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

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 Transformations; 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 $\Delta T$ (a function of the operating clock rate).

\footnote{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).

$T_1, T_2, T_3, T_4$ are defined in Table 1. $T = <T_1, T_2>$ is an SPP-HF-RT for $<e, 9, 11, 12>$ of Figure 1(a) because $T$ propagates a transition only along $<e, 9, 11, 12>$ (SPP) without any hazards. $<T_3, T_4>$ is an MPP-HF-RT both for $<e, 7, 10, 12>$ and $<e, 9, 11, 12>$ 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-

\footnote{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 theoretical 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 \( \{a, b, \overline{c}\} \) denoting the product term \( ab\overline{c} \). If a sum-of-product expression (SOP) consist of one cube it is a single cube else, it is a multiple cube (e.g. \( \overline{a}b + \overline{a}b \)).

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 \( qp + \overline{p}s + r \) where \( p, q, s, r \) are SOPs is Resubstitution
with Complement. If both (either) \(qp, \overline{ps}\) 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: \(A = a_1 \ldots a_t B_1 + \ldots + A_k B_1 + \ldots + A_k B_m\) is Division by Single Cube (DSC) (Figures 3(a), (b)). SR-1[8] (Figures 3(a), (c)) rewrites \(E\) as: \(A = a_1 \ldots a_t B_1 + \ldots + A_k B_m\).

Let \(A_i = B_{a_1} \ldots B_{a_i}; B_m = B_{rb_1} \ldots B_{sb_i}\) and \(g = A_1 B_1 + \ldots + A_k B_1 + \ldots + A_k B_m\) where: \(l, a, b, r, s\) are input literals and \(A_i, B_j\) 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 \(g = a_1 \ldots a_k A + \overline{\alpha}_1 B + \ldots + \overline{\alpha}_k B\), and \(M_b\) is defined as: \(c = a_1 \ldots a_k; f = Ac + B\overline{c}\). By “pushing out NOTs”[5] we get \(a_1 \ldots a_k\) from \(\overline{\alpha}_1 \ldots \overline{\alpha}_k\). Thus DSCC an algebraic resubstitution with “constrained” complement.

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

For D(2,2,3)C (Figure 7) \(M_a\) is defined as \(g = A\overline{\pi} \overline{b} + A\overline{c} + B\overline{\pi} \overline{b} + B\overline{\pi} a\), and \(M_b\) is defined as: \(k = \alpha \overline{\pi} \overline{b}; f = Ak + B\overline{\pi}\). It is an example of algebraic resubstitution with complement not satisfy the “constraint” because, \((\overline{\alpha} \overline{b} + \alpha \overline{c})\) cannot be derived from \(\overline{\alpha} \overline{b} + \alpha \overline{c}\) 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 \(M_a\) 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 $q_1$ in $C_b$ there exists a modified path $p_2$ in $C_a$ such that $V = < V_1, V_2 >$ 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** DSCC, 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. $\forall i \leq k, 1 \leq j \leq m, U_i, V_j$ 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 $A_1 = la_1 \ldots a_i$; 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 $< p >$, 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 $< q >$. Here, $q = p_1 p_2 p_3$ is a modified path in $C_b$ and $p_2 =< r, A_1, c, f >$ 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 $< p >$, 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 $< q >$. 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 $< p >$, $t \in \{r, f\}$ iff $\exists$ a MPP-HF RT for any $< q >$, 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 $p = p_1 p_2 p_3$ be a modified path of $C_a$, $p_2$ is its $m$-segment, and $p_2 \in S = \{< A, d, f >, < a_1, c, d, f >, \ldots, < a_k, c, d, f >, < a_k, c, e, h, f >\}$ (Figure 5(a)). Then, $\exists$ a unique modified path $q = p_1 q_2 p_3$ s.t. $V$ is a MPP-HF RT for $< p >$, $t$ iff $V$ is a MPP-HF RT for $< q >$. For each $p_2 \in S$, the unique $q_2$ 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 irredudant iff $\exists$ a test for all single stuck-at faults in it. Else, it is stuck-at redundant. A circuit is all path irredudant 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 $\text{adj}(T_i, X)$ differs from $T_i$ only in the value it assigns to $X$. In Table 1, $T_2 = \text{adj}(T_1, e)$ and $T_4 = \text{adj}(T_3, e)$. For a proof of Lemmas 3.1, 3.2 see Appendix 1.

**Lemma 3.1.** Let $X$ be a unate variable of $C$, $T_2$ any input vector and $V$ the input pair $< \text{adj}(T_2, X), T_2 >$. 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, $V_r^X = < \text{adj}(T_0^X, X), T_0^X > ( V_f^X = < \text{adj}(T_1^X, X), T_1^X > )$ is an MPP-HF RT for the DF input rising ( input falling ) along $P$.

In Figure 8, $T_1 = < a = 1, b = 1, c = 1, d = 1 >$ and $T_2 = < a = 1, b = 1, c = 1, d = 0 >$ are both tests for $< b, f > s-a-0$. However, $< \text{adj}(T_1, b), T_1 >$ is not a RT for a rising transition along $< b, f, e, h >$ but $< \text{adj}(T_2, b), T_2 >$ is. It does not contradict Lemma 3.2 because $T_1$ 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. $T_2$ of Table 1 is a test for line $< e, 9 > s-a-1$ (Figure 1(a)). RT $< \text{adj}(T_2, e), T_2 >$ is an SPP-HF test for the falling transition along $< e, 9, 11, 12 >$ (Figure 1(a)). Similarly, $< \text{adj}(T_4, e), T_4 >$, constructed from $T_4$ 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 $\beta$ was derived from a test for a stuck-at fault in another network $\beta'$. We construct RTs for a path in $\beta$ from a test for a stuck-at fault in the network $\beta$ 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 \( X_jE_0 + \overline{X_j}E_1 \) 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(\( F_{X_i} \)) and Robust-Testable-Design(\( F_{\overline{X_i}} \)) 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


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 \textit{off-path inputs} of \( P \). If \( P = < a, 6, 10, 12 > \) then \( < c, 6 >, < 7, 10 > \) are two of its \textit{off-path inputs}. Inputs of gates that lie along \( P \) constitute the \textit{on-path inputs} of path \( P \).

If \( k \) is an \textit{off-path input} of \( P \) and an input of gate \( a_j \) then \( k \) is an \textit{off-path input} of \( a_j \) w.r.t \( P \). Line \( < 7, 10 > \) is an \textit{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 \textit{on-path input} of path \( P \) then \( k \) is an \textit{on-path input} of \( a_j \) w.r.t. \( P \).

Signal states of a line \( w \), on application of a pair of input vectors \( < V_1, V_2 > \), are represented using multivalued logic\[11\] as follows: \( S0(S1) \) iff the steady state value at \( w \), on application of both \( V_1, V_2 \), 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 \( V_2 \), is \( 0(1) \) independent of the steady state value, on application of \( V_1 \), and the state of the signal during the transition interval.

**Lemma A1.1.** \[11\] \( V = < V_1, V_2 > \) is a \textit{robust test} for a \( 1 \rightarrow 0 \ ( 0 \rightarrow 1 \) along \( P \) iff \( V_1(a_1) = 1(0), V_2(a_1) = 0(1) \) and each gate \( a_j \) along \( P \) satisfies the following condition.

(i) There is a \( 0 \rightarrow 1 \) transition at the \textit{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 \textit{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 A1.2.** If no \textit{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 A1.1 then, \( V \) is an SPP-HF Robust test.

**Lemma 3.1.** Let \( X \) be a \textit{unate variable} of \( C, T2 \) any input vector and \( V \) the input pair \( < \text{adj}(T2, X), T2 > \). 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 \textit{rising} or all are \textit{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 \textit{rising transition} at input \( g_1 \) of \( G \) and a \textit{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, \( \not\exists \) 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 A1.1. 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 A1.1.

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 A1.1.

**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 A1.1. Hence, even in this case, $V^X_r$ 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 $0 \to 1$ transition arrives at the input of any gate $a_j$ that is traversed by $P$ then $V^X_r$ will robustly propagate that transition to the output of $a_j$. Similarly if a $1 \to 0$ transition arrives at $A$ then $V^X_r$ 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.

**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 $V = \langle V_1, V_2 \rangle$, we use the set $L = \{S_0, S_1, r, f, h, h_0, h_1\}$. Let $V_1(w), V_2(w)$ be the steady state values at $w$ on application of $V_1, V_2$ respectively. $V(w)$ has the value: $S_0(1)$ if $V_1(w) = V_2(w) = 0(1)$ and $w$ is hazard free; $h_0(1)$ if $V_1(w) = V_2(w) = 0(1)$ and there exists a hazard at $w$; $r(f)$ if $V_1(w) = 0(1), V_2 = 1(0)$ and there are no hazards at $w$; and $h_r(h_f)$ if $V_1(w) = 0(1), V_2 = 1(0)$ 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 then $XY$ 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 $V_1(g) = V_1(f)$ and $V_2(g) = V_2(f)$. Thus: $V(g) = S_1$ can imply either $V(f) = S_1$ or $V(f) = h_1$; $V(g) = S_0$ can imply either $V(f) = S_0$ or $V(f) = h_0$; $V(g) = r$ can imply either $V(f) = r$ or $V(f) = h_r$; and $V(g) = f$ can imply $V(f) = f$ or $V(f) = h_f$. 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: $V(g) = S_1 \implies V(f) = S_1, V(g) = S_0 \implies V(f) = S_0, V(g) = r \implies V(f) = r, \text{and } V(g) = f \implies V(f) = f$.

(ii) To show that $V(g) = a \iff V(f) = a, \forall a \in L$, we only need to show that: $V(g) = S_1 \iff V(f) = S_1, V(g) = S_0 \iff V(f) = S_0, V(g) = r \iff V(f) = r, \text{and } V(g) = f \iff V(f) = f$.

**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 $V = \langle V_1, V_2 \rangle$ 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-free" 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 All.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) \( \exists p_{22} \) of Figure 5(a). We next show that for each pair \( p_{22}, q_{22} \) of Table 2 \( < p_{22} >_t \iff < q_{22} >_t, t \in \{ r, f \} \). 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 first pair of Table 2. From Figure 5(a) \( < p_{22} >_t = < A, e, g >_r \)

\[ \iff (A = r)((a_1 = S1 + a_1 = r) \ldots (a_k = S1 + a_k = r))((h = S0) \ldots (k = S0)) \]

\[ \iff (A = r)((a_1 = S1 + a_1 = r) \ldots (a_k = S1 + a_k = r))((a_1 = S1 + B = S0) \ldots (a_k = S1 + B = S0)) \]

\[ \iff (A = r)((a_1 = S1 + a_1 = r) \ldots (a_k = S1 + a_k = r))(B = S0 + (a_1 = S1) \ldots (a_k = S1)) \]

\[ \iff (A = r)((a_1 = S1 + a_1 = r) \ldots (a_k = S1 + a_k = r))(B = S0 + (a_1 = S1) \ldots (a_k = S1) \ldots = \ldots) \]

16
\( S1 \))
\[ \iff (A = r)((c = S1 + c = r)(B = S0) + (c = S1)) \] from Figure 5(b) */
\[ \iff (A = r)((c = S1 + c = r)(h = S0) + (c = S1)(h = S0)) \iff A, d, f >_r= < q_{22} >_r . \] (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 
\[ \iff (A = r)((a_1 = S1) \ldots (a_k = S1))((h = S0) \ldots (k = S0)) \]
\[ \iff (A = r)((a_1 = S1) \ldots (a_k = S1))((a_1 = S1 + B = S0) \ldots (a_k = S1 + B = S0)) \]
\[ \iff (A = r)((a_1 = S1) \ldots (a_k = S1))(B = S0 + (a_1 = S1) \ldots (a_k = S1)) \]
\[ \iff (A = r)((a_1 = S1) \ldots (a_k = S1))((a_1 = S1)(B = S0) + (a_1 = S1) \ldots (a_k = S1)) \]
\[ \iff (A = r)(((c = S1)(B = S0) + (c = S1)) \] from Figure 5(b) */
\[ \iff (A = r)((c = S1)(h = S0) + (c = S1)(h = S0)) \iff A, d, f >_r= < q_{22} >_r . \]
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.
\[ \iff p_{22} >_f= < A, e, g >_f \iff (A = f)((a_1 = S1) \ldots (a_k = S1))((h = S0 + h = f) \ldots (k = S0 + h = f)) \]
\[ \iff (A = f)((a_1 = S1) \ldots (a_k = S1))((a_1 = S1 + B = S0 + (a_1 = r)(B = S1) + (a_1 = S0)(B = f) + (a_1 = r)(B = f)) \ldots (a_k = S1 + B = S0 + (a_k = r)(B = S1) + (a_k = S0)(B = f)) \]
\[ \iff (A = f)((a_1 = S1) \ldots (a_k = S1))((a_1 = S1 + B = S0) \ldots (a_k = S1 + B = S0)) \]
\[ \iff (A = f)((a_1 = S1) \ldots (a_k = S1))(B = S0 + (a_1 = S1) \ldots (a_k = S1)) \iff (A = f)(c = S1)(B = S0 + c = S1) \] from Figure 5(b) */
\[ \iff (A = f)(c = S1) \iff (A = f)(c = S1)(h = S0) \iff A, d, f >_f= < q_{22} >_f . \] (AII.2)
For the second pair of Table 2, from Figure 5(a) \( p_{22} >_r= < a_1, e, g >_r 
\[ \iff (a_1 = r)((A = S1 + A = r)(a_2 = S1 + a_2 = r) \ldots (a_k = S1 + a_k = r))((h = S0) \ldots (k = S0)) \]
\[ \iff (a_1 = r)((A = S1 + A = r)(a_2 = S1 + a_2 = r) \ldots (a_k = S1 + a_k = r))((a_1 = S1 + B = S0) \ldots (a_k = S1 + B = S0)) \]
\[ \iff (a_1 = r)((A = S1 + A = r)(a_2 = S1 + a_2 = r) \ldots (a_k = S1 + a_k = r))(B = S0 + (a_1 = S1) \ldots (a_k = S1)) \]
\[ \iff (a_1 = r)((A = S1 + A = r)(a_2 = S1 + a_2 = r) \ldots (a_k = S1 + a_k = r))(B = S0) \]
\[ \iff (a_1 = r)((A = S1 + A = r)(a_2 = S1 + a_2 = r) \ldots (a_k = S1 + a_k = r)(h = S0)) \] from Figure 5(b) */
\[ \iff < a_1, c, d, f >_r= < q_{22} >_r . \] (AII.3)
\[ \iff p_{22} >_f= < a_1, e, g >_f \iff (a_1 = f)((a_2 = S1) \ldots (a_k = S1)(A = S1)(h = f + h = S0) \ldots (k = f + k = S0)) \]
\[ \iff (a_1 = f)((a_2 = S1) \ldots (a_k = S1)(A = S1)\
(a_2 = 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))\
\iff (a_1 = f)(a_2 = S1) \ldots (a_k = S1)(A = S1)(B = S0) \ldots (a_k = S1 + B = S0)\
\iff (a_1 = f)(a_2 = S1) \ldots (a_k = S1)(A = S1)(B = S0) (h = S0) \}^* \text{ From Figure 5(b) } \}^*/
\iff < a_1, c, d, f > = < q_{22} > f. \hfill (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.

\[ < p_{22} > r = < B, h, g > r \iff (B = f)(a_1 = S0)(e = S0 + e = f) \ldots (k = S0 + k = f)\]
\iff (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_k = S1)(A = f + A = S1) + \ldots + (a_1 = f + a_1 = S1) + \ldots + (a_k = f + a_k = S1)(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))\
\iff (B = f)(a_1 = S0)(a_1 = S0 + \ldots + a_k = S0 + A = S0) \ldots (a_k = S1 + B = S0)(a_k = r) + (B = f)(a_k = S0))\
\iff (B = f)(a_1 = S0) \ldots (a_k = S1 + a_k = r + a_k = S0)\
\iff (B = f)(e = S0) \}^* \text{ From Figure 5(b) } \}^*/
\iff (B = f)(e = S1)(d = S0) \iff < B, h, f > \iff < q_{22} > f. \hfill (AII.5)

\[ < p_{22} > r = < B, h, g > r \iff (B = r)(a_1 = S0 + a_1 = f)(e = S0) \ldots (k = S0)\]
\iff (B = r)(a_1 = f + a_1 = S0)(a_1 = S0 + \ldots + a_k = S0 + A = S0)(a_2 = S0) \ldots (a_k = S0)\
\iff (B = r)(a_1 = f + a_1 = S0)(a_2 = S0) \ldots (a_k = S0)\
\iff (B = r)(e = S0) \}^* \text{ From Figure 5(b) } \}^*/
\iff (B = r)(e = S1)(d = S0) \iff < B, h, f > \iff < q_{22} > r. \hfill (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. \(< p_i > r, t \in \{ r, f \} \) has a “hazard free” RT if \( < q_i > r \) 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 $p_{ij}$ iff $V$ “robustly propagates”, without any hazard, the same transition $t$ along $q_{ij}$. Furthermore, if $p_{ij} \neq p_{j'j}$ then $q_{ij} \neq q_{j'j}$. For DSC this one-one correspondence is shown in Table 8. To complete the proof we show that, for each row of Table 8, $< p_{ij} >_t \iff < q_{ij} >_t, t \in \{r, f\}$.

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

$$< a_1, c_1, g >_t \iff (a_1 = f) (a_2 = S1) \ldots (a_t = S1) (B_1 = S1) (c_2 = S0 + c_2 = f) \ldots (c_m = f + c_m = S0)$$

$$\iff (a_1 = f) (a_2 = S1) \ldots (a_t = S1) (B_1 = S1)$$

$$(a_1 = f) (a_2 = f + a_2 = S1) \ldots (a_t = S1 + a_t = f) (B_2 = f + B_2 = S1) + \ldots +$$

$$(a_t = f) (a_1 = f + a_1 = S1) \ldots (a_{t-1} = S1 + a_{t-1} = f) (B_2 = f + B_2 = S1) +$$

$$(B_2 = f) (a_1 = f + a_1 = S1) \ldots (a_t = S1 + a_t = f))$$

$$\ldots$$

$$(a_1 = f) (a_2 = f + a_2 = S1) \ldots (a_t = S1 + a_t = f) (B_m = f + B_m = S1) + \ldots +$$

$$(a_t = f) (a_1 = f + a_1 = S1) \ldots (a_{t-1} = S1 + a_{t-1} = f) (B_m = f + B_m = S1) +$$

$$(B_m = f) (a_1 = f + a_1 = S1) \ldots (a_t = S1 + a_t = f))$$

$$\iff (a_1 = f) (a_2 = S1) \ldots (a_t = S1) (B_1 = S1)$$

$$(B_2 = S0 + (a_1 = f) (a_2 = S1) \ldots (a_t = S1) (B_2 = f + B_2 = S1))$$

$$\ldots$$

$$(B_m = S0 + (a_1 = f) (a_2 = S1) \ldots (a_t = S1) (B_m = f + B_m = S1))$$

$$\iff (a_1 = f) (a_2 = S1) \ldots (a_t = S1) (B_1 = S1)$$

$$(B_2 = S0 + B_2 = f + B_2 = S1) \ldots (B_m = S0 + B_m = S1 + B_m = f)$$

$$\iff (a_1 = f) (c = f) (B_1 = S1) (B_2 = S0 + B_2 = f + B_2 = S1) \ldots (B_m = S0 + B_m = S1 + B_m = f)$$

$$\iff (a_1 = f) (B_1 = S1) (d_2 = f + d_2 = S0) \ldots (d_m = f + d_m = S0) \iff < a_1, c, d, f >_t$$

$$< a_1, c_1, g >_t \iff (a_1 = r) (a_2 = S1 + a_2 = r) \ldots (a_t = S1 + a_t = r) (B_1 = S1 + B_1 = r) (c_2 = S0) \ldots (c_m = S0)$$

$$\iff (a_1 = r) (a_2 = S1 + a_2 = r) \ldots (a_t = S1 + a_t = r) (B_1 = S1 + B_1 = r) (B_2 = S0) \ldots (B_m = S0)$$

$$< a_1, c_1, d_1, f >_t \iff (a_1 = r) (a_2 = S1 + a_2 = r) \ldots (a_t = S1 + a_t = r) (B_1 = S1 + B_1 = r) (B_2 = S0) \ldots (d_m = S0)$$

$$\iff (a_1 = r) (a_2 = S1 + a_2 = r) \ldots (a_t = S1 + a_t = r) (B_1 = S1 + B_1 = r) (B_2 = S0) \ldots (B_m = S0)$$

From AI11.1, AI11.2 we have $< a_1, c_1, f >_t \iff < a_1, c_1, d_1, f >_t$.

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
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 $< p >$ of $C_a$ where: $p = p_1 p_2 p_3$ is a modified path in $C_a$; $q_2$ is an m-segment; and $q_3 \in \{< l, c_1, g, \ldots, l, c_m, g >\}$ then $V$ is also a “hazard free” RT for the DF $< q >$. 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. $p_3$ is an A-Segment.

Proof. To prove the lemma it is sufficient to show that $< l, c_1, g, \ldots, l, c_m, g > \iff < l, A_1, c, f >$. We use the conditions for MPP-HF-RT. From Figure 4(b),

\[
< l, A_1, c, f > \iff (l = f)(a_1 = S1)\ldots(a_t = S1)(a_2 = S0 + A_2 = f)\ldots(A_k = S0 + A_k = f)(d = S1) \\
\iff (l = f)(a_1 = S1)\ldots(a_t = S1)(a_2 = S0 + A_2 = f)\ldots(A_k = S0 + A_k = f)(B_1 = S1 + \ldots + B_m = S1)
\]

(AIII.3)

For Figure 4(a), $A_1 = l a_1 \ldots a_t$, $c_i = A_1 B_i$, $d_j = A_k B_j$, $1 \leq i,j \leq m$. In addition, define $q_2 = A_2 B_2$, $1 \leq j \leq m$

\[
< l, c_i, g > \iff (l = f)(a_1 = S1)\ldots(a_t = S1)(B_1 = S1)(c_i = S0 + c_i = f)\ldots(c_{i-1} = S0 + c_{i-1} = f)(c_{i+1} = S0 + c_{i+1} = f)\ldots(c_m = S0 + c_m = f)(q_1 = f + q_1 = S0)\ldots(q_m = f + q_m = S0)\ldots(d_1 = S0 + d_1 = f)\ldots(d_m = S0 + d_m = f)
\]

(AIII.4)

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

\[
X(c_1 = f + c_1 = S0)\ldots(c_{j-1} = f + c_{j-1} = S0)(c_{j+1} = f + c_{j+1} = S0)\ldots(c_m = f + c_m = S0) \iff X(B_1 = S0 + B_1 = f + B_1 = S1)\ldots(B_{i-1} = S0 + B_{i-1} = f + B_{i-1} = S1)(B_{i+1} = S0 + B_{i+1} = f + B_{i+1} = S1)\ldots(B_m = S0 + B_m = f + B_m = S1).
\]

From this and AIII.4 we get

\[
< l, c_i, g > \iff X(B_i = S1)(B_1 = S0 + B_1 = f + B_1 = S1)\ldots(B_{i-1} = S0 + B_{i-1} = f + B_{i-1} = S1)(B_{i+1} = S0 + B_{i+1} = f + B_{i+1} = S1)\ldots(B_m = S0 + B_m = f + B_m = S1)(q_1 = f + q_1 = S0)\ldots(q_m = f + q_m = S0)\ldots(d_1 = f + d_1 = S0)\ldots(d_m = f + d_m = S0).
\]

(AIII.5)

Now, $(d_j = f + d_j = S0) \iff (A_k = S0 + B_j = S0 + (B_j = S1)(A_k = f) + (B_j = f)(A_k = S1) + (B_j = f)(A_k = f))$.

(AIII.6)

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

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

(AIII.7)
\[ S_0 \ldots (q_m = f + q_m = S_0) \ldots (d_1 = f + d_1 = S_0) \ldots (d_{i-1} = f + d_{i-1} = S_0) (A_k = f + A_k = S_0) (d_{i+1} = f + d_{i+1} = S_0) \ldots (d_m = f + d_m = S_0). \]  
\[ \text{(AIII.7)} \]

From (AIII.6), for all \( i \neq j \), \( (d_j = f + d_j = S_0) (A_k = S_0 + A_k = f) \iff (A_k = S_0 + A_k = f) (A_k = S_0 + B_j = S_0 + (B_j = S_1) (A_k = f) + (B_j = f) (A_k = S_1) + (B_j = f) (A_k = f)) \iff (A_k = S_0 + A_k = f) (B_j = S_0 + B_j = S_1 + B_j = f). \] Similarly, for all \( i \neq j \), \( (q_j = f + q_j = S_0) (A_2 = S_0 + A_2 = f) \iff (A_2 = S_0 + A_2 = f) (B_j = S_0 + B_j = S_1 + B_j = f). \) From (AIII.7) and these two implications we get (AIII.8)

\[ \iff \text{l, c}_i, g > f \iff X(B_i = S_1) (B_1 = S_0 + B_1 = f + B_1 = S_1) \ldots (B_{i-1} = S_0 + B_{i-1} = f + B_{i-1} = S_1) (B_{i+1} = S_0 + B_{i+1} = f + B_{i+1} = S_1) \ldots (B_m = S_0 + B_m = f + B_m = S_1) (A_2 = f + A_2 = S_0) (A_k = f + A_k = S_0) \]  
\[ \text{(AIII.8)} \]

From (AIII.3) and (AIII.9) we have \( \iff \text{l, c}_1, g > \ldots \iff \text{l, c}_m, g > \iff \text{l, A}_1, c, f > f. \)  
\[ \text{(AIII.8)} \]

From Figure 4(b) we have \( \iff \text{l, A}_1, c, f > r \iff (l = r) (a_1 = r + a_1 = S_1) \ldots (a_i = r + a_i = S_1) (A_k = S_0) \iff (l = r) (a_1 = r + a_1 = S_1) \ldots (a_i = r + a_i = S_1) (A_2 = S_0) \ldots (A_k = S_0) \ldots (d_1 = S_0) \ldots (d_m = S_0) \]  
\[ \iff (l = r) (a_1 = r + a_1 = S_1) \ldots (a_i = r + a_i = S_1) (B_i = r + B_i = S_1) (c_1 = S_0) \ldots (c_{i-1} = S_0) (c_{i+1} = S_0) \ldots (c_m = S_0) (q_1 = S_0) \ldots (q_{m-1} = S_0) \ldots (d_1 = S_0) \ldots (d_m = S_0) \]  
\[ \iff (l = r) (a_1 = r + a_1 = S_1) \ldots (a_i = r + a_i = S_1) (B_i = r + B_i = S_1) (a_1 = S_0 + a_1 = S_0) \ldots (a_i = S_0 + B_{i-1} = S_0) (a_1 = S_0 + a_1 = S_0) \ldots (a_i = S_0 + B_{i-1} = S_0) \ldots (A_2 = S_0 + B_m = S_0) \ldots (A_k = S_0 + B_m = S_0) \ldots (A_k = S_0 + B_m = S_0) \]  
\[ \text{(AIII.10)} \]

Let \( Y = (l = r) (a_i = r + a_i = S_1) \ldots (a_i = r + a_i = S_1). \) Then, \( \forall i \neq j, Y (a_j = S_0 + a_j = S_0 + B_j = S_0) \iff Y (B_j = S_0). \) From this and (AIII.11) we have,

\[ \iff \text{l, c}_i, g > r \iff Y (B_i = r + B_i = S_1) (B_1 = S_0) \ldots (B_{i-1} = S_0) (B_{i+1} = S_0) \ldots (B_m = S_0) (A_2 = S_0 + B_1 = S_0) \ldots (A_2 = S_0 + B_m = S_0) \ldots (A_k = S_0 + B_1 = S_0) \ldots (A_k = S_0 + B_m = S_0) \]  
\[ \text{(AIII.11)} \]

Now, \( (B_j = S_0) (A_2 = S_0 + B_j = S_0) \ldots (A_k = S_0 + B_j = S_0) \iff (B_j = S_0), \forall i \neq j. \) Also, \( (B_i = r + B_i = S_1) (A_2 = S_0 + B_i = S_0) \ldots (A_k = S_0 + B_i = S_0) \iff (B_i = r + B_i = S_1) (A_2 = S_0) \ldots (A_k = S_0). \) Using these identities in (AIII.12) we get (AIII.13).

\[ \iff \text{l, c}_i, g > r \iff Y (B_i = r + B_i = S_1) (B_1 = S_0) \ldots (B_{i-1} = S_0) (B_{i+1} = S_0) \ldots (B_m = S_0) (A_2 = S_0) (A_k = S_0) \]  
\[ \iff (l = r) (a_1 = r + a_1 = S_1) \ldots (a_i = r + a_i = S_1) (B_i = r + B_i = S_1) (B_1 = S_0) \ldots (B_{i-1} = S_0) (B_{i+1} = S_0) \ldots (B_m = S_0) (A_2 = S_0) \ldots (A_k = S_0) \]  
\[ \text{(AIII.13)} \]
\[ \iff (l = r)(a_1 = r + a_1 = S1)\ldots(a_t = r + a_t = S1)(A_2 = S0)\ldots(A_k = S0)(B_1 = S0)\ldots(B_{i-1} = S0)(B_i = r + B_i = S1)(B_{i+1} = S0)\ldots(B_m = S0) \]
\[ \iff (l = r)(a_1 = r + a_1 = S1)\ldots(a_t = r + a_t = S1)(A_2 = S0)\ldots(A_k = S0)(B_1 = S0)\ldots(B_{i-1} = S0)(B_i = r)\ldots(B_m = S0) + (B_1 = S0)\ldots(B_{i-1} = S0)(B_i = S1)(B_{i+1} = S0)\ldots(B_m = S0) \]
\[ \iff (l = r)(a_1 = r + a_1 = S1)\ldots(a_t = r + a_t = S1)(A_2 = S0)\ldots(A_k = S0)(B_1 = S0 + B_1 = r)\ldots(B_{i-1} = S0 + B_{i-1} = r)(B_i = r)\ldots(B_m = S0 + B_m = r) + (B_i = S1) \]
\[ \iff < l, A_1, c, f >, \forall i \text{ */ From All.10 */} \]
Therefore, \[ < l, c_i, g >, +\ldots+ < l, c_m, g >, \iff < l, A_1, c, f >, . \square \]

**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 \( < p >, t \in \{ r, f \} \) iff \( \exists \) a "hazard free" RT for any \( < q >, \) 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 \( < B, h, g >, +\ldots+ < B, k, g >, \iff < B, h, f >, t \in \{ r, f \} \). From (All.5), in the proof of Lemma 2.2, we have \( < B, h, g >, \iff < B, h, f >, . \) It can similarly be shown that \( < B, k, g >, \iff < B, h, f >, . \) Therefore, \( < B, h, g >, +\ldots+ < B, k, g >, \iff < B, h, f >, . \)

From All.6, in the proof of Lemma 2.2, we have \( < B, h, g >, f \iff < B, h, f >, f \). It can be similarly shown that \( < B, k, g >, f \iff < B, h, f >, f \). Therefore, \( < B, h, g >, f +\ldots+ < B, k, g >, f \iff < B, h, f >, f \). \( \square \)

**Lemma 2.9.** Let \( p = p_1p_2p_3 \) be a modified path of \( C_a \), \( p_2 \) its \( m \)-segment, and \( p_{22} \in S = \{ < a_1, c, d, f >, \ldots < a_k, c, d, f >, < a_k, c, e, h, f > \} \) (Figure 5(a)). Then, \( \exists \) a unique modified path \( q = p_1q_2p_3 \) s.t. \( V \) is a "hazard free" RT for \( < p >, \) iff \( V \) is a "hazard free" RT for \( < q >, \). For each \( p_{22} \in S \), the unique \( q_{22} \) is shown in Table 2.

**Proof.** The lemma follows from identities All.1 - All.4 derived in the proof of Lemma 2.2. \( \square \)
<table>
<thead>
<tr>
<th>Table 1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>a</td>
</tr>
<tr>
<td>T1</td>
</tr>
<tr>
<td>T2</td>
</tr>
<tr>
<td>T3</td>
</tr>
<tr>
<td>T4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2 (DSCC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_{22}$</td>
</tr>
<tr>
<td>$&lt; A, d, f &gt;$</td>
</tr>
<tr>
<td>$&lt; a_1, c, d, f &gt;$</td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td>$&lt; a_k, c, d, f &gt;$</td>
</tr>
<tr>
<td>$&lt; a_1, c, e, h, f &gt;$</td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td>$&lt; a_k, c, e, h, f &gt;$</td>
</tr>
<tr>
<td>$&lt; B, h, f &gt;$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 3 (DSCC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
</tr>
<tr>
<td>----</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>S0</td>
</tr>
<tr>
<td>S0</td>
</tr>
<tr>
<td>-</td>
</tr>
<tr>
<td>-</td>
</tr>
<tr>
<td>S1</td>
</tr>
<tr>
<td>S1</td>
</tr>
<tr>
<td>-</td>
</tr>
<tr>
<td>-</td>
</tr>
<tr>
<td>r</td>
</tr>
<tr>
<td>r</td>
</tr>
<tr>
<td>S1</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>S0</td>
</tr>
<tr>
<td>S0</td>
</tr>
<tr>
<td>S0</td>
</tr>
<tr>
<td>f</td>
</tr>
<tr>
<td>f</td>
</tr>
<tr>
<td>S1</td>
</tr>
<tr>
<td>S0</td>
</tr>
<tr>
<td>S0</td>
</tr>
<tr>
<td>S0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 4 (XORC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0</td>
</tr>
<tr>
<td>----</td>
</tr>
<tr>
<td>-</td>
</tr>
<tr>
<td>S1</td>
</tr>
<tr>
<td>S0</td>
</tr>
<tr>
<td>S0</td>
</tr>
<tr>
<td>S1</td>
</tr>
<tr>
<td>S1</td>
</tr>
<tr>
<td>S1</td>
</tr>
<tr>
<td>S0</td>
</tr>
<tr>
<td>S0</td>
</tr>
<tr>
<td>S1</td>
</tr>
<tr>
<td>r</td>
</tr>
<tr>
<td>r</td>
</tr>
<tr>
<td>S1</td>
</tr>
<tr>
<td>S0</td>
</tr>
<tr>
<td>f</td>
</tr>
<tr>
<td>r</td>
</tr>
<tr>
<td>f</td>
</tr>
<tr>
<td>f</td>
</tr>
<tr>
<td>f</td>
</tr>
<tr>
<td>S1</td>
</tr>
<tr>
<td>S0</td>
</tr>
<tr>
<td>r</td>
</tr>
<tr>
<td>f</td>
</tr>
<tr>
<td>r</td>
</tr>
</tbody>
</table>

23
Table 5 (\text{D(2,2,3)C})

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0</td>
<td>S0</td>
<td>-</td>
<td>-</td>
<td>S0</td>
</tr>
<tr>
<td>-</td>
<td>S0</td>
<td>S0</td>
<td>-</td>
<td>S0</td>
</tr>
<tr>
<td>-</td>
<td>S0</td>
<td>-</td>
<td>S0</td>
<td>S0</td>
</tr>
<tr>
<td>S1</td>
<td>S0</td>
<td>S1</td>
<td>S0</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S1</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0</td>
</tr>
<tr>
<td>S1</td>
</tr>
<tr>
<td>S0</td>
</tr>
<tr>
<td>S1</td>
</tr>
</tbody>
</table>

\begin{array}{c|c|c|c|c}
\hline
\text{r} & \text{f} & \text{S1} & \text{S0} & \text{r} & \text{S1} & \text{S0} \\
\hline
\text{S0} & \text{r} & \text{S0} & \text{r} & \text{S1} & \text{S0} & \text{r} & \text{S1} & \text{S0} \\
\hline
\text{r} & \text{S1} & \text{S0} & \text{r} & \text{S1} & \text{S0} & \text{r} & \text{S1} & \text{S0} \\
\hline
\text{f} & \text{r} & \text{S1} & \text{S0} & \text{f} & \text{S1} & \text{S0} & \text{f} & \text{S1} & \text{S0} \\
\hline
\text{S1} & \text{S0} & \text{S1} & \text{S0} & \text{f} & \text{S1} & \text{S0} & \text{f} & \text{S1} & \text{S0} \\
\hline
\end{array}

Table 6 (\text{XORC})

<table>
<thead>
<tr>
<th>q_{22}</th>
<th>p_{22}</th>
</tr>
</thead>
<tbody>
<tr>
<td>\langle a, c, e, k, n, f \rangle</td>
<td>\langle a, d, h, n, q, g \rangle</td>
</tr>
<tr>
<td>\langle a, c, e, k, m, p, f \rangle</td>
<td>\langle a, d, k, p, r, g \rangle</td>
</tr>
<tr>
<td>\langle a, h, k, n, f \rangle</td>
<td>\langle a, e, n, q, g \rangle</td>
</tr>
<tr>
<td>\langle a, h, k, m, p, f \rangle</td>
<td>\langle a, m, p, r, g \rangle</td>
</tr>
<tr>
<td>\langle b, e, k, n, f \rangle</td>
<td>\langle b, h, n, q, g \rangle</td>
</tr>
<tr>
<td>\langle b, e, k, m, p, f \rangle</td>
<td>\langle b, m, p, r, g \rangle</td>
</tr>
<tr>
<td>\langle b, d, h, k, n, f \rangle</td>
<td>\langle b, e, n, q, g \rangle</td>
</tr>
<tr>
<td>\langle b, d, h, k, m, p, f \rangle</td>
<td>\langle b, h, n, q, g \rangle</td>
</tr>
<tr>
<td>\langle A, n, f \rangle</td>
<td>\langle A, q, g \rangle</td>
</tr>
<tr>
<td>\langle B, p, f \rangle</td>
<td>\langle B, r, g \rangle</td>
</tr>
</tbody>
</table>

Table 7 (\text{D(2,2,3)C})

<table>
<thead>
<tr>
<th>q_{22}</th>
<th>p_{22}</th>
</tr>
</thead>
<tbody>
<tr>
<td>\langle A, n, f \rangle</td>
<td>\langle A, k, g \rangle, \langle A, m, g \rangle</td>
</tr>
<tr>
<td>\langle B, p, f \rangle</td>
<td>\langle B, n, g \rangle, \langle B, p, g \rangle</td>
</tr>
<tr>
<td>\langle a, d, e, k, n, f \rangle</td>
<td>\langle a, d, k, g \rangle</td>
</tr>
<tr>
<td>\langle a, d, e, k, m, p, f \rangle</td>
<td>\langle a, p, g \rangle</td>
</tr>
<tr>
<td>\langle b, e, k, n, f \rangle</td>
<td>\langle b, k, g \rangle</td>
</tr>
<tr>
<td>\langle b, e, k, m, p, f \rangle</td>
<td>\langle b, e, n, g \rangle</td>
</tr>
<tr>
<td>\langle a, h, k, n, f \rangle</td>
<td>\langle a, m, g \rangle</td>
</tr>
<tr>
<td>\langle c, h, k, n, f \rangle</td>
<td>\langle c, m, g \rangle</td>
</tr>
<tr>
<td>\langle c, h, k, m, p, f \rangle</td>
<td>\langle c, h, p, g \rangle</td>
</tr>
</tbody>
</table>

Table 8 (\text{DSC})

<table>
<thead>
<tr>
<th>p_{ij}</th>
<th>q_{ij}</th>
</tr>
</thead>
<tbody>
<tr>
<td>\langle a_1, c_1, g \rangle</td>
<td>\langle a_1, c, d_1, f \rangle</td>
</tr>
<tr>
<td>\langle a_1, c_m, g \rangle</td>
<td>\langle a_1, c, d_m, f \rangle</td>
</tr>
<tr>
<td>\cdots</td>
<td>\cdots</td>
</tr>
<tr>
<td>\langle a_t, c_1, g \rangle</td>
<td>\langle a_t, c, d_1, f \rangle</td>
</tr>
<tr>
<td>\cdots</td>
<td>\cdots</td>
</tr>
<tr>
<td>\langle a_t, c_m, g \rangle</td>
<td>\langle a_t, c, d_m, f \rangle</td>
</tr>
</tbody>
</table>
(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.

(a) $M_a$ for DSC, SR-1

(b) $M_b$ for DSC

(c) $M_b$ for SR-1

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