Analytic Approach for Error Masking Elimination in on-line Multipliers.

H. Bederr * , M. Nicolaidis and A. Guyot
Reliable Integrated Systems Group TIMA/INPG
46, avenue Félix Viallet
38031 Grenoble Cedex, France

Abstract

Several systematic design approaches are known to be representatives of the techniques well adapted for testing sequential circuits (partial and full scan, LSSD ...). However in some cases, like for the test of on-line operators, ad-hoc DFT (design for testability) schemes become more suitable. Indeed, on-line arithmetic are used for high precision numbers resulting on high length operators. Thus the length of a test sequence for a scan design approach can grow quite large due to the shift in (shift out) of test values (test responses) and therefore the test application time would become prohibitive. Moreover, the arithmetic nature of these operators imply that some errors detected locally are masked before their observation at the primary outputs.

In this paper we describe an analytic approach for testing on-line multipliers that allows to avoid error masking without adding extra hardware for internal state observability while maintaining a 100% fault coverage. Compared to a DFT approach using parity trees, this method leads to a reduction of the area overhead from 7% to 1% and of the extra pins count from 6 to 3 in the case of the on-line multipliers considered in this paper.

1. Introduction

On-line arithmetic principles were introduced by Ercegovac and Trivedi in 1977 [1,2]. In this arithmetic, operands are serially introduced starting from the most significant digit MSD. Consequently, the MSDs are first obtained in the result, and can thus be exploited while computation is still in progress. This allows dynamically pushing the computation precision to any extent as well as a high degree of parallelism when many operators are pipelined.

The interest of serial operators, whether most significant or least significant digit first, in signal processing applications has been widely published [3-8]. However serial least-significant-digit-first operations are limited to addition and multiplication, whereas on-line arithmetic allows the computation of the most common mathematical functions [9-13].

The test of on-line multipliers was first investigated in [14]. Due to the serial nature of the inputs and outputs, the observability and the controllability are very low and that linear test lengths (i.e. proportional to the operands size) are necessary when traditional test techniques of sequential circuits are to be used (partial and full scan path [15-17]). Since the size of the operands is generally large, the test application time would be unacceptable.

Moreover, some errors detected locally in the multipliers cells can be masked before been observed at the primary outputs.

Efficient ad-hoc DFT (design for testability) techniques can be used to cope with the problems of observability and controllability at a low area overhead price (close to 7% of the original circuit) and few additional pins (three inputs and three outputs) [14]. In these schemes, most of the area overhead involved is due to the addition of parity trees utilised for avoiding error masking.

We propose in this paper a mathematical model and analysis for the error masking occurring in this class of multipliers. In this model, an arithmetic value is associated to each error and the condition of error masking is described by an equation called Error Masking Equation.

Then, we give the necessary conditions to be verified by the test patterns in order to avoid error masking (repetition of test vectors) and we derive test pattern structures such that all the possible errors cannot belong to the space of solutions of the Error Masking Equation. From these structures, actual test patterns are determined.

This way, the parity trees used in the previous DFT scheme [14] can be totally suppressed and only DFT for enhancing controllability (i.e. reducing test length) is required.

This analytical approach results in an area overhead lower to 1% and in adding only three extra inputs while maintaining a 100% fault coverage.

2. On-line multiplier

An on-line operator performs a serial operation starting with the most significant digit first. The on-line multiplier (Fig. 1) multiplies an n-digit multiplicand $A = \sum_{n-1}^{i=0} a_i 2^i$ by an n-digit multiplier $B = \sum_{n-1}^{i=0} b_i 2^i$. Each a_i and b_i $(0 \le i \le n-1)$ is a Signed Binary Digit (SBD) $\in \{-1, 0, +1\}$ coded as a difference of two bits. So the three values of a_i

* Visitor at AT&T BELL Labs, Princeton, under a grant from the French foreign ministry

1063-6889/95 $4.00 © 1995 IEEE

and b_i are -1 (0,1), 0 (0,0) or (1,1) and +1 (1,0).
This representation allows carry-propagation-free addition.
The on-line multiplier is composed of three main parts : a Partial Product Generator, a Sequential Array of Parallel Adders and a Serial Adder (Fig. 1).

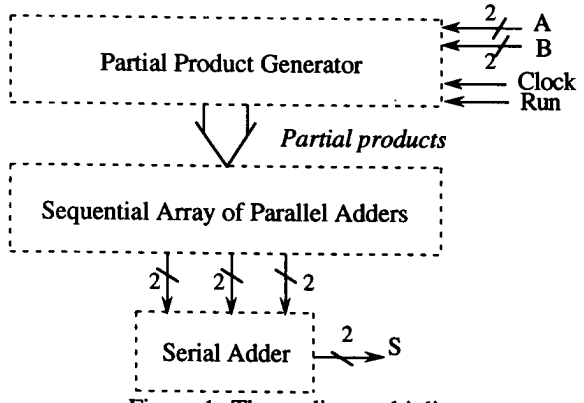

Figure 1- The on-line multiplier

The partial product generator contains: the latches, the flip-flops FF and the SBD multipliers (⊗ Fig. 2). The latches store, one after the other, the signed digits inserted serially in the primary inputs A and B; the flip-flops are arranged in a string where the control pulse "Run" is propagated to command the latches, and the SBD multipliers perform, at each clock pulse, the partial products of the signed digits (stored in the latches) by the incoming digits.

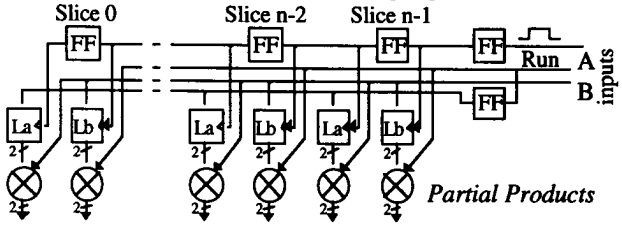

Figure 2- The partial product generator

The sequential array of Parallel Adders "PA" is represented in Fig. 3. In this figure, all the signal lines represent two bits. The basic cell of the array is a parallel adder composed of four PPM cells represented in Fig. 4a The equations of a PPM are given by Fig. 4b. The delay elements of the array (boxes with Δ) store the intermediate results.
Finally the Serial Adder "SA" of Fig. 3 has nearly the same structure as the PA since it is composed of the same basic cell but has no vertical inputs and only two outputs. It is also the single element that has observable outputs.

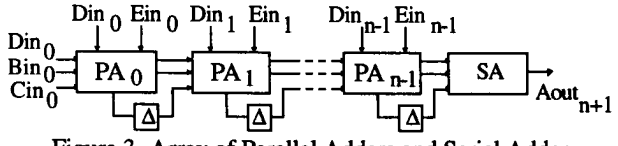

Figure 3- Array of Parallel Adders and Serial Adder

3. Fault model and testing requirements

The fault models adopted for each element are as follows:
- For flip-flops and latches we consider a fault model which includes stuck-at faults and transition faults.
- For PPM and SBD multiplier cells we consider a functional model such that under a fault the function of the cell can be modified to any other logic function. The model considers however that the combinational function of the cell cannot be transformed into a sequential one.

We also assume that the fault is permanent , and that at most one element of the circuit is faulty at the same time (The Single Faulty Cell Model is widely used for the test of regular arrays).
Covering the fault model requires to test exhaustively all the PPMs and SBD multipliers and to propagate the eventual errors to the primary outputs.

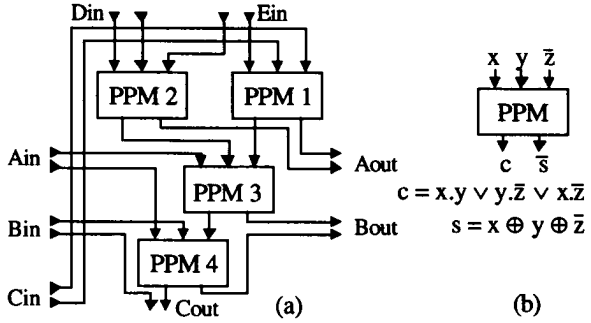

Figure 4- (a) The Parallel Adder (b) The PPM cell table

4. The DFT approach

In the following, we first describe briefly the DFT techniques proposed in [14] for the test of the multiplier.
The observability and controllability of the circuit are very low, since it has few inputs and outputs on the one hand and on the other hand, the serial structure of the partial product generator is not useful for application of periodical test patterns (i.e. deep sequentiality). A way to increase these two factors is to modify the on-line multiplier for improving its testability (Design for Testability).

Controllability : Some slight modifications have been made to the partial product generator originally designed, in order to enhance the parallelism of this structure (Fig. 5). These modifications allow to provide to the sequential Array of Parallel Adders, a set of partial products that offers a regular structure in terms of test patterns : all the partial products are equal or every two partial products are equal. These modifications are :
- two extra inputs SET and TEST, and two latches TA and TB, which are added in order to store during the whole test the neutral element of the SBD multiplication. This modification, added to the duplication of bus B, allows to provide regular patterns to the inputs Din and Ein of the Parallel Adders.
- an AND gate is also added to allow the temporary inhibition of sequential array's flip-flops.
- three multiplexers, a scan path chain with six flip-flops and an extra input are inserted to allow the controllability of the lateral inputs Ain_0, Bin_0 and Cin_0 of the parallel adder PA_0. In the unmodified array these inputs are grounded.

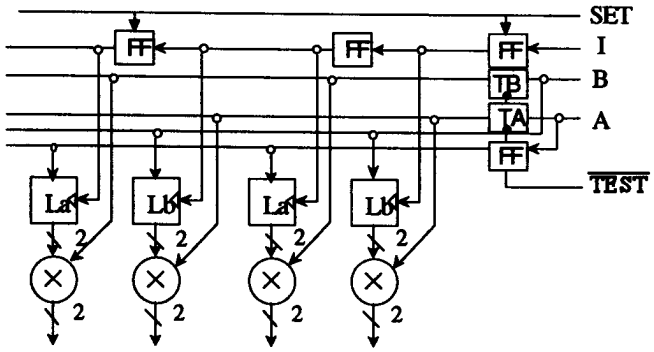

Figure 5- The DFT modifications

Observability : The errors can be recombined and masked before they reach the observable outputs since several clock cycles are needed to propagate them. An error masking occurs in the sequential array of Parallel Adders when a manifestation of a fault is deleted by another error produced by the same fault activated some clock cycles later.

An example of error masking is shown in Fig. 6. We can see that the faulty cell PPM2 of PAi, generates two faulty values at two successive clock cycles (1 → 0 in output s at cycle j and 1 → 0 in output c at cycle j+1) resulting in an error masking at PPM4 of PAi+1 in cycle j+1. We can see in Fig. 6 (cycle j+1) that the outputs of PAi+1 are fault free.

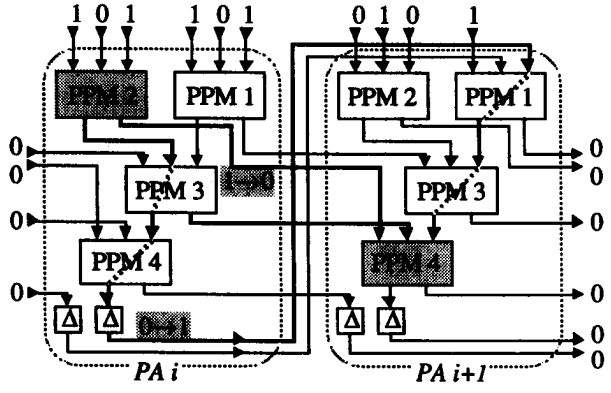

Figure 6- Example of error masking

A way to avoid this kind of masking is to insert parity trees that catch errors before masking. The considerations that determine the number of trees and the nodes to observe are exposed in [14].

The area overhead (in terms of transistors) and the extra pin count of the DFT modifications used for enhancing both observability and controllability are given in column 3 of table I.

Most of this area overhead is due to the parity trees added for enhancing observability. If we use only the circuitry added for enhancing observability, the area overhead becomes insignificant. In this paper we propose an analytic approach that allows to derive test patterns avoiding error masking so the required DFT area overhead becomes less than 1%.

		Design For Testability	
		Control + Observ.	Control
Area overhead	n = 1024	6.84 %	0.64 %
	n = 512	6.86 %	0.66 %
	n = 128	6.89 %	0.69 %
Extra input pins		3	3
Extra output pins		3	0

Table I

5. The analytic approach

In this section, an analytical approach of error masking in the sequential array is presented.

5.1. The Error Masking Equation.

Let us consider a fault in a PPM cell for a given input test vector T. The arithmetic values of the error on one output can be +1, -1 or 0 (e.g. in the case of an error 0→1 in an output with positive sign the error is +1). For each cell, the weight of the output c (carry output) is twice the weight of the output s (sum output). For a PPM cell belonging to the mth cell (PAm) these values are multiplied by 2^m. Thus, the arithmetic value of the error produced on the outputs of the faulty PPM cell when the vector T_i is applied on the cell inputs can be written $(s_i+2c_i)2^m$ with $s_i,c_i \in \{-1,0,+1\}$. The weight of the arithmetic value of a signal produced by a cell at time t_j is twice the weight of the arithmetic value of the same signal produced at time t_{j+1}. Thus if we consider a test sequence $T_0T_1T_2...T_{k-1}T_k$, the arithmetic value of the errors produced by the faulty cell can be written $2^m[(s_0+2c_0)+2^1(s_1+2c_1)+2^2(s_2+2c_2)+.....+2^k(s_k+2c_k)]$. The error will be masked when this value is 0. Thus we obtain the equation :

$(s_0+2c_0)+2^1(s_1+2c_1)+2^2(s_2+2c_2)+...+2^k(s_k+2c_k) = 0$ *(1)*

which will be called the Error Masking Equation.

If there is a fault such that : when one applies the test sequence, the error vector produced by the faulty cell belongs to the space of solutions of the error masking equation, then, the error will be masked and the fault is not detected. For instance, in Fig. 6 the error masking occurs since $s_i = 0$, $c_i = +1$, $s_{i+1} = -1$ and $c_{i+1} = 0$ verify the equation (1). Several other solutions exist for this equation (e.g. $s_i = 0$, $c_i = +1$, $s_{i+1} = +1$, $c_{i+1} = -1$; $s_i = 0$, $c_i = -1$, $s_{i+1} = +1$, $c_{i+1} = -1$, $s_{i+2} = -1$, $c_{i+2} = +1$,...etc). In order to avoid this kind of situations, we can use some constraints on the error values c_i and s_i of the equation (1). These constraints can be obtained from the fault model, the function of the cell and the test sequence.

Fault model error constraints : For a given fault model like for instance the stuck-at fault model and for a given cell implementation, it should be possible to derive error value constraints. However, the fault model considered in the paper admits that, under a fault, the cell function can be changed to any possible function. Thus the fault model does not imply any constraint on the error values.

Functional error constraints : Let us consider, for example, that the output c of a cell has a positive sign and the output

s has a negative sign. For an input vector T_i let c_i=0 and s_i=0, then, the error values of c_i can be +1 (error 0→1 on c) or 0 (no error on c), and the error value of s_i will be -1 (error 0→1 on s) or 0 (no error on s). Then the input vector T_i for which the correct outputs are 00 can be represented as T_{00} and the possible error values on c and s are {0,+1} and {-1,0}. This can be written as $T_{00}\{0,+1\}\{-1,0\}$. Similarly we can derive the possible errors for input vectors generating the correct output 01 and so as for the correct outputs 10 and 11. The complete list will be :
$T_{00}\{0,+1\}\{-1,0\}$; $T_{01}\{0,+1\}\{0,+1\}$; $T_{10}\{-1,0\}\{-1,0\}$ and $T_{11}\{-1,0\}\{0,+1\}$. Similarly we can obtain the error constraints for the cells for which c has a negative sign and s has a positive sign, and so on.

Proposition 1 : If no test vector is repeated in the test sequence, then, the functional constraints can not avoid the error masking.
Proof : Let us consider that the c output of the faulty cell has positive sign, and the s output has negative sign (the proof is similar for the other cases). Let us consider a fault such that the faulty cell produces output error only for the test vectors T_i and T_{i+1}. Then the error masking equation gives $(s_i+2c_i) + 2(s_{i+1}+2c_{i+1}) = 0$, $s_i, c_i, s_{i+1}, c_{i+1} \in \{-1,0,+1\}$. This equation has the following solutions :
$s_i = c_{i+1} = 0, c_i = -s_{i+1} \in \{-1,+1\}$ *(2)*
$s_i = 0, c_i = s_{i+1} = -c_{i+1} \in \{-1,+1\}$ *(3)*
From the functional error constraints, the couples i and i+1 which do not verify the solutions described in (2) are $T_{00}T_{01}$, $T_{00}T_{11}$, $T_{01}T_{01}$, $T_{01}T_{11}$, $T_{10}T_{00}$, $T_{10}T_{10}$, $T_{11}T_{00}$ and $T_{11}T_{10}$ and within these couples the only one which do not satisfy the solutions described in (3) are $T_{00}T_{01}$, $T_{01}T_{01}$, $T_{10}T_{10}$ and $T_{11}T_{10}$. Therefore, any two consecutive vectors must form one of the couples $T_{00}T_{01}$, $T_{01}T_{01}$, $T_{10}T_{10}$ or $T_{11}T_{10}$. This corresponds to the transition diagram of Fig. 7. A sequence which follows this diagram can not include all the input vectors of the cell. But for the considered fault model the cell must be tested for all the input vectors. Q.E.D.

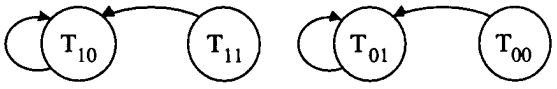

Figure 7- Test vector diagram

Test sequence error constraints : the test sequence can introduce supplementary error constraints if some test vectors are applied several times : if the test vector i is equal to the test vector i+k, then we have the constraint $s_i = s_{i+k}$ and $c_i = c_{i+k}$. Since the fault model constraints and the functional error constraints cannot avoid fault masking it is necessary to introduce test sequence constraints. Thus a necessary condition is to use test sequences which repeat some test vectors.
The space of all the possible sequences is very large and an exhaustive analysis of this space is very complex. In order to limit this complexity we will consider here test sequences with regular structures. An interesting regular sequence is the Up/Down sequence: $T_k, T_{k-1}, \ldots, T_2, T_1, T_0T_0, T_1, T_2, \ldots, T_{k-1}, T_k$ which applies one test sequence in a given order and then in the reverse order. Another sequence is the n-replicant vector sequence : $T_0,T_0,\ldots T_0,T_1,T_1,\ldots,T_1,T_2,T_2,\ldots,T_2,T_{k-1},T_{k-1},\ldots, T_{k-1},T_k,T_k,\ldots T_k$ in which each vector appears n times in consecutive positions.

5.2. The Up/Down sequence.

Let us consider the Up/Down sequence. We can check easily that this sequence does not verify neither the solutions (2) nor the solutions (3) given previously. Thus the error masking equation has no solution in the case where the errors appear in only two consecutive vectors T_i,T_{i+1}. Also we can check that there are no solutions in the case where the errors appear in three consecutive positions. Unfortunately there are solutions when the errors appear within four or more consecutive positions. Following are the c_i's and s_i's values corresponding to some of these solutions; with k+4 being the consecutive positions in which the errors occur (k > 0)

	T_{i-1}	T_i			T_{i+k+1}	T_{i+k+2}
c_i's	+p	-p	-p	-p..... -p	-p	-p+p
s_i's	0	-p	+p	+p.....+p	+p	-p0

k positions (k > 0)

5.3 The n-replicant vector sequence.

Let us consider now the n-replicant vector sequence $T_0T_0\ldots T_0T_1,T_1\ldots T_1T_2T_2\ldots T_2T_{k-1}T_{k-1}\ldots T_{k-1}T_kT_k\ldots T_k$.
Proposition 2 : There is no solution of the error masking equation of the n-replicant vector sequence (n ≥ 2).
Proof: The error masking equation of this sequence is
$(s_0+2c_0)+2(s_0+2c_0)+\ldots+2^{n-1}(s_0+2c_0)+2^n(s_1+2c_1)+$
$2^{n+1}(s_1+2c_1)+\ldots+2^{2n-1}(s_1+2c_1)+$
$+2^{kn}(s_k+2c_k)+\ldots+2^{(k+1)n-1}(s_k+2c_k) = 0$
$\Leftrightarrow(\sum_{n-1}^{i=0})2^i)(A_0+2^nA_1+2^{2n}A_2+\ldots+2^{kn}A_k) = 0$ where
$A_i = s_i+2c_i \Leftrightarrow (A_0+2^nA_1+2^{2n}A_2+\ldots+2^{kn}A_k) = 0$
We have that $\exists$ i : $s_i\neq0 / c_i\neq0$ (otherwise the circuit is not faulty), let r be the greater among these i. Then we must have $A0+2nA1+22nA2+\ldots+2rnAr=0$ *(4)*.
Since $s_r\neq0/c_r\neq0$ and $s_r,c_r \in \{-1,0,+1\}$ then
$A_r \neq 0$ and $|2^{rn}A_r| \geq 2^{rn}$ *(5)*.
On the other hand $|A_0+2^nA_1+2^{2n}A_2+\ldots+2^{(r-1)n}A_{r-1}|$
$\leq 3(1+2^n+2^{2n}+\ldots+2^{(r-1)n}) = 3[(2^n)^r-1]/(2^n-1)$
$\leq 2^{rn}-1$ since $n \geq 2$ *(6)*.
From (5) and (6), the equation (4) can never be satisfied.
Q.E.D.

Note that the only constraint considered in the proof is that the test sequence has the structure of a n-replicant sequence. Thus there are no constraints concerning the sequence $T_0T_1T_2\ldots T_{k-1}T_k$ from which the n-replicant sequence is derived. Any test vectors and any ordering can be used within this sequence and some vectors can be repeated (e.g. $T_1=T_k$ or $T_0=T_1$). A particular case of the n-replicant sequence is the twin vector sequence in which n=2. This sequence has the form $T_0T_0T_1T_1\ldots\ldots T_kT_k$.

5.4. The twin n-vector sequence.

Another regular sequence is obtained by duplicating a subsequence of n vectors instead of duplicating each vector. This sequence will be called twin n-vector sequence, and has the form $(T_0T_1...T_{n-1}T_0T_1...T_{n-1})(T_nT_{n+1}...T_{2n-1}T_nT_{n+1}...T_{2n-1})$ ....$(T_{kn}T_{kn+1}...T_{(k+1)n-1})$. For the shake of clarity the n-vector subsequences are put in parenthesis. We can note that the twin sequence is also a particular case of the twin n-vector sequence in which n=1.

The error masking equation of an twin n-vector sequence can be written as :

$(2^n+1)\{[(s_0+2c_0)+2(s_1+2c_1)+...+2^{n-1}(s_{n-1}+2c_{n-1})]+$

$2^{2n}[(s_n+2c_n)+...+2^{n-1}(s_{2n-1}+2c_{2n-1})]...$

$+2^{2kn}[(s_{kn}+2c_{kn})+...+2^{n-1}(s_{(k+1)n-1}+2c_{(k+1)n-1})]\} = 0,$

$\Leftrightarrow M_0+2^{2n}M_1+....+2^{2in}M_i+.....+2^{2kn}M_k = 0$

where $M_i = (s_{in}+2c_{in})+...+2^{n-1}(s_{(i+1)n-1}+2c_{(i+1)n-1})$,

$i \in \{0,1,....,k\}$ *(7).*

Proposition 3 : If there is an error masking with a twin n-vector sequence it can only happen within the n-vector subsequences.

Proof: We need to show that there is no solution of the equation (7) except the solution $M_0=M_1=...=M_k=0$ *(8).*

Let r be the greater i such that $M_i \neq 0$. The equation (7) becomes $M_0+2^{2n}M_1+....+2^{2in}M_i+.....+2^{2rn}M_r=0$. Since $M_r \neq 0$ we have $|2^{2rn}M_r| \geq 2^{2rn}$ *(9).*

On the other hand $|M_i| \leq 3(1+2+...+2^{n-1})=3(2^n-1)$ and thus $|M_0+2^2M_1+...+2^{2in}M_i+...+2^{2(r-1)n}M_{r-1}| \leq 3(2^n-1)(1+2^{2n}+...+2^{2n(r-1)})=3(2^n-1)(2^{2nr}-1)/(2^{2n}-1) \leq 2^{2rn}-1$ *(10).*

From (9) and (10) the equation (7) has no other solution than the solution (8).

This means that errors can be masked in the twin n-vector sequence, if and only if, they are masked within each n-vector subsequence they occur. Q.E.D.

Thus the solutions of the error masking equations of the n-vector subsequences give the solutions of the twin n-vector subsequences. It is desirable to minimise n, since the greater it is, the larger are the number of solutions. For n=1, we have the twin sequence for which there is no solution (proposition 2).

5.5. Sequences concatenation.

Proposition 3 can be generalised as follows :

Proposition 4 : If we concatenate several twin n-vector sequences (where the n can take any value), then, errors detected within the twin n-vector sequences cannot be masked when concatenating these sequences.

Proposition 5 : If any sequence is applied and then is followed by any twin n-vector sequences, then the errors detected within the twin n-vector sequences cannot be masked in the whole sequence.

Proposition 5 is a generalisation of proposition 4 and thus we only need to prove the former one.

Proof: Let consider the sequence S1S2.....Sq where S_i is a twin n_i vector sequence (with $n_i \geq 2$). Let Sr be the rightmost sequence having error masking values different from 0. By reusing the notation of (7) we can write the error masking equation of Sr as follows:

$(2^{nr}+1)[M_0+2^{2nr}M_1+....+2^{2inr}M_i+....+2^{2knr}M_k] = 0$

where $M_i = (s_{inr}+2c_{inr})+...+2^{n-1}(s_{(i+1)nr-1}+2c_{(i+1)nr-1})$,

$i \in \{0,1,....,k\}$, $n_r \geq 2$, $k \geq 2$.

The Error Masking Equation of the whole sequence S1S2.....Sr is thus: $(s_0+2c_0)+2(s_1+2c_1)+...+2^x(s_y+2c_y)$ $+2^{x+1}(2^{nr}+1)[M_0+2^{2nr}M_1+.....+2^{2knr}M_k] = 0$ *(11).*

where x and y can hold any value. Now we have *(12):*

$|2^{x+1}(2^{nr}+1)[M_0+2^{2nr}M_1+...+2^{2knr}M_k]| \geq 2^{x+1}(2^{nr}+1) \geq 5.$

$2^{x+1}|(s_0+2c_0)+2(s_1+2c_1)+...+2^x(s_y+2c_y)| \leq 3(2^{x+1}-1)$ (13).

From (12) and (13), the equation (11) has no other solution than the solution of each twin n_i vector sequence. Q.E.D.

From the sequences presented above, the n-replicant vector sequences allow complete avoidance of error masking since their error masking equations have no solutions. Unfortunately, as shown below, it is not possible to apply a sequence which has this structure for all the PA cells of the array. In the following we will represent by $IS^i(t_j)$ and $OS^i(t_j)$, the input and output states of the Parallel Adder cell PAi at the clock cycle t_j.

Proposition 6 : There is no input sequence which apply the n-replicant vector sequence to all the PPM cells of the array and which tests these cells.

Proof : Let us consider the cells PAi and PAi+1. An input sequence which applies a twin vector sequence to the PPM cells of PAi and PAi+1 applies also a twin vector sequence to PAi and PAi+1. Therefore it applies the same input state $IS^{i+1}(t_j)$ to PAi+1 during clock cycles t_0 and t_1, t_2 and t_3, t_4 and t_5 etc... Thus $IS^{i+1}(t_0) = IS^{i+1}(t_1)$, $IS^{i+1}(t_2) = IS^{i+1}(t_3)$, $IS^{i+1}(t_4) = IS^{i+1}(t_5)$ etc... Also it applies the same inputs (external and state inputs) on PAi during t_0 and t_1, t_2 and t_3, t_4 and t_5 etc... The output states of PAi will be therefore $OS^i(t_0) = OS^i(t_1)$, $OS^i(t_2) = OS^i(t_3)$, $OS^i(t_4) = OS^i(t_5)$ etc...

Since due to the delay elements Δ, $OS^i(t_j)=IS^{i+1}(t_{j+1})$ we find $IS^{i+1}(t_1) = IS^{i+1}(t_2)$, $IS^{i+1}(t_3) = IS^{i+1}(t_4)$, $IS^{i+1}(t_5) = IS^{i+1}(t_6)$ etc

Thus we have $IS^{i+1}(t_0)=IS^{i+1}(t_1) = IS^{i+1}(t_2) = IS^{i+1}(t_3) = IS^{i+1}(t_4) = IS^{i+1}(t_5)$ etc... Therefore we apply always the same input state to PAi+1 and the PPM cells cannot be tested. This proof can be generalised easily to any n-replicant vectors sequence. Q.E.D.

Similarly by considering the input and output states of the PA cells, we can show that there is no input sequence which applies the Up/Down sequence to all the PPM cells and which tests these cells. As a matter of fact we will construct the test sequence by using the twin n-vector sequences with $n \geq 2$. As we have seen before, error masking cannot be completely avoided within these sequences. Also we have seen that the error masking is reduced if n is held low. Thus we will use preferable sequences with n=2.

Our goal remains, however, to avoid error masking completely. This goal will be reached as follows. We concatenate a twin 2-vector sequence with a twin 4-vector sequence such that :

a- the error masking equation of the resulting global sequence can have, as solutions, only the solutions of the error masking equations of the above two sequences.

b- none of the solutions of the error masking equation of one sequence is also solution of the error masking equation of the other sequence.

Concretely the point (a) has the following meaning : if we concatenate a sequence with error masking equation A = 0 with a sequence with error masking equation B=0 then the error masking equation of the new sequence is $A+2^rB=0$ (r is the length of the first sequence). Thus the solutions of A=0 and B=0 are again solutions of the new equation, but we also have the additional solutions $A\neq0$, $B\neq0$ and $A=-2^rB$. The point (a) requires that there are no additional solutions but from proposition 4 point a) is verified automatically and we just need to construct the sequences in such a way that point b) is also verified.

6. Test sequence for sequential array

Let us now present the strategy used for generating the test sequence for all the PPM cells of the array. The first goal is to test exhaustively the PPM cells of all the PA cells. A way to do that is to generate a test sequence having this property for a PA cell, and to apply this sequence to all the PA cells of the array in parallel. This means that we will apply at each time the same inputs (external and state) to all the PA cells. Due to the relationships of the output state of PAi with the input state of PAi+1, such a sequence cannot test the PPM cells (see [14]). Another less constraining technique is to derive a sequence for two consecutive PA cells (e.g. PAi, PAi+1) and to apply this sequence to all the couples of PAi cells in parallel.

Since in that case the output state of a couple of PA cells is the input state of the next couple, the whole array can be represented as an automaton composed of two PA cells and a feedback loop (dotted lines corresponding to looped around signals of the PAs) as shown in Fig. 8. The test sequence will then be derived from the state transition diagram of the automaton of Fig. 8. This diagram is complex, for simplicity in Fig. 9a and 9b we present only the parts of this diagram used for deriving the test sequence.

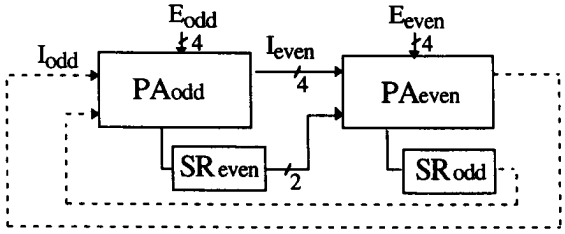

Figure 8-The automaton of two PAs

The problem is to design a twin two-vector sequence and a twin four-vector sequence such that for each PPM cell none of the solutions of the error masking equation of the first is a solution of the error masking equation of the second. Each solution of the first equation is obtained by concatenating the solutions of each vector couple. This combined with the functional constraints limits the number of solutions to 4^8 (the number of vector couples is 8). Similarly the number of solutions of the twin four-vector sequence is 104^2 (the number of vector four-tuples is 2). Supplementary constraints can be introduced by using the same PPM input values in the two sequence. These constraints must ensure the disjointness of the solutions of the two equations.

First we have constructed the twin two-vector sequence and then, the twin-four vector sequence has been derived by selecting the values that ensure this kind of constraints.

The obtained sequences $T_0T_1T_2...T_{14}T_{15}$ (derived from the diagram of Fig. 9a) and $T_{16}T_{17}T_{18}T_{19}T_{20}T_{21}T_{22}T_{23}$ (derived from the diagram of Fig. 9b) are used for deriving the twin 2-vector sequence $T_0T_1T_0T_1...T_{14}T_{15}T_{14}T_{15}$, and the twin 4-vector sequence

$T_{16}T_{17}T_{18}T_{19}.T_{16}T_{17}T_{18}T_{19}T_{20}T_{21}T_{22}T_{23}T_{20}T_{21}T_{22}T_{23}$.

Each vector T_i includes the external inputs E_{odd} and E_{even}, the state input SR_{odd} and SR_{even}, and the internal inputs I_{odd} and I_{even} applied to each PAodd and PAeven cell. The values of these signals are given in table II for the twin 2-vector sequence and in table III for the twin 4-vector sequence. We note that the values of the signals SR_{even} and SR_{odd} at time t correspond to the values generated by PAodd and PAeven at time t-1. The values T_i' used in the state transition diagrams of Fig. 9a and 9b are the values (E_{odd},E_{even}) corresponding to the vector Ti. In Fig. 9a we note that, starting from the initial state (SR_{even},SR_{odd}) = (00,00) and applying the vectors T_0' and T_1', we come back to the state (00,00). Thus, we can apply again T_0' and T_1' and we come back again to the state (00,00). Similarly we can apply twice the other couples of vectors T_2', T_3'etc... to obtain the whole twin 2-vectors sequence. At the end of this sequence, we are again in the state (00,00) and, then, from Fig. 9b we can check that we can apply the twin 4-vector sequence in a similar way.

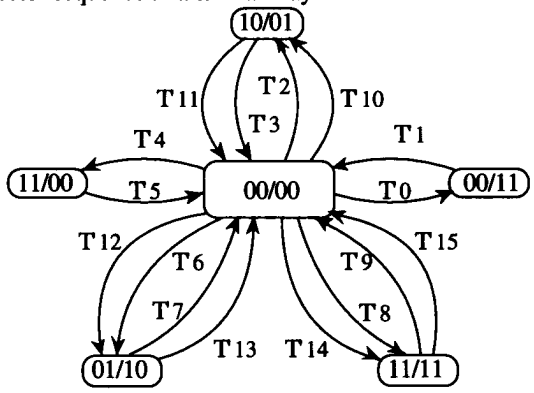

Figure 9a- Transition diagram for the twin 2-vector

As we have shown previously (propositions 3 and 4) masking can occur only inside each couple T_0T_1, T_2T_3 etc and only inside each 4-tuple $T_{16}T_{17}T_{18}T_{19}$,T_{20} $T_{21}T_{22}T_{23}$ of the twin 4-vector sequence. To avoid this masking, we must ensure that there is no any other solution than the all 0's one, which satisfies all the error masking equations corresponding to these couples and

4-tuples. The solutions of the above equations for all the 8 PPM cells of the automaton of Fig. 8, have been computed automatically. It results that none of the equations of the above couples satisfies the equations of the 4-tuples. (These results cannot be detailed here because of space reasons.) Thus, point b given in the previous section is satisfied and error masking is avoided.

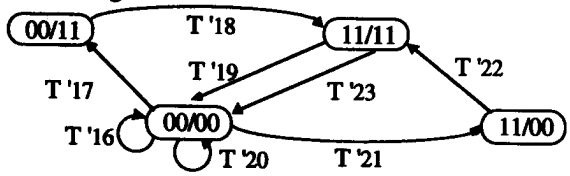

Figure 9b- Transition diagram for the twin 4-vector

Finally we note that we have considered above that we were having a complete controllability of the inputs E_{odd} and E_{even} of the automaton of Fig. 8. In practice the array is accessed through the four primary inputs of the multiplier (see Fig. 1). However the DFT techniques of Fig. 5 enhance the controllability and allow to apply the test sequences. As we have seen, these modifications require less than 0,7% hardware overhead.

Couple of test vector	PA odd			PA even		
	I_{odd}	SR_{odd}	E_{odd}	I_{even}	SR_{even}	E_{even}
T_0	11 00	00	11 00	00 11	00	11 11
T_1	10 10	00	10 10	10 10	11	10 10
T_2	00 00	00	01 00	00 01	00	11 00
T_3	10 10	10	11 10	11 10	01	10 10
T_4	00 11	00	10 01	11 00	00	01 10
T_5	01 10	11	10 10	10 00	00	00 01
T_6	10 01	00	01 11	01 10	00	00 10
T_7	01 10	01	00 10	10 00	10	00 00
T_8	01 11	00	11 10	10 01	00	01 01
T_9	10 00	11	00 01	01 10	11	10 10
T_{10}	11 00	00	11 10	00 01	00	10 01
T_{11}	10 00	10	10 11	11 10	01	11 11
T_{12}	00 01	00	01 01	01 00	00	00 00
T_{13}	11 10	01	10 11	10 10	10	10 11
T_{14}	01 11	00	00 01	01 11	00	00 01
T_{15}	00 00	11	00 00	00 00	11	00 00

Table II

7. Completing the sequence

The sequence derived previously tests the array of parallel adders, and also detects any eventual error of the serial adder. However, it does not test the partial product generator, since the SBD multipliers are not tested exhaustively.

The sequence of this part must test exhaustively each SBD multiplier cell of the partial product generator, and must avoid also error masking. Each SBD cell generates a signed binary digit encoded on two outputs A and B. The weights of the signals A and B are respectively -1 and +1, thus AB=00 and AB=11 encode the value 0, AB=01 encodes the value +1 and AB=10 encodes the value -1. From this representation it results that the arithmetic value of an error on the output A has the value $a \in \{0,+1\}$, and the arithmetic value of an error on B have the value $b \in \{-1,0\}$. For a fault affecting an SBD multiplier cell, the error masking equation corresponding to a test sequence $T_0T_1T_2...T_{k-1}T_k$ is $(a_0+b_0)+2(a_1+b_1)+...+2^k(a_k+b_k)$. For the twin vector se-quence $T_0T_0T_1T_1...T_{k-1}T_{k-1}T_kT_k$ we have the equation 3.

$[(a_0+b_0)+4(a_1+b_1)+...+4^k(a_k+b_k)] = 0 \Leftrightarrow X_0 + 4X_1 + ... + 4^kX_k = 0$ with $X_i = a_i+b_i$ and $-1 \le X_i \le +1 \ \forall \ i \in \{0,1,...k\}$. Then we can check that this equation has an unique solution which corresponds to $(a_i,b_i) \in \{ (0,0), (-1,+1), (+1,-1) \}$. Therefore, either there is no error on the signals Ai,Bi (output signals A and B of the SBD multiplier of position i), that is when $(a_i,b_i)=(0,0)$, or there is an error $0 \rightarrow 1$ on both signals Ai and Bi (when $(a_i,b_i) = (-1,+1)$) or there is an error $1 \rightarrow 0$ on both signals Ai and Bi (when $(a_i,b_i)=(+1,-1)$). These two last cases will change the signals from 00 to 11 or from 11 to 00. As a matter of fact the only masked errors are the errors which transform the values from AB=00 to the values AB=11 and vice versa. However, faults generating these errors do not change the function of the multiplier since AB=00 and AB=11 are the redundant representation of the arithmetic value 0. Thus, the arithmetic result is not changed.

Consequently we will apply to each SBD multiplier a twin vector sequence $T_{24}T_{24}T_{25}T_{25}...T_{38}T_{38}T_{39}T_{39}$ derived by using the complete set $T_{24}T_{25}T_{26}...T_{38}T_{39}$ of the 16 input vectors of the SBD multiplier. This twin vector sequence will be applied in parallel to all the SBD multipliers. Thanks to the DFT modification of fig. 5 this application is trivial and therefore the table containing the set $T_{24}T_{25}T_{26}...T_{38}T_{39}$ is not shown. The application of the twin vector sequence $T_{24}T_{24}T_{25}T_{25}...T_{39}T_{39}$ to the SBD multipliers will generate at the outputs of these cells the sequence $T'_{24}T'_{24}T'_{25}T'_{25}...T'_{39}T'_{39}$ (inputs E_{odd} and E_{even} of the parallel adders).

Finally, the latches and the flip-flops of the partial product generator of Fig. 2, do not require particular test sequence since they are tested by the test sequences derived for the other blocks.

Couple of test vector	PA odd			PA even		
	I_{odd}	SR_{odd}	E_{odd}	I_{even}	SR_{even}	E_{even}
T_{16}	10 10	00	10 10	10 10	00	10 10
T_{17}	11 00	00	00 00	00 11	00	11 11
T_{18}	10 01	00	01 01	01 11	11	11 10
T_{19}	10 10	11	10 10	10 10	11	10 10
T_{20}	10 10	00	10 10	10 10	00	10 10
T_{21}	00 11	00	11 11	11 00	00	00 00
T_{22}	01 11	11	11 10	10 01	00	01 01
T_{23}	10 10	11	10 10	10 10	11	10 10

Table III

8. Sequences concatenation

The test sequences derived in the previous sections are concatenated as shown in Fig. 10. First is applied the twin vector sequence (TV) testing the partial product generator, then, are applied the twin 2-vector and twin 4-vector sequences testing the sequential array and the serial adder. Finally we apply the propagation sequence PS. This sequence is used to propagate to the primary outputs of the multiplier the errors stored in the flip-flops of the array.

This sequence must have a length n' greater or equal to n in order to propagate all these errors. Any vector can be used to construct this sequence since the errors in the internal flip-flops are propagated at least one position in each clock cycle [14]. However this sequence must not mask the errors produced by the other sequences. This is ensured by designing PS as a twin 2-vector sequence and due to proposition 4 error masking is avoided.

Let us now review the error masking for the whole test sequence and the whole multiplier. Concerning the SBD multipliers, TV is a twin vector sequence, the sequence T2-V is a twin 2-vector one, T4-V is twin 4-vector sequence and the sequence PS a twin 2-vector one. For the SBD multiplier, one can show easily proposition 4. Thus error masking is avoided for the SBD multipliers.

Concerning the PA cells of the sequential array the sequence TV is not a twin vector sequence (due to the internal signals of the parallel adder), the other sequences are twin 2-vector and twin 4-vector sequences. According to proposition 5 the errors detected in the sequences T2-V, T4-V and PS are not masked within the whole sequence.

Fig. 10 represents the whole sequence resulting from the concatenation of the TV, T2-V, T4-V and PS sequences. In this Fig. the represented test vectors are the T'i corresponding to the (E_{odd}, E_{even}) couples (external inputs of the parallel adders).

Figure 10- The test sequences

9. Conclusion

In this paper, we have presented an efficient analytic approach ensuring the total avoidance of error masking in a sequential circuit. In particular, this scheme does not require enhancing the observability by means of additional hardware and extra pins in the case of the on-line multiplier considered here. The approach exploits only the arithmetic properties and the serial nature of the on-line operator for deriving and arranging regular test sequences . This strategy can also be extended to the test of a wide class of serial arithmetic operators.

REFERENCES

[1] M.D. Ercegovac and K.S. Trivedi, "On-line algorithms for division & multiplication" *IEEE Trans. on Computer* Vol. C-26 N°7, July 1977.

[2] M.D. Ercegovac, "A general hardware-oriented method for evaluation of functions and computation in digital computers" *IEEE Trans. on Computer* . Vol. C-26 N°7, July 1977.

[3] A. Guyot, Y. Herreros and J.M. Muller, "JANUS, an on-line multiplier/divider for manipulating large numbers" proc. *9th IEEE symposium on Computer Arithmetic* , September 1989

[4] M.J. Irwin and R.M. Owens, "A case for digit serial VLSI signal processors" *Journal of VLSI signal processing,* n°1 1990.

[5] A. F. Murray and P.B. Denyer, "A CMOS Design Strategy for Bit-Serial Signal Processing," *IEEE Journal of Solid-State circuits June 1985.*

[6] N. Kanopoulos, "A Bit-Serial Architecture for Signal Processing," *IEEE Trans. on Circuits and Systems.* pp. 289-291, vol. CAS-32, March 1985.

[7] I-Chen Wu, "AFast 1-D Serial Parallel Systolic Multiplier," *IEEE Trans. on Computers.* pp. 1243 1247, October 1987.

[8] A. Guyot,"Optimal organisation of on-line arithmetic operators in signal processing applications" proc. *International Symposium on Signal, Systems and Electronics* (ISSSE), Paris, September 1992

[9] M.D. Ercegovac, "On-line arithmetic : an overview" proc. *SPIE, Real Time Signal Processing VII.* Vol. 495 August 1984.

[10] M.D. Ercegovac and T. Lang " On-line schemes for computing rotation angles for SVDs" proc. *SPIE, Advanced Algorithms and Architectures for Signal Processing II.* pp.160-169 Aug.1987.

[11] M.D. Ercegovac and T. Lang " On-line arithmetic : a design methodology and applications to signal processing," *VLSI Signal Processing* . Chapter 24, IEEE Press 1988.

[12] A. Guyot, "OCAPI : architecture of a VLSI coprocessor for the GCD and the extended GCD of large numbers" proc.*10th IEEE Symposium on Computer Arithmetic* 1991.

[13] M.D. Ercegovac and T. Lang " Redundant and on-line CORDIC : Application to matrix triangularization and svd" *IEEE Trans. Comput., vol. C-39 no. 6,* June 1990.

[14] H. Bederr, M. Nicolaidis and A. Guyot, "Design for testability of on-line multipliers" proc.*12th IEEE VLSI Test Symposium* 1994, April 1994.

[15] E.B. Eichelberger and T.W. Williams, " A logic designstructure for LSI testing" proc. *14th Design Automation Conference*, pp.462-468, June 1977.

[16] V.D Agrawal, K.T. Cheng, D.D. Johnson and T. Lin "A complete solution to the partial scan problem", proc. *International Test Conference*, pp 44-51, 1987.

[17] H.K.T. Ma, S. Devadas, A.R. Newton and A. Sangiovanni-Vincentelli " An incomplete scan design for sequential machines" proc. *International Test Conference*, pp 730-734, 1988.