2PARMA Project and P2012 Platform

REFLECT and 2PARMA Fall 2012 School: Programming Paradigms for Multi-­‐Core Embedded Systems
October 2, 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) [Y1-Y2]– Belgium

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

2PARMA

PLATFORM 2012

MULTICUBE

OPEN MEDIA PLATFORM

MOSART

MNEMEE

PRO3D

Artemis - SMECY

ENIAC- TOISE

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

Platform 2012: 
A Many-core Programmable Accelerator for Ultra-Efficient Embedded Computing in Nanometer Technology

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 \(\Leftrightarrow\) HWPEs communication.

Cluster Control
ENCORE Boot, HWPE control, etc...

DMA Control
L3 \(\Leftrightarrow\) Sh. Mem.,
L3 \(\Leftrightarrow\) Stream,
Sh. Mem. \(\Leftrightarrow\) 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

Throughput Computing

CPU

GPGPU

SW

Mixed

HW

> 100

P2012 Space

FP7-248716-2PARMA Project

Source: STMicroelectronics
More on P2012 (now STHORM)
IMEC COBRA Platform
Synopsys Virtual Platforms [Y1 and Y2]

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
Clustered General Purpose [Y3]

- Reference multicore x86 architecture
  - AMD NUMA machine with four nodes (clusters)
- Each node is a quad core Opteron 8378 processor
- L1 cache is composed of 64KBytes data cache and 64KBytes instruction cache
- L2 cache is a 512KBytes unified cache
- All cores within a node share unified 6144KBytes L3 cache
- Inter-node communication is supported by a ring network topology

![Clustered General Purpose Diagram]
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
### 2PARMA Use cases [Y3]

<table>
<thead>
<tr>
<th>Use Case</th>
<th>UC1a</th>
<th>UC3</th>
<th>UC4a</th>
<th>UC5</th>
<th>UC6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>CR PHY Layer (CBSE Nuclei)</td>
<td>CR MAC (CBSE Nuclei)</td>
<td>SVC (CBSE MIND)</td>
<td>MultiVIEW (OpenCL) + SVC (MIND)</td>
<td>MultiVIEW (OpenCL)</td>
</tr>
<tr>
<td>Objective</td>
<td>Performance evaluation of multi-core architectures</td>
<td>Performance evaluation of multi-core architectures</td>
<td>Integration &amp; Final Validation &amp; Demonstration</td>
<td>OpenCL Toolchain Integration &amp; Validation</td>
<td>Broader Applicability</td>
</tr>
</tbody>
</table>

**Vertical Dimension:**
Tools Integration

**Horizontal Dimension:**
Validation of the broad applicability of tools and techniques

**Legenda**
- NA = Not Applicable
- 1 = Data and Resource Management are critical for keeping real-time constraints therefore data and task allocation should be managed statically.
- 2 = CBSE Nuclei Toolchain includes runtime task monitoring and management.
- 3 = Noc Trace with reduced functionalities due to the lack of SystemC simulator.
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

PEs

HW Platform

Nucleus Project within UMIC research cluster at RWTH Aachen University
Nucleus Methodology

Transceiver Description

Nuclei

Mapping & Evaluation

Board Support Package

HW Platform

Compile

Flavors

Nucleus Library

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

- Application domain specific knowledge required
- Nucleus IP required
- Synopsys IP required PA, VPU library, ...

User inputs
- Config & constraints
- Waveform description

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

2PARMA

STM P2012 SDK required

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

**Partners role**

**SYNOPSYS:** 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 (persistency)
  - presents data (GUI)

FP7-248716-2PARMA Project
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, 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.

**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
2PARMA RTRM – Different levels

System-level → BBQ

Application level
- Application parameter monitoring and reconfiguration
- Heap management → dmmlib

Platform Level
- Run-time on the chip for thermal/power management, scheduling, ...

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

Design Space Exploration

Platform Parameters

Application Parameters

Adaptive DMM Parameters (example)

<table>
<thead>
<tr>
<th>#CPUs</th>
<th>$mem Size</th>
<th>Instr. width</th>
<th>...</th>
<th>Multiview</th>
<th>SDR</th>
<th>Heap arch</th>
<th>Locking</th>
<th>...</th>
<th># Lists</th>
<th>Fit policy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Objectives of each RTRM level

BBQ objective → to perform an efficient resource partitioning w.r.t run-time optimization goals
Application parameter monitoring and reconfiguration → Continuously maximize output quality while meeting constraints
Heap management → to accommodate the applications’ needs within the allocated memory space (from the OS/system-wide controller)
2PARMA RTRM control approach
Hierarchical control
- System-wide level (e.g., resource partitioning)
- Application specific (tuning of application parameters, DMM)
- Firmware/OS level (frequency selection, thermal alarm, resource availability, adaptive voltage scaling, ...)

Closed control loop enhanced with feed forward
- OP and AWM increase convergence speed

Distributed control
- Several subsystems have their own control loop

Optimal
- User defined goal functions (including overheads)

Robust
- Can adapt to the model characteristics

Adaptive
- The control algorithm is run-time computed

Enforce observability and controllability
- Rich set of sensing and control capabilities, both for applications and platforms
- Bounded and measurable response time
The 2PARMA Run-Time Resource Manager
Overall view

Application-Specific RTRM

System-Wide RTRM

User space

Dynamic Code Generation

App_{cr}  App_{be}

RTLib

Res Accounting  Res Partitioning

Resource Abstraction

MRAPI  Platform Proxy

Platform Driver

Platform Firmware

Legend

- SW Interface (API)
- SW/HW Meta-data

supported platforms

Task Mapping

DDM

Kernel space

supported platforms

Task Mapping

DDM

FP7-248716-2PARMA Project
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)
   - 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 (BBQ) 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
January 14-15, 2010 FP7-248716-2PARMA Project

www.2parma.eu

2PARMA

- Acronym: 2PARMA
- Title: PARalel PAredigma and Run-time MANagement techniques for Many-core Architectures
- Project ID: FP7-ICT-2009-4-249716
- Project Duration: January 2010 - December 2012
- Project Coordinator: Prof. Cristina Silvano, POLIMI
- Project Technical Manager: Prof. William Formentari, POLIMI

- Keywords: Parallelization & Programmability, Continuous Adaptation, Virtualization, Customisation, Methodologies & Tools

Partners

[ 2PARMA Publishable Summary for Year 1 (2010) (PDF Version) ]
[ 2PARMA Project Fiche (PDF Version) ]
[ 2PARMA Flyer (PDF Version) ]
[ 2PARMA Poster (PDF Version) ]
Thanks for your attention
Any question?

Prof. William Fornaciari
Politecnico di Milano
fornacia@elet.polimi.it
home.dei.polimi.it/fornacia

Project Technical Manager of 2PARMA