Professional Elective-II: Embedded Systems (PECCSE601B)

Module - V

Integration and testing of embedded hardware and firmware

Prepared by – Dr. Susmita Biswas
Associate Professor
University of Engineering and Management, Kolkata

Sub topic

- Integration of hardware and firmware
- Boards bring up

Integration of hardware and firmware

Integration of hardware and firmware refers to the process of **embedding firmware into the target hardware board**.

1. **Hardware**: This refers to the physical components of a device, such as microcontrollers, sensors, memory chips, and other electronic parts. Hardware provides the foundation for the device's functionality.

2. **Firmware**: Firmware is the **software** that runs directly on the hardware. It represents the **control algorithm** and **configuration data** necessary to implement the product requirements. Firmware resides in the memory of the embedded processor or controller.

Integration of hardware and firmware

Integration: **The integration process involves embedding the firmware into the target hardware board.** It's like adding intelligence to the product. The goal is to make the hardware and firmware work together seamlessly.

1. If the embedded processor/controller has built-in code memory, the firmware resides there.

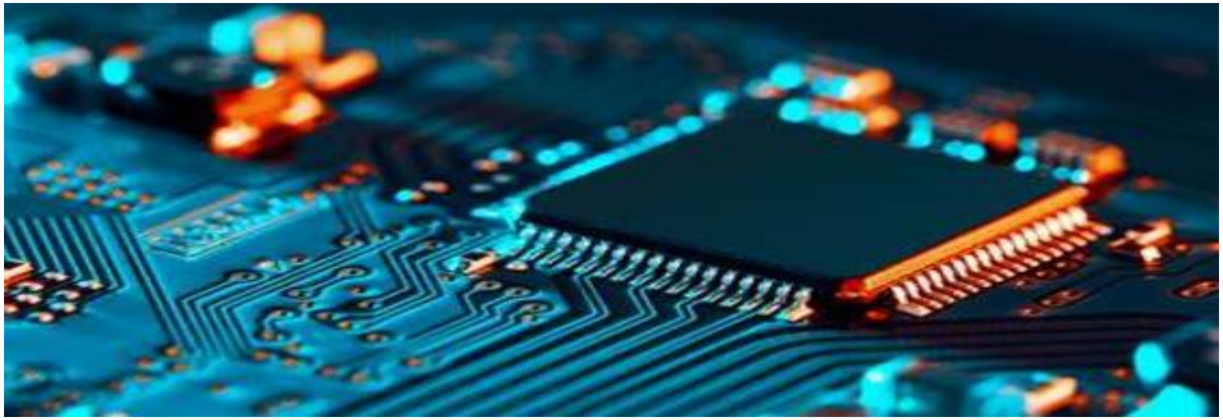
2. If the processor/controller lacks built-in memory or the firmware size exceeds its capacity, an **external dedicated EPROM/FLASH memory chip** is used to hold the firmware. This chip interfaces with the processor/controller.

Integration of hardware and firmware

Techniques for Firmware Integration:

1. Out of Circuit Programming:

- In this approach, the processor or memory chip that needs to be programmed is taken out of the target board.
- The programming is performed externally using a dedicated programming device.
- The programming device contains necessary hardware circuits to generate programming signals.
- A ZIF (Zero Insertion Force) socket with a locking pin holds the device to be programmed.
- Typically used for initial programming or updates during development.



Integration of hardware and firmware

Techniques for Firmware Integration:

2. **In System Programming (ISP)**:

- With ISP, the firmware is programmed directly on the target board without removing any components.
- The processor or memory chip remains in place during programming.
- Special pins or interfaces on the chip allow for in-circuit programming.
- Useful for field updates or production programming.

3. **In Application Programming (IAP)**:

- IAP involves updating the firmware while the system is running.
- The firmware can be modified or replaced without interrupting the normal operation of the embedded system.
- Commonly used in scenarios where continuous operation is critical.

Integration of hardware and firmware

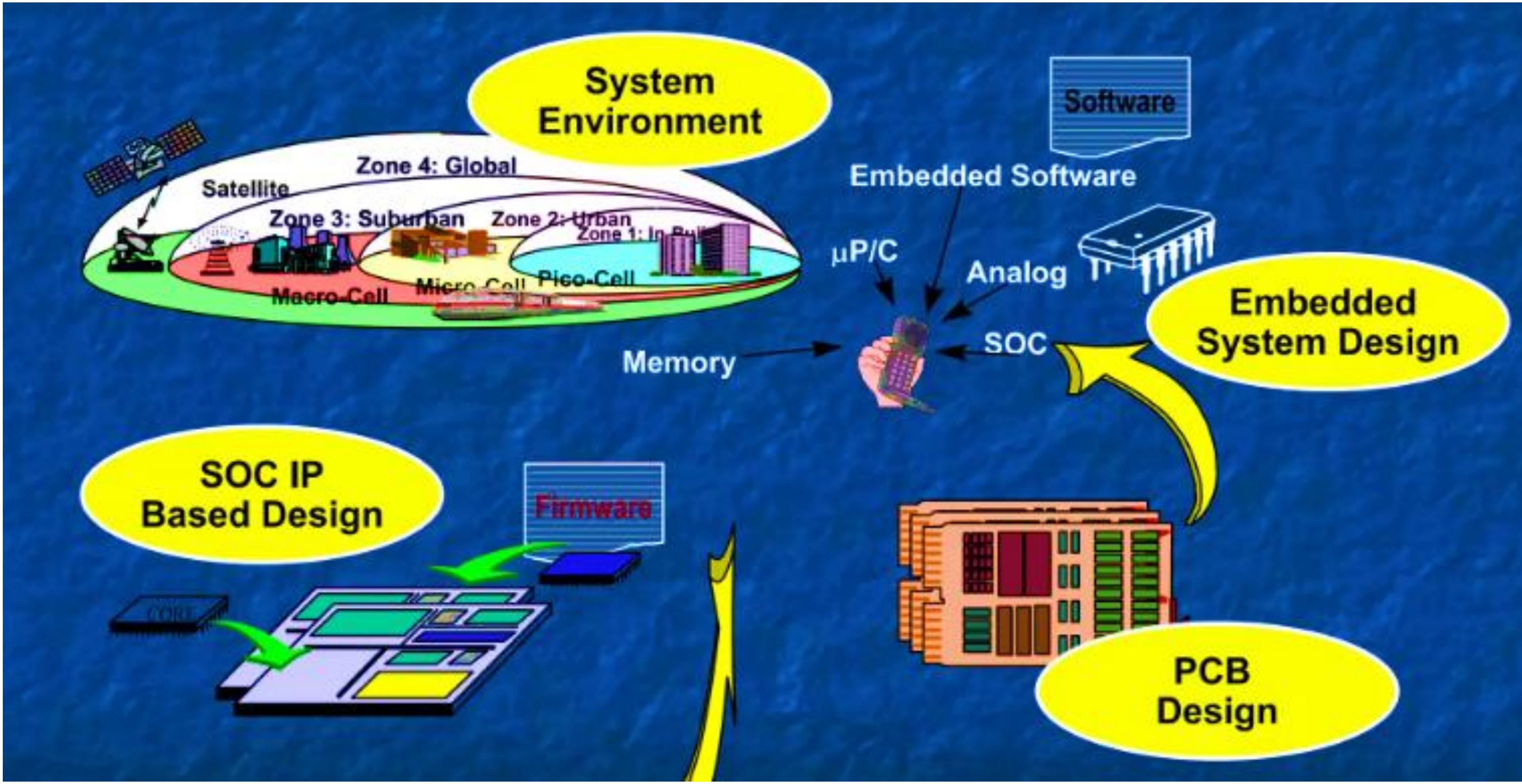
Techniques for Firmware Integration:

4. **Use of Factory Programmed Chip**:

- Some embedded systems use chips that come pre-programmed from the factory.
- These chips already contain the necessary firmware.
- The target board integrates these chips directly, eliminating the need for additional programming steps.

5. **Firmware Loading for Operating System Devices**: Relevant for devices with an operating system.

Integration of hardware and firmware



[Design of embedded systems | PPT (slideshare.net)](https://www.slideshare.net)

Issues while designing embedded system

- **Choosing right platform**
- **Memory and I/O requirements (Cache, Flash memory, Watchdog timer)**
- **Processors choice (Microcontroller, digital signal processor, ASIC, FPGA)**
- **Hardware software tradeoff**
- **Porting issues of operating system in the target board**

Design of embedded systems | PPT (slideshare.net)

Hardware-Software Partitioning

One of the most crucial steps in the design of embedded systems is hardware-software partitioning, i.e. deciding which components of the system are implemented in hardware and which ones in software.

Development of embedded systems

When developing an embedded system for your business app requirements, you might go overboard with budget and timelines. The development lifecycle approach can shorten timelines and improve your product quality.

1. Requirement Analysis:

It is essential to define the scope of work that needs to be done before you begin developing the system. Requirement analysis is a study of the existing systems, how they plan to modernise them, and some of the things they are considering. We will break down this phase into four deliverables to make it easier for the project manager.

a. Need

An individual or a business expresses the need for the system. The need is the system or the business owner’s new idea. It could be a new system, re-engineering an existing system, or maintenance. The client will prepare a proper document with the concept, current gaps, and process issues.

Development of embedded systems

b. Target Audience

Know who the customers of this product would be to define the system scope. You will be developing the scope and total activities for building the system. It is only essential to develop the persona of the customer.

c. Requirements

What are some of the requirements posed by the client for the system in consideration? This includes:

1.Operational attributes
2.Functional aspects
3.Product interface components
4.Layout and design preferences
5.Operational and maintenance needs
6.Generic system requirements

Development of embedded systems

d. Competitors

It is essential to study the competition and identify how they are faring better in this segment. Identify the gaps in the competitor’s system and what you can do to improve it. At the end of this stage, you should have:

- A cost-benefit analysis for the project
- A thorough analysis of the product attributes and functions
- Product scope and feasibility

Development of embedded systems

2. Designing:

Designing the system is a crucial phase of the embedded system development lifecycle. At this point, you are creating the layout and interface for the product. At the same time, you will be making the hardware design and defining the size.

a. Software Designing

Identify all the features and functionality you want to include in the software solution. It is essential to check how you will be enhancing the interface while keeping it accessible and usable. The trick is to work on a simple and intuitive design with the lowest learning curve.

b. Hardware Designing

The hardware should be simple and easy to debug whenever needed. It is essential to design the hardware for lightweight and smaller sizes. It will help you fit the devices into the circuits and improve operations. You need to consider plenty of things, such as the power, the connection, and communication with software at this stage.

Development of embedded systems

3. Development:

The development phase needs to be treated as the ultimate solution builder phase. **Work on translating the designs into a usable product with proper coding and suitable tools.** You should ensure the communication between the hardware and software is well-linked. After checking on the development complexities, you will choose the specialists for the projects. You will need

a. Embedded software engineer
b. Hardware engineer
c. PCB layout engineer
d. Mechanical engineer

Development of embedded systems

4. Implementation:

When the prototype model (after design and development) is ready, you can segment it into two:

1.Alpha Prototype: It is still in the raw product form, and there could be potential functional issues at this point

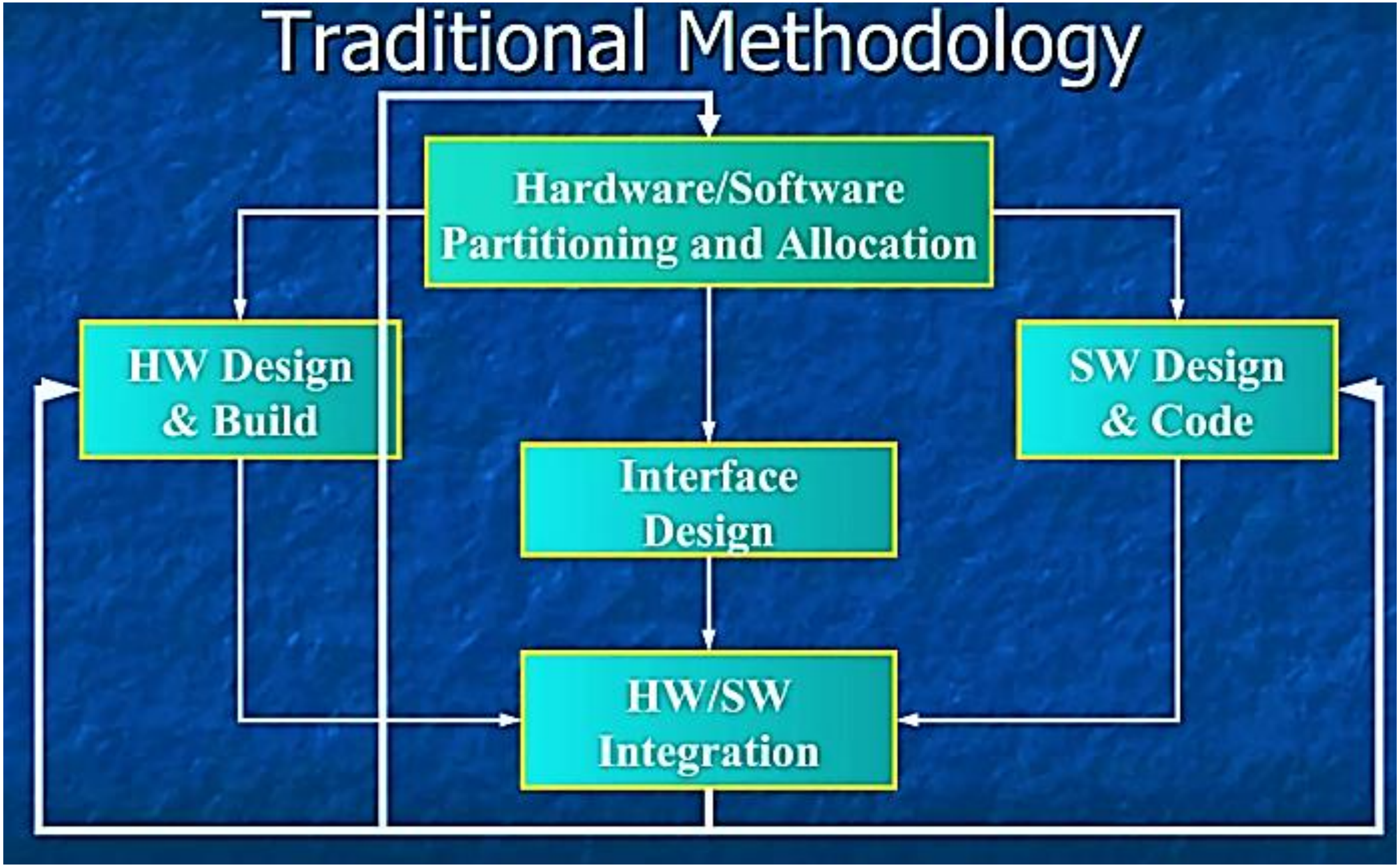
2.Beta Prototype: This is the prototype that you get after clearing all the issues and making the product ready for production

You launch the beta prototype to a closed group of customers to get their feedback and incorporate it into the product.

5. Commercial Launch:

Once your prototype is approved, your product is ready for the end-users. You will need to develop it for commercial use. At this point, you need to stock the necessary parts and raw materials to build a system that can communicate with your software.

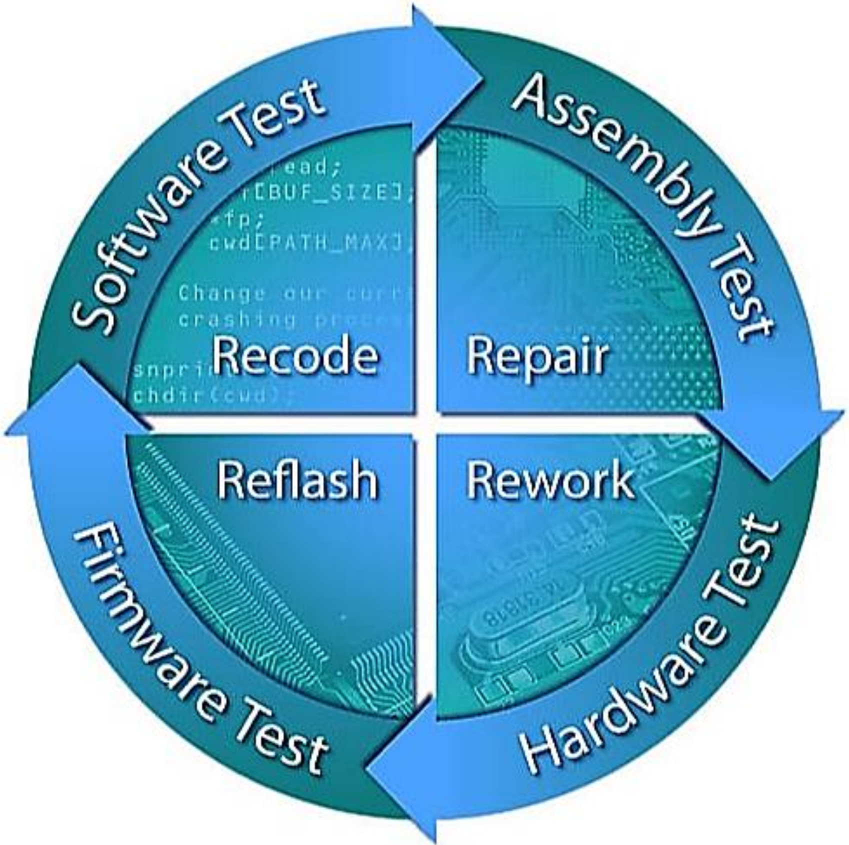
Designing of embedded system



[Design of embedded systems | PPT (slideshare.net)]

Boards bring up

Board bring-up is a **process of testing, validating and debugging an electronics system** that consists of assembly, hardware, firmware, and software elements. The process is done in phases and aims to achieve manufacturing readiness. Board bring-up activities include checking the board assembly, testing the hardware and firmware, evaluating memory and signal integrity, and loading OS and embedded software.


Software Test
Assembly Test
Hardware Test
Firmware Test
Recode
Repair
Reflash
Rework