

Path Delay Fault Testable Modified Booth Multipliers

E. Kalligeros¹, H. T. Vergos^{1,2}, D. Nikolos^{1,2}, Y. Tsiatouhas³ and Th. Haniotakis³

¹Computer Engineering & Informatics Dept., University of Patras, 26 500 Greece

²Computer Technology Institute, 3, Kolokotroni Str., 262 21 Patras, Greece

³ISD S.A, 22, K. Varnali Str, 152 33 Maroussi, Athens, Greece

e-mail : kalliger@ceid.upatras.gr, {vergos, nikolosd}@cti.gr, {tsiatouhas, haniotak}@isd.gr

Abstract

Testing of Modified Booth Multipliers (MBMs) with respect to path delay faults, is studied in this paper. Design modifications are proposed and a path selection method is suggested. The selected paths are Single Path Propagating – Hazard Free Robustly Testable (SPP- HFRT) and based on their delays the delay along any other path of the MBM can be calculated. The number of the selected paths is impressively small compared to all paths of the multiplier. The delay and hardware overhead imposed by the modifications are respectively negligible and small.

1. Introduction

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. The path delay fault model [1] 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.
- b) Since 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 is not robustly testable.

To reduce the number of paths that must be tested for path delay faults, various path selection methods have been proposed [2-5]. None of them though has proven satisfactory in the general case.

It has been shown in [2] that by measuring the delays along a suitable very small set Δ of physical paths, the propagation delay along any other path can be calculated (we will call such a set a *basis* for the circuit). However, to be able to measure the propagation delay along the paths of the basis they must be SPP-HFRT [2]. Unfortunately for most

circuits, a suitable SPP-HFRT basis does not exist.

Multipliers are met in almost all contemporary general and special purpose processors. Array multipliers implementing the modified Booth algorithm with 2-bit recoding feature regularity, short execution time and small area compared to other implementations of multipliers for signed multiplication [6]. Testing them for path delay faults is a very difficult task due to their excessively large number of physical paths (see 3rd column in Table 5). A basis, according to [2], consisting of SPP – HFRT paths does not exist for the MBM.

In this paper modifications of the MBM are proposed leading to a basis Δ' consisting of SPP-HFRT paths. The delay overhead due to the modifications is negligible while the hardware overhead for practical size MBMs is small. We give a method to derive the basis Δ' and we show that it consists of an impressively small percentage of all physical paths. However, due to the prohibitively large number of paths of an MBM, for example the number of paths of the 32 x 32 MBM is equal to 2.4×10^{15} , it is impossible to calculate the delay along all paths in order to derive the maximum path delay. This problem can be overcome using the method proposed in [3] to determine a relatively small number of paths Π 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 [7] that the fact that the circuit functions correctly at a speed does not imply that it will also function correctly at a lower speed. If we can test all primitive path delay faults of a circuit at a speed and during test application no delay fault is detected then the circuit functions correctly at any lower speed [7]. 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 speed), and the circuit functions correctly for all lower speeds.

2. SPP-HFRT paths and delay-verifiable circuits

A two pattern test $T = \langle V_1, V_2 \rangle$ is said to be a robust delay test for a path P, for a rising or falling

transition at the input of the path, if and only if, when P is faulty and test T is applied, the circuit output is different from the expected state at sampling time, independent of the delays along gate inputs not on P [8]. A robust test that propagates the fault effect through only a single path to an output in the circuit will be called a Single-Path Propagating Robust Test (SPP-RT) for that output. A robust test is said to be a Hazard-Free Robust Test (HFRT) if no hazards can occur on the tested path during the application of the test, regardless of the gate delay values. Robust tests may not exist for all path delay faults in an arbitrary circuit.

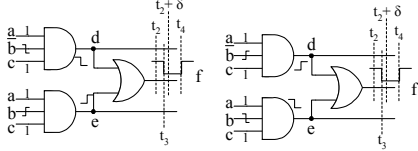


Fig. 1.a Fig. 1.b

It has been shown in [7] that the fact that a circuit functions correctly at a speed does not imply that it will also function correctly at a lower 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 [7]. A circuit which has a strong delay-verification test set is called a delay-verifiable circuit [7]. Figure 1 [7] shows a circuit which does not have a strong delay-verification test set. All faults except $\uparrow bdf$, $\downarrow bdf$, $\uparrow bef$ and $\downarrow bef$ 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 $\langle 101, 111 \rangle$ and $\langle 111, 101 \rangle$ 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 its period is set to the test period τ , but the delayed pulse due to the path delay fault may cause incorrect operation at a lower speed.

Although a strong-delay verification test set does not exist for the circuit of Figure 1 under the definition given in [7], we will show below 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 can be measured applying the test vector pairs $\langle 001, 101 \rangle$, $\langle 001, 101 \rangle$, $\langle 011, 111 \rangle$, $\langle 011, 111 \rangle$, $\langle 101, 111 \rangle$ and $\langle 101, 111 \rangle$ respectively. Then the propagation delay, pd , along the paths bdf and bef can be calculated by :

$$\begin{aligned} pd(\uparrow bdf) &= pd(\uparrow \bar{b}d) + pd(\uparrow adf) - pd(\uparrow ad), \\ pd(\downarrow bdf) &= pd(\downarrow \bar{b}d) + pd(\downarrow adf) - pd(\downarrow ad), \\ pd(\uparrow bef) &= pd(\uparrow be) + pd(\uparrow aef) - pd(\uparrow ae) \quad \text{and} \end{aligned}$$

$$pd(\downarrow bef) = pd(\downarrow be) + pd(\downarrow aef) - pd(\downarrow ae).$$

For the output waveforms of figures 1.a and 1.b we get $pd(\uparrow bdf) = t_4$, $pd(\downarrow bdf) = t_3$, $pd(\uparrow bef) = t_4$, $pd(\downarrow bef) = t_3$, therefore the maximum delay of the circuit is equal to t_4 and for 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 [7] and the calculated maximum propagation delay implies that the circuit will function correctly for any lower speed, therefore is delay verifiable.

3. MBM Design Modifications

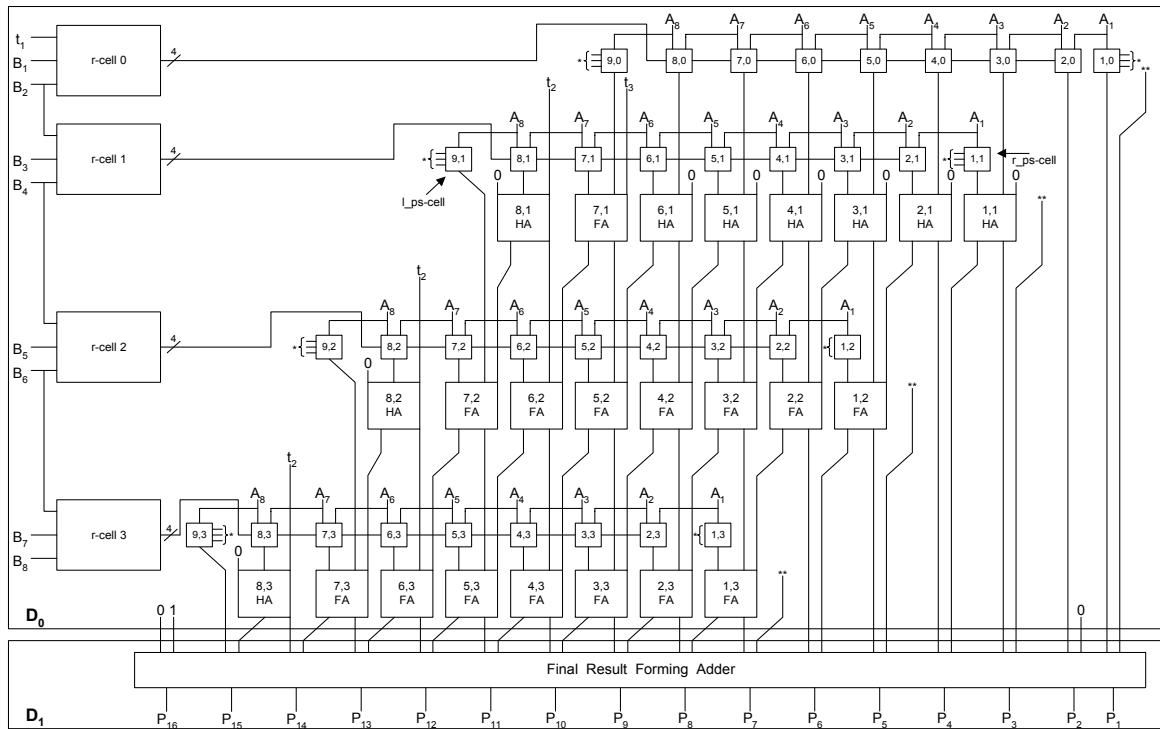
In this paper we consider $n \times n$ MBMs where $n = 2^k$, with sign generate. An $n \times n$ MBM is a combinational circuit with inputs (A_1, A_2, \dots, A_n) and (B_1, B_2, \dots, B_n) and outputs $(P_1, P_2, \dots, P_{2n})$. Figure 2 presents the 8×8 MBM.

We consider that the multiplier consists of two blocks. The first block D_0 contains the logic that forms and adds the partial products and consists of two parts :

- a) The first part is responsible for the 2-bit recoding function, and is implemented by the $n/2$ cells at the left end, named r -cells (Figure 3 presents the assumed r -cell implementation).
- b) The second part is responsible for the generation and addition of the partial sums and is implemented by:
 - b1) $(n-1) \cdot (n/2)$ cells, named ps -cells (Figure 4 presents the implementation of a ps -cell).
 - b2) $n/2$ cells, named l_{ps} -cells. An l_{ps} -cell is the leftmost cell in a ps -cell row. It can be a normal ps -cell with an inverter at the output and its I_4 and I_5 inputs driven from the same signal (A_n) . Since such modifications result in a non robustly testable l_{ps} -cell, in Figure 5 we present the implementation of a robustly testable l_{ps} -cell.
 - b3) $n/2$ cells, named r_{ps} -cells. An r_{ps} -cell is the rightmost cell in a ps -cell row. It can be designed as a normal ps -cell with its I_5 input connected to ground. This results in a non-robustly testable cell. A robustly testable implementation is shown in Figure 6.
 - b4) $(n-1) \cdot [(n/2)-2] + 1$ full adders. We consider every full adder implemented as in [9].
 - b5) $n + (n/2) - 3$ half adders.

The second block D_1 is an $(2 \cdot n)$ -bit adder which forms the final result. D_1 can be implemented as a ripple carry or group carry look ahead adder.

Using multiplexers for making the inputs and outputs of the embedded blocks controllable and observable respectively by the primary ports of the circuit, the path delay fault testing of the circuit is reduced to the path delay fault testing of the blocks



* : Receive the same inputs as the r-cell of the corresponding row,
 ** : Connected to the c2 output of the r-cell of the corresponding row

Fig. 2. An 8 x 8 MBM

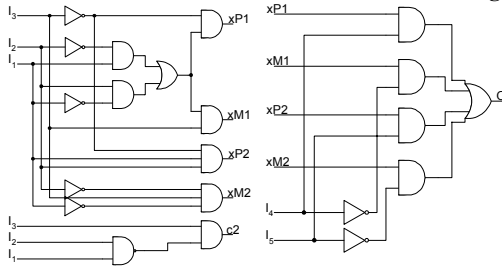


Fig. 3.

Fig. 4.

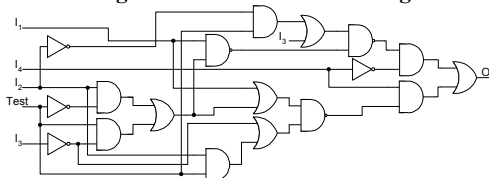


Fig. 5. 1 ps-cell implementation

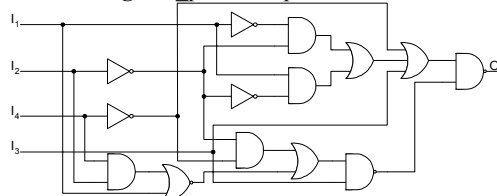


Fig. 6. r_ps-cell implementation

that constitute it [10]. In Figure 7, we show how this technique can be applied to $n \times n$ MBMs.

By adding multiplexers in the original MBM design, we can manipulate the blocks, D_0 and D_1 , individually. We will hereafter deal only with the path delay fault testing of D_0 , since efficient path delay fault testing techniques of both ripple carry and group carry look ahead implementations of D_1

have been presented in [9]. Since both LSP and MSP of D_1 receive the same test inputs, some paths that in [9] are tested in parallel, must be tested explicitly in this case. When each part of D_1 is an ILA with four or more cells, the paths that should be tested explicitly are only those including carry propagation from LSP to MSP, while for the rest parallel robust path delay fault testing can be carried as in [9].

Even after the addition of the multiplexers and the cell modifications proposed above, a suitable SPPHFRT basis does not exist for D_0 . To this end, we also equip the MBM design with 3 extra test inputs. The upper input of the upper r-cell is driven by a test input t_1 . The right input of the leftmost half adder of each row is driven by a test input t_2 . The rightmost input of the upper row is driven by a test input t_3 . During normal circuit operation t_1 is driven to 0 and t_2, t_3 are driven to 1.

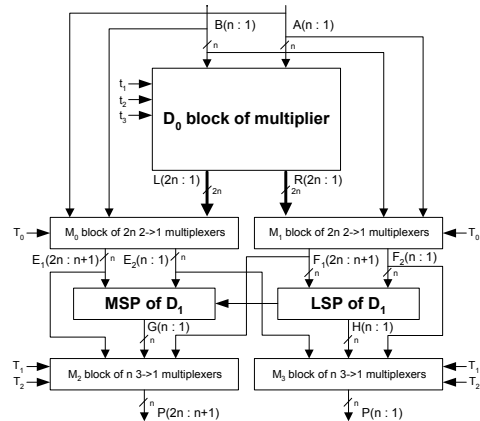


Fig. 7. Modified $n \times n$ MBM

4. Path selection procedure

In this section we derive a basis for D_0 . As sub-path we denote a section of a physical path. Each r-cell has three inputs, I_1 , I_2 and I_3 and five outputs $xP1$, $xM1$, $xP2$, $xM2$ and $c2$ as shown in Figure 3. Each ps-cell has six inputs $xP1$, $xM1$, $xP2$, $xM2$, I_4 and I_5 and one output O , as shown in Figure 4. Each ps-cell is specified as $PS(x, y)$. x specifies a ps-cell among a row of ps cells and y is the number of the r-cell that drives all ps-cells of the same row. When a ps-cell is driven by an r-cell, the outcoming circuit (denoted as rps-cell) has 5 inputs, $I_1 - I_5$ and two outputs O , $c2$. I_1, I_2, I_3, I_4 and I_5 of an rps-cell $rps(x, y)$, with x, y the numbers of the corresponding ps-cell, are connected to $B_{2y}, B_{2y+1}, B_{2y+2}, A_x$ and A_{x-1} inputs of the MBM respectively. The paths of an rps-cell along with their notation are listed in Table 1. Although an rps-cell has 23 physical paths, 21 of them form a basis for the rps-cell circuit. The propagation delay d along the other two paths is calculated by: $d(I_{1X1}) = d(A) + d(H) - d(F)$ and $d(I_{1X2}) = d(B) + d(G) - d(E)$.

From	To	Other Input Signal Values	Notation
I_1	O	$I_2 = 0, I_3 = 0, I_4 = 1, I_5 = X$	A
I_1	O	$I_2 = 1, I_3 = 0, I_4 = 1, I_5 = 0$	B
I_1	O	$I_2 = 1, I_3 = 0, I_4 = 0, I_5 = 1$	C
I_1	O	$I_2 = 0, I_3 = 1, I_4 = 1, I_5 = 0$	D
I_1	O	$I_2 = 0, I_3 = 1, I_4 = 0, I_5 = 1$	I_{1X1}
I_1	O	$I_2 = 1, I_3 = 1, I_4 = 0, I_5 = X$	I_{1X2}
I_2	O	$I_1 = 0, I_3 = 0, I_4 = 1, I_5 = X$	E
I_2	O	$I_1 = 1, I_3 = 0, I_4 = 1, I_5 = 0$	F
I_2	O	$I_1 = 0, I_3 = 1, I_4 = 0, I_5 = 1$	G
I_2	O	$I_1 = 1, I_3 = 1, I_4 = 0, I_5 = X$	H
I_2	O	$I_1 = 1, I_3 = 0, I_4 = 0, I_5 = 1$	I
I_2	O	$I_1 = 0, I_3 = 1, I_4 = 1, I_5 = 0$	J
I_3	O	$I_1 = 0, I_2 = 1, I_4 = 1, I_5 = X$	K
I_3	O	$I_1 = 0, I_2 = 1, I_4 = 0, I_5 = X$	L
I_3	O	$I_1 = 1, I_2 = 1, I_4 = X, I_5 = 1$	M
I_3	O	$I_1 = 0, I_2 = 0, I_4 = X, I_5 = 0$	N
I_1	$c2$	$I_2 = 1, I_3 = 1, I_4 = X, I_5 = X$	A_c
I_2	$c2$	$I_1 = 1, I_3 = 1, I_4 = X, I_5 = X$	H_c
I_3	$c2$	$I_1 = 0, I_2 = X, I_4 = X, I_5 = X$	K_c
I_4	O	$I_1 = 1, I_2 = 0, I_3 = 0, I_5 = X$	W
I_4	O	$I_1 = 1, I_2 = 0, I_3 = 1, I_5 = X$	X
I_5	O	$I_1 = 1, I_2 = 1, I_3 = 0, I_4 = X$	Y
I_5	O	$I_1 = 0, I_2 = 0, I_3 = 1, I_4 = X$	Z

Table 1. Sub-paths along any rps-cell of D_0

A sub-path through an rps-cell will be described either as a triplet of the form (u, x, y) , with $u \in \{A, B, C, D, E, F, G, H, I, J, K, L, M, N, W, X, Y, Z\}$ or as a pair of the form $\{u', y\}$ where $u' \in \{A_c, H_c, K_c\}$.

An l_ps -cell or an r_ps -cell has four inputs (I_1, I_2, I_3, I_4) and one output (O). These cells are not driven by an r-cell so I_1, I_2, I_3 take the same inputs as those of the r-cell of the corresponding row and $I_4 = A_n$ for an l_ps -cell or $I_4 = A_1$ for an r_ps -cell. Every l_ps -cell takes one more input named Test. This input affects the output of the cell only when I_1, I_2, I_3 have the

same value. When $Test=0$ (normal operation) the output of the l_ps -cell is 1 while when $Test=1, O=0$. A XOR gate receiving T_1 and T_2 (control signals of the multiplexer blocks) as inputs drives Test of all the l_ps -cells. Thus in normal operation $Test = 0$ and during testing can be 0 or 1. Tables 2 and 3 list the paths of any l_ps -cell or r_ps -cell respectively.

Each full adder has three inputs I_1, I_2, I_3 and two outputs, S (sum) and C (carry). Table 4 lists the possible sub-paths along any full adder. We assume that every half-adder is a full-adder, with its extra input driven to 0. We assume that the extra input is either I_1 for the leftmost adders of each row or I_3 for all the rest, as shown in Figure 2. These assumptions are only made for the clarity of the analysis and not reflected in hardware. Each full adder of D_0 will be described as $ADDER(x, y)$. A sub-path along an adder will in our notation be a triplet of the form (u, x, y) , where $u \in \{a, b, c, d, e, f, g, h, i, j, m, n, o, p, q, r\}$ is used to specify one of the possible sub-paths along the adder and x, y indicate the specific full adder we refer to.

From Table 4 we can easily see that for the same values of I_1, I_2, I_3 we may have two sub-paths, one starting from an input and ending at the S output and the other starting from the same input and ending to the C output. The sub-paths from an input of an adder to his S output will be denoted as SSO sub-paths, while the sub-paths from an input to its C output will be denoted as SCO sub-paths. Using the above notations, a path along D_0 of the MBM can be described as an ordered set of sub-paths. The first sub-path is of the form of Table 1 while the rest, if any, are either SSO or SCO sub-paths (Table 4).

From	To	Other Input Signal Values	Notation
I_1	O	$I_2 = 0, I_3 = 0, I_4 = 1, Test = 0$	A_1
I_1	O	$I_2 = 1, I_3 = 1, I_4 = 0, Test = 0$	B_1
I_2	O	$I_1 = 0, I_3 = 0, I_4 = 1, Test = 0$	E_1
I_2	O	$I_1 = 1, I_3 = 1, I_4 = 0, Test = 0$	H_1
I_2	O	$I_1 = 0, I_3 = 0, I_4 = 0, Test = 1$	I_1
I_3	O	$I_1 = 1, I_2 = 1, I_4 = 1, Test = 0$	M_1
I_3	O	$I_1 = 0, I_2 = 0, I_4 = 0, Test = 0$	N_1
I_4	O	$I_1 = 1, I_2 = 0, I_3 = 0, Test = 1$	W_1
I_4	O	$I_1 = 0, I_2 = 0, I_3 = 1, Test = 1$	X_1

Table 2. Sub-paths along any l_ps -cell of D_0

From	To	Other Input Signal Values	Notation
I_1	O	$I_2 = 0, I_3 = 0, I_4 = 1$	A_r
I_1	O	$I_2 = 1, I_3 = 0, I_4 = 1$	B_r
I_1	O	$I_2 = 0, I_3 = 1, I_4 = 1$	D_r
I_2	O	$I_1 = 0, I_3 = 0, I_4 = 1$	E_r
I_2	O	$I_1 = 1, I_3 = 0, I_4 = 1$	F_r
I_2	O	$I_1 = 1, I_3 = 1, I_4 = 0$	H_r
I_2	O	$I_1 = 0, I_3 = 1, I_4 = 1$	J_r
I_3	O	$I_1 = 0, I_2 = 1, I_4 = 1$	K_r
I_3	O	$I_1 = 0, I_2 = 0, I_4 = X$	N_r
I_4	O	$I_1 = 1, I_2 = 0, I_3 = 0$	W_r
I_4	O	$I_1 = 1, I_2 = 0, I_3 = 1$	Y_{r1}
I_4	O	$I_1 = 0, I_2 = 1, I_3 = 1$	Y_{r2}

Table 3. Sub-paths along any r_ps -cell of D_0

From	To	Other Input Signal Values	Notation
I ₁	S	I ₂ = 0, I ₃ = 0	a
I ₁	S	I ₂ = 0, I ₃ = 1	b
I ₁	S	I ₂ = 1, I ₃ = 0	c
I ₁	S	I ₂ = 1, I ₃ = 1	d
I ₂	S	I ₁ = 0, I ₃ = 0	e
I ₂	S	I ₁ = 0, I ₃ = 1	f
I ₂	S	I ₁ = 1, I ₃ = 0	g
I ₂	S	I ₁ = 1, I ₃ = 1	h
I ₃	S	I ₁ = 0, I ₂ = 0	i
I ₃	S	I ₁ = 0, I ₂ = 1	j
I ₁	C	I ₂ = 0, I ₃ = 1	m
I ₁	C	I ₂ = 1, I ₃ = 0	n
I ₂	C	I ₁ = 0, I ₃ = 1	o
I ₂	C	I ₁ = 1, I ₃ = 0	p
I ₃	C	I ₁ = 0, I ₂ = 1	q
I ₃	C	I ₁ = 1, I ₂ = 0	r

Table 4. Sub-paths along any full – adder of D₀

An SCO sub-path in a path defines a **step** and the number of consecutive SCO sub-paths defines the **magnitude of the step**. For example, there are two steps in path {(E, 8, 0), (p, 6, 1), (i, 5, 2), (o, 3, 3)}, namely (p, 6, 1) and (o, 3, 3), each one with magnitude equal to 1, while there is a single step in path {(F, 8, 1), (o, 8, 1), (q, 7, 2), (r, 6, 3)}, that of (o, 8, 1), (q, 7, 2), with magnitude equal to 2.

For simplifying the analysis, we divide all possible paths of D₀ in subsets. For an rps-cell let P = W and $\Omega = \{A, B, C, D, E, F, G, H, I, J, K, L, M, N, X, Y, Z\}$. For a l_{rps}-cell let P = W_l and $\Omega = \{A_l, B_l, E_l, H_l, I_l, M_l, N_l, X_l\}$. Finally for an r_{ps}-cell let P = W_r and $\Omega = \{A_r, B_r, D_r, E_r, F_r, H_r, J_r, K_r, N_r, Y_{r1}, Y_{r2}\}$. We define the following variables : V ∈ Ω and R, R' ∈ Ω ∪ {P}. We use the italic fonts to denote sub-paths that may be equal to ∅.

Subset 1. Paths without steps

Theorem 1. If for a specific R value, the delays along all paths of the form {(R, x, y), L} are known, with L ∈ S, where S any subset of sub-paths along the full-adders, then by measuring the delay along a path of the form {(R', x, y), Q}, R' ≠ R and Q ∈ S a specific sub-path along the full-adders, the delay along every path {(R', x, y), L} can be calculated.

Proof: $d(\{(R', x, y), L\}) = d(\{(R, x, y), L\}) + d(\{(R', x, y), Q\}) - d(\{(R, x, y), Q\})$ ■

Case A. Let P_A be the set of all paths starting with an (R, x, 0) sub-path, with 1 ≤ x ≤ n+1, or with an (R, x', y') sub-path, with x' = n, n + 1 for y' > 0, that do not include any steps. All sub-paths of P_A paths excluding the first one, can be described by a triplet of the form (z, x, y) with z ∈ {e, f, g, h}. We define a basis for P_A consisting of the following subsets :

- P_{A1} is the set of paths {(P, x, y), Q} where all sub-paths of Q have the form (e, x', y'). Obviously $|P_{A1}| = 2n - 1$, where |X| denotes the cardinality of set X.
- P_{A2} is the set of paths {(V, x, y), Q}. $|P_{A2}| =$

22n - 7.

- P_{A3} is the set of paths {(P, x, y), S}, where S are all possible sub-paths with only one sub-path of the form (w, x', y') for w = f, g, h and all other sub-paths of the form (e, x'', y''). $|P_{A3}| = 3(n^2/2) - 6n + 5$.

Definition : The **order** of a path or sub-path is equal to the number of the sub-paths of the form (w, x, y), with w ∈ {f, g, h}, that includes.

For example the order of the paths {(W, 6, 0), (e, 4, 1), (e, 2, 2)} and {(E, 7, 0), (f, 5, 1), (g, 3, 2), (e, 1, 3)} is equal respectively to 0 and 2. Observe that P_{A1} ∪ P_{A3} contains all sub-paths without steps, starting with a (P, x, y) sub-path, that have order 0 and 1.

Lemma 1: The propagation delay d(p) along a path p of order u, with u ≥ 2, starting with a (P, x, y) sub-path can be calculated from the propagation delays along paths with order less than u starting with the (P, x, y) sub-path.

Proof: The paths under investigation have the form p₁ = {(P, x, y), T, (w, x', y'), U}. Consider the paths : p₂ = {(P, x, y), T', (w, x', y'), Q} with order 1, p₃ = {(P, x, y), T, (e, x', y'), U} with order u-1 and p₄ = {(P, x, y), T', (e, x', y'), Q} with order zero. Then d(p₁) = d(p₂) + d(p₃) - d(p₄). ■

Then, from Theorem 1, using the paths of P_{A2} we can calculate the delays along any path of P_A. Note that along with the paths of P_{A2} we must measure the propagation delay along the paths {A_c, y}, {H_c, y}, {K_c, y}.

Due to space limitations, we will only present the paths that constitute the SPP-HFRT basis and the required lemmas for the rest of the cases. The proofs can be found in [11].

Case B. Let P_B be the set of all paths starting with an (R, x, y) sub-path, 1 ≤ x ≤ n-1 and y > 0, that do not include any steps. From Tables 1 and 4 we conclude that each path of P_B is of the form {(R, x, y), (w₁, x, y), S}, with w₁ ∈ {a, b, c, d} and S is composed of sub-paths of the form (w₂, x', y'), with w₂ ∈ {e, f, g, h}. A SPP-HFRT basis of P_B consists of the following sets:

- P_{B1} is the set of paths {(P, x, y), (w₁, x, y), Q}, where all sub-paths of Q have the form (e, x', y'). $|P_{B1}| = 2n^2 - 8n + 7$.
- P_{B2} is the set of paths {(V, x, y), (a, x, y), Q}. $|P_{B2}| = (n/2 - 1)(17n - 23)$.

Subset 2. Paths with a single step

Case C. Let P_C be the set of all paths such that : a) they start with an (R, x, 0) sub-path, with 3 ≤ x ≤ n+1, or with an (R, x', y') sub-path, with x' = n for y' = n/2 - 1 and x' = n, n + 1 in any other case, b) include only a single step of magnitude 1. Then, every path of P_C has the form {Q, (w₁, x', y'), (w₂, x' - I, y' + I), S} where Q and S are sub-paths without steps, w₁ ∈ {o, p}, w₂ ∈ {i, j}, 1 ≤ x' ≤ n and 1 ≤ y' ≤ n/2 - 1. A SPP-HFRT basis of P_C consists of the

following sets:

- P_{C1} is the set of paths $\{(P, x, y), T, (z, x', y'), (w_2, x'-1, y'+1), U\}$, where $z = p$ except for ADDER (n, y) for every y , and ADDER $(7, 1)$ where $z = o$. $|P_{C1}| = n^2 - 7(n/2) + 2$.
- $P_{C2,1}$ is the set of paths $\{(P, x' + 2, y' - 1), (a, x' + 2, y' - 1), (o, x', y'), (i, x' - 1, y' + 1), U\}$ where $1 \leq x' \leq n - 3$ and $2 \leq y' \leq n / 2 - 1$. $|P_{C2,1}| = (n - 3)(n/2 - 2)$.
- $P_{C2,2}$ is the set of paths $\{(P, x' + 2, y' - 1), (e, x' + 2, y' - 1), (o, x', y'), (v, x' - 1, y' + 1), U\}$ where $n - 2 \leq x' \leq n - 1$ and $2 \leq y' \leq n / 2 - 1$, $v = i$ for $x = n - 2$ and $v = j$ for $x = n - 1$. $|P_{C2,2}| = n - 4$.

T, U are step free sub-paths with order equal to zero.

Case D. Let P_D be the set of all paths starting with an (R, x, y) sub-path, $1 \leq x \leq n-1$ and $y > 0$, that include only a single step of magnitude 1. An SPP-HFRT basis of P_D consists of the following set:

- P_{D1} is the set of paths $\{(P, x, y), (w_1, x, y), (v, x-1, y+1), S\}$ where $v = i$ except for $x = n - 4$ and $w_1 = m$ where $v = j$. $|P_{D1}| = (n - 1)(n - 3)$. S is a step free sub-path with order equal to zero.

Case E. Let P_E be the set of all paths starting with an (R, x, y) sub-path, $4 \leq x \leq n+1$ for $y = 0$, $2 \leq x \leq n$ for $y = (n/2) - 2$ and $2 \leq x \leq n + 1$ for $1 \leq y \leq (n/2) - 3$. These paths include only a single step of magnitude 2. An SPP-HFRT basis of P_E consists of the following sets:

- P_{E1} is the set of paths $\{(P, x+1, y-1), (z, x+1, y-1), (w_2, x, y), (v, x-1, y+1), U\}$ where $1 \leq x \leq n-2$, $y > 1$. $z = n$ except for $x = n-2$ and $y = 2$ where $z = p$. $v = j$ for $w_2 = q$, $n-1 \leq x \leq n-2$ and $y > 2$ while $v = i$ in any other case. $|P_{E1}| = (n-4)(n-2)$.
- P_{E2} is the set of paths $\{(V', x + 1, y - 1), (o, x+1, y-1), (w_2, x, y), (v, x-1, y+1), U\}$ where $x = n - 1$ and $y > 1$. For $w_2 = q$, $V' = Y$ and $v = j$. For $w_2 = r$, $V' = E$ and $v = i$. $|P_{E2}| = n - 4$.

U is a step free sub-path with order equal to zero.

Lemma 2: The propagation delay along a path that only includes a single step of magnitude $u > 2$, can be calculated given the propagation delays along all paths that only include a single step of magnitude $u-1, 2$ and 1 .

Using Lemma 2, by induction we conclude that the propagation delays along all paths that contain a single step sub-path of any magnitude can be calculated from the propagation delays along all paths containing a single step sub-path of magnitude 1 and 2 .

Subset 3. Paths with multiple steps

Lemma 3: The propagation delay along a path p_1 with u steps, where $u \geq 2$, can be calculated from the propagation delays along paths with zero, one and $u-1$ steps.

Then, since the delays along the paths with zero and one steps of any magnitude are known, the above Lemma implies that we can recursively calculate the

delay along any path of D_0 . The proof that the paths that constitute the basis of D_0 are SPP-HFRT can be found in [11].

5. Conclusions

Path delay fault testing of an MBM is a difficult task due to the excessively large number of its physical paths (see 3rd column of Table 5). A basis consisting of SPP-HFRT paths, according to [2], does not exist for the MBM. To this end, in this work we have proposed minor modifications to the original MBM 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 proposed basis is many orders of magnitude smaller than the number of all physical paths of the original design (Table 5). We have shown that calculating the propagation delay along the paths included in primitive faults we derive the maximum propagation delay of the circuit; the circuit will operate correctly for any lower speed.

MBM size $n \times n$	Original		Modified		
	Equiv. gates	Physical paths	Equiv. gates	Hardware Overhead %	Paths to be tested
16	7695	$1.5 \cdot 10^9$	8654	12.4	7222
32	29151	$2.4 \cdot 10^{15}$	31054	6.5	18654
64	113535	$2.3 \cdot 10^{27}$	117326	3.3	65326

Table 5. Comparisons

References

- [1] G. L. Smith, "Model for Delay Faults Based upon Paths", Proc. ITC - 85, pp. 342 - 349.
- [2] 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.
- [3] 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.
- [4] K.-T. Cheng and H.-C. Chen. "Delays testing for non robust untestable circuits", Proc. of ITC-93, pp. 954-961.
- [5] U. Sparmann, et. al., "Fast identification of robust dependent path delay faults", Proc. of the 32th DAC, pp. 119-125. ACM-IEEE, 1995.
- [6] M. Annaratone, *Digital CMOS Circuit Design*, Kluwer Academic Publishers, 1986.
- [7] W. Ke and P. R. Menon, "Synthesis of Delay - Verifiable Combinational Circuits", IEEE Trans. on Computers, Feb. 1995, pp. 213-222.
- [8] C. J. Lin and M. Reddy, "On Delay Fault Testing in Logic Circuits", IEEE Trans. on CAD, Sep. 1987, pp. 694-703.
- [9] T. Haniotakis et. al, "C-Testable One-Dimensional ILAs with Respect to Path Delay Faults : Theory and Applications", IEEE Int. Symp. DFT-98, pp 155-163.
- [10] D. Nikolos et al., "Path Delay Fault Testing of ICs with Embedded Intellectual Property Blocks", Proc. of DATE '99, March 1999, pp. 112 - 116.
- [11] E. Kalligeros et. al., " Path Delay Fault Testable Modified Booth Multipliers", Computer Technology Institute, Patras, Greece, Technical Report, 1999.