Complete all-optical processing polarization-based binary logic gates and optical processors*

Y. A. Zaghloul and A. R. M. Zaghloul

Georgia Institute of Technology, School of Electrical and Computer Engineering, Ellipsometry Research and Applications Laboratories, Savannah, GA 31407
and
ITR Technologies Inc, POBox 86242, Lincoln, NE 68501-6242
yaz@ieee.org, arzaghloul@ieee.org

Abstract: We present a complete all-optical-processing polarization-based binary-logic system, by which any logic gate or processor can be implemented. Following the new polarization-based logic presented in [Opt. Express 14, 7253 (2006)], we develop a new parallel processing technique that allows for the creation of all-optical-processing gates that produce a unique output either logic 1 or 0 only once in a truth table, and those that do not. This representation allows for the implementation of simple unforced OR, AND, XOR, XNOR, inverter, and more importantly NAND and NOR gates that can be used independently to represent any Boolean expression or function. In addition, the concept of a generalized gate is presented which opens the door for reconfigurable optical processors and programmable optical logic gates. Furthermore, the new design is completely compatible with the old one presented in [Opt. Express 14, 7253 (2006)], and with current semiconductor based devices. The gates can be cascaded, where the information is always on the laser beam. The polarization of the beam, and not its intensity, carries the information. The new methodology allows for the creation of multiple-input-multiple-output processors that implement, by itself, any Boolean function, such as specialized or non-specialized microprocessors. Three all-optical architectures are presented: orthoparallel optical logic architecture for all known and unknown binary gates, single-branch architecture for only XOR and XNOR gates, and the railroad (RR) architecture for polarization optical processors (POP). All the control inputs are applied simultaneously leading to a single time lag which leads to a very-fast and glitch-immune POP. A simple and easy-to-follow step-by-step algorithm is provided for the POP, and design reduction methodologies are briefly discussed. The algorithm lends itself systematically to software programming and computer-assisted design. As examples, designs of all binary gates, multiple-input gates, and sequential and non-sequential Boolean expressions are presented and discussed. The operation of each design is simply understood by a bullet train traveling at the speed of light on a railroad system preconditioned by the crossover states predetermined by the control inputs. The presented designs allow for optical processing of the information eliminating the need to convert it, back and forth, to an electronic signal for processing purposes. All gates with a truth table, including for example Fredkin, Toffoli, testable reversible logic, and threshold logic gates, can be designed and implemented using the railroad architecture. That includes any future gates not known today. Those designs and the quantum gates are not discussed in this paper. * Patent Pending

OCIS codes: (200.0200) Optical computing; (200.3760) Logic-based optical processing; (200.4740) Optical processing; (200.4660) Optical logic.

#73748 - $15.00 USD Received 7 August 2006; revised 17 September 2006; accepted 18 September 2006

References and links

1. Y. A. Zaghloul and A. R. M. Zaghloul, "Unforced polarization-based optical implementation of binary logic," Opt. Express **14**, 7253 – 7269 (2006). Patent Pending.
2. M. M. Mano and C. R. Kime, *Logic and computer design fundamentals*, 2nd ed. (Prentice Hall, New Jersey, 2001).
3. D. S. Kliger, J. W. Lewis, and C. E. Randall, *Polarized light in optics and spectroscopy* (Academic, Boston, 1990).
4. R. M. A. Azzam, A. R. M. Zaghloul, and N. M. Bashara, "Ellipsometric function of a film-substrate system: Applications to the design of reflection-type optical devices and to ellipsometry," J. Opt. Soc. Am. **65**, 252 – 260 (1975). Also in A. R. M. Zaghloul, "Ellipsometric function of a film-substrate system: Applications to the design of reflection-type optical devices and to ellipsometry," Ph. D. dissertation, University of Nebraska-Lincoln, 1975.
5. A. R. M. Zaghloul, R. M. A. Azzam, and N. M. Bashara, "Design of film-substrate single-reflection retarders," J. Opt. Soc. Am. **65**, 1043 – 1049 (1975).
6. A. R. M. Zaghloul, R. M. A. Azzam, and N. M. Bashara, "An angle-of-incidence tunable, SiO_2-Si (film-substrate) reflection retarder for the UV mercury line λ = 2537 Å," Opt. Commun. **14**, 260 – 262 (1975).
7. A. R. M. Zaghloul, M. Elshazly-Zaghloul, W. A. Berzett, and D. A. Keeling, "Thin film coatings: A transmission ellipsometric function (TEF) approach I. Non-negative transmission systems, polarization-devices, coatings, and closed-form design formulae," Appl. Opt., In Press.
8. A. R. M. Zaghloul and M. Elshazly-Zaghloul, "Transmission polarization devices using an unsupported film/pellicle: Closed-form design formulae," SPIE Proceedings of the 2006 Defense and Security Symposium, Orlando, Florida, 17-21 April, 2006.
9. R. M. A. Azzam, "Simultaneous reflection and refraction of light without change of polarization by a single-layer-coated dielectric surface," Opt. Lett. **10**, 107 – 109 (1985).
10. A. R. M. Zaghloul, D. A. Keeling, W. A. Berzett, and J. S. Mason, "Design of reflection retarders by use of nonnegative film-substrate systems," J. Opt. Soc. Am. A **22**, 1637 – 1645 (2005).
11. M. A. Karim and A. A. S. Awwal, *Optical computing: An introduction* (Wiley, New York, 1992).
12. E. Fredkin and T. Toffoli, "Conservative logic," Int. J. Theor. Phys. **21**, 219 – 22 (1982).
13. D. P. Vasudevan, P. K. Lala, J. Di, and J. P. Parkerson, "Reversible-logic design with online testability," IEEE Trans. Instrum. Meas. **55**, 406 – 414 (2006).

1. Introduction

In a recent publication, a new optical-polarization-based representation and implementation of binary logic was introduced [1]. The system is based on the fact that logic one and logic zero both must exist and be quantitative. The proposed system, in one of its simplest implementations, employed the polarization of a beam at a general chosen relative polarization angle to be logic one while the logic zero is represented by another chosen fixed phase change relative to that of logic one. This representation allows the creation of optical gates that are much faster and more superior to their semiconductor counterparts. Only unforced gates which included inverters, XOR, and XNOR gates are possible to design and implement using that methodology.

In this communication, we present a novel method to parallel process optical beams to represent any gate and any Boolean function. The new system is based on splitting the beam into two channels each of which only allows (maximizes the passage of) the beam with certain prespecified polarization and blocks (minimizes the passage of) the beam with the other polarization. That allows for different processes to be independently applied to logic 1 and to logic zero beams with a single control module for each. The two different beams are then steered to the output point, and the system is designed such that only one is active at a time. Throughout, the information is polarization encoded into the beam, where the intensity of the beam itself carries no information. Therefore, the strength or weakness of the beam plays no role in the operation of the devices.

The new parallel processing method retains all of the advantages of the previously introduced polarization optical system [1]. Unlike semiconductors, the optical devices produce virtually no heat, and are bound only by the speed of light and speed of the optical control elements that are only applied once: a single time lag. The system is fully integrable

#73748 - $15.00 USD Received 7 August 2006; revised 17 September 2006; accepted 18 September 2006

with the previously introduced optical gates along with the current widely used electronically-based gates (semiconductor gates). The latter allows for the creation of hybrid systems or add-ons to systems currently in use. The optical output of the novel gates can be cascaded infinitely to implement complex Boolean functions using stand-alone gates. It can also be implemented into a single chip that contains various layers of optical elements. In such a case, the system representation is not only fast but all the components used in the representation are very cheap and are not complex to manufacture. Accordingly, advanced optical microprocessors can be easily and cheaply implemented.

In the following sections, we discuss the design and operation of all types of binary gates, including AND, NAND, OR, NOR, XOR, XNOR, and inverters. In addition, we discuss the concept of a general gate. That allows for all-optical processing of optical signals that applies the current cumulative knowledge of digital design. The same proposed design methodology can be used to design any future gates that are not known today which might lead to simplified digital design for all-optical processing. An easy to follow step-by-step algorithm is presented for the proposed design methodology. We also discuss the design and operation of polarization optical processors (POPs) that implement any complex Boolean expression, and present a step-by-step design algorithm to do the design. An electro-elimination concept is introduced and applied for that purpose. It eliminates all intermediate output-input signals of microprocessor designs to have an all-optical microprocessor. That allows for smart and/or application-specific all-optical processing of input data.

We discuss in detail the design and operation of the logic gates and POPs using the special case of two orthogonal polarization states representation: orthoparallel logic (OPL) design, which leads to the railroad (RR) architecture.

In addition to the standard binary gates, we design and discuss multiple-input gates, and sequential and non-sequential Boolean expressions. We also discuss the design reduction, to simplify the designs, and the operation of all. To clearly explain the operation of any RR-architecture POP, we use the simulation of a bullet train traveling at the speed of light over a railroad system preconditioned by crossovers pre-controlled by the control signals. It is important to note that the control signals are all applied simultaneously, eliminating the need for a timing diagram. Accordingly, no glitches can occur because of the non-existence of inherent propagation delay, and designs are glitch immune. Also, fan-in and fan-out problems encountered in digital system design are eliminated by the RR-architecture [2].

All three architectures are simple, direct, robust, very fast, glitch-immune, virtually consume no energy, require no heat dissipation, no masks, and no photon-electron conversions; not carrier-frequency limited and easily operational in a spectroscopic mode.

As the speed of operation of the introduced devices is bound by the speed of light, it is of unprecedented high values: related to the device dimensions. For example, if the device dimensions are in the millimeter range, the propagation delay would be in the order of pico seconds. On the other hand, if the device dimensions are in the nano range, the device would be much faster: with propagation delays in the order of femto seconds.

Experimental implementations, designs of Fredkin, Toffoli, testable reversible logic, and threshold logic gates, using the railroad architecture are reported elsewhere, to keep the size of the paper within reasonable limits and to focus on the novel concepts introduced. For the same reason, quantum gates are also not discussed. When the design algorithm of Section 5 is comprehended, it becomes evident that any future gates with truth tables that are not known today can be designed and implemented using the one and the same design algorithm.

In this paper, we do not discuss the engineering or packaging of the suggested gates and processors. We focus on the concepts and report on the engineering and packaging aspects of the devices elsewhere.

As discussed in Ref. [1] in some detail, several publications exist that report on the efforts of other research groups on the use of polarization to represent binary logic and gates. All used the two horizontal and vertical linear polarizations to represent the logic zero and logic one, some used spatial masks for beam manipulation, and all resulted in an on/off logic representation where the laser beam has to be regenerated for cascading purposes. In contrast,

#73748 - $15.00 USD Received 7 August 2006; revised 17 September 2006; accepted 18 September 2006

our current technology represents logic zero and logic one by any two orthogonal polarizations and manipulates the polarization of the light beam in accordance with the device performance. Therefore, the information is polarization encoded and the beam is polarization manipulated regardless of its intensity, which does not play any role in the device operation or performance.

2. System components

The components used in the orthoparallel logic OPL and railroad RR architectures design and implementation of digital gates and POPs, to be discussed in the following sections, are beam splitters (BSs), retarders (R), and polarizers (P). A BS is used to split the beam into two identical beams. The R is used to introduce a prespecified relative retardation angle without changing the relative amplitude of the output/input two components of the electric vector of the electromagnetic wave parallel and perpendicular to the plane of incidence or transmission, and the P is used to generate a linearly polarized beam of a prespecified polarization angle of inclination with the plane of incidence, or plane of transmission. The three components are available using the standard crystal-type [3]. They are also available using inexpensive film-substrate systems in reflection or transmission modes [4-8]. In Ref. [1], the Rs and Ps are discussed in detail along with their respective representation in both the complex ρ and τ planes. The BSs are discussed in some detail in Refs. [9, 10].

A spatial mask, properly designed, maybe used to replace the stationary combinations such as BS/P/M/P and BS/P/M/P/R [11]. Polymer spatial masks are easily manufactured today. Also, a birefringent polarizer, linear, circular, or elliptical, can replace the BS/P/M/P combination bringing the efficiency of the device to the high upper nineties.

Electro-optic devices such as magneto-optic, electro-optic, or liquid crystals can be used to effect polarization rotation, instead of using a retarder. The polarization control of the beam and not its intensity are the key in the current technology. Regardless of the beam intensity, the information continues to be present in its polarization state.

A polarization-preserving device (PPD) might be used to keep the polarization state of the beam unchanged, if and when needed for steering purposes [10]**.** Also, mirrors can be used for that purpose.

It is important to emphasize that the three devices of R, P, and BS are becoming, and will continue to become, commercially available in increasingly diverse forms and are not in any way to be limited to the discussed principles of operation now or in the future. The current industry standard processes are sufficient to produce high speed very large scale integrated chips of the proposed architectures.

3. Orthoparallel optical logic representation

Defining the complex ρ plane as the complex plane of the ρ vector representing the relative output/input phase difference Δ and amplitude $\tan \psi$ of the two components of the electric vector of the electromagnetic wave (laser beam) parallel and perpendicular to the plane of incidence. (The τ plane has a similar definition for transmission.) This complex ρ plane is used extensively to illustrate how different designs can be achieved using the new binary logic representation in conjunction with the new parallel processing methodology: orthoparallel logic OPL and railroad RR architectures.

Previously, a general gate was composed of a series of devices as in Ref. [1]. On the other hand, in the current paper a general gate is composed of similar devices but in parallel. Accordingly, two different optical-polarization operations (OPOs) can be performed on each branch after crossing out the polarization representing the other branch: a total of four OPOs.

For simplicity, we select the logic one L1 to be a linearly polarized light at +45° and logic zero L0 to be a linearly polarized light at -45° at both the input and output of any gate or chip. Naturally, an inverter is a simple retarder that introduces 180° phase shift in the ρ plane. Now, for each gate the desired output for each input must be determined. Since a 2-input gate of this design includes a unique output to only one of 4 input possibilities of the AND, NAND, OR, and NOR gates, then one of the two branches does not require logic-controlled operations. On

#73748 - $15.00 USD Received 7 August 2006; revised 17 September 2006; accepted 18 September 2006

the other hand, a 2-input gate that results in a unique output logic (L1 or L0) to two of the four input logics requires one operation in each branch: XOR and XNOR gates. In the following subsections, a brief discussion of the representation, design, and operation of the gates AND, NAND, OR, NOR, XOR, and XNOR is presented. The discussion is concluded with a generalized gate, n-input gate, and a short account on future gates.

3.1 AND gate

Table 1. AND gate 2. NAND gate 3. OR gate 4. NOR gate 5. XOR gate 6. XNOR gate

A	B	Z	A	B	Z	A	B	Z	A	B	Z	A	B	Z	A	B	Z
0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1
0	1	0	0	1	1	0	1	1	0	1	0	0	1	1	0	1	0
1	0	0	1	0	1	1	0	1	1	0	0	1	0	1	1	0	0
1	1	1	1	1	0	1	1	1	1	1	0	1	1	0	1	1	1

An AND gate has the truth table shown in Table 1. The first input column is considered as the optical input (laser beam) and the second input column as the control input. As mentioned earlier, an AND gate requires only one operation at one of the two branches, since it outputs L1 at only one of four input possibilities.

First, the optical input is generated, outside of the gate, by a simple P that either produces a linearly polarized light at 45° or -45° to represent logic one (L1) and logic zero (L0), respectively, or by polarization rotation using, for example, a liquid crystal. Coming into the gate, the input is then split into two beams, inside the gate, using a BS, Fig. 1. One branch includes a P at 45° angle, hence only passes light polarized at 45°, and the other includes a P at -45° angle, hence only passes light polarized at -45°. A properly designed mask or a birefringent polarizer can be used to create the two branches. That results in a branch containing L0, logic zero branch (LZB), and a branch containing L1, logic one branch (LOB), with only one branch at a time having a light beam. In the LZB, no manipulation is necessary. In the LOB, a control R is designed to introduce zero angle change in the complex ρ plane (zero relative phase shift) if the second input is L1 or introduce a 180° angle change if the second input is L0. Or a liquid crystal can be used as discussed in the previous section.

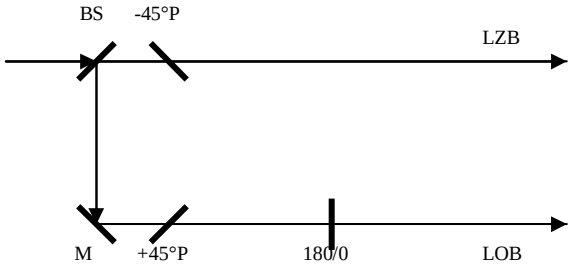

Fig. 1. Orthoparallel Logic OPL architecture of an AND gate., where BS: beam splitter, P: polarizer, 180/0: retarder producing 180°/0 phase shift in the complex ρ plane, LZB: logic zero branch, and LOB: logic one branch. The BS/-45° P/M/+45° P device combination can be replaced by a single properly designed spatial mask or by an appropriate birefringent polarizer. The 180°/0 retarder can be replaced by a 90° polarization rotation electro-optic device such as a liquid crystal. Only one beam leaves the gate by simple steering.

The two beams are then steered to the same output point. Note that the output only includes one of the two beams at a time. No interference occurs. Remember that linearly polarized light and a P at 45° are represented in the complex ρ plane by the point (+1, 0) and that linearly polarized light and a P at -45° are represented by the point (-1, 0): orthogonal polarization states and devices [1].

#73748 - $15.00 USD Received 7 August 2006; revised 17 September 2006; accepted 18 September 2006

Therefore, the operation of the gate is very simple. When the input is L1, only the LOB has a signal and the output is L1 if the second control input is L1, and the branch device is a PPD or nothing at all. If the second control input is L0, the output beam is polarization rotated 180° by a simple R device, and the output is L0. On the other hand, if the input is L0, only the LZB has a signal and the output is L0 in both cases of the second control input, and the branch device does not exist, or a PPD can be used.

3.2 NAND gate

While it is very easy to add a non-control inactive R that introduces a 180° angle in the ρ plane to the output of an AND gate to produce a NAND gate, the gate can otherwise be independently designed using knowledge of the complex ρ plane and the OPL architecture. The truth table of a NAND gate is shown in Table 2, and the OPL gate architecture is shown in Fig. 2. (Note that the NAND gate contains one more optical component than the AND gate: the LZB component.) As always, the first input column is considered as the optical input (laser beam) and the second input column as the control input. Similar to the last design, we create two branches one containing L0, LZB, and one containing L1, LOB. The LZB requires a simple inactive (unchanged) 180° R. The LOB requires a control R introducing a zero degree rotation at the second control input of L0, and a 180° rotation at the second control input of L1: both rotations are in the complex ρ plane. Note that the output contains one less operation than when adding an inverter to the output of an AND gate, while they contain the same optical components: where the 180° is within the gate instead of outside as an inverter.

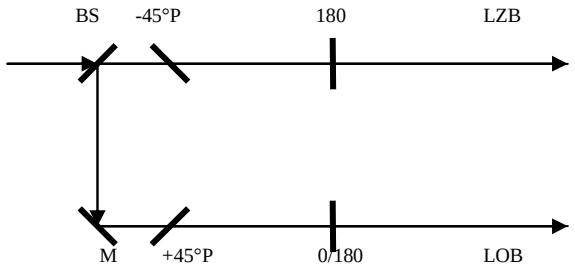

Fig. 2. Same as in Fig. 1, but for a NAND gate.

The operation of the gate is also straight forward. For an L0 optical input, only LZB contains an optical signal and the output beam is polarization rotated 180° for both of the control signals of L0 and L1: an inactive R. On the other hand, for an L1 optical input, only LOB contains an optical signal and the output beam is either unchanged, L1, or 180° polarization changed by the second control input, L0, see Fig. 2. It is clear that the inactive R component is removed in the AND gate and that the second control input is reversed, as expected.

3.3 OR gate

An OR gate can be constructed using the previous gates and inverters, or using only NAND gates, or it can be designed using the knowledge of the complex ρ plane and the OPL architecture. The truth table of an OR gate is shown in Table 3, and the OPL gate architecture is shown in Fig. 3. First, as always, the first input column is considered as the optical input (laser beam) and the second input column as the control input. Similar to the previous designs, we create two branches one containing L0 and one containing L1. The LOB requires no further control. On the other hand, the LZB needs a simple R that introduces a 180° relative phase shift at the second control input of L1, and a zero relative phase shift at the second control input of L0. As always, the two beams are then steered, as needed, to the output point. Only one beam is active at a time.

#73748 - $15.00 USD Received 7 August 2006; revised 17 September 2006; accepted 18 September 2006

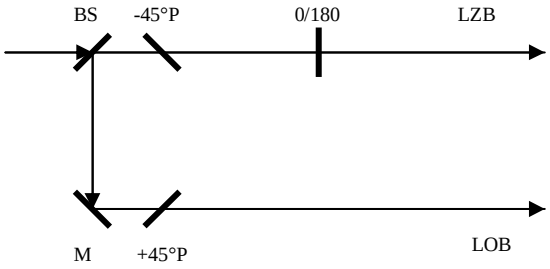

Fig. 3. Same as in Fig. 1, but for an OR gate.

The operation of the gate can be easily followed using Fig. 3, along with Table 3. For an optical input of L1, the LOB is the only active branch and the output is L1 with no beam manipulation required: no optical component or a PPD if needed. On the other hand, for an optical input of L0, the output is either L0 if the second control input is L0 requiring a zero R (a PPD or no device), or L1 if the second control input is L1 requiring a 180° R. As always, the required optical component has two states of operation zero and 180° depending on the second control input.

In comparing the AND and OR OPL gate architectures, it is clear that they have the same optical components with one rearrangement: the control R is moved from the LOB to the LZB with its two polarization states flipped.

3.4 NOR gate

A NOR gate can be constructed using the previous gate and an inverter, using only NAND gates, or it can be designed using knowledge of the ρ plane and the OPL architecture. The truth table and OPL architecture of a NOR gate are given in Table 4 and Fig. 4, respectively. First, as always, the first input column is considered as the optical input (laser beam) and the second input column as the control input. Similar to the previous designs, we create two branches one containing L0 and one containing L1. The LOB requires an inactive R that introduces a 180° phase change in the ρ plane. The LZB requires a control R introducing a 180° phase change at the second control input of L0 and zero phase change at the second control input of L1.

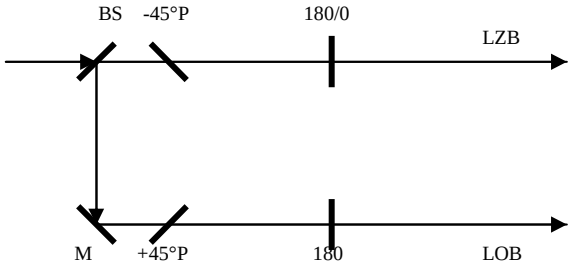

Fig. 4. Same as in Fig. 1, but for a NOR gate.

For this gate, and for an optical input of L1, the output is always L0 for both states of the second control achieved by the inactive 180° R. But for the L0 optical input, the output is L1 for the second control state of L0 (achieved by a 180° phase change of the control R) and is L0 for the second control state of L1 (achieved by a zero phase change of the control R.)

Note that the NOR gate contains one more optical component than the OR gate, which is the same as when an inverter is used with the OR gate. In this case, the R is inside the gate and not of a separate inverter gate, as compared to the OR gate cascaded with an inverter. Also, the output contains one less operations.

#73748 - $15.00 USD Received 7 August 2006; revised 17 September 2006; accepted 18 September 2006

Note also that the NAND and NOR gates are having the same number of optical components, with the optical components interchanged branches, and the two states of the active R also interchanged.

3.5 XOR and XNOR gates

The XOR and XNOR unforced simple realizations were introduced in Ref. [1]. The two gates can also be easily implemented by using the above gates, or by using only NAND or NOR gates. What differentiates between the XOR/XNOR gates and the gates above is the fact that the XOR and XNOR gates require control operations that are symmetric in the LZB and LOB. That fact allows for the creation of the XOR and XNOR logic gates without branching and parallel operations, and by only having one controlled operation on any optical input (L1 or L0). That explained, it is obvious that the XOR and XNOR gates can be implemented using the OPL architecture, by simply having the same controlled retarder on both branches. While usually that is not necessary, there are cretin cases that can benefit from such representation such as the all optical non-single-dimension implementation of complicated Boolean functions that will be discussed later.

A. Parallel architecture

A.1 XOR gate:
An XOR gate can be constructed using the AND or OR gates along with inverters, using only NAND or only NOR gates, or it can be designed using knowledge of the complex ρ plane and the OPL architecture, as all other gates. The truth table of an XOR gate and its OPL architecture are shown in Table 5 and Fig. 5, respectively.

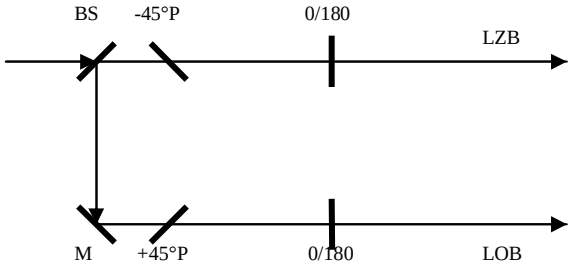

Fig. 5. Same as in Fig. 1, but for an XOR gate.

First, as always, the first input column is considered as the optical input (laser beam) and the second input column as the control input. Similar to the previous designs, we create two branches one containing L0 and one containing L1. From the truth table, it is clear that both branches perform the same operations using similar components. In the case where the optical input is linearly polarized at +45° or -45°, each branch requires a simple R that introduces a 180° phase change at the second control input of L1, and zero phase change at the second control input of L0.

As can be seen from Table 5, and Fig. 5, for an L0 optical input, the output is L0 for a second control input of L0 (achieved by a zero phase change R, or no optical component at all), and is L1 for a second control input of L1 (achieved by a 180° phase-change R). On the other hand, for an optical input of L1, the output is L1 for a second control input of L0 (also achieved by a zero phase change R, or no optical component at all), and is L0 for a second control input of L1 (achieved by a 180° phase-change R.) The parallelism is obvious and can be used to simplify the design to single-branch architecture, as we discuss in Sec. 3. 5. B.

A.2 XNOR gate:
As the XOR gate, the XNOR gate can be constructed using the AND or OR gates along with inverters, using only NAND or only NOR gates, or it can be designed using knowledge of the

#73748 - $15.00 USD Received 7 August 2006; revised 17 September 2006; accepted 18 September 2006

complex ρ plane and the OPL architecture, as all other gates. The truth table of an XNOR gate and its OPL architecture are shown in Table 6 and Fig. 6, respectively.

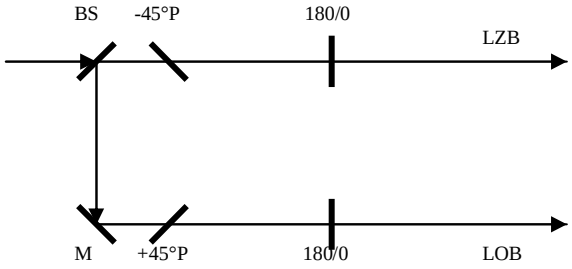

Fig. 6. Same as in fig. 1, but for an XNOR gate.

First, as always, the first input column is considered as the optical input (laser beam) and the second input column as the control input. Similar to the previous designs, we create two branches one containing L0 and one containing L1. From the truth table, it is clear that both branches perform the same operations using similar components. In the case where the optical input is linearly polarized at +45° or -45°, each branch requires a simple R that introduces a 180° phase change at the second control input of L0, and zero phase change at the second control input of L1.

The operation of the gate is easily followed using Table 6 and Fig. 6. For an optical input of L0, the output is L1 for the second control input of L0 (achieved by a 180° phase change R), and is L0 for the second control input of L1 (achieved by a zero phase shift R, or no component at all). On the other hand, for an L1 optical input, the output is L0 for a second control input of L0 (achieved by a 180° phase shift R), and is L1 for the second control input of L1 (achieved by a zero phase shift R, or no component at all.) The parallelism is clearly apparent.

Note that the XOR and XNOR of the OPL architecture type are having the same components in both LZB and LOB, with the same two states of the R interchanged.

B. Single-branch architecture:

As we discussed above, the identical parallel branches of the OPL architecture design of the XOR and XNOR gates lead to a greatly simplified design, where no branching is required and accordingly only one component is used: single-branch architecture.

B.1 XOR gate:

Figure 7 shows the single-branch (SB) architecture of an XOR gate. When the optical input is L0, the output is L0 for a second control input of L0 (achieved by a zero phase change R, or by no component present), and is L1 for a second control input of L1 (achieved by a 180° phase-shift R.) When the optical input is L1, the output is L1 for a second control input of L0 (achieved by a zero phase change R, or by no component present), and is L0 for a second control input of L1 (achieved by a 180° phase-shift R.)

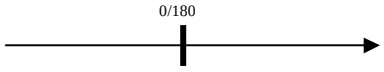

Fig. 7. Single-branch architecture of an XOR gate.

B.2 XNOR gate:

Figure 8 shows the SB architecture of an XNOR gate. When the optical input is L0, the output is L1 for a second control input of L0 (achieved by a 180° phase-shift R), and is L0 for a second control input of L1 (achieved by a zero phase change R, or by no component present.) When the optical input is L1, the output is L0 for a second control input of L0 (achieved by a

#73748 - $15.00 USD Received 7 August 2006; revised 17 September 2006; accepted 18 September 2006

180° phase-shift R.), and is L1 for a second control input of L1 (achieved by a zero phase change R, or by no component present.)

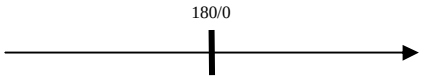

Fig. 8. Single-branch architecture of an XNOR gate.

3.6 Generalized gate

To generalize, we consider a general gate with any truth table designed using the orthoparallel architecture concept, where the laser beam carries the information into and out of the gate, and where L1 and L0 are represented by the +45° and -45° linearly polarized light: points (+1, 0) and (-1, 0) of the complex ρ plane. That special case of linearly polarized light simplifies the design and reduces the cost drastically.

As the laser beam enters the gate it is split into two beams of the same polarization properties using a BS. In the LZB, a -45° P is present to only pass -45° linearly polarized light L0. In the LOB, a +45° P is present to only pass +45° linearly polarized light L1. Alternatively, we can use a properly designed spatial mask, or a proper birefringent polarizer, to replace the three optical components and mirror, as discussed in Sec. 2.

Table 7. Collective table of the control retarder (R) for all gates, Figs. 1 – 6. LZB is the logic zero branch, top branch, LOB is the logic one branch, lower branch, and L0/L1 are the two states of the control R corresponding to the logic zero L0 and logic one L1 states.

	AND	NAND	OR	NOR	XOR	XNOR
LZB (L0/L1)		180	0/180	180/0	0/180	180/0
LOB (L0/L1)	180/0	0/180		180	0/180	180/0

In each branch, an R which acts as a polarization rotator introduces two optical rotations corresponding to the two states of the second control signal. Therefore, we have 4 degrees of freedom to design any gate. One special case is zero and 180° phase-shift retarder. As discussed in Sec. 2, a liquid crystal can be used instead of the R.

Table 7 summarizes the design and operation information of the ±45° OPL special case architecture. The changes from any gate design and operation to its NOT complement is evident in the table. Also, the corresponding changes from an AND gate to a NOR gate is evident. In addition, the parallelism of the XOR and XNOR gates is clear.

The generalized gate opens the door to reconfigurable microprocessors and to programmable optical logic gates.

3.7 N-input gates

The n-input gates are treated as a gate that represents a specific selected Boolean expression. That is treated as a simple special case of a Polarization Optical Processor (POP) in Sec. 5, and a 3-input AND gate is discussed as an example.

3.8 Future gates

The OPL architecture can be used to design any logic gate of any type, with any truth table, and for any purpose. For example, when the Fredkin and the Toffoli gates were introduced, they were novel types of gates at the time [12]. They can be designed using the POP of Sec. 5. In addition, new types of gates are being introduced until very recently, a trend that is expected to continue in the future. For example, three types of reversible gates with a built-in testability are presented in Ref. [13]. Also, those gates can be designed and realized using the POP. A separate publication is devoted to this subject to limit the size of the current paper.

#73748 - $15.00 USD Received 7 August 2006; revised 17 September 2006; accepted 18 September 2006

4. General orthoparallel optical logic (GOPL) realization

As discussed in detail in Ref. [1], for any general state of polarization of the light beam represented by the ρ vector, $|\rho| \angle \theta°$, there exists an orthogonal polarization state represented by a vector of magnitude $1/|\rho|$ with a 180° phase shift. Therefore, those two orthogonal polarizations can be used in the OPL architecture discussed above. Figure 9 shows the architecture of a general OPL (GOPL) AND gate. As before: 1) the input laser beam carries the optical information to the gate, L1 or L0; 2) the LZB has an elliptical polarizer to eliminate the L1 beam; 3) the LOB has an elliptical polarizer to eliminate the L0 beam; 4) the LZB has no control component and L0 beam leaves the gate as is; 5) the LOB has a controlled elliptical polarizer to introduce $1/|\rho|^2 \angle 180°$ for L0 and introduce no polarization change (PPD) for L1.

All gates can be designed and their operation analyzed using the GOPL, similar to the AND gate just discussed. Also, a corresponding table can easily be derived, similar to Table 7.

An interesting special case of $\theta = 0$, where both polarizations are on the real axis of the complex ρ plane, works with only orthogonal linear polarizations. The limiting case is where one polarization is perpendicular to the plane of incidence, or transmission, and the other is parallel, represented by the origin and the point at infinity on the complex ρ plane, respectively. Also of interest is the case of linearly polarized light at ±45°, which is discussed in detail in Sec. 3.

Another interesting case is where the two orthogonal polarizations lie on the unit circle at any general angle of θ and θ+ 180°, with the two special cases of linearly polarized light of ±45° and of circularly polarized light of ±90°.

For most cases, if L0 and L1 of the optical carrier are required to be of any specified polarization state, an R can be used at the output.

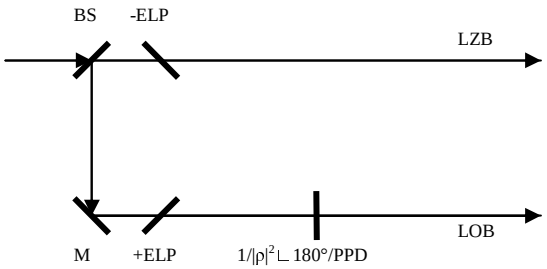

Fig. 9. General OPL architecture of an AND gate.

5. Polarization optical processor (POP), electro-elimination, and railroad architecture

Any Boolean expression can be designed and implemented using the all-optical processing polarization logic gates discussed in the previous sections along with the well established design rules of digital logic. Also, polarization optical processors (POPs) that represent any sets of Boolean expressions, instructions, as a single device can be designed and implemented. In the following subsections, we present an algorithm to do the design, then apply it to three cases. First, a three-input AND gate is discussed as a simple example. Second, a sequential Boolean expression where only one optical output of a gate is the input to another is presented. Third, a non-sequential expression where two optical gate outputs are inputs to another is last presented. Accordingly, the concept of electro-elimination is introduced, where the Boolean expressions are designed as a single entity. It leads to the railroad (RR) architecture. We also discuss some principles of design reduction to reduce the number of optical elements and branches employed in any design.

5.1 Design algorithm

To systematically design any Boolean expression, we present a simple step-by-step algorithm. It can be easily applied to any design problem. It is also easily programmed into a software

#73748 - $15.00 USD Received 7 August 2006; revised 17 September 2006; accepted 18 September 2006

design program to run on any general purpose PC, which is a project currently underway and will be reported elsewhere.

1. Start with the truth table and split the optical input column; first column A of the truth table.
2. Work with the top half of the truth table: the 0s.
3. Split the 0s and 1s of the second column B.
4. Check the output column for translation. (A translation is defined as moving every element a fixed distance in the same direction.)
5. If translation exists, then column B has no effect.
6. If translation does not exist, then invert optical-carrier polarization @ B = 1 and add a new branch. Do not add a new branch for the last input.
7. Repeat steps 3 – 6 for all other input columns in sequence.
8. Check the last optical carrier column to be identical to the output column: use an R if needed.
9. Repeat steps 3 – 8 for the bottom half of the truth table: the 1s.

5.2 Three-input AND gate

The three-input AND gate is a simple case to apply the design algorithm. We start with the truth table, given in Table 8. We generate a new truth table containing the state of the optical carrier, optical beam, as it travels through the POP, by adding a new column after each input column: optical-carrier column a after input column A, optical-carrier column b after input column B, and so on, Table 9. The purpose of introducing the optical-carrier column is to make it easy to identify the polarization state of the laser beam at any point within the system. Now we apply the design algorithm; always refer to Tables 8 and 9, and to Fig. 10.

Table 8. Truth table of a three-input AND gate.

A	B	C	Z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Table 9. Extended truth table for the three-input AND gate.

A	a	B	b	C	c	Z
0	0	0	0	0	0	0
				……...		
0	0	0	0	1	0	0

0	0	1	0	0	0	0
				……...		
0	0	1	0	1	0	0
1	1	0	1	0	1	0
				……		
1	1	0	1	1	1	0

1	1	1	0	0	0	0
				……...		
1	1	1	0	1	1	1

A. Design

1. A. Split the input column A: horizontal continuous line of Table 9 and BS1 and two branches containing two polarizers LZB and LOB of Fig. 10.
   B. Fill out optical-carrier column a: identical to input column A.
2. Focus on the top half of the truth table: the 0s.
3. Split the input column B: horizontal dashed line of upper half of Table 9.
4. Check the output column for translation: it exists.
5. A. Then input column B has no effect.
   B. Accordingly, the carrier beam travels unchanged.

#73748 - $15.00 USD Received 7 August 2006; revised 17 September 2006; accepted 18 September 2006

C. fill out the upper half of the optical-carrier column b; identical to optical-carrier column a.

6. Does not apply.
7. For input column C: It is divided into two quarters (Qs), by the dashed line that divided column B. Each quarter is dealt with separately.
   Q1:
   3. Split Q1; horizontal dotted line of upper half of Table 9.
   4. Check the output column for translation: it exists.
   5. A. Then input column C has no effect.
      B. Accordingly, the carrier beam travels unchanged. (We will continue to use carrier instead of optical-carrier for simplicity.)
      C. Fill out the upper Q, Q1, of the carrier column c: identical to carrier column b.
8. Last carrier checks: carrier column c and output column are identical. Therefore, an R is not needed.
   Q2: Repeat as for Q1. The result is identical.
9. Repeat 3 - 8 for lower half of Table 9: the 1s.

Now we start at Step 3 applied to the lower half.

3. Split the input column B: horizontal dashed line of lower half of Table 9.
4. Check the output column for translation: it does not exist.
5. Does not apply.
6. A. Then invert carrier polarization @ B = 1 and add a new branch.
   B. Accordingly, fill out the lower half of the carrier column b: inverting carrier column a states when B = 1, Table 9.
   C. Add a new branch, Fig. 10.
7. Now, for input column C we have two branches LZB and LOB. First, we work with the LZB: 0s of the carrier b Q4 (last two states of carrier column b).
   3. Split the column: horizontal dotted line separating last two states of input column C, Q4.
   4. Check the output column for translation: it does not exist.
   5. Does not apply.
   6. A. Then invert carrier polarization @ C = 1 and add a new branch.
      B. Accordingly, fill out the last two states of carrier column c: inverting carrier column b state when C = 1, Table 9.
      C. A new branch is not needed because this is the last input.
8. Last carrier checks: carrier column c and output column are identical. Therefore, an R is not needed.

Second, we work with the LOB; the 1s of the carrier b Q3 (last two L1 states of carrier column b).

3. Split the column: horizontal dotted line separating the two-before-last states of input column C, Q3.
4. Check the output column for translation: it exists.
5. A. Then input column C has no effect.
   B. Accordingly, the carrier beam travels unchanged.
   C. Fill out the rest of the carrier column c: identical to carrier column b.
8. Last carrier does not check, then add an R, Fig. 10.

#73748 - $15.00 USD Received 7 August 2006; revised 17 September 2006; accepted 18 September 2006

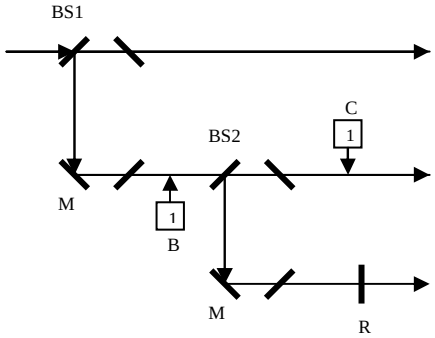

Fig. 10. Railroad-architecture polarization optical processor, RR-architecture POP, design of a three-input AND gate.

B. Design reduction

As it is clear from Fig. 10, the R in the lower branch can be combined with the B carrier-polarization-inversion box inverting @ 1 (CPIB1) before BS2, actually it is a controlled R, and both replaced by a B-CPIB0, with an added C-CPIB0. That requires removing the C-CPIB1, Fig. 11. That design reduction can be systematically achieved by always using a CPIB1 in the upper half of the carrier column A, the 0s, and a CPIB0 in the lower half, the 1s. Table 10 shows the lower half of the extended truth table for that case.

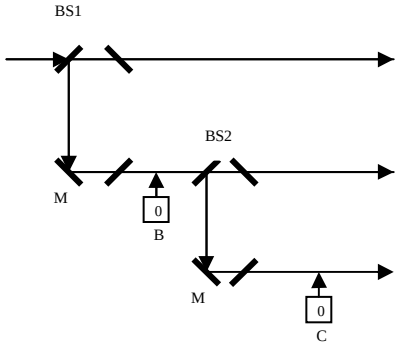

Fig. 11. Reduced design of Fig. 10.

Table 10. The lower half of the extended truth table of the three-input AND gate, inverting @ B = 0.

A	a	B	b	C	c	Z
1	1	0	0	0	0	0
				. .……		
1	1	0	0	1	0	0

1	1	1	1	0	0	0
				……		
1	1	1	1	1	1	1

C. Operation

The operation of the 3-input AND gate design of Fig. 11 is easily understood through following the laser beam as it travels through the gate. First of all, it is important to realize the fact that all control-input signals are applied simultaneously, leading to only a single time lag: time taken to apply only one signal.

When a carrier signal of L0 is applied to the gate, it is split into two by BS1. The lower signal into LOB is blocked by the polarizer, and the upper signal into LZB passes through the polarizer into the output unchanged, L0. Note that B and C signals are not applied to that branch. That case represents the first 4 rows of the truth table.

On the other hand, when a carrier signal of L1 is applied to the gate, it is split into two by BS1. The upper signal into LZB is blocked by the polarizer and the lower signal into LOB passes through the polarizer, unaffected by CPIB0 for a B input of L1 and then split into two by BS2. As an L1 carrier signal, it is blocked by the polarizer of the upper branch (LZB) and

#73748 - $15.00 USD Received 7 August 2006; revised 17 September 2006; accepted 18 September 2006

passes through the polarizer in the LOB, lower branch. For a C signal of L1, the carrier passes unchanged, last row of Table 10; and for a C signal of L0, the carrier state changes to L0, one-before-last row of Table 10.

All other combinations of input signals can be understood the same way.

D. Railroad architecture

The process of changing the carrier path according to the control signal, if and where needed, is similar to the railroad (RR) crossover, hence the railroad architecture. The carrier in this case is similar to a bullet train traveling at the speed of light on an RR system that is preconditioned by crossover actions of the control signals to determine its destination: the output polarization state.

The RR-architecture lends itself easily to software control leading to RR- reconfigurable architecture in which the hardware is computer controlled to change the architecture to any desired one in real time for rapid prototyping purposes for example. That provides ultra fast adaptability and optimization for application-specific needs.

5.3 Sequential Boolean expressions

The sequential Boolean expression is defined here as one that is represented by gates that are all having one optical input and one electrical input, hence can be cascaded in sequence. As an example, Fig. 12 shows the digital design of the Boolean expression ABC+D.

That design can be implemented using the all-optical processing standard gates discussed in Sec. 3, where the microelectronic gates are replaced with the optical ones. It can also be implemented as an RR-architecture POP. Table 11 shows the extended truth table for that expression and Fig. 13 shows the reduced design as an RR-architecture POP: following the design algorithm of Sec. 5.1 and the design reduction rules of Sec. 5. 2. B.

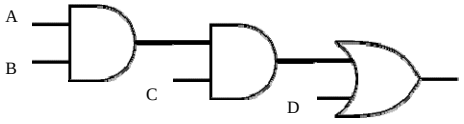

Fig. 12. Digital circuit design of the sequential Boolean expression ABC + D.

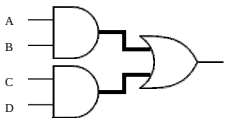

Fig. 14. Digital circuit design of the non-sequential Boolean expression AB+CD.

Note that only three BSs are used, along with five CPIBs. The number of CPIBs required is only two, where the three upper CPIB1s are to be combined into one by directing the output of the three polarizers of the three LZBs together through one CPIB1. Therefore, the design is actually composed of three BSs, two CPIB0, and one CPIB1. Always remember that the output carrier is only one carrier, and that all output paths converge into one by steering. They are left unsteered for clarity.

5.4 Non-sequential Boolean expressions and electro-elimination

The non-sequential Boolean expression is defined here as one that includes gates with two optical inputs. For example see Fig. 14 which represents the Boolean expression AB+CD. Such a Boolean expression can be implemented using the all-optical processing gates as discussed in Sec. 3 as gates of one optical input only when one gate output is converted to electrical input. On the other hand, it can also be implemented using the RR-architecture POP discussed above by applying the design algorithm of Sec. 5.1 eliminating the need to converting any gate output into electrical input, hence electro-elimination. The results are given in Table 12 and Fig. 15. As always, the bullet train simulation simplifies understanding the operation of the design. That is left to the reader as a straight forward exercise to limit the size of the paper and to avoid repetition.

The RR-architecture POP design of Fig. 15 clearly invites elimination and reduction. It is clear that it reduces to a three-BS three-CPIB design.

#73748 - $15.00 USD Received 7 August 2006; revised 17 September 2006; accepted 18 September 2006

The implementation of Fredkin, Toffoli, testable reversible-logic, and other gates by directly applying the design algorithm of Sec. 5.1 are discussed in other publications.

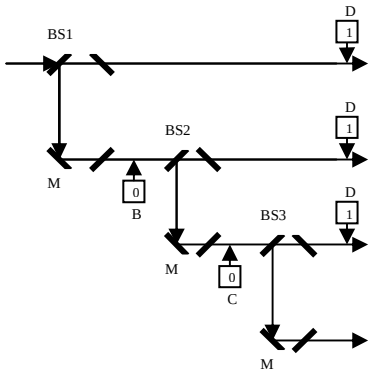

Fig. 13. RR-architecture POP design of the sequential Boolean expression ABC+D.

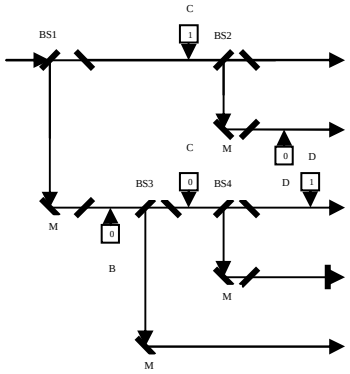

Fig. 15. RR-architecture POP design of the non-sequential Boolean expression AB+CD.

Table 11. Extended truth table for the sequential Boolean expression ABC+D.

A	a	B	b	C	c	D	d	Z
0	0	0	0	0	0	0	0	0
						.-.-.-.-.-.-.-.		
0	0	0	0	0	0	1	1	1
						………..		
0	0	0	0	1	0	0	0	0
0	0	0	0	1	0	1	1	1

0	0	1	0	0	0	0	0	0
0	0	1	0	0	0	1	1	1
0	0	1	0	1	0	0	0	0
0	0	1	0	1	0	1	1	1
1	1	0	0	0	0	0	1	0
						.-.-.-.-.-.-.-.		
1	1	0	0	0	0	1	0	1
						………..		
1	1	0	0	1	0	0	1	0
1	1	0	0	1	0	1	0	1

1	1	1	1	0	0	0	1	0
						.-.-.-.-.-.-.-.		
1	1	1	1	0	0	1	0	1
						………..		
1	1	1	1	1	1	0	1	1
						.-.-.-.-.-.-.-.		
1	1	1	1	1	1	1	1	1

Table 12. Extended truth table for the non-sequential Boolean expression AB+CD.

A	a	B	b	C	c	D	d	Z
0	0	0	0	0	0	0	0	0
						.-.-.-.-.-.-.-		
0	0	0	0	0	0	1	0	0
						…….		
0	0	0	0	1	1	0	1	0
0	0	0	0	1	1	1	0	1

0	0	1	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0
0	0	1	0	1	1	0	1	0
0	0	1	0	1	1	1	0	1
1	1	0	0	0	1	0	1	0
						.-.-.-.-.-.-.-.		
1	1	0	0	0	1	1	1	0
						…….		
1	1	0	0	1	0	0	1	0
1	1	0	0	1	0	1	0	1

1	1	1	1	0	1	0	1	1
						.-.-.-.-.-.-.-.		
1	1	1	1	0	1	1	1	1
						…….		
1	1	1	1	1	1	0	1	1
						.-.-.-.-.-.-.-.		
1	1	1	1	1	1	1	1	1

#73748 - $15.00 USD Received 7 August 2006; revised 17 September 2006; accepted 18 September 2006

6. Conclusion

We introduced the orthoparallel optical logic (OPL) architecture to design and implement any and all binary gates. A clear step-by-step design methodology is presented and applied to design AND, NAND, OR, NOR, XOR, and XNOR gates, in addition to inverters. We also discuss the operation of each gate. In addition, the general gate concept is introduced. The design and operation of all gates are presented using the special case of ±45° OPL, where the logic 1 (L1) and logic 0 (L0) are represented by linearly polarized waves at +45° and -45°, respectively. General OPL architecture is also discussed where L1 and L0 are represented by any two orthogonal polarization states, which opens the door for reconfigurable optical microprocessors and to programmable optical logic gates. The railroad (RR) architecture of polarization optical processors (POPs) is introduced to design any complex Boolean expression, which lead to the concept of electro-elimination to continuously process the information optically without the need to convert the laser beam into an electronic signal. The POP is very easily understood by the simulation of a bullet train traveling at the speed of light on a railroad system preconditioned by the crossovers determined by the control signals. All control signals are applied to the POP simultaneously eliminating the need for a timing diagram and any problems associated with the inherent time delays of electronic microprocessors. Therefore, the POP is not vulnerable to glitches and is inherently stable. The RR-architecture POP can be used to design any binary gate with any truth table. Therefore, it is equally applicable to all types of binary gates, including for example Fredkin gates, Toffoli gate, threshold gates, testable reversible-logic gates, and all future gates.

#73748 - $15.00 USD Received 7 August 2006; revised 17 September 2006; accepted 18 September 2006