

Chapter 42

Real-Time Image and Video Processing Using High-Level Synthesis (HLS)

Murad Qasaimeh

Iowa State University, USA

Ehab Najeh Salahat

Australian National University, Australia

ABSTRACT

Implementing high-performance, low-cost hardware accelerators for the computationally intensive image and video processing algorithms has attracted a lot of attention in the last 20 years. Most of the recent research efforts were trying to figure out new design automation methods to fill the gap between the ability of realizing efficient accelerators in hardware and the tight performance requirements of the complex image processing algorithms. High-Level synthesis (HLS) is a new method to automate the design process by transforming high-level algorithmic description into digital hardware while satisfying the design constraints. This chapter focuses on evaluating the suitability of using HLS as a new tool to accelerate the most demanding image and video processing algorithms in hardware. It discusses the gained benefits and current limitations, the recent academic and commercial tools, the compiler's optimization techniques and four case studies.

1. INTRODUCTION

In the last two decades, the complexity of image and video processing algorithms has been continuously increasing to meet the demands of complex applications. Pure software implementations for most of these algorithms on embedded systems are still far from reaching real-time performance. This forced embedded designers to accelerate them in hardware to meet the required performance. Some used general purpose processing units (GPUs), others used multicore CPUs, but FPGAs showed, since it is introduced, that it is the most suitable platform to implement image processing algorithms in hardware. FPGAs can be configured to exploit spatial and temporal parallelism in image processing algorithm by realizing multiple processing pipelines that could process data concurrently.

DOI: 10.4018/978-1-5225-5204-8.ch042

Increasing the algorithms complexity went along with a growth in FPGA chip's silicon capacity (Trimberger, 2015). Every couple of years, the silicon size increased due to the improvement in transistor length. With this improvement, porting these algorithms manually into embedded systems still hard and time consuming task to do. It required implementing the whole pipeline in Register Transfer Language (RTL) using one of the hardware description languages such as VHDL or Verilog. Several solutions have been proposed to speed-up the design process such as: simplifying the algorithm itself, using softcore CPUs or using High-level synthesis (HLS). Simplifying the algorithm usually leads to scarify in its accuracy and using softcore CPUs is not always an efficient way to reach the applications target performance and power budget.

High-Level synthesis (HLS) offers a way to generate RTL designs from high-level abstraction in an automatic manner while satisfying the design constraints and optimizing the given cost function. The main goal is to efficiently build and verify hardware, by giving embedded systems designers better control over their designs optimization processes. It allows designers to describe the design at high level of abstraction using one of the high-level languages such as C, C++ or even Matlab, while the tool generates the RTL implementation from these system level abstractions. Verifying the generated RTL, to check that the generated hardware architectures functionality matches the given high-level abstraction, is an important part of the process and can be done at high-level also well using RTL/C co-simulation.

Several advantages can be gained by using HLS technique in image processing hardware accelerators (Bailey, 2015). First, the amount of code to be written by designers using HLS is much lower compared to the manual RTL method. This will save designers a lot of time and reduces the risk of making mistakes. It also gives them the flexibility to optimize their designs by tweaking the source code and tool options to explore large design space alternatives. HLS also reduces the verification time, as the tools can be used to generate high level test benches as well. It also makes it possible to handle more complex designs by removing the need for manual coding. This also allows non-experts in hardware to generate hardware accelerators for their algorithms with minimal effort. There have been many a lot of success stories of using HLS tools in accelerating image processing algorithms, but there are still little systematic studies that focused on evaluating these designs.

The objective of this chapter is to evaluate the suitability of using HLS in implementing real-time image and video processing in embedded systems. It also gives detailed explanations of the design process in HLS and the optimization methods necessary to generate efficient circuits. It presents four image processing algorithms and evaluates the performance of its HLS implementations. The layout of this chapter is as follows. Section 2 presents brief background information about FPGA, HLS and other necessary concepts. Section 3 discusses the benefits and limitations of HLS over other design methods. In Section 4, number of HLS optimization techniques has been presented. An overview of the current academic and commercial tools is presented in Section 5. Section 6 shows an example of HLS video C/C++ library. In Section 7, four case studies have been evaluated. Finally, Section 8 briefly introduces the future work and the current open research problems. Section 9 concludes the chapter.

2. BACKGROUND

This section gives brief background information about Field programmable gate array (FPGA), which is one of the most commonly used hardware platform to accelerate image processing algorithms. It also gives an overview to hardware design flow, which is a set of steps used to generate a hardware design

17 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/real-time-image-and-video-processing-using-high-level-synthesis-hls/196991

Related Content

3D Shape Compression Using Holoimage

Nikolaus Karpinsky and Song Zhang (2012). *Depth Map and 3D Imaging Applications: Algorithms and Technologies* (pp. 87-104).

www.irma-international.org/chapter/shape-compression-using-holoimage/60261

Analysis of New Optoelectronic Device for Detection of Heavy Metals in Corroded Soils: Design a Novel Optoelectronic Devices

Gustavo Lopez Badilla and Juan Manuel Terrazas Gaynor (2017). *Developing and Applying Optoelectronics in Machine Vision* (pp. 273-302).

www.irma-international.org/chapter/analysis-of-new-optoelectronic-device-for-detection-of-heavy-metals-in-corroded-soils/161994

The Development, Applications, Challenges, and Analysis of a Cricket Player Face Recognition Dataset

Mahmood UI Haq, Muhammad Athar Javed Sethi, Subhan Ullah and Abd Ullah (2024). *Deep Cognitive Modelling in Remote Sensing Image Processing* (pp. 173-197).

www.irma-international.org/chapter/the-development-applications-challenges-and-analysis-of-a-cricket-player-face-recognition-dataset/351851

Digital Image Encryption Based on Chaotic Cellular Automata

Zubair Jeelani (2020). *International Journal of Computer Vision and Image Processing* (pp. 29-42).

www.irma-international.org/article/digital-image-encryption-based-on-chaotic-cellular-automata/264219

Fall Detection Depth-Based Using Tilt Angle and Shape Deformation

Fairouz Merrouche and Nadia Baha (2018). *International Journal of Computer Vision and Image Processing* (pp. 26-40).

www.irma-international.org/article/fall-detection-depth-based-using-tilt-angle-and-shape-deformation/214072