Correction to “Syndrome-Testable Design of Combinational Circuits”

During the production process of the paper “Syndrome-Testable Design of Combinational Circuits” by Dr. Jacob Savir, which appeared in the June 1980 issue of this IEEE TRANSACTIONS (vol. C-29, pp. 442–451), several corrections supplied by the author were inadvertently omitted from the final version of the paper. We apologize to Dr. Savir for this omission. The following material is the correct version of his Example 2 and Lemma 7, which appear on pages 444 and 447, respectively, of the June issue. Several other errata, which were supplied by the author but were also not corrected in the paper, are included as well.

On p. 444, second column, 21st line, the phrase “such that no fault” should read: “such that no single fault.”

On p. 445, second column, the 4th line should read: “paths emanate with unequal inversion parities.”

Now, where

Thus, $S(C_1) = \frac{1}{16}, S(B_1C_1) = \frac{1}{4}$. 

We can describe $F$ in the form

\begin{align}
F &= A_1x + B_1\overline{x} + C_1
\end{align}

where

\begin{align}
A_1 &= wz + y \\
B_1 &= \overline{y}z \\
C_1 &= \overline{w}y\overline{z}.
\end{align}

Thus,

\begin{align}
A_1\overline{C}_1 &= wz + wy + yz \\
B_1\overline{C}_1 &= \overline{y}z.
\end{align}

Note that $S(x_1^*x_2^*\cdots x_n^*) = 2^{-n}$ and $S(x_1^* + x_2^* + \cdots + x_n^*) = 1 - 2^{-n}$ where $x_i^* \in \{x_i, \overline{x}_i\}, i = 1, 2, \cdots, n$.

Thus, $S(C_1) = \frac{1}{16}, S(B_1C_1) = \frac{1}{4}$

\begin{align}
S(F) &= \frac{S(A_1\overline{C}_1) + \frac{1}{4}}{2} + \frac{1}{16}.
\end{align}

Now, we express $A_1\overline{C}_1$ in the form

\begin{align}
A_1\overline{C}_1 &= A_2y + B_2\overline{y} + C_2
\end{align}

where

\begin{align}
A_2 &= w + v + z \\
B_2 &= 0 \\
C_2 &= wz.
\end{align}

Thus

\begin{align}
A_2\overline{C}_2 &= w\overline{z} + \overline{w}v + \overline{w}z
\end{align}

\begin{align}
B_2\overline{C}_2 &= 0 \\
S(A_1\overline{C}_1) &= S(A_2\overline{C}_2) + \frac{1}{4}.
\end{align}

Now we express $A_2\overline{C}_2$ in the form

\begin{align}
A_2\overline{C}_2 &= A_3z + B_3\overline{z} + C_3
\end{align}

where

\begin{align}
A_3 &= \overline{w} \\
B_3 &= w \\
C_3 &= \overline{w}v.
\end{align}

Thus,

\begin{align}
A_3\overline{C}_3 &= \overline{w}v, S(A_3\overline{C}_3) &= \frac{1}{4} \\
B_3\overline{C}_3 &= w, S(B_3\overline{C}_3) &= \frac{1}{2} \\
S(A_2\overline{C}_2) &= \frac{1 + \frac{1}{4}}{2} + \frac{1}{4} = \frac{5}{8}.
\end{align}

By using (7), (6), and (5) we derive the desired syndrome to be

\begin{align}
S(F) &= \frac{15}{32}.
\end{align}

Lemma 7: Let $g = g(x_1, x_2, \cdots, x_n)$ be a line in a general combinational circuit. Let the equivalent sum of products of the function $F$ with respect to line $g$ be

\begin{align}
F &= Ag + Bg + C.
\end{align}

Then the fault $g/0$ is syndrome-untestable if and only if

\begin{align}
S(A\overline{C}g) &= S(B\overline{C}g)
\end{align}

and, the fault $g/1$ is syndrome-untestable if and only if

\begin{align}
S(A\overline{C}g) &= S(B\overline{C}g).
\end{align}

Proof: We prove relation (9). The proof of (10) is similar.

The fault $g/0$ is syndrome-untestable if and only if

\begin{align}
S(F) &= S(B + C).
\end{align}
Using Lemmas 1 and 2, we have

\[ S(F) = S(ACg) + S(BCg) + S(C) \]

\[ S(B + C) = S(BC) + S(C). \]

Thus, the fault g/0 is syndrome-untestable if and only if

\[ S(ACg) = S(BCg). \quad \text{Q.E.D.} \]

Correspondence

Synthesis of Combinational Logic Using Decomposition and Probability

STEPHEN C. CRIST

Abstract—A new algorithm for the synthesis of single-output two-valued combinational logic using decomposition and probability is described. Two probabilistic quantities, the probability of existence of all pertinent decomposition classes, and the probable cost of an \( N \) variable function are defined. Results for completely specified functions are derived and tabulated. The extension to incompletely specified functions is discussed. A randomly chosen function is assumed throughout the paper.

The probabilistic approach parallels Curtis' worst case cost algorithm [3] with two important exceptions. Decomposition classes with a low probability of existence are not considered, thus reducing the amount of computation required. Also, probable cost is used in lieu of worst case cost as a basis for decisions. The probabilistic approach indicates that for functions of more than five variables, consideration of nontrivial decompositions results in negligible improvement over trivial decompositions.

Index Terms—Combinational logic, decomposition, logic design, switching theory.

I. INTRODUCTION

Several algorithms [3], [4], [6], [11] have been proposed for combinational logic synthesis using decomposition. Most algorithms have in common the procedure of decomposing the original switching function into subfunctions of fewer variables. The subfunctions are similarly decomposed. The process is repeated until only basis functions remain.

Many algorithms [6], [11] can be classified as exhaustive search. Much computation and storage are required to insure that an economical decomposition is not overlooked. Curtis' worst case cost algorithm [3] uses a different philosophy. At no point in this algorithm is more than one decomposition retained for further consideration.

The choice of which decomposition to retain is based on the worst case cost of the new functions generated by the possible decompositions.

Worst case cost is defined as the cost of implementing a function using only decompositions that exist for all functions. (These are referred to as trivial decompositions.) The most economical decomposition class is checked first. If one of these decompositions exists, it is used. If it does not exist, then the next most economical class of decompositions is checked. The procedure is continued until a nontrivial decomposition is found or all applicable nontrivial decompositions have been checked. If no nontrivial decompositions exist, then the recursive decomposition, which exists trivially for any variable partition, is used. (Nontrivial decompositions with a higher worst case cost than the recursive circuit are not considered.) The worst case cost algorithm thus yields a design that is perhaps less economical than one obtained using exhaustive search, but requires considerably less computation.

The algorithm presented herein parallels Curtis' algorithm with two basic differences:

1) The probable cost of an \( N \) variable function with \( d \) DON'T CARES using the \( i \)th decomposition class, denoted \( C_i(N, d) \), replaces worst cost as the decision criterion. The overall probable cost of an \( N \) variable function with \( d \) DON'T CARES is denoted \( C_p(N, d) \).

2) The probability of existence of a decomposition class of an \( N \) variable function with \( d \) DON'T CARES, denoted \( P_i(N, d) \), is assigned a minimum value. Any decomposition class with a lower probability of existence than the minimum is not considered. \( P_i(N, d) \) is the probability that one or more partitions of the same class yield a decomposition. A flowchart of the algorithm is shown in Fig. 1.

There is no set of basis functions universally applicable to all logic families. The set of basis functions used to compute probable cost is the set of all two variable functions. This is consistent with Curtis' worst case cost algorithm and can be taken as a measure of complexity for any logic family. The cost of a two-variable function is taken to be one. Thus, the probable cost of a function is the probable number of two input gates required to synthesize it, assuming all such gates are available.

A later algorithm proposed by Curtis [4] uses a totally different approach. The recursive circuit, given by

\[ F(x_1 - x_N) = G[\eta \{ \phi (x_1 - x_{N-1}), x_N \}, \chi (x_1 - x_{N-1})] \]  (1)

exists trivially and reduces an \( N \) variable function to two \( N - 1 \) variables functions \( \phi \) and \( \chi \) plus two two-variable functions \( G \) and \( \eta \). A recursive decomposition can also be applied to any \( N - 1 \) variable functions generated in (1). Curtis devised procedures whereby multiple recursive decompositions of a particular form can be readily