Real-Time Image Processing on a Focal Plane SIMD Array

Antonio Gentile[1], José L. Cruz-Rivera[2], D. Scott Wills[1], Leugim Bustelo[2], José J. Figueroa[2], Javier E. Fonseca-Camacho[2], Wilfredo E. Lugo-Beauchamp[2], Ricardo Olivieri[2], Marlyn Quiñones-Cerpa[2], Alexis H. Rivera-Ríos[2], Iomar Vargas-Gonzáles[2], Michelle Viera-Vera[2]

[1] School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332-0250
{gentile, scott.wills}@ee.gatech.edu
[2] Department of Electrical and Computer Engineering
University of Puerto Rico in Mayagüez
Mayagüez, Puerto Rico 00681-9042
jcruz@ece.uprm.edu

Abstract. Real-time image processing applications have tremendous computational workloads and I/O throughput requirements. Operation in mobile, portable devices poses stringent resource limitations (size, weight, and power). The SIMD Pixel Processor (SIMPil) has been designed at Georgia Tech to address these problems. In SIMPil, an image sensor array (focal plane) is integrated on top of a SIMD computing layer, where processing elements (PEs) are connected in a torus. A prototype processing element has been implemented in 0.8 μm CMOS technology. This paper evaluates the effectiveness of the SIMPil design on a set of important image applications. A target SIMPil system is described, which is capable of operating in the Tops/sec range in Gigascale technology.. Simulation results indicate sustained operation throughput in the range of 100-1000 Gops/sec. These results support the design choices and suggest that more complex, multistage applications can be implemented to execute at real-time frame rates.

1 Introduction

Large computational workloads and I/O requirements characterize real-time image processing applications. Typical applications may require 10-1000 Gops/s, which cannot be matched by today's general-purpose microprocessor capabilities (0.2-0.7 Gops/s) [1]. In addition, the traditional sequential access to image data further reduces the available performance. When these applications are to operate in mobile, portable devices, the above requirements couple with stringent resource limitations (size, weight, and power) to pose a formidable system design problem. Many SIMD systems [2], [3], [4] have been used for image processing before, but they achieve performance and generality at the expense of I/O coupling and physical size.

Advances in current semiconductor technology have enabled the integration of CMOS image sensors with digital processing circuitry [5]. In a focal plane imaging

system, images are optically focussed into the sensor array, or *focal plane*, and are made available to the underlying processing engine in a single operation. Because large amount of data parallelism is inherent in image processing applications, SIMD (Single Instruction Multiple Data) architectures provide an ideal programming model [6]. The SIMD Pixel Processor (SIMPil) [7], is a focal plane imaging system, being designed at Georgia Tech, which incorporates a specialized SIMD architecture with an integrated array of image sensors.

This paper evaluates the effectiveness of the SIMPil design on a set of important image applications. The objective of this research is to verify the system capability to execute such applications within the inter-frame delay (15-30 ms). A target SIMPil system is described, which is capable of operating in the Tops/sec range in Gigascale technology. Several image processing applications, spanning from spatial and morphological filtering to image compression and analysis, have been implemented and simulated. Simulation results indicate sustained operation throughput in the range of 100-1000 Gops/sec. These results support the design choices and suggest that there is the potential for real-time execution of complex, multistage applications.

The rest of the paper is organized as follows. Section 2 describes the architecture of the SIMPil system and presents a target system for Gigascale integration. Section 3 describes the set of image processing applications implemented on SIMPil. Section 4 presents simulation results and performance evaluation. Finally, conclusions are offered in Section 5.

2 SIMPil System Architecture

The SIMD Pixel Processor (SIMPil) architecture consists of a mesh of SIMD processors on top of which an array of image sensors is integrated. A block diagram for a 16-bit implementation is illustrated in Fig. 1.

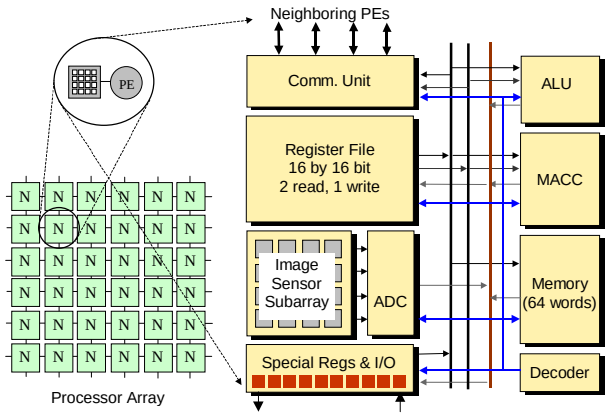

Fig. 1. Block diagram of the SIMPil SIMD system

Each processing element includes a RISC load/store datapath plus an interface to a 4×4 sensor subarray. A 16-bit datapath has been implemented which includes a 32-bit multiply-accumulator unit, a 16 word register file, and 64 words of local memory (the

ISA allows for up to 256 words). The SIMD execution model allows the entire image projected on many PEs to be acquired in a single cycle. Large arrays of SIMPil PEs can be simulated using the SIMPil Simulator, an instruction level simulator, running under Windows95™.

Early prototyping efforts have proved the feasibility of direct coupling of a simple processing core with a sensor device [7]. A 16 bit prototype of a SIMPil PE was designed in 0.8 μm CMOS process and fabricated through MOSIS. The prototypes were successfully tested and run at 25 MHz.

In the rest of the paper, reference is made to a future SIMPil system, integrating 4,096 PEs into a single monolithic device. Table 1 summarizes the most important system parameters [8]. This target system delivers a peak throughput of about 1.5 Tops/sec in a monolithic device, enabling image and video processing applications that are currently unapproachable using today's portable DSP technology.

Table 1. Technology and system parameters for the target SIMPil system

Target VLSI Technology	0.1 μm	Image Sensor Array Size	256×256
Target Clock Rate	500 MHz	Interconnection Network	Torus
Transistor per PE	35 K	Total No. of Transistors (est.)	143.4 M
Image Sensor per PE	4×4	Peak Operation Throughput	1.46 Tops/sec
System Size	64×64	Estimated Power	8.9 W
Chip Size	800 mm^2		

3 SIMPil Application Suite

The SIMPil architecture is designed for image and video processing applications. In general, this class of applications is very computational intensive and requires high throughput to handle the massive data flow in real-time. However, these applications are also characterized by a large degree of data parallelism, which is maximally exploited by focal plane processing. Image frames are available simultaneously at each PE in the system, while retaining their spatial correlation. Image streams can be therefore processed at frame rate, with only nominal amount of memory required at each PE.

To evaluate the set of architectural design choices implemented in the SIMPil system, the following image-processing applications have been implemented and simulated using the SIMPil Simulator. Implementation details can be found in [9].

Spatial filtering. The implementation performs 2D convolution-based filtering. Operations such as shadowing, edge detection, and smoothing are executed using appropriate 3×3-filter masks.

Morphological filtering. Basic morphological operations (erosion, dilation) have been implemented using a 3×3 structuring element. These operations are implemented as intersection and union of shifted versions of the original image. More complex operations, such as opening, closing, inside edge detection, and skeletonization are then implemented by combining the two basic operations.

Vector Quantization. A full-search VQ encoding algorithm has been implemented on the SIMPil architecture. A 256-word codebook is used to achieve a 0.5 bit per pixel encoding of an 8-bpp gray-level image. Compression is achieved by subdividing

the image into small blocks (e.g. 4x4 pixels) and finding the best match for each block among the available codewords. Two levels of parallelism are exploited in this implementation, via codebook distribution among the PEs and by processing multiple input blocks in parallel.

Wavelet decomposition. Discrete wavelet decomposition has been implemented for fingerprint compression and archival. Standard Daubechie's filters have been used to implement the low/high pass filters. A row-column scheme decomposes a gray-level image into 61 frequency bands.

Image rotation. A parallel rotation algorithm has been implemented to perform fast rotations of binary images. The rotation angle γ is first expressed as in (1)

$$\gamma = \alpha + n \cdot \frac{\pi}{2}, \qquad \alpha \in \left[0, \frac{\pi}{2}\right), \qquad \text{and } n = 0 \ldots 3 . \tag{1}$$

Rotations are then executed in two stages: a skew-based rotation of the angle α, and then a set of n fast ninety-degree rotations. This scheme is well suited for a SIMD implementation with regular communication patterns.

Image labeling. This implementation is based on a cluster analysis algorithm. It is used to classify objects in a binary image on the basis of object diameter. The objects are then labeled accordingly.

Quadtree region representation. This implementation operates on binary images to generate a quadtree representation. Quadtrees are based on the principle of recursive decomposition of space. The image is first decomposed in four equal-sized quadrants. If a quadrant is not uniform (entirely filled/empty), it is further decomposed in four more subquadrants. The decomposition stops when uniform quadrants are encountered, or the quadrant contains a single pixel.

These application kernels cover different aspects of the image-processing domain (image compression, image analysis, and image restoration), and they can be staged as components of larger full-featured applications. Larger applications, such as JPEG image encoding, region autofocus and region clustering are currently being implemented, integrating various subcomponents into single larger applications.

4 Results

The applications described in the previous section have been developed and simulated on the SIMPil Simulator. Table 2 summarizes application performances for all the applications implemented. A 64×64 system was used for all the implementations, with the only exception of the image labeling application, for which a 32x32 system was used. A standard NEWS interconnection network was used for all applications with the exception of VQ, for which a toroidal wrapped network was used. In the design of the SIMPil system, memory is traded off for higher throughput. Only a modest amount of memory is available per PE. This choice does not impair SIMPil performance. Both the values of memory used per PE and total system memory indicate that less than 100 memory words are required for most applications. The total system memory size does not exceed 1 MB in almost all cases.

System utilization is calculated as average concurrency, and it is relative to the total system size. Execution time and sustained throughput are computed with

reference to the 500 MHz target platform described in section 2. For each application, the worst case scenario was used. Maximum size images were used in almost all cases. For the image labeling applications, a large number of small objects was used. For the image rotation by skew transforms, the total number of cycles refers to a rotation of 89°.

The simulations show that all applications execute in lower millisecond range, suggesting that they can combined in stages to form complex applications, which still execute at full frame rate (30-60 fps, or 15-30 ms). In most cases, large system utilization is recorded, which in turn provides for a large sustained throughput. The low system utilization for the ninety degree ring rotation algorithm is mainly due to the staged communication patterns used.

Table 2. Application performance comparison (SF: Spatial Filtering, IED: Inside Edge Detection, SKL: Skeletonization, VQ: Vector Quantization, WLT: Wavelet Decomposition, LBL: Image Labeling, QTREE: Quad Tree Decomposition, ROT Skew: Skew-based Rotation, ROT Ring: 90° Ring Rotation)

Application	Memory Size		Total Cycles	System Utilization (%)	Execution Time (μs)	Sustained Throughput (Gops/s)
	PE (16b word)	System (KB)				
SF	62	507.9	11,228	81.6 %	22.46	1,671.2
IED	20	163.8	747	94.2 %	1.49	1,929.2
SKL	20	163.8	14,169	99.3 %	28.34	2,033.7
VQ	51	417.8	81,774	91.7 %	163.55	1,878.0
WLT	56	458.7	23,300	68.5 %	46.60	1,402.9
LBL	86	176.1	170,281	43.9 %	340.56	224.8
QTREE	13	106.5	331,479	87.4 %	662.96	1,790.0
ROT Skew	134	1,097.7	580,636	57.5 %	1,161.27	1,177.6
ROT Ring	39	319.5	10,558	34.5 %	21.12	706.6

In general, the impact of communications across applications is contained within 10% of the total execution time. This illustrates the fact that most applications require limited communications and spend most of their time in local operations. In this, they mostly benefit from the focal plane input of the image data, which distributes the image to each PE in a single operation. A larger amount of communications characterizes the wavelet decomposition application (26%). In this case, a large number of data transfers are needed between different application stages, to rearrange data in preparation of the next stage. A different network organization could be used to reduce the amount of communications required. In general, a good balance between scalar and vector operations is shown by all applications (17-40%). The large degree of data parallelism in the application set is demonstrated by the larger slice of total execution time occupied by vector operations.

5 Conclusions

The demand for real time image processing applications for portable, battery-operated, hand-held devices has grown rapidly in recent years, and it has motivated the research on processing architectures that support focal plane data. The SIMPil

processor has been designed to offer the required computational power and focal plane integration in a compact, monolithic device.

In this paper, a target SIMPil system for Gigascale technology is introduced, and the effectiveness of its design is evaluated for a set of important image processing applications. The application set is described, and performance analysis is given. Simulation results indicate sustained operation throughput in the range of 100-1000 Gops/sec and large system utilization for all implemented applications. Moreover, a balanced hardware resource allocation is shown. These results support the design choices made and suggest that more complex, multistage applications can be implemented on SIMPil to execute at real-time frame rates.

This work was supported in part by NSF, DARPA, and ARL.

References

1. D. S. Wills, J. M. Baker, H. H. Cat, S. M. Chai, L. Codrescu, J. L. Cruz-Rivera, J. C. Eble, A. Gentile, M. A. Hopper, W. S. Lacy, A. Lòpez-Lagunas, P. May, S. Smith, T. Taha, "Processing Architectures for Smart Pixel Systems", *IEEE Journal of Selected Topics in Quantum Electronics*, v.2 n.1, April 1996, p.24-34
2. K. E. Batcher, "Design of a massively parallel processor", *IEEE Transactions on Computers*, pp. 836-840, Sept. 1980
3. T. Blank, "MasPar MP-1 Architecture", *Proceedings of the 35th IEEE Computer Society International Conference*, San Francisco, California, pp.20-24, 1990
4. W. F. Wong, K. T. Lua, "A preliminary evaluation of a massively parallel processor: GAPP", *Microprocessing Microprogramming*, v.29, no. 1, pp. 53-62, 1990
5. E. Fossum, "Digital Camera System on a Chip," *IEEE Micro*, pp. 8-15, May 1998.
6. K. Diefendorff and R. Dubey. "How Multimedia Workloads Will Change Processor Design." IEEE Computer, Vol. 30, No. 9, September 1997, pp. 43-45.
7. H. H. Cat, A. Gentile, J. C. Eble, M. Lee, O. Vendier, Y. J. Joo, D. S. Wills, M. Brooke, N. M. Jokerst, A. S. Brown: "SIMPil: An OE Integrated SIMD Architecture for Focal Plane Processing Applications", *Proceedings of the Third IEEE International. Conference on Massively Parallel Processing using Optical Interconnection* (MPPOI-96), Maui, Hawaii, pp. 44-52, 1996
8. S. M Chai, A. Gentile, D. S. Wills, "Impact of power density limitation in Gigascale Integration for the SIMD Pixel Processor", to appear in *Proceedings of the 20th Anniversary Conference on Advanced Research in VLSI*, Atlanta, Georgia, 1999.
9. A. Gentile, J. L. Cruz-Rivera, D. S. Wills, L. Bustelo, J. J. Figueroa, J. E. Fonseca-Camacho, W. E. Lugo-Beauchamp, R. Olivieri, M. Quiñones-Cerpa, A. H. Rivera-Ríos, I. Vargas-Gonzáles, M. Viera-Vera, "Image Processing Applications on the SIMD Pixel Processor", *Technical Report*, GT-PICA-005-98, Pica Lab, Georgia Institute of Technology, Jun. 1998