unit-1

Embedded system: It is defined as Electro mechanical system which is designed to perform a specific function and is combination of both hardware and software is called "Embedded system".

Ex: washing machines, Electronic toys, DVD player etc

* Classifications of embedded system

1 classification based on generation

=> **First generation**:

-> Embedded systems build around 8-bit microprocessors and 4-bit microcontrollers

Ex: Stepper motor, keypads

=> **second generation**:

-> Embedded system build around 16-bit microprocessor and 8/16 bit microcontroller

Ex: SCADA.

=> **Third generation**:

-> Embedded systems build around 16/32 bit microprocessor/microcontroller, application digital signal processor.

Ex: Robotics, Networking etc.

=> **Fourth generation**:

-> Embedded systems build around system on chips, high performance, tight integration etc.

Ex: Smart phones, MID's etc

2. classification based on complexity:

Small scale: The embedded systems build around low performance low cost 8/16-bit microprocessor and microcontroller.

-> It is suitable for simple applications

medium scale: The embedded system build around medium performance and low cost 16/32 bit microprocerors and Microcontrollers (or) DSP's

large scale: The Embedded systems build around high performance 32/64 RISC processor and controller.

3. Based on deterministic behaviour

-> It is applicable for Real time systems.

i Soft Real time system : missing a deadline may not be ~~created~~ critical, and can be tolerated.

ii Hard Real time system : Missing a task execution time deadline can have loss of finance and human lifes.

4. Based on Triggering

Event Triggered : Activities within the system depends up ~~on~~ occurance of events.

Time Triggered : Activities within the system follows statically computed schedule

* Applications of Embedded systems

-> Household Appliances - Television, DVD players, washing machine, Fridge, oven etc.

-> Home Automation - Air conditioners, Splinkers, detection alarms, Fire alarms.

-> consumer Electronics : cameras, camcorders etc

-> Telecom - cellular Telephones, Telephone switches etc

-> computer peripherals printers, scanners, etc

-> Health care : Different scanners, Ecgs, EEGs etc

-> Card Readers Barcode, smart card readers etc

-> Banking : ATMs, ~~co~~ currency counters etc

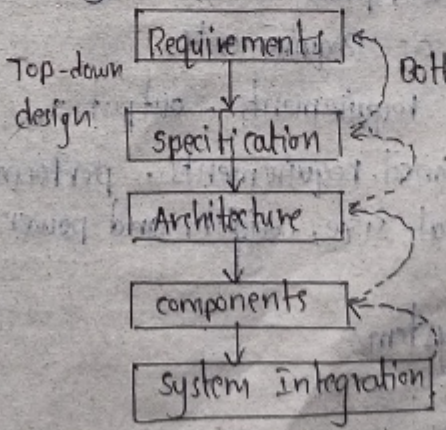
-> Measurement digital multimeters, digital CROs

* Embedded system design process :-

-> There are different steps involved in embedded system design process.

-> These steps depends on the design methodology.

-> Below figure shows the steps in design process



-> Design process steps: Requirements, specification, Architecture, components, system Integram.
-> steps from top to down is top-down design steps.
-> steps from bottom to up is Bottom-up design steps
-> manfacturing cost.
-> performance
-> power consumption.

1. Requirements :

-> Require may be functional or Non-functional.
-> we must capture the basic functions of embedded systems.
-> Informal descriptions gathered from customer is known as Requirements.
-> Functional requirements - output ...
-> Non-functional requirements - performance, cost, physical size, weight, and power consumption.

Ex: Gps system.

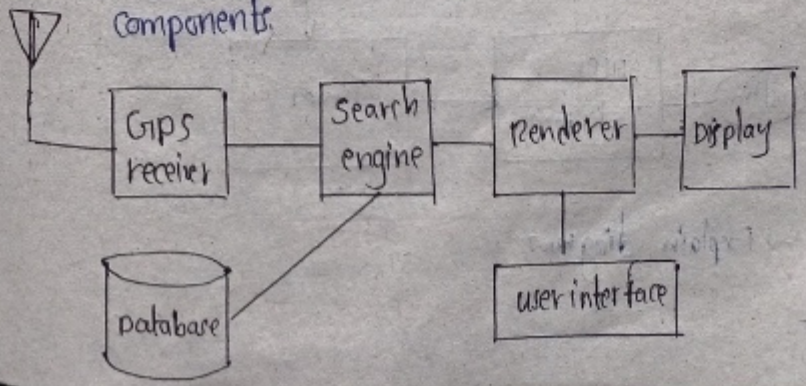
2. Specifications :

-> Requirements is refined into a specification.
-> Specification serves contract blw customers and architects.
-> Specification must be specific, understandable.

Ex: Consider Gps system.
- Map data
- Background action.

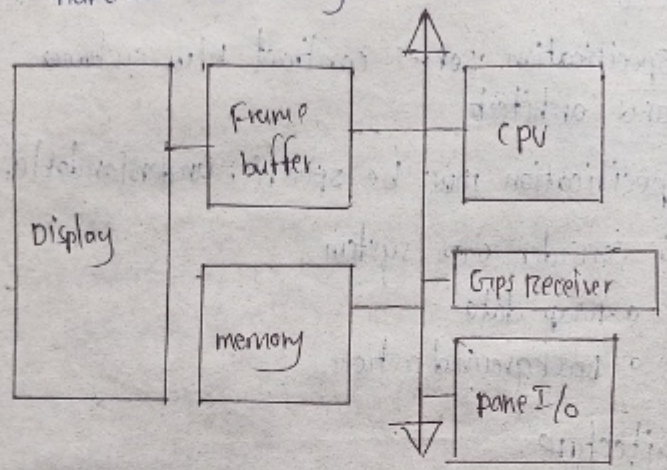
3. Architecture

-> Specification describes the only functions of the system.
-> Implementation is described by architecture.
-> Architecture is a plan for overall structure of the system.
-> Architecture later used for design of components.



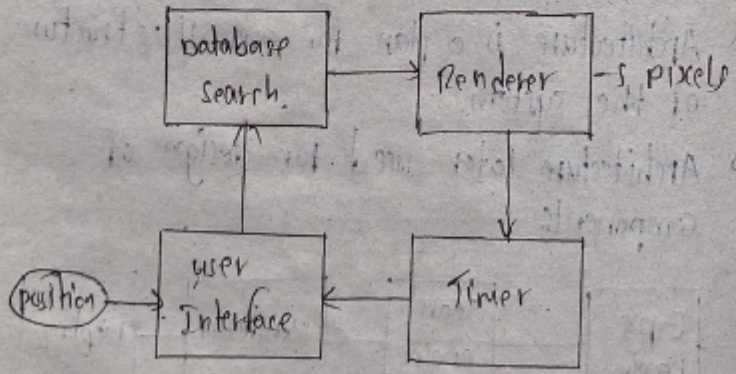
-> Above fig shows block diagram of moving map

-> Hardware block diagram.



-> Explain diagram.

-> Software Block diagram



-> Explain diagram

④ Components

-> Here components means both hardware and software components.

-> The architecture tells us what components we need

-> some components will be ready made (CPU, memory chips).

⑤ System Integration:

-> The ~~compents~~ components built are put together and see how system works.

History of embedded system

-> First modern embedded system:

"Apollo guidance computer (AGC)"

developed by 'charles stark draper' at MIT instrumentation laboratory.

-> It has two modules namely

1. command module
2. Lunar Excursion module

-> RAM size 256, 1k, 2k words

-> ROM size - 4k, 10k, 36k words.

-> clock freq is 1.024 MHz.

-> user interface - display, keyboard

purpose of Embedded systems

-> purpose of Embedded systems

1. Data collection/storage/Representation:

-> Data collection is used to storage, Analysis, manipulation and transmission.

-> Here data can be Text, voice, Image, audio & video,

-> Data is either analog or Digital.

-> Embedded system convert analog to digital.

Eg: Digital camera.

② Data communication:

-> Here Data communication is wired (or) wireless

-> Data transmitted from Analog (or) digital.

-> Data organised from satellite to home network system.

Eg:- Network hubs, Routers etc

③ Data processing:

-> In embedded systems Data (signal) processing like speech coding, audio-video codec etc.

-> In digital signal processors (DSPs).

④ monitering:

-> Embedded systems coming under monitering purpose like medical applications like ECG machine, 2D Eco machine etc.

-> Measuring Instruments like digital CRO, digital multimeter, etc.

-> Monitering is also known as controlling.

⑤ Control:

-> Embedded systems with control function are used for change input variables.

-> ~~Embedde~~ Embedded systems with control function consist of both sensors and Actuators.

-> Sensors are connected at I/p port.

-> Actuators are connected at o/p port.

Ex: Air conditioner.

⑥ Application specific user Interface:

-> Embedded systems which are designed for specific application.

-> Buttons, keyboards, switches.

-> mobile phone — keypad, Touch screen.

* characteristics of Embedded systems:-

1. Application and domain specific:

-> Each embedded system has specific function to perform, they are developed in such a manner.

-> They cannot be used for any other purpose.

② Reactive and Real time:

-> Embedded systems have constant interaction ~~between~~ with real time with sensors.

-> Any changes in the real world in real time captured by sensors.

Ex: Antilock Brake systems.

③ Microprocessor Based:

-> It must be microprocessor or microcontroller based.

④ Small size and weight:

-> product (size, weight, size) etc. is a important factor for chosing a product.

-> products with small size and weight are convenient to handle.

⑤ Memory :

→ Embedded systems must have memory in ROM.

⑥. HW-SW systems

→ Hardware is used for performance

→ software is used for flexibility.

⑦. connected :

→ Embedded systems must have connection b/w I/p and o/p devices

8. Safety critical

→ Must not endge endanger human life, environment

Unit-2

* Core of the Embedded systems :-

-> The core of embedded system falls into any one of the following.

①. General purpose and Domain specific processors.

-> microprocessors.
-> microcontroller.
-> Digital signal processors.
-> Almost 80% of embedded systems are processor / controller based.
-> The ~~ma~~ processor may be microprocessor / microcontroller (or) digital signal processor.

o **Microprocessor** :- A microprocessor is a very small processing unit inside the CPU. is called. microprocessor.

(or)

Microprocessor is a multipurpose programmable device that accepts digital data as input, processes it according to instructions and stores ~~it~~ in it's memory & provide result as output is called. Microprocessor.

-> Microprocessor is a dependent unit.
-> Developers of microprocessors.
- Intel - Intel 4004
- Intel - Intel 4040.
- Intel - Intel 8008.
- Intel - Intel 8080.
- motorola - motorola 6800.

- Microcontroller :- A microcontroller in a embedded systems are small ICs that controls a single process is called Microcontroller.

-> Microcontroller ~~is a~~ includes processor, memory, I/p - o/p peripherals on single chip.

-> Microcontroller are low cost and readily available in market.

Difference b/w microprocessor & Microcontroller.

Microprocessor.	Microcontroller
-> Microprocessor are used for big Applications.	-> Microcontrollers are used to execute a single task within an application.
-> Requires ~~high~~ power consumption.	-> Requires ~~high~~ low power consumption.
-> Requires more instructions	-> Requires less instructions.
-> circuit design is complex.	-> circuit design is simple.
-> It is high cost.	-> It is low cost.
-> It is high speed.	-> speed depends on architecture.
-> It have less registers.	-> It have more registers.
-> Memory & I/p - o/p components are external to it.	-> Memory & I/p-o/p components are internal to it.
-> RAM/ROM are absent.	-> RAM/ROM are present.
-> center of computer system.	-> center of Embedded systems.

- Digital Signal processors :-

-> DSP - Digital signal processors.

-> DSPs are powerful special purpose microprocessor.

-> DSPs are available in 8/16/32 bits.

-> DSPs are designed for today's embedded audio, video, and communication applications.

-> DSPs are 2-3 times faster than general purpose microprocessors.

-> speed of DSPs is more than general purpose microprocessors.

-> DSPs can perform speed operations like addition, substraction, multiplication, division etc.

-> ~~DSPs can perform~~ DSPs have following key units.

- program memory.
- Data memory
- I/o unit
- computational Engine.

Ex: Audio - video, telecommunication, multimedia etc.

Harvard Architecture.	von-Neumann architecture.
-> It is modern computer architecture.	-> It is Ancient computer Architecture.
-> seperate buses are used for transfering data & instruction.	-> common bus for transfering data & instruction.
-> single clock cycle is required to execute instruction.	-> Two-clock cycles are required to execute instruction.
-> It is ~~cheaper~~ costly.	-> It is cheaper.
-> It is complex to use.	-> It is not complex to use.
-> pipeline strategies are possible	-> pipeline strategies are not possible.
-> speed is high.	-> speed is low.

RISC	CISC.
-> RISC- Reduced Instruction set computer.	-> CISC- complex Instruction set computer.
-> It uses more registers.	-> It uses less registers.
-> RAM usage is more.	-> RAM usage is less.
-> code size is more.	-> code size is less.
-> It is more expensive.	-> It is less expensive
-> It consumes low power.	-> It consumes high power.
-> It has less addressing modes	-> It has more addressing modes
-> Emphasis on software.	-> Emphasis on hardware.
-> External memory doesn't required	-> External memory is required.
-> It has fixed instruction format	-> It has variable instruction format.

* Application Specific Integrated circuits:

-> ASIC- Application specific integrated circuits
-> ASIC is specially built for specific purpose.
-> compared with other device, ASIC has improved speed and reliability.
-> ASIC consumes less power.
-> ASIC can be found in most electronic devices.
-> The three different catgories of ASICs,
- Full-custom
- semi-custom
- programmable

o Full custom: It is a design methodology useful for IC's.

-> In full-custom circuit layout resistors transistors, capacitors, analog circuits are placed.
-> It have good performance, high flexibility.

o. semi-custom: semi-custom is the alternative to full-custom.

-> It is divided into two types
(i) Gate array based.
(ii) standard cell based.

o programmable: These are not custom based.

-> It is divided into two types
(i) FPGAs
(ii) PLDs.

* PLD's

-> PLD - stands for programmable logic devices.

Def: programmable logic device is an electronic component, used to build reconfigurable digital circuits is called PLD.

-> PLDs are based on re-writable memory and reprogrammed to change design.

-> programmable logic devices - FPGA and CPLDs

-> Feild programmable Gate arrays and complex programmable logic devices are two major types of PLDs

(i) FPGA: Feild programmable gate array.

-> It is an integrated circuit and thousand times faster than processors

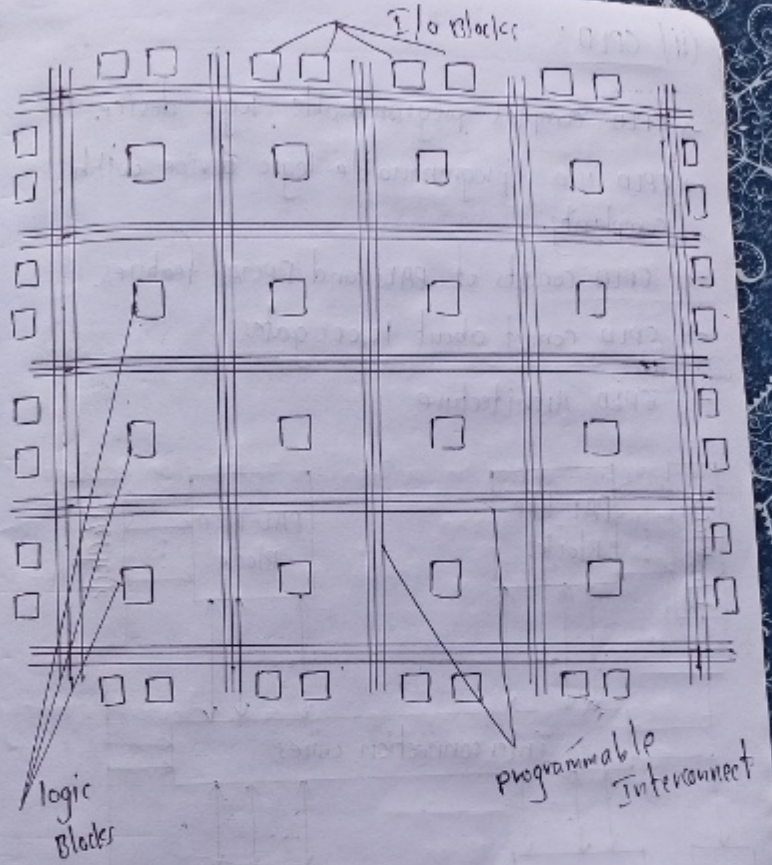
-> Circuit consist of CLB's, memory, and other elements

-> FPGA offers most features

-> It have highest performance.

-> Density of system gates is from 1K to 500K.

-> FPGA architecture,



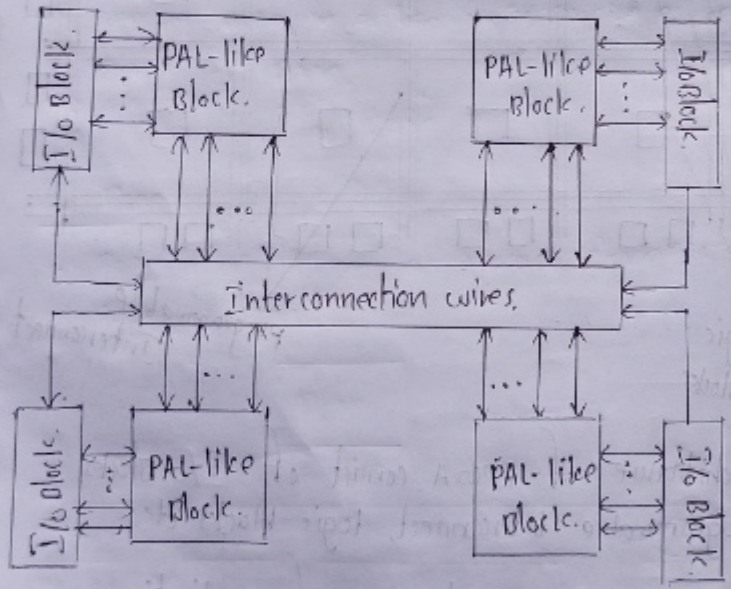
-> Architecture of FPGA consist of I/o blocks, programmable interconnect, logic blocks etc.

-> FPGAs used in wide range of Applications

- instrumentation
- telecommunications
- DSPs, etc.

(ii) CPLD :-

-> CPLD - complex programmable logic device.
-> CPLD is a programmable logic device with complexity.
-> CPLD consists of PALs and FPGAs features.
-> CPLD consist about 10,000 gates.
-> CPLD Architecture.



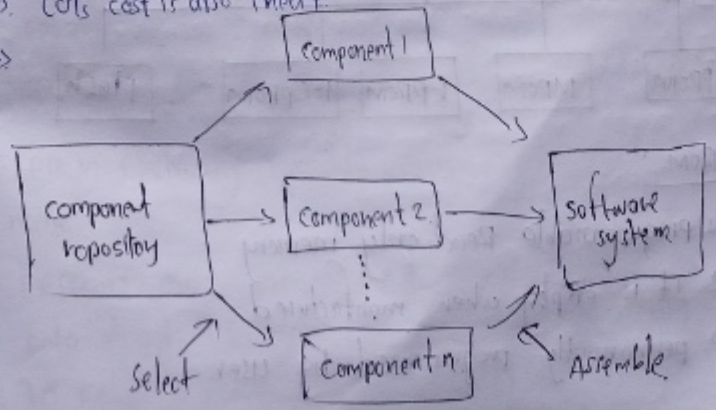
-> CPLD IC consist of I/o Block, PAL-like block, Interconnection wires etc.
-> Interconnection wires plays vital role in CPLDs.

Advantages of PLDs

-> Faster
-> low power consumption.
-> low cost.
-> High reliability.

* COTs

-> COTs - Commertial -off - the shelf
-> COTs remains "as-is".
-> COTs products are easy to design, integration and interoperability.
-> Typical examples of for the COTs products are Remote controlled Toys.
-> COTs includes High performance, high freq, high bandwidth, UV/IR detectors etc.
-> A COTs components contains.
  - GPP
  - ASIP
  - ASIC
  - ASSP
  - PLD.
-> The maiur advantage of COTs are they are readily available in market.
-> COTs cost is also cheaper.
->



Advantages

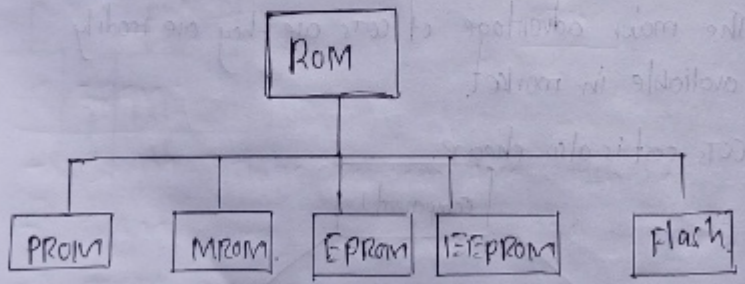
-> Low cost
-> less time required to develop
-> Testing effort is reduced

* Memory

-> Memory is an important part of embedded system.
-> memory used in embedded system is either ROM (or) RAM,
(i) ROM (ii) RAM,

(i) ROM:-

-> ROM is also known as program storage memory.
-> Types of ROM.



PROM:

-> programmable Read-only memory
-> It is empty when manfactured
-> permanantly programmed by user.
-> Data stored in PROM can't be erased by user.
-> PROM is a digital type memory.
-> PROM is also called as one-Time programmable (OTP).
-> It is low cost.

MROM:

-> MROM - Masked Read-only memory.
-> MROM uses hard-wired technology for data storing
-> MROM is also one-time programmable memory (OTP).
-> MROM is low cost.
-> Data stored in MROM can't be erased by user.
-> MROM size is small compared to other memories

EPROM:

-> EPROM - Erasable programmable Read-only memory
-> EPROM - cannot programmed permanantly.
-> Data stored in EPROM can be erased by user.
-> EPROM can be Reprogrammed.

EEPROM

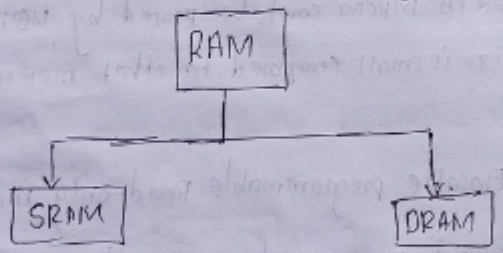
-> EEPROM - Electrically Erasable programmable Read-only memory.
-> EEPROM - can't programmed permanantly
-> Data stored in EEPROM can be ereased electrically.
-> In EEPROM we can delete up to 10,000 times.

Flash memory:

-> Flash is the latest ROM technology.
-> Flash is most popular technology used in today's Embedded systems.
-> Flash is variation of EEPROM.
-> Flash memory stores information in an array.
-> It can be reprogrammed.

(iii) RAM:

-> RAM - Random Access memory
-> RAM is a ~~volatile~~ volatile memory.



i SRAM:

-> SRAM - static Random Access memory
-> SRAM stores the data in the form of voltage.
-> SRAM is made up of flip-flops
-> SRAM is fast memory.
-> SRAM is expensive.
-> SRAM is low density device.

ii DRAM:

-> DRAM - Dynamic Random Access memory
-> DRAM stores the data in the form of charge.
-> DRAM is slow memory.
-> DRAM is cheaper
-> DRAM is high density device

SRAM	DRAM
-> SRAM - static Random Access memory	-> Dynamic Random Access memory
-> more expensive.	-> less expensive.
-> SRAM is faster.	-> DRAM is slower.
-> low capacity.	-> High capacity.
-> SRAM - low density device	-> DRAM - High density device

RAM	ROM
-> RAM - Random Access memory.	-> ROM - Read only memory.
-> Speed is high.	-> speed is low.
-> It is high cost. -> High capability -> It is volatile.	-> It is low cost. -> low capability. -> It is non-volatile.
-> It is temperary storage.	-> It is permanent storage.
-> Direct Access	-> sequential Access
-> large memory.	-> small memory.
-> stores in MBs	-> stores in GBs.
-> Types o SRAM o DRAM.	-> Types o PROM. o EEPROM o MROM o Flash memory. o EPROM.

* Memory According to type of Interface :

-> Memory Acc. to type of interface

- o. parallel interface.
- o Serial interface.

o. parallel Interface :

-> parallel data lines [D0-D7] of 8-bit processor/controller will be connected to the memory

-> The memory capacity expressed in kilobytes.

o. Serial Interface :

-> Serial interface is commonly used for data storage memory EEPROM.

-> The memory capacity expressed in ~~kilobytes~~ kilobits.

-> It is a 2wire interface

* Memory shadowing :

-> memory shadowing is a technique used to solve the execution speed problem.

-> shadow memory is used to store data (program) during its execution.

-> shadow memory consist of shadow bytes.

-> memory shadowing uses RAM to increase speed instead of ROM.

* Memory selection for Embedded systems :

-> The memory requirement for an embedded system in terms of RAM/ROM ~~is solely~~ dependent on the type of Embedded system.

-> lot of ~~the~~ factors considered when selecting the type and size of memory for embedded system.

-> selecting factors

- o speed
- o Data storage size and memory
- o cost.
- o power consumption
- o Bus width.

-> As a rule of thumb, identify system requirement based on type of processor used to design.

-> If Embedded systems design is based on RTOS, the RTOS requires certain amount of RAM for execution and ROM for storing RTOS image.

-> There are two parameters for representing memory:

- o size of memory chip
- o word size of memory.

-> Memory chips come in standard size like ~~512~~ 512 bytes, 1024 bytes, 2048 bytes, 4kb, 8kb, 16kb, 32kb, 64kb, 128kb, 256kb, 512kb.; 1megabyte etc.

-> coord sizes of memory are · 4, 8, 12, 16, 24, 32 etc.

-> Additional RAM/ROM memory required for running user Applications.

-> Flash memory is also used, it is of two types,

1. NAND flash 2. NOR flash.

- NAND flash is low cost, high density.
- NOR flash is high cost, low density.

* I/O - components;

◦ 7 - segment LED :-

- IN MPMC Notes.....

◦ Relay :-

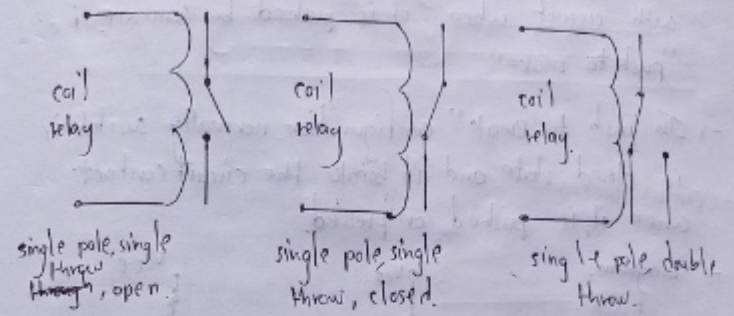
-> A Relay is defined as device that responds to a small current (or) voltage change by activating a switches (or) other devices

-> Relay is a I/O device (or) electro mechanical device.

-> Relay works on electromagnetic principle.

-> when voltage is applied to relay coil, current flows through coil, which generates magnetic field.

-> Diagramatic representation



-> A transistor can be used as relay driver.

◦ Piezo Buzzer ;

-> Def; It is a piezo electric device for generating audio indications in embedded applications.

-> A piezo electric buzzer contains piezo electric diaphgram which produces audio sound when voltage is applied

-> It is of two types

- self-driving
- External driving.

-> self-driving generate sound at predefined tone

-> External driving generates different sounds (or) tones

◦ push Button switch ;

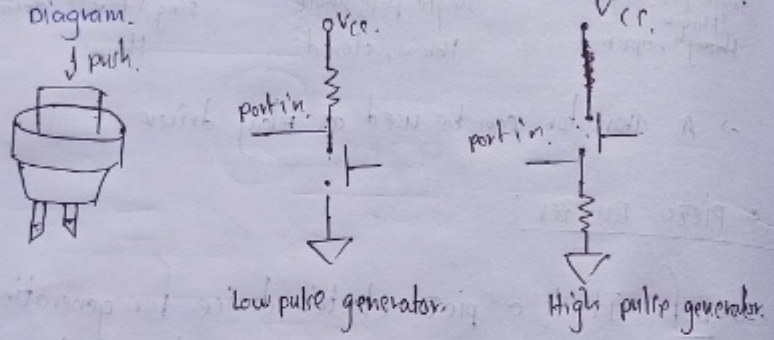
-> push button switch is a input device.

-> push button switch is used for two configuration - "push to make" and "push to Break" etc.

-> The switch is normally in open state, make it's contact with circuit when it is pushed by someone. i.e "push to make".

-> In "push to Break" configuration, normally switch is in closed state and it breaks the circuit contact when it is pushed or pressed.

-> Diagram.



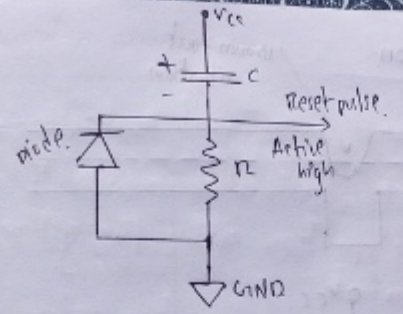
* subsystems :-

o Reset circuit : Reset circuit consist of a resistor, capacitor used to generate an reset pulse signal is called reset circuit.

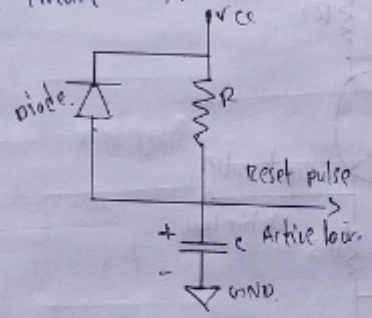
-> The reset signal is either Active high or/ Active low.

-> circuit diagram

-> Reset circuit - Active high.



-> Reset circuit - Active low.



* Brownout protection circuit :-

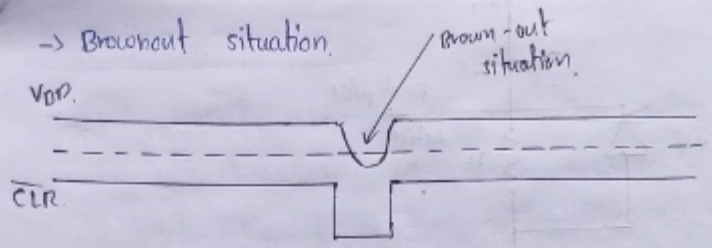
-> Def: Brownout protection circuit, which detects when supply voltage goes below and puts device into a recet state, is called Brownout protection circuit.

-> It is also called as Brown-out Reset.

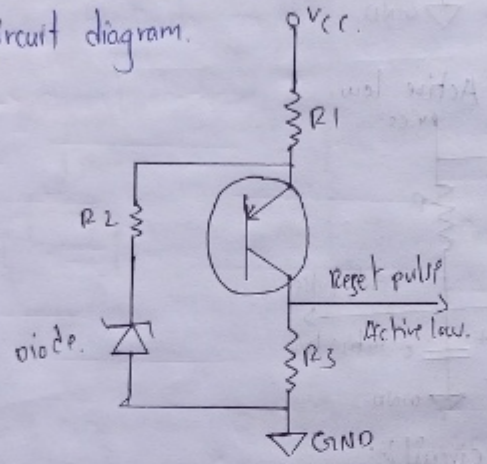
-> A Brown-out protection circuit holds the processor/ controller in reset state.

-> If operating voltage ~~fallow~~ falls below threshold voltage, this circuit rises voltage above the threshold voltage.

-> Brownout situation.



-> circuit diagram.



-> Brown-out protection circuit consist of zener diode transistor, three resistor, Reset pulse, Active low etc.

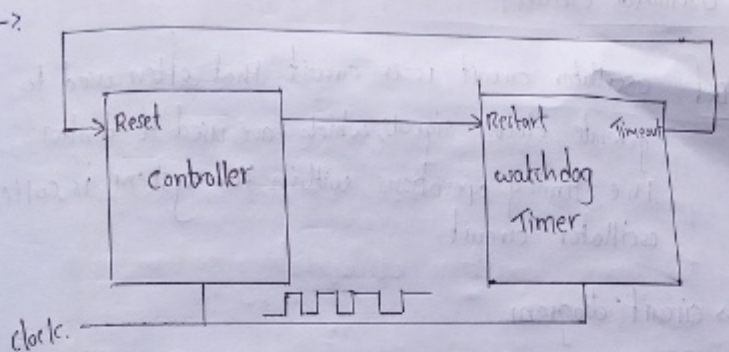
-> circuit shows Brown-out protection circuit with Active low output.

-> Zener diode and Transistor 'Q' plays as heart for this circuit etc.

*. Watch Dog Timer :-

Def: watchdog Timer is a timer ~~that monitors~~ with a special capability of rebooting an mcu, SoC, SiP when it times out is called watchdog timer.

->



-> circuit consist of controller, watchdog timer, Timeout, Reset, clock pulse.

-> watchdog Timer ~~is~~ initialized with Timeout value.

-> key reasons for watchdog timer.

- Automatic Recovery
- Real-time monitering
- compliance

-> Types of watch dog Timers

1. software watchdog Timer.
2. Hardware " "

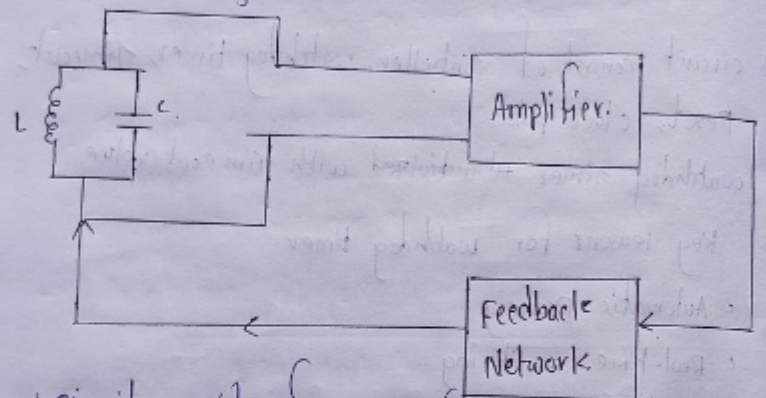
-> watchdog Timer can be Restart and Reset.

-> watch dog Timer moniter the CPU.

* oscillator circuit:

Def: oscillator circuit is a circuit, that often used to generate clock signals, which are used to control the timing operations within the system is called oscillator circuit.

-> circuit diagram.



-> circuit consist of Amplifier, Feedback Network, inductor, capacitor.

-> Clock signal is generated by using different oscillator

-> speed and power consumption depends upon clock frequency.

* Real Time Clock:

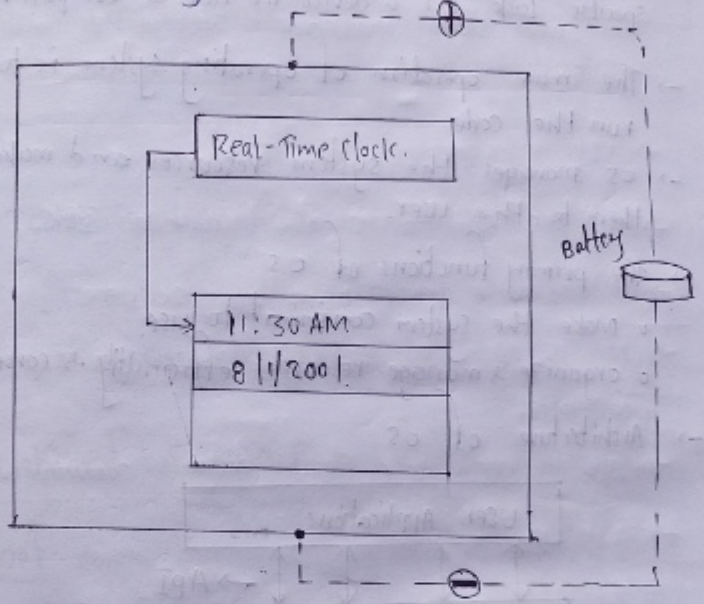
-> Def: A Real Time clock (RTC) is used in embedded systems to keep track time and date even, when the device is not connected to the network. or power off is called RTC.

-> RTC is powered by its own lithium battery.

-> RTC keeps running all time.

-> RTC is an IC.

-> Block diagram or circuit of RTC.



-> Block diagram consist of Battery, Real-time clock.

-> There is interconnection b/w RTC and digital display.

Unit-3

Communication interface :-

Def: Communication interface enables embedded systems to establish communication with each other, within large system.

-> Communication interface is of two types namely
- onboard communication interface.
- External board communication interface.

1. onboard communication interface (Device / Board level)

Def communication channel which interconnects the various components within embedded system is referred as onboard communication interface.

-> It is also called as device (on Board level communication).

-> It is classified into following types
1. I2C (Inter integrated circuit) Bus.
2. SPI (serial peripheral interface) Bus.
3. parallel interface.

1. I2C (Inter Integrated circuit) Bus:-

Def: I2C (Inter integrated circuit) Bus is used to interconnect peripheral devices within small-scale embedded systems is called I2C.

-> I2C is pronounce as "I-square-C".

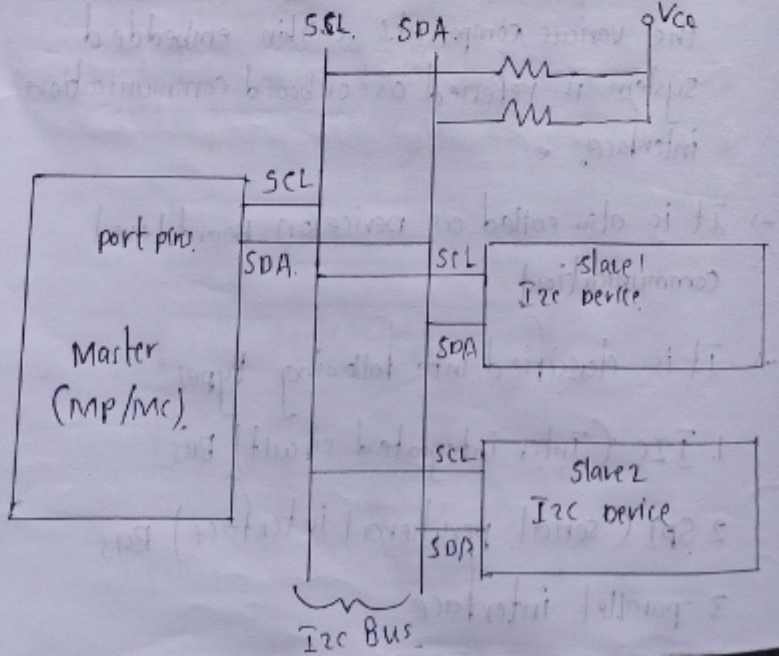
-> It is sometimes also known as IIC.

-> It has been exist more than 20 years.

-> It is equal to SPI but some operation is different.

-> It is used for short distance communication.

-> I2C Bus Interfacing



-> I2C Bus interfacing consist of SCL - serial clock, SDA - serial data.

-> SCL and SDA are port pins.

-> SCL serial clock is responsible for generating clock pulses and SDA is responsible for serial data transmission.

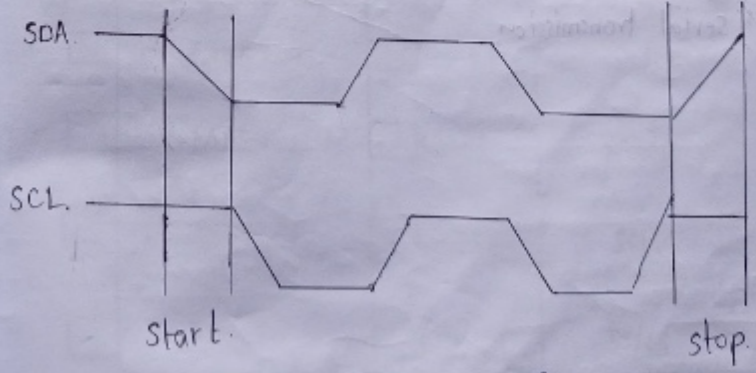
-> I2C is a shared bus, devices connected to I2C bus can act as either master or slave.

-> The master device is responsible for controlling the communication.

-> The slave device is wait for commands from master.

-> Master and slave devices act as either transmitter or receiver.

-> start and stop conditions.



-> Start condition - SDA change from high to low while keeping the SCL high.

-> Stop condition: SDA change from low to high while keeping the SCL high.

-> Read/write bit:

-> A high read/write bit indicates master sending data to the slave.

-> A low read/write bit indicates master receiving data from the slave.

-> Ack:

-> If the data frame is received successfully then Ack bit is sent to the ~~sender~~ sender by the receiver.

-> Features

-> Half-duplex

-> Synchronous communication.

-> Serial transmission.

9. SPI (serial peripheral interface)

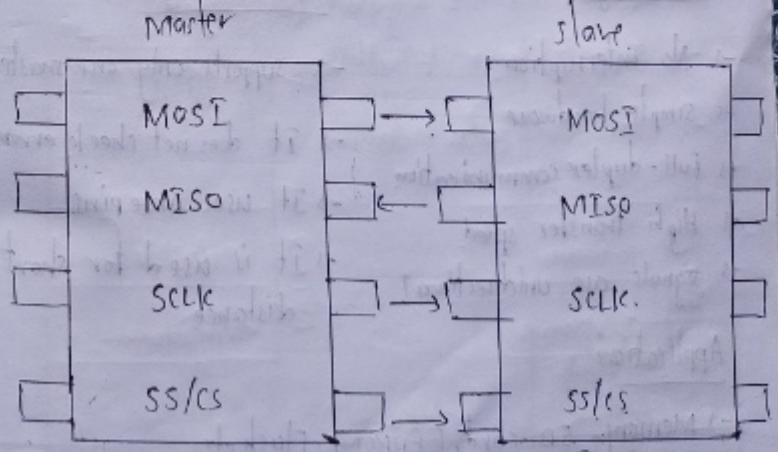
-> SPI (serial peripheral interface) is an interface commonly used in embedded systems for short distance communication between a microcontroller and one (or more) ICs is called SPI.

-> SPI is used in embedded systems for short distance wired communication b/w ICs.

-> It is a full-duplex communication, means data transmitted from both directions.

-> The advantage of SPI is to transfer data without interruption.

-> SPI interface



-> Block diagram shows SPI interface.

-> The SPI interface uses four wires for the communication.

They are:

1. MOSI : stands for master output slave input, used to send data from master to slave.

2. MISO : stands for Master Input slave output used to send data from slave to master.

3. SCLK : It is used to clock signal. (serial clock)

4. SS/CS : stands for slave select / chip select, the master send data to slave.

Advantages

-> No Interruption.
-> simple hardware
-> Full-duplex communication
-> High transfer speed
-> signals are unidirectional

Disadvantages

-> supports only one master
-> It does not check errors.
-> It uses more pins.
-> It is used for short distance

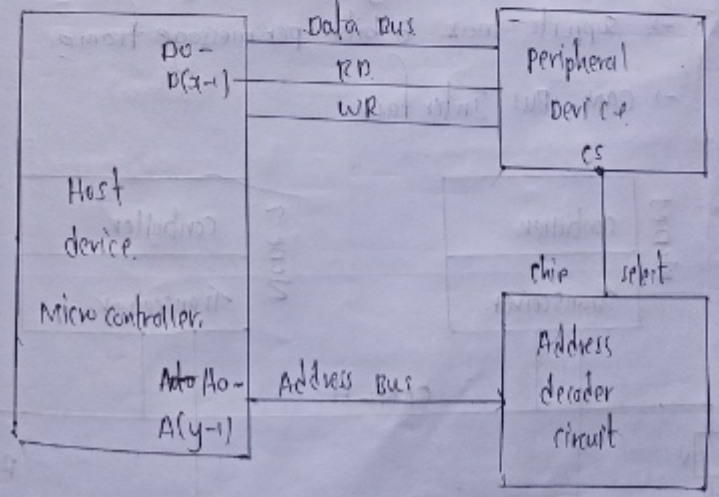
Applications

-> Memory: SD card, EEPROM, Flash etc.
-> Sensors; Temperature, pressure

3. Parallel Interface :

-> In data transmission, parallel communication is a method of conveying multiple binary digits simultaneously.
-> It contrasts with communication
-> The communication channel is the number of electrical conductors used at the physical layer.
-> parallel communication implies more than one such conductor.
-> 8-bit parallel channel will convey eight bits simultaneously.
-> parallel communication is always widely used in ICs.

-> parallel interface



* CAN : (Control Area Network)

-> Def: The CAN is designed to interconnect & communicate microcontroller and other devices without any host computer is called CAN.

-> CAN - Control Area Network.

-> CAN is widely used in automotive & Industrial automation applications.

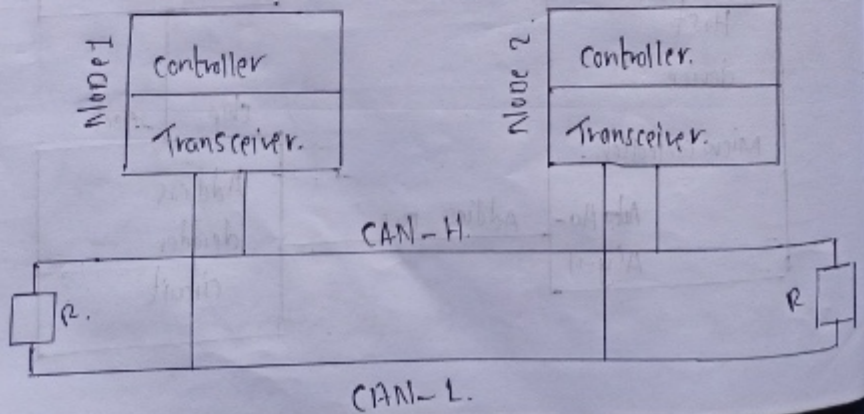
-> It is a serial half duplex and asynchronous type of communication.

-> CAN is a two wired communication namely CAN-H and CAN-L.

-> operates at 1 megabit per second.

-> supports max 8 bits per message frame.

-> CAN Bus Interface.



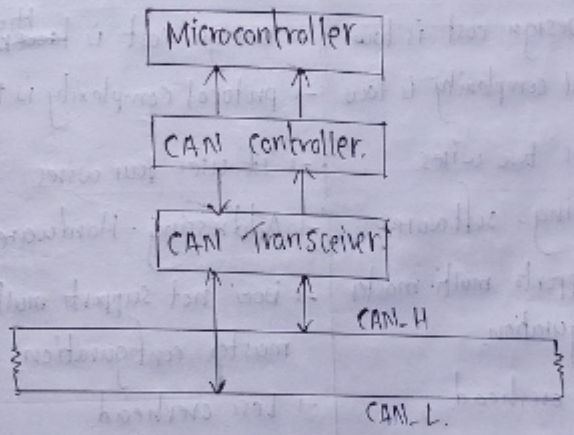
-> Block diagram shows CAN Bus interface.

-> Each Node consist of controller and Transceiver.

-> Two wires namely CAN-H & CAN-L are Two Buses.

-> Terminating resistance is placed at both the ends.

-> There is no interruption of messages.



-> Why CAN?

- High Reliability
- Low cost
- Low weight
- Scalability.

-> CAN Layered Architecture.

physical layer.
↓
Data link layer.
↓
Network layer.
↓
Transport layer.
↓
Application layer.

-> Applications

- Automotive
- Industrial automation
- Robotics.
- Avionics

I2C	SPI
-> I2C stands for Inter Integrated circuit.	-> SPI stands for serial peripheral Interface.
-> It is half-duplex communication.	-> It is full duplex communication.
-> Data transfer speed is slow.	-> Data transfer speed is Fast.
-> Design cost is low.	-> Design cost is ~~lower~~ High.
-> protocal complexity is low.	-> protocal complexity is lower
-> It uses two wires.	-> It uses four wires.
-> Addressing - software.	-> Addressing - Hardware.
-> It supports multi-master configuration.	-> Does not supports multi-master configuration.
-> More overhead.	-> Less overhead.
-> It is multi-master protocal	-> It is single master protocol.
-> more power required.	-> less power required
-> It is slower	-> It is faster.

* External communication Interface

def: The External communication interface is responsible for data transfer blw Embedded systems and other devices (or) modules is called ECI

-> It is classified into two types, they are:
1. wired communication Interface
2. wireless " "

1. RS232C and RS485

-> RS-232 (Recommended standard number)

-> RS-232 is a full duplex, wired, asynchronous serial communication interface.

-> RS-232 is an extension of UART.

-> RS-232 uses a EIA standard, is represented voltage with blw +3 and +25 and +3 and -25.

-> In EIA logic '0' is known as 'space' and logic 1 is known as "mark"

-> RS-232 Transmit and Receive signals.

-> RS-232 Supports two types of connectors namely
1. DB-25 : 25
2. DB-9 : 9.

-> DB-25:25

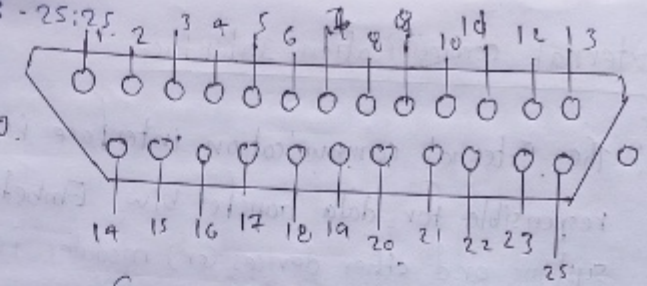


fig: DB:25:25 connector.

-> DB-9 connector.

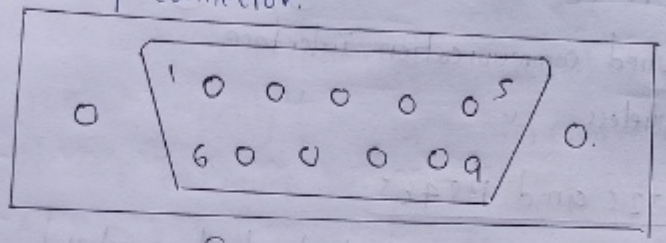


fig: DB-9M connector

-> It is a point-to-point communication.
-> The Devices are called
DTE - Data terminal Equipment
DCE - Data communication Equipment.

-> RS 485 is enhanced version of RS-422.

* USB

-> USB stands for universal serial Bus.
-> USB allows connection of peripheral devices.
-> peripheral devices are keyboards, scanners, printers, audio devices etc.

-> low speed : 10-100kb/s
-> High speed : 400mb/s
-> Full speed : 500kb/s - 10mb/s
-> USB transfer data in packet format.
-> It follows a star topology.
-> USB supports upto 5 meters
-> USB uses two types of connectors
- Type 'A' connector for upstream
- Type 'B' connector for downstream.

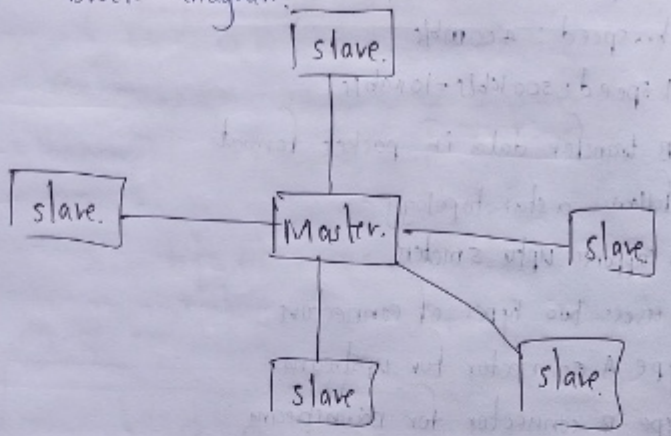
-> Types of data transfer.
- control
- Isochronous
- Bulk
- interrupt etc.

* Bluetooth

Def: Bluetooth is a wireless technology used for short distances for exchanging data over radio waves.

-> Bluetooth is a wireless technology.
-> Bluetooth uses radio technology called frequency.
-> Bluetooth transmit and receive data in the form of packets
-> It uses for short distances only.
-> It is low cost.
-> It requires low power.

-> Block diagram



-> Max No. of Master Nodes - 1.
-> Max no. of slave nodes - 7.
-> Max no. of Nodes in a pironet - 8.

Advantages

-> low cost.
-> Easy to use.
-> It can penetrate through walls.

Disadvantages

-> less secure.
-> can be hacked.
-> slow data transfer.

Applications

-> wireless headsets.
-> digital camera.
-> Bluetooth speakers.

* Zigbee :

Def; Zigbee is a wireless personal Area network.

-> zigbee is a low cost and low powered network.
-> zigbee supports different network configurations. they are
  - master to master.
  - master to slave.
-> It is used for home networking.
-> It is created by Zigbee alliance.
-> Types of Zigbee devices.
  - Zigbee coordinator device: This device is used for connecting the devices.
  - Zigbee Router: This is used for passing the data blw devices.
  - Zigbee end device: This used to control.

-> characteristics :

-> low cost.
-> low power consumption.
-> low, data rate.
-> used for short range.

-> It is robust and flexible.
-> It's basic architecture is piconet.

Advantages
-> low cost.
-> Reliable.
-> Easy to implement.
-> use in smartphone

Disadvantages
-> limited range.
-> No security
-> limited data rate.

Applications
-> Home Networking.
-> commertial
-> light control system.

* GPRS :

-> GPRS - general packet radio service.
-> GPRS is a third step towards internet access.
-> GPRS is also known as GSM-IP.
-> GPRS is a mobile communication, runs on 2G and 3G.
-> GPRS is a high speed data transfer.

-> Working

-> GPRS is a packet switching technology.
-> GPRS supports to cellphones, laptops, handheld devices etc.
-> GPRS takes less time to connect.
-> GPRS have flexible channel allocation.
-> Traffic rules are suitable for GPRS.
-> GPRS have always online feature.
-> High data rate.
-> Easy billing.
-> consultant IP services.
-> GPRS is a wireless communication.
-> cost is effitient.
-> mobility.
-> Localization.

Advantages
-> Easy billing
-> Increased speed.
-> Always on connectivity.

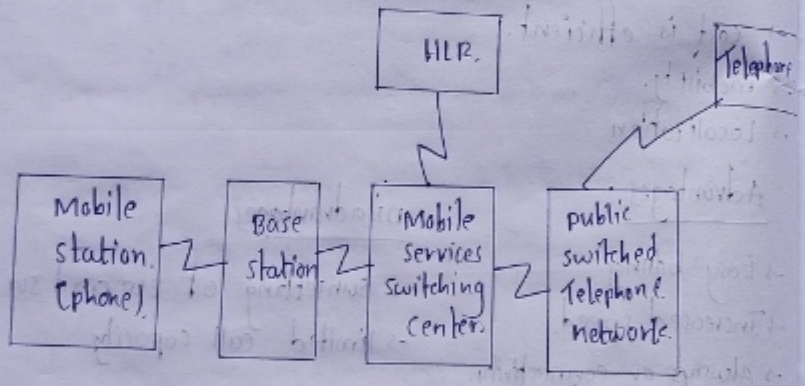
Disadvantages
-> sunsetting of 2G and 3G.
-> limited call capacity

* GSM

-> GSM - Global system for mobile communication.
-> GSM is used to set up communication b/w an embedded system an GSM network is called GSM.
-> GSM operates at different frequencies.
-> GSM was developed by European Telecommunications.
-> operates b/w 900MHZ - 1800MHZ.

-> GSM used all over the world.
-> GSM is a wireless communication.
-> GSM having 4 different types of cells:
  - Macro
  - micro
  - pico
  - umbrella.

-> Architecture.



-> Block diagram shows architecture of GSM.

-> GSM features.
-> clear voice clarity
-> low cost
-> low power devices
-> support multiple devices.

-> GSM supports three services
  a. Data services
  b. Tele services
  c. Supplementary services

Advantages

-> security
-> wide range features

Disadvantages

-> Limited capacity
-> slow data transfer.

Unit- 4

* Embedded Firmware:

Embedded firmware is the flash memory chip that stores specialized software running in a chip in a embedded system is called Embedded Firmware.

-> It is an un-avoidable part of embedded system

-> The Embedded firmware can be developed in various methods like:-

- write a program in high level Language. Embedded c/c++ using IDE.
- write a program with in Assembly language.

* Embedded Firmware Design approaches:

-> The Embedded Firmware is the master brain of Embedded system.

-> The Embedded Firmware provides intelligence to an embedded system.

-> It is a one time process.

-> The Embedded firmware is stored in permanent memory (Rom)

-> There exist two basic design approaches for embedded firmware. They are:

1. Super loop based approach.

2. operating system based Approach.

① Super-loop based Approach:

-> The super-loop based Approach is adopted for applications that are not time critical.

-> It is similar to conventional procedural programming where code is executed task by task.

-> The tasks are executed in a loop.

-> A super loop implementation will look like:

1. configure the common parameter and perform initialization for various hardware components, memory, registers etc.
2. start first task & execute it
3. Execute the second task.
4. Execute the third task.
5. :
6. :
7. : Execute the last task.
8. Jump back to first task and follow the same flow.

-> The C-program code for super loop

```
Void main ( )
{
Configurations ( );
Initializations ( );
while (1)
{
Task 1 ( );
Task 2 ( );
:
:
Task n ( );
}
}
```

-> Here we used while (1) loop.

-> This approach is also referred as "super loop based Approach"

② operating system based Approach:

-> The operating system based approach contains operating systems, which can be either General purpose operating systems (GPOS) or Real Time operating systems (RTOS).

-> operating systems (OS):

- General purpose operating systems (GPOS).
- Real Time " " (RTOS).

-> The general purpose OS based design is very similar to conventional pc based application.

-> microsoft® window XP Embedded is an example of GPOS.

-> PDA, Gaming stations, Tablet PC's etc are examples of GPOS.

-> 'windows CE', windows mobile, QNX, Embedded Linux, Symbian etc are examples of RTOSs.
-> mobile phones, PDAs, are examples of RTOSs.

* Embedded Firmware development languages:

-> Embedded Firmware development languages are of two types they are:
1. Assembly Language.
2. High Level Language.

1. Assembly Language:

-> 'Assembly Language' is the human readable "machine Language"
-> "Machine Language" is a processor understandable Language.
-> Machine language is binary data, it consists of 1s and 0s.
-> Assembly and machine languages are processor dependent.
-> Assembly languige program written for one processor, not work with others.
-> The general format of an assembly language instruction is opcode. followed by operands.
-> opcode tells processor to what to do. and operands provide data required. to perform specific action by opcode.
-> The operand may be. single operand, dual operand or more.
-> The 8051 assembly Instruction.

MOV A, #30 (move decimal 30 to the 8051 register)

e. Here. Move A is opcode. and 30 is the operand.

-> same Instruction in machine language.

01110100 00011110.

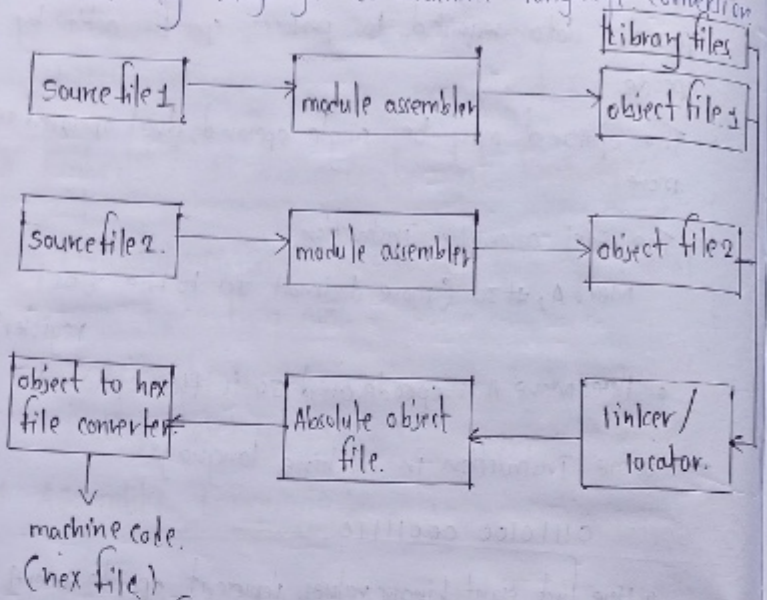
o Here first eight binary values represent opcode mov A, next 8 binary values represent operand 30.

-> Each line of Assembly language program split into four fields.
o Lable o. opcode o operand o comments

Assembly language.- source file to hex file.

-> The Assembly language program written in Assembly code is saved as .asm file or) a .src file.
-> similar to 'c' and other high. level language.
-> This approach is known as 'modular programming'
-> In 'modular programming' the entire code. is divided into submodules.

-> Assembly language to machine language conversion



Advantages of assembly language

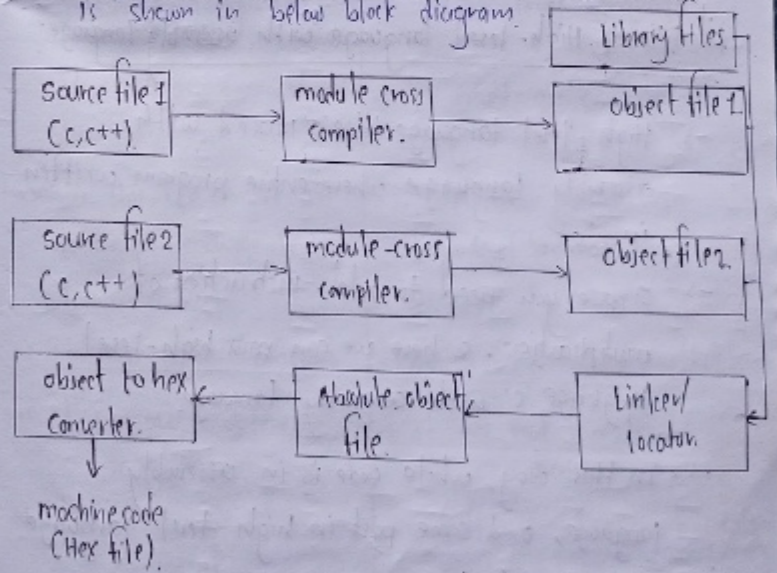
-> High performance.
-> Code reverse Engineering.
-> Efficient code and data memory usage.

Disadvantages

-> High development time.
-> Non portable.

2. High level language :

-> The most commonly used high level language in Embedded firmware is 'C'.
-> The various steps involved in high level language is same that of assembly language.
-> The Embedded firmware is written in high-level languages like C, C++.
-> The various steps involved in the converting high-level language to corresponding machine language is shown in below block diagram.



-> The software program called linker/locator.
-> The software utility called 'object to hex file converter.

→ Mixing of high-level language with Assembly language in three different ways.

i) Mixing assembly language with high level language like 'c'.

-> Assembly language mixed with 'c', where entire program written in 'c'.
-> mixing cand assembly is little complicated.
-> In this way whole code is in high level language, and some part in assembly language.

ii) Mixing High-level language with assembly language:

-> High-level language like 'c' mixed with assembly language, where entire program written in 'c'.
-> Suppose you need to add instruction of multiplication, so here we can mix high-level language 'c' with assembly language.
-> In this way, whole code is in assembly language, and some part in high-level language.

iii) Inline Assembly:

-> Inline assembly is another technique for inserting the target processor at any location, code written in high-level language.
-> Inline assembly avoids the delay.
-> In Inline assembly special keywords are used to start and end.

Eg: #pragma asm
MOV A, #13H
#pragma endasm

* Difference b/w Assembly and High-level language

Assembly Language	High-level language
-> It needs an assembler for conversion	-> It needs an compiler for conversion
-> In this, we convert an Assembly language to machine language	-> In this, we convert an high-level language to assembly language to machine language
-> It is machine dependent	-> It is machine Independent
-> It supports low-level operation	-> It does not low-level operation
-> Easy to access hardware component	-> Difficult to access hardware component
-> More compactness	-> No compactness
-> Requires less memory	-> Requires more memory
-> Low-level language	-> High-level language

Assembly language	machine language	High-level language
-> It is low level language	-> It is low-level language	-> It is high level language
-> It is understand by humans	-> It is understand by compuler	-> It is used in programs
-> Require less memory	-> Require less memory	-> Require more memory
-> Require less time for execution	-> Require less time for execution	-> Require more time for execution
-> program written in mnemonic form	-> program written in binary form	-> program written in English statements
-> It is not user friendly	-> It is less user friendly	-> It is more user friendly

- **Microcontroller**: A microcontroller in a embedded systems are small ICs that controls a single process is called microcontroller.
  - -> Microcontroller ~~is a~~ includes processor, memory, I/p - o/p peripherals on single chip.
  - -> Microcontroller are low cost and readily available in market.

Difference b/w microprocessor & Microcontroller.

Microprocessor.	Microcontroller
-> Microprocessor are used for big Applications.	-> Microcontrollers are used to execute a single task within an application.
-> Requires higher power consumption.	-> Requires low power consumption.
-> Requires more instructions	-> Requires less instructions.
-> circuit design is complex.	-> circuit design is simple.
-> It is high cost.	-> It is low cost.
-> It is high speed.	-> speed depends on architecture.
-> It have less registers.	-> It have more registers.
-> Memory & I/p - o/p components are external to it.	-> Memory & I/p - o/p components are internal to it.
-> RAM/ROM are absent.	-> RAM/ROM are present.
-> center of computer system.	-> center of Embedded systems.

- **Digital signal processors**:-
  - -> DSP - digital signal processors
  - -> DSPs are powerful special purpose microprocessor.
  - -> DSPs are available in 8/16/32 bits.
  - -> DSPs are designed for today's embedded audio, video, and communication applications.
  - -> DSPs are 2-3 times faster than general purpose microprocessors.
  - -> speed of DSPs is more than general purpose microprocessors.
  - -> DSPs can perform speed operations like addition, substraction, multiplication, division etc.
  - -> ~~DSPs can perform~~ DSPs have following key units.
    - o program memory.
    - o Data memory
    - o I/o unit
    - o computational engine.

  Ex': Audio - video, telecommunication, multimedia etc.

Harvard Architecture.	von-Neumann architecture.
-> It is modern computer architecture.	-> It is Ancient computer Architecture.
-> seperate buses are used for transfering data & instruction.	-> common bus for transfering data & instruction.
-> single clock cycle is required to execute instruction.	-> Two-clock cycles are required to execute instruction.
-> It is ~~cheaper~~ costly.	-> It is cheaper.
-> It is complex to use.	-> It is not complex to use.
-> pipeline strategies are possible.	-> pipeline strategies are not possible.
-> speed is high.	-> speed is low.

RISC	CISC.
-> RISC- Reduced Instruction set computer.	-> CISC- complex Instruction set computer.
-> It uses more registers.	-> It uses less registers.
-> RAM usage is more.	-> RAM usage is less.
-> code size is more.	-> code size is less.
-> It is more expensive.	-> It is less expensive.
-> It consumes low power.	-> It consumes high power.
-> It has less addressing modes.	-> It has more addressing modes.
-> Emphasis on software.	-> Emphasis on hardware.
-> External memory doesn't required.	-> External memory is required.
-> It has fixed instruction format.	-> It has variable instruction format.

* Application specific Integrated circuits:

-> ASIC - Application specific integrated circuits.
-> ASIC is specially built for specific purpose.
-> compared with other device, ASIC has improved speed and reliability.
-> ASIC consumes less power.
-> ASIC can be found in most electronic devices.
-> The three different categories of ASICs,
  ◦ Full-custom ◦ semi-custom ◦ programmable

◦ full custom: It is a design methodology useful for ICs.
-> In full-custom circuit layout resistors, transistors, capacitors, analog circuits are placed.
-> It have good performance, high flexibility.

◦ semi-custom: semi-custom is the alternative to full-custom.
-> It is divided into two types
  (i) Gate array based.
  (ii) standard cell based.

◦ programmable: These are not custom based.
-> It is divided into two types
  (i) FPGAs
  (ii) PLDs.

* PLD's

-> PLD - stands for programmable logic devices

Def: programmable logic device is an electronic component, used to build reconfigurable digital circuits. is called PLD.

-> PLDs are based on re-writable memory and reprogrammed to change design.

-> programmable logic devices - FPGA and CPLDs

-> Feild programmable Gate arrays and complex programmable logic devices are two major types of PLDs.

(i) FPGA: Feild programmable gate array.

-> It is an integrated circuit and thousand times faster than processors.

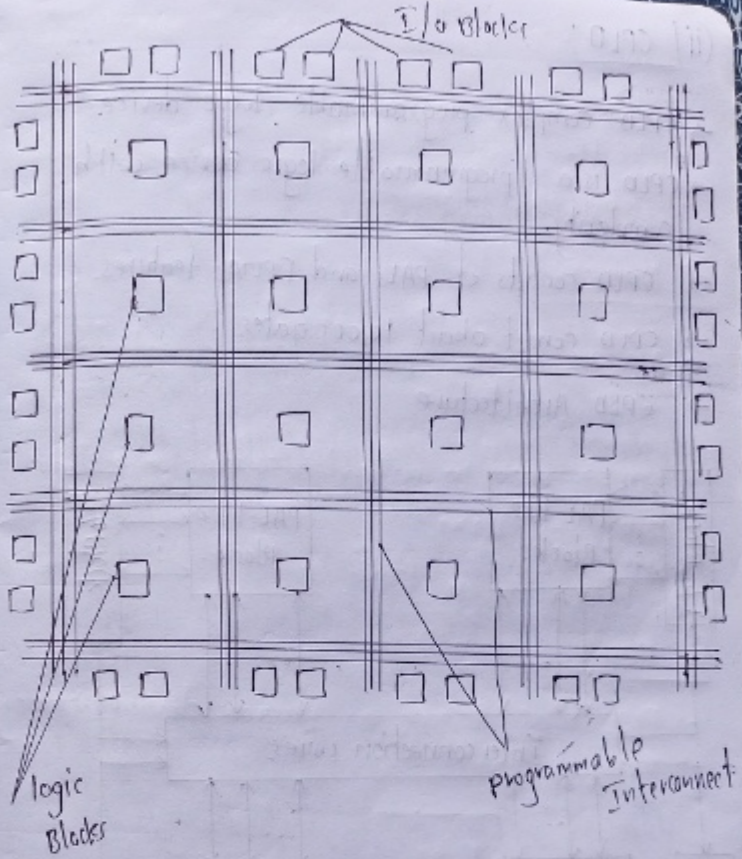
-> Its IC consist of CLB's, memory, and other elements.

-> FPGA offer most features

-> It have highest performance.

-> Density of system gates is from 1k to 500k.

-> FPGA architecture.



-> Architecture of FPGA consist of I/o Blocks, programmable interconnect, logic blocks etc.

-> FPGAs used in wide range of Applications

- instrumentation
- telecommunications
- DSPs. etc

(ii) CPLD :-

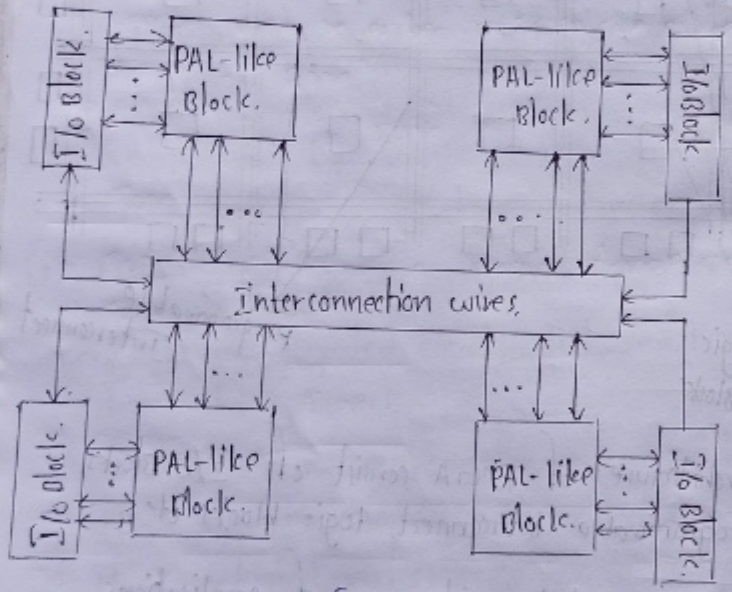
-> CPLD - Complex programmable logic device.

-> CPLD is a programmable logic device with complexity.

-> CPLD consists of PALs and FPGAs features.

-> CPLD consist about 10,000 gates.

-> CPLD Architecture.



-> CPLD IC consist of I/O Block, PAL-like block, Interconnection wires etc.

-> Interconnection wires plays vital role in CPLDs.

Advantages of PLDs

- Faster
- Low power consumption
- Low cost.
- High reliability.

* COTs

-> COTs - Commercial - off - the shelf

-> COTs remains "as-is".

-> COTs products are easy to design, integration and interoperability.

-> Typical examples of for the COTs products are Remote controlled Toys.

-> COTs includes High performance, high freq, high bandwidth, UV/IR detectors etc.

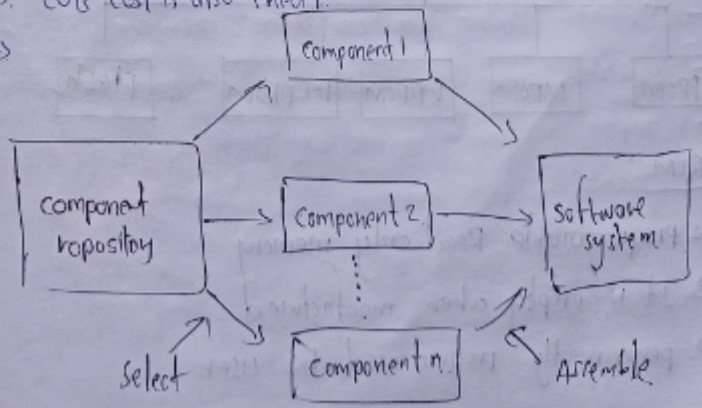
-> A COTs components contains

- GPP
- ASIP.
- ASIC
- ASSP
- PLD.

-> The major advantage of COTs are they are readily available in market.

-> COTs cost is also cheap.

->



Advantages

-> Low cost
-> less time required to develop
-> Testing effort is reduced

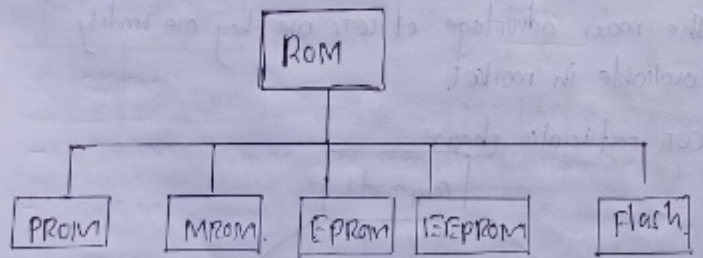
* Memory

-> Memory is an important part of embedded system.
-> memory used in embedded systems is either ROM (or) RAM.

(i) ROM (ii) RAM.

(i) ROM:-

-> ROM is also known as program storage memory.
-> Types of ROM.



PROM:

-> programmable Read-only memory
-> It is empty when manfactured
-> permanantly programmed by user.
-> Data stored in PROM can't be erased by user.
-> PROM is a digital type memory.
-> PROM is also called as one-Time programmable (OTP).
-> It is low cost.

MROM:

-> MROM - Masked Read-only memory.
-> MROM uses hard-wired technology for data storing.
-> MROM is also one-time programmable memory (OTP).
-> MROM is low cost.
-> Data stored in MROM can't be erased by user.
-> MROM size is small compared to other memories

EPROM:

-> EPROM - Erasable programmable Read-only memory
-> EPROM - cannot programmed permanantly.
-> Data stored in EPROM can be erased by user.
-> EPROM can be Reprogrammed.

EEPROM

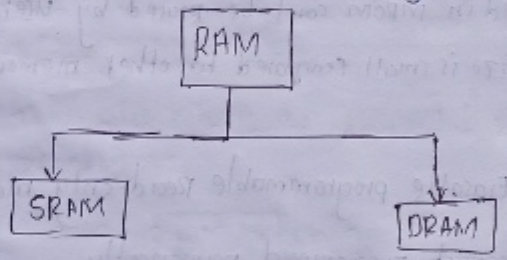
-> EEPROM - Electrically Erasable programmable Read-only memory.
-> EEPROM - can't programmed permanantly
-> Data stored in EEPROM can be ereased electrically
-> In EEPROM we can delete up to 10000 times.

Flash memory:

-> Flash is the latest Ram technology.
-> Flash is most popular technology used in today's Embedded systems.
-> Flash is variation of EEPROM.
-> Flash memory stores information in an array.
-> It can be reprogrammed.

(iii) RAM:

-> RAM- Random Access memory
-> RAM is a ~~vetfal~~ volatile memory.



i SRAM:-

-> SRAM - static Random Access memory
-> SRAM stores the data in the form of voltage.
-> SRAM is made up of flip-flops.
-> SRAM is fast memory.
-> SRAM is expensive.
-> SRAM is low density device.

ii DRAM.

-> DRAM - Dynamic Random Access memory
-> DRAM stores the data in the form of charge.
-> DRAM is slow memory.
-> DRAM is cheaper.
-> DRAM is high density device.

SRAM	DRAM
-> SRAM- static Random Access memory	-> Dynamic Random Access memory
-> more expensive.	-> less expensive
-> SRAM is faster.	-> DRAM is slower.
-> low capacity.	-> High capacity.
-> SRAM- low density device.	-> DRAM- High density device

RAM	ROM
-> RAM- Random Access memory.	-> ROM - Read only memory.
-> Speed is high.	-> speed is low.
-> It is high cost. -> High capability -> It is volatile.	-> It is low cost. -> low capability. -> It is non-volatile.
-> It is temperory storage.	-> It is permanant storage.
-> Direct Access	-> sequential Access
-> large memory	-> small memory.
-> stores in MBs	-> stores in GBs.
-> Types o SRAM o DRAM.	-> Types o PROM. o EEPROM o MROM o Flash memory. o EPROM.

* Memory According to type of Interface :

-> Memory Acc. to type of interface

- a. parallel interface.
- a serial interface.

a. parallel interface :-

-> parallel data lines (D0-D7) of 8-bit processor/controller will be connected to the memory

-> The memory capacity expressed in kilobytes.

a. Serial Interface :-

-> Serial interface is commonly used for data storage memory EEPROM.

-> The memory capacity expressed in ~~kilobytes~~ kilobits.

-> It is a ewire interface

* Memory shadowing :

-> memory shadowing is a technique used to solve the execution speed problem.

-> shadow memory is used to store data (program) during the execution.

-> shadow memory consist of shadow bytes.

-> memory shadowing uses RAM to increase speed instead of ROM.

* Memory selection for Embedded systems :

-> The memory requirement for an embedded system in terms of RAM/ROM ~~is solely~~ dependent on the type of Embedded system.

-> Lot of ~~two~~ factors considered when selecting the type and size of memory for embedded system.

-> selecting factors

- a speed
- a Data storage size and memory
- a cost.
- a power consumption
- a Bus width.

-> As a rule of thumb, identify system requirement based on type of processor used to design.

-> If Embedded systems design is based on RTOS, the RTOS requires certain amount of RAM for execution and ROM for storing RTOS image.

-> There are two parameters for representing memory:

- a. size of memory chip
- a word size of memory.

-> Memory chips come in standard size like ~~512~~ 512 bytes, 1024 bytes, 2048 bytes, 4Kb, 8Kb, 16Kb, 32Kb, 64Kb, 128Kb, 256Kb, 512Kb.; 1 megabyte etc.

-> word sizes of memory are: 4, 8, 12, 16, 24, 32 etc.

-> Additional RAM/ROM memory required for running user Applications.

-> Flash memory is also used, it is of two types,

1. NAND flash 2. NOR flash.

- NAND flash is low cost, high density.
- NOR flash is high cost, low density.

* I/O - components:

o 7 - segment LED:-

- IN MPMC Notes.....

o Relay :-

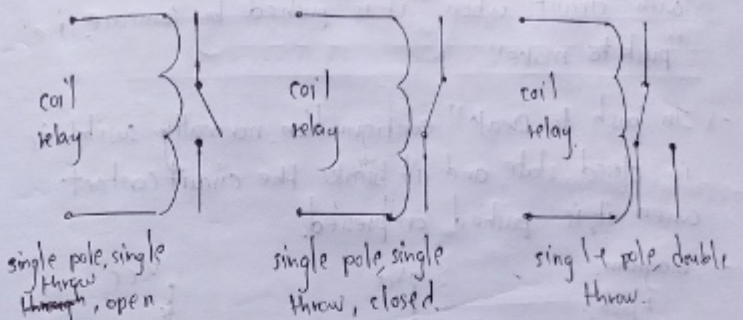
-> A Relay is defined as, device that responds to a small current (or) voltage change by activating a switches (or) other devices

-> Relay is a I/O device. (or) electro mechanical device.

-> Relay works on electromagnetic principle.

-> when voltage is applied to relay coil, current flows through coil, which generates magnetic feild.

-> Diagramatic representation.



-> A transistor can be used as relay driver.

o piezo Buzzer:

-> Def; It is a piezo electric device for generating audio indications in embedded applications.

-> A piezo electric buzzer contains piezo electric diaphgram which produces audio sound when voltage is applied

-> It is of two types.

- self-driving - External driving.

-> self-driving generate sound at predefined tone.

-> External driving generates different sounds (or) tones

o push Button switch:

-> push button switch is a input device.

-> push button switch is used for two configuration - "push to make" and "push to Break" etc.

Unit-5.

* Types of operating systems

-> operating systems are classified into two different types they are:-

1. General purpose operating system (GPPS)
2. Real-time operating system (RTOS).

1. General purpose operating system (GPOS):

Def: The operating systems which are deployed in general computing systems are called GPOS.

-> GPOS are non-deterministic in behaviour.
-> personal computers are best examples for GPOS.
-> Examples of GPOS XP/MS-DOS etc
-> GPOS are user-friendly to users.
-> GPOS contains multiple, buttons, icons, etc.

2. Real-time operating system:

Def: The operating systems which are deployed in embedded systems and demanding real time response is called RTOS.

-> RTOS are deterministic in behaviour.
-> RTOS consumes only known amount of time
-> RTOS works on real time only acc. to user commands.
-> Examples: QNX, Vxworks etc.

- Real-time kernal
- Hard real-time
- soft real-time

o Real-time kernal: The kernal of RTOS is refered as Real time kernal.

-> Basic functions of Real-time kernal
- Task Management
- Memory "
- Time "
- Error handing
- Interrupt handling
- Task scheduling
- Task synchronization

o Hard-real time: A hard-real time has absolute deadlines, if allotted time missed, a system failure will occur. is called hard-real time.

-> Example
  - Air traffic control
  - Medical system
-> It requires deterministic and predictable behaviour
-> It has better utility.

- Soft real time

-> A soft real real time, is, when a system continues to function even if it's unable to execute within an allotted time is called soft real time.
-> Examples
  - Computer games
-> Best examples of soft real time are
  - personal computer
  - Audio-video systems
  - DVD players

Hard real time	soft real time
-> In hard-real time the size of data file is small	-> In soft-real time, the size of data file is large
-> Response time is millisecond	-> Response time is high
-> In this safety is critical	-> In this safety is not critical
-> It is more restrictive	-> It is less restrictive
-> Guarantee response within deadline	-> Does not guarantee within deadline
-> It has more utility	-> It has less utility
-> Includes short database	-> Includes large database
-> It has short term data integrity	-> It has long term integrity
-> Highly predicable	-> less predictable
Ex: satellite launch, Railway signaling	Ex: DVD player, computer games

* Tasks

-> Def: The term "task" refers to something to be done.

-> Task is also known as 'job' in OS.

-> In operating system, a task refers as program in execution.

-> The term 'task' 'job' and 'process' refer to same entity in OS.

* process :

Def: A process is a program in execution is called process.

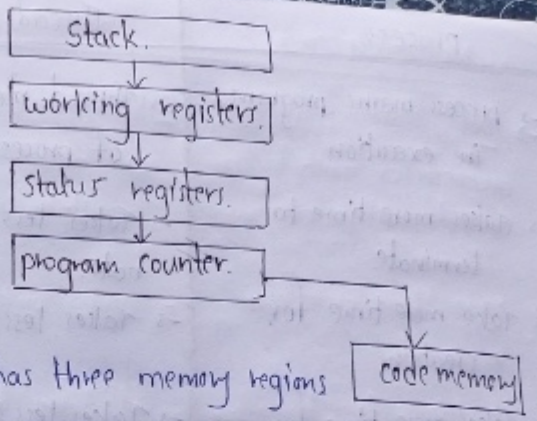
-> process is also know as program in execution.

-> multiple instances of same program can execute simultaneously.

-> process requires various system resources like

- CPU
- Memory
- I/O devices.

-> structure of process



-> process has three memory regions

- stack memory
- Data memory
- code memory.

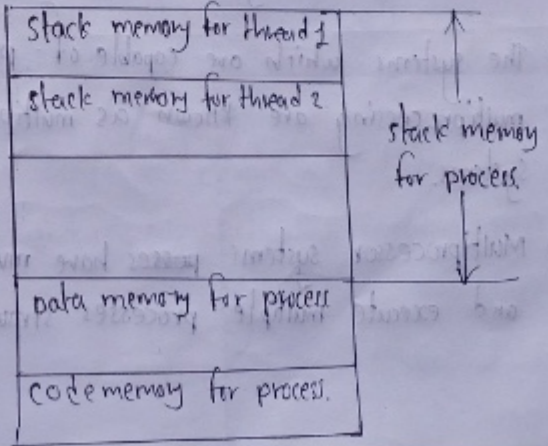
* Threads

-> A thread is a primitive that can execute code.

-> A thread is also known as light weight process.

-> A process can have many threads

-> A thread is a single sequential flow within a process.



Process	Threads
-> process means program in execution	-> Thread means segment of process
-> Takes more time to terminate.	-> Takes less time to terminate.
-> Take more time for creation.	-> Takes less time for creation
-> Takes more time for context switching	-> Takes less time for context switching
-> It is less efficient.	-> It is more efficient.
-> Heavy weight process.	-> less weight process.
-> system called is involved.	-> No system call is involved.

* Multiprocessing & Multitasking:

Multiprocessing: The ability to execute multiple processes simultaneously is referred as multiprocessing.

-> The systems which are capable of performing multiprocessing are known as multiprocessing systems.

-> Multiprocessor systems ~~posses~~ have multiple CPUs and execute multiple processes simultaneously.

-> The ability of operating system to have multiple programs in memory and ready for execution is called multiprogramming.

Multitasking when single resource is used to process multiple tasks then it is called multitasking.

(or)

multitasking is defined as execution of multiple tasks at a time is called multitasking.

Types of multitasking

1. preemptive multitasking:

-> It is a special task assigned to computer operating system.

-> The operating system controls entire process is referred as preemptive.

-> It is used in desk top computers.

2. Non-preemptive multitasking:

-> Non-premptive multitasking is also known as cooperative multitasking.

-> It is mainly used to run present task.

-> It uses windows, macos etc.

Multiprocessing	Multitasking.
-> Def: Running multiple processes on multiple cpus called multiprocessing.	Def: Running multiple tasks on single cpu is called multitasking
-> More than one cpu is used.	-> only one cpu is used.
-> It takes less time for processing.	-> It takes moderate amount of time.
-> It is less economical.	-> It is more economical.
-> The no. of users is one (or) more.	-> The no. of users is more than one.
-> Throughput is Max.	-> Throughput is moderate.
-> Efficiency is max.	-> Efficiency is moderate.
-> Types: o symmetric. o Asymmetric.	-> Types o single user. o multiple user.

* Task communication :

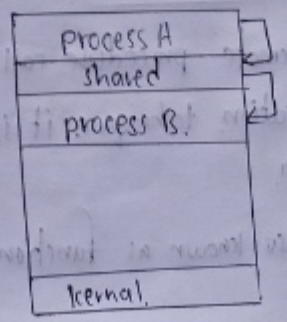
Task : Task is defined as program in execution.
-> Task is also called as 'job'.

Task communication : Task communication consist of all mechanisms serving to exchange information among tasks is called Task communication.

-> The process running on os classified as
- o co-operating process.
- o competing process.

Shared memory

-> shared memory is an operating feature that allows the database server threads and processes to share data by memory.
-> It is used to reduce memory usage.
-> It is faster inter process communication.
-> shared memory requires two or more processes.



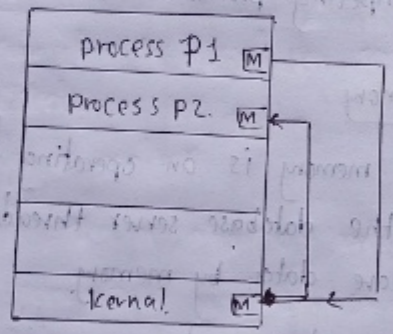
Message passing:

Message passing is a technique for sending messages to communicate what you want done is called message passing.

-> Message passing can be of
- send message
- Receive message

-> Messages can be either fixed (or) variable size.

-> Block diagram.



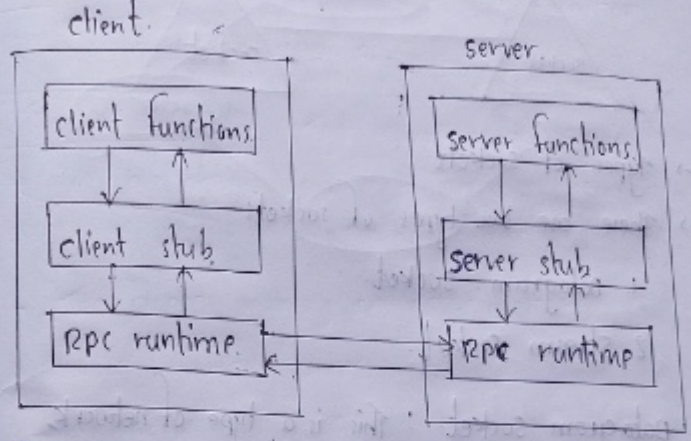
-> Message passing is mainly used for communication

* Remote procedure call (Rpc)

-> The Remote proredure call is an interprocess communication technique. it is used for server application.

-> It is also known as function call (or) subroutine call.

-> Block diagram.



Advantages

-> Good performance.

-> Network communication hidden from user.

Disadvantages

-> No flexibility

-> Not standard

Sockets

A socket is a end point of two way communication. btw two programs running on network.

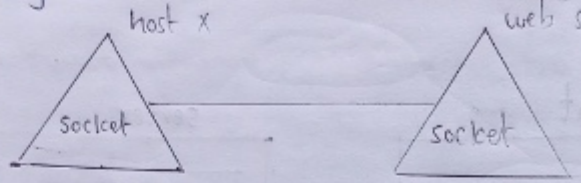
-> Socket provides inter process communication.

-> Socket provides bi-directional FIFO.

-> Each socket has specific address

-> Sockets are employed in client server applications.

-> Diagram.



-> Types of sockets

-> There are two types of sockets

1. Datagram socket.
2. Stream socket.

1. Datagram socket : This is a type of network which has connection-less point for sending and receiving packets.

-> It is similar to mailbox.

2. Stream socket : This is a type of network socket which has connection point for sending and receiving packets.

-> It is similar to phone.

* Task synchronization

Def: Task synchronization is defined as to synchronize data between source and a target is called Task synchronization.

-> Task synchronization Issues

-> Here we discuss Task synchronization issues.

1. Race condition (Racing)
2. Deadlock.

1. Racing :

-> Race condition (or) Racing is the situation in which multiple processes race each other to access and manipulate data concurrently is called Racing.

-> In Racing final value depends on shared data.

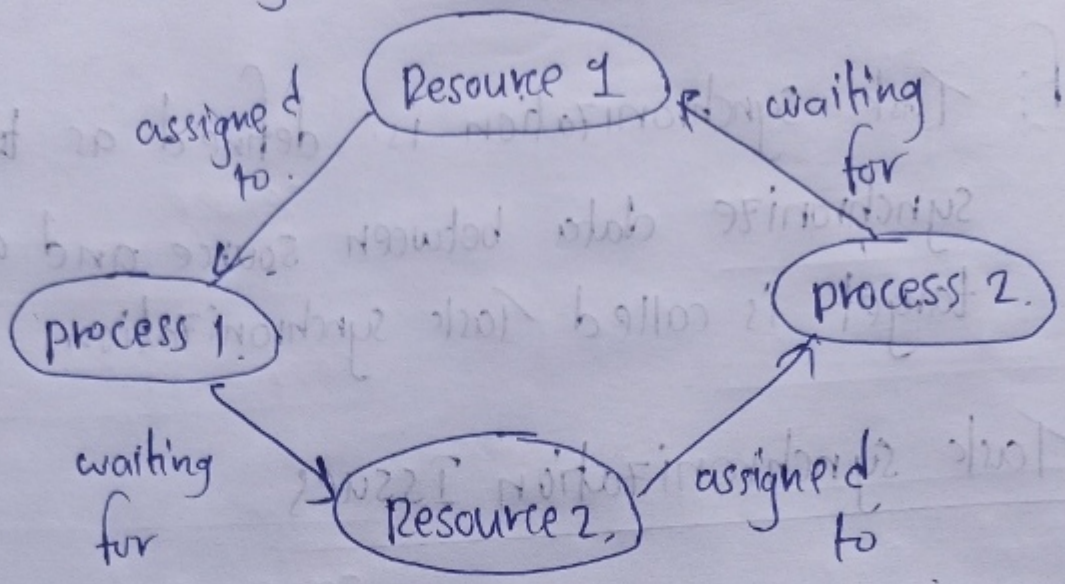
-> Racing condition produces incorrect results.

(2) Deadlock :

-> Deadlock is a database system, having two (or) more transactions running same time, when processing is waiting for data being locked by some other transaction.

-> A deadlock is a cyclic waiting

-> Block diagram



-> Scenarios

- Mutual exclusion
- Hold & wait
- circuit wait