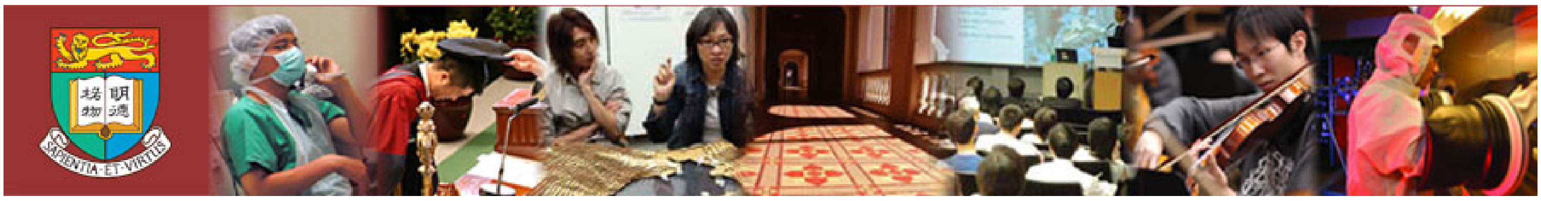

A Soft Coarse-Grained Reconfigurable Array Based High-level Synthesis Methodology: Promoting Design Productivity and Exploring Extreme FPGA Frequency

Cheng Liu, Colin Lin Yu, Hayden Kwok-Hay So
Department of Electrical and Electronic Engineering, The University of Hong Kong
Email: {liucheng, linyu, hso}@eee.hku.hk

I. Introduction

Compared to the use of a typical software development flow, the productivity of developing FPGA-based compute applications remains much lower. Although the use of high-level synthesis (HLS) tools may partly alleviate this shortcoming, the lengthy low-level FPGA implementation process remains a major obstacle to high productivity computing, limiting the number of compile-debug-edit cycles per day. Furthermore, high-level application developers often lack the intimate hardware engineering experience that is needed to achieve high performance on FPGAs, therefore undermining their usefulness as accelerators.

To address the productivity and performance problems, a HLS methodology that utilizes soft coarse-grained reconfigurable arrays (SCGRAs) as an intermediate compilation step is presented. Instead of compiling high-level applications directly to circuits, the compilation process is reduced to an operation scheduling task targeting the SCGRA.

II. Proposed Design Methodology

Figure 1 depicts an overview of the proposed high-level synthesis methodology. As shown in the diagram, the proposed methodology can be divided into two distinct parts. The first part, shown in the top half of the figure, is expected to execute frequently. It should be executed every time a new design iteration is required, a new debug cycle is started, or simply when a new application within the same application domain is implemented. On the other hand, the second part of the design flow, shown in the bottom half of the figure, is expected to execute infrequently, perhaps on a per-application domain basis. Towards the end of the top half of the flow, the scheduling result is merged with the pre-built bitstream from the bottom half to produce the final downloadable bitstream for the target FPGA.

III. Experiment Results

We take 5 computation kernels including matrix multiply (MM), fast Fourier transform (FFT), discrete convolution (CONV),advanced encryption standard (AES) and Viterbi decoder (VD) as our benchmark. Figure 2 and 3 present the performance and compilation time of the benchmark using both AutoESL and the proposed HLS methodology

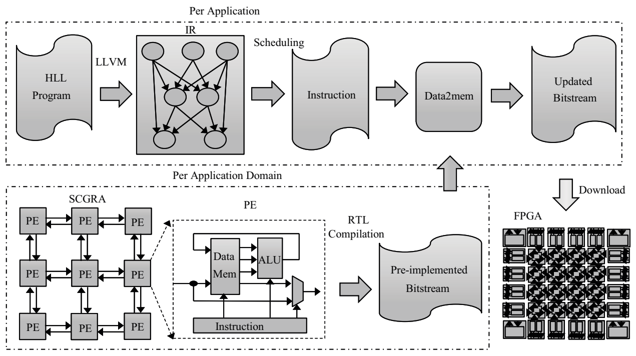

Figure 1. Overview of the proposed soft coarse-grained reconfigurable array based high-level synthesis design methodology.

respectively. It shows that the proposed design methodology achieved 0.8-21x times speedup in the application run time while application compilation time is reduced by 10-100x.

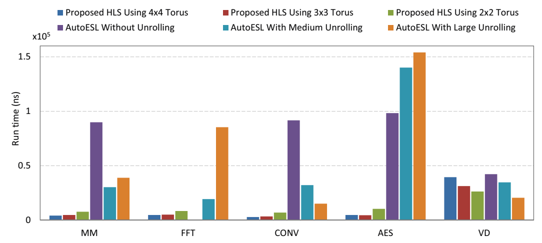

Figure 2. Performance comparison of the benchmark using both AutoESL and the proposed HLS methodology

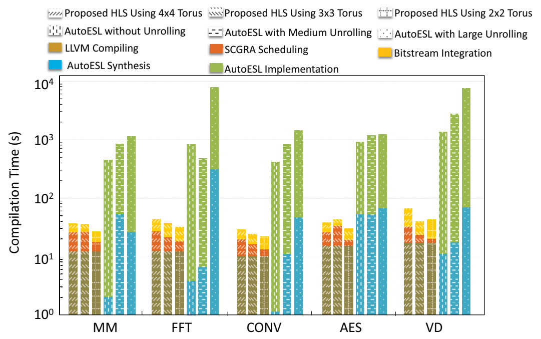

Figure 3. Compilation time comparison of the benchmark using both AutoESL and the proposed HLS methodology