ECE307 Digital Systems Design Lab Introduction to Embedded Systems

Electronic and Communication Engineering Department

Advanced Embedded Systems: Lecture 1 (Date: Jan 19, 2025)

Dr. Wahhab R. Mousa

Communication Channels

- Instructor’s email: WAHHAB.ALBAZRQA@uokufa.edu.iq
- Course Website: www.github.com/myreadings1/ECE307/
- Telegram Group for students:

Contents

- Communication channels in this course
- Course objectives, Structure, and Grading system
- Motivations
- Embedded system features and specifications
- Embedded system design Introduction

Contents

- Course objectives, Structure, and Grading system
- Motivations
- Embedded system features and specifications
- Embedded system design Introduction

Course Goals

- Introduction to Embedded systems
- Understand different design methodologies and metrics
- Software Design: Task Scheduling, ARM Assembly language.
- Hardware Design: FPGAs, Hardware Description Language (VHDL)
- Design embedded control system for different applications

Grading Policy

- Homeworks
- Project and Participation
- Attendance
- Exams (Quizzes, Midterm)
- Final Exam

Reading Material

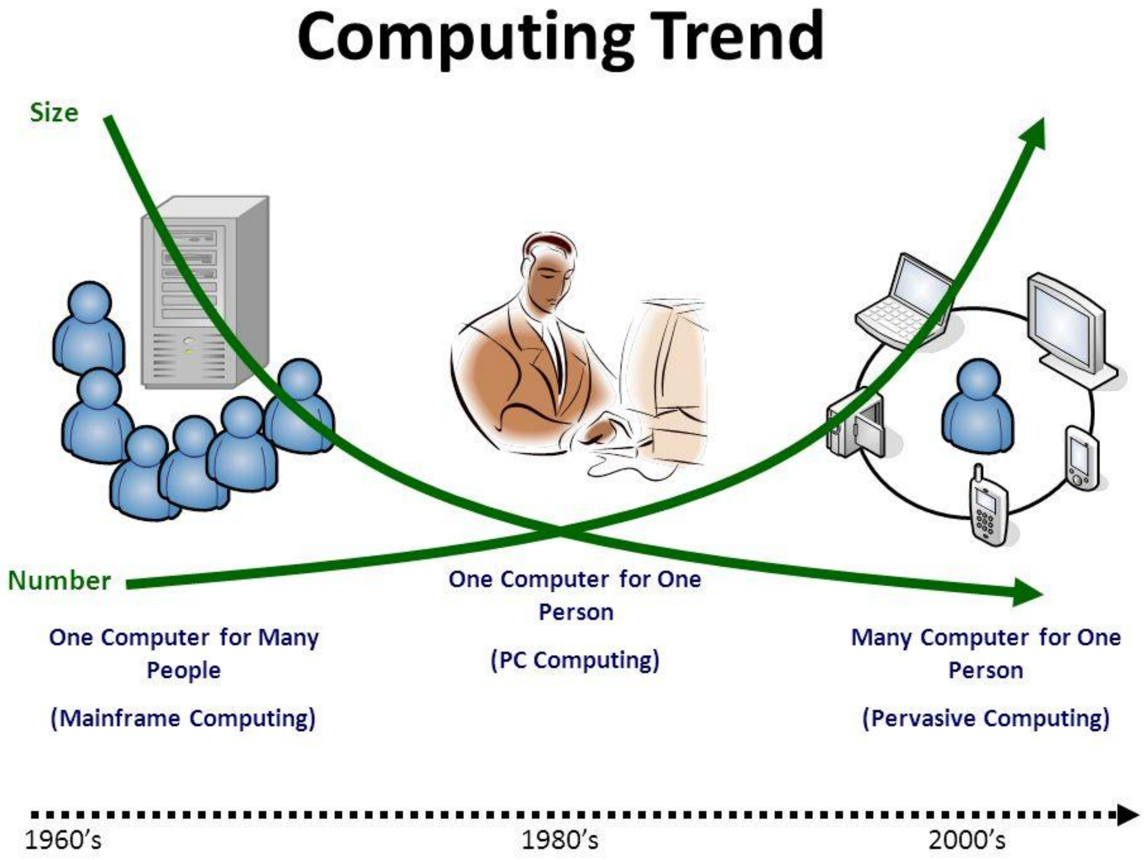
- **Computer Architecture A Quantitative Approach** *5th Edition* John L. Hennessy David A. Patterson
- **Circuit Design and Simulation with VHDL** 2nd edition by Volnei A. Pedroni
- **Embedded Design Handbook** by Intel.
- **Introduction to Embedded Systems: A Cyber-Physical Systems Approach** By Edward Ashford Lee and Sanjit Arunkumar Seshia
- **Embedded Systems Fundamentals with ARM Cortex-M based Microcontrollers: A Practical Approach** by Alexander G. Dean
- Any Embedded systems design book you find!

Contents

- Course objectives, Structure, and Grading system
- Motivations
- Embedded system features and specifications
- Embedded system design Introduction

Motivations

- Market demands
- Amazing growth
- Internet age



Computing Trend

Size

Number

One Computer for Many People

(Mainframe Computing)

One Computer for One Person

(PC Computing)

Many Computer for One Person

(Pervasive Computing)

1960's

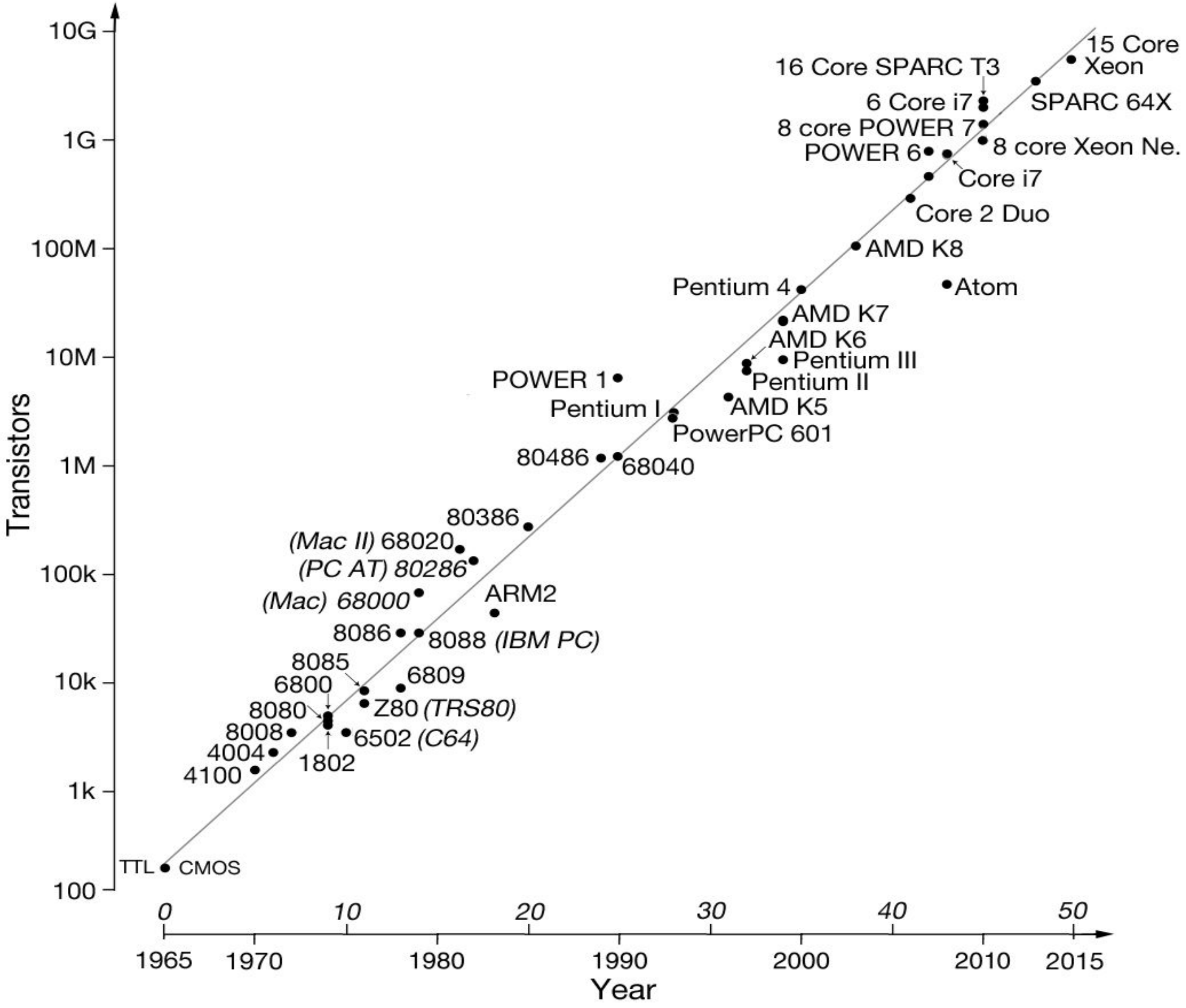
1980's

2000's

Motivations

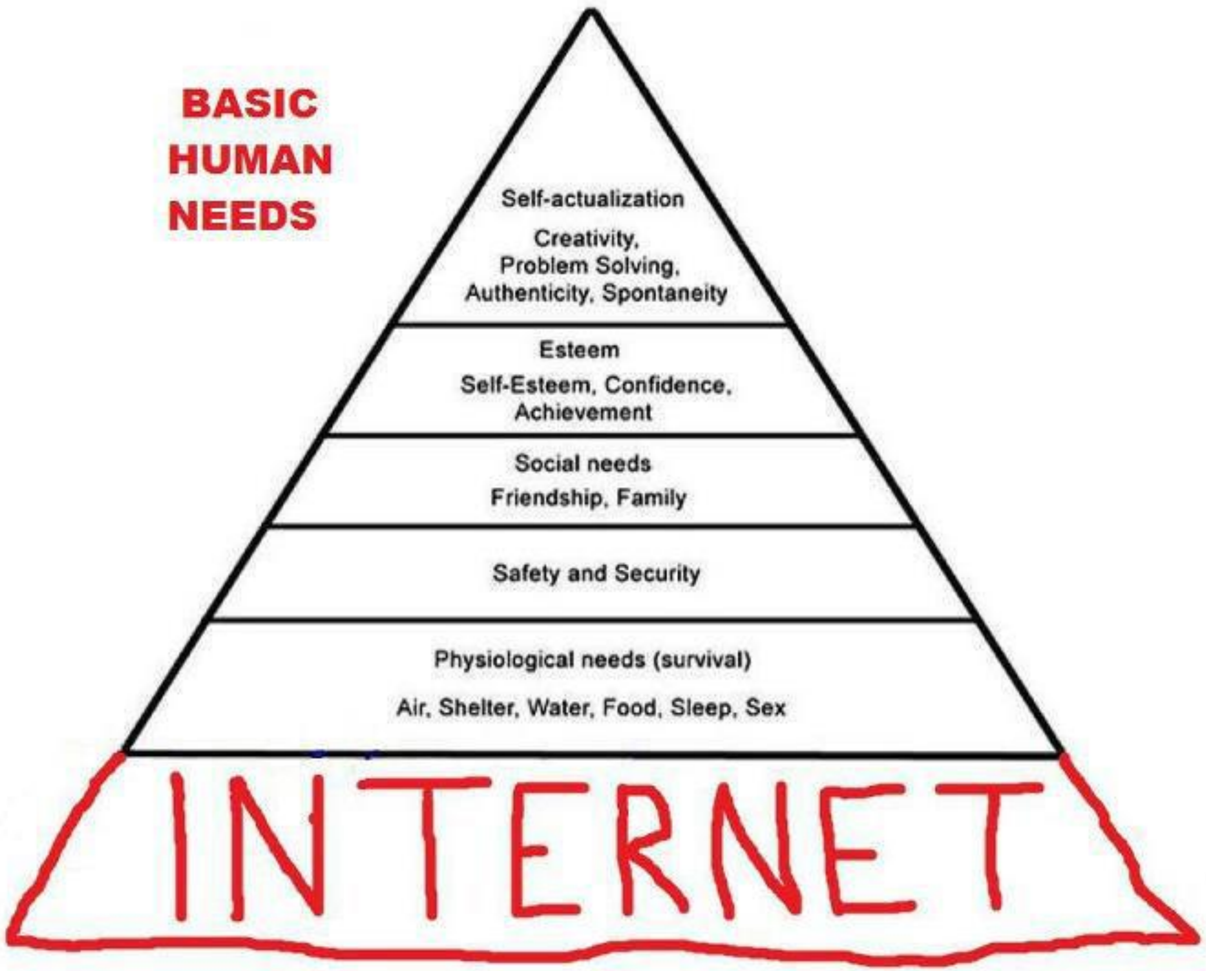
- Market demands
- Amazing growth
- Internet age

Moore's law



Motivations

- Market demands
- Amazing growth
- Internet age



Contents

- Course objectives, Structure, and Grading system
- Motivations
- Embedded system features and specifications
- Embedded system design Introduction

EMBEDDED SYSTEMS

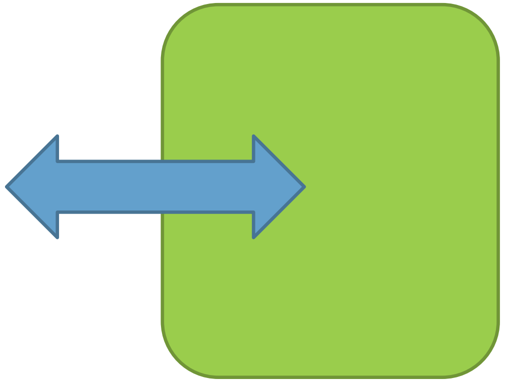
- What is an embedded system?
  - Application specific computer system
  - Built into a larger system
- Why add an embedded system to a larger system?
  - Better performance: make it possible to provide more sophisticated control
  - More functions and features: not possible with other approaches
  - Lower cost: components cost, manufacturing cost, maintenance cost, operation cost
  - Better dependability: adaptive systems to failure, easy to diagnose failure

TERMINOLOGY

- **Integrated circuit (IC):** Electronic circuit with components built into a single piece of silicon, enabling extreme miniaturization, mass production, and cost reduction
- **Microcontroller unit (MCU):** Integrated circuit containing CPU, peripherals, support circuits, and often memory
- **Central processing unit (CPU):** Hardware circuit that executes a program's instructions
- **Instruction:** Command for processor to execute. Consists of an operation and zero or more operands.

INTERFACING

- Many of the external devices use *analog* signals in which the voltage (or current) can take on a continuous range of values (ADC/DAC)
- Weak signals need to be amplified
- High-voltage signals need to be scaled down to safe levels
- Noise must be filtered out



RESPONSIVENESS

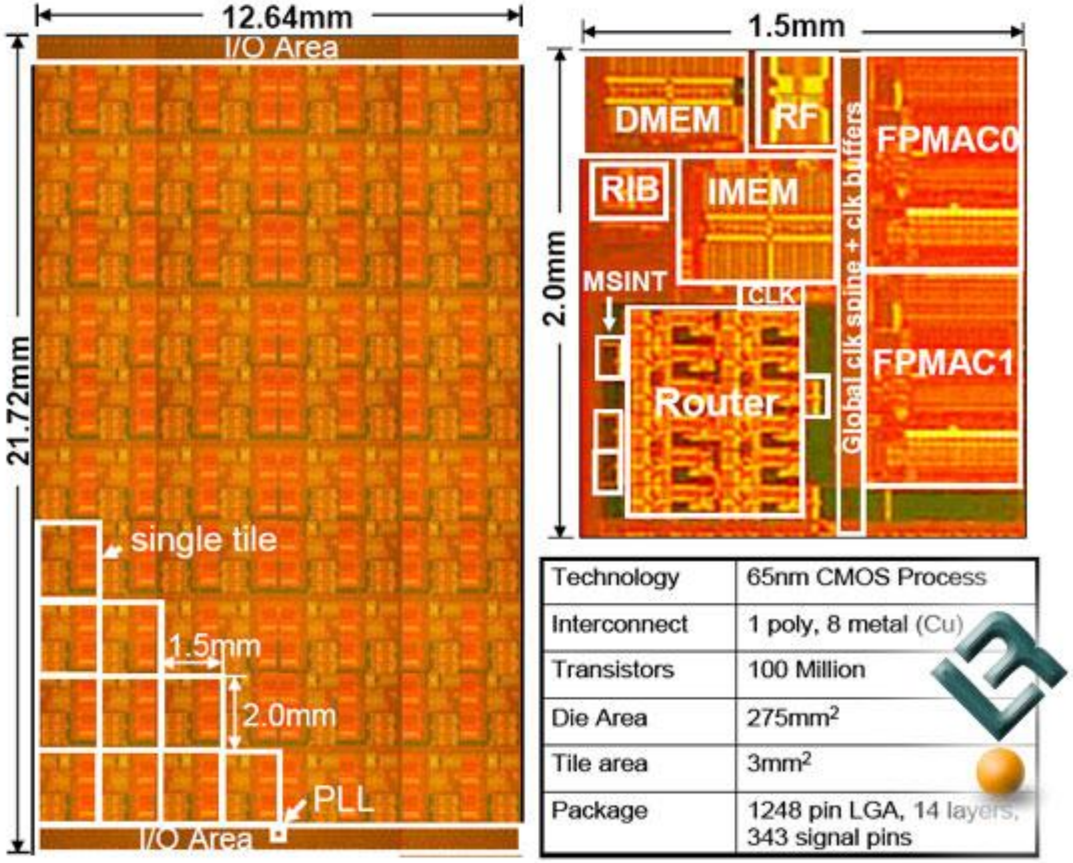
- One challenge for embedded system developers is providing enough responsiveness
  - Raw processing speed
  - Task scheduling: most critical
  - Multiply hardware: multi-core, multi control units

CONCURRENCY VS. PARALLELISM

- Embedded controllers must typically manage multiple activities concurrently, often with precise control of the timing
- Adding more features to a system increases the software's complexity
- **Parallelism**: when different parts of the program physically execute simultaneously.
- **Concurrency**: when different parts of the program conceptually execute simultaneously

INTEL MANY INTEGRATED CORE (MIC)

- **Tera Flops (2008):**
  - **a. 80 core**
  - **b. 10x8 mesh NoC**
  - **c. 100W**
  - **d. 100M transistors**
  - **e. 4GHz**



Technology	65nm CMOS Process
Interconnect	1 poly, 8 metal (Cu)
Transistors	100 Million
Die Area	275mm^2
Tile area	3mm^2
Package	1248 pin LGA, 14 layers, 343 signal pins

MORE SYSTEM FEATURES

- Diagnostics: Some systems require an ability to find and repair faulty components quickly and easily.
- Constraints: Limit the designer options (cost, energy, power, size, weight, environment)
- Security: protected against harm caused by attacks originating from outside the system.

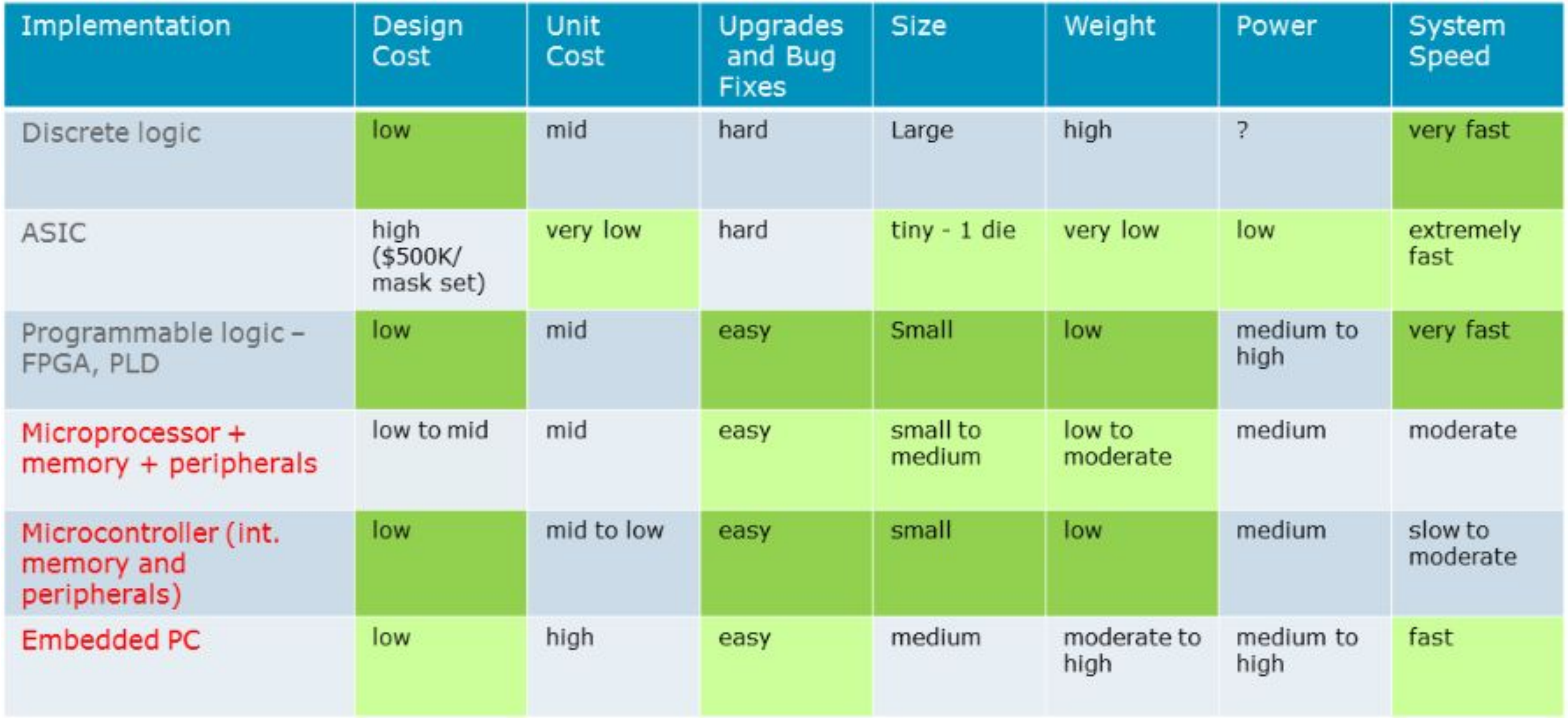
Contents

- Course objectives, Structure, and Grading system
- Motivations
- Embedded system features and specifications
- Embedded system design Introduction

DESIGN OPTIONS

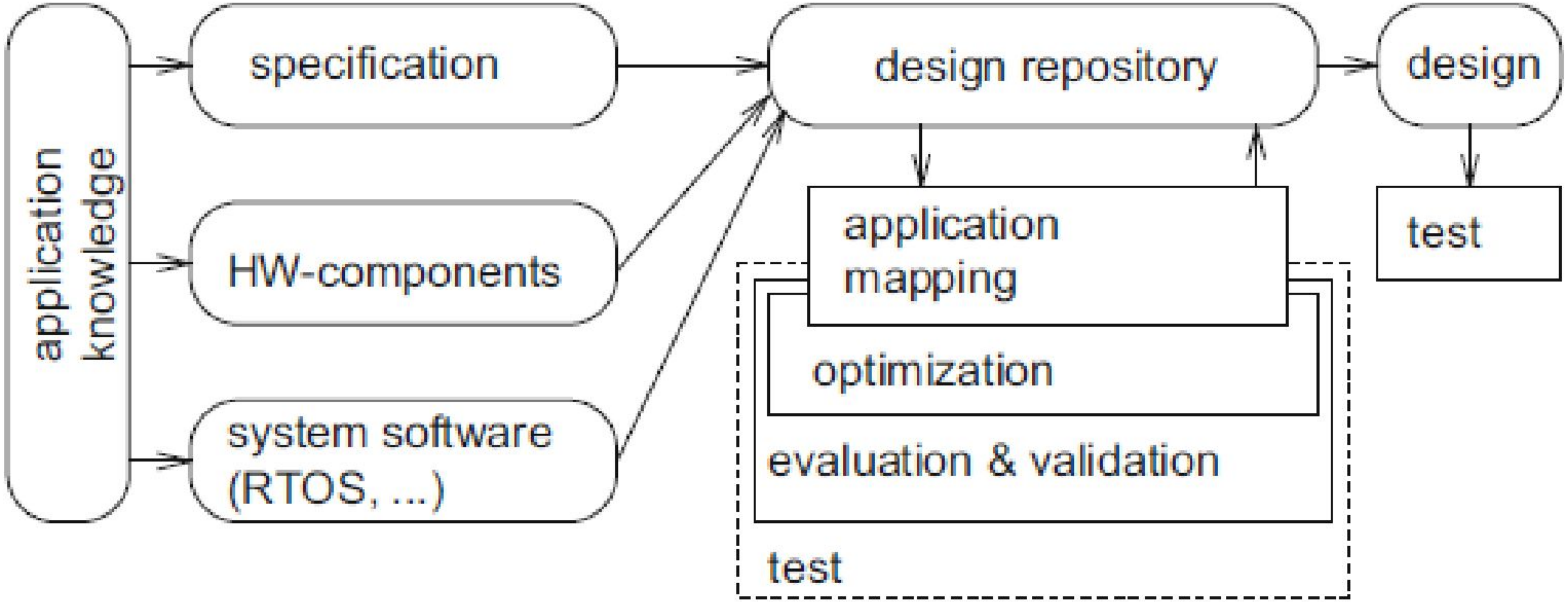
Dedicated hardware

Software running on generic hardware



Implementation	Design Cost	Unit Cost	Upgrades and Bug Fixes	Size	Weight	Power	System Speed
Discrete logic	low	mid	hard	Large	high	?	very fast
ASIC	high ($500K/ mask set)	very low	hard	tiny - 1 die	very low	low	extremely fast
Programmable logic – FPGA, PLD	low	mid	easy	Small	low	medium to high	very fast
Microprocessor + memory + peripherals	low to mid	mid	easy	small to medium	low to moderate	medium	moderate
Microcontroller (int. memory and peripherals)	low	mid to low	easy	small	low	medium	slow to moderate
Embedded PC	low	high	easy	medium	moderate to high	medium to high	fast

SYSTEM DESIGN

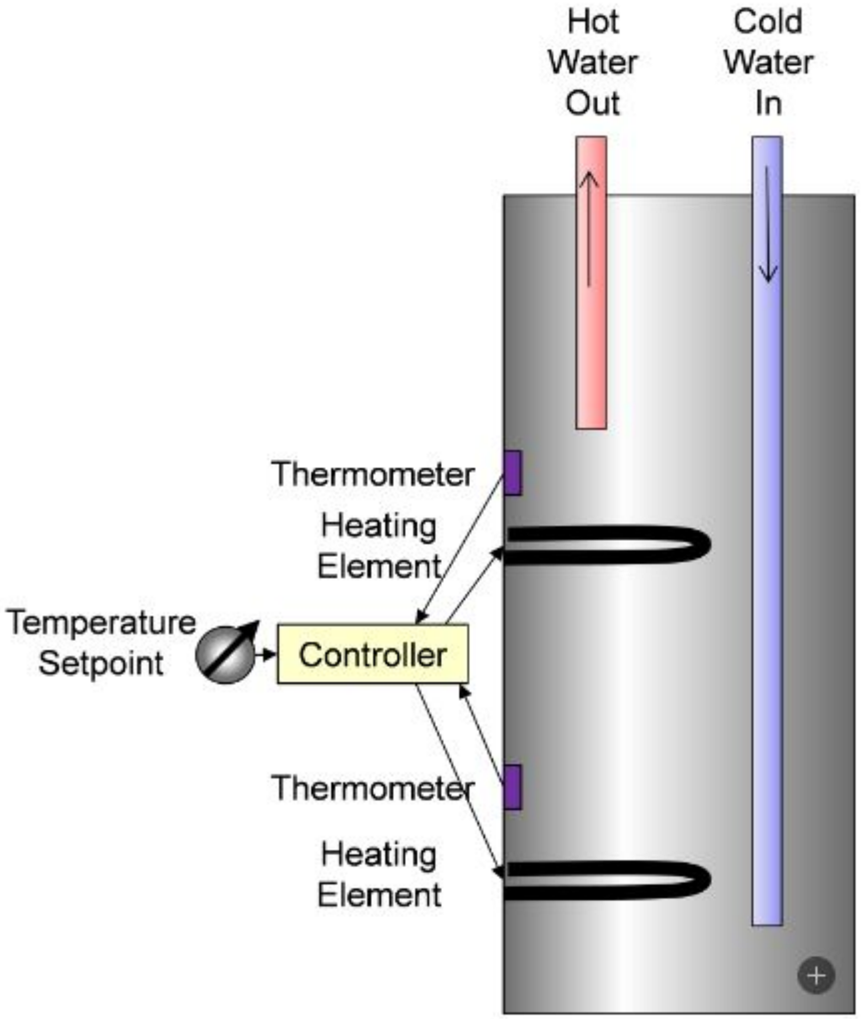


EX: SMART BIKE

- Function: Measure speed and distance
- Constrains: Size, Cost, power, weight
- Input: Wheel rotation indicator, mode key
- Output: Liquid crystal display

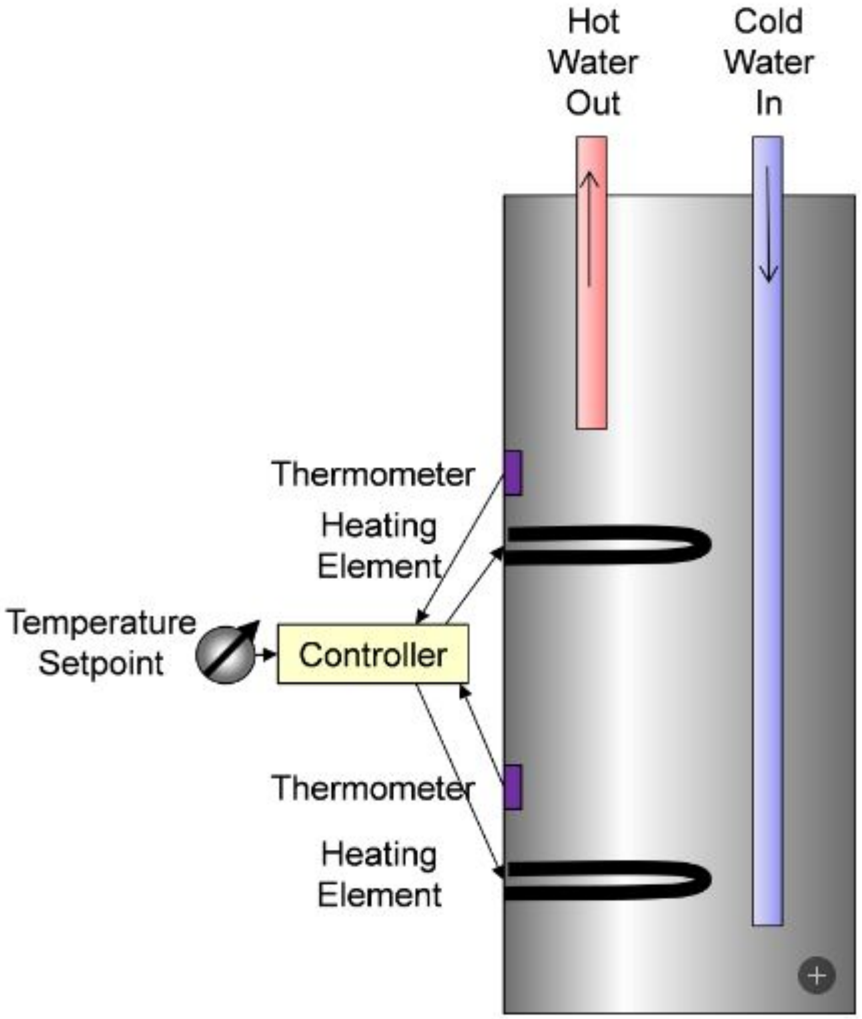
EX2: SMART WATER HEATER

- What are the input devices?
- What are the outputs?
- Name at least two safety features to include, and specify what hardware and/or software is needed for each?
- Describe a useful feature that you can add in the software without requiring additional hardware.



EX2: SMART WATER HEATER

- Describe two useful energy-saving features that you can add in the software if the controller could keep track of the time of day?
- Describe a useful feature that you can add in the software if the water heater included an Internet connection?



THANK YOU
Questions?