Method to analyze the influence of hysteresis in optical arithmetic units

Ana González-Marcos, MEMBER SPIE
José A. Martín-Pereda, MEMBER SPIE
Universidad Politécnica de Madrid
E.T.S. Ingenieros de Telecomunicación.
Ciudad Universitaria
E-28040 Madrid, Spain
E-mail: agonmar@tfo.upm.es

Abstract. A new method to analyze the influence of possible hysteresis cycles in devices employed for optical computing architectures is reported. A simple full adder structure is taken as the basis for this method. Single units, called optical programmable logic cells, previously reported by the authors, compose this structure. These cells employ, as basic devices, on-off and SEED-like components. Their hysteresis cycles have been modeled by numerical analysis. The influence of the different characteristic cycles is studied with respect to the obtained possible errors at the output. Two different approaches have been adopted. The first one shows the change in the arithmetic result output with respect to the different values and positions of the hysteresis cycle. The second one offers a similar result, but in a polar diagram where the total behavior of the system is better analyzed. [DOI: 10.1117/1.1413747]

Subject terms: optical computing; optical logic cell; arithmetic units.

Paper OC-007 received Feb. 10, 2001; revised manuscript received June 22, 2001; accepted for publication June 25, 2001.

1 Introduction

Several hundred papers have been published in the last twenty years concerning different architectures and algorithms able to implement possible future optical computers. They have taken different points of view. Most of them may be grouped into two basic philosophies. The first one is related to analogous structures employed in electronic computers. Such papers adopt a serial type of processing, with similar ideas to currently used structures. For these the main advantage of employing optical concepts comes from the possible alleviation of typical bottlenecks at certain points as well as the higher speed obtained with photonic devices. The second approach is related to the parallel character of light propagation. In it, the principal benefit obtained is the higher information capacity as well as the possibility of application to image processing and pattern recognition. Both approaches have, at the same time, many pros and cons—in some cases pros and cons arising from the same features. Although photonic devices have an inherent capability to work at much higher speed than their electronic counterparts, their power consumption and size are usually larger. Moreover, from the first previously indicated point of view, to employ serial design, when the main advantage of optics is its parallel properties, appears as a serious drawback. But parallel approaches require, at least in most of the published work, structures very difficult to implement in practical systems. This difficulty is more significant if these systems are to be employed in a similar way and in similar situations to the present electronic computers.

Some of the above problems may find solution in the near future if some recently reported proposals, employing photonic materials and concepts, are developed further. In that case, the two previously indicated philosophies may come together. Devices with very high speeds may be stacked in single optical chips, with very low power consumption and the possibility of working in a parallel configuration. This situation has a very important advantage: It may employ many of the previously developed approaches in electronic computers and, at the same time, be compatible with them.

Another consideration of importance is the location and circumstances of the proposed structures. One of the areas where optics and photonics are more active is in optical communications. For this reason, many of the present developments in this area should be employed in any future optical computer. It should be of great interest, for example, to employ the same optical frequencies as in optical communications. Wavelengths around 1.3 and 1.55 μm should preferably be used. Moreover, the present dense wavelength-division multiplexing (DWDM) systems employ frequencies in the third window covering a range of around 15 THz. This implies that any device should work properly in this range and no difference in behavior should be observed when changing from one wavelength to another.

Whether in either of the two previously mentioned cases or when using optical communication devices, some facts remain important. One of them, and it will be the main subject of the present paper, is the influence of the devices' tolerance on the obtained final results. In most of the reported works in optical computing, devices act as ideal components. This assumption may be valid when a system works in a research laboratory or in a field trial with a supervisor in attendance. But if the optical computer is to be a common instrument, real behavior needs to be considered. Devices have a certain tolerance, and operating parameters may change from one device to another. Moreover, they may change from time to time, depending on

external parameters as, for example, temperature. A concrete example of these considerations is related to the well-known self-electro-optic effect device (SEED).[1,2] Its behavior is strongly dependent on temperature. Small changes in operating temperatures affect its absorption properties, and in consequence its nonlinear characteristics may change from time to time. At the same time, if the wavelength of the impinging light changes, its working parameters are affected too. This is the situation with respect to the use of components and wavelengths in optical communications. If these devices are going to be used in logic circuits, these changes may significantly affect the final results. If the final behavior of the optical computer has to be reliable, these problems need to be taken into account.

Thus, the influence of device tolerances on optical arithmetic computation will be the principal task of this work. Some common operations—those performed by such basic circuits as the half adder and full adder—will be taken as examples of our approach. Other more complex operations, such as optical parallel addition and multiplication of words, optical matrix computation, or, in general, optical sequential machines, may be analyzed in a similar way. They will be the subject of future work.

In order to perform the analysis, we will take an optical programmable logic cell (OPLC) as the elementary unit of the system to be studied. We have previously studied this cell and reported its logical properties.[3] We have analyzed its nonlinear behavior, mainly when the cell is acting under certain chaotic conditions. Its resulting irregular behavior was analyzed with fractal concepts.[4] The application of this cell to circuits such as the half and full adder will constitute the first part of this paper. The second part will be devoted to the influence of the tolerance of the elementary devices on these arithmetic operations.

2 Basic Unit of the System: Optical Programmable Logic Cell

Although the basic unit to be employed here has been reported previously in several places,[3,4] some details will be presented here for use in our present work. The basic configuration appears in Fig. 1. The input and output signals are optical binary data, and the employed devices are nonlinear optoelectronic devices with digital characteristics. Figure 1(a) shows a block representation of the basic cell, and Fig. 1(b) its internal configuration. Two optical devices, *P* and *Q*, with a nonlinear behavior, compose the circuit. The outputs of each one of them correspond to the two final outputs, O_1 and O_2, of the cell. The possible inputs to the circuit are four. Two of them, I_1 and I_2, are for input data, and the other two, *g* and *h*, for control signals. The way these four inputs are arranged inside the circuit is also represented in Fig. 1(b). The corresponding inputs to the nonlinear devices, *P* and *Q*, are functions of these signals plus, in the case of *P*, one other coming from inside its own cell and obtained from *Q*.

A practical implementation of this structure that we carried out was based on an optoelectronical configuration. Lines in Fig. 1(b) represent optical multimode fibers. The indicated blocks, placed to combine the corresponding signals, are conventional optical couplers. These couplers, depending on their function, are 2×2, 1×2, or 2×1. In this

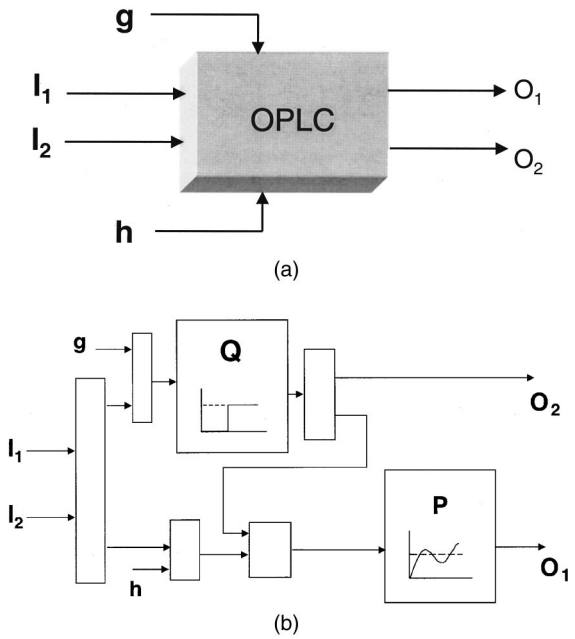


Fig. 1 (a) Block representation of the optically programmable logic cell (OPLC); (b) internal representation of the OPLC. Blocks correspond to nonlinear optical devices—*P* and *Q*—and to optical couplers.

way, inputs arriving at the individual devices are multilevel signals. The working levels are

$$I_P=\frac{I_1+I_2}{2}+g+\frac{O_2}{2} \tag{1}$$

for *P*, and

$$I_Q=\frac{I_1+I_2}{2}+h \tag{2}$$

for *Q*. The output from *Q* is divided into two equal-intensity signals. One of them is the final output O_2, and the other one, $I_3=O_2/2$, becomes a part of the input to *P*. This behavior was computer-simulated.

Therefore, the output of the device *P* depends on the control signal *g* plus one-half the output O_2 of the device *Q*. The output of *Q* depends only on its control signal *h*. We understand *output* as the type of processing, or logical function, that each one of the devices executes on the two binary data inputs.

The characteristics of the nonlinear devices are also shown in Fig. 1(b). The device *Q* is a threshold or switching device, and *P* is a multistate device, having the ideal response of this nonlinear optical device. This response is similar to that of a SEED. Because the input signal is a multilevel signal, as can be seen from Eqs. (1) and (2), the output depends on the relations between:

1. the level of a 1 bit
2. the level of the control signal
3. the level for switching from one state to another (this level is intrinsic to the employed device).

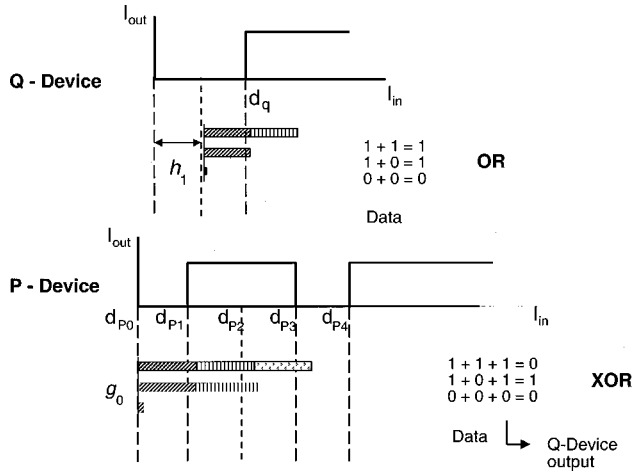


Fig. 2 Example of input/output characteristics for devices *P* and *Q* for a particular situation; $d_{P,Q}$ are the corresponding decision levels.

Table 1 Summary of pairs of logic functions. O_2 is the output from *Q*; "C.s." stands for "Control signal." The outputs for a half adder are italicized; those for a full adder are in boldface.

	Output O_1 from *P*		
C.s. to *P*	O_2 = ***AND***; C.s to *Q*: 0–0.4	O_2 = **OR**; C.s. to *Q*: 0.5–0.9	O_2 = ON; C.s. to *Q*: 1.0–2.0
0–0.4	***XOR***	XOR	NAND
0.5	NAND	NOR	NOR
0.6–0.9	ON	**XNOR**	XNOR
1.0	XNOR	**XNOR**	AND
1.1–1.4	XNOR	ON	OR
1.5	AND	OR	OR
1.6–2.0	OR	OR	ON
2.0–2.5	ON	ON	ON

In order to clarify the above facts, we have represented in Fig. 2 how to obtain one pair of possible outputs. The previously mentioned parameters are indicated in the same figure. Moreover, the nonlinear devices *P* and *Q* appear in the ideal form adopted for them. Other logical operations can be obtained in a similar way.

The value of a 1 bit at either of the two inputs of the cell, namely I_1 or I_2, has been considered as the normalization value for the simulation of an OPLC. As changing parameters have been taken:

1. the decision levels of each device, d_Q and d_P (see Fig. 2)
2. the two control signals, *g* and *h*.

Moreover, we have considered five equidistant levels $d_{P0},\ldots,d_{P4}$ of decision at the device *P*. However, there are just three input levels, according to which the output is able to switch from one state to another.

The behavior of the output O_1, for functions ON or OFF at output O_2, is shown in Fig. 3. The ordinate is the power level *h* of the control signal, and the abscissa, the power level d_{P1} of the decision signal.

The eight Boolean functions—AND, OR, XOR, ON, and the negative NAND, NOR, XNOR, and OFF—can be obtained at the output O_1. Figure 3 shows one of the possible results from this cell. This diagram is called the 2-D *working diagram*. Table 1 sums up all the pairs of logic functions and the conditions to obtain them.

There is one last parameter to be fixed. It is the width of the hysteresis cycles. Figure 4(a) shows a way of modeling the device *P*. As can be seen, it is based on three step functions related according to the rules indicated in the figure. In any case, there is a certain tolerance in the definition

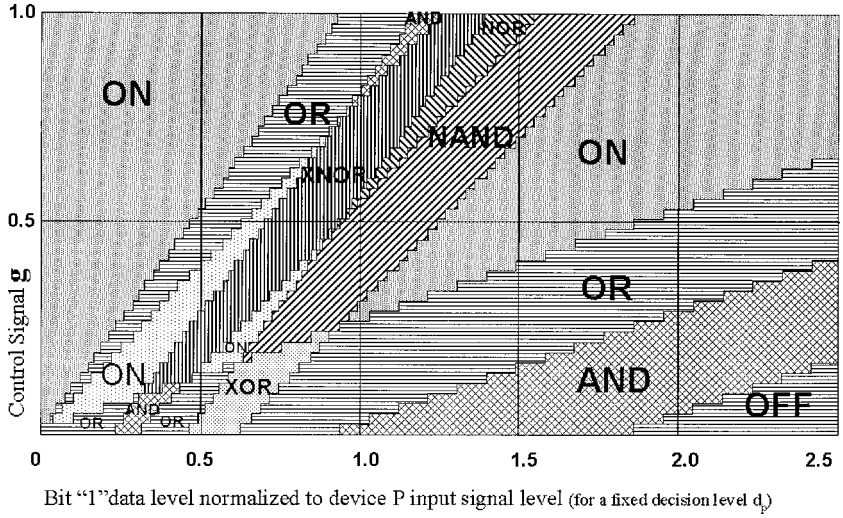


Fig. 3 Behavior of the output O_1 of the OPLC for an ON output at gate O_2. The axes show the power level *h* of the control signal versus the power level d_{P1} of the decision signal. Both signals are normalized to an input bit 1.

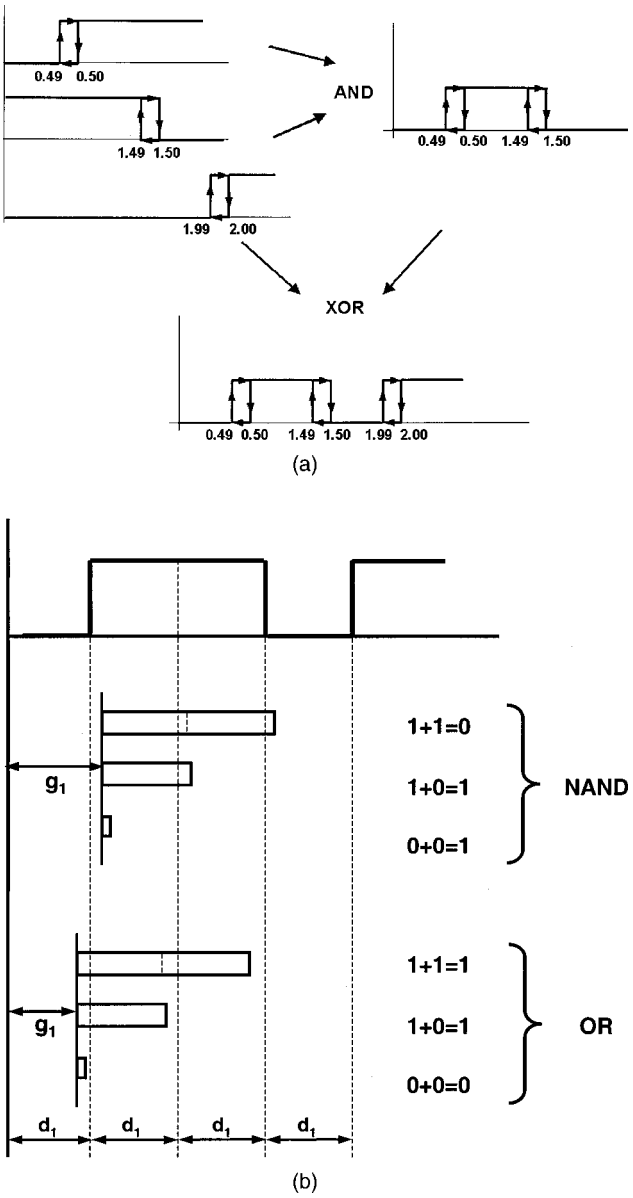


Fig. 4 (a) Internal tolerances of *P*; (b) example of the influence of the external tolerance on the obtained logic functions. Small variations in the control signal g_1 affect the final result.

of the transition from the lower to the higher state. This tolerance may be called the *internal tolerance*. Moreover, there is an *external tolerance* related to the real position of the control value with respect to the decision level. An example appears in Fig. 4(b). The same input signals may give different results depending on where the control signal is located with respect to the decision level. In the example shown, a NAND function is performed in one case, and an OR in the other. These tolerances are of great importance when logic cells are applied in optical computing.

3 Arithmetical Units Based on the OPLC

If a cell as described is to be employed in optical computing architectures, basic configurations as, for example, the half adder and full adder must be implemented. As was indicated, the cell is able to perform the main Boolean functions and, as a consequence, arithmetical operations.

The design of a half adder using our OPLC is based on well-known equations. A simple design appears in Fig. 5(a). As can be seen, the control signals are zero. Hence, O_1 is an XOR function of input data, and O_2 is an AND function. They correspond to the equations for a half-adder: $S_i = A_i \oplus B_i$ and $C_i = A_i B_i$. Table 1 summarizes the OPLC's behavior for the corresponding control levels. In the table are indicated, in italic, the two corresponding outputs for a half adder. Some results are shown in Fig. 5(b). Due to the graphical simulation method employed, the graphical bit representation is not the conventional one. Bits 1 and 0 are not just horizontal lines at the corresponding levels. There are diagonal lines between obtained zeros and ones, and this gives a triangular appearance to the bit series. Indicated in Fig. 5(a) are the input data, the sum, and the carry.

Real binary addition needs to take into account the carry generated from previous bits. The simplest adder is a one-bit adder that serially adds two *n*-bit numbers, *A* and *B*, by taking operand bit pairs at a rate of one per cycle and obtaining resulting bits at the same rate. Such an adder is called a full adder and is designed from the corresponding logic table. This involves three inputs and two outputs. Our OPLC has these characteristics. In this case, the three inputs are the two added bits A_i and B_i and a carry-in C_{i-1}, which is the carry-out from the addition of A_{i-1} and B_{i-1}. The two outputs are a sum bit S_i and a carry-out bit C_i to the next stage. This gives the well-known equations

$$S_i = (A_i \oplus B_i) \oplus C_{i-1}, \tag{3}$$

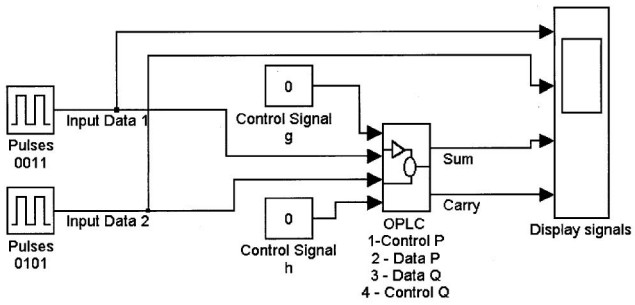
$$C_i = A_i B_i + (A_i \oplus B_i) C_{i-1}. \tag{4}$$

The terms $A_i \oplus B_i$ and $A_i B_i$ are the sum and carry bits without the incoming carry taken into account. They are known as the partial sum and partial carry. In order to understand how the full adder is implemented with the OPLC, one must analyze the two possible cases for the value of previous carrier in Eq. (3) and (4):

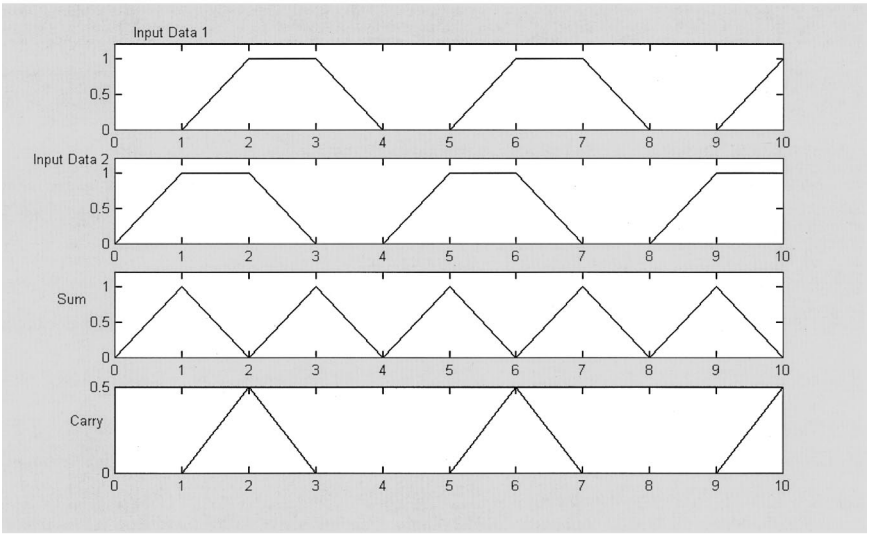
1. $C_{i-1}=0$: $S_i = (A_i \oplus B_i) \oplus 0 = \overline{(A_i \oplus B_i)}0 + (A_i \oplus B_i)1 = 0 + (A_i \oplus B_i) = (A_i \oplus B_i)$, function XOR; $C_i = A_i B_i + (A_i \oplus B_i)0 = A_i B_i$, function AND.
2. $C_{i-1}=1$: $S_i = (A_i \oplus B_i) \oplus 1 = \overline{(A_i \oplus B_i)}1 + (A_i \oplus B_i)0 = \overline{(A_i \oplus B_i)}$, function XNOR; $C_i = A_i B_i + (A_i + B_i)1 = A_i B_i + (A_i + B_i) = A_i + B_i$, function OR.

In Table 1 the corresponding pair of functions needed for the full adder is shown in boldface.

A full adder, in Simulink™ modeling, appears in Fig. 6(a). Every one of the possible input options is presented. It appears, from a detailed study of the full-adder truth table, that the carry output from O_2 must be applied as a control signal to the OPLC. The input carry C_{n-1} is applied as the *g* and *h* control signals. Different Boolean functions are generated depending on their values. Thus the full-adder truth table may be obtained. Figure 6(b) shows the result of



(a)



(b)

Fig. 5 (a) Half-adder Simulink™ model; (b) simulation results for sum and carry, one bit per time step.

our simulation. A 1×2 coupler, with a specific coupler relation, is the only optical component needed to assign the O_2 output to the control-signal input.

4 Device Modeling

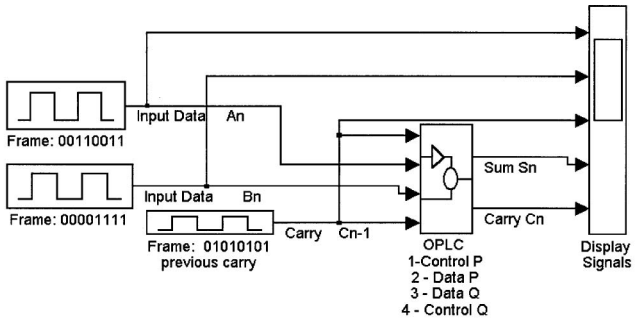
The SEED, or some similar device, is the main component to be employed in the OPLC to configure a logic circuit. Its transfer function for optical output versus optical input power displays nonlinear behavior suitable to perform any logic function. But this behavior is not truly static and well-defined. Its characteristics may vary when, for instance, the wavelength of the incident or control beam changes slightly. Similar changes occur when temperature varies. These factors lead to changes in the transfer function and, as a consequence, in the characteristics of the output light. These changes may alter the logic function performed, and thereby yield a different result for the intended arithmetic operation.

In a previous paper,[4] the effect on the final result of small changes in the input signals was studied. The study was performed with the help of the fractal structure offered by a new working diagram adopted for our optical logic cell. This working diagram represents the input/output characteristic in three dimensions, and it is an extension of the previous 2-D working diagram presented in Fig. 3. The Z axis now gives the Boolean function generated at the output, corresponding to a particular pair of values for the control and input signals. The Boolean function is characterized by a hexadecimal value. The reason for choosing this representation was indicated in that paper.

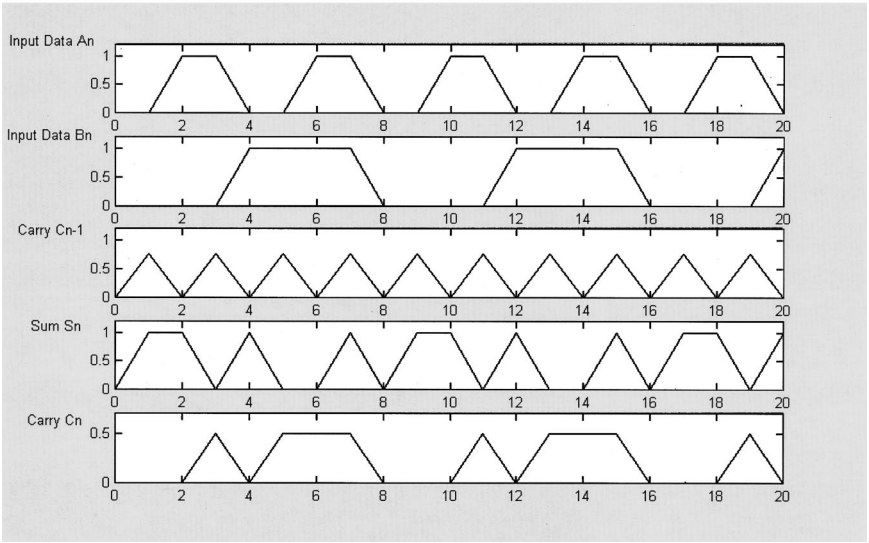
In order to obtain that diagram, the two input data signals must maintain a fixed relation between their frequencies. In this case, the period of one input signal has to be double that of the other one. In this way, the internal input signal to P is a multilevel signal, with a pattern like [2, 1, 1, 0] for every four bits. Then, every four bits of the output signal correspond to a Boolean-function truth table. We convert these four bits to a hexadecimal value and obtain the corresponding point in the working diagram.

The model to be studied takes into account some of the above facts. The main point is that we have simulated the possible variations in the SEED's behavior as small hysteresis cycles at the zones where there is a change, from 1 to 0 or from 0 to 1, in the output level.

Figure 7 shows how we represent the device P and how we have modeled it (see also Fig. 4). As can be seen, it is based on step functions related according to the rules indicated in Fig. 7. As was indicated before, there is a certain tolerance in the definition of the transition from the lower to the higher state. This tolerance, the *internal tolerance*, has as a consequence the presence of the hysteresis cycles. Moreover, this tolerance is the way we take into account



(a)



(b)

Fig. 6 (a) Full-adder Simulink™ model; (b) simulation results for all possible combinations of input data and carriers.

the possible working changes due to either internal or external factors: ambient temperature, device temperature, and input and control beam wavelengths.

5 Dynamical Behavior of an Arithmetic Unit

There are several possibilities to analyze the internal behavior of our OPLC when employed as a part of circuits for optical computing. Its behavior when performing logic functions was studied in a previous paper.[4] In that case, the intended objective was only an analysis of the simple optical logic cell in performing logic operations. The basis of the method employed was a 3-D working diagram where the axes represented control-signal values and tolerances of input signals. The effect of small variations of the input signals on the final results was studied there. The obtained results showed different regions in the working diagram for the different logic functions implemented by the cell. The most interesting result was a fractal structure in certain regions and how this fractal changed with input signal values. The objective of the present paper is to show the ways our logic cell works as part of an arithmetic logic unit. For this purpose we have performed two types of analysis. The first one is just a simple test in order to get an idea of the final results, in comparison with the correct ones, for some values of the device tolerances. The second one is a closer look to see the effects of tolerance values.

The arithmetic unit to be studied is a ripple carry adder, which has not been a popular unit for a long time, but presents some advantage as a regular structure. More details can be found in Ref. 5. This type of adder can be obtained by connecting *n* full adders in a chain, with the carry output from one full adder connected to the carry input of the next full adder. All the operand bits can be fed into the adder in parallel. Also, no storage or delay is needed for the intermediate carries. Figure 8 shows the Simulink™ model of a 4-bit ripple carry adder. Four blocks can be seen, which correspond to the full adder represented in Fig. 6(a) without the pulse generators.

A first analysis was based on the results obtained from a random sequence of numbers applied to the inputs of the ripple carry adder. The result is subtracted from the correct result. Abscissa indicate evolution in time, and ordinates the resulting error values. If no errors are present, the final result should be a straight line. When errors are present, values higher or lower than zero should be obtained. Figure 9 shows the situation when a change in width of the PE1 relays has been imposed. This is only a first indication of

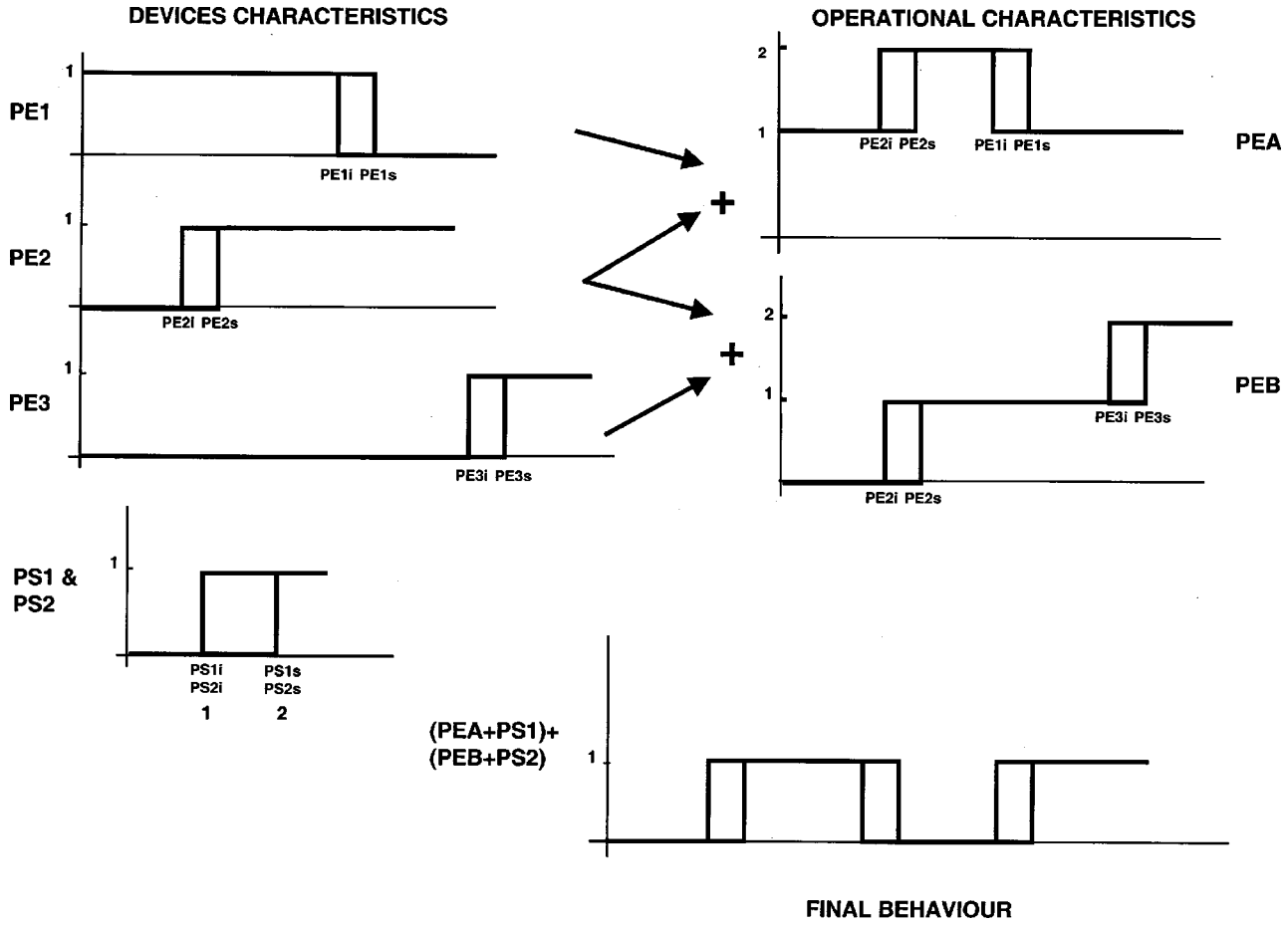


Fig. 7 Modeling of device *P*, in Fig. 1(b), with SEED threshold behavior.

changes in the final result when device characteristics are changed.

The general result reported gives no indication of the influence of particular changes on different classes of device characteristics. As has been pointed out before, our computer simulation of the device *P* is based on the simulation shown in Fig. 7. This figure indicates the way it has been implemented. As can be seen, the initial step is to take three on-off characteristics with a certain tolerance. This tolerance corresponds to the possible hysteresis cycles in real components. They have been named PE1, PE2, and PE3. They are added in the way indicated in the figure, and the results go to two new devices with the indicated characteristics. They are devices PS1 and PS2.

In any real component, the position and width of the hysteresis cycles would change in one or the other direction when some external or internal condition changed. These variations have been analyzed with 12 possible tolerances imposed on the basic units of the OPLC acting as full adders. They are shown in Fig. 8 as ''vector status of hysteresis cycle,'' and they have been designated, as before PE1, PE2, PE3, PS1, and PS2.

Our present simulation changes the position of the hysteresis cycles without changing their widths. Although both devices, *P* and *Q*, could vary their characteristics, we have just considered variations in *P*. These changes were made one by one. Five cycles are analyzed. Some of our results are presented in Fig. 10. The abscissa is the change in position of the hysteresis cycle, and the ordinate is the intended result without any error. The value 0 on the abscissa is the correct cycle position. Green indicates that there are no errors when two numbers are added. Other colors indicate errors. Gradual changes in color indicate gradual changes in error values. Darker colors indicate larger errors. Red means positive error, and blue, negative.

Figure 10(a) shows the results for a change of the hysteresis cycle PS1 from -4.5 to $+4.5$. The precision is 0.1. This means that displacements steps have that value. We observe a vertical green band around the 0 position, between -0.1 and 0.1. However, there are errors in every other region.

In order to obtain a better understanding of this behavior, we have made smaller changes in the imposed variations. Figure 10(b) shows the results when the changes are 0.01, in the range from -1.19 to 1.01. There is a difference between the two cases. In (a) we got a region with no errors between -0.1 and $+0.1$, but when the changes are reduced to 0.01, the region gets much smaller: from -0.01 to $+0.01$. This indicates that this region of the total *P* characteristics has the largest influence on the final result. There is almost no area outside of the indicated region where no error appears.

A similar study performed for hysteresis cycle PS2 gives a very different result. The variation is now from -4.5 to

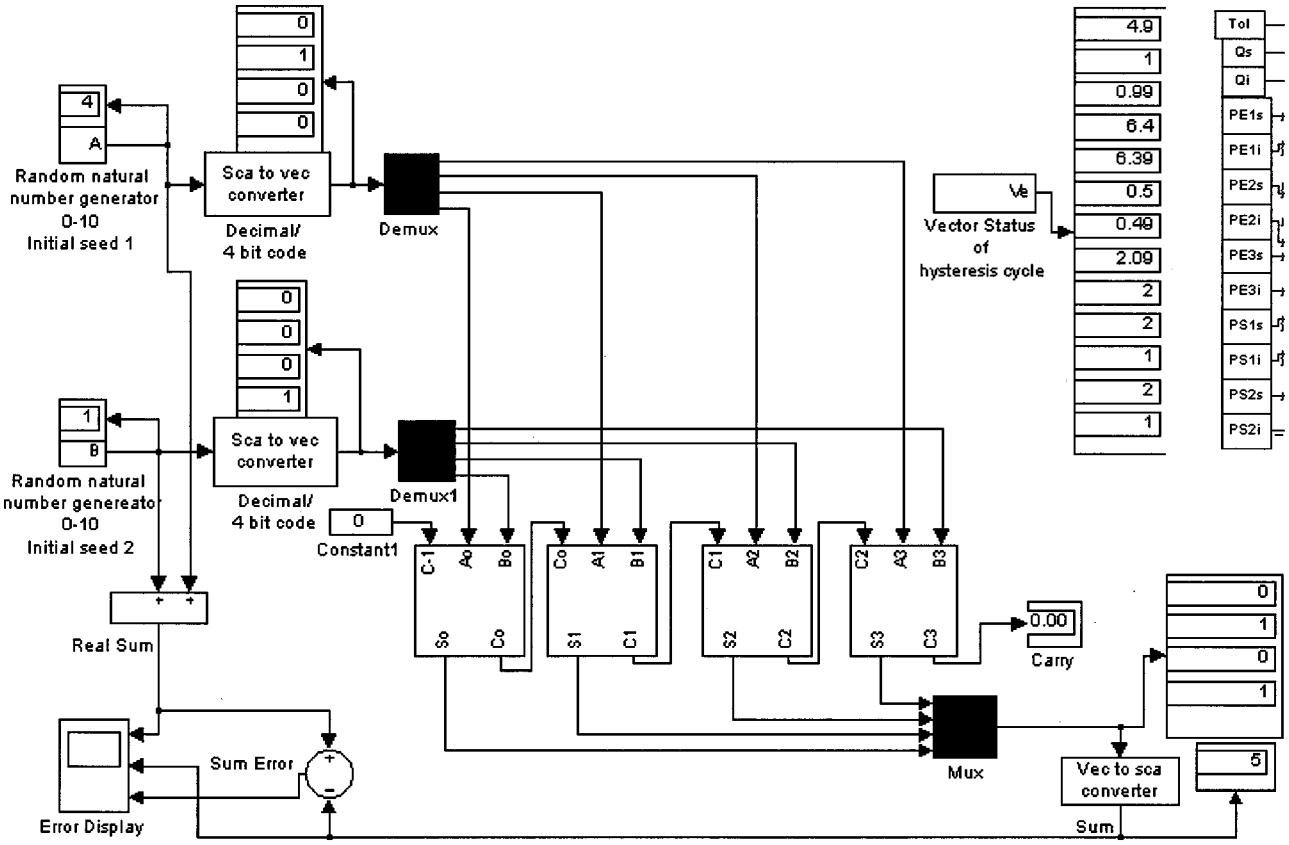


Fig. 8 Simulink™ model of ripple carry adder, with randomly generated decimal numbers (maximum number:16) plus conversion to the binary system.

+4.5. The region with positive changes shows almost no error at all. There is just a small error around 10. Every zone outside this sector is free of errors. On the contrary, negative displacements give errors for almost any possible change. This indicates that these variations are much more influential than positive ones. See Fig. 10(c).

It is possible to infer from these results that changes in position of the hysteresis cycles alter the final result. There

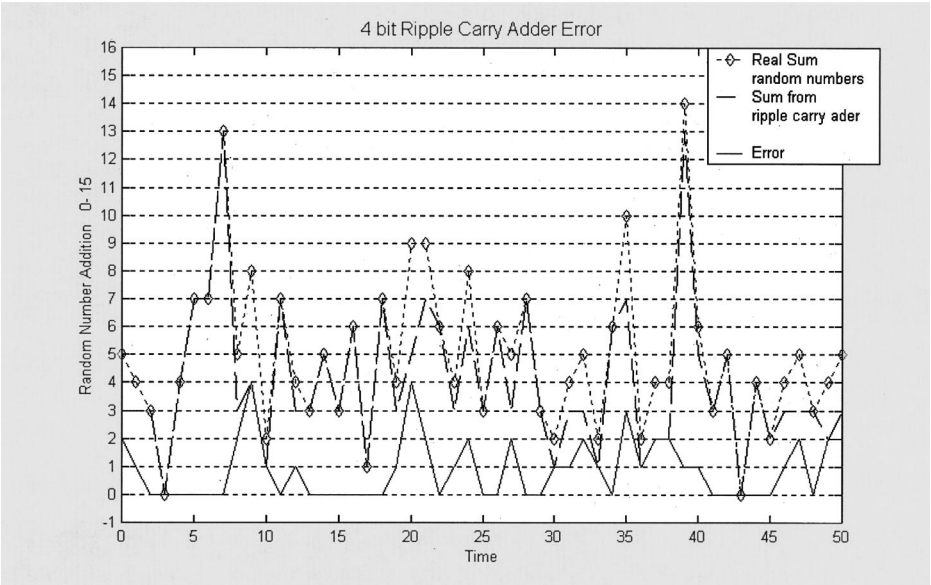


Fig. 9 Addition of two random numbers (between 0 and 15): real sum and obtained result plus resulting error displayed in Fig. 8 as "error display block."

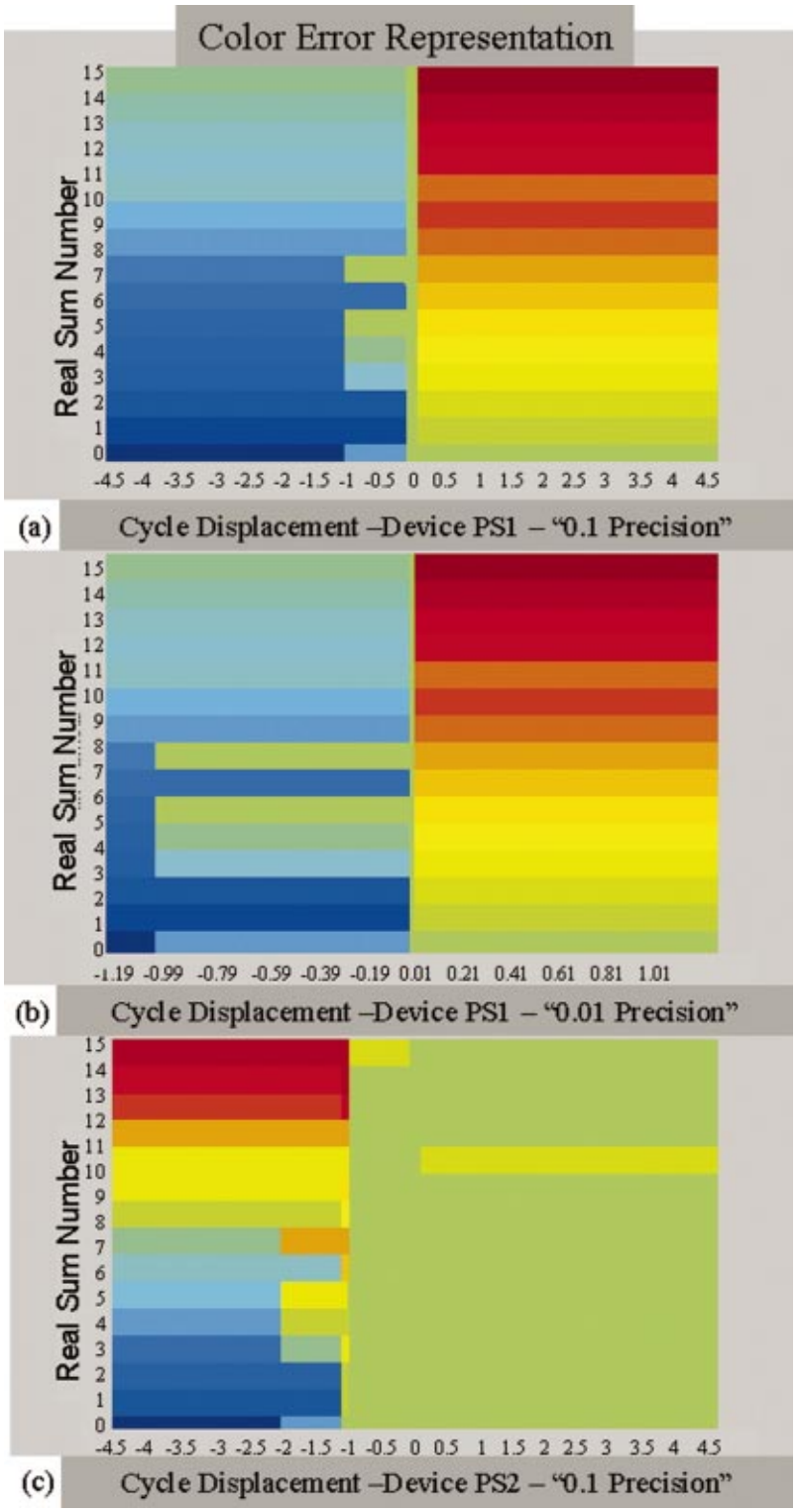


Fig. 10 Error representation for displacements (Tol) of the relay block width: (a) precision 0.1, on the PS1 device (PE1i+To1, PE1s+Tol); (b) precision 0.01, on the PS1 device; (c) precision 0.01, on the PS2 device. Green color indicates no error.

are similar consequences of small modifications in their widths. We do not present the results here, because that is not the objective of the present paper.

Although this method may be adequate to obtain a first indication about the final behavior of the structure, another representation will allow easier interpretation. This new representation is introduced in this paper. Its aims are to clarify how the different regions of our simulated devices influence the final result. Such information has already been partly presented in the previous figures. But this new representation gives a more immediate idea about the zones where the full architecture yields results with some error, as well as their evolution. This knowledge will give information needed for corrections in order to cancel the error. In addition, this new representation has some dynamical characteristics that make it useful for other applications.

The representation is based on a certain type of polar coordinates. Different radial axes correspond to different results. We have restricted these values, in our case, to integers from 0 to 15. Hence, we have 16 radial axes. In the case when the obtained result is the correct one, the value to be represented should be zero. Other obtained values are represented with respect to the correct one. For instance, if the correct result is 7, and the system gives 3, the represented value is 4. If 9 is obtained, the represented value is -2. In this way, it is possible to trace the error evolution when a particular parameter changes its value. The results for some of the analyzed cases are summarized in Figs. 11–14.

Figures 11–14 show the whole behavior of the system when some parameters of *P* change. These results are shown to indicate the possibilities offered by our new representation. We will present some conclusions drawn from the obtained figures. But, as we have indicated, the main result of our paper is to reveal the potentialities of the adopted method.

We have analyzed four possible changes. They correspond to four of the five building blocks to configure the *P* device characteristics. As before, the changes we have considered are displacements of the hysteresis cycle to the right or to the left of the correct position, maintaining the width constant. Some of the obtained results are as follows.

5.1 *Changes in the PE2 Cycle*

The results appear in Fig. 11. Figure 11(a) shows the polar diagram when the hysteresis cycle moves to the right, and Fig. 11(b) when the motion is to the left. The same range of variation appears on both sides: from 0 to 1.19. In the first case, the inner circle corresponds to no errors in the total result (marked with 0); in the second one, it is the outer circle (also marked with 0) that does so. Hysteresis variations to the right correspond, at certain added numbers, to final results larger than the correct ones. Changes to the left give smaller numbers (indicated by the negative numbers that appear on the plot). Is interesting to point out that the larger errors correspond to opposite situations in each of the cases. Displacements to the left give larger errors for numbers smaller than 10, and displacements to the right give larger errors for numbers larger than 10. Moreover, errors are present at different positions of the displacement. There were no errors up to 0.48 when the motion was to the left, and significant errors from 0.24 upward when it was to the right.

5.2 *Changes in the PE3 Cycle*

This situation yields many more differences between the two types of displacements than the previous one. As one may see in Figs. 12(a) and 12(b), there are almost no errors when the motion of the hysteresis cycle is to the right. There is a single error when the two added numbers give 10, but there are no errors in any other situation. We have moved the cycle from 0 to 4.9. On the contrary, displacements to the left give errors for almost every added pair of numbers. There are seven different regions. Only from 0 to -0.3 are there no errors. Larger errors appear when the result is larger than 10.

A point that needs study concerns Fig. 12(a). It shows an error only for 10 as the final result of the arithmetic operation. To interpret this situation needs a deeper analysis of the operations performed in the full adder and how they are affected by particular displacements of the characteristics. This will be done in future work.

5.3 *Changes in the PS1 Cycle*

There is a significant result in this case. Similar evolution appears in the two situations, although for different values of the displacement. As may be seen in Figs. 13(a) and 13(b), a certain spiral curve appears. In the motion to the left, the curve goes to larger errors for smaller values of the added number, whereas in the motion to the right it goes to larger errors for larger numbers. Moreover, in both cases, zero error occurs only for the correct position of the hysteresis cycle. Any displacement gives rise to errors. There are three regions in one case (motion to the left) and two in the other one (motion to the right).

5.4 *Changes in the PS2 Cycle*

This situation is, in some aspects, similar to changes in the PE3 cycle: almost no errors for displacements to the right, and errors for any displacement to the left [Figs. 14(a) and 14(b)]. In this last case, it is important to point out that errors are now both positive and negative, depending on the added numbers. If the latter are larger than 4, there are positive errors in almost every case. If smaller, the errors are negative. A similar situation to that obtained in Fig. 12(a) appears in Fig. 14(a). There is error only when the result of the arithmetic operation is 10. This indicates that displacements to the right for cycles PE3 and PS2 give rise to similar situations. As in the previous situation, this result needs further study.

6 Conclusions

A new general method to analyze the influence of small changes in device characteristics when inserted in a circuit has been presented. The main example considered was the variation of the arithmetic result of a ripple carry adder. Although the method has been applied to that concrete problem, it may be employed in other situations. Its main advantage is the possibility of finding which one of the possible variations in the characteristics of some device has more effect on the final behavior. This influence may depend on the intended application. Our case was a SEED in

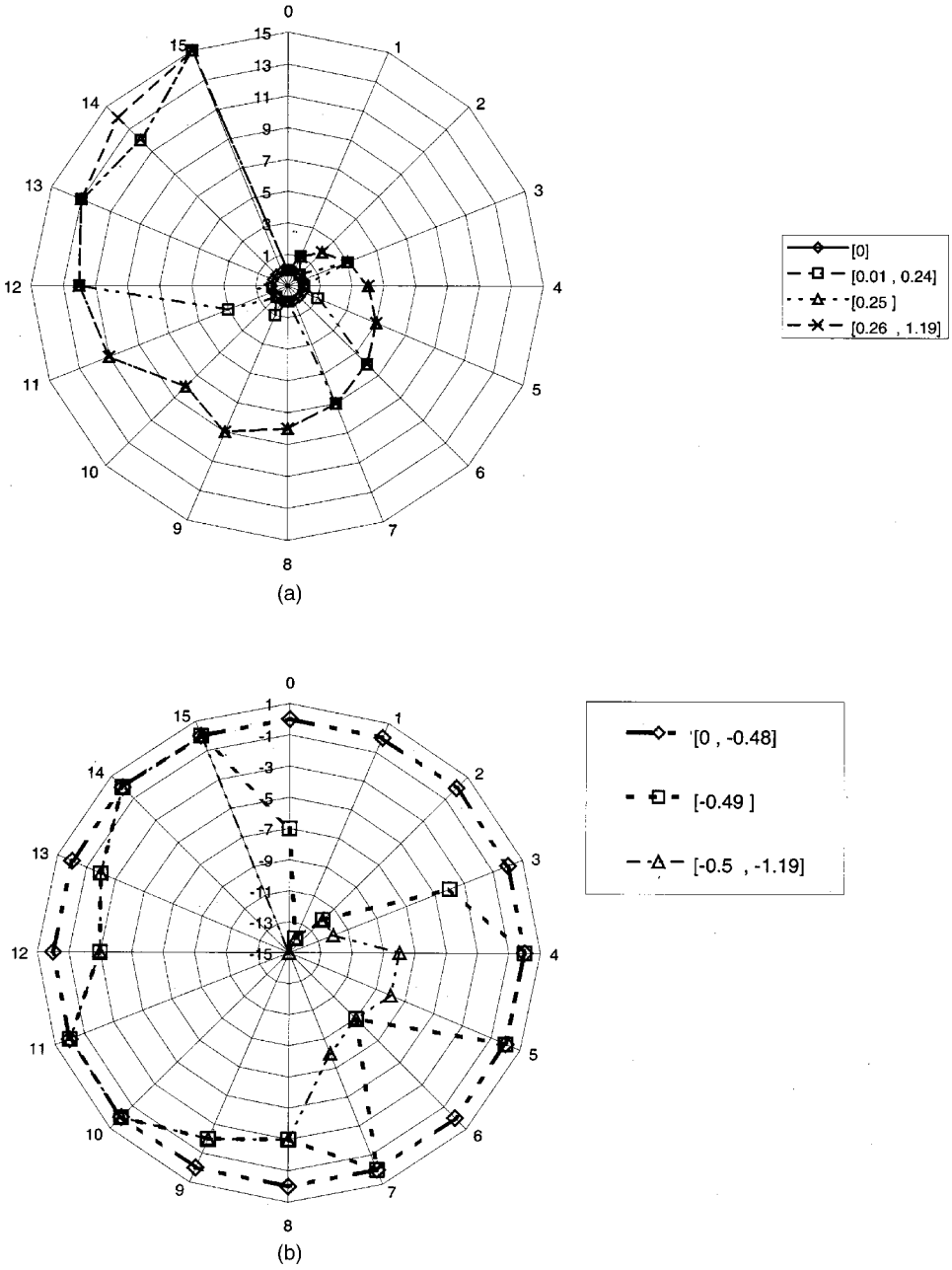


Fig. 11 Polar error representation of obtained results for the PE2 device. Radial axes correspond to the different numbers to be obtained. Radial distances give the error between the correct and the obtained result for a particular displacement of the device hysteresis. (a) Displacements to the right. (b) Displacements to the left.

a full adder. If the influence is known, it may be possible to compensate errors. As can be inferred from the obtained results, small changes in the characteristics of the employed devices—SEED devices in this case—alter the obtained final arithmetic result. Some particular examples have been presented in this paper. Final numerical values change when changes occur in the SEED characteristics. Errors can be compared with correct values in the figures. From the polar representation a quantitative indication of these errors is obtained. It determines the more crucial variations where there are possibilities of getting wrong results, and it indicates regions where possible variations in the device characteristics are less significant. This may have practical consequences. For example, changes due to temperature in a certain device may displace its characteristics towards a particular region. If this region corresponds to the zone with green color in the first representation, or to the 0 circle in the second (polar) one, temperature will not be a crucial parameter. If possible changes move the device to any other region, temperature will be significant. The polar representation gives a measure of the importance of error.

The reported method, employed together with methods previously reported by us to study the influence of small variations in the input signals on the obtained final logic

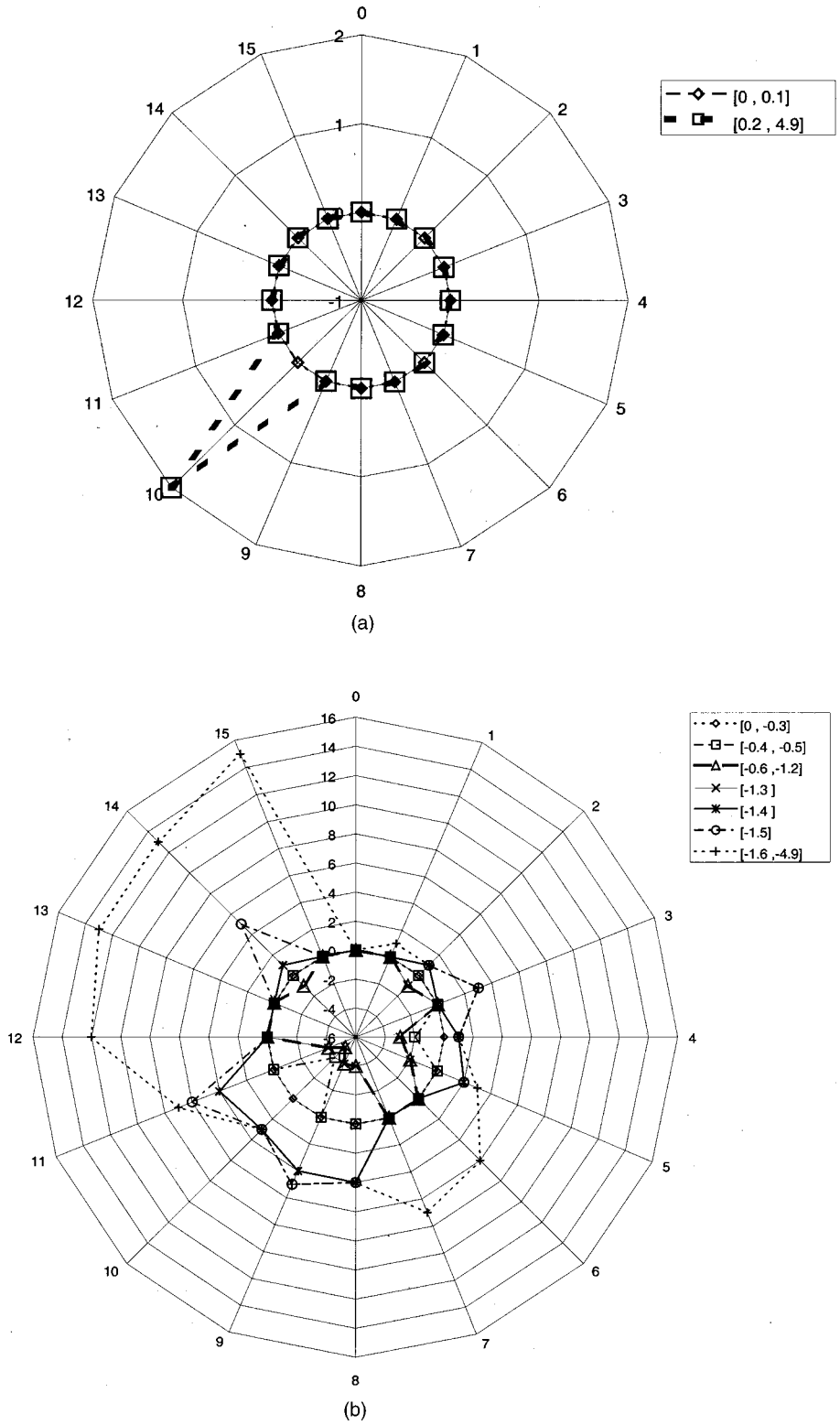


Fig. 12 Polar error representation of obtained results for the PE3 device. Same conventions as in Fig. 11.

functions,[4] may be of great utility for studying the behavior of any optical computing configuration. From an experimental point of view, the influence of possible changes in the characteristics of the employed devices may be known. These changes may be modeled according to similar rules to those reported here for a SEED device. In some cases, for example, there are changes in the current applied to a laser diode. Such a change, as is known, may affect the values of some of its internal parameters, including the refractive index. A change in the refractive index alters the

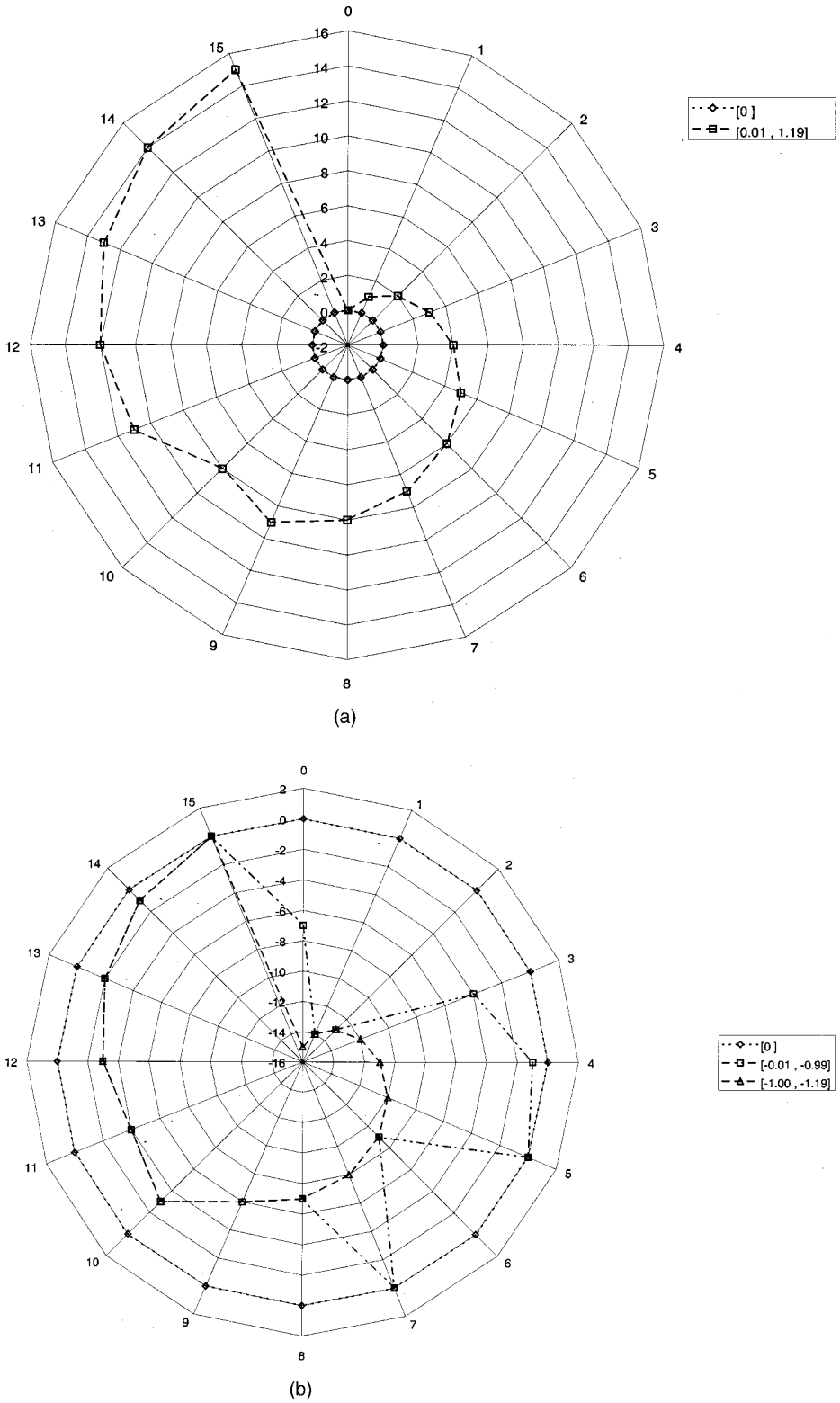
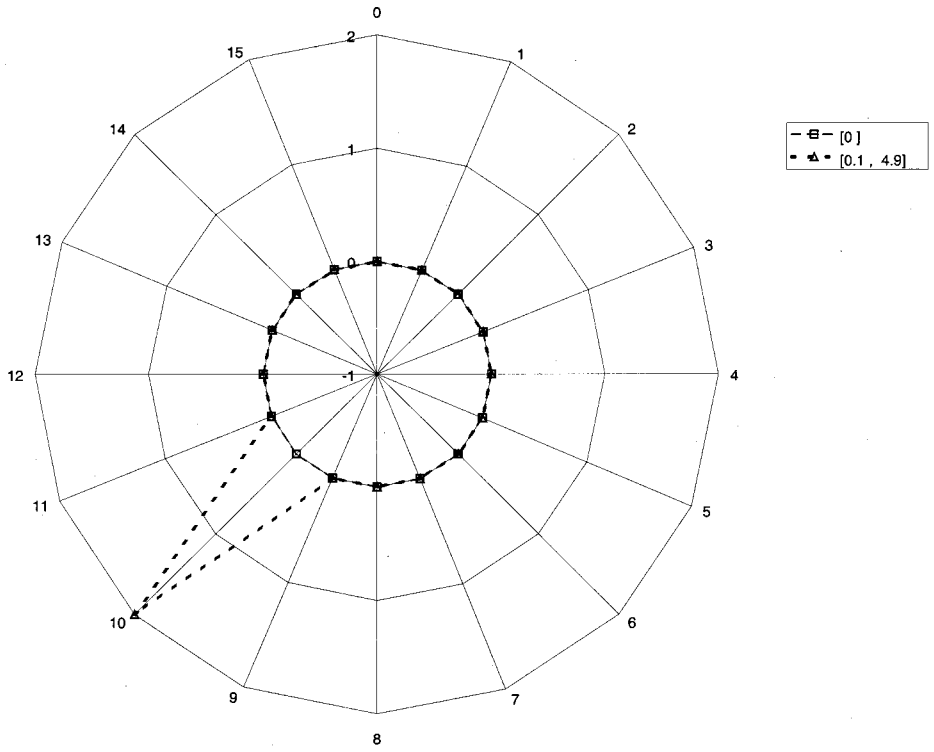
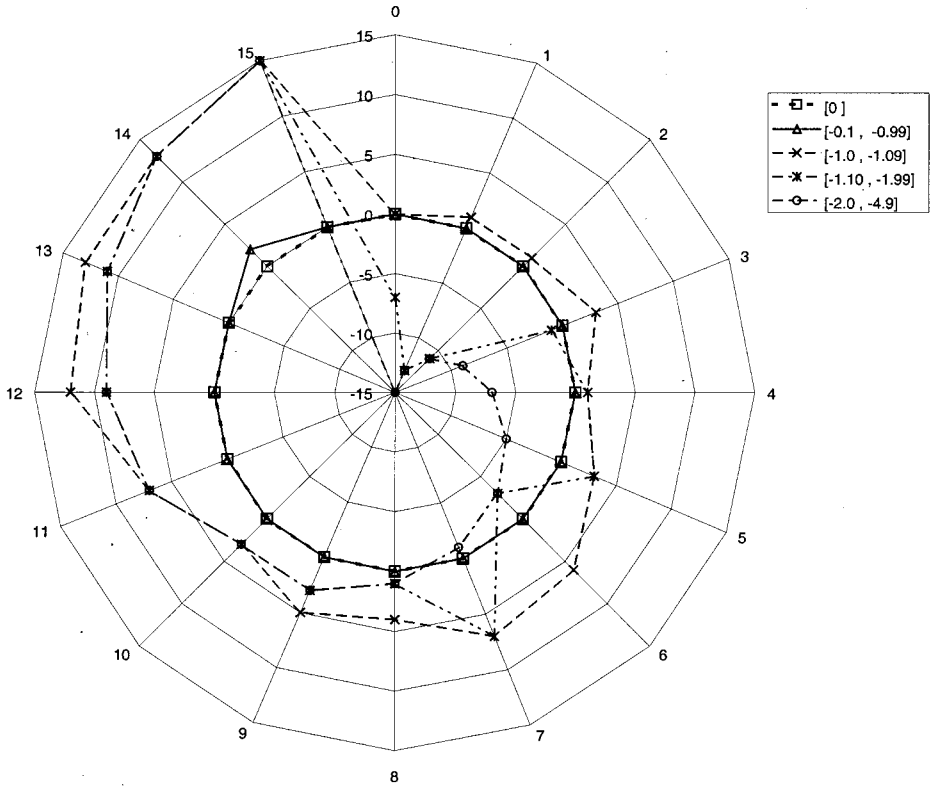


Fig. 13 Polar error representation of obtained results for the PS1 device. Same conventions as in Fig. 11.

resonant characteristics of the Fabry-Perot cavity, and, as a consequence, changes may occur in the wavelength of the laser radiation. These changes alter the whole behavior of the communications system based on that laser. Hence, the study of these variations is of great importance, and it may be conducted with a version of the method reported in this paper. Another possible case for analysis with this technique is the influence of external temperature on the lasing characteristics of a laser diode. In this situation, the increase of the threshold current with temperature is similar



(a)



(b)

Fig. 14 Polar error representation of obtained results for the PS2 device. Same conventions as in Fig. 11.

to the displacements presented previously for the SEED.

Moreover, the method presented in this paper may be useful for analyzing nonlinear behavior as when these types of devices are employed as chaos generators. The different regions that appear and the transitions between them give information about the characteristics of transition points. This fact deserves further analysis.

Acknowledgment

This work was partly supported by CICYT ''Comisión Interministerial de Ciencia y Tecnología,'' grant TIC99-1131, and CAM ''Comunidad Autónoma de Madrid,'' grant 07T/0037/2000.

References

1. D. A. B. Miller, D. S. Chemla, T. C. Damen, T. H. Wood, C. A. Burrus, A. C. Gossard, and W. Wiegmann, ''The quantum well self-electrooptic effect device: optoelectronic bistability and oscillation, and self-linearized modulation,'' *IEEE J. Quantum Electron.* **21**, 1462–1476 (1985). More details are given for example, in H. M. Gibbs, *Optical Bistability: Controlling Light with Light*, Academic Press, Orlando (1985), and P. Bhattacharya, *Semiconductor Optoelectronic Devices*, Prentice Hall (1997).
2. See, for example, M. P. Y. Desmulliez, F. A. P. Tooley, J. A. B. Dines, N. L. Grant, D. J. Baillie, B. S. Wherrett, P. W. Foulk, S. Ashcroft, and P. Black, ''Perfect-shuffle interconnected bitonic sorte: optoelectronic design,'' *Appl. Opt.* **34**, 5077–5090 (1995).
3. A. González-Marcos and J. A. Martín-Pereda, ''Digital chaotic output from an optically processing element,'' *Opt. Eng.* **35**, 525–535 (1996).
4. A. González-Marcos and J. A. Martín-Pereda, ''Analysis of irregular behaviour on an optical computing logic cell,'' *Opt. Laser Technol.* **32**, 457–466 (2000).
5. A. R. Omondi, *Computer Arithmetic Systems*, Prentice Hall, New York (1994).

Ana González-Marcos received her PhD from the Universidad Politécnica de Madrid in 1993. She has been an assistant professor in the Universidad de Alcalá de Henares and in the Universidad Alfonso X el Sabio. She is now an associate professor at the ETS Ingenieros de Telecomunicación, Universidad Politécnica de Madrid (UPM). She has been working on optical computing and optical interconnections. Her present research interest includes fractal applications in optical computing and optical computing architectures.

José A. Martín-Pereda has been a professor of photonic technology and optical communications at ETS Ingenieros de Telecomunicación, Universidad Politécnica de Madrid (UPM), since 1975. His special fields of interest include optical communications, optical computing, and computational neuroscience. He was a member of the Technical Program Committee on International Conferences (ECOC, IOOC) from 1984 to 1994. He was the conference chairman of ECOC'86. He is a member of the Spanish Academy of Engineering, a Senior Member of the IEEE, and a member of OSA, SPIE, and AAAS.