# Lesson 02: **DESIGN PROCESSES FOR AN EMBEDDED-SYSTEM DESIGN**

# Design Process

- Process followed for designing an Embedded System
- Abstraction, Modeling, Hardware and Software Architecture, Extra functional Properties, System Related Family Designs
- Modules
- Mapping

# Abstraction

- Each problem component first abstracted
- For example, Display picture and text as an abstract class
- Robotic system problem abstraction in terms of control of motors in different degrees of freedoms
- Application software abstracted as concurrently running multiple threads and interrupt service threads

# Modeling

- Procedure Oriented
- Objected Oriented
- Sequential processes
- Concurrent processes
- State machine

# Object oriented Model

- Classes
- Objects
- Interfaces
- Package

# Design of Hardware and Software architecture

- Assumed to consists multiple layers
- Each architectural layer be well understood before a design

# Design for Extra functional Properties

- Extra functionalities required in the system being developed be well understood from the design

# System Related Family designs

- Families of related systems developed earlier taken into consideration during designing

# Modular Design

- 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.

# Modules

- Be clearly understood and maintain continuity.
- 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

# Mapping

- Mapping into various representations done considering the 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

- For example, an image is input data to a system; it can have a different number of pixels and colors of each pixel. The system has to store or process each pixel and color
- Transform mapping of image is done by appropriate compression and storage algorithms.
- Transaction mapping is done to define the sequence of the images

# User Interfaces Design

- Designed as per user requirements, analysis of the environment and system functions.
- For example, in an automatic chocolate vending machine system, the user interface is a LCD matrix 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. Same ACVM may be designed with touch screen GUI. Same ACVM may be designed with VUIs. A GUI or VUI or user interface or LCD matrix display

# Interface design validation

- Customer validation
- For example, the customer must validate message's language, screen logo, screen icons and background color, wall paper, menus and dialogs before an interface design can proceed to the implementation stage

# Testing, Verification and Validation

- Testing – to find errors and to validate that the implemented software is as per the specifications and requirements to get reliable product.
- Verification – refers to an activity to ensure that specific functions are correctly implemented.
- Validation – refers to an activity to ensure that the system that has been created is as per requirements agreed upon at the analysis phase, and to ensure its quality

# Modules

- Be clearly understood and maintain continuity.
- 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

# Refinements

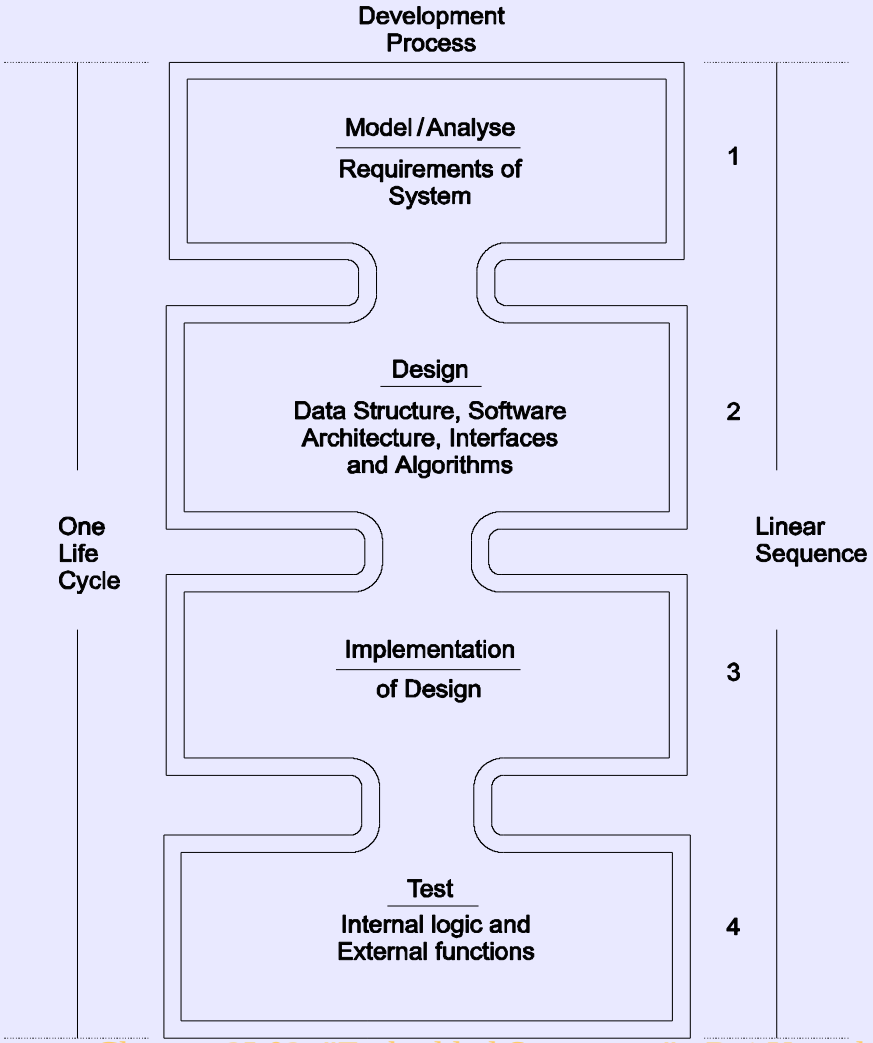
- Each component and module design needs to be refined iteratively till it becomes the most appropriate for implementation by the software team

# Abstraction of Design Process Steps

- A design process bottom-to-top design if it builds starting from the components.
- A design process top-to-down design if it first starts with abstraction of the process and then after abstraction the details are created. Top-to-down design approach is most favoured approach

# Software Design Cycle

# Activities for Software Design during Software-Development Process



# Five levels of abstraction from top level to bottom level in the design process

- Requirements
- Specifications
- Architecture
- Components
- System Integration

# Requirements

Complete clarity of

- required purpose,
- inputs,
- outputs,
- functioning,
- design metrics and
- Validation requirements for finally developed systems specifications.
- Consistency in the requirements

# Specifications

Clear specifications of

Customer expectations from the product.

Needs specifications for

- hardware, for example, peripherals, devices processor and memory specifications
- data types and processing specifications

# Needed specifications

- Expected system behavior specifications,
- constraints of design,
- expected life cycle specifications of the product.
- Process specifications analysed by making lists of inputs on events list, outputs on events, processes activated on each event.

# Architecture

- Data flow graphs
- Program models
- Software architecture layers and
- Hardware architecture
- Interfaces design
- System integration

# Software architectural layers

- How the different elements – data structures, databases, algorithms, control functions, state transition functions, process, data and program flow are to be organized
- What shall be design of data structures and databases that would be most appropriate for the given problem? Whether data organised as a tree- like structure will be appropriate? What will be the design of the components in the data?

# Hardware Components

- Processor, ASIP and single purpose processors in the system
- Memory RAM, ROM or internal and external flash or secondary memory in the system
- Peripherals and devices internal and external to the system
- Ports and buses in the system
- Power source or battery in the system

# Summary

We learnt

- Design processes, models,
- Modular and object oriented concepts
- Requirements
- Specifications
- Architecture
- Components
- System Integration

# End of Lesson 02