Electro-Optical DSP of Tera Operations per Second and Beyond

(Extended Abstract)

Dan E. Tamir[1], Natan T. Shaked[2], Peter J. Wilson[3] and Shlomi Dolev[4]

[1]Department of Computer Science, Texas State University, San Marcos, Texas 78666, USA, dt19@txstate.edu
[2]Department of Electrical and Computer Engineering, Ben-Gurion University of the Negev, P.O. Box 653, Beer-Sheva 84105, Israel, natis@ee.bgu.ac.il
[3]Kiva Design, 104 Forest Trail, Leander, Texas 78641, USA, pete@kivadesigngroup.com
[4]Department of Computer Science, Ben-Gurion University of the Negev, P.O. Box 653, Beer-Sheva 84105, Israel, dolev@cs.bgu.ac.il

Abstract. We present a vector-by-matrix multiplier architecture incorporating the high potential of optical computing within electronics. This architecture stems from advances in optical switching technology, optical communication, and laser on silicon, and overcomes previous bottlenecks implied by the speed of transferring information to the optical vector-by-matrix multiplier. Based on this architecture, we present in detail a feasible electro-optical DSP co-processor that can obtain more than 16-Tera integer operations per second. The use of the new architecture for several principal DSP applications is detailed, showing a significant improvement of at least two orders of magnitudes, over existing DSP technologies. Examples of possible applications, including motion estimation engine, string matching, and geometry engine systems, are provided. In addition, the architecture enables an improvement of previous solutions to bounded NP-complete problems by extensively reducing the size of the solver, while preserving an efficient computation time.

Keywords: Optical Computing, Parallel Processing, Digital Signal Processing, Traveling Sales Person Problem, Bounded NP-Complete Algorithms.

1 Introduction

The need to double the computation speed every *18* months (Moore law) or less still exists. To cope with the frequency limitations of VLSI technologies and still satisfy this need, the microprocessor industry proposes multi-core technologies, essentially suggesting the use of parallel computations. New parallel processing paradigms are being considered and evaluated due to the need for fast communication capabilities among the cores. The revolution in microprocessor architectures can open new opportunities for the use of optical computing. Several new architectures were recently suggested [1-5]. These architectures are based on optical communication

over silicon and utilize the fact that no cross-talks occur in optics, just like the case of the widely-used fiber optics communication.

In this work, we present a new electro-optical architecture that enables a fast vector-by-matrix multiplication. In addition, we present new solutions that enhance the capabilities of this architecture as a co-processor dedicated for DSP-intensive and computationally-intensive applications. The fact that optical matrix multiplication can be carried out in parallel and in a very fast rate must be supported by electronic interface for transferring data to the multiplier in a way that utilizes the multiplier capabilities. Otherwise, the speed of the computation is determined by the data transfer bottleneck.

The operation of a vector-by-matrix multiplication is involved in several computationally-intensive applications such as rendering computer-generated images, beam forming, radar detection, and wireless communication systems. Many of these practical applications require efficient and fast implementation of vector-by-matrix multiplication.

In addition, vector-by-matrix multiplication can be used as a building block for numerous DSP procedures such as convolution, correlation, and certain transformations, as well as distance and similarity measurements [6,7,8]. For example, a discrete Fourier transform (DFT) can be implemented as a special case of vector-by-matrix multiplication.

The electro-optical vector-by-matrix multiplier (VMM) presented in this paper can perform a general vector ($1\times256\times8$ bit) by matrix ($256\times256\times8$ bit) multiplication in one cycle of 8 nanoseconds (that is, at a rate of 125 MHz). This rate is faster than all other VMMs available today and it is expected to improve with the introduction of new electro-optical technologies. In addition, the proposed VMM can serve as a co-processor attached to a DSP or a RISC CPU (referred to as a controller) and significantly enhance the performance of the controller.

Previous commercial electro-optical VMMs (e.g. [1]) were limited to the rate of the electrical driver of the spatial light modulator (SLM) used to represent the VMM matrix. The present paper proposes an alterative architecture where the SLM and its electrical driver are not limited to low rates. Hence, the proposed VMM utilizes the optical advantages more efficiently.

Section 2 introduces the proposed electro-optical design. Section 3 presents possible implementation and provides data that show the design feasibility, as well as explains how this design can be improved along with technology advances. Section 4 presents and analyzes several applications, and Section 5 concludes the current paper.

2. The VMM Electro-Optical Unit

The ability to perform mathematical operations in free space (in the open space, with no wiring), in parallel, and without mutual interactions among the various signals is only a part of the inherent features of optical data processing. These features are utilized in this paper, as well as in many other optical VMM configurations that have been suggested in the technical literature [9-12]. The proposed VMM has two main

components: the optical unit and the electrical driver. The next two sections elaborate on each of these two components respectively.

2.1 The VMM Optical Unit

Several configurations can be utilized to implement the optical component of the VMM. One of these configurations is based on the Stanford multiplier principle [12] illustrated in Fig. 1. As shown in this figure, the input vector of the VMM is represented by a set of light sources, the matrix of the VMM is represented by a slide mask or a real-time SLM and the output (multiplication-product) vector of the VMM is represented by a set of sensitive detectors. The light from each of the sources is spread vertically so that it illuminates a single column in the matrix and then each row in the matrix is summed onto a single detector in the detector array. This VMM configuration can be performed by several optical techniques. One of these techniques uses two sets of lenses. Each of these sets contains a cylindrical lens and a spherical lens and each of these lenses has a focal length of f. A single set of lenses has an equivalent focal length of $f/2$ in vertical/horizontal direction and f in the other direction. As shown in Fig. 1, the first set of lenses is positioned between the input vector (represented by the light sources) and the matrix (represented by the SLM). This set of lenses is positioned so that the light coming from each of the sources illuminates only a single column in the matrix, which means collimating the light diverging vertically from each of the sources, but imaging it in the horizontal direction.

2.2 The VMM Electrical Driver

Figures 2 and 3 present high level descriptions of the electrical components of the system. A possible implementation of the system outlined in these figures is presented in Section 3.

Figure 2 shows the VMM electrical driver. The driver is comprised of *256* single electrical driver (SED) units and has two types of inputs: a $1\times256\times8$ bit input vector A which is the VMM input vector (which is the VCSEL source array driving signal), and a set of *256* vector inputs B_0 to B_{255} (total of $256\times256\times8$ bit). The output of the VMM is a $1\times256\times20$ bit vector C. The VMM output vector C is an aggregation of the set of scalar outputs (C_0 to C_{255}), where the output C_j emerging from SED_j is a single *20*-bit bus which is yielded by the output detector array.

Figure 3 shows the design of one of the $1\times256\times8$ bit SED units. The input vector B_j can be stored in an internal dual-port modular memory buffer / shifter before being directed to the SLM. The output C_j is directed to the external output.

The configuration depicted in Fig. 3 supports a dot-product operation between the input row vector A (being converted into light by the VCSELs) and one column of the entire SLM matrix. Each SED unit performs one vector dot-product operation per cycle of *8* ns (the reciprocal of *125* MHz). Combined together, the *256* units perform a vector ($1\times256\times8$ bit) by matrix ($256\times256\times8$ bit) multiplication operation at a rate of *125* MH.

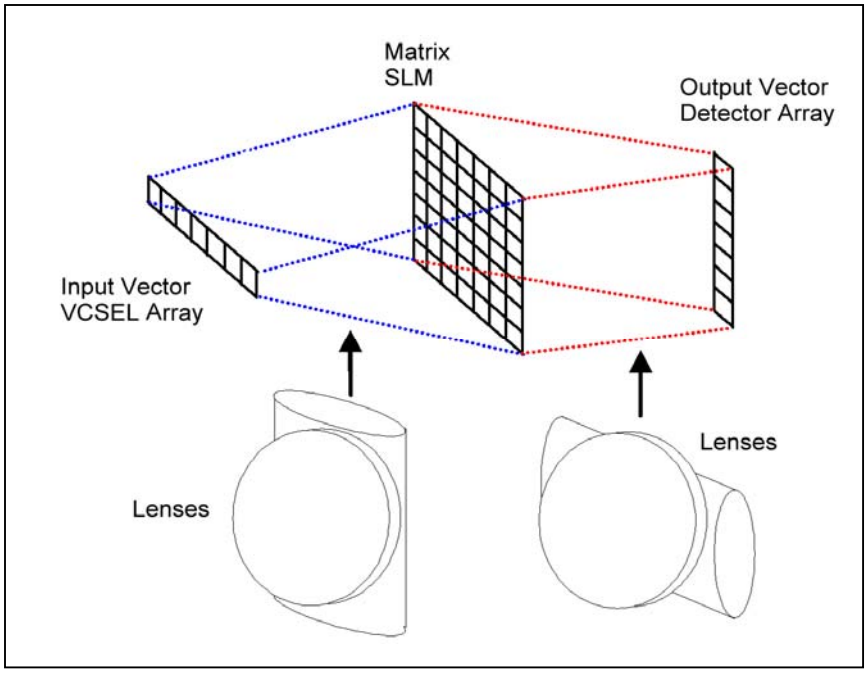

Fig. 1. The Stanford VMM.

Each of the units depicted in Fig. 3 is capable of executing *256×125* million multiply-accumulate instructions per second. This is equivalent to *64* Giga integer operations per second (GIPS). Since the entire VMM system consists of *256* units that are the same as the one unit depicted in Fig. 3, it can perform in peak performance of *16384* GIPS.

The proposed VMM can serve as a co-processor attached to a DSP or a RISC CPU, referred to as the controller. In addition, the controller can utilize other co-processors. The entire system (controller, co-processors and VMM) is depicted in Fig. 4. The figure shows an example where one of the additional co-processors is capable of shuffling vector elements. This may be useful for implementing convolution and correlation. It is assumed that in the typical mode of operation, the controller, along with other co-processors, prepares an input vector and an input matrix to be sent to the VMM for processing. We refer to this as pre-processing. The output of the VMM is sent back to the controller, where it may go through additional processing (referred to as post-processing) before being sent to other devices or back to the VMM. Sound pre-processing and post-processing operations can reduce the amount of VMM operations. For example, reusing the same input vector as much as possible, while altering only some of the matrix values, can reduce the amount of communication between the controller and the VMM and enable the VMM to operate at a peak rate that is greater than *125* MHz.

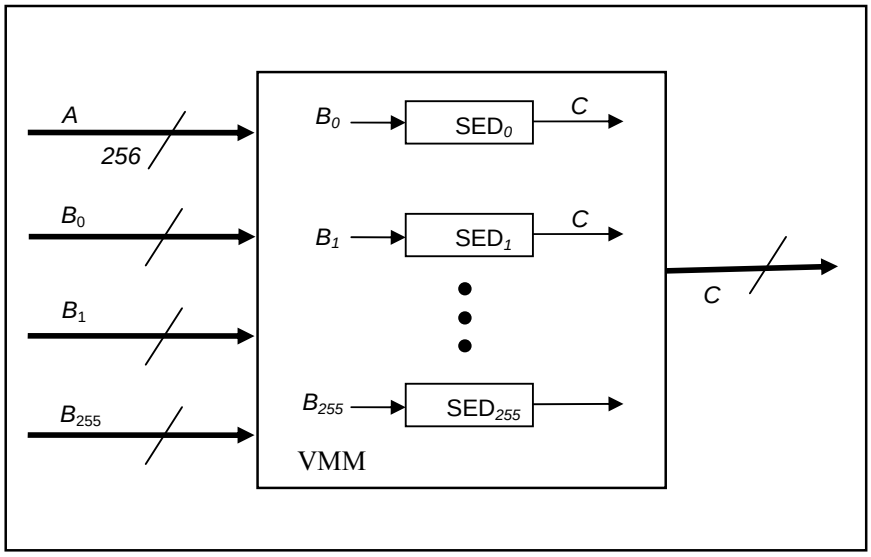

Fig. 2. The VMM electrical driver.

The VMM is capable of performing one generic operation. That is, a vector (*1×256×8* bit) by matrix (*256×256×8*) multiplication. Nevertheless, special cases of vector-by-matrix multiplication can be used as the building blocks for numerous DSP procedures. For example, a vector-by-matrix multiplication with a large number of elements (more than *256* or *256×256* elements), vector-by-matrix multiplication with a small number of elements (less than *256* or *256×256* elements), various matrix-by-matrix multiplications, dot-product operations including convolution, correlation, transforms (e.g. discrete Fourier / cosine transforms), and L2 norm operations, as well as addition, complex, and extended-precision operations. These operations, which are the building blocks of numerous DSP-intensive applications, are further analyzed in a technical report [13].

3. Implementation of the VMM Electrical Driver

Figures 3, 5, and 6 illustrate the components of the VMM electrical driver. These figures include several implementation details showing that it is feasible to construct the system using existing mass-production VLSI technology.

As mentioned above, the driver consists of *256* SED units of *1×256×8* bit each (one of which is shown in Fig. 3). Several SED units are integrated into ALU chips, which are placed on an interface board. A set of SED units integrated into one chip is referred to as the ALU chip (see Fig. 5). The interface board contains several ALU

chips (see Fig. 6). Overall, the interface board contains *256* SED units. There is design flexibility with respect to the number of SED units per ALU chip. This number dictates the required number of ALU chips in the interface board. After analyzing current technology, we have decided to investigate a configuration with *16* ALU chips, each of which contains *16* SED devices. This configuration is further analyzed in the next subsections.

3.1 The SED Unit

Each SED unit contains a relatively small dual-port modular memory of several thousands bytes (say *2048* bytes) and a *256* element SLM (i.e., *256* SLM transistors). The SLM element of SED_j (depicted in Fig. 3) represents row j of the SLM matrix. The input vector B_j can be stored in the internal buffer / shifter before being directed to the SLM. The buffer drives the SLM either directly with the input B_j or with a set of *256* bytes that has been previously stored in the buffer $(\overline{B}_j)$. In this case, the shifter can implement a shift of *1*, *2*, *4*, or *8* bit on one stored row of 256×8 bit, enabling convolution and correlation with bytes and sub-byte units. The output C_j is a single *20*-bit bus which is wired to the output detector array. In the implementation considered here, we assume that the input B_j consists of a *256*-bit bus that operates at *1* GHz. Hence, it loads the 256 SLM elements of the SED or/and the SED buffer at a rate of *125* MByte per second**.**

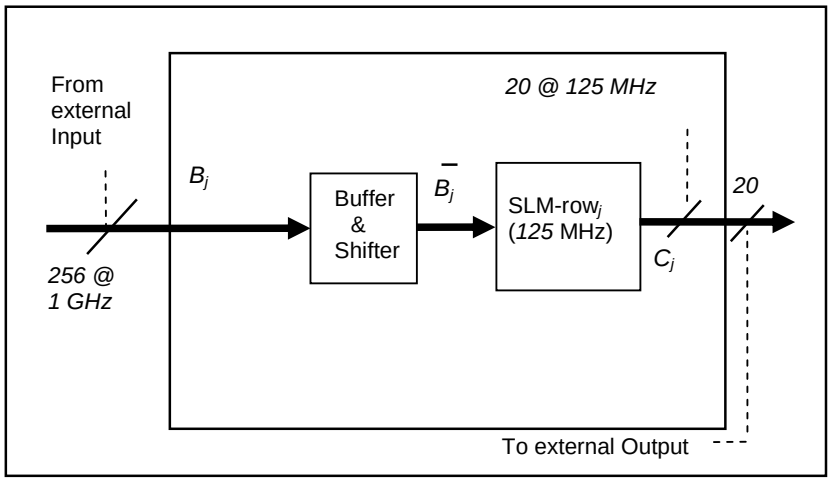

Fig. 3. A *1×256* SLM electrical driver.

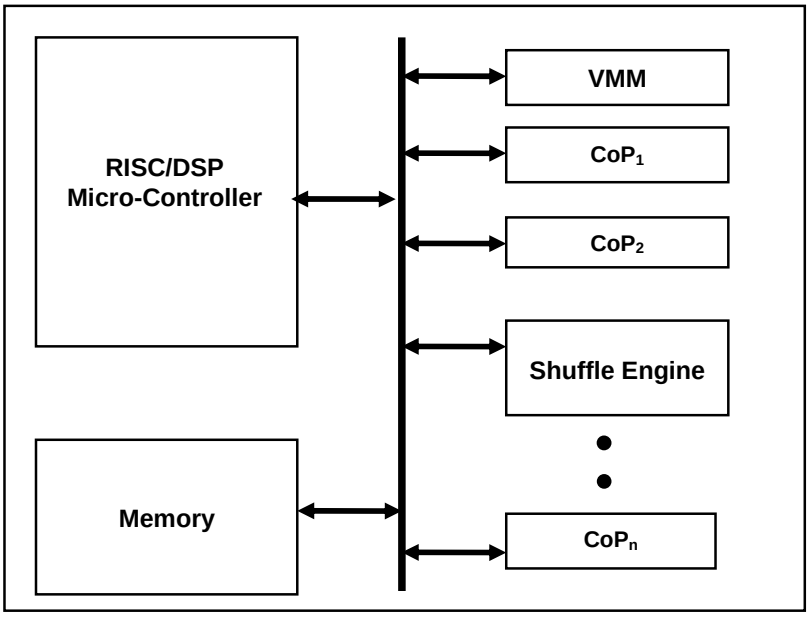

Fig. 4. System architecture.

3.2 The ALU Chip

Several SED units are integrated into one ALU chip (see Fig. 5). In the implementation investigated here, we assume that *16* SED units are integrated into a single ALU chip. This is a reasonable assumption since the estimated number of transistors in each SED unit is less than *100,000* (assuming a *2048×8* bit buffer / shifter implemented as an SRAM organized as a FIFO (first in, first out) queue; with *6* transistors per bit, which is fewer than *100K* transistors). In addition, the number of internal connections within the SED unit is small.

An ALU chip is connected to external memory or I/O devices through an external bus. An internal bus distributes the data to the 16 SED units inside the ALU chip. Each unit gets *256* bit at a rate of *1* GHz, hence a rate of *125* MByte per second. This means that the entire SLM matrix is updated at the same rate as the input row vector (*125* MHz).

Theoretically, a bus of *2048* bit operating at *2* GHZ is needed in order to supply *256* bit of data to each SED unit and sustain a rate of *125* MHz. Practically, we can use an internal bus of *2176* bit, which operates at *2.4* GHz, to drive the individual SED units. The reason for using a bus of *2176* bit is due to the need to synchronize the input data. It is assumed that each set of *16* data lines are synchronized through one synch line. Hence, *2048* data lines require *128* synch lines and the total number of input lines is *2176*. In addition, it is assumed that *20%* redundancy in the number of

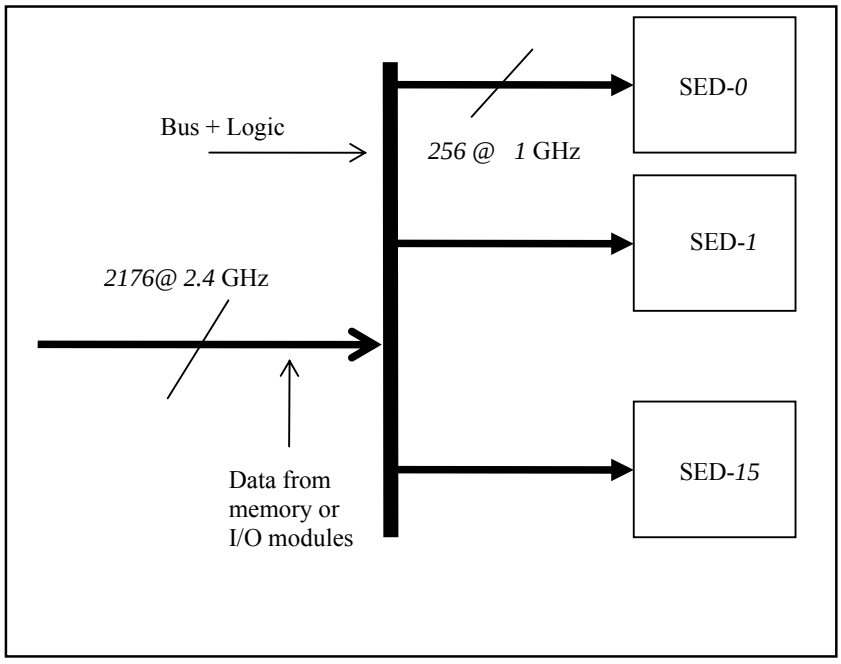

Fig. 5. The ALU chip.

input bits is required in order to supply error correction and handshaking mechanism. This can be achieved by raising the frequency of the bus to *2.4* GHz. While this is a relatively wide bus operating at a relatively high frequency, it is well supported by current *0.65*-nm CMOS technology, where chips such as the IBM Cell contain thousands of pins, thousands of bus lines, and operate at rates that are above *3* GHz [14]. It is also in line with the ITRS 2005 / 2006 reports [15]. Furthermore, this bus is not a true multi-drop bus; rather, it is implemented as a number of narrower point-to-point interconnects (driving a very wide bus quickly and supporting many sinks would be a design challenge).

The proposed device requires a package with several thousand input pins. This is within the state of the art. As predicted in Ref. [16], packages with over *3000* connections are currently available

Within the devices, several sources distribute Gigahertz-rate signals to several destinations. This is also within the capabilities of current technology. As an example, the Tilera TILE 64 implements *64* processors in a *90*-nm CMOS technology [17]. Each processor connects to five point-to-point networks implemented as unidirectional *32*-bit interconnects, which provides an aggregate bandwidth of *1.28* Terabit per second. This bandwidth is provided by *320* bits (lines). Hence, each wire in this older CMOS technology is providing *1280/320=4* Gigabit per second. Furthermore, Intel has described a *5* GHz on-chip network [18].

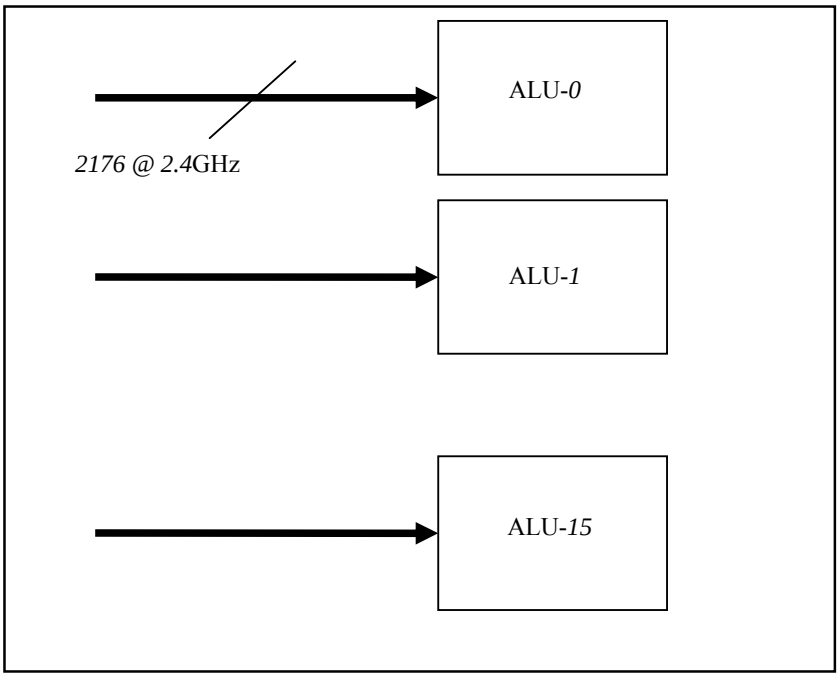

Fig. 6. The interface board.

3.3 The Interface Board

The interface board depicted in Fig. 6 contains *16* ALU chips. In addition, the interface board contains lines that enable transfer of external inputs to the ALU chip. Each ALU chip is fed with data independently of other ALU chips. While there are no internal connections or dependencies between the ALU chips, they work in a source-synchronous mode, where each ALU chip operates at a rate of *2.4* GHz.

The interface board is comparable in complexity to a small-size InfiniBand interconnect board [19]. In fact, in some applications, it is conceivable that the ALU chips occupy more than one interface board. In these cases, the system is equivalent in complexity to several system-on-chip (SOC) units connected to an InfiniBand board. In addition, since there are no dependencies / communication transactions between ALU chips, the interface board does not require a fully-connected switch. The conclusion from the above discussion is that the level of complexity of the interface board is moderate and that this design can be implemented with current of-the-shelve VLSI technology.

3.4 System Synchronization

We consider two synchronization tasks. First, the SED units within an ALU chip have to be synchronized. Second, the ALU chips within an interface board have to be synchronized.

Synchronization of SED units within the ALU chip is relatively straightforward. The SED units operate at *1* GHZ. Each SED contains small amount of logic and the data path within the SED is short (*4* transistors in serial, one of which is an SLM transistor). This is well below the characteristics of current multi-core systems which contain *16* to *32* cores, a data-path with *11* (or more) transistors, and operate at frequencies of *3* GHz and above [17, 18, 20, 21].

Given the characteristics of an ALU chip, synchronizing the ALU chips within the interface board requires a relatively modest level of complexity. The synchronization can be done by distributing a clock signal through the ALU chips. Commercial systems that distribute clock through a clock-tree to hundreds of devices with skew of less than *50* picoseconds (ps) are available. Hence, a 2-branch *8*-level clock distribution unit can be used to synchronize the *16* ALU chips. Using this tree, an uncertainty of the order of less than 50 ps is achievable [17]. Nevertheless, since the source array operates at a rate of *125* MHz, *50* ps is a negligible uncertainty and can be factored into the design. The effect of uncertainty can cause a loss of one least significant bit from an entire set of *256×8* bit. Given the fact that we can use guard bits in the VMM, this uncertainty is tolerable.

4. DSP-Intensive and Computationally-Intensive Applications

The DSP procedures, listed in Section 2 and detailed in a technical report [13], can be used as building blocks for several DSP-intensive applications. As explained in the abovementioned technical report, the proposed architecture is capable of performing *125* million operations - of a (*1*x*256*x*8* bit) vector by a (*256*x*256*x8 bit) matrix multiplication - per second. It can complete *32* billion cross correlations - of a (*1*x*256*x*8* bit) vector by a (*1*x*256*x*8* bit) vector - per second. In addition, the VMM can complete *125* million convolutions - of up to *511* samples with a finite input response filter of up to *256* taps - per second, and *31.25* million discrete Fourier transforms - of *256* complex samples - per second.

Goren *et al* [1] have elaborated on wireless-related processing, such as rake receiver and multi-user detection. Their proposed VMM, however, is at least two orders of magnitude slower than the current proposed VMM. In this section, we analyze the performance of the proposed system as a component in several other DSP-intensive and computationally-extensive applications.

4.1 The VMM as a Motion Estimation Engine

The VMM can be used to implement MPEG/H.264-like motion estimation (ME). Consider a current-frame macro-block of *16×16* pixels stored in the input vector and a search window of *32×48* pixels. The controller loads the SLM matrix with

consecutive macro-blocks from the search window and the VMM implements cross-correlation with *256* blocks in each cycle (32 billion cross correlations per second). The entire *32×48* window contains *1536* instances of overlapping macro-block. Hence, it can be loaded into the SLM in *6* cycles of operation or at a rate of $125/6 = 20.8\overline{3}$ MHz. This is the rate at which the VMM can complete the search. The controller has to find the maximum of the cross-correlation function calculated by the VMM. Some variants of the above analysis can be of interest. For example, if an "informed" search, such as multi-resolution (pyramid) search, is implemented, then each stage can use *256* macro-blocks from the search window. A *3*-stage search in a *32×48* window at *1/4*-pixel resolution can be accomplished at *25* MHz per macro-block. A 1/8-pixel resolution requires one more cycle and can be accomplished in $125/6 = 20.8\overline{3}$ MHz [13].

4.2 String Matching Using the VMM

The motion estimation method described above is a special case of two-dimensional string matching. The VMM has exceptional capability to support string matching. It can be used for exhaustive string matching or to support advanced matching techniques such as the Boyer-Moore, Knuth-Morris-Pratt, or Rabin-Karp algorithms [22]. The string or substring to be matched is stored in the VMM input vector. It can be matched via cross-correlation with *256* other substrings stored in the SLM matrix. Under the current architecture, the VMM can sustain "row" (exhaustive) string matching rate of *256* Gigabit per second. This is at least two orders of magnitude faster than existing hardware architectures of exhaustive search. In the future, we plan to investigate the possibility of loading the SLM matrix at a rate that is higher than *125* MHZ. This can increase the row-string-matching capability of the VMM to the order of Terabit per second. In addition, we plan to investigate the VMM capability to support Basic Local Alignment Search Tools (BLAST) and algorithms for matching nucleotide or protein sequences [23].

4.3 The VMM in a Geometry Engine System

The VMM can be used to support the geometry pipeline of a computer-graphics system, where multitudes of polygons (generally triangles), represented by vertices in a *4*-dimensional homogeneous coordinate space, are subject to affine transformation and perspective/parallel projections. Affine transformations and perspective projections of the polygons require multiplication of *1×4* vectors, representing polygon vertices, by *4×4* matrices. The proposed VMM can complete *2* billion multiplications of a *1×4×8*-bit vector by a *4×4×8*-bit matrix per second. Hence, it can compute the transformation of 666 million triangles per second. The vertices, however, are represented by low resolution *8*-bit elements. To support *32*-bit floating point operations, i.e., operations with *24*-bit mantissa, the VMM has to enable *24*-bit multiplication. This can be achieved by generating *9* partial products. Reuse of data stored in the matrix can enable a rate of *222* million triangles per second. In this case, however, the controller is expected to do extensive pre-processing and post-

processing (e.g., shift-and-add of partial products). This may require another dedicated co-processor. Furthermore, each *8*-bit×*8*-bit multiplication must produce no errors, since these errors can appear in significant bits of the *24*-bit result. Thus, the VMM should be capable of generating an exact *16* bit result. This can be accomplished by adding guard bits to the VMM multiplication unit.

4.4 Bounded NP-Complete Problem Optical Solver

Due to the difficulty of solving high-order instances of bounded NP-complete combinatorial problems, many approximation and heuristic methods have been proposed in the literature. These methods, however, have unpredictable execution time. Therefore, for certain bounded NP-complete applications, where deadlines must be met, such methods are not a good choice and one may prefer to use an exhaustive search. For these cases, a new optical method which can provide a significantly better and guarantied solution-time is proposed in Refs. [2] and [3].

The proposed device is capable of solving bounded NP-complete problems such as the traveling salesman problem (TSP), the Hamiltonian path problem (HPP), etc. by checking all feasible possibilities orders of magnitude faster than a conventional computer. To do this, we use the VMM to perform a fast optical vector-by-matrix multiplication between a weight vector representing the problem weights and a binary matrix representing all feasible solutions. The multiplication product is a vector representing the final solutions of the problem. In the TSP for example, where the required solution is the shortest Hamiltonian tour connecting a certain set of given node coordinates, the multiplication is performed between a grayscale weight vector representing the weights between the TSP nodes and a binary matrix representing all feasible tours among the TSP nodes. The multiplication product is a length vector representing the TSP tour lengths by peaks of light with different intensities. Finding the shortest Hamiltonian tour can be performed by using an optical polynomial-time binary search which utilizes an optical threshold plate. On the other hand in the HPP, a decision whether there is a Hamiltonian path connecting two given nodes on the HPP graph is required. In the HPP, the binary matrix still represents all feasible paths (tours), but the weight vector is also binary. After performing the vector-by-matrix multiplication, any peak of light obtained in the output of the optical system means that a Hamiltonian path exists.

The advantage of the proposed method is that once the binary matrix is synthesized, all TSP and HPP instances of the same order (with the same number of nodes) can be solved optically by only changing the weight vector and performing the vector-by-matrix multiplication in an optical way. In addition, in Refs. [2] and [3] we have presented an efficient method to arrange the tours or paths in the binary matrix so that the binary matrix of N nodes contains the binary matrix of $N-1$ nodes. Therefore, once the binary matrix of N nodes is synthesized, all TSP and HPP instances containing N or fewer nodes can be solved by the VMM. In case the binary matrix contains more than *256*×*256* elements, the binary matrix is stored in the proposed device memory and then uploaded in a high rate to the SLM in parts in order to perform different portions of the vector-by-matrix multiplication each time. Note that in this case the SLM contains binary matrices (rather than *8* bit grayscale

matrices). Hence, a dedicated special-purpose TSP/HPP VMM solver is expected to have performance that is 8 times faster than the performance of the general-purpose VMM presented in former sections.

5. Conclusions

New optical architecture for a super DSP co-processor that is based on of-the-shelf technology is presented. The architecture solves severe bottlenecks of the previous commercial electro-optical designs, gaining orders of magnitude better performance. The architecture supports principal- DSP primitives such as vector-by-matrix and matrix-by-matrix multiplications, convolution and correlation, discrete Fourier transform, L2 norm operations, addition, complex numbers and extended-precision arithmetic. These building blocks enable DSP-intensive applications, as well as an efficient bounded NP-complete problem solution. The device operational rate, in the order of at least 16-Tera integer operations per seconds, is a great opportunity for enhancing existing applications and introducing new computer applications, such as video compression and three-dimensional geometry engine. It is expected that developments in electro-optics technology will enable increasing the performance of this architecture, allowing more SED units and laser elements per device so that higher operational rate can be obtained.

Acknowledgments

This research was supported by Rita Altura Trust Chair in Computer Sciences.

References

1. A. Goren, S. Sariel, Y. Levit, S. Asaf, B. Liberman, T. Sender, Y. Tzelnick, H. Hefetz, E. Moses, and V. Machal, "Vector-matrix multiplication," United States Patent No. 2004/0243657 A1 (2004).
2. N. T. Shaked, S. Messika, S. Dolev, and J. Rosen, "Optical solution for bounded NP-complete problems," *Applied Optics* **46**(5), 711-724 (2007).
3. N. T. Shaked, T. Tabib, G. Simon, S. Messika, S. Dolev, and J. Rosen, "Optical binary-matrix synthesis for solving bounded NP-complete combinatorial problems," *Optical Engineering* **46**(10), 108201: 1-11 (2007).
4. S. Dolev and Y. Nir, "Optical implementation of bounded non-deterministic Turing machines," US Patent No. 7,130,093 B2 (2006).
5. S. Dolev and H. Fitoussi, "The traveling beams optical solutions for bounded NP-complete problems," *Fourth International Conference on Fun with Algorithms (FUN2007)*, *LNCS* **4475**, 120-134 (2007).
6. A. V. Oppenheim and R. W. Schafer, *Discrete Time Signal Processing*, 2nd Ed. (Prentice Hall, 1999).
7. J. Bier, *Byers Guide to DSP Processors*, 2005 Ed. (Berkeley Design Technology Inc, 2006).

8. Anonymous, “Texas Instruments C64 Performance Benchmarks,” http://focus.ti.com/dsp/docs/dspplatformscontentaut.tsp?sectionId=2&familyId=477&tabId=496
9. D. G. Feitelson, *Optical Computing: A Survey for Computer Scientists* (MIT Press, 1988).
10. D. McAulay, *Optical Computer Architectures: The Application of Optical Concepts to Next Generation Computers* (Wiley-Interscience, 1991).
11. M. A. Karim and A. A. S. Awwal, *Optical Computing: An Introduction* (John Wiley & Sons, 1992).
12. J. W. Goodman, *Introduction to Fourier Optics* (McGraw-Hill, 1996).
13. D. E. Tamir, N. T. Shaked, P. J. Wilson, and S. Dolev, “Electro-optical DSP of Tera operations per second and beyond,” Technical report #09-08, Department of Computer Science, Ben-Gurion University of the Negev, Beer Sheva, Israel (2008).
14. M. Gschwind, “Chip multiprocessing and the Cell broadband engine,” *IBM Research Report* **RC2931** (2006).
15. Anonymous, *International Technology Roadmap for Semiconductors; Assembly and Packaging.* (ITRS, 2005).
16. J. Wolf and J. Adams, *2005 Packaging Roadmap Overview* (The International Electronics Manufacturing Initiative (INEMI), 2005).
17. J. Kim, I. Verbauwhede, and M. F. Chang, “Design of an interconnect architecture and signaling technology for parallelism in communication,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* **15**(8), 881-894 (2007).
18. D. Wentzlaff, P. Griffin, H. Hoffmann, L. Bao, B. Edwards, C. Ramey, M. Mattina, C. Miao, J. F. Brown, and A. Agarwal, “On-chip interconnection architecture of the Tilera processor,” *IEEE Micro* **27**(5), 15-31 (2007).
19. Anonymous, “Infiniband technology overview,” http://www.infinibandta.org/about/
20. Y. Hoskote, S. Vangal, A. Singh, N. Borkar, and S. Borkar, “A 5-GHz mesh interconnect for a Teraflops processor,” *IEEE Micro* **27**(5), 51-61 (2007*).*
21. T. H. Wood, E. C. Carr, C. A. Burrus, J. E. Henry, A. C. Gossard, and J. H. English, “High-speed 2×2 electrically driven spatial light modulator made with GaAs/AlGaAs multiple quantum wells (MQWs),” Electronics Letters **23**(17), 916-917 (1987).
22. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and R. C. Stein, Introduction to Algorithms, 2nd
Ed. (MIT Press, 2001).
23. Anonymous, “Basic local alignment search tool (BLAST),” http://www.ncbi.nlm.nih.gov/blast/Blast.cgi