Easily Path Delay Fault Testable Non – Restoring Cellular Array Dividers

G. Sidiropoulos[1,2], H. T. Vergos[1,2] & D. Nikolos[1,2]
[1]Dept. of Computer Engineering and Informatics,University of Patras, 26 500, Patras, Greece
[2]Computer Technology Institute,3, Kolokotroni St., 262 21 Patras, Greece
E-mail : sidiro@ceid.upatras.gr, {vergos, nikolosd}@ cti.gr

Abstract

Testing of NxN Non-Restoring Cellular Array Dividers (NRCAD) with respect to path delay faults, is studied in this paper. Design modifications are proposed and a path selection method is suggested. We prove that the selected paths are Single Path Propagating Hazard Free Robustly Testable (SPP-HFRT) and that by measuring their delays the delay along any other path of the divider can be easily calculated. The number of selected paths is impressively small compared to all paths of the divider. The delay overhead of the modified design for all values of N is negligible, while the hardware overhead is small too. This is the first easily testable, with respect to path delay faults, NRCAD design in the open literature.

1. Introduction

The speed at which a circuit can operate is determined by the maximum delay along any path. As chip manufacturing process becomes more sophisticated, the major failure being experienced shifts from stuck fault to delay type [1]. Imprecise delay modeling, the statistical variations of the parameters during the manufacturing process as well as the occurrence of physical defects in the integrated circuits result in chip malfunction at the desired speed. Gross delay faults [2], model delay defects that affect single lines in the circuit, causing the propagation delay through them to be "very large". The gate delay fault model [3] also addresses defects affecting single lines, however, no assumption is made on the delay size. The path delay fault model [4] addresses distributed or accumulated delays due to the propagation through several lines, each affected by a delay defect. Two major problems are associated with path delay fault testing : a) an excessively large number of physical paths needs to be tested. Usually it is not affordable to test all of them and b) because the single fault assumption is not realistic for the path delay fault model (a single defect usually will affect a large number of paths), a robust test is usually required for detecting a path delay fault. However, for many circuits, a large number of path delay faults can not be robustly tested.

A variety of path selection methods have been proposed [6-12] to alleviate problem a) above. The simplest delay-based approach is to select all paths whose calculated delay exceeds a specific threshold. However, the number of paths selected by this method is so large in general that all the selected paths cannot be tested, especially in the case of optimized circuits [5]. The method given in [7] selects a set of paths such that for each interconnect l of a given circuit the set contains at least one path with the largest calculated delay among the paths through l. The number of the paths selected by this method is moderate. The method, recently proposed in [8], reduces the number of paths to be tested by judging which of two paths has the larger real delay. A common drawback of the above mentioned methods [7, 8] is that the selected paths may not be robustly testable.

A number of functional approaches has also been proposed [9-12]. In those, path selection is achieved by excluding paths that do not have to be tested functionally, such as unsensitizable paths, robust-dependent paths etc. However, these approaches are not practical for large circuits because both computation time and the number of the selected paths are quite large.

It has been shown in [6] that by measuring the delays along a suitable very small set R of physical paths the propagation delays along any other path can be calculated (we will hereafter call such a set of paths a basis). However, to be able to measure the propagation delay along the R paths they must be SPP-HFRT [6]. Unfortunately for most circuits, a basis consisting of SPP-HFRT paths does not exist.

Almost all contemporary general and special purpose processors include a high speed divider circuit. Array divider architectures feature regularity, short execution time and small area; thus they are suitable for VLSI implementations [13]. Testing them for path delay faults is a very difficult task due to: a) their excessively large number of physical paths and b) the fact that all path delay faults can not be robustly tested. In this paper, we focus on N x N NRCADs, originally introduced in [14]. The methods given in [7, 8] can not be applied in the case of a

NRCAD, because many of the longer paths are neither robustly testable nor non-robustly validatable. Also, a basis R, according to [6], consisting of SPP-HFRT paths does not exist for a NRCAD.

In this paper, we propose modifications of the original NRCAD design that enable us to present a method for deriving a basis R' consisting only of SPP-HFRT paths. The number of paths that R' includes is an impressively small percentage of all physical paths of the original design. However due to the prohibitively large number of paths in an NRCAD it is impossible to calculate the delay along all paths in order to derive the maximum delay. This problem can be overcome using the method proposed in [8] to determine a relatively small number of paths T the propagation delay along which must be calculated in order the maximum path delay of the circuit under test to be derived.

It has been shown in [17] that the fact that the circuit functions correctly at a clock speed does not imply that it will also function correctly at any lower clock speed. Therefore, calculating the propagation delays along the paths of T we derive the maximum clock speed under which the NRCAD will operate correctly but we can not be sure that the NRCAD will operate correctly under a lower clock speed. It has been shown in [17] that if we can test all primitive path delay faults of a circuit at a clock speed and under test application no delay fault is detected then the circuit functions correctly at any lower speed. We will show that calculating the propagation delay along all paths included in primitive faults we derive the maximum delay of the circuit (as well as the maximum clock speed), and the circuit functions correctly for all lower clock speeds. Therefore we show that the proposed NRCAD is delay-verifiable.

We consider the inputs and outputs of the divider as primary inputs (PIs) and primary outputs (POs) of the chip. In the case that the divider is embedded in a circuit, its inputs and outputs can easily be made accessible by the PIs and the POs of the chip using, for example, the method proposed in [15].

2. Delay-verifiable circuits

It has been shown in [17] that the fact that a circuit functions correctly at a clock speed does not imply that it will also function correctly at any lower clock speed. A set of path delay tests is called a strong delay-verification test set if the correct response of the CUT at a speed implies correct operation at any lower speed [17]. A circuit which has a strong delay-verification test set is called a delay-verifiable circuit [17].

Figure 1 [17] shows a circuit which does not have a strong delay-verification test set. All faults except $\uparrow \bar{b}df$, $\downarrow \bar{b}df$, $\uparrow \bar{b}ef$ and $\downarrow \bar{b}ef$ are testable by robust tests. In this case, even the exhaustive set consisting of all the vector pairs is not a strong delay verification test set. The signal values in the circuit for the two tests <101,111> and <111,101> are shown in Figures 1.a and 1.b, respectively. If there are no path delay faults, any output pulse that may occur will occur before the sampling time t_2. Faults on the paths from b and $\bar{b}$ may result in an output pulse occurring later. Such faults may or may not be detected at time t_2. Therefore, the correct response for these two tests only guarantees that the circuit will operate correctly if the clock period is the test clock period τ, but the delayed pulse due to the path delay fault may cause incorrect operation at a lower clock speed.

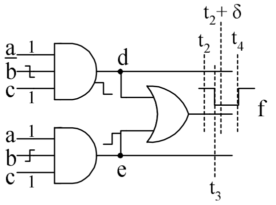

Figure 1.a.

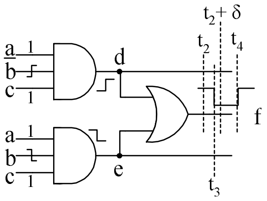

Figure 1.b.

Although the circuit of Figure 1 does not have a strong-delay verification test set, we will show that the circuit is delay-verifiable. The propagation delay along the paths ad, adf, ae, aef, $\bar{b}d$ and be which are SPP-HFRT [16] can be measured applying the test vector pairs <001,101>, <001,101>, <011,111>, <011,111>, <101,111> and <101,111> respectively. The propagation delay, pd, along the paths bdf and bef can be calculated by: $pd(\uparrow \bar{b}df) = pd(\uparrow \bar{b}d) + pd(\uparrow adf) - pd(\uparrow ad)$, $pd(\downarrow \bar{b}df) = pd(\downarrow \bar{b}d) + pd(\downarrow adf) - pd(\downarrow ad)$, $pd(\uparrow bef) = pd(\uparrow be) + pd(\uparrow aef) - pd(\uparrow ae)$ and $pd(\downarrow bef) = pd(\downarrow be) + pd(\downarrow aef) - pd(\downarrow ae)$.

For the output waveforms of figures 1.a and 1.b we get $pd(\uparrow \bar{b}df)=t_4$, $pd(\downarrow \bar{b}df)=t_3$, $pd(\uparrow bef)=t_4$, $pd(\downarrow bef)=t_3$, therefore the maximum delay of the circuit is equal to t_4 and for all lower speeds the circuit will function correctly. From the above discussion it becomes evident that for a circuit having a basis consisting only of SPP-HFRT paths we can calculate the delay along all paths or along the paths included in primitive faults [18] and the calculated maximum propagation delay implies that the circuit will function correctly for any lower clock speed; therefore is delay verifiable.

3. NRCAD Design Modifications

An N x N NRCAD is a combinational circuit with inputs the nominator (n_1, n_2, ... n_{2N-1}) and the denominator (d_1, d_2, ... d_N) and outputs the quotient (q_1, q_2, ... q_N) and the remainder (r_N, r_{N+2}, ... r_{2N-1}). Figure 2 presents the 8x8 NRCAD (ignore the multiplexers at the bottom of Figure

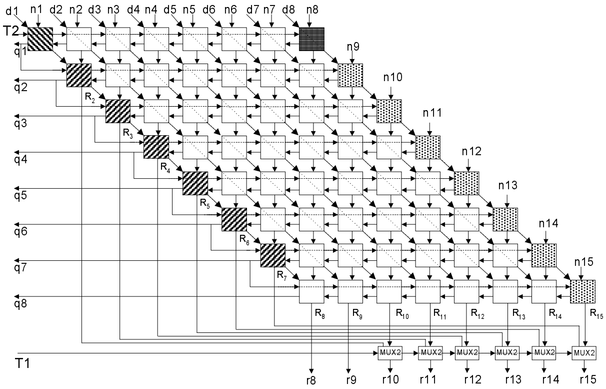
d1 n1 d2 n2 d3 n3 d4 n4 d5 n5 d6 n6 d7 n7 d8 n8
T2
q1
q2
q3
q4
q5
q6
q7
q8
n9
n10
n11
n12
n13
n14
n15
R2
R3
R4
R5
R6
R7
R8 R9 R10 R11 R12 R13 R14 R15
T1
MUX2
r8 r9 r10 r11 r12 r13 r14 r15

Figure 2. 8 x 8 NRCAD.

2). The dashed lines indicate propagation of the signals to the next cell in either horizontal or diagonal direction. The NRCAD is formed as a two dimensional matrix of identical logic cells. Each cell is denoted as cell(x, y), with x indicating the row and y the column that the cell belongs. The implementation of each cell of the divider considered in this paper is presented in Figure 3.a and requires 19 gates, without considering the dashed gate. Since the first row cells have their P input driven by logical 1, they can be implemented by 15 gates. Since the leftmost cell of each row, except the last, does not produce an S output it can be implemented by 9 gates as shown in figure 3.b. The upper and leftmost cell can be implemented by 5 gates. Since the rightmost cell of each row, except the first, has connected P and C_i inputs, it can be implemented by 9 gates as shown in Figure 3.c, without considering the dashed gate. The upper and rightmost cell requires only 8 gates. Summarizing the above, we can express the total area of the divider in gates as A_{total} = 19*[(N-1)*(N-2)+1]+15*(N-2)+9*(N-2)+5+9*(N-1)+8.

In the sequel we propose several design modifications for making the NRCAD design easily testable. Excluding the leftmost cells of all rows and the cells of last row, we augment every other cell with an extra AND gate (the dashed AND gate of figures 3.a, 3.c). An extra test input T_0 is used to drive the second input of the added AND gate for all cells. T_0 is only used during testing. During normal circuit operation T_0 is driven to 1. The hardware overhead of the addition of the AND gate in terms of gate equivalents is (N-1)2/A_{total}. The above relation, for N = 8, 16 and 32 leads to a hardware overhead of 4.7, 5.0 and 5.14% respectively. The critical path of the design is from a primary input of the upper and rightmost CAS cell, through the carry chains of each of the N levels of the NRCAD. The chains of two adjacent levels are connected together through the C_{i+1} output of the leftmost cell which is connected to P input of the rightmost cell of next row. The AND gates that we have added do not add any delay on this critical path and all the rest paths that include any sub-path along an added AND gate have smaller propagation delay times than this critical path.

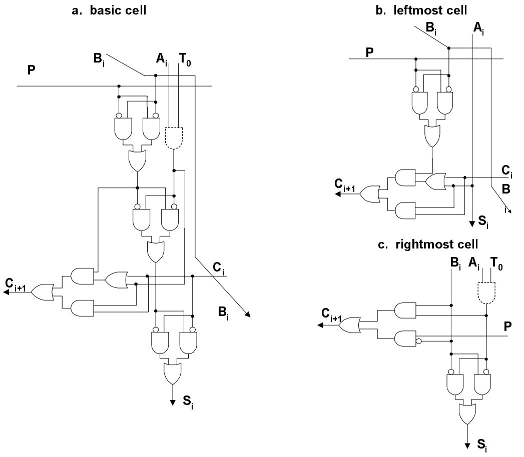

Figure 3. Building blocks of the NRCAD

For providing observability of the S output of the next to the leftmost cell of each row, excluding the first and last rows, we include N-2 2->1 multiplexers to the NRCAD design. All these multiplexers are controlled by the same test input T_1 and connected as shown in Figure 2. T_1 during normal operation is driven to 1 and the remainder bits are observable at the primary outputs, whereas during testing of specific paths is driven to 0 and the S outputs of the next to the leftmost cells become observable at the remainder outputs of the divider. The output of the multiplexer which drives the r_{i+N} primary output is denoted as O_{i+N}. Since the hardware implementation of the multiplexer requires 4 gate equivalents, the hardware overhead due to multiplexer insertion is: 4*(N-2)/A_{total}. This relation, for N = 8, 16 and 32 leads to a hardware overhead of 2.3, 1.4 and 0,64 % respectively. Our simulations showed that the insertion of the multiplexers does not cause any delay overhead.

For providing controllability of the P input for the cells of the first row, we drive all these inputs by a third test input T_2. T_2 is set to 1 for normal circuit operation and is occasionally driven to 0 during testing. The hardware overhead of this change is equal to 4 gates per cell of the first row. Thus the hardware overhead is (4*N)/A_{total} or equivalently 3.0, 1.4 and 0.68 % respectively for N= 8, 16 and 32. This change may increase the critical path of the design by a time equal to the difference of the worst propagation delays between an XOR gate and an inverter, which is overall a negligible delay.

Summarizing the above analysis the proposed modifications induce a small hardware overhead, which for N = 8, 16 and 32 is respectively equal to 10%, 7.8% and 6.5% in terms of gates. Three extra test inputs are required. The delay overhead is negligible.

4. Basis Derivation of the Modified NRCAD

In this section we present a path selection among the paths of the modified NRCAD.

Table 1. Subpaths along each cell.

From	To	Other Input Signal Values	Notation	From	To	Other Input Signal Values	Notation
A	S	P=0, B=1, C=1 / P=1, B=0, C=1	a	P	S	B=0, A=0, C=1	$\bar{f}$
A	S	P=0, B=1, C=0 / P=1, B=0, C=0	b	P	S	B=0, A=1, C=0	$\bar{g}$
A	S	P=0, B=0, C=1 / P=1, B=1, C=1	c	P	S	B=0, A=1, C=1	$\bar{h}$
A	S	P=0, B=0, C=0 / P=1, B=1, C=0	d	P	S	B=1, A=0, C=0	$\bar{i}$
A	S	B=0	a_R	P	S	B=1, A=0, C=1	$\bar{j}$
A	S	B=1	b_R	P	S	B=1, A=1, C=0	$\bar{k}$
B	S	P=0, A=0, C=0	e	P	S	B=1, A=1, C=1	$\bar{l}$
B	S	P=0, A=0, C=1	f	A	C	P=0, B=1, C=0 / P=1, B=0, C=0	o
B	S	P=0, A=1, C=0	g	A	C	P=0, B=0, C=1 / P=1, B=1, C=1	p
B	S	P=0, A=1, C=1	h	A	C	B=1, P=0 / B=1, P=1	o_R
B	S	P=1, A=0, C=0	i	B	C	P=0, A=0, C=1 / P=0, A=1, C=0	q
B	S	P=1, A=0, C=1	j	B	C	P=1, A=0, C=1 / P=1, A=1, C=0	r
B	S	P=1, A=1, C=0	k	B	C	P=0, A=1	q_R
B	S	P=1, A=1, C=1	l	B	C	P=1, A=0	r_R
B	S	A=0	e_R	C	C	P=0, B=1, A=0 / P=1, B=0, A=0	$\bar{o}$
B	S	A=1	f_R	C	C	P=0, B=0, A=1 / P=1, B=1, A=1	$\bar{p}$
C	S	P=0, B=0, A=0 / P=0, B=1, A=1 / P=1, B=0, A=1 / P=1, B=1, A=0	m	P	C	B=0, A=0, C=1 / B=0, A=1, C=0	$\bar{q}$
C	S	P=0, B=0, A=1 / P=0, B=1, A=0 / P=1, B=0, A=0 / P=1, B=1, A=1	n	P	C	B=1, A=0, C=1 / B=1, A=1, C=0	$\bar{r}$
P	S	B=0, A=0, C=0	$\bar{e}$	P	C	A=0, B=0 / A=1, B=0	$\bar{q}_R$

Table 2. Calculation procedure for each cell subpaths.

Path	Calculation	Path	Calculation	Path	Calculation
c	b + d – a	j	i + a – b	$\bar{i}$	$\bar{j}$ + i – j
f	e + a – b	k	l + b – a	$\bar{k}$	$\bar{j}$ + k – j
g	e + l – i	$\bar{f}$	$\bar{e}$ + f – e	$\bar{l}$	$\bar{j}$ + l – j
h	g + b – a	$\bar{g}$	$\bar{e}$ + g – e	$\bar{q}$	$\bar{e}$ + q – e
I	j + b – a	$\bar{h}$	$\bar{e}$ + h – e	$\bar{r}$	$\bar{j}$ + r – j

Table I lists the physical subpaths along a cell with their notation. The index $_R$ is used for paths of the rightmost cells. Considering a single cell, there is no need to measure the propagation delays along all these subpaths, since the propagation delays for c, f, g, h, i, j, k, $\bar{f}$, $\bar{g}$, $\bar{h}$, $\bar{j}$, $\bar{k}$, $\bar{l}$, $\bar{q}$ and $\bar{r}$ subpaths can be calculated, if the propagation delay of the rest subpaths is known. Table II lists the expressions for computing the delay along these subpaths. For example, as Table II indicates, the propagation delay pd along the c subpath can be calculated by : pd(c)=pd(b)+pd(d)– pd(a).

For the clarity of the analysis, we group the physical subpaths of any cell in sets and we define the following variables: $s \in \{a, b, c, d\}$, $s_R \in \{a_R, b_R\}$, $t \in \{e, f, g, h, j, i, k, l\}$, $t_R \in \{e_R, f_R\}$, $u \in \{m, n\}$, $v \in \{\bar{e}, \bar{f}, \bar{g}, \bar{h}, \bar{i}, \bar{j}, \bar{k}, \bar{l}\}$, $w \in \{o, p\}$, $w_R = o_R$, $x \in \{q, r\}$, $x_R \in \{q_R, r_R\}$, $y \in \{\bar{o}, \bar{p}\}$, $z \in (\bar{q}, \bar{r}\}$ and $z_R = \bar{q}_R$.

Definition 1 : A path is denoted as a tuple, (a, b, c) where a is the primary input, b describes the cells sequence which the path traverses and c is the primary output.

Definition 2 : The length of a path is the number of cells that the path traverses. We will denote the length of a path p as $|p|$.

We will examine fourteen cases. In each case, the paths selected along with some belonging to previous cases constitute a basis for the whole set of paths of the case [19] :

Case A. Let P_A be the set of paths that have the form: $n_i \rightarrow S(1,i) \rightarrow S(2,i) \rightarrow \ldots \rightarrow S(i-1,i) \rightarrow O_{i+N} \equiv r_{i+N}$, 1<i<N. By definition 1 these paths can be described as (n_i, M, mux_1, r_{i+N}). We select the following sets of paths :

- P_{A1} is the set of paths (n_i, L, mux_1, r_{i+N}) where L consists only of a-type subpaths.
- P_{A2} is the set of paths (n_i, L_1, s', L_2, mux_1, r_{i+N}), where L_1 and L_2 consist only of a-type subpaths, and s'∈ {b, d}, $|L_1|+|L_2|=i-2$, $0 \le |L_1| \le i-2$.

Case B. Let P_B be the set of paths that have the form: $n_i \rightarrow S(i-N+1,i) \rightarrow S(i-N+2,i) \rightarrow \ldots \rightarrow S(N,i) \rightarrow \mathit{O_i} \equiv r_i$, i≥N. Note that for i={N,N+1} there is no subpath along the output multiplexer (denoted by the *italic* font). These paths can be described as (n_i, s_R, M, $\mathit{mux_2}$, r_i). $\mathit{mux_2}$ denotes a subpath along the multiplexers with $T_1 = 1$. We select the following sets of paths :

- P_{B1} is the set of paths (n_i, s_R, L, mux_2, r_i).
- P_{B2} is the set of paths (n_i , a_R, L_1, s', L_2, mux_2, r_i), $|L_1|+|L_2|=2(n-1)-i$, $0\leq|L_1|\leq 2(n-1)-i$.

Case C. Let P_C be the set of paths that have the form : $n_i \to S(1,i) \to S(2,i) \to ... \to S(k,i) \to C(k+1,i) \to S(k+1,i-1) \to S(k+2,i-1) \to \dots \to S(i-2,i-1) \to O_{i+N-1} \equiv r_{i+N-1}$, $3\leq i\leq N-1$. These paths can be described as (n_i , M_1, w, u, M_2, mux_1, r_{i+N-1}). We select the following set of paths :

- P_{C1} is the set of paths (n_i, $M_{1,w'u}$, w', u, $M_{2,w'u}$, mux_1, r_{i+N-1}) where $(w',u) \in \{(o,m), (p,m), (p,n)\}$ and $M_{1,w'u}$, $M_{2,w'u}$ consist only of d type sub paths if (w',u)=(o,m), and only of a type sub paths if (w',u)=(p,m) or (w',u)= (p,n), $|M_{1,w'u}|+|M_{2,w'u}| = i-3$, $0\leq|M_{1,w'u}|\leq i-3$.

Case D. Let P_D be the set of paths that have the form: $n_i \to S(1,i) \to S(2,i) \to ... \to S(i-1,i) \to C(i,i) \equiv q_i$, $1\leq i\leq N$. These paths can be described as (n_i, $\mathit{s_R}$, M, w, q_i). The subpath s_R exists only when i=N, so the *italic* font is used. We select the following set of paths :

- P_{D1} is the set of paths (n_i, $\mathit{a_R}$, L, w, q_i) where $|L| = i-1$.

Case E. Let P_E be the set of paths that have the form: $n_i \to S(i-N,i) \to S(i-N+1,i) \to ... \to S(i-N+k,i) \to C(i-N+k+1,i) \to S(i-N+k+1,i-1) \to \dots \to S(N,i-1) \to \mathit{O_{i-1}} \equiv r_{i-1}$, $i\geq N$. These paths can be described as (n_i, s_R, M_1, w', u, M_2, $\mathit{mux_2}$, r_{i-1}), where $w' \in \{o, p, o_R\}$. We have to note that $\mathit{mux_2}$ is replaced by mux_1 when i=N and does not exist when i is either equal to N+1 or N+2. We select the following set of paths :

- P_{E1} is the set of paths (n_i, a_R, $M_{1,w'u}$, w', u, $M_{2,w'u}$, $\mathit{mux_2}$, r_{i-1}) where $(w',u) \in \{(o,m), (p,n), (p,m), (o_R,m), (o_R,n)\}$ and $M_{1,w'u}$, $M_{2,w'u}$ consist only of d type sub paths if (w',u)=(o,m), and only of a type sub paths for the other combinations of (w',u), $|M_{1,w'u}|+|M_{2,w'u}|=2(n-k)-i$, $0\leq |M_{1,w'u}|\leq 2(n-k)-i$, k=2 when i=N, k=1 when i>N.

Case F. Let P_F be the set of paths that have the form: $n_I \to S(1,i) \to S(2,i) \to ... \to S(k,i) \to C(k+1,i) \to C(k+1,i-1) \to S(k+1,i-2) \to S(k+2,i-2) \to \dots \to S(i-3,i-2) \to O_{i+N-2} \equiv r_{i+N-2}$, $4\leq i \leq N-1$. These paths can be described as (n_i, M_1, w, y, u, M_2, mux_1, r_{i+N-2}). We select the following set of paths:

- P_{F1} is the set of paths (n_i , L_1, p, y, m, L_2, mux_1, r_{i+N-2}), $|L_1|+|L_2|=i-4$, $0\leq|L_1|\leq i-4$.

Case G. Let P_G be the set of paths starting from n_i and ending at q_{i-2}, with $3\leq i\leq N+2$. These paths can be described as (n_i, $\mathit{s_R}$, M, w, y_1, y_2, q_{i-2}). Note that an s_R subpath exists only when $i \in \{N, N+1, N+2\}$. We select the following set of paths:

- P_{G1} is the set of paths (n_i, $\mathit{a_R}$, $M_{y1,y2}$, w_1, y_1, y_2, q_{i-2}), $(w_1,y_1,y_2) \in \{(p, \bar{o}, \bar{o}), (o, \bar{p}, \bar{p}), (o, \bar{o}, \bar{p})\}$ and $M_{y1,y2}$ consist only of d type sub paths if w_1=o, and only of a type sub paths if w_1=p.

Case H. Let P_H be the set of paths starting from n_i and ending at $q_{i\ -1}$, with $2\leq i\leq N+1$. These paths can be described as (n_i, $\mathit{s_R}$, M, w, y, q_{i-1}). Note that an s_R subpath exists only when $i \in \{N,N+1\}$, so the *italic* font is used. We select the following set of paths :

- P_{H1} is the set of paths (n_i, $\mathit{a_R}$, $M_{w,y'}$, w, y', q_{i-1}) where $(w,y') \in \{(p, \bar{o}), (o, \bar{p})\}$ and $M_{w,y'}$ consists only of a type sub paths if $(w,y')=(p, \bar{o})$ and only of d type sub paths if $(w,y')=(o, \bar{p})$.

Case I. Let P_I be the set of paths that have the form: $n_i \to S(i+N,i) \to S(i+N+1,i) \to ... \to S(i+N+k,i) \to C(i+N+k+1, i) \to C(i+N+k+1,i-1) \to S(i+N+k+1,i-2) \to S(i+N+k+2,i-2) \to \dots \to S(N,i-2) \to \mathit{O_{i-2}} \equiv r_{i-2}$, $i\geq N$. These paths can be described as (n_i, $\mathit{s_R}$, M_1, w, y, u, M_2, $\mathit{mux_2}$, r_{i-2}). We have to note that mux_2 is replaced by mux_1 for i=N, N+1 and does not exist for i=N+2, N+3 and that s_R does not exist when $|L_1|=0$. We select the following set of paths :

- P_{I1} is the set of paths (n_i, $\mathit{a_R}$, L_1, w_y', y, u_y, L_2, $\mathit{mux_2}$, r_{i-2}) where $(w_y',u_y)=(p,m)$ if $|L_1|\neq 0$ and $(w_y',u_y)= (o_R,m)$ if $|L_1|=0$, $|L_1|+|L_2|=2n-k-i$, $0\leq|L_1|\leq 2n-k-i$, k=5 if i=N, k=4 if i=N+1 and k=2 if i>N+1.

Theorem 1. The paths selected in Cases A, B, C, D, E, F, G and I above constitute a basis for every path that starts from n_i, ends at r_j which does not contain v, z sub paths.

Theorem 2. The paths selected in Cases A-I above constitute a basis for every path that starts from n_i, ends at q_j which does not contain v, z sub paths.

Case J. Let P_J be the set of paths that have the form $d_j \to S(i-j+1,i) \to S(i-j+2,i) \to \dots \to r_i$, $1<i<N$. These paths can be described as (d_j, t, M, mux_1, r_{i+N}). We select the following set of paths:

- P_{J1} is the set of paths (d_j, t', M, mux_1, r_{i+N}) where t' ∈ (e, h, i, l), M consists only of d type sub paths if t'=e, M=L if t'=l, if t'=i and i-j is even M have the form (a, b, a, b..a) and if t'=h and i-j is odd M have the form (b, a, b..a).

Case K. Let P_K be the set of paths that have the form $d_j \to S(i-j+1, i) \to S(i-j+2, i) \to \dots \to r_i$, $i\geq N$. These paths can be described as (d_j, t', M, $\mathit{mux_2}$, r_i), where $t' \in \{e, f, g, h, i, j, k, l\} \cup \{e_R, f_R\}$. We select the following set of paths:

- P_{K1} is the set of paths (d_j ,t'', M_1, $\mathit{mux_2}$, r_i), where $t'' \in \{e, h, i, l\} \cup \{e_R, f_R\}$, M consists only of d type sub paths if t'=e, M=L if t'=l, if t'=i and i-j is even M have the form (a, b, a, b..a) and if t'=h and i-j is odd M have the form (b, a, b..a).

Case L. Let P_L be the set of paths that have the form $d_j \to C(i,i+j-1) \to C(i,i+j-2) \to \dots \to C(i,i-1) \to C(i, i) \to q_i$, $1\leq j\leq N$. These paths can be described as (d_j, x', Y, q_i), where $x' \in \{q, r\} \cup \{q_R, r_R\}$. We select the following set of paths:

- P_{L1} is the set of paths (d_j, x'', Y_1, q_i), where $x'' \in \{q\} \cup \{q_R\}$, and Y_1 consists only of $\bar{p}$ type sub-paths.

Theorem 3. The paths selected in Cases A-E and J, K form a basis for any path that starts from d_j and ends at r_i without v, z sub paths.

Case M. Let P_M be the set of paths that have the form $n_j \to S(1,j) \to S(2,j) \to \dots \to S(j,j-1) \to C(j,j) \to S(j+1,i) \to S(j+2,i) \to \dots \to S(i, i) \to O_{i+N} \equiv r_{i+N}$, $1\leq j\leq N-2$, i<N, i>j. These paths can be described as (n_j, M_1, w, v, M_2, mux_1, r_{i+N}). We select the following set of paths:

- P_{M1} is the set of paths (n_j, M_1, w, v', M_2, mux_1, r_{i+N}), where $v' \in \{\bar{e}, \bar{j}\}$, M_1, M_2 consist only of d type sub paths if $v'=\bar{e}$, M_1 have the form (ada…dcac) if $v'=\bar{j}$ and M_2 have the form (abab…ab) if $v'=\bar{j}$.

Case N. Let P_N be the set of paths that have the form $n_j \rightarrow S(1,j) \rightarrow S(2,j) \rightarrow \ldots \rightarrow S(j,j-1) \rightarrow C(j,j) \rightarrow S(j+1,i) \rightarrow S(j+2,i) \rightarrow \ldots \rightarrow S(N,i) \rightarrow r_I$, $2 \leq j \leq N-1$, $i \geq N$, $i>j$. These paths can be described as (n_j, M_1, w, v, M_2, mux_2, r_i). We select the following set of paths:

- P_{N1} is the set of paths (n_j, M_1', w', v', M_2', mux_2, r_i), where $v' \in \{\bar{e}, \bar{j}\}$ and M_1, M_2 have the same form as above (case M).

Theorem 4. The paths selected in cases A-N form a basis for the modified NRCAD.

The proofs of Theorems 1, 2, 3 & 4 can be found in [19].

5. SPP-HFRT property of the derived basis

The paths selected in the previous section, apart from forming a basis for the NRCAD, are all SPP-HFRT. For testing a path of NRCAD under the SPP-HFRT assumption we apply both transitions T (0->1 and 1->0) at the start of the path. All the rest inputs are set to stable values, such that all the on-path gates receive a stable non-controlling value at their off-path inputs. Therefore the transition is robustly propagated to a primary output for observation. The required test vectors for SPP-HFRT propagation of a T along the selected paths are given in Table 3 of [19].

6. Conclusions

Path delay fault testing of a NRCAD is a difficult task due to the excessively large number of its physical paths (see last column of Table 4). In [6] a method has been presented for deriving a minimal set of paths whose delay should be measured in order for the delays along all other paths to be computable. The number of paths of a minimal basis is listed in Table 4. In order to measure the propagation delay along the paths of the optimal basis they should be SPP-HFRT. Unfortunately for most circuits, among them the NRCAD as well, the paths of an optimal basis are not all SPP-HFRT. To this end, in this work we have proposed minor modifications to the original NRCAD design. The proposed modifications impose small hardware and negligible delay overheads. For the modified design, we have derived a basis whose paths are SPP-HFRT. The cardinality of the SPP-HFRT basis although 30% larger than that of the optimal is still many orders of magnitude smaller than the number of all physical paths of the original design (Table 4).

Table 4

NRCAD size	Paths of a minimal basis [6]	Paths of the SPP-HFRT basis	Physical paths
8	528	735	$3{,}7 \times 10^{21}$
16	2464	3231	$5{,}3 \times 10^{80}$
32	10560	13599	$6{,}4 \times 10^{324}$

7. References

[1] J. Savir, "Scan Latch Design for Delay Test", Proc. of ITC-97, pp. 446 - 453.

[2] Z. Brazilai and B. Rosen, "Comparison of AC Self - Testing Procedures", Proc. of ITC-83, pp. 89 - 94.

[3] J. L. Carter, et.al., "Efficient Test Coverage Determination for Delay Faults", Proc. ITC-87, pp. 418 – 427.

[4] G. L. Smith, "Model for Delay Faults Based upon Paths", Proc. of ITC - 85, pp. 342 - 349.

[5] T. W. Williams, et. al., "The interdependence between delay optimization of synthesized networks and testing", Proc. of the 28th DAC, pp. 87-92. ACM/IEEE, 1991.

[6] J. D. Lesser and J. J. Shedletsky, "An Experimental Delay Test Generator for LSI Logic", IEEE Trans. on Computers, C-29 (3), March 1980, pp. 235 – 248.

[7] W.-N. Li, et. al, "On path selection on combinational logic circuits", IEEE Trans. on CAD, Jan. 1989, pp. 56-63.

[8] S. Tani, et.al, "Efficient Path Selection for Delay Testing Based on Partial Path Evaluation", Proc. of 16th IEEE VLSI Test Symp., pp. 188 - 193, 1998.

[9] K.-T. Cheng and H.-C. Chen. "Delay testing for non robust untestable circuits", Proc. of ITC-93, pp. 954-961.

[10] W. K. Lam, et. al., "Delay fault coverage and performance tradeoffs", Proc. of the 30th DAC, pp. 446-452. ACM/IEEE, 1993.

[11] M. A. Gharaybeh, et. al., "Classification and test generation for path - delay faults using single stuck - fault tests", Proc. of ITC-95, pp. 139-148.

[12] U. Sparmann, et. al., "Fast identification of robust dependent path delay faults", Proc. of the 32th DAC, pp. 119-125. ACM-IEEE, 1995.

[13] Kai Hwang, *Computer Arithmetic : Principles, Architecture and Design*, John Wiley Publ., 1979.

[14] H. H. Guild, "Some Cellular Logic Arrays for Nonrestoring Binary Division", The Radio and Elec. Engr., vol. 39, June 1970, pp. 345 – 348.

[15] D. Nikolos et al., "Path Delay Fault Testing of ICs with Embedded Intellectual Property Blocks", Proc. of DATE '99, March 1999, pp. 112-116.

[16] A. K. Pramanick and S. M. Reddy, "On the Design of Path Delay Fault Testable Combinational Circuits", Proc. of Fault Tolerant Computing, 1990, pp. 374-381.

[17] W. Ke and P. R. Menon, "Synthesis of Delay – Verifiable Combinational Circuits", IEEE Trans. on Computers, Feb. 1995, pp. 213-222.

[18] A. Krstic et. al, "Identification and Test Generation for Primitive Faults", Proc. of ITC-96, pp. 423-432

[19] G. Sidiropoulos, H. T. Vergos & D. Nikolos, "Easily Path Delay Fault Testable Non-Restoring Cellular Array Dividers", Computer Technology Institute, Technical Report No. 99/07/05, Greece, 1999.