FPGA: An Efficient And Promising Platform For Real-Time Image Processing Applications

Sparsh Mittal¹, Saket Gupta² and S. Dasgupta³

¹ E&C Department, Indian Institute of Technology Roorkee, sparsuch@iitr.ernet.in
² E&C Department, Indian Institute of Technology Roorkee, freshuce@iitr.ernet.in
³ E&C Department, Indian Institute of Technology Roorkee, sudebfec@iitr.ernet.in

Abstract

Digital image processing (DIP) is an ever growing area with a variety of applications including medicine, video surveillance, and many more. To implement the upcoming sophisticated DIP algorithms and to process the large amount of data captured from sources such as satellites or medical instruments, intelligent high speed real-time systems have become imperative. Image processing algorithms implemented in hardware (instead of software) have recently emerged as the most viable solution for improving the performance of image processing systems. This paper reviews the relative merit of FPGA over softwares and DSPs as a platform for implementation of DIP applications. Our goal is to familiarize applications programmers with the state of the art in compiling high-level programs to FPGAs, and to survey the relevant research work on FPGAs. The outstanding features which FPGAs offer such as optimization, high computational density, low cost etc, make them an increasingly preferred choice of experts in image processing field today.

Keywords: Digital Image Processing (DIP), FPGA, Hardware Descriptive Language, PC

1. Introduction

Digital image processing [1] is an ever growing area with variety of applications in different fields. As image sizes and bit depths grow larger, software has become less useful in the video-processing realm. Generally even specialized image processing programs running on PCs cannot adequately process large amounts of high-resolution streaming data, since PC processors are made to be for general use and hence are unable to efficiently implement many current sophisticated DIP algorithms. Also, to process the large amount of data captured from satellites and ground-based detection systems or 3D data from medical instruments, intelligent high speed real-time systems have become imperative, which may process data before passing it to the human analyst. The requirements demand a system ideally having characteristics viz. high performance, flexibility, easy upgradability, low development cost, and a migration path to lower cost as the application matures and volume increases.

Recently, image processing algorithms implemented in hardware have emerged as the most viable solution for improving the performance of image processing systems. The introduction of reconfigurable devices and system level hardware programming languages has further accelerated the design of DIP in hardware. FPGAs are often used as implementation platforms for real-time image processing applications. A Field Programmable Gate Array (FPGA) is a programmable (or reconfigurable) device [2] in which the final logic structure can be directly configured by the end user. An FPGA consists of an array of uncommitted elements that can be programmed or interconnected (or configured) according to a user’s specification in a virtually limitless number of ways. Being reprogrammable and easily upgradable, an FPGA offers a compromise between the flexibility of general-purpose processors and the hardware-based speed of ASICs.

In this paper we survey implementation of image processing applications on FPGAs with an emphasis on the salient features of FPGAs. The rest of the paper is organized as follows. Section 2 highlights limitations of other implementation alternatives and serves to set the stage for explaining the advantage of FPGAs. In section 3 we evaluate FPGAs on several relevant parameters. Section 4 summarizes prior research in the FPGA implementation of image processing algorithms. Finally, section 5 concludes the work and gives directions for future work.

2. Setting the stage: Limitations of other platforms

2.1 Drawbacks of DSPs and ASICs

System architecture choices for hardware implementation include standard cell ASICs, ASSPs, and programmable solutions such as digital signal processor (DSP) or media processors and FPGAs. Each of these has advantages and disadvantages. ASSPs are inflexible, expensive, and time-consuming to develop. Full custom ASIC design offers highest performance, but they cannot be changed. Hence any error in the hardware design existing after fabrication wastes the entire product. DSPs are specialized microprocessors, typically programmed in C, or with assembly code for improved performance. It is well suited to extremely complex math intensive tasks such as image processing. Knowledge of hardware design is still required, but its advantage is that the learning curve is much lower than other design choices. However, powerful DSPs are costly and their corresponding software applications may not match the performance of hardware.

The reconfigurable computing technology in FPGAs [3], along with many other features of FPGAs make them ideally suited for real-time video processing. Hardware design techniques such as parallelism and pipelining techniques can be developed on an FPGA [4], which is not possible in dedicated DSP designs. The primary reason most engineers choose an FPGA over a DSP is driven by the MIPS requirements of an application [5]. The need for high performance rules out processor-only architectures. A state-of-the-art DSP running at 1 GHz cannot perform H.264 HD decoding or H.264 HD encoding. FPGAs are the only programmable solutions able to tackle this problem.
2.2 Drawbacks of PCs
Software implementation of most image processing algorithms has several limitations and hence it is quite difficult to achieve. Complex operations have to be realized by a large sequence of simple operations, which can only be implemented serially. The range of available operations is limited to common basic operations. The constraint of real-time processing introduces a number of additional complications. These include such issues as limited memory bandwidth, resource conflicts, and the need for pipelining. The CPU is burdened with additional tasks, such as OS requests, user interaction, etc., which is a major drawback in the context of real-time processing. At real-time video rates of 25 frames per second a single operation performed on every pixel of a 768 by 576 color image (PAL frame) equates to 33 million operations per second. (excluding the overhead of storing and retrieving pixel values.) Many image-processing applications require that several operations be performed on each pixel in the image resulting in an even large number of operations per second. As a result it is difficult to meet hard real time requirements with softwares [3].

2.3 The advantages of FPGAs
The salient features of FPGAs that make them superior in speed, over conventional general-purpose hardware like Pentiums is their greater I/O bandwidth to local memory, pipelining, parallelism and availability of optimizing compiler. Complex tasks, which involve, multiple image operators, run much faster on FPGAs than on Pentiums, in fact, ref [6] report 800 time speed up by FPGA using SA-C. There are several reasons for such large speed up which FPGAs have over PCs (see section 3). In comparison to an FPGA, hardware such as Pentium runs at memory speed, not at cache speed. So, even running at much higher clock frequency and having the facility of cache memory, it responds much slower than a comparable FPGA. Frequency of operation in hardware such as Pentium can be increased up to a certain extent to increase the performance or the required data rate to process the image data, but increasing the frequency above certain limits causes system level and board level issues that become a bottle neck in the design. However, the difference in speeds also depends on the particular application involved [6]. In particular, complex image processing applications do enough processing per pixel to be computation bounded, rather than I/O bounded. Here, FPGAs dramatically outperform Pentiums by factors of up to 800. Simpler image processing operators tend to be I/O bounded. In these cases, FPGAs still outperform Pentiums because of their greater I/O capabilities, but by smaller margins (factors of 10 or less).

3. Evaluation of FPGAs as Platform for Developing DIP Applications
3.1 Advantages of FPGAs
Many advantages of FPGAs make them a preferred choice of implementation in DIP realm. Based on our survey, we find many significant features which are as follows:-

1. A characteristic of many image-processing methods is the multiple iterative processing of data sets such as four stages of canny edge detector, which require performing multiple passes over the image. These steps, which have to be performed sequentially on a general-purpose computer, can be fused in one pass in FPGA, as their structure is able to exploit spatial and temporal parallelism. FPGA can perform multiple image windows in parallel and multiple operations within one window also in parallel.
2. By employing several optimizations techniques such as Loop Fusion, Loop Unrolling etc efficient usage of FPGA resources and speed-up in implementations is possible by avoiding many redundant operations.
3. FPGAs are capable of parallel I/O, which allows them to perform read (from memory), process and write (to memory) simultaneously. Many operations such as convolutions, finding square root etc can be executed much faster by using pipelining and parallelism.
4. All of the logic in an FPGA can be rewired, or reconfigured, with a different design as often as the designer likes. This type of architecture allows a large variety of logic designs dependent on the processor’s resources), which can be interchanged for a new design as soon as the device can be reprogrammed.
5. FPGAs provide the flexibility to reprogram and upgrade to new standards [7]. Easy Upgradeability ensures that FPGAs solutions evolve quickly with no risk of obsolescence.
6. The reusability and efficiency of hardware implemented on FPGA, is especially useful in developing Image Processing IP (intellectual property) as it allows an efficient system in terms of cost and performance. Possibility of quick integration of the IP blocks without a need of modification or repetition of verification cycle [7] simplifies debugging and thus greatly reduces the time-to-market.
7. Because of its LUT based architecture, some convolution masks (such as constant coefficient multiplier or KCMs) can be implemented very efficiently [8].
8. High computational density in FPGA together with a low development costs allows even the lowest volume consumer market to bear the development costs of FPGAs. In fact, compared to ASICs, FPGAs are especially useful in a lower volume type of application. As authors in [9] report with low-cost FPGAs high-definition solutions can now be implemented for less than US$1.00 per 1,000 logic elements (LEs)
9. The wide range of FPGAs made available from various companies like Xilinx, Altera, etc fulfill the performance requirement of many of the applications, such as display products [10].

3.2 Limitations of FPGAs
On the other hand, these are the limitations of FPGAs for image processing applications.

1. Currently there are many overheads in FPGA design. This include data transfer times which is the time required to upload (or download) the data, from (or to) reconfigurable processor to (or from) host; time for reconfiguration.
2. FPGA's are excellent choice only for those algorithms which don’t use floating-point mathematics or complex mathematics.
Division, direct multiplication etc. are very complex and expensive on FPGA. Hence, the designers have to reformulate their algorithms and avoid complex mathematics (e.g. implementing a divide by 8 using the bit shifting method of division instead of a divide by 9).
3. Current FPGA's cannot be reconfigured quickly as the process of modifying or combining FPGA circuits is also laborious.
4. The size of memory that can be implemented using standard logic cells on an FPGA is limited, as implementing memory is an inefficient use of FPGA resources.
5. Routines where complex tasks cannot be broken down into simpler tasks must perform a more serial method of processing, which is not entirely efficient with FPGA's.
6. Hardware offers much greater speed than a software implementation, but it comes with a price of increased development time inherent in creating a hardware design. Most software designers are familiar with C, but in order to develop a hardware system, one must either learn a hardware design language such as VHDL or a software-to-hardware conversion scheme, such as streams-C [11], which converts C code to VHDL, or MATCH [12], which converts MATLAB code to VHDL.

4. Implementation of DIP applications on FPGA's
A lot of research has been recently done on utilizing FPGA's as development platform for DIP algorithms. Here we present the related work in the area. The authors in [6] have developed a high level language (called SA-C) for expressing DIP algorithms, and an optimizing compiler that compiles the high-level program written on SA-C and runs them on FPGA's. SA-C is a single-assignment dialect of the C programming language designed to exploit many features of FPGA's [13, 14]. To compare the performance of FPGA's and the Pentium processors, they have implemented SA-C programs compiled to a Xilinx FPGA to equivalent programs running on an 800 MHz Pentium III. For 8 common DIP routines implemented on both these platforms, FPGA's offer 8 to 800 times speed-ups over the Pentium. Experiment results and analysis of various issues such as pipelining, parallelism, optimizations, memory, I/O etc., brings out many prominent features of the FPGA's, relevant to image processing realm. In [15] they present performance numbers for several image-processing routines such as Gaussian, max and Laplace filter etc., written in SA-C.

The authors in [16] present a pipelined architecture of image processing algorithms like median filter, basic morphological operators, convolution and edge detection implemented on FPGA. The hardware modeling is done with the Handel-C language. Moreover, in their work [17], the performance and efficiency of Handel-C language on image processing algorithms is compared at simulation level with another C-based system level language called SystemC and at synthesis level with the industry standard Hardware Descriptive language (HDL), Verilog. Comparison parameters at simulation level include, man-hours for implementation, compile time and lines of code. Comparison parameters at synthesis level include logic resources required, maximum frequency of operation and execution time.

The author in [18] implemented the Rank Order Filter, Erosion, Dilation, Opening, Closing and Convolution algorithms using VHDL and MATLAB on two FPGA platforms. He also integrates the FPGA algorithms into the modeling environment called ACS.

The authors in [19] address the issue of mapping algorithms to hardware. They present some general techniques such as look up tables, raster based methods etc. for dealing with expressions that inefficiently map to hardware. They discuss the effects of and means to deal with the timing, bandwidth, and resource constraints under different processing modes of the system. The efficient mapping of three different types of general operations viz. point, window and global operations is discussed in relation to the hardware constraints.

The authors in [3] report the speed ups that FPGA's offer on image processing methods (such as image denoising and restoration, segmentation, morphological shape recovery etc.) on 2D and 3D images.

In computer vision and image processing, FPGA's have already been used to accelerate real-time point tracking [20], stereo [21], color-based object detection [22], and video and image compression [23] (see also [17]). Crookes presented a hardware FPGA implementation of image filtering to increase the speed [24, 25]. The authors in [26] applied three 2-input bubble sorting algorithm to obtain a triple input sorter and implemented it in FPGA. This algorithm can be utilized to obtain the maximum, middle, and minimum values and hence can be used to realize the 2-D sorting.

5. Conclusion and Future work
With a multi billion-dollar market per year, increases in FPGA speeds and capacities have followed or exceeded Moore’s law for the last several years. This survey clearly demonstrates the outstanding features of FPGA's which make them seem very promising choice for the researchers in the field. FPGA's are great fits for video and image processing applications, such as broadcast infrastructure, medical imaging, HD videoconferencing, video surveillance, and military imaging.

The greater future potential lies in including FPGA's on-chip with the main processor, giving the benefit of general-purpose acceleration without the communication bottleneck created by placing the FPGA in a co-processor. At the moment, applications written directly in VHDL are more efficient (albeit more difficult to develop), but we expect future improvements to the compiler to narrow this gap. Also the disadvantages of FPGA's need to be addressed to make them more efficient and useful.

6. Reference


