Embedded Systems - Video course

1. Introduction to Embedded Computing

1.1 Introduction

1.2 Overview
- 1.2.1 Characteristics of Embedding Computing Applications
- 1.2.2 Concept of Real time Systems
- 1.2.3 Challenges in Embedded System Design

1.3 Design Process
- 1.3.1 Requirements
- 1.3.2 Specifications
- 1.3.3 Architecture Design
- 1.3.4 Designing of Components
- 1.3.5 System Integration

2. Embedded System Architecture

2.1 Instruction Set Architecture
- 2.1.1 CISC and RISC instruction set architecture

2.2 Basic Embedded Processor/Microcontroller Architecture
- 2.2.1 CISC Examples
  - 2.2.1.1 Motorola (68HC11) Example
  - 2.2.1.2 8051
- 2.2.2 RISC Example
  - 2.2.2.1 ARM
- 2.2.3 DSP Processors
- 2.2.4 Harvard Architecture
  - 2.2.4.1 PIC

2.3 Memory System Architecture
- 2.3.1 Caches
- 2.3.2 Virtual Memory
- 2.3.3 Memory Management Unit and Address Translation

2.4 I/0 Sub-system
- 2.4.1 Busy-wait I/0
- 2.4.2 DMA
- 2.4.3 Interrupt driven I/0

2.5 Co-processors and Hardware Accelerators

2.6 Processor Performance Enhancement
- 2.6.1 Pipelining
- 2.6.2 Super-scalar Execution

2.7 CPU Power Consumption
Lab Exercises on:
(i) Digital Circuit implementation
(ii) Hardware Description Language
(iii) Assembly language Programming for different target processors

3. Designing Embedded Computing Platform

3.1 Using CPU Bus
- 3.1.1 Bus Protocols
- 3.1.2 Bus Organisation

3.2 Memory Devices and their Characteristics
- 3.2.1 RAM
- 3.2.2 ROM, UVROM, EEPROM, Flash Memory
- 3.2.3 DRAM

3.3 I/O Devices
- 3.3.1 Timers and Counters
  - 3.3.1.1 Watchdog Timers

NP-TEL

NPTEL

http://nptel.ac.in

Electrical Engineering

Coordinators:

Prof. Santanu Chaudhary
Department of Electrical EngineeringIIT Delhi

3.3.2 Interrupt Controllers
3.3.3 DMA Controllers
3.3.4 A/D and D/A Converters
3.3.5 Displays
3.3.6 Keyboards
3.3.7 Infrared devices

3.4 Component Interfacing
3.4.1 Memory Interfacing
3.4.2 I/O Device Interfacing
3.4.2.1 Interfacing Protocols
3.4.2.1.1 GPIB
3.4.2.1.2 FIREWIRE
3.4.2.1.3 USB
3.4.2.1.4 IRDA

3.5 Designing with Processors
3.5.1 System Architecture
3.5.2 Hardware Design
3.5.2.1 FPGA Based Design

3.6 Implementation
3.6.1 Development Environment
3.6.2 Debugging Techniques
3.6.3 Manufacturing and Testing

3.7 Design Examples
3.7.1 Data Compressor
3.7.2 Alarm Clock
4. Programming Embedded Systems

4.1 Program Design
4.1.1 Design Patterns for Embedded Systems
4.1.2 Models of Program
4.1.2.1 Control and Data flow Graph

4.2 Programming Languages
4.2.1 Desired Language Characteristics
4.2.1.1 Introduction to Object Oriented Programming
4.2.1.2 Data Typing
4.2.1.2.1 Overloading and Polymorphism
4.2.1.3 Control
4.2.1.4 Multi-tasking and Task Scheduling
4.2.1.5 Timing Specifications
4.2.1.6 Run-time Exception handling
4.2.2 Use of High Level Languages
4.2.2.1 C for Programming embedded systems
4.2.2.2 Object Oriented Programming for Embedded Systems in C++
4.2.2.3 Use of Java for Embedded Systems
4.2.3 Programming and Run-time Environment
4.2.3.1 Compiling, Assembling, Linking
4.2.3.2 Debugging
4.2.4 Basic Compilation Techniques
4.2.5 Analysis and Optimization of Execution Time
4.2.6 Analysis and Optimization of Energy and Power
4.2.7 Analysis and Optimization of Program Size
4.2.8 Program Validation and Testing
5. Operating System

5.1 Basic Features of an Operating System

5.2 Kernel Features
5.2.1 Real-time Kernels
5.2.1.1 Polled Loops System
5.2.1.2 Co-routines
5.2.1.3 Interrupt-driven System
5.2.1.4 Multi-rate System

5.3 Processes and Threads

5.4 Context Switching
5.4.1 Cooperative Multi-tasking
5.4.2 Pre-emptive Multi-tasking

5.5 Scheduling
5.5.1 Rate-Monotonic Scheduling
5.5.2 Earliest-Deadline First Scheduling

5.5.3 Task Assignment
5.5.4 Fault-Tolerant Scheduling

5.6 Inter-process Communication
5.6.1 Signals
5.6.2 Shared Memory Communication
5.6.3 Message-Based Communication

5.7 Real-time Memory Management
5.7.1 Process Stack Management
5.7.2 Dynamic Allocation

5.8 I/O
5.8.1 Synchronous and Asynchronous I/O
5.8.2 Interrupt Handling
5.8.3 Device Drivers
5.8.4 Real-time Transactions and Files

5.9 Example Real-time OS
5.9.1 VxWorks
5.9.2 RT-Linux
5.9.3 Psos

5.10 Evaluating and Optimising Operating System Performance
5.10.1 Response-time Calculation
5.10.2 Interrupt latency
5.10.3 Time-loading
5.10.4 Memory Loading

5.11 Power Optimisation Strategies for Processes
6. Network Based Embedded Applications

6.1 Network Fundamentals

6.2 Layers and Protocols
6.2.1 Network Architectures
6.2.2 Network Components: Bridges, Routers, Switches

6.3 Distributed Embedded Architectures

6.4 Elements of Protocol Design

6.5 High Level Protocol Design Languages

6.6 Network Based Design

6.7 Internet-Enabled Systems
6.7.1 Protocols for industrial and control applications
6.7.2 Internetworking Protocols

6.8 Wireless Applications
6.8.1 Blue-tooth
7. Embedded Control Applications

7.1 Introduction

7.2 Open-loop and Closed Loop Control Systems
7.2.1 Examples: Speed Control

7.3 PID Controllers
7.3.1 Software Coding of a PID Controller
7.3.2 PID tuning

7.4 Fuzzy Logic Controller

7.5 Application Examples
7.5.1 Washing Machine
7.5.2 Automotive Systems
7.5.3 Auto-focusing digital camera
7.5.4 Air-conditioner
8. Embedded System Development

8.1 Design Methodologies
8.1.1 UML as Design tool
8.1.2 UML notation
8.1.3 Requirement Analysis and Use case Modeling
8.1.4 Static Modeling