Embedded Systems

UNIT-I

Notes

①

✓ What is an embedded System?

This is an electronic/electro-mechanic System designed to perform a specific function. It is the combination of both Hardware and software (firmware).

Examples: cellphone, microwave, washingmachine, Robotic System, automatic chocolate vending machine etc.

✓ Embedded System Vs General computing System

Embedded System	Generalpurpose Computing System.
① A System which is a Combination of special purpose hardware & Embedded OS for executing a specific set of applications.	① A System which is a combination of a generic hardware and a General purpose OS for executing a variety of applications
② May or may not contain an OS for functioning.	② Contains a general purpose Operating Systems (GPOS)
③ Applications are alterable (Programmable)	③ The firmware of ES is pre-programmed & it is non-alterable.

GPC	
(4) Performance is the key deciding factor in the selection of the system. Always 'Faster is Better'.	(4) Application-specific requirements are the key deciding factors.
(5) Not much focused on reduced operating power requirements.	(5) Highly focused to take advantage of the power saving modes supported by the hardware and operating system.
(6) Response requirements are not time-critical.	(6) For certain type of embedded system like mission-critical systems, the response time is highly critical
(7) Need not be deterministic in execution behaviour.	(7) Execution behaviour is deterministic for certain ES like "Hard Real Time" systems.

Classification of Embedded Systems: Criteria used in the classification of E.S.

* Based on Generation
* Complexity & Performance requirements
* Based on deterministic behaviour
* Based on triggering

Classification Based on Generation: This is based on the order in which ES are evolv

Embedded Systems

UNIT-II

First Generation: Early embedded systems were built around 8-bit microprocessors like 8085 and 4-bit microcontrollers. The firmware is developed in Assembly code.

Ex: Digital Telephone keypads
Stepper motor control unit.

Second Generation: These were built around 16-bit microprocessors & 8 or 16-bit microcontrollers. The instructions set for second generation processors/controllers were much more complex and powerful than the first generation processors/controllers.

Ex: ~~Data~~ Acquisition systems
SCADA Systems.

Third Generation: These were built around 32-bit processors and 16-bit microcontrollers. A new kind of Application specific and domain specific processors/controllers like DSP (Digital Signal Processors) & Application specific Integrated Circuits (ASICs) came into picture. Instruction sets of processors are more complex & powerful - concept of instruction pipelining also evolved.

Embedded Systems spread its ground to areas like robotics, media, industrial process control, networking etc.

Fourth Generation: The advent of System on Chips (SoC), reconfigurable processors and multicore processors are bringing high performance, tight integration and miniaturisation into the embedded device market. These systems are making use of high performance real-time embedded operating systems for their functioning.

Ex : Smart Phone Device
Mobile Internet Device

✓ Classification Based on complexity and performance

Small-scale Embedded Systems: These are simple in application needs, the performance requirements are not time critical. These are buit around low-performance and lowcost 8-bit or 16-bit microprocessors/microcontrollers. These may or may not contain operating system.

Ex Electronic-Toy

Medium-scale Embedded System: These are slightly complex in hardware and firmware requirements. These are buit around medium performance, low cost 16 or 32-bit mp/mc or DSP. These may use OS.

Embedded Systems

UNIT-I ③

Large-scale Embedded Systems | Complex System

These involves highly complex hardware and firmware requirements. These are employed in mission critical applications demanding high-performance. These may requires high-performance 32 bit or 64-bit RISC processors/controllers or Reconfigurable systems on chip (RSOC) or multi-core processors and programmable logic devices. Complex embedded system usually contain a high-performance Real Time Operating System (RTOS) for task scheduling, prioritization and management.

✓ Major Application Areas of Embedded Systems.

* Consumer electronics: Camcorders, Cameras etc
* Household Appliances: Television, DVD player, washing machine, fridge, microwave oven etc.
* Home automation & security systems: Air conditioners, sprinklers, intruder detection alarms, closed circuit television cameras, fire alarms etc.
* Automotive Industry: Anti-lock breaking systems (ABS), Engine control, ignition systems, Automatic Navigation Systems.

* Telecom: cellular telephones, telephone switches, handset multimedia applications etc.
* Computer peripherals: Printers, scanners, fax machines. etc.
* Networking systems: Network routers, switches, hubs, firewalls etc.
* Healthcare: Different kinds of scanner EEG, ECG machines etc.
* Measurement & Instrumentation: Digital multimeter, Digital CRO's, logic analyzer PLC systems.
* Banking & Retail: Automatic teller machine (ATM) and currency counter point of sale (POS).
* Card Readers: Barcode, smart card readers, hand held devices, etc.

Embedded Systems

UNIT - I (4)

Purpose of Embedded Systems:

Embedded systems are applicable to allmost all the areas and alied ares.

Each embedded system is designed to serve the purpose of any one of a combination of the following tasks.

1. Data collection | storage | Representation
2. Data communication
3. Data | signal processing
4. Monitoring
5. Control
6. Application speific user interface

Data Collection | storage | Representation:

ES are designed for the purpose of data collection performs acquisition of data from the external world. Data collection is usually done for storage, analysis, manipulation and transmission. Data may be analog or digital.

The collected data may be stored directly in the system or may be transmitted to some other systems or processed immediately and deleted.

Some embedded systems store the

data for processing and analysis. Such systems incorporate a built-in/plug-in storage memory for storing the captu-red data. Some of them give the user a meaningful representation of the collected data by visual or audible means using display units.

Ex: Measuring instruments with storage memory and monitoring instruments with storage memory used in medical applications.

A digital camera is a typical example of an ES with data collection/storage/Representation of data.

Data Communication: Embedded data communication systems are deployed in applications ranging from complex satellite communication systems to simple home networking systems. The transmission is achieved either by a wire-line medium or by a wire-less medium. Data can either be transmitted by analog means or by digital means. Modern industry trends are setting towards digital communication.

Data (signal) Processing: Data collected by the ES can be used for various kinds of ~~data~~ processing. ES with signal processing can process signals, useful in speech coding.

synthesis, audio video codec, transmission applications, etc.

A digital hearing aid is a typical example of an embedded system employing data processing. Digital hearing aid improves the hearing capacity of hearing impaired persons.

Monitoring: Almost all embedded products coming under the medical domain are with monitoring functions only.

Ex: ECG (Electro cardiogram)
EEG

ECG — used to monitor heartbeat of a patient.

Digital Multimeter, CRO etc. } These can read and display the parameters.

Control:

A system with control functionality contains both sensors and actuators. Mostly sensors are connected to input port and actuators are connected to output port.

AC — used in our home to control the room temperature to a specified limit is a typical example for ES for control purpose.

✓ Core of the Embedded System:

ES are domain and application specific and are built around a central core. The core of the embedded system

1. General Purpose and Domain Specific Processors
   * Microprocessors
   * Microcontrollers
   * Digital Signal Processors
2. Application Specific Integrated Circuits (ASICs)
3. Programmable Logic Devices (PLD's)
4. Commercial Off-the-Shelf Components (COTS)

✓ General purpose and Domain specific processors.

Almost 80% of the embedded systems are processor/controller based.

Microprocessors: It is a silicon chip represents a CPU, which is capable of performing arithmetic as well as logical operations according to a pre-defined set of instructions. The CPU contains Arithmetic Logic Unit (ALU), control unit and working registers. MP is a dependent unit and it requires the combination of other hardware like memory unit, timer unit and interrupt controller, etc. for proper functioning.

Embedded Systems

✓ Classification based on deterministic System behaviour is applicable for 'Real Time' Systems. The application/task execution behaviour for an embedded ~~step~~ system can either be

* Deterministic
* Non-Deterministic

✓ Based on execution behaviour, Real Time embedded Systems are classified into

* Hard Real-Time
* Soft Real-Time

✓ Embedded Systems which are 'Reactive' in nature can be classified based on the trigger. These can be either

* Event Triggered
* Time Triggered.

Embedded Systems

UNIT - I ⑥

Intel claims the credit for developing the first microprocessor unit Intel-4004. this a 4-bit processor.

Intel, AMD, Freescale, IBM, TI, Cyrix, Hitachi, NEC, LSI logic. etc are the key players in the processor market.

The common system architectures for processor design are

* Harvard Architecture
* Von-Neuman Architecture

In case Harvard Architecture - separate buses ~~are~~ there for program memory and data memory, where as Von-Neuman Architecture uses same bus for both of them.

The architectures based on the instruction set are

* Reduced Instruction Set Computing (RISC)
* Complex Instruction Set Computing (CISC)

There may be GPP (General Purpose Processor) or Application-Specific Instruction Set Processor. (ASIP)

GPP is for general computational task

ASIP - incorporate a on chip - peripherals can be for special purpose computing.

✓ **Microcontroller:** A microcontroller is a highly integrated chip contains a CPU, Scratch pad RAM, special and general register arrays on chip ROM/FLASH memory for program storage, timer and interrupt control units and dedicated I/O ports. MC are considered of a super set of MP. Since a MC contains all necessary functional blocks for independent working. MC are cheap, cost effective and are readily available in the market.

Texas Instrument's TMS1000 is the first world's MC. The popular MC is 8051

General purpose MC – 8051

Application Specific Instruction set processor – AVR MC.

Digital Signal Processor (DSP): These are powerful special purpose 8/16/32 bit MP, designed to meet the computational demand and power constraints of embedded audio, video and communication applications. DSPs are 2 to 3 times faster than the general purpose MPs in signal processing applications.

A typical DSP incorporates

* Program Memory, * Data Memory
* Computational Engine
* I/O Unit

Embedded Systems

UNIT-I (7)

Program Memory: Memory for storing program required by DSP to process the data.

Data Memory: Working memory for storing temporary variables and data/signal to be processed.

Computational Engine: This performs the signal processing in accordance with the stored program memory. This incorporates many specialised arithmetic units each of them operates simultaneously to increase the execution speed.

I/O Unit: Acts as an interface b/w the outside world and DSP. It is responsible for capturing signals to be processed and delivering the processed signals.

RISC Vs CISC processors/controllers:

The term RISC stands for Reduced Instruction Set Computing. All RISC processors/controllers posses lesser number of instructions typically in the range of 30 to 40. CISC stands for Complex Instruction Set Computing. These instruction set is complex and these instructions are high in number.

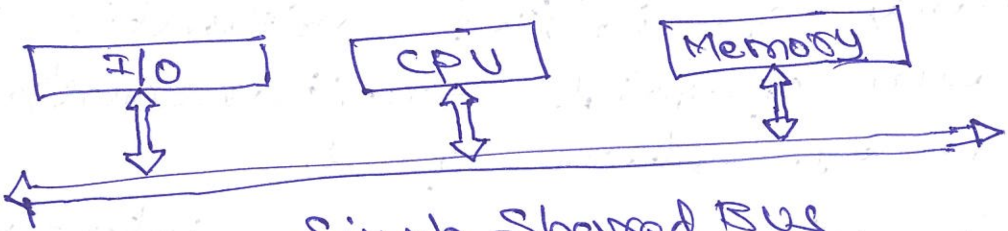
Atmel AVR — is RISC processor — 32 Instructions

8051 — is CISC processor — 255 Instructions

The features like Pipelining, instruction set type, etc determines RISC/CISC criteria.

Harvard - Vs - Von-Neumann Processor/controller Architecture

MC or MP based on Von-Neumann architecture shares a single common bus for fetching both instructions and data. Program instructions and data are stored in a common main memory. These kind of MP/MCs first fetch an instruction and then fetch the data to support instruction from code memory. The two separate fetches slows down the controller's operation. This architecture is also called as Princeton architecture.



Single Shared Bus

The MP/MCs based on Harvard architecture have separate databus and instruction bus. This makes data transfer and program fetching to occur simultaneously on both buses. This allows one instruction to execute while the next instruction is fetched. This results in much faster execution than Von-Neumann architecture.

Big-Endian Vs Little-Endian Processors/controllers:

These specifies the order in which the data is stored in the memory by processor operations in the multibyte system.

① Higher order of data byte at the heigher memory and lower order of data byte at location just below higher memory.

(2) Lower order of data byte at the higher memory and higher order of data byte at location just below the higher memory.

Load Store Operation and Instruction Pipelining:

✓ Application Specific Integrated Circuits (ASIC)

It is a microchip designed to perform a Specific or unique application. It integrates several functions into a single chip and there by reduces the system development cost. This consumes a very small area in the total System and there by helps in the design of smaller systems with high capabilities. These may be pre-fabricated or custom fabricated.

✓ Programmable Logic Devices (PLD)

Logic devices provide specific functions including device-to-device interfacing, data Communication, signal processing, data display timing and control operations and almost every other function a system perform.

These may be * fixed
* Programmable

PLD's offer customers a wide range of logic capacity, features, speed, and votage characteristics - and - these devices can be re-configured to perform any no. of functions at any time.

with PLD, designers use inexpensive s/w tools to quickly develop, simulate and test their designs.

The two major types of PLD are

* Field Programmable Gate Arrays (FPGAS)
* Complex Programmable Logic Devices (CPLDS)

FPGAs - offers highest amount of logic density, the amount of features, and the highest performance. FPGAS are used in wide variety of applications ranging from the data processing and storage, to instrumentation, telecommunications and digital signal processing.

CPLD's - offers smaller amounts of logic - up to about 10,000 gates. But CPLDs offer very predictable timing characteristics and are therefore ideal for critical control applications.

Advantages of PLD: These offers a no. of important advantages over fixed logic devices

* Offers more flexibility during the design cycle and results design changes can be seen immediately in working parts.
* PLDs do not require long lead times for prototypes or production parts.
* PLDs do not require customers to pay large NRE costs and purchase expensive mask sets.

* PLDs allow customers to order just the no. of parts they need, when they need them allowing them to control inventory.
* PLDs can be reprogrammed even after a piece of equipment is shipped to a customer.

Commerical off-the-Shelf Components (COTS)

COTS product is one which is used 'as-is'. These products are designed in such a way to provide easy integration and interoperability with existing system components. COTS component may be developed around a general purpose or domain specific processor or an ASIC or PLD. The ~~COTS~~ advantage of COTS is that they are readily available in the market are cheap and a developer can cut down his/her development time to a great extent. This in turn reduces the time to market your embedded systems.

The major drawback of using COTS components in the embedded design is that the manufacturer of the COTS component may withdraw the product or discontinue the production of the COTS at any time if rapid change in the technology.

Memory

Memory is an important part of a processor/controller based on embedded systems. Some of the processors/controllers contain built in memory and this memory is referred as on-chip-memory. Others do not contain any memory inside the chip and requires external memory to be connected with controller/processor to store the control algorithm. It is called off-chip-memory.

Different types of memory used in Embedded systems

* Program Storage Memory (ROM)
* Read-Write Memory / Random Access Memory (RAM)

ROM :
* MROM [Masked ROM]
* PROM [Programmable Read only Memory]
* EPROM [Erasable Programmable ROM]
* EEPROM [Electrically Erasable PROM]

RAM :
SRAM [Static RAM]
DRAM [Dynamic RAM]

Sensors & Actuators:

Sensors: A sensor is a transducer device that convert energy from one form to another for any measurment or control purpose.

Ex : Temperature sensor.

Embedded Systems

UNIT - I

(10)

Actuators: It is a form of transducer device (Mechanical | Electrical) which converts signals to corresponding physical action (motion). Actuator acts as an output device.

Ex: switch

Communication Interface:

Communication interface is essential for communicating with various subsystems of the ES. & the external world. These may be viewed in two different perspectives.

* Device | Board Level communication Interface.
* Product Level Communication Interface. [External communication Interface].

On Board Communication Interfaces:

Onboard Communication Interface refers to the different communication channels | buses for interconnecting the various integrated circuits & other peripherals within the ES. Following are the interfaces for onboard communication.

* Inter Integrated Circuit (I2C) Bus
* Serial Peripheral Interface (SPI) Bus
* Universal Asynchronous Receiver Transmitter (UART)
* 1-Wire Interface
* Parellel - Interface

External Communication Interface:

The External communication Interface refers to the different communication channel/buses used by the ES to communicate with the external world. Some of them are

* RS-232 C & RS-485
* Universal Serial Bus (USB)
* IEEE 1394 (Firewire)
* Infrared (IrDA)
* Bluetooth (BT)
* Wi-Fi
* ZigBee
* General Packet Radio Service (GPRS)

Embedded Firmware:

Refers to the control algorithms (program) and or the configuration setting that an ES developer dumps into the Code (Program) memory of the ES.

The firmware may be developed using.

① Write the program in the high level languages like Embedded C/C++ using IDE. [IDE can contain an Editor, Compiler, Linker, Debugger, Simulator etc]

② Write the program in Assembly Language using the instructions supported by your application's target processor | controller.

Embedded Systems

UNIT-I (11)

The process of converting the program written in either a high level language or Assembly language - code - to - machine readable binary code is 'HEX File Creation'. The methods used for 'HEX File Creation' is different depending on the programming techniques used.

PCB and Passive Components

Printed Circuit Board (PCB), After finali-sing the components and the inter-connection among them, a schematic design is created and according to the schematic the PCB is fabricated. It also acts as a platform for testing your embedded firmware.

Microprocessor Architecture

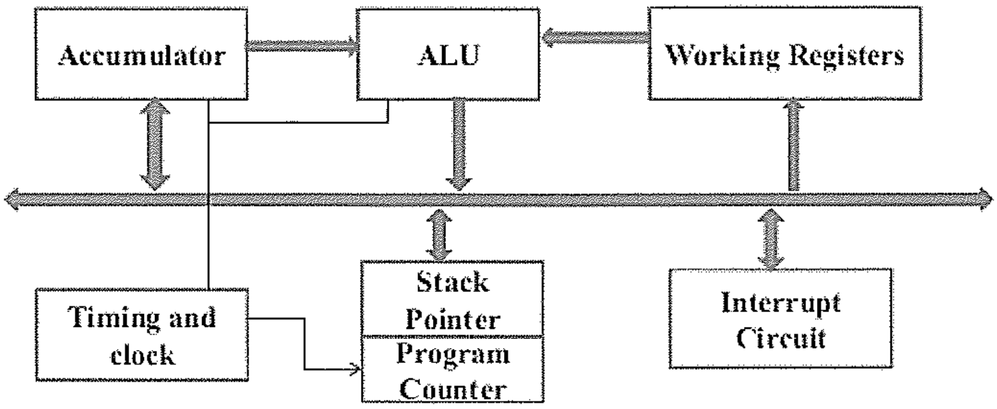
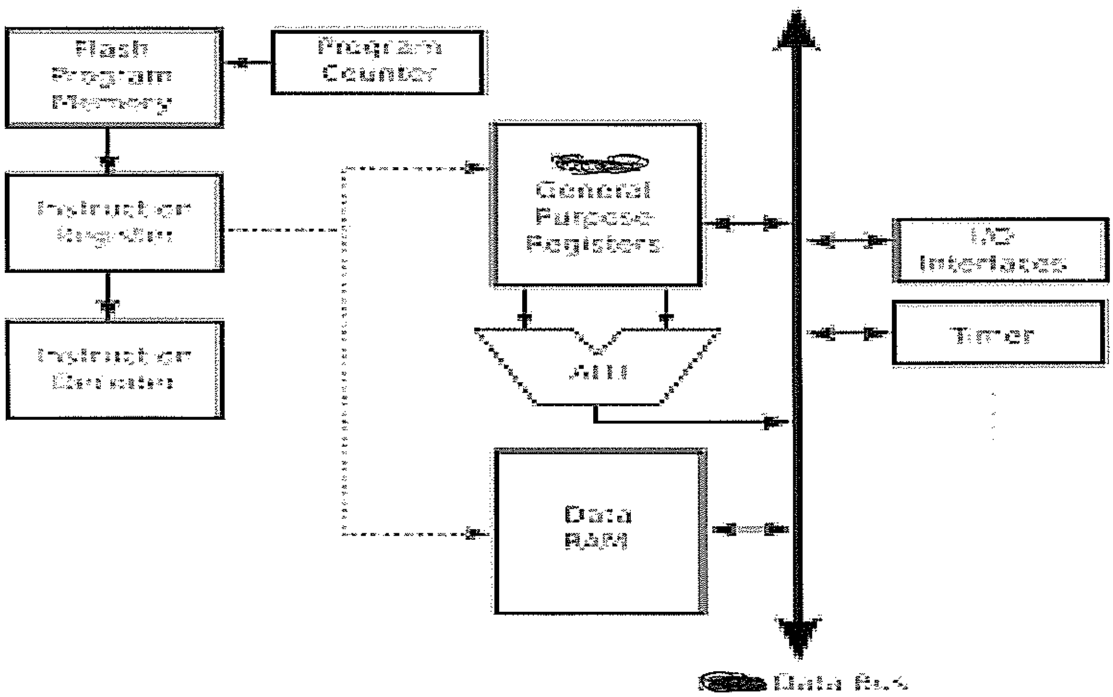


Fig: General Architecture of Microprocessor



General Architecture of Micro Controller