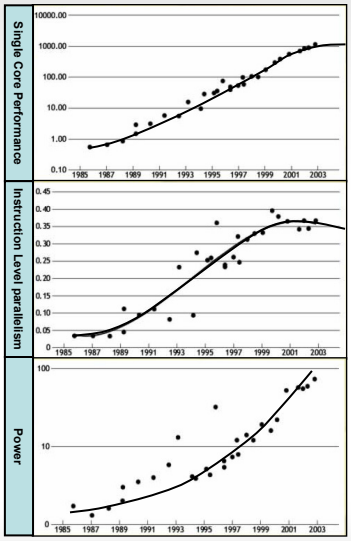
MULTICORE DESIGN SIMPLIFIED
imperas

The Impact of MPSoC Requirements on Software Programming

Frank Schirrmeister
franks@imperas.com

The End of the Single Core Era

MULTICORE DESIGN SIMPLIFIED
imperas



Processor Performance

- Is no longer growing
- Parallelism was addressed in hardware using instruction level parallelism

Instruction Level Parallelism

- Runs out of steam
- Issuing more than four instructions in parallel has little effect on most applications

Power Ceiling

- Has essentially stopped progress in conventional processor core development

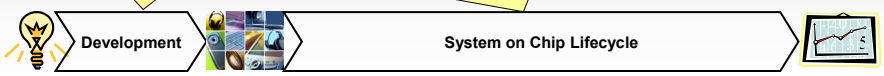
Source: Intel, Imperas

The Business Breakdown

MULTICORE DESIGN SIMPLIFIED imperas

Time to Market still #1 issue

SoC development cost requires to lengthen the lifecycle per product, optimizing **Time in Market**



Proposed approaches:

- Faster processors?
  - Limited by power!
- Dedicated hardware accelerators?
  - Do not address flexibility to optimize Time in Market!

"Von Neumann is a poor use of scaling – all the **energy** is going on the **communication** between the processor and the memory. Its much **better** to use **20 microprocessors running at 100MHz than one at 2GHz**"

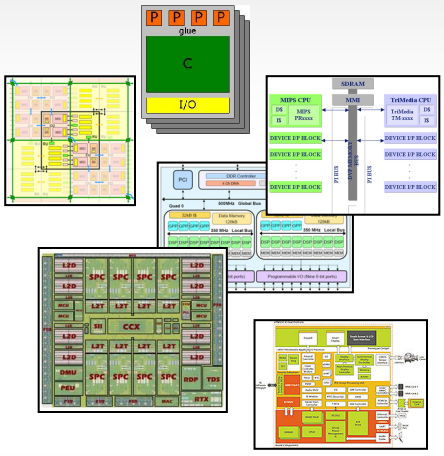
Hugo de Man
IMEC

➔ **Multiprocessor Systems on Chips (MPSoCs) are the *only* feasible alternative …**

MPSoCs are the Solution

But, the challenges move into the software!

MULTICORE DESIGN SIMPLIFIED imperas



- Multiple CPUs
  - Custom
  - 3rd Party IP
- Heterogeneous
  - Homogeneous for very specific applications
- Complex communication
- ***Main Challenge: Software***
  - Complexity explodes
  - Sequential software on individual cores fails
  - Multiple processors present various challenges
  - Partitioning
    - Parallelization
    - Optimization

MPSoC Programming

Challenge: Parallel Software on Parallel Hardware

MULTICORE DESIGN SIMPLIFIED imperas

1. How do I *deliver* the software and programming environment to *customers*?

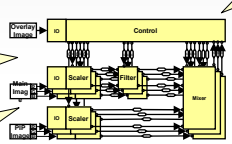
2. How do I build a *programming environment*?

3. Which *programming model and methodology* do I adopt for my customers…?

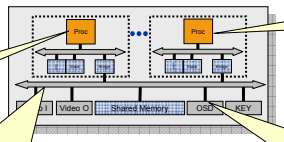
4. How do I *parallelize* the application?

5. Can I *automate* the *software verification*?

6. How do I develop and *verify* a range of reference applications for an MPSoC ?



How do I model my custom CPU?



Which CPU do I choose?

Is this the right bus architecture?

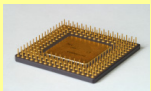
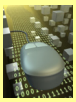
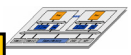
Which hardware IP do I need to develop?

Programming Environments

What does it take? Enable Interaction!

MULTICORE DESIGN SIMPLIFIED imperas

Programming environments enable MPSoC delivery

	Hardware IP Providers	Semiconductor Houses	System Houses	Software IP Providers
				
User Types	Processor & Sub-system Developer	MPSoC Developer	MPSoC Programmer	
	Virtual Processor (or Sub-system) 		 Virtual MPSoC with Programming Environment	
Challenges	1. **Deliver** processor and sub-system **models** to users **early** 2. **Demonstrate** offerings **to customers** 3. **Optimize processor or sub-system** for a range of applications	1. **Deliver** virtual MPSoC **software development environment** to MPSoC programmers early with **programming model and methodology** 2. **Demonstrate** MPSoC 3. **Optimize** MPSoC HW **architecture** for a range of applications	1. **Develop** applications **platform independent** 2. **Verify** and **optimize** legacy software 3. **Add new functionality** to existing design 4. **Optimize** MPSoC hard-ware parameters	

MP Programming Environments

Key Necessary Characteristics & Requirements

MULTICORE DESIGN SIMPLIFIED imperas

Application Programming model handling ***concurrency*** *and* ***communication***
abstractions using e.g. Threads(?), C API, platform independent, *tools/automation to map to MP hardware*

Virtual MPSoC with Programming Environment

- ***MP Application Programming model,***
- ***Tools & Methodology***
- ***MP Environment***
- ***MP Verification & Debug***
- ***MP Simulation***
- ***Models*** *(User Modeling & Model Library)*

Integrated Development Environment ***- Eclipse***
customizable and extendable, providing control, scripting and analysis

Packaged Delivery *to programmers*

Multi-Processor *Debug*

Automated *Software Verification*

Multi-Processor Simulation in the ***100 MIPS speed domain***
of platforms including multiple processors, memory hierarchies, communication, peripherals, and test environment

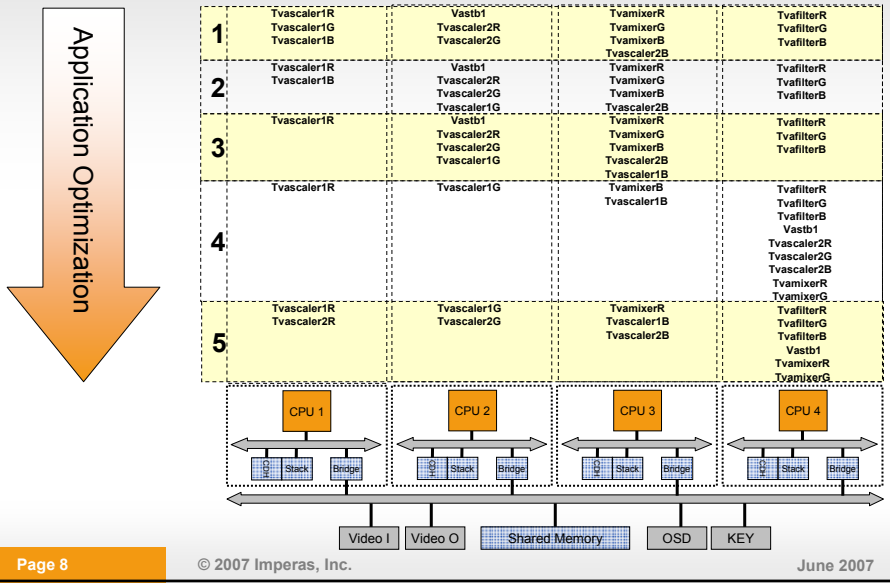
Model Library

Modeling ***open to, and done by users***
CPUs, Sub-systems, Peripherals, MPSoC Platforms and Testbenches

MP Programming Environments

What is required? Application Optimization!

MULTICORE DESIGN SIMPLIFIED imperas



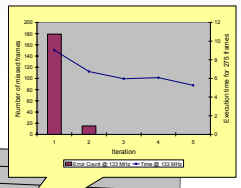
MP Programming Environments

What is required? Optimization on platforms!

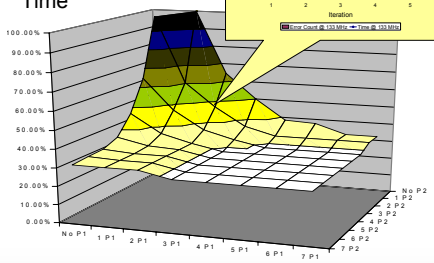
MULTICORE DESIGN SIMPLIFIED imperas

Optimizing applications on different platforms

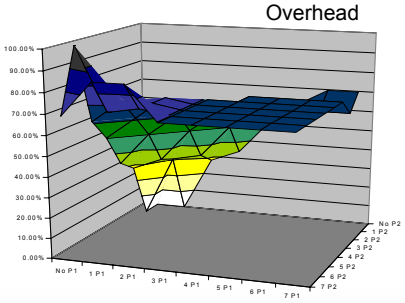
Software Optimization



Application Execution Time



Communication Overhead



MP Programming Environments

Where can you get one?

MULTICORE DESIGN SIMPLIFIED imperas

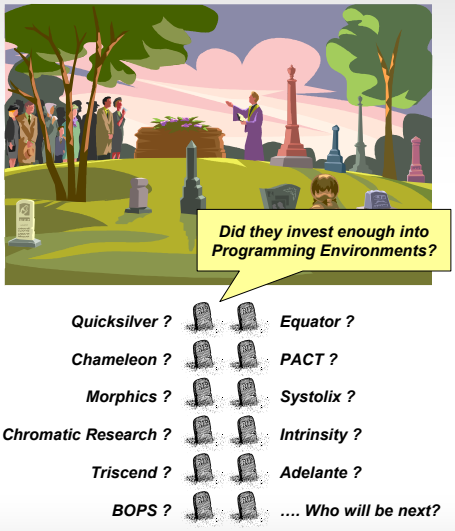
	Do It Yourself 		*Patchwork* 		***Issues***
MP Programming Model, Tools, Automation	Proprietary, non standard, on your own, unique tools	45+ man years	Proprietary, non standard, on your own, unique tools	20+ man years	Adoption – can the MPSoC be programmed? Will anyone adopt your proprietary, one chip approach. Development time, applicability, …
MP Workbench	Enhance Eclipse		Enhance Eclipse		Development time, hiring, not core competence, scaling, getting it right, hosts (Windows / Linux)
MP Verification & Debug	Patch GDB for very basic MP		GDB, Debug Vendor		Inferior solution, time, scaling, expense
MP Simulation	SystemC + "Roll Your Own" OSCI		Vendor SystemC + Vendor		Slow, single core focused, difficult integration, scaling
Model Library	Do it yourself, IP Vendor Models		Models from different vendors		Incompatibilities, speed, inappropriate control, model confusion (CA vs. IA)
Modeling	Do it yourself		Vendor based		Model speed, time to model, proprietary modeling, incompatibilities, verification challenge, inappropriate control
	Huge Effort, Delay, Costs of $9M+		**Big Effort, Delay, Costs of $5M+ plus tool licenses**		***Significant portion of effort is not specific to your chip !!!***

Programming Environments are a must have … or companies will die!

MULTICORE DESIGN SIMPLIFIED imperas

"The **ad hoc approach** to SoC design simply **cannot scale with Moore's Law** because it does not sufficiently reduce the complexity of SoC design. […] The **"software-development environment as afterthought" era** of IC design is **rapidly drawing to a close**."

Kurt Keutzer
U.C. Berkeley
"Programmable platforms will rule"
EETimes



MULTICORE DESIGN SIMPLIFIED imperas

Thank You!

Frank Schirrmeister
franks@imperas.com

