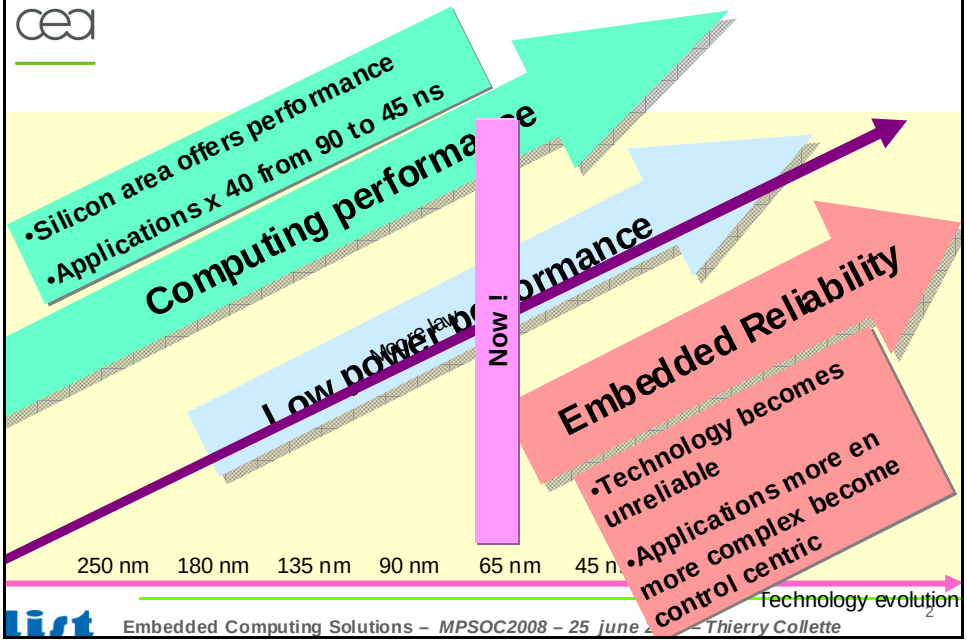
Key technologies for many core architectures

Thierry Collette

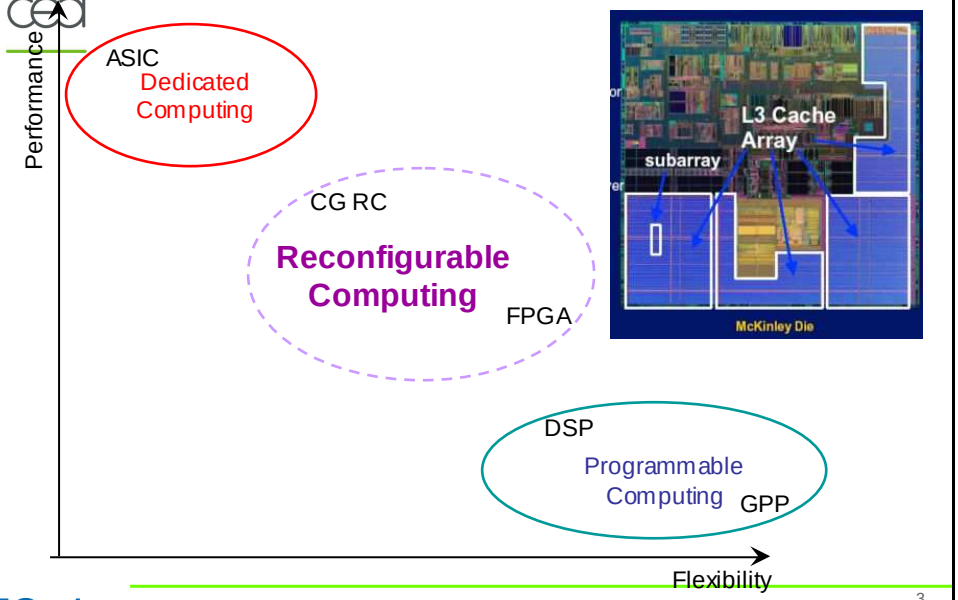
CEA, LIST

thierry.collette@cea.fr

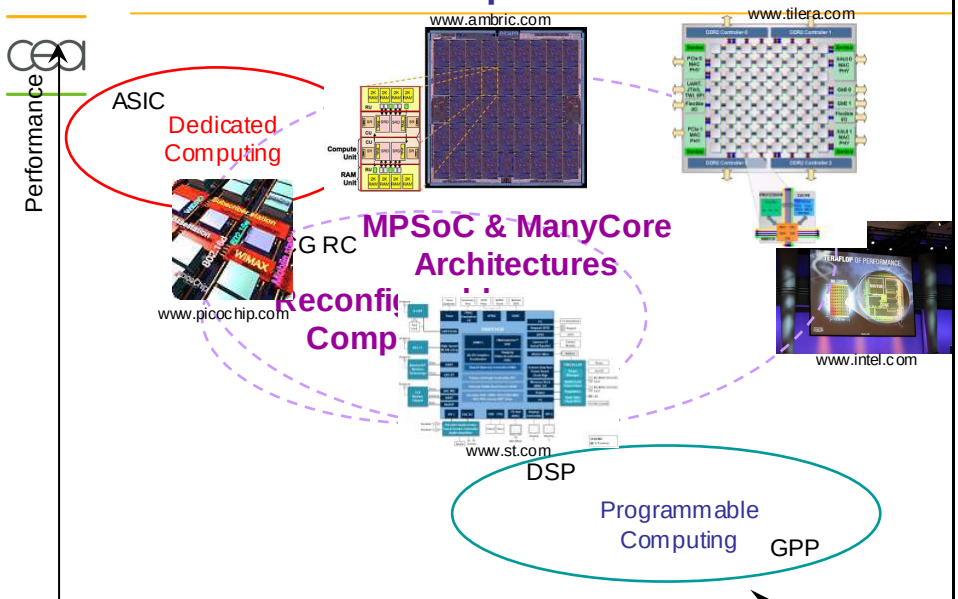
Embedded computing



Temporal acceleration



A new area : The Spatial acceleration



Programming languages

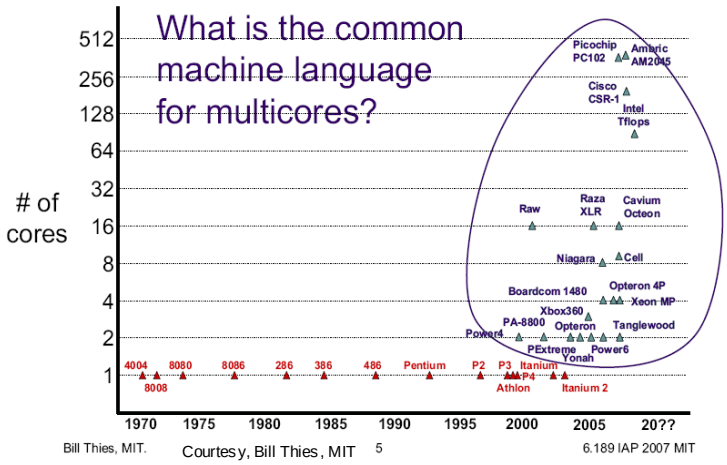
Programming Models/Languages Currently Used/Evaluated (Percentage of Survey Respondents Identifying Each)

	N	Users of Multiple Discrete Processors, N=108	Users of Multicore, N=115	Users of Both, N=123
Normal C/C++	307	88.0%	88.7%	89.4%
Titanium (Java)	42	13.0%	11.3%	12.2%
OpenMP	27	13.0%	4.3%	6.5%
MPI	18	10.2%	3.5%	2.4%
Unified Parallel C	23	7.4%	3.5%	8.9%
CAF (Fortran)	4	2.8%	0.0%	0.8%
Other	28	7.4%	6.1%	10.6%

Courtesy VDC (Multicore expo 2008)

Need of new Programming models?

Why a New Language?



Bill Thies, MIT. Courtesy, Bill Thies, MIT 5 6.189 IAP 2007 MIT

New execution model for manycore architectures

- **More than 100 processor architecture involve :**
  - ✓ New programming model for the users in order to express the parallelism ;
  - ✓ Deterministic and dependable architecture.
- **2 majors consequences :**
  - ✓ On the SW tool part :
    - → Code generation tools have to produce code for processing, communication, memorization and control ;
    - → Synthesis of embedded OS.
  - ✓ On the Architecture part :
    - → The control part is more and more important and complexe ;
    - → Management of matrix of elements with compromises.

Key technologies for manycore architectures (1)

- **Architecture definition :**
  - ✓ Definition and validation of the architecture and its execution model :
    - → Heterogeneous architecture or Homogeneous architecture?
    - → Distributed memory, shared memory or both?
    - → Which topology?
    - → Task management policy : Fully dynamic, fully static, self-time or Static assignment scheduling? Centralized or distributed?
    - → Energy and reliability aware architectures.

Key technologies for manycore architectures (2)

- **Software tools**
  - ✓ The complexity and the efficiency depend on the programming language ;
  - ✓ As for FPGA, Mapping tools, Placing Tools and Routing Tools ;
  - ✓ More the scheduling is complexe, « simpler» are the tools (analogy with VLIW processors and the Superscalar processors).

- **Embedded system Software**
  - ✓ If the mapping, place and route are not done during the compilation, they have to be executed during the run-time,
  - ✓ Processing, Memorisation and **communication** must be ensured at the run time,
  - ✓ **Energy and reliability management.**

Key technologies for manycore architectures (3)

- **Architecture design**
  - ✓ Topology design (Organisation of the matrix of processors, Memory Mapping, Communication supports, I/O, …) ;
  - ✓ Design of the support for embedded system SW (SW/HW tradeoff, centric vs distributed, etc) ;
  - ✓ Design of a power and reliability awared architecture.

- **Component design**
  - ✓ Manycore is possible on advanced technology (ie 65 ns or less) ;
  - ✓ Design of low power elements (Processor for instance) ;
  - ✓ Design of a component of more than 100 processor.

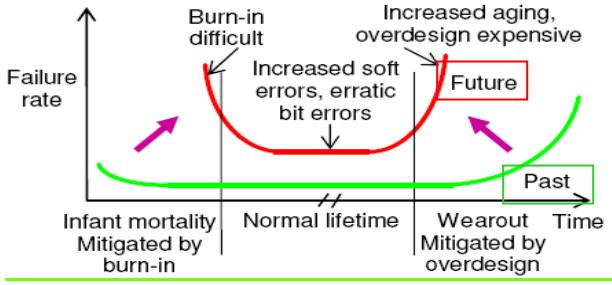
MPPA™* solution (1)

- **Is a component easy to used :**
  - ✓ The user ignore the architecture (ex : the number of processors) ;
  - ✓ Programming language based on C able to express to parallelism :
    - → Think parallel,
    - → Easy to used,
    - → Predictability.
- **Offers transparent access to the matrix of cores :**
  - ✓ Software toolchain
    - → Application code generation and embedded system software code generation,
    - → Dynamic and Dependable computing.

* Multi Purpose Processor Array

MPPA™ solution (2)

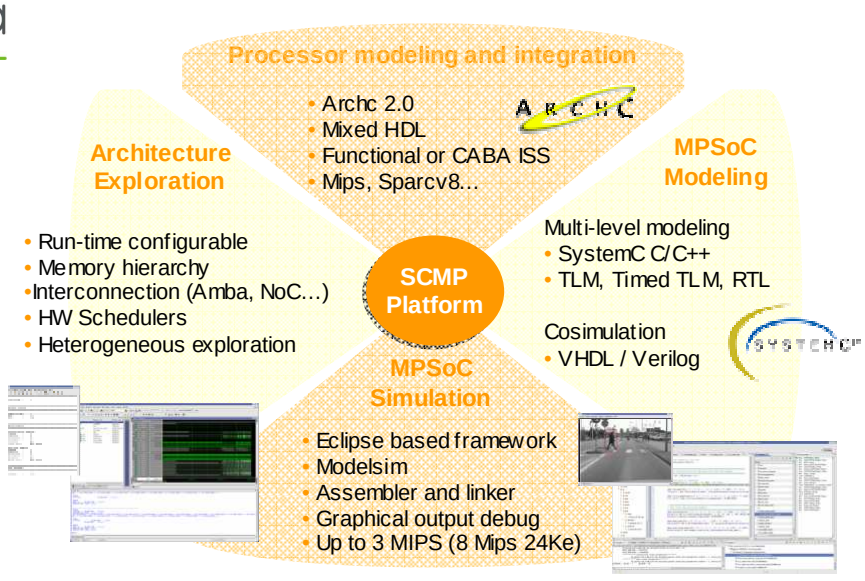
- **Offers high computing performance, low power, reliability and dependability management :**
  - ✓ Advanced execution model (TLP, control,…)
- **Able to cope with the next generations of technologies**
  - ✓ Embedded diagnosis :
    - → Static for computing for manufacturing,
    - → Online for Self Healing computing.



MPPA™ component

- **Not multi, but many processors :**
  - ✓ up to 100
  - ✓ Transparent scalability
- **Dependability and real time with a advanced execution model and the synthesis of services**
- **Advanced embedded OS for the management of performance, energy and reliability**
  - ✓ SW/HW & centralised/distributed solutions.
- **Reliability within a homogeneous platform**
- **Advanced I/O management inside the component**
  - ✓ Many I/O
  - ✓ Real time management

Architecture environment development



Conclusions

- From matrix of gates or CLB to a matrix of Processors (Many more than multi) ;
- Performance, low power, reliability and **dependability;**
- Generation of management services inside the component ;
- Computation, memorization and communication (int&ext) ;
- Parallel and reconfigurable computing on unreliable technology :
  - → Adaptative computing,
  - → Self healing computing.