European Multicore Processing Projects

The following brief project descriptions give a good overview of the variety of topics the research community in Europe is addressing.

HiPEAC

The High Performance and Embedded Architecture and Compilation (HiPEAC, http://www.hipeac.eu) project has a threefold mission: to steer and increase European research in high-performance and embedded computing, to stimulate cooperation between computer architects and tool builders, and to create a visible and integrated community in the HiPEAC area in Europe.

HiPEAC activities can be grouped into three types. First, there are activities to stimulate mobility, such as internships, collaboration grants, and semi-annual cluster meetings where the community discusses research in thematic meetings. The second type of activity seeks to coordinate and steer research in Europe. Research clusters and the roadmap are two examples. The current cluster topics are programming models, simulation, compilation, multicore hardware, interconnects, virtualization, and reconfigurable computing. (See ''ArchExplorer for Automatic Design Space Exploration'' for an example of a cluster collaboration in the simulation area.) The HiPEAC roadmap describes the research vision of the HiPEAC community. Finally, there are the integration and visibility activities. HiPEAC organizes the HiPEAC conference (http://www.hipeac.eu/conference) in January and the ACACES summer school (http://www.hipeac.eu/summerschool), two well-attended scientific activities. Other activities in this category are the quarterly HiPEAC newsletter, journal, and awards; the promotion of startups; industrial workshops; Web seminars; and the many activities organized by the different clusters.

Network coordinator: Koen De Bosschere, Koen.DeBosschere@UGent.be

ACTORS

The EU Adaptivity and Control of Resources in Embedded Systems (ACTORS, http://www.actors-project.eu) project is developing concepts and methodologies for designing resource-aware applications and tools and languages for multicore programming. The need for a more dynamic approach to resource management is driven both by more demanding use cases and more complex hardware, specifically multicore systems. The project partners include researchers in real-time scheduling, video coding, control theory, compiler design, telecommunications, and embedded software. The project's three themes are adaptive resource management, dataflow programming for multicore systems, and reservation-based scheduling.

The project has successfully contributed to the MPEG/ISO standardization and was instrumental in getting the reconfigurable video coding (RVC) standards (ISO/IEC 23001-4 and 23002-4) published as international standards in December 2009. RVC aims at creating more flexible decoders, in terms of both standard specification and implementations, and is based on dataflow programming using the CAL actor language. CAL compilers targeting both multicore systems and FPGAs are made freely available. The work on reservation-based scheduling has resulted in a scheduler patch for the Linux kernel, called SCHED_DEADLINE, which was released at the Linux kernel mailing list September 2009. It was well received, and, in the last years, two new versions have been released to address issues raised by the community.

Project coordinator: Johan Eker, johan.eker@ericsson.com

eMuCo

The Embedded Multicore Processing for Mobile Communications (eMuCo, http://www.emuco.eu) project brings a combination of multicore and virtualization to mobile phones. Although multicore combines better performance with low power consumption through parallelization, virtualization isolates the software from the underlying hardware. Thus, rather than broadly dedicating one processor to each software subsystem, virtualization allows trade-off solutions, in which you can map a given software component to a fraction of a given processor concurrently.

The eMuCo software platform, which was released as open source on 8 June 2010, is based on a small operating system kernel accompanied by various operating system components, allowing manifold usage scenarios. The application layer is a key factor in optimum deployment on multicore architectures, allowing a balance between the execution of multithreaded applications, distributed across multiple cores. From a hardware perspective, the team performed basic investigations for homogeneous and heterogeneous multicore such as the big/little multicore architecture. This architecture aims to have two fully ISA-compatible cores, with a high performance instantiation (the big core) and a low-power instantiation (the little core). It improves performance and power consumption because the respective core can be switched on and off dynamically, and operating systems and applications are shifted dynamically.

Project coordinator: Maria Elizabeth Gonzalez, elizabeth.gonzalez@rub.de

MNEMEE

The Memory Management Technology for Adaptive and Efficient Design of Embedded Systems (MNEMEE, http://www.mnemee.org) project introduces a source-to-source optimization design layer for data management in multiprocessor system-on-chip embedded systems. An integrated and easily managed toolflow automates the process of mapping applications on multicore architectures, reducing design time and enhancing performance.

The MNEMEE project introduces a supplementary source-to-source optimization design layer for data management between optimizations at the application layer and the compiler design layer. An integrated and automatic tool flow lets designers make design trade-offs and operate their systems efficiently in the huge search space. The proposed tool flow contains tools for automatic code parallelization, automatic task mapping, and dynamic memory management. The project targets the optimization of data accesses and memory storage of both

dynamically and statically allocated data and their assignment on the memory hierarchy.

The results of the MNEMEE project will be integrated into the tool flows of the industrial partners that target applications from the multimedia and network domains, such as broadband wireless applications and a low bit-rate speech coder. Target platforms include a high-performance multicore digital signal processing (DSP) device and a dual-core low-power application processor comprising an ARM and a DSP core.

Project coordinator: Stylianos Mamagkakis, mamagka@imec.be

Peppher

The Performance Portability and Programmability for Heterogeneous Many-core Architectures (Peppher, http://www.peppher.eu) project develops a methodology, framework, and guidelines for constructing software, including paths to migration of existing, parallel software that can be ported between different (possibly heterogeneous) many-core processors under preservation of specific quantitative and qualitative performance requirements.

Peppher introduces a flexible and extensible compositional metalanguage for describing and coordinating application components. This metalanguage is used to express functional and nonfunctional component properties (such as resource and performance requirements and constraints) together with abstract specifications of the underlying hardware. The metalanguage lets the Peppher framework generate component variants targeted to different types of cores, and to perform auto-tuning at a coarse-grained level. This is complemented by auto-tuned architecture and context-adaptive algorithmic libraries for further enhancing performance portability. The Peppher framework also provides handles for the Peppher runtime system to schedule components efficiently on the available hardware resources by selecting the most promising variants. Performance predictions can be automatically derived by combining the performance models that are associated with components. Additionally, Peppher is investigating hardware support for performance portability and programmability based on a simulator developed in the project.

Project coordinator: Sabri Pllana, pllana@par.univie.ac.at

Merasa

The Merasa (http://www.merasa.org) project aims to achieve a breakthrough in hardware design, hard real-time support in system software, and WCET analysis tools for embedded multicore processors. The project focuses on developing high-performance multicore processor designs for hard real-time embedded systems hand in hand with timing analysis techniques and tools to guarantee the analyzability and predictability regarding timing of every feature provided by the processor. The developed hardware/software techniques are evaluated by an industrial case study.

The three phases of the Merasa project include a design space exploration phase using a high-level multicore simulator, an architectural refinement phase based on a detailed SystemC multicore simulator, and FPGA prototyping of the developed Merasa multicore and pilot studies. The project investigates moderately scalable embedded multicore architectures between 2 and 16 cores, which are themselves multithreaded due to simultaneous multithreading (SMT) techniques. The reference platform for the project is the TriCore architecture from Infineon Technologies. The developed techniques, however, are applicable to any embedded processor.

An overview of Merasa publications and deliverables is available at http://www.merasa.org. See also "Merasa: Multicore Execution of Hard Real-Time Applications Supporting Analyzability" in this issue.

Project Coordinator: Theo Ungerer, ungerer@informatik.uni-augsburg.de

Mosart

Mosart (http://www.mosart-project.org) addresses two main challenges of prevailing architectures: the global interconnect and memory bottleneck due to a single, globally shared memory with high access times and power consumption; and the difficulties in programming heterogeneous, multicore platforms, in particular, in dynamically managing data structures in distributed memory.

Mosart aims to overcome these challenges through a multicore architecture with distributed memory organization, a network-on-chip (NoC) communication backbone, and configurable processing cores that are scaled, optimized, and customized to achieve diverse energy, performance, cost, and size requirements of different classes of applications. Mosart achieves this by providing platform support for managing abstract data structures, including middleware services and a runtime data manager for NoC-based communication infrastructure; and developing tool support for parallelizing and mapping applications on the multicore target platform and customizing the processing cores for the application. The Mosart approach allows the scaling of the platform and optimization of its constituent elements for various embedded, multimedia, and wireless communication applications. The outcome is a flexible, modular multicore on-chip platform architecture and associated exploration design methods and tools, to allow the scaling of the platform and optimization of its constituent elements for various embedded, multimedia and wireless communication applications.

Project coordinator: Bernard Candaele, Bernard.Candaele@fr.thalesgroup.com

Multicube

The Multi-objective Design Space Exploration of Multiprocessor SoC Architectures for Embedded Multimedia Applications (Multicube, http://www.multicube.eu) project aims at increasing the competitiveness of European industries by optimizing the design of embedded computing systems while reducing design time and costs. The project defined an automatic multi-objective design space exploration framework to be used at design time to find the best power/performance trade-offs while meeting system-level constraints and speeding up the exploration process.

Based on the results of the design-time multi-objective exploration, the Multicube project defined a methodology to be used at runtime to optimize the runtime allocation and scheduling of application tasks. The design exploration flow results in a Pareto-optimal set of design alternatives in terms of power/performance trade-offs. The set of operating points can then be used at runtime to decide how the system resources should be distributed over different tasks running on the

multiprocessor system on chip. The MULTICUBE design space exploration framework leverages a set of open source and proprietary exploration, modeling, and simulation tools to guarantee a wide exploitation of the MULTICUBE project results in the embedded system design community. Industrial partners STMicroelectronics and DS2 defined the design tools' requirements and then validated the tools in some industrial use cases.

Project coordinator: Cristina Silvano, silvano@elet.polimi.it

PREDATOR

PREDATOR (http://www.predator-project.eu) is concerned with safety-critical hard real-time systems, such as aeronautics and automotive systems, that require safe bounds on tasks' worst-case execution time (WCET). These bounds are derived through static analysis of the task executables. Although current approaches can derive precise bounds for single tasks even on complex processors, measures to improve average-case performance make it increasingly difficult to derive safe and precise worst-case bounds. In particular, the trend toward integration of many functions on necessarily powerful hardware poses difficulties. Multicore and MPSoC architectures provide the required high performance, but the composition of multiple components and their connections introduces additional interferences. Without due diligence in hardware, operating system, and application software design, these interferences on shared resources make customary analyses infeasible.

PREDATOR aims to reconcile the objectives of performance and predictability in embedded systems. The goal is to have sound, precise, and efficient analyses of system performance. The project touches all parts of system development, from processor design and system architecture over single-task compilation and analysis to multitask interference analysis, scheduling, task mapping, and communication, and has achieved significant results in all of these areas.

Project coordinator: Reinhard Wilhelm, wilhelm@cs.uni-saarland.de

SARC

The SARC (shorthand for Scalable Architecture, http://www.sarc-ip.org) project sought to define a systematic, scalable approach applicable to a wide range of computing systems, from tiny energy-critical embedded devices to large-scale networked data servers. Various SARC results, some presented in this special issue, show that this approach is a promising path to the design of truly scalable and programmable computer systems in the future. The proposed architectural framework and programming model provide a vehicle for building computing systems with hundreds to thousands of processing elements. SARC's advanced compilation techniques outperform existing technologies in terms of program execution speed up. The SARC memory subsystem improves perceived memory latency, reduces power, and lets programmers express and manage complex memory access patterns. The SARC modular simulation infrastructure allows affordable evaluation of large-scale computing systems. The programming model used in SARC and its runtime system improve the computational performance while advancing programmer productivity. The SARC on-chip network and switch throughputs outperformed systems with similar link bandwidth, number of ports, and silicon area. SARC remote DMA and remote queue hardware primitives improve the latency related to start-up of interprocessor communication. Finally, the ArchExplorer design space exploration framework allows for concurrent evaluation of the hardware and software design spaces.

Project coordinator: Georgi Gaydadjiev, g.n.gaydadjiev@tudelft.nl

SHAPES

SHAPES (http://www.shapes-p.org) proposes a tiled platform to address the challenges of designing multiprocessor systems on chip (MPSoCs) for both embedded systems and large-scale computers. The SHAPES platform aims to maximize the number of floating-point operations per Joule. A heterogeneous tile integrates a RISC processor and a power-efficient floating-point VLIW digital signal processor (DSP). Each tile includes a distributed network processor (DNP) for on-chip and off-chip connectivity. Eight tiles connected by the DNPs constitute the multitile SHAPES chip. The DNP also supports a 3D toroidal mesh for interchip communications, scaling SHAPES to a massive parallel processor optimized for scientific high-performance computing.

The SHAPES software design flow uses a model-driven development approach. The basic idea is to express applications as Kahn process networks, whose restricted semantics make it possible to automate the design flow. This way, efficient implementations can be automatically obtained despite specifying an application in a platform-independent manner at system-level. The DSP's native C programmability simplifies the allocation of processes toward the best target processor (RISC or VLIW DSP). The SHAPES virtual platform is an executable specification of the SHAPES hardware platform. It allows full system simulation of both a single tile and the multitile platform.

Project coordinator: Pier Stanislao Paolucci, pier.paolucci@atmel.com

VELOX

VELOX (http://www.velox-project.eu) aims to deliver seamless transactional memory systems that integrate well at all levels of the system stack. The transactional memory programming paradigm could become the approach of choice for implementing atomic operations in concurrent programming. Combining sequences of concurrent operations into atomic transactions could reduce the complexity of both programming and verification by making parts of the code appear to be sequential without needing to program fine-grained locks. Transactions remove from the programmer the burden of figuring out the interaction among concurrent operations that conflict when accessing the same locations in memory. To make transactional memory effective, these systems will need the right hardware and software support to provide scalability in terms of number of cores, code size, and complexity.

The VELOX project aims to understand how to provide such support by developing an integrated transactional memory stack. This stack spans from the system's underlying hardware to the high-end application, and consists of CPU, operating system, runtime, libraries, compilers, programming languages, and application environments. These fully integrated transactional memory systems will not only improve the understanding of transactional memory designs, but will facilitate the

European software industry's adoption of the transactional memory paradigm, making it a tool of choice for concurrent programming on multicore platforms.

Project coordinator: Osman Unsal, osman.unsal@bsc.es

2PARMA

The Parallel Paradigms and Runtime Management Techniques for Many-core Architectures (2PARMA, http://www.2parma.eu) project aims at analyzing and developing a complete software layer combined with runtime management techniques to exploit the features of future many-core processor architectures. 2PARMA leverages component-based software engineering and develops parallelism-extraction techniques to identify opportunities for parallelization. The project provides OpenCL extensions to express data parallelism for many-core computing fabrics (MCCFs). At the operating system level, logic peripherals will be defined and deployed to the MCCFs.

2PARMA uses portable bytecode representations of software to provide not only portability but also the capability to adapt applications at runtime to the available system resources. Instruction set virtualization will enable developers to tailor the application to the subset of computing resources determined by the co-existence of multiple applications on the computing platform. The project aims at developing intelligent policies to manage system resources at runtime, taking into account the quality-of-service requirements while optimizing resource usage for system-wide performance and energy goals.

Finally, 2PARMA defines an automatic methodology to provide synthetic information about the operating points for each application with respect to the available system resources. Design space exploration methodologies developed in 2PARMA also provide architectural customization to support parallel programming models, especially optimizing communication and memory mapping.

Project coordinator: Cristina Silvano, silvano@elet.polimi.it

hArtes

The Holistic Approach to Reconfigurable Real-Time Embedded Systems (hArtes, http://www.hartes.org) project aims to simplify the design of products based on heterogeneous reconfigurable embedded systems. It does this through the development of a toolchain and a methodology that provides a fast development trajectory from application coding to system design. From the application viewpoint, the complexity of future multimedia devices is becoming too big to design monolithic processing platforms. This is where hArtes's reconfigurable heterogeneous systems become vital. Starting from an existing or new application written in C, the hArtes approach results in an executable with a modified code mapped on a multicore platform, based on a general-purpose processor, a digital-signal processor (DSP), and a field-programmable gate array (FPGA). Moreover, the hArtes toolchain embeds configuration bitstreams for the system's reconfigurable components, thus providing a complete operational system that is supported at both the software and hardware levels. One of the strengths of the hArtes project is its integration with close-to-market application scenarios, mostly focusing on immersive audio and mobile video processing. Some automotive companies are interested in the project outcome. They would particularly benefit from better methodologies and tools to produce optimized real-time embedded products.

Project coordinator: Antonio Cerruto, antonio.cerruto@atmel.com