

Module

1

Introduction

Lesson 2

Introduction to Real Time Embedded Systems Part II

Structure and Design

Instructional Objectives

After going through this lesson the student will

- Learn more about the numerous day-to-day real time embedded systems
- Learn the internal hardware of a typical mobile phone
- Learn about the important components of an RTES
- Learn more about a mobile phone
- Learn about the various important design issues
- Also learn the design flow

Pre-Requisite

Digital Electronics, Microprocessors

Common Examples Of Embedded Systems

Some of the common examples of Embedded Systems are given below:

Consumer electronics cell phones, pagers, digital cameras, camcorders, DVD players, portable video games, calculators, and personal digital assistants etc.



Fig. 2.1(a) Digital Camera



Fig. 2.1(b) Camcorder



Fig. 2.1(c) Personal Digital Assistants

Home appliances microwave ovens, answering machines, thermostats, home security systems, washing machines, and lighting systems etc.



Fig. 2.1(d) Microwave Oven



Fig. 2.1(e) Washer and Dryers

office automation fax machines, copiers, printers, and scanners



Fig. 2.1(f) Fax cum printer cum copier

business equipment electronic cash registers, curbside check-in, alarm systems, card readers product scanners, and automated teller machines



Fig. 2.1(g) Electronic Cash Registers



Fig. 2.1(h) Electronic Card Readers



Fig. 2.1(i) Automated Teller Machines

automobiles Electronic Control Unit(ECU) which includes transmission control, cruise control, fuel injection, antilock brakes, and active suspension in the same or separate modules.



Fig. 2.1(j)ECU of a Vehicle

Mobile Phone

Let us take the same mobile phone as discussed in Lesson 1 as example for illustrating the typical architecture of RTES.

In general, a cell phone is composed of the following components:

- A Circuit board (Fig. 2.2)
- Antenna
- Microphone
- Speaker
- Liquid crystal display (LCD)
- Keyboard
- Battery

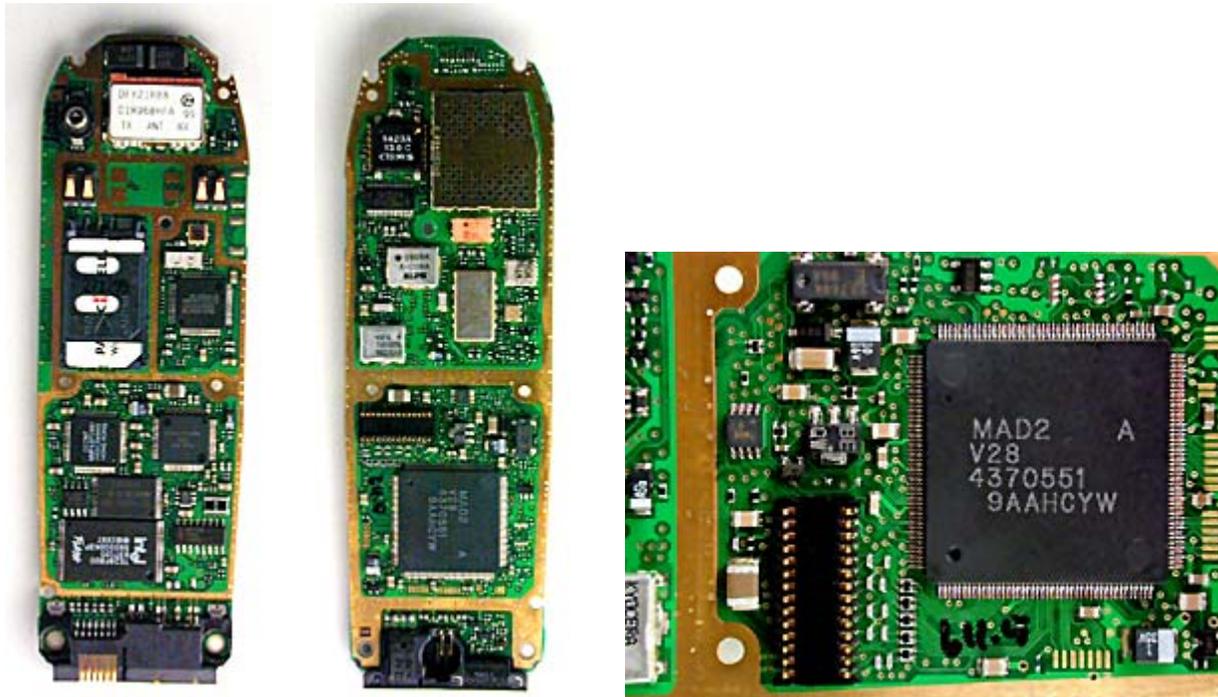


Fig. 2.2 The Cell Phone Circuitry

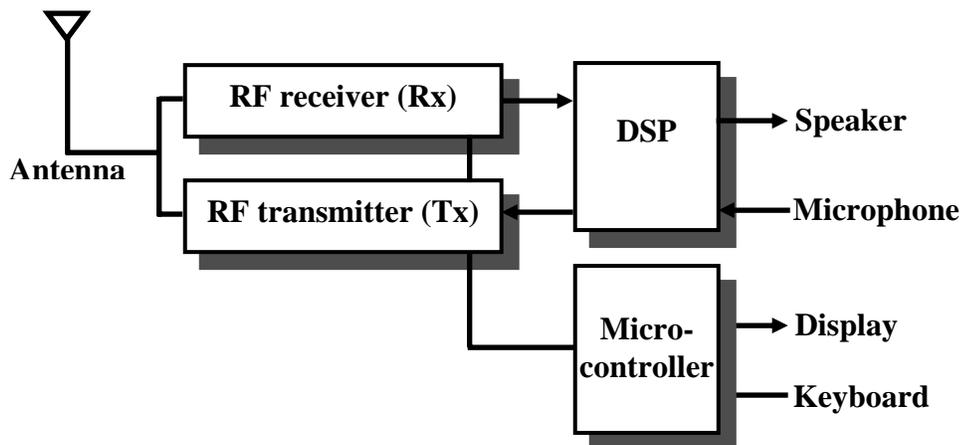


Fig. 2.3 The block diagram

A typical mobile phone handset (Fig. 2.3) should include standard I/O devices (keyboard, LCD), plus a microphone, speaker and antenna for wireless communication. The Digital Signal Processor (DSP) performs the signal processing, and the micro-controller controls the user interface, battery management, call setup etc. The performance specification of the DSP is very crucial since the conversion has to take place in real time. This is why almost all cell phones contain such a special processor dedicated for making digital-to-analog (DA) and analog-to-digital(AD) conversions and real time processing such as modulation and demodulation etc. The Read Only Memory (ROM) and flash memory (Electrically Erasable and Programmable Memory) chips provide storage for the phone's operating system(RTOS) and various data such as phone numbers, calendars information, games etc.

Components of an Embedded System

By this time we know where are our Embedded Systems and what makes them stand out from other systems like Calculators, Desktop Computers, and our Old Television Sets. We have also developed some 6th sense to guess the components of an RTES.

1. Microprocessor

This is the heart of any RTES. The microprocessors used here are different from the general purpose microprocessors like Pentium Sun SPARC etc. They are designed to meet some specific requirements. For example Intel 8048 is a special purpose microprocessor which you will find in the Keyboards of your Desktop computer. It is used to scan the keystrokes and send them in a synchronous manner to your PC. Similarly mobile phones Digital Cameras use special purpose processors for voice and image processing. A washer and dryer may use some other type of processor for *Real Time Control and Instrumentation*.

2. Memory

The microprocessor and memory must co-exist on the same Power Circuit Board(PCB) or same chip. *Compactness, speed and low power* consumption are the characteristics required for the memory to be used in an RTES. Therefore, very low power semiconductor memories are used in almost all such devices. For housing the operating system Read Only Memory(ROM) is used. The program or data loaded might exist for considerable duration. It is like changing the setup of your Desktop Computer. Similar user defined setups exist in RTES. For example you may like to change the ring tone of your mobile and keep it for some time. You may like to change the screen color etc. In these cases the memory should be capable of retaining the information even after the power is removed. In other words the memory should be non-volatile and should be easily programmable too. It is achieved by using Flash¹ memories.

3. Input Output Devices and Interfaces

Input/Output interfaces are necessary to make the RTES interact with the external world. They could be Visual Display Units such as TFT screens in a mobile phone, touch pad key board, antenna, microphones, speakers etc. These RTES should also have open interfaces to other devices such as Desktop Computers, Local Area Networks (LAN) and other RTES. For example you may like to download your address book into your personal digital assistant (PDA). Or you may like to download some mp3 songs from your favorite internet site into your mp3 player. These input/output devices along with standard software protocols in the RTOS provide the necessary interface to these standards.

¹ A memory technology similar in characteristics to EPROM(Erasable Programmable Read Only Memory) memory, with the exception that erasing is performed electrically instead of via ultraviolet light, and, depending upon the organization of the flash memory device, erasing may be accomplished in blocks (typically 64k bytes at a time) instead of the entire device.

4. Software

The RTES is the just the physical body as long as it is not programmed. It is like the human body without life. Whenever you switch on your mobile telephone you might have marked some activities on the screen. Whenever you move from one city to the other you might have noticed the changes on your screen. Or when you are gone for a picnic away from your city you might have marked the no-signal sign. These activities are taken care of by the Real Time Operating System sitting on the non-volatile memory of the RTES.

Besides the above an RTES may have various other components and Application Specific Integrated Circuits (ASIC) for specialized functions such as motor control, modulation, demodulation, CODEC.

The design of a Real Time Embedded System has a number of constraints. The following section discusses these issues.

Design Issues

The constraints in the embedded systems design are imposed by external as well as internal specifications. Design metrics are introduced to measure the cost function taking into account the technical as well as economic considerations.

Design Metrics

A Design Metric is a measurable feature of the system's performance, cost, time for implementation and safety etc. Most of these are conflicting requirements i.e. optimizing one shall not optimize the other: e.g. a cheaper processor may have a lousy performance as far as speed and throughput is concerned.

Following metrics are generally taken into account while designing embedded systems

NRE cost (nonrecurring engineering cost)

It is one-time cost of designing the system. Once the system is designed, any number of units can be manufactured without incurring any additional design cost; hence the term nonrecurring.

Suppose three technologies are available for use in a particular product. Assume that implementing the product using technology 'A' would result in an NRE cost of \$2,000 and unit cost of \$100, that technology B would have an NRE cost of \$30,000 and unit cost of \$30, and that technology C would have an NRE cost of \$100,000 and unit cost of \$2. Ignoring all other design metrics, like time-to-market, the best technology choice will depend on the number of units we plan to produce.

Unit cost

The monetary cost of manufacturing each copy of the system, excluding NRE cost.

Size

The physical space required by the system, often measured in bytes for software, and gates or transistors for hardware.

Performance

The execution time of the system

Power Consumption

It is the amount of power consumed by the system, which may determine the lifetime of a battery, or the cooling requirements of the IC, since more power means more heat.

Flexibility

The ability to change the functionality of the system without incurring heavy NRE cost. Software is typically considered very flexible.

Time-to-prototype

The time needed to build a working version of the system, which may be bigger or more expensive than the final system implementation, but it can be used to verify the system's usefulness and correctness and to refine the system's functionality.

Time-to-market

The time required to develop a system to the point that it can be released and sold to customers. The main contributors are design time, manufacturing time, and testing time. This metric has become especially demanding in recent years. Introducing an embedded system to the marketplace early can make a big difference in the system's profitability.

Maintainability

It is the ability to modify the system after its initial release, especially by designers who did not originally design the system.

Correctness

This is the measure of the confidence that we have implemented the system's functionality correctly. We can check the functionality throughout the process of designing the system, and we can insert test circuitry to check that manufacturing was correct.

The Performance Design Metric

Performance of a system is a measure of how long the system takes to execute our desired tasks.

The two main measures of performance are:

Latency or response time

This is the time between the start of the task's execution and the end. For example, processing an image may take 0.25 second.

Throughput

This is the number of tasks that can be processed per unit time. For example, a camera may be able to process 4 images per second

These are the some of the cost measures for developing an RTES. Optimization of the overall cost of design includes each of these factors taken with some multiplying factors depending on their importance. And the importance of each of these factors depends on the type of application. For instance in defense related applications while designing an anti-ballistic system the execution time is the deciding factor. On the other hand, for de-noising a photograph in an embedded camera in your mobile handset the execution time may be little relaxed if it can bring down the cost and complexity of the embedded Digital Signal Processor.

The design flow of an RTES involves several steps. The cost and performance is tuned and fine-tuned in a recursive manner. An overall design methodology is enumerated below.

Design Methodology (Fig. 2.4)

System Requirement and Specifications

Define the problem

What your embedded system is required to do?

Define the requirements (inputs, outputs, control)

What are the inputs and outputs of your system?

Write down the specifications for them

Specify if the signals are in digital or analogue form. Specify the voltage levels, frequency etc.

The design task can be further segregated into the following steps

System level Design

Find out the possible subsystems of the system and the interconnections between them.

Sub-system or Node Level design

Each of these subsystems can be termed as the nodes. Elaborate on each of these subsystems and further make the block diagram and component level interconnections.

Processor Level Design

Each subsystem may consist of processor, memory, I/O devices. Specification and design at this level is required now.

Task Level Design

Complete interconnection of these subsystems depending on the tasks they would perform.

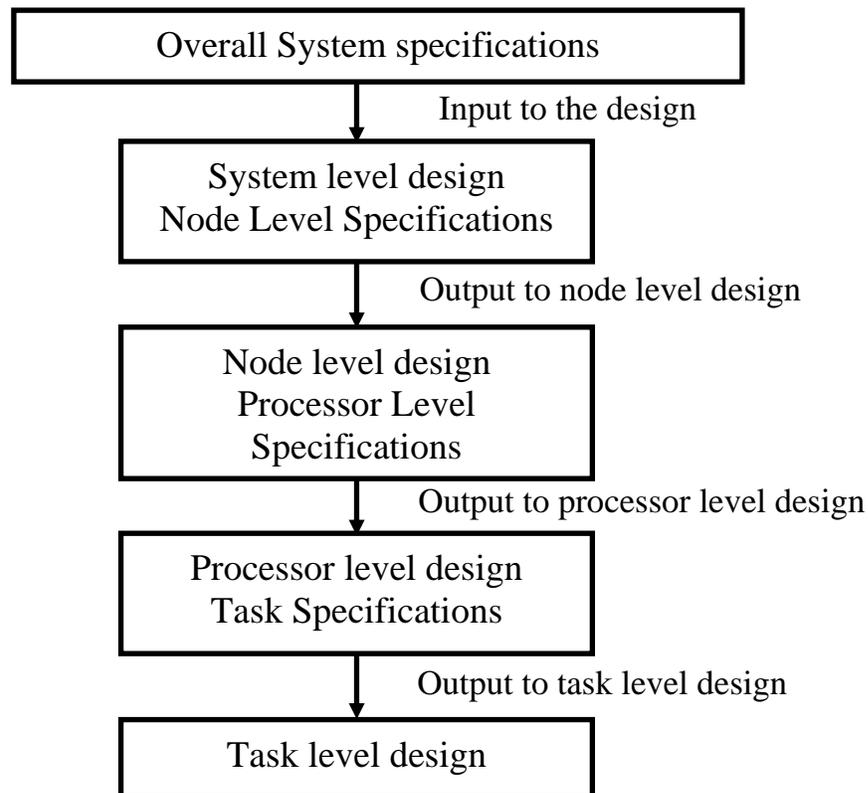


Fig. 2.4 The design approach

Conclusion

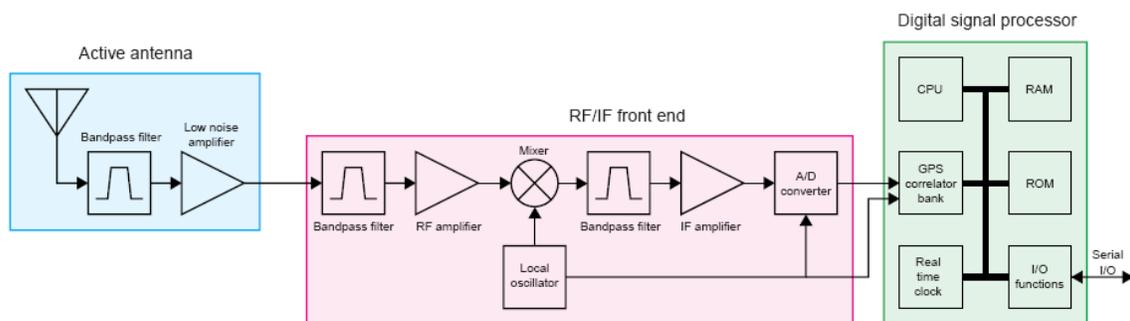
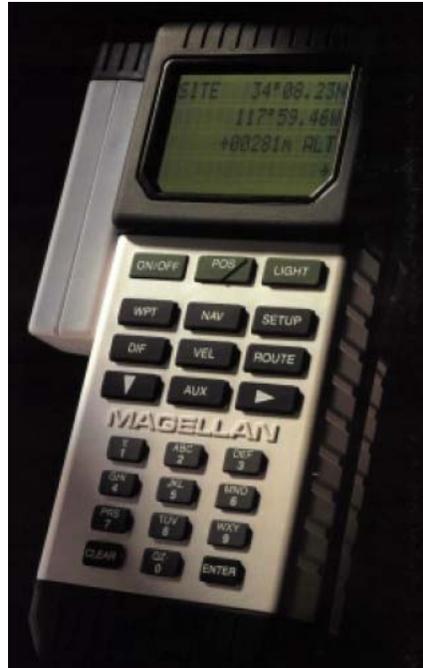
- The scope of embedded systems has been encompassing more and more diverse disciplines of technology day by day. Obsolescence of technology occurs at a much faster pace as compared to the same in other areas. The development of Ultra-Low-Power VLSI mixed signal technology is the prime factor in the miniaturization and enhancement of the performance of the existing systems. More and more systems are tending to be compact and portable with the RTES technology. The future course of embedded systems depends on the advancements of sensor technology, mechatronics and battery technology.
- The design of these RTES by and large is application specific. The time-gap between the conception of the design problem and marketing has been the key factor for the industry.
- Most of the cases for very specific applications the system needs to be developed using the available processors rather than going for a custom design.

Questions

Q1. Give one example of a typical embedded system other than listed in this lecture. Draw the block diagram and discuss the function of the various blocks. What type of embedded processor they use?

Ans:

Example 1: A handheld Global Positioning System Receiver



For details please <http://www.gpsworld.com/>

A GPS receiver receives signals from a constellation of at least four out of a total of 24 satellites. Based on the timing and other information signals sent by these satellites the digital signal processor calculates the position using triangulation.

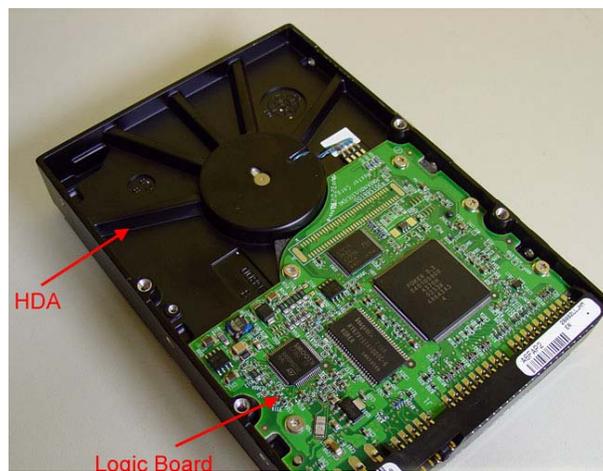
The major block diagram is divided into (1) Active Antenna System (2)RF/IF front end (3) The Digital Signal Processor(DSP)

The Active Antenna System houses the antenna a band pass filter and a low noise amplifier (LNA)

The RF/IF front end houses another band pass filter, the RF amplifier and the demodulator and A/D converter.

The DSP accepts the digital data and decodes the signal to retrieve the information sent by the GPS satellites.

Q2. Discuss about the Hard Disk Drive housed in your PC. Is it an RTES?



Ans:

Hard drives have two kinds of components: internal and external. External components are located on a printed circuit board called logic board while internal components are located in a sealed chamber called HDA or Hard Drive Assembly.

For details browse <http://www.hardwaresecrets.com/article/177/3>

The big circuit is the controller. It is in charge of everything: exchanging data between the hard drive and the computer, controlling the motors on the hard drive, commanding the heads to read or write data, etc.

All these tasks are carried out as demanded by the processor sitting on the motherboard. It can be verified to be single-functioned, tightly constrained,

Therefore one can say that a Hard Disk Drive is an RTES.

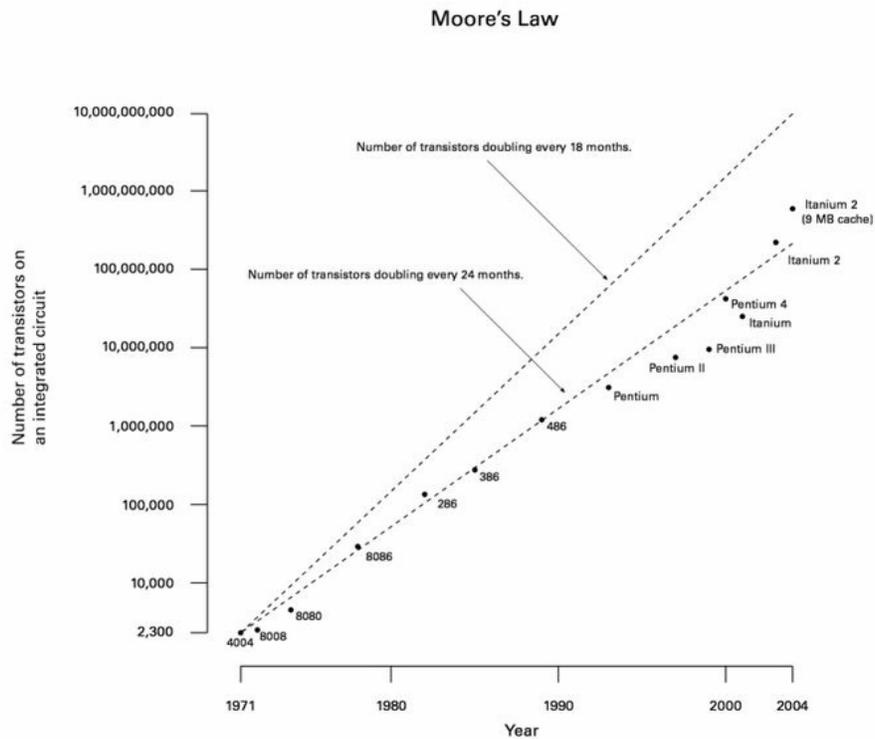
Q3. Elaborate on the time-to-market design metric.

Ans:

The time required to develop a system to the point that it can be released and sold to customers. The main contributors are design time, manufacturing time, and testing time. This metric has become especially demanding in recent years. Introducing an embedded system to the marketplace early can make a big difference in the system's profitability.

Q4. What is Moore's Law? How was it conceived?

Moore's law is the empirical observation that the complexity of integrated circuits, with respect to minimum component cost, doubles every 24 months. It is attributed to Gordon E. Moore, a co-founder of Intel.



References and Further Reading

- [1] Richard Bohuslav Kosik , “Digital ignition & Electronic fuel injection” Department of Computer Science and Electrical Engineering The University of Queensland, Australia, Bachelor’s Thesis, October 2000
- [2] Frank Vahid, Tony Givargis, “Embedded System Design, A Unified Hardware/Software Introduction”,John Wiley and Sons Inc, 2002
- [3] Wayne Wolf, “ Computers as Components”, Morgan Kaufmann, Harcourt India,2001
- [4] A.M Fox, J.E. Cooling, N.S. Cooling, “Integrated Design approach for real time embedded systems”, Proc. IEE-Softw., Vo.146, No.2., April 1999, page 75-85.
- [5] Phen Edwards, Luciano Lavagno, Dward A. Lee.Alberto Sangiovanni- Vincentelli , “Design of Embedded Systems: Formal Models, Validation, and Synthesis”, PROCEEDINGS OF THE IEEE, VOL. 85, NO. 3, MARCH 1997, page-366-390
- [6] J.A. Debardeleben, V. K. Madiseti, A. J. Gadeint, “Incorporating Cost Modeling in Embedded-System Design”, IEEE Design and Test of Computers, July-September-1997, Page 24-35