High-speed and low-power electro-optical DSP coprocessor

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*
[2]*Department of Electrical and Computer Engineering, Ben-Gurion University of the Negev, P.O. Box 653, Beer-Sheva 84105, Israel*
[3]*Kiva Design, 104 Forest Trail, Leander, Texas 78641, USA*
[4]*Department of Computer Science, Ben-Gurion University of the Negev, P.O. Box 653, Beer-Sheva 84105, Israel*
*Corresponding author: dt19@txstate.edu

A fast, power-efficient electro-optical vector-by-matrix multiplier (VMM) architecture is presented. Careful design of an electrical unit supporting high-speed data transfer enables this architecture to overcome bottlenecks encountered by previous VMM architectures. Based on the proposed architecture, we present an electro-optical digital signal processing (DSP) coprocessor that can achieve a significant speedup of 2–3 orders of magnitude over existing DSP technologies and execute more than 16 teraflops. We show that it is feasible to implement the system using off-the-shelf components, analyze the performance of the architecture with respect to primitive DSP operations, and detail the use of the new architecture for several DSP applications. © 2009 Optical Society of America

OCIS codes: 200.0200, 200.4560, 200.2610, 200.4960, 200.3050.

1. INTRODUCTION

More than three decades of investment in complementary metal-oxide semiconductor (CMOS) technology have generated cost-effective, high-performance processing units. Nevertheless, CMOS technology is reaching physical and operational limits that necessitate a change of paradigm [1]. One limitation is due to the physical dimensions of the transistors constituting the building blocks of VLSI circuits [2]. With technologies of 45 nm and below, several side effects in transistor characteristics are amplified and prevent efficient utilization of technology. Another related problem is power consumption [3,4]. Current processing units (e.g., the high performance PowerPC and Pentium processors) are operating at power rates of hundreds of watts [5].

To address these problems, major VLSI design companies such as Texas Instruments (TI), IBM, and Intel concentrate their efforts in two directions. The first route is "evolutionary," exploring new architectures such as multicore processors and ways to address the limitations of current CMOS technology using new materials while essentially keeping the same basic process. For example, in coping with the "power crisis," Intel has announced an experimental high performance multicore system, referred to as the Intel Teraflop, which is expected to achieve 2 teraflops at a power consumption rate of 250 watts [5]. The second approach, the "revolutionary" approach, is seeking new technologies such as optical-based computing. Several new optical-over-silicon architectures were suggested as potential alternatives to CMOS technology [6–10].

Direct comparison between mature, CMOS-based electrical computation devices and emerging optical-based computation devices may be misleading. One should take into account the high potential of optical computing and not just the current state of the art. Otherwise, there will not be sufficient justification to invest in research on most new technologies. Indeed, TI, IBM, and Intel are aware of the potential of optical computing and are investing a significant amount of resources on optical-based computing devices.

Vector-by-matrix multiplication is a fundamental operation in many computer applications. Computational tasks involving vector-by-matrix multiplication include rendering computer-generated images, beam forming, radar detection, and wireless communication systems [6,11–16]. 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. For example, a discrete Fourier transform (DFT) can be implemented as a special case of vector-by-matrix multiplication [11].

Optical implementation of vector-by-matrix multiplication can be carried out in parallel at a very fast rate. This rate, however, must be supported by electronic interface for transferring data to the multiplier. Otherwise, the speed of the computation is determined by the data transfer bottleneck. In 2003 the Lenslet Company presented an advanced optical VMM [6]. The company was able to solve many of the relevant challenges and developed a prototype system referred to as Enlight-256 [14]. One of the prototypes was tested by the Oak Ridge National Labs (ORNL) [15,16]. The ORNL account validates the perfor-

mance breakthrough and operational metrics reported in [14]. Nevertheless, in many applications the performance of the Enlight-256 is severely limited by the rate of the electrical driver for updating the VMM matrix (40 KHz).

In this paper, we propose a system that overcomes these limitations. We present a new electro-optical architecture that enables exploiting the "full speed" of optical vector-by-matrix multiplication. The proposed architecture is based on the Lenslet design, and the data transfer bottleneck is solved by splitting the 256×256-element spatial light modulator (SLM), used to represent the matrix as 256 SLM lines of 1×256 elements each. Splitting the SLM element enables refreshing the matrix at a rate of 125 MHz. As a result, the realistic execution rate of our proposal is at least 2 orders of magnitude higher than the execution rate of existing systems.

The electro-optical vector-by-matrix multiplier (VMM) presented in this paper can perform a general vector ($1 \times 256 \times 8$ bit) by matrix ($256 \times 256 \times 8$ bit) multiplication in one cycle of 8 ns (that is, at a rate of 125 MHz). This means 2 to 3 orders of magnitude execution speedup over current DSP architectures and 1 order of magnitude execution speedup over high-performance multicore systems. This rate is expected to improve with the introduction of new electro-optical technologies. Moreover, this speedup is achieved with lower power dissipation. It is expected that the architecture will consume slightly less power than contemporary DSP architectures and a fraction ($\sim 5\%$) of the power dissipated by multicore systems such as the Intel Teraflop.

It should be noted that all the speedup and instructions-per-watt performance figures given in this paper take into account the word length of the respective architectures. For example, when the execution speed of the VMM is compared with the execution speed of a 16-bit DSP, a factor of 4 is used to represent the execution speed of the DSP. Furthermore, when compared with a floating point device, the floating point device execution rate is multiplied by 9 to account for the 24-bit integer part of a floating point number. This conservative assumption that the DSP can perform two 8-bit operations per operand, per cycle and that a floating point architecture can perform three 8-bit operations, per operand, per cycle somewhat penalizes the actual speedup figures gained by the VMM and reported in this paper.

In addition, the paper presents a feasible scheme that exploits the capabilities of this architecture as a coproces-

Fig. 1.   Electro-optical computing system.

sor dedicated for DSP-intensive and computationally intensive applications such as rendering computer-generated images, beam forming, radar detection, and wireless communication systems.

The rest of the paper is organized in the following way: Section 2 introduces the architecture of the proposed electro-optical design. Section 3 presents an implementation of the system, provides data that shows the design feasibility and explains how this design can be improved along with technology advances. Section 4 details the implementation of several DSP primitive operations that are commonly used in numerous DSP-intensive applications and analyzes the power consumption of the proposed device. Several DSP applications, utilizing the operations described in Section 4, are presented and analyzed in Section 5. Section 6 concludes the paper.

2. ELECTRO-OPTICAL COMPUTING SYSTEM

The VMM can serve as a coprocessor attached to a DSP or a dedicated CPU, referred to as the controller. In addition, the controller can utilize other coprocessors. The entire system (controller, coprocessors, and VMM) is depicted in Fig. 1. In this figure, one of the coprocessors 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 coprocessors, prepares an input vector and an input matrix for VMM processing. We refer to this as preprocessing. The VMM output is sent back to the controller, where it may go through additional processing (referred to as postprocessing) before being sent to other devices or back to the VMM. Sound preprocessing and postprocessing 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. As a result, it can reduce power consumption and enable the VMM to operate at a very high rate.

A. VMM Architecture
A high-level schema of the VMM is presented in Fig. 2. As the figure shows, the VMM is composed of two main components: the optical unit and the electrical driver. The optical unit accepts a series of analog electrical signals, converts the signals into light, and generates a set of light sources representing the input. The matrix is represented by SLM transistors. The output stage of the optical unit contains a set of light-sensitive detectors that convert the light that passes through the SLM into a set of analog electrical signals.

The electrical unit contains an array of 256 analog to digital converters (ADC) and an array of 256 digital to analog converters (DACs). The design presented in this paper assumes that the light detector and the ADCs are independent units allowing for efficient implementations. Nevertheless, off-the-shelf detectors that include the ADC can replace the detector/ADC combination presented here.

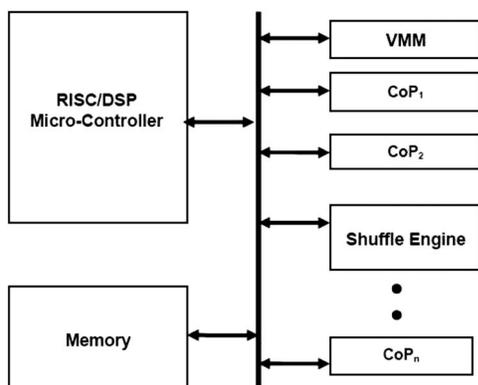In a typical mode of operation, the electrical unit receives digital inputs representing vectors and matrices.
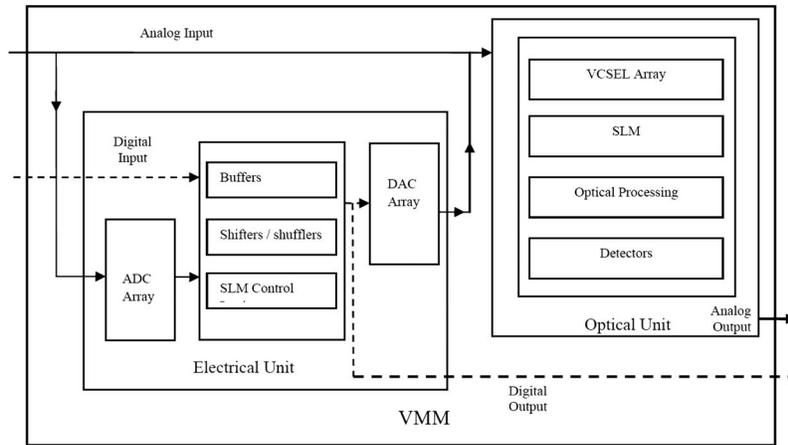
Fig. 2.   VMM architecture.

Data representing vectors feed DACs, thereby producing input vectors for the optical unit. Data representing the matrix are used to set the state of the SLM transistors. In addition, the electrical unit can buffer data and perform limited processing, such as shifting and shuffling. Alternatively, the electrical unit can send the data to the system controller and/or other coprocessors (see Fig. 1).

Due to the availability of the DAC and ADC arrays, the system has additional capabilities. For example, analog signals originated by external devices can be used as direct input for the optical unit or for the electrical unit. Similarly, analog signals generated by the optical unit or by the DAC array of the electrical unit can serve as output to external analog devices. Elaboration on these options is beyond the scope of this paper. The next subsections detail the components of the VMM architecture.

1. VMM Optical Unit

One of the configurations that can be used to implement the optical component of the VMM is based on the Stanford multiplier principle and is illustrated in Fig. 3 [17]. 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 of the matrix. Next, each row in the matrix is summed onto a single detector

Fig. 3.   (Color online) VMM optical unit.

in the detector array. This VMM configuration can be implemented through several optical techniques. One of these techniques uses two sets of lenses each of which contains a cylindrical lens and a spherical lens with focal lengths of *f*. A single set of lenses has an equivalent focal length of $f/2$ in the vertical/horizontal direction and *f* in the other direction. As shown in Fig. 3, 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. The other set of lenses is positioned between the matrix and the output vector (represented by the sensitive detectors) and is rotated 90 deg with respect to the first set of lenses (see Fig. 3). Therefore, these lenses image a row in the matrix onto a vertical position of a single detector in the detector array and spread the light from a single column of the matrix.

Psaltis *et al.* [18] proposed an extended version of the optical VMM by integrating multicolor LED arrays and a photodiode detector array with 1-D adder feedback. Casasent [19] proposed the use of acousto-optic transducers and obtained iterative VMM. Athale *et al.*, [20] proposed to use a binary multiplication via outer products to obtain higher accuracy. Other papers proposed additional versions and applications of the optical VMM. Comprehensive literature reviews in this subject can be found in [21–23].
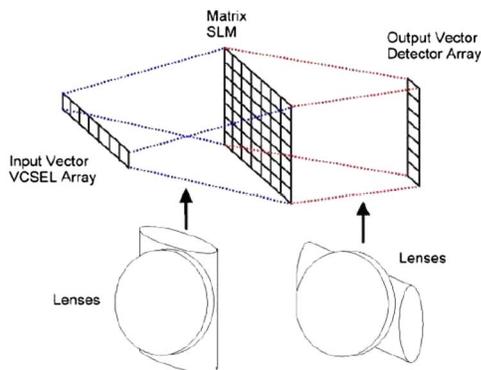
2. VMM Electrical Driver

A high-level description of the electrical components of the system is given in Figs. 4 and 5. Figure 4 shows the VMM electrical driver. The driver is comprised of 256 single electrical driver (SED) units and has two types of inputs: a $1 \times 256 \times 8$ bit input vector *A*, which is the VMM input vector (that is, the VCSEL source array driving signal), and a set of 256 vector inputs B_0 to B_{255} (total of $256 \times 256 \times 8$ bits). The output of the VMM is a $1 \times 256 \times 20$ 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 the *j*th SED unit is a single 20-bit bus that is generated by the output detector array.
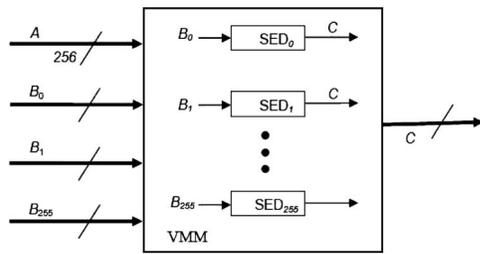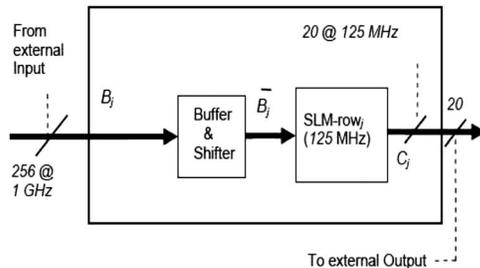
Fig. 4.   VMM electrical driver.

Fig. 5.   A 1×256 SLM electrical driver.

Figure 5 shows the design of a single $1 \times 256 \times 8$ bit SED unit. The input vector B_j can be stored in an internal dual-port modular memory buffer/shifter before it is directed to the SLM. The output C_j is directed to the external output. The configuration depicted in Fig. 5 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 $(1 \times 256 \times 8$ bit$)$ by matrix $(256 \times 256 \times 8$ bit$)$ multiplication operation at a rate of 125 MHz.

Each of the units depicted in Fig. 5 is capable of executing 256×125 million multiply-accumulate instructions per second. This is equivalent to 64 Giga 8-bit integer operations per second (GInt8S). Since the entire VMM system consists of 256 units identical to the unit depicted in Fig. 5, it can exhibit a peak performance of 16,384 GInt8S.

The VMM is capable of performing one generic operation: a vector $(1 \times 256 \times 8$ bit$)$ by matrix $(256 \times 256 \times 8)$ multiplication. Nevertheless, special cases of vector-by-matrix multiplication can be used as the building blocks for several DSP procedures that include 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, operations on complex numbers, and extended-precision operations. These operations, which are the building blocks of numerous DSP-intensive applications, are further analyzed in Section 4.

B. VMM Architecture Evaluation
A prototype of the VMM reported in [6,14] was implemented using an 8-bit DAC and a 10-bit light detector/

ADC combination. It is estimated that the detector/ADC reported in [6] has a bit error rate (BER) of the order of 10^{-4}. Goren asserts that this is a tolerable BER which could be mitigated through operations-redundancy and error correction bits [6].

Generally speaking, the VMM is designed for computations involving 8-bit vectors. Hence, it is expected that the result of VMM operations eventually will be cast into 8-bit integers. Under this assumption, a 10-bit detector/ADC provides acceptable precision. A resolution of 16 bits is adequate for all the applications listed in this paper. Nevertheless, in order to accommodate for the need for higher precision that can result from carry bits as well as multiplying and accumulating vectors of 8-bit integers (e.g., in a dot-product operation), we assume a detection/ADC array that is able to detect at 20-bit precision. Currently, this precision is not commercially available. It is, however, in line with near-future advances in technology as predicted in [24,25] or it can be custom designed [24–29]. Furthermore, high-speed 16-bit detectors are currently available (some of the papers referenced herein report on detectors with up to 20-bit resolution) [27–34]. These detectors introduce shot noise that is at a level of less than −48 dB and a BER of less than 10^{-10}.

The use of an 8-bit DAC and a 20-bit detector/ADC along with guard bits used in the multiplication can provide for an accurate 8-bit design over a dynamic range of more than 48 dB. In fact, as [6] proposes, a 9-bit DAC can be used for conversion of intermediate results (before storing them as 8-bit entities). This could increase the dynamic range to about 56 dB. In this case, however, the shot noise might be the limiting factor. In current technology it might dictate a 48 dB dynamic range. It is expected that future technology will suppress the entire noise level to 56 dB or below, thereby justifying a 9-bit DAC. The next section provides a detailed review of a feasible implementation of the VMM.

3. IMPLEMENTATION OF THE VMM ELECTRICAL DRIVER
This section includes implementation details and demonstrates that it is feasible to implement the electrical portion of the system using existing mass-production VLSI technology. The main components of the VMM electrical driver are illustrated in Figs. 5–7. The driver consists of 256 SED units of $1 \times 256 \times 8$ bit each (see Fig. 5). Several
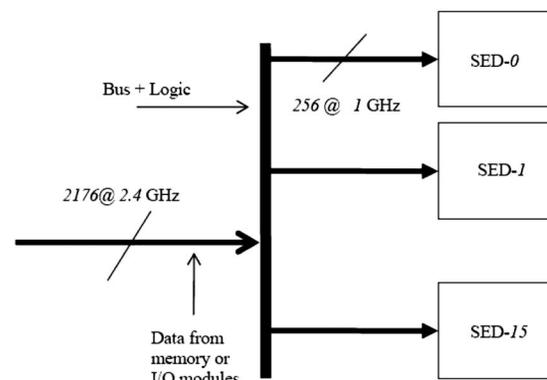

Fig. 6.   ALU chip.

Fig. 7.   Interface board.

SED units are integrated into ALU chips (depicted in Fig. 6) and placed on an interface board (illustrated in Fig. 7). Overall, the interface board contains 256 SED units. There is a design flexibility with respect to the number of SED units per ALU chip. This figure dictates the required number of ALU chips in the interface board. After analyzing the currently available technologies, 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.

A. SED Unit
Each SED unit contains a 256-element SLM (i.e., 256 SLM transistors) and a relatively small dual-port modular memory of several thousand bytes (say 2048 bytes). The SLM element SED_j (depicted in Fig. 5) 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 B_j. In this case, the shifter can implement a shift of 1, 2, 4, or 8 bits on one stored row of 256×8 bits enabling convolution and correlation with bytes and sub-byte units. The output C_j, is a single 20-bit bus that 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 operating at 1 GHz. Hence, it loads the 256 SLM elements of the SED and/or the SED buffer at a rate of 125 MBytes/s.

B. ALU Chip
An ALU chip consists of several SED units (see Fig. 6). 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. In addition, the number of internal connections within the SED unit is small.

An ALU chip is connected to external memory or input/output (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, a rate of 125 MBytes/s. 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 bits of data to each SED unit and sustain a rate of 125 MHz.

Practically, we can use an internal bus of 2176 bits operating at 2.4 GHz to drive the individual SED units. The reason for using a bus of 2176 bits is the need to synchronize the input data. It is assumed that each set of 16 data lines is synchronized through one synch line. Hence, 2048 data lines require 128 synch lines, making the total number of input lines 2176.

In addition, it is assumed that 20% redundancy in the number of input bits is required in order to supply error correction and handshaking mechanisms. 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 and thousands of bus lines and operate at rates above 3 GHz [35]. It is also in line with the ITRS 2005/2006 reports [36]. 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 indicated by International Electronics Manufacturing Initiative (INEMI) 2005 Packaging Roadmap Overview, suggesting that packages with over 4500 connections will be available by 2009 [37].

The proposed device requires fairly high input signaling rates. High-performance serial interconnects such as PCI Express and Rapid IO are driving very-high-frequency source-synchronous signaling schemes with rates in the 4–6 GHz region and might dissipate a noticeable amount of power. These devices, however, are intended to be deployed with relatively few connections. Hence, their power dissipation may be a concern. The proposed VMM electrical interface chip, however, is input-only; thus, the issues associated with driving a large number of high-frequency signals can be relegated to the system to which the proposed chip is to be connected.

Several sources within the devices 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 [38]. Each processor connects to five point-to-point networks implemented as unidirectional 32-bit interconnects, which provides an aggregate bandwidth of 1.28 Terabits/s. This bandwidth is provided by 320 bits (lines). Hence, each wire in this older CMOS technology is providing 1280/320=4 gigabits/s. Furthermore, Intel has described a 5 GHz on-chip network [5,39].

C. Interface Board
The interface board depicted in Fig. 7 contains 16 ALU chips. In addition, the interface board contains lines that enable transfer of external inputs to the ALU chips. 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 interconnect device such as the Infiniband

board [39–42]. 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 or communication transactions between ALU chips the interface board does not require a fully connected switch. We conclude that the level of complexity of the interface board is moderate and that this design can be implemented with current off-the-shelf VLSI technology.

D. System Synchronization
We consider two synchronization tasks: the SED units within an ALU chip have to be synchronized, and the ALU chips within an interface board must 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 (four transistors in serial, one of which is an SLM transistor). This is well below the characteristics of current multicore systems, which occupy 16 to 32 cores, contain a data path with 11 (or more) transistors, and operate at frequencies of 3 GHz and above [5,35,38,39]. 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 signals through a clock-tree to hundreds of devices with skew of less than 50 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 [39]. 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 bits. Given the fact that we can use guard bits in the VMM, this uncertainty is tolerable.

4. PRIMITIVE DSP OPERATIONS
The VMM can perform at a peak rate of 16,384 GInt8S. The following sections analyze the performance of the unit with respect to primitive DSP operations. Section 5 shows how these operations can be enhanced to DSP-intensive applications.

A. Vector-by-Matrix Multiplication
Vector-by-matrix multiplication is an essential operation in several DSP applications such as beam forming, radar signal processing, and multiuser detection [6,11–13]. The proposed system can perform multiplication of a $1 \times 256 \times 8$ bit vector by a $256 \times 256 \times 8$ bit matrix at a rate of 125 million vector-by-matrix multiplications per second. If the matrix and the vector are smaller than 256×256 and 1×256, respectively, then it may be possible to achieve the same rate with multiple small matrices/vectors. If the matrix and/or the vector have more than 256×256 (256×1) elements, then there is an overhead related to de-

composing the matrix into sub matrices. In most of the cases, this overhead is negligible. The next subsection demonstrates this assertion with respect to matrix-by-matrix multiplication.

B. Matrix-by-Matrix Multiplication
The proposed system can complete the multiplication of a $256 \times 256 \times 8$ bit matrix by a $256 \times 256 \times 8$ bit matrix at a rate of 0.48 MHz (125/256 MHz). If the matrices have more than 256×256 elements, then there is an overhead related to decomposing the matrix into submatrices. In most of the cases, this overhead is negligible. If the matrices have fewer than 256×256 elements, then it may be possible to perform preprocessing operations and achieve better performance per vector/matrix. To elaborate, consider implementing two 4×4 matrix multiplications using a single 8×8 matrix: Let A, B, C and D be four 4×4 matrices. Then the following constellation can be used to implement the operations $A \times B$ and $C \times D$:

$$\begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix} \times \begin{bmatrix} C \\ D \end{bmatrix} = \begin{bmatrix} A \times C \\ B \times D \end{bmatrix}. \tag{1}$$

Given the above constellation, the multiplication of 64 4×4 pairs of matrices can be implemented by multiplying a matrix of 4×256 by a matrix of 256×256. Thus, the peak rate for this essential computer graphics operation is $64 \times 125/4$ million matrix multiplications per second (a rate of 2 billion matrix multiplications per second). This is an extremely high rate. Nevertheless, it involves data elements of low resolution (8 bits). We discuss higher precision operations in Subsection 4.D.

C. Dot-Product Operation
The dot product is the basic operation of a single SLM-SED unit. It is an essential part of convolution, correlation, transforms, and implementation of $L2$ norms. Consider the jth SLM unit, SLM_j. Let $A = \{a_i\}_{i=1}^{256}$ be the input row vector and let $B = \{b_i\}_{i=1}^{256}$ be row j of the SLM matrix, then the output of SLM_j is $c_j = \Sigma_{i=1}^{256} a_i b_i$. That is, each of the SLM units performs a vector dot product on two 256-element vectors. The peak rate for the dot-product operation is 125 MHz per SLM. As in the instance of matrix-by-vector and matrix-by-matrix multiplication, we can consider three dot-product cases involving (1) vectors with 256 elements, (2) vectors with more than 256 elements, and (3) vectors with fewer than 256 elements. The dot product of vectors with 256 elements is performed at a peak rate of 125 MHz per SLM. With 256 SLM units, the system can perform 256 dot-product operations (on vectors of 256 elements) in parallel. This is equivalent to a rate of 256×125 million or 32 billion dot products per second. The dot product of vectors with more than 256 elements requires decomposition of the vectors. In most of the cases, the decomposition overhead is negligible (less than 1%). The dot product of vectors with fewer than 256 elements can be done effectively in a way that is similar to that presented in Eq. (1). To put the above discussion in context, consider the dot product of two signals each of which contains 40 samples. This is one of the Berkeley Design Technology Inc. (BDTI) DSP benchmarks [12]. In this case, we can populate the input row vector with 6

subvectors of 40 elements each, and store one subvector of 40 elements per SLM unit. The end result is a peak rate of 6×125 MHz. In other words, the VMM can complete 750 million dot products of 40 samples per second. This is a $25\times$ speedup compared with the best published results of the 40-element dot product achieved on the TI-64XX DSP [13].

1. Convolution and Correlation

Convolution (and correlation) between a signal X and a signal S can be implemented through dot product of X with shifted versions of S. It is assumed that the controller (or a coprocessor) generates the shifted values of the signal S. The VMM can perform a convolution between a signal (referred to as mask) of 256 samples and another signal of arbitrary length at a peak rate of 256×125 MHz (32 GHz). A mask with more than 256 elements requires a negligible decomposition overhead. A mask with less than 256 elements can be handled at the same rate. Additionally, several different filters can be implemented in parallel using the dot product with shifted versions of the input utilizing the principle presented in Eq. (1). We consider another BDTI DSP benchmark, which calls for implementing a 16-tap finite impulse response (FIR) filter on 40 samples of a given signal [12,13]. The VMM can sustain a rate of 125 MHz per SLM in this benchmark. Moreover, The VMM can implement 6 different 16-tap/40 sample FIR filters simultaneously, where each of these filters completes 125 million filters per second, a speedup of 100 over the TI C64XX DSP [13]. We discuss the use of cross correlation for motion estimation and for string alignment in Subsection 5.A and 5.B, respectively.

2. DFT/DCT

The direct DFT can be implemented through vector-by-matrix multiplication, i.e., via a set of complex dot products of a complex signal with complex Fourier transform coefficients. Each dot product represents the DFT at a given frequency. We discuss potential representations of complex data and revisit the DFT in Subsection 4.D.2.

On the other hand, a 1-D DCT can be implemented as a real vector by a real matrix multiplication. The VMM can complete a 256-point DCT at a rate of 125 MHz (125 million transforms per second). In addition, the VMM can perform 16 times 2-D DCT of a 16×16 signal at a rate of 62.5 million transforms per second, an equivalent rate of 1 billion 16×16 DCT per second. This is implemented by running DCT on the rows (16 rows of 16 elements) through a dot product of each row with a 16×16 coefficients matrix and then running the resultant 16 vectors through a column-by-column DCT. In this case, copies of the coefficients matrix of 16×16 elements are stored in the SLM, and the input data is stored in the row input vector. An 8×8 DCT can be implemented at a peak rate of 4 billion transforms per second.

D. Addition, Subtraction, Complex Numbers, and Extended-Precision Arithmetic

Along with pre/postprocessing by the controller, the VMM can be used for addition and subtraction. Furthermore, the VMM can operate on complex vectors. But, the controller has to perform adjustments. Finally, the VMM can generate partial products that the controller can utilize as a part of higher-precision operations.

Complex numbers can be represented as a vector with alternating real and imaginary values. Alternatively, complex numbers can be stored as two vectors; the first includes real values and the second includes the imaginary values. Both may require shuffling and addition or subtraction as a part of the pre, or the post, processing. The second approach requires fewer manipulations by the controller and is further elaborated here. Consider the complex vectors A and B with elements $a_i = u_i + jv_i$, and $b_i = x_i + jy_i$, respectively. Let $U = \{u_i\}_{i=1}^{256}$, $V = \{v_i\}_{i=1}^{256}$, $X = \{x_i\}_{i=1}^{256}$, and $Y = \{y_i\}_{i=1}^{256}$. Then $A \cdot B$ can be obtained from the dot products $[U \cdot X - V \cdot Y]$ and $[U \cdot Y + V \cdot X]$. Therefore, four dot products per element along with addition/subtraction by the controller are required. As a consequence, the VMM can perform one complex dot product of 2 vectors of 256-complex elements at a rate of 31.25 million complex dot products per second (two cycles for the real part and two for the imaginary). Hence, the VMM can complete a complex vector-by-matrix multiplication (1×256 by 256×256) at peak rate of 31.25 MHz. Furthermore, it can compute complex matrix-by-matrix multiplication (256×256 by 256×256) at a rate of 0.12 MHz. Finally, the VMM can complete a complex 256-point DFT at a rate of 31.25 MHz. That is, the VMM can perform 31.25 million complex 256-point DFTs per second.

E. Power Consumption

The key advantage of the proposed VMM is high execution rate at low power consumption. We consider a relative measure of power, based on the number of instructions performed per unit of power consumption. To evaluate the power consumption of the VMM unit we analyze the contribution of the optical unit as well as the contribution of the electrical unit to power consumption.

The optical unit includes 65,536 SLM transistors operating at 125 MHz. The power consumption of these transistors is just a fraction of a watt [43]. In addition, the optical unit includes 256 VSCEL transistors. The power consumption of a single VSCEL transistor is of the order of 2 miliwatts [44,45]. Thus, the power consumption of the VCSEL unit is also a fraction of a watt. Finally, according to [24,25,32–34] the power consumption of the detectors can be extremely low—just a fraction of a watt. Hence, the power consumption of the VMM optical subsystems is very low and is less than one watt. It is estimated that the number of transistors required to implement the electrical part of the VMM is less than 100,000. In current technology, the power consumption of 100,000 transistors operating at 125 MHz is a fraction of a watt. The ADC and DAC, however, are the dominating part of power consumption.

Using off-the-shelf converters can result in a total power consumption of 10 watts. Custom-designed converters would require about 4 watts [46–48]. It should be noted that many potential VMM applications such as digital communication are concerned with underlying analog signals. In this case, the VMM, ADC, and DAC might be eliminated, resulting in additional power saving.

In order for the VMM to function as a DSP coprocessor, other coprocessors should be added to the system. A reasonable assumption for the power consumption of the combination of the VMM (including the electrical and optical components) along with a coprocessor would be of the order of 10 watts. This is in line with [14], where a total power consumption of 40 watts is reported; yet, that figure includes the vector processing unit, which is assumed to consume about 30 watts.

Based on these figures and the fact that the VMM can perform 16,384 Giga 8-bit integer instructions per second, the number of instructions per power unit, measured in Giga Int8 (Gint8) operations per watt is 16,384/10 = 1638.4/watt. In comparison, the high-performance members of the TI C64XX DSP family perform at a peak rate of 48 Gint8 instructions per second at a power consumption rate of about 15 watts (including the power consumed by the ADC and DAC units) [13]. Hence, these devices are rated at about 3.2 GInt8/watt. In terms of number of instructions per second, the Intel Teraflop processor outperforms the TI C64XX DSP by 2 orders of magnitude. On the other hand, its power consumption is higher by an order of magnitude. The Teraflop processor achieves a power rating of 16 GFLOP/watt or the equivalent of 50 Gint8/watt.

To summarize, the VMM is expected to be 2–3 orders of magnitude faster than current DSP technology while consuming about the same power. It is expected to be about 1 order of magnitude faster than the Intel Teraflop, yet its power rating of 1638 GInt8/watt is about 25 times better than the Intel Teraflop processor.

5. DSP-INTENSIVE AND COMPUTATIONALLY INTENSIVE APPLICATIONS

The DSP operations, listed in Section 4, can be used as building blocks for several DSP-intensive applications. As demonstrated in Section 4, the proposed architecture is capable of performing 125 million multiplications, of a $(1 \times 256 \times 8$ bit) vector by a $(256 \times 256 \times 8$ bit) matrix, per second. It can complete 32 billion cross correlations, of a $(1 \times 256 \times 8$ bit) vector by a $(1 \times 256 \times 8$ 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 DFTs, of 256 complex samples, per second.

Goren *et al.* [6] have elaborated on wireless-related processing, such as rake receiver and multiuser detection. Their proposed VMM, however, is at least 2 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.

A. 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-blocks. Hence, it can be loaded into the SLM in 6 cycles of operation or at a rate of $125/6 = 20.8\bar{3}$ 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 multiresolution (pyramid) search is implemented, then each stage can use 256 macro-blocks from the search window. A three-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 $125/6 = 20.8\bar{3}$ MHz.

B. String Matching Using the VMM
The motion estimation method described above is a special case of 2-D string matching. The VMM has exceptional capability to support string matching through cross correlation. 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 [49]. 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 gigabits/s. This is at least 2 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 a 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 [50].

C. 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-D 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 $1 \times 4 \times 8$-bit vector by a $4 \times 4 \times 8$-bit matrix per second equivalent to transforming 666 million triangles per second. The vertices, however, are represented by low-resolution 8-bit elements while current graphical processing units (GPUs) use 32-bit floating point operands. 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 and "shift-adding" nine partial products. On the other hand, it could result in a reduction of the effective execution rate of the VMM by a factor of 9. Nevertheless, efficient reuse of data stored in the matrix enables getting only $3\times$ reduction, resulting in a rate of

222 million triangles per second. In this case, however, the controller is expected to do extensive preprocessing and postprocessing (e.g., shift-and-add of partial products). This may require another dedicated coprocessor. Furthermore, each 8×8-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.

D. 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 often have unpredictable execution times. Therefore, such methods are not a good choice for certain bounded NP-complete applications where deadlines must be met and one may prefer to use an exhaustive search instead. For these cases, an optical method that can provide a significantly better and guaranteed solution time is proposed in [7,8].

The proposed device is capable of solving bounded NP-complete problems such as the traveling salesman problem (TSP) or the Hamiltonian path problem (HPP) by checking all feasible possibilities orders of magnitude faster than can 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. For example, in the TSP, where the required solution is the shortest Hamiltonian path connecting a certain set of vertices, the multiplication is performed between a grayscale weight vector representing the weights of TSP graph edges and a binary matrix representing all feasible paths among the TSP vertices. The multiplication product is a length vector representing the TSP path lengths by peaks of light with different intensities. Finding the shortest Hamiltonian path can be performed by using an optical polynomial-time binary search that utilizes an optical threshold plate. On the other hand, in the HPP a decision regarding whether there is a Hamiltonian path connecting two given vertices on the HPP graph is required. In the HPP, the binary matrix still represents all feasible paths, 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 vertices) can be solved optically by only changing the weight vector and performing the vector-by-matrix multiplication in an optical way. Furthermore, references [7,8] present an efficient method to arrange the paths in the binary matrix so that the binary matrix of N vertices contains the binary matrix of $N-1$ vertices. Therefore, once the binary matrix of N vertices is synthesized, all TSP and HPP instances containing N or fewer vertices can be solved by the VMM. In the case where the binary matrix contains more than

256×256 elements, the binary matrix is stored in the proposed device memory and then uploaded at a high rate to the SLM in parts in order to perform different portions of the vector-by-matrix multiplication. 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 perform $8 \times$ faster than the general-purpose VMM presented in earlier sections.

6. CONCLUSIONS
A high-performance, power efficient optical architecture for a DSP coprocessor is presented. The architecture exploits off-the-shelf technology and can be utilized to solve bottlenecks of previous commercial electro-optical designs. Compared with current technology, the proposed architecture can gain execution speedups of 2–3 orders of magnitude while consuming significantly less power. The architecture supports principal DSP primitives such as vector-by-matrix and matrix-by-matrix multiplications, convolution, correlation, and discrete Fourier transforms. Along with a coprocessor, the architecture can efficiently process integers as well as complex numbers and perform extended-precision arithmetic. These building blocks enable DSP-intensive applications as well as finding optimal solutions to several bounded NP-complete problems. The device operational rate, of the order of at least 16-Tera 8-bit integer operations per second and power rating of 16384.4 Gint8/watt, is a great opportunity for enhancing existing applications and introducing new computer applications such as video compression and a 3-D geometry engine.

While CMOS technology is reaching physical limitations, optical technology still has a lot of room for advancement. Hence, optical technology has the potential for significant improvements through investment from the high-tech industry. It is expected that developments in electro-optical technology will significantly increase the performance of this and other optical architectures. This may result in electro-optical VMM devices including more SED elements and more optical units per device and at the same time operating at higher frequencies.

ACKNOWLEDGMENT

REFERENCES
1. D. Rairigh, "Limits of CMOS technology scaling and technologies beyond-CMOS," http://www.drlock.com/papers/cmos_survey.pdf.
2. G. G. Shahidi, "Evolution of CMOS technology at 32 nm and beyond," in *Proceedings of the IEEE Custom Integrated Circuits Conference* (IEEE, 2007), pp. 413–416.
3. H. Iwai, "CMOS technology after reaching the scale limits," in *Proceedings of the IEEE–EDS 8th International Workshop on Junction Technology* (IEEE, 2008), pp. 2–3.
4. B. Razavi, "Prospects of CMOS technology for high-speed optical communication circuits," IEEE J. Solid-State Circuits **37**, 1135–1145 (2002).

5. Y. Hoskote, S. Vangal, A. Singh, N. Borkar, and S. Borkar, "A 5-GHz mesh interconnect for a teraflops processor," Nat. Phys. **27**, 51–61 (2007).
6. A. Goren, S. Sariel, Y. Levit, S. Assaf, S. Liberman, B. Sender, T. Tzelnick, Y. Hefetz, E. Moses, and V. Machal, "Vector-matrix multiplication," U.S. Patent No. 2004/0243657 A1, December 2, 2004.
7. N. T. Shaked, S. Messika, S. Dolev, and J. Rosen, "Optical solution for bounded NP-complete problems," Appl. Opt. **46**, 711–724 (2007).
8. 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," Opt. Eng. (Bellingham) **46**, 108201:1–11 (2007).
9. S. Dolev and Y. Nir, "Optical implementation of bounded non-deterministic Turing machines," U.S. Patent No. 7, 130,093 B2, October 31, 2006.
10. S. Dolev and H. Fitoussi, "The traveling beams optical solutions for bounded NP-complete problems," in *Proceedings of the Fourth International Conference on Fun with Algorithms* (FUN2007), LNCS 4475 (Springer LNCS, 2007), pp. 120–134.
11. V. Oppenheim and R. W. Schafer, *Discrete Time Signal Processing*, 2nd ed. (Prentice Hall, 1999).
12. J. Bier, *Buyers Guide to DSP Processors*, 2005 ed. (Berkeley Design Technology, 2006).
13. Texas Instruments C64 performance benchmarks," http://focus.ti.com/dsp/docs/dspplatformscontentaut.tsp?sectionId=2&familyId=477&tabId=496.
14. S. Eisenbach, "Optical signal processing practical implementation and applications," Lenslet Marketing Presentation (Lenslet, 2003). http://www.acreo.se/upload/Publications/Proceedings/OE03/LENSLET-OE2003.pdf.
15. "Laboratory directed research and development program—FY 2006 Annual Report" (Oak Ridge National Labs, 2006). http://ornl.gov/adm/ldrd/FY2006/LDRD_FY2006.pdf.
16. J. Barhen and N. Imam, "Sensor data processing for tracking underwater threats using terascale optical core devices," in *Harbor Protection Through Data Fusion Technologies*, E. Shahbazian, G. Rogova, and M. J. DeWeert, eds. (Springer, 2009), pp. 267–282.
17. J. W. Goodman, *Introduction to Fourier Optics* (McGraw-Hill, 1996).
18. D. Psaltis, D. Casasent, and M. Carlotto, "Iterative color-multiplexed, electro-optical processor," Opt. Lett. **4**, 348–350 (1979).
19. D. Casasent, "Acousto-optic transducers in iterative optical vector-matrix processors," Appl. Opt. **21**, 1859–1865 (1982).
20. R. A. Athale, W. C. Collins, and P. D. Stilwell, "High accuracy matrix multiplication with outer product optical processor," Appl. Opt. **22**, 368–370 (1983).
21. D. G. Feitelson, *Optical Computing: A Survey for Computer Scientists* (MIT Press, 1988).
22. D. McAulay, *Optical Computer Architectures: The Application of Optical Concepts to Next Generation Computers* (Wiley-Interscience, 1991).
23. M. A. Karim and A. A. S. Awwal, *Optical Computing: An Introduction* (Wiley, 1992).
24. K. W. Hodapp and D. N. B. Hall, "Introduction to detectors: possible status in 2010–2020," in *Proceedings of the International Astronomical Union* (Cambridge Univ. Press, 2005), Vol. 1, pp. 40–51.
25. A. El Gamal and H. Eltoukhy, "CMOS image sensors," IEEE Circuits Devices Mag. **21**, 6–20 (2005).
26. J. Dubois, D. Ginhac, M. Paindavoine, and B. Heyrman, "A 10,000 fps CMOS sensor with massively parallel image processing," IEEE J. Solid-State Circuits **43**, 706–717 (2008).
27. A. Mambu, J. M. Bussat, M. West, B. C. Seller, M. Watanabe, and A. W. Kay, "An ultrahigh-speed one-dimensional detector for use in synchrotron radiation spectroscopy: first photoemission results," J. Electron Spectrosc. Relat. Phenom. **137**, 691–697 (2004).
28. T. Wallis, "Testing and characterization of image sensors at RAL," Ph.D. dissertation (The University of Surrey, Department of Physics, 2008).
29. L. Mengxiong, "5 GHz optical front end in 0:35 mm CMOS," Ph.D. dissertation (The University of Nottingham, Department of Physics, 2007).
30. I. T. Young, J. J. Gerbrands, and L. J. van Vliet, *Fundamentals of Image Processing* (Delft University of Technology, 2005).
31. M. Blacksell, J. Wach, D. Anderson, J. Howard, S. M. Collis, B. D. Blackwell, D. Andruczyk, and B. James, "Imaging photomultiplier array with integrated amplifiers and high-speed USB interface," Rev. Sci. Instrum. **79**, 10F506:1–4 (2008).
32. Multi anode photomultiplier tubes" (Hamamatsu http://sales.hamamatsu.com/multianode) (2009).
33. "Dalsa image sensor solutions," http://www.dalsa.com/sensors/, and http://www.dalsa.com/sensors/Products/custom.aspx (2009).
34. I. Hehemann, W. Brockherde, H. Hofmann, A. Kemna, and B. J. Hosticka, "A single chip optical CMOS detector with *in-situ* demodulating and integrating readout for next-generation optical storage systems," IEEE J. Solid-State Circuits **39**, 629–635 (2004).
35. M. Gschwind, "Chip multiprocessing and the cell broadband engine," IBM Research Report RC2931 (IBM, 2006).
36. *International Technology Roadmap for Semiconductors; Assembly and Packaging* (ITRS, 2005).
37. J. Wolf and J. Adams, 2005 *Packaging Roadmap Overview* (The International Electronics Manufacturing Initiative, INEMI, 2005).
38. D. Wentzlaff, P. Griffin, H. Hoffmann, L. Bao, B. Edwards, C. Ramey, M. Mattina, C. Miao, J. Brown, and A. Agrawal, "On-chip interconnection architecture of the tilera processor," Nat. Phys. **27**, 15–31 (2007).
39. J. Kim, I. Verbauwhede, and M. F. Chang, "Design of an interconnect architecture and signaling technology for parallelism in communication," IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **15**, 881–894 (2007).
40. F. Mederer, R. Jager, H. J. Unold, R. Michalzik, P. Schnitzer, D. Wiedenmann, and K. Ebeling, "3-Gb/s data transmission with GaAs VCSELs over PCB integrated polymer waveguides," IEEE Photon. Technol. Lett. **13**, 1032–1034 (2001).
41. T. Hino, R. Kuribayashi, Y. Hashimoto, T. Sugimoto, J. Ushioda, J. Sasaki, I. Ogura, I. Hatakeyama, and K. Kurata, "A 10 Gbps×12 channel pluggable optical transceiver for high-speed interconnections," in *Proceedings of the IEEE Electronic Components and Technology Conference* (IEEE, 2008), pp. 1838–1843.
42. "Infiniband technology overview," http://www.infinibandta.org/about/.
43. 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)," Electron. Lett. **23**, 916–917 (1987).
44. 850 NM 4.25 G VCSEL LC TOSA, JDSU, www.jdsu.com/product-literature/pl-slx-00-sg0-cx_ds_cc_ae.pdf.
45. Lasermate Group, Inc. "VSCEL data sheets," http://www.lasermate.com/VCA1XNA1G.html.
46. P. N. Singh, A. Kumar, C. Debnath, and R. Malik, "20 mW, 125 Msps, 10 bit pipelined ADC in 65 nm standard digital CMOS process," in *Proceedings of the IEEE Custom Integrated Circuits Conference* (IEEE, 2007), pp. 189–192.
47. M. H. Zarifi, J. Frounchi, S. Asgarifar, M. B. Nia, and F. A. Dehkhoda, "High speed low power analog to digital converter for biomedical application," in *Proceedings of Canadian Conference on Electrical and Computer Engineering* (IEEE Canada, 2008), pp. 1269–1272.
48. "125 Msps 14-bit wideband ADC," Linear Technology, http://www.linear.com/.
49. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and R. C. Stein, *Introduction to Algorithms*, 2nd ed. (MIT Press, 2001).
50. "Basic local alignment search tool (BLAST)," http://www.ncbi.nlm.nih.gov/blast/Blast.cgi.