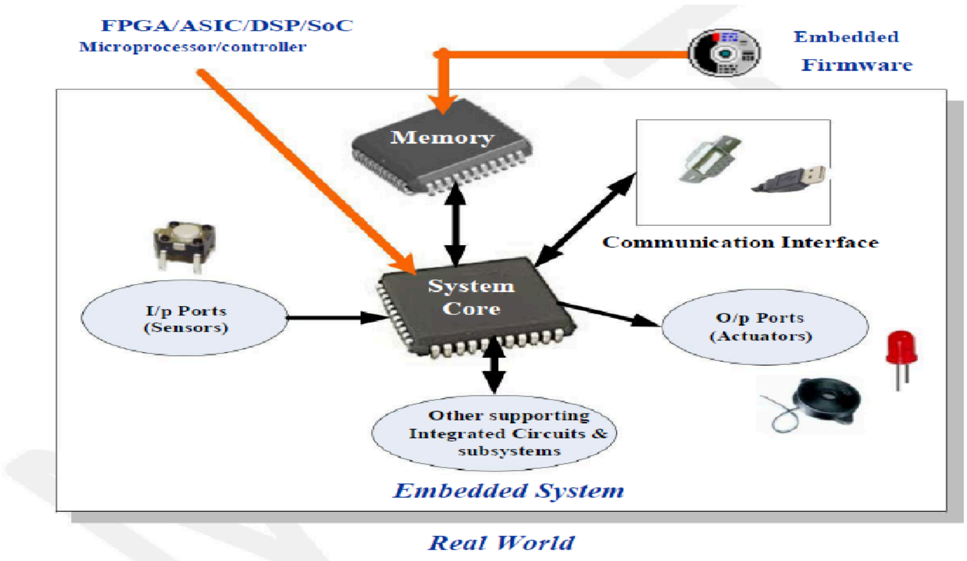
ELEMENTS OF EMBEDDED SYSTEMS:

- An embedded system is a combination of 3 things, Hardware,Software and Mechanical Components and it is supposed to do one specific task only.
- A typical embedded system contains a single chip controller which acts as the master brain of the system.
- Diagrammatically an embedded system can be represented as follows:



- Embedded systems are basically designed to regulate a physical variable (such Microwave Oven) or to manipulate the state of some devices by sending some signals to the actuators or devices connected to the output port system (such as temperature in Air Conditioner), in response to the input signal provided by the end users or sensors which are connected to the input ports. Hence the embedded systems can be viewed as a reactive system.

- The control is achieved by processing the information coming from the sensors and user interfaces and controlling some actuators that regulate the physical variable.

- Keyboards, push button, switches, etc. are Examples of common user interface input devices and LEDs, LCDs, Piezoelectric buzzers, etc examples for common user interface output devices for a typical embedded system.The requirement of type of user interface changes from application to application based on domain.

- Some embedded systems do not require any manual intervention for their operation. They automatically sense the input parameters from the real world through sensors which are connected the input port.

- The sensor information is passed to the processor after signal conditioning and digitization.

- The core of the system performs some predefined operations on input data with the help of embedded firmware in the system and sends some actuating signals to the actuator connect connected to the output port of the system.

- The memory of the system is responsible for holding the code (control algorithm and other important configuration details).

- There are two types of memories used in any embedded system.

- Fixed memory (ROM) is used for storing code or program. The user cannot change the firmware in this type of memory. The most common types of memories used in embedded systems for control algorithm storage are OTP,PROM,UVEPROM,EEPROM and FLASH.

- Some times the system requires temporary memory for performing arithmetic operations or control algorithm execution and this type of memory is known as ‘working memory’.RAM is used in most of the systems as working memory.Various types of RAM like SRAM,DRAM and NVRAM are used for this purpose.

- An embedded system without code (i.e. the control algorithm) implemented memory has all the peripherals, but is not capable of making decisions depending on the situational as well as the real world changes.

- Memory for implementing the code may be present on the processor or may be implemented as a separate chip interfacing the processor
- In a controller based embedded system, the controller may contain internal memory for storing code such controllers are called Micro-controllers with on-chip ROM, eg. Atmel AT89C51.

The Core of the Embedded Systems:

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

- **General Purpose and Domain Specific Processors**
  - Microprocessors
  - Microcontrollers
  - Digital Signal Processors
- **Programmable Logic Devices (PLDs)**
- **Application Specific Integrated Circuits (ASICs)**
- **Commercial off the shelf Components(COTS)**

General Purpose and Domain Specific Processor

- Almost 80% of the embedded systems are processor/ controller based.
- The processor may be microprocessor or a microcontroller or digital signal processor, depending on the domain and application.

Microprocessor:

- A silicon chip representing a Central Processing Unit (CPU), which is capable of performing arithmetic as well as logical operations,according to a pre-defined set of Instructions, which is specific to the manufacturer.
- In general the CPU contains the Arithmetic and Logic Unit (ALU), Control Unit and Working registers
- Microprocessor is a **dependent unit** and it requires the combination of other hardware like Memory, Timer Unit, and Interrupt Controller etc for proper functioning.
- Intel claims the credit for developing the first Microprocessor unit Intel 4004, a 4 bit processor which was released in Nov 1971.It featured 1K data memory,12 bit program counter and 4K program memory,sixteen 4bit general purpose registers and 46 instructions.It ran at a clock speed of 740 kHz.
- Developers of microprocessors.

  Intel – Intel 4004 – November 1971(4-bit)
  Intel – Intel 4040.
  Intel – Intel 8008 – April 1972.
  Intel – Intel 8080 – April 1974(8-bit).
  Motorola – Motorola 6800.
  Intel – Intel 8085 – 1976.
  Zilog - Z80 – July 1976

Microcontroller:

- A highly integrated silicon chip containing a CPU, scratch pad RAM, Special and General purpose Register Arrays, On Chip ROM/FLASH memory for program storage, Timer and Interrupt control units and dedicated I/O ports
- Microcontrollers can be considered as a super set of Microprocessors
- Microcontroller can be general purpose (like Intel 8051, designed for generic applications and domains) or application specific (Like Automotive AVR from Atmel Corporation. Designed specifically for automotive applications)
- Since a microcontroller contains all the necessary functional blocks for independent working, they found greater place in the embedded domain in place of microprocessors
- Microcontrollers are cheap, cost effective and are readily available in the market
- Texas Instruments TMS 1000 is considered as the world"s first microcontroller

Microprocessors	Microcontrollers
A silicon chip representing a Central Processing Unit(CPU), which is capable of performing arithmetic as well as logical operations according to a pre-defined set of Instructions	A microcontroller is a highly integrated chip that contains a CPU, scratch pad RAM, Special and General purpose Register Arrays, On Chip ROM/FLASH memory for program storage, Timer and interrupt control unit and dedicated IO ports.
It is a **dependent** unit. It requires the combination of other chips like Timers, Program and data memory chips, Interrupt controllers etc for functioning.	It is a **self contained** unit and it doesn't require external Interrupt Controller, Timer, UART etc for its functioning.
Most of the time **general purpose** in design and operation.	Mostly **application oriented or domain specific** in operation.
Doesn"t contain a built in I/O port. The I/O Port functionality needs to be implemented with the help of external Programmable Peripheral Interface Chips like 8255	Most of the processors contain multiple **built-in I/O ports** which can be operated as a single 8 or 16 or 32 bit Port or as individual port pins
Targeted for high end market where performance is important.	Targeted for embedded market where performance is not so critical (At present this demarcation is invalid)
Limited power saving options compared to microcontrollers.	Includes lot of power saving features

General Purpose Processor (GPP) Vs Application Specific Instruction Set Processor (ASIP)

- General Purpose Processor or GPP is a processor designed for general computational tasks
- GPPs are produced in large volumes and targeting the general market. Due to the high volume production, the per unit cost for a chip is low compared to ASIC or other specific ICs
- A typical general purpose processor contains an Arithmetic and Logic Unit (ALU) and Control Unit (CU)
- Application Specific Instruction Set processors (ASIPs) are processors with architecture and instruction set optimized to specific domain/application requirements like Network processing, Automotive, Telecom, media applications, digital signal processing, control applications etc

- ASIPs fill the architectural spectrum between General Purpose Processors and Application Specific Integrated Circuits (ASICs)

- The need for an ASIP arises when the traditional general purpose processor are unable to meet the increasing application needs

- Some Microcontrollers (like Automotive AVR, USB AVR from Atmel), System on Chips, Digital Signal Processors etc are examples of Application Specific Instruction Set Processors (ASIPs)

- ASIPs incorporate a processor and on-chip peripherals, demanded by the application requirement, program and data memory.

Digital Signal Processors (DSPs):

- Powerful special purpose 8/16/32 bit microprocessors designed specifically to meet the computational demands and power constraints of today's embedded audio, video, and communications applications .
- Digital Signal Processors are 2 to 3 times faster than the general purpose microprocessors in signal processing applications
- DSPs implement algorithms in hardware which speeds up the execution whereas general purpose processors implement the algorithm in firmware and the speed of execution depends primarily on the clock for the processors
- DSP can be viewed as a microchip designed for performing high speed computational operations for „addition", „subtraction", „multiplication" and „division
- A typical Digital Signal Processor incorporates the following key units
- Program Memory
- Data Memory
- Computational Engine
- I/O Unit

- Audio video signal processing, telecommunication and multimedia applications are typical examples where DSP is employed.

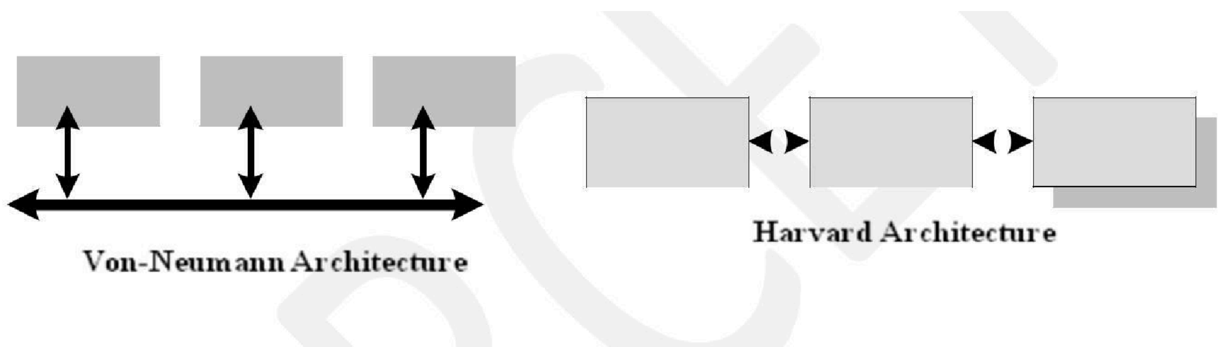
RISC V/s CISC Processors/Controllers:

RISC	CISC
Lesser Number of Instructions	Greater no. of Instructions
Instruction Pipelining and increased execution speed	Generally no instruction pipelining feature
Orthogonal Instruction Set (Allows each instruction to operate on any register and use any addressing mode,	Non Orthogonal Instruction Set (All instructions are not allowed to operate on any register and use any addressing mode. It is instruction specific.)
Operations are performed on registers only, the only memory operations are load and store	Operations are performed on registers or memory depending on the instruction
Large number of registers are available	Limited no. of general purpose registers
Programmer needs to write more code to execute a task since the instructions are simpler ones	A programmer can achieve the desired functionality with a single instruction which in turn provides the effect of using more simpler single instructions in RISC.
Single, Fixed length Instructions	Variable length Instructions

Less Silicon usage and pin count	More silicon usage since more additional decoder logic is required to implement the complex instruction decoding.
With Harvard Architecture	Can be Harvard or Von-Neumann Architecture

Harvard V/s Von-Neumann Processor/Controller Architecture

- The terms Harvard and Von-Neumann refers to the processor architecture design.
- Microprocessors/controllers based on the **Von-Neumann** architecture shares a single common bus for fetching both instructions and data.Program instructions and data are stored in a common main memory.
- Microprocessors/controllers based on the **Harvard** architecture will have separate data bus and instruction bus. This allows the data transfer and program fetching to occur simultaneously on both buses.
- With Harvard architecture, the data memory can be read and written while the program memory is being accessed. These separated data memories and code memory buses allow one instruction to execute while the next instruction is fetched (“**Pre-fetching”)**



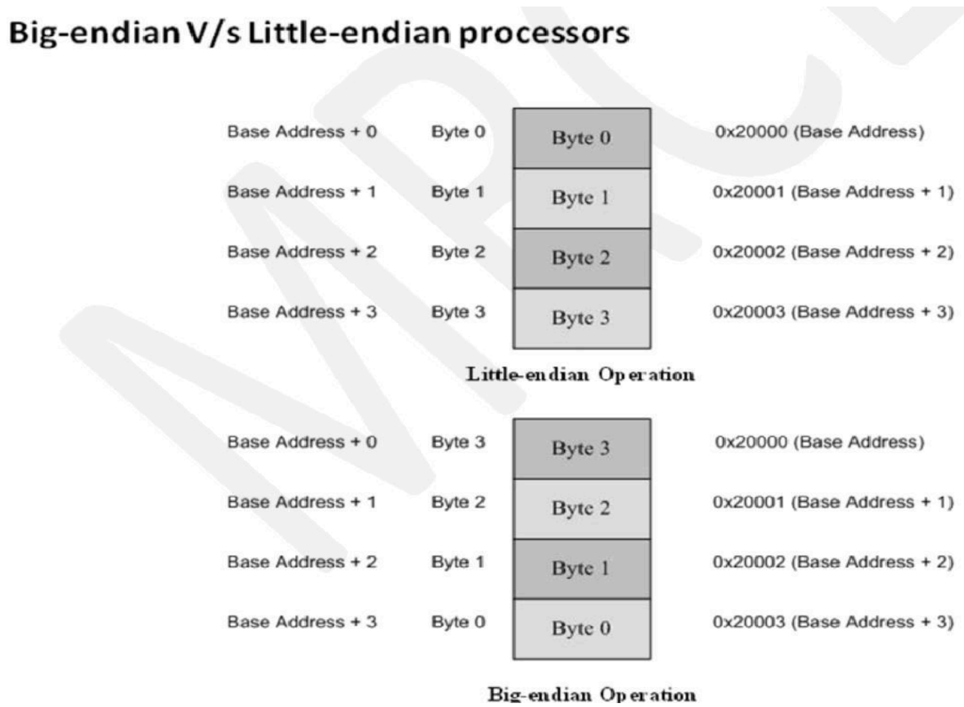
Harvard V/s Von-Neumann Processor/Controller Architecture:

Harvard Architecture	**Von-Neumann Architecture**
Separate buses for Instruction and Data fetching	Single shared bus for Instruction and Data fetching
Easier to pipeline so high performance can be achieved.	Low performance compared to Harvard architecture.
Comparatively high cost	Cheaper
No memory alignment problems	Allow self modifying codes.

Since data memory and program memory are stored physically in different locations,no chances for accidental corruption of program memory.	Since data memory and program memory are stored physically in the same chip, chances for accidental corruption of program memory

Big-endian V/s Little-endian processors:

- Endianness specifies the order in which the data is stored in the memory by processor operations in a multi byte system (Processors whose word size is greater than one byte).
- Suppose the word length is two byte then data can be stored in memory in two different ways
- Higher order of data byte at the higher memory and lower order of data byte at location just below the higher memory
- Lower order of data byte at the higher memory and higher order of data byte at location just below the higher memory
- *Little-endian* means the lower-order byte of the data is stored in memory at the lowest address, and the higher-order byte at the highest address. (The little end comes first)
- *Big-endian* means the higher-order byte of the data is stored in memory at the lowest address, and the lower-order byte at the highest address. (The big end comes first.)



01101001 10001000

Little Endian

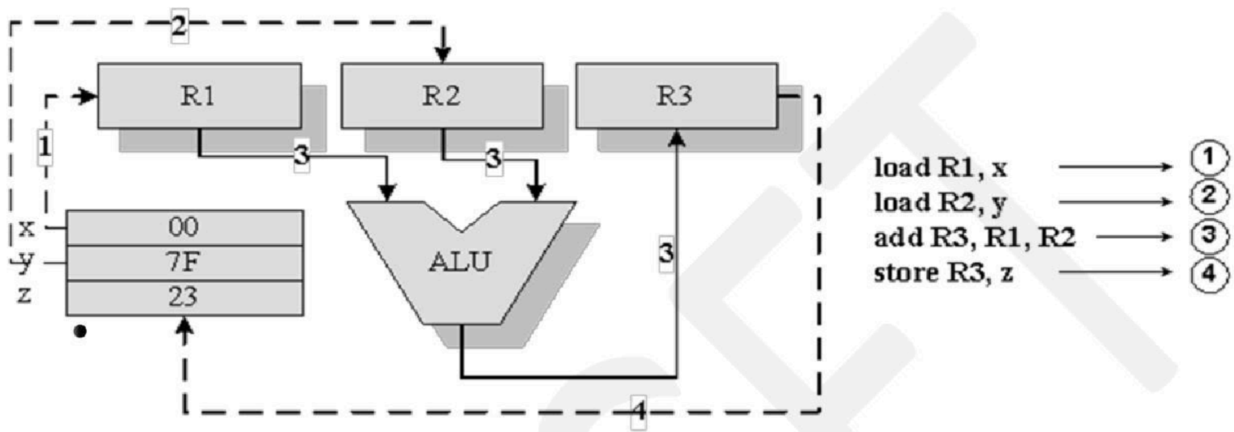
10001000	2000
01101001	2001

Big Endian

01101001	2000
10001000	2001

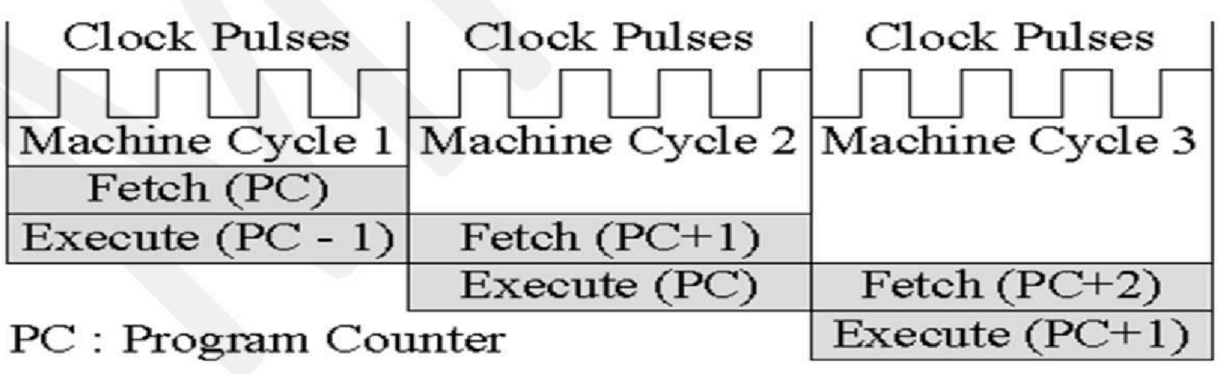
Load Store Operation & Instruction Pipelining:

- The RISC processor instruction set is orthogonal and it operates on registers. The memory access related operations are performed by the special instructions *load* and *store*. If the operand is specified as memory location, the content of it is loaded to a register using the *load* instruction. The instruction *store* stores data from a specified register to a specified memory location.



Load Store Operation

- The conventional instruction execution by the processor follows the fetch-decode-execute sequence
- The „fetch“ part fetches the instruction from program memory or code memory and the decode part decodes the instruction to generate the necessary control signals.

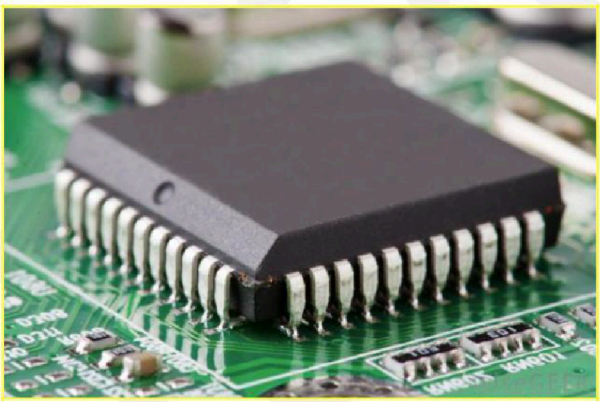


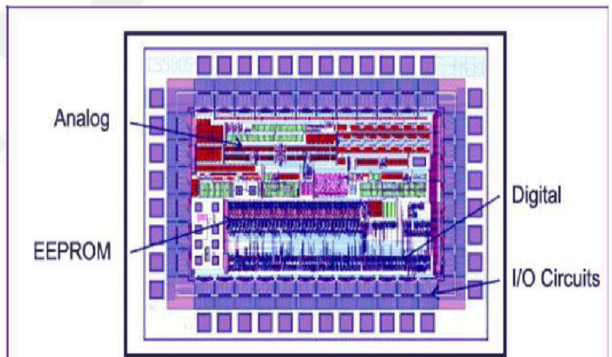
The Single stage pipelining concept

- The execute stage reads the operands, perform ALU operations and stores the result. In conventional program execution, the fetch and decode operations are performed in sequence
- During the decode operation the memory address bus is available and if it possible to effectively utilize it for an instruction fetch, the processing speed can be increased
- In its simplest form instruction pipelining refers to the overlapped execution of instructions

Application Specific Integrated Circuit (ASIC):

- A microchip designed to perform a specific or unique application. It is used as replacement to conventional general purpose logic chips.
- ASIC integrates several functions into a single chip and thereby reduces the system development cost
- Most of the ASICs are proprietary products. As a single chip, ASIC consumes very small area in the total system and thereby helps in the design of smaller systems with high capabilities/functionalities.
- ASICs can be pre-fabricated for a special application or it can be custom fabricated by using the components from a re-usable „*building block*" library of components for a particular customer application





A Typical Mixed Signal ASIC

- Fabrication of ASICs requires a non-refundable initial investment (Non Recurring Engineering (NRE) charges) for the process technology and configuration expenses

- If the Non-Recurring Engineering Charges (NRE) is born by a third party and the Application Specific Integrated Circuit (ASIC) is made openly available in the market, the ASIC is referred as Application Specific Standard Product (ASSP)

- The ASSP is marketed to multiple customers just as a general-purpose product , but to a smaller number of customers since it is for a specific application.

- Some ASICs are proprietary products , the developers are not interested in revealing the internal details.

Programmable Logic Devices (PLDs):

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

- Logic devices can be classified into two broad categories - **Fixed and Programmable**. The circuits in a fixed logic device are permanent, they perform one function or set of functions - once manufactured, they cannot be changed.

- Programmable logic devices (PLDs) offer customers a wide range of logic capacity, features, speed, and voltage characteristics - and these devices can be re-configured to perform any number of functions at any time

- Designers can use inexpensive software tools to quickly develop, simulate, and test their logic designs in PLD based design. The design can be quickly programmed into a device, and immediately tested in a live circuit

- PLDs are based on re-writable memory technology and the device is reprogrammed to change the design

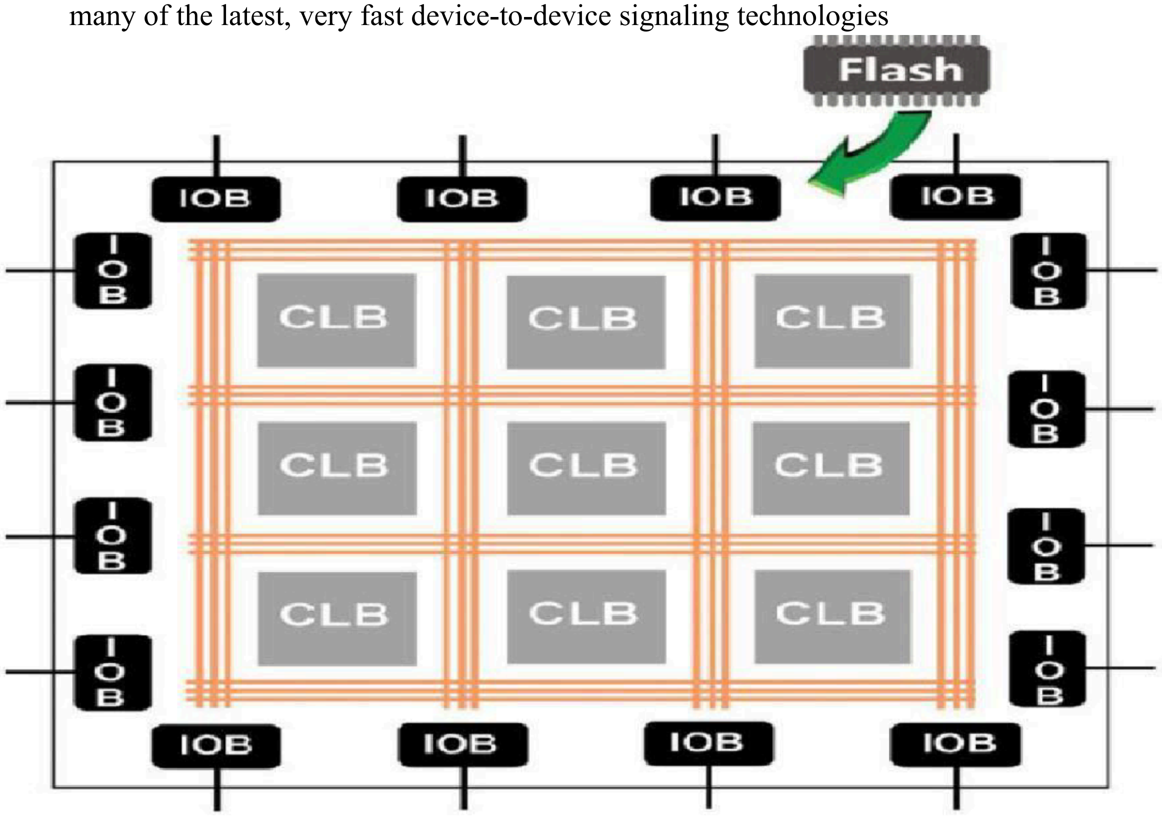
Programmable Logic Devices (PLDs) – CPLDs and FPGA

Field Programmable Gate Arrays (FPGAs) and Complex Programmable Logic Devices (CPLDs) are the two major types of programmable logic devices

FPGA:

- FPGA is an IC designed to be configured by a designer after manufacturing.
- FPGAs offer the highest amount of logic density, the most features, and the highest performance.
- Logic gate is Medium to high density ranging from **1K to 500K** system gates

- These advanced FPGA devices also offer features such as built-in hardwired processors (such as the IBM Power PC), substantial amounts of memory, clock management systems, and support for many of the latest, very fast device-to-device signaling technologies

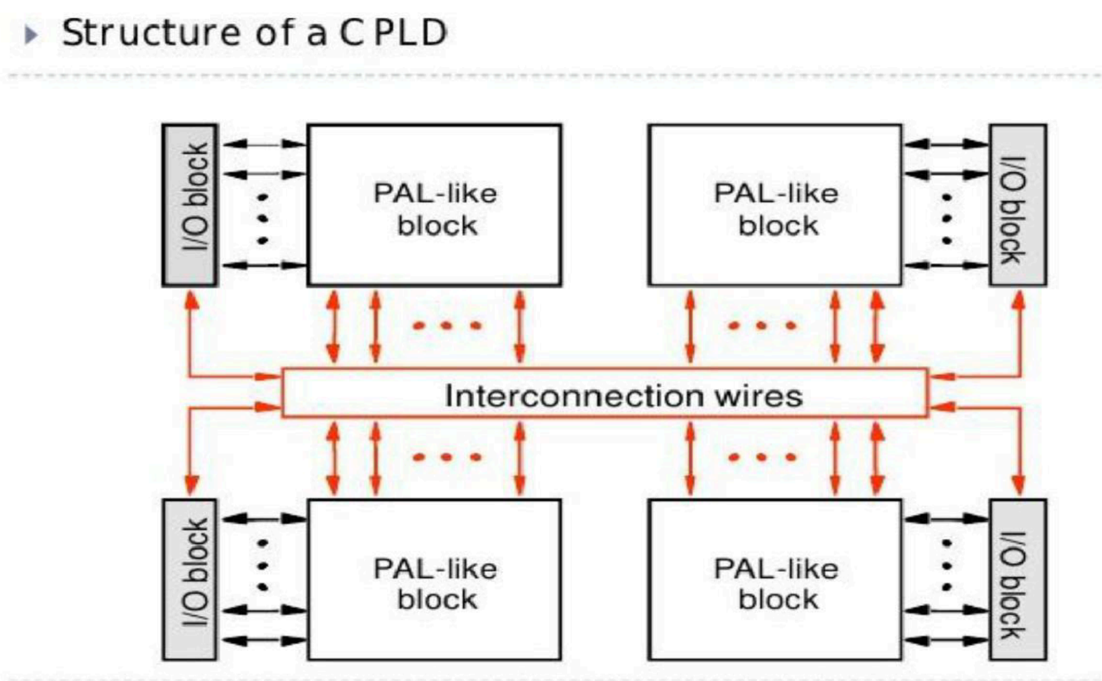


These advanced FPGA devices also offer features such as built-in hardwired processors, substantial amounts of memory, clock management systems, and support for many of the latest, very fast device-to-device signaling technologies.
FPGAs are used in a wide variety of applications ranging from data processing and storage, to instrumentation, telecommunications, and digital signal processing

CPLD:

- A **complex programmable logic device** (**CPLD**) is a programmable logic device with complexity between that of PALs and FPGAs, and architectural features of both.

- CPLDs, by contrast, offer much smaller amounts of logic - up to about 10,000 gates.
- CPLDs offer very predictable timing characteristics and are therefore ideal for critical control applications.



- CPLDs such as the Xilinx **CoolRunner** series also require extremely low amounts of power and are very inexpensive, making them ideal for cost-sensitive, battery-operated, portable applications such as mobile phones and digital handheld assistants.

ADVANTAGES OF PLDs:

- PLDs offer customer much more flexibility during design cycle

- PLDSs do not require long lead times for prototype or production-the PLDs are already on a distributor‟s self and ready for shipment

- PLDs do not require customers to pay for large NRE costs and purchase expensive mask sets

- PLDs allow customers to order just the number of parts required when they need them. allowing them to control inventory.

- PLDs are reprogrammable even after a piece of equipment is shipped to a customer.

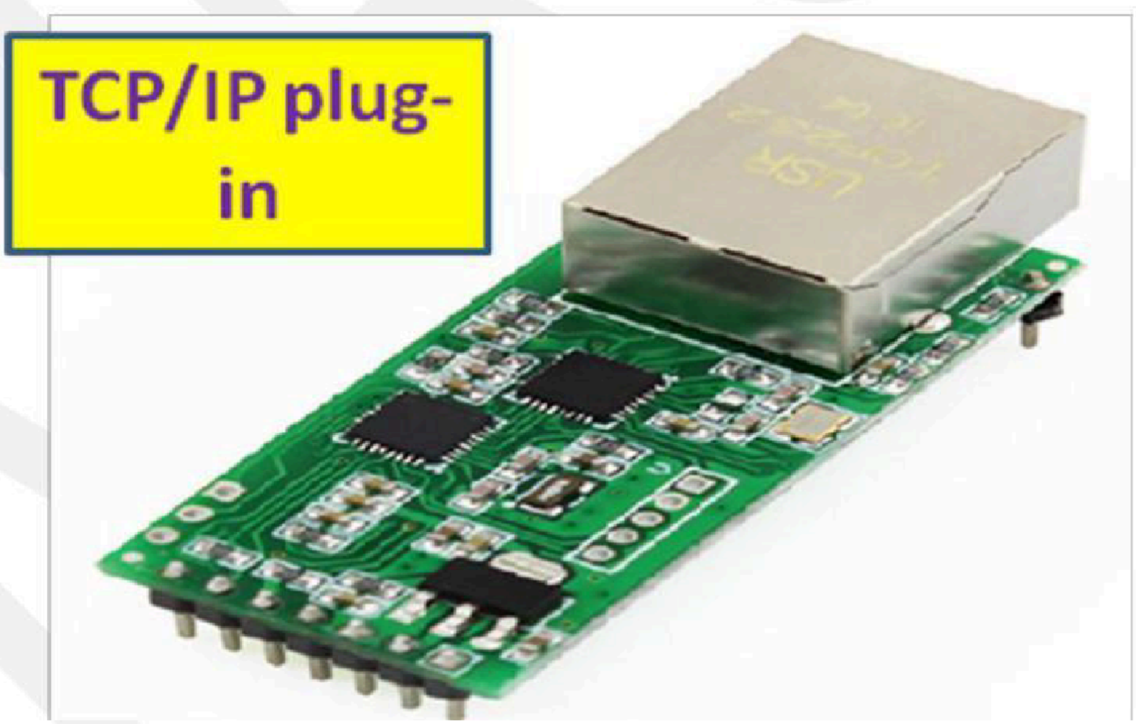
- The manufacturers able to add new features or upgrade the PLD based products that are in the field by uploading new programming file

Commercial off the Shelf Component (COTS):

- COTS products are designed in such a way to provide easy integration and interoperability with existing system components
- Typical examples for the COTS hardware unit are Remote Controlled Toy Car control unit including the RF Circuitry part, High performance, high frequency microwave electronics (2 to

200 GHz), High bandwidth analog-to-digital converters, Devices and components for operation at very high temperatures, Electro-optic IR imaging arrays, UV/IR Detectors etc

- A COTS component in turn contains a General Purpose Processor (GPP) or Application Specific Instruction Set Processor (ASIP) or Application Specific Integrated Chip (ASIC)/Application Specific Standard Product (ASSP) or Programmable Logic Device (PLD)



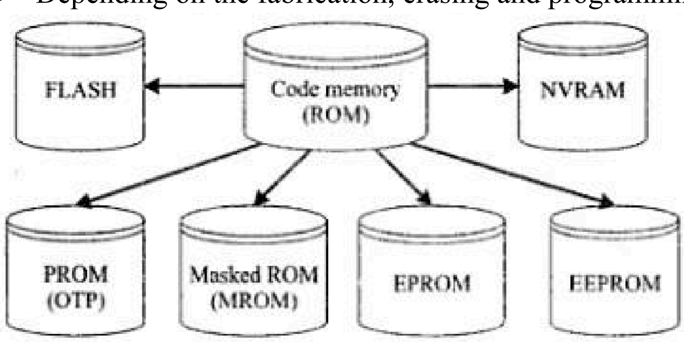
- The major advantage of using COTS is that they are readily available in the market, cheap and a developer can cut down his/her development time to a great extend.
- There is no need to design the module yourself and write the firmware .
- Everything will be readily supplied by the COTs manufacturer.
- The major problem faced by the end-user is that there are no operational and manufacturing standards.
- The major drawback of using COTs component in embedded design is that the manufacturer may withdraw the product or discontinue the production of the COTs at any time if rapid change in technology
- This problem adversely affect a commercial manufacturer of the embedded system which makes use of the specific COTs

Memory:

- Memory is an important part of an embedded system. The memory used in embedded system can be either Program Storage Memory (ROM) or Data memory (RAM)
- Certain Embedded processors/controllers contain built in program memory and data memory and this memory is known as on-chip memory
- Certain Embedded processors/controllers do not contain sufficient memory inside the chip and requires external memory called **off-chip memory or external memory.**

Program Storage Memory(ROM)

- Stores the program instructions
- Retains its contents even after the power to it is turned off. It is generally known as Non volatile storage memory
- Depending on the fabrication, erasing and programming techniques they are classified into



Masked ROM (MROM):

- One-time programmable memory.
- Uses hardwired technology for storing data.
- The **device is factory programmed by masking and metallization process according to the data provided by the end user.**
- The primary advantage of MROM is low cost for high volume production.
- MROM is the least expensive type of solid state memory.
- Different mechanisms are used for the masking process of the ROM, like
    - Creation of an enhancement or depletion mode transistor through channel implant
    - By creating the memory cell either using a standard transistor or a high threshold transistor.
    - In the high threshold mode, the supply voltage required to turn ON the transistor is above the normal ROM IC operating voltage.
    - This ensures that the transistor is always off and the memory cell stores always logic 0.
- The limitation with MROM based firmware storage is the inability to modify the device firmware against firmware upgrades.
- The MROM is permanent in bit storage, it is not possible to alter the bit information

Programmable Read Only Memory (PROM) / (OTP-One time Programmable Memory):

- It is **not pre-programmed by the manufacturer**
- The **end user is responsible for Programming these devices**.
- PROM/OTP has ***nichrome*** **or** ***polysilicon*** **wires arranged in a matrix, these wires can be functionally viewed as fuses.**
- It is programmed by a PROM programmer which selectively burns the fuses according to the bit pattern to be stored.
- Fuses which are not blown/burned represents a logic “1” where as fuses which are blown/burned represents a logic “0”.The default state is logic “1”.
- OTP is widely used for commercial production of embedded systems whose proto-typed versions are proven and the code is finalized.
- It is a low cost solution for commercial production.
- OTPs cannot be reprogrammed

Erasable Programmable Read Only Memory (EPROM):

- Erasable Programmable Read Only (EPROM) memory gives the flexibility to re-program the same chip.
- During development phase , code is subject to continuous changes and using an OTP is not economical.
- EPROM stores the bit information by charging the floating gate of an FET
- Bit information is stored by using an EPROM Programmer, which applies high voltage to charge the floating gate
- EPROM contains a quartz crystal window for erasing the stored information. If the window is exposed to Ultra violet rays for a fixed duration, the entire memory will be erased
- Even though the EPROM chip is flexible in terms of re-programmability, it needs to be taken out of the circuit board and needs to be put in a UV eraser device for 20 to 30 minutes

Electrically Erasable Programmable Read Only Memory (EEPROM):

- Erasable Programmable Read Only (EPROM) memory gives the flexibility to re-program the same chip using electrical signals
- The information contained in the EEPROM memory can be altered by using electrical signals at the register/Byte level
- They can be erased and reprogrammed within the circuit
- These chips include a chip erase mode and in this mode they can be erased in a few milliseconds
- It provides greater flexibility for system design
- The only limitation is their capacity is limited when compared with the standard ROM (A few kilobytes).

Program Storage Memory – FLASH

- FLASH memory is a variation of EEPROM technology.
- FALSH is the latest ROM technology and is the most popular ROM technology used in today‟s embedded designs
- It combines the re-programmability of EEPROM and the high capacity of standard ROMs
- FLASH memory is organized as sectors (blocks) or pages
- FLASH memory stores information in an array of floating gate MOSFET transistors
- The erasing of memory can be done at sector level or page level without affecting the other sectors or pages
- Each sector/page should be erased before re-programming
- The typical erasable capacity of FLASH is of the order of a few 1000 cycles.

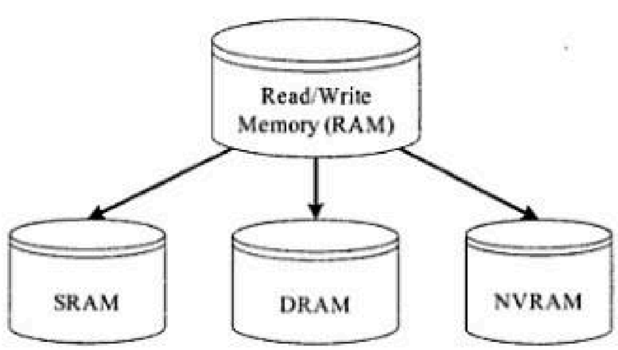
Non Volatile RAM (NVRAM):

- Random access memory with battery backup
- It contains Static RAM based memory and a minute battery for providing supply to the memory in the absence of external power supply
- The memory and battery are packed together in a single package
- NVRAM is used for the non volatile storage of results of operations or for setting up of flags etc
- The life span of NVRAM is expected to be around 10 years
- DS1744 from Maxim/Dallas is an example for 32KB NVRAM

Read-Write Memory/Random Access Memory (RAM)

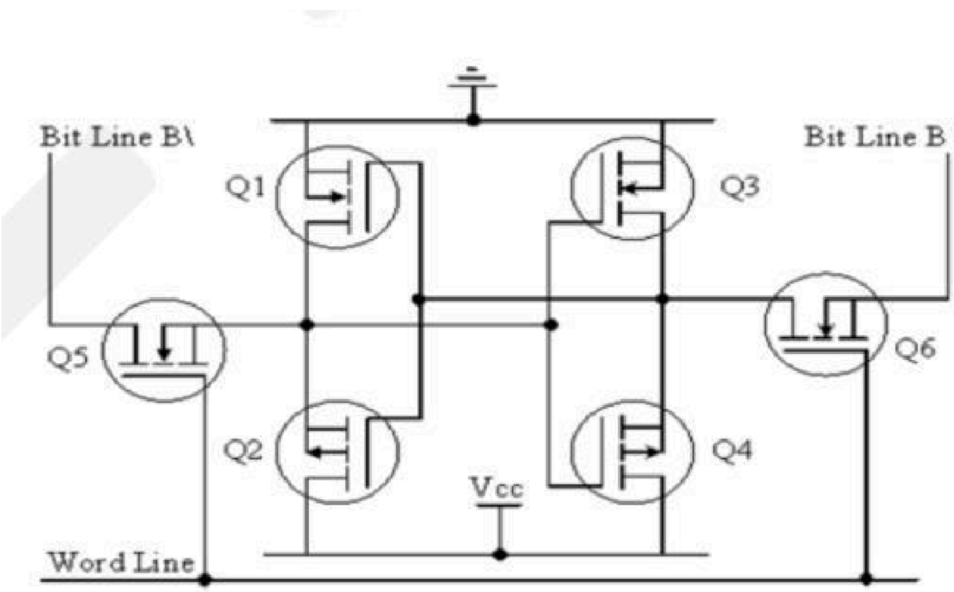
- RAM is the data memory or working memory of the controller/processor
- RAM is volatile, meaning when the power is turned off, all the contents are destroyed

- RAM is a direct access memory, meaning we can access the desired memory location directly without the need for traversing through the entire memory locations to reach the desired memory position (i.e. Random Access of memory location)

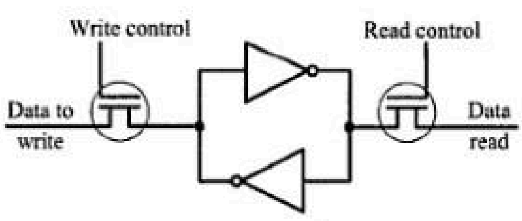


1. Static RAM (SRAM):

- Static RAM stores data in the form of Voltage.
- They are made up of flip-flops
- In typical implementation, an SRAM cell (bit) is realized using 6 transistors (or 6 MOSFETs).
- Four of the transistors are used for building the latch (flip-flop) part of the memory cell and 2 for controlling the access.
- Static RAM is the fastest form of RAM available.
- SRAM is fast in operation due to its resistive networking and switching capabilities



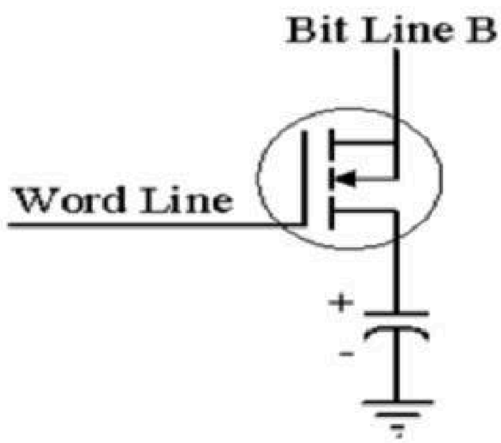
SRAM cell implementation



2. Dynamic RAM (DRAM)

- Dynamic RAM stores data in the form of charge. They are made up of MOS transistor gates.
- The advantages of DRAM are its high density and low cost compared to SRAM
- The disadvantage is that since the information is stored as charge it gets leaked off with time and to prevent this they need to be refreshed periodically

  Special circuits called DRAM controllers are used for the refreshing operation. The refresh operation is done periodically in milliseconds interval



DRAM cell implementation

SRAM vs DRAM

SRAM Cell	DRAM cell
Made up of 6 CMOS transistors (MOSFET)	Made up of a MOSFET and a capacitor
Doesn"t Require refreshing	Requires refreshing
Low capacity (Less dense)	High Capacity (Highly dense)
More expensive	Less Expensive
Fast in operation. Typical access time is 10ns	Slow in operation due to refresh requirements. Typical access time is 60ns. Write operation is faster than read operation.

Memory selection for Embedded Systems:

- o Selection of suitable memory is very much essential step in high performance applications, because the challenges and limitations of the system performance have often decided upon the type of memory architecture.

- o System memory requirement depends primarily on the nature of the application that is planned to run on the system.

- o Memory performance and capacity requirement for low cost systems are small, whereas memory throughput can be the most critical requirement in a complex, high performance system.

- o Following are the factors that are to be considered while selecting the memory devices,

1. Speed
2. Data storage size and capacity

3. Bus width
4. Power consumption
5. Cost

Embedded system requirements:

- Program memory for holding control algorithm or embedded OS and the applications designed to run on top of OS.

- Data memory for holding variables and temporary data during task execution.
- Memory for holding non-volatile data which are modifiable by the application.

- The memory requirement for an embedded system in terms of RAM (SRAM/DRAM) and ROM (EEPROM/FLASH/NVRAM) is solely dependent on the type of the embedded system and applications for which it is designed.
- There is no hard and fast rule for calculating the memory requirements.
- Lot of factors need to be considered for selecting the type and size of memory for embedded system.
- **Example:** Design of Embedded based electronic Toy.
- SOC or microcontroller can be selected based type(RAM &ROM) and size of on-chip memory for the design of embedded system.
- If on-chip memory is not sufficient then how much external memory need to be interfaced.
- If the ES design is RTOS based ,the RTOS requires certain amount of RAM for its execution and ROM for storing RTOS Image.
- The RTOS suppliers gives amount of run time RAM requirements and program memory requirements for the RTOS.
- Additional memory is required for executing user tasks and user applications.

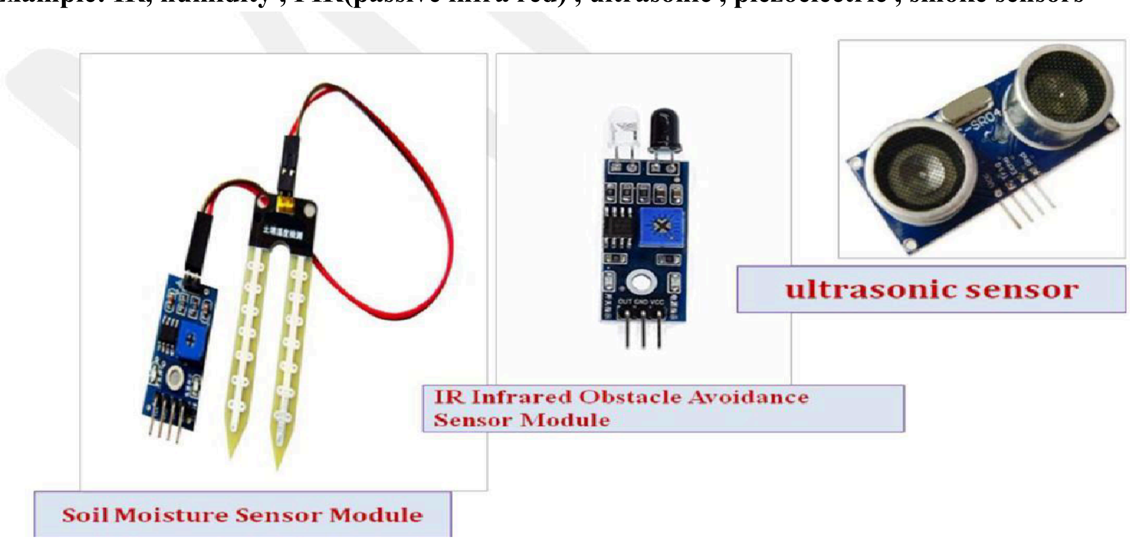
- On a safer side, always add a buffer value to the total estimated RAM and ROM requirements.
- A smart phone device with windows OS is typical example for embedded device requires say 512MB RAM and 1GB ROM are minimum requirements for running the mobile device.
- And additional RAM &ROM memory is required for running user applications.
- So estimate the memory requirements for install and run the user applications without facing memory space.
- Memory can be selected based on size of the memory ,data bus and address bus size of the processor/controller.
- Memory chips are available in standard sizes like 512 bytes,1KB,2KB ,4KB,8KB,16 KB ….1MB etc.
- FLASH memory is the popular choice for ROM in embedded applications .
- It is powerful and cost-effective solid state storage technology for mobile electronic devices and other consumer applications.
- Flash memory available in two major variants
  1. NAND FLASH 2. NOR FLASH
- NAND FLASH is a high density low cost non-volatile storage memory**.**
- NOR FLASH is less dense and slightly expensive but supports Execute in place(XIP).
- The XIP technology allows the execution of code memory from ROM itself without the need for copying it to the RAM.
- The EEPROM is available as either serial interface or parallel interface chip.
- If the processor/controller of the device supports serial interface and the amount of data to write and read to and from the device (Serial EEPROM) is less.
- The serial EEPROM saves the address space of the total system.
- The memory capacity of the serial EEPROM is expressed in bits or Kilobits.
- Industrial grade memory chips are used in certain embedded devices may be operated at extreme environmental conditions like high temperature.

Sensors & Actuators:

- Embedded system is in constant interaction with the real world
- Controlling/monitoring functions executed by the embedded system is achieved in accordance with the changes happening to the Real World.
- The changes in the system environment or variables are detected by the sensors connected to the input port of the embedded system.
- If the embedded system is designed for any controlling purpose, the system will produce some changes in controlling variable to bring the controlled variable to the desired value.
- It is achieved through an actuator connected to the out port of the embedded system.

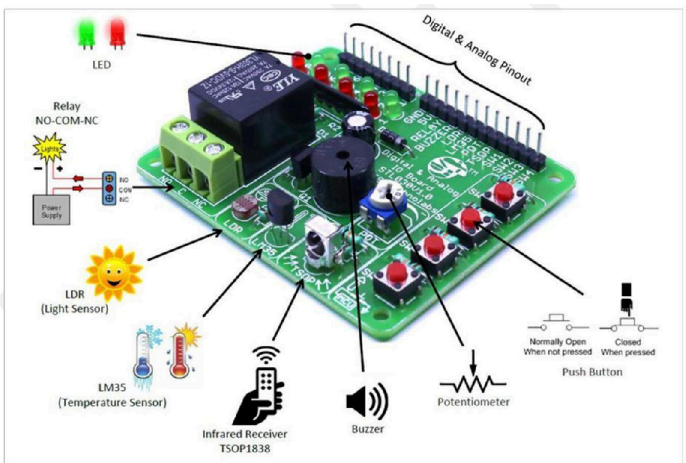
Sensor:

A transducer device which converts energy from one form to another for any measurement or control purpose. Sensors acts as input device.
Eg. Hall Effect Sensor which measures the distance between the cushion and magnet in the Smart Running shoes from adidas
Example: IR, humidity , PIR(passive infra red) , ultrasonic , piezoelectric , smoke sensors



Actuator:

A form of transducer device (mechanical or electrical) which converts signals to corresponding physical action (motion). Actuator acts as an output device
Eg. Micro motor actuator which adjusts the position of the cushioning element in the Smart Running shoes from adidas



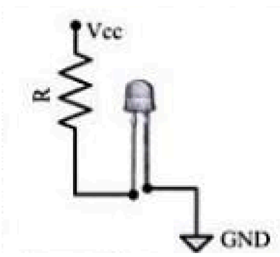
Silicon TechnoLabs Digital Analog Arduino Starter kit

The I/O Subsystem:

- The I/O subsystem of the embedded system facilitates the interaction of the embedded system with external world
- The interaction happens through the sensors and actuators connected to the Input and output ports respectively of the embedded system
- The sensors may not be directly interfaced to the Input ports, instead they may be interfaced through signal conditioning and translating systems like ADC, Optocouplers etc.

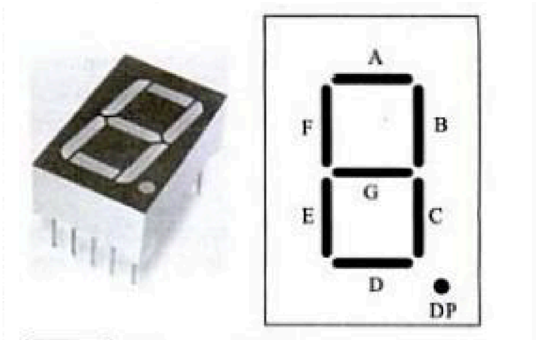
1. I/O Devices - Light Emitting Diode (LED)

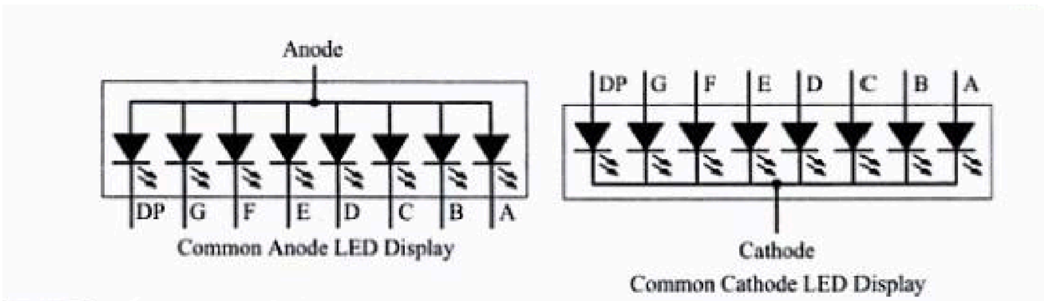
- Light Emitting Diode (LED) is an output device for visual indication in any embedded system
- LED can be used as an indicator for the status of various signals or situations.
- Typical examples are indicating the presence of power conditions like „Device ON", „Battery low" or „Charging of battery" for a battery operated handheld embedded devices
- LED is a p-n junction diode and it contains an anode and a cathode.
- For proper functioning of the LED, the anode of it should be connected to +ve terminal of the supply voltage and cathode to the –ve terminal of supply voltage
- The current flowing through the LED must limited to a value below the maximum current that it can conduct.
- A resister is used in series between the power supply and the resistor to limit the current through the LED



2. I/O Devices – 7-Segment LED Display

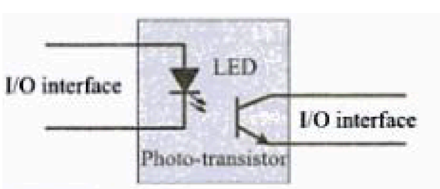
- The 7 – segment LED display is an output device for displaying alpha numeric characters
- It contains 8 light-emitting diode (LED) segments arranged in a special form.
- Out of the 8 LED segments, 7 are used for displaying alpha numeric characters
- The LED segments are named A to G and the decimal point LED segment is named as DP
- The LED Segments A to G and DP should be lit accordingly to display numbers and characters
- The 7 – segment LED displays are available in two different configurations, namely; Common anode and Common cathode
- In the Common anode configuration, the anodes of the 8 segments are connected commonly whereas in the Common cathode configuration, the 8 LED segments share a common cathode line
- Based on the configuration of the 7 – segment LED unit, the LED segment anode or cathode is connected to the Port of the processor/controller in the order „A" segment to the Least significant port Pin and DP segment to the most significant Port Pin.
- The current flow through each of the LED segments should be limited to the maximum value supported by the LED display unit
- The typical value for the current falls within the range of 20mA
- The current through each segment can be limited by connecting a current limiting resistor to the anode or cathode of each segment



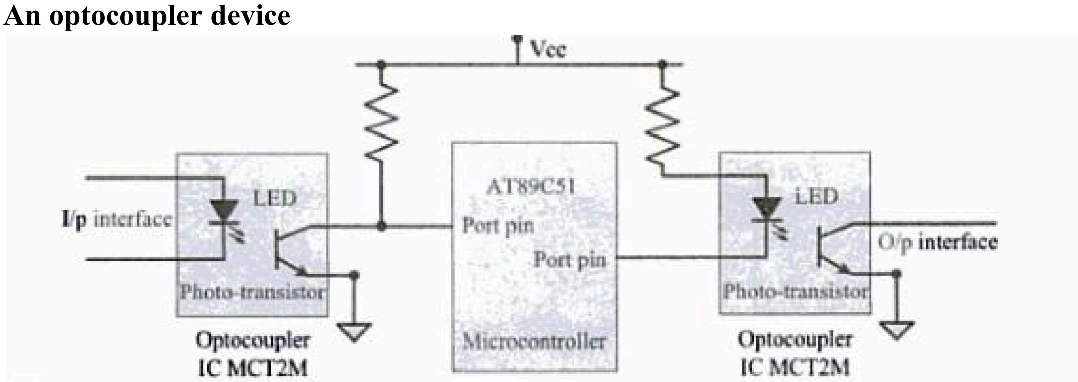


3. I/O Devices – Optocoupler

- Optocoupler is a solid state device to isolate two parts of a circuit.
- Optocoupler combines an LED and a photo-transistor in a single housing (package)
- In electronic circuits, optocoupler is used for suppressing interference in data communication, circuit isolation, High voltage separation, simultaneous separation and intensification signal etc



An optocoupler device



Optocoupler in Input and Output Circuit

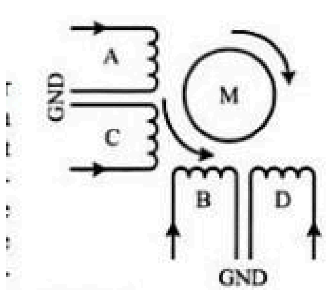
- Optocouplers can be used in either input circuits or in output circuits

4. I/O Devices – Stepper Motor:

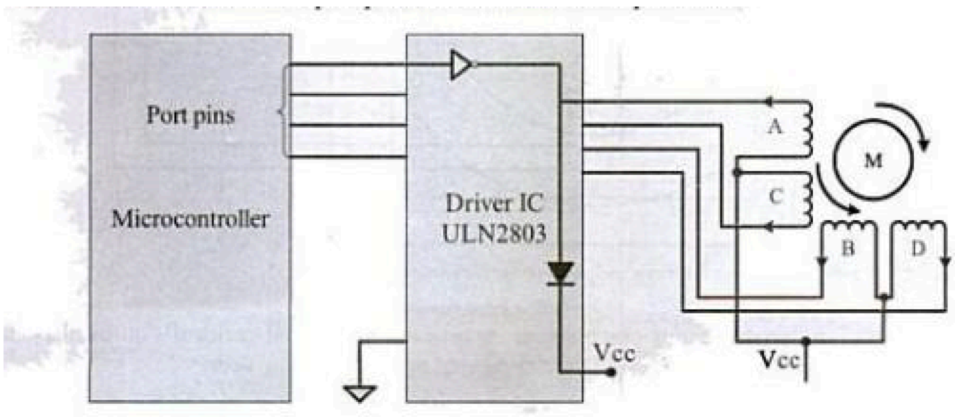
- Stepper motor is an electro mechanical device which generates discrete displacement (motion) in response to dc electrical signals
- It differs from the normal dc motor in its operation. The dc motor produces continuous rotation on applying dc voltage whereas a stepper motor produces discrete rotation in response to the dc voltage applied to it
- Stepper motors are widely used in industrial embedded applications, consumer electronic products and robotics control systems
- The paper feed mechanism of a printer/fax makes use of stepper motors for its functioning.
- Based on the coil winding arrangements, a two phase stepper motor is classified into
  1.Unipolar
  2.Bipolar

Unipolar: A unipolar stepper motor contains two windings per phase. The direction of rotation (clockwise or anticlockwise) of a stepper motor is controlled by changing the direction of current flow. Current in one direction flows through one coil and in the opposite direction flows through the other coil. It is easy to shift the direction of rotation by just switching the terminals to which the coils are connected

Bipolar: A bipolar stepper motor contains single winding per phase. For reversing the motor rotation the current flow through the windings is reversed dynamically. It requires complex circuitry for current flow reversal



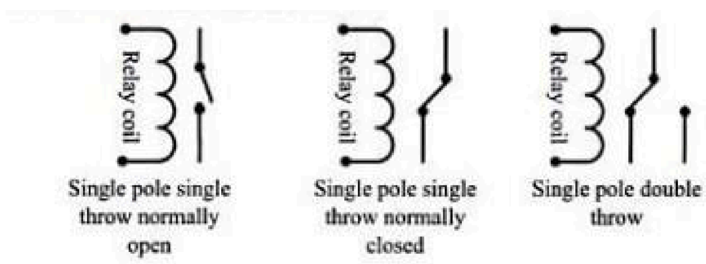
2 phase unipolar stepper motor



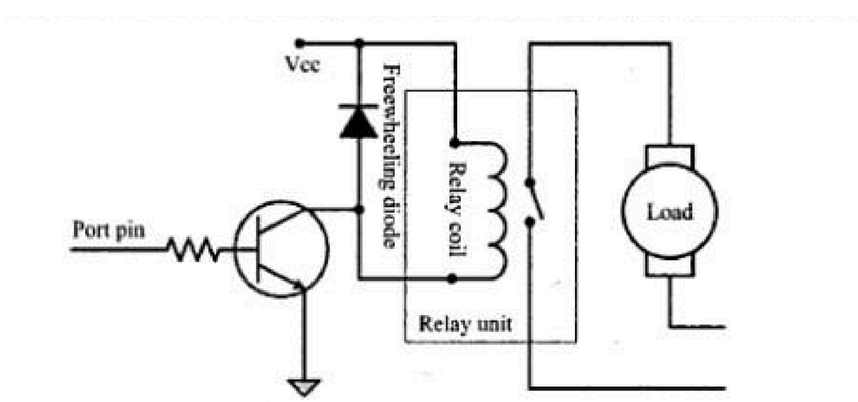
Interfacing of Stepper motor through driver circuit

5. The I/O Subsystem – I/O Devices – Relay:

- An electro mechanical device which acts as dynamic path selectors for signals and power.
- The „Relay" unit contains a relay coil made up of insulated wire on a metal core and a metal armature with one or more contacts.
- „Relay" works on electromagnetic principle.
- When a voltage is applied to the relay coil, current flows through the coil, which in turn generates a magnetic field.
- The magnetic field attracts the armature core and moves the contact point.
- The movement of the contact point changes the power/signal flow path.
- The Relay is normally controlled using a relay driver circuit connected to the port pin of the processor/controller
- A transistor can be used as the relay driver. The transistor can be selected depending on the relay driving current requirements.



Relay Configuration



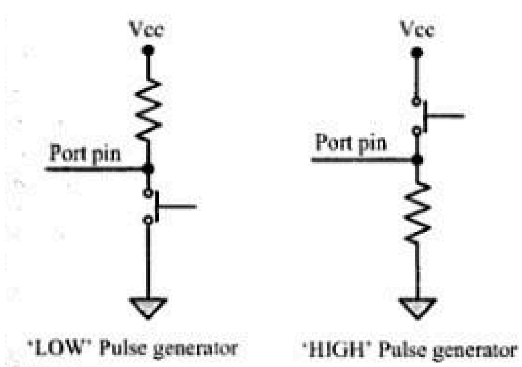
Transistor based Relay Circuit

6. The I/O Subsystem – I/O Devices -Piezo Buzzer:

- It is a piezoelectric device for generating audio indications in embedded applications.
- A Piezo buzzer contains a piezoelectric diaphragm which produces audible sound in response to the voltage applied to it.
- Piezoelectric buzzers are available in two types : 1.Self-driving 2.External driving
- Self-driving contains are the necessary components to generate sound at a predefined tone.
- External driving piezo Buzzers supports the generation of different tones.
- The tone can be varied by applying a variable pulse train to the piezoelectric buzzer.
- A Piezo Buzzer can be directly interfaced to the port pin of the processor/Controller.

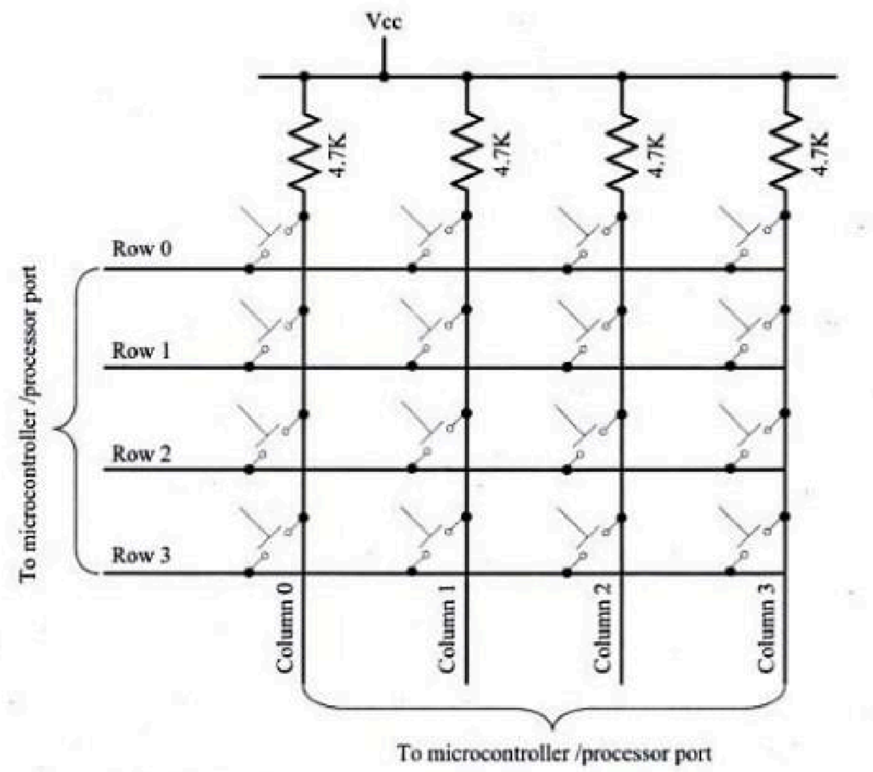
7. The I/O Subsystem – I/O Devices – Push button switch:

- Push Button switch is an input device.
- Push button switch comes in two configurations, namely „Push to Make" and „Push to Break"
- The switch is normally in the open state and it makes a circuit contact when it is pushed or pressed in the „Push to Make" configuration.
- In the „Push to Break" configuration, the switch normally in the closed state and it breaks the circuit contact when it is pushed or pressed .
- The push button stays in the „closed" (For Push to Make type) or „open" (For Push to Break type) state as long as it is kept in the pushed state and it breaks/makes the circuit connection when it is released.
- Push button is used for generating a momentary pulse



8.IO Interfacing Device-Keyboard

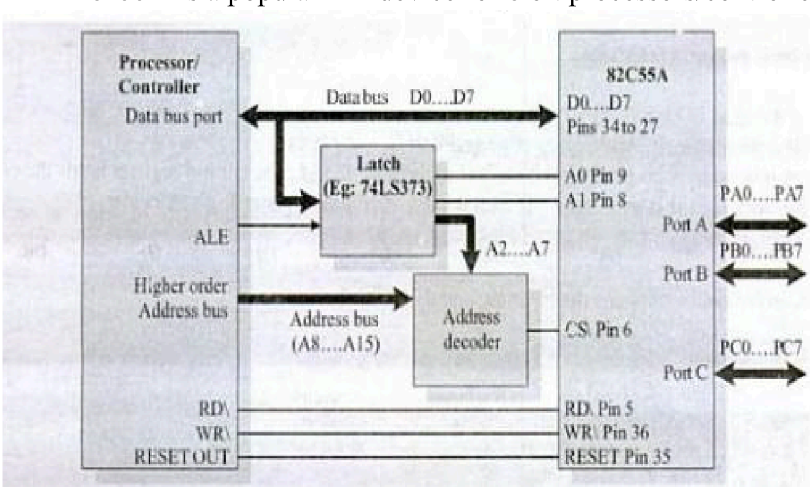
- Keyboard is an input device
- If the number of keys are limited push button can be used and they can be directly interfaced with the port pins.
- Matrix keyboard is used for handling large key requirements.it reduces the number of interface connections.



Matrix Keyboard Interfacing

9.I/O Interfacing device: Programmable Peripheral Interface(PPI)

- PPI devices are used for extending the IO capabilities of processors/controllers.
- Most of the processors/controllers provide a very limited number of IO and data ports and at times it may require more number of IO ports than one supported by the controller/processor.
- A PPI device expands the IO capabilities of the processor/controller.
- 8255A is a popular PPI device for 8 bit processors/controllers.



Interfacing of 8255 with 8 bit microcontroller