PERFORMANCE OF THE ASP
ON THE DARPA ARCHITECTURE BENCHMARK

A Krikelis, R M Lea

Aspex Microsystems Ltd
Brunel University
Uxbridge, United Kingdom, UB8 3PH

ABSTRACT

The Associative String Processor (ASP) is a homogeneous, reconfigurable and programmable, massively parallel processor which offers step-function advantages in cost-performance and application flexibility, due to its unique architecture and its exploitation of state-of-the-art microelectronics. This paper briefly describes the ASP architecture, its implementation and reports the results of an evaluation of its applicability to image processing tasks. In order to provide a realistic demonstration of the above-mentioned advantages, a set of independently defined such tasks (viz. the DARPA Image Understanding benchmark) was chosen for the evaluation and the results are used to compare the performance of the ASP architecture with the performances of other parallel computer architectures when applied to the same computer vision tasks.

INTRODUCTION

Comparing different parallel processing architectures is a very difficult task. Most commercial purveyors promote their machines by quoting only the most favourable performances. Moreover, analysis of parallel algorithms and systems shows that there are always overheads, detracting from performance, which are rarely quoted. Indeed, it is commonly accepted that users can expect parallel processors to provide a speed-up of only $O(\log N)$, where $N$ is the number of processing elements.

The field of image processing in general, and computer vision in particular, provides a strong incentive for massively parallel processors; due to the large data volume, high data-rate and algorithmic complexity of its computational tasks. Indeed, researchers, involved in the areas of algorithm and system development for real-time image understanding, need high performance which is easy to use (including programming) and cost-effective. Not surprisingly, therefore, the image processing workers were among the first to attempt the establishment of a realistic benchmark for massively parallel processors. Early computer vision benchmarking attempts included the Abingdon Cross problem (1982) and the Tanque Verde benchmark suite (1984).

A more recent attempt to construct a computer vision benchmark emerged from the DARPA Image Understanding community in 1986, when the University of Maryland defined a set of representative low and intermediate-level vision tasks [1]. High-level vision (such as recognition) were not included, because it was felt that proposed algorithms were too ill-defined to properly evaluate parallel architectures.

The benchmark was intended to achieve an initial understanding of the general strengths and weaknesses, for computer vision applications, of the growing number of parallel computer architectures and to project the need for future development of parallel architectures to support this field.

This DARPA benchmarking activity has been the most successful to date. Moreover, a second DARPA Image Understanding benchmark suite, based on the experience of the first benchmark and defined by the University of Massachusetts in collaboration with the University of Maryland, has been recently announced [2].

This paper reports the results of the evaluation of the Associative String Processor (ASP) [3], a massively parallel processor emerging from research at Brunel University and being developed by Aspex Microsystems Ltd., for the first DARPA benchmark.

THE BENCHMARK

The first DARPA Image Understanding benchmark, defined in reference [1] and discussed in reference [4], includes the following computer vision tasks.

A. Edge detection within a 512 x 512 pixel image
   A1. 11 x 11 Laplacian
   A2. zero crossing detection
   A3. border following.

B. Connected component labelling within a 512 x 512 pixel image.

C. Hough transform computation within a 512 x 512 pixel image.

D. Geometrical constructions for a set of 1000 planar points.
As indicated in Figure 1, an ASP system [3] comprises a dynamically reconfigurable parallel processing structure of communicating ASP sub-strings, each supported with an ASP Data Buffer (ADB), an ASP Controller and an ASP Data Communications Network.

Each ASP substring is a parallel processing computational structure, comprising a string of identical APEs (Associative Processing Elements), as shown in Figure 2. Each APE is connected to an Inter-APE Communication Network (which runs in parallel with the APE string). All APEs share common bit-parallel Data, Activity and Control Busses and a single feedback line (Match Reply, MR), which are maintained by an external ASP Controller, which also maintains the Link Left and Link Right ports (LKL and LKR) of the Inter-APE Communication Network.

Each APE incorporates an n-bit Data Register and an a-bit Activity Register, an (n+a)-bit parallel Comparator, where the values of n and a are in the ranges 32-128 and 4-8 respectively, depending on the application class for which the ASP is optimised. Moreover, the APE includes a single-bit full-adder, 4 status flags (viz. C to represent arithmetic Carry, M and D to tag Matching and Destination APEs and A to activate selected APEs) and control logic for local processing and communication with other APEs.

In operation, each ASP substring supports a form of set processing, in which the sub-set of active APEs (i.e. those which match broadcast data and activity values) support scalar-vector and vector-vector operations. The Match Reply (MR) line indicates whether none or some APEs match. Matching APEs are either directly activated or source inter-APE communications to indirectly activate other APEs.

Scalar data are directly broadcast or received by the ASP controller via the bit-parallel Data Bus. Input-output vector data could also be exchanged (viz. output dumped and input loaded in a single step) APE-sequentially via the Data Bus with the bit-parallel Primary Data Exchanger (PDX) shown in Figure 2. However, the Vector Data Buffer supports a much faster APE-parallel exchange facility, in which the bit-serial Primary Data Exchanger (PDX) performs the task at a very high data rate, thereby minimising loss of parallel processing efficiency. Similarly, but at a lower data-rate, the Secondary Data Exchanger (SDX) provides a bit-parallel vector data exchange between the Vector Data Buffer and the external ASP Data Buffer (ADB), which is overlapped with parallel processing and, therefore, does not present a sequential processing overhead.

The Inter-APE Communication Network implements a globally-controlled and dynamically-reconfigurable tightly-coupled APE interconnection strategy, which supports cost-effective emulation of common network topologies with two modes of inter-APE communication:

- Circuit-switching: asynchronous bi-directional single-bit communication via multiple signal paths, dynamically configured (programmer-transparently) to connect APE sources and corresponding APE destinations of high-speed activation signals, implementing a fully-connected permutation and broadcast network for

![Figure 1. ASP system](image1)

![Figure 2. ASP substring](image2)
APE selection and inter-APE routing functions

packet-switching: synchronous bi-directional multi-bit communication via a high-speed bit-serial shift register, routing M-bit tag patterns along each APE substring, for data/message transfer.

In order to preserve continuity at the two ends of the Inter-APE Communication Network, the LKL and LXR (shown in Figure 2) allow activation or M-bit signals to be injected and sensed by the external ASP controller and act as the left and right neighbours of the leftmost and rightmost APE in the associative string processor respectively.

ASP IMPLEMENTATION

The ASP concept is particularly well matched to both the opportunities and constraints of VLSI chip fabrication; owing to its high APE packing density, its highly compact inter-APE communications network and, especially, because its I/O requirement is independent of the string length. The feasibility of a 256-APE VLSI ASP chip was demonstrated in 1986 and Aspex Microsystems are developing 256-APE VLSI ASP chips for ASP substring implementation [3].

Moreover, the ASP is highly amenable to defect/fault-tolerance; owing to its construction from a large number of identical APES, lack of location-dependent addressing and simple inter-APE interconnection. Consequently, as reducing feature-sizes and increasing chip sizes drive VLSI chip fabrication technology towards the prospect of ULSI chips and WSI devices, the ASP architecture offers consistency and becomes increasingly more cost-effective. Indeed, research at Brunel University has indicated the potential integrating complete ASP systems with 2,048-APE ULSI chips and 8,192-APE WSI ASP devices.

EVALUATION

In practice, ASP system configurations may be tailored to suit application requirements; the minimum number of APES being 256 (see above) and the maximum being limited by implementation cost.

Two ASP system configurations were chosen for the DARPA benchmark evaluation [4]:

<table>
<thead>
<tr>
<th>DARPA ASP:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of ASP substrings = 512</td>
</tr>
<tr>
<td>Number of APES per substring = 512</td>
</tr>
<tr>
<td>Data Register storage = 96 bits</td>
</tr>
<tr>
<td>Activity Register storage = 5 bits</td>
</tr>
<tr>
<td>Implementation complexity</td>
</tr>
<tr>
<td>32 WSI ASP devices</td>
</tr>
<tr>
<td>or 128 ULSI ASP chips</td>
</tr>
<tr>
<td>or 1,024 VLSI ASP chips plus data communication network, ADB and ASP controller boards</td>
</tr>
<tr>
<td>Clock rate = 20 MHz</td>
</tr>
</tbody>
</table>

| ASP: |
| whichever ASP system configuration offers optimum performance for the particular benchmark task. |

Assuming a clock rate of 20 MHz, the benchmark evaluation results for these configurations are reported in Table 1.

Comparison of the results of Table 1 with those of Tables 2 through 4 indicates the consistency of the performance advantage of the ASP. It is interesting to note the superiority of the two associative architectures. Unfortunately, the DARPA benchmark neglects volume and cost factors, for which highly-compact 1000MOPS/$1000 ULSI/WSI ASPs would excel.

OTHER ARCHITECTURES

Brief details of other parallel computer architectures which have been evaluated according to the first DARPA Image Understanding benchmark are given below [4].

Medium-grain MIMD multiprocessors (see Table 2 for the reported benchmark results)

- BB&N BUTTERFLY: 128 shuffle-exchange connected PEs; each 16 MIPS PE comprising a Motorola 68020 32-bit microprocessor, an AMD-2901 8-bit microprocessor for memory management, a custom-designed VLSI switch circuit supporting 32 MBytes/sec inter-PE communication and up to 4MBytes of local memory with a 64MBytes/sec bandwidth. The multiprocessor incorporates 512 I/O channels, each supporting a data rate of 16MBytes/sec operating with a 16MHz clock.

- Caltech CUBE and MOSAIK: 256 and 16,384 hypercube connected PEs; each 8 MIPS PE comprising Intel 80286/80287 32-bit microprocessors and up to 4.5MBytes of local memory with a 32MBytes/sec bandwidth. The multiprocessors are designed to operate with an 8MHz clock.

Medium-grain systolic arrays (see Table 3 for the reported benchmark results)

- CMU WW-WARP and PC-WARP: 10 linearly connected PEs; each 10 MFLOPS PE comprising a wire-wrapped or printed-circuit board 255 chip implementation of input queues, crossbar, 32-bit processing elements (including a floating point processor), register files, 32 MBytes of local memory (with an 80Mbytes/sec bandwidth), address generator and microengine. Inter-PE communication PEs can be achieved at 32 MBytes/sec and the I/O data rate is 40MBytes/sec. The linear arrays are controlled by an interface unit, comprising 264 chips, which can communicate with the host system through two I/O clusters, based on Motorola 68020 microprocessors. The arrays are designed to operate with an 80 MHz clock.

- CMU IWARP: 72 linearly connected PEs; each 16 MFLOPS PE being targeted for integration on a single chip being developed in collaboration with Intel Corporation.
Columbia NON-VON: up to 1M tree-connected PEs; each PE being based on a custom-designed VLSI 1-bit (and possibly 8-bit in version 3) array processor chip. In version 3, the tree communication network also incorporates mesh interconnection between its leaves. The array processor is designed to operate with a 10MHz clock.

Thinking Machines CONNECTION MACHINE: 65,536 mesh connected PEs; each PE comprising a 1-bit ALU, 8 status flags and 4Kbits of local memory with a 4Mbits/sec bandwidth. 16 PEs are implemented with a custom-designed VLSI chip incorporating a 4 × 4 processor array and one router of a 16Kbytes/sec packet-switching hypercube-connected communications network (overlaying the mesh and implemented separately) and 4 16K-bit static RAM chips. The array can support an I/O data rate of 30Kbits/sec/channel operating with a 10MHz clock.

Fine-grain SIMD array processors (see Table 4 for the reported benchmark results)

- UMass IUA (Image Understanding Architecture): 266,304 three-level mesh connected PEs; each PE comprising a 1-bit ALU, 8 status flags and 4Kbits of local memory with a 4Mbits/sec bandwidth. 16 PEs are implemented with a custom-designed VLSI chip incorporating a 4 × 4 processor array and one router of a 16Kbytes/sec packet-switching hypercube-connected communications network (overlaying the mesh and implemented separately) and 4 16K-bit static RAM chips. The array can support an I/O data rate of 30Kbits/sec/channel operating with a 10MHz clock.

Fine-grain MIMD/SIMD array processor (see Table 4 for the reported benchmark results)

- UMass IUA (Image Understanding Architecture): 266,304 three-level mesh connected PEs; each PE comprising a 1-bit ALU, 8 status flags and 4Kbits of local memory with a 4Mbits/sec bandwidth. 16 PEs are implemented with a custom-designed VLSI chip incorporating a 4 × 4 processor array and one router of a 16Kbytes/sec packet-switching hypercube-connected communications network (overlaying the mesh and implemented separately) and 4 16K-bit static RAM chips. The array can support an I/O data rate of 30Kbits/sec/channel operating with a 10MHz clock.

REFERENCES


