Embedded Systems

Unit-I

Introduction

* What is an Embedded System

→ An embedded System is an electronic / electro-mechanical System designed to Perform a specific function and is a combination of both H/W & firmware (software)

Every embedded System is unique, and the h/w as well as the firmware is highly specialised to the application domain.

EMBEDDED Systems Vs General Computing Systems

General Purpose Computing System	Embedded System
① A System which is a combination of a generic H/w and a General Purpose operating System for executing a variety of applications	① A System which is a combination of Special Purpose h/w & embedded OS for executing a specific set of applications
② Contains a General Purpose operating System (GPOS)	② May or May not Contain an operating System for functioning
③ Applications are alterable (programmable) by the user (It is possible for the end user to re-install the operating System and also add or remove user applications.	③ The firmware of the embedded System is pre-Programmed and it is non-alterable by the end user
④ Performance is the key deciding factor in the selection of the System. Always, Faster is Better	④ Application-specific requirements (like Performance, Power, Memory usage, etc) are the key deciding factors

Thrust – Push Suddenly

	General Purpose Computing System		Embedded System
(5)	Less/not at all tailored towards reduced operating power requirements, options for different levels of power management.	(5)	Highly tailored to take advantage of the power saving modes supported by the H/W and the operating system.
(6)	Response requirements are not time-critical	(6)	For certain category of embedded systems like mission critical systems, the response time requirement is highly critical.
(7)	Need not be deterministic in execution behaviour	(7)	Execution behaviour is deterministic for certain types of embedded systems like Hard Real Time Systems

History of embedded. Systems

→ (1) The first recognised modern embedded system is the Apollo Guidance Computer (AGC) developed by the MIT Instrumentation Laboratory for the lunar expedition. They ran the Interstial guidance systems of both the Command Module (CM) & Lunar Excursion Module (LEM) The Command Module was designed to encircle the moon while the Lunar Module and its crew were designed to go down to the moon surface and land there safely.

Encircle – around the circle

The Lunar Module and featured in total 18 engines There were 16 reaction control thrusters, a descent engine, ascent engine

→ The descent engine was 'designed to' provide thrust. to the lunar Module out of the lunar orbit & land it safely on the moon.

thrust-push

→ MIT's original design was based on 4K words of fixed Memory (Read only Memory) and 256 words of erasable Memory (Random Access Memory).

→ By June 1963, the figures reached 10K of fixed and 1K of erasable Memory. The final configuration was 36K words of fixed Memory and 2K words of erasable Memory.

→ The clock frequency of the first Microchip Proto model used in AGC was 1.024 MHz and it was derived from a 2.048 MHz crystal clock.

→ The computing unit of AGC consisted of ≈ 11 Instructions & 16 bit word logic. Around 5000 ICs (3-Input NOR gates, RTL Logic) supplied by Fairchild Semiconductor were used in this design.

→ The user interface unit of AGC is known as DSKY (display/keyboard). DSKY looked like a calculator type keypad with an array of numerals.

→ The first mass-produced embedded System was the Guidance computer for the Minuteman-I missile in 1961. It was the 'Autonetics D-17' guidance computer, built using discrete transistor logic & a Hard-disk for Main Memory.

→ The first IC ckt was produced in September 1958 but computer using them didn't begin to appear until 1963.

Classification of Embedded Systems-

1. Based on generation
2. complexity and Performance requirements. → timing constraints for a task.
3. Based on deterministic behaviour → Hard Real Time system
   ↙ soft Real Time Systems doesn't guarantee meeting guideline.
4. Based on triggering → Event Triggered system
   → Time Triggered system

The classification based on deterministic system behaviour is applicable for 'Real Time' system. The application / task execution behaviour for an embedded system can be either deterministic or non deterministic.

Based on the execution behaviour, Real Time embedded systems are classified into Hard & soft.

Reactive systems can be either event triggered or time triggered.

Classification Based on Generation

Embedded Systems can be classified into:

First Generation:

The early embedded systems were built around 8-bit µprocessors like 8085 & Z80 & 4-bit µC.

Simple in HW ckts with firmware developed in Assembly code. Digital telephone keypads, stepper motor control units etc.

Second Generation ! The embedded Systems are built around 16-bit µprocessors & 8 or 16 bit µC. The Instruction set for the Second generation Processors/ Controllers were much more Complex and Powerful than the first Generation Processors/ Controllers. Ex! DATA acquisition Systems, SCADA. Systems

Third Generation: With advances in Processor technology, embedded System developers Started making use of Powerful 32 bit Processors & 16-bit µC for their design.

A new concept of application & domain Specific Processors/ Controllers like Digital Signal Processors (DSP) & Application Specific Integrated Circuits (ASICs)

The Instruction set of processors became more complex & powerful and the concept of Instruction pipelining.

Ex: Robotics, Media, Industrial process control, Networking etc.

Fourth Generation: The advent of System on chips (SoC), reconfigurable processors & Multicore processors are bringing high Performance, tight Integration & Miniaturisation into the embedded device Market. The SoC technique Implements a total system on a chip by Integrating different functionalities with a Processor core on an Integrated ckt.

Ex: Smart Phone devices, Mobile Internet devices (MIDs)

Classification Based on Complexity & Performance.

(1) Small-Scale Embedded Systems: These are usually built around low performance & low cost 8 or 16 bit up/uc. A small-scale embedded system may or maynot contain an operating system for its functioning.

(2) Medium-Scale Embedded Systems! These are built around performance, low cost 16 or 32-bit up / uc or dSP.

(3) Large-Scale Embedded Systems / Complex-systems.

The high performance 32 or 64 bit RISC processors/ controllers or Reconfigurable System on chip (RSOC) or Multi core processors & programmable logic devices. They may contain multiple processors/ controllers & co-units / Hardware accelerators for offloading the processing requirements from the main processor of the system.

Decoding/encoding of media, cryptographic function Imple-mentation, etc.

complex embedded systems usually contain a high performance (RTOS) for task scheduling, prioritization & Management.

Major Application Areas of EMbedded Systems.

(1) Consumer electronics: Cam corders, Cameras, etc.

(2) Household appliances: T.V, DVD Players, Washing Machine, fridge, Microwave Oven

(3) Home Automation & Security Systems: Air conditioners, sprinklers, Intruder detection alarms, Closed Ckt T.V. Cameras fire alarms etc.

(4) Automotive Industry: Anti-lock breaking systems (ABS), engine control, ignition systems, automatic navigation systems etc.

(5) Telecom: Cellular telephones, telephone switches, Handset applications, etc.

(6) Computer Peripherals: Printers, Scanners, Fax Machines etc.

(7) Computer N/working Systems: N/W routers, switches, hubs, firewalls, etc.

(8) Health Care: Different Kinds of Scanners, EEG, ECG. Machines etc.

(9) Measurement & Instrumentation: Digital Multimeters, digital CROs, logic analyzers PLC systems, etc.

(10) Banking & Retail: Automatic teller Machines (ATM) & Currency counters, Point of sales (POS)

(11) Card Readers: Bar code, Smart Card readers, handheld devices, etc.

Purpose of Embedded Systems.

Each embedded System is designed to serve the purpose of any one or a Combination of the following tasks.

① Data collection / Storage / Representation.
② Data communication. ③ Data (signal) processing ④ Monitoring
⑤ Control ⑥ Application specific user Interface.

① Data Collection / Storage / Representation

The term "data" refers all kinds of Information, viz, text, voice, Image, video, electrical signals.

Data can be either analog (Continuous) or digital (discrete).

Embedded Systems with analog data capturing techniques collect data directly in the form of analog signals

Where as embedded Systems with digital data Collection mechanism converts the analog signal to corresponding digital signal using analog to digital (A/D) converter and then collects the binary equivalent of the analog data.

Embedded Systems designed for pure applications without storage, used in Control & Instrumentation, Collect data

Some Systems incorporate a built - in / plug - in storage. memory for storing the captured data.

Some Systems are representation of the Collected data by visual (graphical / Quantitative) or audible Means using display units [LCD, LED, etc] ; buzzers, alarms, etc.

Ex: Measuring Instruments with storage Memory

(i) Monitoring Instruments with storage Memory used in Medical Applications.

A digital Camera is a typical example of an embedded system with data Collection/ Storage/ representation of data.

Data Communication

Embedded data Communication system are deployed in applications ranging from complex satellite communication systems to simple Home N/Wking Systems. The data collected by an embedded terminal may require transferring of the same to some other system located remotely.

The transmission is achieved either by a wire-line medium or by a wire-less Medium.

A wireless Medium offers Cheaper connectivity solutions and make the Communication link free from the hassle of wire bundles.

Data can either be transmitted by analog or digital.

Modern Industry trends are settling towards digital communication.

The data communication units ~~is~~ like wireless Modules (Bluetooth, zigbee, Wi-Fi, EDGE, GPRS, etc) or Wire-line Modules (RS-232C, USB, TCP/IP, PS2, etc).

Certain embedded systems act as a dedicated transmission unit b/w the sending & Receiving terminals, offering sophisticated functionalities like data Packetising, encrypting & decrypting. Network hubs, routers, switches etc. are typical examples of dedicated data transmission embedded systems.

Data (signal) processing.

The data (voice, Image, video, electrical signals) collected by embedded systems may be used for various kinds of data processing. Embedded Systems with signal processing functionalities are employed in applications demanding signal processing like speech coding, synthesis, audio video codec, transmission applications, etc

Ex:- A digital hearing aid Improves the hearing capacity of hearing Impaired persons.

Monitoring

All embedded products coming under the medical domain are with monitoring functions only. They are used for determining the state of some variables using input sensors. They cannot Impose Control over variables.

Ex:- The electro cardiogram Machine (ECG) for Monitoring the heartbeat of a patient.

The machine is intended to do the monitoring of the heart beat. It cann't Impose Control over the heart beat. The sensors used in ECG are the different electrodes connected to the patient's body.

Ex:- Digital CRO, digital Multimeters, logic analyzers Control & Instrumentation applications. They are used to current, Voltage etc.

Control

Embedded Systems with control functionalities impose control over some variables according to the changes in input variables. A system with control functionality contains both sensors & actuators. Sensors are connected to the input port for capturing the changes in environmental variable or measuring variable.

The actuators connected to the O/P port are controlled according to the changes in I/P variable to put an impact on the controlling variable to bring the controlled variable to the specified range.

Air conditioner system used in our home to control the room temperature to a specified limit is a typical example for embedded system for control purpose.

An air conditioner contains a room temperature sensing element (sensor) which may be a thermistor & a handheld unit for setting up (feeding) the desired temperature.

The handheld unit may be connected to the central embedded unit residing inside the air conditioner through a wireless link or through a wired link.

The air compressor unit acts as the actuator. The compressor is controlled according to the current room temperature and the desired temperature set by the end user.

The input variable is the current room temperature and the controlled variable is also the room temperature.

The controlling variable is cool air flow by the compressor unit. If the controlled variable and I/P variable are not at the same value, the controlling variable tries to equalise them.

Application Specific User Interface

These are embedded systems with application - Specific user Interfaces like buttons, Switches, Keypad, Lights, bells, display units etc.

In Mobile phone the user Interface is provided through the Keypad, graphic LCD Module, System speaker, vibration alert, etc

→ 'Smart' Running Shoes from ADIDAS - The Innovative Bonding of Lifestyle with Embedded Technology.

→ Adidas launched the 'Smart' running shoes in the market in April 2005.

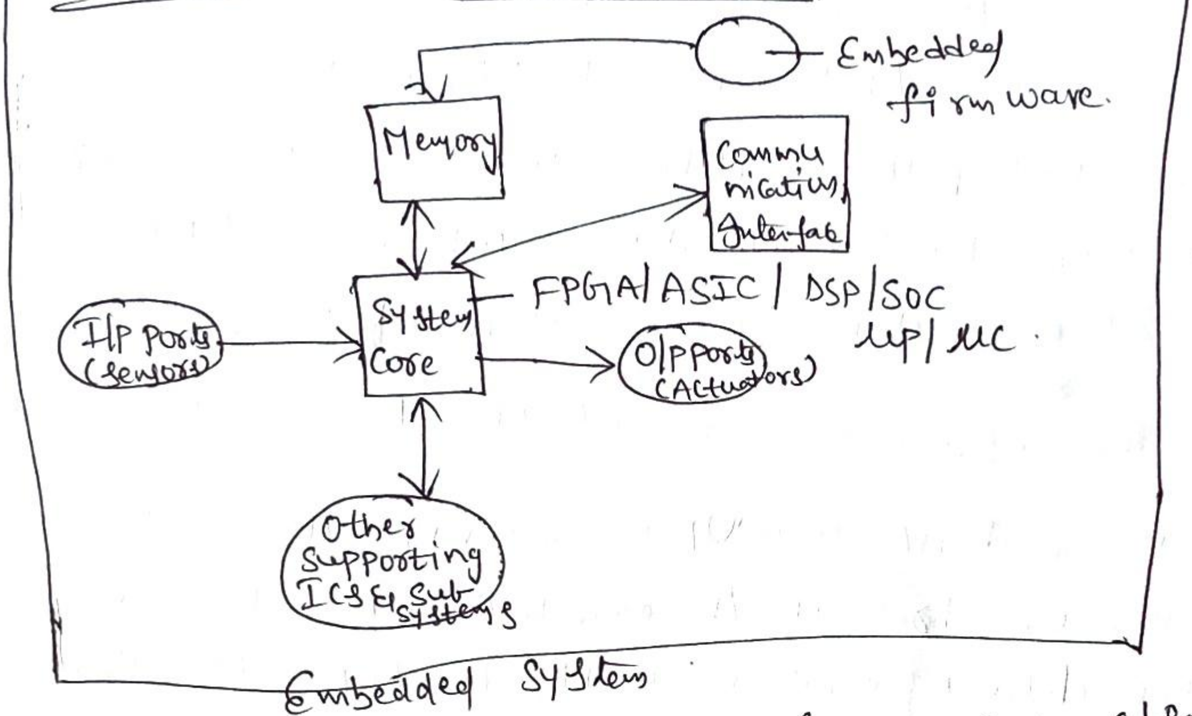
→ The shoe uses a Magnetic sensing system to measure cushioning level, which is adjusted via a digital signal processing unit that controls a motor-driven cable system.

A hall effect Sensor is positioned at the top of the "cushioning element" and the magnet is placed at the bottom of the element.

The cushioning compresses on each impact, the sensor measures the distance from top to bottom of mid-sole (accurate to 0.1mm). About 1000 readings / second are taken and relayed to the Shoe's up. The Microprocessor (MPU) is positioned under the arch of the shoe. It runs an algorithm that compares the compression messages received from the sensor to a preset range of proper cushioning levels. The MPU sends a command to a Micro Motor, housed in the mid-foot. The Micro Motor turns a lead screw to lengthen or shorten a cable secured to the Walls of a plastic-cushioning element

The Typical Embedded System

Elements of an embedded system



Embedded System

A typical embedded system contains a single chip controller which acts as the master brain of the system. The controller can be a µp (e.g. Intel 8085) or a µc (e.g. Atmel AT89C51) or FPGA (Field Programmable Gate Array) device e.g. (Xilinx Spartan) or a Digital Signal Processor (DSP) (e.g. Blackfin Processors from Analog Devices) or Application Specific Integrated Ckt (ASIC), Application Specific standard product (ASSP) (e.g. ADE 7760 Single Phase Energy Metreing IC from) Analog Devices for energy metering applications

Embedded h/w/ s/w Systems are basically designed to regulate a physical variable or to manipulate the state of some devices by sending some control signals to the Actuators or devices connected to the O/p ports of the system, in response to the I/p signals provided by the end users or sensors which are connected to the to the I/p ports.

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 for common user interface I/p devices where as LEDs, LCDs, piezoelectric buzzers, etc are o/p devices.

The sensor information is passed to the processor after signal conditioning & digitisation. Upon receiving the sensor data the processor or brain of the embedded system performs some pre-defined operations with the help of the firmware embedded in the system and sends some actuating signals to the actuator connected to the O/p port of E.S, which in turn acts on the controlling variable to bring the controlled variable to the desired level to make the E.S b/s

The Memory of the system is responsible for holding the control algorithm & other important configuration details. ES, the memory for storing the algorithm or configuration data is of fixed type, which is a kind of ROM

The Memories are OTP, PROM, UVEPROM, EEPROM & FLASH.

Random Access Memory (RAM) is used in most of the systems as the working memory. various types of RAM like SRAM, DRAM & NVRAM.

In a Controller - Based embedded system, the controller may contain internal memory for storing the control algorithms and it may be an EEPROM or FLASH memory

Core of the Embedded System

Embedded Systems are domain & application specific & are built around a central core.

1. General Purpose & Domain Specific processors
   1.1 Microprocessors 1.2 Microcontrollers 1.3 Digital Signal Processors
2. Application Specific Integrated circuits (ASICs)
3. Programmable Logic Devices (PLDs)
4. Commercial off - the - Shelf Components (COTS).

General purpose & Domain Specific Processors

Microprocessors: A μp is a silicon chip representing a CPU, which is capable of performing arithmetic as well as logical operations according to a predefined set of instructions, which is specific to the manufacturer.

The CPU contains the ALU, control unit & working Registers.

A microprocessor is a dependent unit and it requires the combination of other H/W like memory, timer unit, Interrupt controller.

Intel claims the credit for developing the first μp unit Intel 4004, a 4-bit processor which was released in NOV 1971. It featured 1K data memory, 12-bit program counter & 4K program memory, sixteen 4-bit general purpose registers & 46 instructions. The clock speed of 740KHz. In 1972 14 more instructions were added to the 4004 instruction set & the program space is upgraded to 8K.

Also Interrupt Capabilities were added to and it renamed as Intel 4040. It is replaced in April 1972 by Intel 8008 which was similar to Intel 4040, the only difference was that its pc was 14bits wide and the 8008 served as a terminal ~~computer~~ controller.

In April 1974 Intel launched the first 8-bit processor the Intel 8080, with 16-bit address Bus & Program Counter and seven 8-bit registers (A-E, H,L : BC, DE & HL pairs formed the 16-bit register for this processor)

Intel 8080 was the most commonly used processors for Industrial control & embedded applications in the 1975s.

Intel 8080, Motorola also entered the market with their processor, Motorola 6800 with a different architecture & Instruction set compared to 8080.

In 1976 Intel came up with the upgraded version of 8080. Intel 8085, with two newly added Instructions, three Interrupt pins & serial I/O. clock generator & bus controller ckts were built-in & power supply part was modified to a single +5 V supply.

In July 1976 Zilog entered the up Market with its Z80 processor as competitor to Intel → Motorola 6800 - Different Architecture set Compared to 8080

In 20th century 16, 32 & 64-bit processors came into the place of conventional 8-bit processors. Today processors with clock speeds up to 2.4 GHz are available in the Market, ~~Mot~~

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

Harvard Architecture contains seperate buses for program memory & Data Memory, where as processors based on Von-Neumann architecture shares a single system bus for program & data Memory.

Reduced Instruction set computing (RISC) & Complex Instruction Set Computing (CISC) are the two common & Instruction Set Architectures (ISA) available for processor design.

Intel Microprocessors

Microprocessors	Year of Introduction	Data Bus width (Bit)	Address Bus width (Bit)	Memory Size
4004	1971	4	10	1KB
8085A	1976	8	16	64KB
8086	1978	16	20	1MB
8088	1980	8	20	1MB
80186	1981	16	20	1MB
80188	1982	8	20	1MB
80286	1983	16	24	16 MB Real, 4GB Virtual.
80386SX	1984	16	24	16MB
80386EX	1985	16	26	64MB.
80486 DX/SX	1989	32	32	4GB+8KB/16KB Cache.
80386 DX	1986	32	32	4GB
Pentium	1993	64	32	4GB + 16 KB Cache
Pentium over drive (80486)	1994	32	32	4GB + 16 KB Cache
Pentium Pro	1995	64	36	64GB + 16KB L_1 Cache + 256KB L_2 Cache
Pentium II	1997	64	36	64GB + 32KB L_1 Cache + 512KB L_2 cache
Pentium -II Xeon	1998	64	36	64GB + 32KB L_1 Cache + 256KB L_2 cache.
Pentium-III	1999	64	36	64GB + 32KB L_1 Cache + 256KB L_2 cache
Pentium 4	2000	64	36	64GB + 32KB L_1 + Cache + 256KB L_2 cache (1.5GH

Motorola 68000 Family

	MC6800	MC68020	MC68030	MC68040	MC68060
MIPS	2.4	6.5	1	39	>100
MFLOPS	—	0.25	5	3.5	12
Address Range	16 M bytes	4 G Bytes	4G Bytes	4G Bytes	4G Bytes
Data Bus	16-bit	32-bit	32-bit	32-bit	32-bit
clock MHz	8-16	16-33	16-50	25,33,40	50-60
I-Cache	—	256 Bytes	256Bytes	4K Bytes	8K Bytes
D-Cache	—	—	256 Bytes	4K Bytes	8K Bytes
Burst Fill Cache	—	—	16 Byte r/w	16 Byte r/w	16 Byte r/w
on chip MMU	—	—	Yes	Yes	Yes
FPU	68881	68881	68882	On-chip	On-chip.

General Purpose Processor (GPP) Vs. Application – Specific Instruction Set Processor

The General Purpose Processor Contains an ALU & Control Unit (CU)

Application Specific Instruction Set Processors (ASIPs) are Processors with architecture & Instruction set optimized to specific-domains/application requirements like n/w Processing, automative, telecom, media applications, digital signal processing, control applications.

ASIP = Processor + on chip peripherals

ASIPs fill the architectural spectrum b/w GPP Vs ASICs.

→ GPP is Pentium 4/AMD Athlon,

→ MC is Automative AVR, USB AVR from Atmel, Digital Signal Processors

Micro Controllers : A MicroController is a highly integrated chip that contains a CPU, Scratch pad RAM, special & general purpose register arrays, on chips ROM/FLASH memory for program storage, timer & interrupt control units and dedicated I/O ports. Micro controllers can be considered as a super set of μprocessors. μc > CPU+RAM+On chip ROM+ Register arrays, Timer + Interrupt + Dedicated I/O port

Texas Instrument's TMS 1000 is considered as the world's first μC. TI followed Intel's 4004/4040, 4 bit processor design, added some amount of RAM, program storage memory (ROM) & I/O support on a single chip, there by eliminated the requirement of multiple H/w chips for self-functioning. 1974.

In 1977 Intel entered the μC market with a family of controllers coming under one umbrella named MCS-48. The processors came this family were 8038HL, 8039HL, 8040AHL, 8048H, 8049H & 8050AH.

Intel 8048 is recognised as Intel's first μC and it was the most prominent member in the MCS-48™. The inspiration behind 8048 was Fairchild's F8 μp & Intel's goal of developing a low cost & small size processor.

The 8051 processor cores are used in more than 100 devices by more than 20 independent manufacturers like Maxim, Philips, Atmel etc. under the license from Intel.

→ low cost, wide availability, memory efficient instruction set, Boolean processing capability.

The PIC family µc from Microchip Technologies. It is a High performance RISC µc complementing the CISC. Infineon, Freescale, Philips, Atmel, Maxim, Microchip etc The 16-bit applications by developing the Philips XA (extended Architecture) µc series.

Microprocessor Vs Microcontroller

Microprocessor	Microcontroller
① 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	① A µc is a highly integrated chip that contains a CPU, Scratchpad RAM, Special & general purpose register arrays, On chip ROM/ FLASH memory for program storage) timer & interrupt control units & dedicated I/O ports.
② It is a dependent unit. It requires the combination of other chips like timers program & data memory chips, interrupt controllers etc. for functioning.	② It is a self-contained unit and it doesn't require external interrupt controller, timer UART, etc. for its functioning
③ Most of the time general purpose in design & operation	③ Mostly application-oriented or domain-specific.
④ Doesn't contain a built in I/O port. The I/O port functionality needs to be implemented with the help of external PPI chips like 8255	④ 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.
⑤ Targeted for high end market where performance is important.	⑤ Targeted for embedded market where performance is not so critical.

Microprocessor	Microcontroller
Limited Power Saving options Compared to MC	Includes lot of power saving features

→ <u>Digital Signal processors</u>: DSP s are powerful special purpose 8/16/32 bit µprocessors designed specifically to meet the Computational demands & power constraints of today's embedded audio, video & communications applications. Digital Signal Processors are 2 to 3 times faster than the general Purpose µprocessors in Signal processing applications.

<u>Program Memory</u>: Memory for storing the program required by DSP to process the data.

<u>Data Memory</u>: working Memory for storing temporary variables & data / signal to be processed.

<u>Computational Engine</u>: This Incorporates many Specialised arithmetic units & each of them operates simultaneously to Increase the execution speed. It also Incorporates multiple H/w Shifters for Shifting operands & thereby saves execution time.

- <u>I/o Unit</u>: Acts as an Interface b/w the outside World & DSP. It is responsible for Capturing Signals to be processed & delivering the processed signals.

Ex: Audio video signal processing, telecommunication & Multimedia applications are typical examples where DSP is employed.

Digital Signal Processing employs a large amount of real-time calculations. Sum of products (SOP) calculation, convolution, FFT, DFT, etc. are operations performed by DSP.

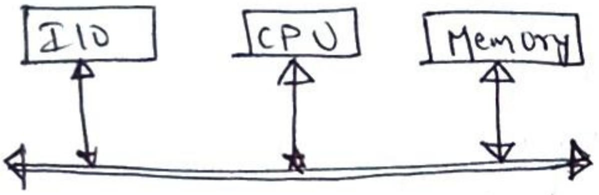
Blackfin processors from Analog Devices is an example of DSP which delivers breakthrough signal-processing performance & power efficiency a full 32-bit RISC MCU programming model. These processors present high-performance, homogeneous S/W targets, which allows flexible resource allocation b/w hard real-time signal processing tasks & non-real time control tasks.

RISC	CISC.
① Lesser Number of Instructions	① Greater number of Instructions
② Instruction pipelining and Increased execution speed.	② Generally no Instruction pipelining feature.
③ Orthogonal Instruction set (Allows each Instruction to operate on any register and use any addressing Mode)	③ Non-orthogonal Instruction set (All Instructions are not allowed to operate on any register and use any addressing Mode. It is Instruction specific)
④ operations are performed on registers only, the only memory operations are load & store	④ operations are performed on registers or Memory depending on the Instruction.
⑤ A large number of registers are available	⑤ Limited Number of general purpose registers

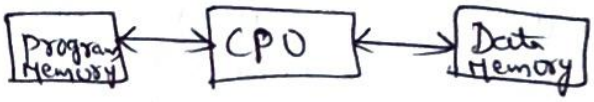
	RISC	CISC
6	A programmer needs to write more code to execute a task since the Instructions are simpler ones	6 Instructions are like macros in C language. A programmer can achieve the desired functionality with a single Instruction which in turn provides the effect of using more simpler single instructions in RISC
7	Single, fixed length Instructions	7 Variable length Instructions
8	Less silicon usage & pin count	8 More silicon usage since more additional decoder logic is required to implement the complex instructions decoding.
9	With Harvard Architecture	Can be Harvard or Von-Neumann Architecture
10	Ex: Atmel AVR MC	Ex: ~~Atmel AVR MC~~ AT89C51.

→ Harvard vs Von-Neumann Processor | Controller Architecture

Harvard Architecture	Von-Neumann Architecture
1 Separate buses for Instruction & data fetching	1 Single shared bus for Instruction & data fetching.
2 Easier to pipeline, so high performance can be achieved	2 Low performance compared to Harvard architecture.
3 comparatively High cost	3 cheaper
4 No Memory alignment problems	4 Allows self modifying codes
5 Since data Memory & program Memory are stored physically in different locations, no chances for accidental corruption of program memory	5 Since data Memory & program Memory are stored physically in the same chip, chances for accidental corruption of Program Memory



Harvard Architecture



Von-Neumann Architecture

* Big-Endian Vs Little-Endian Processors/Controllers

Endianness Specifies the order in which the data is stored in the memory by processor operations in a multibyte 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

① Higher order of data byte at the higher memory & lower order of data byte at location just below the higher memory

② Lower order of data byte at the higher Memory & Higher order of data byte at location just below the Higher Memory.

* Little-Endian – The lower-order byte of the data is stored in memory at the lowest address, The Higher-order byte at the Highest address (The little end Comes first).

Base Add + 0	Byte 0	Byte 0	0x20000 (Base Add)
Base Add + 1	Byte 1	Byte 1	0x20001 (Base Add + 1)
Base Add + 2	Byte 2	Byte 2	0x20002 (Base Add + 2)
Base Add + 3	Byte 3	Byte 3	0x20003 (Base Add + 3)

→ Little-Endian operation.

Big – Endian.

The higher - order byte of the data is stored in Memory at the lowest address, The lower - order byte at the highest address (The big end comes first).

Base Address +0	Byte 3	0X 20000	(Base Address)
Base Address +1	Byte 2	0X 20001	(Base Address +1)
Base Address +2	Byte 1	0X 20002	(Base Address +2)
Base Address +3	Byte 0	0X 20003	(Base Address +3)

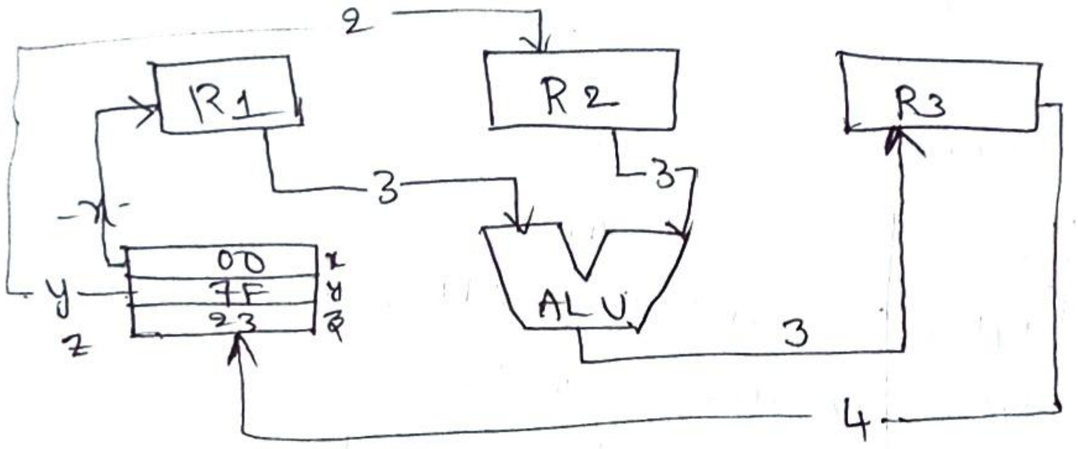
Big-endian operation

Load Store Operation & Instruction Pipelining.

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.

The first instruction load R1, X loads the register R1 with the content of M.L X, the second instruction load R2, Y loads the register R2 with the content of M.L Y. The instruction add R3, R1, R2 adds the ~~the~~ content of registers R1 & R2 & stores the result in reg R3. The next instruction store R3, Z stores the content of reg R3 in M.L Z.

The fetch part fetches the instruction from program memory or code memory & The decode part decodes the instruction to generate the necessary control signals. The execute stage reads & the operands, Perform ALU operations and stores the result.



Load R1, x → ①

Load R2, y → ②

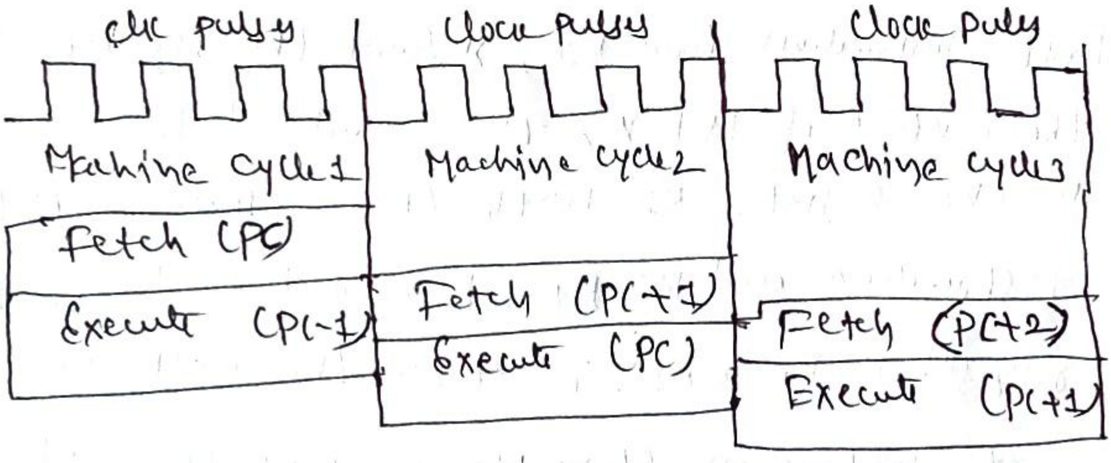
Add R3, R1, R2 → ③

Store R3, z → ④

If the current instruction in progress is a program control flow transfer instruction like Jump or Call instruction.

· whenever the current instruction is executing the PC will be loaded with the address of the next instruction

Incase of Jump or branch instruction, the new location is known only after completion of the jump or branch instruction.



PC: program counter

→ The Single-Stage pipelining concept.

Application Specific Integrated Circuits (ASICs)

ASICs is a microchip designed to perform a specific or unique application. ASICs are proprietary products. As a single chip. ASIC consumes a very small area in the total system and thereby helps in the design of smaller systems with high capabilities / functionalities

ASIC based systems are profitable only for large volume commercial productions. Fabrication of ASICs requires a non-refundable initial investment for the process technology & configuration expenses. This investment is known as NRE (Non-Recurring Engineering cost). i.e it's a one-time investment.

The NRE is borne by a third party & ASIC.

The ASIC is referred to as Application Specific Standard Product (ASSP)

Ex:- The ADE7760 Energy Meter ASIC developed by Analog Devices for Energy metering applications is a typical example for ASSP.

Programmable Logic Devices (PLD's)

Logic Devices are two Categories ① Fixed & ② Programmable. As the name indicates, the ckts in a fixed logic device are permanent, they perform one fun or set of functions - once manufactured, they cannot be changed.

The PLD used for N/w router, a DSL Modem, DVD player, Automotive Navigation systems. There are no NRE costs

& The final design is completed much faster than that of a custom, fixed Logic device.

The PLDs are FPGA & CPLDs.

CPLD & FPGAs.

The largest FPGA now shipping, part of the xilinx Virtex line of devices; provides eight million "System gates" (the relative density of logic)

FPGAs are used. for data processing & storage, to instrumen-n, telecommunications, & DSP.

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

CPLDs, by contrast, offer much smaller amounts of logic- up to about 10,000 Gates. CPLDs such as the Xilinx Cool Runner^{TM} series also require extremely low amount of Power & are very inexpensive, making them ideal for cost-sensitive, Battery-operated, Portable applications such as Mobile Phones & digital Handhelds assistants.

Advantages of CPLDs

(i) PLds don't require customers to pay for large NRE costs and purchase expensive mask sets - PLD suppliers incur those costs when they design their programmable devices.

(2) PLD's offer customers much more flexibility during the design cycle because design iterations are simply a matter of changing the programming file.

(3) PLDs don't require long lead times for prototypes or production parts - the PLDs are already on a distributor's shelf & ready for shipment.

→ PLD suppliers such as Xilinx are 'fabless' companies instead of owning chip manufacturing foundries. Xilinx outsource that job to partners like Thoshiba & UMC whose chief occupation is making chips.

Commercial off-the-shelf components (COTS)

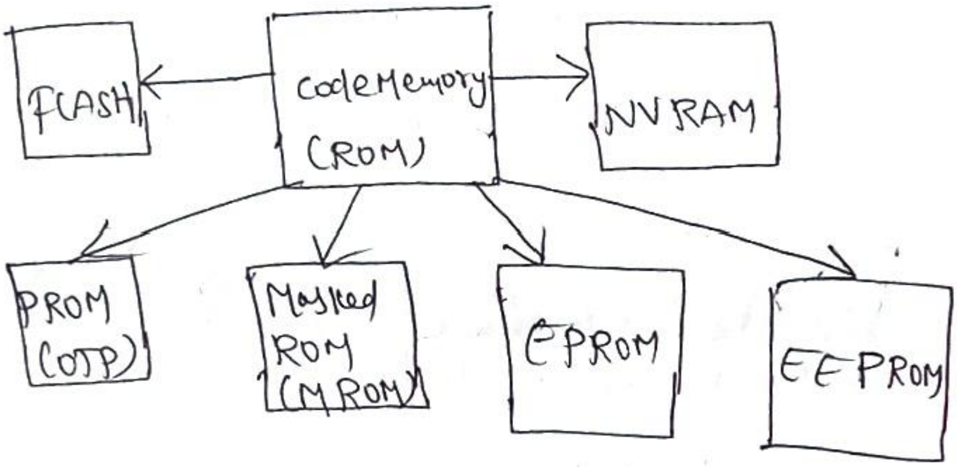
The COTS component itself may be developed around a general purpose or domain specific processor or Application specific integration ckt or programmable logic device.

COTS products are designed in such a way to provide easy integration & interoperability with existing system components. The examples are COTS H/W unit are remote controlled toy car control units including the RF circuitry part, High performance, High freq microwave electronics (2-200GHz), High B.W A/D converter, electro-optic IR imaging arrays, UV/IR detectors, etc.

Memory

The Processors/ Controllers contains built in memory & this memory is referred as On-chip Memory. Others don't contain any memory inside the chip and requires external memory to be connected with the Controller/ processor to store the control algorithms. It is - Called Off-chip Memory

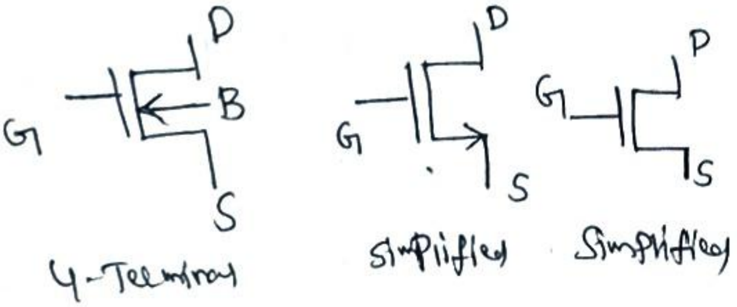
Program Storage Memory (ROM)



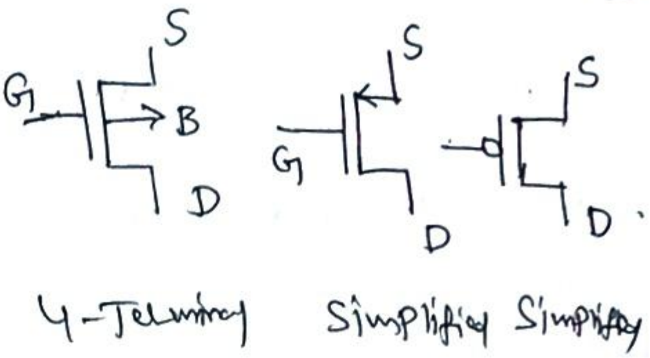
① Masked ROM (MROM)

① Creation of an enhancement or depletion mode trnor through channel implant.

② By creating the memory cell either using a standard transistor or a high treshold trnor. In the high threshold Mode, the supply Voltage required to turned on the transistor is above the normal ROM IC Operating voltage. This ensures that the trnor is always off and the memory cell stores always logic 0

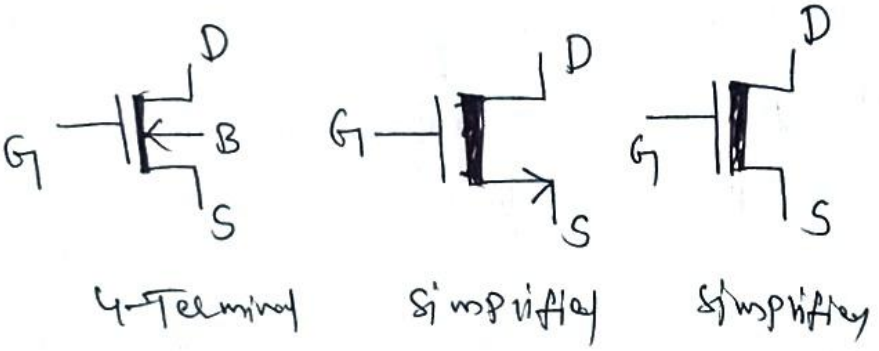


n-channel MOSFET



P-channel MOSFET.

B - Body or Sub-strate.



n-channel depletion-type MOSFETs.

Masked ROM is a good Candidate for storing the embedded firmware for low cost embedded devices.

Programmable Read only Memory (PROM) / (OTP)

Unlike Masked ROM Memory, One Time Programmable Memory (OTP) or PROM is not pre-programmed by the Manufacturer. The end user is responsible for programming these devices. This Memory has nichrome or polysilicon wirs arranged in a matrix.

Fusy which are not blown/ burned represents a logic '1' whereas fuses which are ~~no~~ blown/burned represents a logic '0'. The default State is logic "1".

Electrically Erasable Programmable Read only MemoryC (EEPROM).

The EEPROM Memory can be altered by using electrical signals at the register/ Byte level. They can be erased and reprogrammed in-circuit. These Chips Include a chip erase Mode & In this mode they can be erased in a few msec.

FLASH Memory: This is organised as sectors (blocks) or pages. FLASH memory stores information in an array of floating gate MOSFET transistors. The typical erasable capacity of FLASH is 1000 cycles.

NVRAM. Non-Volatile RAM is a random access memory with battery backup. It contains static RAM based memory & a minute battery for providing supply to the memory in the absence of external power supply.

The life span of NVRAM is 10 years. DS 1644 from Maxim/Dallas is an example of 32KB NVRAM.

Read-write Memory / Random Access Memory (RAM)

RAM is the data Memory or working Memory of the controller Processor can read from it and write to it. RAM is Volatile when the power is turned off, all the contents are destroyed. RAM is a direct acces Memory, meaning we can acces the desired M.L directly without the need for traversing through the entire M.L to reach the desired M.L.

RAM generally falls into three categories; Static RAM, (SRAM), dynamic RAM (DRAM), non-Volatile RAM (NVRAM).

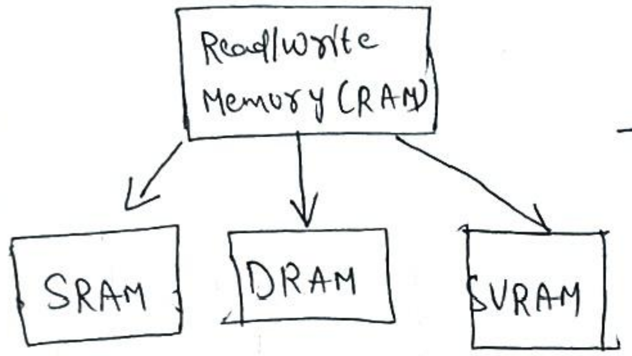
Static RAM (SRAM):

Static RAM Stores data in the form of Voltage They are made up of f/f s Static RAM is the fastest form of RAM available SRAM Cell (bit) is realised using 6 Transistors (or 6 MOSFET s). Four of the Tr/ors are used for building the Latch (f/f) part of the Memory cell & two for controlling the access:

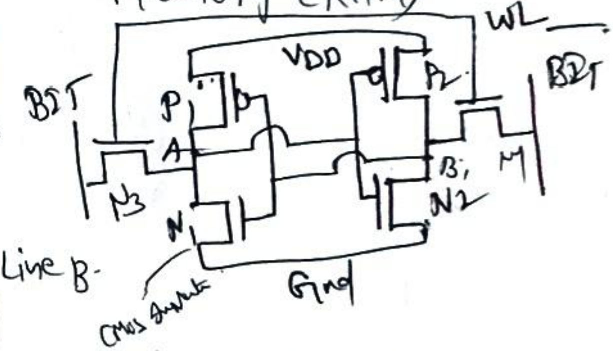
(1) WL is high (V_{DD})
N3,&4

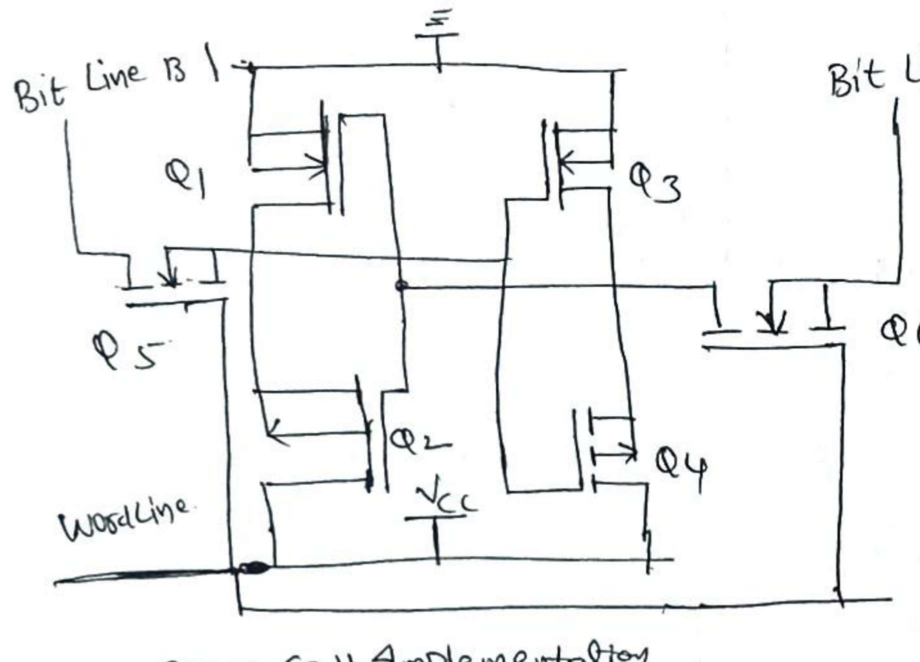
2) BIT, $\overline{BIT}$ = V_{DD}

3) A-B→
0 - 1.
(W/L)N1 > (W/L)N3



→ classification of working Memory (RAM)

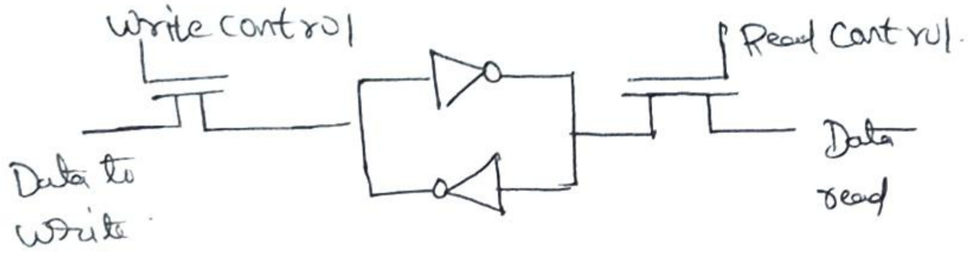




SRAM Cell Implementation

The SRAM Implementation diagram, It is Clear that access to the Memory Cell is controlled by the line Word line, which Controls the access tr/ors (MOSFETs) Q5 & Q6

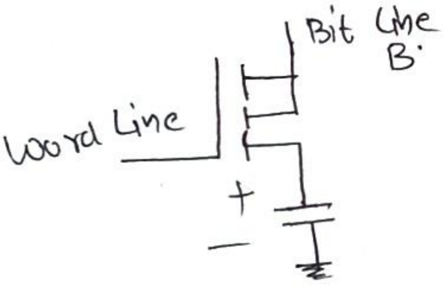
The access transistors controls the Connection to bit lines B & B\. For writing 1, make B=1, B\=0, For writing 0, make B=0 & B1 = 1).



Visualization of SRAM cell

The Four Transistors in the middle form the cross-coupled Inverters. The limitation of SRAM are low capacity & High cost.

Dynamic RAM: It stores data in the form of charge. They are made up of MOS Transistor gates. The advantages of DRAM are its high density & low cost compared to SRAM. The Disadvantage of Information is stored as charge ~~as~~ it gets leaked off with time & to prevent this they need to be refreshed periodically.



DRAM cell Implementation

	SRAM cell	DRAM cell
1	Made up of 6 CMOS Transistors (CMOSFET)	Made up of a MOSFET & a Capacitor
2	Doesn't require refreshing	Requires refreshing
3	Low Capacity (Less dense)	High Capacity (Highly dense).
4	More expensive	Less expensive.
5	Fast in operation	Slow in operation due to refresh requirements

(The time is Gomes write operation is faster than read operation.

NVRAM (Non-Volatile RAM is a random access memory with battery backup. It contains static RAM based memory & a minute battery for providing supply to the memory in the absence of external power supply. The life span of NVRAM is expected to be around 10 years. DS1744 from Maxim/Dallas is an example for 32KB NVRAM.

Memory According to the Type of Interface.

→ A parallel Interface [The parallel data lines (D0-D7) for an 8-bit processor/controller will be connected to D0-D7 of the memory.

→ The serial Interface like I2C (it is a 2 line serial Interface) or It may be an SPI (Serial Peripheral Interface, 2+n line Interface where n stands for the ~~total~~ number of SPI bus devices in the system).

Atmel corporations AT24C512 for serial memory with capacity 512 kbytes & 2-wire Interface.

Memory Shadowing

Generally the execution of a program from a ROM is very slow (120 to 200ns) compared to the execution from a RAM (40 to 70ns). From the timing parameters it is obvious that RAM access is about 3 times as fast as ROM access. Shadowing of memory is a technique adopted to solve the execution speed problem in processor-Based Systems.

In computer systems & video systems there will be a configuration holding ROM called Basic Input ~~or~~ o/p configuration ROM or simply BIOS

2^{16} = 65536 bytes = 64kb.

→ The 2nd parameter that needs to be considered in selecting a memory is the word size of the memory. The word size refers to the number of Memory bits that can be read/write together at a time. 4,8,12,16,24,32 etc. are the word size refers to the number of memory bits that can be read/write 4,8,12,16 24,32,etc. Ensure that the word size supported by the memory chip mathes with the data bus width of the processor/ Controller.

→ FLASH Memory Comes in two major Variants.

① NAND & ② NOR FLASH

NAND FLASH is a high-density low cost non-Volatile Storage Memory. NOR FLASH is less dense & slightly expensive. But it supports the Execute in place (XIP) technique for program execution.

The EEPROM data storage memory is available as either serial interface or parallel interface chips.

Sensors & Actuators

① Sensors

A sensor is a transducer device that converts energy from one form to another for any measurement or control purpose.

② Actuators

Actuator is a form of transducer device (Mechanical or electrical) which convert signals to corresponding Physical Action (motion). Actuator act as an o/p device.

③ The I/o Subsystem

i) Light Emitting Diode (LED): This is an o/p device for visual Indication in any embedded System

→ LED Can beused as an Indicator for the status of various signals or situations. The examples are Indicating the presence of power conditions like Device ON, Battery low, charging of battery for a battery operated handheld embedded device.

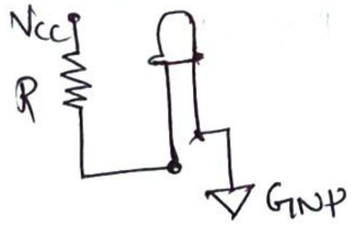
→ LED is a P-N Junction diode and it contains Anode & Cathode. The Anode is connected to an +ve terminal of the supply voltage Cathode is connected to an -ve terminal of the supply Voltage.

→ The current flowing through the LED must be limited to a value below the maximum current that it can conduct. A resistor is used in series b/w the power supply and the LED to limit the current through the LED

→ LED Can be Interfaced to the port pin of a processor/contr in two ways. ① The Anode is directly connected to the port pin and the port pin drives the LED. The port pin 'sources' current to the LED when the port pin is at logic HIGH (logic '1')

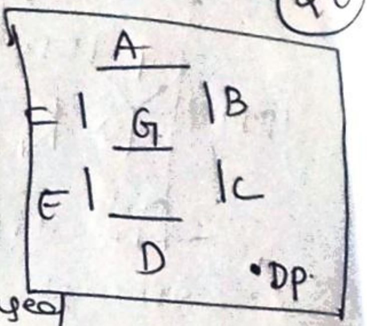
→ ② The cathode of the LED is connected to the port pin of the processor/controller and the anode to the supply voltage through a current limiting rhor. The LED is ON when the port pin is at logic low (Logic '0').

→ In second approach, the current is directly sourced by the power supply & port pin acts as the sink for current. Here the brightness for the LED.



LED Interfacing

7-Segment LED Display

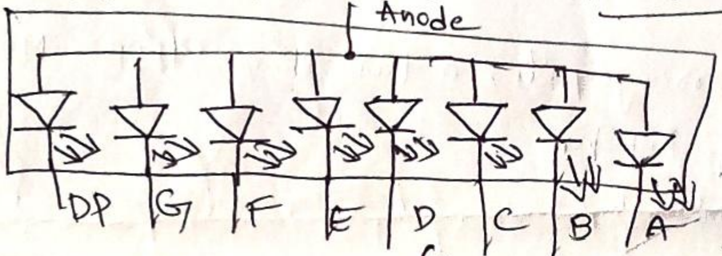


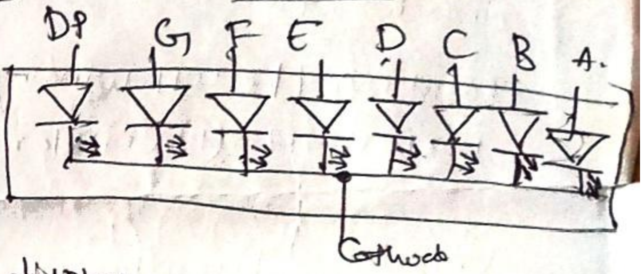
This is an O/p device for displaying alphanumeric Characters. ~~O~~It contains 8 LED segments arranged in a special form. Out of the 8 LED segments, 7 are used for displaying alpha numeric characters & 1 is used for DP (Decimal point).

Ex:- ~~for~~ The character 'd', the segments B, C, D, E & G are lit.

The Number '4', the segments F, G, B & C are lit.

The 7-Segment LED Displays are available in two different configurations, namely; Common Anode & Common Cathode.



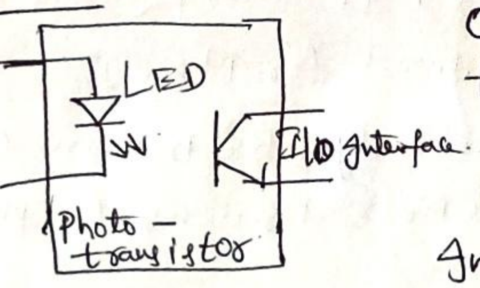


Common Anode LED display.

Common Cathode LED Display

→ Based on the configuration of the 7-Segment LED unit, the LED segment's anode or Cathode is connected to the Port of the processor/controller in the order 'A' segment to the least significant port pin & DP segment to the most significant Port pin

Optocoupler:

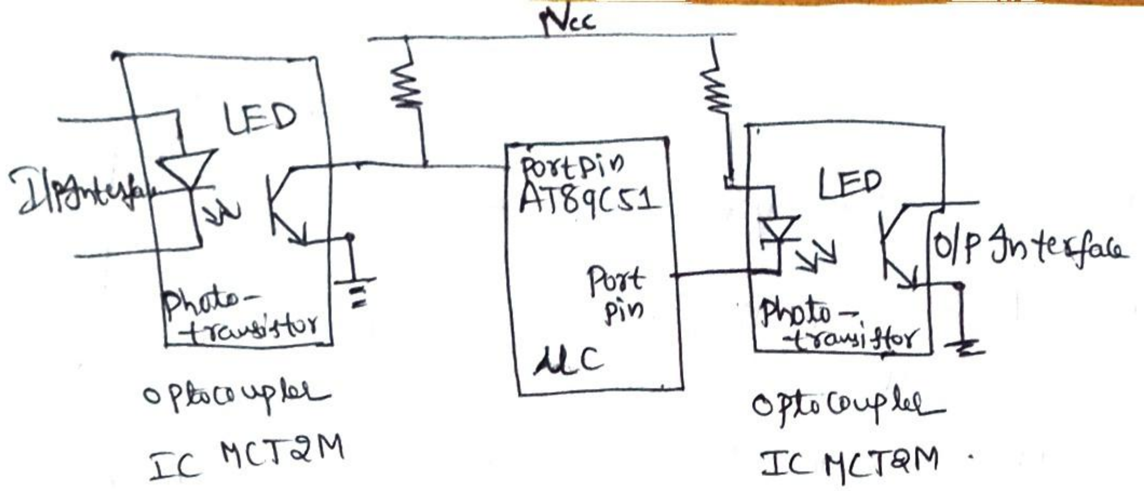


→ An optocoupler device

Optocoupler combines an LED & photo-transistor in a single housing (packages)

An optocoupler is used for suppressing Interference in data communication, ckt Isolation, High voltage separation, simultaneous separation & signal intensification.

Optocouplers can be used in either i/p ckts or in o/p ckts.

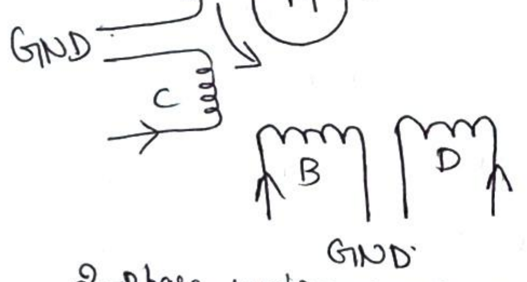


Stepper Motor

This also 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.

Based on the coil winding arrangements, a two-phase Stepper Motor is classified. 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 & in the opposite direction flows through the other coil.



2-phase unipolar stepper Motor

→ Coils A & C carry current in opposite directions for phase I.

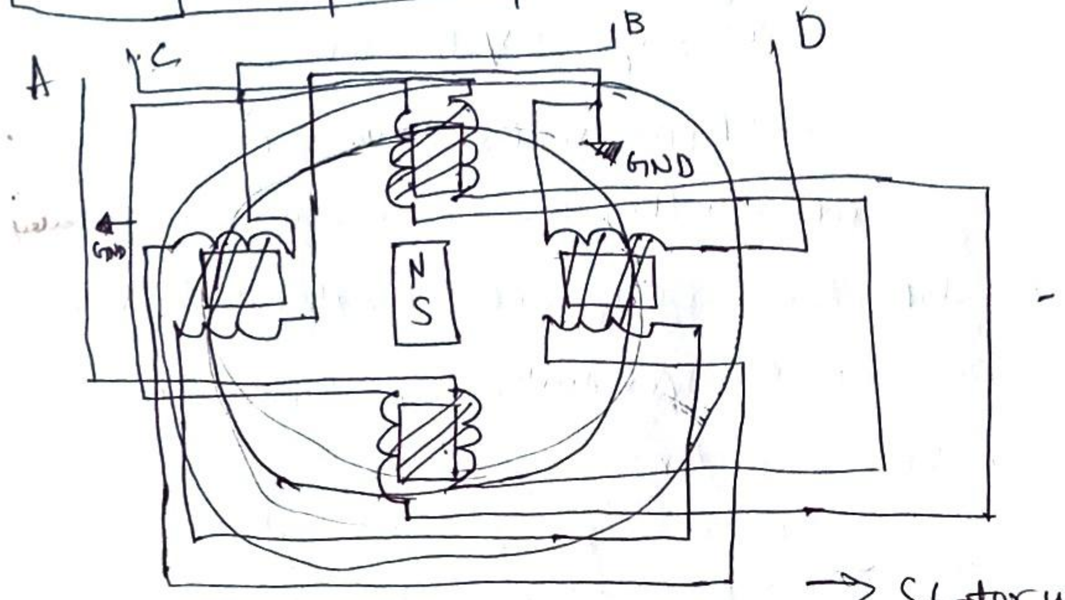
→ Similarly, B & D carry current in opposite directions for phase II.

Bipolar: A bipolar stepper motor contains single winding per phase. For reversing the motor rotation the current flow through the windings is reversed dynamically.

Full Step

Step	Coil A	Coil B	Coil C	Coil D
1	H	H	L	L
2	L	H	H	L
3	L	L	H	H
4	H	L	L	H

=> Both the phases are energised simultaneously



→ Stator winding details for a 2 Phase unipolar Stepper Motor

Wave Step : The only one phase is energised at a time and each coils of the phase is energised alternatively.

Step	Coil A	Coil B	Coil C	Coil D
1	H	L	L	L
2	L	H	L	L
3	L	L	H	L
4	L	L	L	H

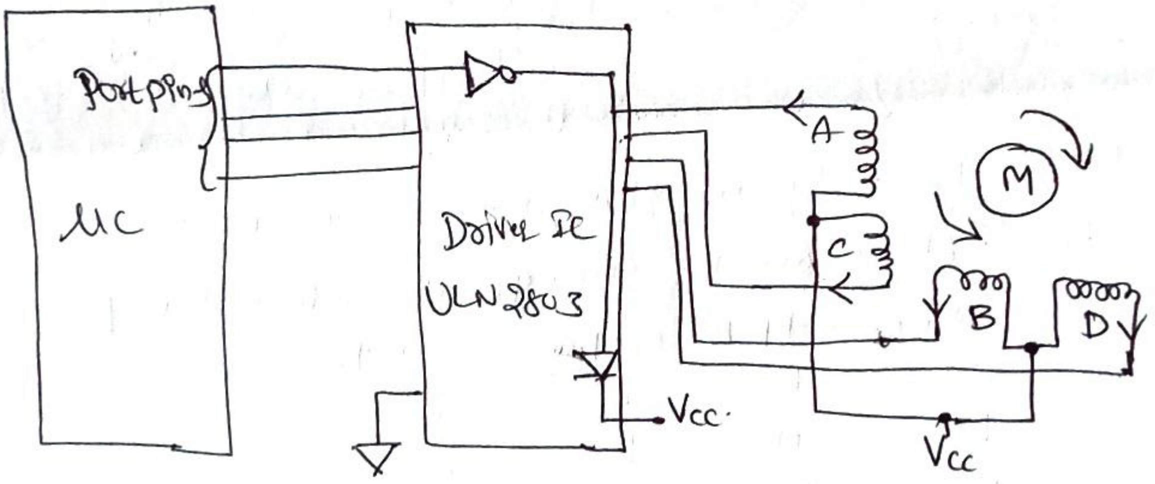
~~Half~~ Half step : It uses the combination of wave & full step. It has the highest torque & stability.

Step	Coil A	Coil B	Coil C	Coil D.
1	H	L	L	L
2	H	H	L	L
3	L	H	L	L
4	L	H	H	L
5	L	L	H	L
6	L	L	H	H
7	L	L	L	H
8	H	L	L	H

The rotation of the stepper motor can be reversed by reversing the order in which the coil is energised.

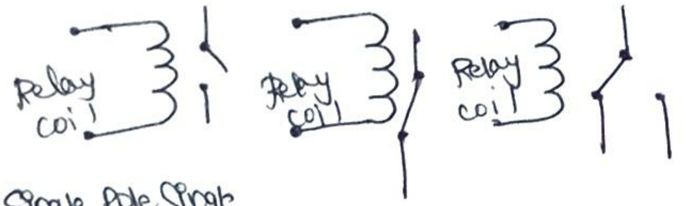
→ The current requirement for stepper motor is little high & Hence the port pins of a μc/ Processor may not be able to drive them directly. The supply voltage required to operate stepper motor varies normally in the range 5V to 24V.

Commercial off-the-shelf stepper motor driver ICs are available in the market and they can be directly interfacing to the μc port. ULN 2803 is an octal peripheral driver array available from ON semiconductors & ST ~~se~~ Microelectronics for driving a 5V stepper Motor.



Interfacing of stepper motor through drive ckt.

Relay: This 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 unit is made up of Metal core & Metal Armature.



Single pole Single throw normally open

Single pole Single throw normally closed

Single pole double throw

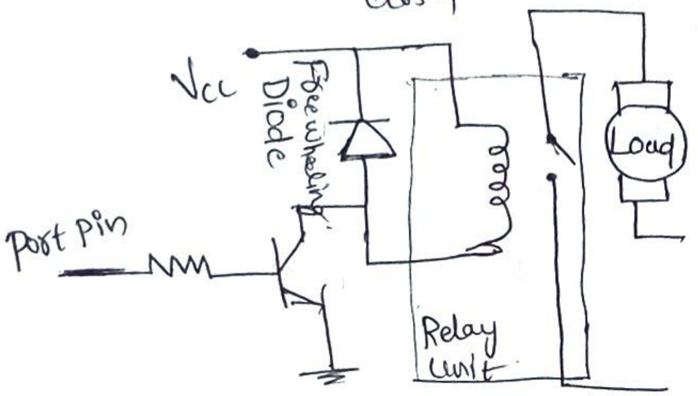


Fig: Transistor Based Relay driving ckt.

→ The Single pole Single Throw configuration has only one path for information flow.

→ The Single pole Double Throw Relay, there are two paths for information flow and they are selected by energising or de-energising the relay.

→ A free-wheeling diode is used for free-wheeling the voltage produced in the opposite direction when the relay coil is de-energised. The free wheeling diode is essential for protecting the relay & the transistor.

* Piezo Buzzer: This is a piezoelectric device for generating audio indications in embedded application. A piezoelectric buzzer contains a piezoelectric diaphragm which produces audible sound in response to the voltage applied to it. The piezo electric buzzers are two types.

① Self-Driving ② External Driving.

Self-Driving: All the necessary components to generate sound at a predefined tone. It will generate a tone on applying the voltage.

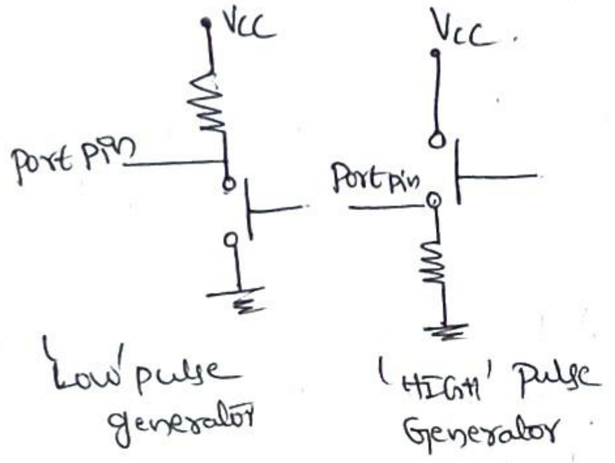
External Driving: This supports the generation of different tones. A piezo buzzer can be directly interfaced to the port pin of the processor/ Control.

* Push Button Switch: This switch comes in two configurations, namely, 'push to make' & 'push to Break'.

The push to make configuration, the switch is normally in the open

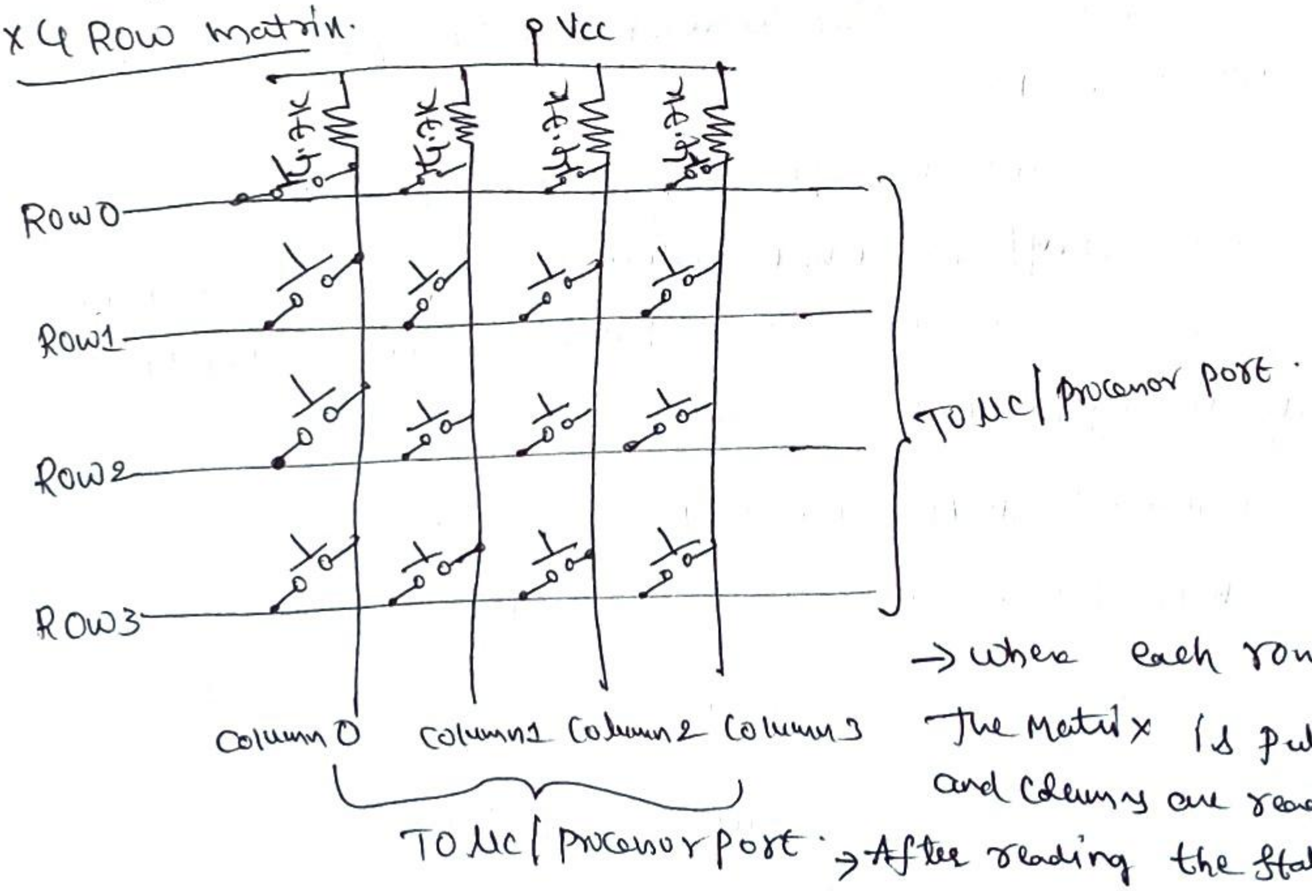
State & it makes a ckt contact when it is pushed or pressed.

The "push to Break" configuration the switch is normally in the closed state & it breaks the ckt contact when it is pushed or pressed.



In embedded application push button is generally used as reset & start switch & pulse generator. The push button is normally connected to the port pin of the host processor/controller

Keyboard: Matrix keyboard is an optimum solution for handing large key requirements. The 16 keys are arranged in a column 4 x 4 Row matrix.



→ when each row of the Matrix is pulled low and columns are read.

→ After reading the status of each columns corresponding to a row, the row is pulled high and next row is pulled low

Matrix keyboard Interfacing

The status of colouns are read

→ In Mechanical the de-bounce keys are Issues.

Programmable Peripheral Interface (PPI)

8255A is a Popular PPI device for 8-bit Processors / Controllers 8255 A Supports 24 I/o Pins and these I/o Pins can be grouped as three 8-bit parallel Ports (Port A & Port B, Port C) or two 8-bit Parallel ports (Port A & port B) with Port C.

1. As 8 Individual I/o Pins.
2. Two 4-bit Ports namely Port C_{UPPER} (C_U) & Port C_{LOWER} (C_L).

The control register holds the Configuration for Port A, Port B, Port C. The bit details of CR is given below.

D7	D6	D5	D4	D3	D2	D1	D0

Bit — Description

D0 – port C lower (C_L) I/o mode selector

D0=1; Sets C_L as I/P Port.

D0=0; Sets C_L as O/P port.

D1 — port B I/o mode selector.

D1 = 1; Sets port B as I/P port.

D1 = 0; Sets Port B as O/P port.

D2 - Mode selector for Port C lower & Port B.

D2 = 0; Mode 0 – port B functions as 8-bit I/o port. Port C as lower functions as 4-bit Port.

D2 = 1; Mode 1 – HandShake Mode. port B uses 3 bits of port C as Hand Shake signals.

D3 – Port C Upper (C_U) I/o mode selector

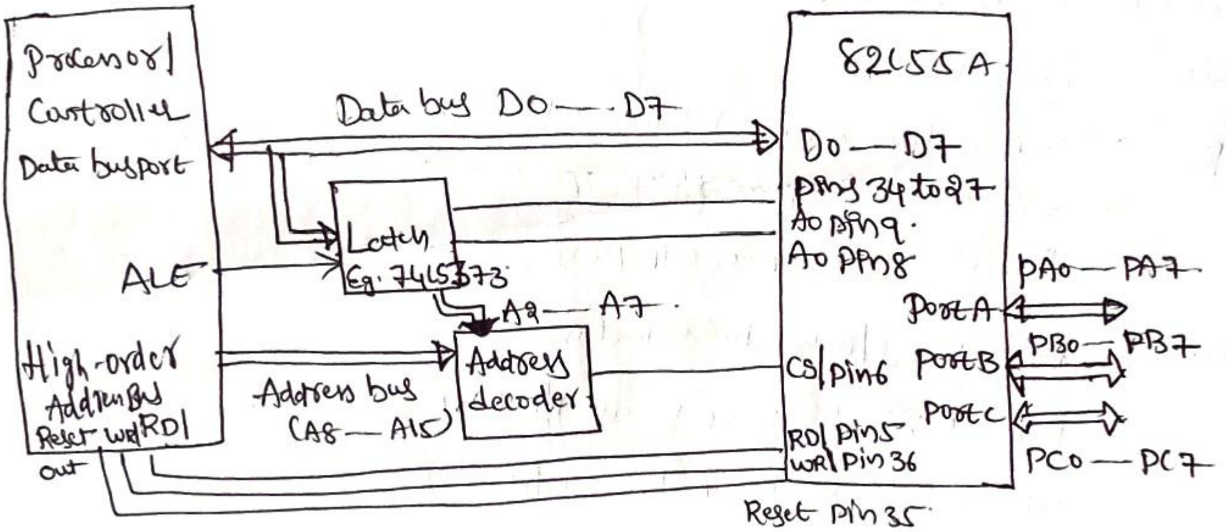
D3 = 1; Sets C_U as I/p Port; D3 = 0; Sets C_U as O/P Port.

D4 – Port A I/O Mode selector

D4 = 1; sets port A as i/P port, D4 = 0; Sets port A as O/P Port.

D5, D6	Mode Selector for port C upper & port A. D6 D5 = 00 ; Mode 0 — Simple I/O Mode D6 D5 = 01 ; Mode 1 - Handshake Mode. Port A uses 3 bits of Port C as handshake signals D6D5 = 1x ; Mode 2. x can be 0 or 1 — port A functions as bi-directional port.

D7	Control / Data Mode Selector for port C.
	D7 = 1 ; I/O Mode.
	D7 = 0 ; Bit Set/reset (BSR) Mode.



Interfacing of 8255 with an 8-bit uc.

Communication Interface.

The communication Interface can be viewed in two different. Perspectives, namely, Device/board level Communication Interface (On board Communication Interface) & product level Communication Interface (External Communication Interface).

The product level Communication Interface ('External Communication Interface) is responsible for data transfer b/w the embedded System.

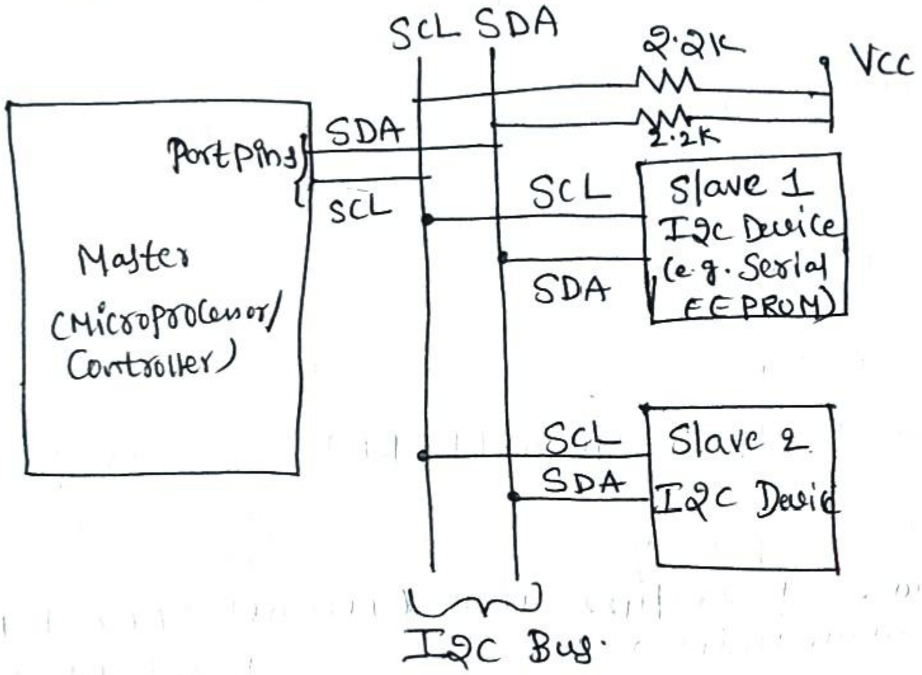
The external Communication Interface can be either a wired media or a wireless media and it can be a serial a parallel Interface - Infrared (IR), Bluetooth (BT), Wireless LAN (Wi-Fi), Radio frequency waves (RF), & GPRS.

Wired Media — RS-232/422, USB, Ethernet IEEE 1394 port, Parallel port.

On-board Communication Interfaces

→ Inter Integrated Ckt (I2C) Bus. The I2c bus comprise of two bus lines, Serial clock (SCL) & Serial Data (SDA) SCL line is responsible for generating synchronisation clk pulses & SDA is responsible for transmitting ~~to~~ the Serial data across ~~bo~~ devices The Master device is responsible for controlling the communication by Initiating / terminating data transfer, Sending data & generating necessary synchronisation clk. Pulses.

The Master device is responsible for controlling the Communication by Initiating / terminating data transfer, sending data & generating necessary Synchronisation clk. pulses. 'Slave' devices wait for the commands from the master & respond upon receiving the commands. 'Master' & 'Slave' devices can act as either transmitter or Receiver. The Synchronisation clock signal is generated by the 'Master device only.



① The Master device pulls the clock line (SCL) of the bus to 'HIGH'

② The Master device pulls the data line (SDA) 'LOW', when the SCL line is at Logic 'HIGH' (i.e. Start condition for data transfer).

③ The master device sends the address (7bit or 10bit wide) of the 'slave' device to which it wants to communicate over the SDA line. Clk pulses are generated at the SCL line for synchronising the bit reception by the slave device. The MSB of the data is always transmitted first. The data on the bus is valid during the 'HIGH' period of the clk signal.

(4) The Master device sends the Read or write bit.
(Bit value = 1 Read operation, Bit value = 0 write operation)

(5) The Master device waits for the ack bit from the slave device whose address is sent on the bus along with the R/W operation command. Slave devices connected to the bus compares the address received with the address assigned to them.

(6) The slave device with the address requested by the master device responds by sending an ack bit. (Bit value = 1) over the SDA line.

(7) The Master device waits for the ack bit from the device upon byte transfer complete for a write operation and sends an ack bit to the slave device for a read operation.

(8) The Master device terminates the transfer by pulling the SDA line 'HIGH' when the clk line SCL is at logic 'HIGH'

The I2C bus supports three different data rates: Standard Mode (Data rate up to 100kbits/sec, (100 kbps)), Fast Mode (Data rate up to 400kbits/sec (400 kbps)) & High Speed mode (Data rate up to 3.4 Mbits/sec (3.4 Mbps)).

Serial Peripheral Interface (SPI) Bus:

The Serial Peripheral Interface Bus (SPI) is a synchronous bi-directional full duplex four-wire serial interface bus. The SPI was introduced by Motorola. The SPI is a single Master Multi-Slave system.

The SPI requires four signal lines for communication.

① Master Out Slave In (MOSI): Signal line carrying the data from master to slave device. It is also known as slave i/p / Slave Data In (SI/SDI).

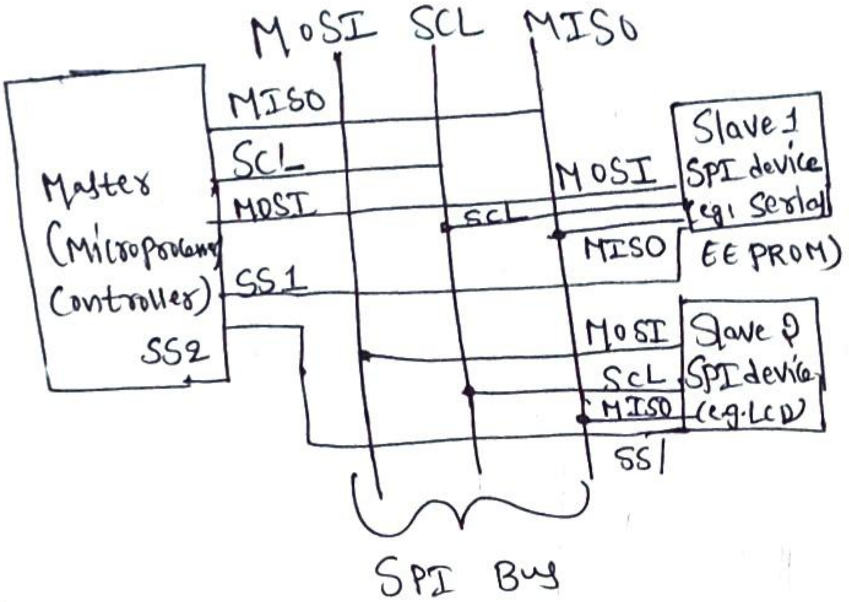
② Master In Slave Out (MISO): Signal line carrying the data from slave to master device. It is also known as Slave o/p (SO/SDO).

③ Serial Clock (SCLK): Signal line carrying the clk signals.

④ Slave Select (SS): Signal line for slave device select. It is an active low signal.

The Master device is responsible for generating the clk signal. It selects the required slave device by asserting the corresponding slave device's slave select signal 'LOW'. The data out line (MISO) of all the slave devices when not selected floats at high Impedance state.

SPI works on the principle of 'Shift register'. The master & slave devices contain a special shift register for the data to transmit or receive. The size of the shift register is device dependent. Normally it is a multiple of 8. During transmission from the master to slave, the data in the master's shift register is shifted out to the MOSI pin and it enters the shift register of the slave device through the MOSI pin of the slave device.


MOSI SCL MISO
MISO
SCL
MOSI
Master (Microprocessor/ Controller)
SS1
SS2
SCL
MOSI
MISO
Slave 1 SPI device (eg: Serial EE PROM)
MOSI
SCL
MISO
Slave 2 SPI device (eg.LCD)
SS
SPI Bus

→ SPI Bus Interfacing

Universal Asynchronous Receiver Transmitter (UART)

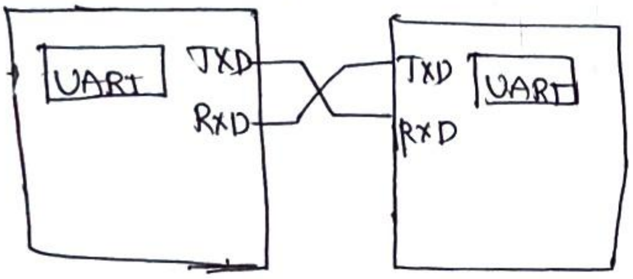
→ UART based on data transmission is an asynchronous form of serial data transmission. UART based Serial data transmission doesn't require a clock signal to synchronise the transmitting end & receiving end for the transmission.

The serial communication settings (Baudrate, number of bits per byte, Parity, number of Start bits and stop bit and flow control). for both TX & RX should be set as identical.

while Sending a byte of data , a start bit is added first and a stop bit is added at the end of the bit stream. The least significant bit of the data byte follows the 'start' bit.

The 'start' bit informs the receiver that a data byte is about to arrive. The receiver device starts polling its 'receive line' as per the baudrate settings. If the baudrate is 'x' bits per second, the time slot available for one bit is 1/x seconds. The receiver unit polls the receiver line at exactly half of the time slot available for the bit.

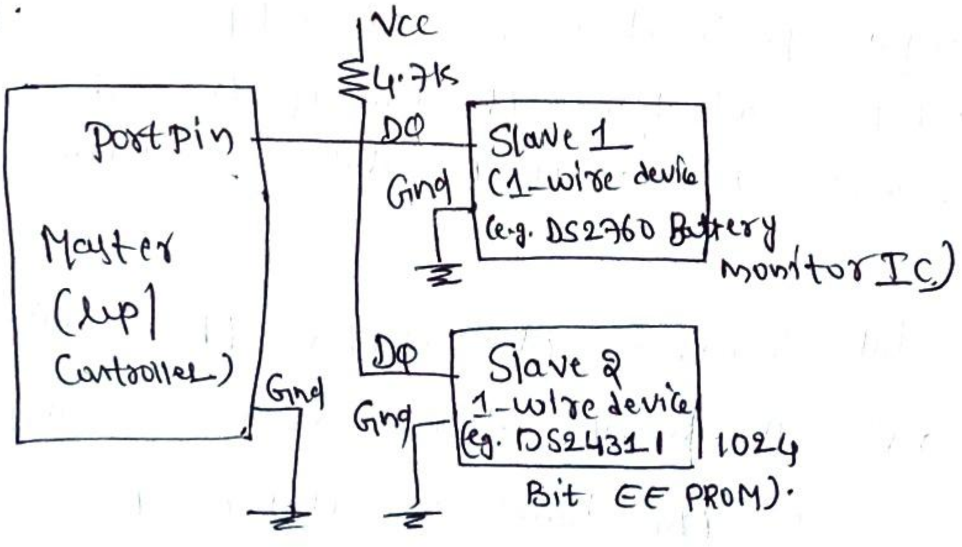
If the Parity is enabled for communication, the UART of the Transmitting device adds a parity bit (bit value is 1 for odd number of 1s in the transmitted bit stream and 0 for even number of 1s)



For proper communication, the 'Transmit line' of the sending device should be connected to the 'Receive line' of the Receiving device.

TxD: Transmitter line
RxD: Receiver line

1-wire Interface: 1-wire Interface is an asynchronous half-duplex communication protocol developed by Maxim Dallas Semiconductor. It is also known as Dallas 1-wire Protocol. It makes use of only a single signal line (wire) called DQ for communication and follows the master-slave communication model.



1-wire Interface bus

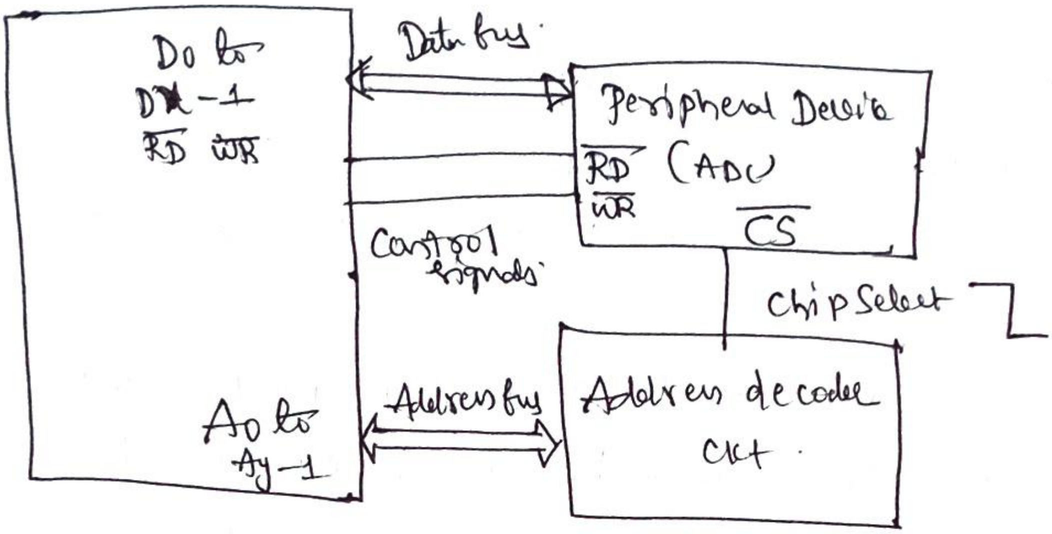
1. The Master device Sends a 'Reset' Pulse on the 1-wire bus
2. The slave device(s) present on the bus respond with a 'presence' pulse.
3. The Master device sends a ROM Command (64-bit address _net Address). This addresses the slave device(s) to which it wants to initiate a communication.
4. The Master device sends a read/write fun Command to R/W the internal Memory or register of the slave device.
5. The Master initiates a Read data/write data from the device or to the device.

All communication over the 1-wire bus is Master initiated. The communication over the 1-wire bus is divided into time slots of 60μs. The 'reset' pulse occupies 8 time slots. For starting a communication, the master asserts the reset pulse by pulling the 1 wire bus 'Low' for at least 8 time slots. If a 'slave' device is present on the bus and is ready for communication it should respond to the Master with a 'presence' pulse, within 60μs of the 'Reset' pulse by the Master.

Parallel Interface

The on-board Parallel Interface is normally used for communicating with peripheral devices which are memory mapped to the Host of the system. The Host / Controller of the E-S contains a Parallel bus & the device which supports Parallel bus can directly connect to this bus system. The device is normally memory mapped to the host processor & a range of address is assigned to it. An address decoder ckt. is used for generating the chip select signal for the device.

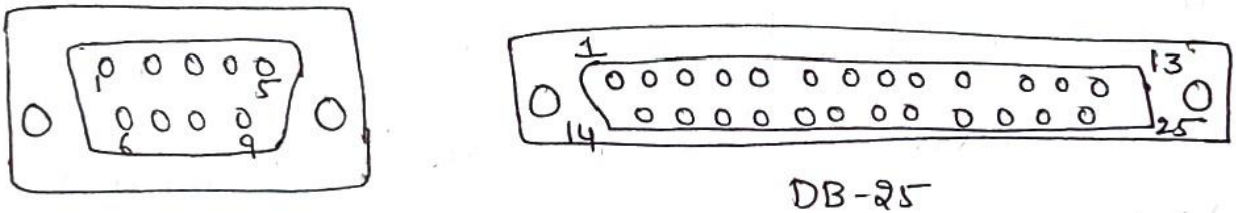
The decoder ckt activates the chip select line & thereby the device becomes active. The width of the parallel interface is determined by the data bus width of the Host processor. It can be 4-bit, 8-bit, 16-bit, 32-bit, 64-bit.



x: Data bus width

y: Address bus width.

UART uses the standard TTL/CMOS logic (Logic 'High' corresponds to bit value 1 & Logic 'LOW corresponds to bit value 0) for bit transmission whereas RS-232 follows the EIA standard for bit transmission. As per the EIA standard, a logic '0' is represented with voltage b/w +3 & +25V & a logic '1' is represented with voltage b/w -3 & -25V. In EIA standard, logic '0' is known as 'Space' and logic '1' as 'Mark'. The RS-232 Interface defines various handshaking & control signals for communication apart from the 'Transmit' & 'Receive' signal lines for data communication. RS-232 supports two different types of connectors, namely, DB-9: 9-Pin Connector & DB-25: 25-Pin Connector



DB-25

DB-9

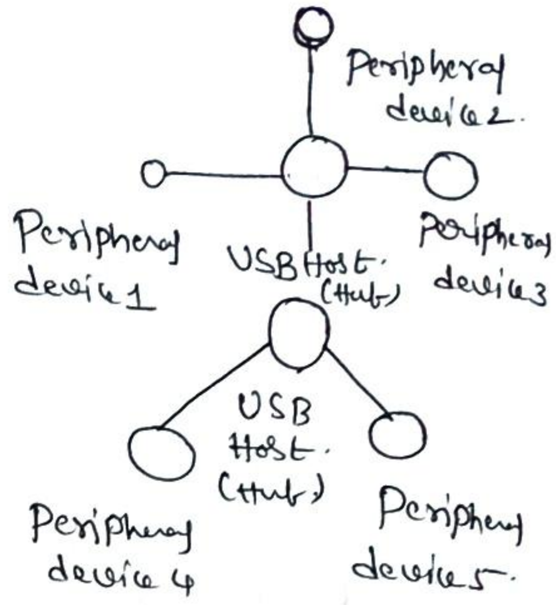
DB-9 & DB-25 RS-232 Connector Interface.

RS-232 single ended transmission
RS-422 differential data communication.

RS-232 is a point-to-point communication interface and the devices involved in RS-232 communication are called DTE (Data Terminal Equipment) & Data Communication Equipment (DCE).

Universal Serial Bus (USB): This is a USB is a wired high speed serial bus for data communication. The first version of USB (USB1.0) was released in 1995 and was created by the USB core group members consisting of Intel, Microsoft, IBM, Compaq, Digital & Northern Telecom.

Pin no.	Pin name	Description
1	V_{Bus}	Carries power (5V)
2	D−	Differential data carrier line.
3	D+	Differential data carries line
4	GND	Ground signal line



USB uses differential signals for data transmission. It improves the noise immunity. USB interface has the ability to supply power to the connecting devices. Two connection lines (Ground & Power) of the USB interface are dedicated for carrying power. It can supply power up to 500mA at 5V. Mini & Micro USB connectors are available for small form factor devices like portable media players.

→ USB Device connection

Star topology

→ Each USB device contains a Product ID (PID) a Vendor ID (VID).

→ USB supports for 4 types of data transfers namely, Control, Bulk, Isochronous & Interrupt.

→ Control transfer is used by USB system S/w to query, configure & issue commands to the USB device.

→ Bulk transfer is used for sending a block of data to a device. Bulk transfer supports error checking & correction.

Ex: Transferring data to a printer is an example for bulk transfer.

→ Isochronous data transfer is used for real-time data communication. This is not support for error checking & re-transmission of data

Interrupt transfer is used for transferring small amount of data.

→ USB supports for four different data rates, Low speed (1.5Mbps), Full Speed (12Mbps), High speed (480Mbps), Super speed (4.8Gbps)

The Low speed & Full speed specifications are defined by USB 1.0 and the High speed specification is defined by USB 2.0. USB 3.0 defines the specifications for Super speed.

IEEE 1394 (Firewire) – This is a wired, Isochronous high speed serial communication bus. It is also known as High performance serial Bus (HPSB). Apple Inc's implementation of 1394 protocol is popularly known as Firewire. i. LINK is the 1394 implementation from Sony corporation and Lynx is the implementation from Texas Instruments. 1394 supports peer-to-peer connection and point-to-multipoint communication allowing 63 devices to be connected on the bus in a tree topology.

The first version IEEE 1394-1995 released in 1995 to the recent version IEEE 1394-2008 released in June 2008. The 1394 Standard supports a data rate of 400 to 3200 Mbits/second. The IEEE 1394 uses differential data transfer and the interface cable supports 3 types of connectors, namely; 4-pin connector, 6-pin connector (alpha connector) & 9-pin connector (β connector)

There are two differential data transfer lines A & B per connector. In a 1394 cable, normally the differential lines of A are connected to B (TPA+ to TPB+ & TPA− to TPB) (∵ TPB−, Differential signal line for signal line B)

The data rate supported by 1394 is far higher than the one supported by USB 2.0 interface. The 1394 H/W implementation is much costlier than USB implementation

* Infrared (IrDA): Infrared communication technique uses Infrared waves of the electromagnetic Spectrum for transmitting the data. IrDA supports point-to-point & point-to-multipoint communication, provided all devices involved in the communication are within the Line of Sight. The typical communication range for IrDA lies in the range 10cm to 1m. The range can be increased by increasing the transmitting power of the IR device. IR supports data rates ranging from 9600 bits/second to 16Mbps. Depending on the speed of data transmission IR is classified into Serial IR (SIR), Medium IR (MIR), Fast IR (FIR), Very Fast IR (VFIR), & ultra fast IR (UFIR). SIR supports transmission rates using from 9600 bps to 115.2 kbps, MIR supports data rates of 0.576 Mbps & 1.152 Mbps. FIR supports data rates upto 4Mbps. VFIR is designed to support high data rates upto 16 Mbps. The UFIR specs are under development and is a data rate up to 100Mbps.

IrDA communication involves a Transmitter unit for transmitting the data over IR & a receiver for receiving the Data. Infrared Light Emitting Diode (LED) is the IR source for Transmitter & at the receiving end a photodiode acts as the receiver.

IrDA Communication has two essential parts; a physical link part & a protocol part. The physical link is responsible for the physical transmission of data b/w devices supporting IR communication & protocol part is responsible for defining the rules of communication.

The IrDA Control Protocol contains for Physical layer (PHY), Media Access Control (MAC) & Logical Link control (LLC).

IrDA is a popular interface for file exchange & data transfer in low cost devices. IrDA was the prominent communication channel in mobile phones before Bluetooth's existence.

* Bluetooth (BT): Bluetooth is a low cost, low power, short range wireless technology for data and voice communication.

Bluetooth was first proposed by 'Ericsson' in 1994. Bluetooth operates at 2.4 GHz of the Radio Frequency spectrum and uses the Frequency Hopping Spread Spectrum (FHSS) technique for communication. The physical link is responsible for the physical transmission of data b/w devices supporting Bluetooth communication & protocol part is responsible for the rules of communication.

The physical link works on the wireless principle making use of RF waves for communication. Bluetooth enabled devices essentially contain a Bluetooth wireless radio for the transmission & reception of data. Each Bluetooth device will have a 48-bit unique identification number.

Bluetooth supports point-to-point & point-to-multipoint (device to multiple device broadcasting) wireless communication. A Bluetooth device can function as either master or slave. When a n/w is formed with one Bluetooth device as master & more than one device as slaves, It is called a piconet. A piconet supports a maximum of seven slave devices.

The Generic Access Profile (GAP) defines the requirements for detecting a Bluetooth device & establishing a connection with it.

→ The easiest communication channel for transferring ringtones, music files, pictures, media files etc.

→ Serial Port profile (SPP) for serial data communication, File Transfer profile (FTP) for file transfer b/w devices, Human Interface Device (HID) for supporting human Interface devices like keyboard & Mouse are examples for Bluetooth profiles.

* Wi-Fi : Wi-Fi or wireless Fidelity is the popular wireless communication technique for networked communication of devices. Wi-Fi follows the IEEE 802.11 standard. Wi-Fi is intended for n/w communication & it supports Internet protocol (IP) based communication.

Wi-Fi based communications require an Intermediate agent called wi-fi router/wireless Access point to manage the communication. Wi-Fi enabled devices contain a wireless adaptor for TX & RX data in the form of radio signals through an antenna. Wi-Fi operates at 2.4GHz or 5GHz of radio spectrum & they co-exist with other ISM band devices like Bluetooth.

For communicating with devices over a Wi-Fi n/w, the device when its wi-fi radio is turned ON, searches the available Wi-Fi n/w in its vicinity and lists out the Service Set Identifier (SSID) of the available n/wks. If the n/w is security enabled, a password may be required to connect to a particular SSID.

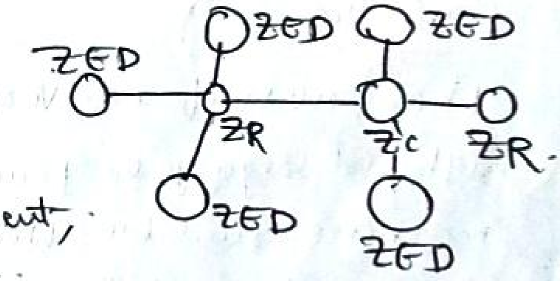
Wi-Fi supports data rates ranging from 1Mbps to 150Mbps depending on the standards (802.11 a/b/g/n) & access/modulation method. wi-Fi offers a range of 100 to 300 feet.

★ **ZigBee**: ZigBee is a low power, low cost, wireles communication protocol based on the IEEE 802.15.4-2006 standard. The ZigBee specifications support a robust mesh n/w containing multiple nodes. ZigBee operates worldwide at the unlicensed bands of Radio spectrum, mainly at 2.400 to 2.484 GHz, 902 to 928 MHz & 868.0 to 868.6 MHz. ZigBee supports an operating distance of upto 100 meters at a data rate of 20 to 250kbps.

ZigBee Coordinator (ZC) / Network coordinator: The ZC is responsible for initiating the ZigBee n/w and it has the capability to store information about the n/w.

ZigBee Router (ZR) / Full function Device (FFD): Responsible for passing information from device to another device.

ZigBee End Device (ZED) / Reduced Function Device (RFD): End Device containing ZigBee functionality for data communication.



→ A Zigbee n/w Model

ZigBee is primarily targeting application areas like Home & industrial automation, energy Management, Home control / Security, Medical / patient tracking, logistics & asset tracking & sensor n/wks & active RFID; Automatic Meter Reading (AMR), smoke detectors, wireless telemetry, HVAC control, Heating control, lighting control, environmental controls, etc.

★ **General Packet Radio Service (GPRS)**: General Packet Radio Service (GPRS) is a communication technique for transferring data over a mobile communication n/w like GSM.

Data is Sent as Packets in GPRS communication. The transmitting device splits the data into several related packets. At the receiving end the data is re-constructed by combining the received data Packets. The GPRS Communication divides the channel into 8 timeslots and transmits data over the available channel.

GPRS is an old technology and it is replaced by like EDGE, High Speed Downlink Packet Access (HSDPA)., which offers higher bandwidths for communication.

*. Embedded Firmware

(1) write the program in high level languages like Embedded C/C++ using an Integrated Development Environment.

(2) write the program in Assembly language using the instructions supported by your application's target Processor/Controller.

The process of converting the program written in either a high level language or processor/Controller specific Assembly Code to machine readable binary code is called 'HEX File creation'. The methods used for 'HEX File creation' is different depending on the programming techniques used.

Two types of control algorithm design exist in embedded firmware development. The first type of control algorithm development is known as the infinite loop or 'Super loop' based approach, where the control flow runs from top to bottom and then jumps back to the top of the program in a conventional procedure. It is similar to the while (1) {} based technique in C

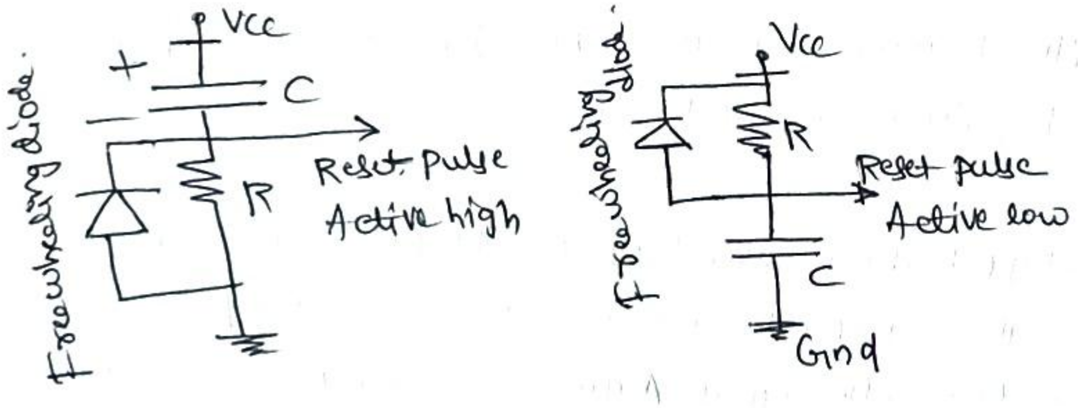
The second method deals with splitting the functions to be executed into tasks and running these tasks using a scheduler which is part of a General purpose or Real Time Embedded operating System (GPOS/ RTOS).

Other System Components

Reset circuit.

The Reset signal can be either active high or active low. Since the processor operation is synchronised to a clock signal, the reset pulse should be wide enough to give for the clk oscillator to stabilise before the internal reset state starts. The reset signal to the processor can be applied at power ON through an external passive reset ckt comprising a capacitor & resistor or through a standard Reset IC like MAX810 from Maxim Dallas.

The reset pulse width can be adjusted by changing the resistance value R & capacitance value C.

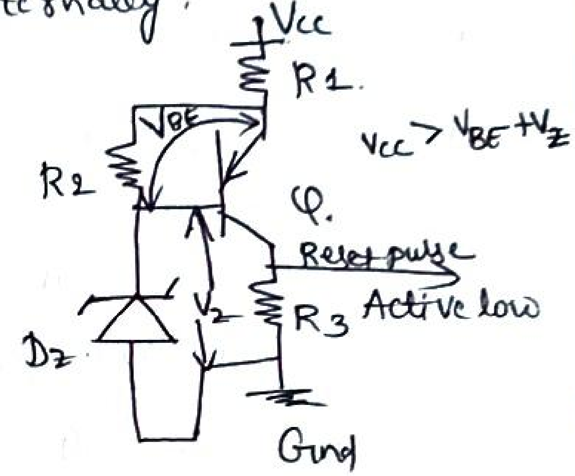


RC based Reset ckt

Brown-out Protection circuit

Brown out Protection ckt prevents the processor/ controller from unexpected program execution behaviour when the supply voltage to the processor/ controller falls below a specified voltage.

A brown-out protection ckt holds the processor/Controller in reset state, when the operating voltage falls below the threshold, until it rises above the threshold voltage. Certain processors/Controllers support built in brown-out protection ckt which monitors the supply voltage internally.

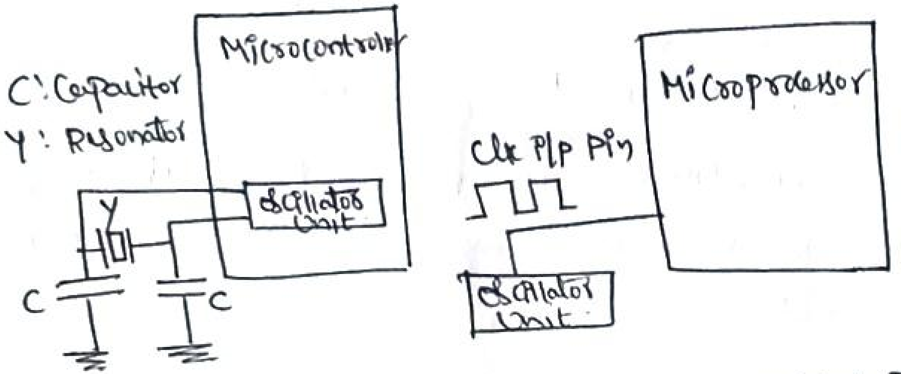


Brown-out protection ckt with Active low O/P

The Zener Diode D_Z & transistor Q forms the heart of this ckt. The transistor conduct always when the supply voltage V_{cc} is greater than that of the sum of V_{BE} & V_Z (zener voltage). The transistor stops conducting when the supply voltage falls below the sum of V_{BE} & V_Z. Select the Zener diode with required voltage for setting the low threshold value for V_{cc}. The values of R1, R2 & R3 can be selected based on the electrical characteristics of the transistor in use. Microprocessor Supervisor ICs like DS1232 from Maxim Dallas also provides Brown-out Protection.

Oscillator Unit

A µp/µc is a digital device made up of digital combinational & sequential ckts. The instruction execution of a µp/controller occurs in sync with a clk signal. A Quartz crystal is normally mounted in a hermetically sealed metal case with two leads protruding out of the case. The speed of operation of a processor is primarily dependent on the clk frequency. The total system power consumption is directly proportional to the clk frequency. The power consumption increases with increase in clk frequency. The accuracy of the crystal oscillator or ceramic resonator is normally expressed in +/- ppm (parts per million)



Oscillator circuitry using Quartz crystal & Quartz crystal oscillator.

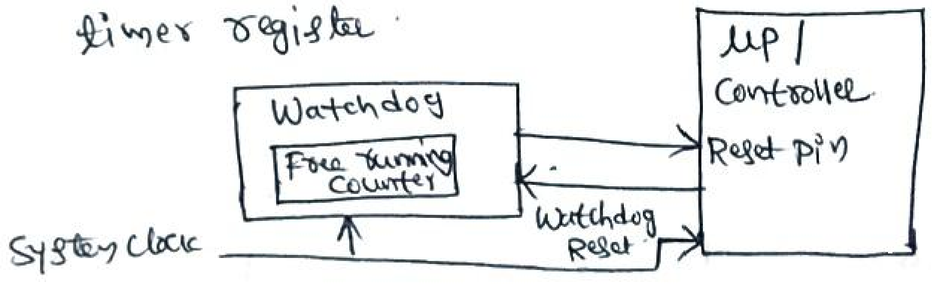
Real-Time clock (RTC)

RTC holds Information like current time (In hours, Minutes & seconds) in 12 hour / 24 hour format, date, month, year, day of the week, etc and supplies timing reference to the system.

RTCs are available in the form of ICs from different semiconductor manufacturers like Maxim / Dallas, ST Micro electronics etc. The OS Kernel identifies the Interrupt in terms of the Interrupt Request (IRQ) number generated by an Interrupt controller.

Watchdog Timer

The watchdog Timer Increments or decrements a free running counter with each clock pulse and generates a reset signal to reset the processor if the count reaches zero for a down counting watchdog, The highest count value for an up counting Watchdog. If the watchdog counter is in the enabled state, the firmware can write a zero to it before starting the execution of a piece of code and the watchdog will start counting. If the firmware execution completes before the expiration of the watchdog Timer you can reset the count by writing a 0 to the watchdog timer register.



Watchdog timer for firmware execution supervision

PCB AND PASSIVE Components

PCB acts as a platform for testing your embedded firmware. The passive components are R, C, Diodes, etc on your board. They are the co-workers of various chips contained in your embedded H/W. They are very essential for the proper functioning of your embedded system.