the Boolean identities. However, $\mathfrak{F}$ and $\bar{\mathfrak{F}}$ do not have to be in the minimal form.

$G3$) For each X_i in F' generate all the X_j which are one variable less specified. If there are multiple copies of the same X_j only one need be retained in the total list.

$G4$) For all X_j in the list compute

$$\hat{X}_j = \bar{\mathfrak{F}} X_j \tag{16}$$

and if $\hat{X}_j \neq \phi$ leave the X_j in the X_j list.

$G5$) If $\hat{X}_j = \phi$ in (16), replace X_j with all one variable less specified terms from X_j and add to the X_j list and go to $G4$.

$G6$) When the X_j list is complete, any (preferably minimal) set of minterms from $\bar{\mathfrak{F}}$ that cover all the X_j is the abnormal true tests.

$G7$) For each X_k in $\bar{F}'$ go through $G3$–$G5$ forming a partial X_m list and the $\hat{X}_m$ by

$$\hat{X}_m = \mathfrak{F} \cdot X_m. \tag{17}$$

$G8$) When the X_m list is complete, any (minimal) covering set of minterms from $\mathfrak{F}$ is the abnormal false tests.

The commentors' conclusion 3) applies to any near-minimal solution. As to the question arising from their conclusion 1), one does not have to perform the NR procedure on both F and $\bar{F}$ sides. The better side to work on would be the side that has fewer terms in the expression. Statistically speaking, one would expect a lesser number of test patterns from this side as well as smaller computation efforts.

REFERENCES

[1] C. D. Latino and J. G. Bredeson, "Comments on "Cause-effect analysis for multiple fault detection in combinational networks,'" this issue, p. 1326.

Comments on "Parallel Processing Algorithms for the Optimal Control of Nonlinear Dynamic Systems"

DANIEL TABAK

In a recent publication[1] the authors have presented, in detail' parallel processing algorithms, solving several classes of optimal control problems. Considerable attention is given to dynamic programming. On p. 777 the authors state: "The purpose of this paper is to describe a continuing effort in the development of algorithms for solving optimal control problems for nonlinear dynamic systems on parallel processors [4]–[11]." The author's references [4]–[11] range through the period of 1971–1973. In fact, the idea of solving dynamic programming problems on parallel processors has been brought up and documented in a much earlier correspondence [1]. The solution of other types of optimization problems by parallel processors has been discussed by the author of this correspondence at the Asilomar Conference on Circuits and Systems in 1967, and documented in its Proceedings. At that time, the ideas were discussed in general form and no detailed formulations of the algorithms were given. It is certainly pleasant to see that the work in this area has been carried on to a much more advanced extent and some of the optimal control algorithms worked out in detail. The authors should be congratulated on their valuable contribution. It will be even more interesting to see the algorithms practically implemented on parallel processors.

REFERENCES

[1] D. Tabak, "Computational improvement of dynamic programming solutions by multiprocessing techniques," *IEEE Trans. Automat. Contr.* (Corresp.), vol. AC-13, p. 596, Oct. 1968.

Manuscript received August 23, 1974.
The author is with the Department of Electrical Engineering, University of the Negev, Beer-Sheva, Israel, on leave from RPI, Hartford, Conn.
[1] R. E. Larson and E. Tse, "Parallel processing algorithms for the optimal control of nonlinear dynamic systems," *IEEE Trans. Comput.*, vol. C-22, pp. 777–786, Aug. 1973.

Comments on "A Two's Complement Parallel Array Multiplication Algorithm"

P. E. BLANKENSHIP

The two's complement multiplication algorithm discussed by Baugh and Wooley[1] is indeed a viable one, and we have used it several times in the past to great advantage in applications where modestly high performance signed multiplies were required, but the design was constrained to commercially available, standard components. This approach to sign compensation, which was first described to me in 1969 by C. M. Rader, does give rise to a uniform partial product array which can be efficiently summed by any of several techniques.

A method was devised here that used a Wallace tree summation approach and combined effective partial product formation with the first level additions by applying multiplier bit pairs as direct control inputs to programmable ALU-type devices. The programmable units are clearly necessary only on the first level of combination, subsequent levels requiring simple full adders. With standard 10 000 series ECL functions in a point-to-point wire wrap interconnect environment, 12 × 12 and 16 × 8 bit configurations were constructed, yielding measured worst case settling times of 45 ns.

As a point of further interest, it might be noted that the P_{n+m-1} and P_{n+m-2} columns of Fig. 3 of the above paper[1] can be modified in several different ways to reduce the number of terms to be summed. The lower left-hand corner of the Baugh–Wooley array is

$$\begin{array}{cc} & x_{n-1} \cdot y_{m-1} \\ & \bar{x}_{n-1} \\ 1 & \bar{y}_{m-1} \\ \hline P_{n+m-1} & P_{n+m-2} \end{array} \cdots.$$

S. D. Pezaris suggests the following alteration, which has the advantage of employing the same AND functions as the rest of the array:

$$\begin{array}{cc} & 1 \\ 1 & \bar{x}_{n-1} \cdot \bar{y}_{m-1} \\ \hline P_{n+m-1} & P_{n+m-2} \end{array} \cdots.$$

An alternate approach which we found useful for our programmed ALU/Wallace tree structure requires only the INCLUSIVE-OR of the sign bits in each column:

$$\begin{array}{cc} x_{n-1} \cup y_{m-1} & x_{n-1} \cup y_{m-1} \\ \hline P_{n+m-1} & P_{n+m-2} \end{array} \cdots.$$

All three approaches are equivalent, as can be easily verified by direct evaluation of the logical expressions for a binary full adder. Also, in many applications the P_{n+m-1} bit constitutes redundant information and its formation can be omitted entirely. Certainly P_{n+m-1} agrees with P_{n+m-2} in all but the special case wherein the product of the greatest negative operands is desirable.

Manuscript received February 12, 1974. This work was supported by the Department of the Air Force.
The author is with the M.I.T. Lincoln Laboratory, Lexington, Mass. 02173.
[1] C. R. Baugh and B. A. Wooley, "A two's complement parallel array multiplication algorithm," *IEEE Trans. Comput.*, vol. C-22, pp. 1045–1047, Dec. 1973.

Comments on "A Two's Complement Parallel Array Multiplication Algorithm"

DAVID KROFT

***Abstract*—This correspondence presents a simpler proof for Baugh and Wooley's two's complement parallel array multiplication algorithm, as demonstrated in a recent paper.[1] The above algorithm**

Manuscript received March 29, 1974; revised June 24, 1974. This work was supported by Control Data Canada Ltd., Mississuaga, Ont., Canada.
The author is with Canadian Development Division, Control Data Canada Ltd., Mississauga, Ont., Canada.
[1] C. R. Baugh and B. A. Wooley, *IEEE Trans. Comput.*, vol. C-22, pp. 1045–1047, Dec. 1973.