

# Optical Arithmetic Using High-Radix Symbolic Substitution Rules\*

Kai Hwang and D. K. Panda

University of Southern California  
Department of Electrical Engineering-Systems  
Los Angeles, CA 90089-0781

**Abstract:** New optical representations and *symbolic substitution*(SS) rules are presented for performing high-radix arithmetic in optics. A set of SS rules are proposed for high-radix optical arithmetic, which satisfies the arithmetic completeness property. Trade-off parameters like representational efficiency, projected speed gain, and estimated implementation cost are analyzed. The SS mechanism together with the signed-digit(SD) representation reinforces massive parallelism in optics. A digit-plane architecture, blending very well with the SS technique and SD representation, is considered for implementing high-radix arithmetic. An *optical adder*, exploiting massive parallelism, is proposed. The set of SS rules and their implementations on a digit-plane architecture provide the basis to achieve pipelining, systolization, and on-line arithmetic in future optical computers.

## 1 Introduction

Optics has emerged as a promising technology for eliminating the performance bottleneck of electronic computing systems. The inherent parallelism, high speed, non interfering communication, and wide bandwidth of optics provide a natural platform to perform high speed arithmetic computations [6,7]. Optical arithmetic computations, promising a significant improvement in computing power over the present day electronic computers, show hope for powerful digital optical computers in the future. At this juncture of *arithmetic with optics*, the field of *optical arithmetic* needs attention from the *computer arithmetic* research community for the successful realization of digital optical computers.

Recently, several researchers have demonstrated arithmetic computations in optics using different number systems. The *symbolic substitution*(SS) technique proposed in [3,8] has been adopted in the literature to carry out arithmetic in optics. The *signed-digit*(SD) number representation, originally proposed by Avizienis [2], was introduced to optics by Drake et al [5]. A number of symbolic substitution

algorithms for optical arithmetic computations using SD representations have been recently proposed [10,11,12,13]. Residue number systems for performing optical arithmetic have also been studied [4]. Optical parallel array logic systems(OPALS) to carry out arithmetic logic operations based on SS technique have been proposed in [14,15].

In this paper, we investigate the feasibility of performing high-radix arithmetic in optics. We present optical representations for high-radix SD numbers. We demonstrate that  $2(r+1)^2 + (r+5)$  symbolic substitution rules are sufficient to carry out radix- $r$  addition, subtraction, multiplication, division, and negation. Radix-4 arithmetic rules are illustrated as an example. We analyze different high-radix implementation in optics from the point of view of *representational efficiency*, *projected speed gain*, and *estimated implementation cost*. We consider a digit-plane architecture for implementing high-radix SD optical arithmetic. An *optical adder* based on a digit-plane architecture is presented.

The rest of the paper is organized as follows. We first represent high-radix signed-digit numbers in optics. Then, we demonstrate the set of SS rules exhibiting sufficiency in radix- $r$  optical arithmetic. Different trade-off parameters are analyzed to choose the suitable high-radix for optical arithmetic. The digit-plane architecture is presented together with the optical adder. Finally, we project the potential of high-radix SD optical arithmetic.

## 2 Optical Number Representation

High-radix arithmetic in electronic computers [1] demonstrate faster execution time compared to the binary case. Several studies are available in the literature [16] for high-radix arithmetic. In general, a string of  $l$  bits is equivalent to  $m$  radix- $r$  digits, where  $m = \frac{l}{\lceil \log_2 r \rceil}$ . For practical implementations,  $r = 2^k$  for some integer  $k > 0$  and  $l = m \times k$  [9]. The signed-digit(SD) number system with its inherent redundancy is ideal for implementing high-speed parallel arithmetic [2,9]. In a radix- $r$  SD number system, the digit set  $\Sigma_r$  assumes  $2\alpha + 1$  values, i.e.  $\Sigma_r = \{-\alpha, \dots, -1, 0, 1, \dots, \alpha\}$ , where  $\lceil \frac{r-1}{2} \rceil \leq \alpha \leq r-1$ . The absolute lower and upper limits of  $\alpha$  result in minimal and maximal redundancy digit sets respectively.

\*This work was supported in part by an AT&T grant on *Optical multicomputer networks for balanced parallel processing*, 7/1/1988 - 6/30/1990.

The two-level intensity (dark and bright) of light provides a natural mechanism for encoding binary data in optical digital computers. High-radix digit sets can be easily represented by grouping few pixels together. Both the minimal and maximal digit sets of radix- $r$  SD number system can be represented by a group of  $\lceil \log_2 r \rceil + 1$  pixels as shown in figure 1. Radix-4(8) digit sets require 3(4) pixels respectively. As shown in figures 1 (b) and 1 (c), the group of pixels (4 for radix-8) can have vertical/square organization. This alternative representation leads to flexibility in implementation. The combination of all pixels being bright can represent the don't care (X) state. Alternatively, it can be reserved for *radix point* separating the fraction portion of a number from the integer part.

Without any loss of generalization, in this paper, we assume a vertical representation. Any  $m \times n$  matrix of single-digit radix- $r$  numbers can be represented as a matrix of  $(m(\lceil \log_2 r \rceil + 1) \times n)$  dark and bright pixels in optics. Figure 2 (a) shows an example of radix-4 representation. For  $k$ -digit SD elements, the matrix can be separated into  $k$  digit planes with attached weights  $r^{k-1}, r^{k-2}, \dots, r^0$ . Figure 2 (b) shows the optical representation of a  $2 \times 2$  matrix consisting of 2-digit SD numbers.

### 3 High-Radix Symbolic Substitution Rules

In this section, we emphasize on the concept of symbolic substitution (SS) rules and its implementation in optics to carry out arithmetic computations. We derive the basic set of SS rules for radix- $r$  arithmetic and show the completeness property of this set of rules.

#### 3.1 Basic Concept of Symbolic Substitution

The SS technique proposed by Huang [3,7] forms a mathematical basis for several of the optical computing systems being studied today. Since the data is represented as spatial patterns (spatial arrangement of the digit-set as shown in the last section) of bright and dark pixels, mathematical operators such as ADD, SUBTRACT, AND, NOT etc can be represented as substitution rules specifying how to manipulate these patterns. The symbolic substitution (SS) can be defined [12] as a 3-tuple set  $S=(R, P, Q)$ , where  $R$  represents a one-to-one mapping from the domain  $P$  to the range  $Q$ ,  $R:P \rightarrow Q$ .  $R(r_1, r_2, \dots, r_n)$  is the set of SS rules, where  $n$  is the total number of rules.  $P$  is the set of search patterns  $P=(P_1, P_2, \dots, P_n)$  and  $Q$  is the set of replacement patterns  $Q=(Q_1, Q_2, \dots, Q_n)$ . Each SS rule,  $r_i, i = 1, 2, \dots, n$ , can be defined as follows:

$$r_i(P_j) = \begin{cases} Q_j & \text{if } i = j \\ \phi & \text{if } i \neq j \end{cases}$$

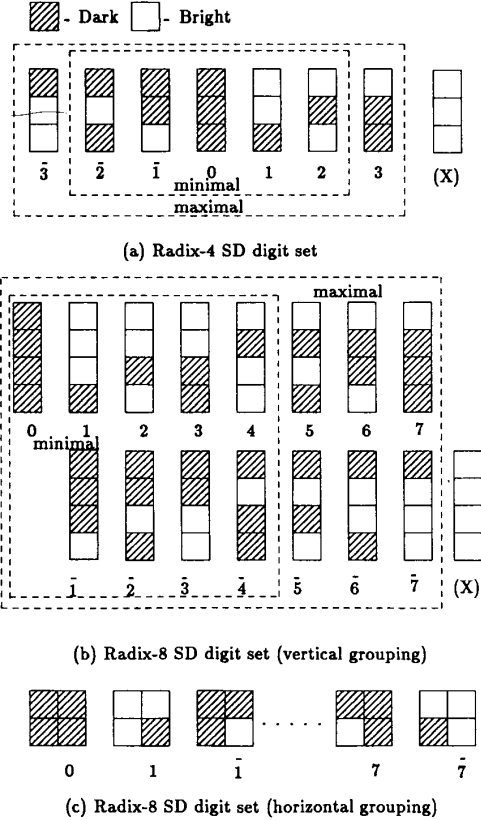


Figure 1: Optical representations of radix-4 and radix-8 SD digit sets.

where,  $\phi$  represents an empty pattern (a number of dark pixels).

Figure 3 (a) shows an example of a SS rule. Here, the mapping  $R$  represents the addition of two digits (2 and 2 in radix-4 arithmetic) generating sum and carry digits 0 and 1 respectively. The LHS of the rule is the search pattern in the input and the RHS of the rule is the replacement pattern. Figure 3 (b) demonstrates the application of this SS rule to an input plane generating the corresponding output plane.

The SS rules for an arithmetic operator are derived from the truth table specifications of that operator over the digit set. We examine this aspect of SS in the next section. The arithmetic operation in optics is carried out by applying all the related rules to the input data in a parallel manner as shown in figure 4. The input data plane is replicated and passed through the SS rule units. In the first phase of symbolic substitution, each SS rule unit tries to recognize its LHS pattern of symbols in the input plane. Only, in case of a match, the unit substitutes the LHS pattern by its RHS pattern. The output planes from different

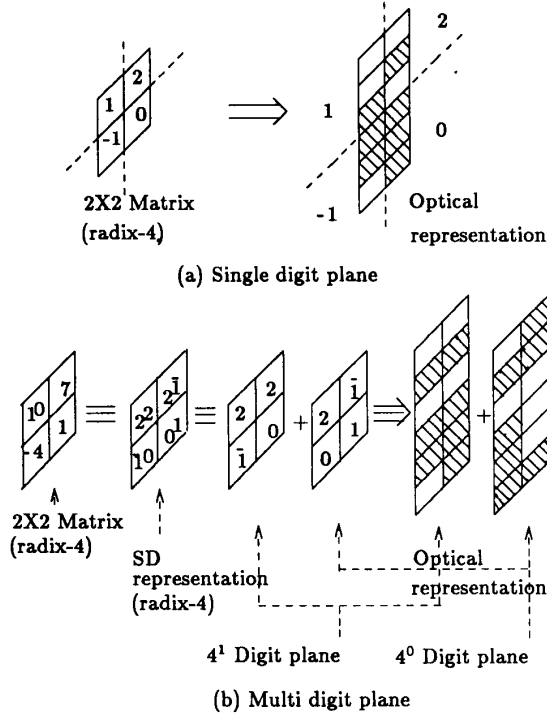


Figure 2: Optical representations for single and multi-digit planes.

SS rule units are combined (overlapped) together generating the output data plane. For detailed implementation of symbolic substitution rules, the readers are referred to [3,12].

### 3.2 Rules for High-Radix Arithmetic

As suggested earlier, in this section, we derive the high-radix SS rules from the truth table specifications of the operators NOT, ADD, and MULT. We consider the three fundamental operators NOT, ADD, and MULT. Considering the division by repeated multiplication (convergence division) [9], the above three basic arithmetic operators span the set of arithmetic operations.

We consider the minimal redundancy digit set for high-radix modified SD arithmetic. In a radix- $r$  SD number system, the minimal redundancy digit set  $\sum_{r(min)}$  consists of  $r+1$  values [2]. With a minimal redundancy representation, addition/subtraction of two SD numbers is possible in two steps by allowing the propagation of the transfer digit over two digital positions to the left. This classical *two-transfer addition* scheme, proposed by Avizienis, requires only three types of adder blocks (Type I, II, and III). The truth table corresponding to type-I radix- $r$  adder block ( $x_i + y_i = rt'_{i+1} + w'_i$ ,  $|t'_{i+1}| = 1$  if  $|x_i + y_i| \geq \frac{1}{2}r$ ,

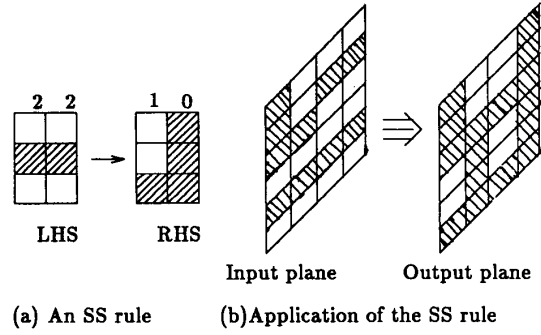


Figure 3: The basic concept of symbolic substitution.

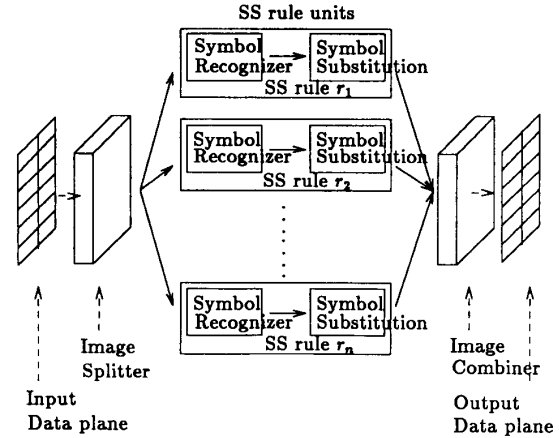


Figure 4: Optical implementation of SS rules.

$x_i, y_i, t'_{i+1}, w'_i \in \sum_{r(min)}$ ) is of size  $(r+1) \times (r+1)$ . The table represents all possible combinations of  $x_i$  and  $y_i$  over the digit set consisting of  $(r+1)$  digits. Each of these combinations can be treated as a symbolic substitution rule from the  $\{x_i \times y_i\}$  domain to the  $\{t'_{i+1} \times w'_i\}$  domain. This contributes  $(r+1)^2$  SS rules for type-I adder block.

The type-II adder block ( $w'_i + t'_i = rt''_{i+1} + w''_i$ ,  $|t''_{i+1}| = 1$  if  $|w'_i + t'_i| > \frac{1}{2}$ ,  $w'_i, t''_{i+1}, w''_i \in \sum_{r(min)}$  and  $t'_i \in \{\bar{1}, 0, 1\}$ ) can be represented by a truth table of size  $3 \times (r+1)$ . Similarly, the type-III adder block ( $s'_i = w''_i + t'_i$ ,  $w''_i \in \sum_{r(min)}$ ,  $t'_i \in \{\bar{1}, 0, 1\}$ ) also represents a truth table of size  $3 \times (r+1)$ . The only difference between the type-I and type-II adder is the carry generation logic (carry=1 in type-II adder if  $|w'_i + t'_i| > \frac{1}{2}$ , where as carry=1 in type-I adder if  $|x_i + y_i| \geq \frac{1}{2}$ ). Based on this logic, only 4 entries in the truth table of type-II adder differs from that of type-I adder. This requires 4 additional SS rules. By incorporating default 0's as the carry output in the truth table of type-III adder, the truth table becomes identical to that of type-II adder. Hence,  $(r+1)^2 + 4$  SS rules are sufficient to carry out radix- $r$  SD additions. The truth tables corresponding to type-I, II, and III adders for radix-4 arithmetic are shown in figure 5.

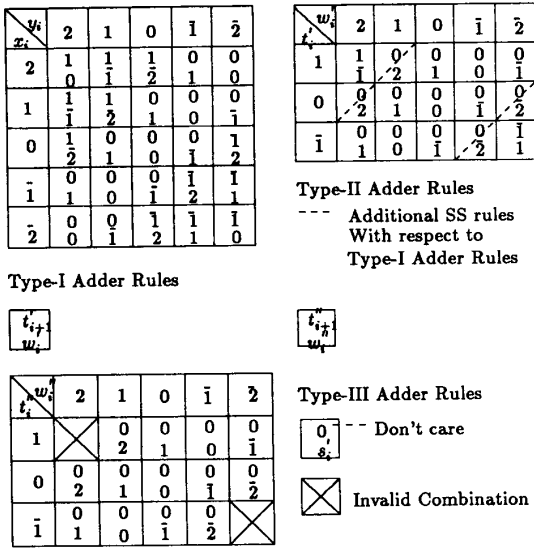
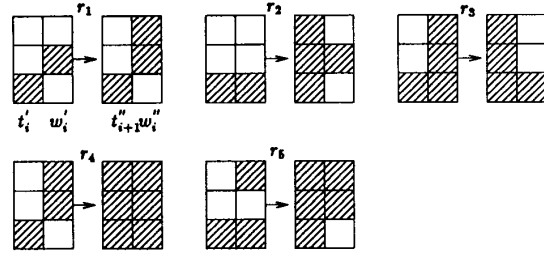


Figure 5: Truth tables for radix-4 adder blocks.

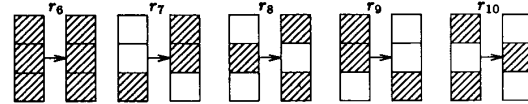
The SS rules corresponding to the first row of type-II adder rules for radix-4 arithmetic are shown in figure 6 (a). Without any loss of generality, it is assumed that the two input digits are represented by two consecutive vertical columns in the input plane. Similar assumption also holds good for carry and sum digits in the output plane. The organization of input and output digits in SS rules can be modified to match the implementation scheme.

The negation(NOT) operation is quite straight forward in optics. It can be achieved by complementing(bright replaced by dark and vice versa) the intensity of input pixels. Radix- $r$  SD digit set comprises of  $(r + 1)$  digits. Hence,  $(r + 1)$  SS rules as shown in figure 6 (b), are required to implement the negation(NOT) operation. Fast multiplication of two numbers  $X = x_{n-1}, x_{n-2}, \dots, x_0$  and  $Y = y_{n-1}, y_{n-2}, \dots, y_0$  is achieved by generating the partial products  $p_i = y_i \times X$ ,  $(0 \leq i \leq n - 1)$  and getting the weighted sum of the partial products [9]. The weighted sum is carried out either by shift-and-add or by binary tree addition. For binary numbers, the partial product  $p_i$  is generated in one step by bit-wise AND operation of the multiplier bit  $y_i$  with the multiplicand X. For high-radix SD numbers, the partial product generation can not be achieved in one step. It can be easily verified that adder circuits similar to *two-transfer addition* are essential for generating partial products for high-radix SD multiplication.

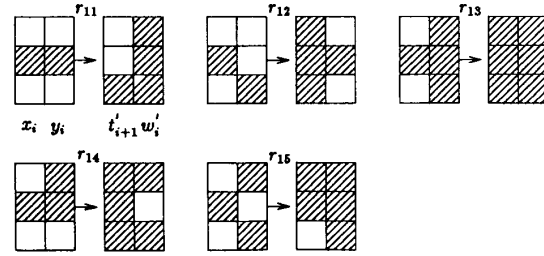
We demonstrate such an adder circuit in figure 7, where the conventional type-I adder blocks of *two-transfer adder* are replaced by type-IV multiplication(Mult) blocks. The Mult blocks perform digit-wise multiplication ( $2 \times 2 = 10, \bar{2} \times 1 = 0 \bar{2}$  in radix-4 arithmetic) generating sum and



(a) Few of the SS rules for radix-4 SD addition



(b) SS rules for radix-4 SD NOT operation



(c) Few of the SS rules for radix-4 SD Multiplication

Figure 6: Optical SS rules for radix-4 arithmetic.

carry. The next two layers are identical to the adder circuits discussed earlier. The truth table corresponding to type-IV Mult block ( $x_i \times y_i = rt'_{i+1} + w_i, |t'_{i+1}| = 1$  whenever  $|x_i \times y_i| \geq \frac{1}{2}r$ ) is of size  $(r + 1) \times (r + 1)$ . This requires additional  $(r + 1)^2$  SS rules to incorporate multiplication in radix- $r$  arithmetic. Figure 8 shows the truth table for radix-4 Mult block. The SS rules corresponding to the first row of the type-IV Mult truth table are shown in figure 6 (c).

The mentioned SS rules corresponding to addition (type I, II, and III adder blocks), negation(NOT), and multiplication(type IV Mult block) form the basis for all arithmetic operations. Hence, these  $2(r + 1)^2 + (r + 5)$  SS rules are sufficient for radix- $r$  modified signed-digit arithmetic in optics.

## 4 Cost-Performance Analysis

In this section, we consider different trade-off parameters like *representational efficiency*, *estimated implementation cost*, and *projected speed gain* for different high-radix arithmetic in optics. The *cost* and *speed gain* parameters are analyzed on a relative scale (with respect to binary arithmetic) to compare the merits and demerits of high-radix

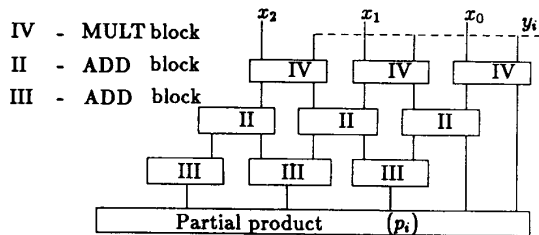


Figure 7: Partial product generation circuit for high-radix multiplication.

$y_i \backslash x_i$	2	1	0	$\bar{1}$	$\bar{2}$
2	1 0	0 2	0 0	0 2	$\bar{1}$ 0
1	0 2	0 1	0 0	0 $\bar{1}$	0 2
0	0 0	0 0	0 0	0 0	0 0
$\bar{1}$	0 2	0 $\bar{1}$	0 0	0 1	0 2
$\bar{2}$	$\bar{1}$ 0	0 2	0 0	0 2	1 0

Type-IV Mult Rules

$$\begin{matrix} t_{i+1} \\ w_i \end{matrix}$$

Figure 8: Truth tables for radix-4 Mult block.

arithmetic. Some of these parameters result in conflicting requirements for a suitable radix usage in optical arithmetic. Hence, the optimal choice of a suitable radix for an optical digital computer has to depend on several other factors like technology, application, and overall cost.

As we discussed in section 2, high-radix digit sets are represented in optics by grouping few pixels together. Both the minimal and maximal digit sets of radix- $r$  SD number system are represented by a group of  $\lceil \log_2 r + 1 \rceil$  pixels. Though, these pixels result in  $2^{\lceil \log_2 r + 1 \rceil + 1}$  combinations, few of the combinations are used depending on the radix and redundancy (maximal or minimal). On this basis, we analyze the representational efficiency (useful number of combinations/total number of combinations) for different radix implementations. This efficiency parameter is directly related to the storage efficiency for data in optics. Figure 9 demonstrates this efficiency for different radix representations. From the graph, it is evident that maximal redundancy digit set is efficient for high-radix arithmetic in optics. But, lower radix implementation using minimal redundancy set also promises comparable storage efficiency. It can be also observed that radix-10 representation results in very poor efficiency both for maximal and minimal redundancy. Hence, radix-10 arithmetic is quite inefficient from representational (storage) point of view in optics.

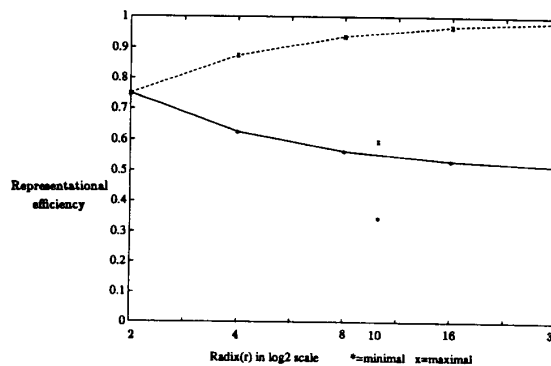


Figure 9: Representational efficiency vs radix.

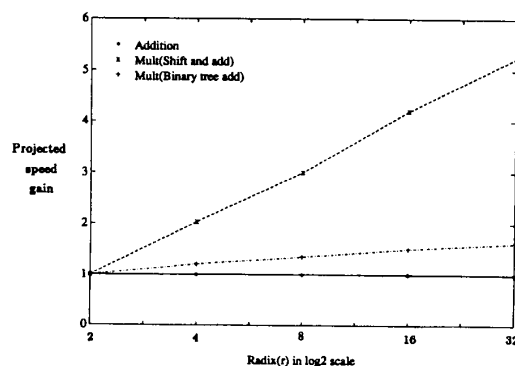


Figure 10: Projected speed gain over binary arithmetic vs radix.

Next we analyze the projected speed gain of high-radix arithmetic over binary. Figure 10 shows the trade-off of the speed gain. We have assumed the numbers to be 64-bit (binary) for our comparison. Since we use modified SD arithmetic, the addition time is independent of the radix. Radix- $r$  multiplication of two  $n$ -digit numbers based on shift-and-add scheme results in a speed gain of  $\frac{n-1}{\lceil \log_2 r \rceil - 1}$  compared to binary multiplication. This gives almost a  $\log_2 r$  speed-up. At the expense of more hardware, the multiplication of two numbers can also be carried over by generating the partial products and adding them in a binary tree fashion. Using radix- $r$  representation, this scheme provides a speed gain of  $\frac{\log_2 n}{\lceil \log_2 \frac{n}{\log_2 r} \rceil}$ , which is considerably smaller compared to the shift-and-add scheme. Hence, high-radix multiplication (shift-and-add) is favorable for optical arithmetic. It is to be noted that, the speed gain, as shown in figure 10, represents the gain by considering arithmetic over two numbers. But, optics together with a digit-plane architecture (discussed in the next section), has the potential to perform arithmetic over arrays of data in parallel. Under these circumstances, the speed gain with high-radix arithmetic will be significantly high.

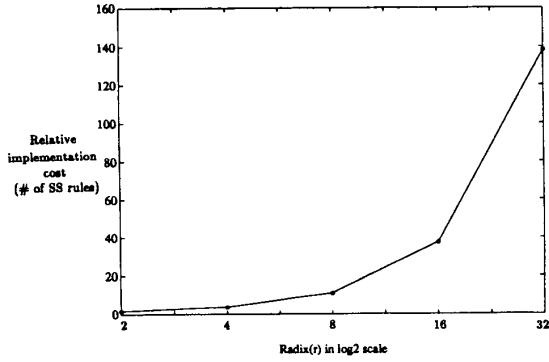


Figure 11: Estimated implementation(relative) cost vs radix.

Since symbolic substitution units are the basic building blocks for optical arithmetic, we consider the cost of an optical computer directly proportional to the number of SS rules. Since, the technology for optical computers are not matured, we consider the relative cost i.e. the ratio of the costs of high-radix arithmetic to binary. Figure 11 shows the relative implementation cost as a function of radix. The cost being quadratic as a function of radix, does not appeal much for high-radix optical arithmetic at the first sight. As we mentioned earlier, optics has the potential for parallel arithmetic over arrays of numbers giving rise to significant speed gain for high-radix arithmetic. Hence, the increase in cost associated with high-radix arithmetic is justified. Trading-off the speed-gain and the high cost, it can be easily observed that high-radix arithmetic (radix-4, 8, and 16) will win over binary in optics.

## 5 An Optical Arithmetic Processor

In this section, we consider a digit plane architecture for efficient implementation of high-radix optical arithmetic computations. Similar to the bit-plane architecture [12], a digit plane architecture is proposed for high-radix arithmetic to exploit the natural data-parallelism inherent in data-intensive applications. Such applications include matrix computations and image processing.

As we discussed in section 2 (figure 2), an array of data can be easily represented in optics by multiple digit-planes. The array of data can be stored/retrieved to/from optical memory as digit-planes. The processing (arithmetic operations) can also be done in a digit-plane manner. We explain this concept by demonstrating the architecture of an optical adder. The SD arithmetic, as we discussed in section 3, provides additional parallelism to the digit-plane architecture in optics.

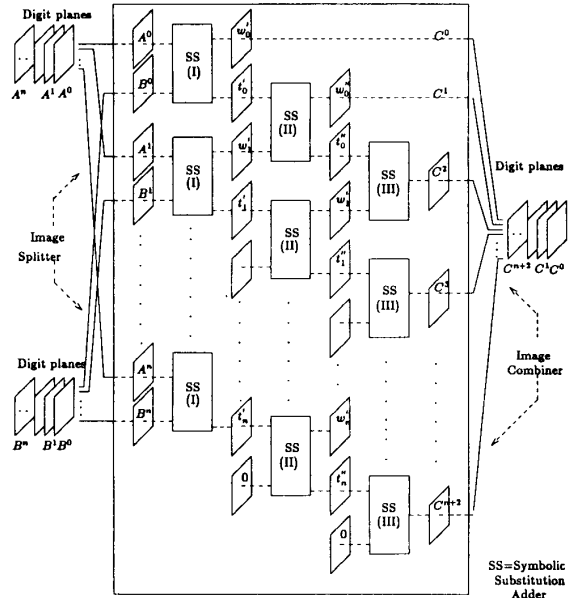


Figure 12: Optical signed digit adder using digit-plane architecture.

The proposed adder as shown in figure 12 is based on a digit-plane architecture, symbolic substitution, and SD addition. The adder adds two input data arrays (A, B) and produces the output data array (C). For high-radix arithmetic, these data arrays can be stored in memory as digit-planes ( $A^n, \dots, A^0; B^n, \dots, B^0$ ) and ( $C^{n+2}, \dots, C^0$ ). This optical adder implements SD addition and is similar to the *two-transfer adder* in [2]. The input digit-planes of A and B are separated by the image splitter so that digit-planes with similar weights ( $(A^0, B^0), (A^1, B^1), \dots, (A^n, B^n)$ ) are fed to the same SS unit. The three layers of SS units implement the SS rules corresponding to type-I, II, and III adders discussed in section 3.2. In the first step, the rules of SS-I units are applied to the input digit-planes generating interim sum digit-planes  $\{w'_0, w'_1, \dots, w'_n\}$  and carry digit-planes  $\{t'_0, t'_1, \dots, t'_n\}$ . Similar operations are carried over in the second and third steps generating the digit-planes ( $C^{n+2}, \dots, C^0$ ). Finally, the image combiner combines these planes to output the result data array C.

This optical adder demonstrates massive parallelism in optical arithmetic. Different levels of parallelism in arithmetic computations exhibited by the adder can be summarized as follows:

1. Parallel arithmetic on an array of data using optics.
2. Parallel application of SS rules within an SS unit to perform arithmetic.

### 3. Parallel operations on digit-planes using high-radix.

The digit-plane architecture provides a basic framework for high-radix arithmetic in optics. Since optical arithmetic is carried out by SS rules on an input data plane, the digit-plane architecture can be used to achieve *pipelining* in optical arithmetic. This broadens the scope of optical arithmetic to *pipelining*, *systolization*, and *on-line arithmetic*.

## 6 Conclusions

In this paper, we presented optical representations for high-radix redundant signed-digit code. We proposed a set of  $O(r^2)$  symbolic substitution rules to carry out radix- $r$  optical arithmetic. We analyzed different trade-off parameters like *representational efficiency*, *estimated implementation cost*, and *projected speed gain* for carrying out high-radix arithmetic in optics. High-radix optical arithmetic provides speed gain over binary arithmetic at the expense of higher cost (number of symbolic substitution rules) and lower representational (data storage) efficiency. But, the speed gain becomes multi-folded by performing parallel arithmetic on arrays of data. This parallelism was shown to be achievable by using digit-plane architecture and signed-digit redundant code. This multi-folded gain balances out the higher cost associated with high-radix optical arithmetic and forces high-radix (radix 4, 8, and 16) arithmetic to win over binary arithmetic in optics. We presented the architecture of an optical adder and demonstrated the feasibility of digit-plane architecture providing a basic framework for high-radix arithmetic in optics. The digit-plane architecture combined with symbolic substitution and signed-digit code has potential to incorporate pipelining, systolization, and on-line arithmetic in optics. This promises a significant improvement in computing power of the optical digital computers over the present day electronic counterparts.

## References

- [1] D. E. Atkins. Design of the arithmetic units of ILLIAC III: use of redundancy and higher radix methods. *IEEE Transaction on Computers*, C-19(18):720–733, 1970.
- [2] A. Avizienis. Signed-digit number representations for fast parallel arithmetic. *IRE Transactions on Electronic Computers*, 389–400, September 1961.
- [3] K. H. Brenner, A. Huang, and N. Streibl. Digital optical computing with symbolic substitution. *Applied Optics*, 25(25):3054–3060, September 1986.
- [4] C. D. Capps, R. A. Falk, and T. L. Houk. Optical arithmetic/logic unit based on residue arithmetic and symbolic substitution. *Applied Optics*, 27(9):1682–1686, May 1988.
- [5] B. L. Drake, R. P. Bocker, M. E. Lasher, R. H. Patterson, and W. J. Miceli. Photonic computing using the modified signed-digit number representation. *Optical Engineering*, 25(1):038–043, January 1986.
- [6] D. G. Feitelson. *Optical Computing: A Survey for Computer Scientists*. MIT Press, 1988.
- [7] A. Huang. Architectural considerations involved in the design of an optical digital computer. In *Proc. IEEE*, pages 780–786, 1984.
- [8] A. Huang. Parallel algorithms for optical digital computers. In *Proceedings 10th International Optical Computing Conference*, pages 13–17, 1983.
- [9] K. Hwang. *Computer Arithmetic*. John Wiley and Sons, 1979.
- [10] K. Hwang and D. DeGroot(eds.). *Parallel Processing for Supercomputer and AI*. McGraw Hill, N. Y., Mar 1989.
- [11] K. Hwang and A. Louri. Optical multiplication and division using modified signed-digit symbolic substitution. *Optical Engineering, Special Issue on Optical Computing*, April 1989.
- [12] A. Louri. *A Bit Plane Architecture and 2-D Symbolic Substitution Algorithms for Optical Computing*. PhD thesis, University of Southern California, Department of EE Systems, August 1988.
- [13] A. Louri and K. Hwang. A bit-plane architecture for optical computing with two-dimensional symbolic substitution. In *15th Intl. Symposium on Computer Architecture*, pages 18–27, May 1988.
- [14] J. Tanida and Y. Ichioka. OPALS: optical parallel array logic systems. *Applied Optics*, 25(10):1565–1570, May 1986.
- [15] J. Tanida and Y. Ichioka. Optical logic array processor using shadowgrams. *Journal of Optical Society of America*, 73(6):800–809, June 1983.
- [16] P. K. Tu and M. D. Ercegovac. A radix-4 on-line division algorithm. In *Proceedings of 8th Symposium on Computer Arithmetic*, pages 181–187, May 1987.