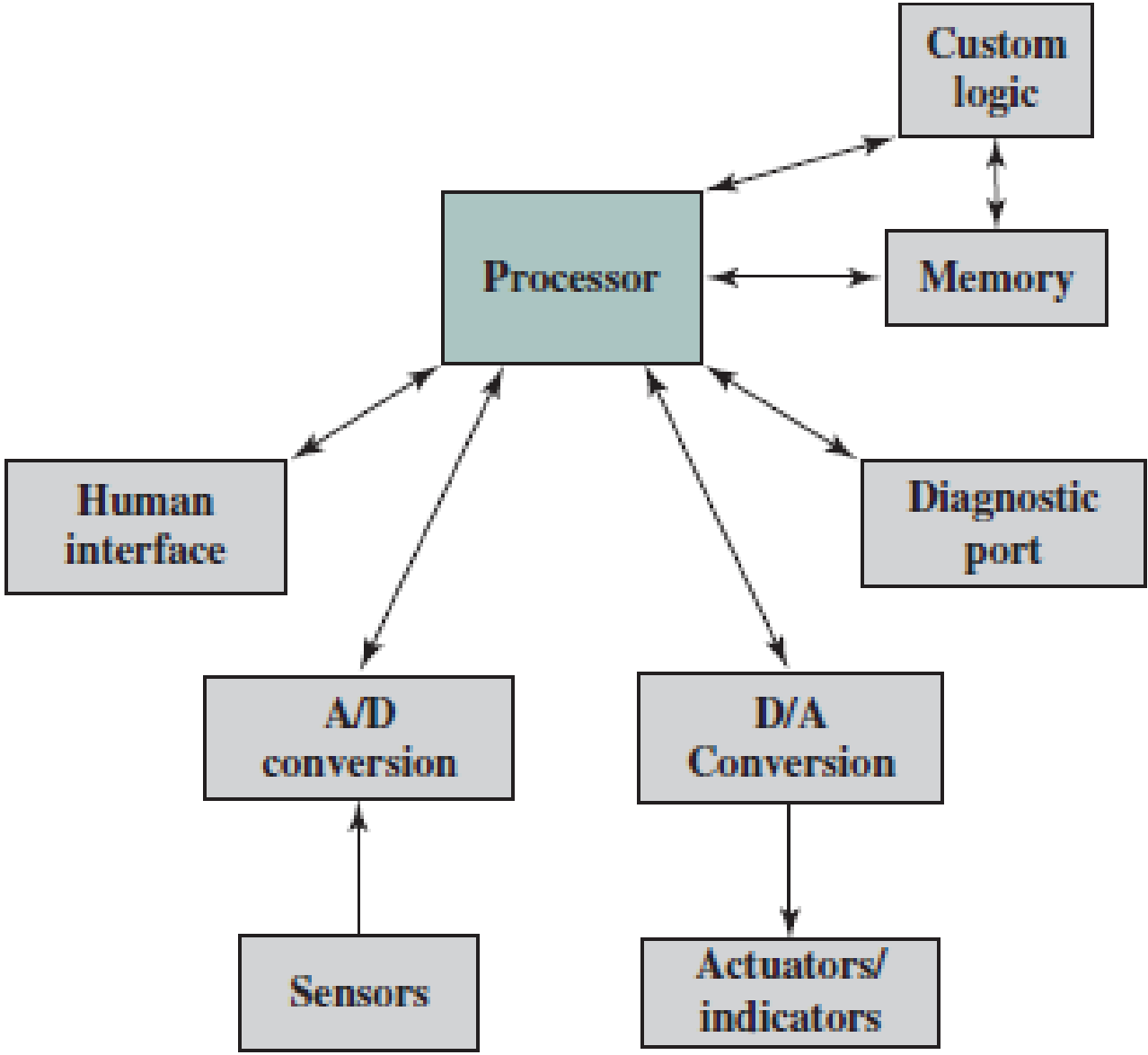
EMBEDDED SYSTEM

Introduction

- The term *embedded system* refers to the use of electronics and software within a Product.
- Today many devices that use electric power have an embedded computing system.
- In near future virtually all such devices will have embedded computing systems.
- Examples include cell phones, digital cameras, video cameras, calculators, microwave ovens, home security systems, washing machines, lighting systems, thermostats, printers, various automotive systems (e.g., transmission control, cruise control, fuel injection, anti-lock brakes, and suspension systems), tennis rackets, toothbrushes.
- Real-time constraints imposed by the interaction with the environment.

General organization of an Embedded System



The Internet of Things

- The **Internet of things (IoT)** is a term that refers to the expanding interconnection of smart devices, ranging from appliances to tiny sensors.
- The embedding of short-range mobile transceivers into a wide array of gadgets and everyday items, enabling new forms of communication between people and things, and between things themselves.
- The Internet now supports the interconnection of billions of industrial and personal objects, usually through cloud systems.
- These devices are low-bandwidth, low- repetition data- capture, and low-bandwidth data- usage appliances that communicate with each other and provide data via user interfaces.

Embedded Operating Systems

- There are two general approaches to developing an embedded operating system (OS).
- The first approach is to take an existing OS and adapt it for the embedded application. For example, there are embedded versions of Linux, Windows, and Mac, as well as other commercial and proprietary operating systems specialized for embedded systems.
- The other approach is to design and implement an OS intended solely for embedded use. An example of the latter is TinyOS, widely used in wireless sensor networks.

Application Processors versus Dedicated Processors

- **Application processors** are defined by the processor's ability to execute complex operating systems, such as Linux, Android, and Chrome.
- Most embedded systems employ a **dedicated processor**, which, as the name implies, is dedicated to one or a small number of specific tasks required by the host device.

Microprocessors versus Microcontrollers

- **Microprocessor** chips included registers, an ALU, and some sort of control unit or instruction processing logic.
- A **microcontroller** is a single chip that contains the processor, non- volatile memory for the program (ROM), volatile memory for input and output (RAM), a clock, and an I/O control unit. Also called a "computer on a chip".
- Microcontrollers come in a range of physical sizes and processing power. Processors range from 4-bit to 32-bit architectures.
- Microcontrollers tend to be much slower than microprocessors, typically operating in the MHz range rather than the GHz speeds of microprocessors.
- Usually microcontroller does not provide for human interaction.
- The microcontroller is programmed for a specific task, embedded in its device, and executes as and when required.

Microprocessors versus Microcontrollers

Microprocessor	Microcontroller
Since memory and I/O are connected externally, the circuit becomes large in size.	Since memory and I/O are present together, the internal circuit is small in size.
It cannot be used in compact systems	It can be used in compact systems.
Cost is high	Cost is low
RAM, ROM, I/O units, and other peripherals are not embedded on a single chip.	RAM, ROM, CPU and other peripherals are embedded on a single chip.
Do not have power saving mode.	Have power-saving mode.
Used in personal computers.	Used in embedded systems.
Less number of registers.	More number of registers.
Uses an external bus.	Uses an internal controlling bus.
Complex and expensive due to large number of instructions to process.	Simple and inexpensive due to less number of instructions to process.
Can run at a very high speed.	Can run up to 200MHz or more.

Embedded versus Deeply Embedded Systems

- A subset of embedded systems is referred to as **deeply embedded systems.**
- A deeply embedded system uses a microcontroller rather than a microprocessor, is not programmable once the program logic for the device has been burned into ROM (read- only memory), and has no interaction with a user.
- Deeply embedded systems are dedicated, single-purpose devices that detect something in the environment, perform a basic level of processing, and then do something with the results.
- Deeply embedded systems often have wireless capability and appear in networked configurations, such as networks of sensors deployed over a large area (e.g., factory, agricultural field).
- The Internet of things depends heavily on deeply embedded systems. Typically, deeply embedded systems have extreme resource constraints in terms of memory, processor size, time, and power consumption.