Information Society Technologies

HiPEAC info 9

COMPILATION ARCHITECTURE

appears quarterly | January 2007

Network of Excellence on High Performance Embedded Architectures and Compilers

Welcome to HiPEAC 2007 Conference

www.hipeac.net

ACACES 2007: July 15 to July 20, 2007, L'Aquila, Italy

Transactions on HiPEAC: Next Call for papers on March 1, 2007

Message from the HiPEAC coordinator

Mateo Valero
Coordinator
UPC Barcelona
mateo@ac.upc.edu

Dear colleagues,

When you read these lines, a new year will have begun. Let me start 2007 with a few words about what it is undeniably the greatest HiPEAC asset: our people. That goes from the whole Steering Committee, really committed to achieving excellence in all aspects of our network, to all the students that eagerly collaborate in clusters. Whenever I share one of our usual working sessions with you, I have the rewarding feeling that we are really forging our common progress, of which I will now give a brief update.

Last summer HiPEAC launched a new instrument open to all members, HiPEAC travel grants. Since then, 14 travel grants have been awarded. The first students were sent to their host institutions in October. The aim of this new initiative is to foster HiPEAC members' joint research and publications.

In October, the second HiPEAC Industrial Workshop took place at Philips Eindhoven, in conjunction with the HiPEAC cluster meeting and a CoSy tutorial. The next HiPEAC Industrial Workshop is scheduled for April 2007 at IBM Haifa, Israel, in parallel with a general cluster meeting. The main purpose of the industrial workshops is to let European industry be aware of research activities of the academic members on topics of their interest; thus, papers for the industrial workshops are selected by industry researchers.

On November 7, HiPEAC passed its second official European Community review. The progress achieved in engaging more industry members within HiPEAC activities was very much appreciated; we intend to continue to foster industry-academia collaboration by means of industrial workshops, internships and cluster calls. A new cluster call will be opening in January and new internship calls are expected in March 2007.

HiPEAC wants to spread by welcoming new members, especially from the Central and Eastern European countries. During PACT'07 in Bucarest we will devote a special session to present our work to make HiPEAC known and raise interest among research groups in this area.

HiPEAC participated in the IST 2006 event in Helsinki in collaboration with ARTIST2 by presenting two talks in the "Computing System" session. You can find the presentations at the IST event 2006 website.

Our next major community event is the HiPEAC Conference, which will take place January 28-30, 2007 in the medieval heart of Ghent, Belgium. The proceedings are published by Springer LNCS and the best papers will be invited for a special issue of Transactions on HiPEAC. Co-located with the conference are several workshops and tutorials, and an internship meeting where companies will present their internships for 2007. The conference will be followed by a general HiPEAC cluster meeting on January 31-February 1. Several companies have generously sponsored the event. I am sure you don't want to miss this great opportunity to meet our community.

Happy 2007!

Mateo Valero
HiPEAC Coordinator ■

Message from the project officer

During November I have been very busy with the final preparations for the IST 2006 Event. I am now happy to say all efforts have been rewarded, not only because this year's event has been the biggest such event ever organised (more that 4,500 delegates, over 250 speakers in 47 conference sessions, 160 exhibits, and 104 networking sessions) but also because, according to several independent reports, the event has been of high quality.

There in Helsinki I met a considerable number of people from the HiPEAC community and would like to thank all of them for helping to make IST2006 such a success.
If you did not participate, I would like to recommend you to reserve a few minutes of your time for visiting the following Web pages:

1. The Conference session on Computing Systems.
http://ec.europa.eu/information_society/istevent/2006/cf/document.cfm?doc_id=2500
Computing Systems is a new objective in the Workprogramme 2007/2008. We envisage the publication of the 1st FP7 Call by the end of 2007 with a closing date of the 24th of April 2007. The session included four motivating speeches and my presentation, which gave you more details on the Call.

Mercè Griera i Fisa Merce.Griera-i-Fisa@ec.europa.eu
Please take note of the European Commission's new e-mail addresses!

ACACES 2007: Third International summer school on advanced computer architecture and compilation for embedded systems

July 15 to July 20, 2007, L'Aquila, Italy
We are proud to announce the third HiPEAC Summer School, which will take place during the third week of July in L'Aquila, a small town about 100 km northeast of Rome at the TILS Campus.
We start on Sunday evening with an opening ceremony. On Monday, the 12 courses start spread over two morning slots and two afternoon slots. Per slot there are three parallel courses of which you can take one. The courses have been allocated to slots in such a way that it will be possible to create a summer school program that matches your research interests. The topics of this year's Summer School will be presented by the following world-class experts.

On Monday evening there will be an invited talk by Rudy Lauwereins, VP IMEC .
On Wednesday afternoon, the participants are given the opportunity to present their own work to the other participants during a huge poster session.
On Friday evening there is a farewell dinner and party.

Students and lecturers will be accommodated in hotel-standard private rooms on campus, where they will stay for one week. This will provide plenty of opportunity to have discussions with the teachers and with the other participants in the relaxing surroundings of the TILS Campus. Long after-the-lecture discussions at the bar or the pool table are one of the major assets of this summer school. At the end of the event you will receive a certificate of attendance detailing the courses you took.

You can arrange to be picked up in Rome on Sunday July 15, 2007 and to be taken back on Saturday July 21, 2007 to either Rome airport or downtown Rome (for those who want to spend some extra time in the eternal city).

Unfortunately, the number of partipicants will be limited. Therefore, we have an admission procedure to guarantee a fair distribution of the available places among all qualified applicants from the various countries and institutions. If you are a member of a HiPEAC institution you can ask for a grant that covers the registration fee.

In this newsletter, you will find a summer school poster. Please post is at some visible place in your department. You can find more information about the summer school at http://www.hipeac.net/summerschool.

We look forward to seeing you there!

Koen De Bosschere
Summer school organizer ■

Pradip Bose	IBM, USA	Power-efficient, reliable micro-architectures
Jack Davidson	University of Virginia, USA	Compilation Techniques for Embedded Systems
Michel Dubois	University of Southern California, USA	Coherence, Store Atomicity and Memory Consistency Models
Paolo Faraboschi	Hewlett Packard Laboratories, Spain	Embedded VLIW Architectures and Compilers
Manolis Katevenis	FORTH and Univ. of Crete, Greece	Queueing and Flow Control in Interconnection Switches
Peter Marwedel	University of Dortmund, Germany	Memory architecture aware compilation
Kathryn McKinley	University of Texas at Austin, USA	Optimizing and Measuring Managed Languages
Kunle Olukotun	Stanford University, USA	Chip Multiprocessor (CMP) Architectures
Lawrence Rauchwerger	Texas A&M University, USA	Parallel Programming Models
Barbara Ryder	Rutgers University, USA	Advanced Program Analyses for Object-oriented Systems
Guri Sohi	University of Wisconsin-Madison, USA	Memory Dependence Prediction and its Applications
Hans van Someren	ACE Associated Compiler Experts, The Netherlands	Constructing Compiler Components for Reusability

2. The Conference session on How to Submit a Proposal
http://ec.europa.eu/information_society/istevent/2006/cf/conference-detail.cfm?id=1047
The session includes four presentations covering the rules for participation, how to submit a proposal and the selection procedures. This information is important because the next call will be the first in FP7 and rules and procedures are not exactly the same as for FP6.

3. The Networking Session on Multi-core Processing and ARTEMIS
http://ec.europa.eu/information_society/istevent/2006/cf/network-detail.cfm?id=913
The goal of this community is to build a research framework that actually renders MPSOC platforms transparent (platform abstraction), enabling application and software developers to efficiently and optimally plan MPSOC designs. Their research covers two objectives of the next call: the objective on Computing Systems and the objective on Embedded Systems Design.

The next gathering of the HiPEAC community is the Second HiPEAC conference in Ghent. If you have any questions related to the Call content and on the new rules and procedures, do not hesitate to approach me.
Looking forward to seeing you in Ghent!
Mercè Griera-I-Fisa ■

France

HiPEAC France consists of 2 institutions: CNRS and INRIA.

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

INRIA FUTURS

The Centre National de la Recherche Scientifique (CNRS) is a government-funded research organization. As the largest organization supporting fundamental research in Europe, CNRS is involved, mostly through joint labs with universities, in all scientific fields including Compiler and Architecture research.

Two CNRS labs participate in HiPEAC activities:

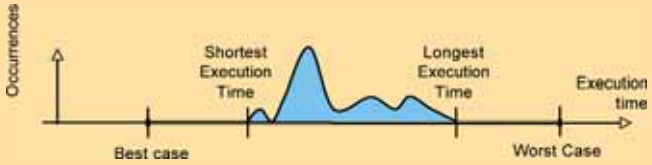
- IRIT, associated with the University of Toulouse.
- LIP6, associated with the University of Paris 6.

Université Paul Sabatier TOULOUSE III

UNIVERSITE PIERRE & MARIE CURIE

Traces is an IRIT Research group on Architecture and Compilation for Embedded Systems. The research interests of the TRACES team of the IRIT lab include hardware issues for real-time embedded systems. The main goal is to guarantee that the execution time of an application code will meet the system deadlines. Research topics include:

- characterizing the temporal properties of off-the-shelves components. This collaboration with Airbus targets solutions to use these components in such a way that safe and tight WCET (Worst-Case Execution Time) estimates can be computed.
- exploring new architectures that reconcile high-performance with predictability.
- exploring tools for modeling of modern processor architectures and for WCET calculus.



The execution time of an application code depends on the input data. Unfortunately, the input data domain is generally too large to be exhaustively explored. This is why methods have been proposed over the last few years to compute the Worst-Case Execution Time from time measures of short code segments (e.g. basic blocks). The estimation of the WCET requires three steps. First, a static analysis of the code identifies all the possible execution paths. Second, the target hardware is modeled to determine the individual execution times of the basic blocks taking into account every possible history. Finally, the information on control flow and the partial execution times are combined to evaluate an upper bound of the global execution time. Research in the TRACES team includes modelling processors for WCET estimation, and in particular new architectures like the multi-threaded processor CarSOC developed by Theo Ungerer's team in Augsburg.

People:
Hugues Cassé
Marianne De Michiel
Djemai Kebbal
Christine Rochange
Pascal Sainrat (leader)

Device Under Test at IRIT

The **SOC** (System On-Chip) department of the LIP6, Laboratory of Computer Sciences, University Paris 6, focuses on the design of highly complex integrated circuits, the development of advanced CAD tools for VLSI, and hardware architecture and software for embedded systems.

In the context of software for embedded systems, the SOC department focuses on iterative compilation techniques. More precisely, it explores the possibilities of integrating dynamic information (information from code execution) into the optimization process.
Currently, most compiler optimization techniques rely on a machine model embedded in the compiler. However, as hardware complexity increases, embedding an accurate and reliable machine model in a compiler is increasingly difficult, which, in turn, limits the efficiency of purely static optimization techniques. For that reason, dynamic or dynamically-assisted compiler optimizations have received increased attention in the past few years. The principle is to use run-time information to fine-tune static program transformation parameters and thus to compensate for the lack of accuracy of the embedded machine model.

In the SOC department, this approach is considered in different works:
- selection of compiler optimizations for performance,
- selection of regions to be compressed,
- extraction of instruction parallelism for VLIW (Very Long Instruction) architectures,
- ...etc.

For these works, a major concern is the performance sensitivity to the data used to profile the application. Thus, the problems of sensitivity to the input data sets and of their representativeness are addressed.

People:
Nathalie Drach (leader)
Karine Heydemann

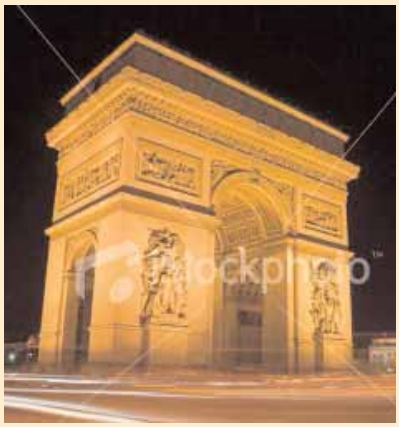
INRIA, which stands for Institut National de Recherche en Informatique et Automatique (National Research Institute in Computer Science and Control) is the leading French institution dedicated to computer science, with more than 800 faculty distributed among 138 research projects. Approximately 1000 PhDs are currently advised at INRIA. INRIA researchers benefit from permanent full-time research positions.

The institute addresses both theoretical and applied computer science research, and has spun off more than 80 companies since it was founded in 1984. It is a decentralized organization distributed across 6 research units in France:
Paris/Rocquencourt, Rennes, Grenoble/Lyon, Nice/Sophia, Nancy, and the tri-located Futurs unit at Paris/Saclay, Lille, Bordeaux.

Three INRIA units are involved in HiPEAC:
- ALCHEMY at Futurs/Saclay,
- CAPS at Rennes, and
- COMPSYS at Lyon.

Alchemy is a joint INRIA/University of Paris Sud research group.
The general research topics of the Alchemy group are architectures, languages and compilers for high-performance embedded and general-purpose processors.
Alchemy investigates scalable architecture and compiler/programming solutions for high-performance general-purpose and embedded processors.
Alchemy stands for:
Architectures, Languages and Compilers to Harness the End of Moore Years, referring to both traditional processor architectures implemented using the current photo-litho-graphic processes, and novel architecture/language paradigms compatible with future and alternative technologies. The current emphasis of Alchemy is on the former part, but we are progressively increasing our efforts for the latter part.
The research goals of Alchemy span from short term to long term. The short-term goals target existing complex processor architectures, and thus focus on improving program performance on these architectures (software-only techniques). The medium-term goals target the upcoming CMPs (Chip Multi-Processors) with a large number of cores, which will result from the now slower progression of core clock frequency due to technological limitations. The main challenge is to take advantage of the large number of cores for a wide range of applications, considering that automatic parallelization techniques have not yet proved to be an adequate solution.
In Alchemy, we explore joint architecture/programming paradigms as a pragmatic alternative solution. Finally, even longer term research is conducted with the goal of harnessing the properties of future and alternative technologies for processing purposes.
Most of the research in Alchemy attempts to jointly consider the hardware and software aspects, based on the premise that many of the limitations of existing architecture and compiler techniques stem from the lack of cooperation between architects and compiler designers. However, Alchemy addresses the aforementioned research goals through two different, though sometimes complementary, approaches.
One approach considers that, in spite of their complexity, architectures and programs can still be accurately and efficiently modelled (and optimized) using analytical methods.
The second approach considers that the architecture/program pair already has or will reach a complexity level that will evade analytical methods, and explores a complex systems approach; the principle is to accept that the architecture/program pair is more easily understood (and thus optimized) based on its observed behaviour rather than inferred from its known design.

People:
Cédric Bastoul
Hugues Berry
Albert Cohen
Christine Eisenbeis
Frédéric Gruau
Olivier Temam (leader)

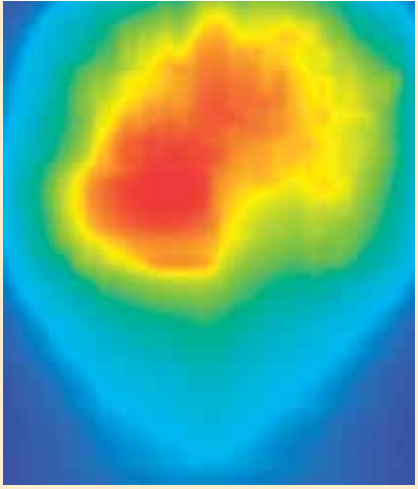


CAPS is an Irisa/Inria project-team, located in Rennes. IRISA is a joint lab for computer science and applications from CNRS, Inria, University of Rennes I and INSA.
CAPS is four permanent researchers from Inria and the University of Rennes I.
Research in the CAPS project-team ranges from processor architecture to software plat-forms for performance tuning, including compiler/architecture interactions, processor simulation techniques and worst case execution time (WCET) evaluation techniques.
Peak performance is one of the objectives, however, finding tradeoffs between hardware complexity and performance, and performance and power consumption (or code size) is also a major issue, while accurately evaluating (more precisely bounding) the execution time is the challenge for embedded real time systems.

The research undertaken at CAPS in computer architecture covers memory hierarchy, branch prediction, and superscalar implementation, as well as SMT and multi-core processors. CAPS researchers have proposed several new complexity-effective structures for caches and branch predictors. CAPS also pursues research on architecture exploiting thread-level parallelism on a single chip (multi-core, SMT) as well as research on temperature management at the architectural level.
Performance, but also power consumption or hardware system cost, depends on the processor architecture but can also be managed at the compiler/code generation level. For instance, code size is often an issue with embedded systems.
CAPS is exploring tradeoffs leveraging code compression and interpretation.

For heterogeneous SOCs (Systems On a Chip) featuring special purpose hardware and one or more execution cores, CAPS is exploring speculative thread extraction for the different hardware components.

In hard real-time embedded systems, the task WCETs must be correctly evaluated, so that it can be proven that the task's temporal constraints (typically, deadlines) will be met. CAPS researchers also study WCET-oriented (as opposed to average-performance oriented) compilation and measurement-based WCET estimation.
Some of the research prototypes developed by the project-team during the past few years have been transferred to industry through the CAPS Entreprise start-up.



Microprocessor temperature map obtained with a thermal model developed by the CAPS team

People:
François Bodin
Pierre Michaud
Isabelle Puaut
André Seznec (leader)

Compsys is an Inria Rhône-Alpes project, located in Lyon, at École normale supérieure de Lyon and, more precisely, is part of the Computer Science Laboratory for the study of Parallelism (LIP). The name of the lab dates back to the early developments of parallel computers, but now the lab focuses on various aspects, including complexity theory, automatic deduction, grid computing, compilation, computer arithmetic and high-performance networks. Compsys is currently composed of five permanent researchers (from different institutions: CNRS, ENS-Lyon, Inria, and Insa-Lyon), three post-docs, and five PhD students.
The Compsys project (Compilation and Embedded Computing Systems) was created by researchers previously working in automatic parallelization and high-level program transformations for the synthesis of systolic arrays. Some of the optimization techniques developed at this time (in the '90s) seemed to be frequently rediscovered, or at least applied in different forms, to the automatic design of embedded computing systems. The objective of Compsys was to first learn the requirements for high-level synthesis of systems and circuits, so as to try to adapt and extend some code optimization techniques, primarily designed for compilers/parallelizers for high-performance computing, to the special case of embedded computing systems. In particular, today, Compsys works on micro-code optimizations for specialized processors and on high-level synthesis of hardware accelerators.
The main characteristic of Compsys is to focus on combinatorial optimization problems (graph algorithms, linear programming, polyhedra) coming from code optimization problems arising in this field (register allocation, instruction cache optimization, memory allocation, scheduling, optimizations for power, automatic generation of software/hardware interface, etc.) and to validate the techniques developed by Compsys.
The main new achievements and current developments of Compsys are:
- The study of a new approach for register allocation, in two steps, with a spilling phase (optimization of loads and stores) followed by an assignment phase including register coalescing (optimization of register-to-register moves); complexity study, developments of heuristics and algorithms, integration within the STMicroelectronics compiler.
- The introduction of new mathematical tools (related to critical lattices) to optimize the memory reuse for multi-dimensional arrays; theoretical and software developments (software tool Cl@k) with integration in a compiler for source-to-source program transformations.
- The development of methodologies for Systems-on-Chip: prototypes of high-level synthesis tools, developments of software/hardware interfaces, in particular for the SocLib platform, traffic analysis, power consumption analysis for embedded operating systems, ...

People:
Alain Darte (leader)
Paul Feautrier
Antoine Fraboulet
Fabrice Rastello
Tanguy Risset

ARM: Research challenges at the heart of advanced digital products

ARM designs the technology that lies at the heart of advanced digital products, from wireless, networking and consumer entertainment solutions to imaging, automotive, security and storage devices. ARM's comprehensive product offering includes 16/32-bit RISC microprocessors, data engines, 3D processors, digital libraries, embedded memories, peripherals, and software and development tools, as well as analog functions and high-speed connectivity products.

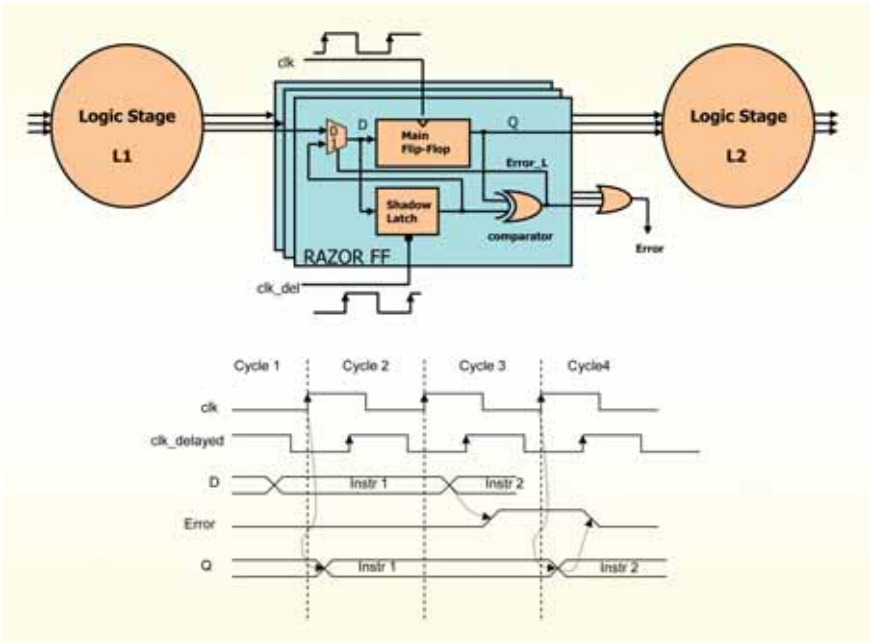
Low power computing: Razor

An old adage says, "If you're not failing some of the time, you're not trying hard enough." To address the power challenges that current on-chip densities pose, Razor adapts this precept to circuit design.

With increasing clock frequency and silicon integration achievable through aggressive process scaling, designers find it increasingly difficult to meet the dual and often conflicting constraints required to engineer low power and yet robust circuits. Traditionally, microprocessors are designed to be always correct even under the worst-case combination of silicon grade and ambient conditions. Safety margins are added during design time to ensure worst-case computational correctness. However, with increased process variability and design uncertainties, these safety margins often lead to overly conservative designs and contribute significantly to the overall power consumption of a processor.

Razor is a novel design methodology that eliminates the safety margins and uses self-checking mechanisms to ensure computational correctness in the presence of errors. Thus, it represents a radical departure from the traditional design paradigm of "worst-case" and "always-correct" to "typical-case" and "usually-correct". Using in situ error detection and correction mechanisms, a "Razor"-ed pipeline is able to dynamically adjust its supply voltage or frequency to operate at the exact point of failure. In addition, it can also tune its system parameters to operate below this point of failure so as to achieve a targeted error-rate. In the Razor prototype that was built in ACAL, we managed to achieve 50% energy savings on average over the worst-case at 120MHz, by operating at a 0.1% error rate.

ARM is closely collaborating with Michigan University in this project.



A razor flip-flop for a pipeline stage and its operation with a timing error in cycle 2 and recovery in cycle 4

ARM R&D

ARM's R&D team performs research into generating innovative technology 3 to 10 years ahead of the products.

ARM's R&D team is a dynamic, growing team with world-class expertise in the fields of energy efficient 16/32-bit RISC microprocessors, data engines, software and development tools, digital libraries, analog functions, and high-speed connectivity. They have a successful record of creating technologies which are now a key part of the ARM architecture (NEON, TrustZone, Thumb-2, Jazelle) as well products now available from ARM (IEM). The group has a wide range of activities ongoing; a lot of these activities are carried out in close collaboration with academic institutions in Europe as well as in the US.

Transparent Specialization

On another front, the embedded device industry is witnessing the growing challenge of providing multi-purpose high performance cores that support an unprecedented diversity in applications while maintaining increasingly stringent cost, low-power and time-to-market constraints.

Traditionally, application specific integrated circuits (ASICs) are utilized to do the heavy lifting in system-on-chip designs, where critical portions of applications are mapped directly to hardware implementations.

With the recent "convergence of embedded products", like in the most recent smart phones for example, it is now more and more difficult to support specialized circuitry for every single class of applications.

Transparent customization addresses this problem by defining fine grain programmable accelerators which execute in parallel with (or replace) the ALU. These fine grain accelerators execute acyclic data flow graphs more efficiently than traditional ALUs thanks to the collapsing of dependent instructions. Furthermore, the use of such accelerators is transparent in the sense that we do not need to add special instructions to support the accelerator. Instead, the hardware dynamically extracts subgraphs from the existing ISA that can be mapped to the accelerator and programs it accordingly. This transparency assures portability of the binaries across

several generations and implementations of accelerators.

Challenges: However, for transparent specialization to be successful, some major challenges remain. The first challenge is to find the appropriate granularity of the programmable accelerator. On the one hand, a small accelerator that can map relatively small dataflow subgraphs will be able to map a large number of subgraphs at a relatively low cost, but will have a lower impact on performance. On the other hand, a larger accelerator will have a greater impact on performance provided we find sufficiently large subgraphs that can efficiently exploit its resources. Finding the right trade-off along with finding recurring patterns of instructions are still important research challenges.

ARM has been actively looking at these challenges and collaborating with other industrials and institutions from the EU through HiPEAC. Many discussions and networking events have evolved to collaborative activities in this area, like the research cluster on "Identification and Specialization of Frequent Patterns of Computations in Programs" in which ARM has closely collaborated with the University of Cyprus, and INRIA. ARM has been also closely working on this topic with the University of Michigan.

Parallel architectures

For multi-processing in embedded systems, time-slicing one processor has usually been sufficient in the past, but higher-bandwidth data and processing requirements on the datastream and more-complex human interfaces are pushing uniprocessor solutions to their limit, without going to multi-gigahertz clock speeds. ARM has recently released its first symmetric multiprocessing core, the ARM MPCore. MPCore supports from one to four ARMv6-compliant processors combined with multi-CPU optimized cache-coherency logic and interrupt controller.

Unlike multicores in PCs or servers, one challenge embedded systems need to consider is how to program applications that exploit multicores as well as other IP blocks in the SOC such as video processing or radio processing IP blocks. While the number of cores on which the application executes is not necessarily large, the challenge resides in how to partition the application over the several cores and IP blocks so that the application executes the most efficiently in term of performance and energy consumption. Dynamically adapting parallelism according to existing resources and exploring parallel programming paradigms is one of the areas ARM is investigating through close collaboration with INRIA-Futurs, a member of the HiPEAC consortium.

In conclusion, the challenges ahead of us in embedded systems are no longer the "eight-year old challenges of high performance architectures". Energy constraints as well as the convergence of embedded products requiring high performance processing gives rise to an exciting set of challenging problems and research issues.

Krisztian Flautner, Sami Yehia

■

People

Krisztián Flautner is the director of research at ARM Ltd. His research interests include high-performance, low power processing platforms to support advanced software environments. Flautner received a PhD in computer science and engineering from the University of Michigan. He is a member of the ACM and the IEEE.
krisztian.flautner@arm.com

Sami Yehia obtained his Master and PhD in Computer Architecture from University of Paris-Sud, France in 2000 and 2004 respectively. He joined ARM Ltd as a senior research engineer in December 2004. His research interests include computer architecture, instruction customization and hardware specialization.
sami.yehia@arm.com

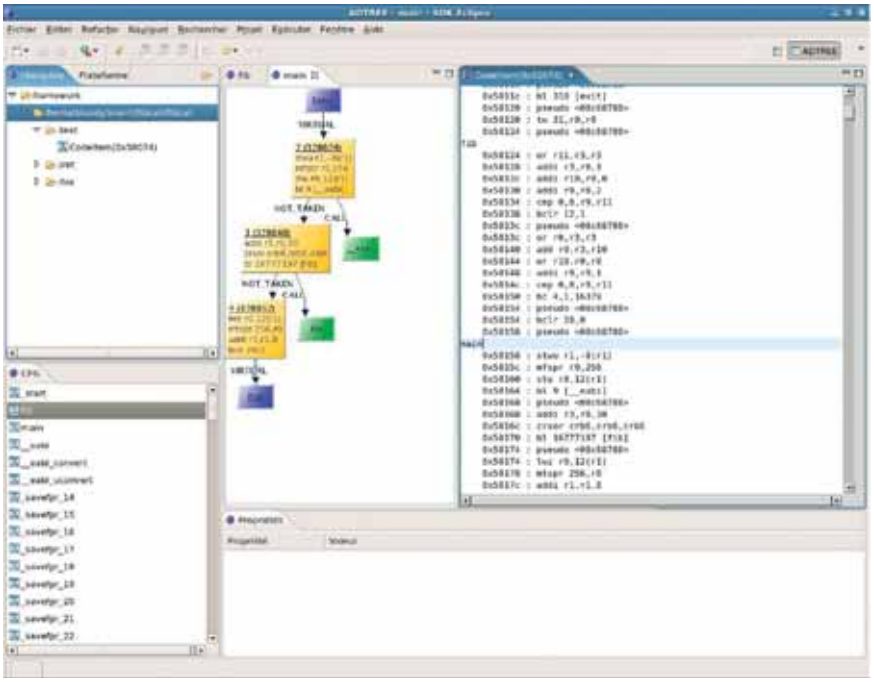
ARM History

ARM was established in November 1990 as Advanced RISC Machines Ltd., a UK-based joint venture between Apple Computer, Acorn Computer Group and VLSI Technology. Apple and VLSI both provided funding, while Acorn supplied the technology and the 12 founding ARM engineers. Acorn, developer of the world's first commercial single-chip RISC processor, and Apple, intent on advancing the use of RISC technology in its own systems, chartered ARM with creating a new processor standard. ARM immediately differentiated itself in the market by creating the first low-cost RISC architecture. Conversely, competing architectures, which were more commonly focused on maximising performance, were first used in high-end workstations.

With the introduction of its first embedded RISC processor, the ARM6™ family of processors, in 1991, ARM signed VLSI as its initial licensee. One year later, Sharp and GEC Plessey entered into licensing agreements, with Texas Instruments and Cirrus Logic following suit in 1993. Over the years, ARM has significantly expanded both its IP portfolio and its Partner base. After the 1993 addition of Nippon Investment and Finance (NIF) as a shareholder, the company began establishing a global presence, opening new offices in Asia, the US and Europe. In April 1998, the company listed on the London Stock Exchange and Nasdaq. More recently, in December 2004, ARM acquired Artisan Components.

ARM is now a global corporation with more than 1,250 employees and facilities in 12 countries on three continents, with design centers in: Blackburn, Cambridge and Sheffield in the UK; Sophia Antipolis in France; Bangalore in India; Sunnyvale, San Diego and Walnut Creek in California; Cary in North Carolina and Austin in Texas. The company also maintains sales, administrative and support offices in Belgium, China, France, Germany, Israel, Japan, Korea, Taiwan, Singapore, the UK and the US.

OTAWA: a framework for a fast implementation of program analyses and WCET techniques.



For many years, the compilation and architecture domains have seen the development of generic and experimental frameworks as soon as a modelling problem has become mature enough. For example, we can quote SUIF [1], Salto [2] and many more. These frameworks, even if they leave room for improvement, have sped up the development of new techniques making the re-use of existing algorithms easier.

We think that Worst-Case Execution Time computation techniques have reached this point [3] and OTAWA – Open Tool for Adaptative WCET Analysis – will provide an analogous platform. Similar to other frameworks, it includes criteria ensuring a large range of use: multi-architecture support, genericity, openness, re-usability, extensibility. At the same time, OTAWA has been designed using the concepts and the experience provided by existing frameworks in order to avoid a maximum number of design pitfalls.

OTAWA comes from the need for open and generic software usable in our team: (1) a tool that may be used by our team to develop new algorithms or new analyses in the WCET area and (2) a software platform allowing implementation of an adaptive experimental approach for WCET computation. As this last task was requiring a lot of development, we retargeted the tool to become as generic as possible in order to maximize the benefits of the development effort. This led to the current implementation of OTAWA.

The running system includes an architecture abstraction layer that is heavily based on an object-oriented approach. An annotation layer plays a central role for maximizing re-use and combination of code analyses: annotations are typed pieces of information that are hooked to the code. They are processed by code analyses to compute the WCET or other program information: each analysis may in turn use annotations built by other analyses and produce new annotations that are inserted in to the code and made available to following analyses. As the analyses performed in WCET computation are typically designed to work with a partial knowledge of the program, it is easy to combine them: the precedence requirements between analyses are minimal.

OTAWA is currently developed by the TRACES team at IRIT, France. The core components are fully working and include support for the PowerPC architecture with a binary loader and a generic cycle-level simulator. We will soon add support for processing ARM and S12X binaries to prove the modularity of our architecture. OTAWA already includes high-level structured program representations like Control Flow Graphs, Context Trees, Abstract Syntax Trees, dominance trees or generic facilities to perform analyses like flow fact insertion, applying Data Flow Analysis engines and so on. Although OTAWA is not dedicated to a specific method of WCET computation, it provides some basic support for methods based on the Implicit Path Enumeration Technique and Extended Timing Schema.

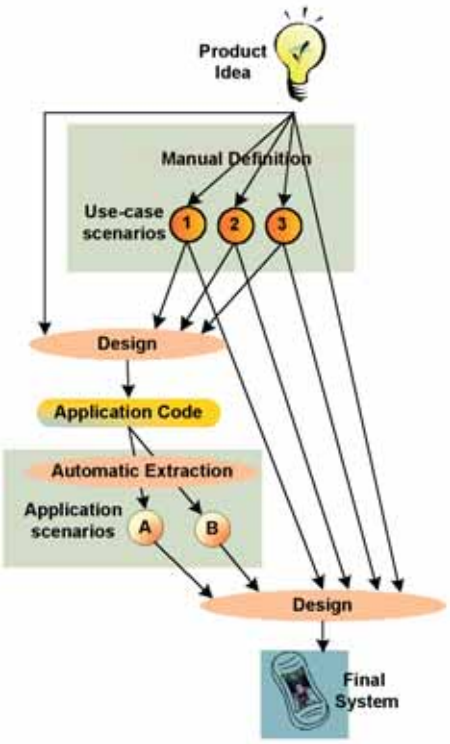
Finally, we have used OTAWA for managing the instruction cache in WCET computation in two projects. In the first, we compare two implementations in the IPET method, the Conflict Flow Graph [4] and categories [5]. In the second project, we have adapted the categories approach to Extended Timing Schema. These works have first validated our view of the process of computing WCET and second proved the usability of OTAWA.

OTAWA has now reached a milestone in its development making it usable outwith the team environment. Sources will be soon distributed freely on http://www.irit.fr/

Contact information: Pascal Sainrat (pascal.sainrat@irit.fr)

[1] R.P. Wilson, R.S. French, C.S. Wilson, S.P. Amarasinghe, J.M. Anderson, S.W.K. Tijang, S.-W.] Liao, C.-W. Tseng, M.W. Hall, M.S. Lam, J.L. Hennessy. SUIF: An Infrastructure for Research on Parallelizing and Optimizing Compilers. ACM SIGPLAN Notices, V29, N12, 1994.
[2] R. Rohou, F. Bodin, A. Seznec, G. Le Fol, F. Charot, F. Raimbault. Salto : system for assembly-language transformation and optimization. Technical Report RR-2980, INRIA, 1996.
[3] J. Engblom, A. Ermedahl, M. Sjodin, J. Gustafsson, and H. Hansson. Towards industry-strength worst case execution time analysis. Technical Report ASTEC 99/02, Advanced Software Technology Center (ASTEC), April 1999.
[4] Y.-T. S. Li, S. Malik, A. Wolfe, Cache Modeling for Real-Time Software: Beyond Direct Mapped Instruction Caches, Proceedings of ACM SIGPLAN Workshop on Language, Compiler and Tool Support for Real-time Systems, June 1997.
[5] C.A. Healy, R.D. Arnold, F. Mueller, D.B. Whalley, M.G. Harmon, M. G, Bounding Pipeline and Instruction Cache Performance, IEEE Transactions on Computers, January 1999, vol. 48.1.

Joint Research in Scenario Based Embedded System Design



Scenario based design has been used for a long time in different areas, including embedded system design. These scenarios describe, in an early phase of the development process, the use of a future system. They appear as narrative descriptions of envisioned usage episodes, or as unified modeling language (UML) use-case diagrams. Moreover, these scenarios, also called use-case scenarios, focus on the application's functional and timing behaviors and on its interaction with the users and the environment, not on the resources required by an application to meet its constraints.

In this research we concentrate on a different type of scenario, so called application scenarios, which may be derived from the possible run-time behavior of the application. Exploiting these scenarios may reduce the system cost by enabling better design decisions. While use-case scenarios classify the application's behavior based on the different ways it can be used, application scenarios classify it based on cost aspects.

The figure depicts a design trajectory using use-case and application scenarios. It starts from a product idea, for which the stakeholders define the product's functionality as use-case scenarios. These scenarios characterize the system from a user perspective and they are used in a user-centric development process to design both the software and hardware components. In order to optimize the design of the system, the extraction and exploitation of application scenarios augments this trajectory (the bottom gray box within the figure). The run-time behavior of the application can be automatically classified from a cost perspective into several application scenarios, where the cost within a scenario is always fairly similar and between scenarios it is fairly different. For each individual scenario, more specific and aggressive design decisions can be made.

Example Let us consider that we want to design a small portable MP3 player as a USB stick. At first sight, there are at least two main use-case scenarios: (i) the player is connected to the computer and music files are transferred between them, and (ii) the player is used to play music. These scenarios can be divided in more detailed use-case scenarios.

For example, for the second one, we may have the song selection, playing, or fast forward scenarios. Let us consider the playing use-case scenario. From the software point of view, this use-case can be split in two different application scenarios: playing songs (i) in mono mode and (ii) in stereo mode. If these scenarios are detected in the application, and the information about them is used during the system design, the system battery lifetime may be increased: in case of playing in mono mode less computational power is needed, thus a lower supply voltage may be used to meet the timing constraints of the decoding.

Our methodology based on application scenarios tackles the following four issues:

- Identification: given the application, how is it classified into scenarios?
- Detection: given a particular run-time situation, to which scenario does it belong?
- Exploitation: given a particular scenario, what can be done to optimize the system cost?
- Switching: when and how does the application switch from one scenario to another?

For more details about our work, please check our website: http://www.es.ele.tue.nl/scenarios.

Current research partners are: Eindhoven University of Technology, IMEC Leuven, and Ghent University. Other interested parties are welcome to join us.

Contact information: Valentin Gheorghita (s.v.gheorghita@tue.nl),
Technical University Eindhoven ■

Community news

Technical University of Valencia Joins HyperTransport Consortium

The HyperTransport Technology Consortium, a non-profit organization dedicated to developing, promoting and licensing the industry's lowest latency, highest bandwidth interconnect technology, has announced the addition of the Technical University of Valencia (TUV) to its academic membership roster. TUV brings extensive knowledge and background in interconnect technology to the Consortium. The HyperTransport Consortium is working with the Technical University of Valencia's Parallel Architectures Group, a leading research group that develops advanced applied research in the areas of interconnection networks, multiprocessor architectures and processor microarchitecture.

Mateo Valero was awarded the 'Premio Nacional de Investigación 2006 Leonardo Torres Quevedo'

For his career in the domain of computer architecture, and more in particular for his fundamental contributions to vector computing.

Per Stenstrom was nominated IEEE Fellow for contributions to design of high-performance memory systems.

Buffer Management Strategies to Reduce HoL Blocking

By Teresa Nachiondo (tnachion@ disca.upv.es)
Dr. Jose Flich
Prof. Jose Duato
Universidad Politécnica de Valencia
July 21, 2006

Head-of-line (HoL) blocking is one of the main problems arising in high-speed switches, due to the use of FIFO queues. Virtual Output Queues (VOQs) is the best-known solution to the HoL blocking problem. However, the cost of implementing VOQs increases quadratically with the number of output ports in the switch fabric, thus this solution leads to very high costs and it is not scalable at all. In this thesis we present a new family of methodologies, to reduce the HoL blocking effect on interconnection networks using efficiently the resources (mainly memory queues) of the network.

The TM3270 Media-processor

By Jan-Willem van der Waerdt (jan-willem.van_de_waerdt@nxp.com)
Prof. Georgi Gaydadjiev
Delft University of Technology
October 10, 2006

This thesis presents several enhancements to the TM3270 VLIW media-processor: a new data prefetching technique, and architectural enhancements, such as additions to the TriMedia Instruction Set Architecture (ISA). Examples of ISA enhancements include collapsed load operations, two-slot operations and H.264 specific CABAC decoding operations. The speedup of individual ISA enhancements are measured in terms of both static (VLIW instruction count) and dynamic (processor cycle count) performance complexity, and both at the level of individual kernels and complete video applications. Combined, the TM3270 enhancements result in speedups of more than a factor two, for the evaluated video applications.

Design of a Scalable and Efficient Congestion Management Technique for Interconnection Networks

By Pedro Garcia (pgarcia@info-ab.uclm.es)
Prof. Francisco Quiles
Prof. Jose Duato
Universidad de Castilla La Mancha
November 20, 2006

RECN (Regional Explicit Congestion Notification) is a new congestion management technique for interconnection networks. RECN is based on accurately detecting the origin of any congestion tree appearing in the network, in order to identify which packets contribute to congestion. RECN uses a small number of queues that are dynamically assigned for storing only congested packets, separating so them from the non-congested ones. In this way, RECN avoids the appearance of HOL blocking and is able to keep network performance at maximum, even in congestion situations.

The SSA Representation Framework: Semantics, Analyses and GCC Implementation

By Sebastian Pop (sebpop@gmail.com)
Albert Cohen
Francois Irigoin
Georges-Andre Silber
INRIA Futurs
December 13, 2006

The Static Single Assignment (SSA) language is one of the intermediate representations commonly used in industrial compilers. However, there was little interest from the static program analysis community in this intermediate representation due to the weak formal grounds of the SSA. The thesis presents a denotational semantics of the SSA language, allowing formal definitions of static analyses on the SSA language based on the classical abstract interpretation framework. From a practical point of view, we present the implementation of the formally described static analyses on the SSA in an industrial compiler: the GNU Compiler Collection (GCC).

Upcoming events

Tenth Workshop on Computer Architecture Evaluation using Commercial Workloads (CAECW-10)
Phoenix, Arizona, February 11, 2007, http://trappist.elis.ugent.be/~leeckhou/caecw10/

HPCA-13, 13th International Symposium on High-Performance Computer Architecture
Hyatt Regency Phoenix – Phoenix, Arizona, February 10-14, 2007, http://www.ece.arizona.edu/~hpca/

ODES-5, Workshop on Optimizations for DSP and Embedded Systems
San Jose, California, March 11, 2007, http://www.ece.vill.edu/~deepu/odes/odes.html

CGO-2007, 2007 International Symposium on Code Generation and Optimization
San Jose, California, March 11-14, 2007, http://www. cgo.org/

CC, International Conference on Compiler Construction
Braga, PortugalMarch, March 24 - April 1, 2007, http://cc2007.cs.brown.edu/

ISPASS-2007, 2007 IEEE International Symposium on Performance Analysis of Systems and Software
San Jose, California, USA, April 8-10, 2007, http://ispass.org/

DATE, Design, Automation and Test in Europe
Nice, France, April 16-20, 2007, http://www.date-conference.com/

ACM International Conference on Computing Frontiers
Ischia, Italy, May 7-9, 2007, http://www.computingfrontiers.org/

PLDI 2007, Programming Language Design and Implementation
San Diego, CA, USA, June 10-13, 2007, http://ties.ucsd.edu/PLDI/

ISCA 2007, The 34th International Symposium on Computer Architecture
San Diego, CA, USA, June 9-13, 2007, http://www.cse.ucsd.edu/isca2007/

LCTES 2007, Conference on Languages, Compilers, and Tools for Embedded Systems
San Diego, CA, USA, June 10-13, 2007, http://www.cs.purdue.edu/lctes07/

Workshop on Experimental Computer Science,
San Diego, CA, June 13-14, 2007, http://www.cs.huji.ac.il/~feit/exp/

ACACES 2007, Third HiPEAC Summer School
L'Aquila, Italy, July 15-20, 2007, http://www.hipeac.net/summerschool

HiPEAC 2008 Conference
Kiruna/Jukkasjârvi, Sweden, January 27-29, 2008

Contributions

If you are a HiPEAC member and you want to contribute to this newsletter, please contact Thomas Van Parys at **Thomas.VanParys@HiPEAC.net**

HiPEAC Info is a quarterly newsletter published by the HiPEAC network of excellence.
Funded by the 6th European Framework Programme (FP6), under contract no. IST-004408.
Website : http://www.hipeac.net.
Subscriptions: http://www.hipeac.net/newsletter

in the next issue:
presentation of HiPEAC The Netherlands