Appl. Phys. B 46, 111–120 (1988)

Applied Physics B Photophysics and Laser Chemistry

Digital Optical Computing

K.-H. Brenner

Physikalisches Institut, Universität, D-8520 Erlangen, Fed. Rep. Germany

Received 17 November 1987/Accepted 14 March 1988

Abstract. The motivation for a digital optical computer is based on the shortcomings inherent in of electronic computers. Optics has solutions to offer especially in interconnects. Devices based on nonlinear optical effects are still in the early stages of their development. Hence at an intermediate stage hybrid opto-electronic computers might emerge. Their architectures should make use of the special attributes of optics. A specific approach called symbolic substitution logic is outlined.

PACS: 42.30, 42.80, 85.60

Recent advances in the development of devices for optical switching have revitalized the idea of constructing computers that operate with light as the carrier of information. In the sixties this idea was already being pursued by various groups. The motivation at that time was the unique advantage of optical communication. Switching devices at that time were mostly based on the electro-optic effect, which is fast but requires a considerable amount of energy. Architecturally, optical computers were considered similar to electronic computers. One of the major reasons why optical computing was stopped at this time was that silicon technology was in a state of rapid growth, mostly due to miniaturization. Today, miniaturization is still an ongoing process, however not at the same rate it used to be a few years ago. Other important quality factors like computing speed have also increased but by far not at the rate at which miniaturization has progressed. Looking at the increase in speed over the years, one can detect a saturation effect. This becomes more clear if we look at the expense versus speed. In the low frequency domain up to 20 MHz prizes are low. Going beyond 100 MHz the expense grows more than linearly. A Cray-XMP computer running at 105 MHz clock cycle frequency is, according to benchmarks, only a factor of 15 times faster than an IBM AT personal computer but certainly more than a factor of thousand more expensive. A similar observation shows that there is a clear inverse relationship between device complexity and speed. Technology for fast transistors does exist. Emitter Coupled Logic (ECL) and GaAs transistors are able to run at clock frequencies up to 100 GHz corresponding to 10 ps switching time. Looking at ring oscillators, we only see switching times around 30 ps. Simple logic gates consisting of a few transistors operate in the 200 ps domain. Chips with several hundred transistors were demonstrated to run at 1 ns clock cycle time. Fast CPUs using several thousand transistors run at around 10 ns. So between a transistor and a CPU there is a drop of three orders of magnitude in speed. The reason why CPUs cannot run at the same speed as the individual transistors lies in the interconnections. A computer consists of logic gates and interconnections. Both determine the operation speed of the system. Todays computers are communication limited. The demand for faster computers is clearly present. There are still a lot of unsolved problems in science that are in principle computable but in practice incomputable because the computation on the fastest machine available would take longer than a PhD thesis. So in order to increase computing speed, it is not sufficient to look for faster transistors. One also has to improve interconnections.

The main limiting factors are:

i) Band Limitation. Every electronic wire can be considered as a complex low pass filter due to intrinsic R–L–C time constants. The maximum operating frequency is determined not only by the real part, the loss, but also by the imaginary part, the velocity dispersion. Digital signals are not sinusoids. They can be decomposed into several Fourier components that have to be transmitted with correct amplitude and phase in order to maintain the logic levels and the transients.

ii) Clock Skew. Up to now electronics is still in a domain, where propagation times in the system are small compared to the switching time of gates. For a 100 MHz clock, the free space propagation length is 3 m. In other words, if the total diameter of the computer is less that 3 m everything occurs quasi-simultaneously in the system, because latches are able to compensate minor path length differences. For a 10 GHz computer the diameter would be limited to 3 cm. The alternative to that would be to guarantee that all the path lengths in the system are equal with a precision better than roughly 3 mm.

iii) Cross Talk. An electronic wire without shielding can be considered as a Hertzian dipole. Such a dipole has radiative loss which results not only in amplitude loss, but also, because a dipole acts as an antenna, in cross talk. In case of the Hertzian dipole, the radiated energy grows with the fourth power of the frequency. Other sources of cross talk arise due to signal coupling via the power source. Switching many transistors simultaneously results in a voltage drop near these transistors. This effect can be prevented by placing charge reservoirs (capacitors) near these transistors. This method however restricts the integration density.

iv) Power consumption in terminating resistors:

Impedance matching is a critical factor in high speed logic design. In order to assure maximum power transfer through a wire, the output impedance of one stage has to match the input impedance of the next stage. Even at present speeds the power dissipated in terminating resistors exceeds the power required by switching.

The present approach to increasing computing power is parallel processing. The number of operations per second is then determined by the speed of one individual processor, by the total number of processors and by the degree to which a problem can be parallelized. The drawback of this method however is that parallel machines further increase the importance of interconnections because they require a higher interconnect density.

1. Optical Interconnections

Optical information transfer has a series of unique capabilities to offer: Propagation in waveguides provides an increase in bandwidth compared to electronic interconnections. This increase is however moderate compared to free space optical interconnections. Therefore we want to emphasize free space interconnections in this section. According to Fig. 1 they can be categorized with respect to regularity and fan-out. For all four categories shown in Fig. 1 the following properties are valid.

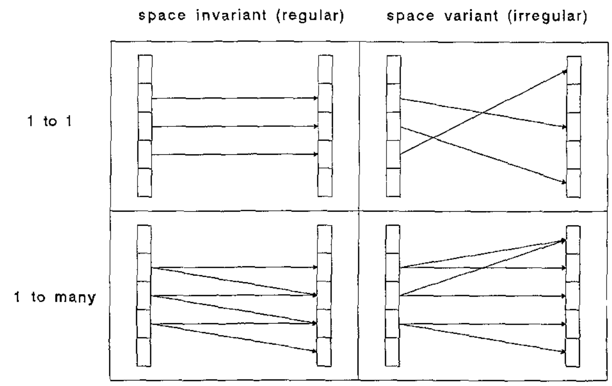


Fig. 1. Different categories of free-space interconnections

1.1. Bandwidth. In free space there is no dispersion. The bandwidth therefore is limited only by the carrier frequency. Thus communication up to 100 THz would be possible.

1.2. Noninterfering Propagation. Unlike electronic interconnections optical interconnections do not influence each other in free space. Many light rays can cross each other without interaction. For a fixed interconnect scheme, no wire routing is necessary as in electronics, where one wire occupies a certain volume that is forbidden for other wires. This is a direct consequence of the fact that photons do not interact in free space whereas electrons do.

1.3. Connectivity. An optical imaging system can be considered as an interconnection between the object plane and the image plane. If one spot in the object plane is switched, for example from dark to bright, the corresponding spot in the image plane will also change brightness. The total number of spots that can be communicated independently is determined by the numerical aperture of the optical system. An ordinary lens is able to interconnect 10^6 spots regularly or 10^3 spots irregularly.

1.4. No Contact Interconnects. Interconnections in free space do not require physical contact as electronic interconnections do. This is not only a reliability issue but it also has topological advantages. The space that is used for one interconnect can be reused by other interconnections.

1.5. Equidistance. Fermat's principle guarantees that the path lengths in a two-lens optical imaging system are equal. This is not only true for all the rays emanating from one spot but also for all spots. The degree of equidistance cannot only be measured with a precision less than a wavelength but it can also be maintained with standard positioning devices.

Realizing that electronic systems are communication limited and that optics has advantages to offer especially in signal communication, might lead to the conclusion, that electronic computers with optical interconnections is the right way to go. One would then use electrons to perform logic, because electrons interact, and photons for communication, where no interaction is desired. For the near future this certainly is a valuable approach. Optical interconnects on the systems level, on the board level and on the chip level are being researched now. For the longer term a third possibility is feasable. Photons do interact in nonlinear optical materials. Therefore photons can also be used for switching. Thus by placing nonlinear material where interaction is desired, and linear material where no interaction is desired, photons offer more flexibility than electrons [1].

2. Devices for Optical Computing

The intensification of research into optical computing was mostly initiated by the discovery of optical bistability [2, 3]. A bistable optical component shows two different transmission characteristics for the same input control value. Switching between these characteristics is obtained by changing the input power. One of the most frequently used components is the nonlinear Fabry Perot cavity. Depending on whether there is a saturable absorber in the cavity or a nonlinear refractive material one obtains either absorptive or dispersive bistability. More recent devices consist of a slab of nonlinear optical material which is coated with partially reflecting mirrors (Fig. 2). For dispersive nonlinear materials the refractive index depends on the intensity inside the cavity. The origin of this nonlinearity can be a shift of the band-gap energy in semiconductors like in GaAs or InSb or it can be of thermal origin as in ZnSe.

Bistability is a result of two mechanisms: nonlinearity and feedback. For a linear Fabry Perot cavity, the transmission is determined by the optical path length between the mirrors. In a nonlinear Fabry Perot cavity, the light intensity itself changes the refractive index and thus the optical path length. Depending on whether the device is initially transmitting or initially reflecting, an inverting or a noninverting switching characteristic can be obtained. Because of this internal feedback mechanism between optical path length and intensity in the cavity, these devices show a discontinuous behaviour when the input power exceeds a certain threshold. They also show hysteresis. If the input power is reduced again, the device remains in the second state until a certain (lower) threshold is reached.

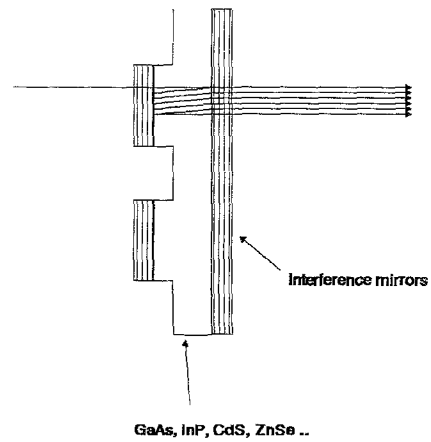


Fig. 2. A nonlinear Fabry-Perot cavity

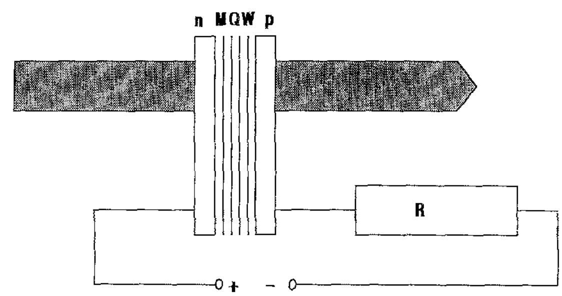


Fig. 3. Self-Electrooptic Effect Device

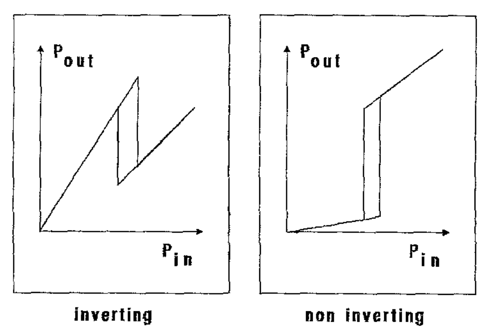


Fig. 4. Inverting and noninverting characteristics of bistable optical elements

A second class of bistable optical devices is known by the name of Self-Electrooptic Effect Devices (SEED) [4]. Here light is modulated by the absorption of the device. Through a mechanism called the quantum-confined Stark effect, the excitonic absorption peak in multiple quantum well structures shifts without broadening when an external electrical field is applied. In the SEED-configuration (Fig. 3) the absorbed light itself induces a current through the device and therefore a change of the E-field across the device. Depending on the wavelength of the input light, both inverting and noninverting characteristics can be obtained. Because of this electric feedback mechanism the SEED also shows hysteresis with similar input-output characteristics as the nonlinear Fabry Perot cavity (Fig. 4).

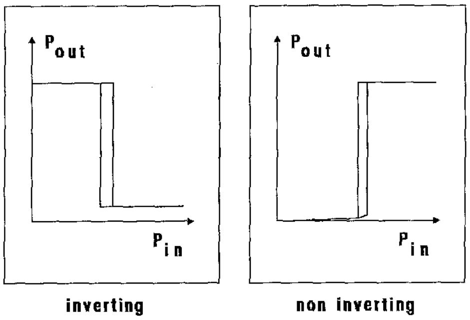


Fig. 5. Desired characteristics for an optical switch

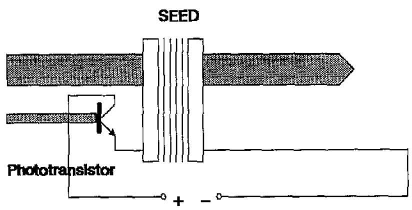


Fig. 6. A three-terminal optical device using a SEED and a phototransistor

An inverting device for example is operated by applying a bias beam with its power slightly below the switching threshold. An additional signal beam will switch the device to the nontransmitting state. If the signal beam is removed, the device returns to the transmitting state. Memory function can be obtained by choosing the power of the bias beam only slightly below the upswitching threshold but above the downswitching threshold. An external signal beam applied once will switch the device, but after removal of the signal beam the device will remain in this state.

Although optical bistability was a valuable driving force for optical computing, the input-output characteristics of bistable elements is not the most favourable from the system designer's point of view [5]. As shown in Fig. 5, a more desirable characteristic would show approximately constant levels around the switching point and an almost vanishing hysteresis loop. Such a characteristic allows larger fan-in devices, i.e. devices that can accept more inputs. Furthermore, it would show better output contrast and higher gain. These quantities determine the fan-out i.e. the maximum number of devices that can be driven by one device. Another point of consideration is that the method of physically adding the amplitudes of signal and power beam requires that the noise level on the power beam should not exceed in absolute terms the noise on the signal beam. However, in order to have gain, the power beam must be several times more intense than the signal beam. Therefore the relative noise level in the power beam must be smaller than the relative noise level in the signal beam. In electronics this problem was solved by going to three-terminal devices. Here the power beam is decoupled from the signal beam. The T-SEED [6,7] as shown in Fig. 6 is a step in this direction. Here the signal beam is detected by a phototransistor. This transistor modulates the absorption of the device and therefore the transmitted power beam.

3. Logic and Architectures for Digital Optical Computers

The early approaches to optical computing concerned ways to implement optical logic. Later, architectures based on this logic were suggested.

3.1. Optical Implementations of Logic

A simple mechanism for implementing optical logic was proposed by Bartelt and Lohmann [8]. The basic principle of this method is that binary states are coded by the direction of the light. Thus logic operations are possible by filtering in the Fourier plane. This kind of coding has been achieved with gratings [9], anisotropic scatterers, and with Wollaston prisms [10].

A different method for implementing logic was proposed by Tanida and Ichioka [11] and goes by the name of shadow casting. Here data were coded as spatial masks. A "one" in the first plane for example was coded as a square block where the upper half was opaque and the lower half transparent. A "zero" was coded the other way round. In the second plane, a "one" was coded as a square block where the left half was opaque and the right half transparent. When the two masks were placed behind each other at a certain distance, together with a decoding maks and illuminated by a point source, all the 16 possible binary logic operations could be obtained by changing the position of the illuminating light source.

Coding binary states by the two polarization states of light is an attractive alternative to intensity coding. Polarization logic based on anisotropic crystals as logic elements has been proposed by Watrasiewicz [12], based on liquid crystal light valves by Jenkins et al. [13], based on ferroelectric liquid crystals by Handschy et al. [14] and based on anisotropic Fabry-Perot Resonators by Korpel and Lohmann [15].

3.2. Optical Architectures

The fore runner of all computer architectures is the classical finite state machine. In this architecture logic and latches provide universal programmability. Also there is no bottleneck in data transport because all the latches are accessible through an individual interconnection. Conceptually, there is also no connectivity problem because the interconnections are hidden in the logic box. So in principle any output of the logic block may be a logic combination of all input elements. In a practical electronic realisation, this degree of connectivity is not available. In addition to that, with the von Neumann approach, the output lines no longer address the latch directly. A binary addressing scheme reduces the N necessary lines down to $\log_2 N$. An additional binary to 1 out of N decoding scheme is used. At the input it is no longer necessary to supply N individual inputs to the logic box because, due to the addressing scheme, only one latch is active at a time anyway. Thus it suffices to shunt all the outputs of the latch to one line. The input of the logic box is thus reduced to one line.

Present optical architectures can be categorized into parallel optical logic architectures, bit-serial architectures and symbolic substitution architectures.

Parallel optical logic approaches are designed more for special purpose applications like image processing and not so much for general purpose computers. They consist essentially of hardware for logically combining two arrays of binary information and of a fixed interconnection. The relatively slow processing time is assumed to be compensated by the high degree of parallelism.

In the OPALS (Optical Parallel Array Logic System) architecture [11] and also in the sequential optical processor [13], arrays of binary information are processed in parallel according to one logic operation. In the feedback path, a fixed optical interconnection serves to permute the data within the array.

Bit-serial architectures are conceptually similar to traditional electronic architectures, replacing electronic components by corresponding optical ones. These approaches mainly use the high temporal bandwidth of optical communication. A bit-serial computer [16] with directional couplers as switches, and optical fibres as interconnects, is currently being developed at the Center for Optoelectronic Computing Systems in Boulder, Colorado. At AT & T Bell Laboratories a concept for a fully integrated optical computer is being pursued [17]. The interconnections are made of waveguides on planar structures and of coupling gratings for vertical coupling between adjacent layers. The switch is a waveguide Fabry-Perot Resonator which is based on the QWEST effect.

3.3. Symbolic Substitution

Symbolic substitution [18, 19, 20] is an approach that uses both the temporal bandwidth and the high connectivity of optics to construct a general prupose programmable optical computer.

Symbolic Substitution Logic (SSL). Processing in SSL occurs on a space quantized plane or a matrix. Each

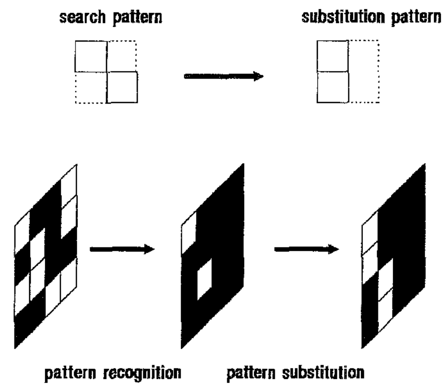


Fig. 7. Principle of symbolic substitution

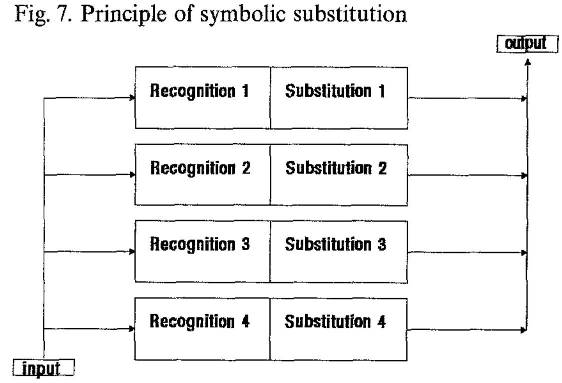
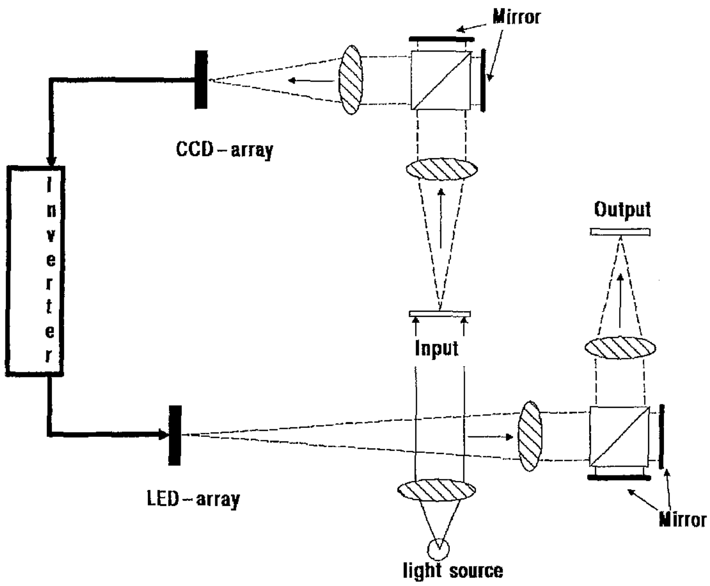
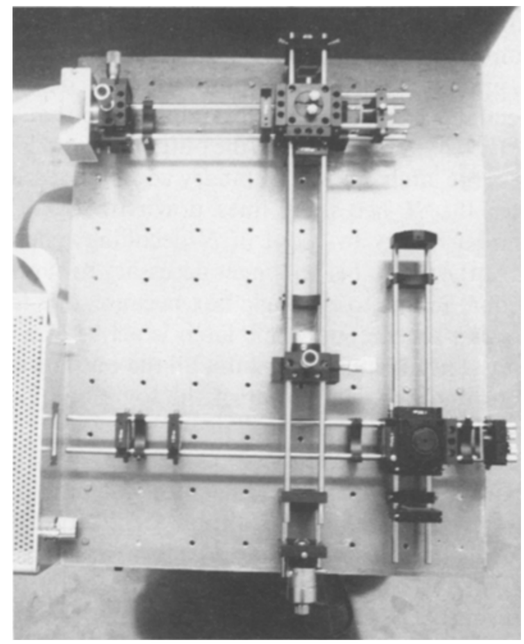


Fig. 8. Parallel execution of substitution rules in a transformation block

element can have a finite number of states. A convenient but not necessary choice is to work with binary states. The information can be encoded optically by intensity, phase, polarization, position or direction. In our example (Fig. 7) we use intensity coding. Logical One will be represented by light intensity and logical Zero by no light intensity. The elementarty operation in SSL is recognition and substitution of patterns. A pattern here is a spatial configuration of binary values. Thus a pattern could be represented by a vector of tuples $(s, \mathbf{d})_i$, $0 < i < M$ describing the state s and the corresponding relative position $\mathbf{d}$ of this state. The number of vector elements M is given by the number of specified cells in the pattern. The pattern is only defined relative to some arbitrary origin. The first goal in SSL is to find all the occurrences of a given pattern within a matrix. More precisely, the goal is to find those origins in the matrix where the states s_i in their corresponding relative directions $\mathbf{d}_i$ equal those of the search pattern. Only those locations will be marked by logical One in an intermediate plane. The final goal is to substitute these Ones with a new, different pattern. In detail, the task is to start from those locations, marked One and to scribe the new states s_k at the corresponding relative offsets $\mathbf{d}_k$. Figure 7 illustrates these operations. A binary matrix contains the search pattern, a white two-dot pattern at relative locations (1, 0) and (0, 1) on two different locations. The matrix in the middle shows the





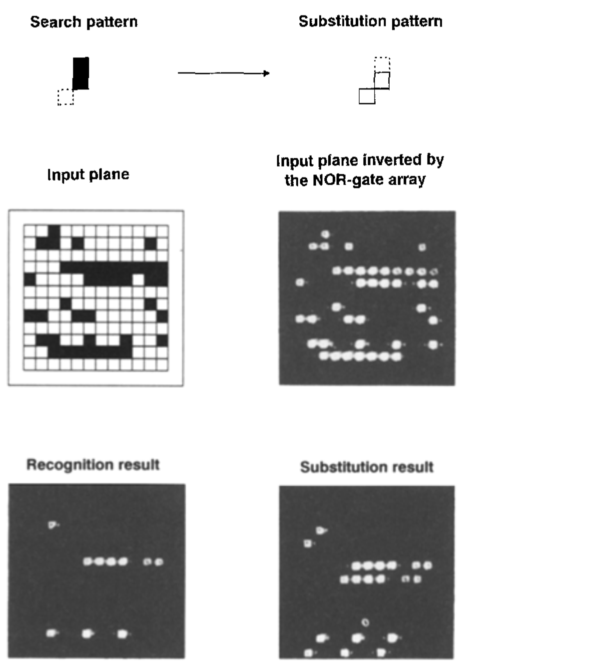


Fig. 9. (a) Setup to implement one substitution rule. (b) Optical implementation. (c) Experimental result. Top: the substitution rule, middle: the input plane, bottom: recognition and substitution results

recognized origins of these patterns. The matrix on the right-hand side shows how these patterns are replaced by the substitution pattern. In the example, a two-dot pattern consisting of two dots at relative positions (0, 0) and (0, 1) is used. One recognition-substitution operation is called a rule. Every rule consists of a left-hand side: the search pattern and a right-hand side: the substitution pattern. In general, more than one substitution rule is applied to a matrix simultaneously. As shown in Fig. 8, the input pattern is replicated by using beam splitters in order to provide several copies of the input plane to the recognition-substitution stages. We call a module performing several substitution operations in parallel a transformation block. For parallel substitutions, two situations are possible. The first is that no rule scribes a certain location. Thus a default state (no light) exists. The second case is that several rules scribe the same location. In this situation the physical implementation determines what happens. In intensity logic a logical One will overwrite a Zero, whereas in polarization logic this mixed state can be used as an "always match", i.e. a "don't care" state that can serve both as a One and as a Zero. Several specializations of SSL are possible. So it can be useful to allow recognitions not on every location but only on a predetermined raster. The elementary operations "pattern recognition" and "pattern substitution" have been chosen because they can be implemented optically in parallel. Figure 9a shows an optical implementation of a recognition-substitution system for intensity coding. Figure 9b shows a photograph of the setup. The input matrix is imaged onto an optoelectronic inverter array via a beam splitter and two mirrors that are tilted to produce the shifts. Therefore on the inverter array two shifted copies of the input matrix appear. The inverter array in this system could be an array of nonlinear Fabry-Perot cavities or an array of SEEDs. It serves two purposes. First it performs a logic NOR-function, yielding a logical One output only if all the inputs to this cell are Zero, and

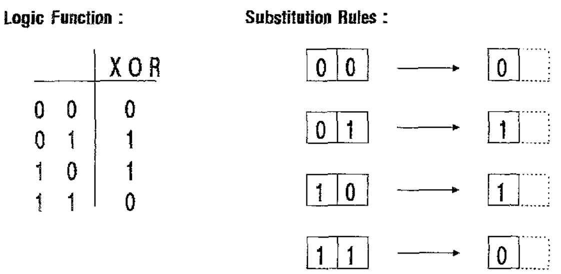


Fig. 10. Substitution rules for performing Boolean logic

second it restores the optical signal levels. The substitution part is similar to the recognition part. The intermediate image is again split into two branches. In each branch, a shift and its corresponding state are generated. In the case of patterns with more than two specified cells, holograms can be used instead of beam splitters [21]. Figure 9c shows an experimental result. The substitution rule is shown on top. A two-dot dark pattern is to be transformed into a diagonal two-dot bright pattern. The input plane contains the search pattern ten times. The intermediate plane, where the recognition result occurs, and the substitution result are shown at the bottom.

Logic with Symbolic Substitution. With symbolic substitution, logic is performed differently from conventional logic. In conventional logic a physical mechanism acts on the input quantities producing an output state. For example an electronic AND gate operates as a current source. If both inputs are high, no current will flow and a high level will be generated at the output. In SSL, logic is not based on an elementary physical mechanism. Here a logic function with two inputs is realized by four substitution rules. Figure 10 shows the rules for an Exclusive-OR function as an example. For two inputs there are exactly four different input combinations. For each of these combinations a substitution rule specifies the output. Thus each of the sixteen possible logic functions can be realized with the same hardware.

Optical Adder Using Symbolic Substitution. A ripple carry adder can be described by a logic function with two inputs and two outputs. One output generates the sum by using an Exclusive-OR function, the other performs the carry-generation using an AND function. As shown in Fig. 11 both functions can be combined in one rule. Figure 11 shows the four rules for the four different input combinations. If these rules are applied to an array of binary numbers, these numbers will be added in parallel. To this end, the substitution rules have to be applied m times, when m is the number of bits used for addition.

A Programmable Optical Processor. The basic operation in SSL is a space invariant pattern recognition

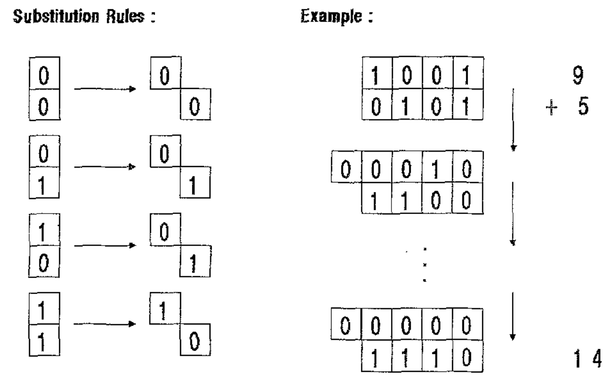


Fig. 11. Substitution rules for a ripple-carry adder

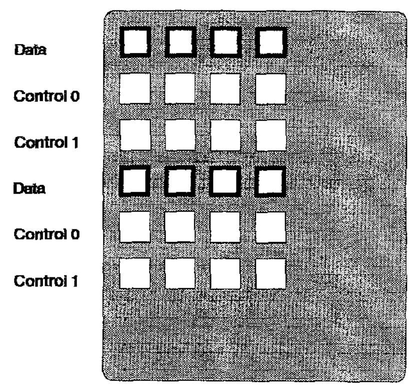


Fig. 12. Layout for the data plane

and substitution. In other words the same operation is performed independent of the position in the matrix. Such operations are useful where the problem is space invariant in nature, like in image processing or in solving partial differential equations with constant coefficients. For a universally programmable processor space dependent operations are also necessary because it should be possible to perform different operations in different regions of the array. Furthermore two types of operation are required: data transport within the array and logic operations between elements of the array. These operations should be programmable. In the following we want to construct a concept for a space variant programmable processor by using space invariant SSL. To this end a distinction between data and control is introduced. A possible layout for data and control within the matrix is shown in Fig. 12. We assume that the data are combined with control information before entering a transformation block.

Programmable Movement of Data. Figure 13 shows the substitution rules for horizontal data movement. The substitution rules are constructed so that the control bit combination (0, 0) results in no shift, the combi-

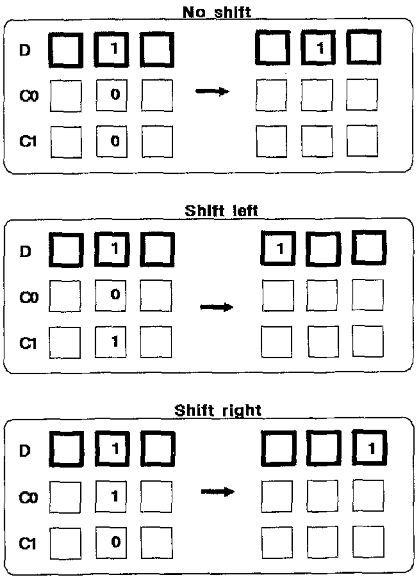


Fig. 13. Substitution rules for programmed data shifts

nation (0, 1) in a left shift and the combination (1, 0) in a right shift. For this operation six rules are necessary. Three of them are shown in Fig. 13. A similar block for vertical data movement can be constructed analogously. Then, in addition to left and right shifts, also up and down shifts are available.

Programmable Logic Operations. The minimal requirements for a functionally complete logic module are the identity operation and a complete set of boolean functions, i.e. a set of operators from which all the other ones can be constructed. Examples are OR and NOT

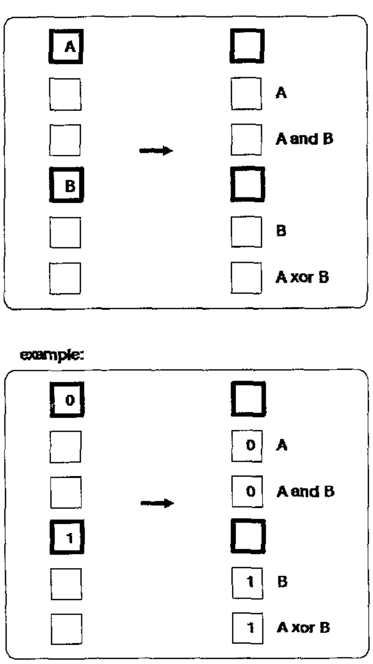


Fig. 14. Substitution rules for logic

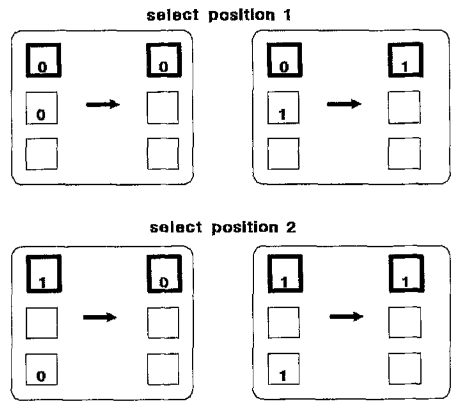


Fig. 15. Substitution rules for selection

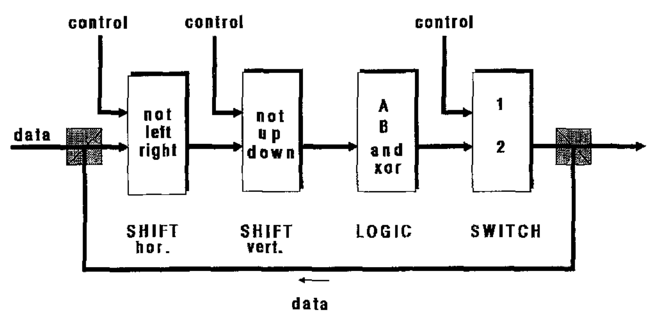


Fig. 16. A minimal optical processor

or also AND and Exclusive-OR. Using SSL it turns out to be convenient to divide the task of performing logic into two subtasks. First, from two input data we generate four different boolean functions: the identity operators for both inputs and the AND and Exclusive-OR combination of those inputs as shown in Fig. 14. Then using external control we select which of these four Boolean operators is used in the subsequent operations. The selection mechanism is achieved with four substitution rules as shown in Fig. 15.

A Minimal Programmable Optical Processor. The shift modules, the logic module and the switch module together form a minimal optical processor (Fig. 16) that supports all logic and connection primitives [22]. Incoming data enter from the left side. Temporary data circulate in the loop and can also exit at the right side. External control enters from the top. Figure 17 demonstrates three simple examples of operation. In Fig. 17a a shift of a horizontal register is shown. Vertical registers can be shifted similarly by applying shift control to the vertical shift module. In Fig. 17b a swap operation of two registers is shown. To this end one horizontal register is programmed to move up and the other horizontal register is programmed to move down. In Fig. 17c we perform a programmed binary addition of two horizontal registers. The data first

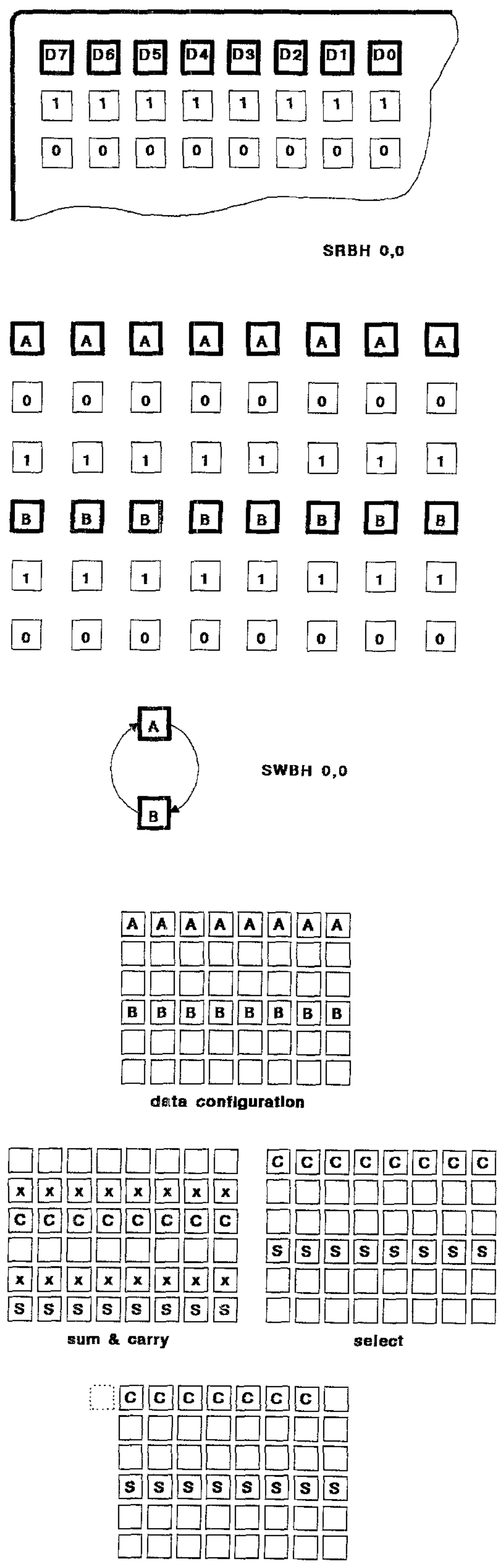


Fig. 17a–c. Examples for the operation of the minimal optical processor: (a) shift a whole register one position to the right, (b) exchange of two registers (swap), (c) addition of two registers

move through the shift modules without any shift. In the logic module, the AND and Exclusive-OR results are generated. The select module therefore selects position 2. Then the data cycle back to the input and enter the horizontal shift module, where the upper register is shifted one position to the left. This loop, iterated as many times as there are bits in the register, will generate the sum of those two registers.

An interesting feature of this processor is that the subdivision into registers is not determined by the hardware but rather by the external control. Therefore the layout and thus also the interconnections between individual cells can be changed at any cycle. Another aspect is the modular design. The smallest module, the recognition-substitution modul, is the basic building block. From this module transformation blocks are formed, and these in turn form a simple processor.

4. Conclusion

The use of optics in computing, as well as the transition to higher clock frequencies, require new and different architectures, better adapted to optical processing. In this paper we have demonstrated that optical computers do not have to be special purpose machines but that is quite possible to construct universally programmable machines. To actually build such machines, further progress is also needed in the area of device development.

Acknowledgement. The author would like to thank Adolf Lohmann for stimulating discussions and valuable comments.

References

1. A.W. Lohmann: Appl. Opt. **25**, 1543 (1986)
2. *Optical Bistability III*, ed. by H.M. Gibbs, P. Mandel, N. Peyghambarian, S.D. Smith, Springer Proc. Phys. Vol. 8 (Springer, Berlin, Heidelberg 1986)
3. "From Optical Bistability towards Optical Computing, The EJOB Project" (North-Holland, Amsterdam 1987)
4. D.A.B. Miller, D.S. Chemla et al.: IEEE J. QE-**21**, 1462 (1985)
5. M. Prize, N. Streibl, M. Downs: "Computational properties of nonlinear optical devices", in Topical meeting on photonic switching, Technical Digest, Lake Tahoe, March 87 Incline Village, Nevada
6. D.A.B. Miller: Private communication. See also D.A.B. Miller, this issue
7. P. Wheatley, M. Whitehead, J. Midwinter et al.: Opt. Lett. **12**, 784 (1987)
8. H. Bartelt, A.W. Lohmann: Appl. Opt. **22**, 2519 (1983)
9. J. Weigelt: Opt. Eng. **26**, 28 (1987)
10. A.W. Lohmann, J. Weigelt: Appl. Opt. **26**, 131 (1987)
11. J. Tanida, Y. Ichioka: Appl. Opt. **25**, 1565 (1986)
12. B.M. Watrasiewicz: Opt. Laser Technol. **7**, 213 (1975)

13. B.K. Jenkins, A.A. Sawchuk, T.C. Strand, R. Forchheimer, B.H. Soffer: Appl. Opt. **23**, 3455 (1984)
14. M.A. Handschy, K.M. Johnson, W.T. Cathey: Opt. Lett. **12**, 611 (1987)
15. A. Korpel, A.W. Lohmann: Appl. Opt. **25**, 1528 (1986)
16. H.F. Jordan: "Bit serial optical computer" in Topical Meeting on Optical Computing, Technical Digest Volume 11, Optical Society of America, March 1987 Incline Village, Nevada
17. L.C. West: IEEE Computer **20**, 34 (1987)
18. A. Huang: "Parallel algorithms for optical digital computers", Proc. of the 10th Intern. Optical Computing Conf. 1983, IEEE Catalog No.: 83 CH 1880-4, p. 13
19. K.-H. Brenner, A. Huang, N. Streibl: Appl. Opt. **25**, 3054 (1986)
20. K.-H. Brenner: Appl. Opt. **25**, 3061 (1986)
21. J.N. Mait, K.-H. Brenner: Appl. Opt. (submitted)
22. K.-H. Brenner: Appl. Opt. May 1, 1988 (in press)