Correspondence

Multiple Stuck-Fault Detection and Location in Multivalued Linear Circuits

CHUEN-LIANG CHEN AND MIN-WEN DU

Abstract—**In this correspondence, we present procedures for constructing universal fault detection test sets as well as fault location test sets for multivalued linear circuits, under a multiple stuck-fault model. The bin packing problem is involved in the procedures. The sizes of the fault detection test set and the fault location test set constructed for an n-variable v-valued linear tree circuit are $1 + \lceil n/(v - 1) \rceil$ and $1 + \lceil n/\lfloor \log_2 v \rfloor \rceil$, respectively. It has been proved that the sizes listed above are optimal for some cases.**

Index Terms—**Bin packing problem, linear circuit, multivalued logic, stuck-fault detection/location, universal test set.**

I. Introduction

Recently, more and more researchers have focused their attention upon multivalued logic. It is now worth studying the reliability problem of multivalued circuits as we have done for 2-valued circuits. In this correspondence, we shall study the fault detection problem as well as the fault location problem in multivalued linear circuits.

A v-valued logic function $f(x_1, x_2, \cdots, x_n)$ is said to be linear if it can be expressed as

$$a_0 \oplus \underbrace{x_1 \oplus x_1 \oplus \cdots \oplus x_1}_{a_1 \text{ times}} \oplus \underbrace{x_2 \oplus x_2 \oplus \cdots \oplus x_2}_{a_2 \text{ times}} \oplus \cdots \oplus \underbrace{x_n \oplus x_n \oplus \cdots \oplus x_n}_{a_n \text{ times}}$$

where " $\oplus$ " is a mod-v-addition operator. For the sake of brevity, we shall express it as

$$(a_0 + a_1x_1 + a_2x_2 + \cdots + a_nx_n) \bmod v$$

in the remainder of this correspondence. A v-valued logic circuit is said to be linear if it realizes a v-valued linear function. It is known that a 2-valued linear circuit usually consists of some mod-2-adders (EXCLUSIVE-OR gates). Similarly, we can construct a v-valued linear circuit by using mod-v-adders.

We say that a line of a v-valued circuit has a stuck-at-s fault, $0 \leq s \leq v - 1$, if this line generates a constant value s for all input combinations. Here, we shall consider multiple stuck fault model. That is, some input/output lines of mod-v-adders may have stuck faults.

II. Parameter Vector

In this section, the characteristics of a linear circuit will be utilized to form a parameter vector which is to be used in constructing test sets.

Assume that C is a fault-free v-valued linear circuit with n input lines and one output line; these input lines are driven by variables $x_1, x_2, \cdots, x_n$.

Definition 1: The *parameter vector* of C, denoted as $P(C)$, is an $(n + 1)$-tuple $(a_0, a_1, \cdots, a_n)$, where:

1) $a_0 = 0$;[1]

2) a_i = the number of paths from input-line i to the output line, $1 \leq i \leq n$.

Let $f_C(X)$, $X = (x_1, x_2, \cdots, x_n)$, denote the function realized on circuit C. We can get the following lemma straightforwardly.

Lemma 1: If $P(C) = (a_0, a_1, \cdots, a_n)$, then

$$f_C(X) = (a_0 + a_1x_1 + a_2x_2 + \cdots + a_nx_n) \bmod v.$$

Now, assume that m lines of circuit C, denoted as stuck-lines 1, 2, $\cdots$, m, are, respectively, stuck at values $s_1, s_2, \cdots, s_m$ where $0 \leq s_i \leq v - 1$ for $1 \leq i \leq m$. Let C' denote this faulty circuit.

Definition 2: The *parameter vector* of a possibly faulty circuit C', denoted as $P(C')$, is an $(n + 1)$-tuple $(a_0', a_1', \cdots, a_n')$, where:

1) $a_0' = (\Sigma_{i=1}^{m} b_i' s_i) \bmod v$;

2) a_i' = the number of paths from input-line i to the output-line which do not pass through any stuck-line, $1 \leq i \leq n$;

3) b_i' = the number of paths from stuck-line i to the output-line which have no other stuck-fault except that at stuck-line i itself, $1 \leq i \leq m$.

Let $f_{C'}(X)$ denote the function realized on circuit C'. We note that it is still a linear function and also closely relates to $P(C')$.

Lemma 2: If $P(C') = (a_0', a_1', \cdots, a_n')$, then

$$f_{C'}(X) = (a_0' + a_1'x_1 + a_2'x_2 + \cdots + a_n'x_n) \bmod v.$$

From the above definitions, it can be proved that the parameter vector $P(C')$ needs to satisfy the following certain constraints:

Theorem 1: $0 \leq a_0' \leq v - 1$ and $0 \leq a_i' \leq a_i$ for $1 \leq i \leq n$.

Example 1: Consider the 3-valued linear circuit C shown in Fig. 1(a). Its parameter vector $P(C)$ is (0, 1, 4, 1, 1). Here, $a_2 = 4$ because there are four paths from line b to line m, they are

$$b \rightarrow c \rightarrow e \rightarrow f \rightarrow i \rightarrow m \tag{1}$$

$$b \rightarrow c \rightarrow e \rightarrow g \rightarrow j \rightarrow m \tag{2}$$

$$b \rightarrow d \rightarrow e \rightarrow f \rightarrow i \rightarrow m \tag{3}$$

$$b \rightarrow d \rightarrow e \rightarrow g \rightarrow j \rightarrow m. \tag{4}$$

Now, assume that both lines d and j are stuck at 2. Let C' denote the faulty circuit; its equivalent fault-free circuit is shown in Fig. 1(b). Name lines d and j as stuck-lines 1 and 2. Then, $P(C')$ can be derived as

1) $b_1' = 1$ because there are two paths from line d to line m, i.e., $d \rightarrow e \rightarrow f \rightarrow i \rightarrow m$ and $d \rightarrow e \rightarrow g \rightarrow j \rightarrow m$, but the second path has other stuck fault on line j;

2) $b_2' = 1$;

3) $a_0' = (1 \times 2 + 1 \times 2) \bmod 3 = 1$;

4) $a_2' = 1$ because paths (2), (3), and (4) are stuck;

5) $a_3' = 0$ because the only possible path $h \rightarrow j \rightarrow m$ is stuck;

6) $a_1' = a_4' = 1$.

Totally speaking, $P(C') = (1, 1, 1, 0, 1)$.

We can now formulate the fault detection problem and fault

Manuscript received June 29, 1984; revised November 10, 1985, and February 14, 1986.

C.-L. Chen was with the Institute of Computer Engineering, National Chiao-Tung University, Hsinchu, Taiwan. He is now with the Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan 10764.

M.-W. Du is with the Department of Computer Science, Worcester Polytechnic Institute, Worcester, MA 01609, on leave from the Institute of Computer Engineering, National Chiao-Tung University, Hsinchu, Taiwan 30049.

IEEE Log Number 8610489.

[1] When some input lines of the circuit concerned are driven by constants, a_0 maybe not be equal to 0. We do not consider this case here. In fact, it can be shown that this feature will not affect the construction of test sets.

0018-9340/86/1200-1068$01.00 © 1986 IEEE

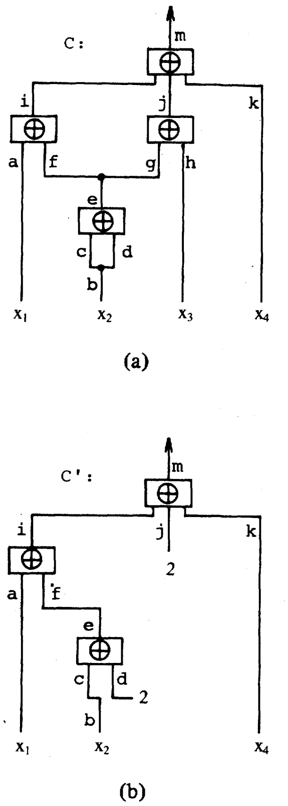

Fig. 1. Example of circuits.

location problem by using the notation of parameter vectors. Such formulations will isolate our problems from the structures of the circuits. Hence, the test sets constructed by our procedures will be universal test sets [1]. In the following discussion, we will use the "congruence" notation. We say that a is congruent to a' modulo v, denoted as $a \equiv a'$ (modulo v), if (a mod v) = (a' mod v). And, $P(C)$ is congruent to $P(C')$ modulo v, denoted as $P(C) \equiv P(C')$ (modulo v), if $a_i \equiv a_i'$ (modulo v) for $0 \le i \le n$.

Originally, the fault detection problem is to examine whether $f_{C'}(X) = f_C(X)$. Because $f_C(X)$ and $f_{C'}(X)$ closely relate to $P(C)$ and $P(C')$ (stated in Lemmas 1 and 2), we may roughly imagine the fault detection problem as determining whether $P(C') = P(C)$. However, sometimes a faulty circuit may produce an undetectable fault with respect to the fault-free function. For example, considering circuit C shown in Fig. 1(a), $P(C')$ will be (0, 1, 1, 1, 1) if both lines d and g are stuck at 0. We see that $P(C') \ne P(C)$ but C and C' in fact realize the same function. This is due to the fact that $P(C') \equiv P(C)$ (modulo 3). In order to take care of this situation, the fault detection problem needs to be formulated as "*to examine whether $P(C) \equiv P(C')$ (modulo v).*"

As to the fault location problem, originally it is to determine which fault class the circuit under test C' belongs to, i.e., to determine what $f_{C'}(X)$ is. Similarly, we may imagine the fault location problem as to see what $P(C')$ is. However, sometimes two different faults may be undistinguishable. For example, considering circuit C again, assume that lines d, g, and h of faulty circuit C' are all stuck at 0, while line h of faulty circuit C'' is stuck at 0. It is easy to see that $P(C') \ne P(C'')$ but they are in fact the same in the sense of "congruence." Hence, the fault location problem needs to be formulated as "*to determine which parameter vector $P(C')$ is congruent to.*"

III. Fault Detection Procedure

In this section, we shall use parameter vectors to solve the fault detection problem. The test sets can be obtained by solving a bin packing problem [2].

The so-called bin packing problem is, given the bin capacity $B > 0$ and a set E of "weighted elements" (the weight of each element is between 0 and B), to find a partition of E into disjoint subsets E_1, E_2, $\cdots$, E_N such that the sum of the weights of the elements in each E_i is not more than B and such that N is as small as possible.

Let a_i and a_i' be defined as in Definitions 1 and 2.

Lemma 3: When $\Sigma_{i=1}^k a_i \le v - 1$, if

$$f_{C'}(\underbrace{1, \cdots, 1}_{k \text{ times}}, 0, \cdots, 0) = \left[a_0' + \left(\sum_{i=1}^k a_i\right)\right] \bmod v$$

then $a_i' \equiv a_i$ (modulo v) for $1 \le i \le k$.

Proof: If $a_j' \not\equiv a_j$ (modulo v) for a certain $1 \le j \le k$, then $0 \le \Sigma_{i=1}^k a_i' < \Sigma_{i=1}^k a_i$ because of Theorem 1, and then $\Sigma_{i=1}^k a_i' \not\equiv \Sigma_{i=1}^k a_i$ (modulo v) because $\Sigma_{i=1}^k a_i \le v - 1$. Therefore,

$$f_{C'}(\underbrace{1, \cdots, 1}_{k \text{ times}}, 0, \cdots, 0) = \left[a_0' + \left(\sum_{i=1}^k a_i'\right)\right] \bmod v$$

$$\ne \left[a_0' + \left(\sum_{i=1}^k a_i\right)\right] \bmod v. \quad \text{Q.E.D.}$$

Lemma 4: When $a_1 > v - 1$, if $f_{C'}(1, 0, 0, \cdots, 0) = (a_0' + a_1)$ mod v then $a_1' \equiv a_1$ (modulo v).

Proof: If $a_1' \not\equiv a_1$ (modulo v), then $f_{C'}(1, 0, 0, \cdots, 0) = (a_0' + a_1')$ mod $v \ne (a_0' + a_1)$ mod v. Q.E.D.

In fact, the proposition of Lemma 4 "$a_1 > v - 1$" is unnecessary. However, we still write it down to emphasize the different opportunities when we adopt Lemmas 3 or 4.

Let $\hat{a}_i = \min(a_i, v - 1)$ for $1 \le i \le n$; and let $\delta(C)$ denote the size of the optimal fault detection test set for a v-valued linear circuit C.

Theorem 2: $\delta(C) \le N + 1$, if $\{\hat{a}_1, \hat{a}_2, \cdots, \hat{a}_n\}$ can be partitioned into N disjoint subsets, say $\{\hat{a}_{1(=k_0+1)}, \hat{a}_2, \cdots, \hat{a}_{k_1}\}$, $\{\hat{a}_{k_1+1}, \hat{a}_{k_1+2}, \cdots, \hat{a}_{k_2}\}$, $\cdots$, $\{\hat{a}_{k_{N-1}+1}, \hat{a}_{k_{N-1}+2}, \cdots, \hat{a}_{n(=k_N)}\}$ after renaming all $\hat{a}_i$, such that

$$\sum_{i=k_{j-1}+1}^{k_j} \hat{a}_i \le v - 1 \quad \text{for } 1 \le j \le N.$$

Proof: A possible test set with size $N + 1$ is $\{Y_j | 0 \le j \le N\}$, where

1) $Y_0 = (0, 0, \cdots, 0)$;

2) $Y_j = (0, \cdots, 0, \underset{\substack{\uparrow \\ (k_{j-1}+1)\text{-th}}}{1}, \cdots, \underset{\substack{\uparrow \\ k_j\text{-th}}}{1}, 0, \cdots, 0)$ for $1 \le j \le N$.

First, because $f_{C'}(Y_0) = a_0'$, we use Y_0 to see what a_0' is. Then, we use other Y_j to examine whether $a_i' \equiv a_i$ (modulo v) for $1 \le i \le n$, according to Lemmas 3 and 4. Q.E.D.

Theorem 2 suggests a procedure for constructing fault detection test sets. Surely, we will get a smaller test set if we can partition $\{\hat{a}_1, \hat{a}_2, \cdots, \hat{a}_n\}$ in as few subsets as possible. In fact, this is the bin packing problem. Bin packing problem has been proved to be NP-hard [2]. However, some heuristic methods for solving it have been proposed and were summarized in [2]. We can use these well-developed methods to help our constructions of the test sets.

Procedure 1:

Input: An n-variable v-valued linear circuit C, whose parameter vector $P(C) = (a_0, a_1, \cdots, a_n)$.

Output: A multiple stuck fault detection test set S for this circuit C.

Step 1: Solve the following bin packing problem: the bin capacity $B = v - 1$; the set of elements $E = \{e_1, e_2, \cdots, e_n\}$ where the weight of e_i is $w(e_i) = \hat{a}_i = \min(a_i, v - 1)$, $1 \le i \le n$. Assume that the solution is $\{E_j | 1 \le j \le N\}$.

Step 2: Let $S = \{(0, 0, \cdots, 0)\}$.

Step 3: For each E_j, $1 \le j \le N$, add n-tuple $(y_1, y_2, \cdots, y_n)$ into S where

$$y_i = \begin{cases} 0 & \text{if } e_i \notin E_j; \\ 1 & \text{if } e_i \in E_j. \end{cases}$$

Example 2: Consider a 10-valued linear circuit C with $P(C) = (0, 2, 3, 3, 4, 12, 2)$. Then, the associated bin packing problem is to pack the set of elements $E = \{e_1, e_2, e_3, e_4, e_5, e_6\}$ with weights $\{2, 3, 3, 4, 9, 2\}$ (note that $w(e_5) = \hat{a}_5 = \min(12, 9) = 9$) into some bins with capacity $B = 9$. One of the optimal partitions is as follows: $E_1 = \{e_1, e_4, e_6\}$, $E_2 = \{e_2, e_3\}$ and $E_3 = \{e_5\}$. Thus,

$$\{(0, 0, 0, 0, 0, 0), (1, 0, 0, 1, 0, 1), (0, 1, 1, 0, 0, 0), (0, 0, 0, 0, 1, 0)\}$$

will be a test set.

IV. Fault Location Procedure

Now, let us discuss how to construct fault location test sets. We derive two lemmas first. The first lemma is related to the "mixed radix number system" [3]. It is known that any integer I, $0 \le I \le (\Pi_{i=1}^{k} r_i) - 1$, can be uniquely represented as a k-tuple $(d_1, d_2, \cdots, d_k)$ with respect to a set of radices $\{r_1, r_2, \cdots, r_k\}$ where $I = d_1 + d_2 r_1 + d_3 r_1 r_2 + \cdots + d_k r_1 r_2 \cdots r_{k-1}$ and $0 \le d_i \le r_i - 1$ for $1 \le i \le k$.

Lemma 5: When $\Pi_{i=1}^{k}(a_i + 1) \le v$, if $f_{C'}(R_1, \cdots, R_k, 0, \cdots, 0) = [a_0' + (\Sigma_{i=1}^{k} d_i R_i)] \bmod v$ where $R_i = \Pi_{j=1}^{i-1}(a_j + 1)$ and $0 \le d_i \le a_i$ for $1 \le i \le k$, then $a_i' \equiv d_i$ (modulo v) for $1 \le i \le k$.

Proof: We prove it by contradiction. We know that $0 \le d_i \le a_i \le v - 1$ and $0 \le a_i' \le a_i \le v - 1$, by the proposition and Theorem 1. Hence, if $a_j' \not\equiv d_j$ (modulo v) for a certain $1 \le j \le k$, then $a_j' \ne d_j$, and then $\Sigma_{i=1}^{k} a_i' R_i \ne \Sigma_{i=1}^{k} d_i R_i$ because of the property of mixed radix number system.

And, it is easy to derive that $\Sigma_{i=1}^{k} a_i R_i = [\Pi_{i=1}^{k}(a_i + 1)] - 1 \le v - 1$. Thus, $0 \le \Sigma_{i=1}^{k} a_i' R_i \le v - 1$ and $0 \le \Sigma_{i=1}^{k} d_i R_i \le v - 1$. Combining the result of the above paragraph, we know that $\Sigma_{i=1}^{k} a_i' R_i \not\equiv \Sigma_{i=1}^{k} d_i R_i$ (modulo v) if $a_j' \not\equiv d_j$ (modulo v) for a certain $1 \le j \le k$. Therefore, $f_{C'}(R_1, R_2, \cdots, R_k, 0, \cdots, 0) = [a_0' + (\Sigma_{i=1}^{k} a_i' R_i)] \bmod v \ne [a_0' + (\Sigma_{i=1}^{k} d_i R_i)] \bmod v$. Q.E.D.

Lemma 5 illustrates the kernel idea of our procedure. We can interpret it as that all values of a_i', $1 \le i \le k$, can be uniquely determined in the sense of congruence when we know the values of a_0' and $f_{C'}(R_1, R_2, \cdots, R_k, 0, \cdots, 0)$.

Lemma 6: When $a_1 > v - 1$, if $f_{C'}(1, 0, 0, \cdots, 0) = (a_0' + d) \bmod v$ where $0 \le d \le v - 1$, then $a_1' \equiv d$ (modulo v).

Proof: It can be proved as Lemma 4. Q.E.D.

Let $\lambda(C)$ denote the size of the optimal fault location test set for a v-valued linear circuit C.

Theorem 3: $\lambda(C) \le M + 1$, if $\{\hat{a}_1, \hat{a}_2, \cdots, \hat{a}_n\}$ can be partitioned into M disjoint subsets, say $\{\hat{a}_{1(=k_0+1)}, \hat{a}_2, \cdots, \hat{a}_{k_1}\}$, $\{\hat{a}_{k_1+1}, \hat{a}_{k_1+2}, \cdots, \hat{a}_{k_2}\}, \cdots, \{\hat{a}_{k_{M-1}+1}, \hat{a}_{k_{M-1}+2}, \cdots, \hat{a}_{n(=k_M)}\}$ after renaming all $\hat{a}_i$, such that:

$$\prod_{i=k_{j-1}+1}^{k_j} (\hat{a}_i + 1) \le v \quad \text{for } 1 \le j \le M. \tag{5}$$

Proof: A possible test set with size $M + 1$ is $\{Z_j | 0 \le j \le M\}$ where

1) $Z_0 = (0, 0, \cdots, 0)$

2) $Z_j = (0, \cdots, 0, \underset{\substack{\uparrow \\ (k_{j-1}+1)-th}}{R_{k_{j-1}+1}}, R_{k_{j-1}+2}, \cdots, \underset{\substack{\uparrow \\ k_j-th}}{R_{k_j}}, 0, \cdots, 0)$, $1 \le j \le M$,

with

$$R_i = \prod_{h=k_{j-1}+1}^{i-1} (\hat{a}_h + 1) \quad \text{for } k_{j-1} + 1 \le i \le k_j.$$

We use Z_0 to see what a_0' is; then use other Z_j's to determine which values the other a_i''s are congruent to, according to Lemmas 5 and 6. Q.E.D.

As does Theorem 2, Theorem 3 suggests a procedure for constructing fault location test sets. This procedure is very similar to Procedure 1; we do not write it down explicitly here. Instead, we just point out their differences:

1) In Step 1, the encountered bin packing problem is based on

$$\sum_{i=k_{j-1}+1}^{k_j} \log(\hat{a}_i + 1) \le \log v$$

which is obtained by performing the "logarithm" operation on both sides of (5).

2) In Step 3, the construction of test is according to Theorem 3 not Theorem 2.

Example 3: Consider a 10-valued linear circuit C with $P(C) = (0, 1, 2, 1, 2, 15, 1)$. Then, the associated bin packing problem is to pack the set of elements $E = \{e_1, e_2, e_3, e_4, e_5, e_6\}$ with weights $\{\log 2, \log 3, \log 2, \log 3, \log 10, \log 2\}$ (note that $w(e_5) = \log(\hat{a}_5 + 1) = \log(\min(15, 9) + 1) = \log 10$) into some bins with capacity $B = \log 10$. We may partition E into $E_1 = \{e_1, e_3, e_6\}$, $E_2 = \{e_2, e_4\}$, and $E_3 = \{e_5\}$. Then, one of the possible test sets is

$$\{(0, 0, 0, 0, 0, 0), (1, 0, 2, 0, 0, 4), (0, 1, 0, 3, 0, 0), (0, 0, 0, 0, 1, 0)\}.$$

If the corresponding output values are 4, 9, 2, and 0, we can guarantee that $P(C') \equiv (4, 1, 2, 0, 2, 6, 1)$ (modulo 10), by solving the following.

$$f_{C'}(0, 0, 0, 0, 0, 0) = 4 = a_0'$$

$$f_{C'}(1, 0, 2, 0, 0, 4) = 9 = (a_0' + a_1' + 2a_3' + 4a_6') \bmod 10$$

$$f_{C'}(0, 1, 0, 3, 0, 0) = 2 = (a_0' + a_2' + 3a_4') \bmod 10$$

$$f_{C'}(0, 0, 0, 0, 1, 0) = 0 = (a_0' + a_5') \bmod 10.$$

V. A Special Case—Linear Tree Circuits

A linear circuit is called a linear tree circuit if it is constructed in a tree structure. In this section, we shall focus our attention upon linear tree circuits.

Let T be an n variable v-valued linear tree circuit. It is clear that each parameter a_i of $P(T)$ is always equal to 1 for $1 \le i \le n$. This fact simplifies the construction of test sets greatly because the involved bin packing problems in Theorems 2 and 3 degenerate to trivial cases. And, we have the following theorem.

Theorem 4:

$$\delta(T) \le 1 + \left\lceil \frac{n}{v-1} \right\rceil$$

and

$$\lambda(T) \le 1 + \left\lceil \frac{n}{\lfloor \log_2 v \rfloor} \right\rceil.$$

When v is 2 (i.e., considering 2-valued circuits), it can be seen that the test sets constructed by Fujiwara [4] and those constructed by us are the same. We may treat Fujiwara's methods as special cases of ours. In [4], Fujiwara also illustrated that $\delta(T) = \lambda(T) = 1 + n$ when $v = 2$. It seems that such optimality can be extended to other cases. In fact, we have examined that $\delta(T)$ is $1 + \lceil n/(v - 1)\rceil$ when $n \leq 2(v - 1)$. We strongly conjecture that $\delta(T)$ is exactly $1 + \lceil n/(v - 1)\rceil$ for all n and v.

As to $\lambda(T)$, we also can prove that $\lambda(T)$ is exactly $1 + \lceil n/\lfloor \log_2 v\rfloor\rceil$ for certain cases, particularly for the case that v is a power of 2.

Lemma 7: Consider linear tree circuit T. For any $(n + 1)$-tuple $(a_0', a_1', \cdots, a_n')$ such that:

1) $0 \leq a_i' \leq 1$ for $1 \leq i \leq n$

2) $\begin{cases} a_0' = 0 & \text{if } a_i' = 1 \text{ for all } 1 \leq i \leq n \\ 0 \leq a_0' \leq v - 1 & \text{otherwise} \end{cases}$

there exists a possibly faulty circuit T' such that

$$P(T') = (a_0', a_1', \cdots, a_n').$$

Proof: We can prove it by constructing a corresponding T' for each tuple. Q.E.D.

Theorem 5: $\lambda(T) = 1 + \lceil n/\lfloor \log_2 v\rfloor\rceil$, when T is a v-valued linear tree circuit and v is a power of 2.

Proof: A trivial lower bound of $\lambda(T)$ is $\lceil \log_v G\rceil$ where G is the number of all possible distinguishable fault classes. By Lemma 7, $G \geq v(2^n - 1) + 1$. Thus, we have the following derivations:

$$\begin{aligned} \lambda(T) &\geq \lceil \log_v [v(2^n - 1) + 1]\rceil \\ &= 1 + \left\lceil \log_v \left(2^n - 1 + \frac{1}{v}\right)\right\rceil \\ &= 1 + \left\lceil \frac{n - \epsilon}{\log_2 v}\right\rceil \quad [\text{where } 0 < \epsilon < 1] \\ &= 1 + \left\lceil \frac{n}{\lfloor \log_2 v\rfloor}\right\rceil \quad [\text{note that } \log_2 v = \lfloor \log_2 v\rfloor]. \end{aligned}$$

And, by Theorem 4, we know that $\lambda(T) = 1 + \lceil n/\lfloor \log_2 v\rfloor\rceil$. Q.E.D.

VI. Conclusion

We have discussed how to construct universal fault detection test sets and fault location test sets for any multivalued linear circuits, under multiple stuck-fault model. We also studied a special case—linear tree circuits. The sizes of the fault detection test set and the fault location test set constructed for an n variables v-valued linear tree circuit are $1 + \lceil n/(v - 1)\rceil$ and $1 + \lceil n/\lfloor \log_2 v\rfloor\rceil$, respectively. Besides, it has been proved that these sizes are optimal for some cases.

References

[1] S. B. Akers, Jr., "Universal test sets for logic networks," *IEEE Trans. Comput.*, vol. C-22, pp. 835–839, Sept. 1973.

[2] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA: Freeman, 1979, pp. 124–127.

[3] D. E. Knuth, *The Art of Computer Programming. Volume 2: Seminumerical Algorithms*, 2nd ed. Reading, MA: Addison-Wesley, 1981, pp. 179–197.

[4] H. Fujiwara, "On closedness and test complexity of logic circuits," *IEEE Trans. Comput.*, vol. C-30, pp. 556–562, Aug. 1981.

A Special-Function Unit for Sorting and Sort-Based Database Operations

L. RASCHID, T. FEI, H. LAM, AND STANLEY Y. W. SU

Abstract—**Achieving efficiency in database management functions is a fundamental problem underlying many computer applications. Efficiency is difficult to achieve using the traditional general-purpose von Neumann processors. Recent advances in microelectronic technologies have prompted many new research activities in the design, implementation, and application of database machines which are tailored for processing database management functions. To build an efficient system, the software algorithms designed for this type of system need to be tailored to take advantage of the hardware characteristics of these machines. Furthermore, special hardware units should be used, if they are cost-effective, to execute or to assist the execution of these software algorithms.**

In this correspondence, we present a special-function unit (SFU) which provides efficient support to a hardware sorting algorithm and other sort-based database operations. The main features of the SFU are the use of an automatic retrieval memory (ARM), which automatically reduces the memory space to only those "marked" locations that contain relevant data, and the parallel operation of several memory units within the SFU to support the sorting algorithm. The SFU-based sorting algorithm and the other sort-based database operations have an order of complexity N where N is the number of data words. The proposed hardware and software algorithms can be used to support complex, nonnumeric computational tasks found in many applications.

Index Terms—**Database machines, hardware sorter, sort-based algorithms for database operations, special-function processor.**

I. Introduction

In a computerized society, making decisions relies on efficient access to large databases. There are two fundamental problems in processing large databases, namely, the reduction of the search space and the efficient execution of frequent, primitive database operations.

A. Reduction of Search Space

A database is an integration of information relevant to a community of users. At any given time, a particular user is interested only in a portion or a "view" of a database. The external model is a mechanism for defining a view, and a retrieval query further reduces the size of the database defined by the view.

Several software techniques have been used to localize a subdatabase. One method of physically creating separate subdatabases for different applications has prohibitive time and space requirements. Another common method, which uses software indexing or clustering techniques to directly access portions of a large database, involves considerable software overhead.

Recent research in database machines has introduced several hardware techniques, namely, the cellular logic approach used in [10], [13], and [17], the database filter approach used in [1], [2], [9], and [20], and the cache memory approach used in [3], [4], and [14]. The time for staging data into main memory in these systems is still significant.

In our opinion, the ideal solution would be a memory that can *physically* "shrink" in size each time a subdatabase is established so that data elements in the subdatabase can be accessed directly and irrelevant data can be bypassed by the hardware, thus effectively shrinking the memory. This correspondence introduces a storage

Manuscript received July 15, 1984; revised April 15, 1986. This work was supported by the National Science Foundation, under Grant ECS-8402252.

The authors are with Database Systems Research and Development Center, University of Florida, Gainsville, FL 32611.

IEEE Log Number 8610490.

0018-9340/86/1200-1071$01.00 © 1986 IEEE