

I) Introduction to ES

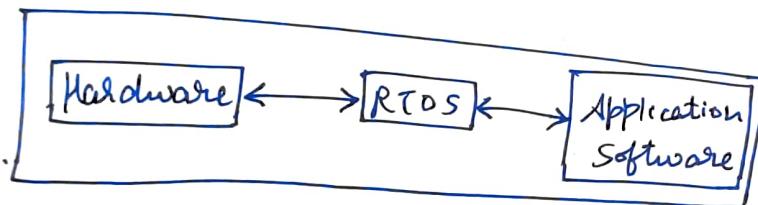
Embedded Systems - special purpose system in which computer is completely encapsulated by the device it controls

- ↓
- performs predefined tasks, usually with very specific requirements
- Built to function with little or no human intervention (often within a large system)
- Specifically designed keeping in consideration the tasks that need completion in the most efficient way

* An Embedded System has 3 components

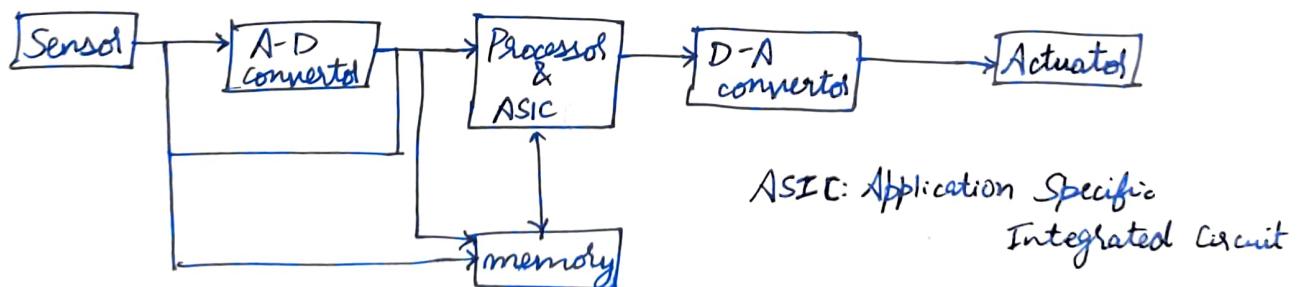
1. Hardware
2. Application Software
3. Real Time Operating System (RTOS)

supervises application software and provides mechanism to let the processor run a process as per scheduling by following a plan to control latencies.



RTOS defines the way a system works. It sets the rules during the execution of application program. Small scale E.S may not have RTOS.

Basic Structure of an embedded system



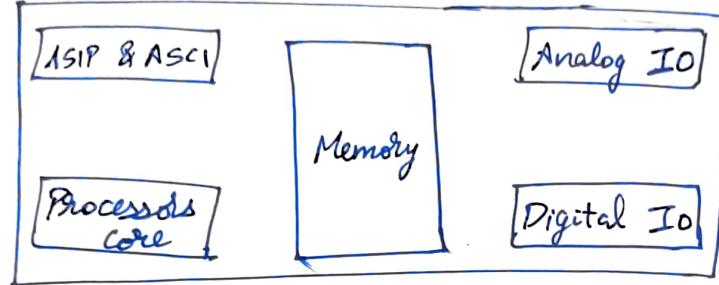
Sensor: measures physical quantity and converts it to an electrical signal which can be read by an observer / by an electronic instrument like A/D converter

Processor & ASIC: Processes data to measure the O/P and store it to memory

Actuator: Compares the O/P given by D-A converter to the actual expected output stored in it and stores the approved O/P

Features of an Embedded System

- * Single functional : performs specialized operation & does the same repeatedly
- * Tightly constrained :
- * Reactive & Real Time :
- * Microprocessors based :
- * Memory :
- * Connected peripherals :
- * HW - SW systems



Applications of Embedded System

- * Consumer electronics : TV, digital cameras, printers, PS4
- * Household appliances : Refrigerators, washing machines, AC
- * Medical equipment : MRI, CT, ECG machines
- * Automobiles : Fuel injection systems, anti lock braking systems, music & entertainment systems
- * Industrial applications : Assembly lines, systems for feedback
- * Aerospace : Systems for navigation & guidance, GPS
- * Communications : Routers & satellite phones

Advantages of Embedded System

- * Convenient for mass production. This results in low price per piece
- * Highly stable & reliable
- * Made for specific tasks
- * Very small in size, hence can be carried and loaded anywhere
- * Fast, uses less power
- * Optimize the use of resources available
- * Improve product quality

- (3)
- Characteristics of Embedded System**
- All Embedded systems are task specific. They do the same task repeated / continuously over their lifetime
 - * Embedded systems are created to perform task within a certain time frame. It must ∴ perform fast enough
 - * They have minimal or no user interface (UI)
 - * Some ES are designed to react to external stimuli & work accordingly
 - * ES are built to achieve certain efficiency levels. They are small, work with less power and not too expensive.
 - * ES cannot be changed/upgraded by users. Hence must rank high on reliability and stability. They are expected to function for long durations without the user experiencing any difficulties.
 - * Microcontroller / Microprocessors are used to design ES
 - * ES need connected peripherals to attach I/P and O/P devices
 - * Hardware of ES : used for security & performance
Software of ES : used for features
 - * Firmware of ES is preprogrammed and it is non alterable by the end user

Classification of ES

* Based on Generation

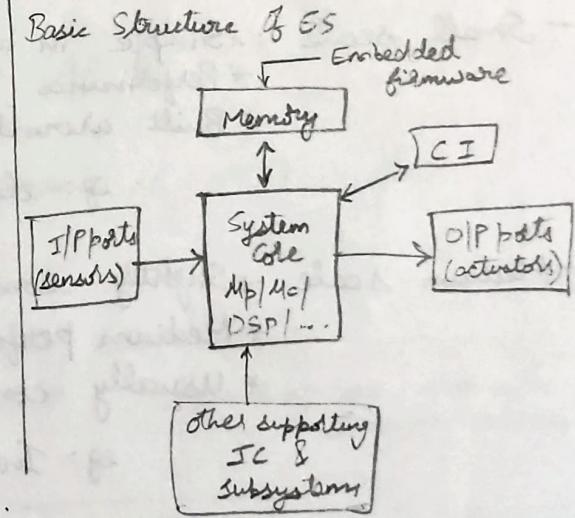
- i) 1G 8M_b 8M_c - Digital telephone keypads
- ii) 2G 16M_b 8M_c - SCADA systems
- iii) 3G 32M_b 16M_c - Robotics, media
- iv) 4G 64M_b 32M_c - Smart Phones

* Based on Complexity & Performance

- i) Small scale ES 8/16 - electronic Toy
- ii) Medium scale ES 16/32 - Industrial machines
- iii) Large scale ES 32/64 - mission critical applications

* Based on Deterministic Behaviour

- i) Deterministic/Non Deterministic
- ii) Real time ES: Hard/Soft - camera



→ Generation

- First Generation (1G):
 - * 8bit microprocessor (μp) & microcontrollers (μc)
 - * Simple in hardware circuit & firmware development
- eg: Digital telephone keypads
- Second Generation (2G):
 - * 16 bit μp & 8 bit μc
 - * More complex & powerful than 1G μp & μc
- eg: SCADA systems
- Third Generation (3G):
 - * 32bit μp & 16 bit μc
 - * Digital Signal Processors (DSPs), ASIC evolved
- eg: Robotics, Media, etc
- Fourth Generation (4G):
 - * 64 bit μp & 32bit μc
 - * System On Chips (SOC), Multicore processors evolved
 - * Highly complex & very powerful
- eg: Smart Phones

→ Complexity & Performance

- Small scale:
 - * Simple in application need
 - * Performance not time critical
 - * Built around low performance & low cost 8 or 16 bit $\mu p/\mu c$
- eg: electronic Toy
- Medium scale:
 - * Slightly complex in hardware and firmware requirement
 - * Medium performance & low cost 16 or 32 bit $\mu p/\mu c$
 - * Usually contain OS
- eg: Industrial machines
- Large scale:
 - * Highly complex hardware & firmware
 - * 32 & 64 bit RISC $\mu p/\mu c$ or PLDs or Multicore processors
 - * Response Time is critical
- eg: Mission critical applications

RISC : Reduced Instruction set computer

PLD : Programmable Logic Devices

Deterministic Behaviour

Classification applicable for "Real Time" systems

- * Task execution behaviour Real Time ES are divided into Hard & Soft
 - eg: Digital camera

→ Triggering

- * ES which are "Reactive" in nature can be based on triggering
 - Event triggered
 - Time triggered

eg: Thumb impression reader & ATM.

(III) Core Of ES, General Purpose & Domain specific processors

- * ES are domain & application specific and are built around a central core

- * The core of the ES falls into any of the foll. categories

1. General purpose^(GPP) and Domain Specific Processors

1.1 Microprocessors

1.2 Microcontrollers

1.3 Digital Signal Processors (DSPs)

2. Application Specific Integrated Circuits (ASICs)

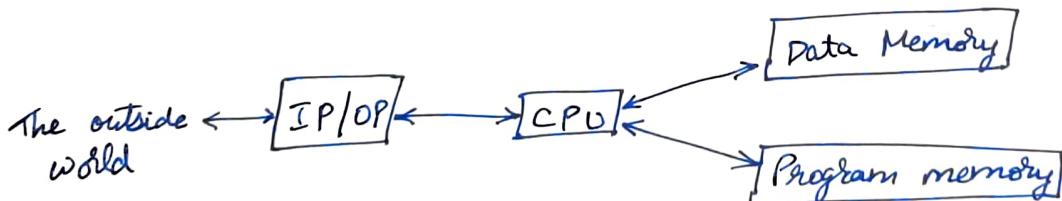
3. Programmable logic Devices (PLDs)

4. Commercial off-the shelf components (COTS)

1. General Purpose and Domain Specific Processor

- * 80% of the ES are processor/controller based

- * The processor may be 4b or 4c or DSP depending on domain & application



→ Microprocessors

- * is a Si chip representing a CPU

- * Mp is a dependent unit and it requires the combination of other hardware like memory, timer unit and interrupt controller etc for proper functioning

- * Architectures used for processor design - Harvard and Von-Neumann

→ Processors in a System

- * Processor has two essential units -
 - i) Program Flow Control Unit (CU)
 - ii) Execution Unit (EU)

* CU includes a fetch unit for fetching instructions from memory.

* EU has circuits that implement the instructions pertaining to data transfer operation & data conversion from one form to another.

* EU includes ALU and also circuits that execute instructions for a program control task that execute instructions for a program control task such as interrupt, or jump to another set of instructions.

* A processor runs the cycles of fetch and executes the instructions in the same sequence as they are fetched from memory.

→ Single Purpose Processors

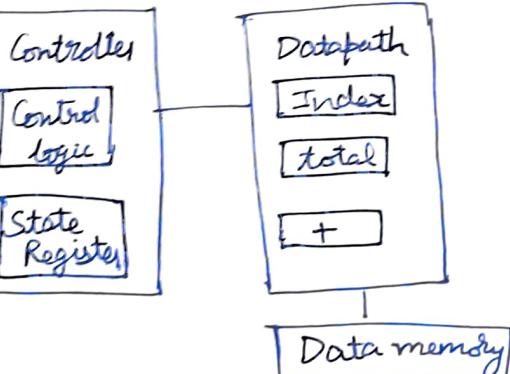
- * digital circuit designed to execute exactly one program aka coprocessor, accelerator / peripheral

* Features:

- * Contains only the components needed to execute a single program
- * No program memory

* Benefits:

- * Fast
- * low power
- * Small size



→ Application Specific Processors

* Programmable processor optimized for a particular class of applications having common characteristics

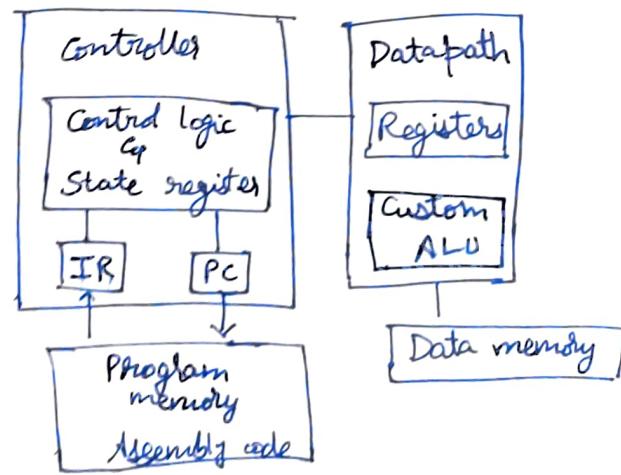
* Compromise b/w General purpose & Single purpose processors

* Features:

- * Program Memory
- * Optimized data path
- * Special functional units

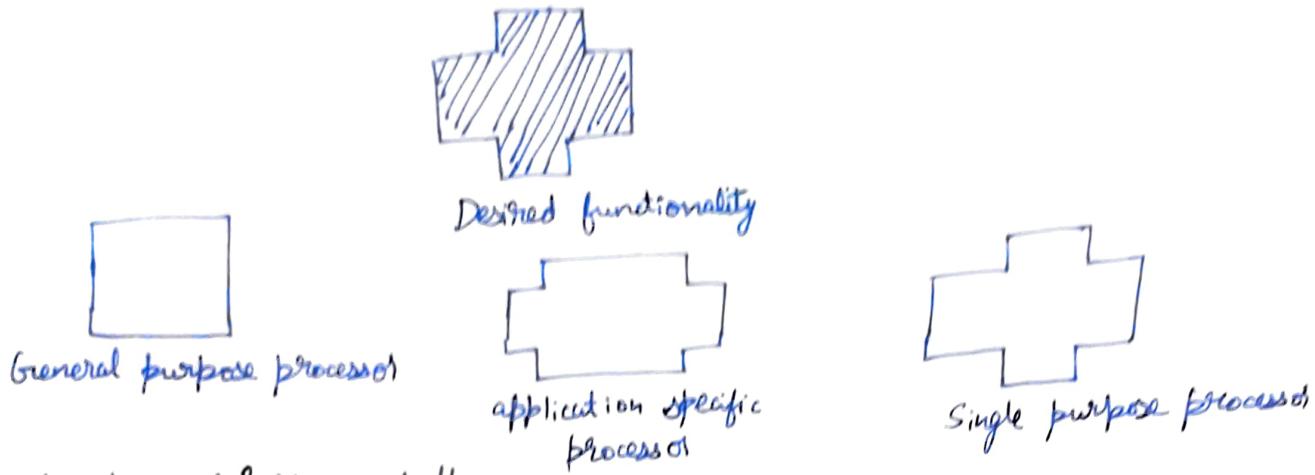
* Benefits

- * Some flexibility, good performance, size & power



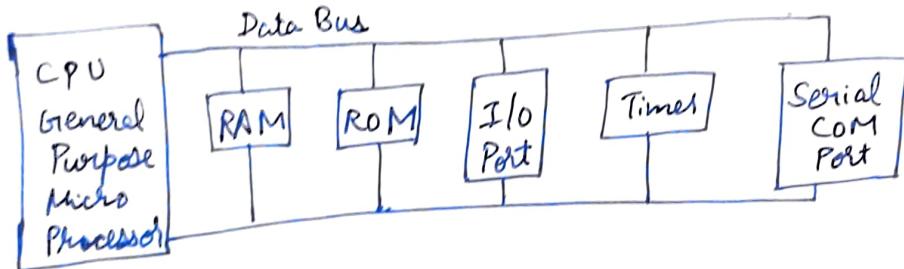
Processor Technology

Processors vary in their customization for the problem at hand



IV) Microprocessor & Microcontroller, #Microprocessor

- * Single VLSI chip having a CPU. In addition it may have other units
- * Earlier generation microprocessors' fetch & execute cycle was guided by a clock freq. of 1 MHz
Processors now operate at a clock freq. of 2 GHz



#Microcontroller

- * Single chip VLSI unit (also called microcomputer) which although have limited computational capabilities, possesses enhanced I/O capability and a number of on-chip functional units
- * MC are particularly used in ES for real time control applications with on-chip program memory and devices

CPU	RAM	ROM
I/O Port	Timer	Serial COM Port

F Microprocessor

- * Multitasking in nature.
Can perform multiple tasks at the same time
eg. computer - music with text editing
- * General purpose in design / application
- * RAM, ROM, I/O ports and timers can be added externally and can vary in numbers
- * It is dependent unit
- * Designers can decide the no. of memory or I/O ports needed
- * External support of external memory and I/O ports makes a microprocessor based system healthier and costlier
- * Limited Power Saving Options
- * External devices require more space and their power consumption is high

Microcontroller

- * Single task oriented
eg: washing machine is designed for washing clothes only.
- * Application oriented / Domain specific
- * RAM, ROM, I/O ports and timers cannot be added externally. These components are to be embedded on a chip and are fixed in numbers.
- * It is self contained unit
- * Fixed no. of memory or I/O makes a microcontroller ideal for a limited but specific task.
- * They are light weight and cheaper than a microprocessor
- * Lot of Power Saving Features
A microcontroller based system consumes less power and takes less space

→ Types of Architectures

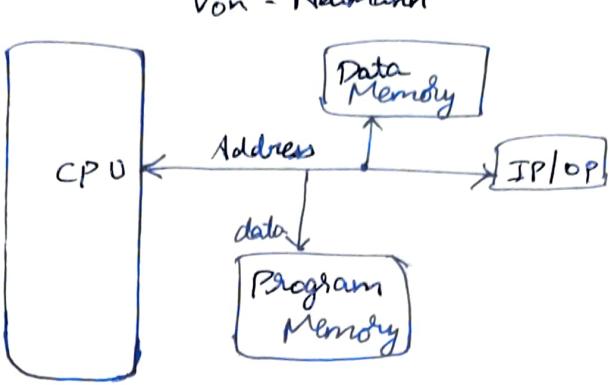
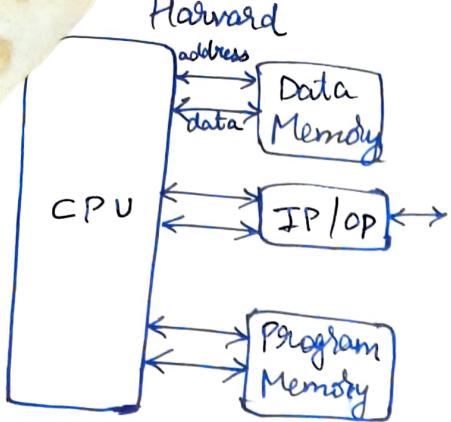
#

Harvard Architecture

- * It has separate buses for instruction as well as data fetching
- * Easier to pipeline so high performance can be achieved
- * Comparatively high cost
- * Since data and program memory are stored physically in different locations, no chances exist for accidental corruption of program memory

Von-Neumann architecture

- * It shares single common bus for instruction and data fetching
- * Low performance as compared to Harvard architecture.
- * It is cheaper
- * Accidental corruption of program memory may occur if data memory and program memory are stored physically in the same chip.



→ Types of instruction codes : CISC & RISC

Complex Instruction Set Computer

Reduced Instruction Set Computer

#

CISC

- * Larger set of instructions. Easy to program
- * Simple design of compiler, considering larger set of instructions
- * Many addressing modes causing complex instruction formats
- * Instruction length is ^{longer} variable
- * Higher clock cycles per second
- * Emphasis is on hardware
- * Control unit implements large instruction set using micro-program unit
- * Slower execution, as instructions are to be read from memory and decoded by decoder unit.
- * Pipelining is not possible

RISC

- * Smaller set of instructions. Difficult to program
- * Complex design of compiler
- * Few addressing modes, fix instruction format
- * Instruction length doesn't vary
- * Low clock cycles per second
- * Emphasis is on software
- * Each instruction is to be executed by hardware
- * Faster execution, as each instruction is to be executed by hardware
- * Pipelining of instructions is possible, considering single clock cycle.

1) # Digital Signal Processing (DSPs)

- * Powerful special purpose 8/16/32 bit microprocessor designed to meet the computational demands and power constraints of today's embedded audio, video and communication applications.
- * DSPs are 2-3 times faster than general purpose microprocessors in signal processing applications. Because of the architectural difference b/w DSP and general purpose microprocessors.
- * DSP implement algorithms in hardware which speeds up the execution whereas general purpose processor implement the algorithm in software and the speed of execution depends mainly on the clock for the processors.
- * DSP includes:
 - Program memory: Memory for storing the program to work on data.
 - Data memory: Memory for storing temporary variables and data/signal to be processed.
 - Computational engine: It performs signal processing in accordance with the stored program memory.
It incorporates many specialized arithmetic units and each of them operates simultaneously to ↑ execution speed.
It also includes multiple hardware shifters for shifting operands and saves execution time.
 - I/O unit: It acts as an interface b/w outside world & DSP.
It is responsible for capturing signals to be processed and delivering the processed signals.
eg: AV signal processing, Telecommunication & multimedia applications.
SOP calculation, convolution, FFT (Fast Fourier Transform), DFT (Discrete Fourier Transform) etc are some of the operations performed by DSP

Application Specific Integrated Circuits (ASIC) by - paulmeyer

- * ASIC is a microchip to perform a specific and unique application.
- * Because of using single chip it integrates several functions thereby reduces system development cost.
- * Most of the ASICs are proprietary (having some trade name) products, it is referred as Application Specific Standard Products (ASSP)
- * Single ASIC chip consumes a very small area in the total system. Thereby helps in the design of smaller system with high capabilities / functionalities.
- * The developer of such chips may not be interested in revealing the internal detail of it.

Programmable Logic Device(s) (PLDs) - F1MUX, DEMUX

- * It is an electronic component used to build digital circuits which are reconfigurable.
- * A logic gate has a fixed function but a PLD does not have a defined function at the time of manufacture
- * PLDs offer customers a wide range of logic capacity, features, speed, voltage characteristics.
- * PLDs can be reconfigured to perform any no. of functions at any time.
- * A variety of tools are available for the designers of PLDs which are inexpensive and help to develop, simulate and test the designs.
- * PLDs - 2 major types
 1. CPLD (Complex Programmable Logic Device):
CPLDs offer much smaller amount of logic up to 1000 gates
 2. FPGA (Field Programmable Gate Arrays):
offers highest amount of performance as well as highest logic density, most features

Commercial off-the-shelf components (COTS) - eg: MS office

- * It is used 'as is'
- * COTS components itself maybe developed around a general purpose (d) domain specific processor (d) ASIC (d) PLDs
- * Major advantage : they are readily available in the market, developer can cut down his development time to a great extent.

* Major drawback: manufacturers of COTS components may withdraw the product or discontinue the production of COTS at any time if rapid change in technology occurs

* Advantages of COTS:
1) Ready to use
2) Easy to integrate
3) Reduces development time

I/O devices: Sensors & Actuators

→ Sensor:
* Used for taking input
* It is a transducer that converts energy from one form to another for any measurement & control purpose.
eg: Temperature sensor

→ Actuator:
* Used for output
* It is a transducer that may be either mechanical or electrical which converts signals to corresponding physical actions
eg: LED

LED:
* p-n junction contains a CATHODE and ANODE
 |
 | connected to -ve end of power supply | connected to +ve end of power supply
* Max current flowing thru LED is limited by connecting a resistor in series b/w power supply & LED

Communication Interfaces

23rd January, 2023 (13)

1. Onboard CI

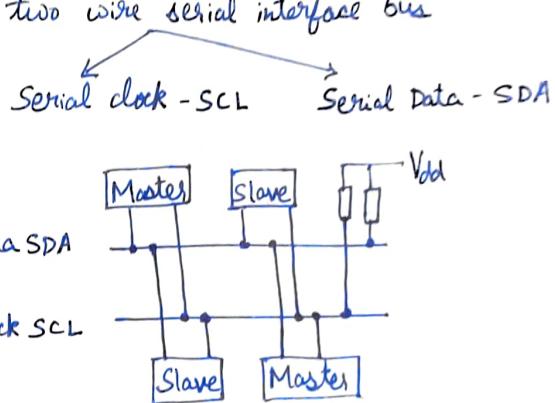
- * used for internal communication of the ES i.e. communication b/w different components present on the system

On board Communication Interfaces Examples

- * Inter Integrated Circuit (I₂C) - synchronous, bi-directional, half duplex, two-wired
- * Serial Peripheral Interface (SPI)
- * Universal Asynchronous Receiver Transmitter (UART)
- * I-wire interface
- * Parallel interface
 - eg: I₂C
 - * It is synchronous
 - * Bidirectional, half duplex, two-wire serial interface bus

→ I₂C

- * SCL generates synchronization clock pulses
- * SDA transmits data serially across devices
- * I₂C is a shared bus system to which many devices can be connected
- * Devices connected by I₂C can either act as master/slave
- * Master device: responsible for controlling communication by initiating/terminating data transfer.
- * Slave device: wait for commands from master and respond to those commands.



External or Peripheral communication interfaces

Examples

- * RS-232 C & RS-485 → asynchronous, full duplex, four-wired
- * Universal Serial Bus (USB)
- * IEEE 1394 (Fire wire)
- * Infrared (IRDA)
- * Bluetooth
- * Wi-Fi
- * Zig Bee
- * General Packet Radio Service (GPRS)

→ RS 232 & RS 485

- * It is wired, asynchronous, serial, full duplex communication.
- * RS 232 interface was developed by EIA (Electronics Industries Associates)
- * RS 232 is the extension to UART for external communications
- * RS 232 logic levels use: +3 to +25V to signify a "Space" (Logic 0)
-3 to -25V to signify a "Mark" (Logic 1)
- * RS 232 is used only for point to point connections
- * It is susceptible to noise and hence is limited to short distances only.
- * RS 485 supports up to 32 transmitters and 32 receivers.

→ Purpose of ES

- * Data collection/Storage/Representation
- * Data Communication
- * Digital Signal Processing
- * Monitoring
- * Control
- * Application Specific User Interface

eg: Digital Camera

eg: wireless modules: Bluetooth, WiFi wire-line modules: USB, TCP, IP

eg: Digital hearing aid

eg: Electrocardiogram machine, digital CRO

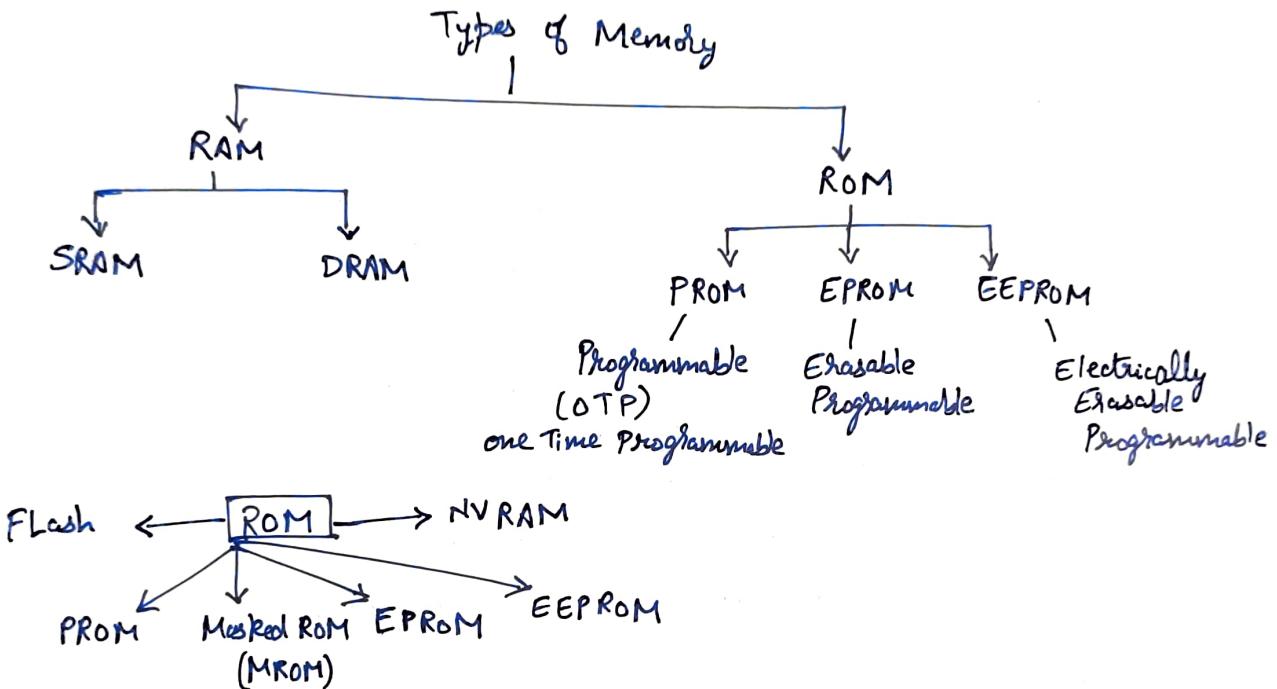
eg: Air Conditioner System

eg: Mobile Phone

→ Summary On ES

- * A system is an arrangement where its entire component assemblies work according to the specific defined rules
- * An ES is a combination of computer software & hardware which is either fixed in capability or programmable
- * eg: of ES is laser printer which manages various aspects of the printing
- * An ES requires real time performance
- * Reliability measure is the survival probability of the system when the function is critical during run-time
- * Fault-Tolerance is the capability of a computer system to survive in the presence of faults
- * ES must meet various timing and other constraints
- * Flexibility is building systems with built-in debugging opportunities which allow remote maintenance.
- * Portability is the measure of ease of using the same embedded software in various environments
- * A microcontroller is a single-chip VLSI unit which is also called microcomputer.
- * Applications of ES: Robotic Science, Medical, Automotive, Networking, Home devices, Automobiles & Industrial control
- * Major advantage of ES: it is able to cover a wide range of environments

- Memory is required for holding data temporarily during certain operations
- * On chip memory : built in memory
- * off chip memory : external memory connected with controllers / processor for storing controllers algorithm
- * Program Storage Memory
- * Also called as code storage memory of an ES - stores the program instructions.
It retains its contents even after power is turned off. (non volatile)



→ MROM

- * One Time Programmable device
- * Uses hardwired technology to store data
- * This is low cost for high volume of production
- * Good for storing embedded firmware for low cost embedded device
- * Limitation: inability to modify the device firmware against firmware upgrades.

→ PROM/OTP

- * The end user is responsible for programming this memory
- * This memory consists of polysilicon/nichrome wires functionally viewed as fuses.
- * Fuses not blown/burned - logic '1'
- * Fuse blown/burned - logic '0'
- * OTP is used in commercial production of ES

→ EEPROM

- * EEPROM gives flexibility to re-program the same chip
- * Stores bit information by charging the floating gate of FET to a high voltage
- * Contains a quartz crystal window, if exposed to UV rays for 20-30 mins entire memory will be erased.
- * Limitation: Erasing the memory using UV rays is tedious & time consuming process.

→ EEPROM

- * Information contained in this memory can be erased and re-programmed in circuit using electrical signals at register/Byte level.
- * It provides greater flexibility for system design
- * Limitation: capacity (only a few kilobytes)

→ Flash

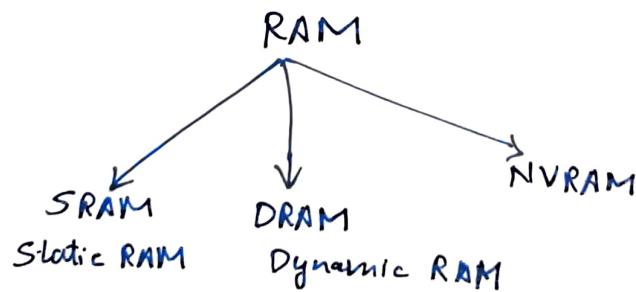
- * Latest ROM technology which combines - re-programmability of EEPROM + high capacity of standard ROMs
- * It is organised as sectors/pages.
- * Stores info in array of floating gate of an MOSFET
- * Each sector is erased before re-programming and it is done at sector level/page level without affecting sectors/page
- * Typical erasable capacity = 1000 cycles per second

Eg: W27C512 from WINDBOND (64 kB FLASH memory)

→ NVRAM (Non Volatile RAM)

- * RAM with battery backup
- * Lifespan of NVRAM = 10 years

Eg: DS1644 from Maxim/Dallas (32 kB NVRAM)



- * RAM is data/working memory of controller/processor where it can read from it & write to it
- * Volatile
- * It is direct access memory

RAM (SRAM)

(17)

Fastest RAM (Resistive Networking & switching capabilities)

- * Made up of flip flops and stores memory in the form of voltage
- * realised using 6 transistors (MOSFETs)

 └ 4 for building latch part of memory cell

 └ 2 for controlling the access

- * Major limitation: Low capacity & High cost.

→ Dynamic RAM (DRAM)

- * Stores data in the form of charge

- * Advantage: High density & Low cost

- * Disadvantage: Since it is stored as charge, it will get leaked off with time, so refreshing is needed.

- * Special circuits called DRAM controllers are used for refreshing the operation.

SRAM

- * Made up of 6 CMOS Transistors (MOSFET)

- * Doesn't require refreshing

- * Low capacity

- * Fast in operation.

Typical access time is 10ns

DRAM

- * Made up of a MOSFET and a capacitor

- * Requires refreshing

- * High capacity

- * Slow in operation.

Typical access time is 60ns

- * Less expensive

- * Write operation is faster than read operation

→ NVRAM

- * RAM with battery backup

- * Contains static RAM based memory

- * Lifespan = 10 years

VIII) Introduction to ARM processor

* ARM : Advanced RISC machines

/ Reduced Instruction Set computing

- * ARM is one of the most licensed and thus widespread processor cores in the world
- * Used especially in portable devices due to low power consumption and reasonable performance.
- * Used in PDA (Personal Digital Assistants), cell phones, multimedia players, handheld game console, digital TV and cameras.
- * ARM has several processor cores that are grouped into no. of families based on they are implemented with.
- * The architecture of ARM processors has continued to evolve with every family.

- No. of physical features that have driven the ARM processor design

- * Portable ES require some form of battery power. The ARM processor has been specifically designed to be small to reduce power consumption & extend battery operation - essential for applications such as mobile phones and personal digital assistants (PDA)
- * High code density is another major requirement since ES have limited memory due to cost/physical size restrictions. High code density is useful for applications that have limited on board memory such as mobile phones & mass storage devices.
- * ES are price sensitive and use slow and low cost memory devices.

* Another important requirement is to reduce the area of the die taken up by the Embedded processor.

For a single chip soln. the smaller the area used by embedded processor
the more space available for specialized peripherals
↓
reduces the cost of design & manufacturing since fewer discrete chips are reqd. for end product

- * ARM has incorporated hardware debug technology within the processor so that software engineers can view what's happening while the processor is executing code. With greater visibility → software engineers can resolve issues faster.

SC characteristics

* Instructions : RISC processors have reduced no. of instruction classes.
Each instruction is of fixed length.
CISC processors - instructions are often of variable size and take many cycles to execute

* Pipeline : The processing of instructions is broken down into smaller units that can be executed in parallel by pipelines.

There is no need for an instruction to be executed by a microprogram called microcode as on CISC processors

* Registers : RISC machines have a large general purpose registers set.
↳ act as fast local memory store for all data. / can contain either data/address

* Load-store architecture :

Load : Data is put into registers

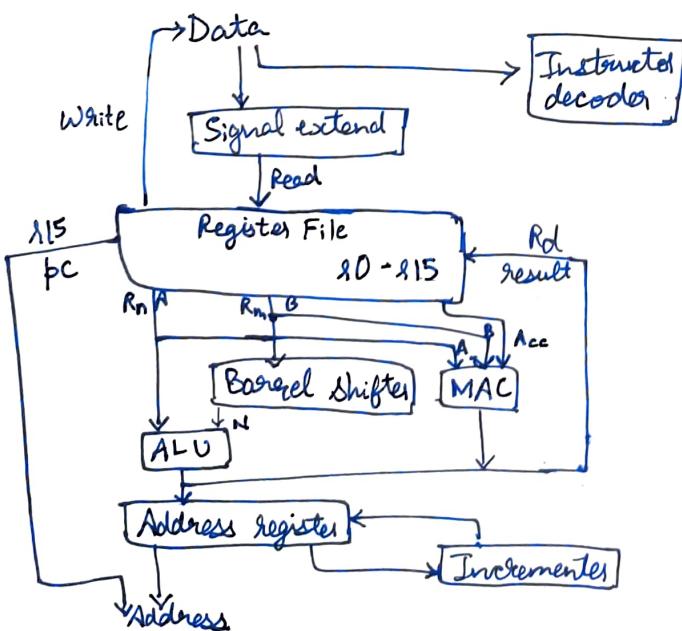
Store : Data from registers to memory location

→ Features of ARM7

- * 32 bit processor
- * 32 bit ALU
- * 32 bit databus
- * 32 bit instructions
- * 32 bit address bus
- * Von Neumann Architecture
- * Three stage pipelining

IX) ARM core data flow Model

29th January, 2023



enters the processor core through data bus

Instruction Decoder translates instructions before they are executed

* Data items are placed in register file-storage bank made of 32 bit registers.

* Sign extend converts signed 8-bit & 16-bit numbers to 32-bit values.

* Two source registers - R_n & R_m

Single destination register - R_d

* Source operands are read from register file using internal buses A & B. and these buses are connected to basic processing units

* ALU takes register values R_n & R_m and computes the result.

* Barrel shifter computes the result for any no. of shifts within a clock cycle.

* This is achieved as it is combinational logic (not sequential)

* Multiply-Accumulate Circuit (MAC) is used to perform both multiplication & addition.
eg: Matrix addition & multiplication

* Load and store instructions use the ALU to generate an address to be held in the address register and broadcast on the address bus.

* As ARM is Von-Neumann architecture, so same bus is used to load instructions and data. Hence input data bus entering the processor core is connected to

i) Instruction Decode Block

ii) Direct Register bank

iii) Sign Extend hardware block (which again is connected to register file)

→ i) Instruction Decoder translates instructions before they are executed

→ ii) Data items are placed in register file storage bank made up of 32-bit registers. Since ARM core is a 32-bit processor, most instructions treat the registers as holding signed/unsigned 32-bit values.

→ iii) Sign extend hardware converts signed 8-bit & 16-bit numbers to 32-bit values as they are read from memory and placed in a register.

Basic Processing Units : i) ALU (Arithmetic Logic Unit)

ii) Barrel Shifter

iii) MAC (Multiply & Accumulate Unit)

→ i) ALU

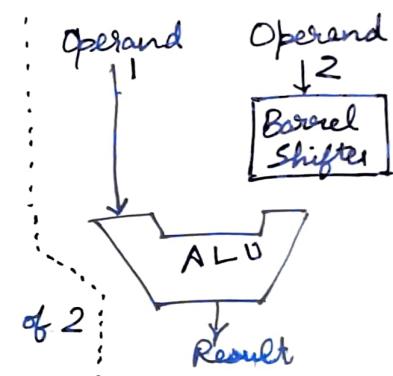
- * ALU takes values R_n and R_m and computes results
- * ALU performs add, sub etc (Mathematic operations)
OR/AND etc (Logical operations) on the data present in data registers

eg: ADD R3, R2, RI $\Rightarrow R3 = R2 + RI$

- * Load and store instructions use the ALU to generate an address to be held in address register and broadcast on address bus.

→ ii) Barrel Shifter

- * A barrel shifter is a digital circuit that can shift a data word by a specified no. of bits in one clock cycle
- Operand 2 (B) can be directly loaded (o)
- Shifted by specified no. of times
- Can achieve fast multiples/division by a power of 2



→ iii) MAC

- * MAC is used to perform both Multiply and Add
- * The result of any operation can be written back to registers bank
- * Supports basic summation operation on data present in registers

eg: Matrix addition & multiplication, summation operations

$$\begin{bmatrix} 1 & 4 & 6 \end{bmatrix} \cdot \begin{bmatrix} 2 & 3 \\ 5 & 8 \\ 7 & 7 \end{bmatrix} = \begin{bmatrix} 64 & 89 \end{bmatrix}$$

$$(1 \cdot 2) + (4 \cdot 5) + (6 \cdot 7) = 64$$

result = 2 → Acc
(accumulated)

Address register:

- * Contains the address (location) from which data/instructions has to be fetched.
- * The register is connected to an incrementer unit.

Registers and Current Program Status Register

- ARM processors like all RISC processors uses a load-store architecture.
- * This means it has two instruction types for transferring data in and out of the Processor.
 - * There are no data processing instructions that directly manipulate data in memory.
 - * Data processing is carried out solely in data items that are placed in the register file which is a storage bank made up of 32 bit registers.

* Most instructions treat the registers as holding signed/unsigned 32 bit values.

* The sign extension hardware converts signed 8 bit & 16 bit into 32 bit values as they are read from memory and placed in a register.

- ARM has 37 registers in total all of which are 32-bits long.

- 1 dedicated program counter
- 1 dedicated current program status register
- 5 dedicated saved program status registers
- 30 general purpose registers

→ 20 registers are hidden from a program at different times
These registers are called banked registers

They are available only when the processor is in a particular mode

* Stack Pointer: stores the head of the stack in the current process mode

* Link Register: core puts the return address when it calls a subroutine

* Program Counter: contains the address of the next instruction to be fetched

* In ARM state, the registers R0 - R15 are orthogonal by the processor.

* Any instruction that you can apply to R0 can equally well apply to other registers.

Current Program Status Register (CPSR)

* 32 bit register

* contains the present state of an internal operation

* ARM uses CPSR to monitor and control internal operations.

Processor Modes

1 determines which registers are active and the access rights to the CPSR register itself.

* ARM has 7 operating modes

1. User (Unprivileged mode under which most tasks run)
2. FIQ (entered when a high priority (fast) interrupt is raised)
3. IRQ (entered when a low priority (normal) interrupt is raised)
4. Supervisor (entered on reset and when a software interrupt instruction is executed)
5. Abort (used to handle memory access violation)
6. Undefined (used to handle undefined instructions)
7. System (privileged mode using the same registers as User mode)

→ Non-Privileged modes

- * Read access to control field to CRSP
- * Read-write access to conditional flags
- * User

→ State and instruction sets

- * The state of the core determines which instruction set is being executed. 3 instruction sets :
 - i) ARM-Arm state
 - ii) Thumb-Thumb state
 - iii) Jazelle-Jazelle state