University of San Carlos | Department of
COMPUTER ENGINEERING

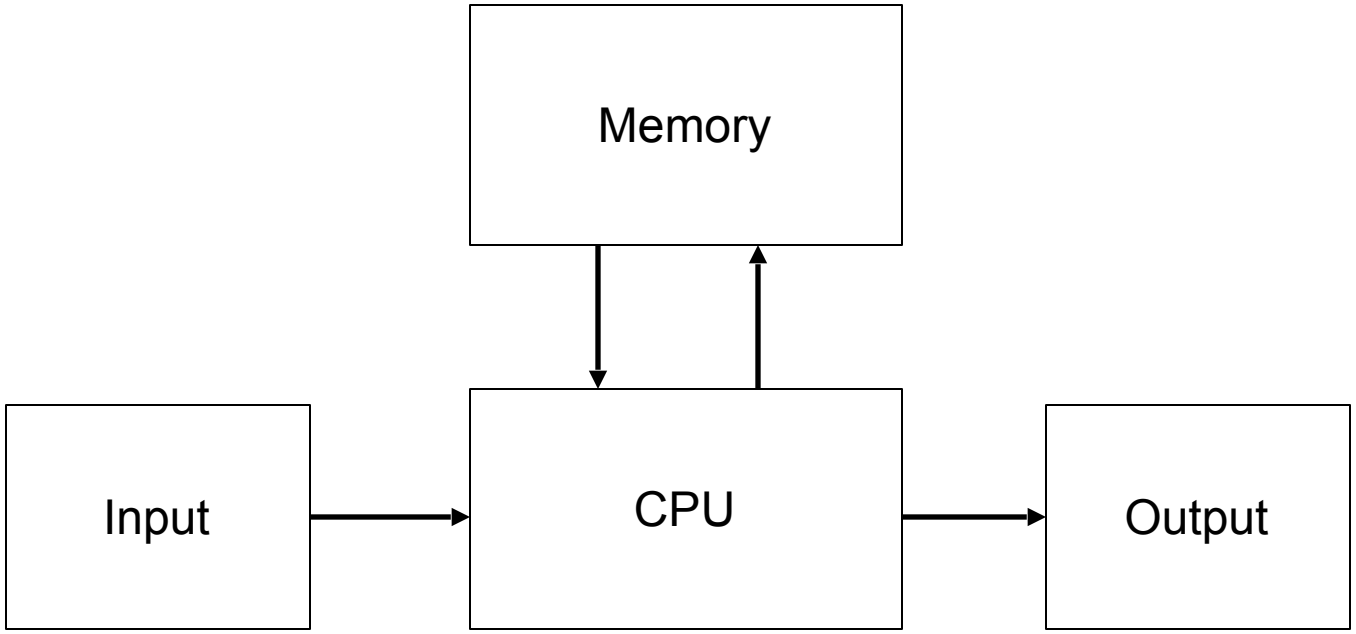
CpE 3201
Embedded Systems

Introduction to Embedded Systems

Microcomputer Concepts

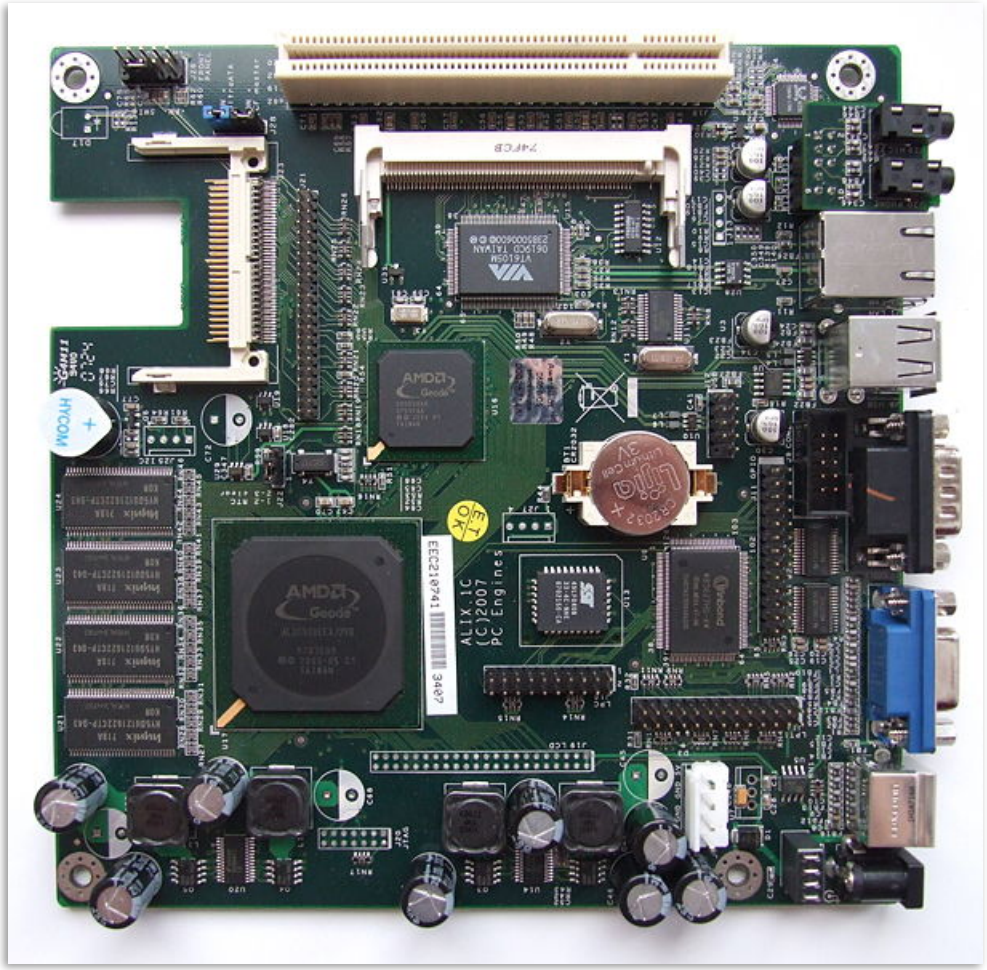
- Microcomputers is composed of three basic components:
  - CPU
  - Memory
  - I/O

Microcomputer



Embedded Systems

- A computer system designed to do one or a few dedicated and/or specific functions often with real-time computing constraints
- Its is controlled by one or more main processing cores that are typically either microcontrollers (MCU) or digital signal processors (DSP)



Embedded Computer

Embedded Computer vs PC

- An embedded computer is frequently a computer that is implemented for a particular purpose.
- In contrast, an average PC computer usually serves a number of purposes: checking email, surfing the internet, listening to music, word processing, etc…

PC (general purpose computer)

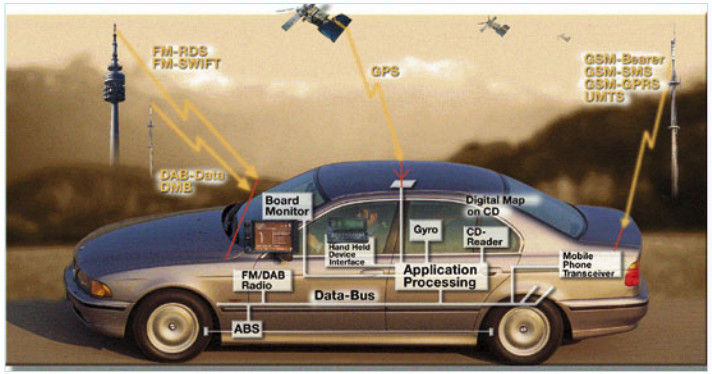
- Checking Email
- Surfing the Web
- Multimedia
- Complex Mathematical Calculations

Embedded Computer

- Control Speed of a DC Motor

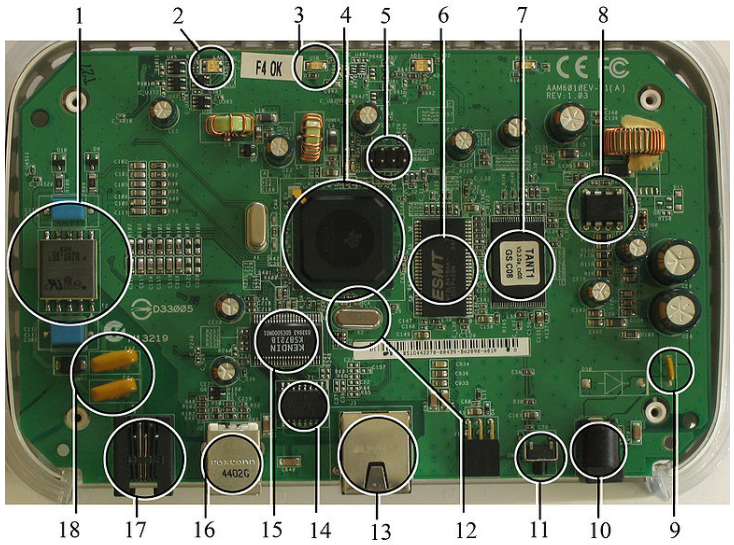
Applications of Embedded Computers

- Household
  - Digital timers in microwave ovens, washing machines
  - Digital clocks and watches
- Automotive
  - BCM
  - GPS
  - ABS



Applications of Embedded Computers

- Communications
  - Modems
  - Routers and switches
  - Network Interface Cards
- Multimedia
  - Music Players



Interface

- Embedded computers rarely have a generic interface, however. Even if embedded systems have a keypad and an LCD display, they are rarely capable of using many different types of input or output. An example of an embedded system with I/O capability is a security alarm with an LCD status display, and a keypad for entering a password

In General…

- An embedded system is:
  - a system built to perform its duty, completely or partially independent of human intervention
  - specially designed to perform a few tasks in the most efficient way
  - Interacts with physical elements in our environment, viz. controlling and driving a motor, sensing temperature, etc.

Real-Time Systems

- From an implementation viewpoint, there is a major difference between a computer and an embedded system
- Embedded systems are often required to provide ***Real-Time response***

Real-Time Systems (definition)

- Defined as a system whose correctness depends on the timeliness of its response.
- Examples of such systems are flight control systems of an aircraft, sensor systems in nuclear reactors and power plants.
- For these systems, delay in response is a fatal error.

History

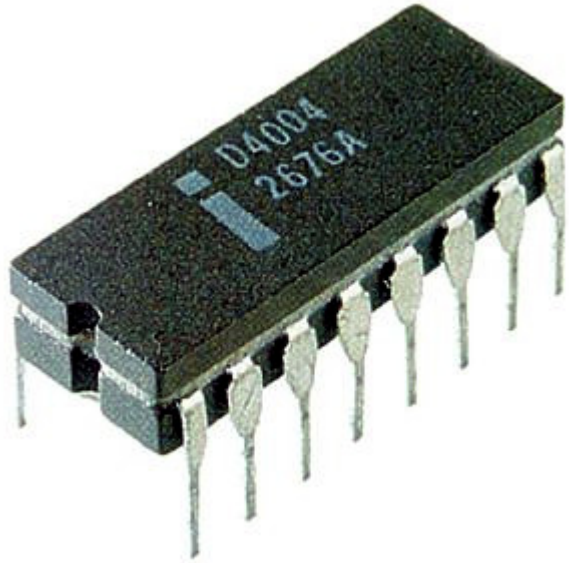
- The first recognizably modern embedded systems was the Apollo Guidance Computer, developed by Charles Stark Draper at the MIT Instrumentation Laboratory.
- The Apollo guidance computer was considered the riskiest item in the Apollo project as it employed the then newly developed monolithic integrated circuits to reduce the size and weight

History

- Autonetics D-17 guidance computer for the Minuteman missile was released in 1961.
- It was built from transistor logic and had a hard disk for main memory.

Evolution (uProcessor)

- The first microprocessor for example, the Intel 4004, was designed for calculators and other small systems but still required many external memory and support chips.



Evolution (uProcessor)

- In 1978 National Engineering Manufacturers Association released a "standard" for programmable microcontrollers, including almost any computer-based controllers, such as single board computers, numerical, and event-based controllers.

Microcomputer as a Controller

- Microcomputers is not only used in computing and processing data, it is also used to control electronic and electromechanical devices.
- To perform control functions, microcomputers execute a certain program that defines the function

Processors in Embedded Systems

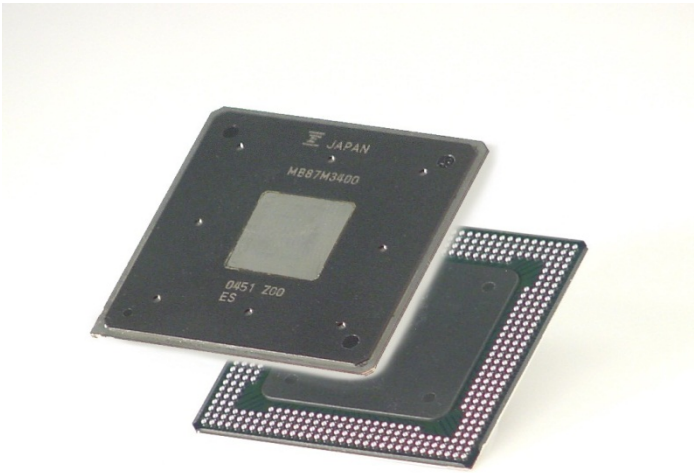
- Categories
  - Microprocessor (uP)
  - Microcontrollers (uC)
- Uses different CPU architectures as Von Nuemann as well as various degrees of Harvard Architecture.

Embedded Systems Solutions

- Ready-made Computers
  - common design style uses a small system module, perhaps the size of a business card, holding high density BGA chips such as an ARM-based System-on-a-chip processor and peripherals, external flash memory for storage, and DRAM for runtime memory

Embedded Systems Solutions

- ASIC & FPGA
  - A common array of n configuration for very-high-volume embedded systems is the system on a chip (SoC) which contains a complete system consisting of multiple processors, multipliers, caches and interfaces on a single chip



WiMax SoC

Peripherals

- Serial Communication Interfaces (SCI): RS-232, RS-422, RS-485 etc.
- Synchronous Serial Communication Interface: I2C, SPI, SSC and ESSI (Enhanced Synchronous Serial Interface)
- Universal Serial Bus (USB)
- Multi Media Cards (SD Cards, Compact Flash etc.)
- Networks: Ethernet, LonWorks, etc.

Peripherals

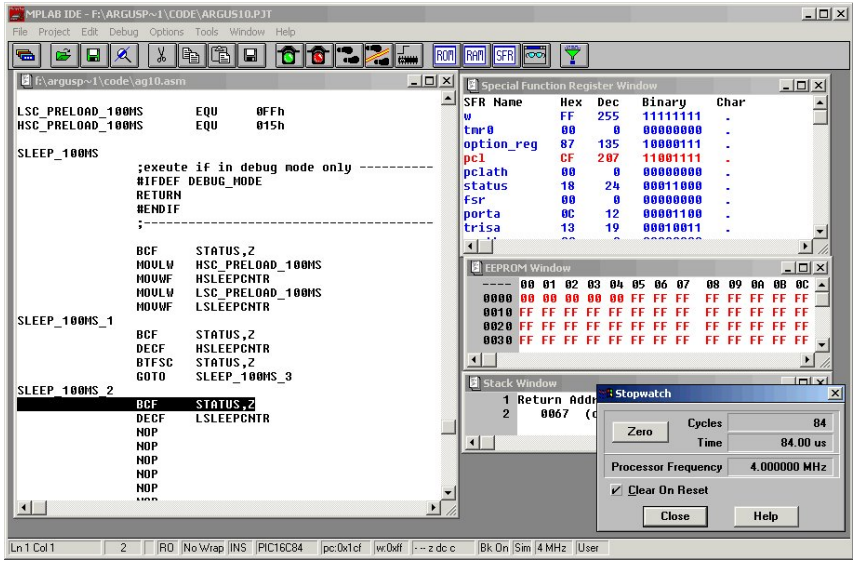
- Fieldbuses: CAN-Bus, LIN-Bus, PROFIBUS, etc.
- Timers: PLL(s), Capture/Compare and Time Processing Units
- Discrete IO: aka General Purpose Input/Output (GPIO)
- Analog to Digital/Digital to Analog (ADC/DAC)
- Debugging: JTAG, ISP, ICSP, BDM Port, BITP, and DP9 ports.

Development Tools

- In circuit debuggers or emulators
- Utilities to add a checksum or CRC to a program, so the embedded system can check if the program is valid
- For systems using digital signal processing, developers may use a math workbench such as Scilab / Scicos, MATLAB / Simulink, EICASLAB, MathCad, Mathematica or FlowStone DSP to simulate the mathematics

Development Tools

- Custom compilers and linkers may be used to improve optimization for the particular hardware
- An embedded system may have its own special language or design tool, or add enhancements to an existing language such as Forth or Basic



Development Tools

- Another alternative is to add a real-time operating system or embedded operating system, which may have DSP capabilities like DSPnano RTOS.
- Modeling and code generating tools often based on state machines

Software Architecture

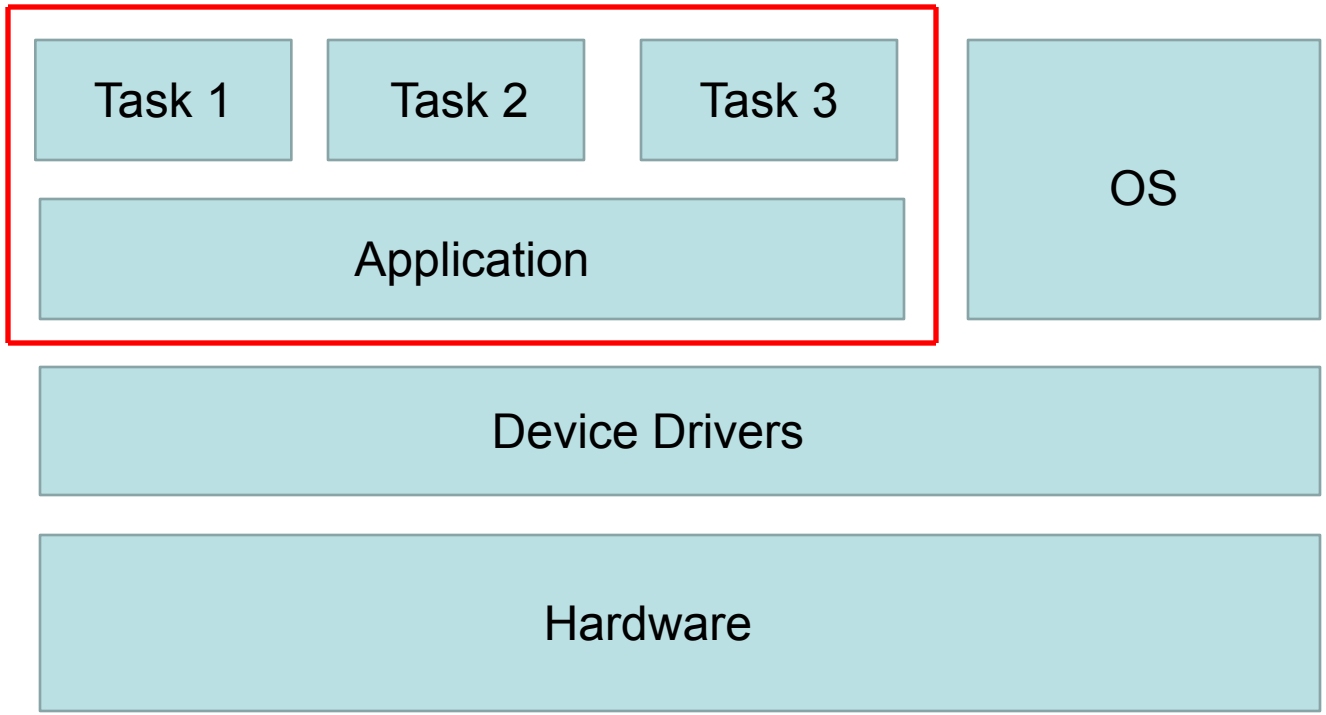
- Simple Control Loop
  - In this design, the software simply has a loop
  - The loop calls subroutines, each of which manages a part of the hardware or software

Software Architecture

- Interrupt Controlled System
  - Tasks performed by the system are triggered by different kinds of events
  - An interrupt could be generated for example by a timer in a predefined frequency, or by a serial port controller receiving a byte

Software Architecture

- Cooperative multitasking
  - A low-level piece of code switches between tasks or threads based on a timer (connected to an interrupt)
  - This is the level at which the system is generally considered to have an "operating system" kernel
  - Depending on how much functionality is required, it introduces more or less of the complexities of managing multiple tasks running conceptually in parallel



Example of Cooperative Multitasking Software Architecture

Software Architecture

- Microkernels & Exokernels
  - A *microkernel* is a logical step up from a real-time OS. The usual arrangement is that the operating system kernel allocates memory and switches the CPU to different threads of execution. User mode processes implement major functions such as file systems, network interfaces, etc.
  - *Exokernels* communicate efficiently by normal subroutine calls. The hardware, and all the software in the system are available to, and extensible by application programmers.

Software Architecture

- Monolithic Kernels
  - A relatively large kernel with sophisticated capabilities is adapted to suit an embedded environment
  - This gives programmers an environment similar to a desktop operating system like Linux or Microsoft Windows, and is therefore very productive for development;
  - Downside, it requires considerably more hardware resources, is often more expensive, and because of the complexity of these kernels can be less predictable and reliable

University of San Carlos | Department of
COMPUTER ENGINEERING

CpE 3201
Embedded Systems

End of Lecture

References:

- Jonathan W. Valvano. Embedded Microcomputer Systems Int'l, 3rd ed. Connecticut: Cengage Learning, 2011.
- Ian McLoughlin. Computer Architecture: An Embedded Approach. New York: McGraw-Hill Education (Asia), 2011.
- Ramesh S. Gaonkar. Fundamentals of Microcontrollers and Applications in Embedded Systems. Thomson Delmar Learning, 2007.