社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2007－ARC－171 (21)
2007－EMB－ 3 (21)
2007／1／23

Compiler-in-the-Loop Exploration of Embedded Processor Architectures

Nikil Dutt
ACES Laboratory
Center for Embedded Computer Systems
University of California, Irvine
Irvine, CA 92697-3425, USA
dutt@uci.edu

EXTENDED ABSTRACT

As the software content of embedded systems continues to increase, there is a critical need to provide embedded system designers with customized embedded processors that not only deliver the high performance required of demanding applications, but which also simultaneously satisfy a multitude of other (often conflicting) design goals such as energy consumption, reduced code size, system reliability, etc. These multi-dimensional demands of embedded processors necessitate customization of architectural features to deliver the desired design goals. Indeed, embedded processors often have idiosyncratic design features, custom algorithms, and light-weight versions of features present in high-performance processor architectures (in order to reduce the energy consumption and size/cost). Consequently, code generation for the embedded processors is a challenging task. However, if the compiler is able to exploit the specialized architectural features of the embedded processors, it can make a tremendous difference in the resulting power, performance, etc. of the embedded system. Existing embedded processor system design/exploration techniques either do not adequately consider the effects of a compiler on the design, or (when they try) often include the compiler's effects in an ad-hoc manner, which may lead to inaccurate evaluation of design choices and therefore result in suboptimal design decisions. Of course, the designers of embedded processors are presented with a large space of architectural alternatives, only some of which will satisfy all of the design goals for the embedded systems. Thus embedded processor architects critically need tools that quantitatively and qualitatively allow early evaluation and exploration of embedded processor architecture features.

A key notion in this effort is the use of an Architecture Description Language (ADL) to concisely describe the embedded processor features, and a method to generate a customized toolchain (including a compiler) to enable early architectural exploration. We use the phrase "Compiler-in-the-Loop (C-I-L) Exploration" to denote a systematic method that includes the effects of a compiler during architectural evaluation of embedded processors. Our work in this area has demonstrated the need and usefulness of this C-I-L methodology at several levels of embedded processor design, including the instruction set architecture level, the processor pipeline design level, the memory design level, and at the processor-memory interface level. At each level of design abstraction, we have shown that our C-I-L methodology results in a more meaningful exploration of the design space, leading to better design decisions very early in the embedded processor exploration flow.

ACKNOWLEDGEMENTS

This work was done jointly with Aviral Shrivastava (currently at Arizona State University), and was partially supported by SRC and NSF.

REFERENCES

[1] A. Halambi, A. Shrivastava, N. Dutt, A. Nicolau. A customizable compiler framework for embedded systems. SCOPES 2001.

[2] A. Shrivastava, E. Earlie, N. Dutt, A. Nicolau, "PBExplore: A Framework for Compiler-in-the-Loop Exploration of Partial Bypassing in Embedded Processors" DATE 2005.

[3] A. Shrivastava, I. Issenin and N.D. Dutt, "Compilation Techniques for Energy Reduction In Horizontally Partitioned Cache Architectures," CASES 2005.

[4] A. Shrivastava, P. Biswas, A. Halambi, N. Dutt, and A. Nicolau: Compilation framework for code size reduction using reduced bit-width ISAs. ACM TODAES, Jan 2006.

[5] P. Mishra, A. Shrivastava, and N. Dutt, "Architecture Description Language (ADL)-driven Software Toolkit Generation for Architectural Exploration of Programmable SOCs," ACM TODAES, July 2006.

[6] A. Shrivastava, N.D. Dutt, A. Nicolau, and E. Earlie, "Retargetable Pipeline Hazard Detection for Partially Bypassed Processors," IEEE TVLSI, Aug 2006.

[7] Q. Zhu, A. Shrivastava, and N.D. Dutt, "Functional Validation for Partially Bypassed Processor Pipelines," DATE 2007.