Finding and Applying Loop Transformations for Generating Optimized FPGA Implementations

Harald Devos, Kristof Beyls, Mark Christiaens,
Jan Van Campenhout, Erik H. D'Hollander, and Dirk Stroobandt

Parallel Information Systems, ELIS-Dept., Faculty of Engineering,
Ghent University, Belgium
{harald.devos, kristof.beyls, mark.christiaens, jan.vancampenhout,
erik.dhollander, dirk.stroobandt}@elis.ugent.be
http://www.elis.ugent.be/paris

Abstract. When implementing multimedia applications, solutions in dedicated hardware are chosen only when the required performance or energy-efficiency cannot be met with a software solution. The performance of a hardware design critically depends upon having high levels of parallelism and data locality. Often a long sequence of high-level transformations is needed to sufficiently increase the locality and parallelism. The effect of the transformations is known only after translating the high-level code into a specific design at the circuit level. When the constraints are not met, hardware designers need to redo the high-level loop transformations, and repeat all subsequent translation steps, which leads to long design times.

We propose a method to reduce design time through the synergistic combination of techniques (a) to quickly pinpoint the loop transformations that increase locality; (b) to refactor loops in a polyhedral model and check whether a sequence of refactorings is legal; (c) to generate efficient structural VHDL from the optimized refactored algorithm.

The implementation of these techniques in a tool suite results in a far shorter design time of hours instead of days or weeks. A 2D-inverse discrete wavelet transform was taken as a case study. The results outperform those of a commercial C-to-VHDL compiler, and compare favorably with existing published approaches.

1 Introduction

Multimedia applications have made their way into all kinds of devices, from small mobile systems up to desktop computers, with varying ranges of computational power and requirements. They are part of a large class of signal processing applications that often need hardware acceleration. FPGAs (Field Programmable Gate Arrays) are a popular way to speed up designs [16]. They consist of a large array of elementary hardware blocks (LUTs, Memories, Multipliers, ...). The function of these blocks and the connections between them are programmed (configured) to implement a certain hardware design. All blocks can operate in