LUND UNIVERSITY

LTH FACULTY OF ENGINEERING

# Design of Embedded Systems

Krzysztof Kuchcinski
krzysztof.kuchcinski@cs.lth.se

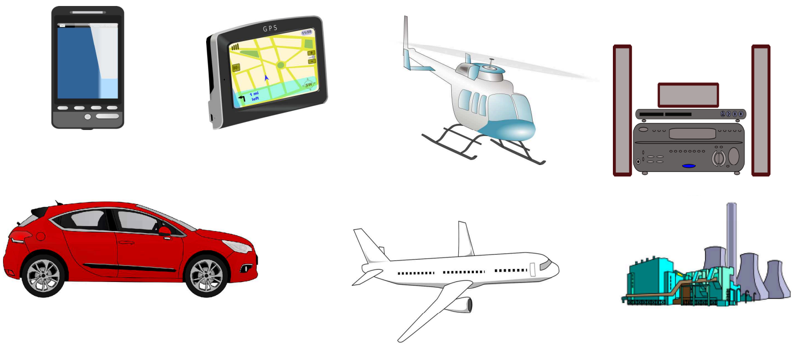
## Outline

- Introduction
- Course Organization
- General introduction, definition of the field
- Embedded Systems Examples
- Embedded Systems Design Methodologies



## Introduction

# Examples of Embedded Systems





# Examples of Embedded Systems (cont'd)

- Anti-lock brakes
- Auto-focus cameras
- Automatic teller machines
- Automatic toll systems
- Automatic transmission
- Avionic systems
- Battery chargers
- Camcorders
- Cell phones
- Cell-phone base stations
- Cordless phones
- Cruise control
- Curbside check-in systems
- Digital cameras
- Disk drives
- Electronic card readers
- Electronic instruments
- Electronic toys/games
- Factory control
- Fax machines
- Fingerprint identifiers
- Home security systems
- Life-support systems
- Medical testing systems
- Modems
- MPEG decoders
- Network cards
- Network switches/routers
- On-board navigation
- Pagers
- Photocopiers
- Point-of-sale systems
- Portable video games
- Printers
- Satellite phones
- Scanners
- Smart ovens/dishwashers
- Speech recognizers
- Stereo systems
- Teleconferencing systems
- Televisions
- Temperature controllers
- Theft tracking systems
- TV set-top boxes
- VCR's, DVD players
- Video game consoles
- Video phones
- Washers and dryers

Source: Embedded Systems Design: A Unified Hardware/Software Introduction, (c) 2000 Vahid/Givargis



# Course Organization

## Course Organization

- Twelve **lectures**.
- Four **lab assignments** plus an introductory lab.
  - carried out in groups of two students
  - use special software and hardware (lab E:4115 and E:4119)
  - all labbs postponed; more information later
  - lab responsible: Flavius Gruian



## Course Organization (cont'd)

- **Seminars**
  - 27/05, ~~8:15 and 10:15 in E:2116~~ probably Zoom meetings at 10:15
  - 29/05, ~~8:15 and 10:15 in E:2116~~ probably Zoom meetings at 10:15
- Obligatory **examinations**
  - 1 June 2020 at 14-19, MA:8A, 8B
  - 25 August 2020 at 14-19, E:2116
- Books:
  - Peter Marwedel, Embedded System Design, 2nd Edition, Springer, 2011 (ISBN: 978-94-007-0256-1)
    or
    3rd Edition, Springer, 2018 (ISBN: 978-3-319-56045-8)



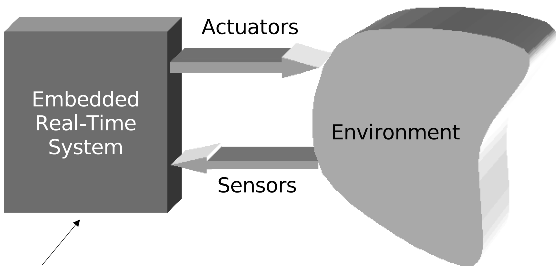
## Lectures
## Preliminary Schedule

| Date | Content |
|---|---|
| 20-03-23 | Introduction, motivation, etc. |
| 20-03-27 | Design methodology (HW/SW co-design, etc) |
| 20-03-30 | VHDL introduction |
| 20-04-03 | VHDL for synthesis |
| 20-04-06 | Computational models |
| 20-04-27 | Design representations |
| 20-05-04 | System partitioning |
| 20-05-08 | Allocation, assignment |
| 20-05-11 | and scheduling |
| 20-05-15 | Communication synthesis |
| 20-05-18 | Testability |
| 20-05-25 | Low-power design \| ARM presentation |

# General introduction, definition of the field

## Embedded Systems

"A device that includes a programmable computer but is not itself a general-purpose computer"




Execution deadlines,
Power and energy consumption constraints,
:



## Embedded Systems (cont'd)

- Computing systems embedded within electronic devices
- Hard to define. Nearly any computing system other than a desktop computer
- Billions of units produced yearly, versus millions of desktop units
- Perhaps 50 per household and per automobile

Source: Embedded Systems Design: A Unified Hardware/Software Introduction, (c) 2000 Vahid/Givargis

## Embedded Systems (cont'd)

- Non User-Programmable.
- Based on programmable components (e.g., Micro-controllers, DSP's...) but often contain application specific hardware (IC's, ASIC's).
- Reactive Real-Time Systems:
  - React to external environment,
  - Maintain permanent interaction,
  - Ideally never terminate,
  - Are subject to external timing constraints (real-time).



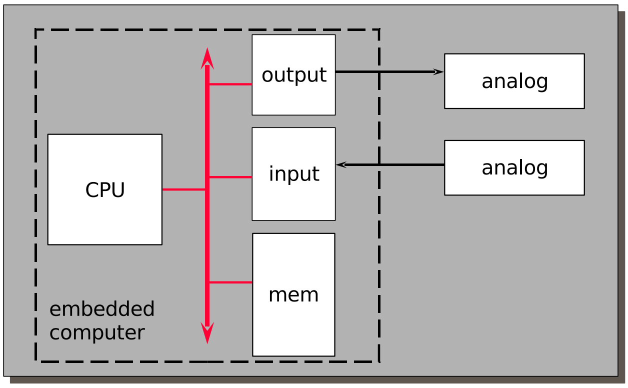
## Characteristics Embedded Systems

- Sophisticated functionality.
- Real-time operation.
- Low manufacturing cost.
- Low power.
- Designed to tight deadlines by small teams.
- "Resource conscious" vs. "Unlimited resources" programming



## Embedded Systems Examples

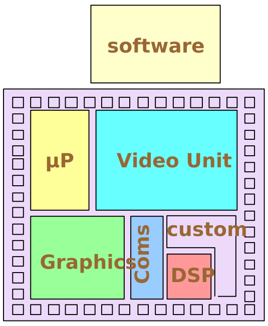
# A Single Processor Embedded System




Source: W. Wolf, Computers as Components: Principles of Embedded Computing Systems Design



# SoC Embedded System

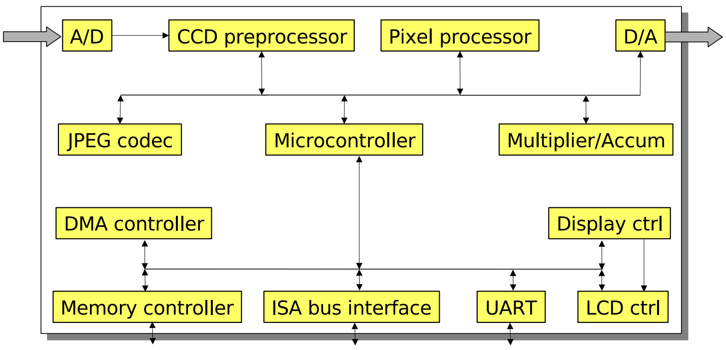



- Assembly of "prefabricated component" often purchased from external vendors ("IP")
  - "black box" hierarchy
- Design & Verification at the System level
  - rather than the logic level
  - Interface and communication
- Great Importance of Software

Source: Alberto Sangiovanni-Vincentelli, 35th DAC

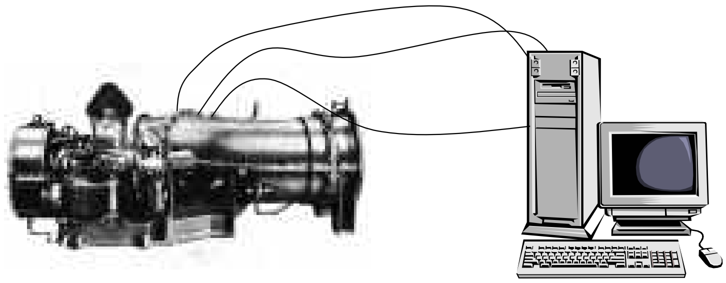


# A Digital Camera Example




Source: F. Vahid and T. Givargis, Embedded System Design: A Unified Hardware Software Approach

# Real-time gas turbine testing system

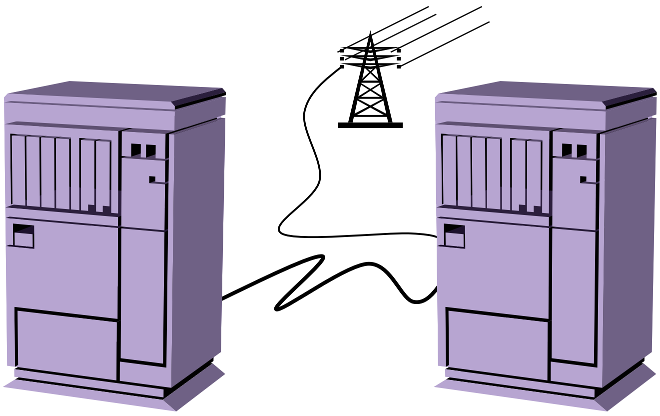


MI-2 helicopter engine

"Minicomputer" 8kB RAM
cassette tape



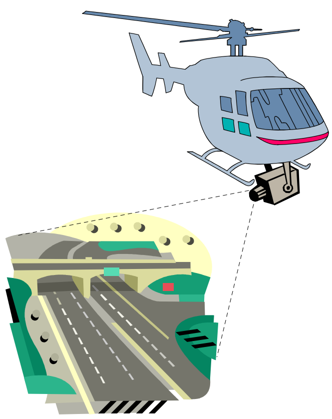
# TELEX-I and TELEX-II systems



PDP-11 PDP-11



# WITAS project



- Autonomous system.
- Real-time system.
- Image processing.
- Mission planning.
- Incorporation of GIS systems.
- Interface with ground operator.
- ...

http://www.ida.liu.se/ext/witas

# Typical Hardware Components of DSP System

| Component class | Implements | Compiler | Specification |
|---|---|---|---|
| DSP processor | Low data-rate DSP<br>Slow control loops<br>Appl. Spec. alg. | (Retargetable)<br>code generator<br>High level synth. | Assembly<br>C |
| Microcontroller | User interface<br>Slow control loops | C compiler | C |
| Hardware accelerator | High data-rate DSP<br>RT level synth. | High level synth. | C, VHDL<br>Verilog |
| Communication blocks and memory | Internal & external communication<br>Storage & buffering | Memory mgmt.<br>(A)synchronous interface synth. | Data-sheets |
| Others | Usually FSMD's<br>- clock generators<br>- DMA blocks | RT level synth.<br>Asynchronous synth. | VHDL |

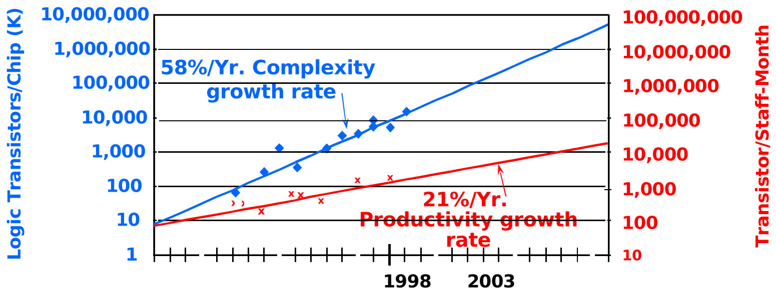


# Embedded Systems Design Methodologies

# Importance of Embedded System Design Methodologies

- Hardware complexity.
- Heterogeneous systems containing hardware (both digital and analog) and software.
- Heterogeneous components (CPU's, DSP's, ASIC's, buses, point-to-point links, etc.).
- Heterogeneous requirements - performance, cost, power consumption, etc.
- System-on-chip.
- Shorter design cycles required by time-to-market constraints.
- ...

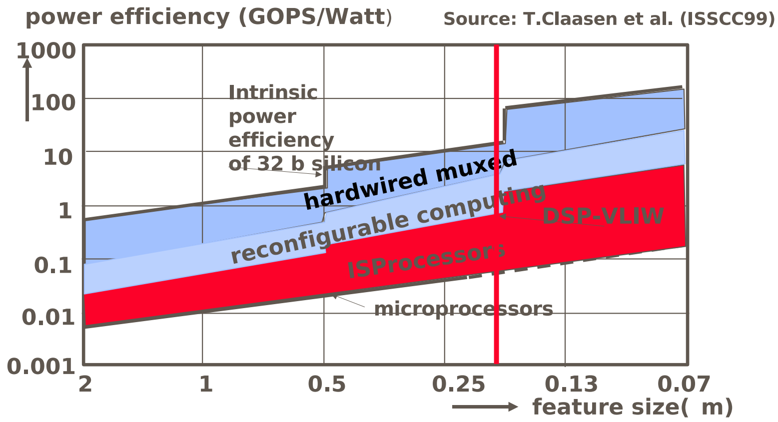
## Design Complexity and Designer Productivity Gap




Source: Bryan Preas, Xerox PARC, 35th DAC

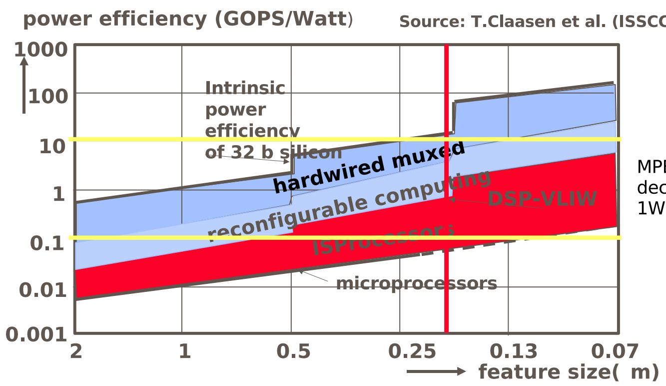


## Flexibility and Energy Efficiency



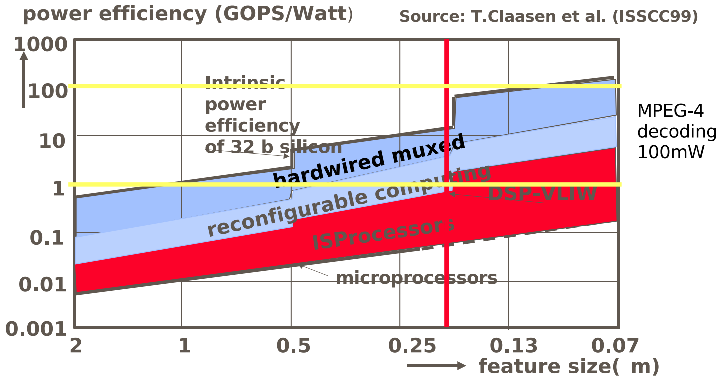



## Flexibility and Energy Efficiency



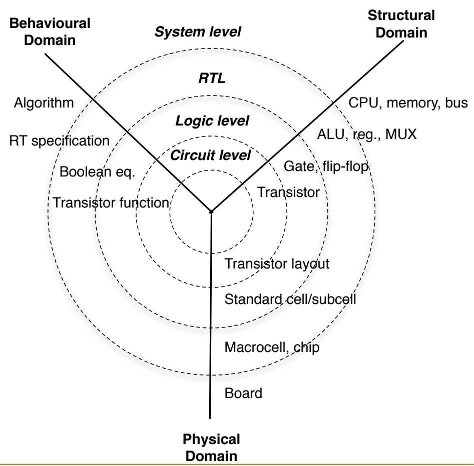

MPEG-4 decoding 1W

# Flexibility and Energy Efficiency






# Design Domains and Abstraction Levels



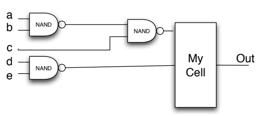



# Design Domains

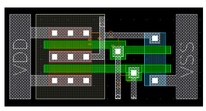
- behavioral representations - describe only circuit's function, for example

```
if clock=high then counter:= counter+1
```

- structural representations - components and their interconnections, for example




- physical representations - either a geometrical layout or a topological constraint.



## Software vs. Hardware Design short summary

- Software
  - flexibility,
  - reconfigurability, easy update, etc.,
  - complex functionality,
  - cost,
  - ...
- Hardware
  - speed,
  - power consumption,
  - cost in large volumes,
  - ...



## Design of Embedded Systems

- Need to be done using high-level specification, programming and hardware description languages - not assembly languages and gate/transistor level design.
- Requires efficient design space exploration and synthesis/compilation tools.
- Different design requirements has to be taken into account, e.g., cost, performance, testability, quality of service, power consumption.
- Multi-language design framework.



## Importance of High-Level Design Methods

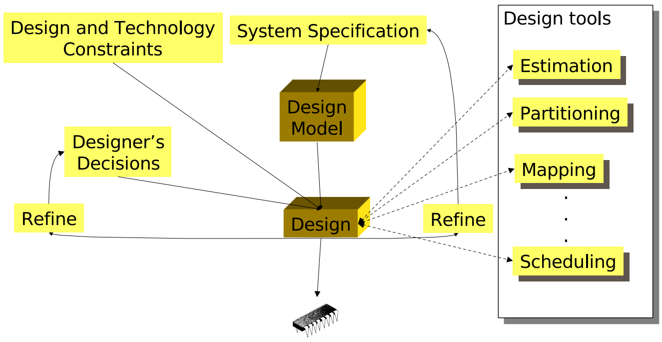
System Verification Processing Speeds

| System implementation | Processing time (s/frame) |
|---|---|
| Behavioral model | 1 200 (20 min/frame) |
| RTL model | 144 000 (1.6 days/frame) |
| Gate model | 228 000 (2.6 days/frame) |
| Gate model on hardware accelerator | 1 200 |
| Rapid prototype | 0.5 |
| Target hardware | 0.05 |

Source: Paul Clemente, Ron Crevier, Peter Runstadler "RTL and Behavioral Synthesis, A Case Study", VHDL Times, vol. 5, no. 1.

# General Design Flow

Specify-Explore-Refine






# Specification and Programming

- Specification languages, such as UML, SDL.
- Programming languages, such as C, C++, Java, Esterel, assembly languages.
- Hardware description languages, such as VHDL, Verilog, SystemC.

**Example:** combining SystemC and C++ gives a unified simulation environment for hardware and software.



# Hardware Description Languages

- Cover several levels of design abstraction as well as behavioral and structural description domain.
- Contain typical features of programming languages, such as data types and program statements.
- Special features:
  - time concept,
  - structure description,
  - parallelism.
- VHDL (IEEE standard), Verilog, SystemC.

## Design Representations (Computational Model)

- Used to represent/model digital systems under design.
- Generated by a compiler from system specification or coded directly in the model.
- Represent the semantics, structure and timing of the system.
- Usually based on some kind of annotated graph representation.
- Used internally by design automation systems or by the modeler/designer.



## Design – Synthesis

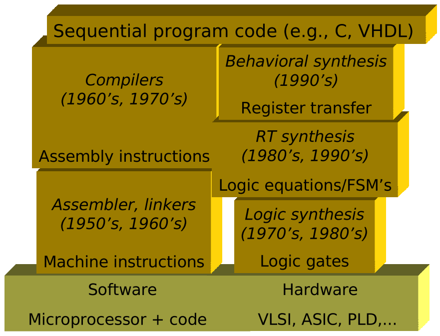
- Software is translated into target code for a processor.
- Real-time operating system might be used.
- Hardware synthesis – translation of a behavioral representation of a design into a structural one.
- Communication synthesis – generates hardware and software which interconnects system components.



## Hardware Synthesis Levels

- System level – accepts as an input specification in a form of communicating concurrent processes. The synthesis task is to generate the general system structure defined by processors, ASIC's, buses, etc.
- High level – the input specification is given as a behavioral level description of an algorithm describing the functionality of a design and generates implementation at RT-level. Basic synthesis steps are scheduling, allocation and binding.
- Logic level – it can be divided into combinational and sequential logic synthesis.
- Physical design – it accepts a gate-level netlist and produces final implementation of the design in a given technology.

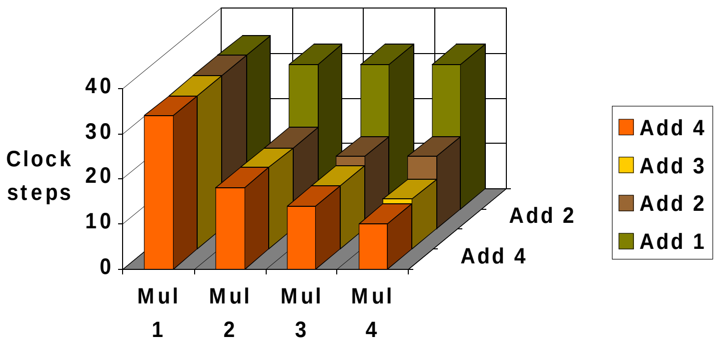
# Hardware/Software Co-design enabling technologies




Source: F. Vahid and T. Givargis, Embedded System Design: A Unified Hardware Software Approach

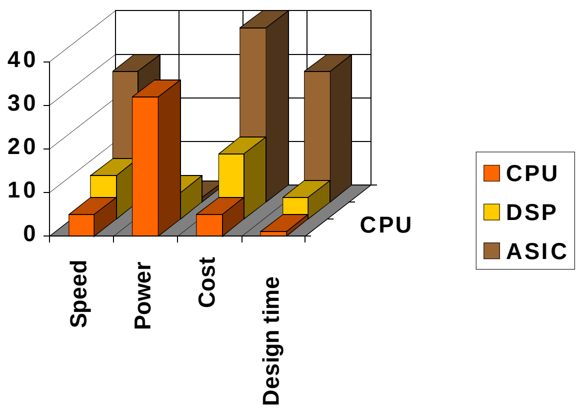


# Discrete Cosine Transform Partial Design Space

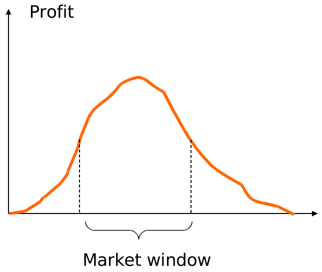





# Design Space Exploration



## Time-to-market constraint




- need time for new product development,
- the biggest profit is in the market window time,
- missing the market window can be costly



## Summary

- Embedded systems are important class of electronic systems which can be found everywhere,
- Combine hardware and software solutions,
- Cover several engineering and research areas:
  - microelectronics,
  - real-time systems,
  - software development,
  - etc.
- Need careful design which optimizes different design parameters.