IES SEM4 Q&A

1.What is an Embedded System? Explain the different applications of Embedded System.

Embedded System

- An embedded system is a combination of computer hardware and software designed for a specific function.
- Embedded systems may also function within a larger system.
- The systems can be programmable or have a fixed functionality.

Applications:-

- **GPS Systems**

The GPS is a navigation system that synchronizes data about location, time, and velocity using satellites and receivers.

- **Fitness Trackers**

Wearable fitness trackers can monitor your health and track activities such as sleeping, running, and walking.

- **Medical Devices**

Embedded systems have been used in medical equipment at healthcare institutions for a long time.

- **Automotive Systems**

Automobile embedded systems are created and installed to improve vehicle safety.

The number of traffic fatalities has decreased in recent years as a result of car safety systems.

- **ATMs**

An automated teller machine (ATM) is a computerized banking machine that communicates across a network with a host bank computer.

- **Banking Sector**

Embedded systems are also used in the banking sector for security purposes in several domains.

- **Defense and Aerospace**

Missile Guidance Systems, Systems for navigation and guidance, GPS, Space Exploration (Rovers) use the embedded systems.

- **Signal Systems**

Signaling systems use embedded technology, which ensures your safety during your journey.

- **Automobiles Sector**

Nowadays, modern cars have a variety of embedded systems that perform a variety of activities based on their uses in the vehicle.

- **Home Appliances**

Embedded systems are also employed in a variety of household appliances that you use daily and on which you are completely reliant.

2.Differentiate between RISC and CISC processors.

1.RISC is a reduced instruction set.

1.CISC is a complex instruction set.

2.The number of instructions is less as compared to CISC.

2.The number of instructions is more as compared to RISC.

3.The addressing modes are less.

3.The addressing modes are more.

4.It works in a fixed instruction format.

4.It works in a variable instruction format.

5.The RISC consumes low power.

5.The CISC consumes high power.

6.The RISC processors are highly pipelined.

6.The CISC processors are less pipelined.

7.It optimizes the performance by focusing on software.

7.It optimizes the performance by focusing on hardware.

8.Requires more RAM.

8.Requires less RAM.

3.Explain the various purposes of embedded system.

Embedded System

- An embedded system is a combination of computer hardware and software designed for a specific function.
- Embedded systems may also function within a larger system.
- The systems can be programmable or have a fixed functionality.

Purpose of Embedded System

- The purpose of embedded systems is to control a specific function within a device.
- They are usually designed to only perform this function repeatedly, but more developed embedded systems can control entire operating systems.

There are 3 major purpose of Embedded System

- **Receiving Information**

Receiving information or data from various sources. and also molding and checking the compatibility of this data.

- **Processing information**

Using calculations, numerical, legitimate, and investigative methods to analyze meaning out of data, and process significant information, find out the desired conclusions, observations, and desired results.

- **Conveying Information**

Provide the outcomes, information, and so forth to the various entities which could utilize it seriously, and giving an interface to such elements.

4.What are the characteristics of an Embedded System?

Embedded System

- An embedded system is a combination of computer hardware and software designed for a specific function.
- Embedded systems may also function within a larger system.
- The systems can be programmable or have a fixed functionality.

Characterstic

Unlike general computer systems, embedded systems work only for a particular function in a time-bound manner.

For instance, a washing machine can not multitask like a laptop.

In this regard, here are some unique characteristics of an embedded system.

- **Sophisticated Functionality**

The functionality of no two embedded system applications is bound to be the same. The functionality of a washing machine is different from that of a microwave.

- **Real-Time Operation**

It doesn't mean live operation. It means the software programs hardware to operate in a time-bound fashion. It could also have two modes: Hard and Soft.

- **Low Manufacturing Cost**

As an embedded system design aims for any particular application, it involves less manufacturing cost as compared to a versatile general computing system. As a result, embedded systems also require less power to perform operations.

- **Processor and Memory**

Depending on the type, processor and memory requirements may vary. For instance, small embedded systems would require less memory, but sophisticated systems demand more memory and run on multi-core processors.

- **Tight Design Constraint**

There are many design constraints to consider around the cost, performance, size, and power of an embedded system to realize its absolute performance. These design factors are kept to a minimum to justify their simple function.

5.What is Read-Write Memory? Explain the categories of Read-Write Memory.

Read Write memory

- Read-write memory (RWM) is computer memory that can be read from and written to.
- This type of memory can be contrasted with read-only

memory, which cannot be modified after it is written.

- In general, the use of read-write memory represents the need for users to continually update the data that is held on hardware storage devices.
- Read-write memory comes in all kinds of physical setups.

Categories

RAM generally falls into three categories:

Static RAM, dynamic RAM and non-volatile RAM.

Static RAM (SRAM)

- Static RAM stores data in the form of voltage. They are made up of flip-flops. Static RAM is the fastest form of RAM available.
- In typical implementation, an SRAM cell (bit) is realised using six transistors (or 6 MOSFETs).
- Four of the transistors are used for building the latch (flip-flop) part of the memory cell and two for controlling the access.

- SRAM is fast in operation due to its resistive networking and switching capabilities.

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.
- The MOSFET acts as the gate for the incoming and outgoing data whereas the capacitor acts as the bit storage unit.

NVRAM Non-volatile RAM

- It is a 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.

6. Explain Automotive Communication Buses.

Defination

- The number of sensors, actuators, entertainment and navigation systems and their corresponding electronic control units in the typical automobile has been growing exponentially.

- Digital devices and systems must communicate via an electrical or optical signal employing a well defined protocol.
- These signals and protocols constitute a communications bus.

Controller Area Network (CAN):

- The CAN bus was originally proposed by Robert Bosch, pioneer in the Automotive embedded solution providers.
- It supports medium speed and high speed data transfer.
- CAN is an event-driven protocol interface with support for error handling in data transmission.
- It is generally employed in safety systems like airbag control; powertrain systems like engine control and Antilock Brake System (ABS); and navigation systems like GPS.

Local Interconnect Network (LIN):

- The LIN bus is a single master multiple slave (up to 16 independent slave nodes) communication interface.

- LIN is a low speed, single wire communication interface with support for data rates up to 20 Kbps and is used for sensor/actuator interfacing.
- LIN bus follows the master communication triggering technique to eliminate the possible bus arbitration problem that can occur by the simultaneous talking of different slave nodes connected to a single interface bus.
- LIN bus is employed in applications like mirror controls, fan controls, seat positioning controls, window controls, and position controls where response time is not a critical issue.

Media-Oriented System Transport (MOST):

- The Media-oriented system transport (MOST) is targeted for automotive audio/video equipment interfacing, used primarily in European cars.
- A MOST bus is a multimedia fibre-optic point-to-point network implemented in a star, ring or daisy-chained over optical fibre cables.

- The MOST bus-specifications define the physical (electrical and optical parameters) layer as well as the application layer, network layer, and media access control.
- MOST bus is an optical fibre cable connected between the Electrical Optical Converter (EOC) and Optical Electrical Converter (OEC), which would translate into the optical cable MOST bus.

7. What is Direct Memory Access (DMA)?

Defination

- DMA is a technique for transferring blocks of data directly between two hardware devices.
- In the absence of DMA the processor must read the data from one device and write it to the other one byte or word at a time.
- **DMA Presence Advantage**: The DMA Controller performs the entire transfer with little help from the Processor.
- **DMA Absence Disadvantage**: If the amount of data to be

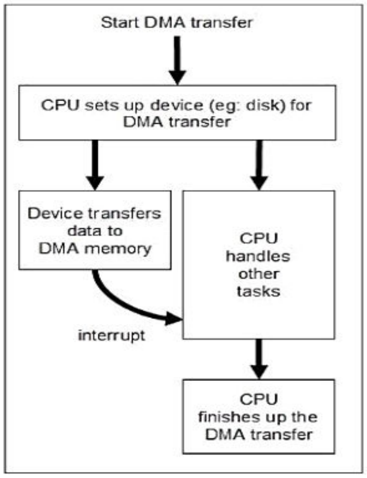
transferred is large or frequency of transfer is high the rest of the software might never get a chance to run.

Working

- Direct memory access (DMA) is a means of having a peripheral device control a processor's memory bus directly.

- DMA permits the peripheral, such as a UART, to transfer data directly to or from memory without having each byte (or word) handled by the processor.

- Thus DMA enables more efficient use of interrupts, increases data throughput, and potentially reduces hardware costs by eliminating the need for peripheral-specific FIFO buffers.


Start DMA transfer
CPU sets up device (eg: disk) for DMA transfer
Device transfers data to DMA memory
CPU handles other tasks
interrupt
CPU finishes up the DMA transfer

DMA Controller

- In a typical DMA transfer, some event notifies a separate device called the DMA controller that data needs to be transferred to memory.
- The DMA controller then asserts a DMA request signal to the CPU, asking its permission to use the bus.
- The CPU completes its current bus activity, stops driving the bus, and returns a DMA acknowledge signal to the DMA controller.

8. Explain the working of embedded system with respect to Washing Machine.

- The actuator part of the washing machine consists of a motorised agitator, tumble tub, water drawing pump and inlet valve to control the flow of water into the unit.
- The sensor part consists of the water temperature sensor, level sensor, etc. The control part contains a microprocessor /controller based board with interfaces to

the sensors and actuators.

- The sensor data is fed back to the control unit and the control unit generate; the necessary actuator outputs.
- The control unit also provides connectivity to user interfaces like keypad for setting the washing time, selecting the type of material to be washed like light, medium, heavy duty, etc.
- Washing machine comes in two models, namely, top loading and front loading machines.
- In top loading models the agitator of the machine twists back and forth and pulls the cloth down to the bottom of the tub.
- On reaching the bottom of the tub the clothes work their way back up to the top of the tub where the agitator grabs them again and repeats the mechanism.
- In the front loading machines, the clothes are tumbled and plunged into the water over and over again.

9. Differentiate between Microprocessor and Microcontroller.

1.Microprocessor is the heart of Computer system.

1.Micro Controller is the heart of an embedded system.

2.It is only a processor, so memory and I/O components need to be connected externally.

2.Micro Controller has a processor along with internal memory and I/O components.

3.Memory and I/O has to be connected externally, so the circuit becomes large.

3.Memory and I/O are already present, and the internal circuit is small.

4.You can't use it in compact systems.

4.You can use it in compact systems.

5.Cost of the entire system is high.

5.Cost of the entire system is low.

6.Most of the microprocessors do not have power saving features.

6.Most of the microcontrollers offer power-saving mode.

7.It is mainly used in personal computers.

7.It is used mainly in a washing machine, MP3 players, and embedded systems.

8.Microprocessors are based on Von Neumann model

8.Micro controllers are based on Harvard architecture.

9.It uses an external bus to interface to RAM, ROM, and other peripherals.

9.It uses an internal controlling bus.

10.It's complex and expensive, with a large number of instructions to process.

10.It's simple and inexpensive with less number of instructions to process.

11.Microprocessors are not efficient.

11.Microcontrollers are efficient.

12.Microprocessors have a zero status flag.

12.Microcontroller doesn't have a zero status flag.

10. Write 8051 C program to generate delay using timer register.

Code:-

```
#include<REG51.H>

void T0Delay(void);

void main(void)              {

while (1)         {

P1=0xFF;

ToDelay();

P1=0x00;

ToDelay();

}

}

void ToDelay()       //delay function              {
```

```
TMOD=0x01;

TL0=0xFC;

TH0=0x65;

TR0=1;

while (TF0==0);

TR0=0;

TF0=0;

}
```

11.Explain the following:-

(a)RXd (b)TXD (c)INT0 & INT1 (d)T0 & T1 (e)PSEN

RXD Pin no. 10

- Pin 10 is used as a RXD (serial data receive pin) which is for serial input pin. By using this input signal microcontroller receives data for serial communication.

TXD Pin no. 11

- Pin 11 is used as a TXD (serial data transmit pin) which is serial output pin. By using this output signal microcontroller transmits data for serial communication.

INT0 & INT1 Pin no. 12 and 13

- Pins12 and 13 are used for External Hardware Interrupt 0 and Interrupt 1 respectively.
- When this interrupt is activated (i.e. when it is low), 8051 gets interrupted means it stopped whatever it is doing and jumps to the vector table where ISR's (Interrupt Service Routine) are stored and starts performing Interrupt Service Routine (ISR) from that vector location.

T0 and T1 Pin no. 14 and 15

- Pin 14 and 15 are used for Timer 0 and Timer 1 external input. They can be connected with 16-bit timer/counter.

PSEN (Program store Enable) Pin No 29

- This is also an active low pin, i.e., it gets activated after applying a low pulse.
- It is an output pin and used along with the EA pin in 8031

based (i.e. ROMLESS) Systems to allow storage of program code in external ROM.

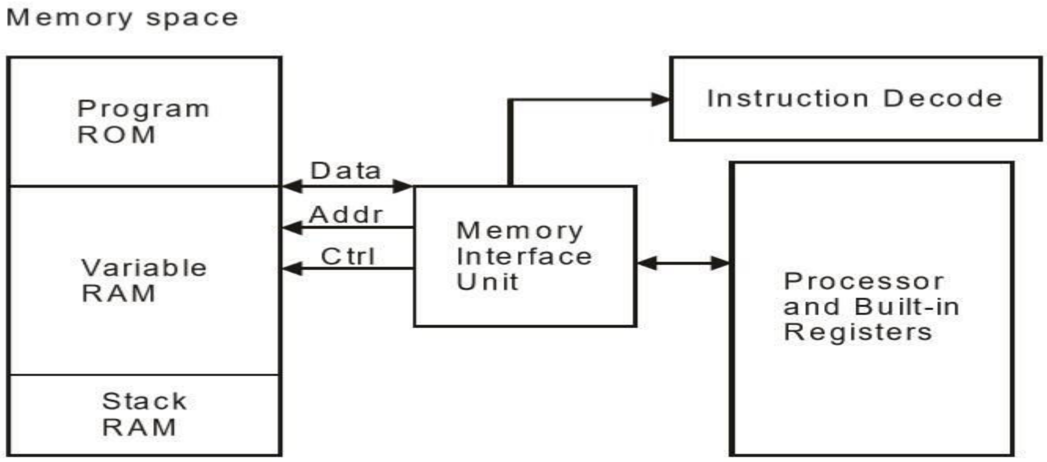
12. Write 8051 C program to convert packed BCD 0x29 to ASCII and display the bytes on P1 and P2.

Code:-

```
#include <reg51.h>
void main(void)            {
unsigned char x,y,z;
unsigned char mybyte=0x29;
x=mybyte&0x0F;
P1=x|0x30;
y=mybyte&0xF0;
y=y>>4;
P2=y|0x30;
}
```

13. Explain the Von-Neumann Memory Model for 8051.

- The 8051 microcontrollers work with 8-bit data bus.
- So they can support external data memory up to 64K and external program memory of 64k at best.
- Collectively, 8051 microcontrollers can address 128k of external memory.
- When data and code lie in different memory blocks, then the architecture is referred as Harvard architecture.
- In case data and code lie in the same memory block, then the architecture is referred as **Von Neumann architecture**.



<u>Von Neumann Architecture</u>

- The Von Neumann architecture was first proposed by a computer scientist John von Neumann.
- In this architecture, one data path or bus exists for both instruction and data.
- As a result, the CPU does one operation at a time.
- It either fetches an instruction from memory, or performs read/write operation on data.
- Von-Neumann architecture supports simple hardware.
- It allows the use of a single, sequential memory.

14. What are the factors to be considered in selecting a controller? (ANY 5 Point)

1. Know Hardware Requirements

- The first thing to consider while choosing a microcontroller is to consider what hardware devices your project need.
- You must have an idea what hardware peripherals you need to connect with your microcontroller.

2. Know Software Requirements

- Just as hardware of the project, you need to know the software required by your project.
- The software requirement is also important to consider.

3. Architecture

- The architecture of a microcontroller is the internal structure.
- When you have the idea about hardware and software requirements of the project, you can look for different type of microcontrollers to find the matching ones.
- They have a number of families and come in a variety of types that include 4 bit, 8 bit, 16 bit and 32 bit. 32 bit microcontrollers are the ones used in embedded systems.

4. Cost And Power Requirements

- Cost factor is very important while developing a project or an embedded system.
- If you require complex functions and operations from your microcontroller, cost will be higher.

- If it is for a simple application, than a cheap microcontroller can serve the purpose.

5. Memory Requirement

- The memory size is very serious issue to consider.
- How much memory is required by your code and How much RAM you need are worth considering

6. Search Microcontrollers

- Up till this step, you have all the information required to start looking for microcontrollers.
- You can use online information available for different microcontrollers.
- Also, you can take help of some professional who can direct you to some family of microcontrollers.

7. Part Compatibility

- Microcontrollers come in a variety of types.
- It is important to check whether the type you selected is compatible with other tools or not.

- If you are developing a system for mass production and use, then make sure that the microcontroller you select would be available for your product.

8. Know Software Tools

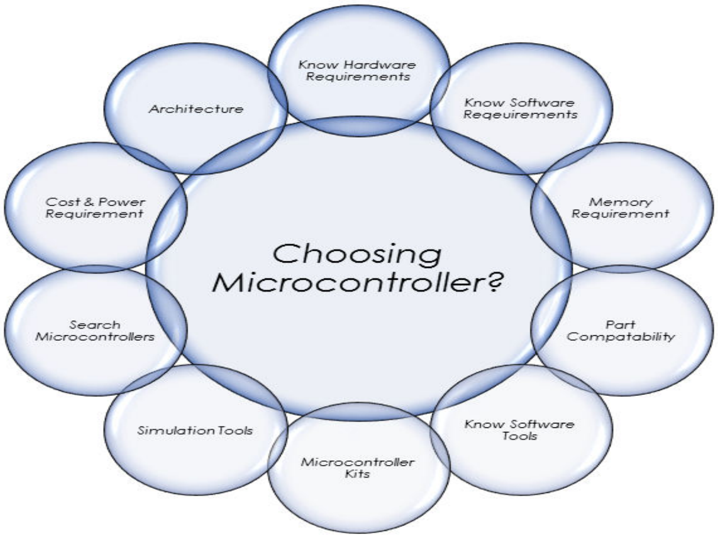
- Microcontrollers come with some standard software tools with them.
- These tools are provided by the manufacturers and are very useful.
- They come with assemblers, compilers, debuggers and simulating tools.
- So it is good to know about all the products available with the microcontroller.

9. Simulation Tools

- It is good to use simulation software to check the working of microcontroller.
- As I told in the previous point, some of the microcontrollers come with simulation tools that are very useful in the development process.

10. Start With Microcontroller Kits

- Microcontroller kits are very useful for starting your work.
- If you are beginner, then experimenting with kit can be very interesting.
- You can start with mini projects or you can build small circuits for testing purposes.
- If you already know how to work with microcontrollers then you can skip this step.



15. Explain Timer/Counter in Mode 0.

<u>Timer:-</u>

- A timer is a specialized type of clock which is used to measure time intervals.
- A timer that counts from zero upwards for measuring time elapsed is often called a stopwatch.
- It is a device that counts down from a specified time interval and used to generate a time delay, for example, an hourglass is a timer.

Counter:-

- A counter is a device that stores (and sometimes displays) the number of times a particular event or process occurred, with respect to a clock signal.
- It is used to count the events happening outside the microcontroller.
- In electronics, counters can be implemented quite easily using register-type circuits such as a flip-flop.

Timer in Mode 0:-

Mode 0 (13-Bit Timer Mode)

- Both Timer 1 and Timer 0 in Mode 0 operate as 8-bit counters (with a divide-by-32 prescaler).
- Timer register is configured as a 13-bit register consisting of all the 8 bits of TH1 and the lower 5 bits of TL1.
- The upper 3 bits of TL1 are indeterminate and should be ignored.
- Setting the run flag (TR1) does not clear the register

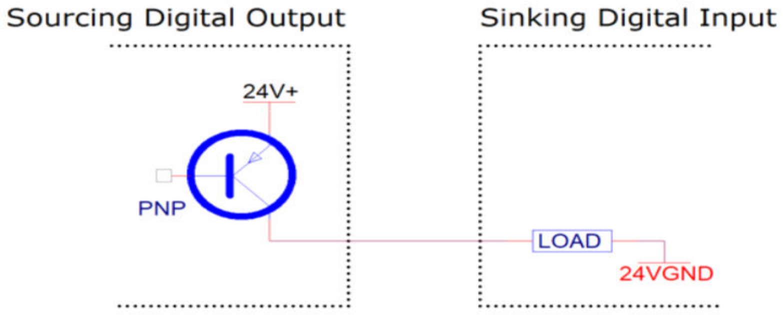
16. What is Sink current and Source current?

- Current source and current sink are two terms that are important to understand, when you choose the type of digital input or output module you want to use for you PLC system.
- Current source or current sink can only direct current in one direction, which means that the circuit will not operate if you connect it the wrong way.
- A PLC I/O circuit needs one terminal for current to enter, and another for current to exit.

- Two terminals are associated with every I/O point.
- Source current is the ability of the digital output/input port to supply current.
- Sink current is the ability of the port to receive current.
- When you have a simple circuit where a digital input connects to a digital output, you need three components; a voltage source, a ground, and a load.
- A sourcing digital I/O provides the voltage, a sinking I/O the ground, and the digital input provides the load.

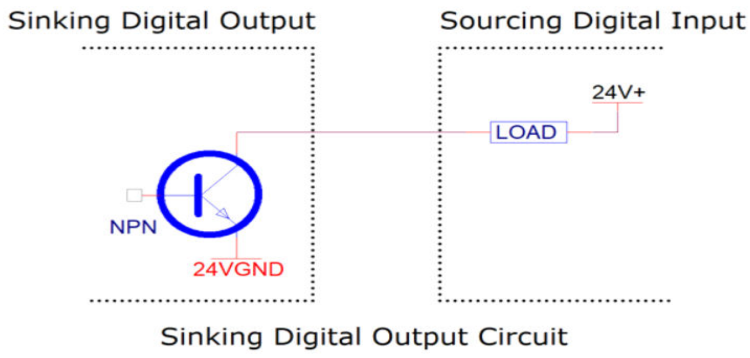
This means that the input and output works in pairs:

i. Digital source output - Digital sink input



Sourcing Digital Output Circuit

ii. Digital sink output - Digital source input



17. Explain the Types of Operating Systems.

What is an Embedded Operating System?

- An embedded operating system is a special type of computer operating system designed to optimize the functional efficiency of an embedded system.
- In the same way that your desktop machine requires an operating system like Linux, Mac OS, or Windows to run basic applications, embedded systems also require an operating system that supports their basic functionality.

Types of Embedded Operating System

- **Single System Control Loop**

A single system control loop is the most basic type of embedded operating system.

This type of OS exercises digital control over a single measured variable.

A simple embedded device like a smart home temperature control or thermostat might use this type of OS.

The user programs the desired set-point for the temperature variable.

- **Multi-tasking Operating System**

When an embedded system must perform several tasks or functions at the same time, it requires a special type of functionality called multitasking.

A multitasking operating system is programmed to allocate processing time between various system functions according to application needs.

If the system has more than one processor, it may be possible to perform more than one function simultaneously.

Multitasking operating systems include a scheduling algorithm that establishes rules for organizing and prioritizing tasks.

- **Real-Time Operating System**

A real-time operating system is meant to process inputs immediately as they are received, with the minimum possible delay between receiving a target input and generating the corresponding output.

Real-time operating systems are bound by strict time constraints and processing tasks must be completed within the specified time constraints to avoid a system failure.

- **Rate Monotonic Operating System**

Many embedded systems are developed using a special type of task scheduling algorithm known as rate monotonic scheduling.

The purpose of this unique type of algorithm is to ensure that the system can meet the requirements of a range of tasks, all of which may require special priority treatment and timing constraints.

A rate monotonic assigns the highest levels of priority to tasks that take the least time.

18. What are the objectives of EDLC(Embedded Product

Development Life Cycle)?

Defination:-

- Just like the SDLC used in Software Development, there is EDLC used in Embedded product development.
- Embedded Product Development Life Cycle is an 'Analysis -Design -Implementation' based standard problem solving approach for Embedded Product Development.

Objective:-

EDLC has three primary objectives are:

- **Ensure that high quality products are delivered to user**

Quality in any product development is Return On Investment achieved by the product

The expenses incurred for developing the product the product are:-

Initial investment

Developer recruiting

Training

Infrastructure requirement related

- **Risk minimization defect prevention in product development through project management**

In which required for product development lòose ˎor 'tight' project management

'project management is essential for 'predictability
co-ordination and risk minimization

Resource allocation is critical and it is having a direct impact on investment.

Example:- Microsoft @ Project Tool

- **Maximize the productivity**

Productivity is a measure of efficiency as well as Return On Investment

This productivity measurement is based on total manpower efficiency

Productivity in which when product is increased then investment is fall down

Saving manpower

19. What are simulators?

Simulators:

- Simulator is a tool used for simulation of an embedded system.
- Code tested for microcontroller unit by simulating code on the host computer.
- Simulator is used for model the behavior of the complete microcontroller in software.

Functions of simulators:

Let's see the functions performed by simulator are:

- It defines the processing or processor device family with various version of target system.
- It monitors the detailed information of a source code and symbolic arguments as the execution goes for each single step of operation.

- It simulates the ports of target system for each single step of execution.
- It provides the working status of RAM.
- It monitors the response of system and determines the throughput.
- It provides the complete meaning of the present command.
- It monitors the detailed information of the simulator commands entered from the keyboard or selected from the menu.
- It facilitates synchronization of internal peripherals and delays.

20. Explain the types of files generated on Cross Compilation.

<u>Cross-Compilation</u>

Cross-compilation is the process of converting a source code writtenin high level language to a target processor/controller understandablemachine code.

Types of Files Generated during the cross-compilation process are:

i. List file(.lst)

ii. Hex file(.hex)

iii. Pre-processor output file

iv. Map file(.MAP)

v. Object file(.obj)

List file(.LST File)

- Contains information about the cross-compilation process.
- Cross compiler details
- Formatted source text
- Symbol tables

Pre-processor Output File

- Contains the pre-processor output for the pre-processor instructionsused in the source file.

- The pre-processor output file is a valid C source file.

Object File(.OBJ File)

- List of some of the details stored in an object file
- Reserved memory for global variables
- Public symbol names
- External symbol references

Map File(.MAP)

- Linking and locating of relocatable object files generate a list filecalled 'linker list file' or 'map file'.

Map file contains information about the link/locate process.

- Page header
- Command line
- CPU details
- Input modules
- Memory map
- Program size

- Warnings and errors

HEX File(.HEX)

- Hex file is the binary executable file created from the source code.
- The format of hex file varies across the family ofprocessors/controllers.

Disassembler/Decompiler

- Decompiler is the utility program for translating machine codes into corresponding high level language instructions.
- Disassemblers/decompilers are deployed in reverse engineering.