2PARMA Project

PARallel PARadigms and Run-time MAnagement techniques for Many-core Architectures

INA-OCMC’2012

Prof. William Fornaciari
Politecnico di Milano
fornacia@elet.polimi.it
home.dei.polimi.it/fornacia
Project Technical Manager of 2PARMA

FP7-248716-2PARMA Project
List of Project Partners

1. Politecnico di Milano (POLIMI) – Italy (Coordinator)
2. STMicroelectronics (STM) – Italy
3. Fraunhofer Institut for Telecommunications / Heinrich-Hertz Institut (HHI) – Germany
4. Interuniversitair Micro-Electronica Centrum (IMEC) – Belgium
5. Institute of Communication and Computer Systems (ICCS) – Greece
6. RHEINISCH-WESTFAELISCHE TECHNISCHE HOCHSCHULE AACHEN (RWTH) – Germany
7. Synopsys (CoWare) – Belgium

**Start date**: January 1st, 2010  
**Project duration**: 3 years  
**Total effort**: 408 PM  
**EC Contribution**: 2.74 M€
Back-ground & side-ground projects

- PRO3D
- Artemis - SMECY
- ENIAC- TOISE
- HiPEAC2
- HiPEAC2
Scientific and Technical Objectives

Main Goals

• Programmability of Many-core Computing Fabrics
• Virtualisation and Continuous Adaptation
• Design Space Exploration
• Runtime Adaptivity

Project Outcomes

• Integrated Compiler Toolchain and OS Layer
• DSE Toolchain
• Run-time Resource Manager

The 2PARMA project focuses on the definition of suitable parallel programming models, instruction set virtualisation, run-time energy/power and resource management policies and mechanisms as well as design space exploration methodologies for Many-core Computing Fabrics.
The 2PARMA project focuses on the flexible family of parallel and scalable computing processors, which we call Many-core Computing Fabric Template, composed of many processing cores interconnected by an on-chip network.
Platform 2012: Scalable Architecture Template


The node:
- From 8 to 16 cores
- Shared memory architecture
- Floating point extension instead of vectorial extension
- Capability to add HW blocks in the cluster

FP7-248716-2PARMA Project

Source: STMicroelectronics
P2012 Cluster

- **Cluster Architecture evolutions** supporting many applications.

- **Heterogeneous Cluster RTL implementation(s):**
  - ENCore<N>,
  - Cluster Controller,
    - Cluster Control,
    - DMA Control,
  - Debug logic,
  - Stream-flow logic,
  - User-def. HWPEs.

**User-defined HWPEs**

**Stream-flow logic**

*Inter-HWPE communication, L3 or shared Memory ⇔ HWPEs communication.*

**Cluster Control**

- ENCore Boot, HWPE control, etc...

**DMA Control**

- L3 ⇔ Sh. Mem., L3 ⇔ Stream, Sh. Mem. ⇔ Stream

**Configurable** (N, EFUs, banking factor, ...)

- “Computing Farm” (supplied by “Computing”)

- For SW PEs and/or intensive control

**Debug** Multicore Debug

Source: STMicroelectronics
Platform 2012: Positioning

GOPS/mm²/W in 32 nm

General-purpose Computing

CPU

Throughput Computing

GPGPU

P2012 Space

HW IP

Source: STMicroelectronics
Synopsys Virtual Platforms

1. Abstract model mimicking P2012 in support of CBSE methodology

1. ARM based multi-core platform supporting Linux OS for design tool validation
   - Integration with DSE tools
   - Investigate RTRM support
2PARMA Target Applications

- Scalable Video Coding (SVC) – HHI

- Cognitive Radio
  - Physical Layer – RWTH-ISS
  - MAC Layer – RWTH-iNETs
  - Reconfigurable Radio – IMEC

- Multi-view Image Processing - IMEC
Applications/Architecture Integration

Cognitive Radio
- MAC
- PHY
- REC

Scalable Video Coding

Multi-view

Run-time Management and Virtualisation

OS Integration Layer

FP7-248716-2PARMA Project
The 2PARMA Run-Time Resource Manager
Overall view

User space
Dynamic Code Generation

App cr

App be

RTLib
Res Accounting
Res Partitioning
Resource Abstraction
MRAPI
Platform Proxy

Kernel space

Platform Driver

supported platforms
Task Mapping
DDM

Platform Firmware

Application-Specific RTRM
System-Wide RTRM

Legend

SW Interface (API)
SW/HW Meta-data

FP7-248716-2PARMA Project
WP2: Programmability of Parallel Computing Systems to Exploit Task/Data Parallelism

**Partners role**

**RWTH**: Component-based SE toolchain to support applications design

**POLIMI**: OpenCL compilation toolchain. Dynamic compilation to adapt parallelism to system resources

**STM**: Operating System Integration. Development of device drivers to support run-time management
Mapping OpenCL to Computing Fabrics

- Analysis of OpenCL programming model (strengths and weaknesses) vs P2012 Computing Fabric
- Specification of Computing Fabric-specific extensions to OpenCL
- Proposed OpenCL extensions targeting Computing Fabrics
- OpenCL front-end compilation toolchain for LLVM to be publicly released soon
- Dynamic compilation to adapt parallelism to system resources
Nucleus Methodology

- Transceiver Description
- Nuclei
- Mapping & Evaluation
- Board Support Package
- HW Platform

Nucleus Project within UMIC research cluster at RWTH Aachen University
Component-based Software Engineering Toolchain

User inputs
- Config & constraints
- Waveform description

Libraries
- Nucleus Library
- Board Support Package
- Flavor Library
- Platform Description
- 2PARMA IP – WP4

Nucleus Mapper

Code Generator

Synopsys Virtual Platform

STM P2012 Platform

STM P2012 SDK required

Application domain specific knowledge required

Nucleus IP required

Synopsys IP required
PA, VPU library, …

Nucleus Project within UMIC research cluster at RWTH Aachen University
WP3: Co-exploration of Architectural Platforms and Programming Models

**Application Design and Compilation Toolchain**
- Applications Description
- Component Interfaces and SW-Channels
- Component-Based Toolchain
- Compiler to Bytecode and Native Code
- Portable Bytecode
- Native Code
- Instruction Set Virtualization

**Runtime Platform**
- Run-Time Management
- Computing Fabrics Device Drivers
- Many-Core Computing Fabric

**Partners role**

**SYNOPSIS:** EDA tools and virtual platform provider

**IMEC:** Support for run-time task mapping and scheduling

**POLIMI:** Design Space Exploration for supporting run-time system management

**HHI:** Profiling methodology for parallel computing platform. Efficiency analysis of parallel programming models.
HHI NoCTrace Architecture

- NoCTrace backend
  - collects data

- NoCTrace frontend
  - processes data
  - manages data (persistence)
  - presents data (GUI)
HHI NoCTrace Backend

- Extract/record program counter and transaction data in SystemC kernel
- Perform automatic design analysis to get to know where to place the probes
Automatic Multi-Objective Design
Space Exploration

Source: Politecnico di Milano - MPEG2 decoder - MIPS-based multiprocessor - 70nm tech. node - 10 frames
Automatic Multi-Objective Design
Space Exploration

Source: Politecnico di Milano - MPEG2 decoder, generated with Multicube Explorer
Based on the **design-time exploration**, we derive a set of Pareto **operating points** corresponding to power, resources (number of cores) and QoS (average time per frame).

- The operating points will be used by the **Run-time Resource Manager** to achieve QoS requirements (average time per frame) while meeting overall resources (number of cores) and minimizing power consumption.
Run Time Management of multi-core platform

- The system state can change due to some events:
  - A new application executed or
  - QoS requirements modified

8-Core IMEC Virtual Platform Architecture:

FP7-248716-2PARMA Project
WP4: Runtime Resource Management

Partners role

**ICCS:** Adaptive dynamic data management

**IMEC:** Adaptive run-time task mapping and scheduling

**POLIMI:** Run-time QoS constrained resource and power manager at the OS-level

**STM:** Operating System Integration, Development of Device Drivers
Run-time resource management (RTRM): Overview

RTRM offers an integrated solution and the mechanisms to perform optimal WM selection at run-time.
What is a working mode?

Platform Parameters → Application Parameters → Adaptive DMM Parameters (example)

Design Space Exploration

<table>
<thead>
<tr>
<th>Platform Params</th>
<th>App. Params</th>
<th>Adaptive DMM Params</th>
</tr>
</thead>
<tbody>
<tr>
<td>#CPUs</td>
<td>$mem Size</td>
<td>Instr. width</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

FP7-248716-2PARMA Project
Example of Intra-Heap MTh-DMM Design Space

Intra-Thread Level Design Space

E. Block Structure Decisions

- Block Sizes
  - One Size
  - Many Sizes
- Block Recorded Info
  - Size
  - Status
- Block Tags
  - None
  - Header
  - Boundary Tags
- Pointers
- Block Structure
  - SLL
  - DLL
  - Dynamic Array

F. Pool Organization Decisions

- Pool Structure Based on Block Size
  - One Pool per Size
  - Single Pool
- Pool Structure Based on Blocks Order
  - One Pool per Block Order
  - Single Pool
- Pool Structure Based on Block Address
  - One Pool per Block Address
  - Single Pool
- Pool Structure
  - SLL
  - DLL
  - Dynamic Array
Project Promises

1. Increased performance, power-efficiency and reliability of Many-core Computing Fabrics by means of:
   - Fine grained platform configuration
   - Dynamic resource management techniques

2. Improved time to market for high-performance applications requiring hardware acceleration by means of:
   - ISA virtualisation on computing fabrics.
   - Supporting the full toolchain, including compilation and OS support

3. Reinforced European scientific and technological leadership in the multi-core computing architectures both at:
   - Industry side (STM as a large company and COWARE as a SME)
   - Academic side (POLIMI, HHI, IMEC, ICCS and RWTH)

4. The project will contribute to Free and Open Source projects:
   - LLVM Static and Dynamic compiler
   - MULTICUBE Explorer framework
   - Runtime Resource Manager on top of Linux OS

5. The project will also spearhead the evolution of standards in the field of parallel programming models and languages (such as OpenCL)

FP7-248716-2PARMA Project
Demo of the Barbeque RTRM
Dynamic Resource Partitioning

Grant resources to critical workloads while optimize resource usage by best-effort workloads

- Considering a mixed-workload scenario
  - critical workload (High Priority)
    - Application: SVCDemo
      - app using SVC Video Decoding library by HHI (x86_32 version)
  - other workloads runs concurrently (Lower Priority)
    - Applications: 1 SVCDemo + 3 BbqRTLlibTestApp
      - BbqRTLlibTestApp as background "disturbing" apps

- Goals: application integration and dynamic resource partitioning
  - according to apps priority and resources demand
  - synthetic recipes (no DSE based), mechanisms evaluation (behaviors, overheads, stability,...)
  - no policies nor platform control assessment
Barbeque RTRM
Framework Demo on Dynamic Resource Partitioning