Automatic Topology-Based Identification of Instruction-Set Extensions for Embedded Processors

Laura Pozzi, Miljan Vuletić, and Paolo Ienne
EPFL, IN-F Ecublens, 1015 Lausanne, Switzerland
{Laura.Pozzi, Miljan.Vuletic, Paolo.Ienne}@epfl.ch

The need for high performance in ASIC embedded processors, coupled with aggressive energy and area goals, is pushing researchers and designers toward *processor specialisation* for a given application-domain. In this paper, specialisation is addressed through introduction of Ad-hoc Functional Units—special arithmetic/logic units added to a traditional architecture to perform domain-specific complex operations.

Goal. The aim of this work is to identify automatically and with very low effort some relatively small clusters of combinatorial operations which can be collapsed in single opcodes. Traditionally, the design of ad-hoc extensions has been lengthy and burdensome, hence affordable only on mature products.

Topology-Based Identification. The proposed identification algorithm is applied to the Data Flow Graphs of the application and extracts subgraphs with some desired topological features. These are both useful to enforce microarchitectural constraints (e.g., read/write ports) and to ensure optimality (e.g., largest possible subgraph). Specifically, this work addresses Multiple Inputs Single Output graphs (MISO), and in particular MISO graphs of maximal size. The significance of this type of subgraphs for automated IS extension identification has been shown both theoretically [1] and through experimental results [2].

Experiments. The experimental setup consists of the following stages: (1) A first pass of compilation into the SUIF [3] intermediate representation is applied. (2) Profiling is performed and annotated in the SUIF representation. (3) Code analysis for instruction identification follows. The MISO identification algorithm [2] is applied to all basic blocks, and all candidates found are ranked in terms of potential speedup. (4) Instruction selection is run; the best n candidates are chosen. For the unextended processor, each node is considered to execute in one cycle; subgraphs are considered to execute in one cycle, when collapsed into special instructions (an assumption validated in [2]).

opcodes no	2	4	8	16
gsmencode	15.4%	24.9%	35.5%	41.9%
jpegtran	11.6%	15.1%	18.9%	22.9%
mpeg2encode	21.7%	33.9%	37.4%	37.8%
adpcmencode	23.5%	29.6%	29.6%	29.6%
md5	24.0%	38.9%	54.0%	54.9%

Table 1. Potential cycle savings.

Results. Results obtained on the *MediaBench* suite are shown in Table 1, for a varying number of ad-hoc opcodes. Only MISOs with at most 6 inputs are considered, to enforce a reasonable microarchitectural constraint. With a reasonable number of ad-hoc opcodes assigned (e.g., 8), one observes dynamic cycle savings between 18% and 54% with typical values around 30% (typical speedups of $\times$ up to above $\times$). Note that one can implement several complex operations on the same functional unit to keep the total number of read/write ports reasonable. A more comprehensive description of the method used and a comparison with the state of the art can be found in [2].

Conclusions. Interesting cycle reductions have been shown, and significantly, the requirements on the standard toolset of the original processor are very modest: together with the present identification technique, the possibility of inlining the new opcodes in standard C code and of modelling them in the compiler and simulator would be enough to complete a fully automatic processor specialisation flow from C code to hardware and software.

References

[1] L. Pozzi. *Methodologies for the Design of Application-Specific Reconfigurable VLIW Processors*. Ph.D. thesis, Politecnico di Milano, Milano, Jan. 2000.

[2] L. Pozzi et al. Automatic topology-based identification of instruction-set extensions for embedded processors. TR 01/377, EPFL, DI-LAP, Lausanne, Dec. 2001.

[3] R. Wilson et al. SUIF: An infrastructure for research on parallelizing and optimizing compilers. *SIGPLAN Notices*, 29:31–37, Dec. 1994.

Proceedings of the 2002 Design, Automation and Test in Europe Conference and Exhibition (DATE'02)
1530-1591/02 $17.00 © 2002 **IEEE**