Journal of Embedded Computing 2 (2006) 1–3
IOS Press

Embedded processors and systems: Architectural issues and solutions for emerging applications

S. Bartolini[a,1], P. Foglia[b,*,1] and C.A. Prete[b,1]
[a]*Department of Information Engineering, University of Siena, Siena, Italy*
[b]*Department of Information Engineering, University of Pisa, Pisa, Italy*

The design of embedded systems concerns hardware, software and mixed hardware-software issues. The increasing demand of complex consumer applications, such as the management of multimedia content and value-added services provided via mobile devices [1], is pushing towards the development of new architectures, operating systems and development environments. Among the most promising architectures, there are heterogeneous systems on chip. In a typical design, such systems integrate microprocessor(s)/microcontroller(s) and DSP cores. In them, microprocessor(s)/microcontroller(s) are devoted to the execution of the control, coordination and interfacing parts of the application, DSP are devoted to the support of numerically intensive tasks. Examples of this tendency can be verified in some commercial products such as the Texas-Instrument OMAP or the Atmel Diopsis families, comprising integrated ARM/DSP cores [2,3].

This scenario requires studies on the chip architecture (e.g. how many units, which kind of units, in which way they communicate, how the chips are interconnected: these are some of the open questions [6–8]) and on the paradigms for synchronization/communication at hardware and software level. In addition, advances in development tools for the management and exploitation of the cross-interaction among cores are highly desirable, starting from the system design phase up to the compilation, linking, run-time loading and tuning phases. Without such integrated approach, designers may be forced to make designing and optimizing choices for the different cores/modules singularly. This fact can cause an inefficient overall system even if it originates from the union of optimal subsets of design parameters [9,10,14].

Also the research concerning real-time OS needs to keep up with the increased complexity derived from the heterogeneous environment. This is a crucial point that needs to be solved, at least in terms of support provided by development environments. The new applications have to meet complex real-time requirements, and designers need of tools that support the easy implementation and verification of such requirements. Without advances in this field, the increasing chip complexity will augment the difficulty in estimating worst-case execution times. This situation can induce designers to choose conservative and simple scheduling solutions, in order to be able to meet the application deadlines and, above all, to certify the timing features of the systems [4,5].

This issue of JEC covers some of these challenges with articles on diverse aspects of embedded systems, focusing in particular on the architectural level. They are consequence of an open call for submission, and a selection of papers collected from various editions of the MEDEA workshop. This workshop, held in conjunction with the PACT conference since 2000,

*Corresponding author. P. Foglia, E-mail: Foglia@iet.unipi.it.
[1]Members of the HIPEAC EU network of excellence.

ISSN 1740-4460/06/$17.00

has moved from decoupled architectures to memory hierarchies' topics with emphasis on embedded and application-specific systems. In this field, there is still a lot of work, deriving from the challenges and opportunity offered by the new class of applications and the incoming limit of technology.

As for the architectural issues, new families of embedded processors have to cope with increasing performance and reduced power consumption. Some possible solutions include the introduction of ad-hoc instructions. Sami Yehia et al., in their work, introduce a new instruction, called load squared, to reduce the memory latency of indirect memory accesses (a memory access pattern generated by the use of complex data structures and sophisticated software conventions). Their load-squared operation is performed by memory-side logic, and the micro-architecture decides which load should be replaced by loads-squared instruction. They first analyze the potential of Loads Squared occurrences for the SpecInt, SpecFP and Olden benchmark suite, then show that performance is improved significantly in some cases, but never degraded. Surendra et al. analyze instruction reuse, a micro architectural technique that improves the execution time of a program by removing redundant computations at run-time. In particular, they analyze the power aspect of instruction reuse, propose an optimization that exploits communication reuse to reduce the power dissipated over a high capacitance bus, and examine the impact of multithreading on instruction reuse in the context of packet header processing applications.

In embedded systems and, in particular, in portable embedded systems, power consumption issues play a key-role for the market value of many emerging products. In fact, the demand of computationally intensive applications (e.g. multimedia) has to be fulfilled with carefully designed solutions able to maximize the energy efficiency of their implementations [12–14]. In this field, together with the dynamic-power issues, the scaling-down of lithographic technology brings also the static-power problems into the embedded domain. Zhang presents two cost-effective approaches to reduce the leakage energy of the instruction cache without significant performance or dynamic energy overheads. The proposed compiler-guided approach exploits static information to identify as many transitional instructions as possible and provides useful hints for the processor to pre-activate the next sub-bank at runtime, for avoiding performance degradation. He proposes also a hybrid approach to exploit both the static and runtime information for the next sub-bank prediction. The paper from Sato at al. presents a mixed hardware management technique to reduce the static power due to sub-threshold leakage. In particular, the paper proposes a cache having fast and slow ways, according to their silicon threshold voltage and a policy that aims to maintain the most used data within the fast ways. This approach allows for significant energy saving with negligible performance degradation, demonstrating its applicability in the embedded scenario.

Many applications that are converging to the domain of portable embedded systems (e.g. PDA, Smartphones) involve information security issues that have to be faced in order to enable the application deployment. For instance, access to value-added services and electronic payments requires information security to be guaranteed at various levels: from the user authentication, the communication protocols on the involved networks, up to the encryption and the non-accessibility of secret information (cryptographic keys, sensible data) on the mobile device, both via software and via hardware [11]. Moreover, security issues reside also in the possibility to certify the integrity of the software installed on a device that is employed to access a secure service. Without such certification, many systems could be hacked through an access device with modified software. Milenkovic et al. analyze various solutions for code certification through integrated approaches at hardware and software level. In particular, the paper highlights the most suitable approaches to be employed in embedded devices.

Due to the increased intelligence of networking applications, network processors are becoming more and more demanding of computing and memory resources. J. Verdu et al. analyze the effect of traffic aggregation (consequence of the increased Internet traffic) on the memory performance of networking applications. After classifying networking application in no-state, steless and stateful categories, they show that the impact on memory is different: the no-state applications are insensitive to the traffic aggregation. Stateless programs are susceptible to increment the data cache miss rate, while stateful applications are the most sensitive to the traffic aggregation level.

Finally, the SuperH embedded processor is an interesting proposal for a variety of consumer applications that run on mobile devices (digital still/video cameras, smart/cell phones), require high computational resources and need extreme power efficiency. Arakawa et al., describe the main architectural features of the 0.13 um SuperH processor both in the standard and in the low-power version. In particular, they describe the

details of the processor architecture and the memory subsystem that are suited for the multimedia application domain. The authors analyze also the features of a newly introduced low-power mechanism.

Taken as a whole, the articles in this special issue illustrate some of the active topics in the domain of the emerging embedded systems. Such systems are pushed by the demand of complex and computationally hungry applications and have to be shaped at the hardware, architectural and software level to meet the hard requirements imposed by the strict available resources. Almost all such resources constraints originate from the limited form-factor and battery capacity needed for portability. In this scenario, power consumption vs. computational power is one of the key trade-off that drives the research from various points of view (e.g. security, memory performance, multimedia management, etc).

Acknowledgements

We would like to thank Roberto Giorgi, Alessio Bechini, Luca Fanucci, Gianluca Dini, Alexander Milenkovic, Avi Mendelson, Stephen Keckler, Naohiko Shimizu, Theo Ungerer, Massimo Alioto, Sören Moch for their invaluable help in the reviewing process.

References

[1] M. Schlett, Trends in Embedded-Microprocessor Design, *IEEE Computer* **31**(8) (August 1988), 44–49.

[2] S. Ramamurthi and J. Srage, Multimedia Technologies on Terminals Based on the OMAP Platform. TI White Paper, SWPY006, June 2002.

[3] Atmel Corporation. Diopsis 740 Dual Core DSP. Atmel Corporation, 2003.

[4] M.J. Pont, Patterns for time-triggered embedded systems: building reliable applications with the 8051 family of microcontrollers. ACM Press/Addison-Wesley Publishing Co., 2001.

[5] M.J. Pont and M.P. Banner, Designing embedded systems using patterns: A case study, *The Journal of Systems and Software* **71**(3) (May, 2004), 201–213.

[6] Z. Chisti, M.D. Powell and T.N. Vijaykumar, *Optimizing Replication, Communication, and Capacity Allocation in CMPs*, In Proceeding of the 32nd International Symposium on Computer Architecture, June 2005, Madison, WC, 357–368.

[7] R. Kumar, V. Zyuban and D.M. Tullsen, *Interconnections in Multi-Core Architectures: Understanding Mechanisms, Overheads and Scaling*, In Proceeding of the 32nd International Symposium on Computer Architecture, June 2005, Madison, WC, 408–419.

[8] S. Balakrishnan, R. Rajwar, M. Upton and K. Lai, *The Impact of Performance Asymmetry in Emerging Multicore Architecture*, In Proceeding of the 32nd International Symposium on Computer Architecture, June 2005, Madison, WC, 506–517.

[9] C. Haubelt and J. Teich, *Accelerating design space exploration using Pareto-front arithmetics*, Proceedings of the ASP-DAC (Design Automation Conference) 2003 Asia and South Pacific, Jan. 2003, 525–531.

[10] A.K. Rath and S. Dehuri, *Non-dominated sorting genetic algorithms for heterogeneous embedded system design*, Proceedings of International Conference on Intelligent Sensing and Information Processing, 2004, Bhubaneshwar, India, 46–50.

[11] P. Schaumont and I. Verbauwhede, Domain-specific codesign for embedded security, *IEEE Computer* **36**(4) (April, 2003), 68–74, ISSN: 0018–9162.

[12] C. Zhang, F. Vahid and W. Najjar, A highly configurable cache architecture for embedded systems, *Proceedings. 30th Annual International Symposium on Computer Architecture* (June, 2003), 136–146.

[13] H. Sbeyti, S. Niar and L. Eeckhout, *Adaptive prefetching for multimedia applications in embedded systems*, Proceedings Design, Automation and Test in Europe Conference and Exhibition, (Vol. 2), Feb. 2004, 1350–1351.

[14] P. Stravers, *Homogeneous multiprocessing for the masses*, ESTImedia 2004. 2nd Workshop on Embedded Systems for Real-Time Multimedia, 2004, iv+106–108.