Embedded Motion Control

Real-Time Concepts for Embedded Systems

Chapter 1: Introduction

Group 1:

Rein Appeldoorn	0657407
Jeroen Graafmans	0657171
Bart Peeters	0748906
Ton Peters	0662103
Scott van Venrooij	0658912

TU/e Technische Universiteit Eindhoven University of Technology

Where innovation starts

Outline

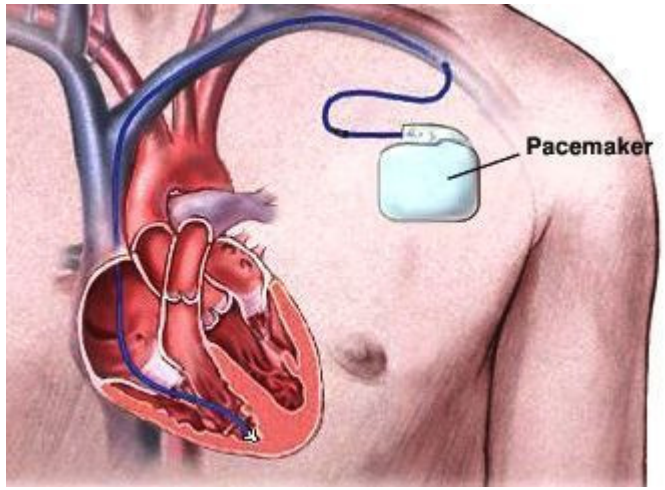
- Real-Life examples of embedded systems
- Definition embedded systems
- Definition real-time embedded systems
- Future
- Points to remember

Examples (1)

- **Consumer electronics**
  - **Digital cameras**
  - **DVD players**
  - **Printers**
  - **Mobile phone**

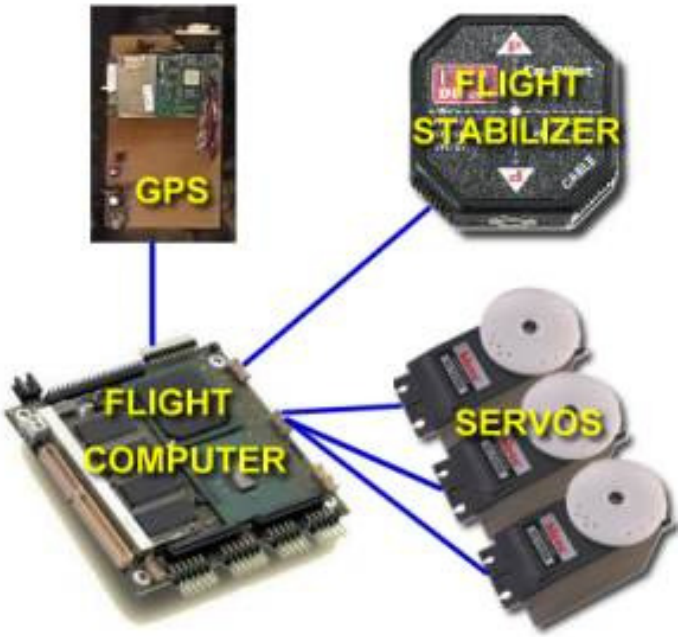
Examples (2)

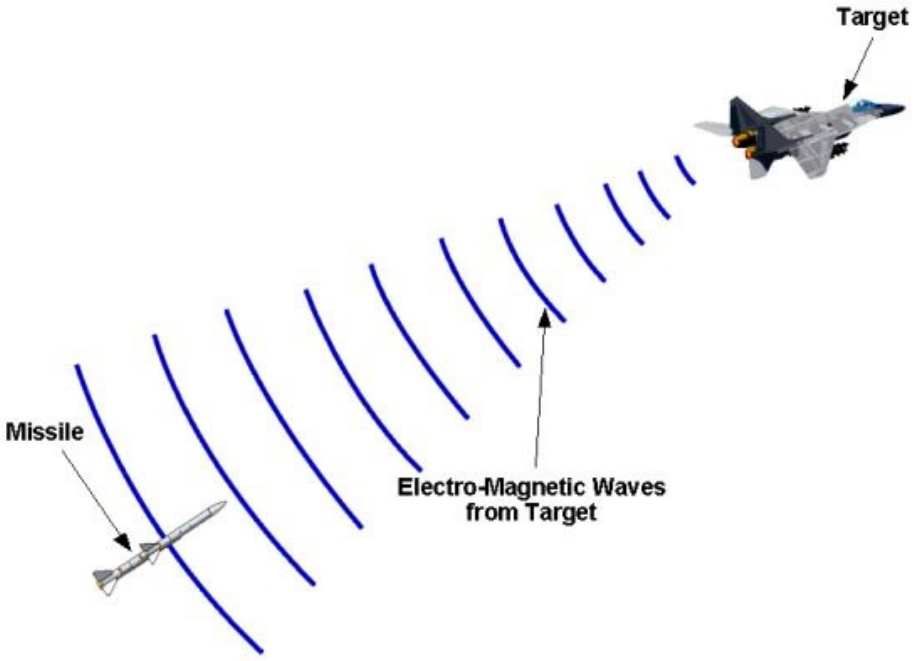
- **Medical equipment**
  - **Cardiac arrhythmia monitors**
  - **Cardiac pacemakers**



Examples (3)

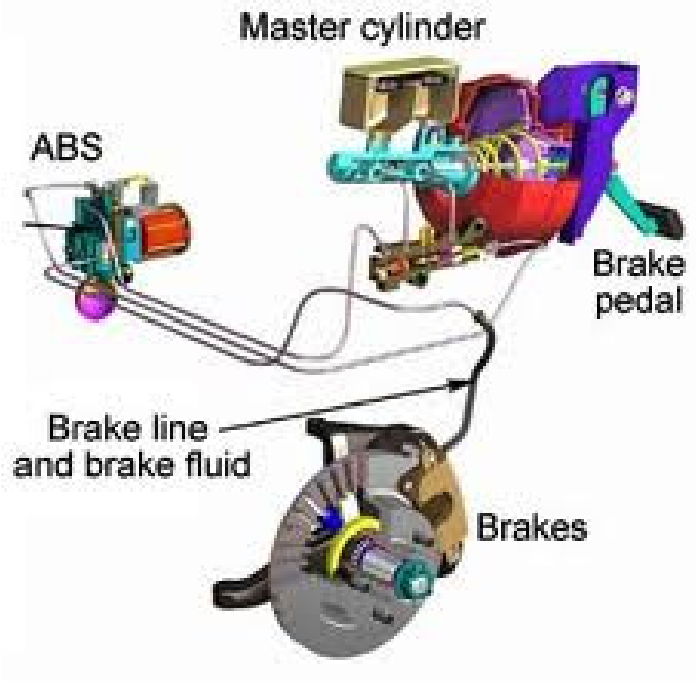
- **Advanced avionics**
  - **Flight control systems**
  - **Missile guidance systems**





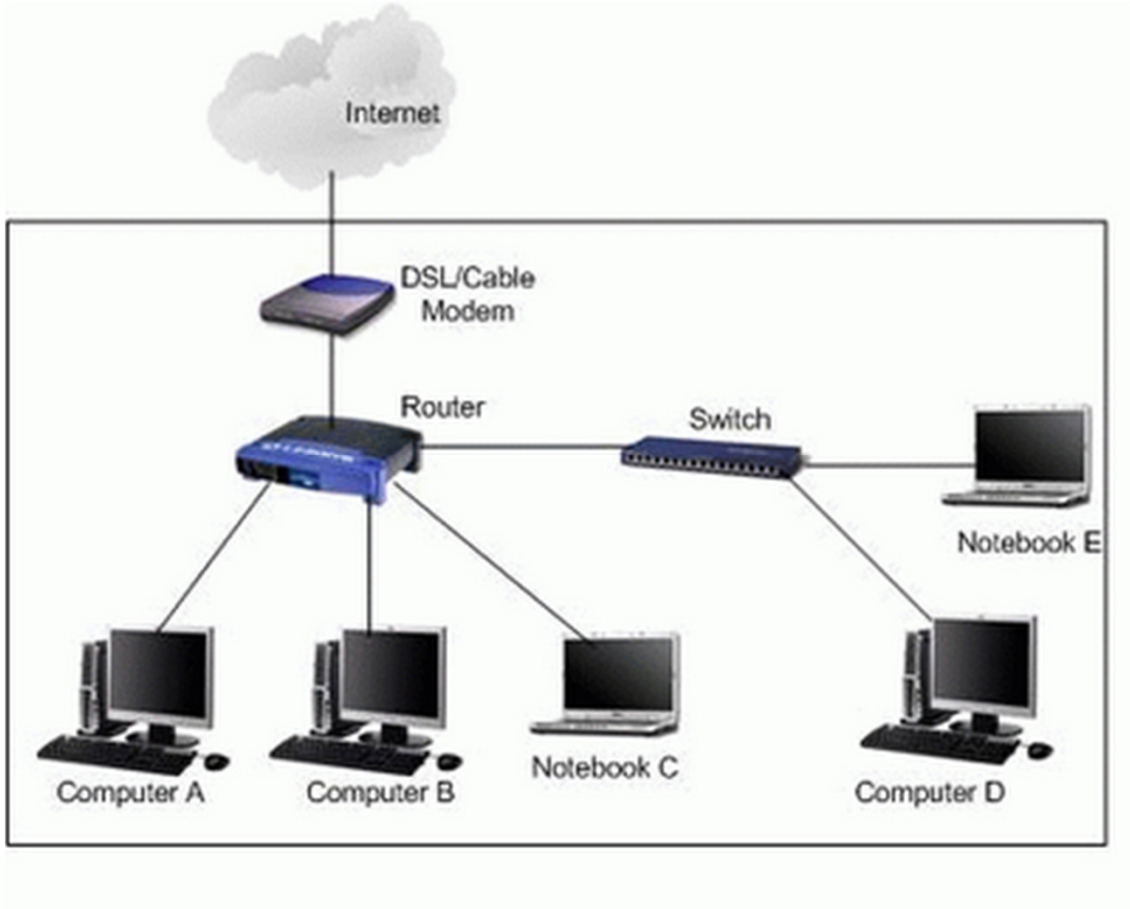
Examples (4)

- **Automotive designs**
  - **Fuel injection systems**
  - **Auto-braking systems**



Examples (5)

- **Internet**



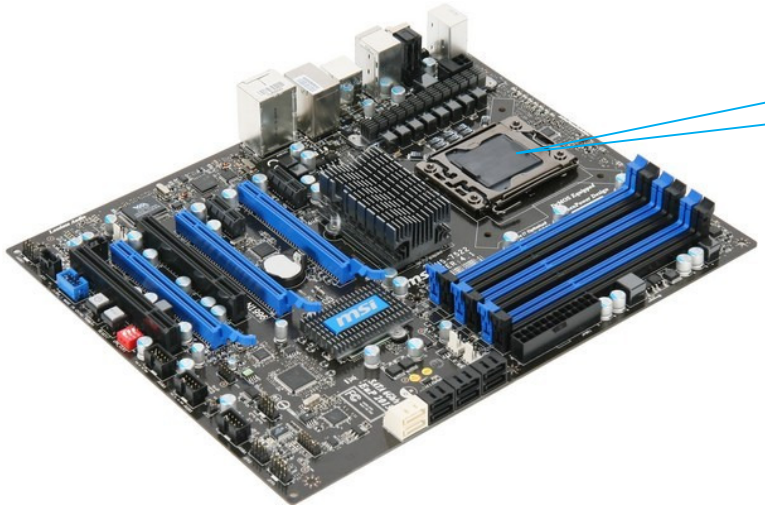
Defining the embedded system

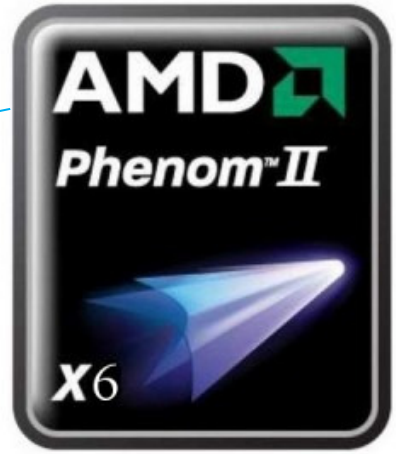
- **General definition:**

“Embedded systems are computing systems with tightly coupled hardware and software integration, that are designed to perform a dedicated function.”

PC Processors

- **Personal computers have stock processors**
  - **Full scale of features**
  - **Memory management, extra costs**
  - **Compatible with multiple operating systems**
  - **Not designed for a specific task**

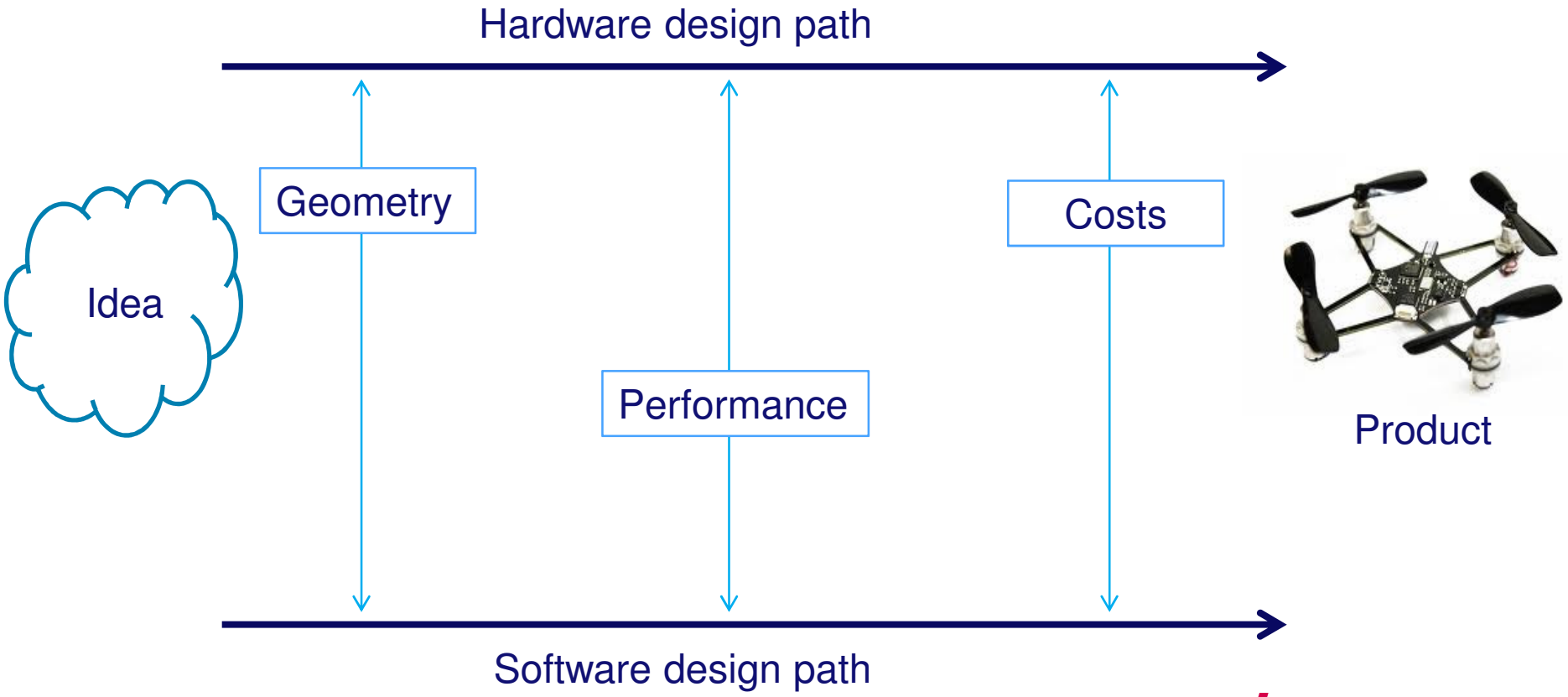




Embedded Processors

- **Embedded systems have processors with special purposes:**
  - **Power**
  - **Geometry**
  - **Price**
  - **Heat productivity**

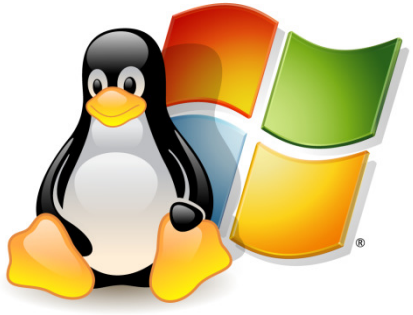
Hardware/Software development for E.S.



Cross-Platform Development

- **Host system**
  - **System on which you develop**
- **Target system**
  - **The embedded system**
- **Cross-compiling**
  - **Same processor architecture**

Cross-Platform Development



HS: Linux/Mac/Windows

- Develop C, C++ Code
- Compile to executable
  - Debug your code

Flash executable to storage on E.S.

TS: Embedded system

- Run the compiled executable

Software Storage and Upgradeability

- Code of an embedded system needs to be stored
  - ROM (Read only Memory)
    - Non-volatile content
    - Without external power source
  - RAM (Random Access Memory)
    - External power source
    - Faster

ROM

1. Mask Programmed ROM
2. Field Programmed ROM (PROM)
3. Erasable Programmable ROM (EPROM)
4. Electrically Erasable Programmable ROM (EEPROM)
5. Flash Memory

ROM

1. **Mask Programmed ROM**
2. Field Programmed ROM (PROM)
3. Erasable Programmable ROM (EPROM)
4. Electrically Erasable Programmable ROM (EEPROM)
5. Flash Memory

- **Programmed during manufacturing process**
- **Content can't be changed**

- **Advantage: Cheap**

ROM

1. Mask Programmed ROM
2. **Field Programmed ROM (PROM)**
3. Erasable Programmable ROM (EPROM)
4. Electrically Erasable Programmable ROM (EEPROM)
5. Flash Memory

- **Custom-Programmed once**
- **Content can't be changed**

- **Advantage: Cheap and Custom**

ROM

1. Mask Programmed ROM
2. Field Programmed ROM (PROM)
3. **Erasable Programmable ROM (EPROM)**
4. Electrically Erasable Programmable ROM (EEPROM)
5. Flash Memory

- **Custom-Programmed**
- **Content can be changed**
- **Changes only by completely erasing the content**

- **Advantage: Reprogrammable**

ROM

1. Mask Programmed ROM
2. Field Programmed ROM (PROM)
3. Erasable Programmable ROM (EPROM)
4. **Electrically Erasable Programmable ROM (EEPROM)**
5. Flash Memory

- **Custom-Programmed**
- **Content can be changed**
- **Every byte can be reprogrammed separately**

- **Advantage: Reprogrammable (byte by byte)**

ROM

1. Mask Programmed ROM
2. Field Programmed ROM (PROM)
3. Erasable Programmable ROM (EPROM)
4. Electrically Erasable Programmable ROM (
5. **Flash Memory**

- **Custom-Programmed**
- **Content can be changed**
- **Blocks (e.g. 512-byte) can be reprogrammed**

- **Advantage: Reprogrammable and faster than EEPROM**

Which ROM to choose

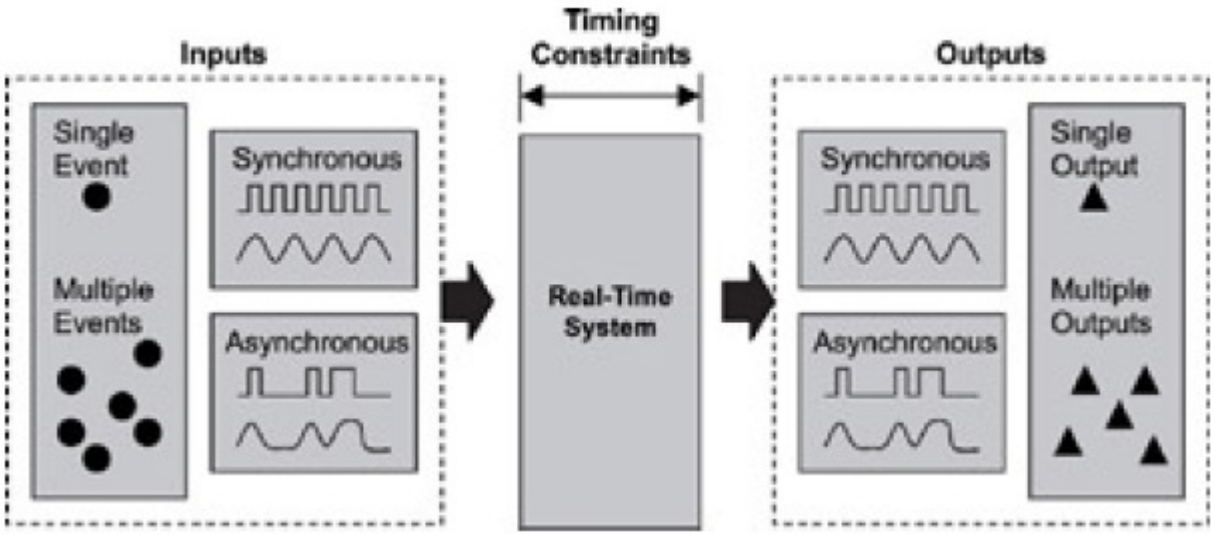
- **ROM/PROM: Cheap**
- **EPROM: Rewritable (at location)**
- **EEPROM/Flash: Rewritable from distance**

RAM

- Dynamic RAM (DRAM)
  - Needs periodic refreshing
- Static RAM (SRAM)
  - Retains content as long as power is supplied
- Non-Volatile RAM (NVRAM)
  1. SRAM with backup power
  2. SRAM with EEPROM (saves data when power is off)

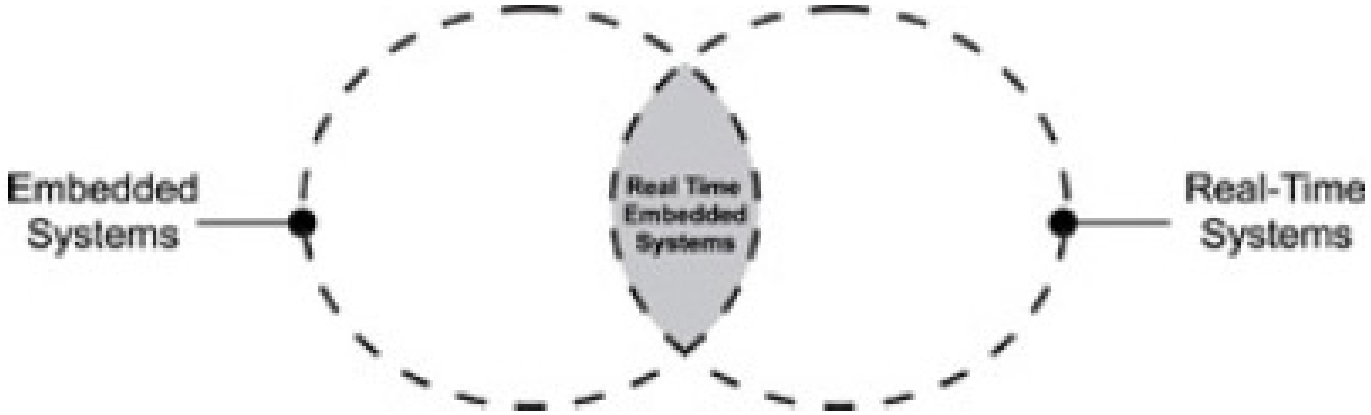
Real-time systems

- **Real-time systems**
  - **Systems that respond to external events in a timely fashion**



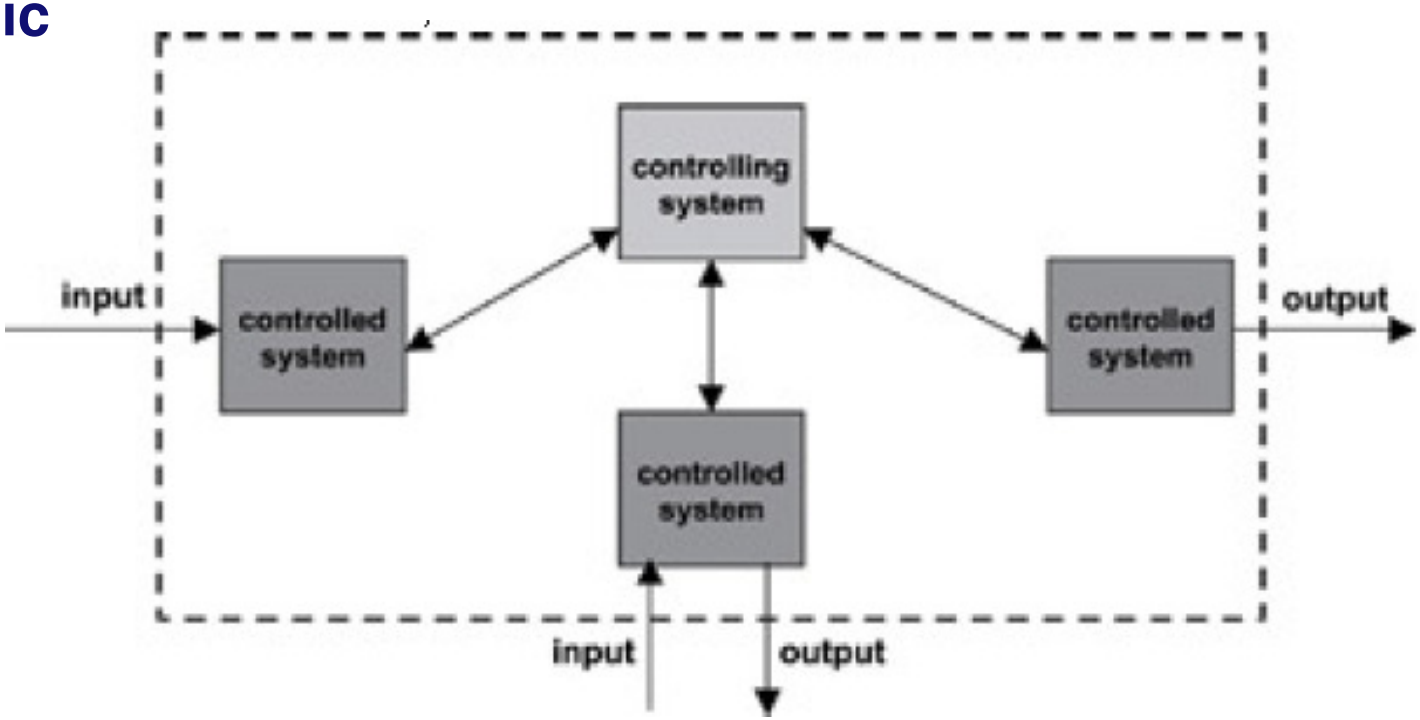
Real-time embedded systems

- **Relationship between real-time systems and embedded systems:**



Real-time systems

- **Interaction**
  - **Periodic**
  - **Aperiodic**



Real-time systems

- **Example**
  - **Real-time weapons defense system**

  - **http://www.youtube.com/watch?v=jZ-53a2JsNg**

Real-time systems

- **Two essential characteristics**
  - **Logical or functional correctness**
  - **Timing correctness**
- **Deterministic**

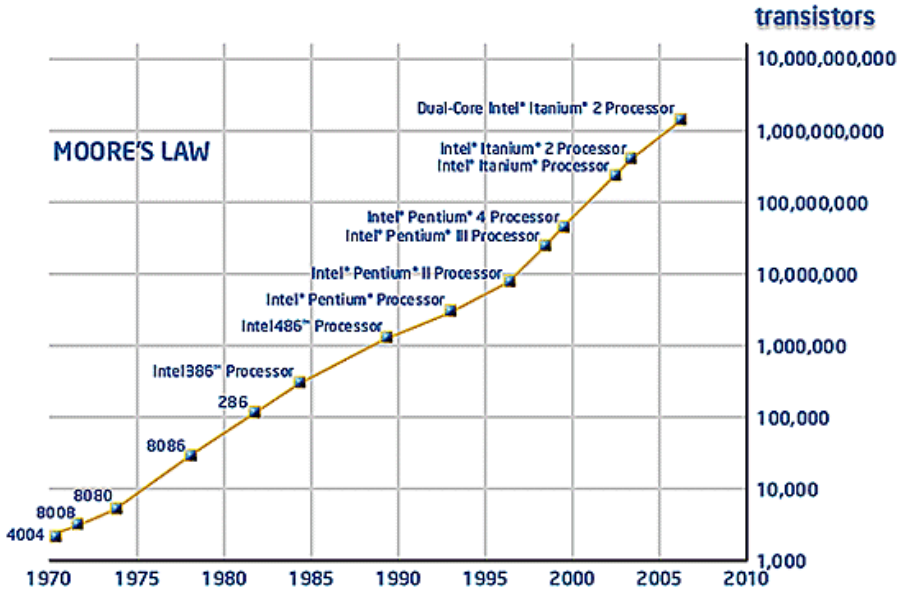
Real-time systems

- **Hard**
  - **Near-zero flexibility**
  - **Missed deadline: catastrophic**
- **Soft**
  - **Degree of flexibility (non-zero)**
  - **Missed deadline: non-catastrophic**

Trends of Embedded Systems (1)

- **Processing power increases according to Moore's Law**

  "The number of transistors per integrated circuit doubles every 18 months."



Trends of Embedded Systems (2)

- **Product markets now dictate six- to nine-month cycles**
- **Connectivity is now a requirement**
- **Electronics-based products become more complex**

Trends of Embedded Systems (3)

- **New/smarter classes of products**

- **Embedded systems will reshape the world**

Future of Embedded Systems (4)

- **Google Glass Project**
- **http://www.youtube.com/watch?v=9c6W4CCU9M4**

Points to remember

- **Embedded systems:**
  - **Built for a specific application.**
  - **Generally built using embedded processors**
- **Real-time systems:**
  - **Timing is as important as functionality**
  - **Hard and soft real-time systems**
- **Real-time embedded systems**
  - **Embedded systems with real-time behaviors**

Questions