Offline Phase Classification and Tuning for Multi-Configuration Hardware

Frederik Vandeputte* Lieven Eeckhout Koen De Bosschere

Ghent University, Electronics and Information Systems Department
Sint-Pietersnieuwstraat 41, 9000 Gent, Belgium

Abstract—*Adaptive processing can be used for reducing energy consumption of modern microprocessors while maintaining near optimal performance. In previous work, a multitude of approaches were presented to adapt various hardware structures to the phase behavior of a program execution. However, these approaches were unable to efficiently deal with a large number of possible hardware configurations in case of multiple adaptive structures.*

In this paper, we present a solution to this configuration space search problem by extending the temporal adaptation approach. We first identify phases through profiling of fixed length intervals and subsequently determine the optimal hardware configuration per phase using an efficient offline search algorithm. During program execution, we then inspect the phase behavior and adapt the hardware on a per-phase basis. With our approach, we are able to achieve an energy reduction of 37% on average (up to 61%) with an average 2.9% performance degradation.

1 Introduction

Energy dissipation is a major design issue for modern microprocessors. To address this several researchers have proposed to apply adaptive processing [1, 2, 3, 4, 5, 9, 10, 11, 14, 15]. The idea is to dynamically tune or resize several hardware resources to the application needs without affecting overall performance. As such, the microprocessor adapts itself to the execution phases thereby reducing energy consumption.

A common problem for all adaptive processing approaches is how to efficiently deal with a large number of possible hardware configurations in case of multiple adaptive structures. In this paper, we present a solution to the problem of efficient adaptation in multi-configuration hardware.

The biggest benefit of our approach is that the configuration space exploration is done much more efficiently. It does not require (i) an exhaustive configuration space search, nor (ii) a complete benchmark execution for each configuration point. Our approach uses an efficient search algorithm in combination with simulation of small instruction intervals (1M instructions in this study).

*This research was funded by Ghent University and by the Fund for Scientific Research-Flanders (FWO-Flanders).

Table 1: Baseline simulation model.

Width	8-wide - double fetch speed
FUs	8 ialu 4 mem 2 falu 2 imul 2 fmul
Buff.	128 ROB - 32 LSQ
BPred	8k 2lev + 8k bimod + 8k meta
L1I	8KB 2WSA 32b 2 lat.
L1D	16KB 4WSA 32b 2 lat.
L2U	1024KB 4WSA 64b 20 lat.
Mem	151 lat.

2 Methodology

For our research, we used Simplescalar/Alpha v3.0 [7] in combination with Wattch v1.02 [6] to collect detailed performance and energy consumption data. Our baseline 8-issue processor model is depicted in Table 1. We used a subset of the SPEC2000 benchmark suite. These programs were all simulated from start to completion. We used the train inputs for our offline phase analysis and per-phase optimal hardware configuration exploration; the reported results are obtained using the reference inputs.

3 General Framework

This section presents our adaptation processing approach that is capable of efficiently dealing with multiple adaptive hardware structures. Our framework is based on the one proposed by Sherwood *et al.* [13] with several modifications and simplifications to deal with the offline phase analysis and adaptation. We first collect so called *footprints* during a profiling run. These footprints contain information concerning the basic blocks being executed over a fixed length interval (1M instruction intervals in our case). Once these footprints are collected, we extract execution phases by clustering intervals with similar footprints. This is done offline. For each phase, we select a representative interval for which we determine the optimal architectural configuration through an offline configuration space exploration. The

The remainder of this paper is not included as this paper is copyrighted material. If you wish to obtain an electronic version of this paper, please send an email to bib@elis.UGent.be with a request for publication P104.082.pdf.