

EVALUATING THE TERASYS SYSTEM FOR IMAGE PROCESSING TASKS

Patrick D. Krolak* Patrick G. Mullins†

Center for Productivity Enhancement
University of Massachusetts Lowell
Lowell, MA 01854 U.S.A.

ABSTRACT

We discuss the evaluation of a recently designed novel architecture, called TERASYS for image processing. Our research covers the implementation of various benchmarks; implemented in a new language called data-parallel bit C, (dbC). We explore the TERASYS architecture, the dbC programming environment and report the results of an evaluation of applicability of both to image processing tasks. An initial performance comparison between TERASYS and other parallel architectures is presented. Direction of future efforts is also outlined.

1 INTRODUCTION

The field of image processing in general, provides a strong incentive for massively parallel processors, because some of the characteristics of image processing problems (e.g. voluminous data, data layout) are often ideally suited to implementation on these machines. Presently, there exists a variety of image processing architectures of the SIMD massively parallel type, e.g. CLIP7A, AIS5000, SLiM, PCLIP-II, [1, 2, 3, 4].

Regardless of the type, image processing systems must have fast execution, great price performance and a broad portfolio of image processing capabilities. Up to a decade ago, it was difficult to make the critical decision of choosing an image processor system. It proved unwise to rely on the vendors of these machines to provide an evaluation of their machines, because they quoted only the most favorable performance statistics. Consequently, researchers have devised ways of benchmarking these image processing systems in order to create a more truthful evaluation. Benchmarking attempts to produce, in some fashion, a figure of merit from which the performance of a particular parallel processing system may be judged, for the further purpose of justifying a particular design, [5].

In our work, we focus on evaluating a novel architecture called TERASYS for image processing tasks. TERASYS is a massively parallel processing system that belongs to the SIMD (single instruction, multiple data) category. Our ongoing research involves the implementation of various benchmarks on TERASYS and the evaluating its performance. Our initial work on implementing the *Abingdon Cross Bench-*

mark is discussed in this paper. Furthermore, our research covers the comparison of TERASYS with other image processing architectures, such as the GAPP-II, MasPar MP-1216 and AIS5000.

The rest of this paper is organized as follows. Section 2 provides a short description of the TERASYS system. In addition, it discusses some aspects of the dbC programming environment. Section 3 briefly discusses our current implementation of the benchmark, preceded by an outline on the image processing benchmarks that will be covered in our research. In Section 4, we present the results of running the application on TERASYS. In addition, we include a performance comparison between TERASYS and other parallel architectures. Finally, Section 5 summarizes our work and discusses possible future work.

2 TERASYS SYSTEM AND dbC PROGRAMMING ENVIRONMENT

This section describes the architecture of TERASYS and presents the overall system and structure of the processing elements. In addition, it comments on the the dbC programming environment available on TERASYS. Most of the information presented here is taken directly (with permission) from the TERASYS reference manual, [6] and the dbC reference manual, [7]. The reader may refer to these manuals for more information. TERASYS is a novel memory architecture developed at the Supercomputing Research Center¹. TERASYS is a SIMD (single instruction, multiple data) machine that utilizes special processor chips, called PIM or Process-in-Memory, chips. The main components of the TERASYS system are: (1) Sun SPARC-2 front end workstation, (2) Processor Array, (3) Sbus Interface and (4) PIM Interface board. Figure 1 displays a simple sketch of these components.

2.1 TERASYS ARCHITECTURE

All programs are stored on the SPARC-2 front end machine. The SPARC-2 workstation is used for code development and image display. Communication between the SPARC-2 front end and the PIM chip cards, is performed through the *SBus*. Packed commands are issued from the SPARC-2 at 280 nsec intervals. The TERASYS interface board splits them into two commands, and issues them to the PIM chip cards. Therefore, the command issue rate between the SBus interface and the PIM interface is 140 nsec per command.

¹The Supercomputing Research Center is located in Bowie, Maryland.

*Patrick D. Krolak is professor of Computer Science at the University of Massachusetts Lowell, and is director of the Center for Productivity Enhancement at the university.

†Patrick G. Mullins is a research associate at the Center for Productivity Enhancement.

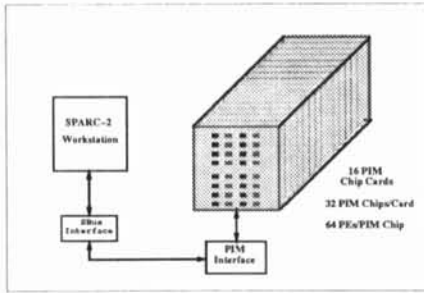


Figure 1: The TERASYS Schematic

2.2 PROCESSOR ARRAY

The processor array is made up of pipelined Processor-in-Memory (PIM) chips, [8]. The PIM chip is a memory chip with additional logic on chip for 64 one-bit processing elements. The current system has 64 processing elements (PEs) per PIM chip, 32 PIM chips per card and 16 PIM chip cards; for a total of 32,768 PEs.

Processing Elements - PEs Each processing element (PE) is tightly coupled to its own private (local) memory and can directly read and write to a 2048 bit column of attached memory. The PE is a programmable bit-serial processor that is essentially divided into an upper and lower half. The upper half performs the actual computations on the data, while the lower half performs routing and masking operations. A simplified version of the PE is shown in Figure 2. There are

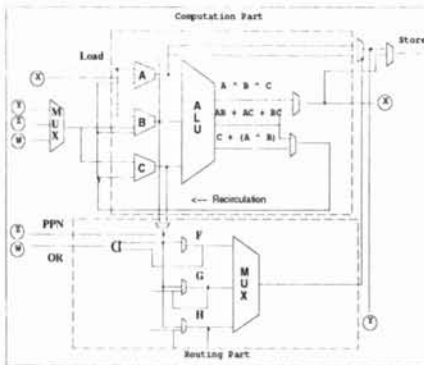


Figure 2: A Processing Element

three primary registers for each processor, denoted the A, B, and C registers. These registers feed an Arithmetic Logic Unit (ALU) which performs the bit serial instructions. The registers have three primary input lines from which to receive data, each of which also can be inverted to receive the logical not of that input. The PE can input data through a MUX from either the parallel prefix network, the global OR network, or the internal mask/register control. Data are brought in from the processors' attached memory through the load line, circulate through the logic as specified by the program, and are written back to memory via the store line. The PE is fully pipelined. At each clock cycle the processors can either load data from memory or store data to memory, but not both at the same time. Also, on each clock cycle, the ALU produces three outputs that either can be selected for storage (under mask control) or selected for recirculation. Additionally, data can be sent to other processors via the

routing network.

2.3 dbC PROGRAMMING ENVIRONMENT

The programming language implemented on TERASYS is data-parallel bit C, (dbC). dbC is a superset of ANSI C that has been extended to include programming languages supporting SIMD programming language constructs. Some of the dbC features include, but not limited to:

- Parallel and serial statements can be intermixed within a program.
- Arbitrary length operations on data parallel operands is possible.
- Parallel user-defined types, structures, and unions can be used.
- Parallel control constructs such as while loops, for loops and if-else are supported.
- The user can call predefined interprocessor communication intrinsics.

dbC supports a simple, easy-to-use programming model so that programmers can become productive rapidly on TERASYS. It allows efficient data parallel computation on TERASYS and on other SIMD machines, such the Connection Machine CM-2.

dbC uses intrinsic functions to perform host-processor, processor-host, and interprocessor communications. The intrinsics allow for greater language flexibility in supporting programmer defined processor array configurations than can be accomplished with specific statements in the language. dbC supports two forms of interprocessor communication. Interprocessor communication can be done using general router communication or nearest neighbor (net) communication. Intrinsic functions exist for each of these communications. Nearest neighbor communication involves a network that uses a toroidal wrap around the edges. This form of communication is accomplished with the use of the Parallel Prefix Network (PPN). General router communication permits any processor to send or receive data from any other processor. Generally, router communication is slower and more expensive than nearest neighbor communication and, therefore, the user should use the nearest neighbor communication network in preference to this method whenever possible.

2.3.1 SIMD INSTRUCTION CODE

dbC translates all code involving parallel operands into a generic three address memory-to-memory SIMD assembly code. The generic SIMD is mapped to the hardware of TERASYS using TWIST. TWIST is an acronym for the Terasys Workstation Interface Software Tools, [9]. The TWIST microcode libraries allows complete low-level access to the parallel architecture of TERASYS. The libraries perform basic operations such as:

- parallel memory allocation and release
- basic arithmetic and logical operations on poly operands of arbitrary bit length,
- nearest neighbor communication and generalized communication.

- data transfer between host memory and PIM memory.

3 EVALUATION OF TERASYS FOR IMAGE PROCESSING

Our research covers the implementation of various image processing benchmarks. In this section we briefly describe these benchmarks followed by a description of our current work.

3.1 THE ABINGDON CROSS BENCHMARK

The Abingdon Cross Benchmark, [10, 11, 12], devised during the Abingdon Workshop in 1982 is used to benchmark image-processing systems. The Abingdon Cross Benchmark has become the *de facto* standard among many researchers in the image processing community. It has proven to be an initial step in the direction of rigorous benchmarking. It is not a specific algorithm, but finds the medial axis of a cross-shaped structure buried in noise. (see Figure 3 (a)). Kendall

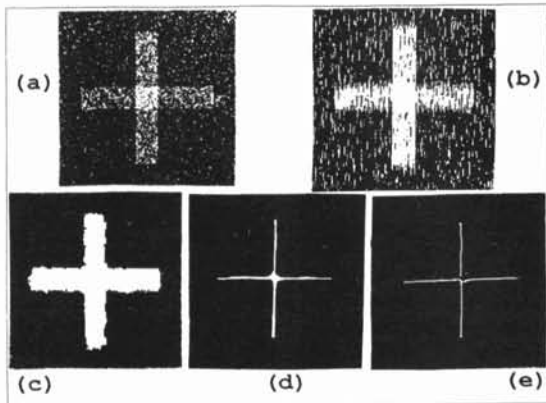


Figure 3: Abingdon Cross Benchmark solution

Preston² at Carnegie Mellon University has spent the last decade gathering benchmark results from researchers in the image processing community. More than 70 groups have submitted results on the performance of the benchmark on various architectures.

3.2 OTHER BENCHMARKS

The Preston-Seigart Benchmark is a Carnegie Mellon University (CMU) successor to the Abingdon Cross Benchmark. It was developed by Kendall Preston and Carol Seigart, out of the growing concern that the Abingdon Cross Benchmark was unfair, in some respects, [12]. The Image Benchmarking Toolkit (IBT) is a proposed standard for image processing benchmarking. The tool was proposed as a standard by the Image Benchmark Technical Work Group, (IBTWG). We are using proposed Standard 0.1. The Image Understanding Benchmark (IUB) was developed through the cooperative effort of the University of Massachusetts Amherst and the University of Maryland. It is based on the DARPA Image Understanding Workshop, 1987, where vision experts felt that standard imaging benchmarks (such as the Abingdon Cross Benchmark) did not address the key issues facing image recognition and understanding.

²Author of the Abingdon Cross Benchmark.

3.3 IMPLEMENTATION

In the proposed benchmark, *a priori* knowledge is considered: The dimensions of the cross are known; the midrange of the gray levels is also known. Our implementation of the benchmark includes two main factors: Data Layout Model and Image Processing (IP) operations. The initial image data is grey-level data (i.e. each pixel is represented by eight bits of data). In setting up the memory for the implementation, we focused on the problem of mapping of a 2-D image to the 1-D processor array of TERASYS. We make the following assumptions:

- The image is square (256×256 pixels)
- # of pixels = $k \times$ # of processors for some integer $k > 1$
- # of processors = $j \times \sqrt{\#pixels}$ for some integer $j > 1$

Each processor is assigned a segment of a column of pixels. The segment size $s = \frac{\#pixels}{\#processors}$.

The basic IP operations used in our implementation are: (1) Vertical Averaging, (2) Horizontal Averaging, (3) Erosion and (4) Skeletonization. The first step computes a 1×13 vertical averaging on a binarized cross, [Figure 3 (b)]. The second step computes 13×1 averaging of the cross, [Figure 3 (c)]. The third step employs 13 iterations of erosion on the cross, [Figure 3 (d)]. Erosion is a form of thinning. Thinning is an image processing operation in which binary valued image regions are reduced to lines that approximate the center skeletons of the region, [13]. It is required that the lines of the *thinned* result are connected for each single image region. Finally, in the fourth step, a thinning (skeletonization) process reduces the image to single-line thickness, while preserving connectivity, [Figure 3 (e)].

4 RESULTS

The Abingdon Cross Benchmark on TERASYS is evaluated from point of view of two criteria; *Quality Factor* and *Price Performance Factor*. The Quality Factor is the ratio of the size of the image processed divided by the execution time. The Price Performance Factor is the total number of pixels processed divided by the product of the execution time and the price of the system in US dollars. These two criteria are used in the comparison of TERASYS with other Image Processing (IP) systems.

4.1 TIMINGS

Benchmark Operations	Number of Iterations	Time (ms)	Percentage of Total
Vertical Averaging	1	1	7.8
Horizontal Averaging	1	3	23.2
Erosion	13	4	30.4
Skeletonization	2	5	38.6
Total	17	13	100%

Table 1: Benchmark results on TERASYS

Table 1 shows the timings of running the benchmark, using 32768 processors and an image size of 256×256 pixels. The

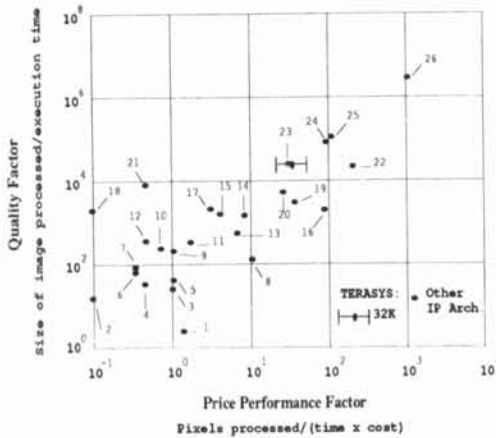


Figure 4: Abingdon Cross Benchmark Results

Graph Number	Architecture
*	TERASYS (32K)
1	T800 (Scottish Regional Transputer Support Center)
2	Magiscan-2 (Joyce-Lobel) QF = 10)
3	T800 Array (Scottish Regional Transputer Support Center)
4	THAPIX 5500 (Recognition Concepts Inc.)
5	PIP4000 (ADS Company Ltd.)
6	PNICOM 327 (Perceptive Systems)
7	IP8500 (ETH Zurich)
8	MVP/AT (Matrox) (QF = 100)
9	TOSPIX II (Toshiba)
10	TAS-Plus (Leitz GmbH)
11	TEHACON (Irragon)
12	IP9200 (Perceptics)
13	VICOM VME-II (Vicom)
14	PIXAR/1-Chal? (Vicom-Pixar) (QF = 1000)
15	Scope-20 (Symbolics)
16	MP150 (Noesis Vision)
17	VItec-1 (Visual Information Technologies)
18	CM-2 (Thinking Machines - using C*)
19	MaxVideo (DataCube)
20	DAP 510 (Active Memory Technology)
21	CM-2 (Thinking Machines - using PARIS)
22	AIS5000 (Applied Intelligent Systems)
23	Zephyr-8 (Wavetracer)
24	MP-1208 (MasPar)
25	MP-1216 (MasPar)
26	GAPP-II (Martin Marietta)

Table 2: Architectures in Figure 4

Quality Factor is $256/0.013 = 19,692$. The TERASYS system is a non-commercial and so, it was difficult to establish a specific price for the system. We estimate that the price of the system ranges over the values \$75,000 and \$150,000. Therefore, the Price Performance Factor ranges over the values $256^2/(0.013 \times \$150,000) = 33.6$ and $256^2/(0.013 \times \$75,000) = 67.2$.

4.2 TERASYS VERSUS OTHER IP ARCHITECTURES

The performance of the Abingdon Cross Benchmark on TERASYS compared with other image processing systems is shown in Figure 4 with its legend given in Table 2³. Accounting for the range in Price Performance Factor, the reader may observe that the performance of TERASYS is competitive with the leading machines.

³The graph contained in this paper is a modification of the results published by Kendall Preston and is used with permission of **Advanced Imaging**, copyright Sept. 1992.

5 SUMMARY AND FUTURE WORK

We have briefly described a SIMD machine (TERASYS) and the fundamentals of a high level language (dbC) for it. In addition, we focused on the benchmarking of TERASYS for image processing tasks using the Abingdon Cross Benchmark. We have shown that TERASYS has performed tremendously well with respect to the image processing benchmark. In comparison to other image processing systems, this paper shows that TERASYS has great potential as being a viable architecture for low level image processing tasks. The language dbC appears to be able to support convenient programming models with very efficient mappings to the hardware. In our future work, we are interested in implementing the more rigorous benchmarks: *IBT* and *IUB*. We believe that TERASYS is suitable for other application areas. Therefore, we are looking other applications. Specifically, we are in the process of implementing cellular automata applications on TERASYS. [14].

Acknowledgments We gratefully acknowledge the support from Supercomputing Research Center contract SRC # 06750 and the assistance of Bill Holmes, Maya Gokhale, and Howard Gordon of the Supercomputing Research Center. We would like to thank Linda Wilkens, David Woods and Aaron Euright for their contributions to earlier stages of this work. We are indebted to the anonymous referees for their reviews of an earlier draft of this paper.

References

- [1] Fountain, Terry J. et al. The CLIP7A Image Processor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(3):310-319, May 1988.
- [2] Schmitt, Lorenz A. and Wilson, Stephen S. The AIS-5000 Parallel Processor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(3):320-330, May 1988.
- [3] Sunwoo, Myung Hoon and Aggarwal, J.K. A sliding Memory Plane Array Processor. *IEEE Transactions on Parallel and Distributed Systems*, 5(6):601-612, June 1993.
- [4] Pfeiffer Jr., Joseph J. Design Considerations for a Pyramidal Cellular Logic Processor. In *The 2nd Symposium on the Frontiers of Massively Parallel Computations*, pages 511-514. Piscataway, NJ, 1988. IEEE.
- [5] Duff, M.J.B. How not to Benchmark Image Processors. In Uhr, L., et al., editor. *Evaluation of Multicomputers for Image Processing*, pages 3-21, Orlando, FL 32887, 1986. Academic Press, Inc.
- [6] K. Iobst and T. Turnbull. Terasys reference manual. Technical Report SRC-TR-93-xxx, Supercomputing Research Center, 1993.
- [7] Judith Schlesinger and Maya Gokhale. dbC Reference Manual. Technical Report TR-92-068, Revision 2, Supercomputing Research Center, 1993.

- [8] H. Doug Sweely. Terasys Demonstration Hardware Manual. Technical report, Supercomputing Research Center, 1993.
- [9] Judith Schlesinger and Dan Kopetzky. Terasys, Microcode, and TWIST. Technical report, Supercomputing Research Center, February 1994.
- [10] Preston, Kendall, Jr. The Abingdon Cross Benchmark Survey. *IEEE Computer*, 8(34):9-18, July 1989.
- [11] Preston, Kendall. Benchmark Results: The Abingdon Cross. In Uhr, L., et al., editor, *Evaluation of Multi-computers for Image Processing*, pages 23-54, Orlando, FL 32887, 1986. Academic Press, Inc.
- [12] Preston, Kendall, Jr. Benchmark for Image Processing. *Advance Imaging*, 5(5):30-37, May 1990.
- [13] O'Gorman, Lawrence. $k \times k$ Thinning. *Computer Vision, Graphics and Image Processing*, 51:195-215, 1990.
- [14] Patrick G. Mullins and Patrick D. Krolak. A Discrete Simulation of 2-D Fluid Flow on Terasys. Technical Report in progress, University of Massachusetts at Lowell, 1994.