

Fig. 4. A second example graph.

This modulo 3 technique does not work for graphs of unequal edge lengths, however, since the unequal lengths can put the labels "out of synchronization" (which can be verified by adding edge *BH* of length 1 to the first example problem). In the algorithm described here, the 2-bit vertex labels indicate if an edge has been traversed and, if so, in what direction, with synchronization being provided by edge length modification.

While this algorithm can be used with sequential computer, it has been designed expressly to exploit the highly parallel search and arithmetic properties of associative memory so as to gain improved execution speeds; and it is with associative memory that the saving in storage required for graph representation becomes important because of the higher cost per bit.

REFERENCES

[1] E. F. Moore, "The shortest path through a maze," *Ann. Computation Laboratory of Harvard University*, vol. 30. Cambridge, Mass.: Harvard University Press, 1959, pp. 285-292.
 [2] M. Pollack and W. Wiebenson, "Solutions of the shortest-route problem—a review," *Operations Research*, vol. 8, pp. 224-230, March-April 1960.
 [3] C. Y. Lee, "An algorithm for path connections and its applications," *IEEE Trans. Electronic Computers*, vol. EC-10, pp. 346-365, September 1961.
 [4] S. B. Akers, Jr., "A modification of Lee's path connection algorithm," *IEEE Trans. Electronic Computers (Short Notes)*, vol. EC-16, pp. 97-98, February 1967.
 [5] M. H. Lewin, "Retrieval of ordered lists from content-addressed memory," *RCA Rev.*, pp. 215-229, June 1962.
 [6] R. H. Fuller and G. Estrin, "Some applications for content-addressable memories," *1963 Fall Joint Computer Conf., AFIPS Proc.*, vol. 24. Baltimore, Md.: Spartan, 1963, pp. 495-508.

A New Algorithm for Inner Product

S. WINOGRAD, MEMBER, IEEE

Abstract—In this note we describe a new way of computing the inner product of two vectors. This method cuts down the number of multiplications required when we want to perform a large number of inner products on a smaller set of vectors. In particular, we obtain that the product of two $n \times n$ matrices can be performed using roughly $n^3/2$ multiplications instead of the n^3 multiplications which the regular method necessitates.

Index Terms—Algorithm, inner product, matrix inversion, matrix multiplication, solution of linear equations.

Manuscript received January 18, 1968; revised March 13, 1968. The author was with IBM Research Center, Yorktown Heights, N. Y. 10598. He is now with the Dept. of Elec. Engrg. and Computer Sciences, University of California, Berkeley, Calif. 94720

I. THE ALGORITHM

Let $x = (x_1, \dots, x_n)$ and $y = (y_1, \dots, y_n)$ be two vectors. For each vector we compute the number

$$\xi = \sum_{j=1}^{\lfloor n/2 \rfloor} x_{2j-1} \cdot x_{2j}$$

(where $\lfloor t \rfloor$ denotes the integer part of t), and

$$\eta = \sum_{j=1}^{\lfloor n/2 \rfloor} y_{2j-1} \cdot y_{2j}.$$

The inner product (x, y) is then given by

$$(x, y) = \begin{cases} \sum_{j=1}^{\lfloor n/2 \rfloor} (x_{2j-1} + y_{2j})(x_{2j} + y_{2j-1}) - \xi - \eta & \text{if } n \text{ is even} \\ \sum_{j=1}^{\lfloor n/2 \rfloor} (x_{2j-1} + y_{2j})(x_{2j} + y_{2j-1}) - \xi - \eta + x_n y_n & \text{if } n \text{ is odd.} \end{cases}$$

Consider the case where N n -dimensional vectors are given, and it is desired to perform T inner products involving these vectors. The total number of multiplications required is then $N\lfloor n/2 \rfloor + T\lfloor (n+1)/2 \rfloor = Nn + (T-N)\lfloor (n+1)/2 \rfloor$ as compared with $Tn = Nn + (T-N)n$. So if $T > N$, the new method requires fewer multiplications than the regular method. The total number of additions required is $N(\lfloor n/2 \rfloor - 1) + T(n + \lfloor n/2 \rfloor + 1)$, while the regular method requires only $T(n-1)$ additions. If $T \gg N$, then the total number of operations in the new method is about the same as the total number of operations in the regular method; therefore, the new method is faster when the time required to multiply is longer than the time required to add.

II. APPLICATIONS

A. Matrix Multiplication

Let A be an $m \times n$ matrix, and B an $n \times p$ matrix. Performing the product $A \cdot B$ is equivalent to giving $N = m + p$ vectors and performing $m \cdot p$ inner products. The total number of multiplications is $(m+p)n + (m \cdot p - m - p)\lfloor (n+1)/2 \rfloor$ compared with $m \cdot p \cdot n$ multiplications in the regular way of performing matrix multiplication. The number of additions is $(m+p)(\lfloor n/2 \rfloor - 1) + mp(n + \lfloor n/2 \rfloor + 1)$ compared with $mp(n-1)$ in the regular way. If we assume m, n , and p are large, then the new method requires about $\frac{1}{2} mnp$ multiplications and $\frac{3}{2} mnp$ additions, while the regular method requires mnp multiplications and mnp additions.

B. Matrix Inversion

Let A be an $n \times n$ matrix to be inverted. Gaussian elimination method requires n^3 multiplications (we count a division as a multiplication) and $n^3 - 2n^2 + n$ additions. If $n = m \cdot k$ then we can view A as an $m \times m$ matrix whose entries are $k \times k$ matrices. We can invert A by Gaussian elimination of this $m \times m$ matrix, where addition and multiplication means addition and multiplication between $k \times k$ matrices and inversion means inverting a matrix. Applying the above method for the multiplications and assuming that the inversion of the $k \times k$ matrices is done by the regular Gaussian elimination, we obtain (assuming k is even) that

$$\frac{n^3}{2} + \frac{3}{2} n^2 + \frac{n}{2} (k^2 - k)$$

multiplications and

$$\frac{3}{2} n^3 \left(1 + \frac{4}{3k}\right) - n^2 \left(\frac{1}{2} + \frac{3}{k}\right) + n \left(2 - \frac{5}{2} k - \frac{1}{2} k^2\right)$$

additions are required.

C. Solutions of Linear Equations

Consider the system of linear equations $Ax = b$, where A is an $n \times n$ matrix. Solving these equations by Gaussian elimination requires $\frac{1}{2}(n^3 + 3n^2 - n)$ multiplications and $\frac{1}{2}(2n^3 + 3n^2 + n)$ additions.

As in the previous section, if $n = m \cdot k$ we perform the Gaussian elimination method on the $m \times m$ matrix whose entries are $k \times k$ matrices. Assuming that k is even this method requires

$$\frac{n^3}{6} + \frac{n^2}{4} (7k - k^2) + \frac{n}{6} (11k^2 + 3k + 6)$$

multiplications and

$$\frac{n^3}{2} + n^2 \left(\frac{3}{2} - \frac{1}{4k} - \frac{1}{2k} \right) + n \left(\frac{3}{2} k^2 - \frac{20}{6} k + \frac{7}{4} - \frac{1}{2k} \right)$$

additions.

ACKNOWLEDGMENT

The author wishes to thank M. O. Rabin for his suggestions for the format of this note.

A Variable Counter Design Technique

EDWARD L. RENSCHLER, MEMBER, IEEE

Abstract—Two design techniques are presented which allow one counter circuit to divide a fixed-reference frequency by a wide range of counts. In the examples used the output frequency is preselected on three 10-position selector switches, providing for division of the input reference frequency by N , where $1 \leq N \leq 999$. The techniques described are not dependent on the type of digital logic used and are therefore applicable to any family of binary logic modules.

Index Terms—Counter, divider, programmable counter.

An externally programmable counter is desirable in many frequency-synthesizing systems. For this counter to be of value, one must be able to change the count state, in steps of one, over a range of three or four hundred. In addition, this count change must be done externally to the counter itself. This means that no wiring can be changed and that the count decision must be made through logic. Two techniques that accomplish this are presented here. The techniques are described in general terms and will be directly applicable to any form of binary logic. The counters used as examples in this paper have the constraint

$$1 \leq N \leq 999, \tag{1}$$

where N is the desired count.

The basic counter system is shown in Fig. 1. A description of the system operation is as follows. The desired count N is selected by setting each of the three 10-position input switches to the desired decimal number. Only three input switches are used here because of the constraint given in (1). N can be represented, for a 3-digit number, as

$$N = \sum_{i=0}^2 \sum_{j=0}^3 10^i 2^j B_{ij} \tag{2}$$

which is a 12-bit BCD word.

Each of the three stages of decimal-BCD conversion logic will be identical. One stage is illustrated in Fig. 2. The logic used here is a positive-logic NAND function.

The first technique for performing this variable-count function is shown in Fig. 3. When the shift pulse arrives at the input gating logic, each of the BCD numbers is preset into the appropriate counter

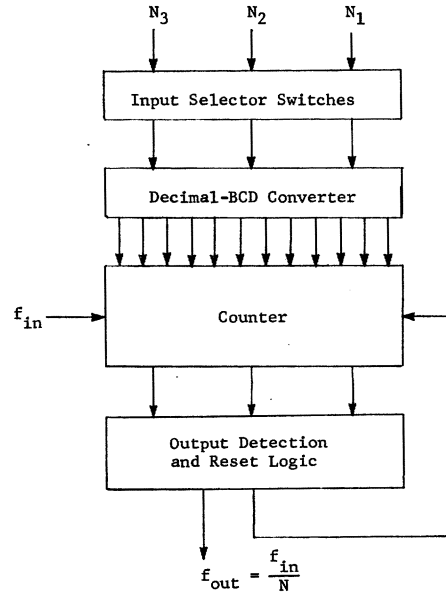


Fig. 1. Basic programmable counter.

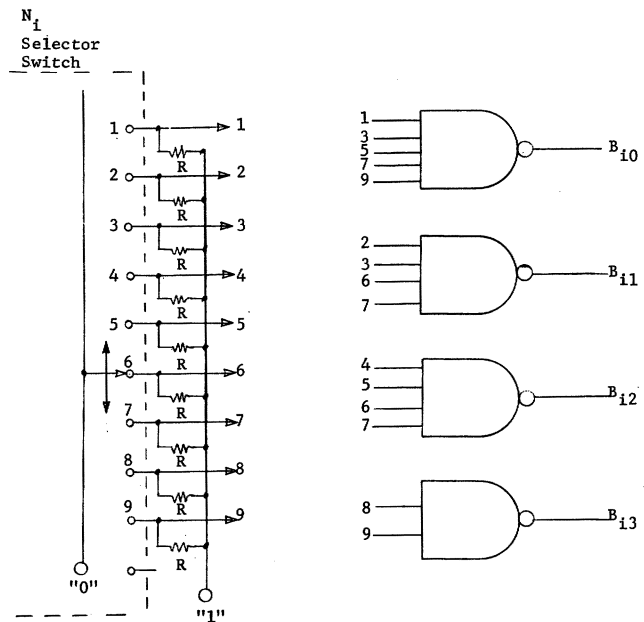


Fig. 2. Decimal-BCD conversion logic.

stage. Each counter is a clocked BCD decade *down* counter. All three counter stages are driven from the same clock through gating, as opposed to the ripple-counter approach. This makes the counter system completely synchronous and is done deliberately to avoid the well-known problems of ripple counters. The count continues until the three stages are simultaneously in the zero state. When this occurs, the counter has counted down from the preset number N to 0. The zero state is detected by the zero state detector, and a trigger is produced which fires the "one shot," which in turn produces the shift pulse necessary to preset the number N again into the counter. The counter is reloaded during the zero count, and upon the next clock pulse the divide-by- N process begins again.

The one shot is necessary to insure that the number N is properly preset into the counter. If the one shot were not included, the dura-

Manuscript received October 9, 1967; revised April 18, 1968. The author is with the Semiconductor Products Div., Motorola, Inc., Phoenix, Ariz.