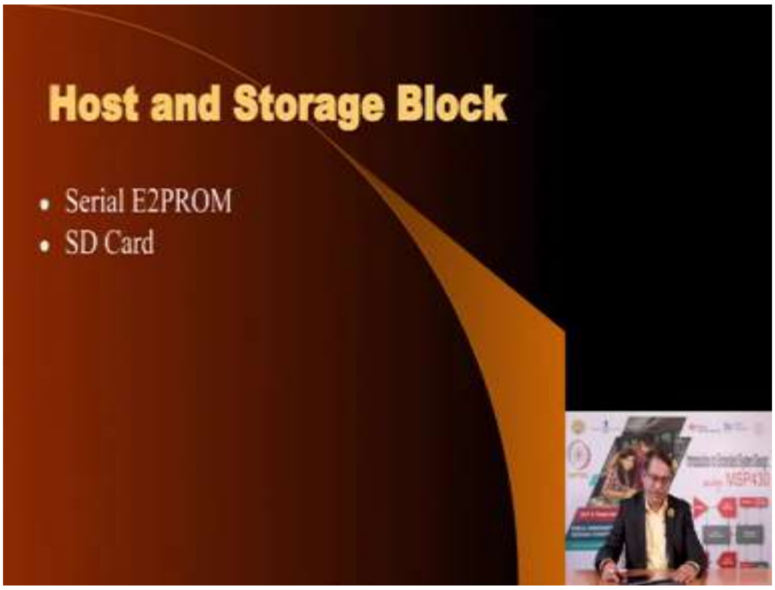
Communication Links Block

- Inter-device Vs Intra-device
- Intra-device: UART, SPI, I2C
- Inter-device: UART, LIN (Local Interconnect Network), CAN, WiFi, Ethernet, USB, Bluetooth

Then, you want to consider various ways of communicating with the outside world. Broadly these are of 2 types: inter-device and intra-device. Inter-device means within the device if certain components want to communicate with each other for sharing information, what are the various protocols available? Or intra-device, oh that is the inter-device- between two devices.

And intra-device means communication within a given like a television within the given system would be classified as intra-device communication protocols. And for intra-device, we have a UART, SPI and I squared C bus. For inter-device, we have several protocols such as UART, local internet interconnect network, we have control area network, WIFI, Ethernet, USB and Bluetooth. These are some of the inter-device communication protocols.

(Refer Slide Time: 30:39)

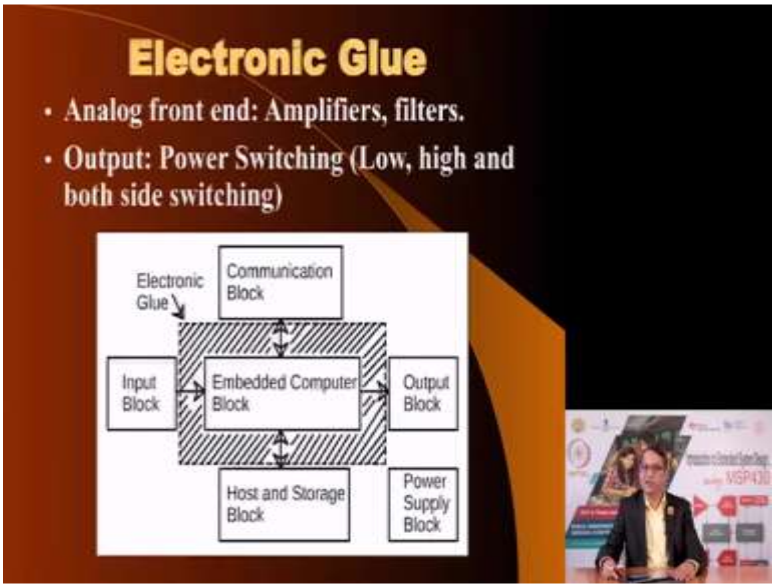


Host and Storage Block

- Serial E2PROM
- SD Card

Then we have the host and storage block where we may have device to store information and you could use a serial E squared PROM or you could use SD card.

(Refer Slide Time: 30:50)

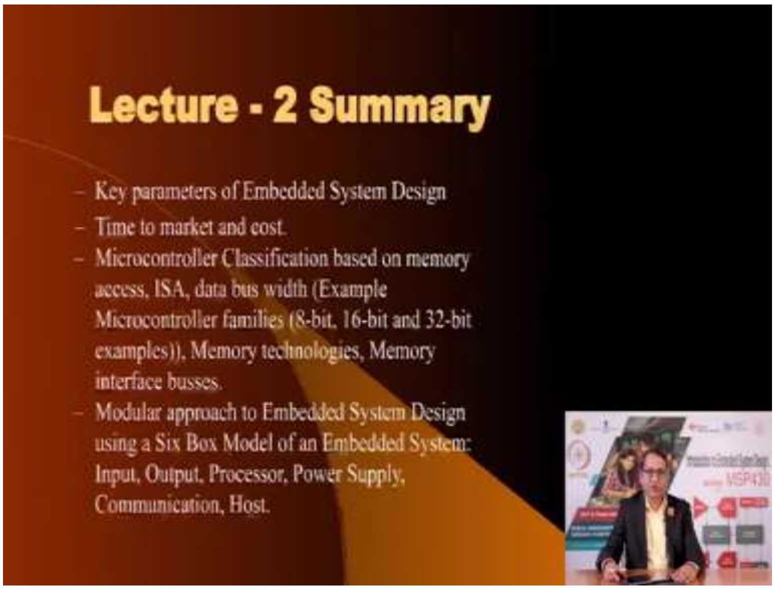


Electronic Glue

- Analog front end: Amplifiers, filters.
- Output: Power Switching (Low, high and both side switching)

Then we have the electronic glue which connects to all these devices. And what are examples of electronic glue? You may require amplifiers you may require filters as the case may be or for the output you may require to switch high amounts of large amount of voltages or currents. For that we have 3 types of drivers. One is called low side driver, high side driver and a both sides switching device which we will consider in detail in subsequent lectures.

(Refer Slide Time: 31:18)



Lecture - 2 Summary

- Key parameters of Embedded System Design
- Time to market and cost.
- Microcontroller Classification based on memory access, ISA, data bus width (Example Microcontroller families (8-bit, 16-bit and 32-bit examples)), Memory technologies, Memory interface busses.
- Modular approach to Embedded System Design using a Six Box Model of an Embedded System: Input, Output, Processor, Power Supply, Communication, Host.

So, here we are at the end of this current lecture where we have seen the various parameters in choosing the microcontroller appropriate for a given application through time to market and cost considerations. Then we have seen the classification of microcontrollers based on the way it communicates with talks to memory, the way the number of bits are available and the instruction set architecture.

And we have introduced to the idea of visualizing an embedded system through 6 blocks and we will go in further detail into each of these blocks in subsequent lectures. I hope you enjoyed this lecture as much as I did in presenting it to you. I will see you very soon. Thank you very much.