IV. CONCLUSIONS

All results shown above agree with intuitive reasoning. However, it enables one to make comparisons using probability figures and come up with an optimum number of logic gate inputs that will satisfy a given input reliability requirement. We strongly feel that optimal logic circuit design should adopt the probabilistic design approach. It is our sincere hope that this work will trigger some further research interest in this area.

ACKNOWLEDGMENT

The author wishes to thank Prof. E. Klingshirm and G. Butters for their helpful discussions.

REFERENCES

[1] J. Von Neuman, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," in *Automata Studies*. Princeton, N.J.: Princeton Univ. Press, 1956.
[2] E. F. Moore and C. Shannon, "Reliable circuits using less reliable relays," *J. Franklin Inst.*, vol. 262, no. 3, part I, pp. 191–208; no. 4, part II, pp. 281–297, 1956.
[3] K. P. Parker and E. J. McCluskey, "Analysis of logic circuits with faults using input signal probabilities," in *Dig. 1974 Int. Symp. Fault-Tolerant Comput.*, Champaign, Ill., June 1974, pp. 1-8–1-12.
[4] R. C. Ogus, "The probability of a correct output from a combinational circuit," in *Dig. 1974 Int. Symp. Fault-Tolerant Comput.*, Champaign, Ill., June 1974, pp. 1-13–1-19.
[5] W. H. Pierce, *Failure-Tolerant Computer Design*. New York: Academic, 1965.
[6] H. Mine and Y. Koga, "Basic properties and a construction method for fail-safe logic systems," *IEEE Trans. Electron. Comput.*, vol. EC-16, pp. 282–289, June 1967.
[7] R. C. Ogus, "Fault-tolerance of the iterative cell array switch for hybrid redundancy," *IEEE Trans. Comput.*, vol. C-23, pp. 667–681, July 1974.
[8] W. H. Von Alven, Ed., *Reliability Engineering*, ARINC Res. Corp. Englewood Cliffs, N.J.: Prentice-Hall, 1964.

A Proof of the Modified Booth's Algorithm for Multiplication

LOUIS P. RUBINFIELD

***Abstract*—A simplified proof of a modification of Booth's multiplication algorithm by MacSorley to a form which examines three multiplier bits at a time is presented. In comparison with the original Booth's algorithm, which examines two bits at a time, the modified algorithm requires half the number of iterations at the cost of somewhat increased complexity for each iteration.**

***Index Terms*—Modified Booth's algorithm, multiplicand, multiplier, partial product.**

Many multiplication algorithms exist which increase the speed of operation over the classic shift and add method. These algorithms may be divided into two categories: variable shift methods and uniform shift methods. The variable shift methods are disadvantageous for clocked systems since the time required for a multiply is data dependent. Booth's algorithm [1], a uniform shift method, examines two bits of the multiplier at a time to determine the correct multiple of the multiplicand to be added to the partial product. This method requires no sign correction for a two's complement number and the decoding of the multiplier may be begun from either direction. The major disadvantage of the algorithm is that the process still requires n shifts and an average of $n/2$ additions for an n bit multiplier.

Manuscript received June 3, 1974; revised March 21, 1975. This work was supported in part by the Advanced Research Projects Agency of the Department of Defense under Contract SD-302 and in part by the Division of Research Resources of the National Institutes of Health under Grant RR-00396.

The author is with the Computers Systems Laboratory, Washington University, St. Louis, Mo. 63110.

TABLE I
MODIFIED BOOTH'S ALGORITHM FOR A LEAST TO MOST SIGNIFICANT SCAN OF BITS

Multiplier Bits $y_{i-1}\ y_i\ y_{i+1}$	Operation
0 0 0	$PP_i \leftarrow (1/4)PP_{i+2}$
0 0 1	$PP_i \leftarrow (1/4)PP_{i+2} + X$
0 1 0	$PP_i \leftarrow (1/4)PP_{i+2} + X$
0 1 1	$PP_i \leftarrow (1/4)PP_{i+2} + 2X$
1 0 0	$PP_i \leftarrow (1/4)PP_{i+2} - 2X$
1 0 1	$PP_i \leftarrow (1/4)PP_{i+2} - X$
1 1 0	$PP_i \leftarrow (1/4)PP_{i+2} - X$
1 1 1	$PP_i \leftarrow (1/4)PP_{i+2}$

For
i $n-1, n-3, \cdots, 3, 1$.
PP_i ith partial product, $PP_{n+1} = 0$.
X Multiplicand.
Y Multiplier, $n+1$ bits wide, $y_n = 0$.

An increased multiplication speed can be achieved by examining more than two bits of the multiplier at a time. A suggested modification by MacSorley [2] and used extensively in present day computers (see [3], for example) decodes three bits of the multiplier at a time. The decoding of the multiplier and the action prescribed is shown in Table I. Again, the examination of the multiplier may be started at either end. However, it is often advantageous to begin the examination with the least significant bit. The following proof of the algorithm shows that the decoding and subsequent action of the modified Booth's algorithm in Table I is correct.

First let Y be the fractional multiplier in two's complement form. Also let y_i be the ith bit of the multiplier with y_0 the sign bit and y_{n-1} the least significant bit of an n bit number. Furthermore, append to the right of y_{n-1} an additional bit, $y_n = 0$, which will not change the numerical value of Y. Assume that n is even. Thus the multiplier may be written as

$$Y = -y_0 + \sum_{i=1}^{n-1} y_i 2^{-i} \tag{1}$$

where $y_i \in \{0,1\}$, for $i = 0,1,2,\cdots,n-1$. Thus,

$$Y = -y_0 + \sum_{i=1\ \text{odd}}^{n-1} y_i 2^{-i} + \sum_{i=2\ \text{even}}^{n-2} y_i 2^{-i}. \tag{2}$$

Adding and subtracting the rightmost term of (2) and recombining yields

$$Y = -y_0 + \sum_{i=1\ \text{odd}}^{n-1} y_i 2^{-i} + \sum_{i=2\ \text{even}}^{n-2} y_i 2^{-i+1} - 2 \sum_{i=2\ \text{even}}^{n-2} y_i 2^{-i-1}. \tag{3}$$

Rearranging (3) so that all powers of 2 are the same and the limits of the sums are equal yields

$$Y = \sum_{i=1\ \text{odd}}^{n-1} (y_i + y_{i+1} - 2y_{i-1}) 2^{-i} \tag{4}$$

for $y_j \in \{0,1\}$, for all $j = 0,1,2,\cdots,n-1$. Thus the correct multiple of the multiplicand is found by calculating the sum

$$z_i = y_i + y_{i+1} - 2y_{i-1}. \tag{5}$$

With X the multiplicand, the product XY may now be written as

$$XY = \sum_{i=1\ \text{odd}}^{n-1} X z_i 2^{-i}. \tag{6}$$

Expanding the sum yields

$$XY = (1/2)\{z_1X + (1/4)[z_3X + \cdots + (1/4)[z_{n-3}X + (1/4)[z_{n-1}X]]\cdots]\}. \quad (7)$$

If we define the partial product at each iteration as PP_i, for $i = n+1, n-1, n-3, \cdots, 5, 3, 1$ with $PP_{n+1} = 0$, then (7) may be written as an iterative process,

$$PP_i \leftarrow z_iX + (1/4)PP_{i+2} \quad (8)$$

for $i = n-1, n-3, \cdots, 5, 3, 1$. The final product, XY, is found to be one half of the last partial product $(1/2PP_1)$.

Equation (8) now defines the modified Booth's algorithm as an iterative process for the examination of the multiplier starting with the least significant bits. That is, the counter i starts at $i = n-1$ and ends with $i = 1$. The $(1/4)PP_{i+2}$ in (8) indicates that the partial product must be shifted right two places prior to its addition to the multiple of the multiplicand, z_iX.

For the first iteration, $i = n-1$, the correct multiple is found to be [from (5)]

$$z_{n-1} = y_{n-1} + y_n - 2y_{n-2}.$$

Since y_n is always zero, an extra bit position for y_n is concatenated to the right of the least significant bit of the multiplier register, y_{n-1}. This bit position is set to zero when the multiplier is loaded. Thus the decoding window looks at this extra bit and two bits to the left of it.

To summarize, the steps in the modified Booth's algorithm are as follows: 1) The three least significant bits of the multiplier augmented by $y_n = 0$ are examined and decoded (see Table I). 2) The resulting multiple of the multiplicand is added to or subtracted from the previous partial product, forming a new partial product. (Initially the partial product is zero.) 3) The new partial product and the multiplier are shifted right two places. 4) The above operations are repeated $n/2$ times ($n \geq 0$ and even). After the final addition or subtraction, the partial product is shifted right one place to form the final product.

REFERENCES

[1] A. D. Booth, "A signed binary multiplication technique," *Quart. J. Mech. Appl. Math.*, vol. 4, part 2, 1951.

[2] O. L. MacSorley, "High-speed arithmetic in binary computers," *Proc. IRE*, vol. 49, pp. 67–91, Jan. 1961.

[3] S. Anderson, J. Earle, R. Goldschmidt, and D. Powers, "The IBM system 360 model 91: Floating-point execution unit," *IBM J. Res. Develop.*, vol. II, no. 1, Jan. 1967.

A New Method of Formulating a Minimum Edge Set

JUHANI NIEMINEN

***Abstract*—Mayeda and Ramamoorthy have reduced the problem of finding a minimum number of test points which detect an improperly operating functional element of a single entry-single exit (SEC) system graph, to the problem of finding a minimum number of edges under which this system graph is 1 distinguishable. This correspondence shows that the determination of the minimum edge set is equivalent to the determination of a minimum covering of a certain matrix M (S) of zeros and ones. A way of finding the matrix M (S) is proposed.**

***Index Terms*—Directed graphs, fault location, minimum covering, minimum number of test points, one distinguishability.**

Manuscript received November 26, 1973; revised March 19, 1975.

The author is with the Department of Technical Sciences, Finnish Academy, Lauttasaarentie, Helsinki, Finland.

I. INTRODUCTION

A single entry single-exit (SEC) graph G is a directed graph without directed circuits and with one source vertex s and one sink vertex s', i.e., the indegree of s and outdegree of s' are zeros in G. Any discrete sequential system can be shown to be isomorphic to a directed graph and any directed graph with directed circuits can be transformed to a SEC graph [1]. A system described by a SEC graph operates improperly, if it contains a faulty functional element, i.e., G contains a faulty vertex. It is of importance to find a minimum set of test points determining whether or not all vertices of the system are operating properly. According to Mayeda and Ramamoorthy [1], such a set of test points is given, if one can find a minimum set $M \subset E(G)$ of edges of G, with respect to which the SEC graph G of the system is 1 distinguishable.

A SEC graph G is said to be k distinguishable with respect to an edge set M, if the set generates a partition D of vertices of G such that there is a set in D containing k vertices but there are no sets in D containing more than k vertices. In their paper Mayeda and Ramamoorthy derived a necessary condition for 1 distinguishability of a SEC graph [1, theorem 4], and it was sharpened in [6, theorem 14–2–7]. In this correspondence we shall concentrate on the determination of a minimum edge set $M_{\min}$ of G giving a 1 distinguishable partition of the vertices of G. This determination problem is equivalent to the problem of finding a minimum cover of a matrix M (S) obtained from the graph G and its complement $\bar{G}$, as it will be shown.

The relations between the test points and the minimum edge set $M_{\min}$ are reported in [1] and [6]. In the second section we shall briefly recall some central concepts and results of [1] needed here. As the main reference we have used [1]; as a general reference, where some further results are also given, one can use [6, chap. 14], to which the reader is referred. Other interesting approaches to the determination of test points in a SEC graph are given in [2]–[5].

II. NOTATIONS AND PRELIMINARY RESULTS

A graph $G = (V(G), E(G))$ is a pair of sets, where $V(G) = \{v_1, \cdots, v_n\}$ is the set of vertices in G and $E(G) = \{e_1, e_2, \cdots, e_p\}$ is the set of edges joining the vertices in G.

A set d of vertices of a graph is called a dominating set of G, when each vertex not in d is the endpoint of some edge from a vertex in d. A dominating set d^* of G is a minimum dominating set, if $|d^*| \leq |d|$ for any dominating set d of G.

Let V_i be a subset of the vertices of G; $\bar{V}_i$ denotes its complement in $V(G)$, i.e., $\bar{V}_i = V(G) - V_i$. Let V_i and V_j be two nonempty subsets of $V(G)$ such that $V_i \cap V_j = \emptyset$, then $E(V_i \times V_j)$ is the set of all edges in G that are connected between a vertex in V_i and a vertex in V_j. Clearly $S = E(V_1 \times \bar{V}_1)$ is a cutset of G, i.e., the removal of its edges from G separates G into two components. A cutset $S = E(V_1 \times \bar{V}_1)$ is a directed cutset of a directed graph G if every edge in S emanates from a vertex in V_1 and terminates on a vertex in $\bar{V}_1$. Assume that $E(V_i \times V_j)$, $V_i \cap V_j = \emptyset$, is a cutset of G such that all its edges emanate from a vertex in V_i and terminate on a vertex in V_j. As $E(V_i \times V_j)$ is a cutset of G, the vertex sets V_i and V_j can be completed with the vertices of $V(G) - \{V_i \cup V_j\}$ to the sets V_i' and V_j' such that $V_j' = \bar{V}_i'$ and $E(V_i \times V_j) = E(V_i' \times \bar{V}_i')$. Hence any directed cutset of G is of the type $E(V_i \times \bar{V}_i)$. A directed cutset S_{ij} is said to separate the vertices v_i and v_j of G, if $v_i \in V_i$ and $v_j \in \bar{V}_i$.

Let $e = (v_{1e}, v_{2e})$ be a directed edge from v_{1e} to v_{2e} in a SEC graph