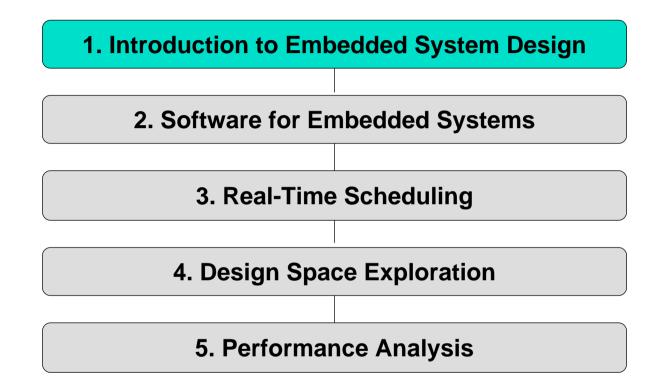
#### 1. Introduction to Embedded System Design

#### © Lothar Thiele ETH Zurich, Switzerland





## **Contents of Lectures (Lothar Thiele)**



The slides contain material from the "Embedded System Design" Book and Lecture of Peter Marwedel and from the "Hard Real-Time Computing Systems" Book of Giorgio Buttazzo.



## Topics

- General Introduction to Embedded Systems
- Hardware Platforms and Components
  - System Specialization
  - Application Specific Instruction Sets
    - Micro Controller
    - Digital Signal Processors and VLIW
  - Programmable Hardware
  - ASICs
  - System-on-Chip



### **Embedded Systems**

Embedded systems (ES) = information processing systems embedded into a larger product

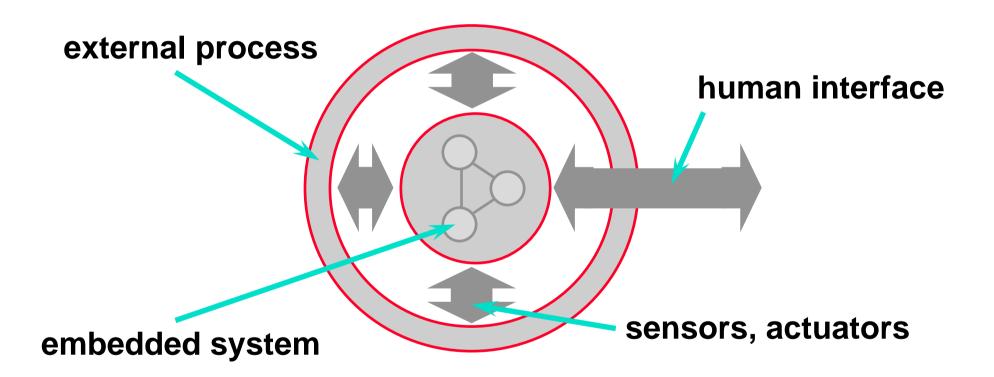


Main reason for buying is not information processing

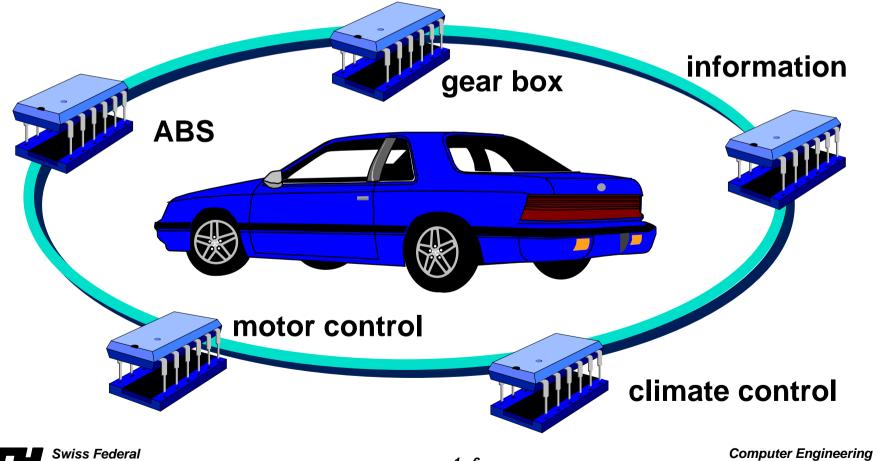




#### **Embedded Systems**



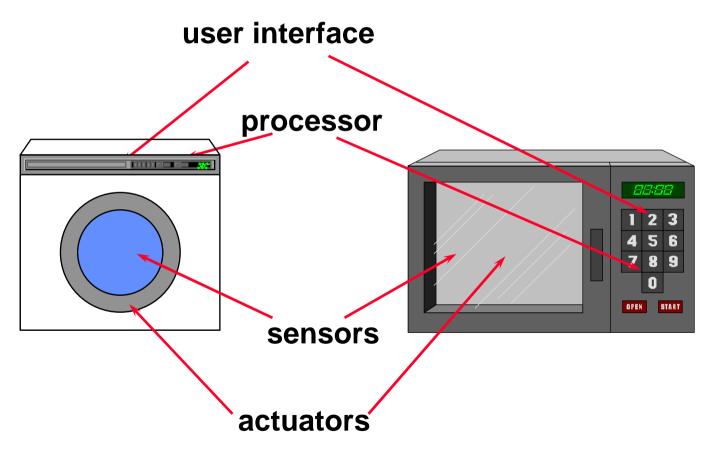
Car as an integrated control-, communication and information system.

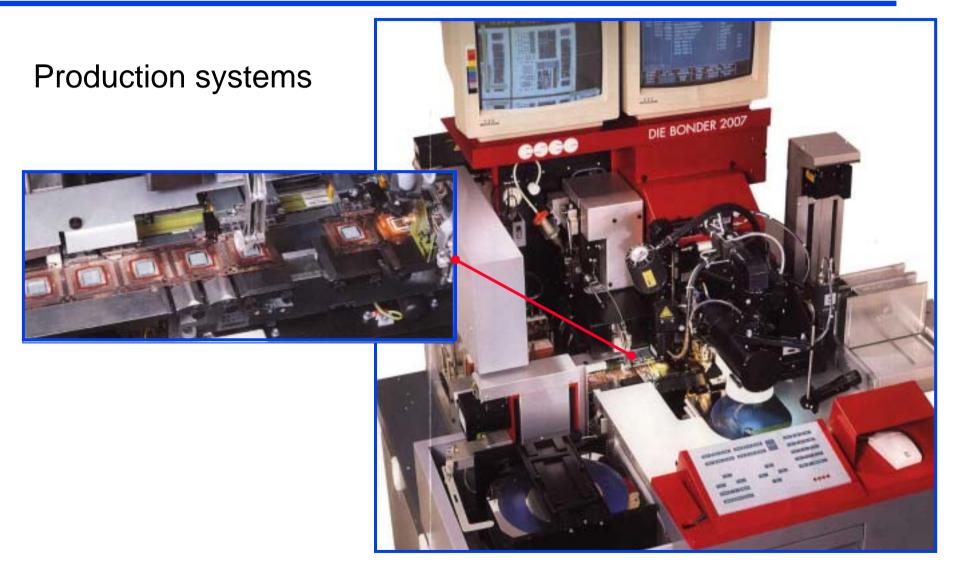




Institute of Technology

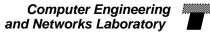
Consumer electronics, for example MP3 Audio, digital camera, home electronics, ....





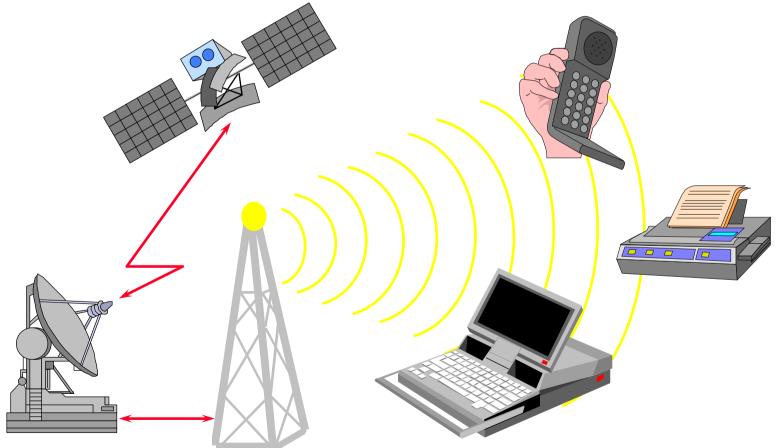


1 - 8



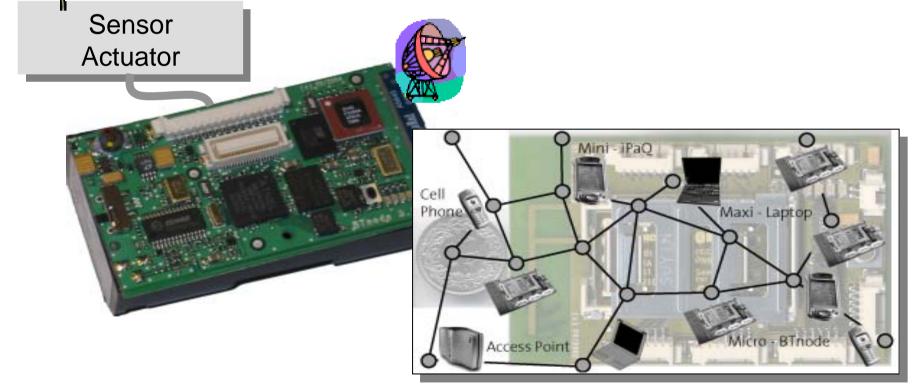


Information systems, for example wireless communication (mobile phone, Wireless LAN, ...), end-user equipment, router, ...



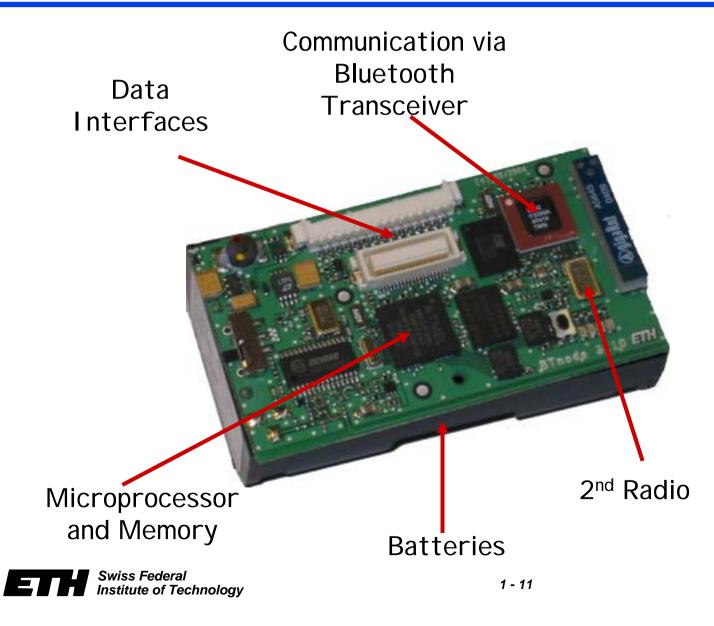
## **Communicating Embedded Systems**

- Example: BTnodes (http://www.btnode.ethz.ch)
  - complete platform including OS
  - especially suited for pervasive computing applications





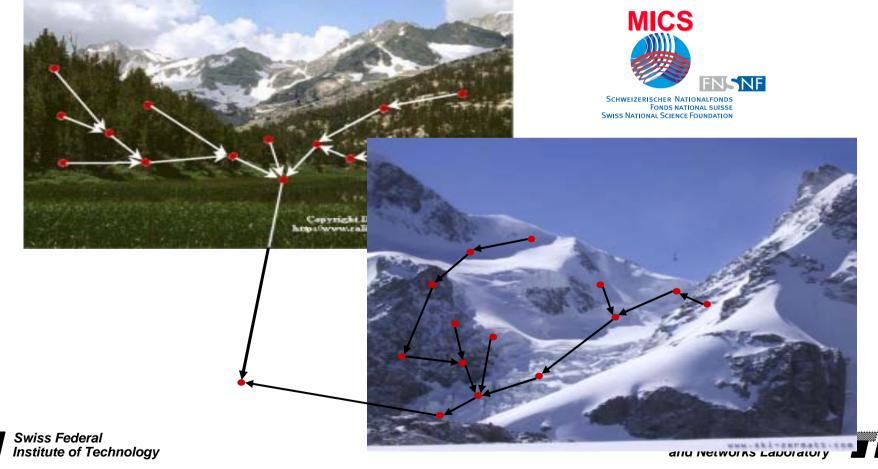
### **BTnode Platform**



- generic platform for ad-hoc computing
- complete platform including OS
- especially suited for pervasive computing applications

## **Communicating Embedded Systems**

- sensor networks (civil engineering, buildings, environmental monitoring, traffic, emergency situations)
- smart products. wearable/ubiquitous computing



## **Characteristics of Embedded Systems (1)**

- Must be dependable:
  - Reliability: R(t) = probability of system working correctly provided that is was working at t=0
  - Maintainability: M(d) = probability of system working correctly d time units after error occurred.
  - **Availability:** probability of system working at time t
  - Safety: no harm to be caused
  - Security: confidential and authentic communication

Even perfectly designed systems can fail if the assumptions about the workload and possible errors turn out to be wrong.Making the system dependable must not be an after-thought, it must be considered from the very beginning.





## **Characteristics of Embedded Systems (2)**

- Must be efficient:
  - Energy efficient
  - **Code-size** efficient (especially for systems on a chip)
  - Run-time efficient
  - Weight efficient
  - Cost efficient
- Dedicated towards a certain application: Knowledge about behavior at design time can be used to minimize resources and to maximize robustness.
- Dedicated user interface (no mouse, keyboard and screen).

## **Characteristics of Embedded Systems (3)**

- Many ES must meet *real-time constraints*:
  - A real-time system must *react to stimuli* from the controlled object (or the operator) within the time interval dictated by the environment.
  - For real-time systems, right answers arriving too late (or even too early) are wrong.

"A real-time constraint is called hard, if not meeting that constraint could result in a catastrophe" [Kopetz, 1997].

- All other time-constraints are called soft.
- A guaranteed system response has to be explained without statistical arguments.



# **Characteristics of Embedded Systems (4)**

- Frequently connected to physical environment through sensors and actuators,
- Hybrid systems (analog + digital parts).
- Typically, ES are *reactive systems*:

"A reactive system is one which is in continual interaction with is environment and executes at a pace determined by that environment" [Bergé, 1995]

Behavior depends on input and current state. automata model often appropriate,



## Comparison

- Embedded Systems
  - Few applications that are known at design-time.
  - Not programmable by end user.
  - Fixed run-time requirements (additional computing power not useful).
  - Criteria:
    - cost
    - power consumption
    - predictability
    - ...

- General Purpose Computing
  - Broad class of applications.
  - Programmable by end user.
  - Faster is better.
  - Criteria:
    - cost
    - average speed

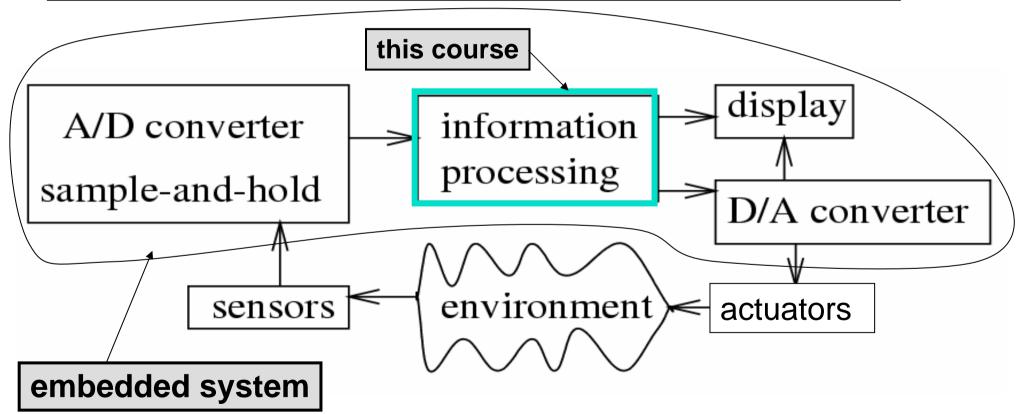
## Topics

- General Introduction to Embedded Systems
- Hardware Platforms and Components
  - System Specialization
  - Application Specific Instruction Sets
    - Micro Controller
    - Digital Signal Processors and VLIW
  - Programmable Hardware
  - ASICs
  - System-on-Chip

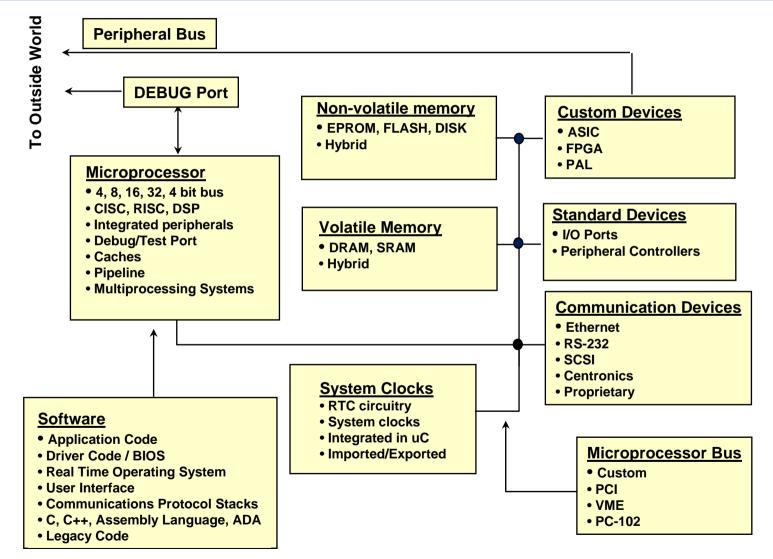


#### **Embedded System Hardware**

Embedded system hardware is frequently used in a loop (*"hardware in a loop"*):



## **Typical Architecture**





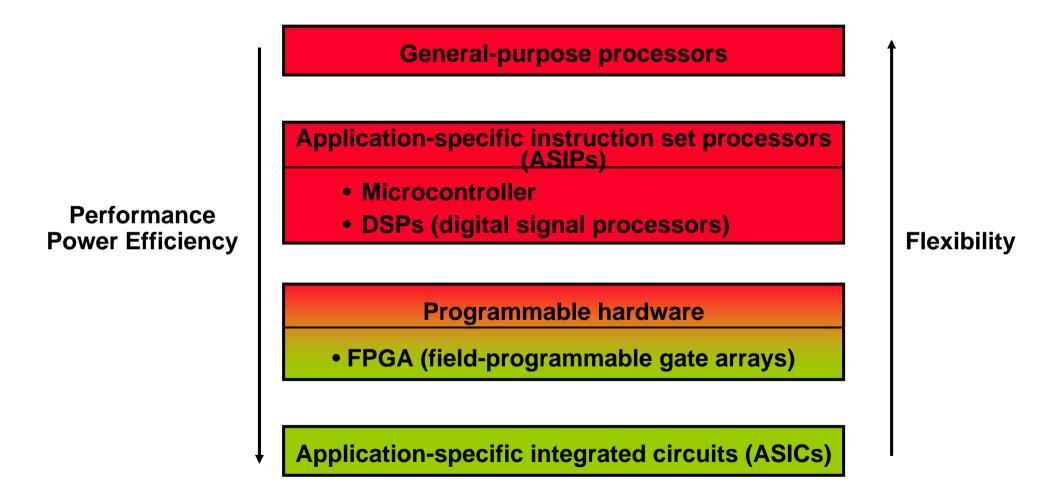
Swiss Federal Institute of Technology

## Topics

- General Introduction to Embedded Systems
- Hardware Platforms and Components
  - System Specialization
  - Application Specific Instruction Sets
    - Micro Controller
    - Digital Signal Processors and VLIW
  - Programmable Hardware
  - ASICs
  - System-on-Chip



#### **Implementation Alternatives**



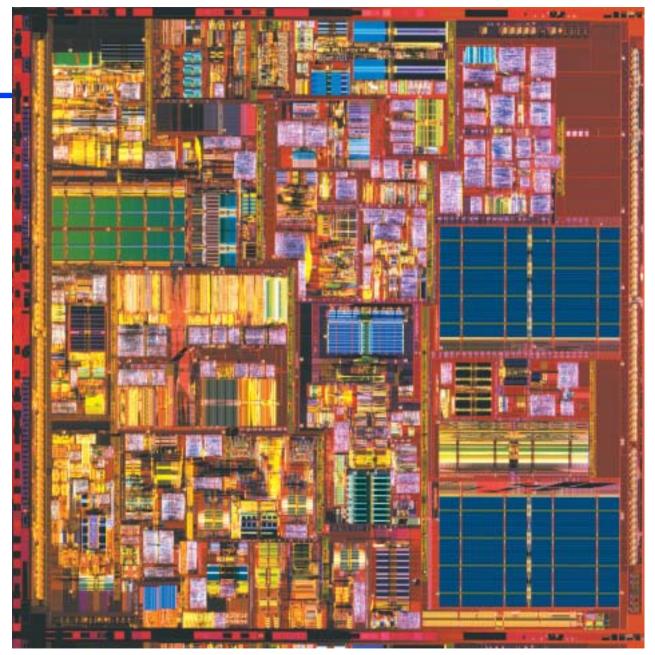


## **General-purpose Processors**

- High performance
  - Highly optimized circuits and technology
  - Use of parallelism
    - superscalar: dynamic scheduling of instructions
    - super-pipelining: instruction pipelining, branch prediction, speculation
  - complex memory hierarchy
- Not suited for real-time applications
  - Execution times are highly unpredictable because of intensive resource sharing and dynamic decisions
- Properties
  - Good average performance for large application mix
  - High power consumption



#### **Pentium P4**







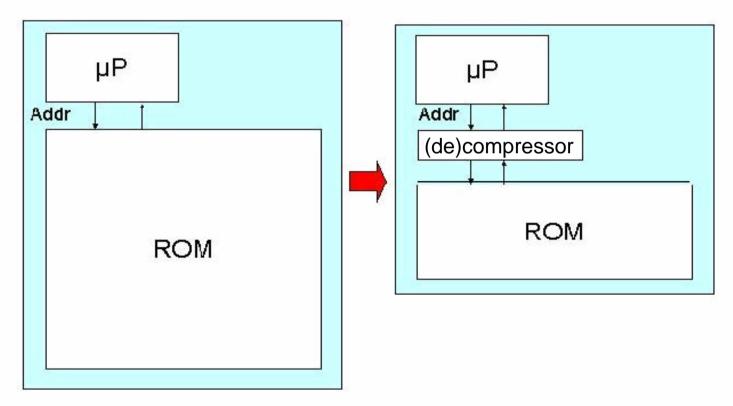
## System Specialization

- The main difference between general purpose highest volume microprocessors and embedded systems is specialization.
- Specialization should respect flexibility
  - application domain specific systems shall cover a class of applications
  - some flexibility is required to account for late changes, debugging
- System analysis required
  - identification of application properties which can be used for specialization
  - quantification of individual specialization effects



### **Example: Code-size Efficiency**

- CISC machines: RISC machines designed for run-time-, not for code-size-efficiency.
- Compression techniques: key idea

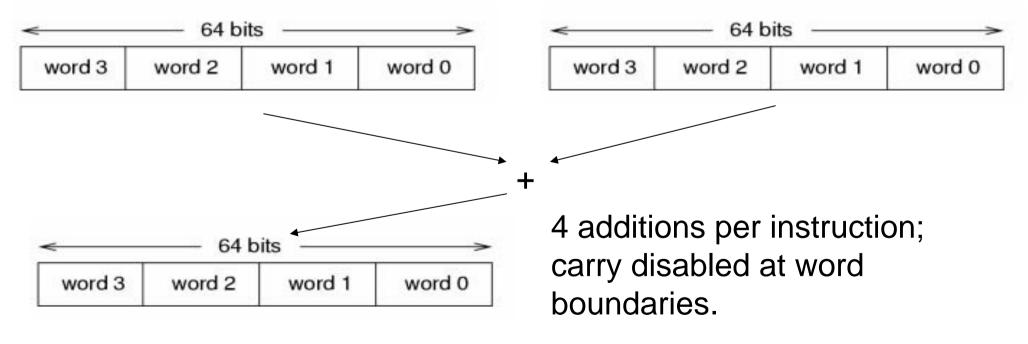


## **Example: Multimedia-Instructions**

Multimedia instructions exploit that many registers, adders etc are quite wide (32/64 bit), whereas most multimedia data types are narrow

(e.g. 8 bit per color, 16 bit per audio sample per channel)

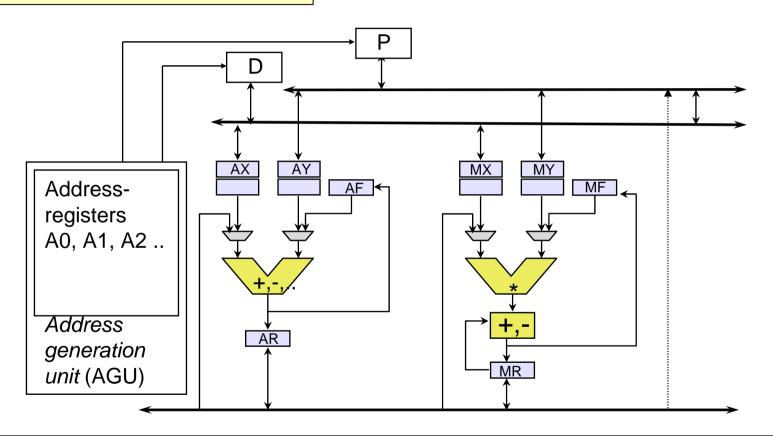
2-8 values can be stored per register and added. E.g.:





## **Example: Heterogeneous registers**

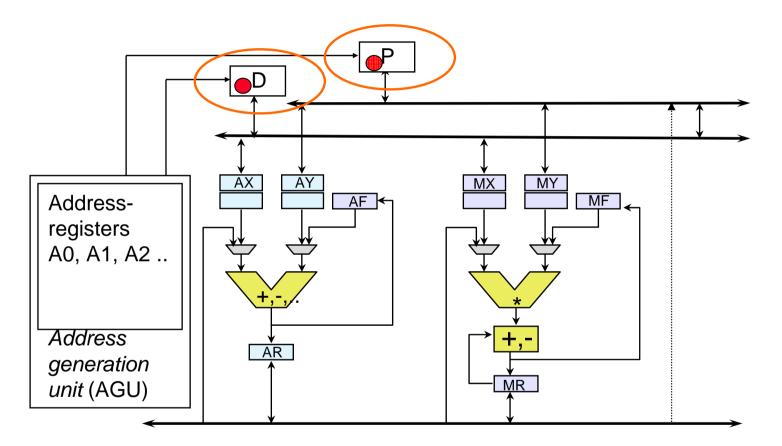
#### Example (ADSP 210x):



Different functionality of registers An, AX, AY, AF, MX, MY, MF, MR



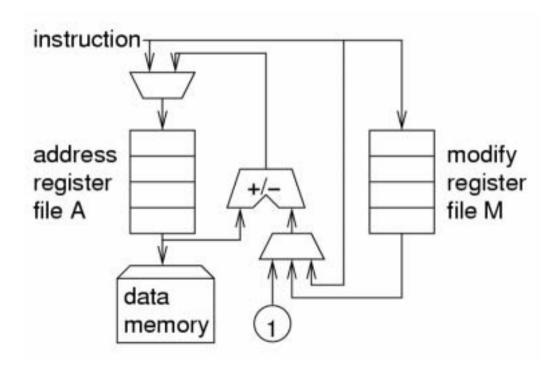
#### **Example: Multiple memory banks or memories**



#### Simplifies parallel fetches

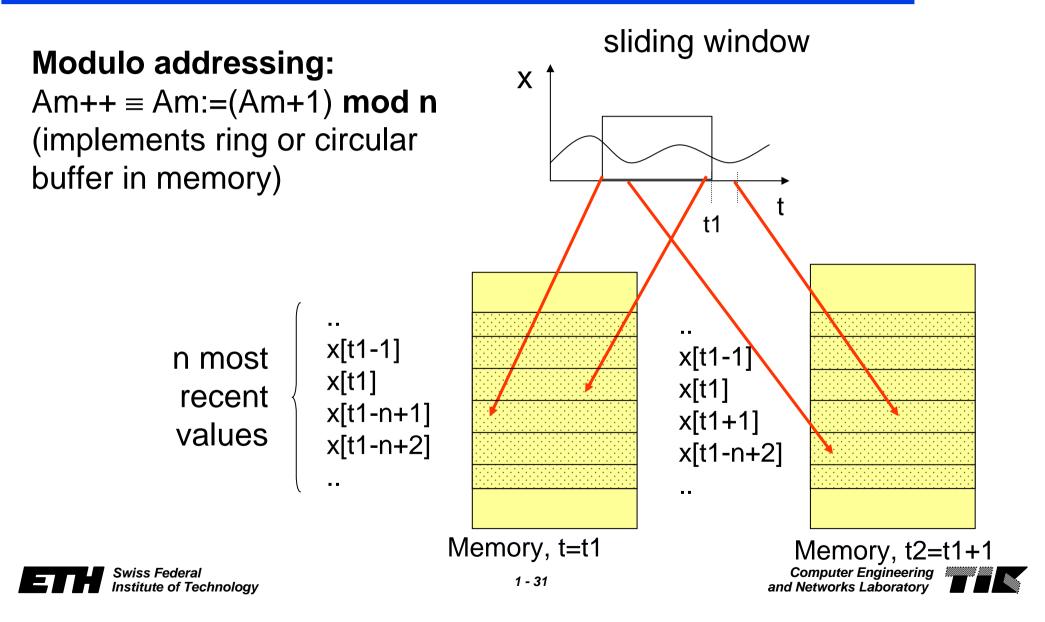
## **Example: Address generation units**

#### Example (ADSP 210x):



- Data memory can only be fetched with address contained in A,
- but this can be done in parallel with operation in main data path (takes effectively 0 time).
- $A := A \pm 1$  also takes 0 time,
- same for  $A := A \pm M$ ;

#### **Example: Modulo addressing**



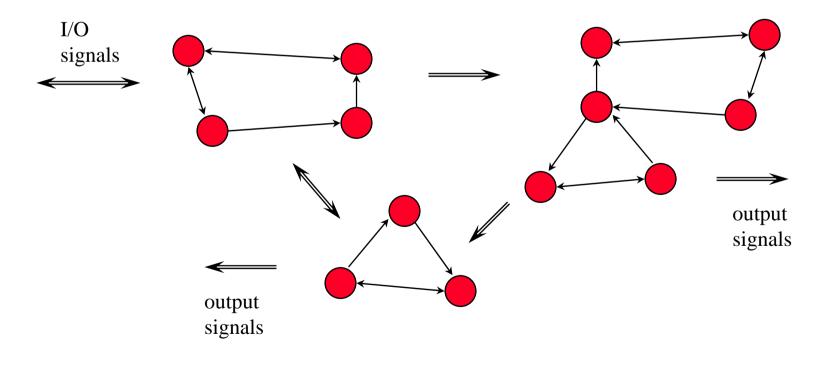
## Topics

- General Introduction to Embedded Systems
- Hardware Platforms and Components
  - System Specialization
  - Application Specific Instruction Sets
    - Micro Controller
    - Digital Signal Processors and VLIW
  - Programmable Hardware
  - ASICs
  - System-on-Chip



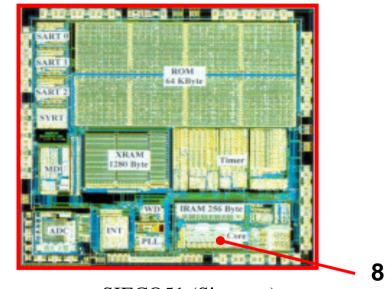
## **Control Dominated Systems**

- Reactive systems with event driven behavior
- Underlying semantics of system description ("input model of computation") typically (coupled) Finite State Machines



### **Microcontroller**

- control-dominant applications
  - supports process scheduling and synchronization
  - preemption (interrupt), context switch
  - short latency times
- Iow power consumption
- peripheral units often integrated
- suited for real-time applications



SIECO51 (Siemens)

8051 core

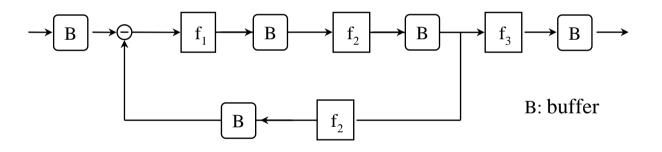
## Topics

- General Introduction to Embedded Systems
- Hardware Platforms and Components
  - System Specialization
  - Application Specific Instruction Sets
    - Micro Controller
    - Digital Signal Processors and VLIW
  - Programmable Hardware
  - ASICs
  - System-on-Chip



## **Data Dominated Systems**

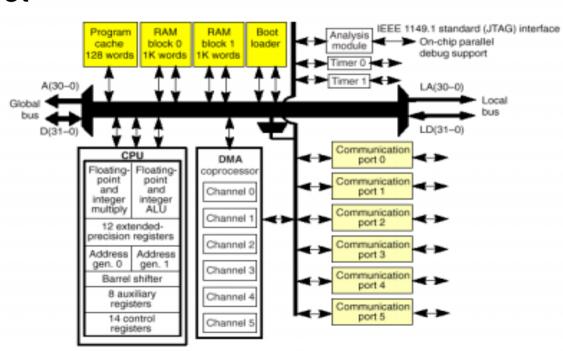
- Streaming oriented systems with mostly periodic behavior
- Underlying semantics of input description e.g. *flow* graphs ("input model of computation")



Application examples: signal processing, control engineering

# **Digital Signal Processor**

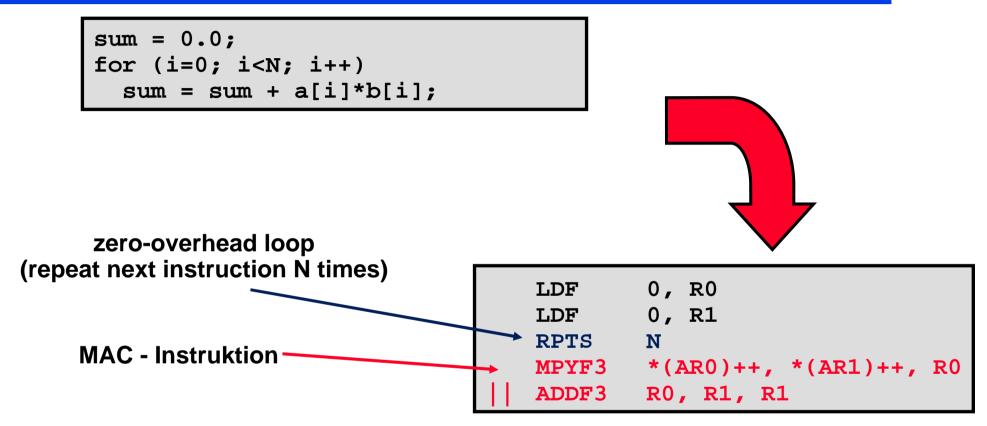
- optimized for data-flow applications
- suited for simple control flow
- parallel hardware units (VLIW)
- specialized instruction set
- high data throughput
- zero-overhead loops
- specialized memory
- suited for real-time applications



TMS320C40 Block Diagram



## MAC (multiply & accumulate)



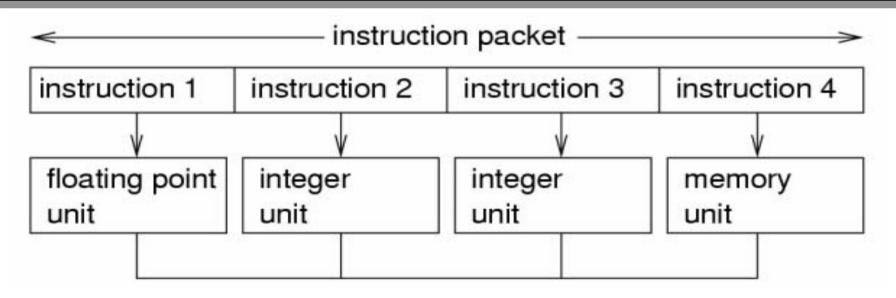
TMS320C3x Assembler (Texas Instruments)



# Very Long Instruction Word (VLIW)

Key idea: detection of possible parallelism to be done by compiler, not by hardware at run-time (inefficient).

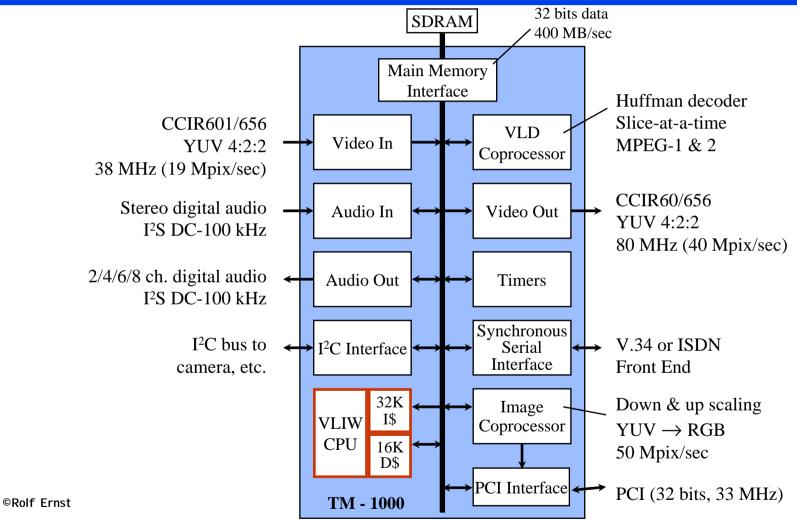
VLIW: parallel operations (instructions) encoded in one long word (instruction packet), each instruction controlling one functional unit. E.g.:







## **Example: Philips TriMedia TM1000**





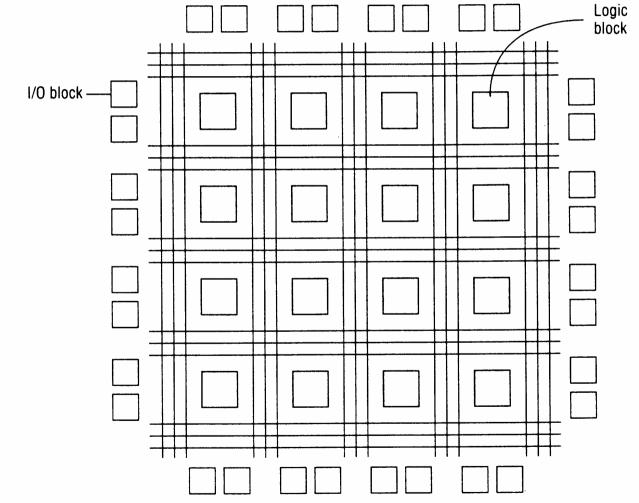
# Topics

- General Introduction to Embedded Systems
- Hardware Platforms and Components
  - System Specialization
  - Application Specific Instruction Sets
    - Micro Controller
    - Digital Signal Processors and VLIW
  - Programmable Hardware
  - ASICs
  - System-on-Chip



#### **FPGA – Basic Strucutre**

- Logic Units
- I/O Units
- Connections



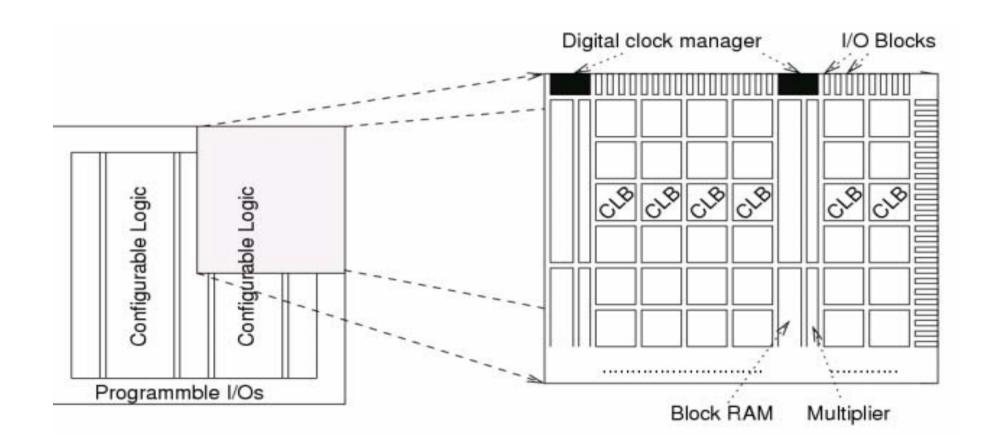




### **FPGA - Classification**

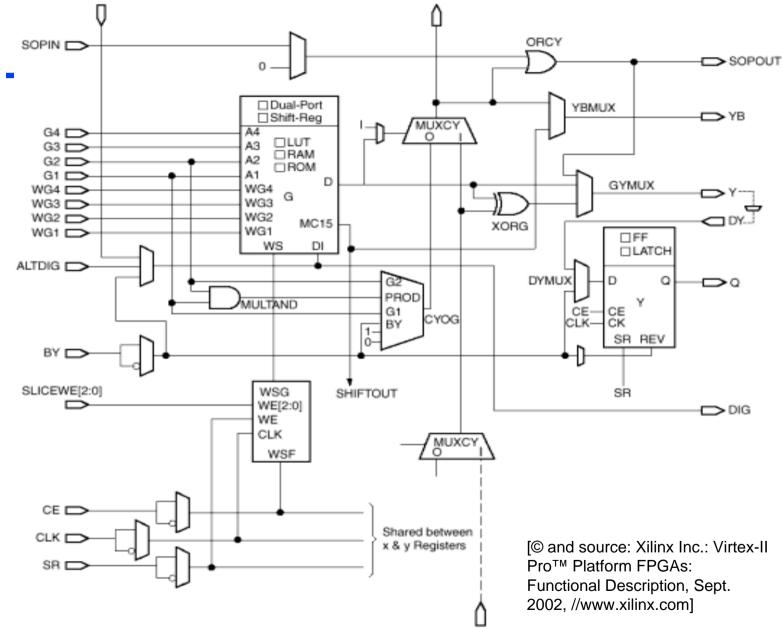
- Granularity of logic units:
  - Gate, tables, memory, functional blocks (ALU, control, data path, processor)
- Communication network:
  - Crossbar, hierarchical mesh, tree
- ► Reconfiguration:
  - fixed at production time, once at design time, dynamic during run-time

#### **Floor-plan of VIRTEX II FPGAs**





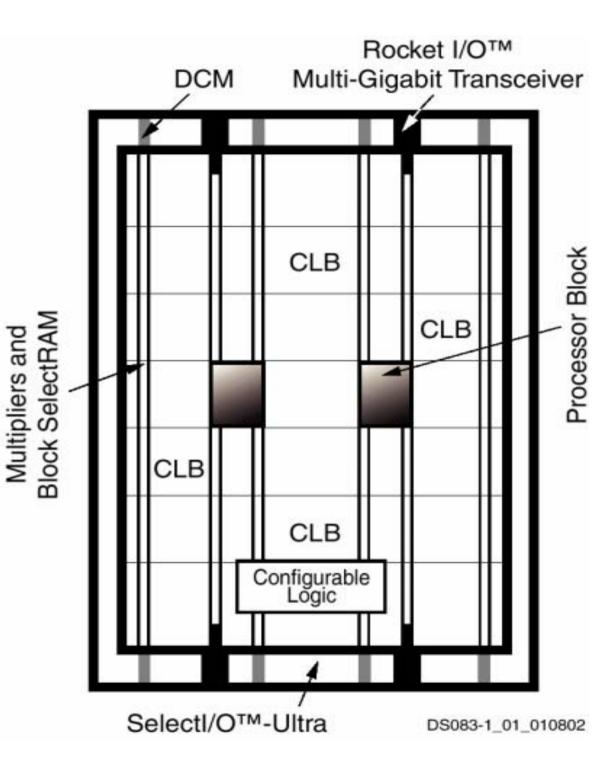




Swiss Federal Institute of Technology



## Virtex II Pro Devices include up to 4 PowerPC processor cores



[© and source: Xilinx Inc.: Virtex-II Pro<sup>™</sup> Platform FPGAs: Functional Description, Sept. 2002, //www.xilinx.com]



Swiss Federal Institute of Technology

# Topics

- General Introduction to Embedded Systems
- Hardware Platforms and Components
  - System Specialization
  - Application Specific Instruction Sets
    - Micro Controller
    - Digital Signal Processors and VLIW
  - Programmable Hardware
  - ASICs
  - System-on-Chip

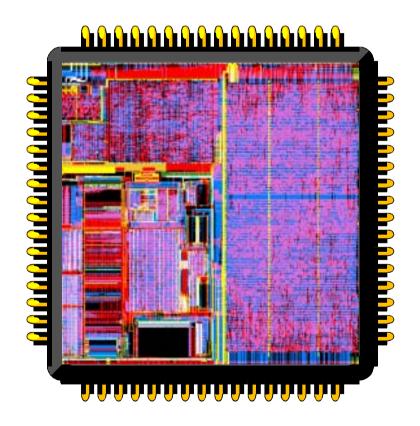
# **Application Specific Circuits (ASICS)**

Custom-designed circuits necessary

- if ultimate speed or
- energy efficiency is the goal and
- Iarge numbers can be sold.

#### Approach suffers from

- long design times,
- lack of flexibility (changing standards) and
- high costs (e.g. Mill. \$ mask costs).

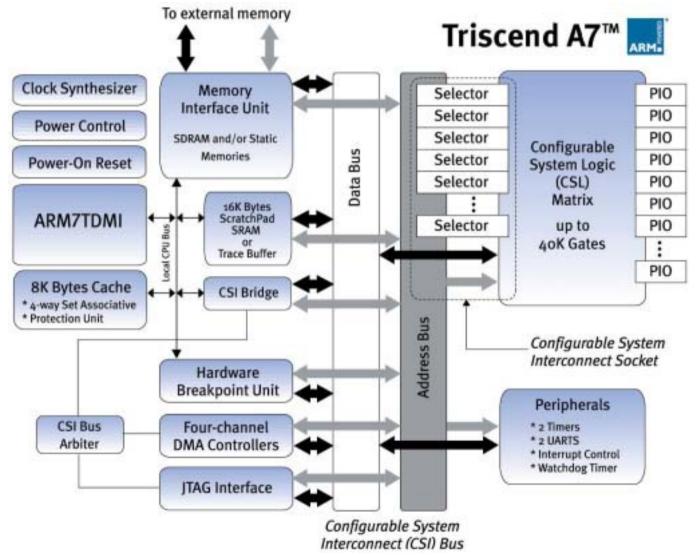




# Topics

- General Introduction to Embedded Systems
- Hardware Platforms and Components
  - System Specialization
  - Application Specific Instruction Sets
    - Micro Controller
    - Digital Signal Processors and VLIW
  - Programmable Hardware
  - ASICs
  - System-on-Chip

## **Configurable System-On-Chip**





### System-on-a-Chip

