**LECTURE NOTES ON EMBEDDED SYSTEM**

**SEMESTER- 6**

**BRANCH- ELECTRONICS & TELECOMMUNICATION**

**UNIT- 1**

COURSE CONTENT:

* 1. Introduction to embedded system

 Embedded system overview.

 What are they?

 A shortlist of embedded system.

 Some common characteristics of embedded system.

 An embedded system example- A digital camera.

* 1. Embedded system technologies.

Technology definition.

Technology for embedded systems.

Processor technology.

IC technology.

Design technology.

* 1. Processor technology.

General purpose processor- software.

Basic architecture of single purpose processor- hardware.

* 1. Application-specific processors.

Microcontrollers.

Digital signal processors

* 1. IC technology.

Full custom / VLSI.

Semi custom ASIC.

PLD.

4. Peripherals.

 Watchdog timers, LCD Controllers.

 Analog to digital convertors.

 Real time clock.

 DS 12887 RTC chip & its interfacing.

 Motor control: Relay & Optoisolator, Stepper motor interfacing, DC motor interfacing.

INTRODUCTION TO EMBEDDED SYSTEM

Overview:

System is a way of working, organizing or doing one or many according to a fixed plan, program or set of rules.

A system is also an arrangement in which all its units assemble and work together according to the plan or program.

Consider a watch. It is a time display system. It consists of hardware, needle, battery, etc. its components follow a set of rules to show time. If one of its parts fails, the watch will stop working.

Thus, in a system, all its components depend on each other.

Embedded system:

Embedded means something that is attached to another thing.

 An embedded system can be a computer hardware system having software embedded in it.

An embedded system can be an independent system or it can be a part of a large system.

Definition: An embedded system is a system that has embedded software and computer hardware, which makes it a system dedicated for an applications or specific part of an application or product or a part of large system.

An embedded system has three components-

1. It has hardware.
2. It has application software.
3. It has real time operating system (RTOS) that supervises the application software and provide mechanism to let the processor run a process as per the plan.

Shortlists of embedded system:

1. Consumer electronics (cell phones, pagers, digital cameras, calculators, portable video games).
2. Home appliances (microwave ovens, home security systems, washing machines, lighting systems).
3. Office automation (fax machine, printers, scanners).
4. Business equipment (cash register, alarm system, card reader, ATM).
5. Automobiles (fuel injection, antilock brakes, active suspension).

Characteristics of embedded systems:

1. Single functioned: an embedded system usually performs a specialized operation and does the same repeatedly. Example- pager.
2. Tightly constrained: all computing system have constraints on design metrics, but those on an embedded system can be especially tight. Design metrics is a measure of an implementation features such as its cost, size, power & performance. Embedded system must be of size to fit on a single chip, must perform fast enough to process data in real time and consume minimum power to extend battery life.
3. Reactive & real time: many embedded systems must continually react to changes in the systems environment and must compute certain result in real time without any delay.

Digital camera:

* A charged coupled device (CCD) contains an array of light sensitive photocells that capture an image.
* The A2D & D2A circuit converts analog images to digital and digital to analog respectively.
* CCD processor provides commands to CCD to read the image.
* Pre-processor modifies data to confirm with input requirements of another program.
* Pixel coprocessor aids in rapidly displaying images.
* Coprocessor manipulates or perform some other specialized function more quickly than the basic processor.
* JPEG Codec (Joint Photographic Experts Group) compresses & decompresses an image using JPEG compression standard, enabling compact storage of image in the limited memory of camera.
* Multiplier/accumulator circuit perform a particular frequently executed multiply/accumulate computation faster than the microcontroller could.
* DMA controller enable direct memory access by other devices while microcontroller is performing other function.
* Memory controller controls access to a memory chip found in the camera.
* ISA bus interface enables a faster connection with a PCs ISA bus.
* UART enables communication with PCs serial port for uploading video frame.
* LCD control & display control circuit control the display of images on the camera’s LCD display.
* Microcontroller is the heart of the of the system which controls all the activities of all other circuit.

EMBEDDED SYSTEM TECHNOLOGY:

Definition: Technology is defined as a manner of accomplishing a task, especially using technical processes, methods or knowledge.

There are three types of embedded system technology:

1. Processor technology.
2. IC technology.
3. Design technology.

Processor technology relates to the architecture of the computation engine used to implement a system’s derived functionality.

IC technology involves the manner in which we map a digital (gate level) implementation onto an IC. Ic technology is independent from processor technology, any type of processor can be mapped to any type of IC technology.

Design technology involves the manner in which we convert our concept of desired system functionality into an implementation.

PROCESSOR TECHNOLOGY:

Processor is usually associated with programmable software processor.

It can also be non programmable.

So, each processor differs in its specialization towards a particular function.

Processor may be-

General purpose processor or single purpose processor.

General purpose processor (software)-:

It is also called microprocessor.

It is a programmable device that is suitable for a variety of applications to maximize the number devices sold.

Features:

1. Program memory: since the designer does not know what program will run in the processor, hence the program cannot be built in the digital circuit.
2. General data path: data path (ALU, registers, buses) must be general enough to handle a variety of computations. Data path has large register file & one or more general purpose ALUs.

Benefits:

* Flexibility is high because only program is to be changed.
* Time to market and one time cost of design is low.
* Performance may be fast for computation.

Drawbacks:

* Unit cost may be high for large quantities.
* Performance may be slow for certain application.
* Size & power may be large due to unnecessary processor hardware.

Single processor (hardware):

A single processor is a digital circuit designed to execute exactly one program.

Eg. Digital camera, JPEG Codec.

An embedded system designer may create a single purpose processor by designing a custom digital circuit or a pre-designed single purpose processor.

This part of implementation is referred as ‘hardware’ portion.

It also includes coprocessor, accelerator & peripherals.

Benefits:

* Performance may be fast.
* Size & power may be small.
* Unit cost may be low for large quantities.

Drawback:

* Design time and one time cost may be high.
* Low flexibility.
* Unit cost high for small quantities.
* Performance may not match general purpose processor for some applications.

APPLICATION SPECIFIC PROCESSOR:

Also called Application Specific Instruction Set Processor (ASIP) is a compromise between general and single purpose processor.

An ASIP is a programmable processor effective for particular class of applications having common characteristics, such as embedded control, DSP or Telecommunication.

The designer optimize the data path for the application by adding special function unit for common operation & eliminating other frequently used units.

ASIP provides flexibility, good performance, power, size & large one time cost.

There are two types of ASIP-

Microcontroller & DSP.

Microcontroller has been optimized for embedded control applications.

DSP designed to perform common operation on digital signals such as encoding & decoding of video & audio.

Microcontrollers:

It is a processor that has been optimized for embedded control applications.

These applications monitors and set numerous single bit control signals but do not perform large amount of data computation.

It includes several peripheral devices, such as timers, analog to digital convertors and serial communication device on same IC.

It includes program and data memory on same IC.

It provides the programmer with direct access to a no. of pins of the IC.

With peripherals and memory on same IC, reduces the no. of IC required.

Hence, microcontrollers are compact and low power implementations.

Pin access to programmer provides easy monitor to sensors, to set actuators and to transfer data with other devices.

Special instructions improves performance for embedded system applications.

DSP (digital signal processor):

DSP are highly optimized for processing large amount of data.

 Large amount of data is in the form of digitized signal, such as photo image captured in digital camera, a voice packet going through router or audio clip played by digital keyboard.

DSP contains numerous register files, memory blocks, multipliers and other arithmetic units.

DSP provides instructions that are control to digital signal processing, such as filtering & transforming vectors or matrix of data.

In DSP, frequently used arithmetic functions, such as multiply and accumulate, are implemented in hardware and thus execute order of magnitude faster than a software implementation running on general purpose processor.

DSP allows execution of some function in parallel, resulting in a boost of performance.

DSP allows peripherals that are useful in signal processing on a single IC.

DSP may include no. of A2D & D2A convertor, PWM, DMA controllers, timers & counters.

IC TECHNOLOGY:

All processors are implemented on an integrated circuit (IC).

IC technology involves the manner in which we map a digital (gate level) implementation onto an IC.

An IC, also called chip, is a semiconductor device consisting of a set of connected transistors and other devices.

Different processors are used to build semiconductors. Eg. CMOS.

IC technology is independent of processor technology.

Any type of processor can be mapped to any type of IC technology.

Types of IC technology:

Full Custom, Semi-Custom ASIC & PLD.

In full custom IC, complete IC is fabricated by the designer.

In semi-custom IC, lower layer are fully or partially built and upper ayer is left for users to finish.

PLD is a pre-manufactured IC that can be purchased and then configured to implement desired circuit.

Full Custom / VLSI:-

• In a full-custom IC technology, we optimize all layers for our particular embedded system’s digital implementation.

• Such optimization includes placing the transistors to minimize interconnection lengths, sizing the transistors to optimize signal transmissions and routing wires among the transistors.

 • Once all the masks are completed, then we send the mask specifications to a fabrication plant that builds the actual ICs.

• Full-custom IC design, often referred to as VLSI (Very Large Scale Integration) design, has very high NRE cost and long turnaround times (typically months) before the IC becomes available, but can yield excellent performance with small size and power.

• It is usually used only in high-volume or extremely performance-critical applications.

SEMICUSTOM ASIC (GATE ARRAY AND STANDARD CELL):

• In an ASIC (Application-Specific IC) technology, the lower layers are fully or partially built, leaving us to finish the upper layers.

• In a gate array technology, the masks for the transistor and gate levels are already built (i.e., the IC already consists of arrays of gates).

• The remaining task is to connect these gates to achieve our particular implementation.

• In a standard cell technology, logic-level cells (such as an AND gate or an ANDORINVERT combination) have their mask portions pre-designed, usually by hand.

• Thus, the remaining task is to arrange these portions into complete masks for the gate level, and then to connect the cells.

• ASICs are by far the most popular IC technology, as they provide for good performance and size, with much less NRE cost than full-custom IC’s.

PLD:

• In a PLD (Programmable Logic Device) technology, layers implement a programmable circuit, where programming has a lower-level meaning than a software program.

• The programming that takes place may consist of creating or destroying connections between wires that connect gates, either by blowing a fuse, or setting a bit in a programmable switch.

• Small devices, called programmers, connected to a desktop computer can typically perform such programming.

• PLD's of two types, simple and complex. One type of simple PLD is a PLA (Programmable Logic Array), which consists of a programmable array of AND gates and a programmable array of OR gates.

• Another type is a PAL (Programmable Array Logic), which uses just one programmable array to reduce the number of expensive programmable components.

• One type of complex PLD, growing very rapidly in popularity over the past decade, is the FPGA (Field Programmable Gate Array), which offers more general connectivity among blocks of logic, rather than just arrays of logic as with PLAs and PALs, and is thus able to implement far more complex designs. PLDs offer very low NRE cost and almost instant IC availability.

• They are typically bigger than ASICs, may have higher unit cost, may consume more power, and may be slower (especially FPGAs). They still provide reasonable performance, though, so are especially well suited to rapid prototyping.

UNIT -5:

Peripherals:

Watch-dog timer:

* **For those embedded systems that can't be constantly watched by a human, watchdog timers may be the solution.**
* Most embedded systems need to be self-reliant. It's not usually possible to wait for someone to reboot them if the software hangs. Some embedded designs, such as space probes, are simply not accessible to human operators. If their software ever hangs, such systems are permanently disabled. In other cases, the speed with which a human operator might reset the system would be too slow to meet the uptime requirements of the product.
* A watchdog timer is a piece of hardware that can be used to automatically detect software anomalies and reset the processor if any occur. Generally speaking, a watchdog timer is based on a counter that counts down from some initial value to zero. The embedded software selects the counter's initial value and periodically restarts it. If the counter ever reaches zero before the software restarts it, the software is presumed to be malfunctioning and the processor's reset signal is asserted. The processor (and the embedded software it's running) will be restarted as if a human operator had cycled the power.
* A watchdog timer can get a system out of a lot of dangerous situations. However, if it is to be effective, resetting the watchdog timer must be considered within the overall software design. Designers must know what kinds of things could go wrong with their software, and ensure that the watchdog timer will detect them, if any occur.
* Another possibility is that an unusual number of interrupts arrives during one pass of the loop. Any extra time spent in ISRs is time not spent executing the main loop. A dangerous delay in feeding the motor new control instructions could result.
* A watchdog timer is a useful tool in helping your system recover from transient failures. Since it is so common to find watchdogs built into modern microcontrollers, the technique is effectively free. If you are working on a mission-critical system, then either common sense or a regulatory body will insist that you use a watchdog. It's always a good idea to make your systems more self-reliant.

LCD Controller:

* LCD controller An LCD (Liquid crystal display) is a low-cost, low-power device capable of displaying text and images.
* LCDs are extremely common in embedded systems, since such systems often do not have video monitors standard for desktop systems.
* LCDs can be found in numerous common devices like watches, fax and copy machines, and calculators.
* The basic principle of one type of LCD (reflective) works as follows.
* First, incoming light passes through a polarizing plate.
* Next, that polarized light encounters liquid crystal material. If we excite a region of this material, we cause the material’s molecules to align, which in turn causes the polarized light to pass through the material. Otherwise, the light does not pass through.
* Finally, light that has passed through hits a mirror and reflects back, so the excited region appears to light up.
* Another type of LCD (absorption) works similarly, but uses a black surface instead of a mirror.
* The surface below the excited region absorbs light, thus appearing darker than the other regions.
* One of the simplest LCDs is 7-segment LCD. Each of the 7 segments can be activated to display any digit character or one of several letters and symbols.
* Such an LCD may have 7 inputs, each corresponding to a segment, or it may have only 4 inputs to represent the numbers 0 through 9.
* An LCD driver converts these inputs to the electrical signals necessary to excite the appropriate LCD segments.
* A dot-matrix LCD consists of a matrix of dots that can display alphanumeric characters (letters and digits) as well as other symbols.
* A common dot-matrix LCD has 5 columns and 8 rows of dots for one character.
* An LCD driver converts input data into the appropriate electrical signals necessary to excite the appropriate LCD bits.
* Each type of LCD may be able to display multiple characters. In addition, each character may be displayed in normal or inverted fashion.
* The LCD may permit a character to be blinking (cycling through normal and inverted display) or may permit display of a cursor (such as a blinking underscore) indicating the "current" character.
* This functionality would be difficult for us to implement using software. Thus, we use an LCD controller to provide us with a simple interface, perhaps 8 data inputs and one enable input.
* To send a byte to the LCD, we provide a value to the 8 inputs and pulse the enable. This byte may be a control word, which instructs the LCD controller to initialize the LCD, clear the display, select the position of the cursor, brighten the display, and so on.
* Alternatively, this byte may be a data word, such as an ASCII character, instructing the LCD to display the character at the currently-selected display position.

Stepper motor controller:

* A stepper motor is an electric motor that rotates a fixed number of degrees whenever we apply a "step" signal.
* In contrast, a regular electric motor rotates continuously whenever power is applied, coasting to a stop when power is removed.
* We specify a stepper motor either by the number of degrees in a single step, such as 1.8Ε, or by the number of steps required to move 360Ε, such as 200 steps.
* Stepper motors obviously abound in embedded systems with moving parts, such as disk drives, printers, photocopy and fax machines, robots, camcorders, VCRs, etc. Internally, a stepper motor typically has four coils.
* To rotate the motor one step, we pass current through one or two of the coils; the particular coils depends on the present orientation of the motor.
* Thus, rotating the motor 360Ε requires applying current to the coils in a specified sequence.
* Applying the sequence in reverse causes reversed rotation. In some cases, the stepper motor comes with four inputs corresponding to the four coils, and with documentation that includes a table indicating the proper input sequence.
* To control the motor from software, we must maintain this table in software, and write a step routine that applies high values to the inputs based on the table values that follow the previously-applied values.
* In other cases, the stepper motor comes with a built-in controller (i.e., a special purpose processor) implementing this sequence.
* Thus, we merely create a pulse on an input signal of the motor, causing the controller to generate the appropriate high signals to the coils that will cause the motor to rotate one step.

Analog-digital converters:

* An analog-to-digital converter (ADC, A/D or A2D) converts an analog signal to a digital signal, and a digital-to-analog converter (DAC, D/A or D2A) does the opposite. Such conversions are necessary because, while embedded systems deal with digital values, an embedded system’s surroundings typically involve many analog signals. Analog refers to continuously-valued signal, such as temperature or speed represented by a voltage between 0 and 100, with infinite possible values in between. "Digital" refers to discretely-valued signals, such as integers, and in computing systems, these signals are encoded in binary.
* By converting between analog and digital signals, we can use digital processors in an analog environment. For example, consider the analog signal.
* The analog input voltage varies over time from 1 to 4 Volts.
* We sample the signal at successive time units, and encode the current voltage into a 4-bit binary number.
* We want to generate an analog output voltage for the given binary numbers over time. We generate the analog signal.
* We can compute the digital values from the analog values, and vice-versa, using the following ratio:

 e/Vmax = d/(2n-1)

* Vmax is the maximum voltage that the analog signal can assume, n is the number of bits available for the digital encoding, d is the present digital encoding, and e is the present analog voltage.
* This proportionality of the voltage and digital encoding is shown graphically. In our example of suppose Vmax is 7.5V.
* Then for e = 5V, we have the following ratio: 5/7.5 = d/15, resulting in d = 1010 (ten). The resolution of a DAC or ADC is defined as Vmax/(2n -1), representing the number of volts between successive digital encodings.
* The above discussion assumes a minimum voltage of 0V. Internally, DACs possess simpler designs than ADCs. A DAC has n inputs for the digital encoding d, a Vmax analog input, and an analog output e.
* A fairly straightforward circuit (involving resistors and an op-amp) can be used to convert d to e.
* ADCs, on the other hand, require designs that are more complex, for the following reason.
* Given a Vmax analog input and an analog input e, how does the converter know what binary value to assign in order to satisfy the above ratio?
* Unlike DACs, there is no simple analog circuit to compute d from e. Instead, an ADC may itself contain a DAC also connected to Vmax.
* The ADC "guesses" an encoding d, and then evaluates its guess by inputting d into the DAC, and comparing the generated analog output e’ with the original analog input e (using an analog comparator).
* If the two sufficiently match, then the ADC has found a proper encoding. So now the question remains: how do we guess the correct encoding?
* This problem is analogous to the common computer-programming problem of finding an item in a list.
* One approach is sequential search, or "counting-up" in analogdigital terminology. In this approach, we start with an encoding of 0, then 1, then 2, etc., until we find a match.
* Unfortunately, while simple, this approach in the worst case (for high voltage values) requires 2n comparisons, so it may be quite slow.
* A faster solution uses what programmers call binary search, or "successive approximation" in analog-digital terminology.
* We start with an encoding corresponding half of the maximum. We then compare the resulting analog value with the original; if the resulting value is greater (less) than the original, we set the new encoding to halfway between this one and the maximum (minimum).
* We continue this process, dividing the possible encoding range in half at each step, until the compared voltages are equal.
* This technique requires at most n comparisons. However, it requires a more complex converter.
* Because ADCs must guess the correct encoding, they require some time.
* Thus, in addition to the analog input and digital output, they include an input "start" that starts the conversion, and an ouput "done" to indicate that the conversion is complete.

Real-time clocks:

* Much like a digital wristwatch, a real-time clock (RTC) keeps the time and date in an embedded system.
* Read-time clocks are typically composed of a crystal-controlled oscillator, numerous cascaded counters, and a battery backup.
* The crystal-controlled oscillator generates a very consistent high-frequency digital pulse that feed the cascaded counters.
* The first counter, typically, counts these pulses up to the oscillator frequency, which corresponds to exactly one second.
* At this point, it generates a pulse that feeds the next counter.
* This counter counts up to 59, at which point it generates a pulse feeding the minute counter.
* The hour, date, month and year counters work in similar fashion.
* In addition, real-time clocks adjust for leap years.
* The rechargeable back-up battery is used to keep the real-time clock running while the system is powered off.
* From the micro-controller’s point of view, the content of these counters can be set to a desired value, (this corresponds to setting the clock), and retrieved. Communication between the micro-controller and a real-time clock is accomplished through a serial bus, such as I 2C.
* It should be noted that, given a timer peripheral, it is possible to implement a real-time clock in software running on a processor.
* In fact, many systems use this approach to maintain the time.
* However, the drawback of such systems is that when the processor is shut down or reset, the time is lost.