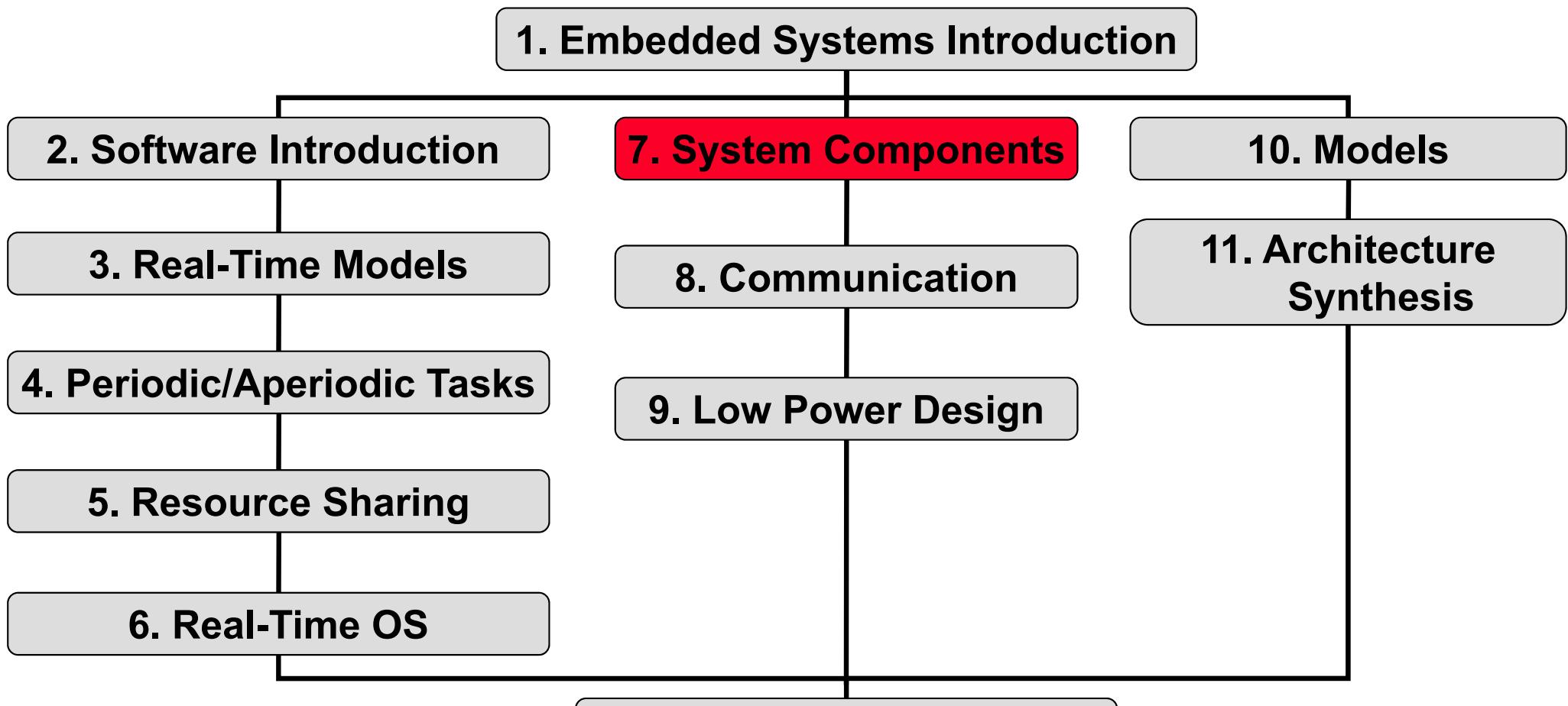


# Embedded Systems

## 7. System Components

Lothar Thiele

# Contents of Course



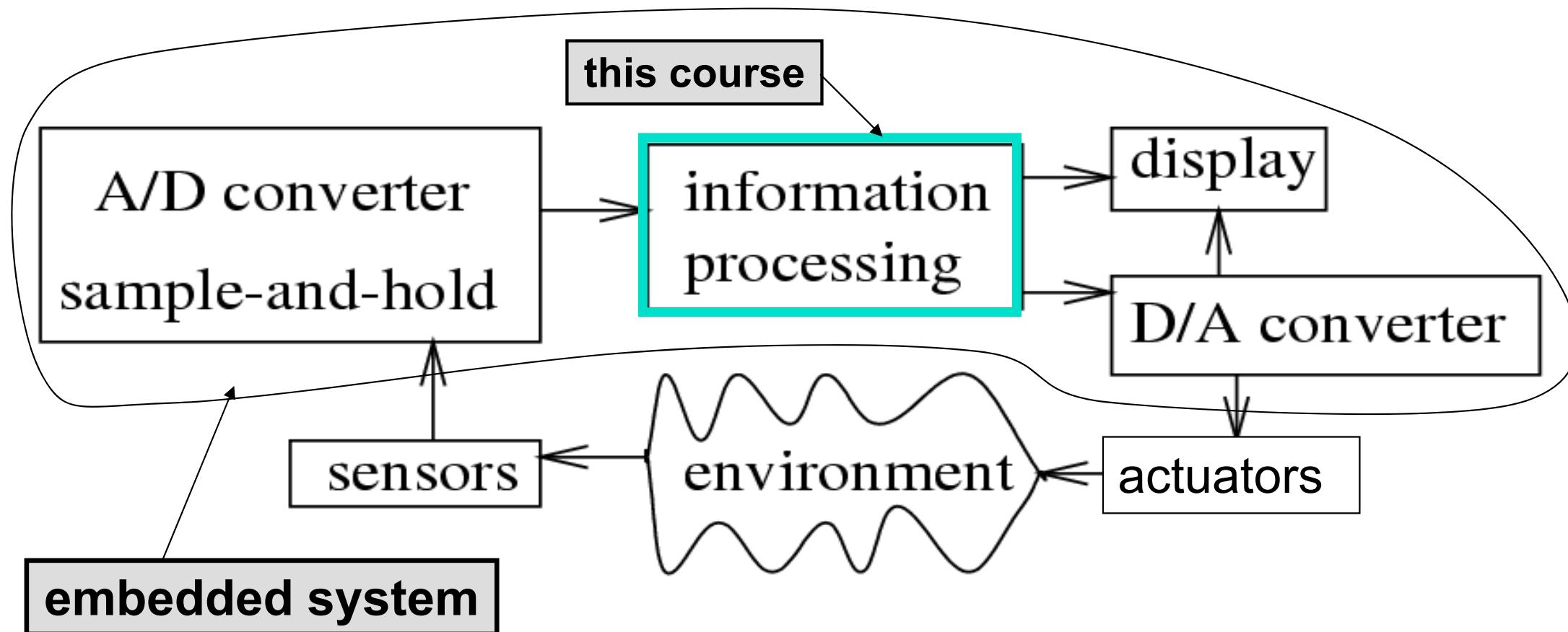
*Software and  
Programming*

*Processing and  
Communication*

*Hardware*

# Embedded System Hardware

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

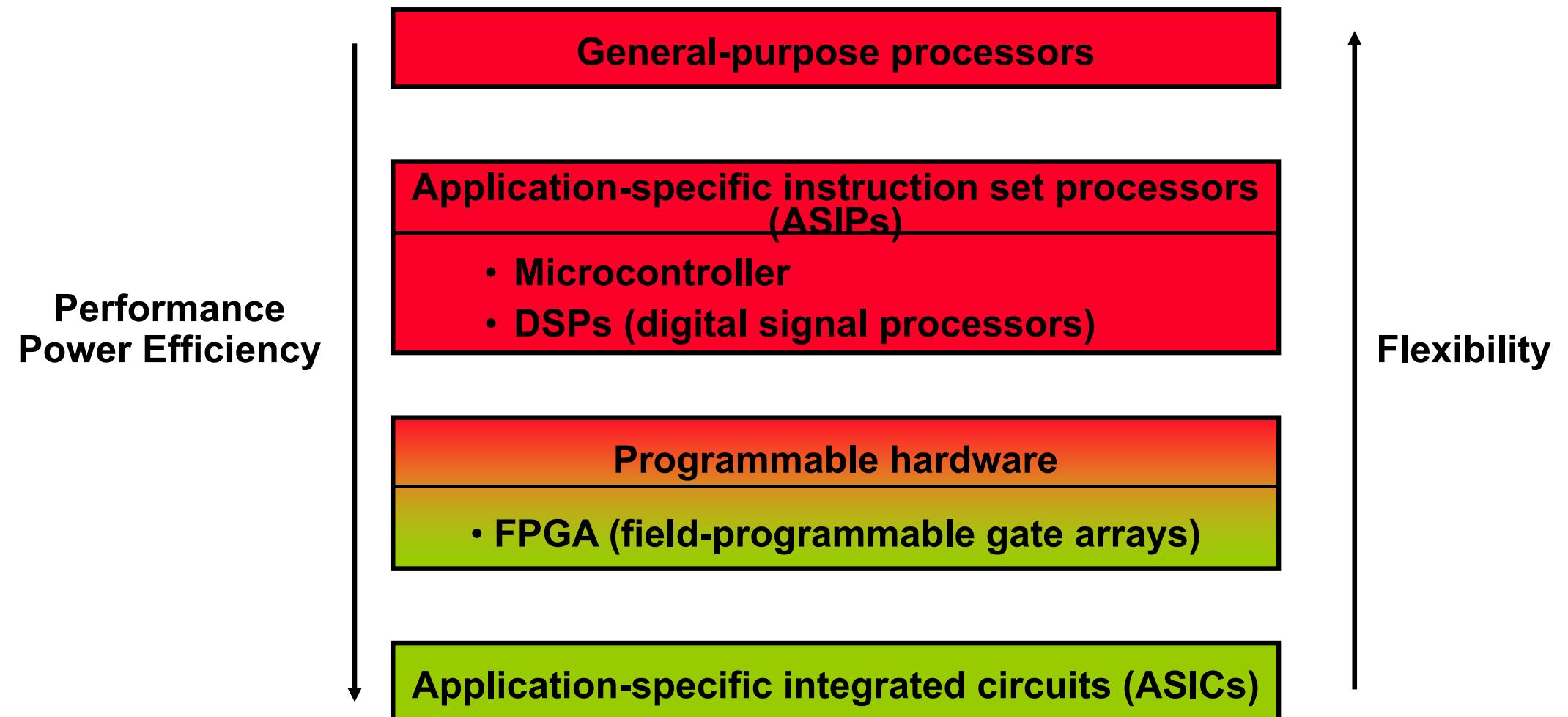


# Topics

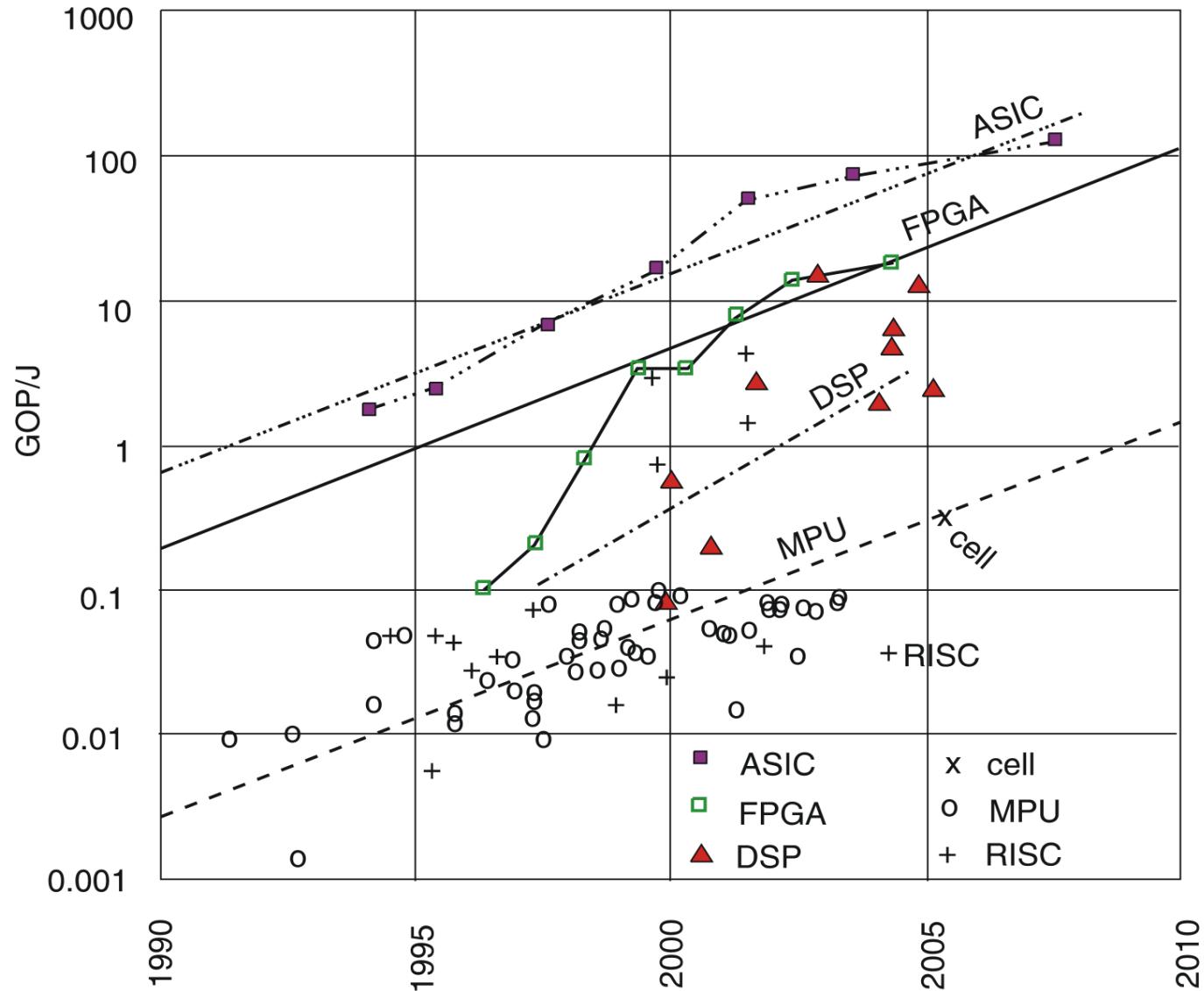
---

- ▶ ***System Specialization***
- ▶ Application Specific Instruction Sets
  - Micro Controller
  - Digital Signal Processors and VLIW
- ▶ Programmable Hardware
- ▶ ASICs
- ▶ System-on-Chip

# Implementation Alternatives



# Energy Efficiency



© Hugo De Man,  
IMEC, Philips, 2007

# 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

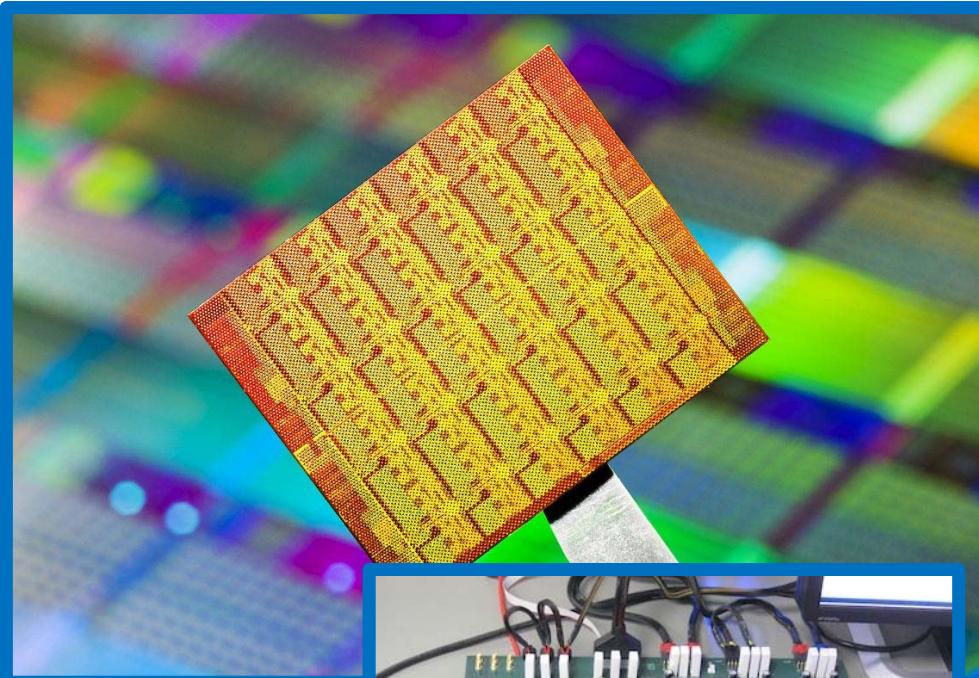
# General-purpose Processors

---

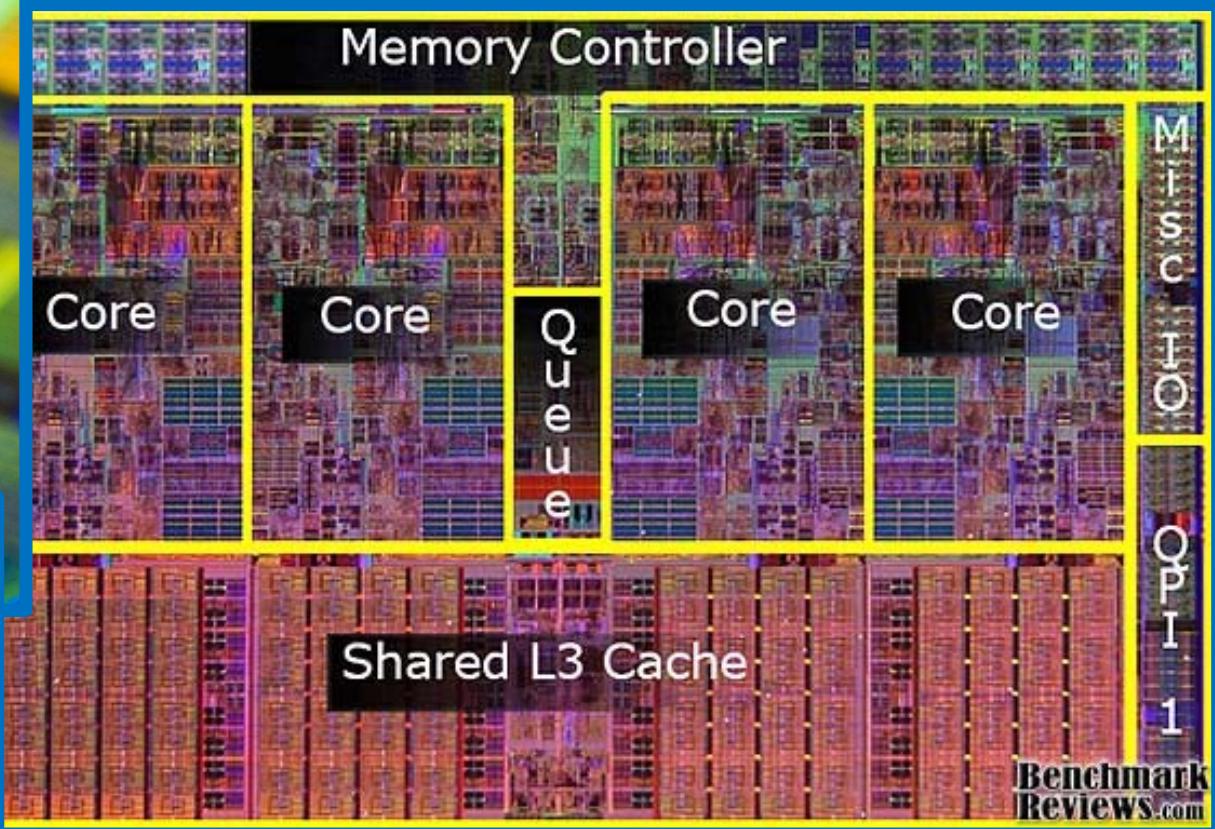
## ► *Multicore Processors*

- Potential of providing higher execution performance by exploiting parallelism
- Especially useful in high-performance embedded systems, e.g. autonomous driving
- Disadvantages and problems for embedded systems:
  - Increased interference on shared resources such as buses and shared caches
  - Increased timing uncertainty
  - Often, there is limited parallelism in embedded applications

# Multicore Examples

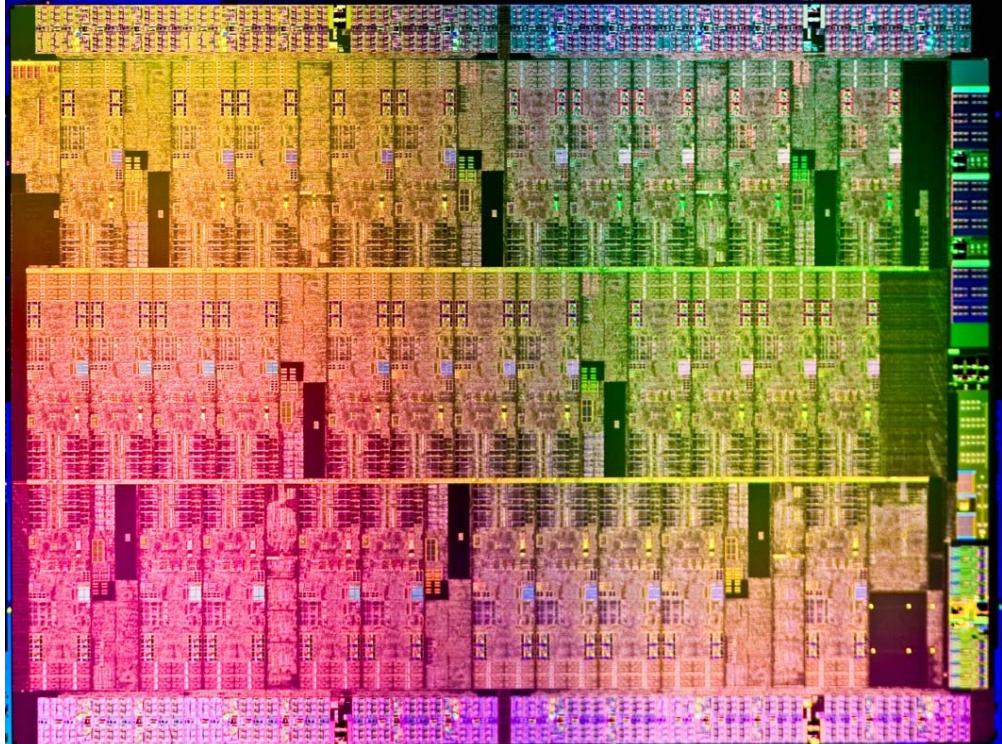


48 cores

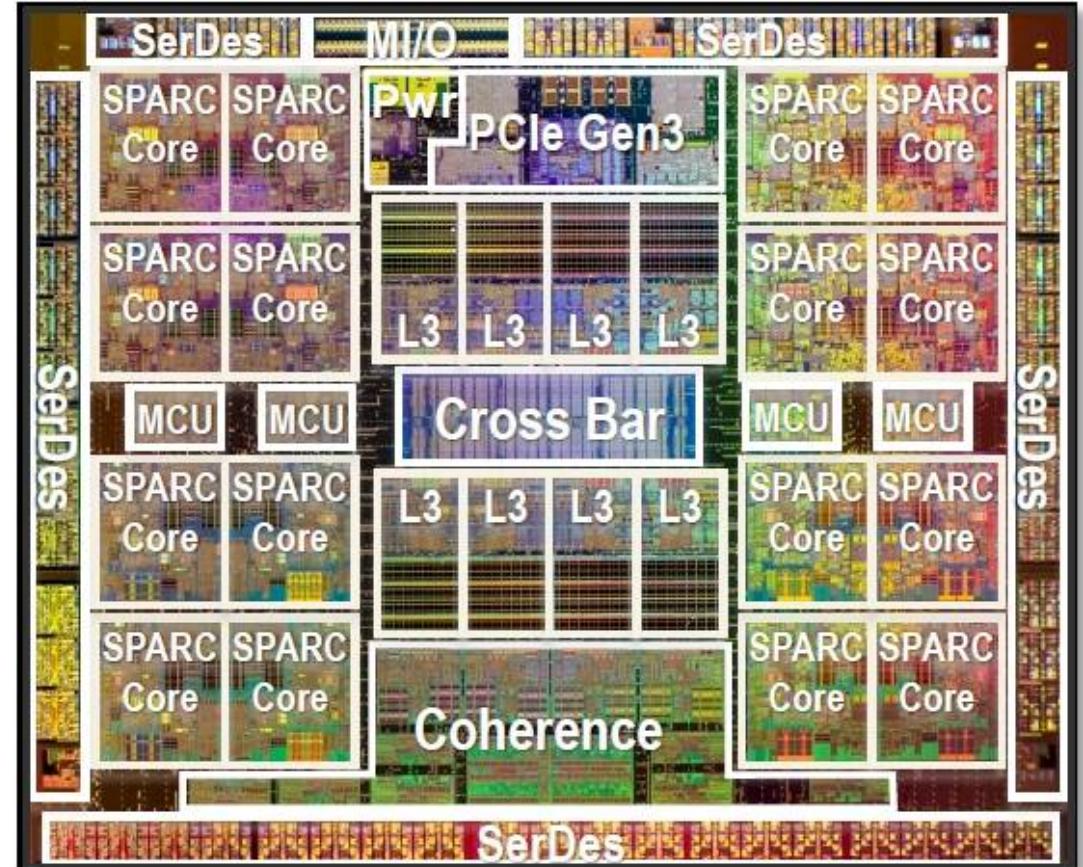


4 cores

# Multicore Examples



Intel Xeon Phi  
(5 Billion transistors,  
22nm technology,  
350mm<sup>2</sup> area)



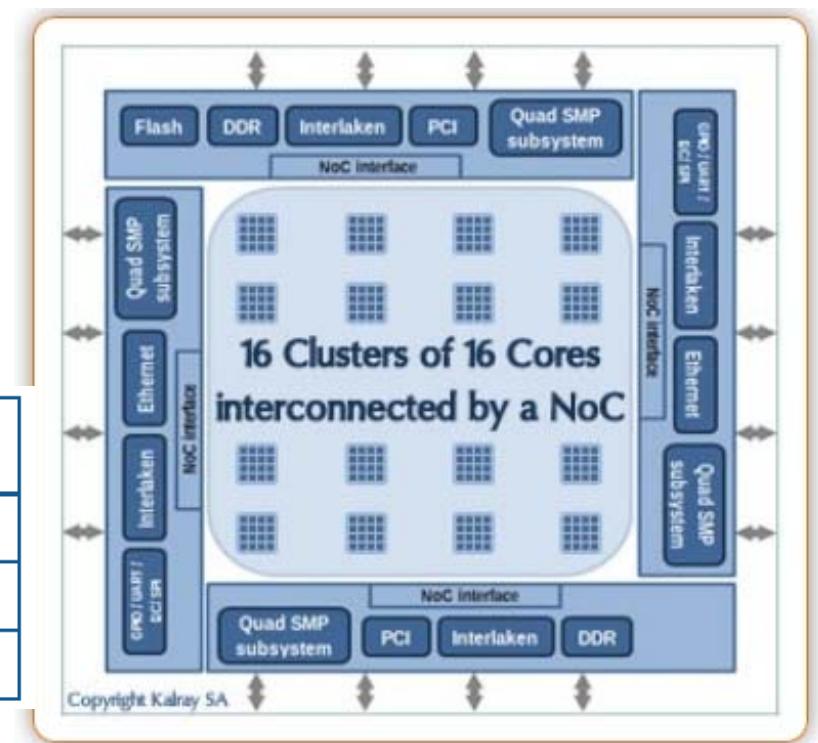
Oracle Sparc T5

# Embedded Multicore Example

- ▶ Recent development:
  - Specialize multicore processors towards real-time processing and low power consumption
  - Target domains:



Core Generation	Number of Processing Cores	GFLOPS/W	GOPS/W
Andey	256	25	75
Bostan (2014)	256	50	80
Coolidge (2015)	64/256/1024	75	115

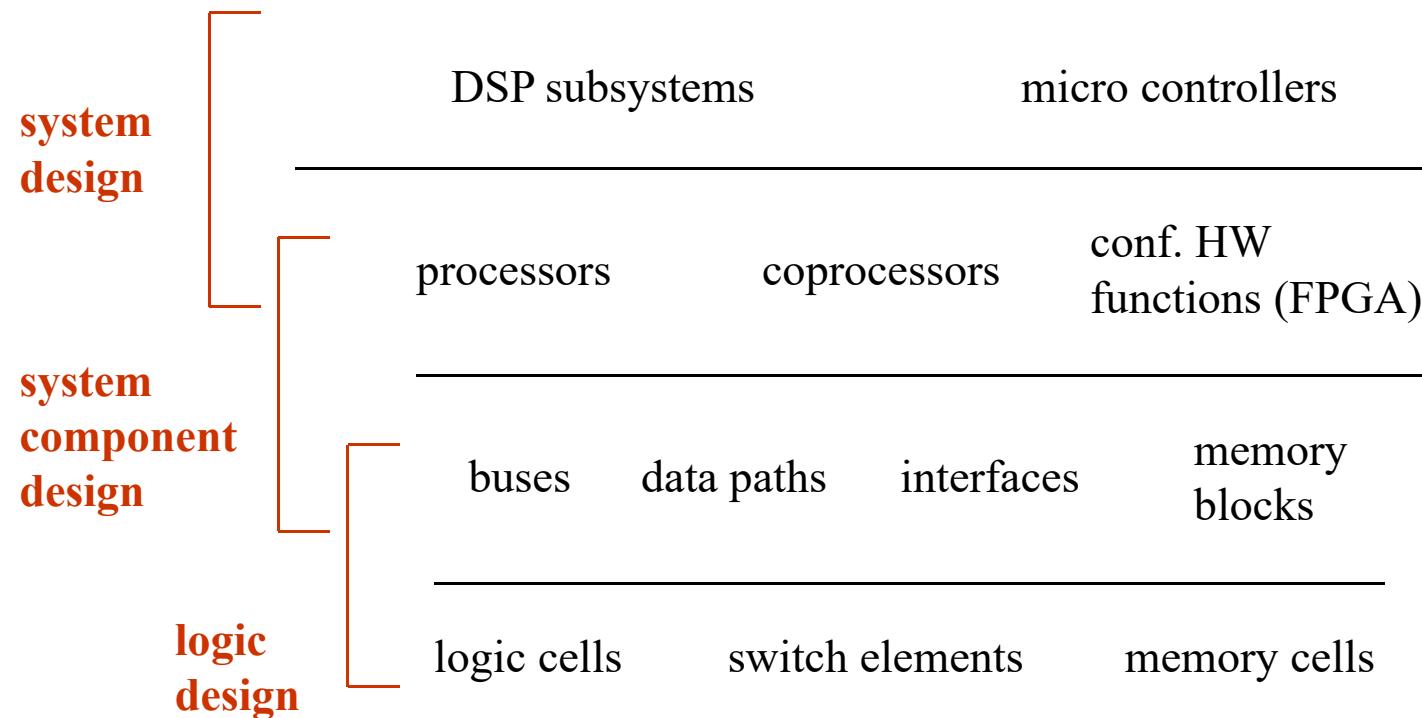


# System Specialization

---

- ▶ 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

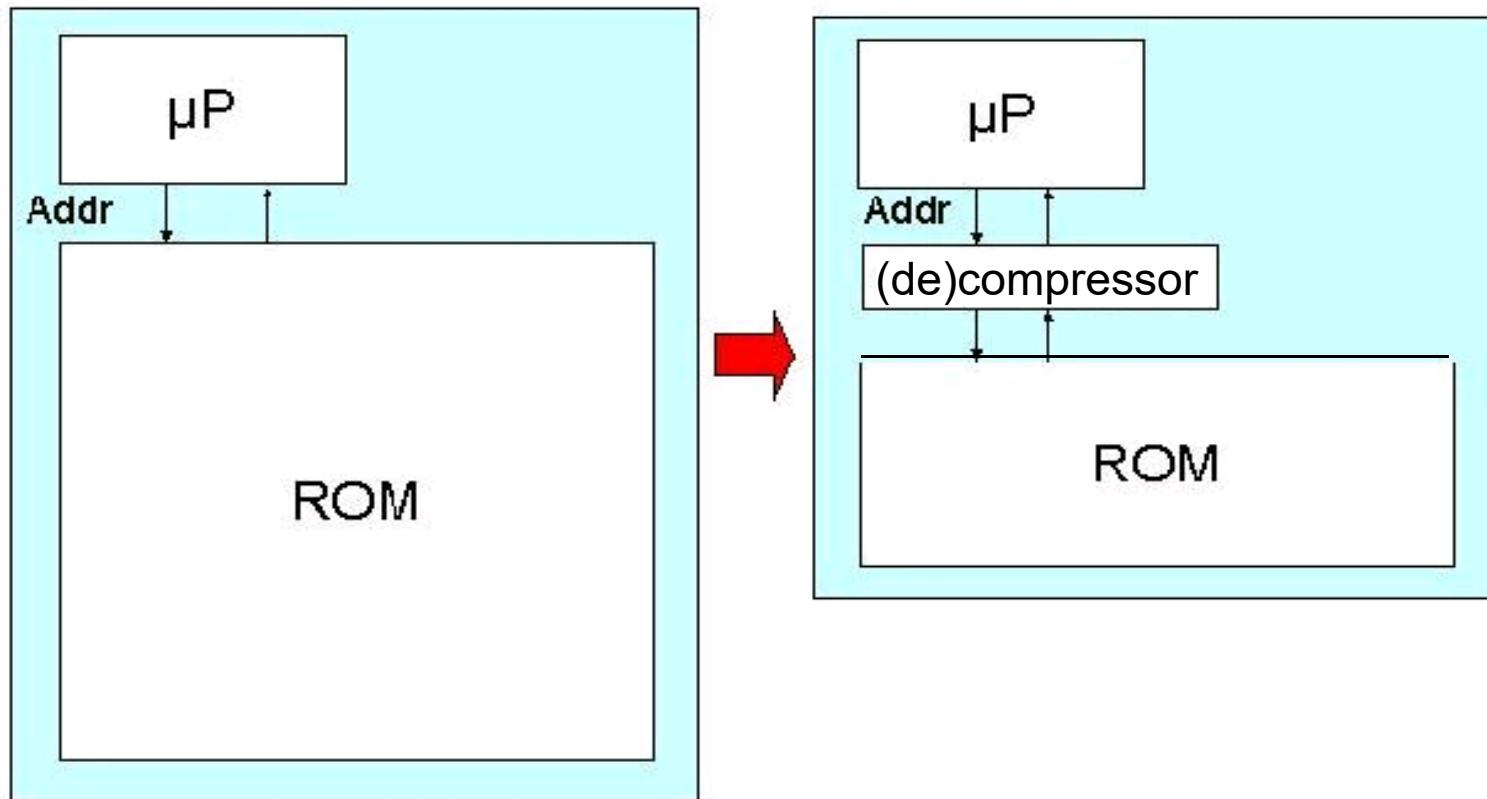
# Architecture Specialization Techniques



A simple system design classification

# Example: Code-size Efficiency

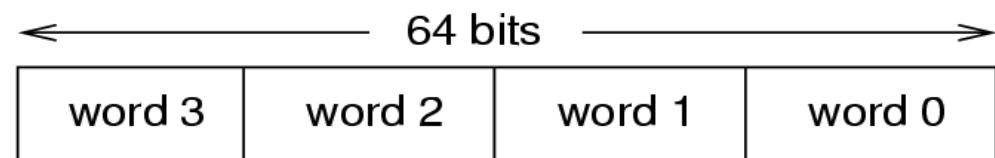
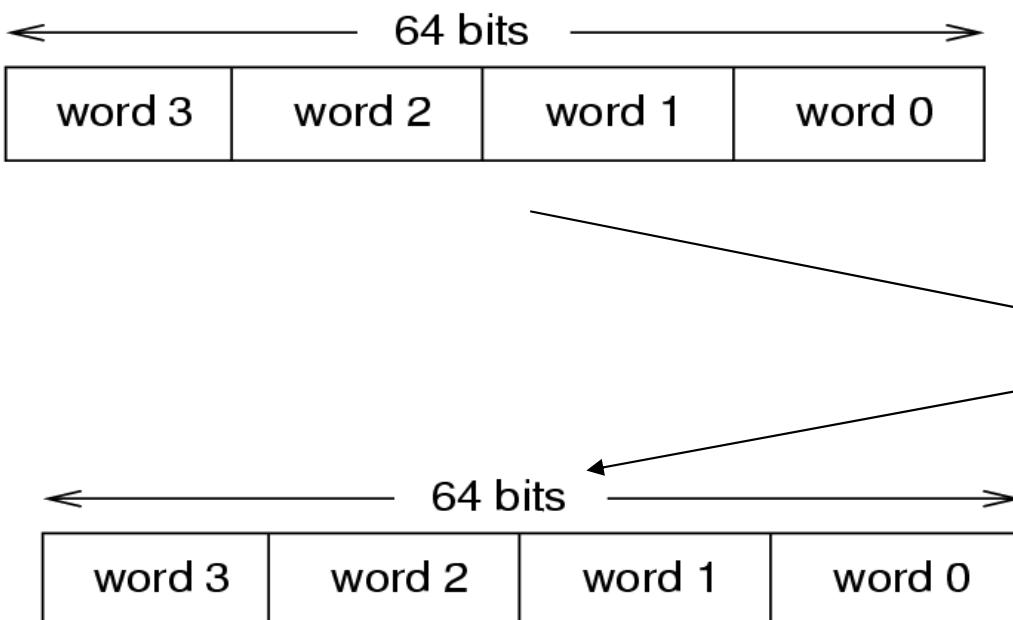
- ▶ RISC (Reduced Instruction Set Computers) machines designed for run-time-, not for code-size-efficiency.
- ▶ ***Compression techniques***: key idea



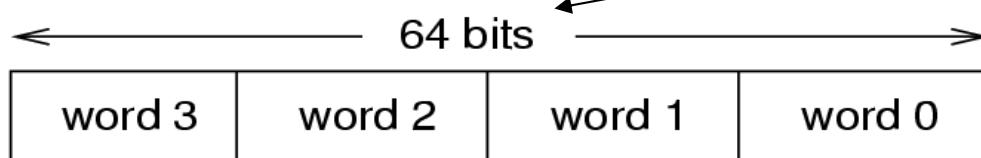
# Example: Multimedia-Instructions

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.



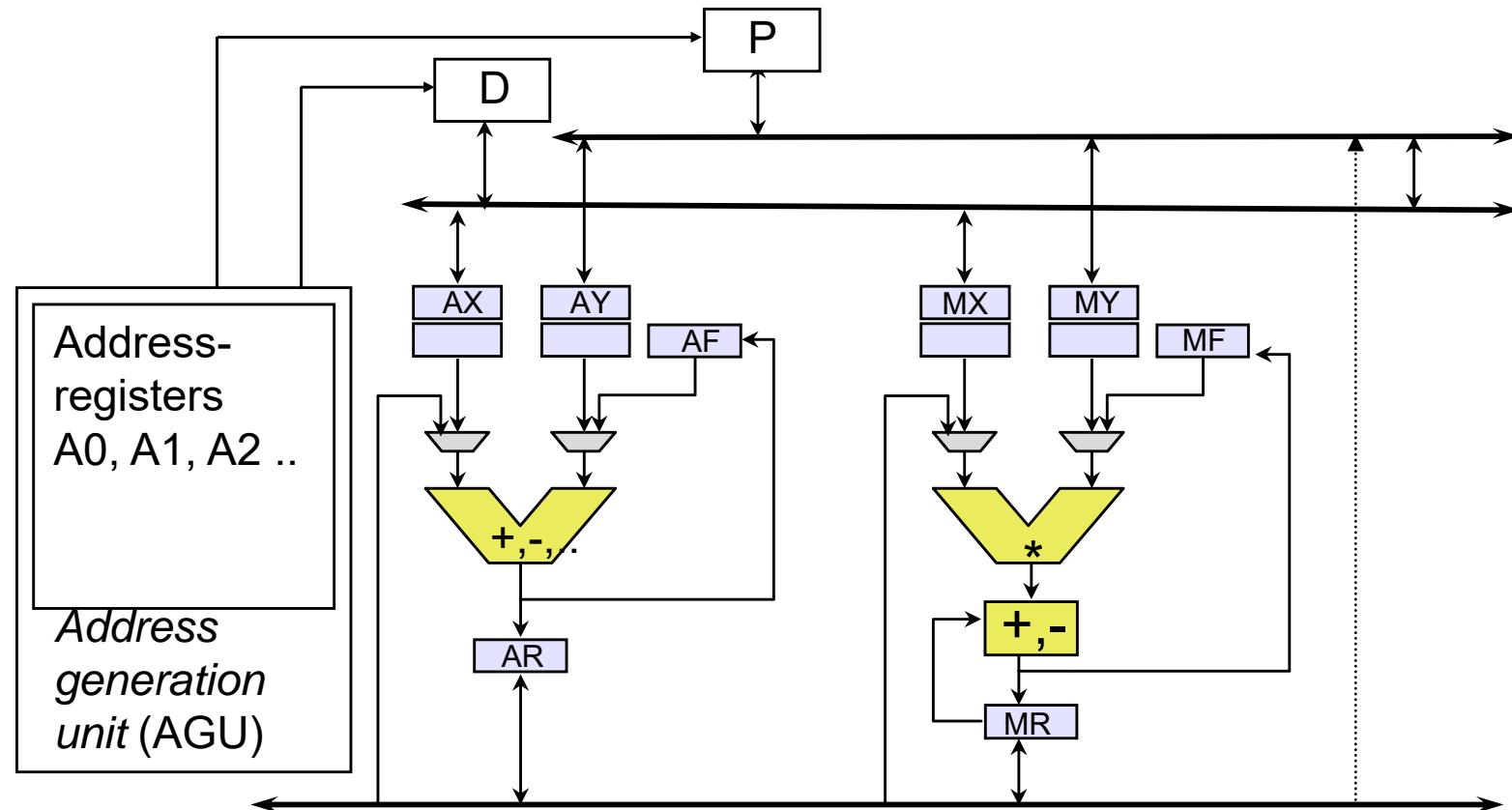
$$\begin{matrix} & + \\ \diagdown & & \diagup \\ \text{word 3} & \text{word 2} & \text{word 1} & \text{word 0} \end{matrix}$$



4 additions per instruction;  
carry disabled at word  
boundaries.

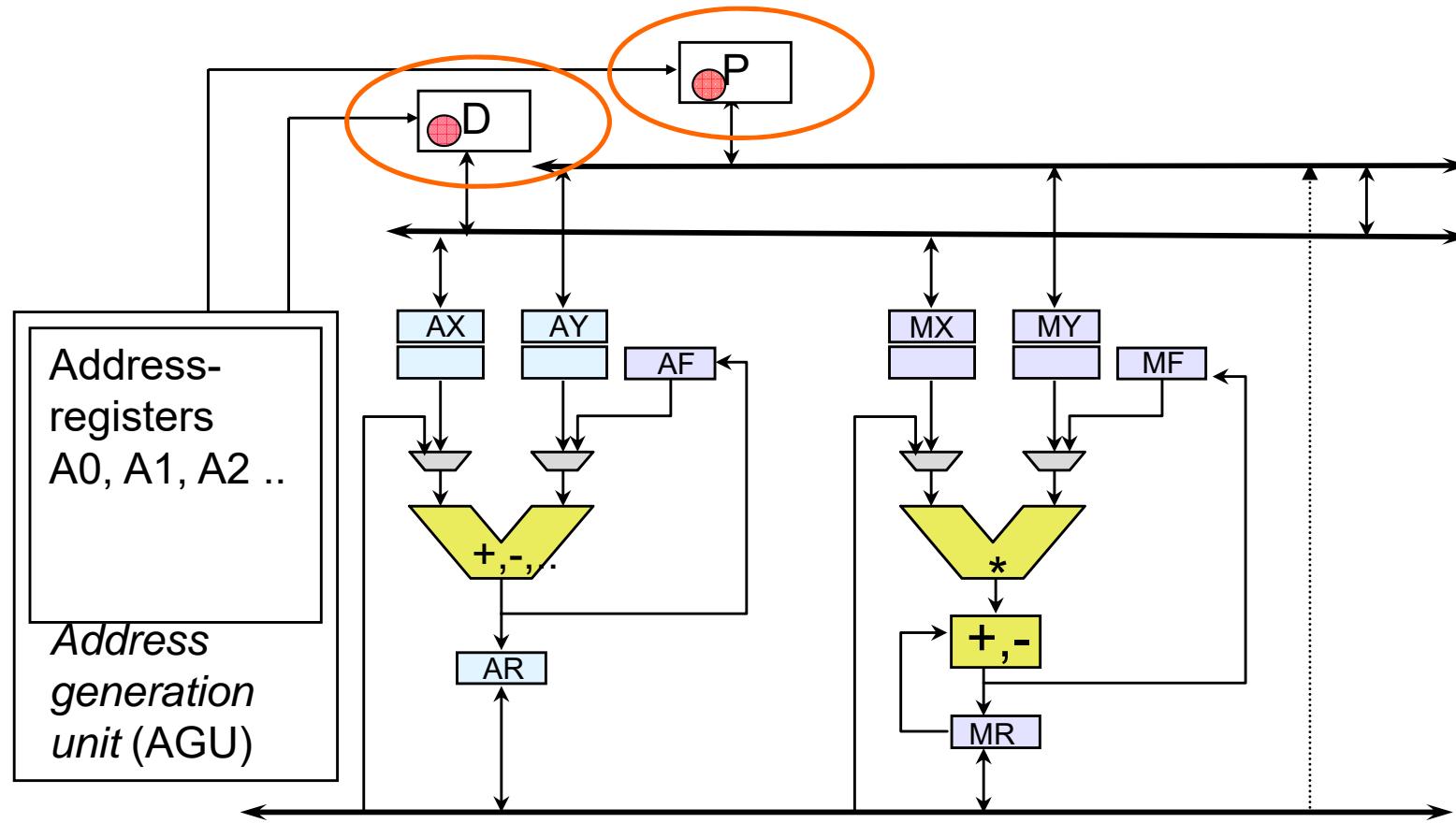
# Example: Heterogeneous registers

Example (ADSP 210x):



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

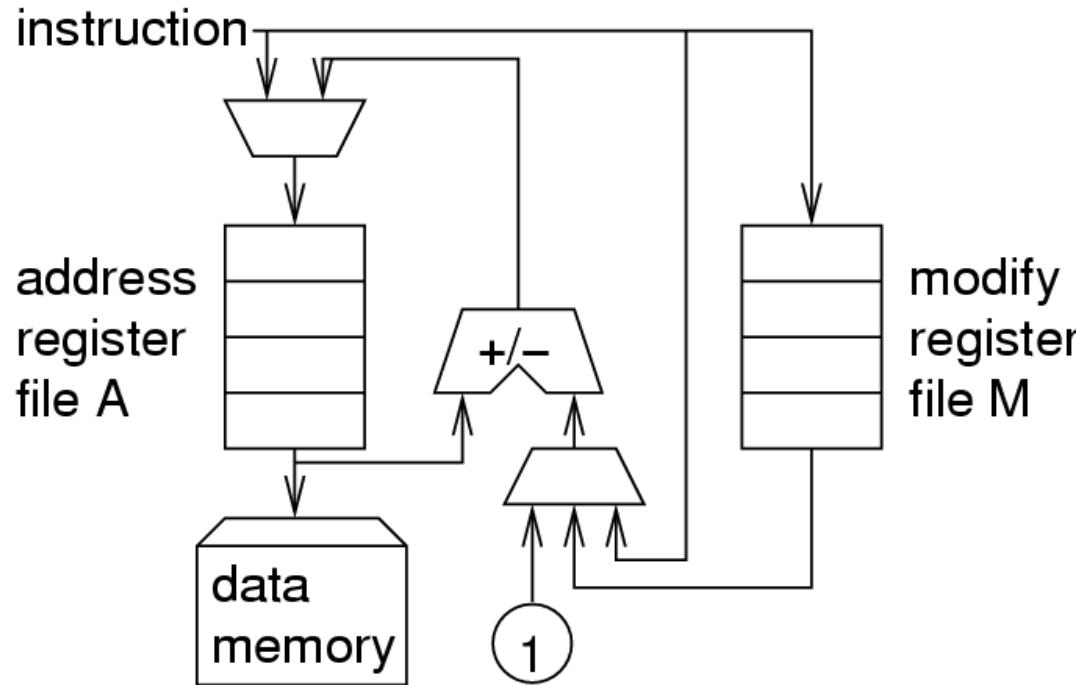
# Example: Multiple memory banks or memories



Simplifies parallel fetches

# Example: Address generation units

Example (ADSP 210x):



- Data memory can only be fetched with address contained in register file A, but its update can be done in parallel with operation in main data path (**takes effectively 0 time**).
- Register file A contains several precomputed addresses  $A[i]$ .
- There is another register file M that contains modification values  $M[j]$ .
- Possible updates:
  - $M[j] := \text{'immediate'}$
  - $A[i] := A[i] \pm M[j]$
  - $A[i] := A[i] \pm 1$
  - $A[i] := A[i] \pm \text{'immediate'}$
  - $A[i] := \text{'immediate'}$

# Example: Modulo addressing

## Modulo addressing:

$$A_{m++} \equiv A_m := (A_m + 1) \bmod n$$

(implements ring or circular buffer in memory)

$x[t]$ : value accessed at time  $t$

$\dots$

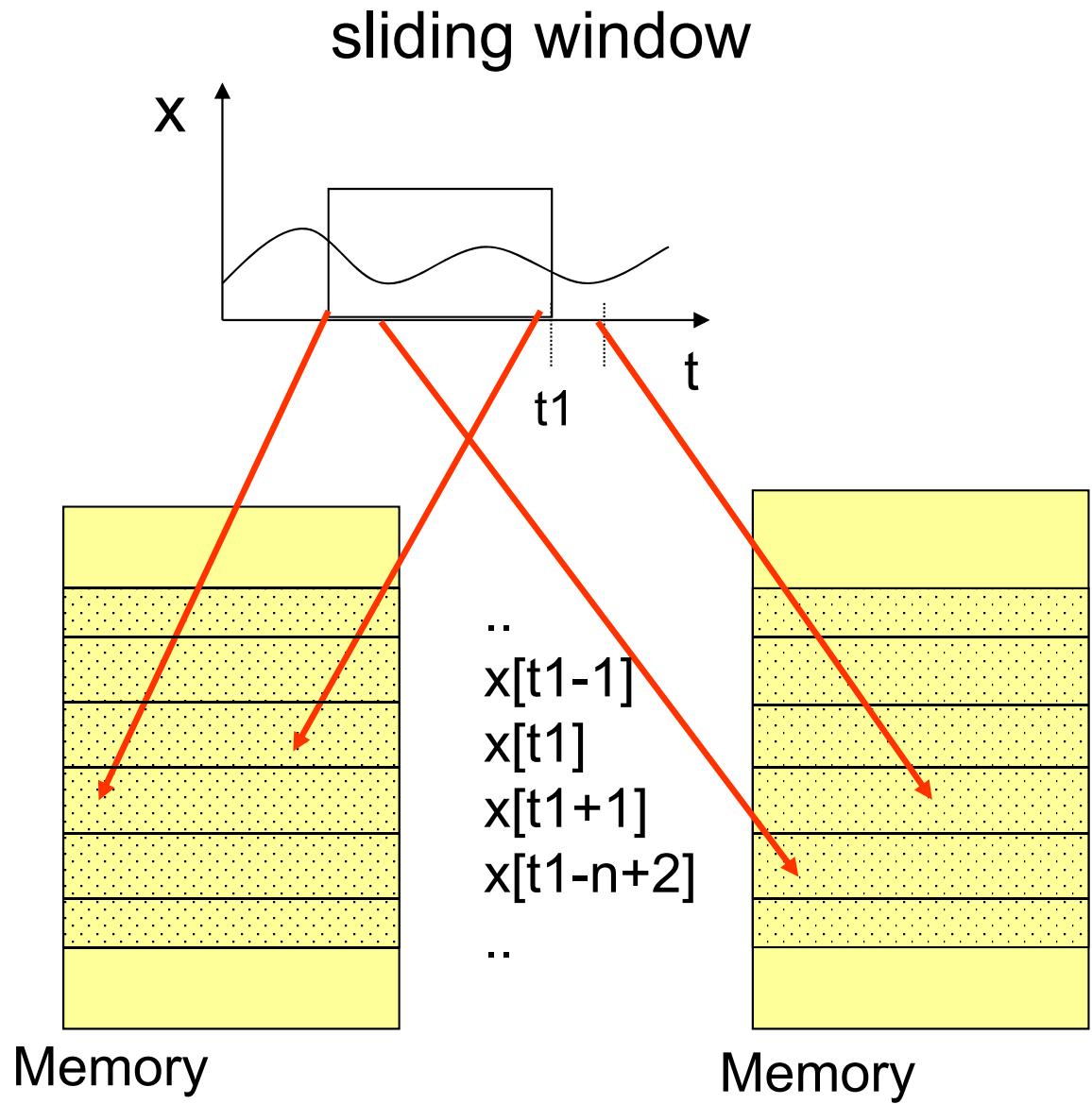
$x[t_1-1]$

$x[t_1]$

$x[t_1-n+1]$

$x[t_1-n+2]$

$\dots$



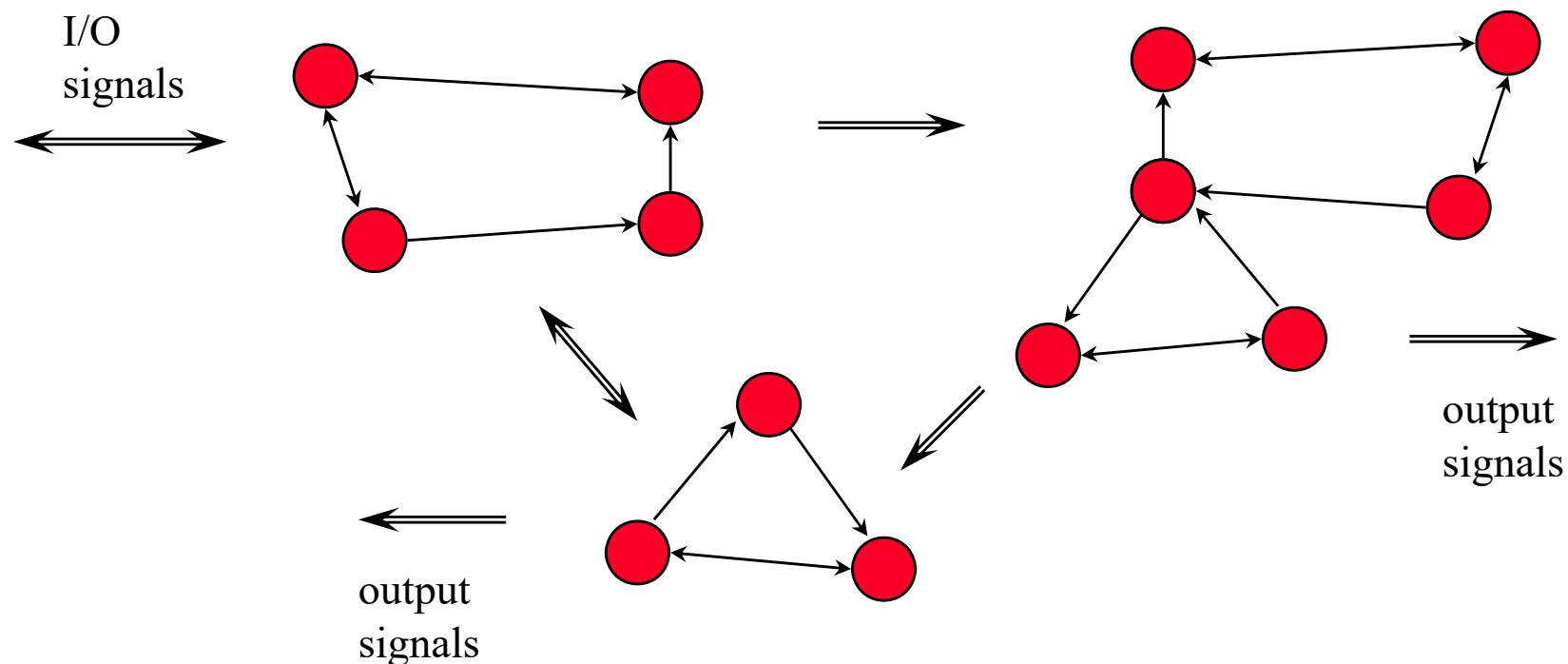
# Topics

---

- ▶ System Specialization
- ▶ Application Specific Instruction Sets
  - *Micro Controller*
  - Digital Signal Processors and VLIW
- ▶ Programmable Hardware
- ▶ ASICs
- ▶ System-on-Chip

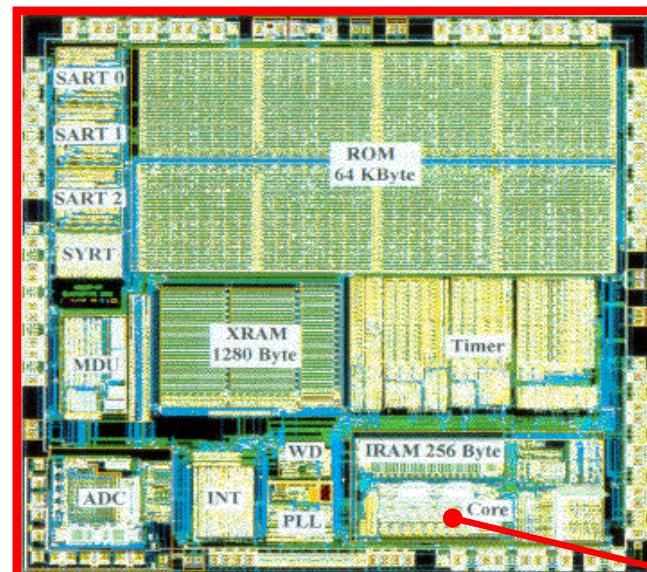
# Control Dominated Systems

- ▶ Reactive systems with ***event driven behavior***
- ▶ Underlying semantics of system description (“input model of computation”) typically (coupled) Finite State Machines or Petri Nets



# Microcontroller

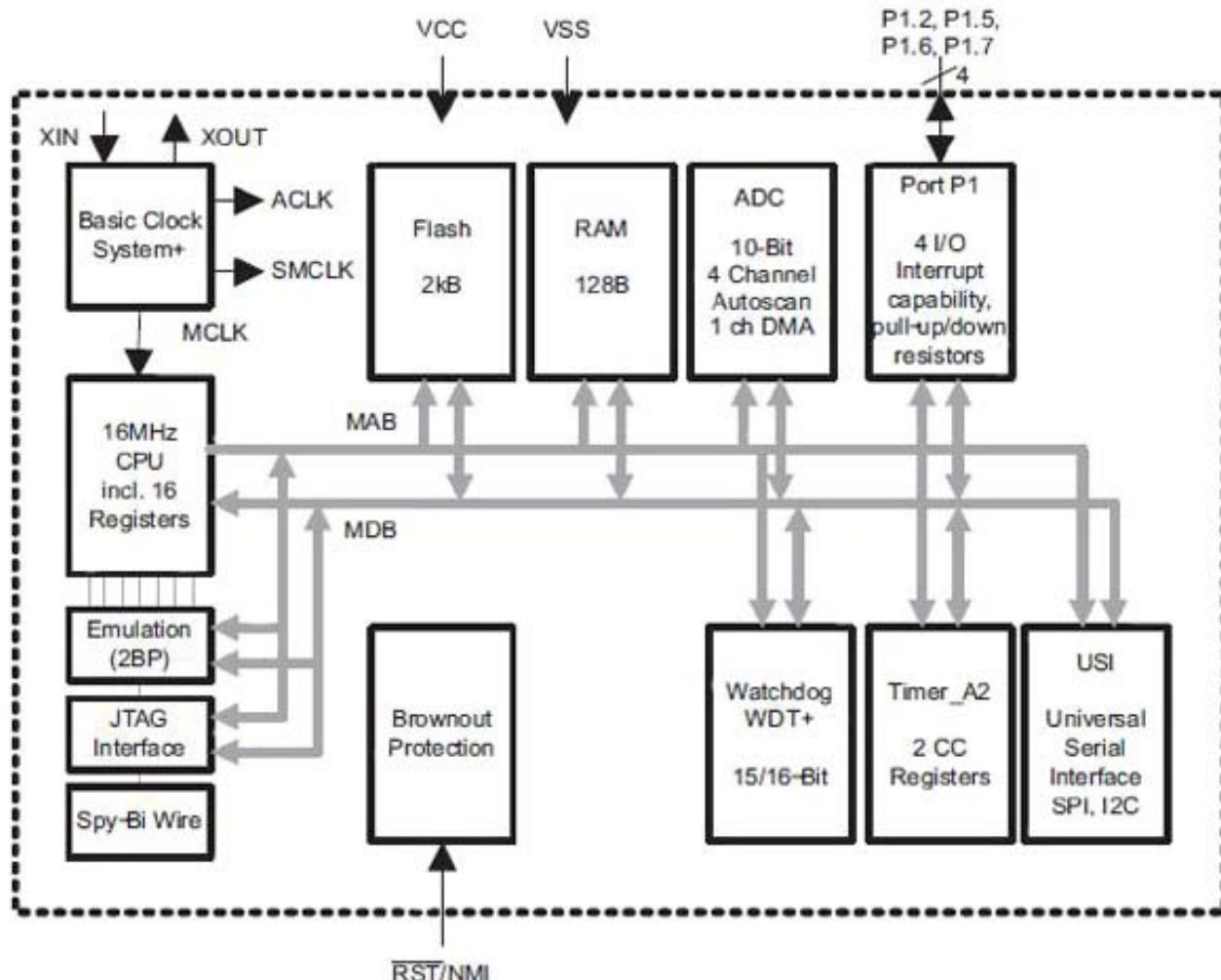
- ▶ control-dominant applications
  - supports process scheduling and synchronization
  - preemption (interrupt), context switch
  - short latency times
- ▶ low power consumption
- ▶ peripheral units often integrated
- ▶ suited for real-time applications



8051 core

SIECO51 (Siemens)

# Microcontroller as a System-on-Chip



MSP 430 RISC Processor (Microchip)

- complete system
- timers
- I<sup>2</sup>C-bus and par./ser. interfaces for communication
- A/D converter
- watchdog (SW activity timeout): safety
- on-chip memory (volatile/non-volatile)
- interrupt controller

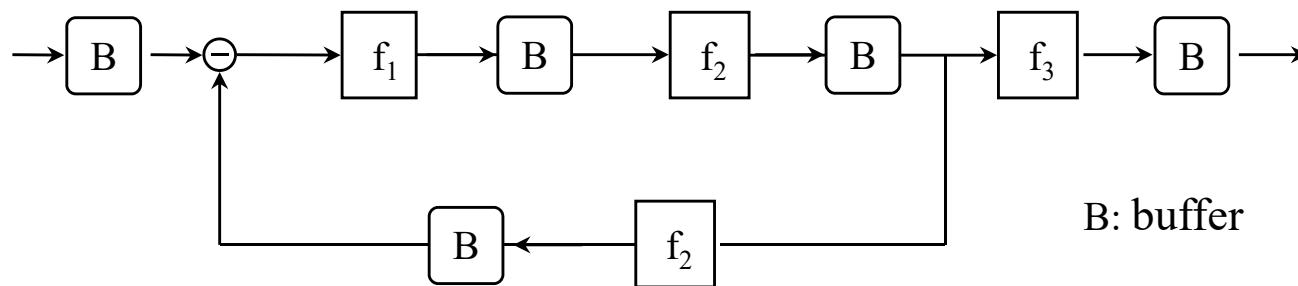
# Topics

---

- ▶ 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”)

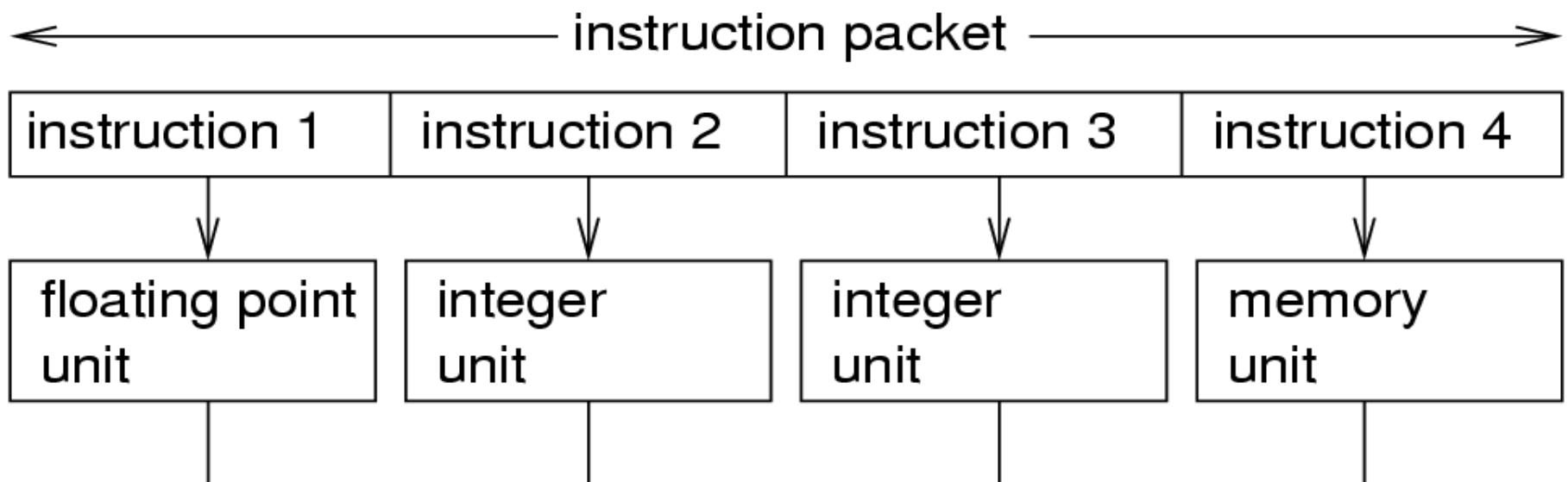


- ▶ ***Application examples:*** signal processing, control engineering

# Very Long Instruction Word (VLIW)

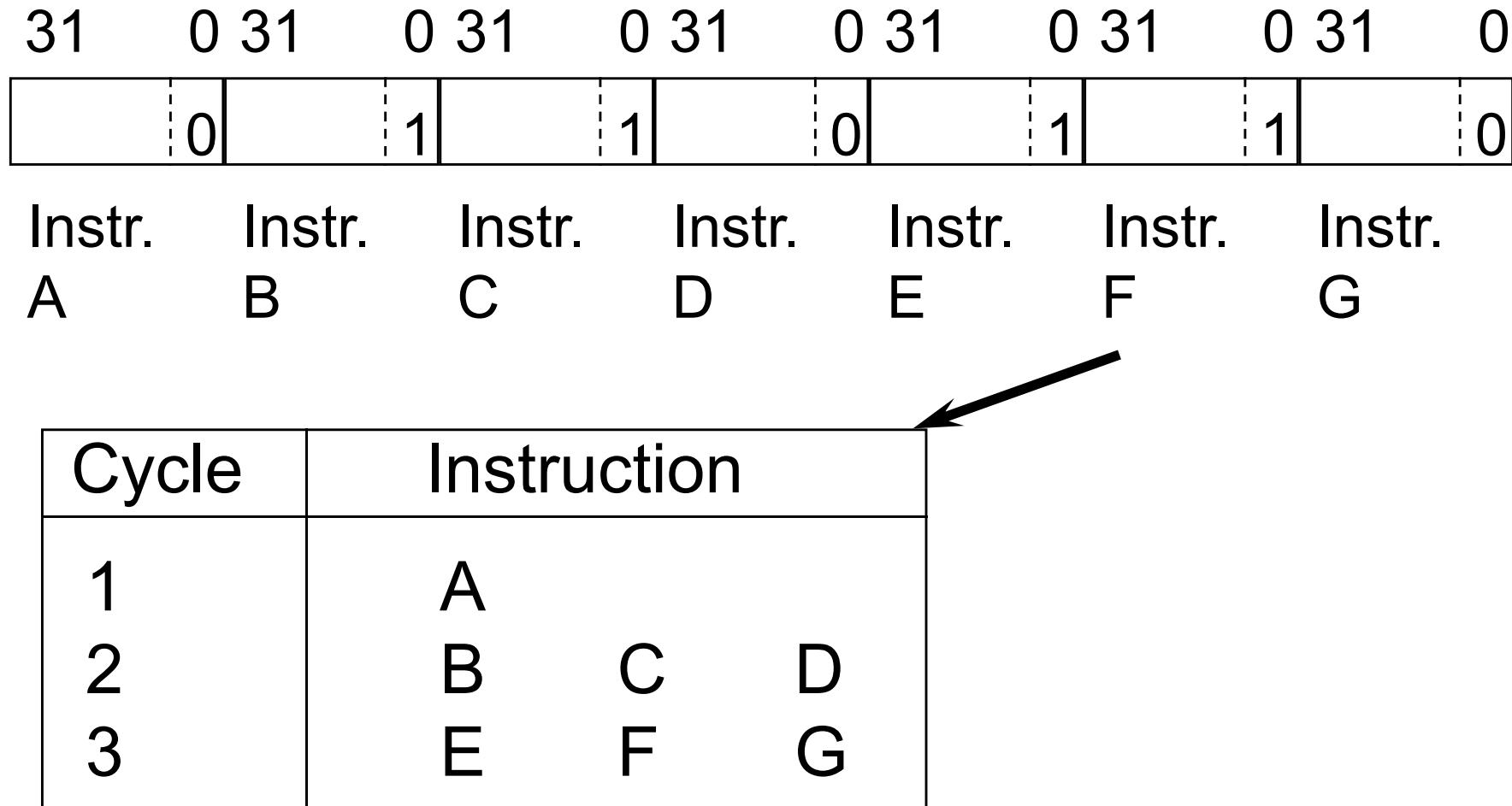
Key idea: detection of possible parallelism to be done by compiler, not by hardware at run-time (inefficient).

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



# Explicit Parallelism Instruction Computers

## The TMS320C62xx VLIW Processor as an example of EPIC:

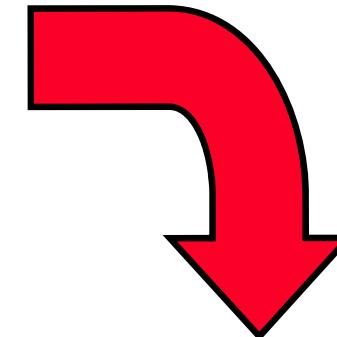


# MAC (multiply & accumulate)

```
sum = 0.0;  
for (i=0; i<N; i++)  
    sum = sum + a[i]*b[i];
```

zero-overhead loop  
(repeat next instruction N times)

MAC - Instruktion



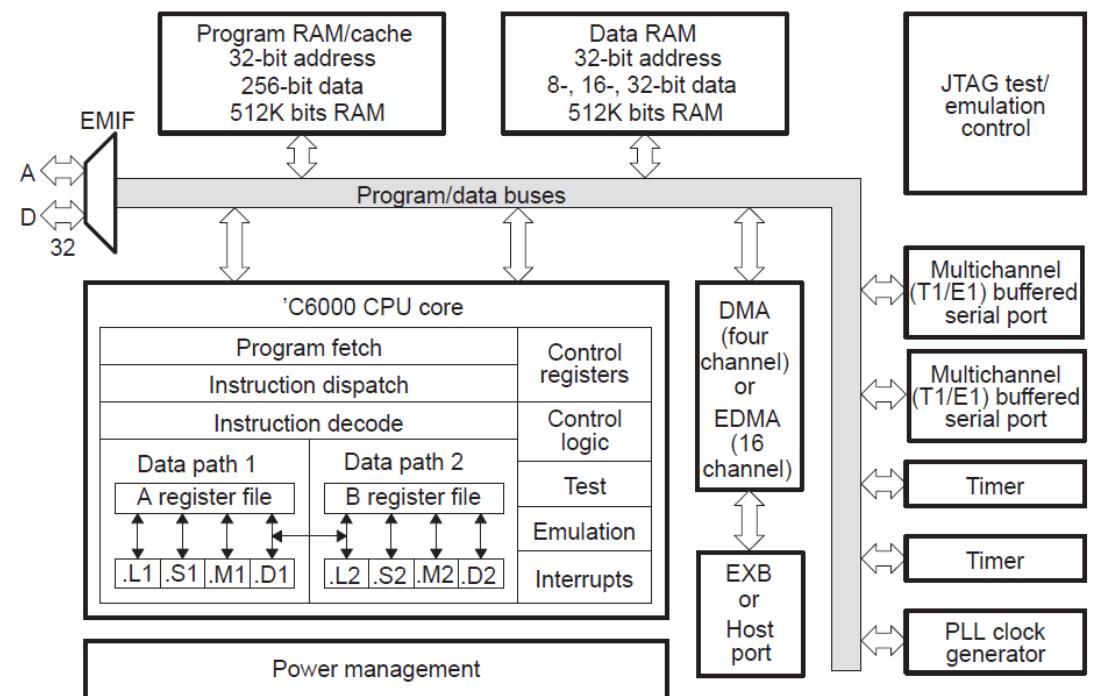
LDF	0, R0
LDF	0, R1
RPTS	N
MPYF3	* (AR0)++, * (AR1)++, R0
ADDF3	R0, R1, R1

TMS320C3x Assembler  
(Texas Instruments)

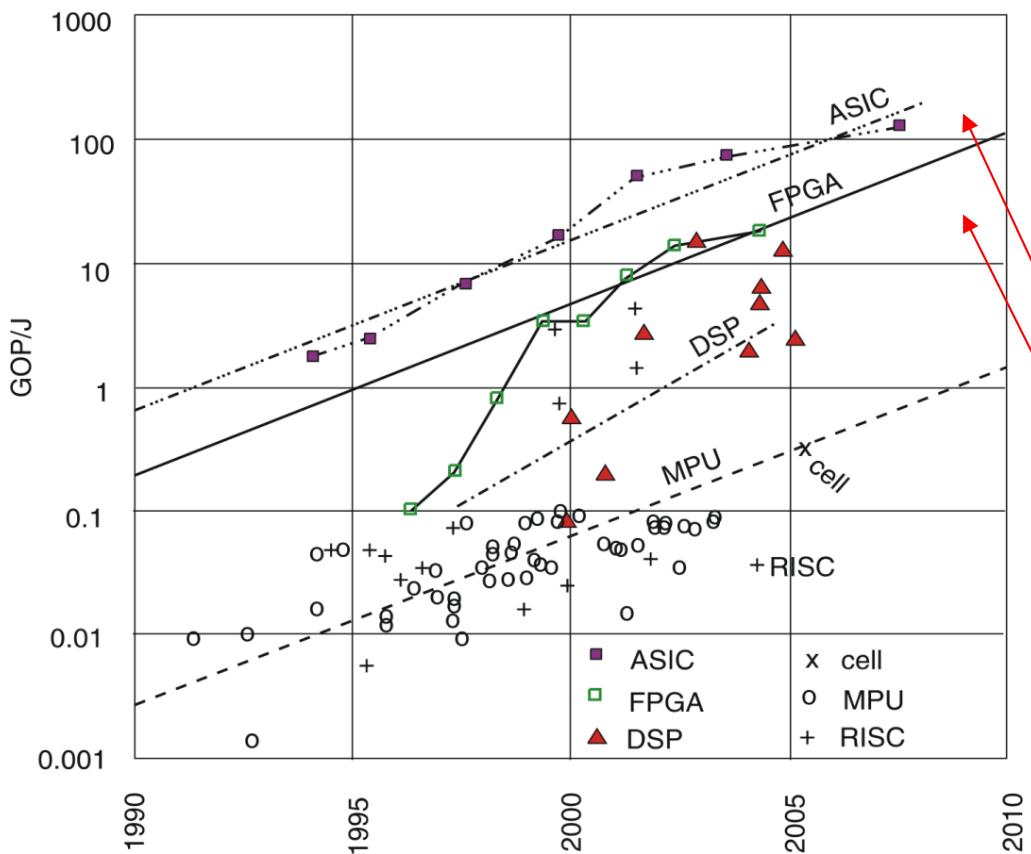
# Digital Signal Processor

- ▶ 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

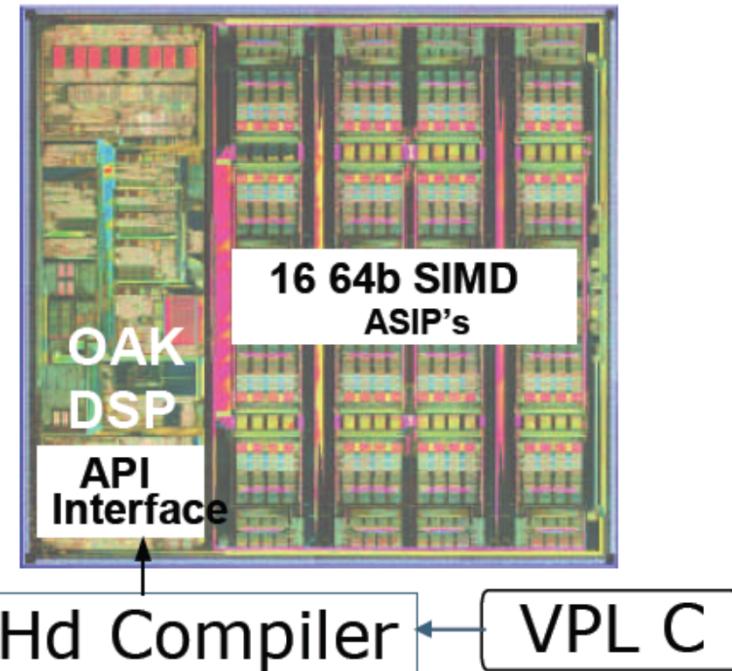
Figure 2-1. TMS320C62x/C67x Block Diagram



# Example Infineon

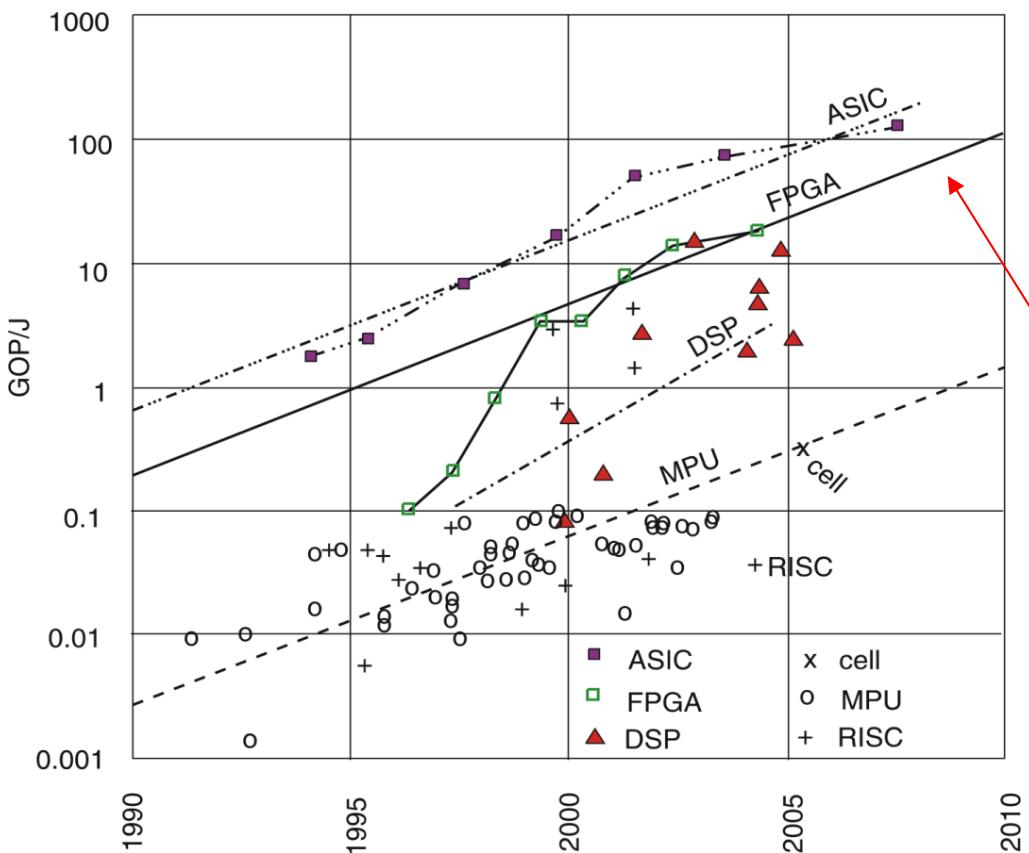


**VIP for car mirrors  
Infineon**

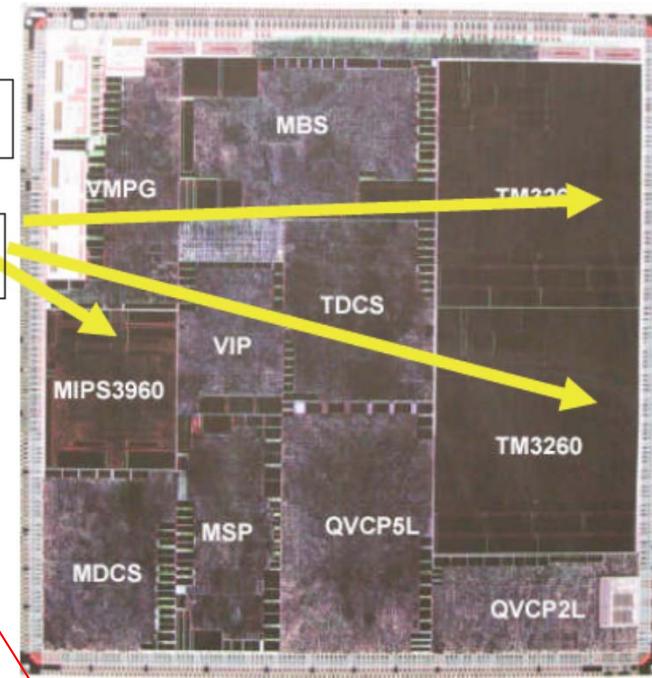


**200MHz , 0.76 Watt**  
**100Gops @ 8b**  
**25Gops @ 32b**

# Example NXP Trimedia VLIW



Nexperia Digital Video Platform  
NXP



**1 MIPS, 2 Trimedia  
60 coproc, 250 RAM's  
266MHz, 1.5 watt 100 Gops**

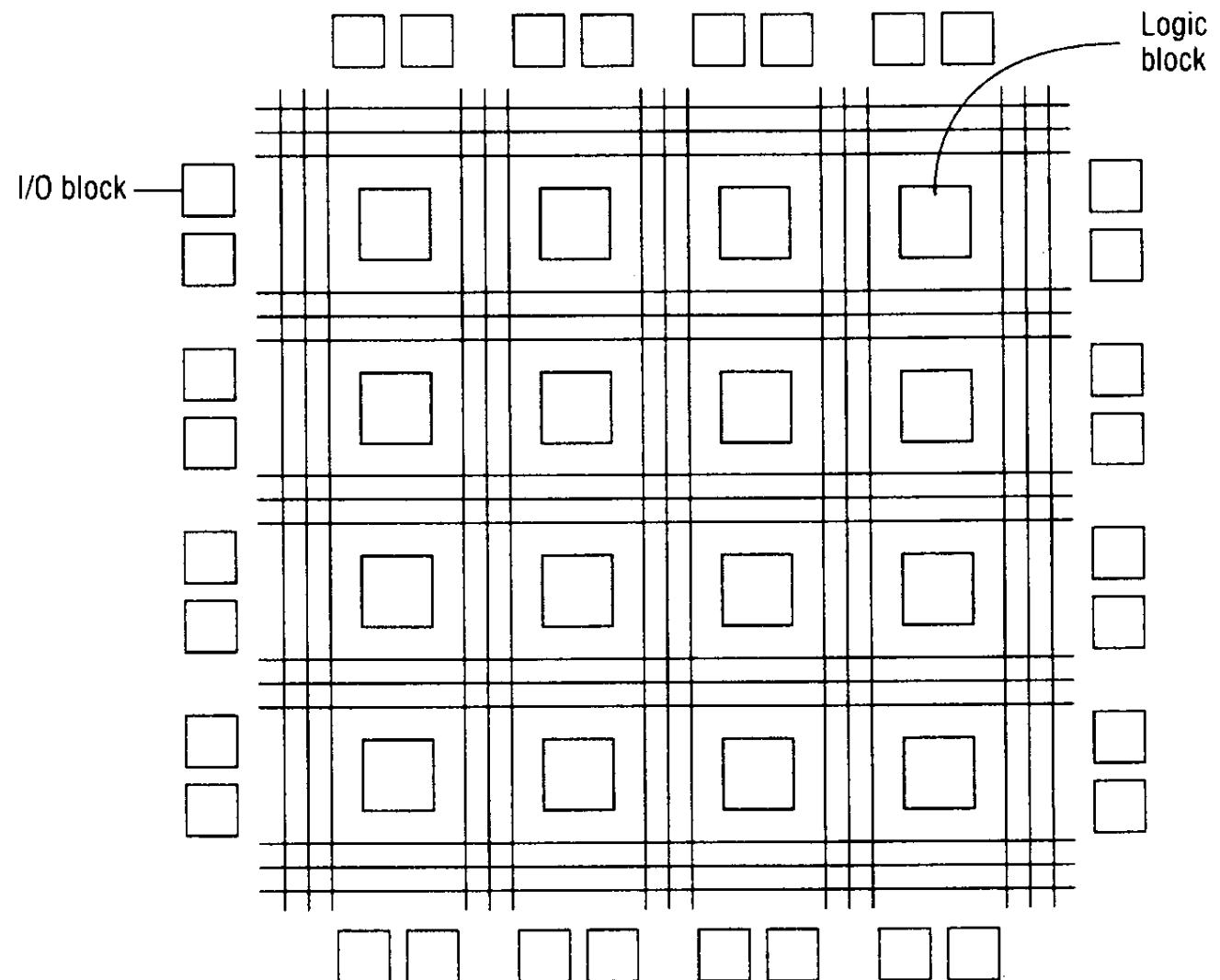
# Topics

---

- ▶ System Specialization
- ▶ Application Specific Instruction Sets
  - Micro Controller
  - Digital Signal Processors and VLIW
- ▶ ***Programmable Hardware***
- ▶ ASICs
- ▶ System-on-Chip

# FPGA – Basic Structure

- ▶ Logic Units
- ▶ I/O Units
- ▶ Connections

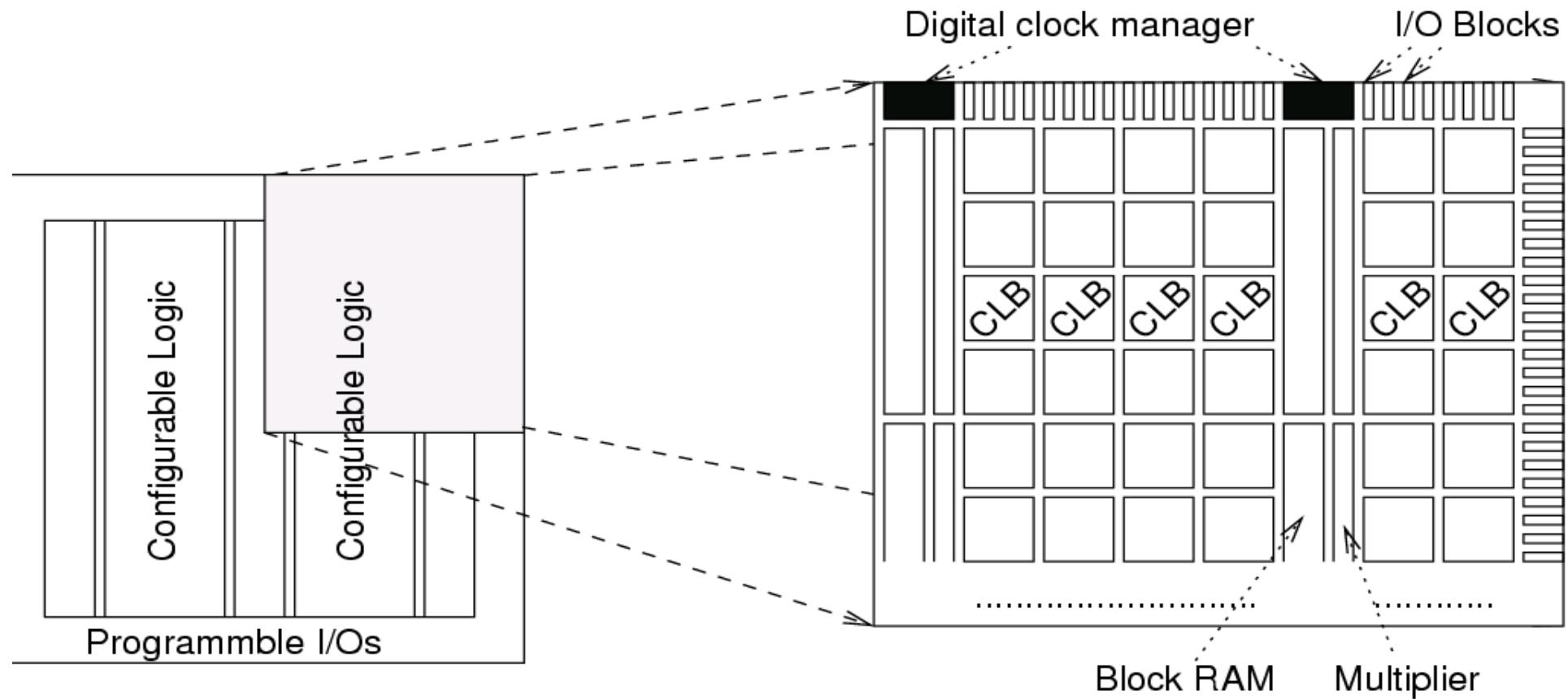


# FPGA - Classification

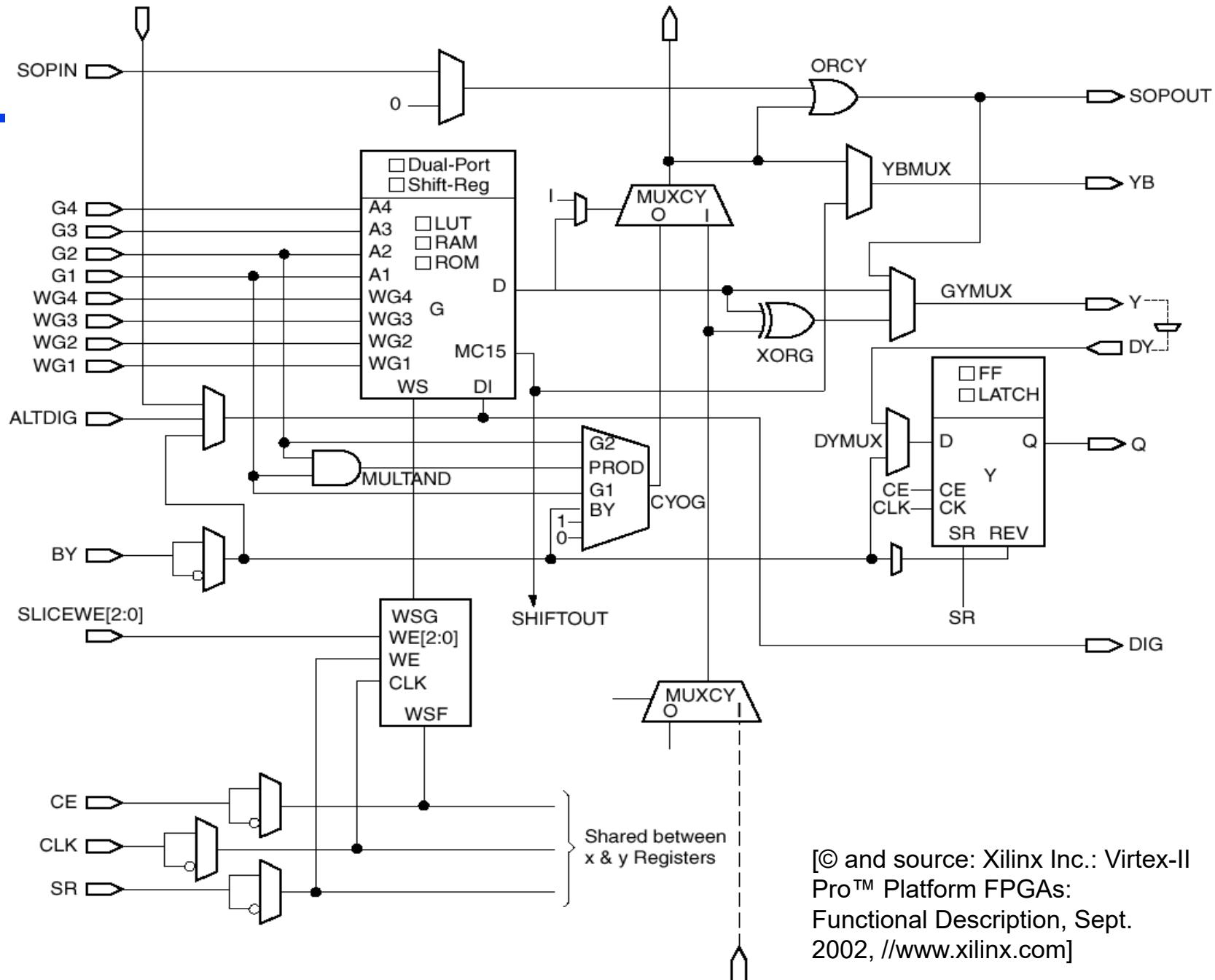
---

- ▶ ***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



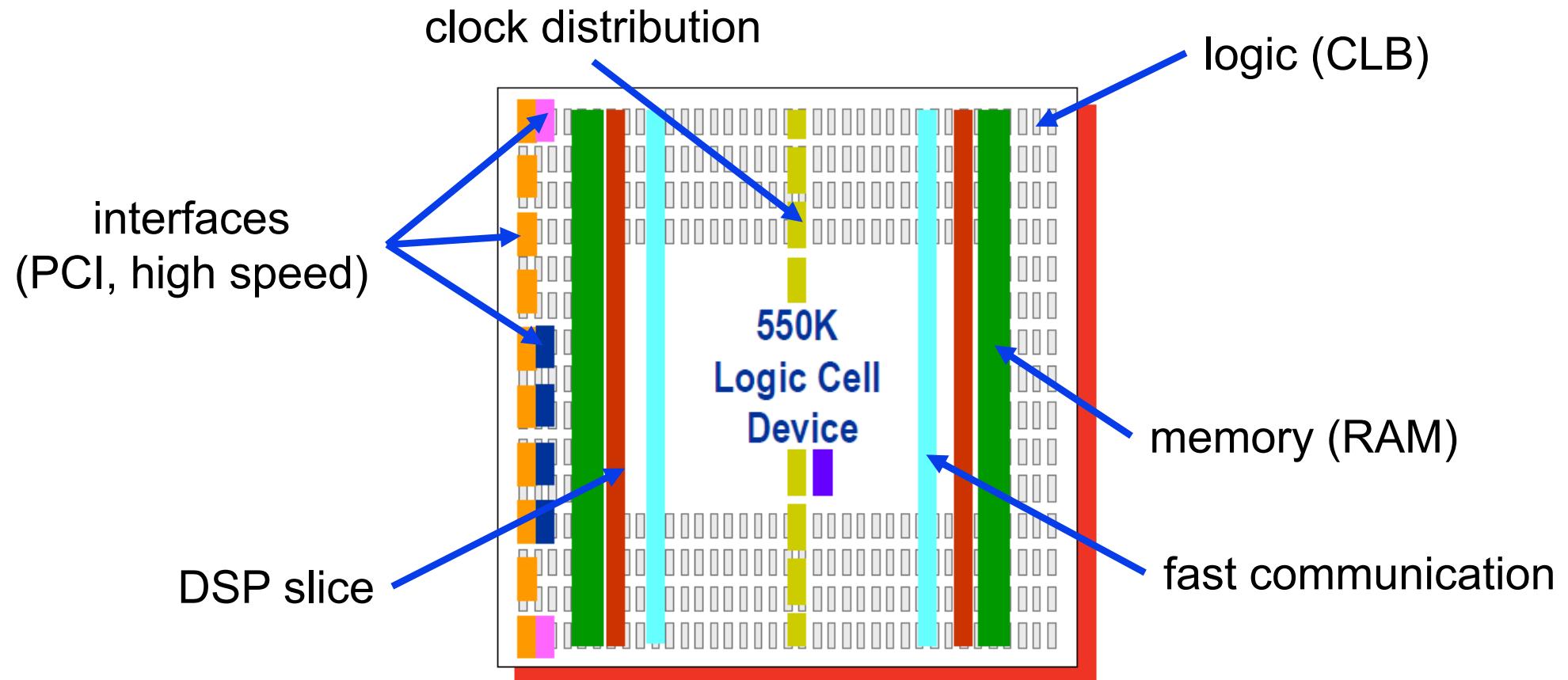
# Virtex Logic Cell



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

# Example Virtex-6

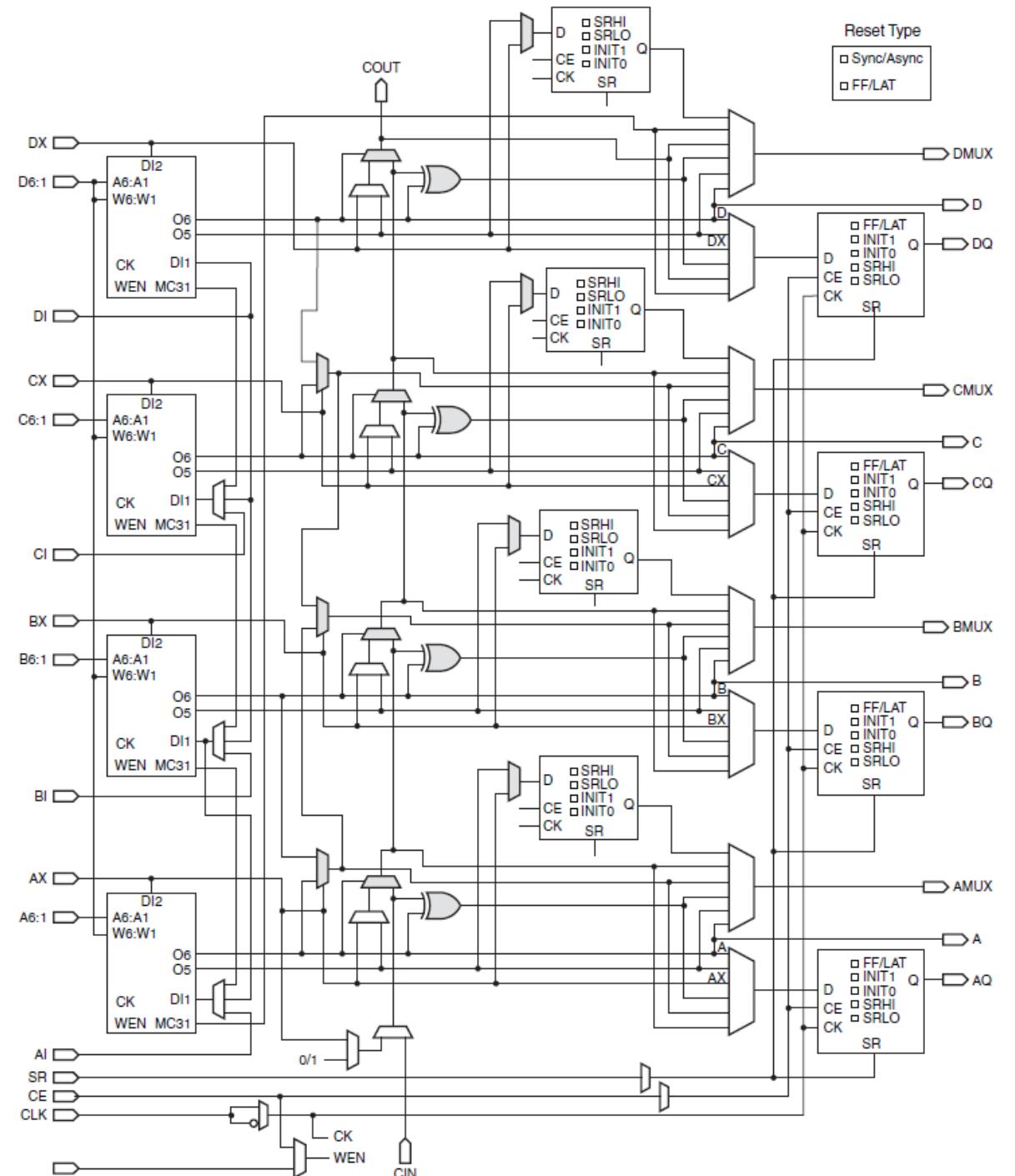
- ▶ Combination of flexibility (CLB's), Integration and performance (heterogeneity of hard-IP Blocks)



# XILINX Virtex UltraScale

Effective LEs (K)	3,435
Logic Cells (K)	2,863
UltraRAM (Mb)	432.0
Block RAM (Mb)	94.5
DSP Slices	11,904
I/O Pins	832

## Virtex-6 CLB Slice



# Topics

---

- ▶ System Specialization
- ▶ Application Specific Instruction Sets
  - Micro Controller
  - Digital Signal Processors and VLIW
- ▶ Programmable Hardware
- ▶ **ASICs**
- ▶ System-on-Chip

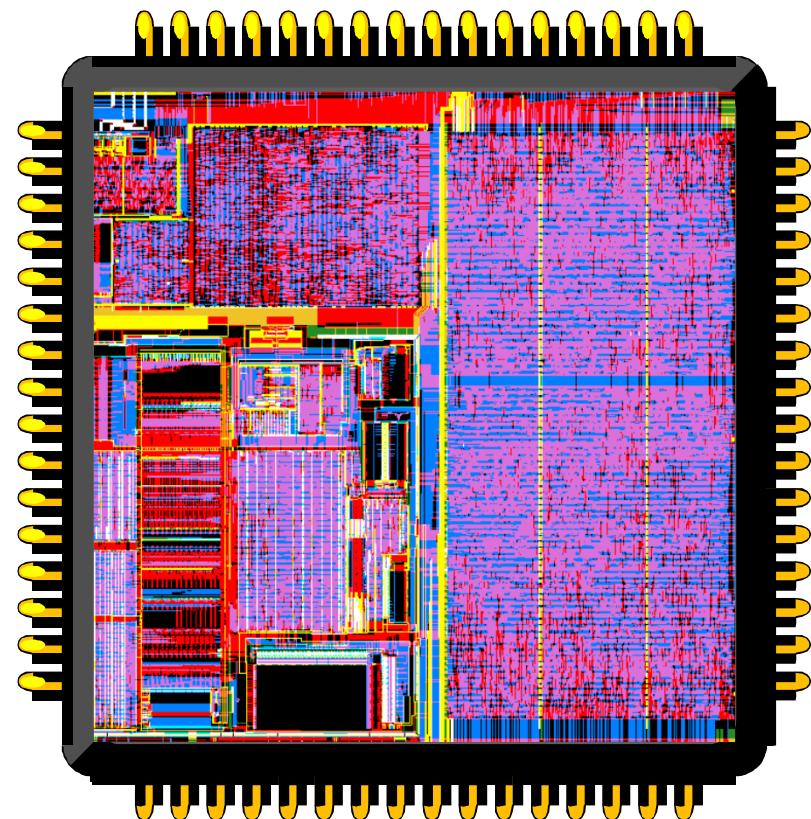
# Application Specific Circuits (ASICs)

Custom-designed circuits necessary

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

Approach suffers from

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

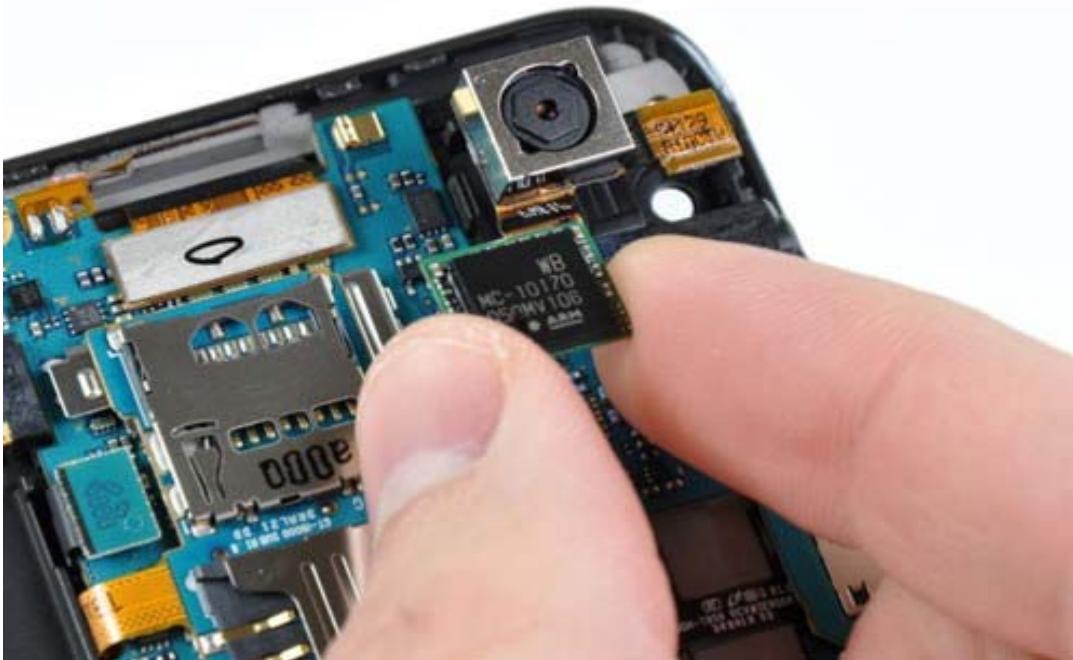


# Topics

---

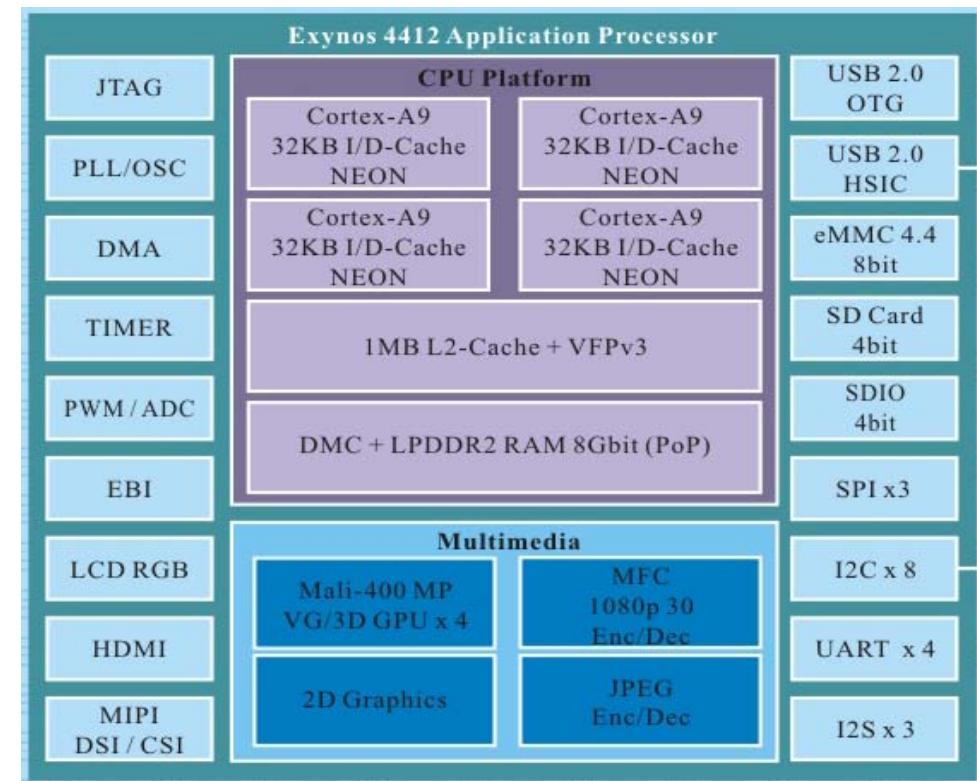
- ▶ System Specialization
- ▶ Application Specific Instruction Sets
  - Micro Controller
  - Digital Signal Processors and VLIW
- ▶ Programmable Hardware
- ▶ ASICs
- ▶ ***System-on-Chip***

# System-on-Chip



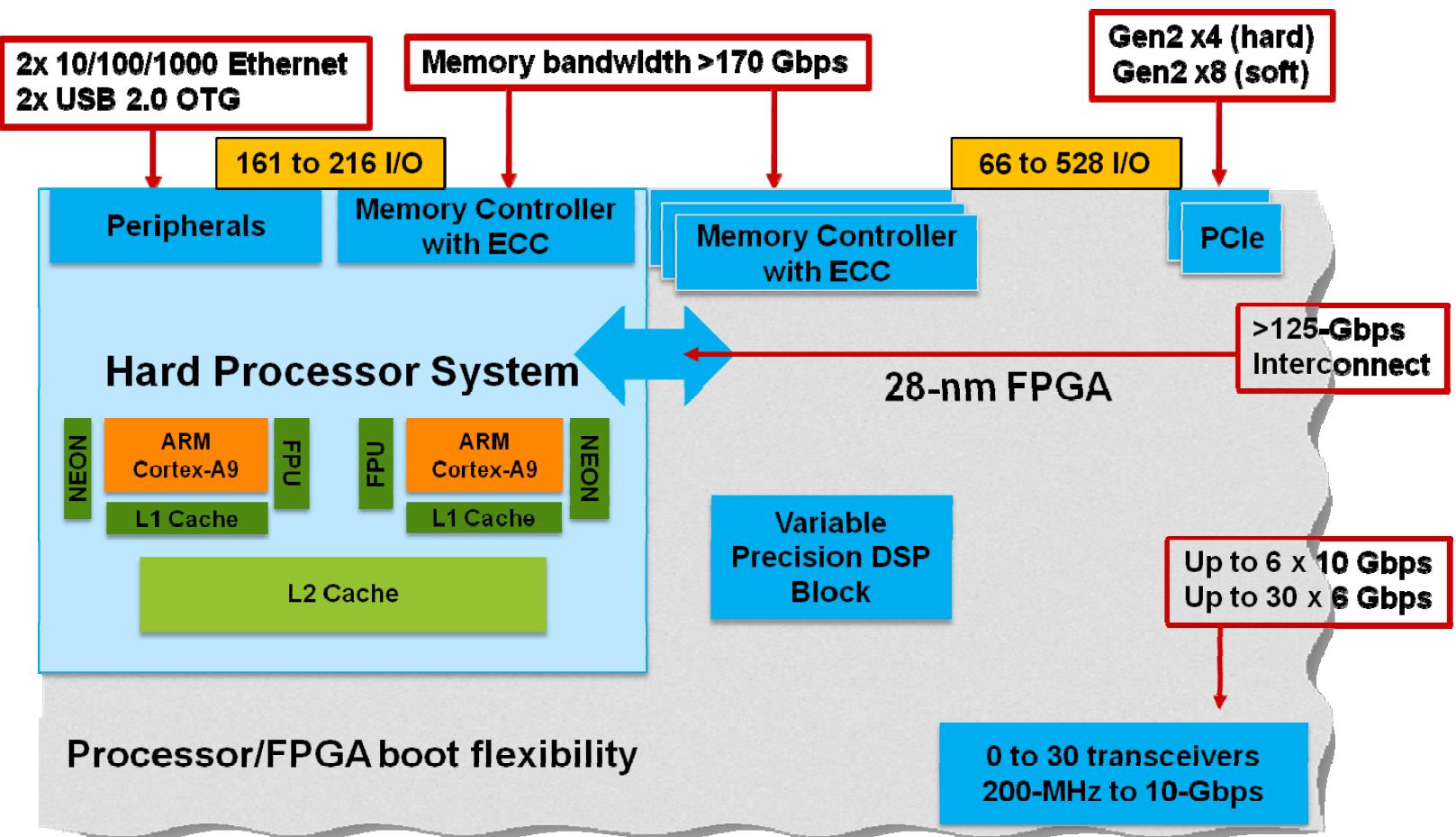
Samsung Galaxy Note II

- Eynos 4412 System on a Chip (SoC)
- ARM Cortex-A9 processing core
- 32 nanometer: transistor gate width
- Four processing cores



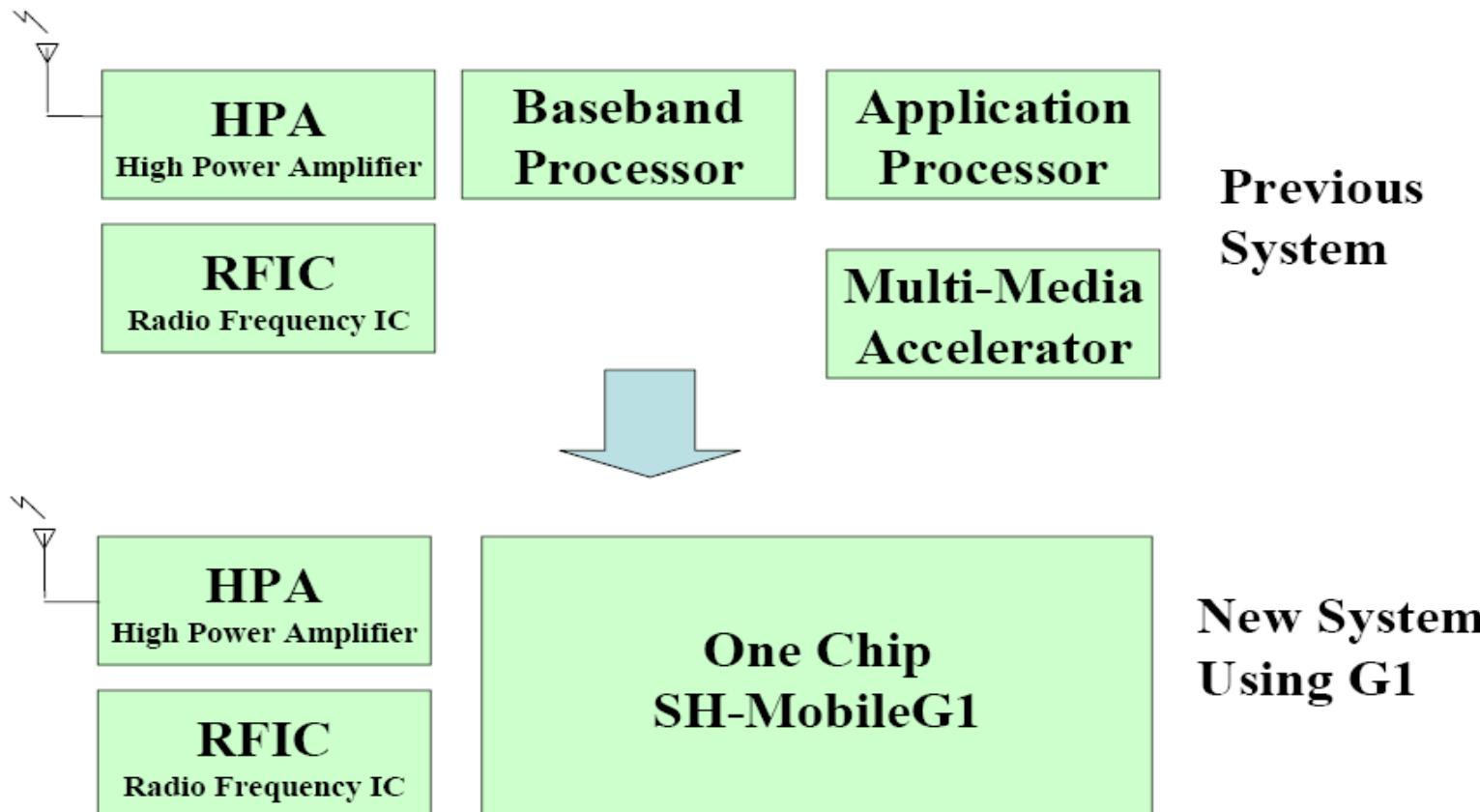
# Configurable System-On-Chip

**Example:**  
Altera's SoC  
FPGA integrates a  
dual-core ARM  
Cortex-A9  
processor system  
with a low power  
FPGA fabrics



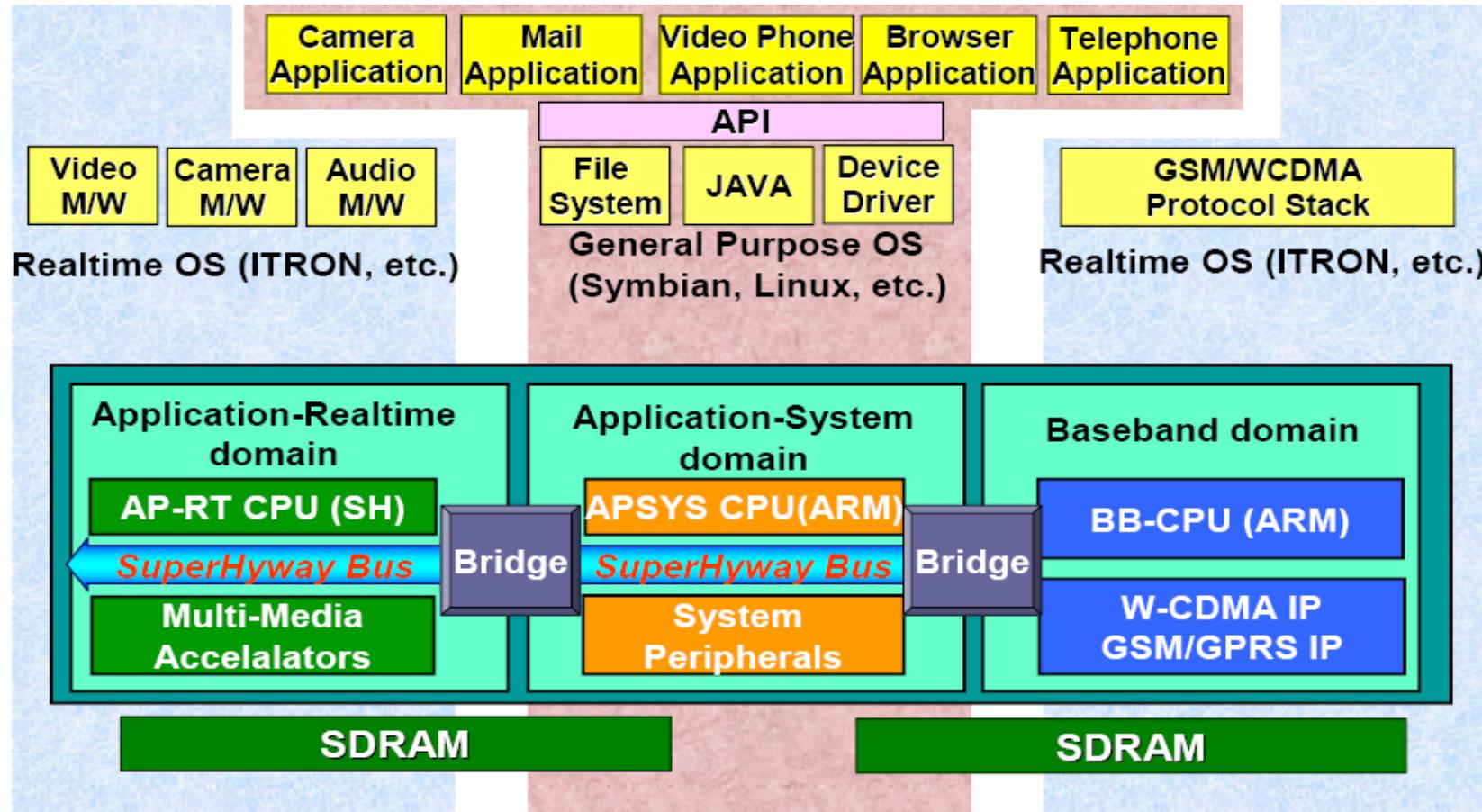
# Trend: multiprocessor systems-on-a-chip (MPSoCs)

## 3G Multi-Media Cellular Phone System



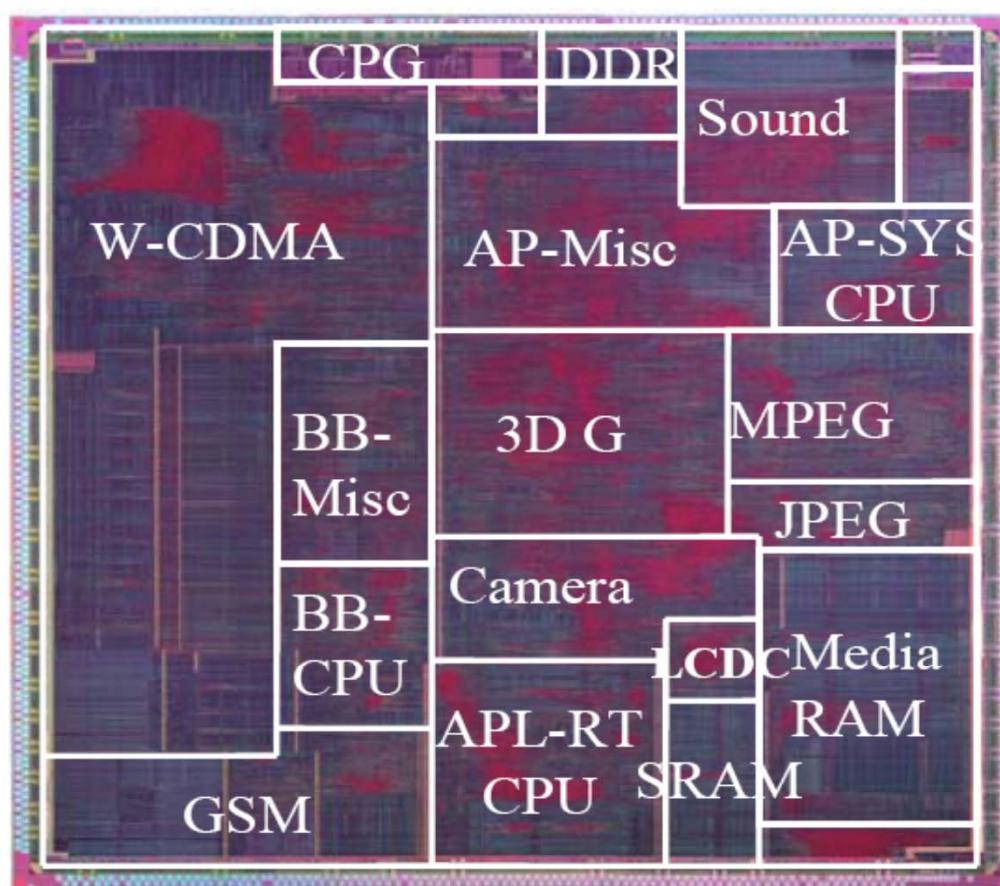
# Multiprocessor systems-on-a-chip (MPSoCs)

## A Sample of System Architecture using G1



# Multiprocessor systems-on-a-chip (MPSoCs)

## SH-MobileG1: Chip Overview



Die size	11.15mm x 11.15mm
Process	90nm LP 8M(7Cu+1Al) CMOS dual-Vth
Supply voltage	1.2V(internal), 1.8/2.5/3.3V(I/O)
# of TRs, gate, memory	181M TRs, 13.5M Gate 20.2 Mbit mem

# Multiprocessor systems-on-a-chip (MPSoCs)

## G1 Module Diagram

