Research Article | Vol. 26, No. 21 | 15 Oct 2018 | OPTICS EXPRESS 28002
Optics EXPRESS

Automated logic synthesis for electro-optic logic-based integrated optical computing

ZHOUFENG YING,[1,4] ZHENG ZHAO,[1,4] CHENGHAO FENG,[1] ROHAN MITAL,[1] SHOUNAK DHAR,[1] DAVID Z. PAN,[1] RICHARD SOREF,[2] AND RAY T. CHEN[1,3*]

[1]*Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, Texas 78712, USA*
[2]*Department of Engineering, University of Massachusetts Boston, Boston, Massachusetts 02125, USA*
[3]*Omega Optics, Inc., 8500 Shoal Creek Blvd., Bldg. 4, Suite 200, Austin, TX 78757, USA*
[4]*These authors contributed equally to this paper*
**chenrt@austin.utexas.edu*

Abstract: Integrated optical computing attracts increasing interest recently as Moore's law approaches the physical limitation. Among all the approaches of integrated optical computing, directed logic that takes the full advantage of integrated photonics and electronics has received lots of investigation since its first introduction in 2007. Meanwhile, as integrated photonics matures, it has become critical to develop automated methods for synthesizing optical devices for large-scale optical designs. In this paper, we propose a general electro-optic (EO) logic in a higher level to explore its potential in integrated computing. Compared to the directed logic, the EO logic leads to a briefer design with shorter optical paths and fewer components. Then a comprehensive gate library based on EO logic is summarized. At last, an And-Inverter Graphs (AIGs) based automated logic synthesis algorithm is described as an example to implement the EO logic, which offers an instruction for the design automation of high-speed integrated optical computing circuits.

© 2018 Optical Society of America under the terms of the OSA Open Access Publishing Agreement

1. Introduction

Directed logic that functions as a network of switching elements is regarded as a logic paradigm to take full advantage of integrated photonics and electronics [1,2]. Each element in the circuit determines the direction of the light and the output function is directly linked to the operands along the path. The operation of each element is independent of the states of other elements in the circuit. With all the control operands applied simultaneously, the delay time is much smaller when compared with electronic counterparts and consequently leads to a substantial reduction in the whole latency, in contrast to transistors based electronic logic circuits where delays are cascaded. The advantages of electronics lie in the accessibility and maturity of electrical components and CMOS fabrication line. The advantage of photonics in a directed logic is also well retained since the light flows through each element to transport signals and achieve computing at ultrafast speed. Many architectures and generic design methods have been studied to realize this type of electric-optical computing circuit and various structures based on directed logic have been demonstrated experimentally [3–9].

However, one disadvantage of directed logic is that the building blocks of the directed logic provided in [1] are very limited so that many simple logic functions can only be achieved by complex circuits. In addition, many advanced optical logic gates have been emerging recently that can contribute to efficiency in the circuit design. Another reconfigurable optical directed logic architecture which is mapped directly from truth tables is proposed in [10]. Compared to the original logic gate, the advantages of this architecture include the shorter optical paths, better scalability, and ease of programming. However, this method suffers from the huge number of components which is proportional to N^2, where N is the number of variables/electrical operands.

#338410 https://doi.org/10.1364/OE.26.028002
Journal © 2018 Received 11 Jul 2018; revised 10 Sep 2018; accepted 20 Sep 2018; published 10 Oct 2018

Optics EXPRESS

On the other hand, as integrated photonics matures, more and more fundamental components, such as modulators [11], photodetectors [12], and interconnects [13], are available nowadays with satisfying performance. Foundries also start to provide services in integrated photonics to help customers and researchers develop and fabricate mature chips with guaranteed performance just like the CMOS electronics. At the same time, electronic design automation (EDA) companies are considering automating the design and optimization of large-scale integrated optical circuits, which is called the electronic/photonic design automation (EPDA).

In this paper, we introduce a general EO logic that makes the most of the electronics and photonics with abundant mature logic gates to explore its potential in integrated optical computing. Compared to the directed logic, the electro-optic (EO) logic has the advantages of briefer designs, much shorter optical paths, and higher scalability. The high-level architecture is proposed and then a comprehensive gate library based on EO logic is summarized. Finally, an And-Inverter Graphs (AIGs) based automated logic synthesis algorithm is described as an example to implement the EO logic, which offers an instruction for the design automation of a high-speed integrated optical computing circuits.

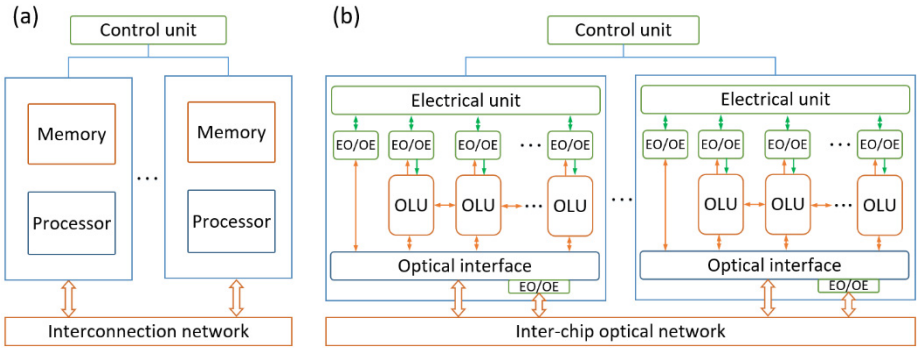

Fig. 1. Architecture of (a) a SIMD parallel computer and (b) proposed integrated EO computing system. OLU: optical logic unit.

2. EO computing architecture

Figure 1 shows a conventional single instruction multiple data (SIMD) architecture, which employs a central control unit, multiple processors and memories, and an interconnection network [14]. The control unit broadcasts instructions to all processors and the interconnection network. It enables the communication between processors so that the calculation results from one processor may be transferred to another processor, where they become the operands in a subsequent instruction. A similar architecture is proposed for an integrated EO computing system by replacing most electrical modules with optical counterparts, as shown in Fig. 1(b). As the core logic parts in the system, the optical logic units, which can be identical or different, are cascaded in series to function as a cluster. They communicate directly using light in the most efficient way so that it will not introduce much latency as long as no OE/EO conversion modules are inserted in between. OE/EO modules actually act as channels through which each optical logic unit will be controlled by electrical units. The operation of the simultaneous injection of all these operands results in the minimal latency in computing. The results then will either be converted back into electrical signals to electrical units or be directly transported to the optical interface, which consists of optical routers, multiplexers, demultiplexers, and combiners, etc. It could travel a long way to another chip through the inter-chip optical network. An OE/EO module is also added between the optical interface and the inter-chip optical network in case that signal processing such as signal recovering or degraded signal rebuilding in the electrical domain is required. It is

because mature optical repeaters are not available nowadays. Electrical signals are able to access the optical network through OE/EO modules and the optical interface as well.

3. Characteristic

Optical logic units in this EO computing system receive optical signals from adjacent units or/and the optical interface which collects signals from the other blocks. Results are delivered in two ways. One is forwarding the light to the next optical logic unit or the optical interface to participate in computing in the optical format. The other way is going through an OE module to terminate the optical path and get involved in the electrical calculation. Assuming that the inputs from various sources are $I_1, I_2 \cdots I_m$, and that the outputs are $S_1, S_2 \cdots S_n$, the control signals from electrical units are $e_1, e_2 \cdots e_l$, as shown in Fig. 2, then the function of the logic module can be expressed as

$$\begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_n \end{bmatrix} = [T]_{n\times m} \begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_m \end{bmatrix}, \tag{1}$$

where T is an operation matrix of the optical logic module, consisting of $n\times m$, elements which can be $e_1, e_2 \cdots e_l$ or constant values like 1 and 0 or their combinations. This express reveals several characteristics of the system which can also assist in the circuit design.

Firstly, the interface along the propagation direction is optical. In other words, the vector bases are all optical; electrical control signals only exist in the transfer matrix T. For example, when an EO modulator is controlled by an electrical signal, the output of the modulator turns out to be a beam of modulated light. Therefore, when designing a cascading optical system, we need to consider carefully what signals should be in the optical domain. Secondly, inside the OLU, the light will go all the way through to the output without any OE/EO conversion. Only in this way will the latency remain at a minimum without any extra conversion time. Another feature is that no product between two optical inputs is allowed. It is because, practically speaking, the optical multiplication between two light beams can only be realized by all-optical modulators which could bring a tremendous energy overhead based on current all optical devices [15,16].

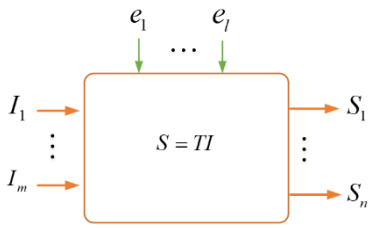

Fig. 2. The input and output interfaces of an OLU.

4. Building blocks

In order to be comparable to the CMOS transistors in terms of power and bandwidth, EO circuits have to obey the abovementioned characteristics due to the reality that mature building blocks in integrated photonics with high bandwidth and low power consumption are very limited. As far as we know, optical modulators, passive routing structures and photodetectors, are key components that can contribute to the optical computing system. Therefore, in this part, we will discuss various fundamental operations that can be realized by these structures within these characteristics.

Optics EXPRESS

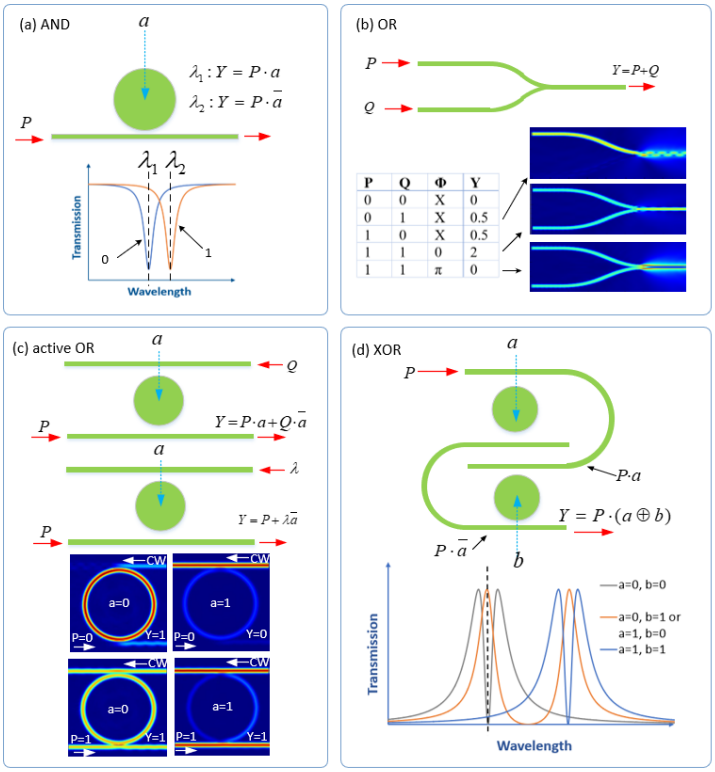

Fig. 3. The schematics of (a) AND gates, (b) OR gates, (c) active OR gates, and (d) XOR gates.

Table 1. Truth table

A	B	AND	OR	XOR
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

4.1 AND gate

All the logic operations can be expressed as a sum of products or a product of sums of the logic inputs. Therefore, AND and OR operations play a pivotal role in a logic system. The truth table is shown in Table 1. Fortunately, in the EO computing system, the modulation itself is a realization of AND intrinsically, which maps the electrical signals directly into optical ones. It should be noted that this AND operation can only be applied to the case with one optical operand and one electrical operand along with an optical output, denoted as $Y = P \cdot a$ or $Y = P \cdot \bar{a}$, as shown in Fig. 3(a). The applied voltage changes properties of the modulator such as the refractive index and finally switches the output states. The EO modulators include but are not limited to the microdisk resonators shown in the figure. For example, MZI modulators, EAM or other kinds of microresonator-based modulators such as nanobeam modulators [17] can be promising candidates as well. Among them, wavelength selective modulators could achieve the function and its negation simultaneously and independently at two separated wavelengths. For a microresonator, these different states are known as pass/block or block/pass modes. When cascading several modulators in series, one can easily obtain the AND operations between several electrical signals, written as

Optics EXPRESS

$Y = P \cdot a \cdot b \cdot c \cdots$. As mentioned above, due to the immaturity of low power all-optical modulators, the AND operations between two optical signals are not considered here.

4.2 OR gate

Passive optical components are capable of operating two or more optical signals such as OR without any electrical support, which means it is a zero-power-consumption process if not taking propagation and insertion loss into consideration. A simple combiner such as a Y branch or a multimode interference (MMI) coupler can be used as an OR gate to process two optical signals, as shown in Fig. 3(b). However, as can be seen in the truth table in Fig. 3(b), this type of OR gate is not ideal because of the uneven output intensity. For instance, when only one of the inputs is 1, the output turns out to be 0.5 with 3 dB loss due to the mode mismatch. When they are both one and in phase, the output will be a perfect sum of the two inputs without any loss. In the worst case, the output power will be totally lost when two one inputs are out of phase. If the intensity evenness really matters in a certain circumstance, especially in a cascading system, two potential solutions could be used to solve this problem. The first is to rewrite the logic function in order to cancel out the undesirable state. To be specific, after noticing that the output for $(P,Q) = (1,0)$ and $(P,Q) = (0,1)$ are identical, we try to strike out the complex logic state $(P,Q) = (1,1)$ by linking two inputs in logic. As an example, assuming the inputs are P and $\lambda \cdot a$, the function we need to achieve is $Y = P + \lambda \cdot a$. Obviously, unrelated random signals P and a will surely result in the unevenness of the outputs. In order to correlate them, we rewrite the function as

$$Y = P + \lambda \cdot a = P \cdot \bar{a} + \lambda \cdot a, \tag{2}$$

so that state (1,1) will not emerge for the two addends. Specifically, we can add another modulator before the combiner to eliminate the undesired state. The second way is to use an active OR gate consisting of a four-port modulator, which extricates itself from the 3 dB loss. As shown in Fig. 3(c), the electrical signal a will eventually choose one from the two inputs P and Q to propagate to the output with the function of $Y = P \cdot a + Q \cdot \bar{a}$. If setting the input Q to be λ, one could obtain $Y = P + \lambda \cdot a$ using Eq. (2). Therefore, it serves as a useful method to add an optical signal that may come from the previous stage with an electrical signal that needs to be applied to the optical link at this stage.

4.3 XOR gate

An XOR gate is also a fundamental gate in a logic system that produces a true output when the number of true inputs is odd. Since $a \oplus b = \bar{a} \cdot b + a \cdot \bar{b}$, it can be perfectly realized by the active OR gate mentioned in the previous section after the two inputs are set as $P \cdot a$ and $P \cdot \bar{a}$, which fortunately can be achieved by another active OR gate as well. As can be seen in the Fig. 3(d), the upper modulator produces $P \cdot a$ and $P \cdot \bar{a}$ at two ports, respectively. For the second modulator, the output would be $Y = (P \cdot a) \cdot \bar{b} + (P \cdot a) \cdot b = P \cdot (a \oplus b)$.

Optics EXPRESS

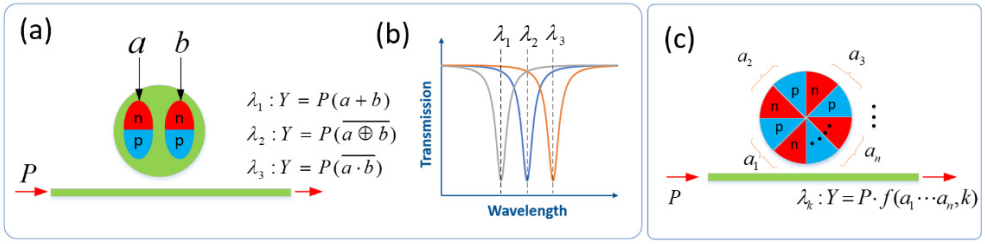

Fig. 4. A two-operand modulator (a) and the output functions at three different wavelengths (b). (c)A generalized case for multi-operand modulators.

4.4 Multi-operand modulators

Recent researches have come to the theoretical and experimental demonstration of two-segments EO modulators, utilizing two electrical driving signals to control the transmission characteristic in order to further increase the versatility and packing density of the fundamental unit, as shown in Fig. 4(a). As can be seen in Fig. 4(b), the transmission spectrum now will shift from λ_1 to λ_2 when one of the inputs is logic high and will shift more to λ_3 when both are high. The working wavelength determines the output function. At the working wavelengths of $\lambda_1, \lambda_2, \lambda_3$, the corresponding output function can be written as, $Y = P(a+b)$, $Y = P(\overline{a \oplus b})$, and $Y = P(\overline{a \cdot b})$, respectively. The negations, i.e. $Y = P(\overline{a+b})$, $Y = P(a \oplus b)$, and $Y = P(a \cdot b)$ can be generated at the drop port if dual-bus modulators are used. Obviously, compared to the electrical XOR/XNOR gates abovementioned, this approach substantially reduces the design complexity and footprint. We can also generalize it into a more generic case that a single modulator may be controlled by multiple electrical signals marked as $a_1, a_2, a_3 \cdots$ as shown in Fig. 4(c). At a certain working wavelength λ_k, the output turns out to be a counting function, which means when the number of input logic 1 equals $k-1$, the output will be pulled down to logic 0; otherwise, it will be pulled up to logic 1. Various logic circuits have been designed theoretically based on multi-operand modulators such as comparators, encoders, decoders, and adders [18].

4.5 Multiple wavelengths

Wavelength multiplexing wherein various wavelengths share the same channel contributes to the flexibility of designing the EO computing circuit. First, channel sharing leads to operation sharing. For instance, in Fig. 3(a), if two optical inputs, namely P with the wavelength of λ_1 and Q with the wavelength of λ_2, are injected into the modulator simultaneously, functions $Y_1 = P \cdot a$ and $Y_2 = Q \cdot \overline{a}$ will be generated successfully. Therefore, two related functions similar to Y_1 and Y_2 can be calculated in parallel. Second, channel sharing for a broadband photodetector means addition. For the same case above, if the output is terminated by a photodetector, the received electrical output will be $Y = P \cdot a + Q \cdot b$, which serves as a lossless OR gate. Some demonstrations on EO logic using wavelength multiplexing have been published [17,19].

5. Automated design algorithm

As the fundamental optical logic gate mentioned above matures, design automation starts to come onto the stage very naturally, similar to the CMOS auto-design. We summarize and categorize the abovementioned building blocks in Fig. 5, where only two-operand logic gates are listed just as a simple example.

Optics EXPRESS

In the figure, the first row shows the And-Inverter Graph (AIG) representation and the corresponding Boolean logic functions of operand a and b. The AIG is a directed acyclic graph, which is a widely-used data structure in traditional logic synthesis to represent Boolean functions [20]. The components of an AIG consist of two-input nodes (shown as circles) representing logical AND of the two inputs; terminal nodes labeled with primary input names; and edges optionally containing markers indicating logical NOT. The validity of AIG is based on the universality of the {AND, NOT} operations, in the sense that any logic can be represented as a combination of them. The second to fourth rows list the typical optical implementations of the corresponding AIGs and logic functions in the first row. The implementations in the three rows are categorized by the types of the inputs: both a and b are electrical inputs; a is electrical input and b is optical input (and vice versa); both a and b are optical. The modulators colored in green denote that the high/low voltage controls the off/on-resonance, and the modulators colored in yellow denotes the reverse, which means that the high/low voltage controls the on/off -resonance, as shown in Fig. 3(a).

Table 2. AIG-based synthesis algorithm

Algorithm 1 AIG-Based Synthesis
Input: Combinational Boolean function
Output: Optical implementation using EO logic gates
1: Convert the given function into an AIG
2: **for** each node in AIG **do**
3: Record all matching library gates
4: Pruning infeasible library gate choices
5: Select the best gates for each AIG node

Given a combinational Boolean function, the algorithm shown in Table 2 first converts the function to an AIG, which in our work, is generated from a state-of-the-art open-source logic synthesis tool ABC [21] (Line 1, Table 2). The resultant AIG is optimized by the embedded AIG-rewriting/balancing techniques to reduce the number of AIG node [20]. Then for each AIG node, the algorithm finds the library gates that implement the sub-AIG, so that the functionality rooted at this node is equivalent to the AIG representation of the library gate. To this end, traditional mapping algorithm can be adopted [22,23] (Line 3, Table 2). Basically, for each node n from the bottom to top of the given, optimized AIG, the mapping algorithm searches for the subtrees, where the node n is at their bottom, that matches any of the library gates. The search of this round stops if no such matching is possible. For example, at the AIG node with children c and d, as shown in Fig. 6(c), the search will find two smaller subtrees, $c\bar{d}$ and $c+\bar{d}$, and a bigger subtree of $c \oplus d$. All subtrees will be recorded as potential mappings. The subtree solutions of different rounds of search can overlap. Dynamic programming [23] is then used to determine the best mapping solution for the whole AIG.

Optics EXPRESS

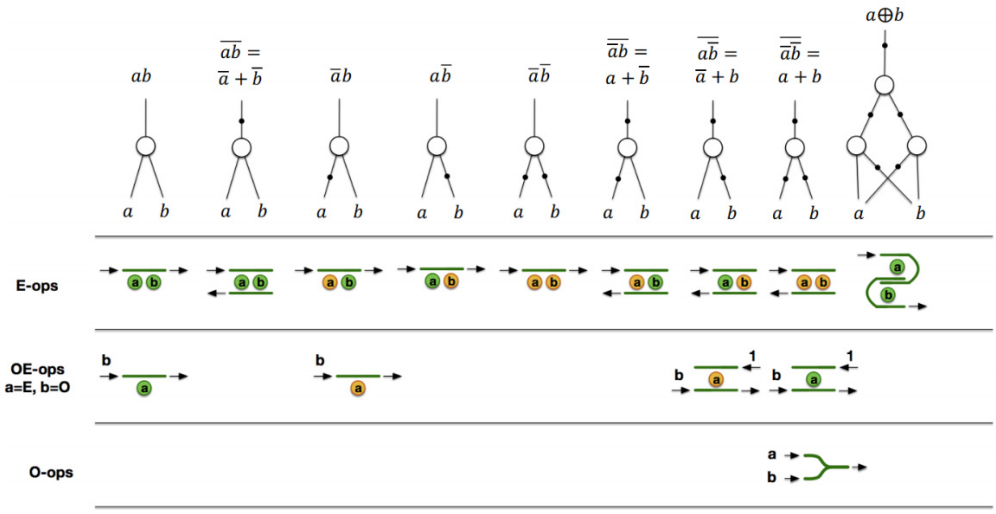

Fig. 5. A two-operand library of EO logic components.

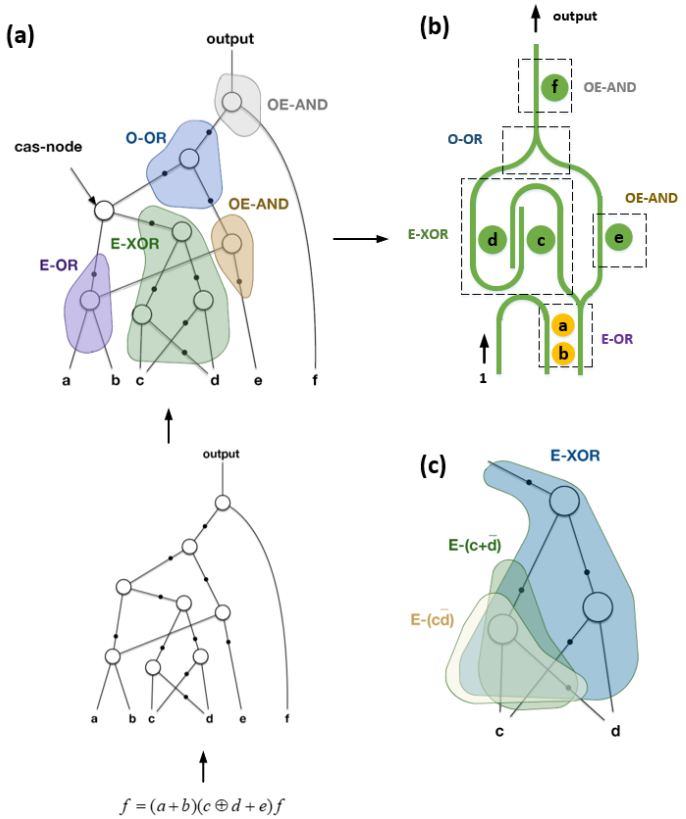

Fig. 6. (a) AIG and mapping of a random function. (b) Optical implementation based on EO logic. (c) An example of different mappings.

However, one critical difference of the mapping of optical gates and CMOS gates needs to be noted: the inputs have specific types (electrical or optical) and the mapping should handle them properly. Infeasible mapping, where some implementation cannot be realized due to the mismatch of input types (i.e. necessitate EO or OE conversions), can be created from the

Optics EXPRESS

previous step. These solutions will be pruned if cannot be handled by a special graph transformation as follows.

The transformation is performed by introducing of cascading nodes (CAS-node). Whenever an AND-node takes two optical inputs produced by two logic cones and thus cannot be mapped to any of the library implementations, it is nonetheless possible to reconnect one of the optical inputs of one of the logic cones to the constant light input of the other cone. Taking Fig. 6 as an example, the left and right logic cones are mapped by E-OR and E-XOR gates. Their outputs are both optical and are connected to an AND-node. Even though the AND-node cannot be realized by the library gates, we can reconnect the light output of the E-OR to the constant light input port of the E-XOR, which emulates the logical AND of light.

Finally, the feasible mapping will be selected based on the cost (e.g., the number of optical modulators) (Line 5, Table 2). It is worthwhile to notice that this algorithm does not guarantee the availability of feasible mapping. In the case when feasible mapping is not obtainable, the more general Binary Decision Diagram (BDD)-based algorithm can be leveraged [22]. Though BDD does guarantee the mapping feasibility, it usually entails a larger size compared to AIG-based mapping because AIG can make full use of its abundant library.

An example of the AIG-based algorithm is shown in Fig. 6. In Fig. 6(a), the sub-AIG mappings are circled and marked by the optical implementations. The cascading node is introduced to guarantee the feasibility of the mapping. Note that the optical phase at the O-OR gate needs to be well controlled in order to eliminate possible logic errors. Details have been discussed in 4.2. The final implementation is shown in Fig. 6(b), with six modulators in total in the whole design. As a comparison, if we use the reconfigurable optical directed-logic circuits proposed in [10], the number of the modulators will be around 64.

In order to show the universality of this algorithm, we present another example about a real application of a scalable four-bit full adder (the carry path) in Fig. 7. The function for each bit is $C_{out} = C_{in} \cdot (a \oplus b) + a \cdot b$. Figure 7(b) shows the optical implementation using the AIG-based algorithm. As we can see, the periodic pattern is well retained in the optical circuits. Again, for the directed logic circuits, it will need about $2^8 = 256$ modulators to implement the same function, leading to a much more complex circuit.

It should be noted that only two-operand library is shown here. It could be future enlarged to more-operand, multi-operand, and multi-wavelength libraries, which are believed to have the potential to further improve the performance of the EO logic computing circuits. In addition, AIG is general algorithm that can also involve all-optical non-linear components as long as we update the library accordingly. If we focus on the present library, the latency of the circuit is affected by several factors, such as the EO switching time. The switching time will not accumulate since all the gate could be set simultaneously. Normally, it will take tens of picoseconds or more for a resonator-based modulator to switch and only several picoseconds or even less for a non-resonator-based modulator, for example, an EAM. Another fact is the propagation time that will accumulate. For a non-resonator-based gate, it can be easily calculated according to the optical path and it is negligible in many cases. While for a resonator, it will be a little more complicated. If it is an 1x1 gate, such as the first gate in the library, the output will not be affected by the resonators since the light will directly pass through without going into the resonator at the off-resonance state, and light will be dropped at the on-resonance state. While for a 2x2 resonator, which is the basic component in directed logic, the latency accumulates when the light goes through the drop port [10]. Overall, the latency could be optimized to achieve desired performance based on the characteristics of the components including resonator-based and non-resonator-based as well as potential all-optical gates in the library.

Optics EXPRESS

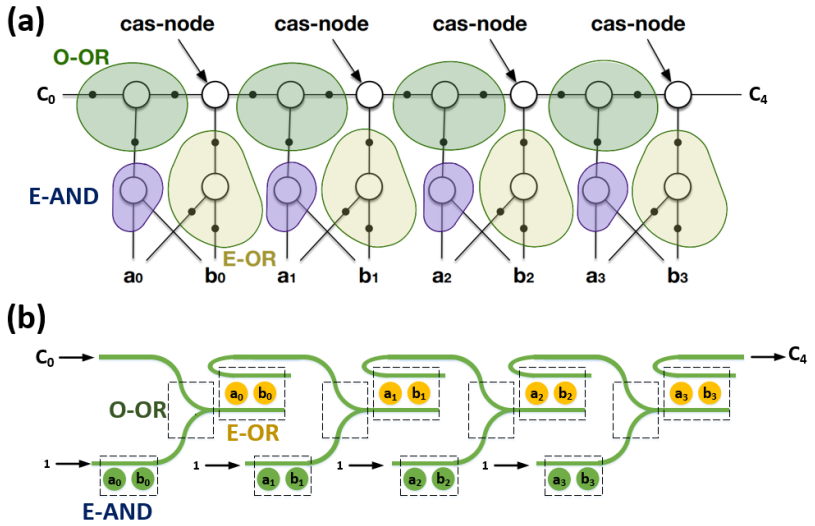

Fig. 7. (a) AIG and mapping of a carry path for a four-bit full adder. (b) Optical implementation based on EO logic.

6. Conclusion

We have proposed a new architecture of the integrated EO computing system based on EO logic, which makes the utmost use of electronics and photonics. Characteristics of optical logic units including continuity and independence are explored. A comprehensive library has been summarized based on the most recent high-speed EO components. Finally, an AIG-based automated logic synthesis algorithm is described with two real examples to implement the EO logic, which paves the way to future large-scale design automation of high-speed integrated optical computing circuits.

Funding

Air Force Office of Scientific Research (AFOSR) (FA 9550-17-1-0071, FA9550-17-1-0354).

Acknowledgment

The authors acknowledge support from the Multidisciplinary University Research Initiative (MURI) program through the Air Force Office of Scientific Research (AFOSR), monitored by Dr. Gernot S. Pomrenke.

References

1. J. Hardy and J. Shamir, "Optics inspired logic architecture," Opt. Express **15**(1), 150–165 (2007).
2. Z. Ying, S. Dhar, Z. Zhao, C. Feng, R. Mital, C.-J. Chung, D. Z. Pan, R. A. Soref, and R. T. Chen, "Electro-optic ripple-carry adder in integrated silicon photonics for optical computing," IEEE J. Sel. Top. Quantum Electron. **24**(6), 1–10 (2018).
3. Y. Tian, L. Zhang, R. Ji, L. Yang, and Q. Xu, "Demonstration of a directed optical encoder using microring-resonator-based optical switches," Opt. Lett. **36**(19), 3795–3797 (2011).
4. C. Qiu, X. Ye, R. Soref, L. Yang, and Q. Xu, "Demonstration of reconfigurable electro-optical logic with silicon photonic integrated circuits," Opt. Lett. **37**(19), 3942–3944 (2012).
5. L. Zhang, J. Ding, Y. Tian, R. Ji, L. Yang, H. Chen, P. Zhou, Y. Lu, W. Zhu, and R. Min, "Electro-optic directed logic circuit based on microring resonators for XOR/XNOR operations," Opt. Express **20**(11), 11605–11614 (2012).
6. Y. Tian, L. Zhang, Q. Xu, and L. Yang, "XOR/XNOR directed logic circuit based on coupled-resonator-induced transparency," Laser Photonics Rev. **7**(1), 109–113 (2013).
7. Z. Ying, Z. Wang, Z. Zhao, S. Dhar, D. Z. Pan, R. Soref, and R. T. Chen, "Silicon microdisk-based full adders for optical computing," Opt. Lett. **43**(5), 983–986 (2018).
8. Z. Zhao, Z. Wang, Z. Ying, S. Dhar, R. T. Chen, and D. Z. Pan, "Optical computing on silicon-on-insulator-based photonic Integrated circuits," in *IEEE International Conference on ASIC (ASICON)* (2017), pp. 472–475.

Optics EXPRESS

9. L. Zhang, R. Ji, L. Jia, L. Yang, P. Zhou, Y. Tian, P. Chen, Y. Lu, Z. Jiang, Y. Liu, Q. Fang, and M. Yu, "Demonstration of directed XOR/XNOR logic gates using two cascaded microring resonators," Opt. Lett. **35**(10), 1620–1622 (2010).
10. Q. Xu and R. Soref, "Reconfigurable optical directed-logic circuits using microresonator-based optical switches," Opt. Express **19**(6), 5244–5259 (2011).
11. G. T. Reed, G. Mashanovich, F. Y. Gardes, and D. J. Thomson, "Silicon optical modulators," Nat. Photonics **4**(8), 518–526 (2010).
12. J. Michel, J. Liu, and L. C. Kimerling, "High-performance Ge-on-Si photodetectors," Nat. Photonics **4**(8), 527–534 (2010).
13. H. Subbaraman, X. Xu, A. Hosseini, X. Zhang, Y. Zhang, D. Kwong, and R. T. Chen, "Recent advances in silicon-based passive and active optical interconnects," Opt. Express **23**(3), 2487–2510 (2015).
14. R. Duncan, "A survey of parallel computer architectures," Computer **23**(2), 5–16 (1990).
15. M. Xiong, L. Lei, Y. Ding, B. Huang, H. Ou, C. Peucheret, and X. Zhang, "All-optical 10 Gb/s AND logic gate in a silicon microring resonator," Opt. Express **21**(22), 25772–25779 (2013).
16. F. Li, T. D. Vo, C. Husko, M. Pelusi, D.-X. Xu, A. Densmore, R. Ma, S. Janz, B. J. Eggleton, and D. J. Moss, "All-optical XOR logic gate for 40Gb/s DPSK signals via FWM in a silicon nanowire," Opt. Express **19**(21), 20364–20371 (2011).
17. R. Soref and J. Hendrickson, "Proposed ultralow-energy dual photonic-crystal nanobeam devices for on-chip N x N switching, logic, and wavelength multiplexing," Opt. Express **23**(25), 32582–32596 (2015).
18. D. Gostimirovic and W. N. Ye, "Ultracompact CMOS-compatible optical logic using carrier depletion in microdisk resonators," Sci. Rep. **7**(1), 12603 (2017).
19. Y. Tian, Z. Liu, H. Xiao, G. Zhao, G. Liu, J. Yang, J. Ding, L. Zhang, and L. Yang, "Experimental demonstration of a reconfigurable electro-optic directed logic circuit using cascaded carrier-injection micro-ring resonators," Sci. Rep. **7**(1), 6410 (2017).
20. A. Mishchenko, S. Chatterjee, and R. Brayton, "DAG-aware AIG rewriting: a fresh look at combinational logic synthesis," in *2006 43rd ACM/IEEE Design Automation Conference* (2006), pp. 532–535.
21. A. Mishchenko, "ABC: a system for sequential synthesis and cerification," https://people.eecs.berkeley.edu/~alanmi/abc/abc.htm.
22. Z. Zhao, Z. Wang, Z. Ying, S. Dhar, R. T. Chen, and D. Z. Pan, "Logic synthesis for energy-efficient photonic integrated circuits," in *2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC)* (2018), pp. 355–360.
23. K. Keutzer, "DAGON: technology binding and local optimization by DAG matching," in *24th ACM/IEEE Design Automation Conference* (1987), pp. 341–347.

128.62.17.55 {ts '2018-11-26 16:06:30'}