2

RECENT DEVELOPMENTS IN THE DESIGN OF IMAGE AND VIDEO PROCESSING ICs

Konstantinos Konstantinides and Vasudev Bhaskaran
Hewlett-Packard Laboratories

1. Introduction

Hardware support for image processing is very important in emerging applications such as desktop publishing, medical imaging, and multimedia. In the past, computational needs for intensive image processing tasks, such as image analysis of satellite data and pattern recognition, were satisfied with custom, complex, and expensive image processing architectures [1], [2]. However, the latest scientific workstations have enough compute power for low-cost desktop image processing. Furthermore, traditional image processing operations, such as texture mapping and warping, are now combined with conventional graphics techniques. Hence, there is an increased interest for accelerated image and video processing on low cost computational platforms, such as personal computers and scientific workstations.

Many developers provide already some type of image processing support, based mostly on general purpose microprocessors or digital signal processors (DSPs), such as the i860 from INTEL, or the TMS320 family from Texas Instruments. However, new application areas, such as high-definition TV (HDTV) and video teleconferencing demand processing power that existing general purpose DSPs cannot provide.

Until recently, commercially-available image processing ICs performed only low-level imaging operations, such as convolution, and had very limited, if any, programming capability. This was due to the well defined operation and data representation of the low-level imaging functions; furthermore, there was a general feeling that the image processing market was quite small. Emerging standards, such as JPEG (Joint Photographic Experts Group) and MPEG (Moving Picture Experts Group) for image and video compression, and the opening of new market areas, such as multimedia computing, make it easier now for manufacturers to invest in the design and development of a new generation of image and video processing ICs. For example, image compression ICs are now becoming widely available. Thus, a new generation of general purpose image processors will soon emerge to extend the capabilities of general purpose DSPs.

M. A. Bayoumi et al. (eds.), *VLSI Signal Processing Technology*

In this paper we present a brief overview of the specific requirements in image processing and some recent developments in the design of image and video processing ICs. Both application-specific (ASIC) and programmable image processors are discussed. For the application-specific processors, special emphasis is given to the processing requirements of recent standards in image and video compression. We close with a discussion on a "generic" general purpose image processing architecture. As general purpose DSPs share many common features (a Harvard architecture, multiple data memories, etc.), we expect that future general purpose image processors will share many of the features embodied in the generic design.

2. Image Processing Requirements

Image processing is a broad field that spans a wide range of applications such as document processing, machine vision, geophysical imaging, multimedia, graphics arts, and medical imaging. A careful examination of the imaging operations needed for these applications suggests that one can classify the image processing operations into low, intermediate, and high levels of complexity. In low-level processing (i.e. filtering, scaling, and thresholding), all operations are performed in the pixel domain and both input and output are in the form of a pixel array. In intermediate-level processing, such as edge detection, the transformed input data can no longer be expressed as just an image-sized pixel array. Finally, high-level processing, such as feature extraction and pattern recognition, attempts to interpret this data in order to describe the image content.

Because of this large range of operations, it seems that every conceivable type of computer architecture has been applied at one time or another for image processing 2 and that so far, there is no single architecture that can efficiently address all possible problems. For example, Single Instruction Multiple Data (SIMD) architectures are well suited for low-level image processing algorithms; however, due to their limited local control they cannot address complex, high-level, algorithms [3]. Multiple Instruction Multiple Data (MIMD) architectures are better suited for high-level algorithms, but they require extensive support for efficient inter-processor communication, data management, and programming. Regardless of the specifics, every image processing architecture needs to address the following requirements: processing power, efficient data addressing, and data management and I/O.

2.1 Processing Power

To better understand the computational requirements of image processing algorithms, consider first the 1-D space. In 1-D digital signal processing, a large class of algorithms can be described by

$$Y(i) = \sum_{k=1}^{p} C_k X(i-k), \tag{1}$$

where $\{X(i)\}$ and $\{Y(i)\}$ denote the input and output data, and $\{C_1, C_2, \ldots, C_p\}$ are

algorithm-dependent coefficients. Using a general purpose DSP that has a multiply-accumulate unit and can access simultaneously the input data and the coefficients, an output sample can be generated every p cycles.

In image processing, a large class of algorithms can be described by

$$Y(i,j) = \sum_{m=1}^{p} \sum_{n=1}^{q} C_{mn} X(i-m, j-n), \tag{2}$$

where X and Y are the input and output images, and C is a p x q matrix of algorithm-dependent coefficients. From (2), a single DSP requires now at least pq cycles to generate an output pixel. In a multi-DSP system, at least q DSPs have to be used to generate an output sample every p cycles, provided that all processors have direct access to the data and there exists efficient inter-processor communication and a data distribution mechanism. Thus, the first main requirement in image processing is *processing power.*

2.2 2-D Addressing

Another major requirement in image processing is *efficient data addressing.* In conventional DSPs, a simple p-modulo counter is adequate for the data addressing in (1) [4]. However, in image processing, the addressing schemes may be far more complicated. For example, Fig. 1 shows an N x M image, and a p x p kernel of coefficients positioned with its left upper corner at address A of the image. From (2), to compute an output sample, one possible scheme to sequentially access the image data is to generate data addresses in the following sequence.

$$\begin{array}{llll} A, & A+1, & \cdots & A+(p-1), \\ A+N, & A+N+1, & \cdots & A+N+(p-1), \\ A+2N, & \cdots & \cdots & A+2N+(p-1), \\ \cdots & \cdots & \cdots & \cdots \\ A+(p-1)N, & A+(p-1)N+1, & \cdots & A+(p-1)(N+1). \end{array}$$

Fig. 2 shows a 2-D address arithmetic unit (AAU) as implemented on the Video DSP by Matsushita [5]. This AAU has two parts, the controller, and the address generator. The AAU operation is controlled by the values in five registers: SA, NX, NY, DX, and DY, as shown in Fig. 2 . In our case, SA=A, NX=p, NY=p, DX=1, and DY=$N-(p-1)$. To generate addresses from A to $A+(p-1)$, the adder is incremented by DX=1. After each row of the kernel is processed, the "Row End" signal enables DY=$N-(p-1)$ to access the adder, and an address is generated for the beginning of the next row. Operations continue until the "End of Block" signal is enabled.

The HSP45240 IC from Harris Semiconductor is a programmable 24-bit address sequence generator for image processing applications. It can be configured to generate addresses for filtering, FFT, rotation, warping, zooming, etc. The Harris IC is ideal for block oriented address generation. Five configuration registers allow a user to define such parameters as: the beginning address of a sequence, the block size,

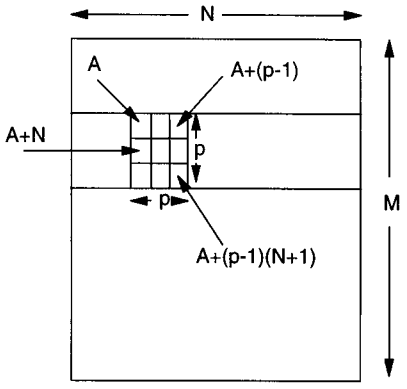

Fig. 1 : 2-D addressing in image processing.

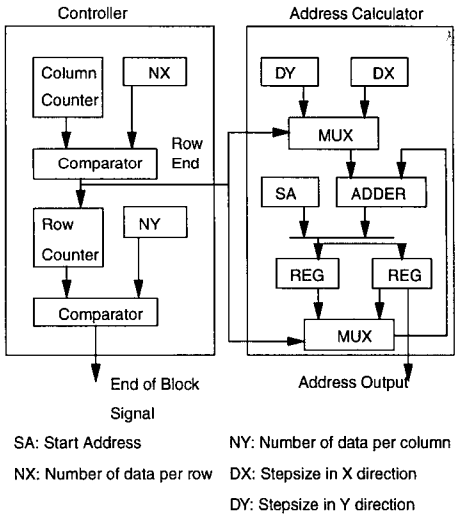

Fig. 2 : Block diagram of a 2-D address generator.

address increment, number of blocks, etc. A crosspoint switch can be used to reorder the address bits. This allows for *bit-reverse* addressing schemes used in FFT computations.

Recent image processor ICs have very long instruction words (VLIW) for the control of multiple AAUs, ALUs and function units. A programmer has to

simultaneously control not only the two to three address generators, but also all the other function units. Kitagaki et al. [6]. presented an AAU design that allows the user to control the address generator controller using high-level code words. Given the high-level code word, an address microsequencer then generates the full sequence of addresses required in a particular application. The design allows for 17 different high-level and commonly used image processing addressing modes, such as: block raster scan, neighborhood search, 1-D and 2-D indirect access, and has special modes for FFT and affine transformation addressing.

2.3 Data Storage and I/O

A 512 x 512, 8-bit grayscale image requires 262 Kbytes of storage and a 24-bit, 1024 x 1024 color image requires approximately 3 Mbytes of storage. Considering that many applications require data processing in real-time (12-30 frames per second or more), it is easy to see why *efficient data storage and I/O* are extremely important in image processing architectures.

Systolic architectures address this problem by distributing the data across an array of processors and allow only local data transfers. Many DSPs achieve efficient I/O by integrating a local Direct Memory Access (DMA) engine with the rest of the architecture 4; furthermore, I/O operations and computations are usually performed in parallel. As we will see next, similar techniques can also be applied into the design of image processor ICs.

Image processing architectures can be divided into the following broad categories: dedicated image processor systems, image processing acceleration systems, and image processor ICs. The dedicated image processor systems usually include a host controller and an imaging subsystem that includes memory, customized processing ICs, and custom or commercially-available math processing units, connected in a SIMD, MIMD, or mixed-type architecture. Most of these systems are being developed either at universities for research in image processing architectures, or at corporate laboratories for in-house use or for specific customers and application areas 3, [7], [8].

Image processing acceleration systems are usually being developed as board level subsystems for commercially-available personal computers or technical workstations. They use standard bus (VME, EISA, NuBus, etc.) interface to communicate with the main host, and usually include memory, one to four commercially-available DSPs (such as the TMS320C40 from Texas Instruments) or high performance microprocessors (such as the INTEL i860), a video port, and a frame buffer. Most of the board-level subsystems also include floating-point arithmetic unit(s) and have a Bus-based architecture [9].

In this paper, the emphasis will be on image processor ICs. These are either application-specific circuits (for example, image compression ICs) with limited programming capability, or programmable circuits designed for a specific range of applications (HDTV, video conferencing, etc.). No single article can cover all existing

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.