Executing ARMv8 Loop Traces on Reconfigurable Accelerator via Binary Translation Framework

Nuno Paulino, João Canas Ferreira, João Bispo, João M.P. Cardoso
INESC TEC and
Faculty of Engineering of the University of Porto
Porto, PORTUGAL
{nuno.m.paulino, joao.c.ferreira, joao.bispo, joao.paiva.cardoso@inesctec.pt}

***Abstract*—Performance and power efficiency in edge and embedded systems can benefit from specialized hardware. To avoid the effort of manual hardware design, we explore the generation of accelerator circuits from binary instruction traces for several Instruction Set Architectures.**

***Index Terms*—accelerator, instruction traces, binary acceleration, HW/SW partitioning, heterogeneous systems, ARMv8**

One of the requirements of edge computing and embedded systems is power efficiency. Systems with heterogeneous hardware are an efficient approach for promoting both an increase in computing performance and a reduction in energy consumption, since well defined computing kernels can be offloaded to specialized units. Further gains can be attained by designing application-specific circuits on a per-case basis.

This work focuses on providing this heterogeneity to embedded and edge devices, while also easing hardware design effort, by automating generation of specialized hardware. We rely on low-level information such as analysis of the program binary, or on retrieved instruction traces. We thus aim to prevent interference with software programming flows, and to provide modest but ubiquitous acceleration from embedded applications. We have demonstrated this in previous work regarding acceleration of loop traces [PFC19a], where MicroBlaze applications are executed on automatically generated modulo-scheduled accelerators capable of reconfiguration via Dynamic Partial Reconfiguration.

Using a redesigned binary translation framework, we are currently expanding the applicability of our approach to other Instruction Set Architectures (ISAs), and exploring additional accelerator architecture, such as custom instruction units and nested loop accelerators. The framework can generate Control and Dataflow Graphs (CDFGs) representing portions of binary code, i.e., binary segments, which are extracted from either static analysis, or from instruction traces [PFC19b]. The former analysis is performed by inspection of the ELF file, and the latter by offline simulation via a combination of QEMU [Bel05] emulation and *gdb*. Different types of binary segments can be detected, among which, repeating loop traces.

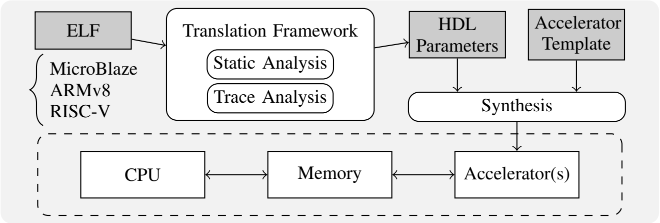

Fig. 1. Generation of accelerator circuits from binary analysis of MicroBlaze, ARMv8, or RISC-V binaries

We will present the use of the binary translation framework for generation of hardware specifications for a loop accelerator. After detection and selection of loop traces, their CDFG representations are used to generate a specialized instance of a loop accelerator template. Speedups over CPU-only execution are achieved by exploiting memory access parallelism, Instruction Level Parallelism (ILP), and loop pipelining. For ARMv8 binaries, the latent ILP results in a potential to execute 5.6 Instruction per Clock Cycle (IPC) in hot Basic Blocks, and 7.6 IPC through single-cycle custom instructions.

On-going work is addressing the reduction of accesses to main memory by optimizing away memory access instructions present in the CDFGs. We will also present preliminary support for detection of static and trace binary segments from RISC-V binaries. Future work will focus on hardware architectures other than a single loop accelerator as targets. For instance, single-cycle sub-graph accelerators (i.e., custom instruction engines), multiple concurrently executing loop accelerators, or architectures such as CGRAs [FGD+19].

References

[Bel05] Fabrice Bellard. QEMU, a Fast and Portable Dynamic Translator. In *USENIX Annual Technical Conference, FREENIX Track*, pages 41–46. USENIX, 2005.

[FGD+19] Luis Fiolhais, Fernando Gonçalves, Rui Duarte, Mário Véstias, and José Sousa. Low Energy Heterogeneous Computing with Multiple RISC-V and CGRA Cores. pages 1–5, 05 2019.

[PFC19a] Nuno Paulino, João C. Ferreira, and João M.P. Cardoso. Dynamic Partial Reconfiguration of Customized Single-Row Accelerators. *IEEE Trans. on VLSI Systems*, 27(1):116–125, 2019.

[PFC19b] Nuno Paulino, João C. Ferreira, and João M.P. Cardoso, Ferreira. Improving Performance and Energy Consumption in Embedded Systems via Binary Acceleration: A Survey. *ACM Comput. Surv.*, 52(6), 2019.

This work was supported by the PEPCC project, ”PTDC/EEI-HAC/30848/2017”, financed by FCT (Fundação para a Ciência e Tecnologia - Portuguese Fundation for Science and Technology).