

**Scalable Software Hardware Architecture Platform
for
Embedded Systems**



**Four levels of parallelism
to be managed in the
DIOPSIS based
SHAPES multi-tiled architecture**

Pier Stanislao PAOLUCCI

technology director – ATMEL Roma

& (part-time) permanent staff researcher – INFN Roma

for

the SHAPES Consortium



Tiled architectures - motivations

- Modern MPSoCs must integrate hundreds of million-gates to satisfy high performance demands from applications.
- Tiled architectures provide a scalable design style addressing two key issues: global wiring and design complexity on deep submicron technologies.
- The usage of a set of “small” mesochronous processing tiles (a few million gates each), with “short” intra-tile wires, solves the clocking issue.
- The replication of stable and validated processors and tiles maintains a manageable complexity at system level.
- Business motivations – maintain high unit price and Gross Profit Margin– escape from commodity model – create a roadmap

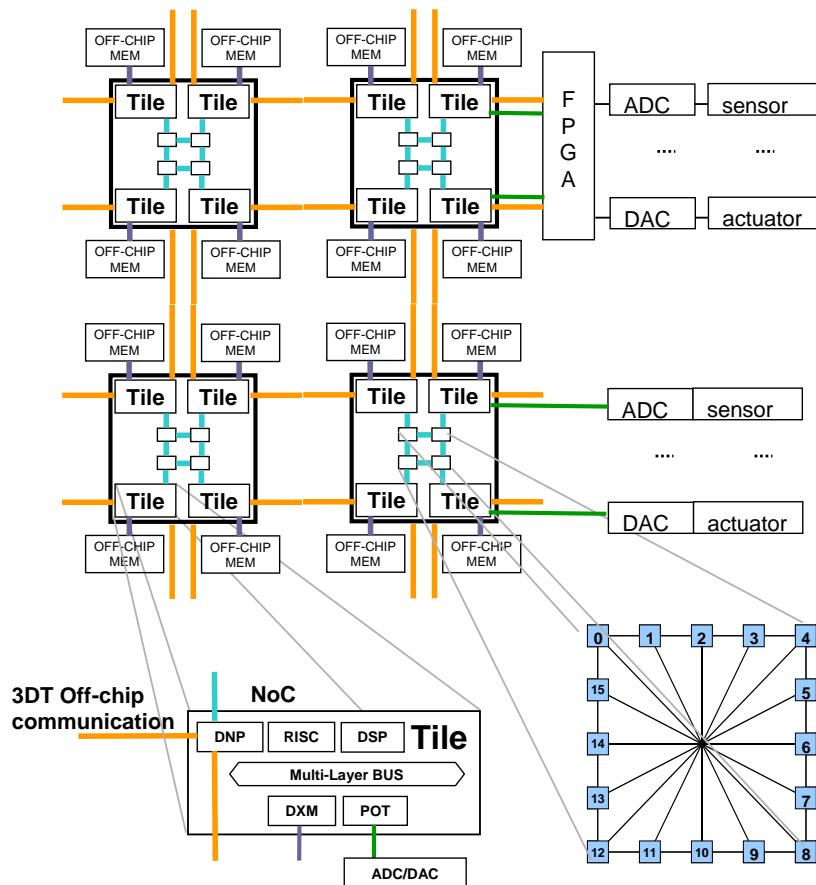


- SHAPES multi-tiled system:
 - based on a multi-processor heterogeneous tile
 - evolution of the ATMEL DIOPSIS MPSoC
 - NoC based on ST Microelectronics Spidergon technology
 - Each multiprocessor tile typically includes:
 - a VLIW floating-point DSP (native C programmable Gigaflops mAgicV Digital Signal Processor, from ATMEL)
 - a RISC controller (ARM926)
 - a DNP (Distributed Network Processor, from INFN)
 - Each tile always includes several banks of intra-tile memory!!!
 - Each tile can be individually connected with:
 - a DXM (Distributed External Memory),
 - a set of peripherals (e.g. ADC/DAC),
 - A routing fabric connects on-chip and off-chip tiles, weaving a distributed packet switching network.
 - 3D next-neighbours toroidal connections are adopted for off-chip networking and maximum system density.



Tiled HW Architecture

- Communication Centric, not Processor Centric
- Homogeneous SW interface for on-chip and off-chip scalable connection and I/O
- Virtual tunnelling on packed switching NoC (Network on Chip) and off-chip 3DT

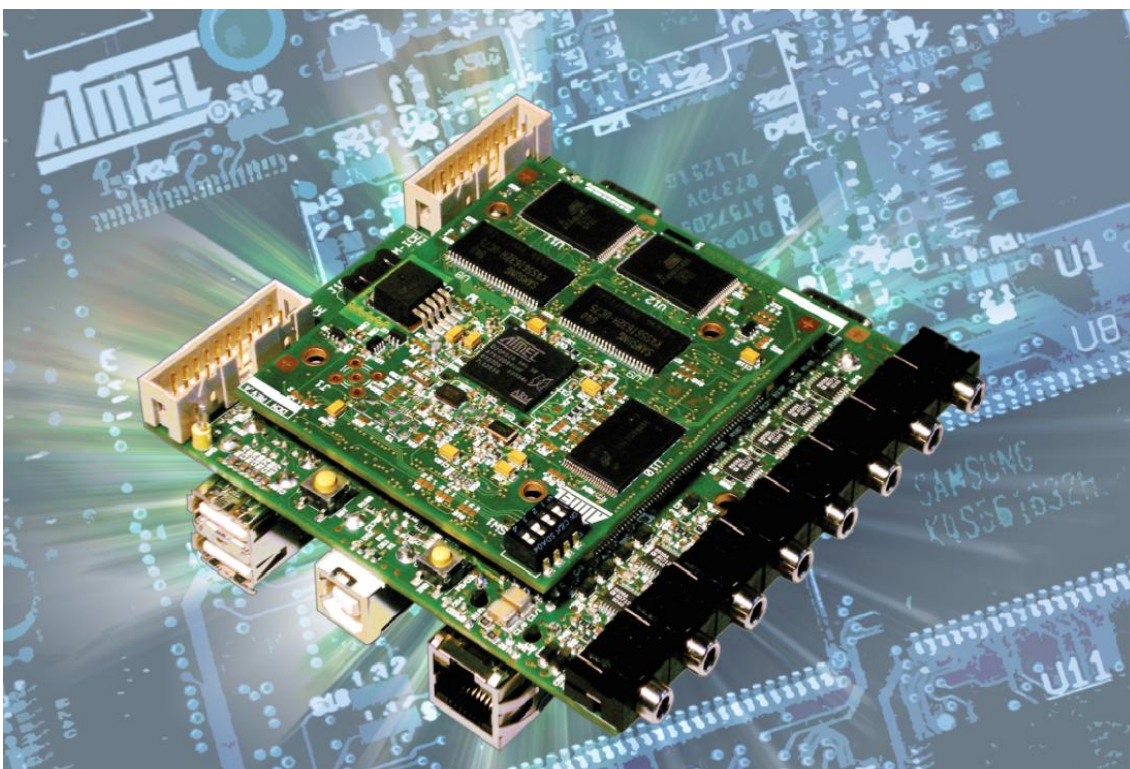


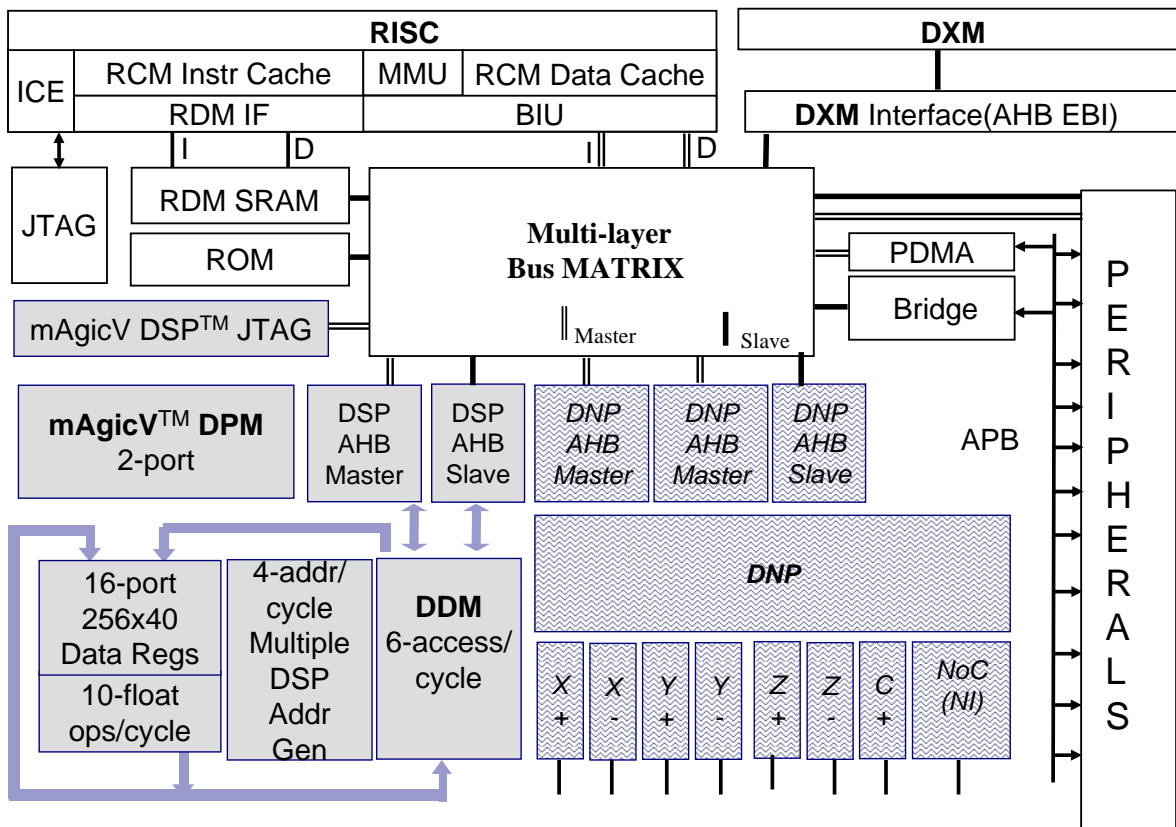
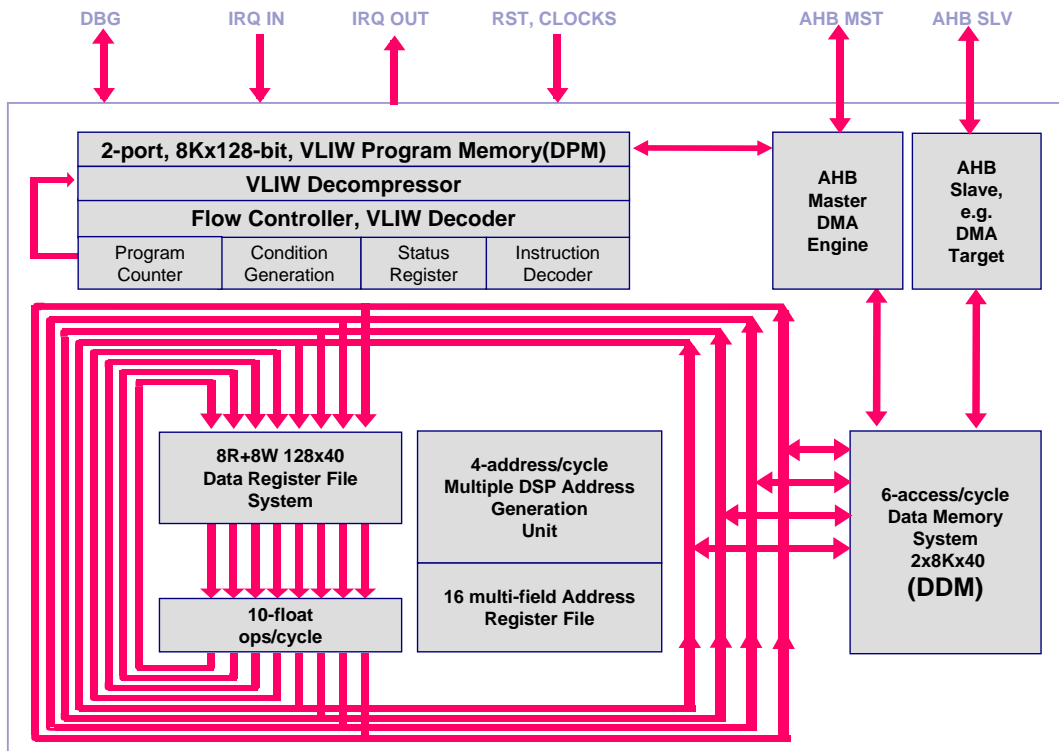


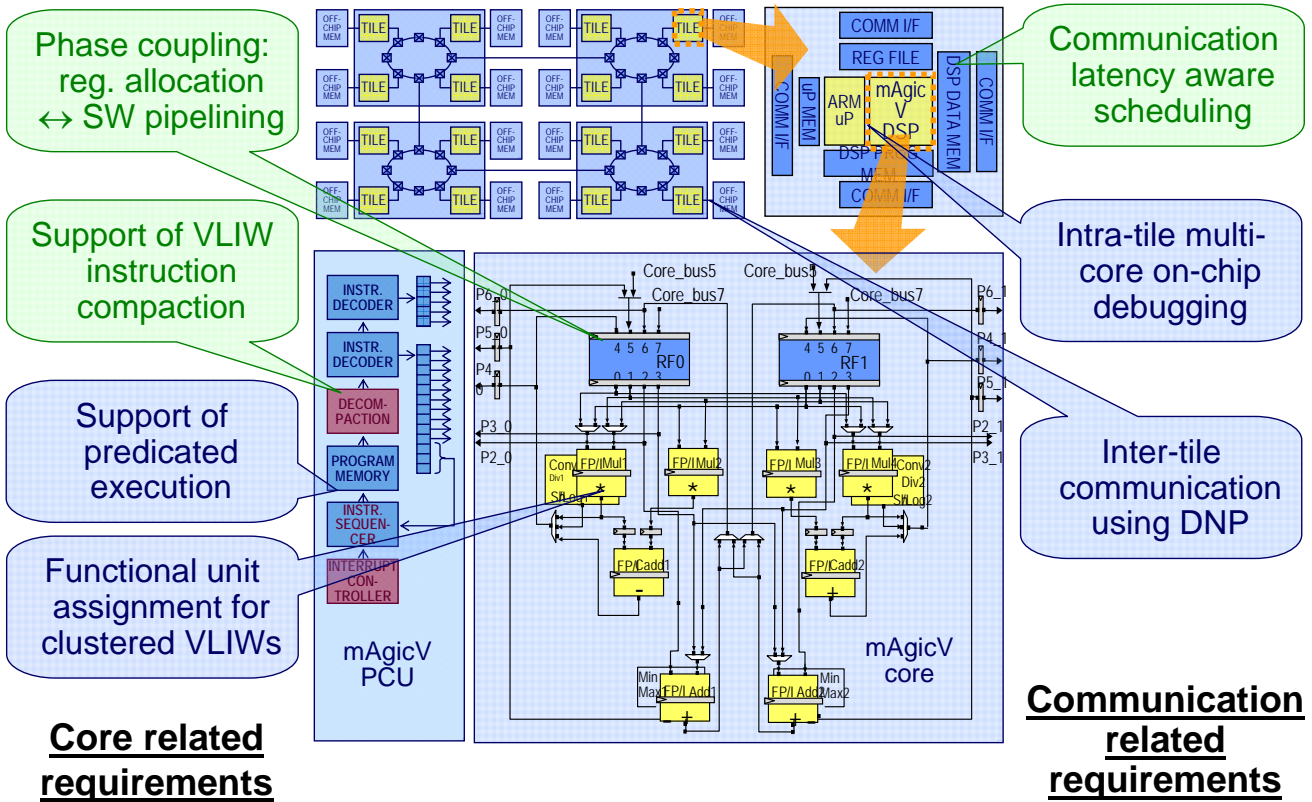
- The SW challenge is to provide a simple and efficient programming environment able to manage four levels of parallelism:
 - **Level 1.** Inside each processor: e.g. exploiting the 15 operations / clock cycle of the multiple-issue mAgicV floating-point VLIW DSP of ATMEL, and its software-managed multi-bank memory system;
 - **Level 2.** Inside each tile, e.g. activating the parallel operation of the RISC, DSP and DNP processors inside the tile, supported by a SW managed multi-layer bus matrix, which permits multiple simultaneous intra-tile data transfers between the intra-tile processors and the Distributed External Memory attached to each tile.
 - **Level 3.** Inside each multi-tile chip, managing the inter-tile DNP-NoC-DNP packet transfers, for a number of simultaneous data transfers which should grow proportionally to the number of tiles;
 - **Level 4.** At multi-chip system level, using the 3D toroidal off chip Network.

Diopsis 940 overview

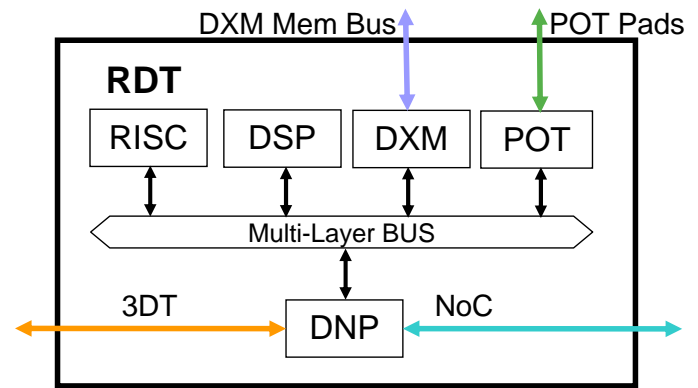
Diopsis940 Development Kit



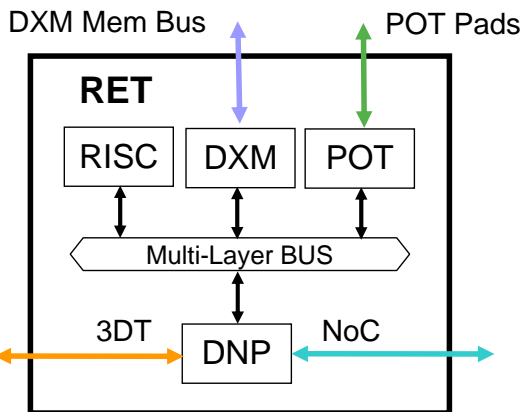




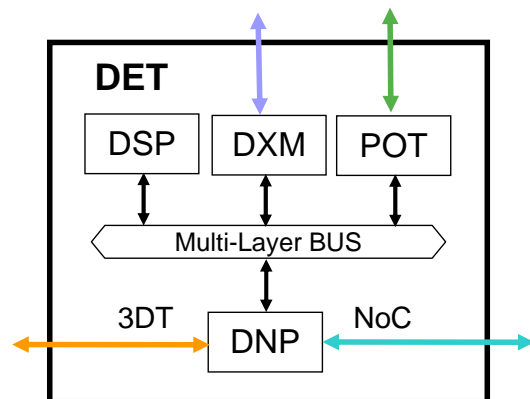
Different Types of Tiles



RDT: RISC + DSP Elementary Tile



RET: RISC Elementary Tile



DET: DSP Elementary Tile

INSIDE THE TILE

- Multilayer Bus Matrix sustains multiple simultaneous transfers
- RISC max one per tile
- DSP one or more per tile
- DNP: Distributed Network Processor (always one per tile)
- DDM: on-chip Distributed Data Mem (inside the DSP)
- DPM: on-chip Distributed Progr Mem (inside the DSP)
- DXM: Distributed eXternal Mem Interface (max one per tile, outside the RISC and DSP)
- POT: Peripherals On Tile
- RDM: Risc (tightly coupled) Data Memory
- RPM: Risc (tightly coupled) Program Memory
- RCM: Risc Cache Memory

AT THE CHIP LEVEL

- MTC: Multiple Tile Chip (composed of multiple Tiles)
- NOC: Network On Chip (connecting Tiles)
- 3DT: 3 Dim Toroidal Connection (outside the chip)

FUNDAMENTAL TYPE OF TILE

- RDT includes:
 - RISC: (includes on chip memories RDM and RPM) +
 - DSP (includes on chip memories DDM and DPM)
 - DNP + DXM (off-chip mem) + POT (e.g. DAC/ADC conv)

POSSIBLE TILE VARIANTS

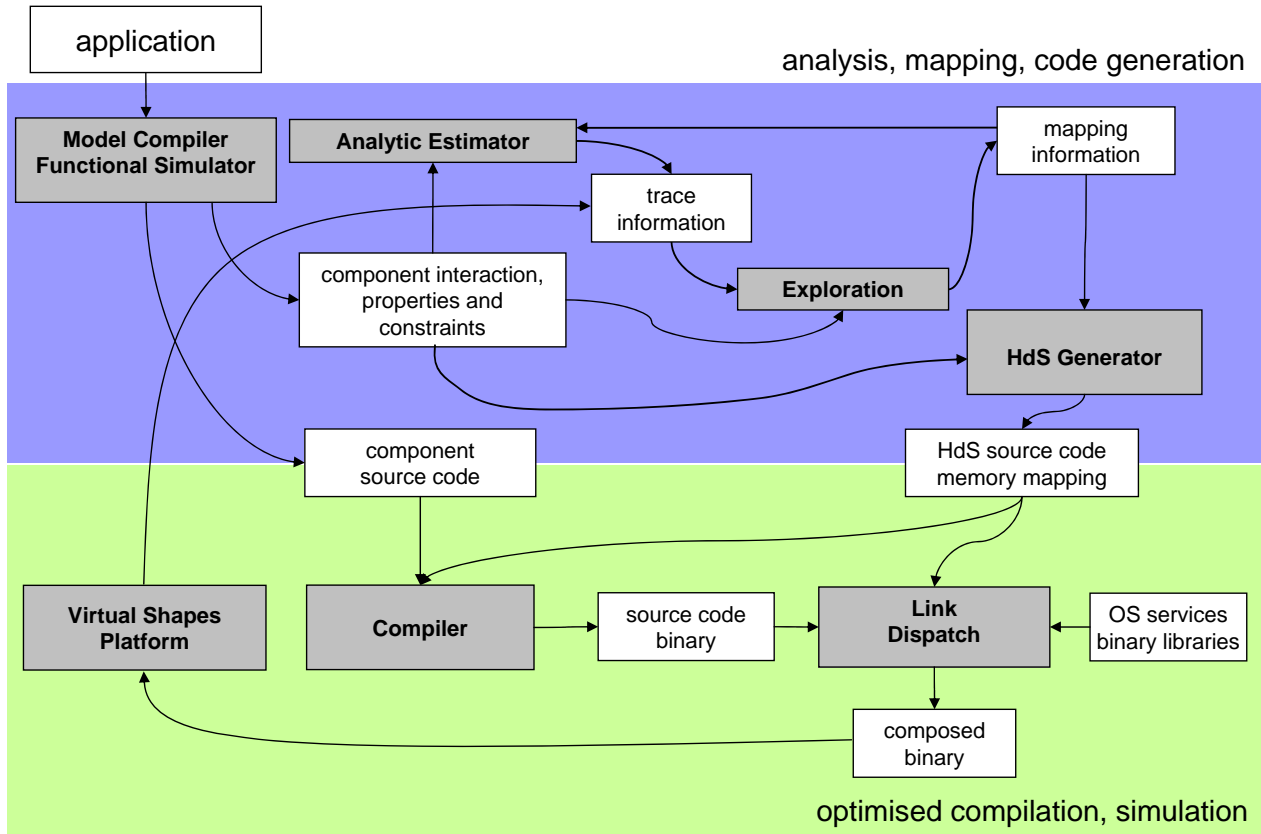
(subset of RDT)

- RET := RDT minus DSP
- DET := RDT minus RISC
- DDT := DET minus DXM

Scalability

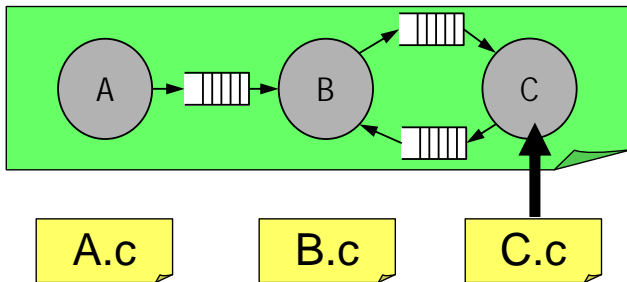
Efficiency

Predictability



13

SW – Native C Application Programming Environment, Ideal for DSP people



Application Description using the style preferred by DSP experts: a Network of “Processing blocks”, e.g FFT, FIR, DAC in, ...

Complete functional debug/simulation on your PC!!!

Processing Blocks are written in C plus message passing

Graphical description of the “Network of Processes”

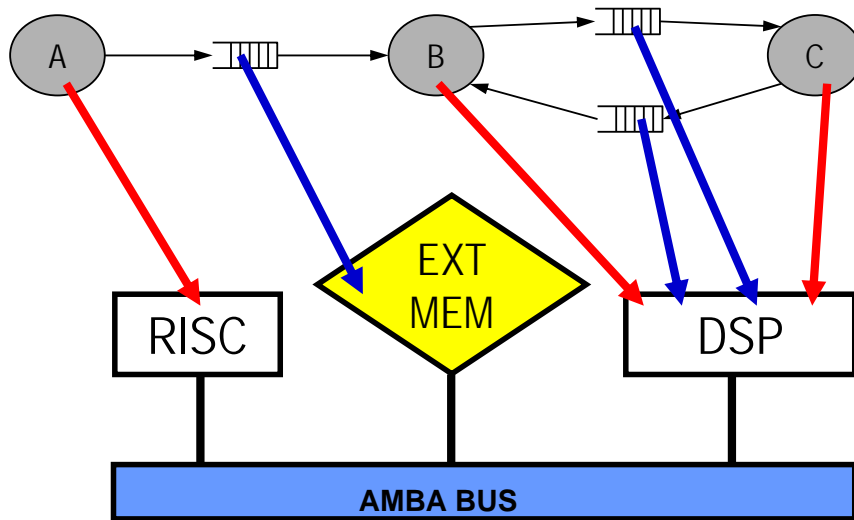
Automatic mapping on:

- Diopsis940HF: 1 RISC + 1 DSP
- Systems with Multiple Diopsis/mAgicV like tiles
- Reduces porting from PC to Diopsis940hF by a factor 3x

14



- Automated Mapping on Diopsis = Binding + Scheduling
- Automated generation of Diopsis executable/OS services
- Automated performance analysis



System SW

ETH Zurich - *Distributed Operation Layer & Automated Mapping*

TIMA Lab and THALES - *Hardware dependent Software Layer and RTOS*

TARGET Compiler Tech. - *Retargetable Compilers*

RWTH Aachen Univ. - *Fast Simulation of Heterogeneous Multi Proc. Systems*

System HW

ATMEL Roma - *Tile:*

Evolution of (Diopsis®: mAgicV VLIW DSP™ + RISC) + INFN DNPT™

INFN Roma - *DNPT™ Distributed Network Processor + 3D Toroidal Eng.:*

Evolution of APE Massive Parallel Processors

STMicroelectronics + Univ. of Cagliari and Pisa - *Network on Chip:*

Evolution of Spidergon™ Packet Switching Network on Chip

Univ Roma 1 Sapienza - *Deep Sub-micron Issues*

Parallel Application benchmarking

Fraunhofer IDMT - *multi-loudspeaker Audio Wave Field Synthesis*

ESAOTE, MedCom, Fraunhofer IGD - *Ultrasound scanner*

INFN - *Physical Modelling - Lattice Quantum Chromo Dynamic*

ATMEL - *Multi-microphone arrays for voice-extraction*



- 4 hierarchical levels of parallelism to be managed on tiled SHAPES architectures
- SHAPES investigating a possible solution for applications with inherent large degree of parallelism
- Solving solvable problems with good Gross Profit Margin and good market growth rate...
 - Gaming
 - Surveillance
 - Vocal Command → Speech Recognition
 - Robotics
- Do we really need to apply many-cores to sequential problems?
- After all, the brain is a highly parallel engine executing applications with inherent large degree of parallelism. It seems working without cache coherence, transactional memories and the like...
- TRIVIA?: let us focus on solving solvable problems