AP
Asian Journal of Physics Vol. 23, No 3 (2014) 403-414

About the future of optical computing

Joseph Shamir
Department of Electrical Engineering
Technion – Israel Institute of Technology, Haifa 32000, Israel

Interest in signal processing and computing using light was triggered by the invention of the laser and flourished during about three decades starting in 1960. Optical methods were first introduced for analog signal processing paradigms but early attempts toward optical digital processing failed due to fundamental misconceptions. To assess the trends for the future, the first part of the paper is devoted mainly to a historical overview while the second part shows that some novel concepts and advanced technology may revitalize optical signal processing also within the digital computing world. This latter development is demonstrated by digital logic functions implemented on simple electro-optic networks.

1 Introduction

For many centuries optical methods were used for imaging purposes employing technologies developed by experienced craftsmen. Fundamental mathematical relations were introduced only in the 19th century by Ernst Abbe and Lord Rayleigh [1]. For a long time these mathematical relations were exploited only for improving imaging systems and no other applications were explored. This state of affairs changed at the middle of the 20th century when a close relationship between light propagation and processes involved in communications were established and concluded in several publications [2, 3]. These publications demonstrated how communications theory, mainly Fourier analysis, can be applied to optics. However, for further progress a highly coherent light source was needed and this was achieved with the invention of the laser. The first attempt to implement a real application was made in 1960 by L J Cutrona and collaborators [4]. This work was followed by B VanderLugt in 1964, when he introduced the optical correlator containing the spatial matched filter [5]. Combined with laser holography this was the beginning of an exciting era of analog optical signal processing. Unfortunately, an intensive research effort that continued for about two decades was significantly dwindled due to technical difficulties, fundamental limitations and the harsh competition with electronic computing. Moreover, attempts to copy the paradigm of digital electronic computation to digital photonic computation were doomed from the start because of the difference between the physical nature of electrons and photons.

Today, further development of electronic computing started to run into severe difficulties causing a revived interest in optical computing along with quantum computing and other possible paradigms. The information in electronic computers is moved and processed by electrons. Electrons are charged fermions and therefore, each electron "feels" the presence of all the others and is affected by them. In other words, the flow of electrons can be easily controlled and switched from one route to another by other electrons. This is the main attribute of electrons in computing but it also is their detriment. We also must spend energy to move electrons from one place to another and their motion can be altered unintentionally if not properly shielded. In contrast, photons are bosons with no charge and one photon does not care about others. They can cross each other's path with no interaction and after they were generated in a light source they propagate with the speed of light with no need for additional energy supply. Considering these differences it is obvious why the attempts to copy the processes involved in electronic computation into computation with photons failed. However, with respect to computing, photons perform a very complex mathematical computation while propagating in space. They *solve the wave equation* with given boundary conditions.

The next two sections review some historical markers of optics in computing while the rest of the paper is devoted to the relatively new paradigm of Directed Logic (DL) to indicate a possible rout to incorporate optics into novel computing architectures.

Corresponding author :
e-mail: jsh@ee.technion.ac.il (Joseph Shamir)

2 Background on optical computing

As indicated in the introduction, a propagating photon solves the wave equation under given boundary conditions determined by the optical system architecture (free space, optical system, set of waveguides, etc.) and the information input presented usually over a single, 2D surface (not necessarily planar). This is the case also in the work cited earlier [2-5]. In all these processors photons are generated at the input as particles, they propagate through space and optical systems as waves and are finally detected as particles. Hence we coined the name Wave-Particle Duality (WPD) processor [6, 7] to all this class of signal processors. The main attributes of WPD processors are their extended parallelism with huge information content, energy efficiency (no energy is spent for the propagation) and high speed.

A typical WPD processor is the optical correlator (Fig 1). In a digital processor we would take the input function and perform on it an energy consuming fast Fourier transform (FFT), multiply the result with the reference (filter) function (again an energy dissipating operation), then perform another FFT and, finally read out the result. By contrast, in the WPD optical correlator light reads out the input information and processes it by propagating through the system with the speed of light with no additional energy input until it is detected at the output plane. It is important to note that this mathematical operation, essentially the solution of a wave equation, is performed by each photon but we need several photons to obtain a statistically meaningful result at the output. Although VanderLugt [5] did not call this a WPD processor, its advantages were clear to him and to a whole community of scientists of that time. The initial excitement lasted for about a decade and this was followed by another decade during which new ideas were introduced to improve performance and extend possible applications [8-22]. Nevertheless, optical correlators are still not in wide use. One of several reasons is that the optical correlator is basically an analog processor and not a digital one.

A WPD processor with a more digital flavor is the optical matrix-matrix multiplier for which one possible implementation is shown in Fig 2. The input plane, H, contains an $n \times n$ matrix of holograms illuminated by a reference wave R. The reconstructed wave is projected onto an $n \times n$ matrix presented on a spatial light modulator (SLM). The light transmitted by the SLM is focused onto an $n \times n$ detector array, D, in such a way that the ij-th element of D receives light originating only from the ij-th element of H. It can be shown [23] that the result obtained on D is a matrix multiplication of the two matrices, one on H and the other one on the SLM. Other interpretations of this architecture are a weighted n^4 interconnection network between n^2 sources and n^2 receivers [24-26] or a neural network [27]. Within these interpretations the interconnection weights are determined by the holograms on H while the inputs are displayed over the SLM. The cited literature [6, 7, 23-27] introduces several other architectures serving the same purpose and it provides a detailed analysis of their operation and characteristics.

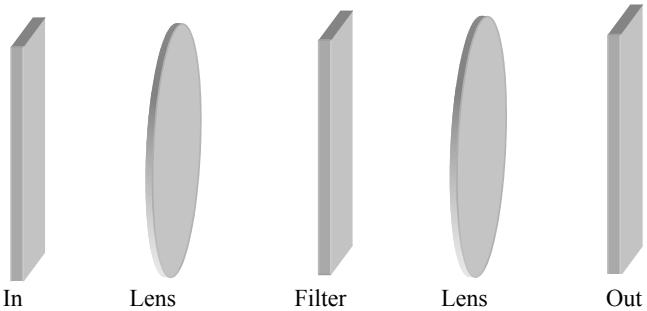


Fig 1. Optical correlator. A two-dimensional input function is presented over the input surface (In) and Fourier transformed by the first lens. This transform is multiplied by the function displayed over the filter plane and the product is again transformed to yield the correlation function over the output surface.

The high efficiency and elegancy of WPD processors are very appealing but, nevertheless, they were not accepted for widespread technological applications. The fact that these processors are basically analog is not the only reason for this state of affairs. A whole list of additional reasons is provided in Ref [28] with three most notable: The need for bulky optical systems, technological difficulties and, obviously, the harsh competition of electronic computing, which, historically, preceded optical computing.

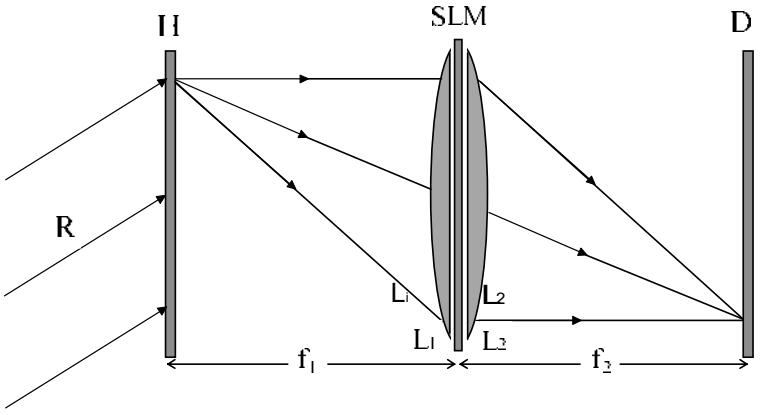


Fig 2. A matrix-matrix multiplier or an n^4 interconnection network.

3 Rediscovering optical computing

Recognizing the attributes and limitations of optics it became evident that optics should not be required to do the job alone and its attributes must be complemented with those of other media, particularly digital electronics. The primary advantage of optics lies in linear processes at high speed with massive parallelism and low expenses in energy (WPD processor) while the execution of non-linear calculations is more suitable for digital electronic processing. With this in mind the hybrid electro-optic correlator was introduced in 1990 [29-31] to implement complicated learning and optimization algorithms. The optical WPD processor with a coherent optical correlator architecture performed the correlations while all non-linear processes and calculations were implemented electronically in a digital computer. This paradigm, however, still uses bulky systems and addresses processes that are basically analog. Our main objective in this paper is to consider optics for essentially digital operations.

As indicated above early research to use optics for digital computing failed because it attempted a direct translation of electronics into optics. Deviation from this approach is marked by the introduction of reversible logic [32-35]. Conventional logic gates discard information and, as a consequence, are dissipative in terms of information and energy [32]. In contrast, reversible logic is based on lossless logic elements. Probably the best known element of this kind is the Fredkin gate [34]. Essentially, a Fredkin gate is a cross-bar switch adapted to perform logic operations. The schematic representation of a Fredkin gate is depicted in Fig 3. Traditionally a Fredkin gate has three information channels, the control channel C and two data input and output channels. While the signal in the control channel is transmitted unchanged it controls the two other channels, that are referred to as the data channels, in the following way:

$$b_1 = a_1; \quad b_2 = a_2 \text{ If } C = 0; \quad b_1 = a_2; \quad b_2 = a_1 \text{ If } C = 1. \tag{1}$$

It is appropriate to note that this definition is the complement of the original definition by Fredkin and Toffoli [34] since we found the present definition more convenient for our applications. The Fredkin gate is lossless and reversible in the sense that if all three outputs are detected, the input can be reconstructed. The energy needed to change the state of the gate between $C = 0$ and $C = 1$ is ignored in most publications.

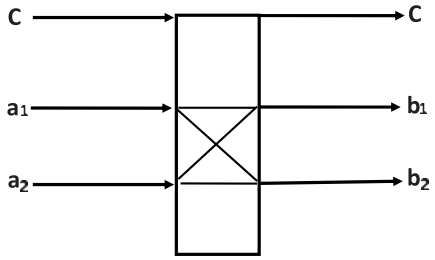


Fig 3. Basic definition of the Fredkin gate

The controlled switching operation of the Fredkin gate can be represented mathematically in several ways [35], out of which we found it convenient to employ here the matrix representation given as,

$$\begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = M(C) \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \tag{2}$$

where,

$$M(0) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix};\ M(1) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{3}$$

The original idea of Fredkin gates was to use the two input channels for the data and the control channel, *C*, for controlling the gate's state. However, already from the start, the proposal to use these reversible gates for the implementation of traditional Boolean logic functions [34-36] mixed up the role of the data and control channels. In principle, Boolean logic functions can be performed by one, or a combination of several reversible elements. The elements of Fig 4 depict possible layouts to implement the three primitive logic operations, NOT, AND and OR, using a single Fredkin gate. Looking at the architectures of these implementations we observe that each logic operation is accompanied by two additional outputs filling up all the three output channels of the gate. In the terminology of reversible logic the outputs irrelevant to the specific operation are considered garbage. Unfortunately, by disposing the garbage reversibility is lost and energy dissipation can no longer be avoided. Furthermore, the mixing of the control and data channels is problematic since the physical nature of the control channel is usually completely different than the two data channels. In practice, while the original data channels flow with no theoretical lower limit on energy dissipation, the control channel must use energy, at least for changing the state of the gate. Moreover, to maintain a certain state, the switching energy between the two states must exceed significantly the thermal noise of the order of kT (k is Boltzman's constant and T is the absolute temperature). In practice this energy will be much higher than the kT ln 2 theoretical limit [32, 35] on a single logic operation. Obviously, these traditional ways to implement Boolean logic operations with reversible logic gates reinstate dissipation of at least the same order of magnitude as conventional logic gates. Thus, the whole idea of reversible computing is contradicted.

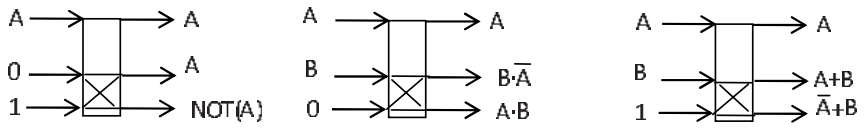


Fig 4. Implementation of logic elements using a Fredkin gate. (a) NOT gate, (b) AND gate and (c) OR gate.

4 Directed logic networks

To alleviate the above indicated difficulties encountered with conventional reversible logic the paradigm of directed logic (DL) [37] networks was introduced. A DL network comprises an array of interconnected Fredkin gates where input data is fed to the control channel of the gates. In the original

approach of DL a single conveyor light beam is introduced into one of the data ports of the system and logic functions are implemented by this light as it propagates through the consecutive Fredkin gates. Thus, information in the gate channels flows along the net with no obstruction as in a WPD processor and the need to mix signals of different physical nature is eliminated. Two examples of DL networks for the implementation of Boolean logic functions are shown in Fig 5. Note that together with the required function we obtain also its complement on a different output port. An apparent disadvantage of the DL approach as compared to traditional Fredkin gate logic operations is a loss of the idealized simplicity indicated in Fig 4. Within the DL paradigm we need three gate elements instead of just one and one of the input signals must be fed to two elements in parallel. However, in view of the above discussion it is not clear what is the actual penalty paid for the simplicity of the original Fredkin gate in terms of practical system complexity, processing speed and energy dissipation. Energy dissipation is particularly large when the state of the gate is switched. While in the original Fredkin gate implementations the gates are switched during the input and the processing operation, in DL the gate is switched only with the input data. It turns out that the advantages of DL significantly out weight its disadvantages and the feasibility of the practical optical implementation of DL elements and small networks has been already demonstrated in several laboratory experiments [38-42].

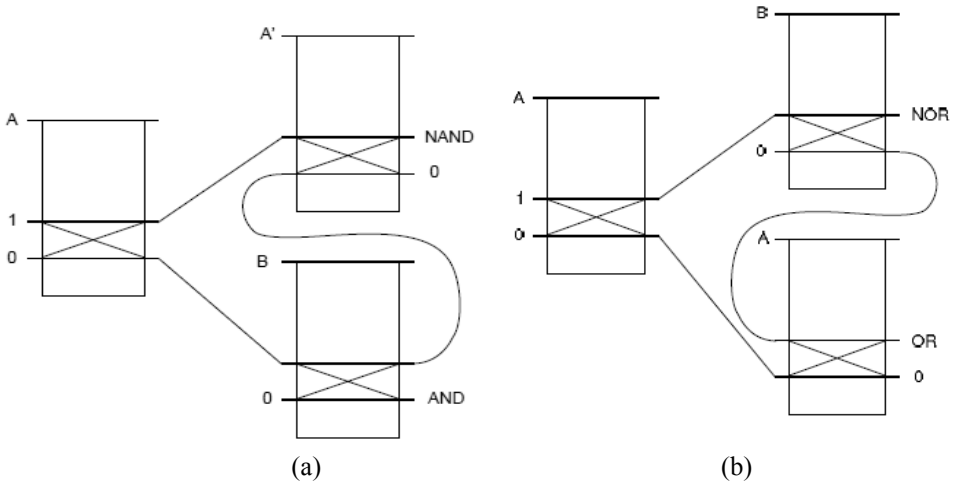


Fig 5. DL implementation of Boolean logic gates (a) AND gate (b) OR gate. A and B are the input data.

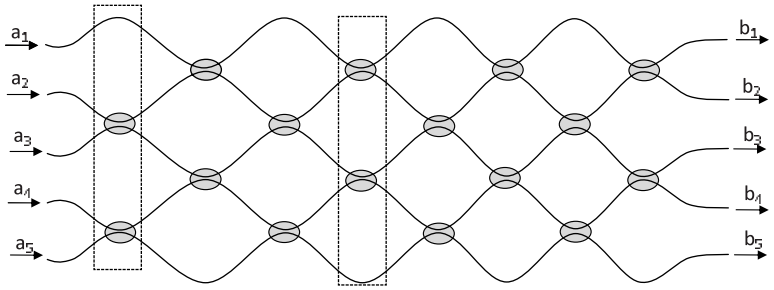


Fig 6. A generic interconnection network. Each dashed rectangle delineates a single switching layer.

The impressive capability of the DL paradigm is demonstrated here by a modified DL architecture based on a programmable waveguide interconnection network [43, 44]. A generic interconnection network composed of a waveguide array interconnected by controllable waveguide couplers is shown in Fig 6. Each 2×2 coupler is essentially a Fredkin gate [36] and there is a clear distinction between the data input ports

(the original control channels) and the waveguides that route the light. For brevity a waveguide will be referred to as a light channel while the control channels will be the data input ports.

In the past it was shown that a complete permutation network between $2N$ input channels and $2N$ output channels can be implemented by $2N$ switching layers [43]. Two such layers are marked on the figure by two dashed rectangles. The usefulness of this architecture for arithmetic processors [45] was also demonstrated and here we focus on logic operations [46]. Within the original DL paradigm we supply a light input vector with only one non-zero element and the result of a logic operation is detected on one of the output ports. As was indicated earlier, a Boolean logic operation is essentially a non-linear process. In a DL processor the non-linearity is achieved by routing the conveyor light to the appropriate output port.

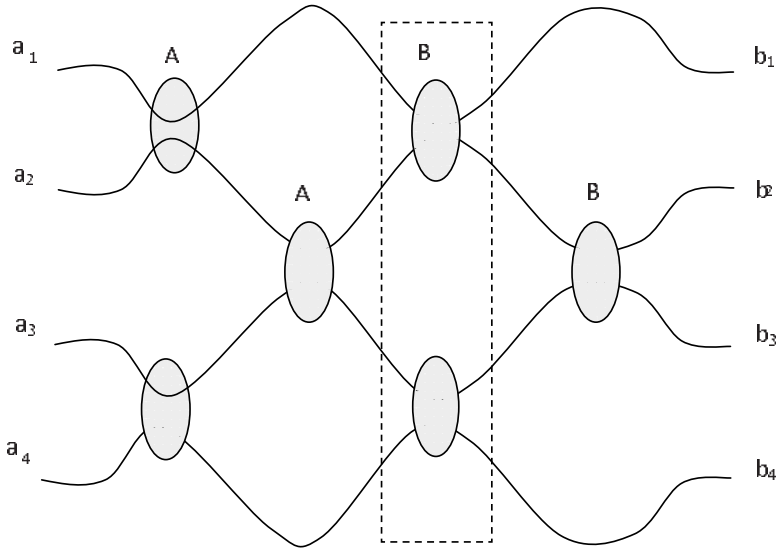


Fig 7. A 4 × 4 interconnection network implementing logic operations (see text for detail).

While the original DL procedure already has a significant improvement over the basic hard-wired DL network here we allow several non-zero input elements to improve the processing efficiency and to perform several logic operations in parallel. For the implementation of each logic function described below we employ a limited section of the complete interconnection net, allowing for additional operations to be performed in parallel.

A specific $N = 2$ section of the interconnection network is shown in Fig 7. To facilitate all possible permutations, such an array needs four layers of gates along the length of the net in the direction of the light propagation. For the implementation of a primitive logic operation we need only one light channel for the input and one for the output. We also do not need all the possible gates and all the four channels.

In the example of Fig 7, we show the input data A and B as fed to two gate elements each. As in the case of Eq (3) the matrix representation can be expanded to a $2N \times 2N$ matrix for each switching layer. In our example, $N = 2$ generates a 4×4 element matrix for each switching layer. With the four switching layers we have to multiply these four matrices to obtain the complete transfer matrix of the net. Denoting the transfer matrix of the whole network by $M(A, B)$, where A and B represent the values of the input data, we shall have the four possibilities, $M(0, 0)$, $M(1, 0)$, $M(0, 1)$ and $M(1, 1)$. When all the switches controlled by the input data are off, $M(0, 0)$ is obviously the unit matrix. The two mixed states, $M(1,0)$ and $M(0,1)$ are obtained as,

$$M(1,0)=\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix}\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix}\begin{pmatrix}0&1&0&0\\1&0&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix}=\begin{pmatrix}0&1&0&0\\0&0&1&0\\1&0&0&0\\0&0&0&1\end{pmatrix}=M(0,1) \tag{4}$$

The two mixed states are equal since the only difference between them is the position of the unit matrix. In a similar way we calculate the fourth matrix to be,

$$M(1,1)=\begin{pmatrix}0&0&1&0\\1&0&0&0\\0&1&0&0\\0&0&0&1\end{pmatrix} \tag{5}$$

Applying these matrices to the four-element input vector, a_j (j = 1, 2, 3, 4) we obtain the respective four-element output vectors $b_j(A, B)$,

$$\begin{pmatrix}b_1(0,0)\\b_2(0,0)\\b_3(0,0)\\b_4(0,0)\end{pmatrix}=\begin{pmatrix}a_1\\a_2\\a_3\\a_4\end{pmatrix};\quad\begin{pmatrix}b_1(1,1)\\b_2(1,1)\\b_3(1,1)\\b_4(1,1)\end{pmatrix}=\begin{pmatrix}a_3\\a_1\\a_2\\a_4\end{pmatrix};$$

$$\begin{pmatrix}b_1(1,0)\\b_2(1,0)\\b_3(1,0)\\b_4(1,0)\end{pmatrix}=\begin{pmatrix}b_1(0,1)\\b_2(0,1)\\b_3(0,1)\\b_4(0,1)\end{pmatrix}=\begin{pmatrix}a_2\\a_3\\a_1\\a_4\end{pmatrix} \tag{6}$$

The output vectors of Eq (6) represent a specific example of how the input vector can be manipulated by the input data. Viewing the process as a network of logic processors we recall that we have two inputs, A and B, and we need only one output. Let us assume first a light input vector a = (1, 0, 0, 0) and observe the light output channels. Comparing the results with the truth table of basic logic operations (Fig 8) we observe that the output vector will implement a NOR operation on b_1, a XOR operation on b_3, and an AND operation on b_2. It is important to note that by detecting only the outputs on b_2 and b_3 we have actually implemented a Half Adder with the sum, S and carry C that can be represented by the basic operations as,

$$S = A \oplus B; \quad C = A \,.\, B \tag{7}$$

A	B	$A + B$	$A \oplus B$	$A \bullet B$	$\overline{A + B}$	$\overline{A \bullet B}$
0	0	0	0	0	1	1
1	0	1	1	0	0	1
0	1	1	1	0	0	1
1	1	1	0	1	0	0

Fig 8. Truth table for three primitive logic operations: OR, XOR, AND, NOR and NAND.

The implementation of a Full Adder needs two simultaneous output ports, either of which or both can be zero or one. This required a modification of the original DL approach to allow more than one of the light input vector components to be nonzero. For example, if we take a = (0, 1, 1, 0), we shall implement an OR operation on b_1 and, at the same time, an NAND operation on b_2 and an NXOR operation on b_3. The significant improvement achieved by more than a single light channel input can be demonstrated with the input vector a = (1, 0, 0, 1) and a simplification of the active gate layout as shown in Fig 9. The AND and OR operations are implemented in parallel using only three gates.

Finally, we present here one of several possible architectures to implement a Full Adder as shown in Fig 10 together with the adder truth table. In this case the light input is a five-component vector, out of which three are "1" and only two are zero. Columns 1-5 of the table show the outputs from the 5 channels corresponding to the various inputs. The input bits are A and B while C_{in} is the carry from a previous stage. The adder outputs are the sum, S and carry out C_o that can be expressed as a combination of the basic logic operations as,

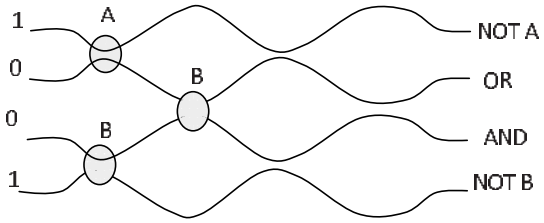


Fig 9. Parallel implementation of OR and AND operations. For clarity only the active gates are marked

$$S = (A \oplus B \oplus C) + A \,.\, B \,.\, C, \tag{8}$$

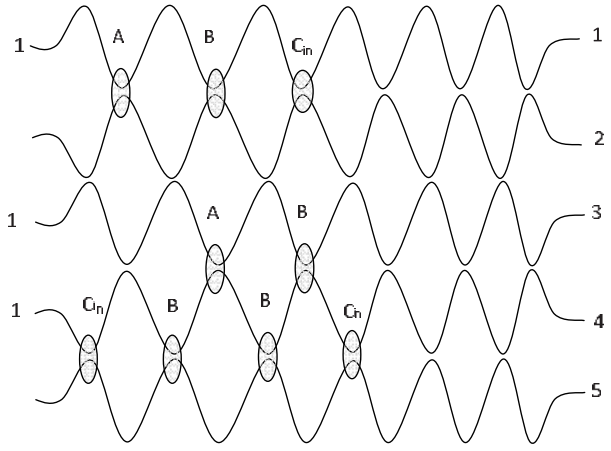
and

$$C_o = (A \,.\, B + A \,.\, C + B \,.\, C) + A \,.\, B \,.\, C. \tag{9}$$

Observation of the table reveals that the sum, S is obtained on channel 2 while the carry out, C_o, corresponds to channel 5. This is a good example which demonstrates the power of this approach. The complete Full Adder was implemented on a nominally lossless network and we pay with energy only for the data input and the detection of the output data. The detected signals can be fed into the next stage of the processor.

5 Conclusions

The introduction of coherent laser radiation more than half a century ago lead to a revolution in understanding the physics of light. It is safe to state that the first decade after the invention of the laser was the period of childhood of electro-optics where all the scientific groundwork was laid down. Coherent light enabled wide use of high precision interferometry, and various methods of non-destructive evaluation. While these applications of light were already practiced in the past and the laser contributed to their improvement and extension, coherent optical signal processing and holography played a central role in generating new insight and physical understanding.



A	B	C_i	S	C_o	1	2	3	4	5
0	0	0	0	0	1	0	1	1	0
1	0	0	1	0	0	1	1	1	0
0	1	0	1	0	0	1	1	1	0
0	0	1	1	0	0	1	1	1	0
1	1	0	0	1	1	0	1	0	1
1	0	1	0	1	1	0	1	1	1
0	1	1	0	1	1	0	1	1	1
1	1	1	1	1	0	1	1	0	1

Fig 10. A section of an interconnection array for the implementation of a Full Adder and its truth table. For illustration purposes in a larger section of the net only the gates necessary for this application are shown

Nevertheless, from a technological point of view, after an initial excitement, interest in optical signal processing declined due to the bulky and inflexible architectures needed and the vigorous competition by microelectronics based digital computing.

The decline of interest in optical computing was reversed when it was realized that optics cannot and should not attempt to replace electronic computing. Instead, optics should complement electronics wherever it can perform better, paving the way to integrated electro-optic systems. Within this general approach, the present paper discussed in some detail the relatively novel paradigm of directed logic demonstrating its potentials to reduce energy dissipation and increase the speed and parallelism in evaluating logic operations. Miniaturization is now also possible by using nano-optic technologies including nano lasers, metamaterials and plasmonics.

References

1. Strutt J W (Lord Rayleigh), Investigations in optics, with special reference to the spectroscope, *Phil Mag*, 8(1880)40-55. Reprinted in, The collection of optics papers of Lord Rayleigh, J W Strutt, Ed, Optical Society of America, Washington, DC, (1994).
2. Elias P, Gray D S, Robinson D Z, Fourier treatment of optical processes, *J Opt Soc Am,* 42(1952)127-134.
3. Elias P, Optics in communication theory, *J Opt Soc Am,* 43(1953)229-232.
4. Cutrona L J, Leith E N, Palermo C J, Porcello L J, Optical data processing and filtering systems, *IRE Trans Inf Theory*, IT- 6(1960)386-400.
5. VanderLugt A B, Signal detection by complex spatial filtering, *IEEE Trans Inf Theory*, IT-10(1964)139-145.
6. Caulfield H J, Shamir J, Wave-particle duality considerations in optical computing, *Appl Opt*, 28 (1989)2184-2186.
7. Caulfield H J, Shamir J, Wave-particle duality processors—characteristics, requirements and applications, *J Opt Soc Am* A, 7(1990)1314-1323.
8. Caulfield H J, Maloney W T, Improved discrimination in optical character recognition, *Appl Opt*, 8(1969)2354-2356.
9. Braunecker B, Hauch R, Lohmann A W, Optical character recognition based on nonredundant correlation measurements, *Appl Opt*, 18(1979)2746-2753.
10. Hester C F, Casasent C, Multivariant technique for multiclass pattern recognition, *Appl Opt*, 19(1980)1758-1761.
11. Hsu Y, Arsenault H, Optical pattern recognition using circular harmonic expansion, *Appl Opt*, 21(1982)4016-4019.
12. Rosen J, J Shamir J, Circular harmonic phase filters for efficient rotation-invariant pattern recognition, *Appl Opt*, 27(1988)2895-2899.

13. Casasent D, Psaltis D, New optical transforms for pattern recognition, *Proc IEEE*, 65(1977)77-84.
14. Rosen J, Shamir J, Scale invariant pattern recognition with logarithmic radial harmonic filters, *Appl Opt*, 28 (1989)240-244.
15. Casasent D, Unified synthetic discriminant function computational formulation, *Appl Opt*, 3(1984)1620-1627.
16. Mahlab U, Shamir J, Phase only entropy optimized filter by simulated annealing, *Opt Lett*, 14(1989)146-148.
17. Mahlab U, Shamir J, Optical pattern recognition based on convex functions, *J Opt Soc Am A*, 8(1991)1233-1239.
18. Vijaya Kumar B V K, Tutorial survey of composite filter designs for optical correlators, *Appl Opt*, 31(1992) 4773-4801.
19. Gabor D, Light and information, in Progress in Optics, (E Wolf, Ed), Vol 1, (1964)109-153.
20. Piestun R, J Shamir J, Synthesis of three-dimensional light-fields and applications, *Proc IEEE*, 90(2002)220-244.
21. Shamir J, Fundamental speed limitations on parallel processing, *Appl Opt*, 26(1987)1567.
22. Lohmann A W, Marathay A S, Globality and speed of optical parallel processors, *Appl Opt*, 28(1989)3838-3842.
23. Shamir J, Caulfield H J, Johnson R B, Massive holographic interconnection networks and their limitation, *Appl Opt*, 28(1989)311-324.
24. Caulfield H J, Parallel N4 weighted optical interconnections, *Appl Opt*, 26(1987)4039-4040.
25. Shamir J, Caulfield H J, Parallel optical processors-some basic considerations, *Int J Opt Comp*, 2(1991)73-78.
26. Jeon H I, Shamir J, Johnson R B, Caulfield H J, Kinser J, Hester C, Temmen M, The use of fixed holograms for massively- interconnected, low-power neural networks, in Neural Networks for Perception Vol 2, H Wechsler, Ed pp. 282309, Academic Press, Boston, (1992).
27. Caulfield H J, Shamir J, Fixed hologram neural networks" in Real- time optical information processing, B Javidi and J L Horner, Eds, pp. 255305, Academic Press, 1994,
28. Shamir J, Half a century of optics in computinga personal perspective, *Appl Opt*, 52(2013)600-612
29. Rosen J, Mahlab U, Shamir J, Adaptive learning with joint transform correlators, *Opt Eng*, 29(1990)1101-1106.
30. Shamir J, Adaptive pattern recognition correlators, *Opt Eng*, 36(1997)2675-2689.
31. Shamir J, Iterative procedures in electrooptical pattern recognition, in Optical Pattern Recognition, F T S Yu, Ed, pp. 221-261, Cambridge University Press, Cambridge, UK, 1998.
32. Landauer R, Irreversibility and heat generation in the computing process, *IBM Jornal of Research and Development*, 5(1961)183-191.
33. Bennett C H, Logical reversibility of computation, *IBM Jornal of Research and Development*, 17(1973)525-532.
34. Fredkin E, Toffoli T, Conservative Logic, *Int J Theoret Phys*, 21(1982)219-253.
35. Feynman R P, Quantum mechanical computing, *Optics News* 11, 11-20 February (1985).
36. Shamir J, Caulfield H J, Miceli W, Seymour R J, Optical Computing and the Fredkin Gate, *Appl Opt*, 25(1986) 1604-1607.
37. Hardy J, Shamir J, Optics inspired logic architecture, *Opt Exp*, 15(2007)150-165.
38. Xu Q, Soref R, Reconfigurable optical directed-logic circuits using microresonator-based optical switches, *Opt Exp*, 19(2011)5244-5259.
39. Zhang L, Ji R Q, Jia L X, Yang L, Zhou P, Tian Y H, Chen P, Lu Y Y, Jiang Z Y, Liu Y L, Fang Q, Yu M B, Demonstration of directed XOR/XNOR logic gates using two cascaded microring resonators, 35(2010)1620-1622.
40. Tian Y H, Zhang L, Ji R Q, Yang L, Zhou P, Chen H T, Ding J F, Zhu W W, Lu Y Y, Jia L X, Fang Q, Yu M B, Proof of concept of directed OR/NOR and AND/NAND logic circuit consisting of two parallel microring resonators, *Opt Lett*, 36(2011) 1650-1652 1650.
41. Zhang L, Ji R Q, Tian Y H, Yang L, Zhou P, Lu Y Y, Zhu W W, Liu Y L, Jia X L, Fang Q, Yu M B, Simultaneous implementation of XOR and XNOR operations using a directed logic circuit based on two microring resonators, *Opt Exp*, 19(2011)6524-6540.

42. Papaioannou S, Vyrsokinos K, Tsilipakos O, Pitilakis A, Hassan K, Weeber J C, Markey L, A Dereux A, Bozhevolnyi S I, Miliou A, Kriezis E E, Pleros N, A 320 Gb/s-Throughput Capable 2×2 Silicon Plasmonic Router Architecture for Optical Interconnects, *J Lightwave Tech*, 29(2011)3185-3195.
43. Shamir J, Caulfield H J, High-efficiency rapidly programmable optical interconnections, *Appl Opt*, 26(1987)1032-1037.
44. Shamir J, Three-dimensional optical interconnection gate array, *Appl Opt*, 26(1987)3455-3457.
45. Mirsalehi M M, Shamir J, Caulfield H J, Residue arithmetic processing utilizing optical Fredkin gate arrays, *Appl Opt*, 26(1987)3940-3946.
46. Shamir J, Parallel optical logic operations on reversible networks, *Opt Comm*, 291(2013)133-137.

[*Received*: 1.12.2013; *accepted*: 15.1.2014]