Image and video processing on FPGAs: An Exploration Framework for Real-Time Applications

Paulo da Cunha Possa*. Zied El Hadhri.**
Carlos Valderrama***

**University of Mons, Department of Electronics and Microelectronics,*
Mons, 7000 Belgium (Tel: +32-6537-4222; e-mail: Paulo.Possa@umons.ac.be).
***(e-mail: Zied.ELHADHRI@student.umons.ac.be)*
****(e-mail: Carlos.Valderrama@umons.ac.be)*

Abstract: This work presents a design framework for real-time image and video processing enabling exploration and evaluation of different processing techniques. The goal of our educational approach is to develop a flexible and easily customizable environment for prototyping different processing techniques on Field Programmable Gate Arrays (FPGAs), targeting specific applications. In this paper we give an overview of different requirements and techniques of video processing featuring FPGAs. Three real-time video processing algorithms were combined to show the advantages and characteristics of our approach. Within the framework, the modules running in parallel can be easily swapped at run-time according to the application specific needs.

Keywords: Video processing, field programmable gate array, embedded systems, hardware description language, object tracking, pattern detection.

1. INTRODUCTION

Innovations regarding video/image processing and a fast evolution of video standards, such as digital cinema and HDTV, are now taking place. The progress on image capture and display resolution, advanced compression techniques, and smart cameras are the main reasons that are pushing the limits of technology and processing power (Altera, 2007; Bodba, 2007).

Video resolution requirements rose significantly over the last few years. Moving from standard definition (SD) to high definition (HD), with a resolution exceeding 2MP (Megapixel) and a refresh rate between 30 to 60 frames per second, represents a 5.5x increase in processed data. The new video surveillance standards impose a change from Common Intermediate Format (CIF – 352 × 288 pixels per frame) to D1 format (704 × 576), with some industrial cameras providing HD at 1280 × 720. Other video applications like military surveillance, medical imaging, and machine vision are also processing very high resolution images (Altera, 2007; Poynton, 2003).

The new generation of advanced compression techniques reinforces the streaming capability, the compression rate for a given quality, and reduces latency. With an improved resolution and increased compression ratio, there is a crucial need of processing power, all by keeping the architecture rather flexible to follow the last upgrade of the standard. (Altera, 2007).

Digital Signal Processors (DSPs), Application-specific Integrated Circuits (ASICs) and Graphic Processor Units (GPUs) are the platforms commonly used to implement image and video processing algorithms requiring simultaneous computations on multiple pixels/frames. Looking into the architecture, a typical TI DSP processor may have two Arithmetic Logic Units (ALUs) to carry out Multiply & Accumulate (MAC) operations. Comparatively, an FPGA can have more than 200 MAC blocks processing pixels in parallel. Some FPGAs now have dedicated hard-core DSP/MAC blocks for faster processing power (Altera, 2007; Kalomiros and Lygouras, 2008).

FPGAs hold a clear advantage compared to conventional DSPs to perform digital signal processing which is their scalability (the capacity to replicate functions as required) and inherent parallelism. With the arrival of the new age of FPGAs, as a mature technology with increased volume, the growing need for faster and cost-effective systems was overcome (Kalomiros and Lygouras, 2008; Serrano, 2008).

The concept of latency becomes critical with real-time processing applications like video or television signal processing. Therefore, in addition to embedded hardware multipliers and a larger number of memory blocks, computationally demanding functions (e.g. convolution filters, motion estimators, two-dimensional Discrete Cosine Transforms (2D DCTs), and Fast Fourier Transforms (FFTs)) are better provided in the form of specialized hardware components available in modern embedded multimedia architectures. Such components are also available as Intellectual Property (IP) cores to develop FPGA-based applications (Kalomiros and Lygouras, 2008).

Nowadays, students should evolve from software development to architecture design in order to satisfy such requirements. They must master algorithmic selection and processing power requirements working in a design framework built around the latest video/image standards. By adopting a video processing framework based on FPGAs, we can provide real-time exploration in a flexible and evolving environment, populated by a growing set of technology bricks. The proposed exploration framework was built around the Altera DE2 Development and Education Board.

This work is presented as follows. Firstly, we will give an overview of requirements and available techniques concerning video processing on FPGAs. We will analyse previous work to establish the bases of our proposal. Secondly, we will detail the characteristic of our video processing framework. Some examples of video processing functions will be merged to illustrate the design methodology. Finally, we will conclude with an analysis of the results and provide future directions.

2. RELATED WORK

Many works concerning video and image processing on FPGAs can be found with the most diverse applications. From entertainment and video monitoring to medical imagery, the level of design complexity varies according to the methodology and its use.

Within those design environments, we are interested on how the different processing algorithms can be combined in a flexible way to satisfy specific application requirements. One good example is the work of Li et al. (2009) about a multipurpose reconfigurable platform, where more than one process can be applied simultaneously on the incoming video signal. On that approach, a user-specific functional module implements most of the required functionalities, such as four-direction edge detection and rescaling. This functional box can be extended depending on different application scenarios. Another interesting example is given by Meng et al. (2008). They have synthesized a real-time human motion recognition algorithm that can be enhanced to fit intelligent surveillance systems requirements. In that approach, the motion type identification is made by training the system with a reference database using six Support Vector Machine (SVM) classifiers. Referring to the work of Sen et al. (2008), we can see how FPGAs can be exploited in medical imaging by carrying out image registration which consists in aligning two images representing the same features to detect some specific information for diagnosing of different diseases.

3. VIDEO PROCESSING FRAMEWORK

The board chosen to host our framework was the Altera DE2 Development and Education Board (Fig. 1). The motivation of this choice was the need for a module aiming particularly an educational goal, with accessible components for debugging (e.g. toggle switches, debounced pushbutton switches, and LEDs) and a complete set of peripherals, including a 24-bit audio codec, an USB host/slave controller, 10/100 Ethernet controller, 8MB SDRAM, a TV decoder, and a VGA 10-bit DAC. The DE2 board hosted FPGA is the low cost Altera's FPGA device EP2C35 from the family Cyclone II. The EP2C35 contains 33,216 Logic Elements, 105 M4K RAM blocks, 35 embedded multipliers, and 4 PLLs.

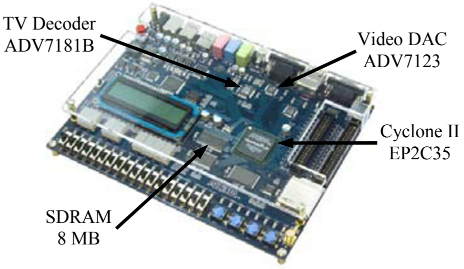

Fig. 1. Altera DE2 Development and Education Board.

The proposed video processing framework takes advantage of the already available TV Decoder (ADV7181B) and Video DAC (ADV7123) to create a low cost environment for video processing. Fig. 2 shows a simplified diagram of the video framework architecture. The input signal is analogue composite video in NTSC format. The TV Decoder converts the input signal to the digital ITU-R BT.656 format.

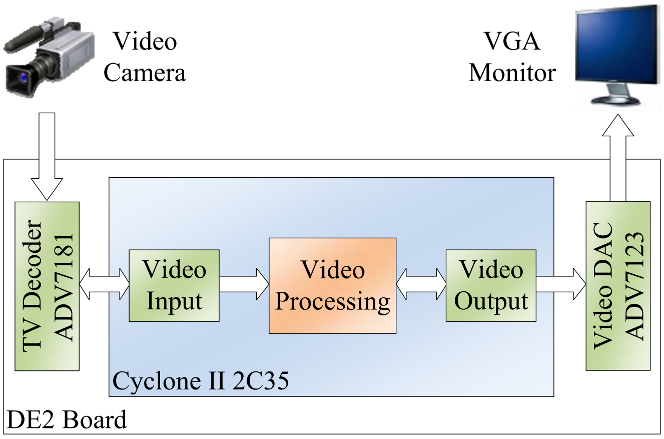

Fig. 2. Simplified diagram of the video framework architecture.

The Video Input and Output modules (Fig. 2) are based on the DE2 TV box demonstration supplied with the DE2 board by Altera. Between these two modules can be placed video processing operators customized to perform a specific application in real-time.

In the Video Input module (Fig. 3), the ITU-R 656 Decoder block extracts YCbCr 4:2:2 (YUV 4:2:2) video signals from the ITU-R 656 data stream. As the input format is interlaced, a Video De-Intrelacer block is used. This block converts the input video stream into a progressive format aided by a SDRAM frame buffer. After that, the chroma components of the video stream are up-sampled by the YUV 4:2:2 to 4:4:4 block. Finally, the video stream is converted from YUV colour format to RGB, which is the input format of the Video Processing block. The two other blocks are auxiliary. The I2C AV Configuration block is dedicated to send the configuration parameters to the TV Decoder by I2C protocol. The Locked Detector & Initiation Controller detects initial instabilities in the TV Detector and control the initialisation sequence of the entire system.