19MCE304- DESIGN OF EMBEDDED SYSTEMS

The design process for embedded systems

The design process for embedded systems typically involves several stages, each crucial for ensuring the final product meets the required specifications and performs reliably. Here's a step-by-step outline of the embedded system design process:

1. Requirement Analysis

- **Define Objectives**: Clearly state what the embedded system is supposed to achieve.
- **Specify Requirements**: Gather detailed functional and non-functional requirements, including performance, power consumption, cost constraints, and environmental conditions.
- **Feasibility Study**: Assess the technical and economic feasibility of the project.

2. System Specification

- **Functional Specifications**: Document the functions the system must perform.
- **System Architecture**: Define the overall system architecture, including hardware and software components.
- **Interface Requirements**: Specify how the system will interact with other systems, devices, and users.

3. System Design

- **Hardware Design**:
  - **Component Selection**: Choose appropriate microcontrollers, sensors, actuators, memory, and other components.
  - **Schematic Design**: Create circuit diagrams and schematics.
  - **PCB Layout**: Design the printed circuit board (PCB) layout.
- **Software Design**:
  - **System Software**: Design the real-time operating system (RTOS) or firmware.

    - **Application Software**: Develop the high-level application code.
    - **Middleware**: Implement necessary middleware for communication, data management, etc.

4. Prototyping

- **Hardware Prototyping**: Assemble a prototype based on the hardware design.
- **Software Prototyping**: Develop initial versions of the software components.
- **Integration**: Integrate hardware and software components to create a working prototype.

5. Testing and Validation

- **Unit Testing**: Test individual components and modules for correct functionality.
- **Integration Testing**: Verify that integrated components work together as expected.
- **System Testing**: Conduct comprehensive testing of the entire system to ensure it meets all requirements.
- **Validation**: Validate the system against the original specifications and requirements.

6. Optimization

- **Performance Optimization**: Improve the system's performance, such as processing speed and power efficiency.
- **Resource Optimization**: Minimize the use of resources like memory and power.
- **Cost Optimization**: Reduce the cost of components and manufacturing without compromising quality.

7. Documentation

- **Design Documentation**: Create detailed documentation of the system design, including schematics, code, and interface descriptions.
- **User Manuals**: Prepare user manuals and guides for end-users and technicians.
- **Maintenance Documentation**: Document procedures for maintenance and updates.

8. Pre-Production and Pilot Run

- **Pre-Production**: Produce a small batch of units to test the manufacturing process and identify any issues.
- **Pilot Run**: Conduct a pilot run to test the system in real-world conditions and gather feedback.

9. Production

- **Mass Production**: Begin full-scale manufacturing of the embedded system.
- **Quality Control**: Implement quality control measures to ensure consistency and reliability in production.

10. Deployment and Maintenance

- **Deployment**: Distribute the final product to users or clients.
- **Monitoring and Support**: Provide ongoing support and monitor the system for issues.
- **Updates and Upgrades**: Release firmware updates and upgrades as needed.

11. Post-Mortem Analysis

- **Review**: Conduct a review of the project to identify successes and areas for improvement.
- **Lessons Learned**: Document lessons learned to improve future projects.

Iterative Development

Many embedded system designs follow an iterative approach, where the design, prototyping, testing, and validation stages are repeated multiple times to refine the system and address any issues that arise. This iterative process helps in improving the system's functionality, reliability, and performance before final deployment.

1.8 DESIGN PROCESS IN EMBEDDED SYSTEM

The concepts used during a design process are as follows.

1. *Abstraction:* Each problem component is first abstracted. For example, in the design of a robotic system, the problem of abstraction can be in terms of control of arms and motors.
2. *Hardware and Software architecture:* Architectures should be well understood before a design.
3. *Extra functional Properties:* Extra functionalities required in the system being developed should be well understood from the design.
4. *System Related Family of designs:* Families of related systems developed earlier should be taken into consideration during designing.
5. *Modular Design:* Modular design concepts should be used. System designing is fast by decomposition of software into modules that are to be implemented. Modules should be such that they can be composed (coupled or integrated) later. Effective modular design should ensure effective (i) function independence, (ii) cohesion and (iii) coupling.
   (a) Modules should be clearly understood and should maintain continuity.
   (b) Also, appropriate protection strategies are necessary for each module. A module is not permitted to change or modify another module functionality. For example, protection from a device driver modifying the configuration of another device.
6. *Mapping:* Mapping into various representations is done from software requirements. For example, data flow in the same path during the program flow can be mapped together as a single entity. Transform and transaction mapping design processes are used in designing. For example, an image is input data to a system; it can have a different number of pixels and colours. The system does not process each pixel and colour individually. Transform mapping of image is done by appropriate compression and storage algorithms. Transaction mapping is done to define the sequence of images.
7. *User Interface Design:* User interface design is an important part of design. User interfaces are designed as per user requirements, analysis of the environment and system functions. For example, in an automatic chocolate vending machine (ACVM) system, the user interface is an LCD multiline graphics display. It can display a welcome message as well as specify the coins needed to be inserted into the machine for each type of chocolate. The same ACVM may be designed with touchscreen User Interface (GUI), or it may be designed with Voice User Interfaces (VUIs). Any of these interface designs has to be validated by the customer. For example, the ACVM customer who installs the machine must validate message language and messages to be displayed before an interface design can proceed to the implementation stage.
8. *Refinements:* Each component and module design needs to be refined iteratively till it becomes the most appropriate for implementation by the software team.

The software design process may require use of Architecture Description Language (ADL). It is used for :presenting the following: (i) Control Hierarchy (ii) Structural Partitioning (iii) Data Structure and Hierarchy v) Software Procedures.

Figure 1.11 shows the activities for software-design cycle during an embedded software-development process and the cycle may be repeated till tests show the verification of specifications.

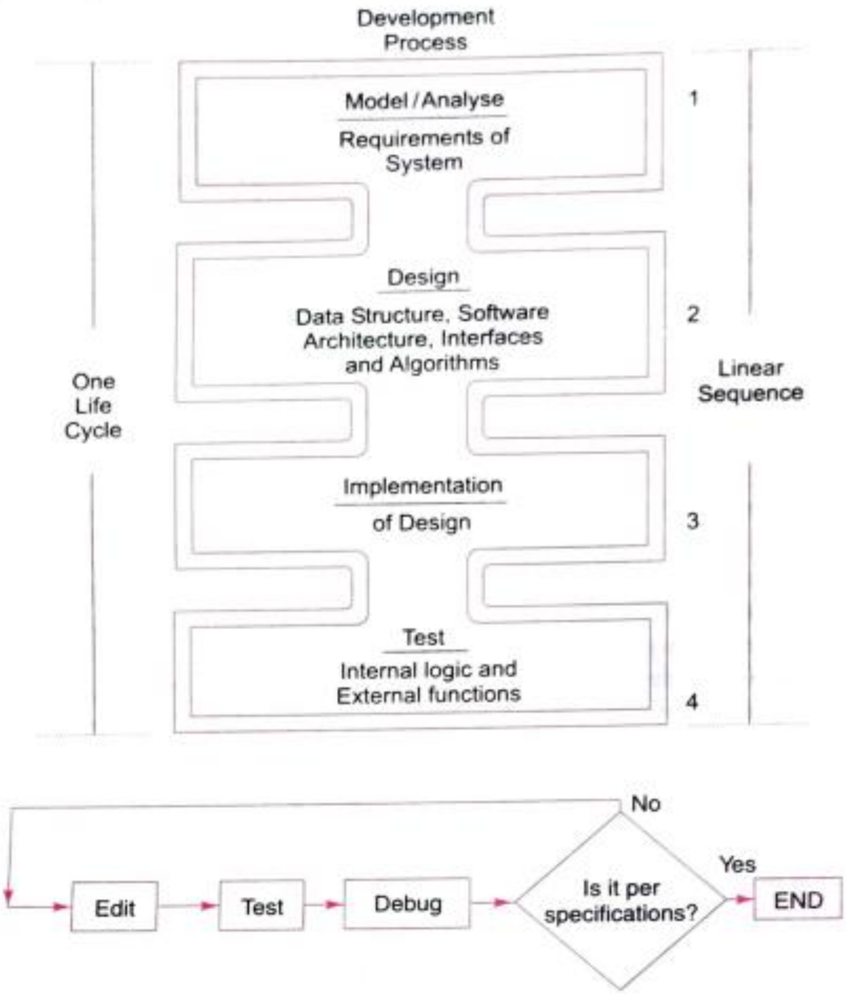

Fig. 1.11 Activities for software design during an embedded software-development process