A FAST ALGORITHM FOR THE PROPER DECOMPOSITION OF BOOLEAN FUNCTIONS

by A. THAYSE

Abstract
A fast algorithm for the computation of the set of $\Delta$-operators of a given Boolean function is presented. This algorithm together with some new theorems on functional decomposition derived in this paper allow the building of a simple scheme for disjunctive and nondisjunctive decompositions of Boolean functions.

1. Introduction
The general problem of functional decomposition of Boolean functions has proven to be of a high computational complexity. One approach to this problem is the decomposition chart of Ashenurst 1). Ashenurst's theory has been used by Curtis 2) as a basis for a more general theory in order to develop a systematic method of deriving economical multiple-stage switching circuits. Unfortunately, the procedure proposed by Ashenurst and Curtis requires the testing of $2^n - 2^n$ decomposition charts, where $n$ is the number of input variables. Another approach has been proposed by Akers 3). The algebraic decomposition condition obtained by Akers however requires the computation of $\Phi$-operators: this leads to algebraic expressions of overwhelming size.

In this paper, we will show how an algebraization of the Ashenurst's theory directly leads to a decomposition condition much simpler than that of Akers. The algebraization is performed through a systematic use of the Boolean differential operators defined in ref. 4. In particular, we present a fast algorithm for the simple, disjunctive or not, decompositions of Boolean functions when the latter are represented by the sets of all their prime implicants and of all their prime implicates.

In secs 2 and 3 the problem is stated and the main results by Ashenurst and Curtis are briefly recalled. The theoretical results of this paper relative to decomposition conditions and to algebraic expressions of the $\Delta$-operator are gathered in secs 4 and 5 respectively. Finally an algorithm for simple decompositions is presented in sec. 6.

The notations used in this paper are those of ref. 4.

2. Statement of the problem and decompositions classification
The functional decomposition of Boolean functions is generally presented as a mathematical tool for designing economical combinational switching net-
works. Indeed, it is often possible to express a function $f(x)$ of $n$ variables as a composite function of functions, as in the following equation for example:

$$ f(x) = F[y(x_0), x_1], $$(1)

where $x_0$ and $x_1$ are subsets of the set of variables $x$. Sometimes a composite expression can be found for a Boolean function $f$, so that in the composite expression $F$ and $y$ are essentially simpler functions. Thus, if we wish to design a switching network for a Boolean function, we may accomplish this by designing networks for the various simpler functions of the composite representation.

Decompositions can be classified in different ways. According to the number of subfunctions, there are two types of decompositions: the simple decomposition and the complex decomposition. A simple decomposition is necessarily of the type of eq. (1), that is, $F$ contains only one subfunction $y$. If $F$ contains more than one subfunction the decomposition is said to be complex. The following definitions are also classical.

A multiple decomposition is characterized by an expression of the form

$$ f(x) = F[y_0(x_0), \ldots, y_m(x_m), x_{m+1}], \quad x_1 \in x. $$

An iterative decomposition is characterized by an expression of the form

$$ f(x) = F[y_0(y_1(\ldots y_m(x_m), \ldots, x_1), x_0), x_{m+1}]. $$

Multiple and iterative decompositions are evidently sub-classes of the complex decomposition class. The decompositions can also be classified according to the fact that the sets $x_i$ of variables are disjoint or not. If $x_i \cap x_j = \phi$ (empty set) for each pair $x_i, x_j \in x$, the decomposition is said to be disjunctive. If not it is said to be nondisjunctive. The proper subset $x_{lp}$ of $x_l$ is classically defined as $x_{lp} = x_l \cap (\cup x_j)$. The decompositions can finally be classified according to the fact that some of the proper subsets $x_{lp}$ are empty or not. A decomposition is said to be proper if each of its proper subsets $x_{lp}$ is non-empty. If not, it is said to be improper. The simplest nontrivial improper decompositions of $f$ are characterized by expressions of the form

$$ f(x) = F[y_0(x_0), y_1(x_0), x_1], \quad f(x) = F[y_0(y_1(x_0), x_1), x_0]. $$

According to the three ways of classification quoted hereabove, a set of decomposition types can be defined which is summarized in fig. 1. The way suggested in this paper to solve the various kinds of decomposition problems arising by considering the different types of decomposition is noted in the entries of fig. 1. This will be detailed in the next sections. The following observations may however already be done. It is clear that disjunctive improper decompositions can-
not exist. Moreover the simplest improper decompositions are all trivial; they are necessarily of one of the three following forms:

\[ f(x) = F[y(x_c), x], \]
\[ f(x) = F[y(x), x_c], \]
\[ f(x) = F[y(x), x]. \]

3. The results by Ashenurst and Curtis

Consider a simple decomposition of \( f(x) \) which is characterized by a relation of the form (1). The set \( x_0 \) will be called the bound set and the set \( x_1 \) the free set. Ashenurst gave necessary and sufficient conditions for the existence of a simple disjunctive decomposition with a given bound set in terms of a decomposition chart.

**Theorem 1a**

Let \( x_0, x_1 \) be a partition on \( x \). A switching function \( f \) is decomposable with bound set \( x_0 \) and free set \( x_1 \) if and only if the decomposition chart, with the variables in \( x_0 \) defining the columns and the variables in \( x_1 \) defining the rows, has at most four distinct kinds of rows which can be classified into the following categories:

1. all 0's;
2. all 1's;
3. a fixed pattern of 0's and 1's;
4. the complement of 3.

Curtis stated the following theorem.
Theorem 1b

If a decomposition chart satisfies the criteria on the rows by Ashenurst (see theorem 1a), then it has column multiplicity \( \nu \leq 2 \).

Ashenurst has stated a set of theorems relating simple disjunctive decompositions to complex ones. As a result it has been shown that we can derive all the complex disjunctive decompositions of a given function from the set of all its simple decompositions. Therefore it is important to build algorithms for finding all the simple decompositions of a given function.

The fundamental connection between simple disjunctive and nondisjunctive decompositions can be revealed by means of the following theorem.

Theorem 2

Let \( x_0, x_1, x_2 \) be a partition on \( x \). A switching function \( f \) is decomposable with bound set \( x_0, x_2 \) and free set \( x_1, x_2 \) if and only if the 2\( p \) subfunctions (\( p \) being the dimension of \( x_2 \)) \( f(x_0, x_1, e_2), 0 \leq e_2 < 2^p \), are each decomposable with bound set \( x_0 \) and free set \( x_1 \).

Curtis has stated a set of theorems relating simple nondisjunctive decompositions to complex ones. All these theorems are in most cases immediate generalizations of the corresponding theorems by Ashenurst.

All the types of decompositions quoted in fig. 1 have been covered by the above theorems, except the improper complex decomposition. A general theory for improper decomposition is, until now, unavailable. However, some particular improper decompositions have been considered. Curtis stated the following theorem.

Theorem 3

Let \( x_0, x_1 \) be a partition on \( x \). A switching function \( f \) possesses an improper decomposition given by \( F[y_0(x_0), \ldots, y_{k-1}(x_0), x_1] \) if and only if the decomposition chart, with the variables in \( x_0 \) defining the columns and the variables in \( x_1 \) defining the rows, has column multiplicity \( \nu \leq 2^k \).

4. Fundamental theorems for simple decompositions

Theorem 4

Let \( x_0, x_1, x_2 \) be a partition on \( x \). A switching function \( f \) has a simple decomposition of the form \( F[y_0(x_0, x_2), x_1, x_2] \) if and only if

\[
\frac{df}{dx_i} = \frac{df}{dx_0} g_i(x_0, x_2), \quad \{x_i \mid x_i \in x_0\},
\]

where \( g_i \) is a function of \( x_0 \) and \( x_2 \) only.
Proof
Assume first that $x_2 = \phi$ (disjunctive decomposition). The conditions of theorem 1b on the columns of the decomposition chart imply that in that part of the decomposition chart characterized by $\Delta f/\Delta x_0 = 1$ the columns are necessarily a fixed pattern of 0's and 1's or its complement. As a consequence $\partial f/\partial x_i$ is there only a function of $x_0$. This states relation (6) for the disjunctive case. The proof of the theorem results then immediately from the conditions of theorem 2.

Corollary 1
Let $x_0, x, x_1$ be a partition on $x$. A switching function has a simple disjunctive decomposition of the form $F[y(x_0), x_1]$ if and only if

$$\frac{\partial f}{\partial x_i} = \frac{\Delta f}{\Delta x_0} g_i(x_0), \quad \{x_i | x_i \in x_0\}. \quad (7)$$

Corollary 2
The conditions (6) and (7) are equivalent to (8) and (9) respectively:

$$\frac{Sf}{Sx_0} = \frac{\Delta f}{\Delta x_0} g(x_0, x_2), \quad (8)$$

$$\frac{Sf}{Sx_0} = \frac{\Delta f}{\Delta x_0} g(x_0), \quad (9)$$

with $g = \sum g_i$.

The condition (6) directly leads to the Akers formula. Indeed, it can easily be shown that if $f(x)$ effectively depends on each of its variables (that is if $\partial f/\partial x_i \neq 0$) then

$$g_i = \frac{\Delta}{\Delta x_1} \left( \frac{\partial f}{\partial x_i} \right) + \left[ \frac{\Delta}{\Delta x_1} \left( \frac{\partial f}{\Delta x_0} \right) \right]' \quad (10)$$

In view of (10) and (6) one obtains:

$$\frac{\partial f}{\partial x_i} = \frac{\Delta f}{\Delta x_0} \left\{ \frac{\Delta}{\Delta x_1} \left( \frac{\partial f}{\partial x_i} \right) + \left[ \frac{\Delta}{\Delta x_1} \left( \frac{\partial f}{\Delta x_0} \right) \right]' \right\}, \quad (11)$$

which is equivalent to the Akers expression

$$\frac{\Delta}{\Delta x_1} \left( \frac{\partial f}{\partial x_0} \right) \frac{\partial f}{\partial x_i} = \frac{\Delta f}{\Delta x_0} \frac{\Delta}{\Delta x_1} \left( \frac{\partial f}{\partial x_i} \right) \quad (12)$$
On the other hand, by derivating the expression (6) with respect to $x_k \in x_1$ one obtains successively:

$$\frac{\partial f}{\partial x_k x_i} = \frac{\partial}{\partial x_k} \left( \frac{\Delta f}{\Delta x_0} \right) g_i,$$

$$\frac{\partial f}{\partial x_j} \frac{\partial f}{\partial x_i x_k} = \frac{\Delta f}{\Delta x_0} \frac{\partial}{\partial x_k} \left( \frac{\Delta f}{\Delta x_0} \right) g_i g_j, \quad (x_i, x_j \in x_0, x_k \in x_1),(13)$$

$$\frac{\partial f}{\partial x_j} \frac{\partial f}{\partial x_i x_k} = \frac{\partial f}{\partial x_i} \frac{\partial f}{\partial x_j x_k}.$$

Relation (13) constitutes the necessary conditions for disjunctive decomposition by Shen, McKellar and Weiner $^5)$. The Akers and the Shen theorems, initially proved only for the disjunctive case, remain thus valid for the nondisjunctive one.

The following expansion has been established by the author $^4)$:

$$f(x) = f(e) \oplus \sum_{i=0}^{n-1} x_i^{(e_i')} \left( \frac{\partial f}{\partial x_i} \right)_{x_j=e_j} \quad , \quad j > i, \quad e = (e_0, \ldots, e_{n-1}). \quad (14)$$

Let us assume that a simple decomposition holds for $f$ so that the conditions (6) are satisfied. From (6) and (14) one deduces:

$$f = \sum_{x_i=x_0} x_i^{(e_i')} \left( \frac{\partial f}{\partial x_i} \right)_{x_j=e_j} \oplus f(e) \oplus \sum_{x_i=x_1, x_2} x_i^{(e_i')} \left( \frac{\partial f}{\partial x_i} \right)_{x_j=e_j}$$

$$= \frac{\Delta f}{\Delta x_0} \left[ \sum_{x_i=x_0} x_i^{(e_i')} (g_i)_{x_j=e_j} \right] \oplus f(e_0, x_1, x_2), \quad (15)$$

with $e_0$ a vector of fixed values for $x_0$. The following theorem then holds.

**Theorem 5**

If $f(x) = F[y(x_0, x_2), x_1, x_2]$ then

$$F = A(x_1, x_2) B (x_0, x_2) \oplus C(x_1, x_2), \quad (16)$$

with

$$A = \frac{\Delta f}{\Delta x_0},$$

$$B = \sum x_i^{(e_i')} (g_i)_{x_j=e_j}, \quad x_i \in x_0,$$

$$C = f(e_0, x_1, x_2).$$
Corollary

If \( f(x) \) is simply decomposable,

- a disjunctive decomposition \( f = A' + B' \) exists if and only if \( f(e_0, x_1, x_2) = 1 \);
- a conjunctive decomposition \( f = A B \) exists if and only if \( f(e_0, x_1, x_2) = 0 \);
- an exclusive-OR decomposition \( f = A \oplus B \) exists if and only if \( \Delta f/\Delta x_0 = 1 \).

The conditions of theorem 4 will now be used to build an algorithm for the simple decomposition of Boolean functions. This however requires that the set of \( \Delta \)-operators of a given function is easily available. Computational expressions for the \( \Delta \)-operator will be obtained in the next section.

5. The computation of the \( \Delta \)-operator

Let us consider the representation of a Boolean function by means of the set of all its prime implicants \( p_i \) and by means of the set of all its prime implicates \( q_j \); one has

\[
f(x) = \sum_{p_i} p_i = \prod_{q_j} q_j.
\]  

(17)

If \((x_0, x_1)\) is a partition of \( x \), let us define \( p_{x_1} \) and \( q_{x_1} \) as follows:

\[
p_{x_1} = \sum_k p_k, \quad \{p_k | p_k \text{ does not contain } x_1 \text{ for each } x_i \in x_1\},
\]

\[
q_{x_1} = \prod_k q_k, \quad \{q_k | q_k \text{ does not contain } x_1 \text{ for each } x_i \in x_1\}.
\]

(18)

The following theorem holds ((\( p' q \)\)\(_{x_1}\) holds for \((p_{x_1})' q_{x_1}\)):

**Theorem 6**

\[
\Delta f/\Delta x_1 = (p' q)_{x_1}.
\]

(19)

**Proof**

(a) \((p' q)_{x_1}\) is independent of \( x_1 \) (obvious).

(b) \((p' q)_x\) is 0 or 1 and is 0 if and only if \( f \) is itself 0 or 1. Indeed, \( f = 0 \) if and only if \( q_x = 0 ; f = 1 \) if and only if \( p_x = 1 \).

(c) If \( e_0 \) is a fixed value of \( x_0 \), then \([p' q]_{x_1} \big|_{x_0=e_0} \) is 0 or 1 and is 0 if and only if \( f(e_0, x_1) \) is itself 0 or 1.

This immediately results from points 1 and 2 and from the fact that all the prime implicants and all the prime implicates of \( f \) are considered.

(d) The proof of the theorem then immediately results from the properties of the \( \Delta \)-operator derived by Akers 3).

**Corollary**

\[
\partial f/\partial x_i = (p' q)_{x_i}.
\]

(20)
The formal expression of theorem 6 will now slightly be modified in order to allow the building of an iterative computation scheme for the set of $\Delta$-operators. Let us define $\hat{p}_{x_1}$ and $\hat{q}_{x_1}$ as follows:

$$\hat{p}_{x_1} = \sum_k p_k \{ p_k | p_k \text{ depends on all the variables of } x_0 \text{ and only of these variables} \};$$

$$\hat{q}_{x_1} = \prod_k q_k \{ q_k | q_k \text{ depends on all the variables of } x_0 \text{ and only of these variables} \}. \quad (21)$$

The following relation directly comes from theorem 6 and expressions (21):

$$\frac{\Delta f}{\Delta x_i} = \prod_{x_j \in x_0} \frac{\Delta f}{\Delta x_1 x_i} (\hat{p}' \hat{q})_{x_1}. \quad (22)$$

Expression (22) allows us to compute immediately the $\Delta$-operators of order $p$ when the $\Delta$-operators of order $(p + 1)$ and the prime implicates and implicants of $f$ are known. Since $\Delta f/\Delta x = 0$ or 1 all the $\Delta$-operators may then trivially be computed when starting with the $\Delta$-operator of the highest order. Let us also note that this computation scheme requires using each of the $\hat{p}_{x_1}$ and $\hat{q}_{x_1}$ once and only once. Let us finally note that several good programs exist which determine all the prime implicants and prime implicates of Boolean functions 6). This achieves a very fast computation scheme for the $\Delta$-operators.

6. The basic algorithm

Let us first deal with the disjunctive case.

(1) Compute the set of $\Delta$-operators of the function $f$ to be tested by applying relation (22). Since one is only interested in nontrivial decompositions, that is, decompositions whose bound set contains at least two variables, only the $\Delta$-operators of order $\leq n - 2$ ($n = \text{number of variables in } f$) are to be computed.

(2) Candidate bound sets can firstly be eliminated by using the following necessary condition which directly derives from theorem 4: If $\Delta f/\Delta x_0 = h_0(x_0)$ and $\sum \delta f/\delta x_i = h_1(x_0)$, $x_i \in x_0$, then $x_0$ may constitute a bound set of a possible decomposition only if $x_0 \setminus x_i \in x_0$.

(3) Verify if the conditions of theorem 4 are satisfied for the remaining sets of variables. This verification can be performed e.g. by using one of the two following computation schemes.

(a) The $\Delta$-operators are naturally obtained by the intermediate of step 1 as a product of prime implicants. One has

$$\frac{\partial f}{\partial x_i} = \frac{\Delta f}{\Delta x_0} g_i. \quad (23)$$
The prime implicants of $A_i/Dx_0$ and of $g_i$ are obtained from their conjunctive forms respectively. The prime implicants of $g_i$ are each the product of two monomials $A_i(x_1)$ and $B_i(x_0)$, $A_i(x_1)$ being the product of variables $x_j \in x_1$ and $B_i(x_0)$ being the product of variables $x_j \in x_0$. The conditions of theorem 4 become then

$$\frac{\Delta f}{\Delta x_0} \left[ \sum_i A_i(x_1) \right] = 0,$$

which can be trivially verified.

(b) Obtain the $\delta f/\delta x_i$ as a sum of prime implicants. Again each of the prime implicants is a product of two monomials $A_i(x_1)$ and $B_i(x_0)$ (see (a)). The conditions of theorem 4 become then

$$\sum_i \hat{B}_i(x_0) \leq \frac{\delta f}{\delta x_i} + \left( \frac{\Delta f}{\Delta x_0} \right).$$

Again the form of the $A$-operators obtained after step 1 allows a trivial verification of condition (25).

Example

The $\hat{p}_e$ and $\hat{g}_e$ are explicitly indicated in the literal form of $f$:

$$f = x_1' x_5' (\hat{p}_{2346}) + x_1' x_6 (\hat{p}_{2345}) + x_2' x_3' x_5' (\hat{p}_{146}) + x_2' x_3' x_6 (\hat{p}_{145}) + x_2' x_4' x_5' (\hat{p}_{136}) + x_2' x_4' x_6 (\hat{p}_{135}) + x_2 x_3 x_4 x_5' (\hat{p}_{16}) + x_2 x_3 x_4 x_6 (\hat{p}_{15}) + x_1 x_2 x_3' x_5' x_6' (\hat{p}_4) + x_1 x_2 x_4' x_5' x_6' (\hat{p}_3) + x_1 x_2' x_3 x_4 x_5' x_6' (\hat{p}_0) =$$

$$= (x_1 + x_5' + x_6) (\hat{g}_{234}) (x_2 + x_3 + x_5' + x_6) (\hat{g}_{14}) (x_2 + x_4 + x_5' + x_6) (\hat{g}_{13}) \times$$

$$\times (x_1' + x_2' + x_3 + x_5) (\hat{g}_{46}) (x_1' + x_2' + x_3 + x_6) (\hat{g}_{45}) \times$$

$$\times (x_1' + x_2' + x_4 + x_5) (\hat{g}_{36}) (x_1' + x_2' + x_4 + x_6) (\hat{g}_{35}) \times$$

$$\times (x_2' + x_3' + x_4' + x_5' + x_6) (\hat{g}_i) (x_1' + x_2 + x_3' + x_4' + x_5) (\hat{g}_6) \times$$

$$\times (x_1' + x_2 + x_3' + x_4' + x_6') (\hat{g}_5).$$

In table I, the $A$-operators obtained after step 1 were more simply denoted $A(e)$. The following set of variables satisfy the conditions of theorem 4:

$$(x_1, x_2, x_3, x_4), \ (x_2, x_3, x_4), \ (x_3, x_4), \ (x_5, x_6).$$

In view of (27) and of the theorems by Ashenurst the function $f$ can be designed as follows:

$$f = F[y_1(x_1, x_2, y_2(x_3, x_4)), y_3(x_5, x_6)],$$

$$F = y_1 \oplus y_3,$$

$$y_1 = x_1 (x_2 \oplus y_2),$$

$$y_2 = x_3 x_4,$$

$$y_3 = x_5' + x_6.$$
Let us now examine the nondisjunctive case. The same basic algorithm as for the disjunctive case may be used except that point (2) (relative to the necessary condition) must be dropped. In addition, the conditions (24) and (25) are also too strong. For example, condition (24) should be substituted by the following one.

If the $A_i$ which do not satisfy condition (24) are only functions of $X_1 \in X_1'$, then a nondisjunctive decomposition of the form

$$f = F[y(x_0, x_{11}), x_1]$$

holds. Let us again consider the preceding example. It can easily be verified that the following nondisjunctive decompositions hold, e.g.

$$f = F[y(x_1, x_2, x_3, x_4), x_2, x_4, x_5, x_6],$$

$$f = F[y(x_1, x_2, x_5, x_6), x_1, x_3, x_4, x_6].$$

### Table I

<table>
<thead>
<tr>
<th>$e$</th>
<th>$\Delta(e)$</th>
<th>$e$</th>
<th>$\Delta(e)$</th>
<th>$e$</th>
<th>$\Delta(e)$</th>
<th>$e$</th>
<th>$\Delta(e)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1234</td>
<td>1</td>
<td>123</td>
<td>1</td>
<td>12</td>
<td>1</td>
<td>1</td>
<td>$x_2 \oplus x_3 x_4$</td>
</tr>
<tr>
<td>1235</td>
<td>1</td>
<td>124</td>
<td>1</td>
<td>13</td>
<td>$x_2 + x_4$</td>
<td>2</td>
<td>$x_1$</td>
</tr>
<tr>
<td>1236</td>
<td>1</td>
<td>125</td>
<td>1</td>
<td>14</td>
<td>$x_2 + x_3$</td>
<td>3</td>
<td>$x_1 x_4$</td>
</tr>
<tr>
<td>1245</td>
<td>1</td>
<td>126</td>
<td>1</td>
<td>15</td>
<td>$x_6' + (x_2 \oplus x_3 x_4)$</td>
<td>4</td>
<td>$x_1 x_3$</td>
</tr>
<tr>
<td>1246</td>
<td>1</td>
<td>134</td>
<td>1</td>
<td>16</td>
<td>$x_5 + (x_2 \oplus x_3 x_4)$</td>
<td>5</td>
<td>$x_6'$</td>
</tr>
<tr>
<td>1256</td>
<td>1</td>
<td>135</td>
<td>$x_2 + x_4 + x_6'$</td>
<td>23</td>
<td>$x_1$</td>
<td>6</td>
<td>$x_5$</td>
</tr>
<tr>
<td>1345</td>
<td>1</td>
<td>136</td>
<td>$x_2 + x_4 + x_5$</td>
<td>24</td>
<td>$x_1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1346</td>
<td>1</td>
<td>145</td>
<td>$x_2 + x_3 + x_6'$</td>
<td>25</td>
<td>$x_1 + x_6'$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1356</td>
<td>1</td>
<td>146</td>
<td>$x_2 + x_3 + x_5$</td>
<td>26</td>
<td>$x_1 + x_5$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1456</td>
<td>1</td>
<td>156</td>
<td>1</td>
<td>34</td>
<td>$x_1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2345</td>
<td>$x_1 + x_6'$</td>
<td>234</td>
<td>$x_1$</td>
<td>35</td>
<td>$x_6' + x_1 x_4$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2346</td>
<td>$x_1 + x_5$</td>
<td>235</td>
<td>$x_1 + x_6'$</td>
<td>36</td>
<td>$x_5 + x_1 x_4$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2356</td>
<td>1</td>
<td>236</td>
<td>$x_1 + x_5$</td>
<td>45</td>
<td>$x_6' + x_1 x_3$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2456</td>
<td>1</td>
<td>245</td>
<td>$x_1 + x_6'$</td>
<td>46</td>
<td>$x_5 + x_1 x_3$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3456</td>
<td>1</td>
<td>246</td>
<td>$x_1 + x_5$</td>
<td>56</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>256</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>345</td>
<td>$x_1 + x_6'$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>346</td>
<td>$x_1 + x_5$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>356</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>456</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
7. Conclusion

A fast algorithm for the computation of the set of $A$-operators of a Boolean function has been presented. This algorithm together with some new theorems on functional decomposition derived in this paper allow us to build some fast decomposition schemes for Boolean functions.

Acknowledgement

The author is indebted to Dr M. Davio for many interesting discussions during the preparation of this paper.

MBLE Research Laboratory

REFERENCES

6) G. Bioul, MBLE internal technical Note N75, 1971.