A Study of Theoretical Issues in the Synthesis of Delay Fault Testable Circuits [1]

Sreejit Chakravarty
226 Bell Hall
Dept. of Computer Science
State University of New York
Buffalo, NY 14260

(TECH. REPORT No. 93-38.
Revised Version of 92-31)

ABSTRACT

Several useful multilevel logic optimization transformations (MLOT), not known to preserve path delay fault testability (PDFT), are shown to indeed do so. We show that, while minimizing area, a number of MLOTs can also be used to "improve" PDFT! A sufficient condition for identifying PDFT unate circuits is presented. We show how these results can be used to: improve a known method for synthesizing PDFT circuits; and to prove the PDFT of designs not known to be so.

INDEX TERMS: Delay Fault Testable Circuits; Logic Optimization; Logic Synthesis; Testability Enhancing Tranformations; Testability Preserving Transformations.

1 Introduction

Delay testing attempts to verify the timing specifications of circuits. Two models for delay testing are: **path delay testing**[6, 10, 11, 14, 18]; and **gate delay testing**[6]. We assume *path delay testing* that uses the **path delay fault model** discussed below.

Along every physical path, from an input to an output of the combinational circuit, two distinct *transitions* - P_f, *input falling* and P_r, *input rising* - can propagate. These two transitions correspond to two delay faults (DFs). Delay testing ascertains if, for every physical path P, both P_f, P_r can propagate within a predetermined interval of time ΔT (a function of the operating clock rate).

[1]Research Supported by NSF Grant No. MIP-9102509.

Tests for DFs consists of a pair of input vectors $< T1, T2 >$. **Initialization vector** $T1$ is applied at time t_0. All circuit signals are allowed to stabilize under its influence. Then **test vector** $T2$ is applied at time t_1. Circuit outputs are sampled at time t_2, $t_2 - t_1 = \Delta T$.

A test for a single DF may not detect that DF if multiple DFs are present. **Robust tests** (**RTs**)[11, 18] for a DF, along path P, are tests not invalidated by DFs along other paths. RTs are classified depending on whether they are: hazard-free (**HF**) or not (**NHF**); and single path propagating (**SPP**) or multiple path propagating (**MPP**).

$T1, T2, T3, T4$ are defined in Table 1. $T =< T1, T2 >$ is an SPP-HF-RT for $< e, 9, 11, 12 >_f$ of Figure 1(a) because T propagates a transition only along $< e, 9, 11, 12 >$ (SPP) without any hazards. $< T3, T4 >$ is an MPP-HF-RT both for $< e, 7, 10, 12 >_f$ and $< e, 9, 11, 12 >_f$ because transitions propagate along both paths. The transition reaches y after the larger of the delays along these paths. C is **SPP-HF delay testable** (**MPP-HF delay testable**) or simply **SPP-HF Testable** (**MPP-HF Testable**) iff all DFs in C have an SPP-HF-RT (MPP-HF-RT).

An **atomic transformation**(**AT**) T transforms a circuit C into a functionally equivalent circuit C_T. ATs used in multilevel optimization systems include Inverter-Migration, Algebraic Resubstitution[2], Extended Factorization[14], transformations based on Dual Extraction[15]etc. All these ATs preserve *stuck-at testability*[7, 15].

Known synthesis methods for delay fault testable (**DFT**) circuits [1, 10, 13, 14] are area "inefficient". Question is: can known area optimization ATs be used to reduce the area of the initial DFT circuit ? More precisely, an AT is an SPP-HF (MPP-HF) **testability preserving transformation** (**TPT**) iff $\forall C$ if C is SPP-HF (MPP-HF) testable then so is C_T. If an atomic transformation is a SPP-HF or MPP-HF TPT then it can be used for the above purpose. Inverter Migration, Algebraic Resubstitution[5], Algebraic Resubstitution with "constrained" complement[5, 4] are SPP-HF TPTs whereas Extended Factorization[14] is an MPP-HF TPT.

We show that a number of other useful ATs[15] like *Division by Single Cube and its Complement* (**DSCC**), *Division by XOR and its complement* (**XORC**), *Division by D(2,2,3)*[2] *and its complement* (**D(2,2,3)C**) are SPP-HF TPTs. Note that XORC, D(2,2,3)C are two useful instances of *algebraic resubstitution with complement* that do not satisfy the "con-

[2]Abbreviation for a two cube, with the two terms containing two literals each and its support having three variables

straint" [4, 5] but are SPP-HF TPTs. These results help in the exploration of a much larger design space.

TPTs are useful when a DFT circuit is available. Often it isn't. In addition, a designer may not want to trade off "testability" with "area". We show that a number of area optimization ATs, if carefully applied, can be used to improve testability while reducing area. Use of ATs, like SR-1, to enhance testability[8] motivates the notion of **Testability Enhancing Transformations** (**TETs**) discussed in Section 2. We show that: *Division by Multiple Cube* (**DMC**)and DSCC are TETs; and *Division by Single Cube* (**DSC**), XORC and D(2,2,3)C are not TETs.

Understanding characteristics of DFT circuits is useful in enhancing known synthesis procedures and ascertain if known implementations are DFT. Such characteristics for two level circuits are known[10]. We present a sufficient condition that helps in identifying unate, multilevel circuits that are MPP-HF DFT. Our result is stronger than[10] in that it identifies multilevel unate MPP-HF DFT circuits. It is weaker than [10] in that it identifies conditions for MPP-HF DFT and not SPP-HF DFT circuits. Applications of these theoreticals results are discussed.

Fanout free circuits contain no fanout points. In **Tree Circuits**, **factored form**[2] or **Multilevel Expressions**(Figure1(a)) only primary inputs can fanout.

2 Testability Properties of ATs

The following definitions are from [2]. A **cube** is a collection of input literals like $\{a, b, \bar{c}\}$ denoting the product term $ab\bar{c}$. If a sum-of-product expression (SOP) consist of one cube it is a **single cube** else, it is a *multiple cube*(e.g. $a\bar{b} + \bar{a}b$).

The **support of SOP f** ($sup(f)$) is the set of variables in f. SOP f is **algebraic** iff no cube of f is contained in any other cube of f. Product of two SOPs FG is **algebraic** iff both F, G are *algebraic* and $sup(F) \bigcap sup(G) = \Phi$.

For two SOPs f, g **resubstitution** of g in f consists of expressing f as a function of g and the original inputs of f. For example, $f = a + bc, g = a + b, f = g(a + c)$ i.e. $f = pg + r$, p, r being SOPs. It is an **algebraic (Boolean) resubstitution** iff pg is (is not) an *algebraic product*. **Division** is rewriting f as $qp+r$. It is **algebraic** (**Boolean**) iff qp is an *algebraic*(*Boolean*) *product*. Rewriting f as $qp + \bar{p}s + r$ where p, q, s, r are SOPs is **Resubstitution**

with Complement. If both (either) $qp, \overline{p}s$ are (is not) *algebraic products* then it is **Algebraic (Boolean) Resubstitution with Complement**.

The general form of an AT (Figure 2) transforms C_a to C_b by replacing M_a with M_b. Let $B_1, \ldots, B_m$ be single cubes and $a_1, \ldots, a_t$ be input literals. Then, rewriting $a_1 \ldots a_t B_1 + \ldots + a_1 \ldots a_t B_m$ as: $A = a_1 \ldots a_t; E = AB_1 + \ldots + AB_m$ is *Division by Single Cube* (**DSC**) (Figures 3(a), (b)). SR-1[8] (Figures 3(a), (c)) rewrites E as: $A = a_1 \ldots a_t; E = A(B_1 + \ldots + B_m)$.

Let $A_1 = la_1 \ldots a_t; B_m = rb_1 \ldots b_s$; and $g = A_1B_1 + \ldots A_1B_m + \ldots + A_kB_1 + \ldots + A_kB_m$ where: $l, a_i's, b_j's, r$ are input literals and $A_i's, B_j's$ are single cubes. Rewriting g as: $c = A_1 + \ldots + A_k; d = B_1 + \ldots + B_m; g = cd$ is *Division by Multiple Cube* (**DMC**) (Figure 4).

For DSCC(Figure 5) M_a is defined as $g = a_1 \ldots a_k A + \overline{a_1}B + \ldots + \overline{a_k}B$, and M_b is defined as: $c = a_1 \ldots a_k; f = Ac + B\overline{c}$. By "pushing out NOTs"[5] we get $\overline{a_1 \ldots a_k}$ from $\overline{a_1} + \ldots + \overline{a_k}$. Thus DSCC an *algebraic resubstitution with "constrained" complement.*

For XORC (Figure 6) M_a is defined as $g = (a\overline{b} + \overline{a}b)A + (\overline{a}\overline{b} + ab)B$, and M_b is defined as: $k = \overline{b}a + \overline{a}b; f = Ak + B\overline{k}$. It is an *algebraic resubstitution with complement* not satisfying the "constraint", because $\overline{(a\overline{b} + \overline{a}b)}$ cannot be derived from $\overline{a}\overline{b} + ab$ by "pushing out NOTs".

For D(2,2,3)C (Figure 7) M_a is defined as $g = A\overline{a}b + Aac + B\overline{a}\overline{b} + B\overline{c}a$, and M_b is defined as: $k = ac + \overline{a}b; f = Ak + B\overline{k}$. It is an example of *algebraic resubstitution with complement* not satisfy the "constraint" because, $\overline{(\overline{a}b + ac)}$ cannot be derived from $\overline{a}\overline{b} + \overline{c}a$ by "pushing out NOTs".

First, properties of ATs in preserving DFT are discussed. Both C_a, C_b (Figure 2) contain **modified**, **unmodified** parts, and **modified paths**, **unmodified paths**. *Modified* (*Unmodified*) *paths* traverse only *modified (* unmodified *) part* of the circuit. Moreover, for every *unmodified path* q_1 in C_b $\exists$ an identical *unmodified path* $p_1 = I(q_1)$ in C_b. For a proof of Lemma 2.1, stated below, see Appendix II.

Lemma 2.1. For every *unmodified path* q_1 of C_b an input-pair $V =< V_1, V_2 >$ is an MPP-HF-RT (SPP-HF-RT) for a transition along q_1 iff V is an MPP-HF-RT (SPP-HF-RT) for the same transition along $p_1 = I(q_1)$.

Every *modified* path q_2 traversing M_b has 3 segments q_{21}, q_{22} and q_{23}. Every *modified* path p_2 traversing M_a also has 3 segments p_{21}, p_{22}, p_{23}. Segment p_{22} (q_{22}) that lie completely within M_a (M_b) is the **m-segment** of p_2 (q_2). For a proof of Lemma 2.2, stated below, see Appendix II. Theorem 2.3, also stated below, follows from Lemmas 2.1, 2.2.

Lemma 2.2. If C_b of Figure 2 is derived from C_a using either DSCC, XORC or D(2,2,3)C then for every *modified path* q_2 in C_b $\exists$ a *modified path* p_2 in C_a s.t. $V =< V1, V2 >$ is an MPP-HF-RT (SPP-HF-RT) for a transition along p_2 iff V is an MPP-HF-RT (SPP-HF-RT) for the same transition along q_2.

Theorem 2.3. DSCC, XORC, D(2,2,3)C are MPP-HF as well as SPP-HF TPTs.

A result along the lines of Theorem 2.3 is claimed in [16] and a very similar result was obtained in [3]. The proof sketch in [3] is different from the one we present here. Some of the derivations in our proofs are used to prove the rest of the results in the section.

Note that if an AT T is an SPP-HF TPT (MPP-HF TPT) then it does not imply that T is also an MPP-HF TPT (SPP-HF TPT). In addition, it is important to know if a TPT is an MPP-HF TPT or an SPP-HF TPT. This is because even if C_a of Figure 2(a) is not an SPP-HF DFT circuit it may be an MPP-HF DFT circuit. In this case we need to know if C_b of Figure 2(b) is at least an MPP-HF DFT circuit. From [3] it is not clear which of MPP or SPP "hazard free" RTs are being assumed. Thus, Theorem 2.3 is stronger than the result in [3] since we prove it for SPP-HF as well as MPP-HF RTs.

The next results show how some of these ATs can or cannot be used to enhance DFT of the given circuit. Let N_a (N_b) be the number of DFs in C_a (C_b) of Figure 2 that do not have a "hazard free" RT. The AT T, used to derive C_b from C_a, is a *Testability Enhancing Transformation* (**TET**) iff T is s.t.: (i) the number of paths in C_b is less than or equal to the number of paths in C_a; (ii) $N_b \leq N_a$; and (iii) when certain conditions are satisfied, $N_a > N_b$. Note that while considering TETs we do not worry if the "hazard free" RT is MPP or SPP. **SR-1** is a TET (Theorem 1 of [8]). We next show that some ATs are TETs while others are not. For a proof of Theorem 2.4, stated below, see Appendix III.

Theorem 2.4 DSC, XORC, D(2,2,3)C are not TETs.

For DMC, DSCC assume that $A_1, \ldots, A_k$, $B_1, \ldots, B_m$ are *single cubes* s.t. $\forall 1 \leq i \leq k, 1 \leq j \leq m, U_i, V_j$ are respectively the number of literals in A_i, B_j. In Figure 4(a), there are $m \sum_{i=1}^{k} U_i + k \sum_{j=1}^{m} V_j$ *path-segments*. In Figure 4(b) there are $\sum_{i=1}^{k} U_i + \sum_{j=1}^{m} V_j$ *path-segments*. Since $m, k \geq 2$ therefore the number of *path-segments* in M_a is greater than the number of *path-segments* in M_b. From this we have the following lemma.

Lemma 2.5. For DMC, the number of *modified paths* in C_a is greater than the number of *modified paths* in C_b. Also, the number of *unmodified paths* in C_a is equal to the number of *unmodified paths* in C_b.

In Figure 4 $A_1, \ldots, A_k$ are the **A-cubes** and $B_1, \ldots, B_m$ are the **B-cubes**. If an *m-segment* starts at a literal, say r, which is "part" of a *B-cube* B_i (*A-Cube* A_j), for some $i(j)$, then the *m-segment* is a **B-segment** (**A-segment**). Every *m-segment* in Figure 4 is either an *A-segment* or a *B-segment*. To simplify the notation we assume: an *A-segment* of Figure 4(b) to be of the form $< l, A_1, c, f >$ where $A_1 = la_1 \ldots a_t$; and a *B-segment* of Figure 4(b) to be of the form $< r, B_m, d, f >$ where $B_m = rb_1 \ldots b_s$. Lemma 2.6 (Lemma 2.7) is for *A-Segment* (*B-Segment*).

Lemma 2.6. Let C_b be derived from C_a using DMC. For $t \in \{r, f\}$, if V is a MPP-HF RT for any of the DFs $< p >_t$ of C_a where: $p = p_1 q_2 p_3$ is a *modified path* in C_a; q_2 is an *m-segment*; and $q_2 \in \{< l, c_1, g >, \ldots, < l, c_m, g >\}$ then V is also a MPP-HF RT for the DF $< q >_t$. Here, $q = p_1 p_2 p_3$ is a *modified path* in C_b and $p_2 =< l, A_1, c, f >$ i.e. p_2 is an *A-Segment*.

Lemma 2.7. Let C_b be derived from C_a using DMC. For $t \in \{r, f\}$, if V is a MPP-HF RT for any of the DFs $< p >_t$ of C_a where: $p = p_1 q_2 p_3$ is a *modified path* in C_a; q_2 is an *m-segment*; and $q_2 \in \{< r, d_1, g >, \ldots, < r, d_m, g >\}$ then V is also a MPP-HF RT for the DF $< q >_t$. Here, $q = p_1 p_2 p_3$ is a *modified path* in C_b and $p_2 =< r, B_m, d, f >$ i.e. p_2 is a *B-Segment*.

For a proof of Lemma 2.6 see Appendix III. The proof for Lemma 2.7 is very similar to the proof for Lemma 2.6 and therefore omitted.

Lemma 2.8. Let $p = p_1 p_2 p_3$ where p_2 is the *m-segment* $< B, h, f >$ of Figure 5(b). There exists a MPP-HF RT for $< p >_t, t \in \{r, f\}$ iff $\exists$ a MPP-HF RT for any $< q >_t$ where $q = p_1 q_2 p_3, q_2 \in \{< B, h, g >, \ldots, < B, k, g >\}$ is an *m-segment* in Figure 5(a).

Lemma 2.9. Let $p = p_1 p_{22} p_3$ be a *modified path* of C_a, p_{22} its *m-segment*, and $p_{22} \in S = \{< A, d, f >, < a_1, c, d, f >, \ldots < a_k, c, d, f >, < a_k, c, e, h, f >\}$ (Figure 5(a)). Then, $\exists$ a unique *modified path* $q = p_1 q_{22} p_3$ s.t. V is a MPP-HF RT for $< p >_t$ iff V is a MPP-HF RT for $< q >_t$. For each $p_{22} \in S$, the unique q_{22} is shown in Table 2.

The proof of Lemma 2.8, 2.9 are given in Appendix III. The next theorem follows from Lemmas 2.4 - 2.9.

Theorem 2.10. DMC, DSCC are TETs.

In Lemmas 2.6, 2.7, 2.8 we have shown that: $\forall t \in \{r, f\}$

$< p_1 < l, c_1, g > p_3 >_t + \ldots + < p_1 < l, c_m, g > p_3 >_t \Longrightarrow < p_1 < l, A_1, c, f > p_3 >_t$;

$< p_1 < r, d_1, g > p_3 >_t + \ldots + < p_1 < r, d_m, g > p_3 >_t \Longrightarrow < p_1 < r, B_m, d, f > p_3 >_t$; and

$< p_1 < B, h, g > p_3 >_t + \ldots + < p_1 < B, k, g > p_3 >_t \Longrightarrow < p_1 < B, h, f > p_3 >_t$

In [8] heuristics for carefully selecting the proper SR-1 transformation to improve testability was suggested. Similar heuristics can be derived from the first two (third) implications suggests a heuristic for DMC(DSCC). We leave it to the interested reader to experimentally evaluate the effectiveness of these heuristics.

3 Testability of Unate Circuits

The factored form $F = (a+b)((d+e)+(a+c))$ is SPP-HF DFT. This factored form cannot be derived from any SOP form using "algebraic transformations". Therefore results on delay fault testability of SOP circuits and delay fault testability preserving properties of "algebraic transformations" cannot be used to identify this factored form to be DFT. It can be derived from a product-of-sum form by Boolean factorization. However, Boolean factorization does not, in general are not TPTs. In short, known theoretical results fail to identify F to be a DFT circuit. However, this circuit has a property, which we will state below, makes it a DFT circuit. This property will be shown to be a sufficient condition for a circuit to be a DFT circuit.

NAND, NOR, NOT (OR, AND) are gates of **odd parity** (**even parity**). Path P from X to Y is of **even** (**odd**) parity iff, excluding node X, P traverses an even (odd) number of *odd parity* gates.

An input variable X is a **unate variable** iff every pair of reconverging , starting at X, have the same parity. Else it is a **binate variable**. In Figure 1: e is *unate*; and a, b, c, d, f are *binate*. A circuit C is a **unate circuit** iff every input of C is *unate*.

A circuit is **stuck-at irredundant** iff $\exists$ a test for all single stuck-at faults in it. Else, it is **stuck-at redundant**. A circuit is **all path irredundant** iff $\forall$ input variables X, and $\forall$ single path p starting from X to a primary output $\exists$ an input vector T_0 (T_1) that detects the fault X *s-a-0* (*s-a-1*) and T_0 (T_1) propagates the fault along p, and no other path, to the same primary output. In Figure 8 $< a = 1, b = 0, c = 1, d = 0 >$ propagates fault b *s-a-1* along the single path $< b, e, f, h >$ and vector $< a = 1, b = 0, c = 1, d = 1 >$ propagates fault

b *s-a-1* along multiple paths $< b, e, f, h >$, $< b, e, g, h >$. *All-path-irredundancy* is a stronger condition that *stuck-at irredundancy*.

For input vector T_i and input variable X the input vector $adj(T_i, X)$ differs from T_i only in the value it assigns to X. In Table 1, $T2 = adj(T1, e)$ and $T4 = adj(T3, e)$. For a proof of Lemmas 3.1, 3.2 see Appendix I.

Lemma 3.1. Let X be a *unate variable* of $C, T2$ any input vector and V the input pair $< adj(T2, X), T2 >$. Then, if there are no hazards at the inputs of C on application of V: (i) there are no hazards at any line in C; and (ii) for any gate G in C if there are any transitions at the inputs of G then either all are *rising* or all are *falling* transitions.

Lemma 3.2. Let X be a *unate input* of the *all-path-irredundant* circuit C, $P =< X, a_2, \ldots, a_n = Y >$ a single path starting at X and $T_0^X (T_1^X)$ a test for the fault "line $< X, a_2 >$ s-a-0 (s-a-1)" s.t. $T_0^X (T_1^X)$ propagates the fault along the single path P. Then, $V_r^X =< adj(T_0^X, X), T_0^X >$ ($V_f^X =< adj(T_1^X, X), T_1^X >$) is an *MPP-HF* RT for the DF *input rising* (*input falling*) along P.

In Figure 8, $T1 =< a = 1, b = 1, c = 1, d = 1 >$ and $T2 =< a = 1, b = 1, c = 1, d = 0 >$ are both tests for $< b, f >$ s-a-0. However, $< adj(T1, b), T1 >$ is not a RT for a rising transition along $< b, f, e, h >$ but $< adj(T2, b), T2 >$ is. It does not contradict Lemma 3.2 because $T1$ propagates the fault $< b, e >$ s-a-0 along the multiple path consisting of lines b, f, g, e, h.

RTs constructed using Lemma 3.2 can be MPP-HF or SPP-HF. $T2$ of Table 1 is a test for line $< e, 9 >$ s-a-1 (Figure 1(a)). RT $< adj(T2, e), T2 >$ is an SPP-HF test for the falling transition along $< e, 9, 11, 12 >$(Figure1(a)). Similarly, $< adj(T4, e), T4 >$, constructed from $T4$ of Table 1, a test for line $< e, 9 >$ s-a-1, is an MPP-HF RT for a falling transition along $< e, 9, 11, 12 >$ (Figure 1(b)).

Relationship between tests for stuck-at faults and RTs for DFs were studied in [17]. In [17] RTs for a path in a network β was derived from a test for a stuck-at fault in another network β'. We construct RTs for a path in β from a test for a stuck-at fault in the network β itself. However, unlike the method in [17], our method can be used to construct RTs for paths starting at unate inputs only. From Lemmas 3.1, 3.2 we have Theorem 3.3.

Theorem 3.3. *All-path-irredundancy* is a condition for *unate circuits* to be MPP-HF DFT.

Cor. 3.4. A *unate, multilevel, multi-output tree circuit* is MPP-HF DFT if every component is stuck-at-irredundant.

4 Application

4.1 Carry Lookahead Adders

The *carrychain* of the *CarryLookAhead* Adder (Figure 9) is all-path-irredundant and for each i, cone of c_i is unate. Therefore, Theorem 3.3 implies $\exists$ an MPP-HF RT for all paths starting at the inputs and terminating at c_i. Thus, the *carrychain* is DFT. Each B_i is DFT. Moreover, $(\forall i)c_i$ is a function of $C_0, x_1, y_1, \ldots, x_{i-1}, y_{i-1}$. Therefore, c_i is independent of x_i, y_i. This and Theorem 4.4 from [6] implies that carry lookahead adders are MPP-HF DFT. This also implies that adders using group carrylookahead are also MPP-HF DFT. Prior to this only ripple carry adders were known to be DFT[6].

4.2 Unate Decomposition

Shannon's Expansion of Boolean expression E w.r.t. a **splitting variable** X_j is given by $X_j E_0 + \overline{X_j} E_1$ where: E_1, E_0 are **cofactors** of E w.r.t. X_j.

Theorem 4.1.[10] Let C_0, C_1 be MPP-HF (SPP-HF) DFT circuits implementing *cofactors* E_0, E_1 of X_j. Then, C derived using decomposition rules of Figure 10 are MPP-HF (SPP-HF) DFT circuits. The following Unate Decomposition paradigm to synthesize DFT circuits[10] can be enhanced as discussed below.

Robust-Testable-Design(F)

Step 1. Minimize the function F for a two level design.

Step 2. Check if the DFs in the two-level circuit of Step 1 are detectable. If yes, stop.

Step 3. Apply "a heuristic" to select the splitting variable X_i.

Step 4. Call *Robust-Testable-Design*($F_{\overline{X_i}}$) and *Robust-Testable-Design*(F_{X_i}) to realize the DFT circuit using Theorem 4.1.

MultiLevel Decomposition

Step 1. Given a multilevel circuit G convert it into a *Tree Circuit* C_1.

Step 2. Convert C_1 into a *stuck-at irredundant* circuit C_E by identifying the irredundant s-a faults and eliminating them.

Step 3. If C_E is *MPP-HF* DFT then return. Else, choose a *binate variable* X_j as the *splitting variable* using heuristics from[10].

Step 4. Determine the *cofactors* C_E^0, C_E^1 of C_E w.r.t X_j. Use *MultiLevel Decomposition* to construct *MPP-HF* DFT versions C_0, C_1 respectively of C_E^0, C_E^1.

Step. 5. Use the rules of Figure 10 to construct an *MPP-HF* DFT version C of C_E.

Cor 3.4 guarantees that Step 4 of MultiLevel Decomposition will terminate. Thus, it is a *complete procedure* for synthesizing DFT circuits.

The important difference between the above two procedures is that, unlike *Procedure-Robust-Testable-Design*, *Multilevel Decomposition* is not constrained to start with a two-level design but can start with any multilevel *Tree Circuit* (which includes two-level designs). This distinction is important because for some Boolean functions every two-level expression is very large. As a result, because of resource constraint, it may not be possible to use *Procedure-Robust-Testable-Design.* On the other hand, there often exists a small Tree Circuits for such Boolean functions. In this case *MultiLevel Decomposition* can be used.

Thus our decomposition algorithm is more powerful and flexible than previously known decomposition algorithm for synthesizing DFT circuits. This, in turn allows the exploration of a much larger design space while synthesizing DFT circuits.

4.3 Pre and Post-Decomposition

The above algorithms, although complete, are unsatisfactory because the resulting designs have large area overhead. To alleviate this problem a paradigm that seems to be evolving is a three step process: **Pre-Expansion; Decomposition; and Post-Expansion**. We have seen above how our results improves the Decomposition Phase.

In Pre-Expansion attempts are made to covert the given untestable circuit into a DFT circuit using TETs or extra-inputs. We have shown that in addition to SR-1, TPTs DMC, DSCC, but not DSC, XORC and D(2,2,3)C, can be used in this phase. Thus, using these additional ATs, a larger design space can be explored during Pre-Expansion.

During Post-Expansion TPTs are used to reduce the area of the synthesized DFTs. Our results state that in addition to algebraic factorization based ATs other ATs based on dual extraction can also be used. Thus, even in this phase our results enable the exploration of a much larger design space.

Conclusion

We addressed a number of issues pertaining to the synthesis of DFT circuits. They enhance the evolving paradigm for synthesizing DFT circuits by making it possible to explore a larger design space. Hopefully these results constitutes some useful steps, in the right direction, in our attempt to develop a practical system for synthesizing DFT circuits.

References

[1] P. Ashar, S. Devadas, and K. Keutzer, "Testability Properties of Multilevel logic networks derived from binary decision diagrams," In *Proc. Santa Crutz Conf. Adv. Research in VLSI*, April 1991.

[2] R. K. Brayton, et. al., "MIS: A Multiple-Level Logic Optimization System," *IEEE Transactions on CAD/ISCAS*, November 1987, pp. 1062 - 1081.

[3] B. Becker, R. Dreschsler, H. Hengster, "Local Circuit Transformations Preserving Robust Path-Delay-Fault Testability," Tech. Report, Department of Informatiks, Univ. of Frankfurt, FRG, January 1993.

[4] M. Bryan, S. Devadas and K. Keutzer, "Testability Preserving Circuit Transformations," *1990 IEEE Int'l Conference on Computer-Aided Design*, pp. 456 - 459.

[5] S. Devadas and K. Keutzer, "Synthesis of Robust Delay-Fault Testable Circuits: Practice", *IEEE Transactions on CAD/ISCAS*, March 1992, pp. 277-300.

[6] S. Devadas and K. Keutzer, "Synthesis of Robust Delay-Fault Testable Circuits: Theory," *IEEE Transactions on CAD/ISCAS*, January 1992, pp. 87-101.

[7] G. Hactel et. al., "On the Properties of Algebraic Transformations and the Multifault Testability of Multilevel Logic," In *Proc. 1989 IEEE International Conference on Computer-Aided Design*, November 1989, pp. 422 - 425.

[8] N. K. Jha, I. Pomerantz, S. M. Reddy, R. J. Miller, "Synthesis of Multi-Level Combinational Circuits for Complete Robust Path Delay Fault Testability," In *Proc. IEEE/ACM International Fault-Tolerant Computing Symposium*, pp. 280-287, 1992.

[9] S. Kundu and A. K. Parmanick, "Testability Preserving Boolean Transformation for Logic Synthesis," *IEEE VLSI Test Symposium*, pp. 331 - 338, 1993.

[10] S. Kundu, S. M. Reddy and N. K. Jha, "On the Design of Robustly Testable CMOS Combinational Logic Circuits," *IEEE Transactions on CAD/ISCAS*, August 1991, pp. 1036 - 1048.

[11] C. J. Lin and S. M. Reddy, "On Delay Fault Testing in Logic Circuits," *IEEE Transactions on CAD/ISCAS*, Vol. 6, No. 5, Sept. 1987, pp. 694-703.

[12] E. Mendelson, *Introduction to Mathematical Logic.* Second Edition. D. Van Nostrand.

[13] I. Pomerantz and S. M. Reddy, "Achieving complete delay fault testability by extra inputs," In *Proc. Intl. Test Conference*, Oct., 1991.

[14] A. K. Pramanick and S. M. Reddy, "On the Design of Path Delay Fault Testable Combinational Circuits," In *Proc. 20th IEEE International Fault- Tolerant Computing Symposium*, June 1990, pp. 374 - 381.

[15] J. Rajski and J. Vasudevamurthy, "The Testability Preserving Concurrent Decomposition and Factorization of Boolean Expressions" *IEEE Trans. on CAD*, Vol. 11, No. 6, pp. 778 - 793, June 1992.

[16] J. Rajski, J. Vasudevamurthy and A. El-Maleh, "Recent Advances in Synthesis with Testability", *IEEE VLSI Test Symposium*, pp. 254 - 256, 1992.

[17] A. Saldhana, R. K. Brayton, A. L. Sangiovanni-Vincentelli, "Equivalence of Robust Delay-Fault and Single Stuck-Fault Test Generation," *IEEE/ACM Design Automation Conference*, pp. 173 - 176, 1992.

[18] G. L. Smith, "Model for Delay Faults Based Upon Paths," In *Proc. 1985 IEEE International Test Conference*, Nov. 1985, pp. 342 - 349.

Appendix I

(Proof of Lemmas 3.1, 3.2)

The following definitions are illustrated using Figure 1(a). Let $P =< a_1, \ldots, a_n >$ be a path that passes through gates $a_1, \ldots, a_n$. Then, inputs of gates $a_2, \ldots, a_n$ that do not lie along P are **off-path inputs** of P. If $P =< a, 6, 10, 12 >$ then $< c, 6 >, < 7, 10 >$ are two of its *off-path inputs*. Inputs of gates that lie along P constitute the **on-path inputs** of path P.

If k is an *off-path input* of P and an input of gate a_j then k is an **off-path input of** a_j w.r.t P. Line $< 7, 10 >$ is an *off-path input* of gate 10 w.r.t. path $< a, 6, 10, 12 >$. For all a_j, if an input k of a_j is an *on-path input* of path P then k is an **on-path input** of a_j w.r.t. P.

Signal states of a line w, on application of a pair of input vectors $< V1, V2 >$, are represented using multivalued logic[11] as follows: **S0(S1)** iff the steady state value at w, on application of both $V1, V2$, is $0(1)$ and there are no hazards during the transition interval; and **U0(U1)** iff the steady state value at w, on application of $V2$, is $0(1)$ independent of the steady state value, on application of $V1$, and the state of the signal during the transition interval.

Lemma AI.1. [11] $V =< V1, V2 >$ is a *robust test* for a $1 \rightarrow 0$ ($0 \rightarrow 1$) along P iff $V1(a_1) = 1(0)$, $V2(a_1) = 0(1)$ and each gate a_j along P satisfies the following condition.

(i) There is a $0 \rightarrow 1$ transition at the *on-path input* $< a_{j-1}, a_j >$ of a_j w.r.t. P. If(a_j is a NAND or AND gate) then for each off-path input I of a_j w. r. t. P $V(I) = S1$ or $U1$. If(a_j is a NOR, OR gate) then for each off-path input I of a_j w. r. t. P $V(I) = S0$.

(ii) There is a $1 \rightarrow 0$ transition at the *on-path input* $< a_{j-1}, a_j >$ of a_j w.r.t. P. If(a_j is a NAND or AND gate) then for each off-path input I of a_j w. r. t. P $V(I) = S1$. If(a_j is a NOR, OR gate) then for each off-path input I of a_j w. r. t. P $V(I) = S0$ or $U0$.

Cor AI.2. If no *off-path input* of P is assigned either $U0, U1$; there are no hazards at any of these inputs; and V satisfies the conditions of Lemma AI.1 then, V is an SPP-HF Robust test.

Lemma 3.1. Let X be a *unate variable* of $C, T2$ any input vector and V the input pair $< adj(T2, X), T2 >$. Then, if there are no hazards at the inputs of C on application of V: (i) there are no hazards at any line in C; and (ii) for any gate G in C if there are any transitions at the inputs of G then either all are *rising* or all are *falling* transitions.

Proof. (i) Among all the primary inputs of C, on application of V, there is a transition only at the unate input X. Let, if possible, $\exists$ G s.t. on application of V, there is a *rising transition* at input g_1 of G and a *falling transition* at another input g_2 of G. Clearly,

G must be a *reconverging gate* of X s.t. $\exists$ two *disjoint paths* of different parity from X to G. This contradicts the fact that X is a *unate variable*. Hence, the proof.

(ii) Let, if possible, $\exists$ a line g, which is the output of a gate G s.t. $\exists$ a hazard at g on application of V. Therefore, $\exists$ a line $h \in Cone(g)$, which is the output of gate H, s.t. $\exists$ a hazard at h on application of V. (H could be G itself.) Moreover, $\nexists$ any line $w \in Cone(h)$ s.t. $\exists$ a hazard at w on application of V. Such an h exists because all the inputs are hazard free on application of V.

Thus, $\exists$ inputs h_1, h_2 of H s.t. $\exists$ a *rising transition* (*falling transition*) at $h_1(h_2)$. This contradicts (i). Hence the proof. $\Box$

Lemma 3.2. Let X be a *unate input* of the *all-path-irredundant* circuit C, $P =< X, a_2, \ldots, a_n = Y >$ a path starting at X and $T_0^X(T_1^X)$ a test for the fault "line $< X, a_2 >$ s-a-0 (s-a-1)" s.t. $T_0^X(T_1^X)$ propagates the fault along P. Then, $V_r^X =< adj(T_0^X, X), T_0^X >$ ($V_f^X =< adj(T_1^X, X), T_1^X >$) is an *MPP-HF* RT for the DF *input rising* (*input falling*) along P.

Proof. The proof for V_r^X is presented. The proof for V_f^X is very similar. For notational simplicity we assume that every gate has a fanin of at most two. The proof of the lemma when gates have larger fanin is very similar.

To show that V_r^X is an MPP-HF robust test for the *input rising* along P we show that V_r^X satisfies the conditions of Lemma AI.1. Since, T_0^X is a test for line $< X, a_2 >$ s-a-0 therefore $T_0^X(X) = 1$. From the definition of $adj(T_0^X, X)$ it follows that $adj(T_0^X, X)(X) = 0$. Therefore, V_r^X result in a *rising transition* at X. It is HF because of Lemma 3.1. We next show that V_r^X "robustly propagates" this transition through gates $a_2, \ldots, a_n$ that is traversed by P, as required by Lemma AI.1.

Consider any gate a_j in P and refer to Figure 11. In Figure 11, for each gate shown, input A is the *on-path input* of a_j w.r.t. P and B is the *off-path input* of a_j w.r.t. P. The transition shown at A is the transition that propagates from X to a_j.

First consider a $0 \rightarrow 1$ transition at A. On application of V_r^X, there cannot be any hazard at the lines in the given circuit C and there can only be unidirectional transitions at the inputs of any gate (in particular a_j) (Lemma 3.1). Thus, on application of V_r^X, B can have the transition $S0, S1$, or $0 \rightarrow 1$. However, not all these values are possible. We have four cases.

Case 1. a_j is a NOR gate. T_0^X is a test for line $< X, a_2 >$ s-a-0 that propagates the fault along P. Therefore, $T_0^X(B) = 0$. Thus, on application of V_r^X, line B cannot have the value $S1$ nor can there be a $0 \rightarrow 1$ transition at B. Therefore, the only possible value at B, on application of V_r^X, is $S0$. This is shown in Figure 11 (top, left hand corner). This satisfies the conditions of Lemma AI.1.

Case 2. a_j is an AND gate. Using an argument similar to Case 1 we have $T_0^X(B) = 1$. Therefore, $V_r^X(B) \neq S0$. Thus, as shown in Figure 11, $V_r^X(B) \in \{0 \rightarrow 1, S1\}$ both of which satisfies the conditions of Lemma AI.1. Hence, even in this case, V_r^X robustly propagates the fault to the output of a_j.

Case 3. a_j is a NAND gate. Similar to the AND case.

Case 4. a_j is an OR gate. Similar to the NOR case.

Thus, if a $0 \rightarrow 1$ transition arrives at the input of any gate a_j that is traversed by P then V_r^X will robustly propagate that transition to the output of a_j. Similarly if a $1 \rightarrow 0$ transition arrives at A then V_r^X robustly propagates it to the output of a_j. To help the reader go through the four possible cases in Figure 11 we have presented the possible signal values at B for these four cases. $\Box$

Appendix II

(Proof of Lemmas 2.1, 2.2)

In subsequent proofs, to denote the state of the signal at line w on application of the pair $V =< V1, V2 >$, we use the set $L = \{S0, S1, r, f, hr, hf, h0, h1\}$. Let $V1(w), V2(w)$ be the steady state values at w on application of $V1, V2$ respectively. $V(w)$ has the value: $S0(S1)$ iff $V1(w) = V2(w) = 0(1)$ and w is hazard free; $h0(h1)$ iff $V1(w) = V2(w) = 0(1)$ and there exists a hazard at w; $r(f)$ iff $V1(w) = 0(1), V2 = 1(0)$ and there are no hazards at w; and $hr(hf)$ iff $V1(w) = 0(1), V2 = 1(0)$ and there exists a hazard at w.

We use some ideas from propositional calculus[12]. $V(w) = a$ denotes the proposition that line w has the value $a \in L$. If X, Y are two propositions than XY denotes the "logical and" of these two propositions and $X + Y$ denotes the "logical or" of these two propositions.

Note that, in Figure 2, if M_a, M_b implement the same Boolean function then $V1(g) = V1(f)$ and $V2(g) = V2(f)$. Thus: $V(g) = S1$ can imply either $V(f) = S1$ or $V(f) = h1$; $V(g) = S0$ can imply either $V(f) = S0$ or $V(f) = h0$; $V(g) = r$ can imply either $V(f) = r$ or $V(f) = hr$; and $V(g) = f$ can imply $V(f) = f$ or $V(f) = hf$. This leads to the following observation.

Observation AII.1. (i) To show that *V(g) is hazard free implies V(f) is hazard free* we only need to show that: $V(g) = S1 \Longrightarrow V(f) = S1$; $V(g) = S0 \Longrightarrow V(f) = S0$; $V(g) = r \Longrightarrow V(f) = r$; and $V(g) = f \Longrightarrow V(f) = f$.

(ii) To show that $V(g) = a \iff V(f) = a, \forall a \in L$ we only need to show that: $V(g) = S1 \iff V(f) = S1$; $V(g) = S0 \iff V(f) = S0$; $V(g) = r \iff V(f) = r$; and $V(g) = f \iff V(f) = f$.

Lemma 2.1. If C_b of Figure 2 is derived from C_a using either DSCC, XORC or D(2,2,3)C then for every *unmodified path* q_1 of C_b an input-pair $V =< V_1, V_2 >$ is an MPP-HF-RT (SPP-HF-RT) for a transition along q_1 iff V is an MPP-HF-RT (SPP-HF-RT) for the same transition along $p_1 = I(q_1)$.

Proof. To prove the lemma it is enough to show that for any input-pair V, $V(g)$ is "hazard-fee" implies that $V(f)$ is "hazard-free". We use Observation AII.1. For each of DSCC

(Figure 5), XORC(Figure 6) and D(2,2,3)C (Figure 7) we note that M_a, M_b are functionally equivalent.

For DSCC refer to Table 3. The combination of values, over L, at the inputs $a_1, \ldots, a_k$, A, B of Figure 5(a) that set g to $S0, S1, r, f$ are shown in Table 3. Some entries are ordered pairs of the form p,q. This implies that the corresponding variable can have either the value p or q. The entry '-' represents the fact that the corresponding variable can be assigned any value from the set L. The reader can verify that these are exactly the combination of input values that will also set f of Figure 5(b) respectively to $S0, S1, r, f$.

Tables 4, 5 are respectively for XORC, D(2,2,3)C. The claim for these two cases can be verified from these tables. □

Lemma 2.2. If C_b of Figure 2 is derived from C_a using either DSCC, XORC or D(2,2,3)C then for every *modified path* q_2 in C_b $\exists$ a *modified path* p_2 in C_a s.t. $V = < V1, V2 >$ is an MPP-HF-RT (SPP-HF-RT) for a transition along p_2 iff V is an MPP-HF-RT (SPP-HF-RT) for the same transition along q_2.

Proof. It is enough to show that given any q_2 we can always define p_2 satisfying the conditions of the lemma. Given, $q_2 = q_{21}q_{22}q_{23}$ define $p_2 = q_{21}p_{22}q_{23}$ where the two paths differ only in the *m-segment*.

First, consider DSCC (Figure 5). Using Observation AII.1(ii) and Table 3 it can be verified that $(V(g) = a) \iff (V(f) = a), \forall a \in L$. Thus, it is enough to show that $\forall q_{22}$ of Figure 5(b) $\exists$ p_{22} of Figure 5(a) s.t. if V "robustly propagates" a "hazard-free transition" along p_{22} then V also "robustly propagates" a "hazard-free transition" along q_{22}. Let $< p >_t$ denote the proposition that, on application of an input-pair V, an input transition "t" "robustly propagates" along the path p and there are no hazards along p, $t \in \{r, f\}$. Whether we are considering multiple paths or single paths will become clear from the context.

In Table 2 for each q_{22} of Figure 5(b) $\exists$ a p_{22} of Figure 5(a). We next show that for each pair p_{22}, q_{22} of Table 2 $< p_{22} >_t \Longrightarrow < q_{22} >_t, t \in \{r, f\}$. We show this by deriving a propositional expression for $< p_{22} >_t$ using Figure 5(a) and verifying that this expression implies $< q_{22} >_t$. There are two cases to be considered. One for MPP-HF-RTs and the other for SPP-HF-RTs.

We first consider the case for MPP-HF-RT for the <u>first pair of Table 2</u>. From Figure 5(a) $< p_{22} >_r = < A, e, g >_r$
$\iff (A = r)((a_1 = S1 + a_1 = r) \ldots (a_k = S1 + a_k = r))((h = S0) \ldots (k = S0))$
$\iff (A = r)((a_1 = S1 + a_1 = r) \ldots (a_k = S1 + a_k = r))((a_1 = S1 + B = S0) \ldots (a_k = S1 + B = S0))$
$\iff (A = r)((a_1 = S1 + a_1 = r) \ldots (a_k = S1 + a_k = r))(B = S0 + (a_1 = S1) \ldots (a_k = S1))$
$\iff (A = r)((a_1 = S1 + a_1 = r) \ldots (a_k = S1 + a_k = r)(B = S0) + (a_1 = S1) \ldots (a_k =$

$S1))$
$\Longleftrightarrow (A=r)((c=S1+c=r)(B=S0)+(c=S1))$ /* From Figure 5(b) */
$\Longleftrightarrow (A=r)((c=S1+c=r)(h=S0)+(c=S1)(h=S0)) \Longleftrightarrow < A,d,f >_r = < q_{22} >_r$. **(AII.1)**

We next consider the case for SPP-HF-RT for the first pair. From Figure 5(a) $< p_{22} >_r = < A,e,g >_r$
$\Longleftrightarrow (A=r)((a_1=S1)\ldots(a_k=S1))((h=S0)\ldots(k=S0))$
$\Longleftrightarrow (A=r)((a_1=S1)\ldots(a_k=S1))((a_1=S1+B=S0)\ldots(a_k=S1+B=S0))$
$\Longleftrightarrow (A=r)((a_1=S1)\ldots(a_k=S1))(B=S0+(a_1=S1)\ldots(a_k=S1))$
$\Longleftrightarrow (A=r)((a_1=S1)\ldots(a_k=S1)(B=S0)+(a_1=S1)\ldots(a_k=S1))$
$\Longleftrightarrow (A=r)((c=S1)(B=S0)+(c=S1))$ /* From Figure 5(b) */
$\Longleftrightarrow (A=r)((c=S1)(h=S0)+(c=S1)(h=S0)) \Longleftrightarrow < A,d,f >_r = < q_{22} >_r$.

Note that the above derivation for SPP-HF-RT is a simplification of the derivation for the MPP-HF-RT case. This will be true in general. Henceforth, we will only present the derivation for the MPP-HF-RT case and leave the derivation for the SPP-HF-RT case to the interested reader.

$< p_{22} >_f = < A,e,g >_f \Longleftrightarrow (A=f)((a_1=S1)\ldots(a_k=S1))((h=S0+h=f)\ldots(k=S0+h=f))$
$\Longleftrightarrow (A=f)((a_1=S1)\ldots(a_k=S1))((a_1=S1+B=S0+(a_1=r)(B=S1)+(a_1=S0)(B=f)+(a_1=r)(B=f))\ldots(a_k=S1+B=S0+(a_k=r)(B=S1)+(a_k=S0)(B=f))$
$\Longleftrightarrow (A=f)((a_1=S1)\ldots(a_k=S1))((a_1=S1+B=S0)\ldots(a_k=S1+B=S0))$
$\Longleftrightarrow (A=f)((a_1=S1)\ldots(a_k=S1))(B=S0+(a_1=S1)\ldots(a_k=S1)) \Longleftrightarrow (A=f)(c=S1)(B=S0+c=S1)$ /* From Figure 5(b) */
$\Longleftrightarrow (A=f)(c=S1) \Longleftrightarrow (A=f)(c=S1)(h=S0) \Longleftrightarrow < A,d,f >_f = < q_{22} >_f$. **(AII.2)**

For the second pair of Table 2, from Figure 5(a) $< p_{22} >_r = < a_1,e,g >_r$
$\Longleftrightarrow (a_1=r)((A=S1+A=r)(a_2=S1+a_2=r)\ldots(a_k=S1+a_k=r))((h=S0)\ldots(k=S0))$
$\Longleftrightarrow (a_1=r)((A=S1+A=r)(a_2=S1+a_2=r)\ldots(a_k=S1+a_k=r))((a_1=S1+B=S0)\ldots(a_k=S1+B=S0))$
$\Longleftrightarrow (a_1=r)((A=S1+A=r)(a_2=S1+a_2=r)\ldots(a_k=S1+a_k=r))(B=S0+(a_1=S1)\ldots(a_k=S1))$
$\Longleftrightarrow (a_1=r)(A=S1+A=r)(a_2=S1+a_2=r)\ldots(a_k=S1+a_k=r)(B=S0)$
$\Longrightarrow (a_1=r)(A=S1+A=r)(a_2=S1+a_2=r)\ldots(a_k=S1+a_k=r)(h=S0)$ /* From Figure 5(b) */
$\Longleftrightarrow < a_1,c,d,f >_r = < q_{22} >_r$. **(AII.3)**

$< p_{22} >_f = < a_1,e,g >_f \Longleftrightarrow (a_1=f)((a_2=S1)\ldots(a_k=S1)(A=S1)(h=f+h=S0)\ldots(k=f+k=S0))$

$\Longleftrightarrow (a_1 = f)((a_2 = S1) \ldots (a_k = S1)(A = S1)$
$(a_1 = S1 + B = S0 + (B = S1)(a_1 = r) + (a_1 = S0)(B = f) + (a_1 = r)(B = f)) \ldots$
$(a_k = S1 + B = S0 + (B = S1)(a_k = r) + (a_k = S0)(B = f) + (a_k = r)(B = f))$
$\Longleftrightarrow (a_1 = f)(a_2 = S1) \ldots (a_k = S1)(A = S1)(B = S0) \ldots (a_k = S1 + B = S0)$
$\Longleftrightarrow (a_1 = f)(a_2 = S1) \ldots (a_k = S1)(A = S1)(B = S0)$
$\Longleftrightarrow (a_1 = f)(a_2 = S1) \ldots (a_k = S1)(A = S1)(h = S0)$ /* From Figure 5(b) */
$\Longleftrightarrow < a_1, c, d, f >_f = < q_{22} >_f$. **(AII.4)**

The two cases for the next three pairs of Table 2 are very similar to the second pair and therefore omitted. For the last pair of Table 2 we have the following.

$< p_{22} >_f = < B, h, g >_f \Longleftrightarrow (B = f)(a_1 = S0)(e = S0 + e = f) \ldots (k = S0 + k = f)$
$\Longleftrightarrow (B = f)(a_1 = S0)(a_1 = S0 + \ldots + a_k = S0 + A = S0 + (a_1 = f + a_1 = S1) \ldots (a_k = f + a_k = S1)(A = f) + (a_1 = f) \ldots (a_k = f + a_k = S1)(A = f + A = S1) + \ldots + (a_1 = f + a_1 = S1) \ldots (a_k = f)(A = f + A = S1)) \ldots (a_k = S1 + B = S0 + (B = S1)(a_k = r) + (B = f)(a_k = r) + (B = f)(a_k = S0))$
$\Longleftrightarrow (B = f)(a_1 = S0)(a_1 = S0 + \ldots + a_k = S0 + A = S0) \ldots (a_k = S1 + (B = f)(a_k = r) + (B = f)(a_k = S0))$
$\Longleftrightarrow (B = f)(a_1 = S0) \ldots (a_k = S1 + a_k = r + a_k = S0)$
$\Longrightarrow (B = f)(c = S0)$ /*From Figure 5(b) */
$\Longrightarrow (B = f)(e = S1)(d = S0) \Longleftrightarrow < B, h, f >_f \Longleftrightarrow < q_{22} >_f$ **(AII.5)**

$< p_{22} >_r = < B, h, g >_r \Longleftrightarrow (B = r)(a_1 = S0 + a1 = f)(e = S0) \ldots (k = S0)$
$\Longleftrightarrow (B = r)(a_1 = f + a_1 = S0)(a_1 = S0 + \ldots a_k = S0 + A = S0)(a_2 = S0) \ldots (a_k = S0)$
$\Longleftrightarrow (B = r)(a_1 = f + a_1 = S0)(a_2 = S0) \ldots (a_k = S0)$
$\Longrightarrow (B = r)(c = S0)$ /* From Figure 5(b) */
$\Longleftrightarrow (B = r)(e = S1)(d = S0) \Longrightarrow < B, h, f >_r = < q_{22} >_r$. **(AII.6)**

The reader can verify that Tables 6, 7 can be used respectively for XORC and D(2,2,3)C in a manner analogous to how Table 2 was used for DSCC.□

Appendix III

(Proof of Theorem 2.4 and Lemmas 2.6, 2.9)

Theorem 2.4. DSC, XORC, D(2,2,3)C are not TETs.

Proof. We first prove the case for DSC (Figure 3). It is sufficient to show that $\forall$ *modified-path* p_i in C_a $\exists$ a *modified-path* q_i in C_b s.t. $< p_i >_t, t \in \{r, f\}$ has a "hazard free" RT iff $< q_i >_t$ has a "hazard free" RT. Furthermore, for any two paths p_i, p_j, if $p_i \neq p_j$ then $q_i \neq q_j$. Since the transformation does not affect the *unmodified-paths* in C_a, C_b we need not consider them.

Let $p_i = p_{i1}p_{i2}p_{i3}$ be any *modified-path* where p_{i2} is the *m-segment* in M_a. Define path q_i to be $p_{i1}q_{i2}p_{i3}$ where q_{i2} is an *m-segment* in M_b. It is enough to show that $\forall$ *m-segment* p_{i2} in M_a $\exists$ an *m-segment* q_{i2} in M_b s.t. an input-pair V "robustly propagates", without any hazard, the transition t along p_{i2} iff V "robustly propagates", without any hazard, the same transition t along q_{i2}. Furthermore, if $p_{i2} \neq p_{j2}$ then $q_{i2} \neq q_{j2}$. For DSC this one-one correspondence is shown in Table 8. To complete the proof we show that, for each row of Table 8, $< p_{i2} >_t \Longleftrightarrow < q_{i2} >_t, t \in \{r, f\}$.

Detailed proof for only row 1 is shown. Proofs for the other cases are similar and left to the interested reader. We show that $< a_1, c_1, g >_t \Longleftrightarrow < a_1, c, d_1, f >_t$.

$< a_1, c_1, g >_f \Longleftrightarrow (a_1 = f)(a_2 = S1)\ldots(a_t = S1)(B_1 = S1)(c_2 = S0 + c_2 = f)\ldots(c_m = f + c_m = S0)$
$\Longleftrightarrow (a_1 = f)(a_2 = S1)\ldots(a_t = S1)(B_1 = S1)$
$(a_1 = S0 + \ldots a_t = S0 + B_2 = S0+$
$(a_1 = f)(a_2 = f + a_2 = S1)\ldots(a_t = S1 + a_t = f)(B_2 = f + B_2 = S1) + \ldots +$
$(a_t = f)(a_1 = f + a_1 = S1)\ldots(a_{t-1} = S1 + a_{t-1} = f)(B_2 = f + B_2 = S1)+$
$(B_2 = f)(a_1 = f + a_1 = S1)\ldots(a_t = S1 + a_t = f))$
$\ldots$
$(a_1 = S0 + \ldots a_t = S0 + B_m = S0+$
$(a_1 = f)(a_2 = f + a_2 = S1)\ldots(a_t = S1 + a_t = f)(B_m = f + B_m = S1) + \ldots +$
$(a_t = f)(a_1 = f + a_1 = S1)\ldots(a_{t-1} = S1 + a_{t-1} = f)(B_m = f + B_m = S1)+$
$(B_m = f)(a_1 = f + a_1 = S1)\ldots(a_t = S1 + a_t = f))$
$\Longleftrightarrow (a_1 = f)(a_2 = S1)\ldots(a_t = S1)(B_1 = S1)$
$(B_2 = S0 + (a_1 = f)(a_2 = S1)\ldots(a_t = S1)(B_2 = f + B_2 = S1))$
$\ldots$
$(B_m = S0 + (a_1 = f)(a_2 = S1)\ldots(a_t = S1)(B_m = f + B_m = S1))$
$\Longleftrightarrow (a_1 = f)(a_2 = S1)\ldots(a_t = S1)(B_1 = S1)$
$(B_2 = S0 + B_2 = f + B_2 = S1)\ldots(B_m = S0 + B_m = S1 + B_m = f)$
$\Longleftrightarrow (a_1 = f)(c = f)(B_1 = S1)(B_2 = S0 + B_2 = f + B_2 = S1)\ldots(B_m = S0 + B_m = S1 + B_m = f)$ /*From Figure 3(b) */
$\Longleftrightarrow (a_1 = f)(B_1 = S1)(d_2 = f + d_2 = S0)\ldots(d_m = f + d_m = S0) \Longleftrightarrow < a_1, c, d, f >_f$

$< a_1, c_1, g >_r \Longleftrightarrow (a_1 = r)(a_2 = S1 + a_2 = r)\ldots(a_t = S1 + a_t = r)(B_1 = S1 + B_1 = r)(c_2 = S0)\ldots(c_m = S0)$
$\Longleftrightarrow (a_1 = r)(a_2 = S1 + a_2 = r)\ldots(a_t = S1 + a_t = r)(B_1 = S1 + B_1 = r)(B_2 = S0)\ldots(B_m = S0)$. **(AIII.1)**
$< a_1, c, d_1, f >_r \Longleftrightarrow (a_1 = r)(a_2 = S1 + a_2 = r)\ldots(a_t = S1 + a_t = r)(B_1 = S1 + B_1 = r)(B_2 = S0)\ldots(B_m = S0)$
$< a_1, c, d_1, f >_r \Longleftrightarrow (a_1 = r)(a_2 = S1 + a_2 = r)\ldots(a_t = S1 + a_t = r)(B_1 = S1 + B_1 = r)(d_2 = S0)\ldots(d_m = S0)$
$\Longleftrightarrow (a_1 = r)(a_2 = S1 + a_2 = r)\ldots(a_t = S1 + a_t = r)(B_1 = S1 + B_1 = r)(B_2 = S0)\ldots(B_m = S0)$ **(AIII.2)**
From AIII.1, AIII.2 we have $< a_1, c_1, f >_r \Longleftrightarrow < a_1, c, d_1, f >_r$.

For XORC, the one-one correspondence is shown in Table 6. For D(2,2,3)C the one-one correspondence is shown in Table 7. The proof for these two cases is very similar to that for DSC and left to the interested reader. □

Lemma 2.6. Let C_b be derived from C_a using DMC. For $t \in \{r, f\}$, if V is a "hazard free" RT for any of the DFs $< p >_t$ of C_a where: $p = p_1 q_2 p_3$ is a *modified path* in C_a; q_2 is an *m-segment*; and $q_2 \in \{< l, c_1, g >, \ldots, < l, c_m, g >\}$ then V is also a "hazard free" RT for the DF $< q >_t$. Here, $q = p_1 p_2 p_3$ is a *modified path* in C_b and $p_2 =< l, A_1, c, f >$ i.e. p_2 is an *A-Segment*.

Proof. To prove the lemma it is sufficient to show that $< l, c_1, g >_t + \ldots + < l, c_m, g >_t \Longrightarrow < l, A_1, c, f >_t, t \in \{r, f\}$. We use the conditions for MPP-HF-RT. From Figure 4(b), $< l, A_1, c, f >_f$
$\Longleftrightarrow (l = f)(a_1 = S1) \ldots (a_t = S1)(A_2 = S0 + A_2 = f) \ldots (A_k = S0 + A_k = f)(d = S1)$
$\Longleftrightarrow (l = f)(a_1 = S1) \ldots (a_t = S1)(A_2 = S0 + A_2 = f) \ldots (A_k = S0 + A_k = f)(B_1 = S1 + \ldots + B_m = S1)$ **(AIII.3)**

For Figure 4(a), $A_1 = la_1 \ldots a_t$, $c_i = A_1 B_i, d_j = A_k B_j, 1 \leq i, j \leq m$. In addition, define $q_j = A_2 B_j, 1 \leq j \leq m$
$< l, c_i, g >_f \Longleftrightarrow (l = f)(a_1 = S1) \ldots (a_t = S1)(B_i = S1)(c_1 = S0 + c_1 = f) \ldots (c_{i-1} = S0 + c_{i-1} = f)(c_{i+1} = S0 + c_{i+1} = f) \ldots (c_m = S0 + c_m = f)(q_1 = f + q_1 = S0) \ldots (q_m = f + q_m = S0) \ldots (d_1 = S0 + d_1 = f) \ldots (d_m = S0 + d_m = f)$ **(AIII.4)**

Let $X = (l = f)(a_1 = S1) \ldots (a_t = S1)$. Therefore, $X \Longrightarrow (A_1 = f)$. Also, $(c_j = f + c_j = S0) \Longleftrightarrow (A_1 = S0 + B_j = S0 + (A_1 = S1)(B_j = f) + (A_1 = f)(B_j = f) + (A_1 = f)(B_j = S1))$. Therefore, $X(c_j = f + c_j = S0) \Longrightarrow X(B_j = S0 + B_j = f + B_j = S1)$. From this we get,

$X(c_1 = f + c_1 = S0) \ldots (c_{j-1} = f + c_{j-1} = S0)(c_{j+1} = f + c_{j+1} = S0) \ldots (c_m = f + c_m = S0) \Longrightarrow X(B_1 = S0 + B_1 = f + B_1 = S1) \ldots (B_{i-1} = S0 + B_{i-1} = f + B_{i-1} = S1)(B_{i+1} = S0 + B_{i+1} = f + B_{i+1} = S1) \ldots (B_m = S0 + B_m = f + B_m = S1)$. From this and AIII.4 we get

$< l, c_i, g >_f \Longrightarrow X(B_i = S1)(B_1 = S0 + B_1 = f + B_1 = S1) \ldots (B_{i-1} = S0 + B_{i-1} = f + B_{i-1} = S1)(B_{i+1} = S0 + B_{i+1} = f + B_{i+1} = S1) \ldots (B_m = S0 + B_m = f + B_m = S1)(q_1 = f + q_1 = S0) \ldots (q_m = f + q_m = S0) \ldots (d_1 = f + d_1 = S0) \ldots (d_m = f + d_m = S0)$. **(AIII.5)**

Now, $(d_j = f + d_j = S0) \Longleftrightarrow (A_k = S0 + B_j = S0 + (B_j = S1)(A_k = f) + (B_j = f)(A_k = S1) + (B_j = f)(A_k = f))$. **(AIII.6)**

From (AIII.6), $(B_i = S1)(d_i = f + d_i = S0) \Longleftrightarrow (B_i = S1)(A_k = S0 + A_k = f)$. Similarly, $(B_i = S1)(q_i = f + q_i = S0) \Longleftrightarrow (B_i = S1)(A_2 = S0 + A_2 = f)$. Using this in (AIII.5) we get (AIII.7).

$< l, c_i, g >_f \Longrightarrow X(B_i = S1)(B_1 = S0 + B_1 = f + B_1 = S1) \ldots (B_{i-1} = S0 + B_{i-1} = f + B_{i-1} = S1)(B_{i+1} = S0 + B_{i+1} = f + B_{i+1} = S1) \ldots (B_m = S0 + B_m = f + B_m = S1)(q_1 = f + q_1 = S0) \ldots (q_{i-1} = f + q_{i-1} = S0)(A_2 = f + A_2 = S0)(q_{i+1} = f + q_{i+1} =$

$S0)\dots(q_m = f + q_m = S0)\dots(d_1 = f + d_1 = S0)\dots(d_{i-1} = f + d_{i-1} = S0)(A_k = f + A_k = S0)(d_{i+1} = f + d_{i+1} = S0)\dots(d_m = f + d_m = S0)$. **(AIII.7)**

From (AIII.6), for all $i \neq j$, $(d_j = f + d_j = S0)(A_k = S0 + A_k = f) \Longleftrightarrow (A_k = S0 + A_k = f)(A_k = S0 + B_j = S0 + (B_j = S1)(A_k = f) + (B_j = f)(A_k = S1) + (B_j = f)(A_k = f)) \Longleftrightarrow (A_k = S0 + (A_k = f)(B_j = S0 + B_j = S1 + B_j = f)) \Longrightarrow (A_k = S0 + A_k = f)(B_j = S0 + B_j = S1 + B_j = f)$. Similarly, for all $i \neq j$, $(q_j = f + q_j = S0)(A_2 = S0 + A_2 = f) \Longrightarrow (A_2 = S0 + A_2 = f)(B_j = S0 + B_j = S1 + B_j = f)$. From (AIII.7) and these two implications we get (AIII.8)

$< l, c_i, g >_f \Longrightarrow X(B_i = S1)(B_1 = S0 + B_1 = f + B_1 = S1)\dots(B_{i-1} = S0 + B_{i-1} = f + B_{i-1} = S1)(B_{i+1} = S0 + B_{i+1} = f + B_{i+1} = S1)\dots(B_m = S0 + B_m = f + B_m = S1)(A_2 = f + A_2 = S0)\dots(A_k = f + A_k = S0)$ **(AIII.8)**

$\Longrightarrow < l, c_i, g >_f \Longrightarrow X(B_i = S1)(A_2 = f + A_2 = S0)\dots(A_k = f + A_k = S0)$ **(AIII.9)**

From (AIII.3) and (AIII.9) we have $< l, c_1, g > + \dots + < l, c_m, g > \Longrightarrow < l, A_1, c, f >_f$.

From Figure 4(b) we have $< l, A_1, c, f >_r \Longleftrightarrow (l = r)(a_1 = r + a_1 = S1)\dots(a_t = r + a_t = S1)(A_2 = S0)\dots(A_k = S0)(d = r + d = S1)$

$\Longleftrightarrow (l = r)(a_1 = r + a_1 = S1)\dots(a_t = r + a_t = S1)(A_2 = S0)\dots(A_k = S0)$

$((B_1 = r)(B_2 = r + B_2 = S0)\dots(B_m = r + B_m = S0) + \dots(B_1 = r + B_1 = S0)\dots(B_{m-1} = r + B_{m-1} = S0)(B_m = r) + B_1 = S1 + \dots B_m = S1)$ **(AIII.10)**

From Figure 4(a) we have $< l, c_i, g >_r \Longleftrightarrow (l = r)(a_1 = r + a_1 = S1)\dots(a_t = r + a_t = S1)(B_i = r + B_i = S1)\ (c_1 = S0)\dots(c_{i-1} = S0)(c_{i+1} = S0)\dots(c_m = S0)(q_1 = S0)\dots(q_m = S0)\dots(d_1 = S0)\dots(d_m = S0)$

$\Longleftrightarrow (l = r)(a_1 = r + a_1 = S1)\dots(a_t = r + a_t = S1)(B_i = r + B_i = S1)(a_1 = S0 + \dots + a_t = S0 + B_1 = S0)\dots(a_1 = S0 + \dots + a_t = S0 + B_{i-1} = S0)(a_1 = S0 + \dots + a_t = S0 + B_{i+1} = S0)\dots(a_1 = S0 + \dots + a_t = S0 + B_m = S0)(A_2 = S0 + B_1 = S0)\dots(A_2 = S0 + B_m = S0)\dots(A_k = S0 + B_1 = S0)\dots(A_k = S0 + B_m = S0)$ **(AIII.11)**

Let $Y = (l = r)(a_1 = r + a_1 = S1)\dots(a_t = r + a_t = S1)$. Then, $\forall i \neq j, Y(a_1 = S0 + \dots + a_t = S0 + B_j = S0) \Longleftrightarrow Y(B_j = S0)$. From this and (AIII.11) we have,

$< l, c_i, g >_r \Longleftrightarrow Y(B_i = r + B_i = S1)(B_1 = S0)\dots(B_{i-1} = S0)(B_{i+1} = S0)\dots(B_m = S0)(A_2 = S0 + B_1 = S0)\dots(A_2 = S0 + B_m = S0)\dots(A_k = S0 + B_1 = S0)\dots(A_k = S0 + B_m = S0)$ **(AIII.12)**

Now, $(B_j = S0)(A_2 = S0 + B_j = S0)\dots(A_k = S0 + B_j = S0) \Longleftrightarrow (B_j = S0), \forall i \neq j$. Also, $(B_i = r + B_i = S1)(A_2 = S0 + B_i = S0)\dots(A_k = S0 + B_i = S0) \Longleftrightarrow (B_i = r + B_i = S1)(A_2 = S0)\dots(A_k = S0)$. Using these identities in (AIII.12) we get (AIII.13).

$< l, c_i, g >_r \Longleftrightarrow Y(B_i = r + B_i = S1)(B_1 = S0)\dots(B_{i-1} = S0)(B_{i+1} = S0)\dots(B_m = S0)(A_2 = S0)(A_k = S0)$ **(AIII.13)**

$\Longleftrightarrow (l = r)(a_1 = r + a_1 = S1)\dots(a_t = r + a_t = S1)(B_i = r + B_i = S1)(B_1 = S0)\dots(B_{i-1} = S0)(B_{i+1} = S0)\dots(B_m = S0)(A_2 = S0)\dots(A_k = S0)$

$\Longleftrightarrow (l = r)(a_1 = r + a_1 = S1)\ldots(a_t = r + a_t = S1)(A_2 = S0)\ldots(A_k = S0)(B_1 = S0)\ldots(B_{i-1} = S0)(B_i = r + B_i = S1)(B_{i+1} = S0)\ldots(B_m = S0)$
$\Longleftrightarrow (l = r)(a_1 = r + a_1 = S1)\ldots(a_t = r + a_t = S1)(A_2 = S0)\ldots(A_k = S0)((B_1 = S0)\ldots(B_{i-1} = S0)(B_i = r)(B_{i+1} = S0)\ldots(B_m = S0) + (B_1 = S0)\ldots(B_{i-1} = S0)(B_i = S1)(B_{i+1} = S0)\ldots(B_m = S0))$
$\Longrightarrow (l = r)(a_1 = r + a_1 = S1)\ldots(a_t = r + a_t = S1)(A_2 = S0)\ldots(A_k = S0)((B_1 = S0 + B_1 = r)\ldots(B_{i-1} = S0 + B_{i-1} = r)(B_i = r)(B_{i+1} = S0 + B_{i+1} = r)\ldots(B_m = S0 + B_m = r) + (B_i = S1))$
$\Longrightarrow < l, A_1, c, f >_r, \forall i$ /* From AIII.10 */
Therefore, $< l, c_1, g >_r + \ldots < l, c_m, g >_r \Longrightarrow < l, A_1, c, f >_r$. □

Lemma 2.8. Let $p = p_1p_2p_3$ where p_2 is the *m-segment* $< B, h, f >$ of Figure 5(b). There exists a "hazard free" RT for $< p >_t, t \in \{r, f\}$ iff $\exists$ a "hazard free" RT for any $< q >_t$ where $q = p_1q_2p_3, q_2 \in \{< B, h, g >, \ldots, < B, k, g >\}$ is an *m-segment* in Figure 5(a).

Proof. We only need to show that $< B, h, g >_t + \ldots + < B, k, g >_t \Longrightarrow < B, h, f >_t, t \in \{r, f\}$. From (AII.5), in the proof of Lemma 2.2, we have $< B, h, g >_r \Longrightarrow < B, h, f >_r$. It can similarly be shown that $< B, k, g >_r \Longrightarrow < B, h, f >_r$. Therefore, $< B, h, g >_r + \ldots + < B, k, g >_r \Longrightarrow < B, h, f >_r$.

From AII.6, in the proof of Lemma 2.2, we have $< B, h, g >_f \Longrightarrow < B, h, f >_f$. It can be similarly shown that $< B, k, g >_f \Longrightarrow < B, h, f >_f$. Therefore, $< B, h, g >_f + \ldots + < B, k, g >_f \Longrightarrow < B, h, f >_f$. □

Lemma 2.9. Let $p = p_1p_{22}p_3$ be a *modified path* of C_a, p_{22} its *m-segment*, and $p_{22} \in S = \{< A, d, f >, < a_1, c, d, f >, \ldots < a_k, c, d, f >, < a_k, c, e, h, f >\}$ (Figure 5(a)). Then, $\exists$ a unique *modified path* $q = p_1q_{22}p_3$ s.t. V is a "hazard free" RT for $< p >_t$ iff V is a "hazard free" RT for $< q >_t$. For each $p_{22} \in S$, the unique q_{22} is shown in Table 2.

Proof. The lemma follows from identities AII.1 - AII.4 derived in the proof of Lemma 2.2. □

Table 1						
	a	b	c	d	e	f
T1	0	0	0	1	1	1
T2	0	0	0	1	0	1
T3	1	1	0	1	1	1
T4	1	1	0	1	0	1

Table 2 (DSCC)	
q_{22}	p_{22}
$< A, d, f >$	$< A, e, g >$
$< a_1, c, d, f >$	$< a_1, e, g >$
$\ldots$	$\ldots$
$< a_k, c, d, f >$	$< a_k, e, g >$
$< a_1, c, e, h, f >$	$< a_1, c, h, g >$
$\ldots$	$\ldots$
$< a_k, c, e, h, f >$	$< a_k, d, k, g >$
$< B, h, f >$	$< B, h, g >, \ldots, < B, k, g >$

Table 3 DSCC				
A	B	a_1	$\ldots$	a_k
		S0		
S0	S0	-	$\ldots$	-
S0	-	S1	$\ldots$	S1
-	S0	S0	$\ldots$	-
-	S0	-	$\ldots$	S0
		S1		
S1	-	S1	$\ldots$	S1
-	S1	S0	$\ldots$	-
		$\ldots$		
-	S1	-	$\ldots$	S0
		r		
r	S0	r,S1	$\ldots$	r,S1
S1	S0	r,S1	$\ldots$	r,S1
		$\ldots$		
S0	S1	f	$\ldots$	f,S1
S0	S1	f,S1	$\ldots$	f
S0	f	S1,f	$\ldots$	S1,f
		f		
f	S0	f,S1	$\ldots$	f,S1
S1	S0	f,S1	$\ldots$	f,S1
S0	S1	r	$\ldots$	r,S1
S0	S1	r,S1	$\ldots$	r
S0	r	S1,r	$\ldots$	S1,r

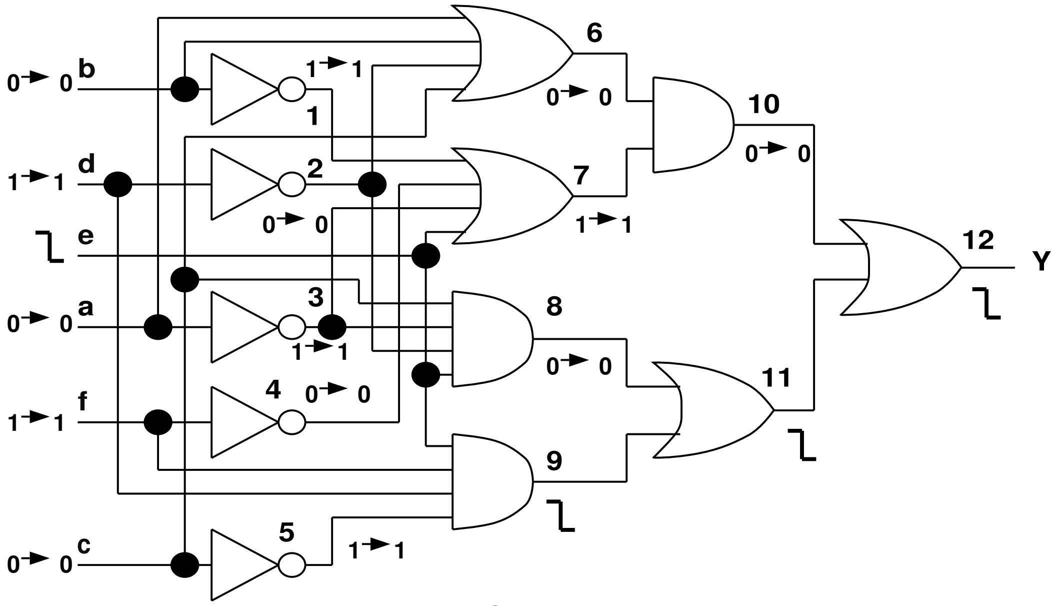
Table 4 XORC			
	S0		
a	b	A	B
-	-	S0	S0
S1	S1	S0	-
S0	S0	S0	-
S0	S1	-	S0
S1	S0	-	S0
	S1		
S1	S0	S1	-
S0	S1	S1	-
S0	S0	-	S1
S1	S1	-	S1
	r		
S0	S1	r	-
S1	S0	r	-
S0	r	r,S1	S0
f	S1	r,S1	S0
r	S0	r,S1	S0
f	S1	r,S1	S0
	f		
S0	S1	f	-
S1	S0	f	-
S0	f	f,S1	S0
r	S1	f,S1	S0
f	S0	f,S1	S0
r	S1	f,S1	S0

Table 5 D(2,2,3)C				
a	b	c	A	B
S0	S0	-	-	S0
-	S0	S0	-	S0
-	S0	-	S0	S0
S1	S0	S1	S0	-
		S1		
S0	S1	-	S1	-
S1	-	S1	S1	-
S0	S0	-	-	S1
S1	-	S0	-	S1
		r		
f	r,S1	S0	r,S1	S0
S0	r	S0	r,S1	S0
S0	S1	S0	r,S1	-
r,S1	S0	r,S1	r,S1	S0
S1	S0	S1	r,S1	-
		f		
r	f,S1	S0	f,S1	S0
S0	f	S0	f,S1	S0
S0	S1	S0	f,S1	-
f,S1	S0	f,S1	f,S1	S0
S1	S0	S1	f,S1	-

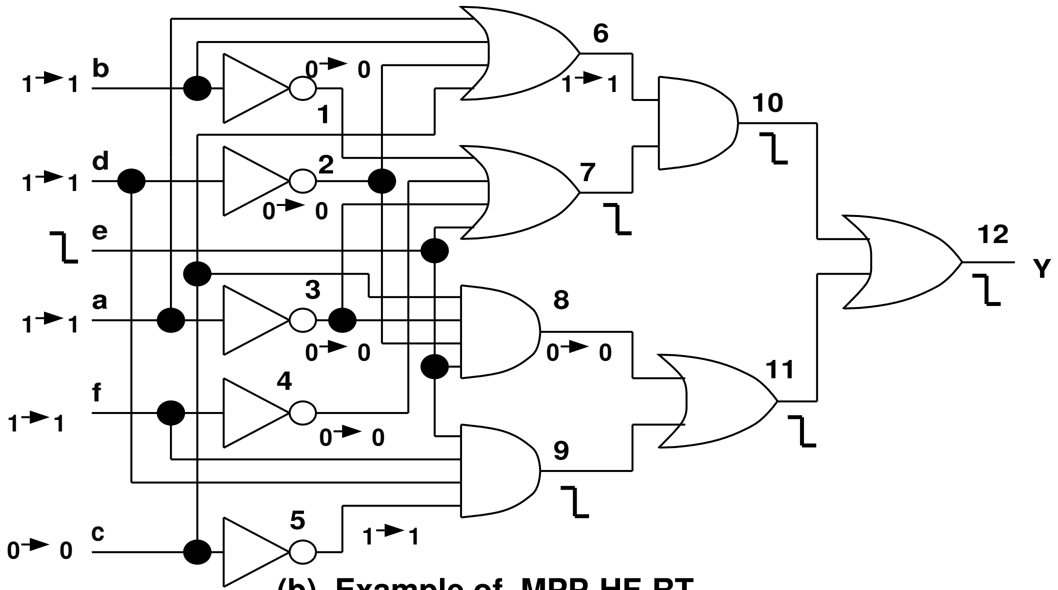
Table 6 (XORC)	
q_{22}	p_{22}
$< a, c, e, k, n, f >$	$< a, d, h, n, q, g >$
$< a, c, e, k, m, p, f >$	$< a, d, k, p, r, g >$
$< a, h, k, n, f >$	$< a, e, n, q, g >$
$< a, h, k, m, p, f >$	$< a, m, p, r, g >$
$< b, e, k, n, f >$	$< b, h, n, q, g >$
$< b, e, k, m, p, f >$	$< b, m, p, r, g >$
$< b, d, h, k, n, f >$	$< b, e, n, q, g >$
$< b, d, h, k, m, p, f >$	$< b, h, n, q, g >$
$< A, n, f >$	$< A, q, g >$
$< B, p, f >$	$< B, r, g >$

Table 7 (D(2,2,3)C)	
q_{22}	p_{22}
$< A, n, f >$	$< A, k, g >, < A, m, g >$
$< B, p, f >$	$< B, n, g >, < B, p, g >$
$< a, d, e, k, n, f >$	$< a, d, k, g >$
$< a, d, e, k, m, p, f >$	$< a, p, g >$
$< b, e, k, n, f >$	$< b, k, g >$
$< b, e, k, m, p, f >$	$< b, e, n, g >$
$< a, h, k, n, f >$	$< a, m, g >$
$< c, h, k, n, f >$	$< c, m, g >$
$< c, h, k, m, p, f >$	$< c, h, p, g >$

Table 8 (DSC)	
p_{i2}	q_{i2}
$< a_1, c_1, g >$	$< a_1, c, d_1, f >$
$< a_1, c_m, g >$	$< a_1, c, d_m, f >$
$\ldots$	$\ldots$
$< a_t, c_1, g >$	$< a_t, c, d_1, f >$
$\ldots$	$\ldots$
$< a_t, c_m, g >$	$< a_t, c, d_m, f >$



(a) Example of SPP-HF-RT.



(b) Example of MPP-HF-RT.

Figure 1: Illustration of SPP-HF, MPP-HF RTs.

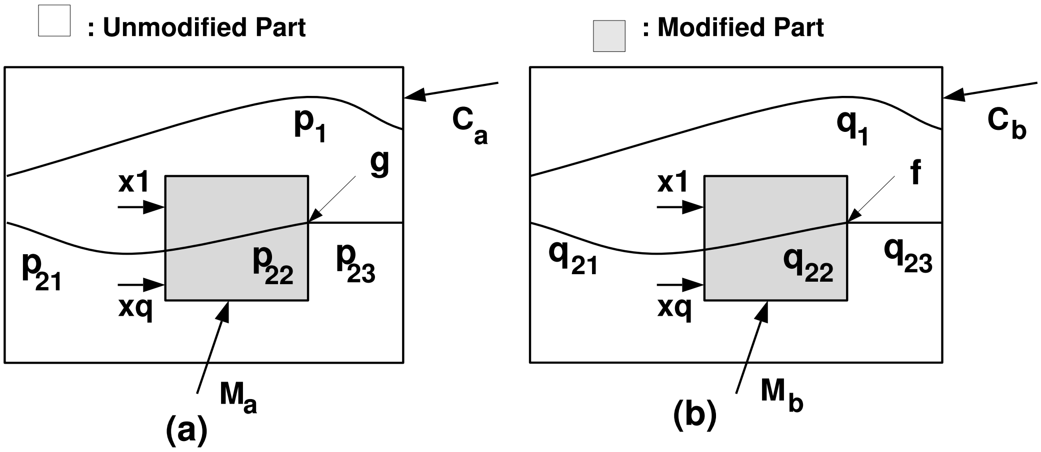


Figure 2: Atomic Transformations.

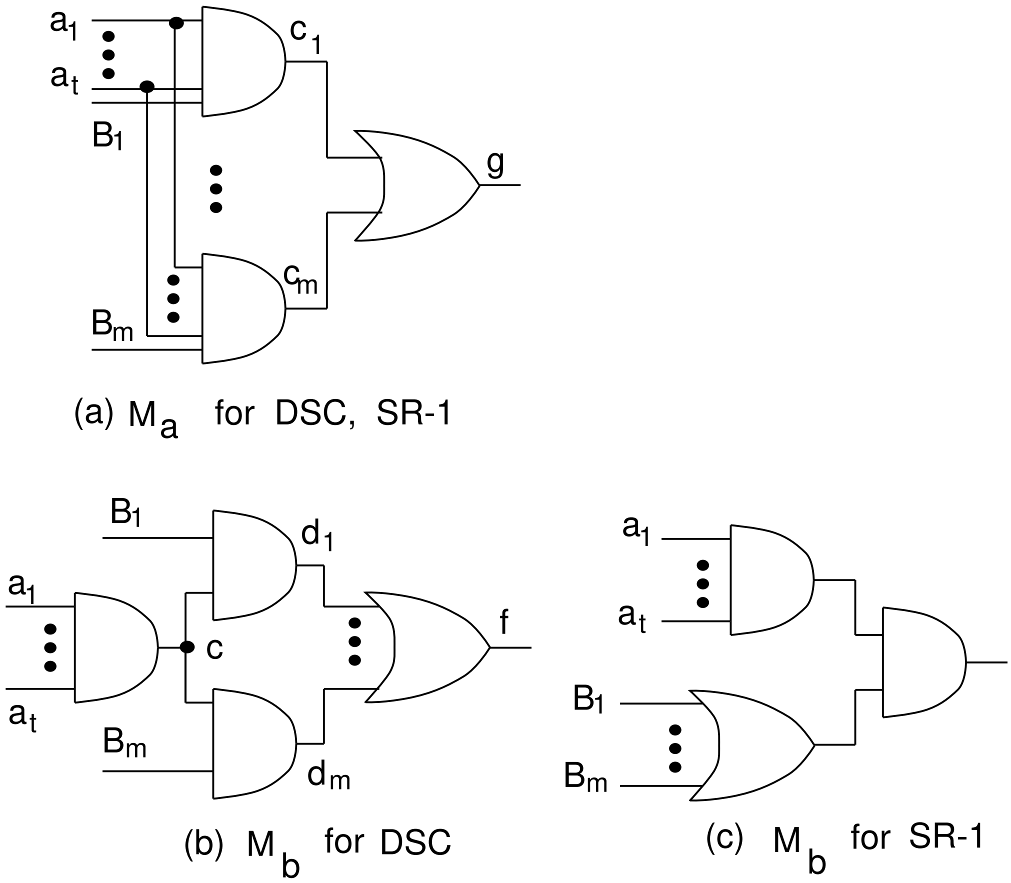


Figure 3: Division by Single Cube, SR-1.

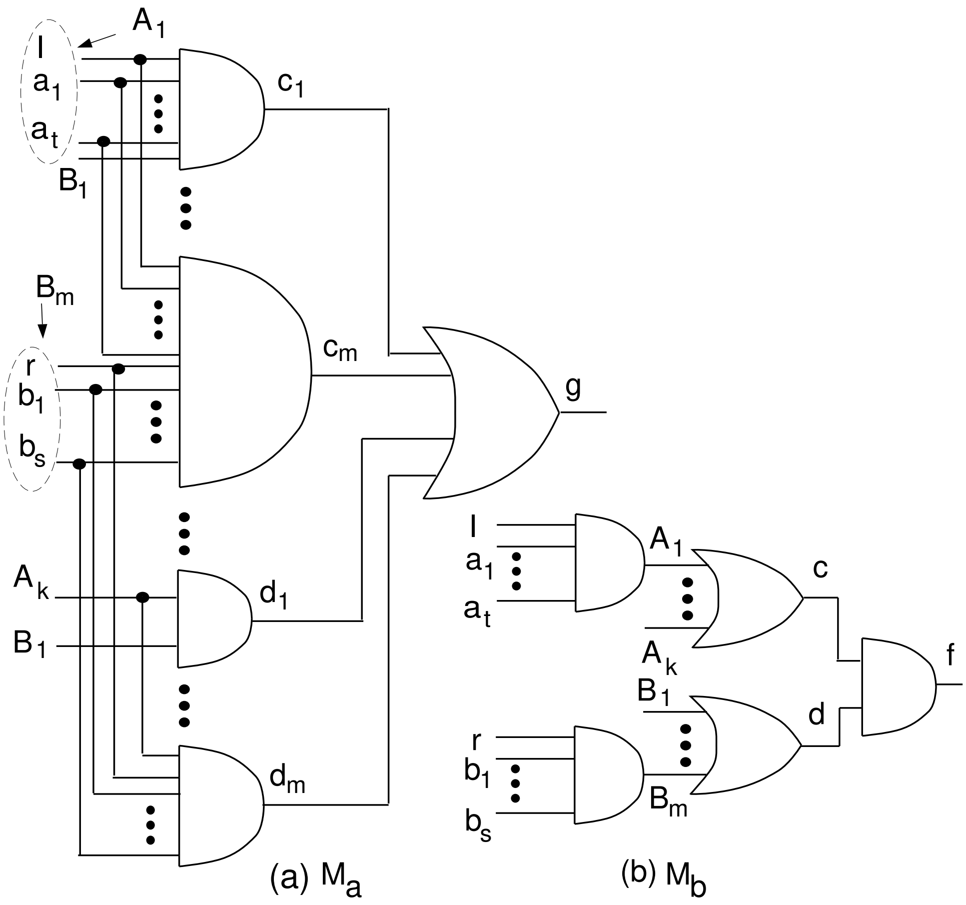


Figure 4: Division by Multiple Cube.

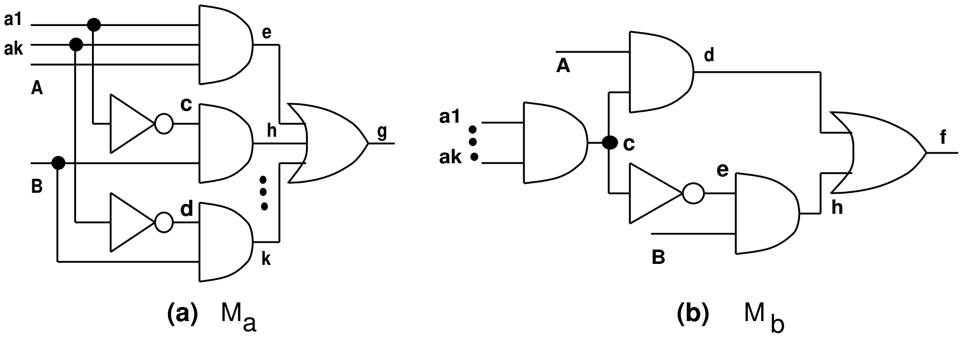


Figure 5: Division by a Single Cube and its Complement.

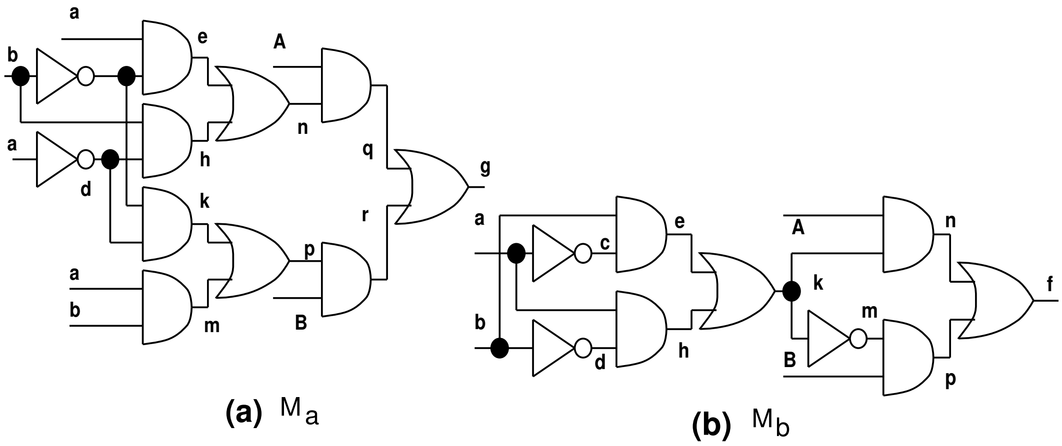


Figure 6: Division by XOR and its Complement.

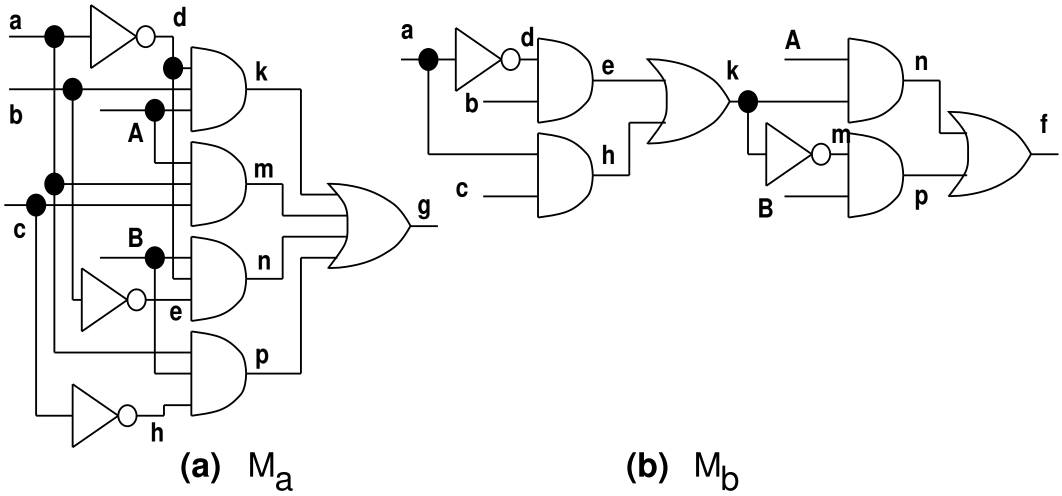


Figure 7: Division by $D(2,2,3)$ and its Complement.

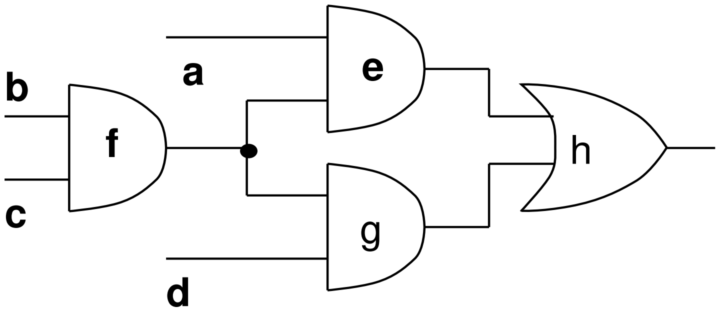


Figure 8:

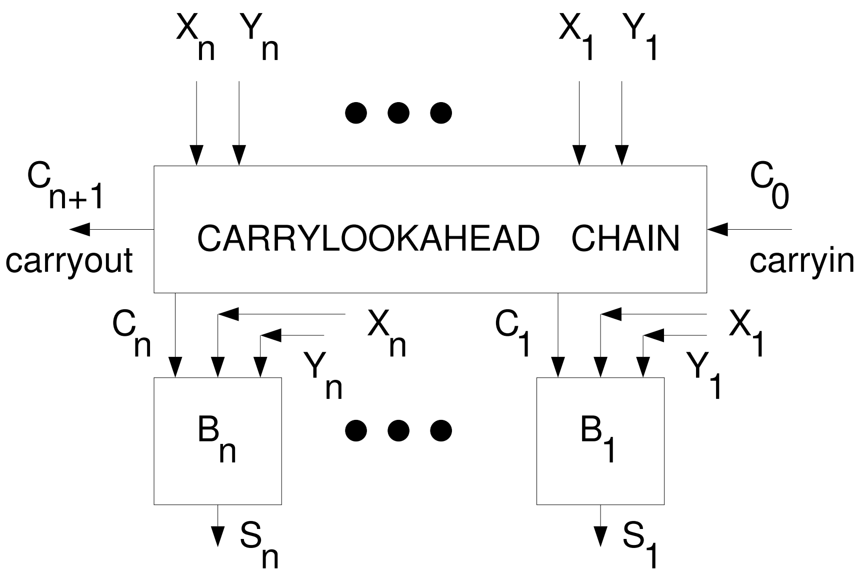


Figure 9: Block Diagram of a Carrylookahead Adder.

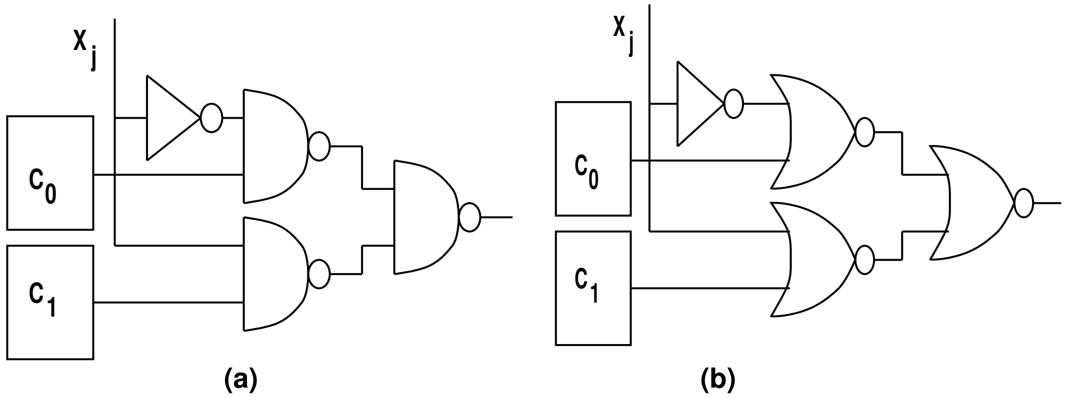


Figure 10: Decomposition Rules for Shannon's Expansion.

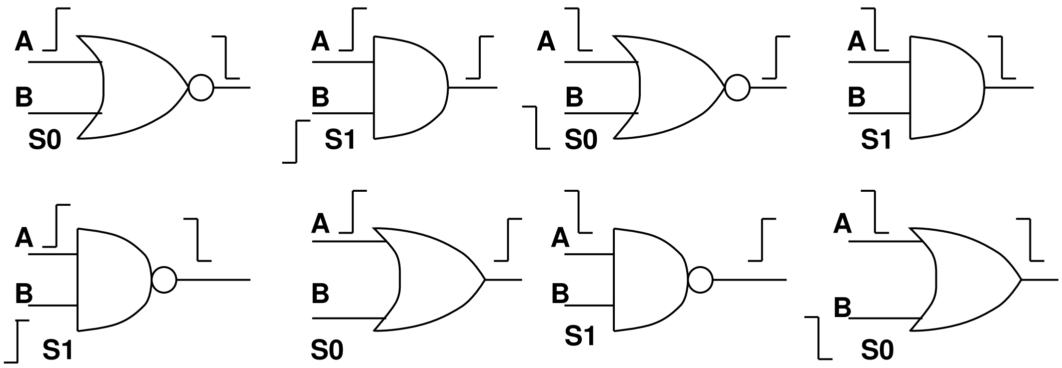


Figure 11: Possible Transitions at off-path inputs