Lesson 9: Challenges in Embedded System Design: Optimizing the Design Metrics and Formalism of System Design

Amount and type of hardware needed

- Optimizing the microprocessors, ASIPs and single purpose processors in the system
- Optimizing according to the performance, power dissipation, cost and other design metrics the system
- Optimizing hardware (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 and power source or battery in the system).

Taking into account the design metrics

- Design metrics examples –power dissipation, physical size, number of gates and engineering, prototype development and manufacturing costs.

Optimizing the Power Dissipation

- Clock Rate Reduction
- Operating Voltage Reduction
- Wait, Stop and Cache Disable Instructions – Clever real- time programming. It is by using of 'Wait' and 'Stop' instructions and disabling or controlling certain units when not needed is one method of saving power during program execution

Disable use of certain structural units of the processor to reduce power dissipation

- Caches—when not necessary and
- Keep in disconnected state those structure units that are not needed during a particular software-portion execution, for example, display screen, timers or IO units
- Control of power requirement, for example, by screen auto-brightness control

Process Deadlines

- Meeting the deadline of all processes in the system while keeping the memory, power dissipation, processor clock rate and cost at minimum is a challenge

Flexibility and Upgradeability

- Ability to offer the different versions of a product for marketing and offering the product in advanced versions later on.

Reliability

- Designing reliable product by appropriate design and thorough testing, verification and validation is a challenge.

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

Formalism of System Design

Formalism of Steps

- Requirements and Specifications of hardware and software
- Define architectures of hardware and software
- Coding and implementation as per architecture
- Testing, validation and verification of system

Diagrammatic model

- UML (Universal Modeling Language)

UML (Universal Modeling Language)

- Conceptual design modeling
- Classes and Objects to describe identity, attributes, components and behaviour
  - Inheritances in classes and objects
  - Interfaces with the objects and their implementation in the objects,
- Structural description of the design components

UML (*Contd.*)

- Behavioral description in terms of states, state machine and signals
- Events description

Disable use of certain structural units of the processor to reduce power dissipation

- Caches—when not necessary and
- Keep in disconnected state those structure units that are not needed during a particular software-portion execution, for example, display screen, timers or IO units
- Control of power requirement, for example, by screen auto-brightness control

Process Deadlines

- Meeting the deadline of all processes in the system while keeping the memory, power dissipation, processor clock rate and cost at minimum is a challenge

Flexibility and Upgradeability

- Ability to offer the different versions of a product for marketing and offering the product in advanced versions later on.

Reliability

- Designing reliable product by appropriate design and thorough testing, verification and validation is a challenge.

Testing, Verification and Validation

- Testing – to find errors and to validate that the implemented software is as per the specifications and requirements to get a 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 the 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

Summary

We learnt

- Challenges in system design
- Meeting design metrics
- Power dissipation reduction
- Enabling and controlling of processes and hardware units
- Flexibility, upgradeability and reliability

Summary

We learnt

- Formalism of System design Processes
- Use of UML for diagrammatically modeling the system design Processes

End of Lesson 9