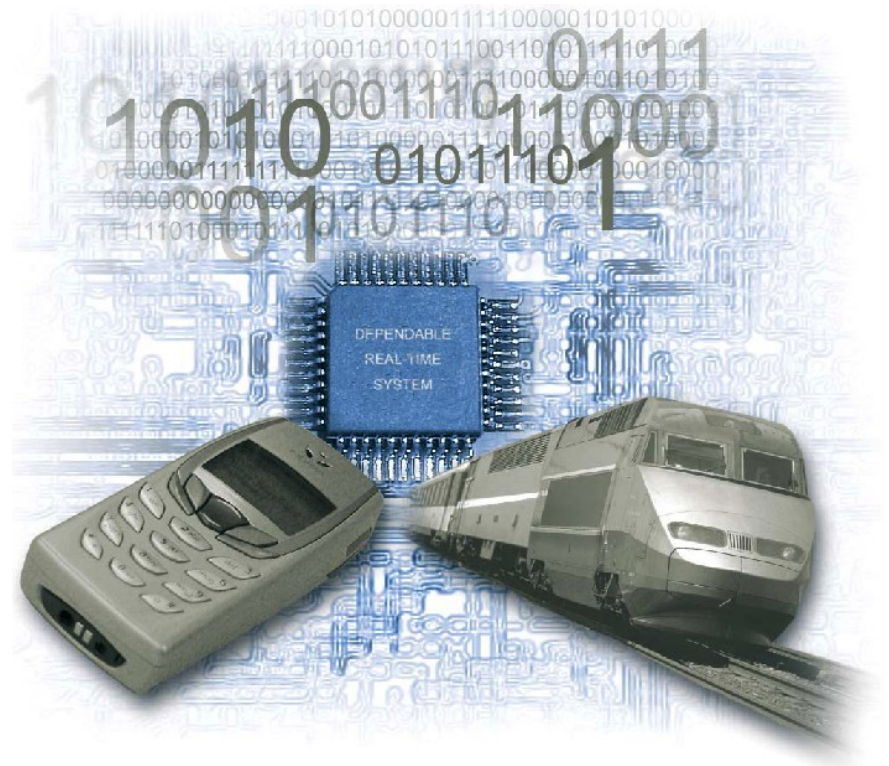


# Embedded System Hardware - Processing -

Peter Marwedel  
Informatik 12  
TU Dortmund  
Germany

2009/11/18



# Very long instruction word (VLIW) architectures

---

- Very long instruction word (“instruction packet”) contains several instructions, all of which are assumed to be executed in parallel.
- Compiler is assumed to generate these “parallel” packets
- Complexity of finding parallelism is moved from the hardware (RISC/CISC processors) to the compiler; Ideally, this avoids the overhead (silicon, energy, ..) of identifying parallelism at run-time.
- ☞ A lot of expectations into VLIW machines
- Explicitly parallel instruction set computers (EPICs) are an extension of VLIW architectures: parallelism detected by compiler, but no need to encode parallelism in 1 word.

# Energy savings for VLIW-architectures

## Basic equations

Power:

$$P \sim V_{DD}^2,$$

Maximum clock frequency:

$$f \sim V_{DD},$$

Energy to run a program:

$$E = P \times t, \text{ with: } t = \text{runtime (fixed)}$$

Time to run a program:

$$t \sim 1/f$$

## Changes due to VLIW processing, with $\alpha$ operations per clock:

Clock frequency reduced to:

$$f' = f / \alpha,$$

Voltage can be reduced to:

$$V_{DD}' = V_{DD} / \alpha,$$

Power for VLIW processing:

$$P^\circ = P / \alpha^2 \text{ per operation,}$$

Power for  $\alpha$  operations per clock:

$$P' = \alpha \times P^\circ = P / \alpha,$$

Time to run a program is still:

$$t' = t,$$

Energy required to run program:

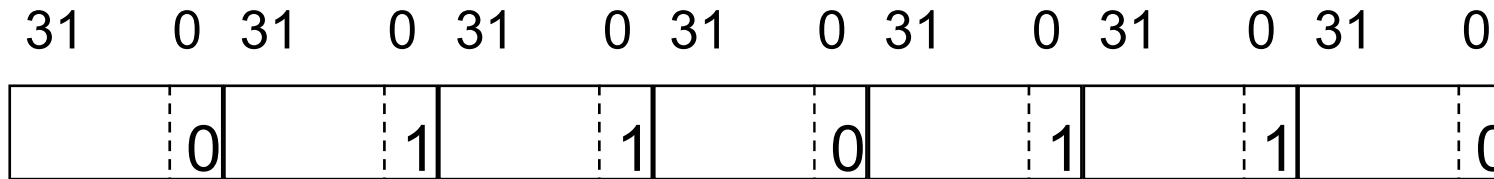
$$E' = P' \times t = E / \alpha$$

👉 **VLIW processors popular for low power applications.**

▪ Same argument used for multi-cores

# EPIC: TMS 320C6xx as an example

Bit in each instruction encodes end of parallel execution



Instr. A    Instr. B    Instr. C    Instr. D    Instr. E    Instr. F    Instr. G

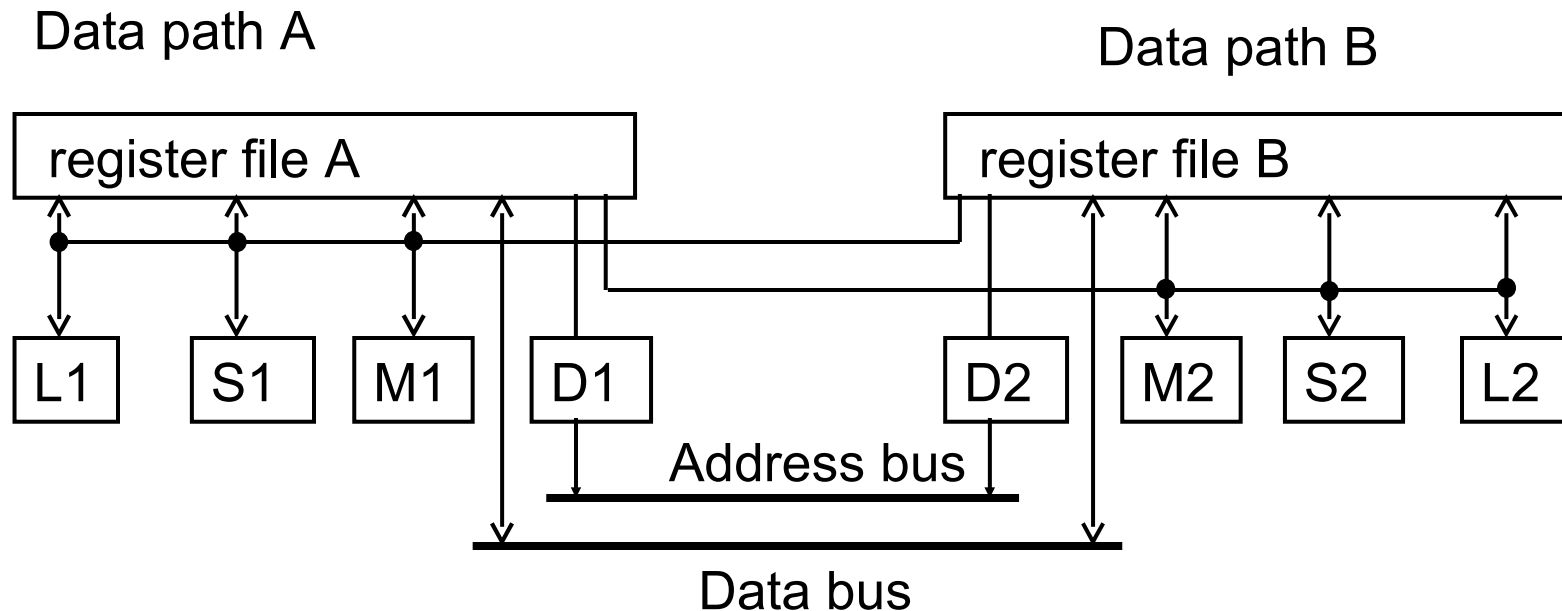
Cycle	Instruction		
1	A		
2	B	C	D
3	E	F	G

Instructions B, C and D use disjoint functional units, cross paths and other data path resources. The same is also true for E, F and G.

Parallel execution cannot span several packets.

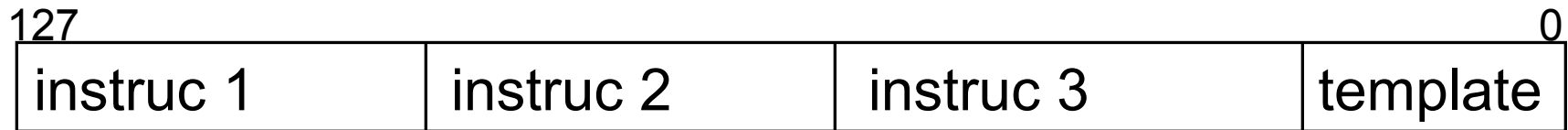
# Partitioned register files

- Many memory ports are required to supply enough operands per cycle.
  - Memories with many ports are expensive.
- ☞ Registers are partitioned into (typically 2) sets, e.g. for TI C60x:



# More encoding flexibility with IA-64 Itanium

3 instructions per **bundle**:



There are 5 instruction types:

- A: common ALU instructions
- I: more special integer instructions (e.g. shifts)
- M: Memory instructions
- F: floating point instructions
- B: branches

*Instruction  
grouping  
information*

The following combinations can be encoded in templates:

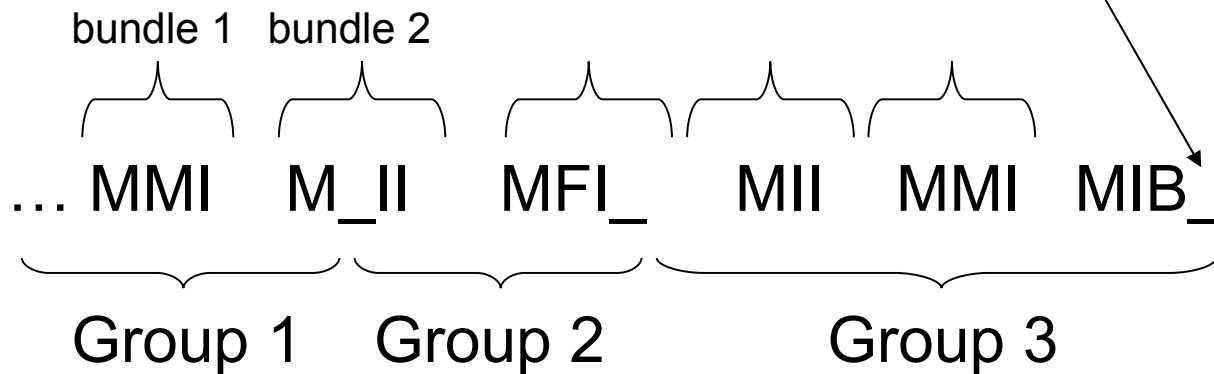
- MII, MMI, MFI, MIB, MMB, MFB, MMF, MBB, BBB, MLX  
with LX = *move 64-bit immediate* encoded in 2 slots

# Templates and instruction types

End of parallel execution called **stops**.

Stops are denoted by underscores.

Example:



Very restricted placement of stops within bundle.

Parallel execution within groups possible.

Parallel execution can span several bundles

# Instruction types are mapped to functional unit types

---

There are 4 functional unit (FU) types:

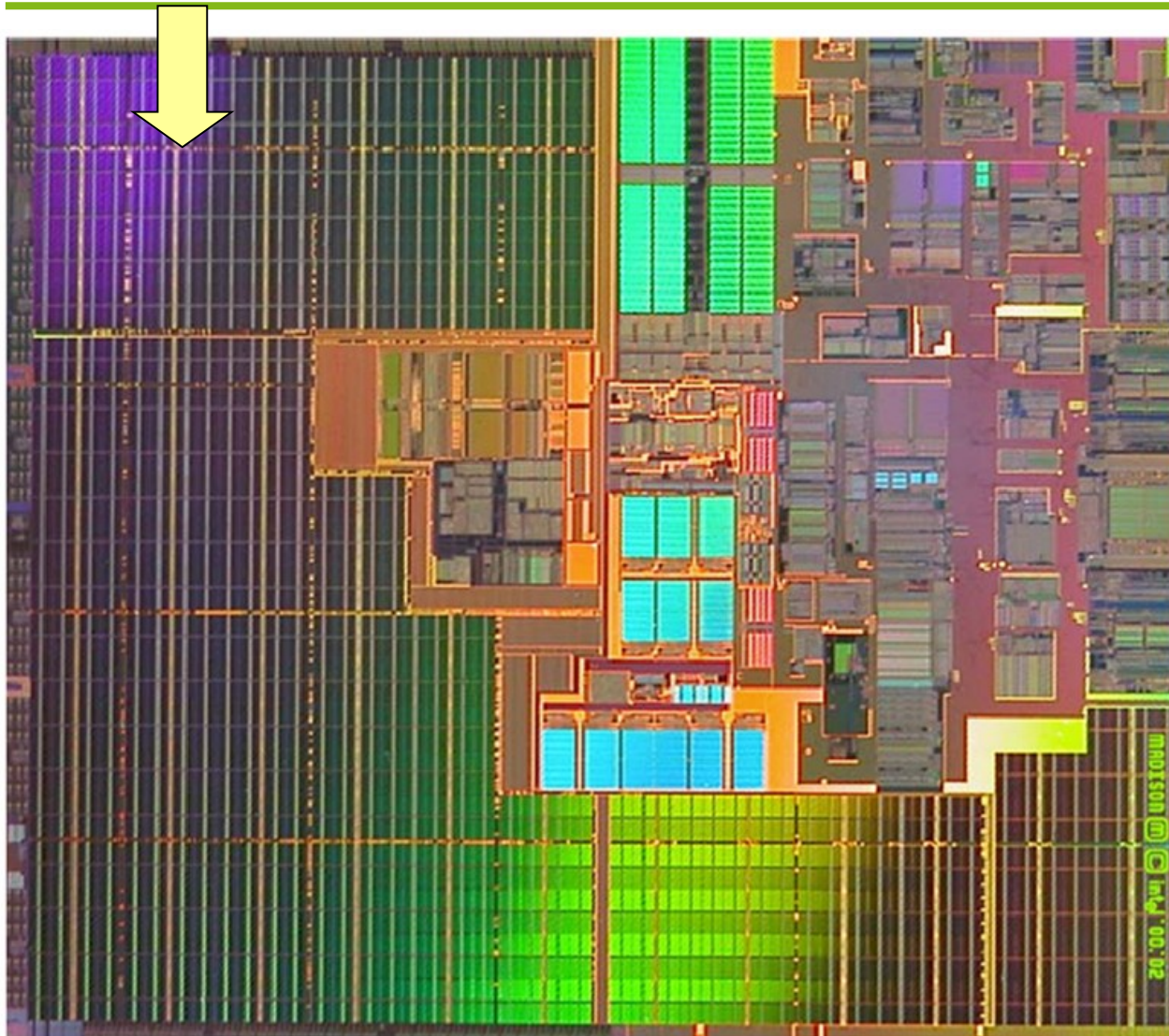
- M: Memory Unit
- I: Integer Unit
- F: Floating-Point Unit
- B: Branch Unit

Instruction types → corresponding FU type,  
except type A (mapping to either I or M-functional units).



# Implementation: Itanium 2 (2003)

L3 cache

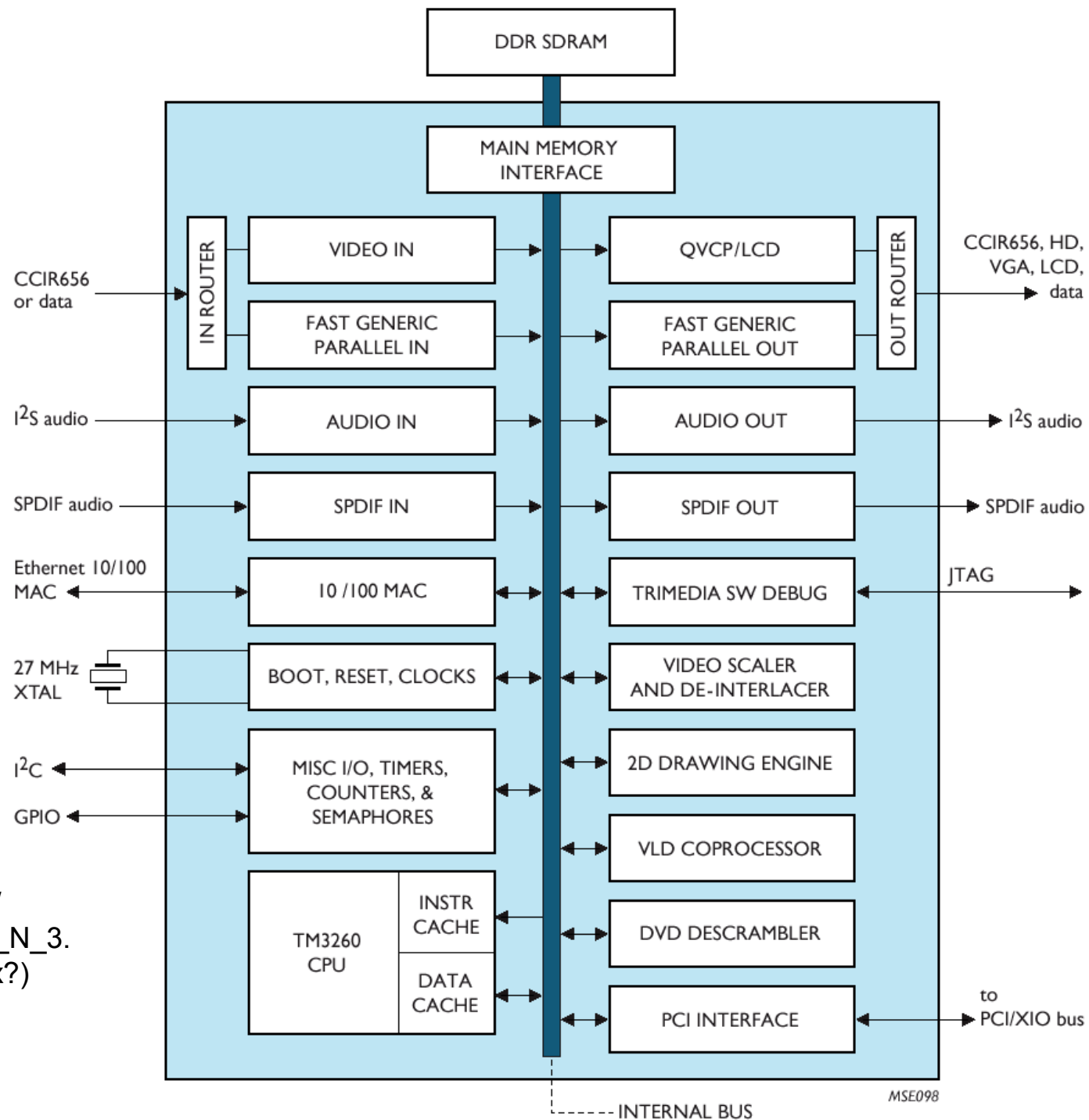


- 410M transistors
- 374 mm<sup>2</sup> die size
- 6MB on-die L3 cache
- 1.5 GHz at 1.3V

[[ftp://download.intel.com/design/itanium2/download/madison\\_slides\\_r1.pdf](ftp://download.intel.com/design/itanium2/download/madison_slides_r1.pdf)]

# Philips TriMedia- Processor

For multimedia-applications, up to 5 instructions/cycle.



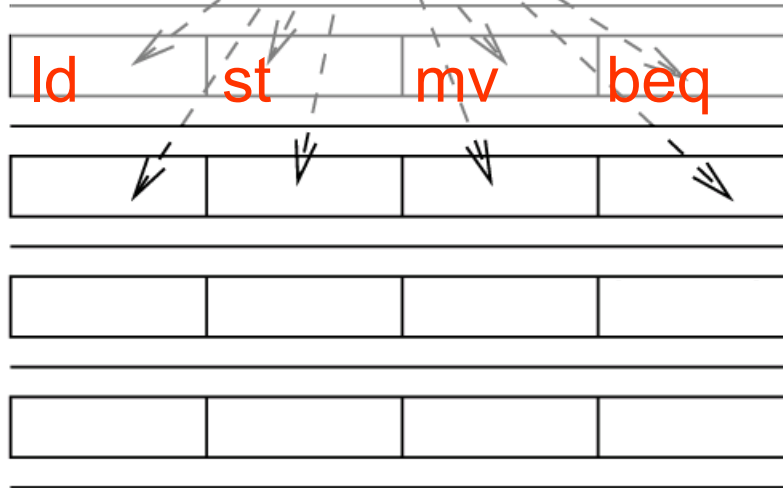
[http://www.nxp.com/acrobat/datasheets/PNX15XX\\_SER\\_N\\_3.pdf](http://www.nxp.com/acrobat/datasheets/PNX15XX_SER_N_3.pdf) (incompatible with firefox?)  
© NXP

# Large # of delay slots, a problem of VLIW processors

add    sub    and    or

sub    mult    xor    div

pipeline stages

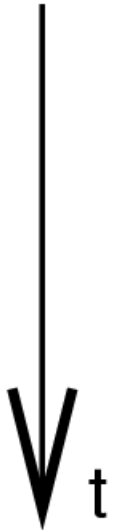


instruction fetch

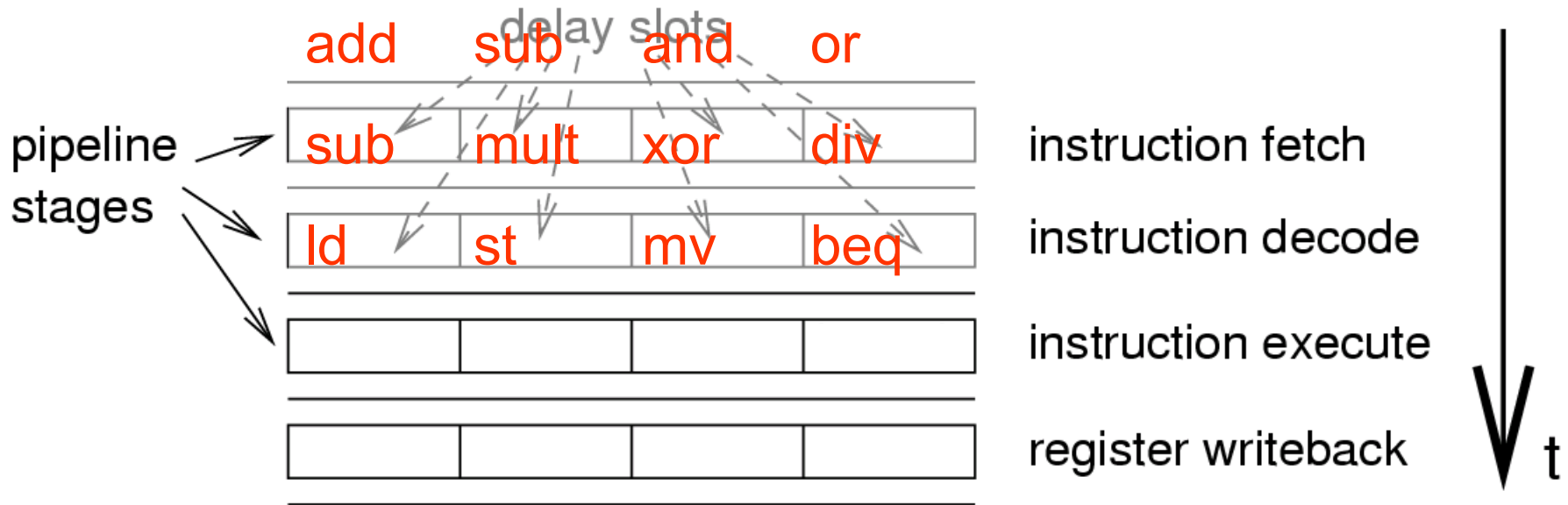
instruction decode

instruction execute

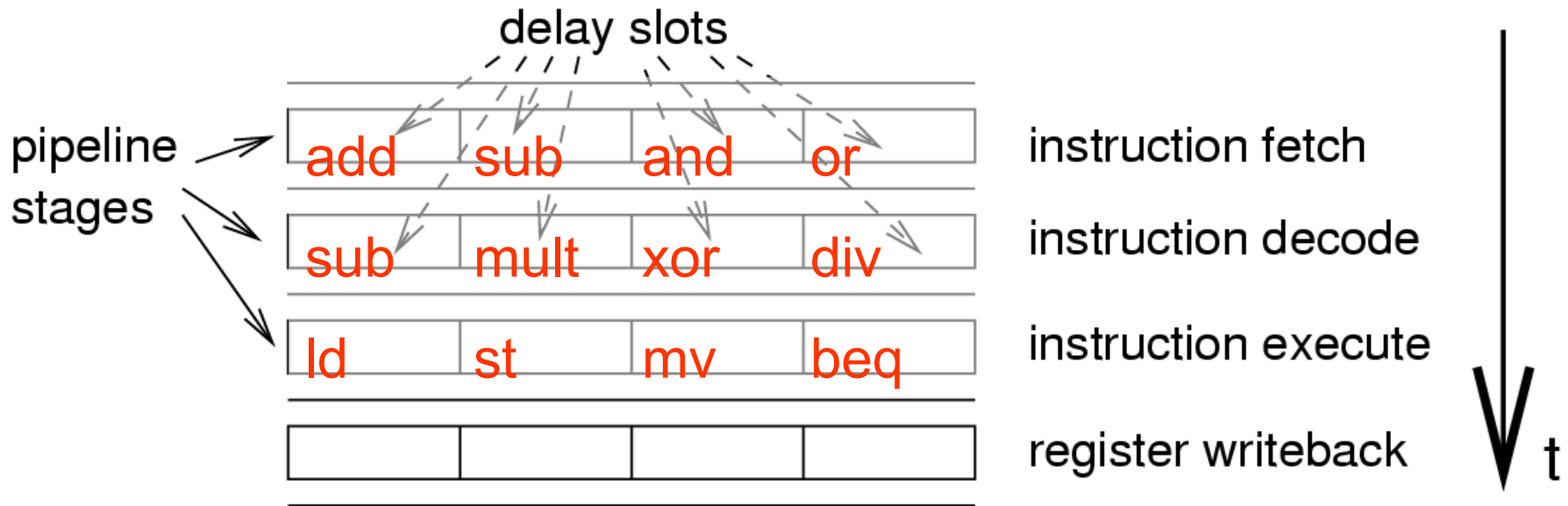
register writeback



# Large # of delay slots, a problem of VLIW processors



# Large # of delay slots, a problem of VLIW processors



The execution of many instructions has been started before it is realized that a branch was required.

Nullifying those instructions would waste compute power

- ☞ Executing those instructions is declared a feature, not a bug.
- ☞ How to fill all “delay slots” with useful instructions?
- ☞ Avoid branches wherever possible.

# Predicated execution: Implementing IF-statements “branch-free”

---

Conditional Instruction “[c] I” consists of:

- condition c
- instruction I

**c = true => I executed**  
**c = false => NOP**

# Predicated execution: Implementing IF-statements “branch-free“: TI C6x

```
if (c)
{ a = x + y;
  b = x + z;
}
else
{ a = x - y;
  b = x - z;
}
```

## Conditional branch

```
          [c] B L1
              NOP 5
              B L2
              NOP 4
              SUB x,y,a
L1:        || SUB x,z,b
              ADD x,y,a
              || ADD x,z,b
L2:
```

max. 12 cycles

## Predicated execution

```
          [c] ADD x,y,a
|| [c] ADD x,z,b
|| [!c] SUB x,y,a
|| [!c] SUB x,z,b
```

1 cycle

# Microcontrollers

## - MHS 80C51 as an example -

---

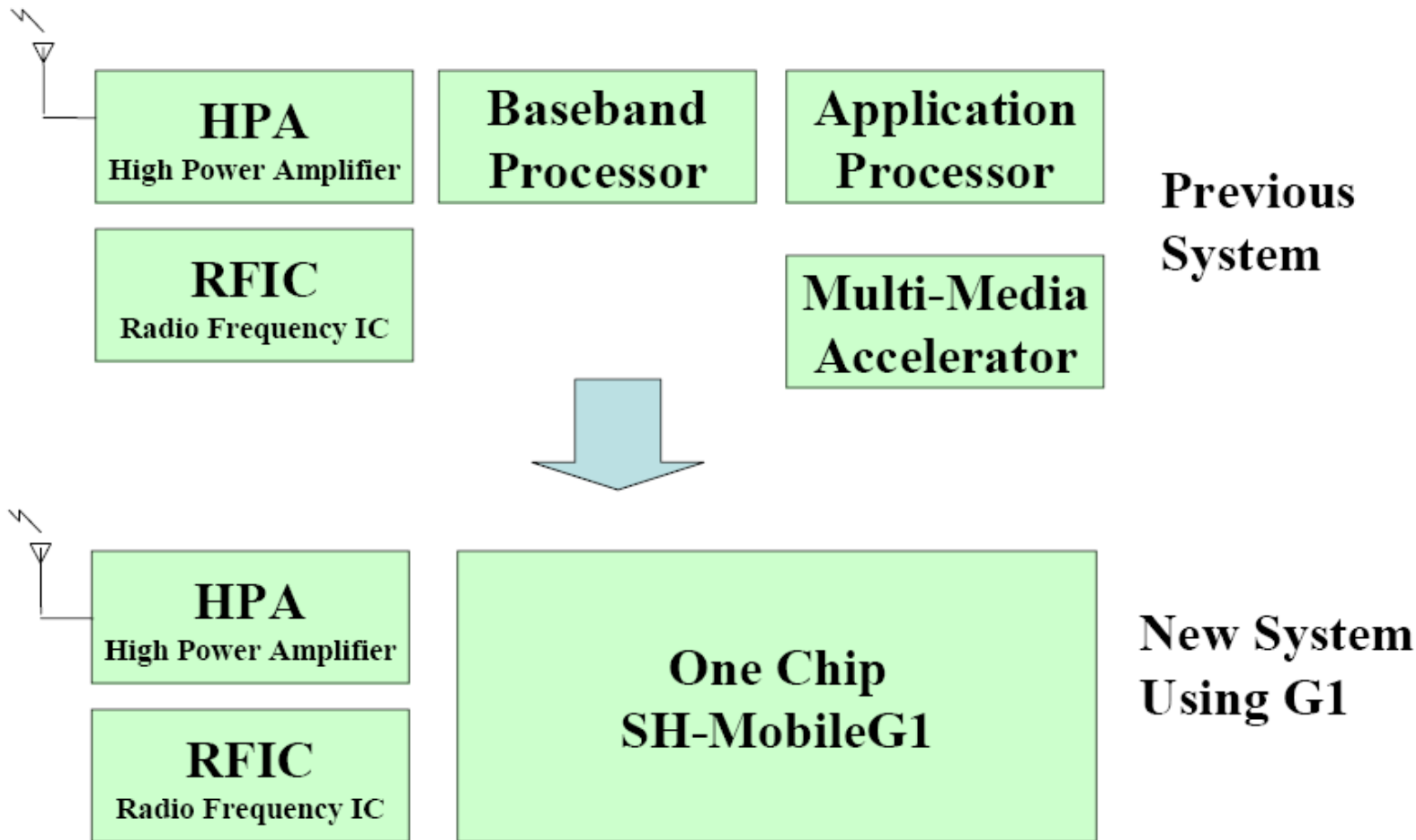
- 8-bit CPU optimised for control applications ←-----
- Extensive Boolean processing capabilities ←-----
- 64 k Program Memory address space
- 64 k Data Memory address space
- 4 k bytes of on chip Program Memory
- 128 bytes of on chip data RAM ←-----
- 32 bi-directional and individually addressable I/O lines ←-----
- Two 16-bit timers/counters ←-----
- Full duplex UART ←-----
- 6 sources/5-vector interrupt structure with 2 priority levels ←-----
- On chip clock oscillators ←-----
- Very popular CPU with many different variations ←-----

Features for Embedded Systems



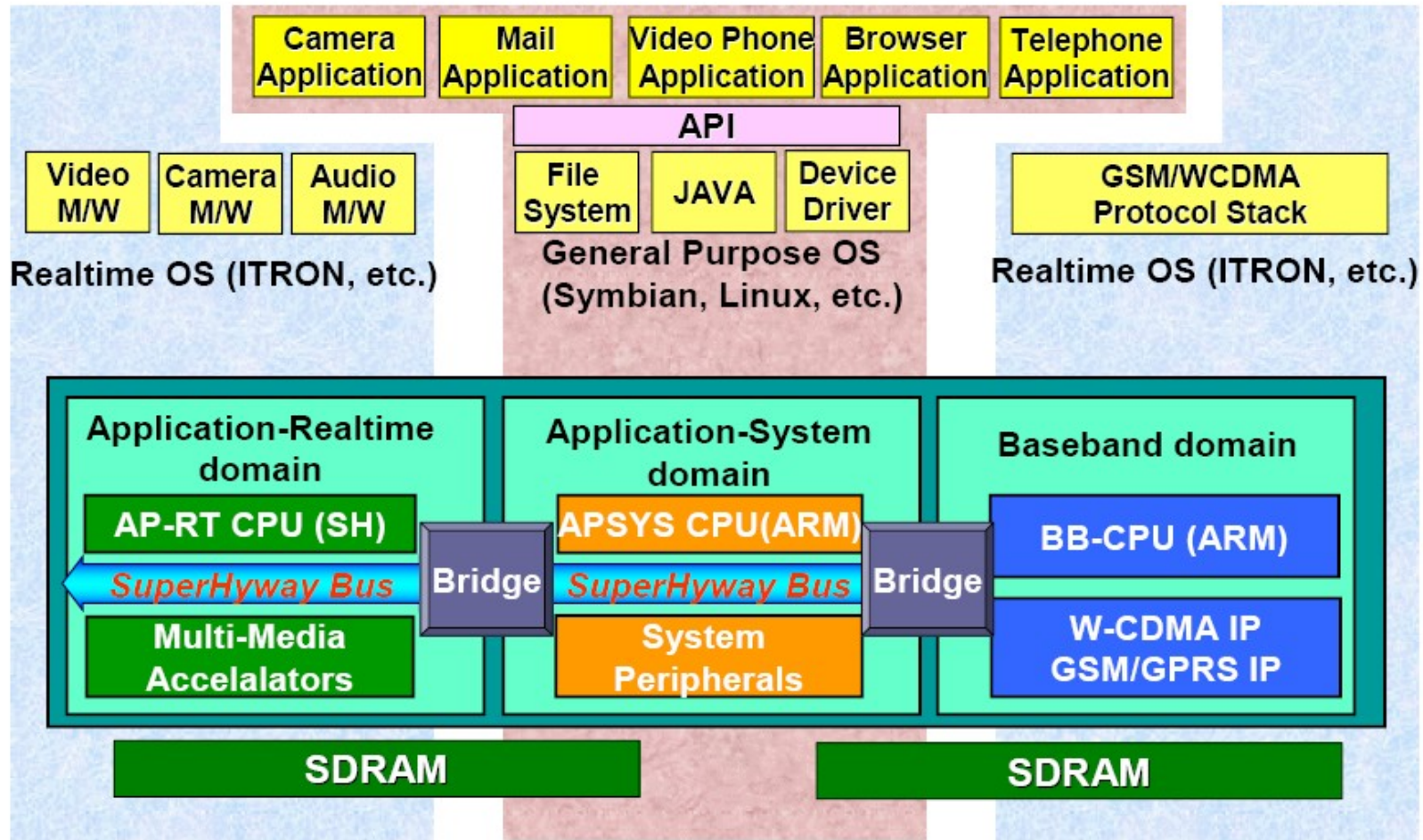
# Trend: multiprocessor systems-on-a-chip (MPSoCs)

## 3G Multi-Media Cellular Phone System



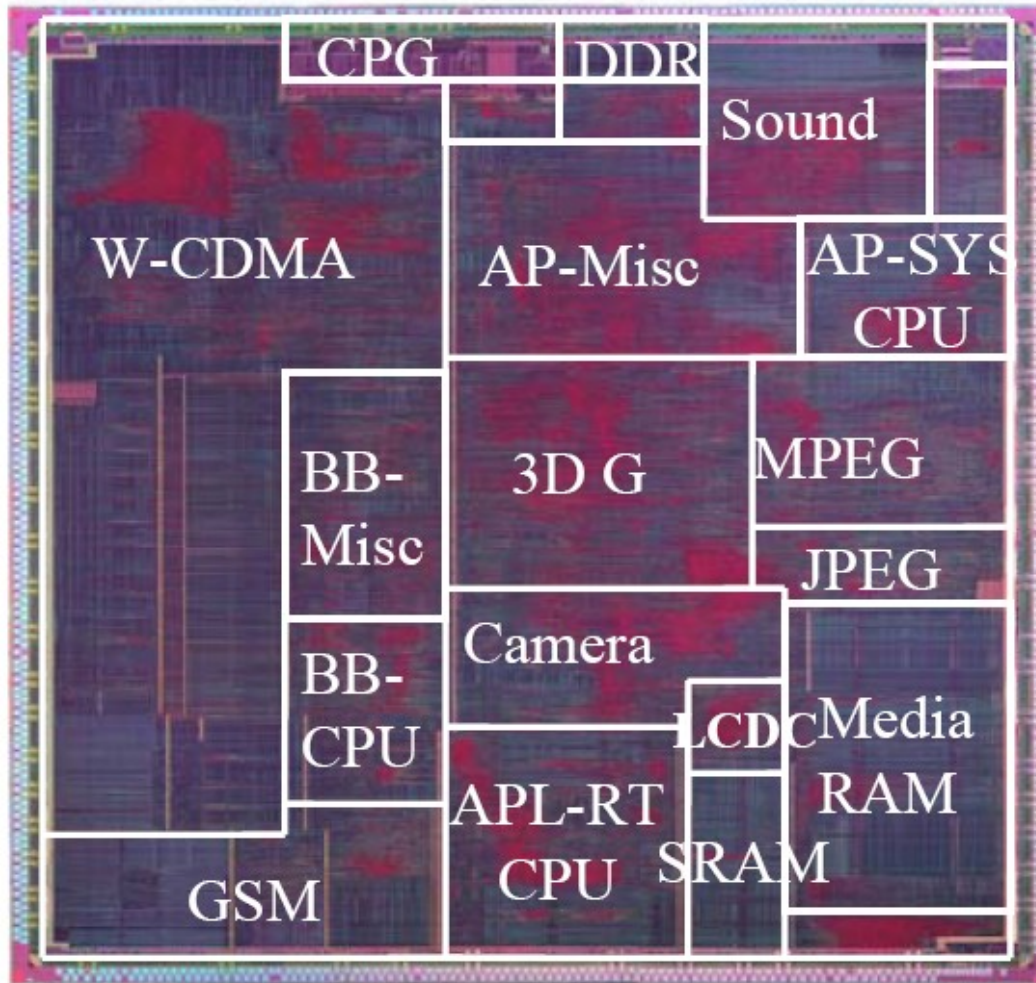
# Multiprocessor systems-on-a-chip (MPSoCs) (2)

## A Sample of System Architecture using G1



# Multiprocessor systems-on-a-chip (MPSoCs) (3)

## SH-MobileG1: Chip Overview

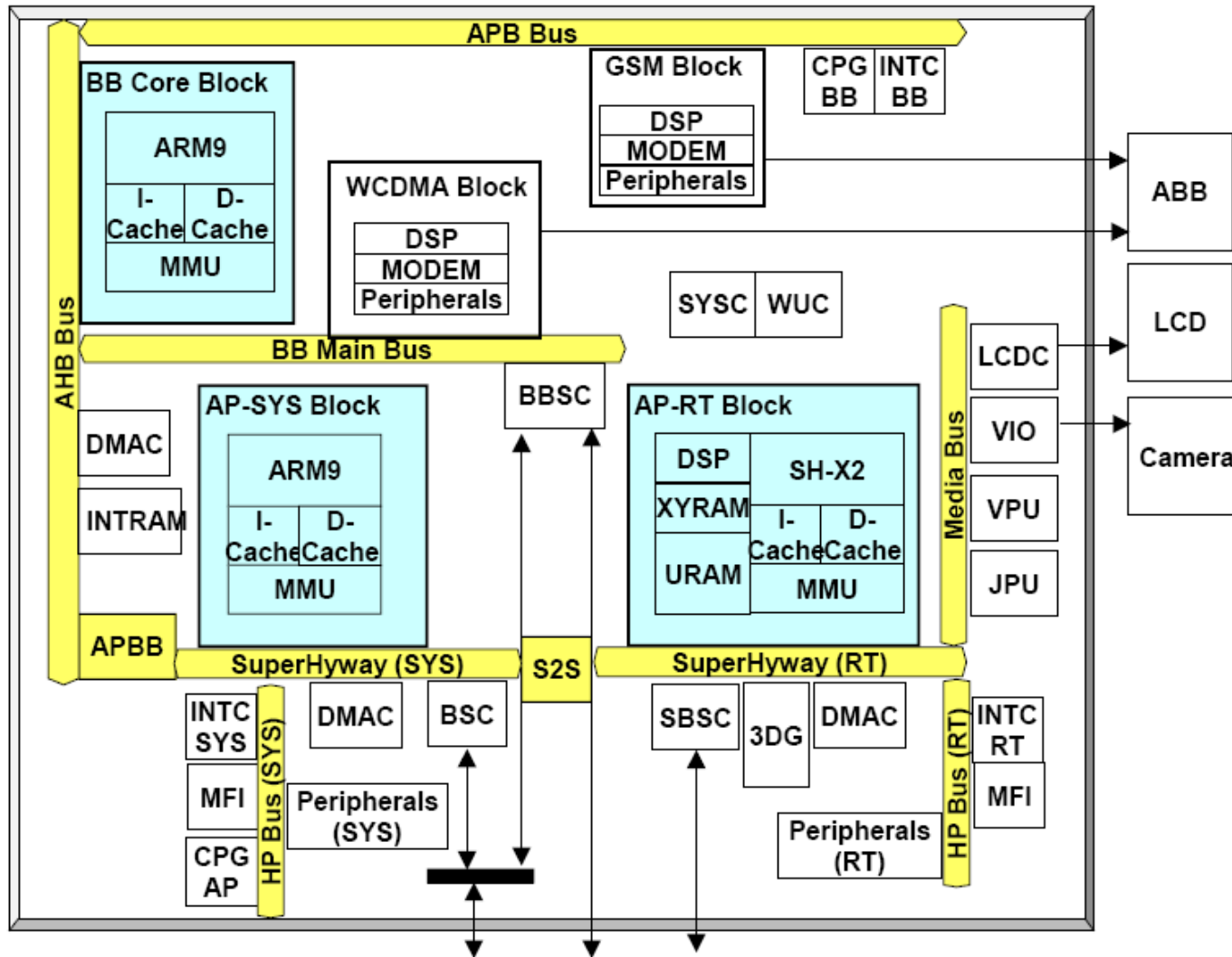


Die size	11.15mm x 11.15mm
Process	90nm LP 8M(7Cu+1Al) CMOS dual-Vth
Supply voltage	1.2V(internal), 1.8/2.5/3.3V(I/O)
# of TRs, gate, memory	181M TRs, 13.5M Gate 20.2 Mbit mem

<http://www.mpsoc-forum.org/2007/slides/Hattori.pdf>

# Multiprocessor systems-on-a-chip (MPSoCs) (4)

## G1 Module Diagram



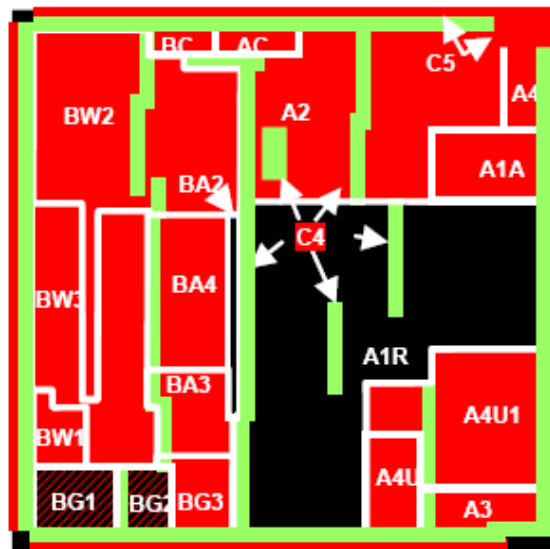
<http://www.mpsoc-forum.org/2007/slides/Hattori.pdf>



# Multiprocessor systems-on-a-chip (MPSoCs) (5)

## Leakage Current in Usage Scenes

### (2) Telephony (W-CDMA)



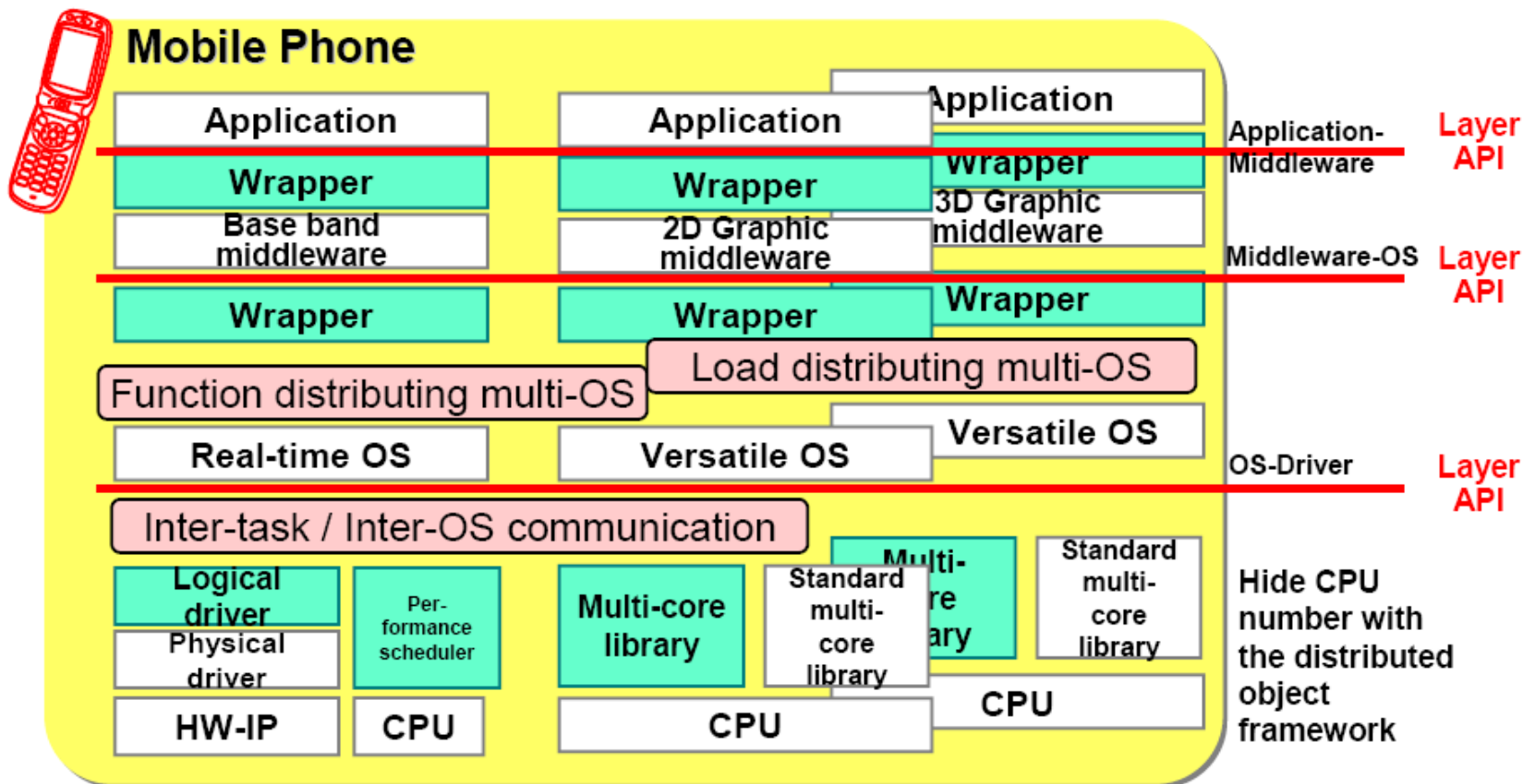
■ Power on  
■ Power off

Baseband part	Control	ON
	W-CDMA	ON
	GSM	ON / OFF
Application part	System-domain	ON
	Realtime-domain	OFF
Measured Leakage Current (@ Room Temp, 1.2V)		407 $\mu$ A

# Multiprocessor systems-on-a-chip (MPSoCs) (6)

## EXREAL Platform™ Software Interconnect

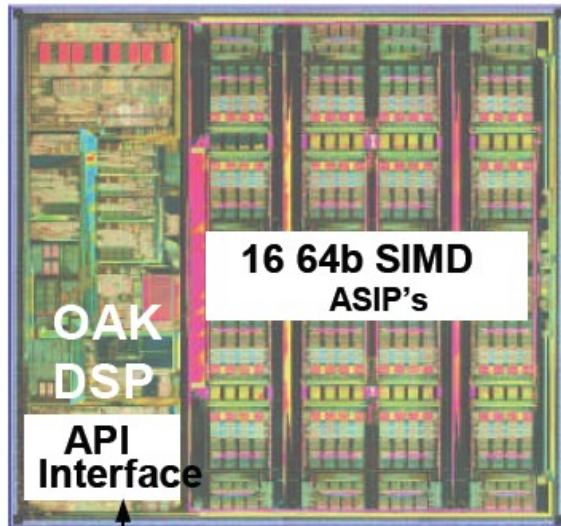
- Promote reuse of software assets through Wrapper and standardized layer API
- Control operating frequency and power on/off through the performance scheduler



<http://www.mpsoc-forum.org/2007/slides/Hattori.pdf>

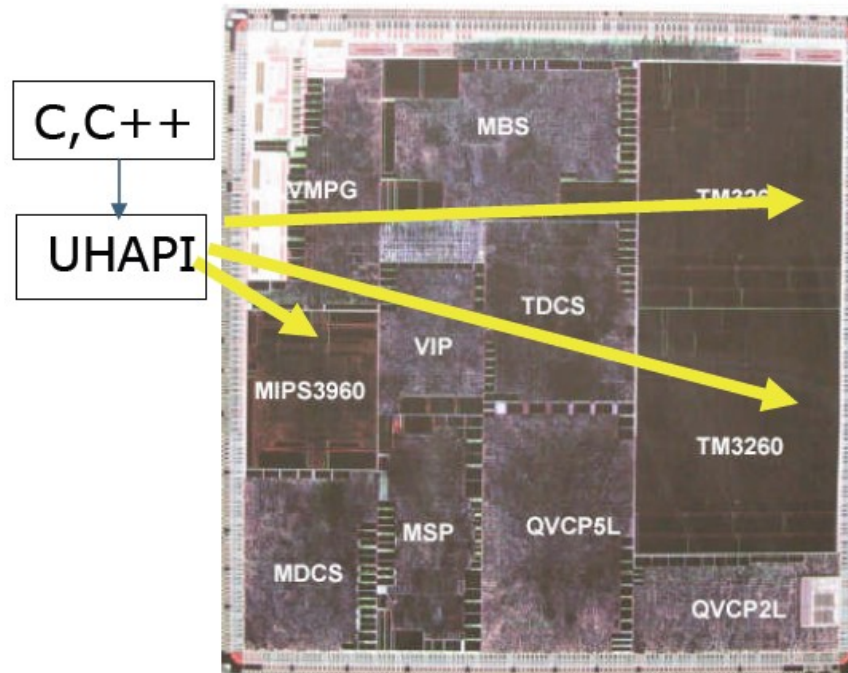
# Multiprocessor systems-on-a-chip (MPSoCs) (7)

## VIP for car mirrors Infineon



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

## Nexperia Digital Video Platform NXP



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

~50% inherent power efficiency of silicon

# Embedded System Hardware - Reconfigurable Hardware -

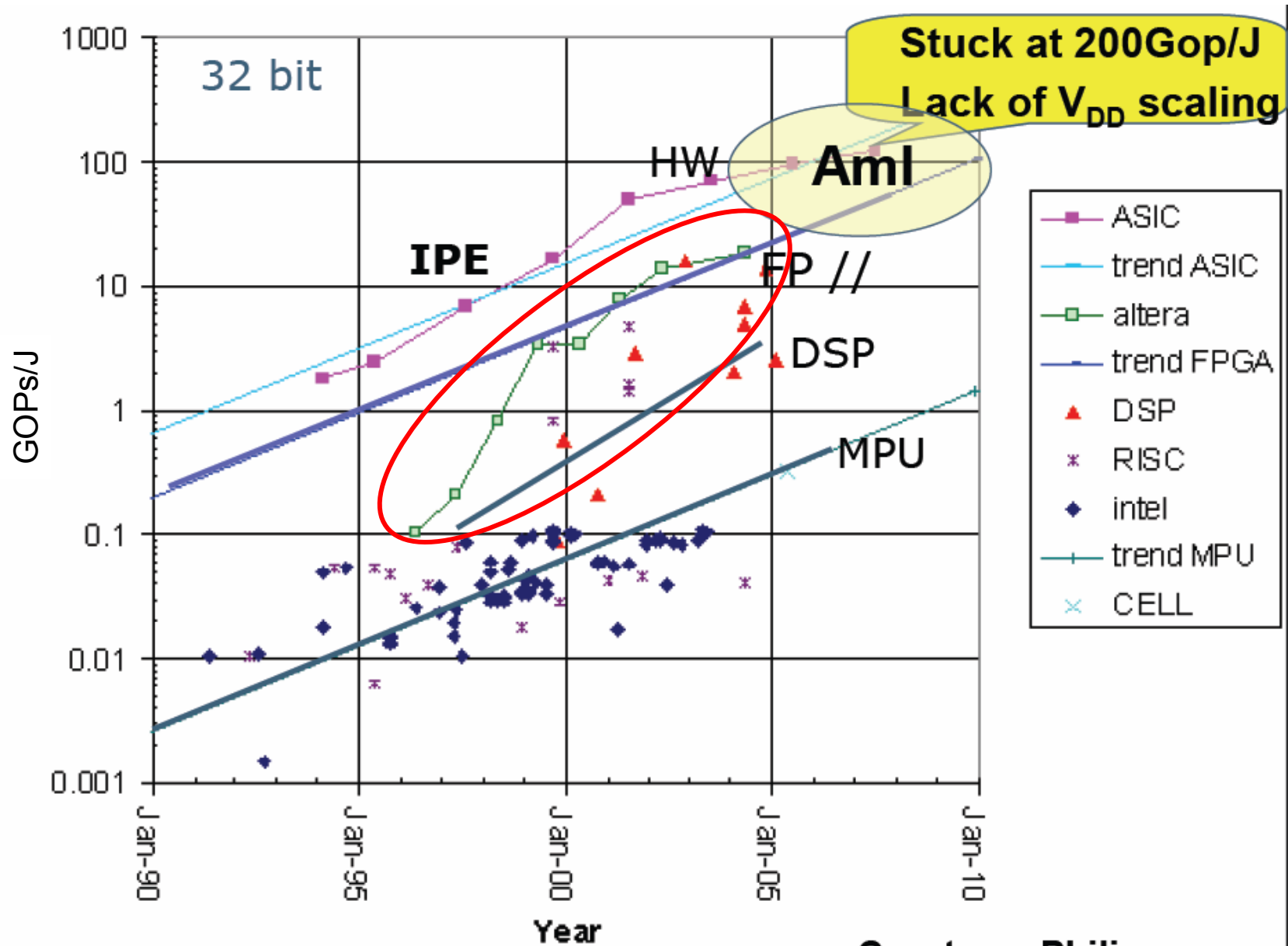
Peter Marwedel  
Informatik 12  
TU Dortmund  
Germany

**2009/11/10**





# Energy Efficiency of FPGAs



Courtesy: Philips

# Reconfigurable Logic

---

Full custom chips may be too expensive, software too slow.

Combine the speed of HW with the flexibility of SW

☞ HW with programmable functions and interconnect.

☞ Use of configurable hardware;

common form: field programmable gate arrays (FPGAs)

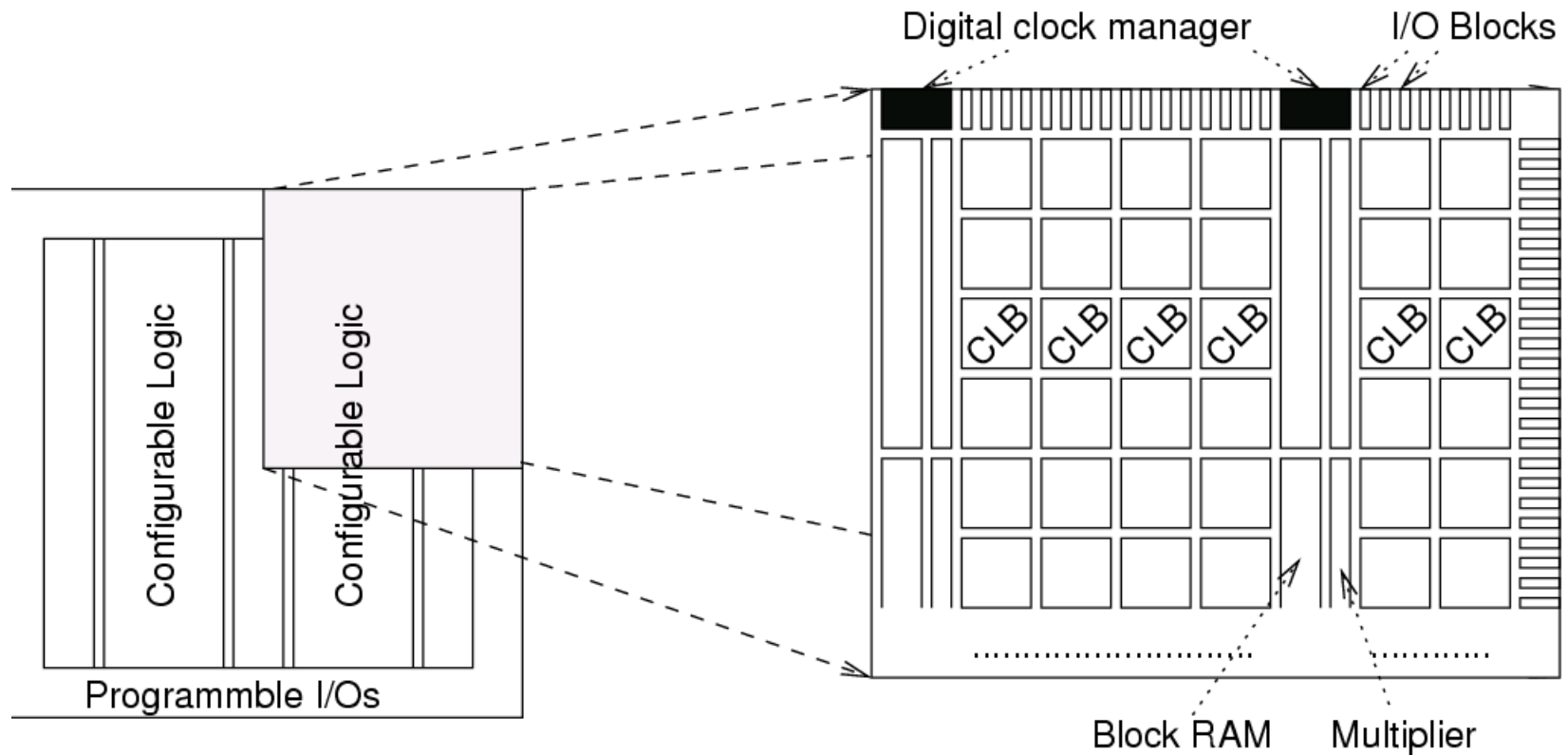
Applications: bit-oriented algorithms like

- encryption,
- fast “object recognition“ (medical and military)
- Adapting mobile phones to different standards.

Very popular devices from

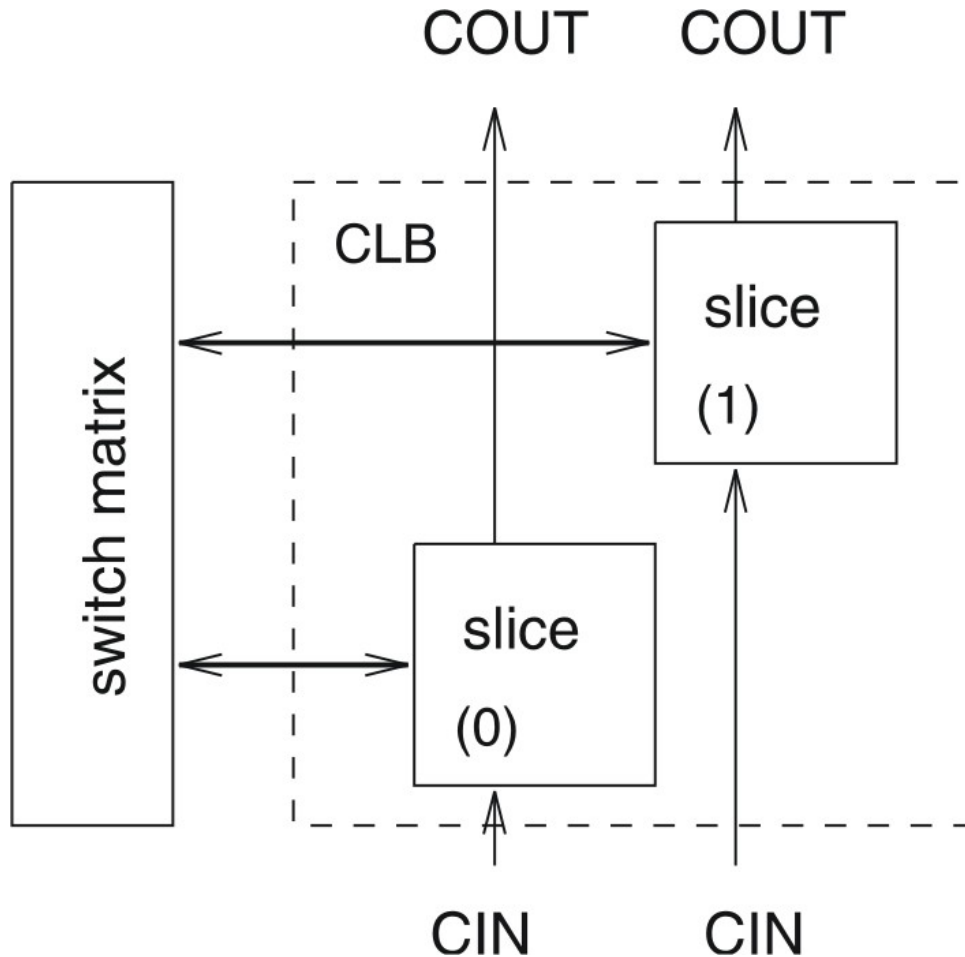
- XILINX (XILINX Vertex II are recent devices)
- Actel, Altera and others

# Floor-plan of VIRTEX II FPGAs

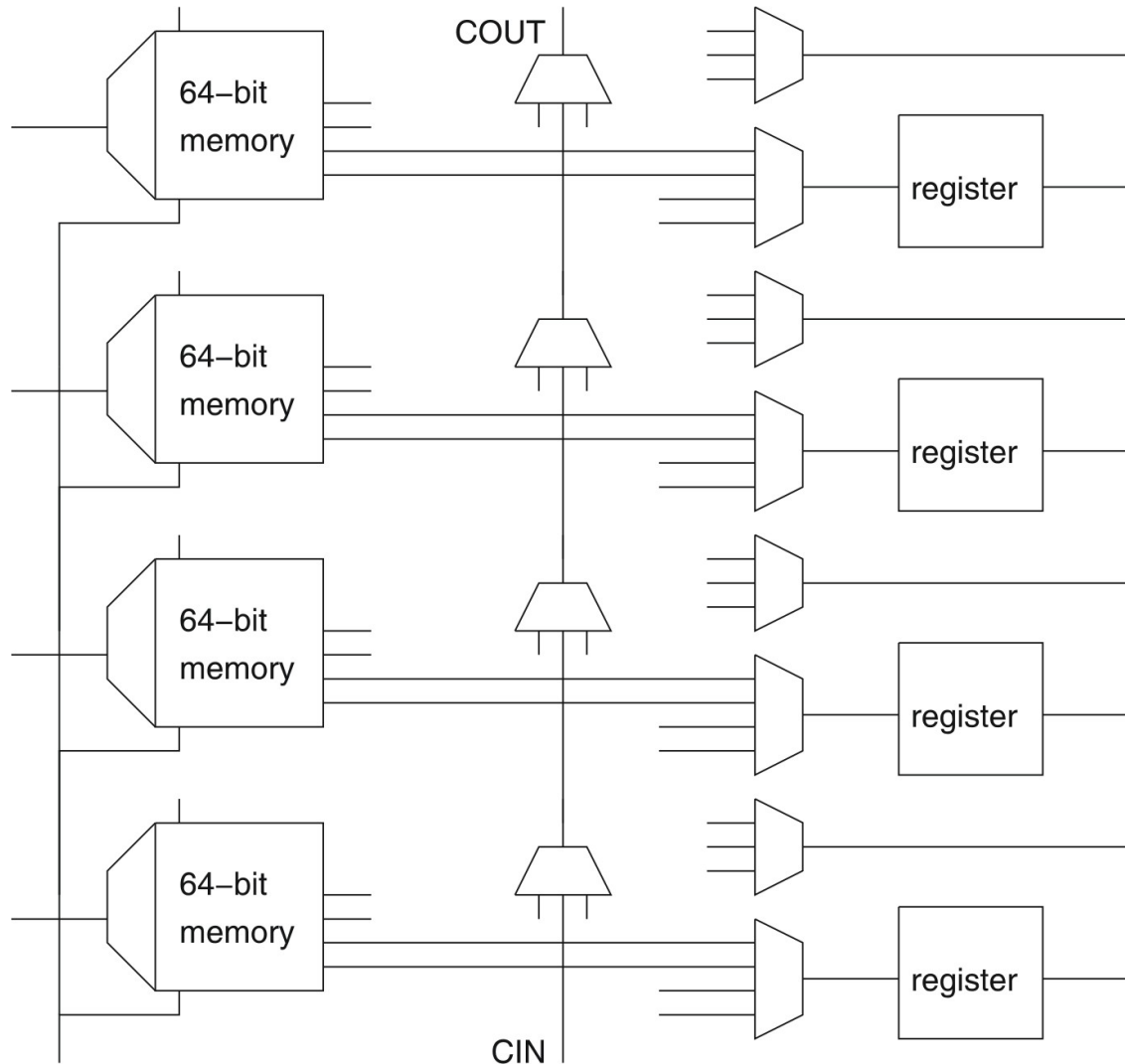


More recent: Virtex 5, but no floor-plan found for Virtex 5.

# Virtex 5 Configurable Logic Block (CLB)



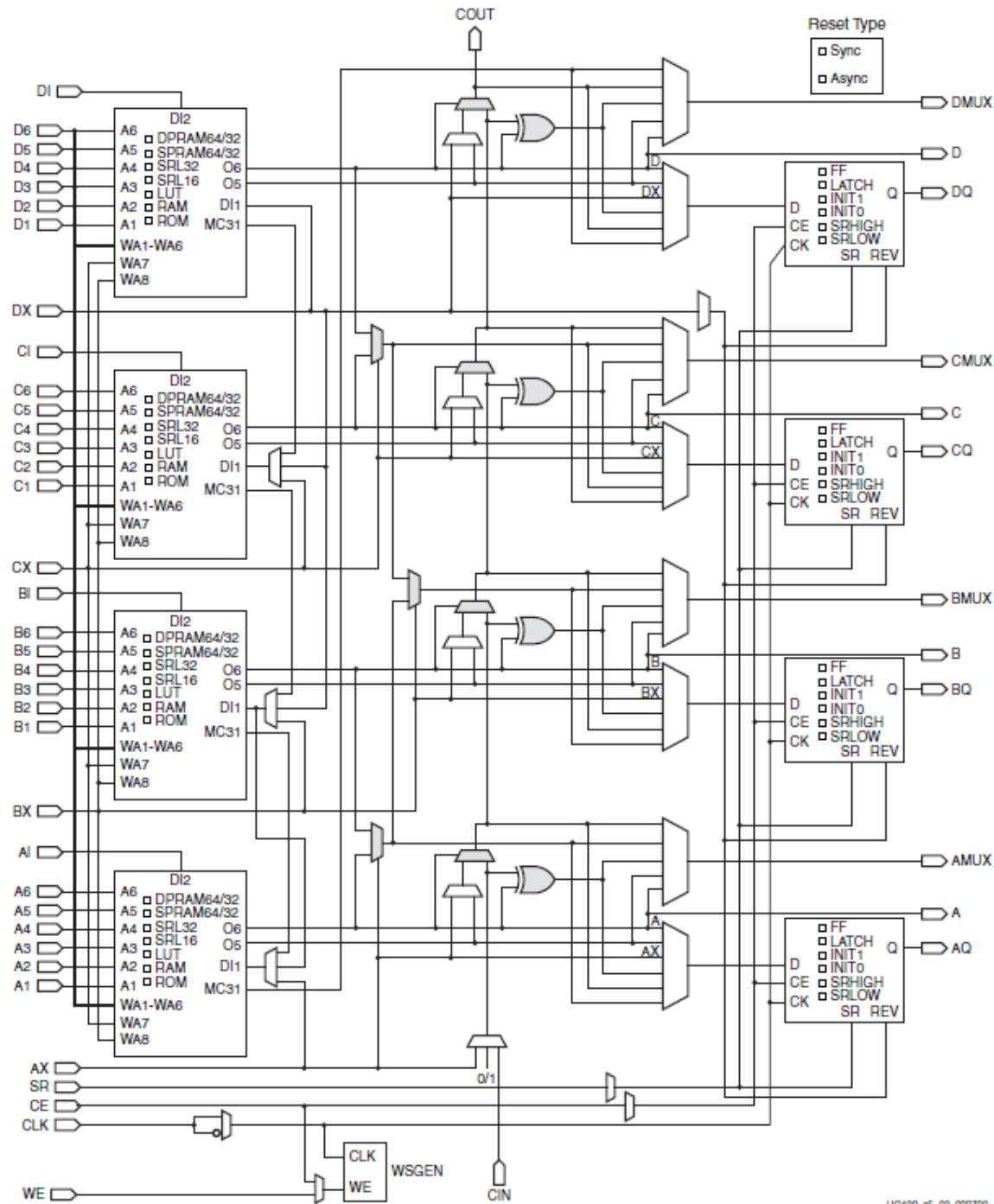
# Virtex 5 Slice (simplified)



Memories typically used as look-up tables to implement any Boolean function of  $\leq 6$  variables.

# Virtex 5 SliceM

SliceM supports using memories for storing data and as shift registers





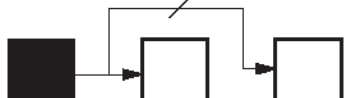
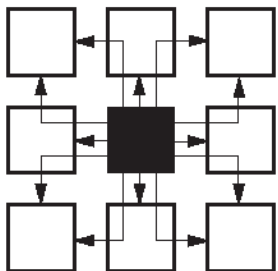
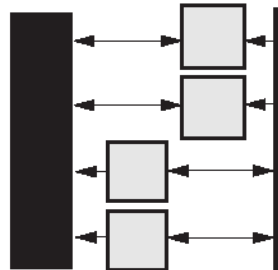
# Resources available in Virtex 5 devices

Device	CLB Array Row x Column	Number of 6-Input LUTs	Maximum Distributed RAM (Kb)	Shift Register (Kb)	Number of Flip-Flops
XC5VLX20T	60 x 26	12,480	210	105	12,480
XC5VLX30	80 x 30	19,200	320	160	19,200
XC5VFX30T	80 x 38	20,480	380	190	20,480
XC5VLX30T	80 x 30	19,200	320	160	19,200
XC5VSX35T	80 x 34	21,760	520	260	21,760
XC5VLX50	120 x 30	28,800	480	240	28,800
XC5VLX50T	120 x 30	28,800	480	240	28,800
XC5VSX50T	120 x 34	32,640	780	390	32,640
XC5VFX70T	160 x 38	44,800	820	410	44,800
XC5VLX85	120 x 54	51,840	840	420	51,840
XC5VLX85T	120 x 54	51,840	840	420	51,840
XC5VSX95T	160 x 46	58,880	1,520	760	58,880
XC5VFX100T	160 x 56	64,000	1,240	620	64,000
XC5VLX110	160 x 54	69,120	1,120	560	69,120
XC5VLX110T	160 x 54	69,120	1,120	560	69,120
XC5VFX130T	200 x 56	81,920	1,580	790	81,920
XC5VTX150T	200 x 58	92,800	1,500	750	92,800
XC5VLX155	160 x 76	97,280	1,640	820	97,280
XC5VLX155T	160 x 76	97,280	1,640	820	97,280
XC5VFX200T	240 x 68	122,880	2,280	1140	122,880
XC5VLX220	160 x 108	138,240	2,280	1140	138,240
XC5VLX220T	160 x 108	138,240	2,280	1140	138,240
XC5VSX240T	240 x 78	149,760	4,200	2100	149,760
XC5VTX240T	240 x 78	149,760	2,400	1200	149,760
XC5VLX330	240 x 108	207,360	3,420	1710	207,360
XC5VLX330T	240 x 108	207,360	3,420	1710	207,360

[© and source: Xilinx Inc.:  
Virtex 5 FPGA User  
Guide, May, 2009  
//www.xilinx.com]

# Interconnect for Virtex II

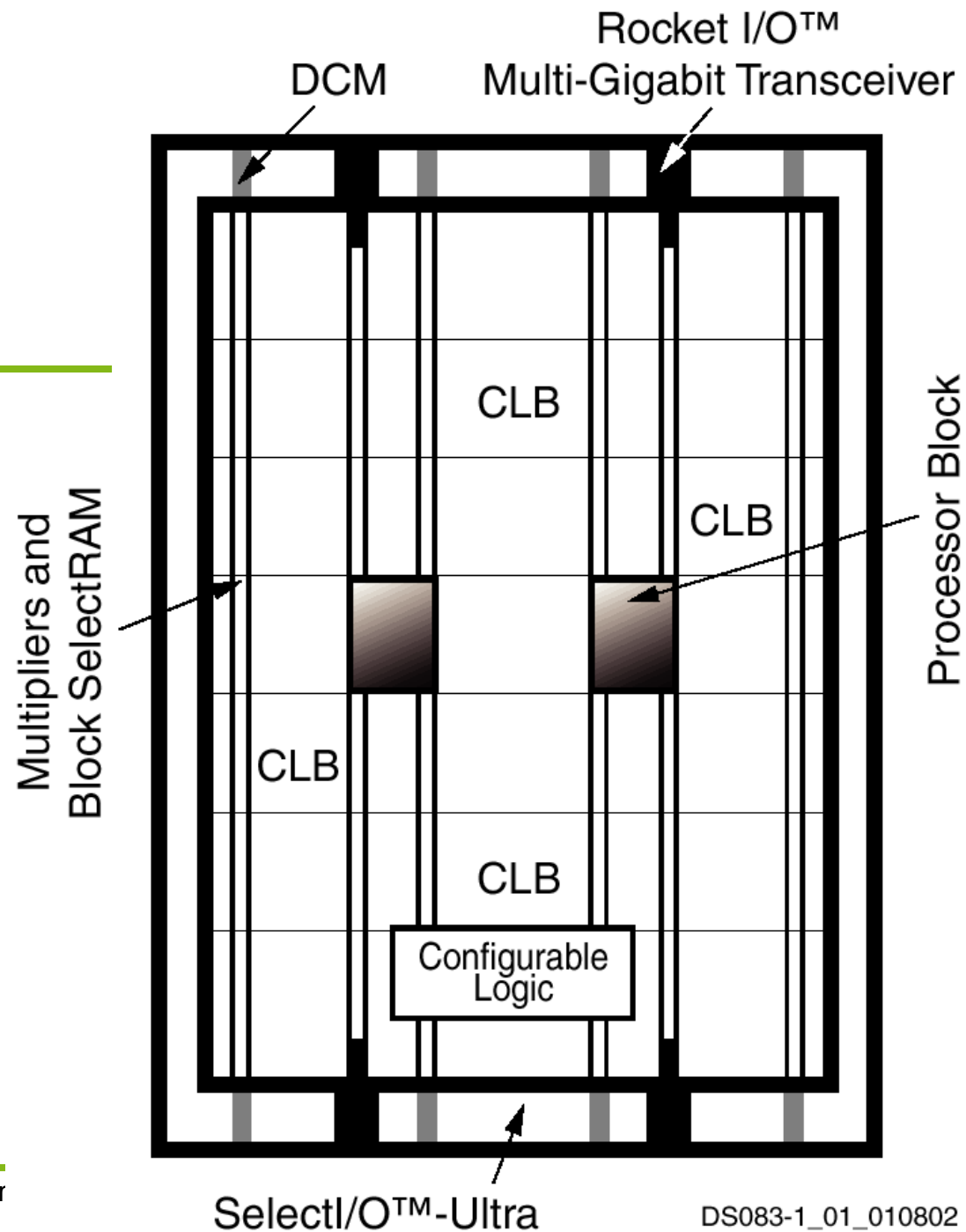
Hierarchical Routing Resources;  
no routing plan found for Virtex 5.

<p>24 Horizontal Long Lines 24 Vertical Long Lines</p>	
<p>120 Horizontal Hex Lines 120 Vertical Hex Lines</p>	
<p>40 Horizontal Double Lines 40 Vertical Double Lines</p>	
<p>16 Direct Connections (total in all four directions)</p>	
<p>8 Fast Connects</p>	



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

Virtex 5 Devices include up to 2 PowerPC processor cores



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

# Memory

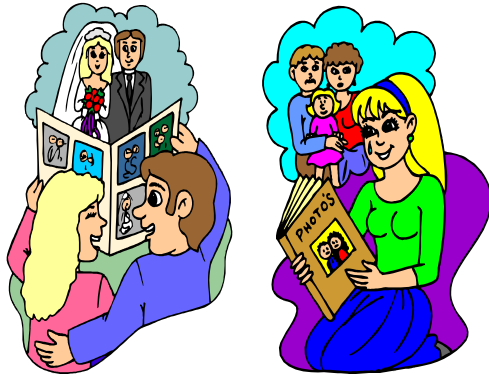
Peter Marwedel  
Informatik 12  
TU Dortmund  
Germany

2008/11/15

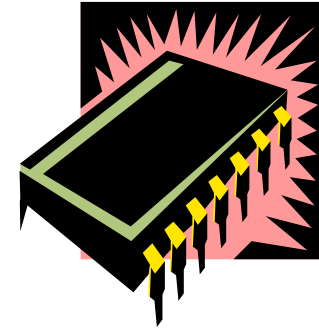


# Memory

Memories?



Oops!  
Memories!

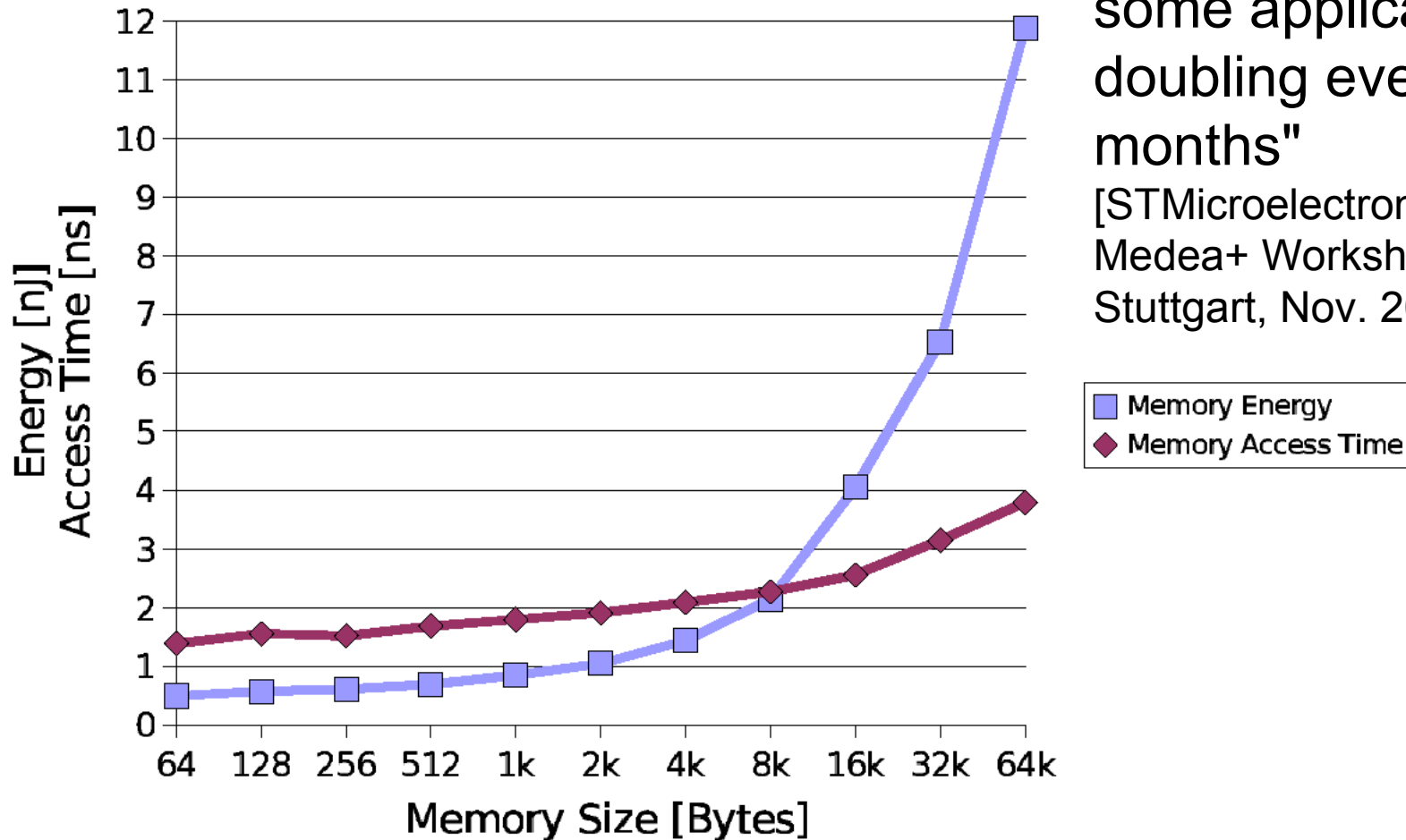


For the memory, efficiency is again a concern:

- speed (latency and throughput); predictable timing
- energy efficiency
- size
- cost
- other attributes (volatile vs. persistent, etc)

# Access times and energy consumption increases with the size of the memory

Example (CACTI Model):

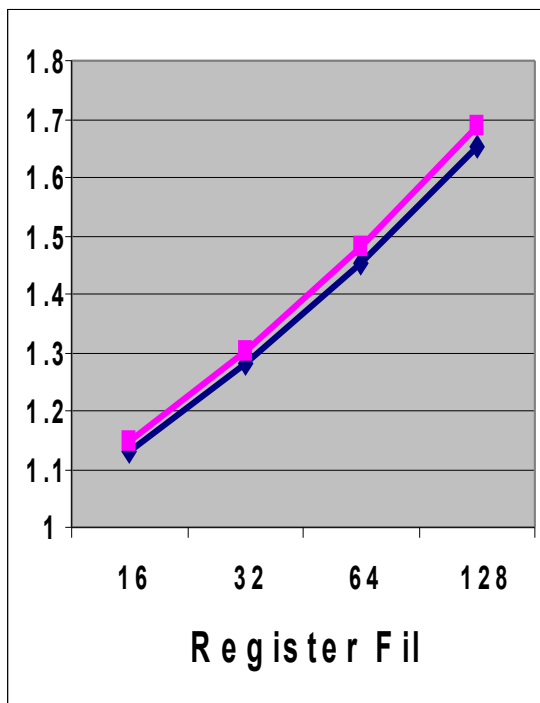


"Currently, the size of some applications is doubling every 10 months"

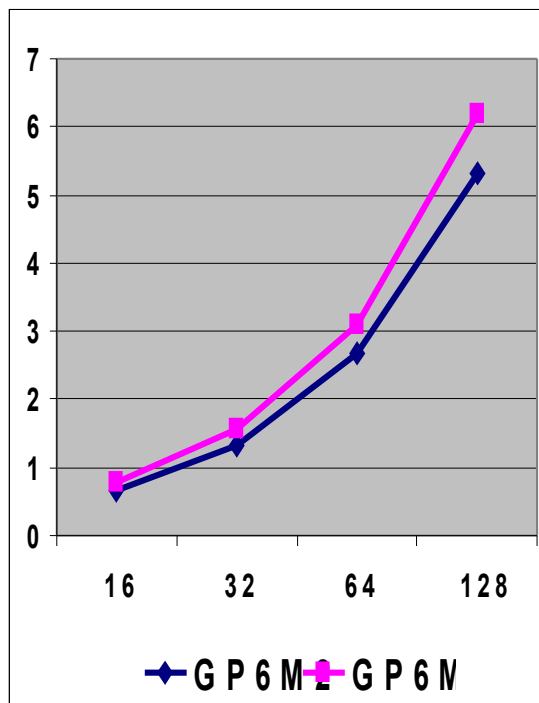
[STMicroelectronics, Medea+ Workshop, Stuttgart, Nov. 2003]

# Access times and energy consumption for multi-ported register files

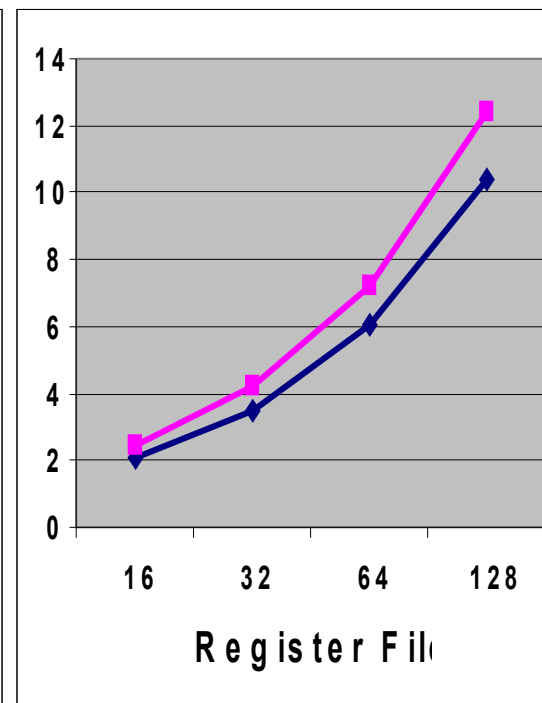
Cycle Time (ns)



Area ( $\lambda^2 \times 10^6$ )



Power (W)

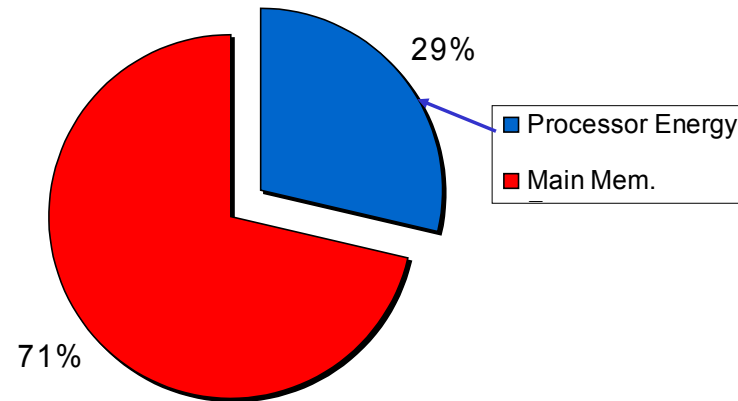


Rixner's et al. model [HPCA'00], Technology of 0.18  $\mu\text{m}$

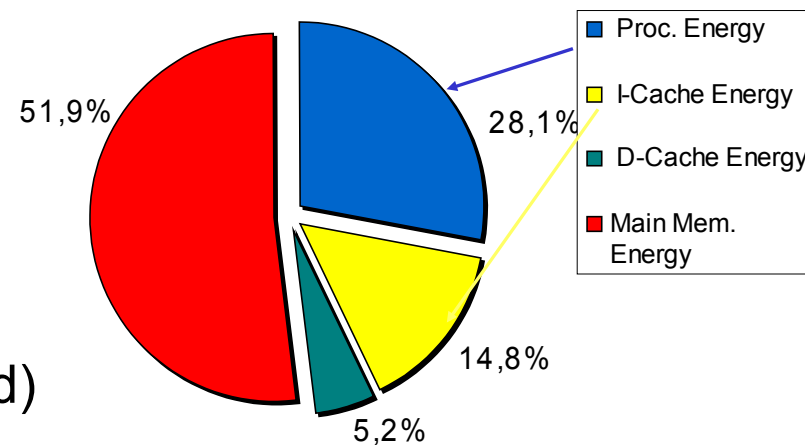
Source and © H. Valero, 2001

# Memory system frequently consumes >50 % of the energy used for processing

Cache (\$)-less monoprocessor



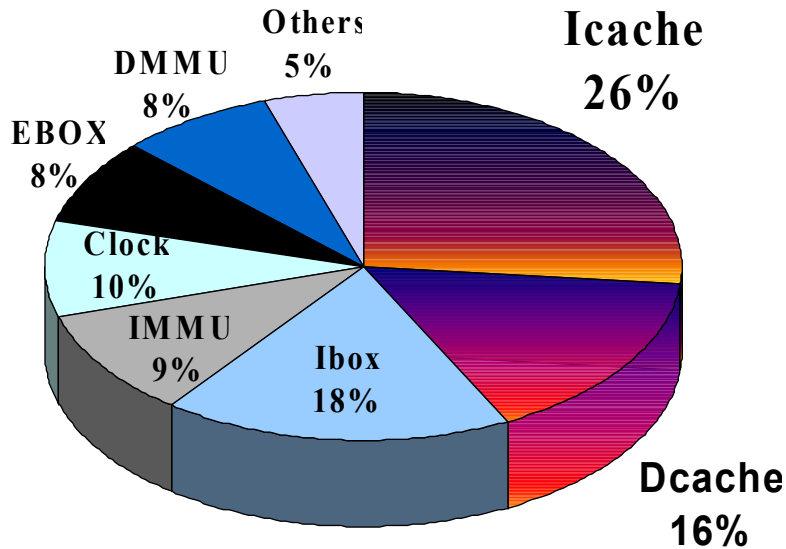
Multiprocessor with cache (\$)



Average over 200 benchmarks analyzed by Verma (U. Dortmund)

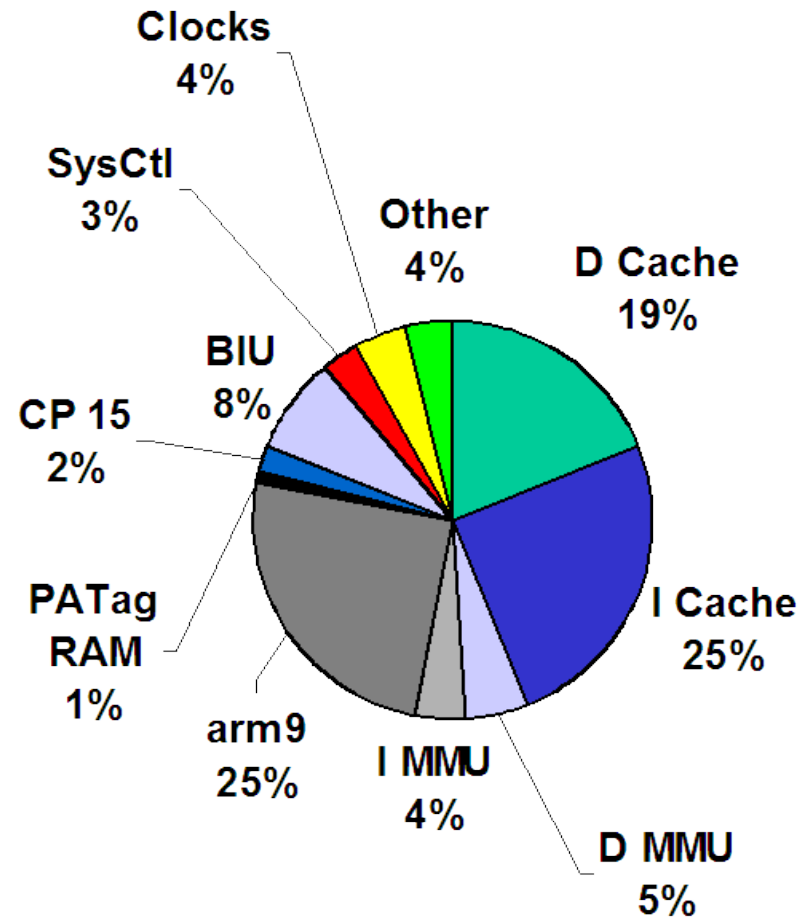
[M. Verma, P. Marwedel: Advanced Memory Optimization Techniques for Low-Power Embedded Processors, Springer, 2007]

# Similar information according to other sources



Strong ARM

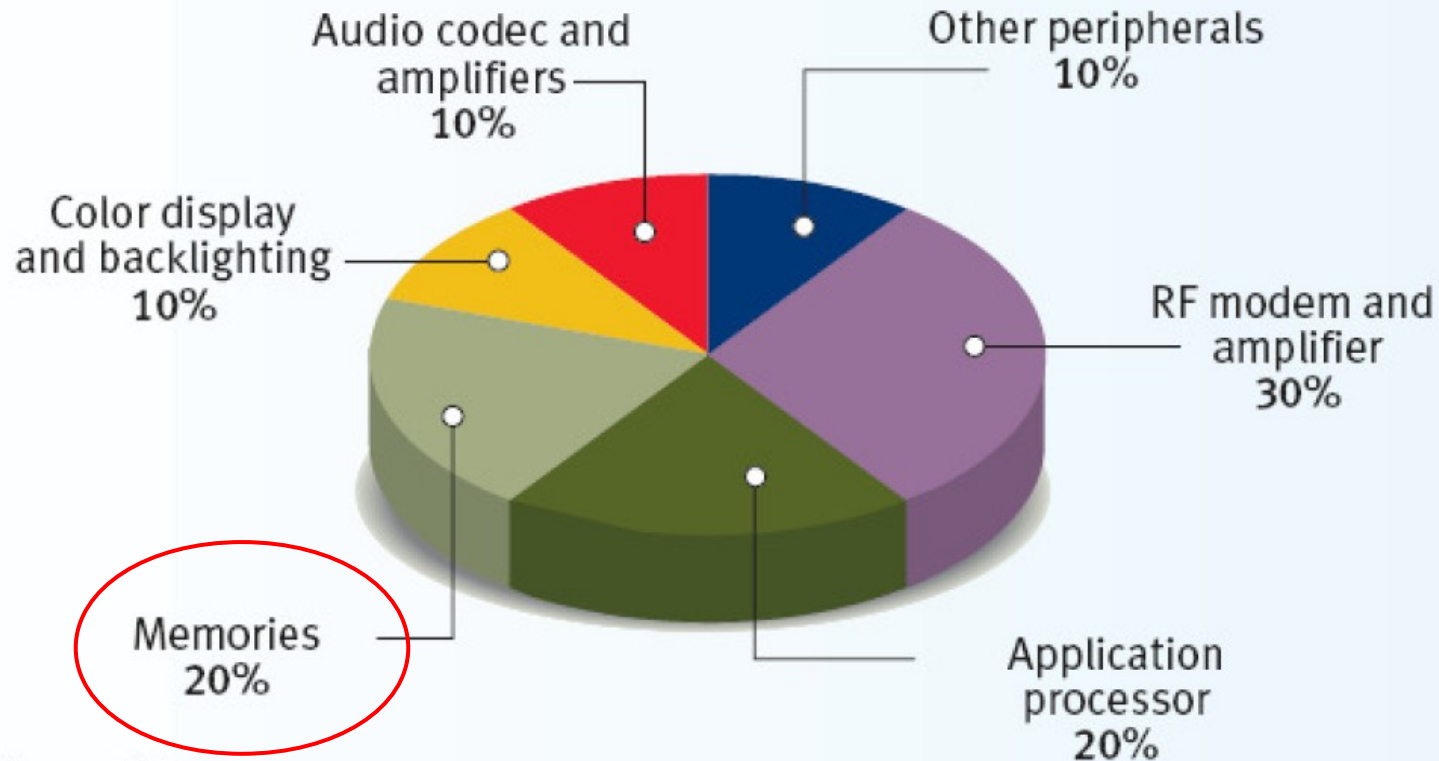
IEEE Journal of SSC  
Nov. 96



[Based on slide by and ©: Osman S. Unsal, Israel Koren, C. Mani Krishna, Csaba Andras Moritz, U. of Massachusetts, Amherst, 2001]

[Segars 01 according to Vahid@ISSS01]

# Energy consumption in mobile devices



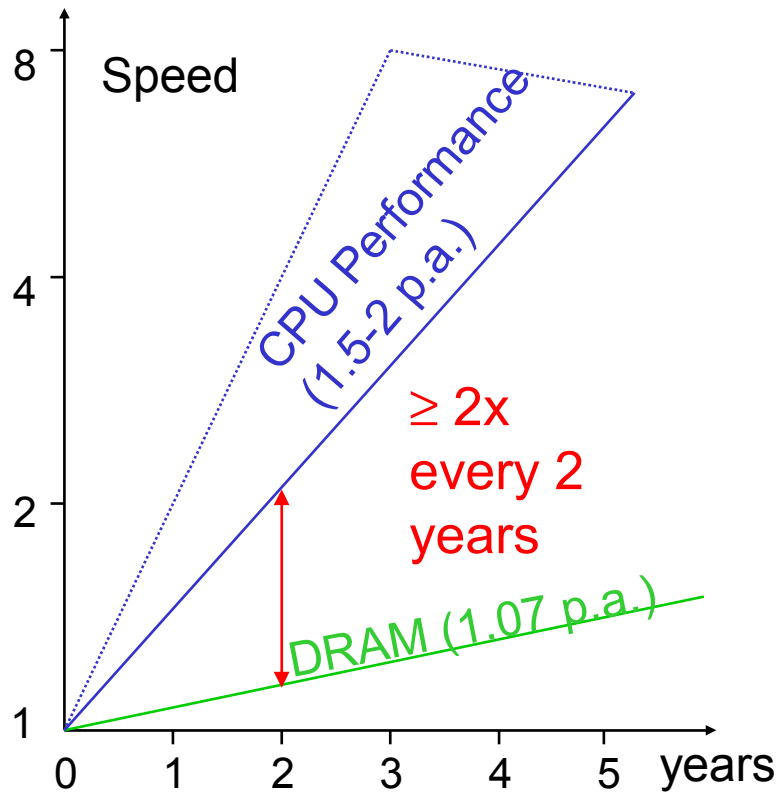
Source: Siemens

[O. Vargas (Infineon Technologies): Minimum power consumption in mobile-phone memory subsystems; Pennwell Portable Design - September 2005;] Thanks to Thorsten Koch (Nokia/ Univ. Dortmund) for providing this source.



# Trends for the Speeds

Speed gap between processor and main DRAM increases



Similar problems also for embedded systems & MPSoCs

☞ In the future:

Memory access times  $\gg$  processor cycle times

☞ “Memory wall” problem

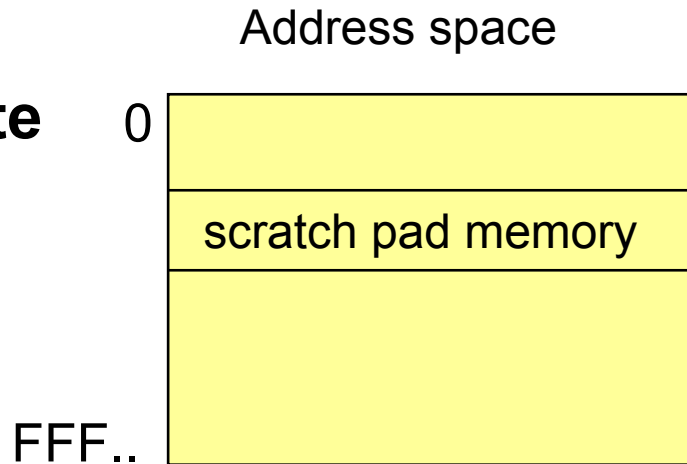
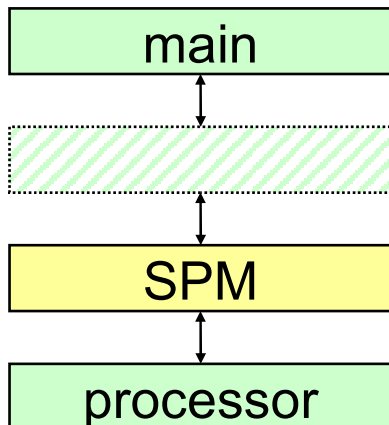


[P. Machanik: Approaches to Addressing the Memory Wall, TR Nov. 2002, U. Brisbane]

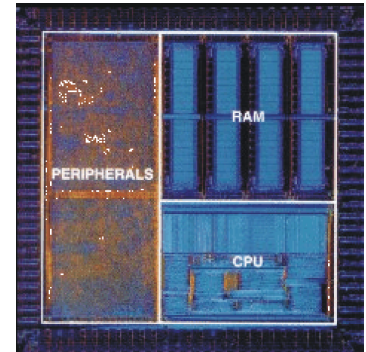
# Hierarchical memories using scratch pad memories (SPM)

**SPM is a small, physically separate memory mapped into the address space**

## Hierarchy

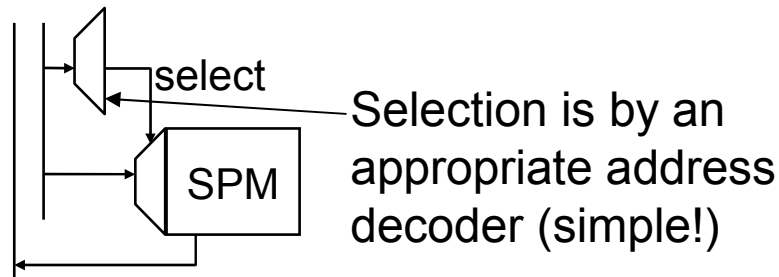


## Example



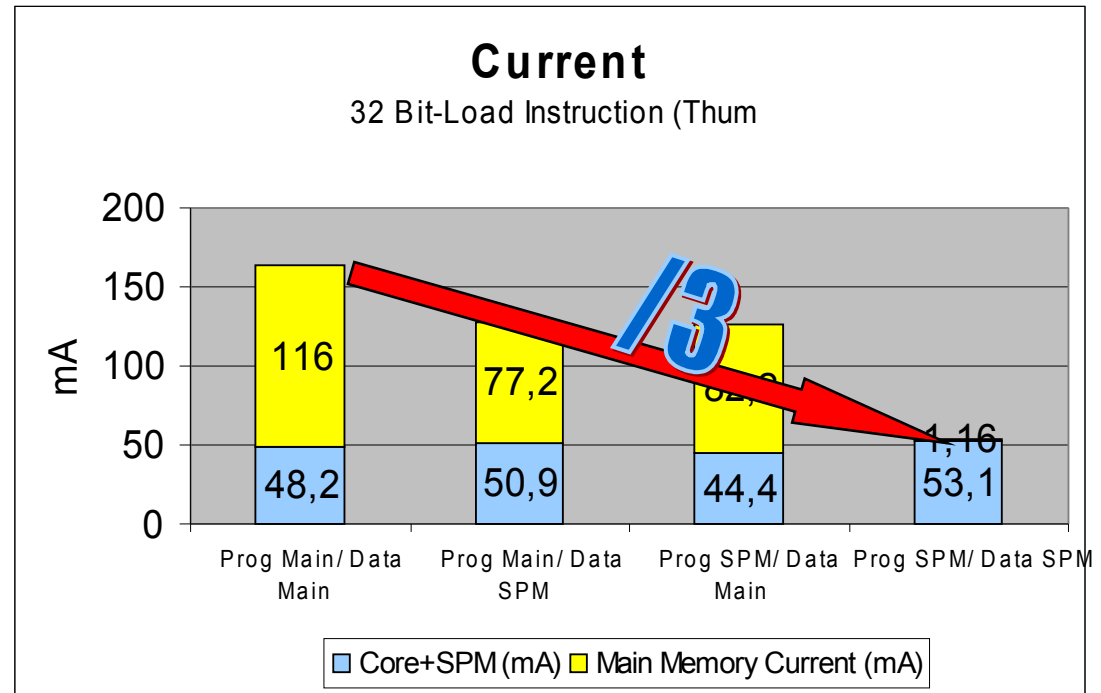
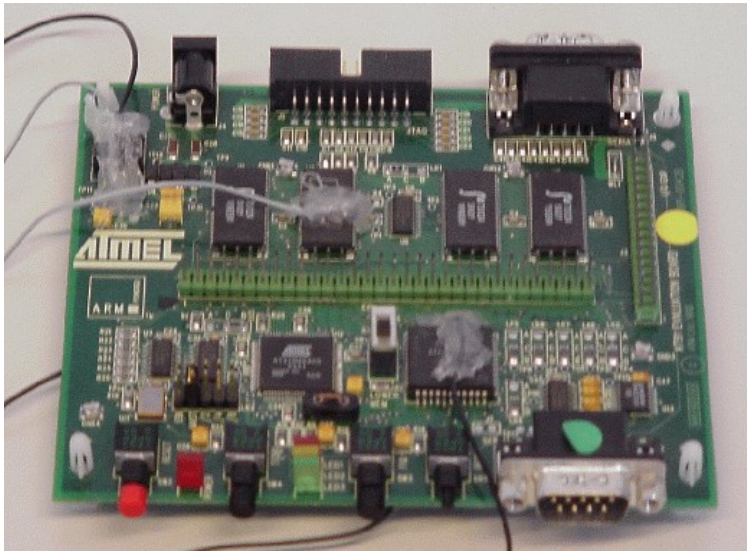
ARM7TDMI cores, well-known for low power consumption

no tag memory



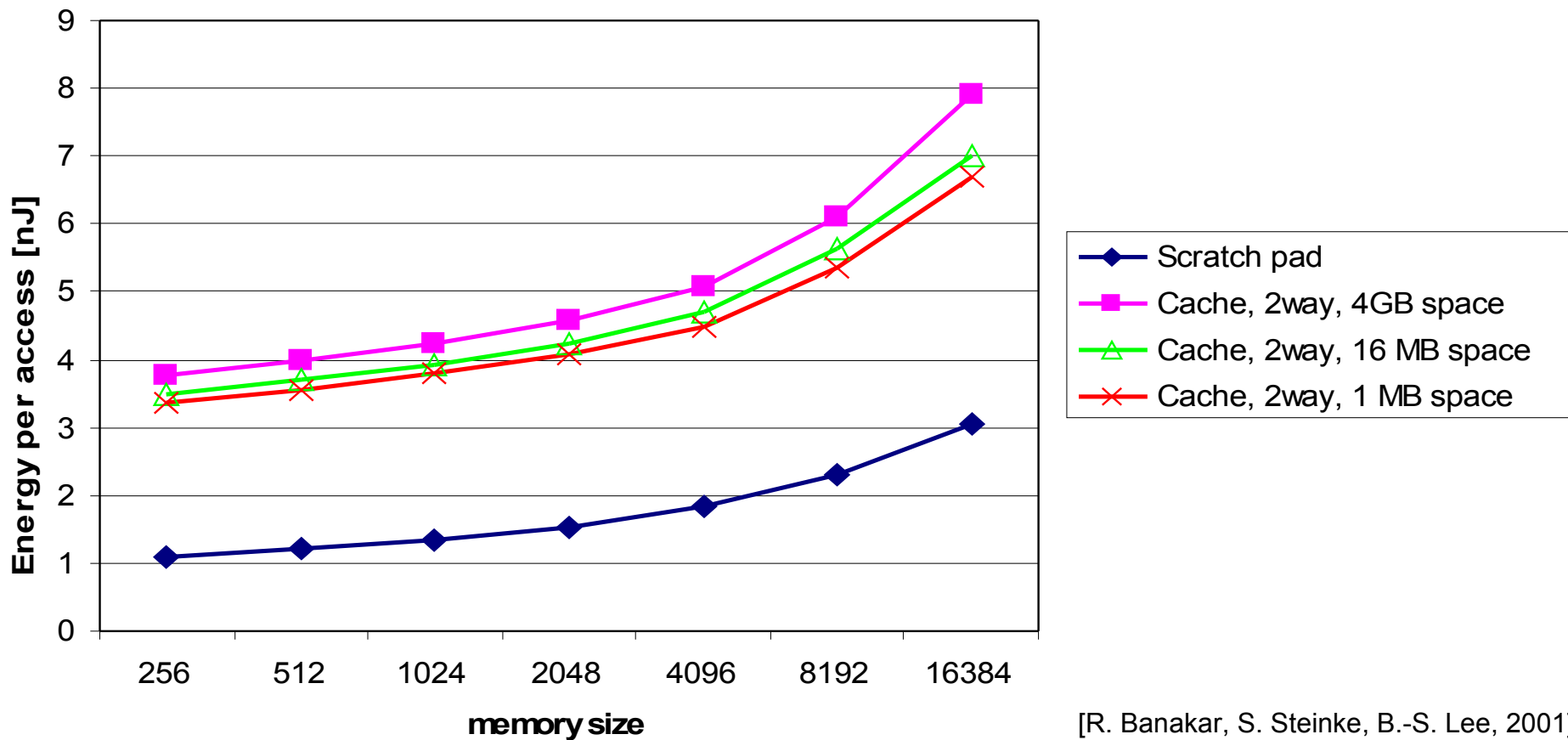
# Comparison of currents using measurements

E.g.: ATMEL board with ARM7TDMI and ext. SRAM



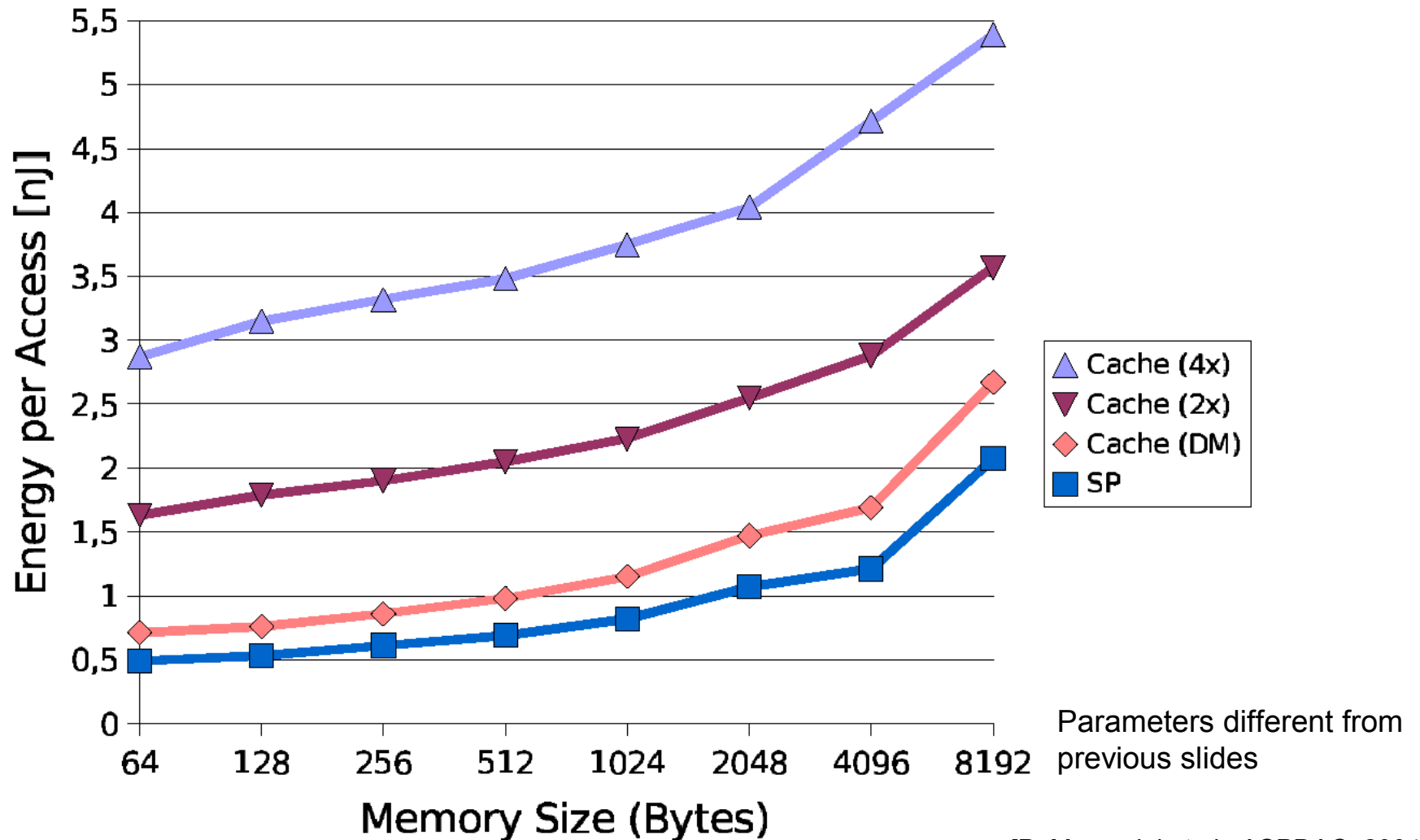
# Why not just use a cache ?

## 2. Energy for parallel access of sets, in comparators, muxes.



[R. Banakar, S. Steinke, B.-S. Lee, 2001]

# Influence of the associativity



[P. Marwedel et al., ASPDAC, 2004]

# Summary

---

- Processing
  - VLIW/EPIC processors
  - MPSoCs
- FPGAs
- Memories
  - “Small is beautiful”  
(in terms of energy consumption, access times, size)