Dr. Surendra Shrestha
surendra@ioe.edu.np, surendtha@gmail.com

Education:

- **Post Doc. (Graphene Tech.), University Polytechnica de Madrid , Spain**
- **PhD (Major: Nanoscience)**, Sun Moon University, S. Korea
- **M.Sc. Engg.,** Tashkent Electro-Technical Institute of Communication, Uzbekistan

PROFESSIONAL EXPERIENCE:

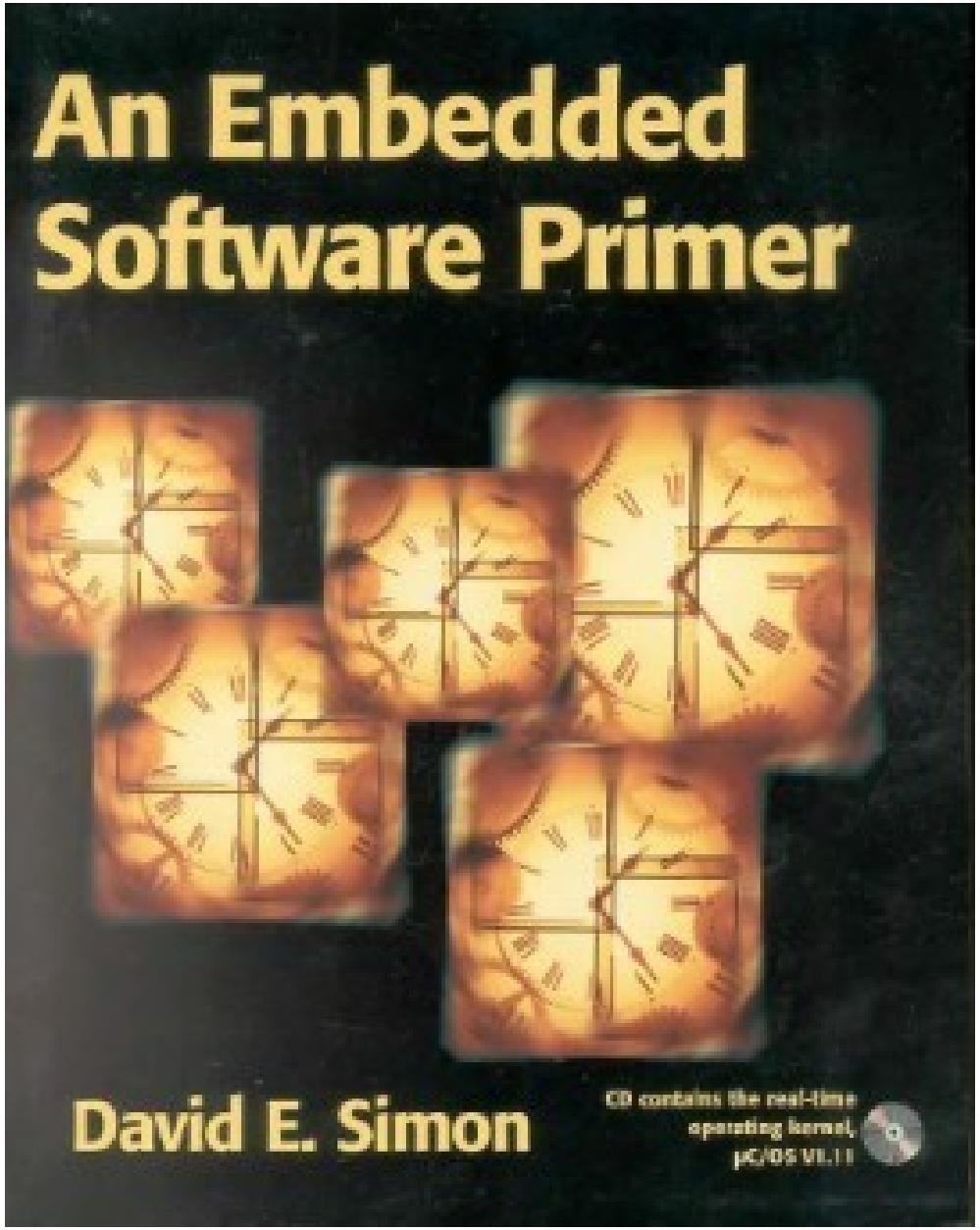
-**Associate Professor,** Department of Electronics and Computer Engineering, Pulchowk Campus, Institute of Engineering, Pulchowk, Lalitpur, Nepal

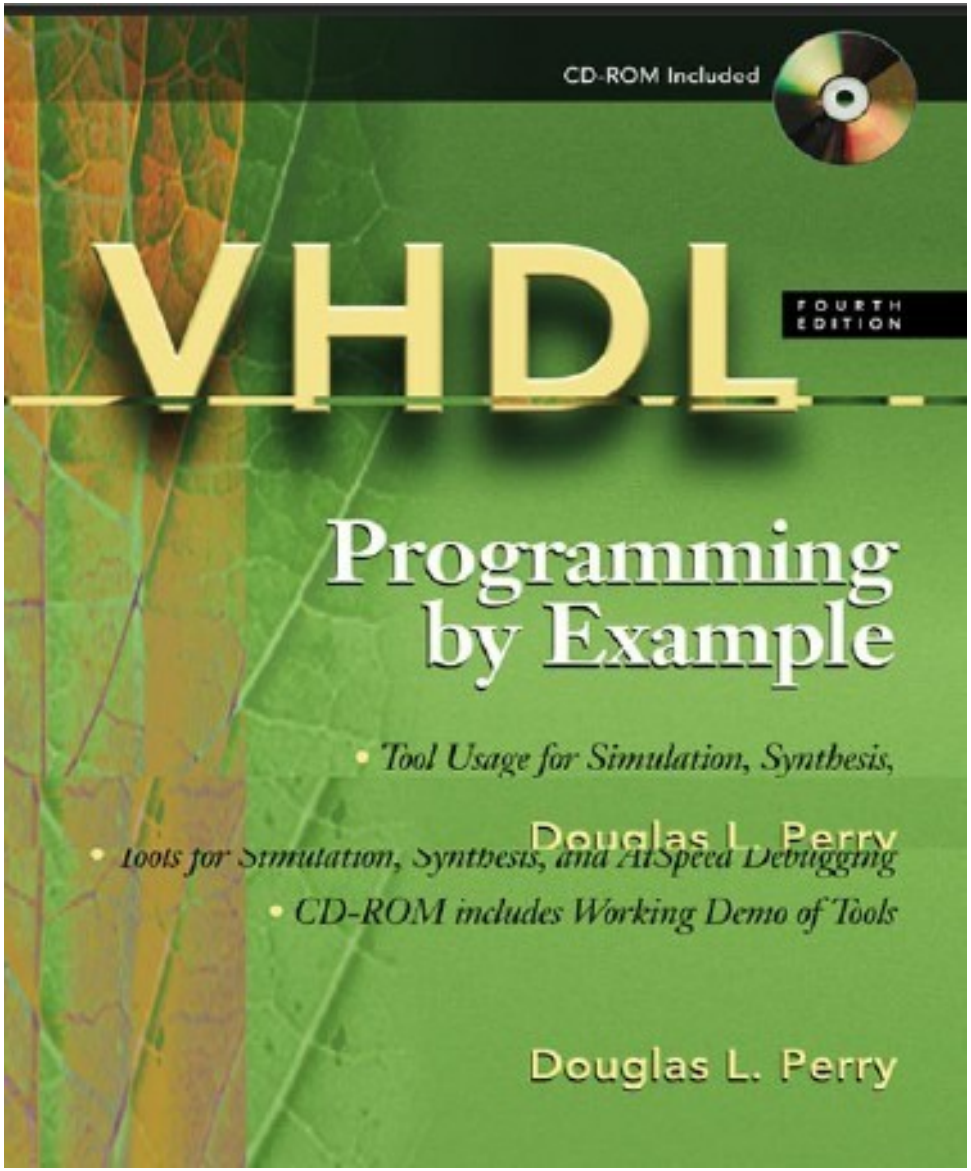
-**Program Coordinator**, M.Sc. In ICE, Department of Electronics and Computer Engineering, Pulchowk Campus, Institute of Engineering, Pulchowk, Lalitpur, Nepal

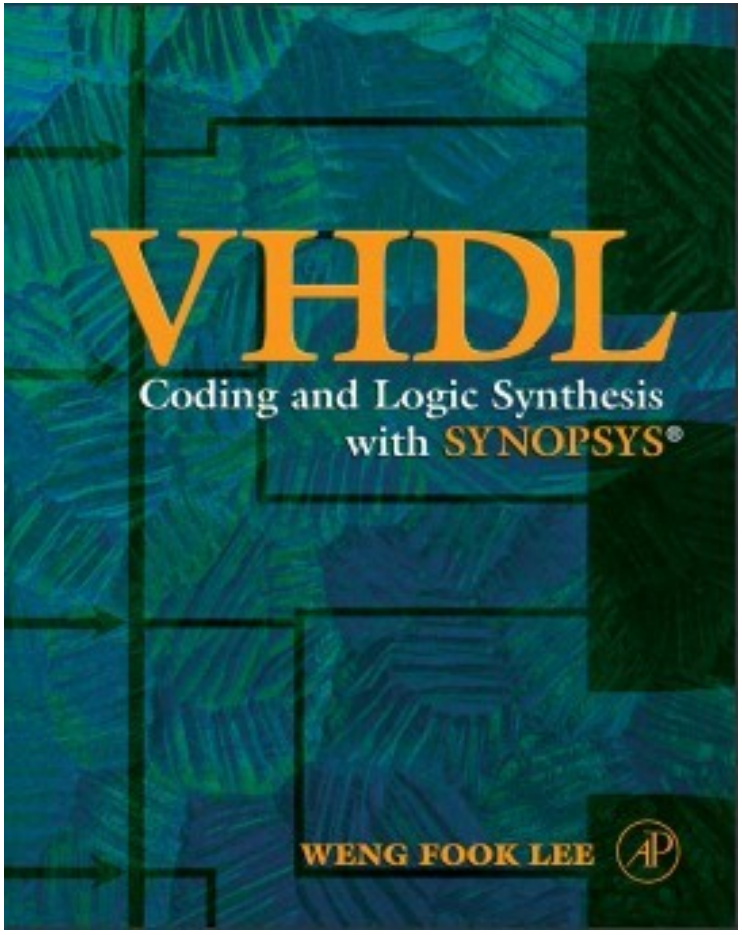
EMBEDDED SYSTEMS

Reference Books:

- David E. Simon, “An Embedded Software Primer”, Addison-Wesley, 2005
- Muhammad Ali Mazidi, “8051 Microcontroller and Embedded Systems”, Prentice Hall, 2006
- Frank Vahid, Tony Givargis, “Embedded System Design”, John Wiley & Sons, 2008
- Douglas L. Perry, “VHDL Programming by example”, McGraw Hill, 2002
- Shibu K V, “Introduction to EMBEDDED SYSTEMS”, McGrawHill, 2009


An Embedded Software Primer
David E. Simon
CD contains the real-time operating kernel, µC/OS VI.11


CD-ROM Included
VHDL
FOURTH EDITION
Programming by Example
• Tool Usage for Simulation, Synthesis,
• Tools for Simulation, Synthesis, and AtSpeed Debugging
• CD-ROM includes Working Demo of Tools
Douglas L. Perry
Douglas L. Perry


VHDL
Coding and Logic Synthesis with SYNOPSYS®
WENG FOOK LEE
AP

Unit	Hour	Numbers of question	Mark Distribution
1	3		4
2	4		8
3	6		8
4	5		8
5	6		8
6	8		12
7	3		8
8	3		8
9	3		8
10	4		8
Total	**45**	**10**	**80**

1. Introduction to Embedded System [3 Hrs]

1.1 Embedded Systems overview
1.2 Classification of Embedded Systems
1.3 Hardware and Software in a system
1.4 Purpose and Application of Embedded Systems

1.1 Embedded Systems overview

An Embedded System

is an electronic/electromechanical system designed to perform a specific function and is a combination of both hardware and firmware (software).

is a system built to perform its duty, completely or partially independent of human intervention.

is specially designed to perform a few tasks in the most efficient way.

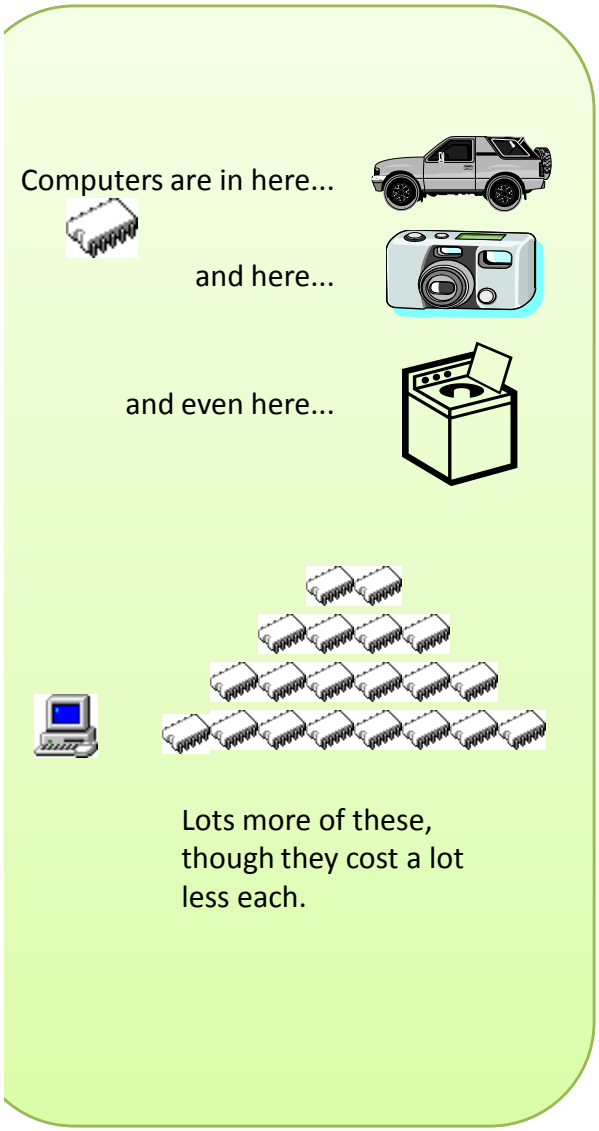
Interacts with physical elements in our environment, controlling and driving a motor, sensing temperature, …

Embedded systems overview

- Computing systems are everywhere
- Most of us think of “desktop” computers
  - PC’s 
  - Laptops
  - Mainframes 
  - Servers
- But there’s another type of computing system
  - Far more common...

Embedded systems overview

- **Embedded computing systems**
  - **Computing systems embedded within electronic devices**
  - **Hard to define. Nearly any computing system other than a desktop computer**
  - **Billions of units produced yearly, versus millions of desktop units**



General Purpose Computing System	Embedded System
• A system which is a combination of a generic hardware and a General Purpose Operating System for executing a variety of application • Contains a General Purpose Operating System (GPOS) • Applications are alterable (programmable) by the user (It is possible for the end user to re-install the operating system, and also add or remove user application)	• A system which is a combination of special purpose hardware and embedded OS for executing a specific set of application • May or may not contain an operating system for functioning • The firmware of the embedded system is pre- programmed and it is non- alterable by the end-user (there may be exceptions for system supporting OS kernel image flashing through special hardware settings)

General Purpose Computing System …

- Performance is the key deciding factor in the selection of the system. Always, ‘Faster is Better’
- Less/not at all tailored towards reduced operating power requirements, options for different levels of power management.
- Response requirements are not time-critical
- Need not be deterministic in execution behavior

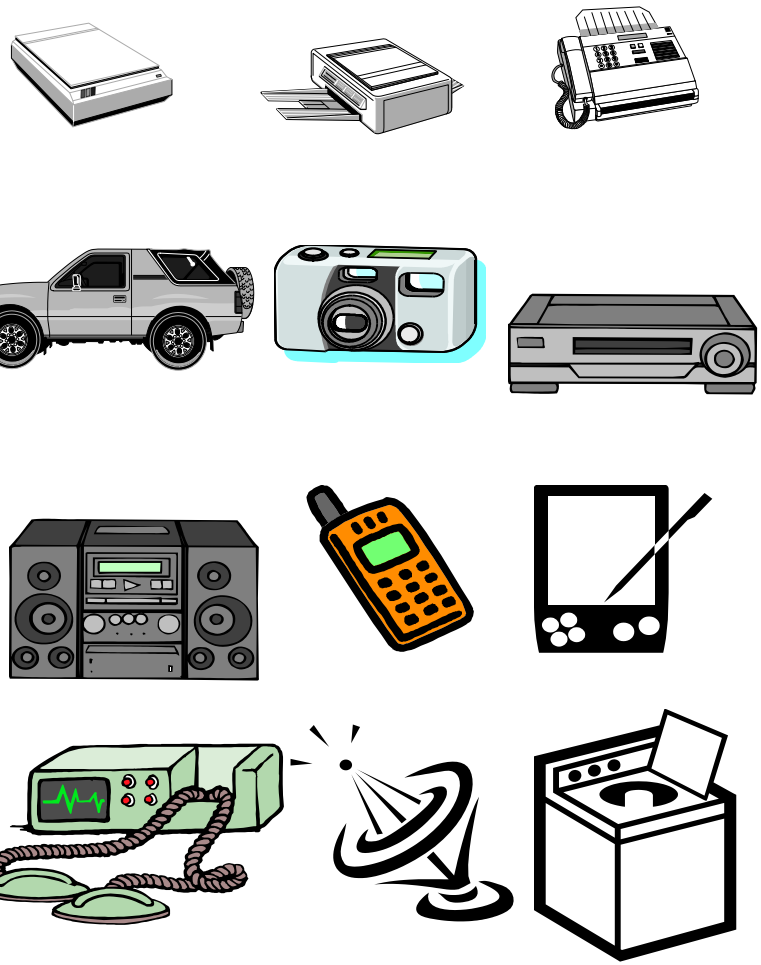
Embedded System …

- Application-specific requirement (like performance, power requirements, memory usage, etc.) are the key deciding factors
- Highly tailored to take advantage of the power saving modes supported by the hardware and the operating system
- For certain category of ESs like mission critical systems, the response time requirement is highly critical
- Execution behavior is deterministic for certain types of ESs like ‘Hard Real Time’ systems

A “short list” of embedded systems

- Anti-lock brakes
- Auto-focus cameras
- Automatic teller machines
- Automatic toll systems
- Automatic transmission
- Avionic systems
- Battery chargers
- Camcorders
- Cell phones
- Cell-phone base stations
- Cordless phones
- Cruise control
- Digital cameras
- Disk drives
- Electronic card readers
- Electronic instruments
- Electronic toys/games
- Factory control
- Fax machines
- Fingerprint identifiers
- Home security systems
- Life-support systems
- Medical testing systems

- Modems
- MPEG decoders
- Network cards
- Network switches/routers
- Pagers
- Photocopiers
- Point-of-sale systems
- Portable video games
- Printers
- Satellite phones
- Scanners
- Smart ovens/dishwashers
- Speech recognizers
- Stereo systems
- Teleconferencing systems
- Televisions
- Temperature controllers
- Theft tracking systems
- TV set-top boxes
- VCR’s, DVD players
- Video game consoles
- Video phones
- Washers and dryers

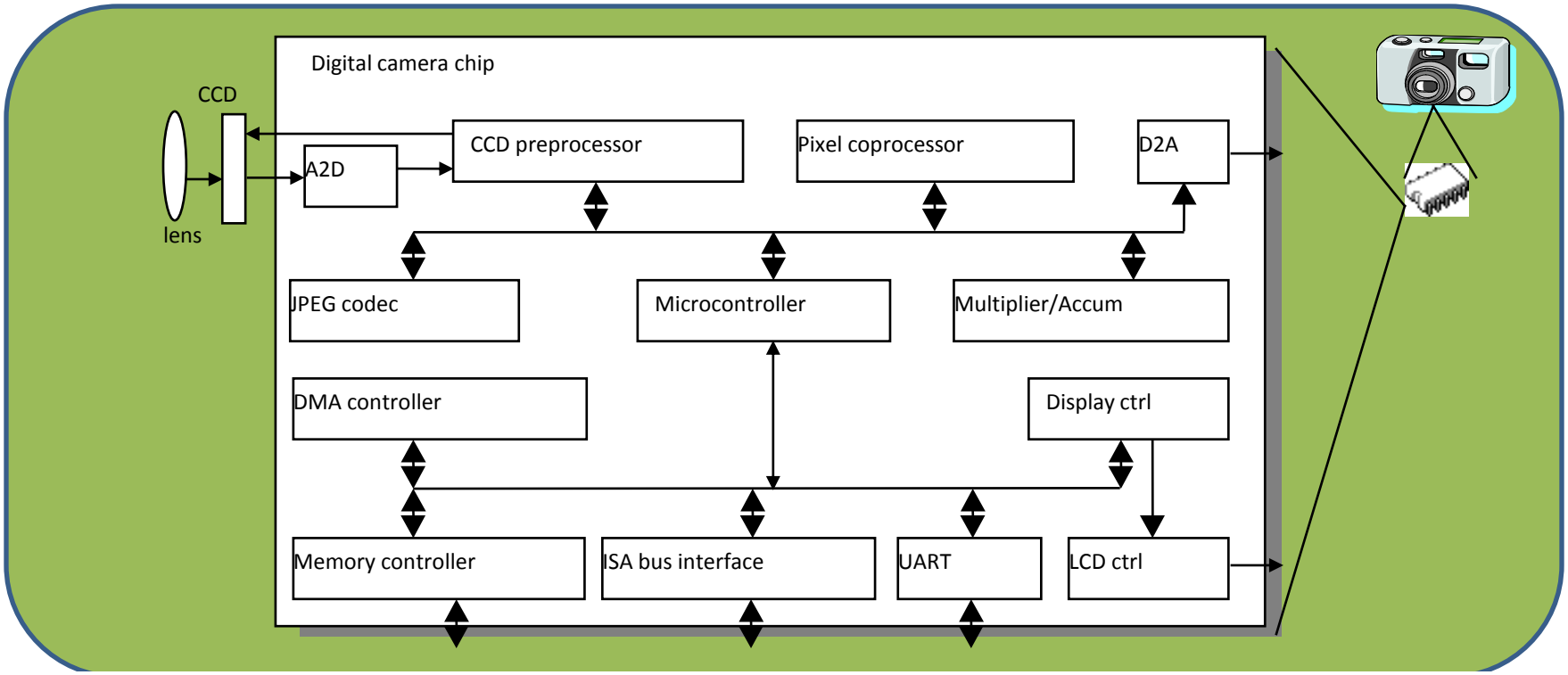


And the list goes on and o

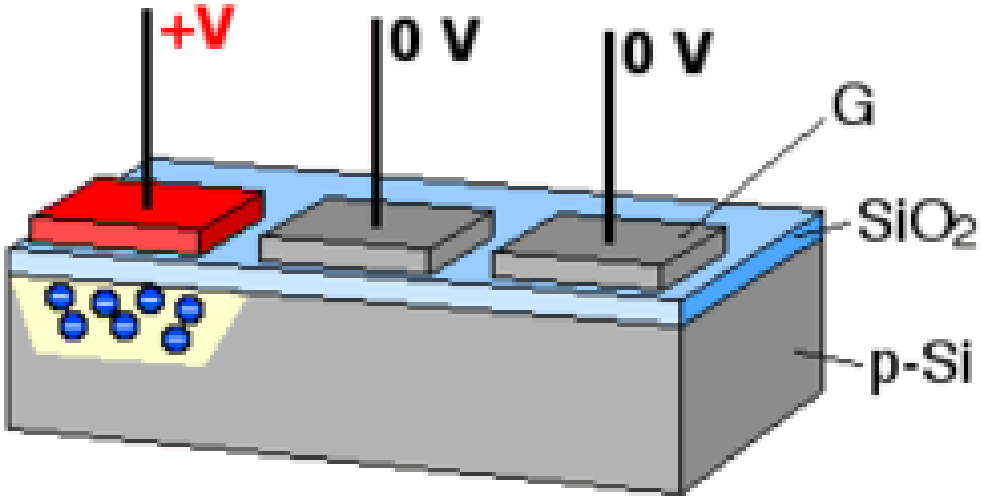
Some common characteristics of ESs

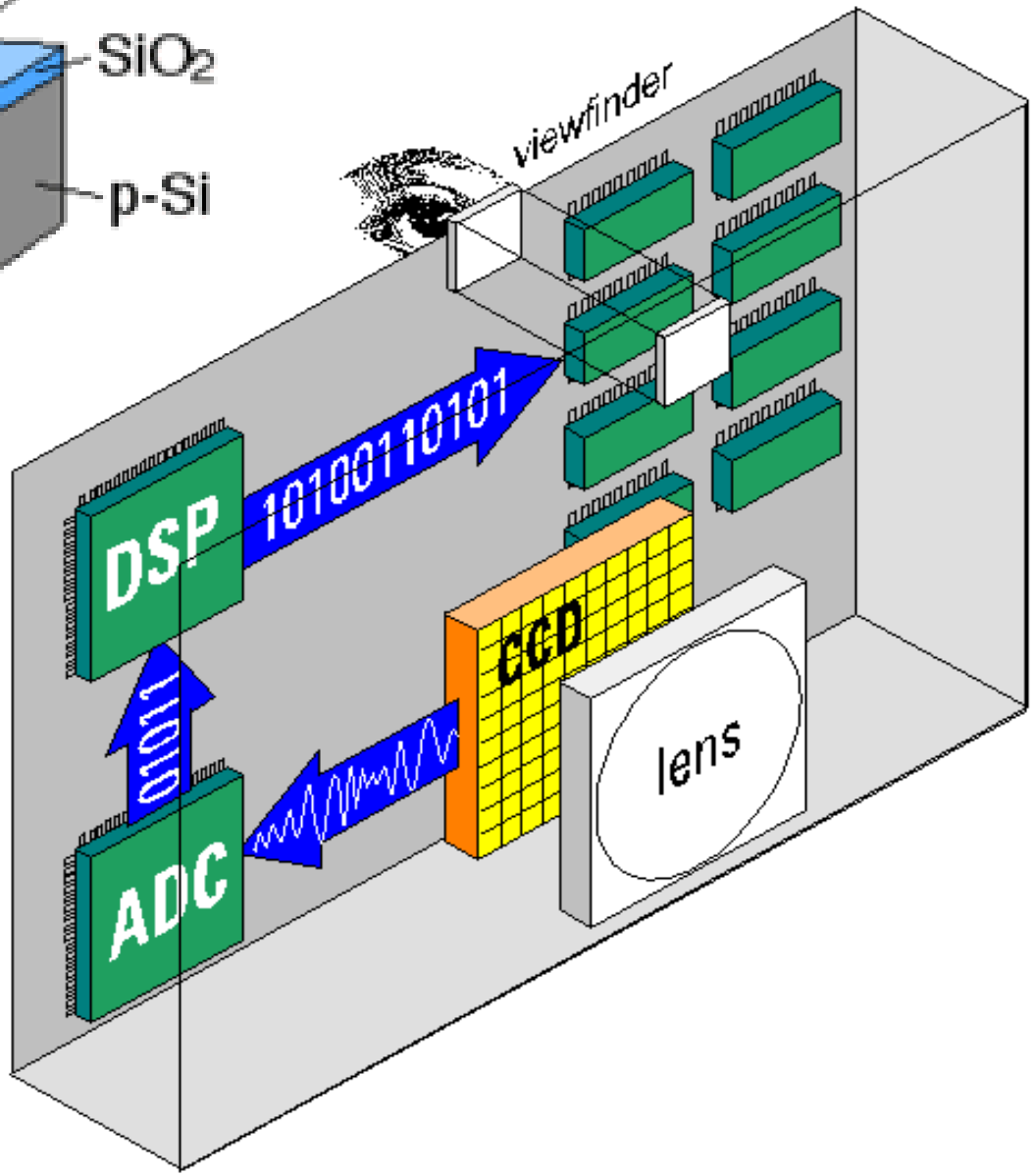
- **Single-functioned**
  - Executes a single program, repeatedly
- **Tightly-constrained**
  - Low cost, low power, small, fast, etc.
- **Reactive and real-time**
  - Continually reacts to changes in the system's environment
  - Must compute certain results in real-time without delay

An embedded system example – *a digital camera*



- **Single-functioned -- always a digital camera**
- **Tightly-constrained -- Low cost, low power, small, fast**
- **Reactive and real-time -- only to a small extent**





1.2 Classification of Embedded Systems

Based on different criteria:

1. **Based on generation**
2. **Complexity and performance requirements**
3. **Based on deterministic behaviour**
4. **Based on triggering**

Classification based on Generation:

- **First Generation:** ES were built around 8 bit microprocessors like 8085, and Z80, and 4 bit microcontrollers. Simple in hardware circuits with firmware developed in Assembly code. e.g. telephone keypads, stepper motor control unit.
- **Second Generation:** ES are built around 16 bit microprocessors and 8 or 16 bit microcontrollers, following the first generation ESs. The instruction set for the second generation processors/controllers were much more complex and powerful then 1st generation. Some of 2nd G ESs contained embedded operating systems for their operation. Data Acquisition System, SCADA (***Supervisory Control And Data Acquisition***) system.

Classification based on Generation: …

- **Third Generation:** With advances in processor tech., ES developers started making use of powerful 32 bit processor 16 bit microcontrollers for their design. e.g. DSPs, Application Specific Integrated Circuits (ASICs), processors like Intel, Pentium, Motorola 68K.

- **Fourth Generation:** The advent of System on Chip (SoC), reconfigurable processors and multicore processors are bringing high performance, tight integration and miniaturization into the embedded device market. The SoC technique implements a total system on a chip by integrating different functionalities with a processor core on an IC. Smart phone devices, mobile internet device.

Classification based on Complexity & Performance:

- **Small-Scale ESs:** ESs which are simple in application needs where the performance requirements are not time critical fall under this category. e.g. electronic toy, built around low performance and low cost 8 or 16 bit microprocessors/microcontrollers.

- **Medium-Scale ESs:** ESs are slightly complex in hardware and firmware requirements fall under this category. e.g. low cost 16 or 32 bit microprocessors / microcontrollers or DSP. They usually contain an embedded operating system (either general purpose or real time operating system) for functioning.

Classification based on Complexity & Performance: …

- **Large-Scale ESs/Complex Systems:** ESs which are highly complex hardware and firmware requirements fall under this category. They are employed in mission critical applications demanding high performance. Such systems are commonly built around high performance 32 or 64 bit RISC processor/controllers or Reconfigurable System on Chip (RSoC) or multi-core processor and programmable logic devices. e.g. multiple processor/controllers and co-units/hardware accelerators for offloading the processing requirements from the main processor of the system, Decoding/encoding of media, cryptographic function implementation.
  RTOS for task scheduling, prioritization and management.

1.4 Major Application of Embedded Systems

1.Consumer electronics: camcorders, cameras

2.Household Appliances: TV, DVD players, washing machine

3.Home automation and security systems: Aircon, CCTV, fire alarms

4.Automatic industry: engine control, ignition system, navigation

5.Telecom: Cell Phone, tel. switches, hand set multimedia appl.

6.Computer peripherals: Printer, scanners, fax machines

7.Computer Networking System: Network routers, switches, hubs

8.Healthcare: diff. kind of scanners, EEG, ECG machine

9.Measurement & Instrumentation: digital multi-meter, CROs

10.Banking & Retail: ATM, currency counter, point of sales (POS)

11.Card Reader: Barcode, smart card reader, hand held devices

1.4 Purpose of Embedded Systems

1. **Data Collection/ Storage/ Representation**
2. **Data Communication**
3. **Data (signal) processing**
4. **Monitoring**
5. **Control**
6. **Application specific user interface**

Design challenge – optimizing design metrics

- Obvious design goal:
  - Construct an implementation with desired functionality
- Key design challenge:
  - Simultaneously optimize numerous design metrics
- Design metric
  - A measurable feature of a system's implementation
  - Optimizing design metrics is a key challenge

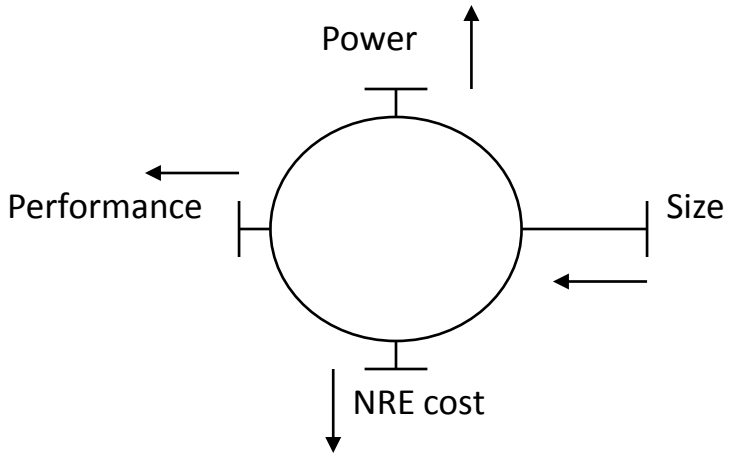
Design challenge – optimizing design metrics

- **Common metrics**
  - **Unit cost:** the monetary cost of manufacturing each copy of the system, excluding NRE cost
  - **NRE cost (Non-Recurring Engineering cost):** The one-time monetary cost of designing the system
  - **Size:** the physical space required by the system
  - **Performance:** the execution time or throughput of the system
  - **Power:** the amount of power consumed by the system
  - **Flexibility:** the ability to change the functionality of the system without incurring heavy NRE cost

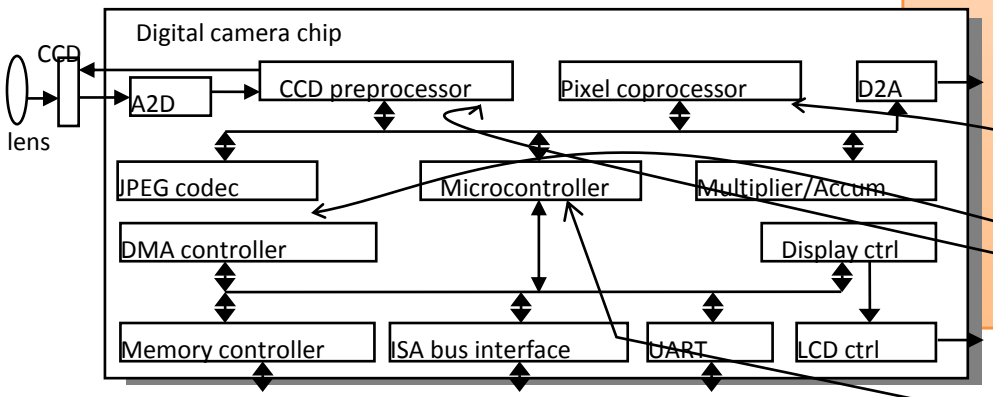
Design challenge – optimizing design metrics

- Common metrics (continued)
  - Time-to-prototype: the time needed to build a working version of the system
  - Time-to-market: the time required to develop a system to the point that it can be released and sold to customers
  - Maintainability: the ability to modify the system after its initial release
  - Correctness, safety, many more

Design metric competition -- improving one may worsen others



- Expertise with both **software and hardware** is needed to optimize design metrics
  - Not just a hardware or software expert, as is common
  - A designer must be comfortable with various technologies in order to choose the best for a given application and constraints



Hardware

Software

UART: Universal Asynchronous Receiver Transmitter