ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΜΣ Κυβερνοασφάλεια και Επιστήμη Δεδομένων
MSc Cybersecurity and Data Science
Dept of Informatics
University of Piraeus

EMBEDDED SYSTEMS

INTRODUCTION

Mihalis Psarakis and Dimitris Agiakatsikas

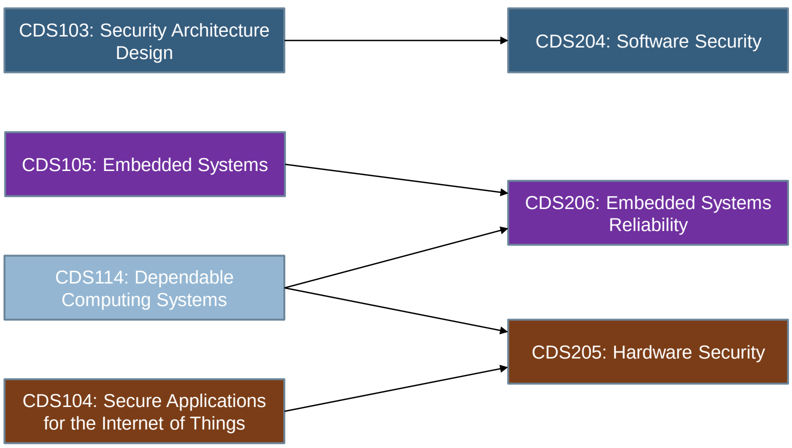
Outline

2

- Introduction to Embedded Systems
- Embedded computing platforms (CPUs, FPGAs, ASICs)
  - Course CDS104 (Secure Applications for the Internet of Things): focus on CPUs
  - Course CDS105 (Embedded Systems): focus on FPGAs
- Design embedded systems with FPGAs
  - VHDL Basics (4 lectures)
  - FPGA architecture (1 lecture)
  - FPGA design flow with VHDL (3-4 lab courses)
  - RISC-V processor (2 lab courses)
- Labs: Design, simulation and validation of FPGA designs.
  - Use of EDA tools (Xilinx Vivado) and
  - FPGA development boards (Zybo boards).
- Project: Implementation of an embedded application on FPGA

Role of the course in the track

3



What is an embedded system?

4

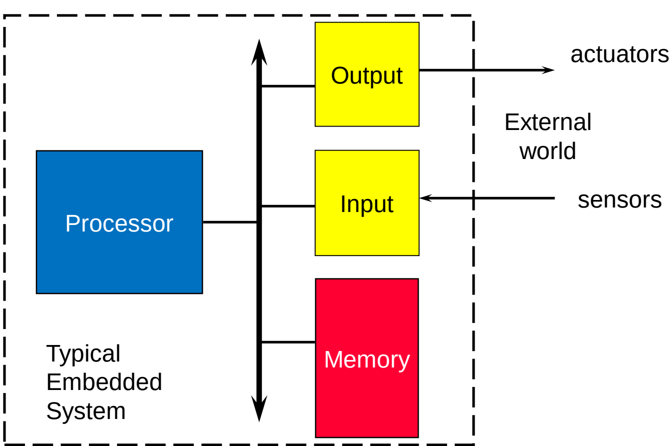
- Several definitions:
  1. "a digital system that provides service as part of a larger system" – G. De Micheli
  2. "any device that includes a programmable computer but is not itself a general-purpose computer" – M.Wolf
  3. "a less visible computer" - E. Lee
  4. "a single-functioned, tightly constrained, reactive computing system" – F. Vahid
  5. "a computer system that has a dedicated function within a larger mechanical or electronic system." – Wikipedia

Where are the embedded systems?

5

Typical structure

6



Embedded processor types

7

- Computational microprocessors (32- or 64-bit datapaths)
  - CPU of workstations, PCs, or high-end portable devices (PDAs)
  - x86, PowerPC, SPARC, etc.
- Embedded general purpose microprocessors (32-bit datapaths)
  - Designed for a wide range of embedded applications
  - Often scaled-down version of computational microprocessors
  - ARM, PowerPC, MIPS, x86, 68K, etc.
- Microcontrollers (4-, 8-, or 16-bit datapaths)
  - Integrate processing unit, memory, I/O buses, and peripherals
  - Often low-cost, high-volume devices
- Domain-specific processors (datapath size varies greatly)
  - Designed for a particular application domain
  - Digital signal processors, multimedia processors, graphics processors, network processors, security processors, etc.

Design constraints

8

- Small size, low weight
  - Portable devices
  - Transportation applications
- Low power
  - Autonomous for many hours (laptops last only 2-3 hours)
- Harsh environment
  - Heat, vibration, shock
  - Power fluctuations, RF interference
  - Radiation
- Safety-critical operation
  - Must NOT function incorrectly
- Cost sensitivity
  - Small difference in cost adds up over several millions units

Design metrics

9

- NRE cost (Non-Recurring Engineering cost):
  - The one-time cost of designing the system
- Unit cost:
  - the cost of manufacturing each copy of the system, excluding NRE cost
- 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 metrics

10

- 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
- Dependability
  - Reliability/Availability
  - Safety
  - Security

Real-time operation

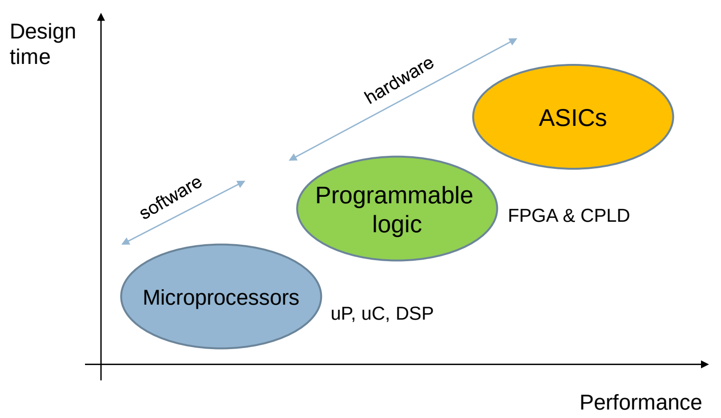
- Correct operation of real-time systems means:
  - Working correctly (functionally correct) but also producing outputs **in time!**
    - i.e., correct result at the right time
- Hard real time: System designed to meet all deadlines
  - A missed deadline is a design flaw
  - Example: ABS brake, nuclear reactor monitoring system
  - System hardware (over) designed for worst-case performance
- Firm real-time: System designed to meet all deadlines, but occasional missed deadline is allowed
  - Example: multimedia systems
  - System hardware designed for average case performance
- Soft real time: System designed to meet as many deadlines as possible
  - Best effort to complete within specified time, but may be late
  - Example: network switch or router
  - System hardware designed for average case performance

Software vs hardware

- Many functions can be done by **software** on a general purpose microprocessor OR by **hardware** on an application specific ICs (ASICs)
  - Example: game console graphic, industrial control
  - Leads to Hardware/Software Co-design concept
- Where to place functionality?
  - Dilemma: A Sort algorithm
    - faster in hardware, but more expensive.
    - more flexible in software but slower.
- Must be able to explore these various trade-offs:
  - Cost
  - Speed
  - Reliability
  - Form (size, weight, and power constraints.)

Software vs hardware

13



What Are FPGAs

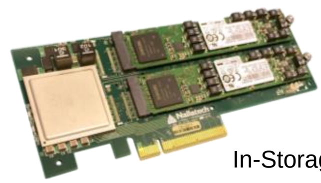
14

- ***Field-Programmable*** Gate Array
- Can be configured to act like any circuit
- Can do many things, but mostly used for acceleration

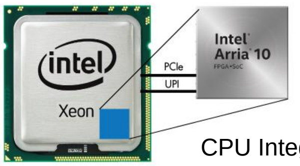
FPGAs Come In Many Forms

15

PCIe-Attached



In-Storage



CPU Integrated

In-Network

How Is It Different From CPU/GPUs

16

- GPU – The other major accelerator
- CPU/GPU hardware is fixed
  - “General purpose”
  - we write programs (sequence of instructions) for them
- FPGA hardware is not fixed
  - “Special purpose”
  - Hardware can be whatever we want
  - Will our hardware require/support software? Maybe!
- Optimized hardware is very efficient
  - GPU-level performance
  - 10x power efficiency

Parallel execution in custom hardware

17

- Example -- Calculating gravitational force: $\frac{G \times m_1 \times m_2}{(x_1 - x_2)^2 + (y_1 - y_2)^2}$
- 8 instructions on a CPU → 8 cycles**
- Much fewer cycles on a special purpose circuit

$A = G \times m_1$ | $C = x_1 - x_2$ | $E = y_1 - y_2$

$B = A \times m_2$ | $D = C^2$ | $F = E^2$

$G = D + F$

$Res = B / G$

4 cycles with basic operations

$A = G \times m_1 \times m_2$ | $B = (x_1 - x_2)^2$ | $C = (y_1 - y_2)^2$

$D = B + C$

$Res = B / G$

3 cycles with compound operations

May slow down clock

$Res = (G \times m_1 \times m_2) / ((x_1 - x_2)^2 + (y_1 - y_2)^2)$

1 cycle with even further compound operations

How Is It Different From ASICs

18

- ASIC (Application-Specific Integrated Circuit)
  - Special chip purpose-built for an application
  - E.g., ASIC bitcoin miner, Intel neural network accelerator
  - Function cannot be changed once expensively built
- \+ FPGAs can be ***field-programmed***
  - Function can be changed completely whenever
  - FPGA fabric ***emulates*** custom circuits
- \- Emulated circuits are not as efficient as bare-metal
  - ~10x performance (larger circuits, faster clock)
  - ~10x power efficiency