A SPECIALLY DESIGNED TRANSIENT FAULTS INJECTION TECHNIQUE AT THE VHDL LEVEL AND MODELING

Trailokya Nath Sasamal[1]& Anand Mohan[2]
[1,2]Department of Electronics Engineering, Institute of Technology, BHU
Varanasi-221005, India
Email: sasamal.trailokyanath@gmail.com

ABSTRACT

This paper presents a technique to improve verification at VHDL level by a specially designed transient faults injection block. By this technique fault insertion time, can be randomized. A probabilistic model of faulty periods, the time period where at least one fault exists and a fault analysis to derive the optimum faulty period is presented. Distribution functions are derived to represent the case of false alarm, where a transient fault is flagged as permanent, and the case of a miss, where too many faults coexist thus overcoming the checker's capability to detect them. These derivations are compared with the results of a simulation program representing the model. The VHDL coding utilized the Xilinx ISE 11.1, and the simulation has been performed in ISim simulator.

Keywords:*Transient fault, Faulty period, Checker, Modeling, Simulation*

1. INTRODUCTION

The first step in a modern digital system design is to specify it in a high level language such as VHDL. Before the translation of the specification into an actual implementation, the design needs to be evaluated based on several criteria, e.g. area, testability, power consumption etc. The capability to verify a testable system (in the presence of faults) at the VHDL level before it is implemented allows design modifications to achieve the desired goal. This makes the case for a fault injection system that provides such capability. In general faults are separated into two categories: permanent and transient. Permanent faults that exist in logic circuits are normally identified during offline testing by the manufacturer of the IC, so the transient fault is of major concern after a chip is in the hands of the consumer. The ability to simulate the occurrence of a transient fault in the VHDL description of a system is extremely important to verify the performance of an on-line testable system In addition the ability to insert permanent faults on single bits or a data word must also be taken into consideration. These features enable the performance of a system under faulty conditions to be effectively verified before the system is implemented. Any internal signal can be accessed at the VHDL level for the purpose of injecting faults, thus ensuring greater controllability and observability of the system. The fault injection system proposed in this paper will be contained within the instruction VHDL of a system. It is platform independent and is able to simulate on any VHDL simulation software without extensive knowledge of simulation VHDL.

More than one transient faults can be coexist.A checker detects a fault. Before the fault disappears and before the end of the fault period, other faults occur masking the first one and causing the checker to miss their detection. Single error detectors are not designed to detect multiple errors and therefore, the existence of two or more errors can cause a miss. This paper discusses a fault analysis to derive the optimum faulty period based on the arrival rate of faults and their lifetimes.

If a fault lasts longer than the retry period, it will be considered permanent fault, if the fault disappears before the end of the retry period, it will be considered transient. Faults can overlap, thus creating relatively longer faulty periods, and possibly masking each other, thus causing the checkers to miss their detection. The retry period should be long enough to allow a transient fault to disappear, and short enough to prevent multiple faults from overlapping.

2. FAULT INSERTION

In the proposed system, transient faults are injected randomly. The ability to predetermine a rate at which faults are inserted is very important. During transient fault injection, random bits in a data word are selected by the system fault insertion. This is a key component of the proposed injection system that enables the designer to simulate faults at more realistic intervals on varying bits in a data word without having to modify the VHDL description every time a fault is inserted in the system. If there is a single input bit or a signal that is directed to the system, a transient fault will always occur on that bit at the interval chosen by the user. This allows the user to focus solely on a single bit for fault insertion when transient fault insertion is desired. If a larger data word is sent to the injection system, it will choose on which bit the fault be injected. This is especially useful in on and offline testing by honing in on a specific bit or inserting faults randomly across a data word.

The proposed fault injection system is comprised of blocks as shown in RTL schematic Figure1.The main component of transient fault injection system is the ability to insert faults at desired intervals. For this two PN sequence which generated from two, 8-bit LFSR has been used. The Two LFSR's run in parallel generating pseudo-random sequences.Based on the percentage of time that is chosen to insert a fault, a certain number of bits in the two LFSR's are compared by the BIT MATCHER logic block. If that number of bits matches, then a fault is inserted into the system. The data flow through the system that accomplishes this is shown in Figure 2.

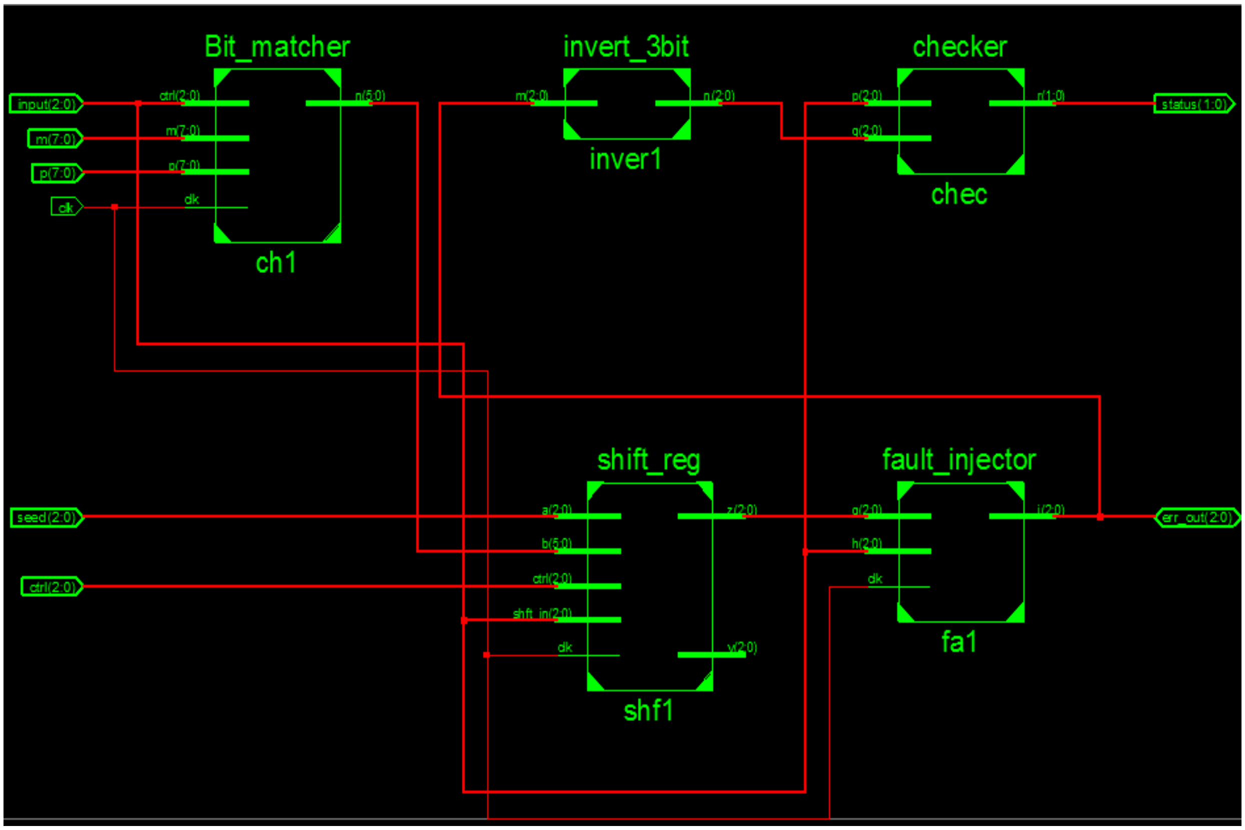


Figure 1.RTL Schematics of fault injection system

The BIT MATCHER block constantly monitors the output of both of the LFSR's, determining whether a fault is to be injected or not. The output of two LFSR are feed to the BIT MATCHER circuit with a 3 bit control input .Control inputs are for transient injection and range from "001" → 50% injection to "110" → < 1% injection. A control input of "000" is 0% fault injection while a control code of "111" is 100% fault injection. The control input is used to control the point in time at which the fault is injected. By incrementing this control input by '1' for each 3 bit pattern, fault injection is dropped by ½ from the previous rate. Figure 3represents results from a program that was written to simulate two 8-bit LFSR's running in parallel and certain numbers of bits being matched. A 3-bit control code (**Ctrl**) that is processed by the BIT MATCHER Block determines how many bits need to be matched in the two LFSR's to control the percentage at which faults are injected. The initial seed to each of the LFSR's must be different in order to produce two different pseudorandom binary sequences.

In order to determine the bit on which a fault will be inserted during transient fault injection, a SHIFT REGISTER is used. Every clock cycle, logic'1' is shifted through a data word that is the length of the data word sent to the injection system. The purpose of the '1' is to determine on which bit the fault will be injected. When the FAULT INJECTOR Block has seen that a fault is to be injected, it views the data output word of the SHIFT REGISTER according to the bit that is a '1' , FAULT INJECTOR block flip the corresponding bit of 3-bit input word. CHECKER block consist of a TWO RAIL CHECKER which shows the status fault existence.

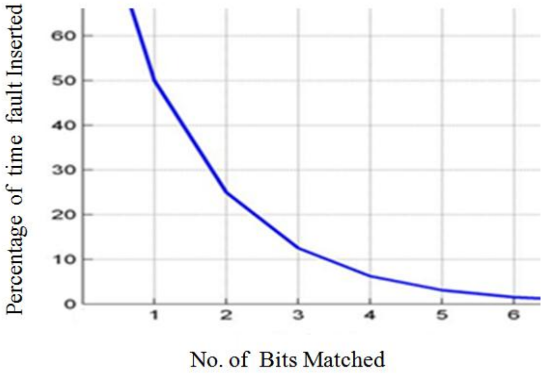


Figure 3. Percentage of time a fault is inserted in the system per the No. of bits matched

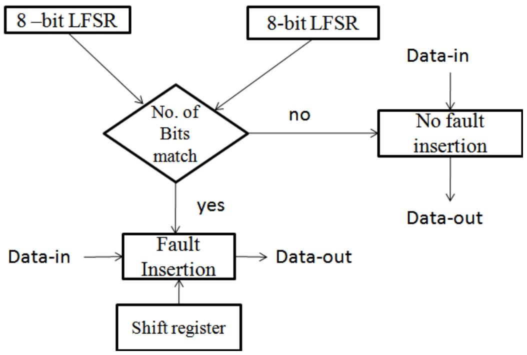


Figure 2.Flow chart of Transient faults insertion process

3. TRANSIENT FAULT MODEL

Several models for fault analysis have been used by previous researchers, mostly Markov models **,**representing the effects of permanent faults, intermittent faults, and environmental disturbances. Some researchers have used the Weibull distribution to represent these faults **.**The Weibull distribution is useful when considering the aging effect on the failure rate. However, in this paper we primarily focus our attention on transient faults caused by external disturbances independent of the aging factor.

Assumptions

1. Faults occur with a Poisson distribution.
2. The lifetime of a transient fault is exponentially distributed.
3. Faults can overlap to create faulty periods where at least one fault exists.
4. The faults are statistically independent and thus linearly uncorrelated.
5. Each transient fault creates one transient error.
6. The checkers are not prone to transient faults.
7. The checkers are failure free in the presence of zero errors. They always detect single errors; they miss the detection of double errors with probability P_f, and they always miss the detection of 3or more coexisting errors.

Nomenclature

Faulty period *time* period where at least one fault exists.
False alarmthe decision that a transient fault is permanent
A Miss the decision that the system is fault free, while one or more faults exist.

Notation

λ average arrival rate of a fault
$1/\mu$ average lifetime of a transient fault
P(k) Pr{ ***k*** faults exist in the system}
D(t) Pr{a faulty period is less than or equal to *t}*
F(t) Pr{False Alarm}
$1/\mu'$ average length of a faulty period.
G(t) Pr{ a miss}
G´(t) Pr{two or more faults overlapping in time (0, *t)}*
G´´(t) Pr{ three or more faults overlapping in time (0, *t)}*
P_f Pr{ a checker misses the detection of double errors}
$1/\lambda_2$average time between the occurrence of the first fault and a second fault that overlaps the first one.
$1/\lambda_3$ average time between the occurrence of the first fault and a third fault where all three faults are overlapping.

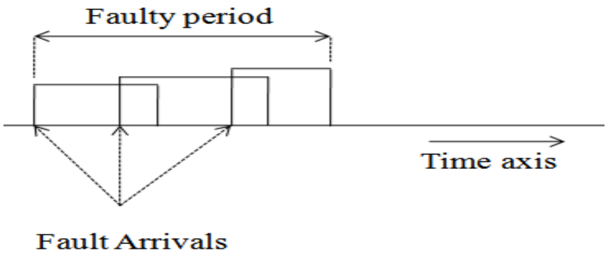


Figure 4.Time diagram representing the behavior of transient faults.

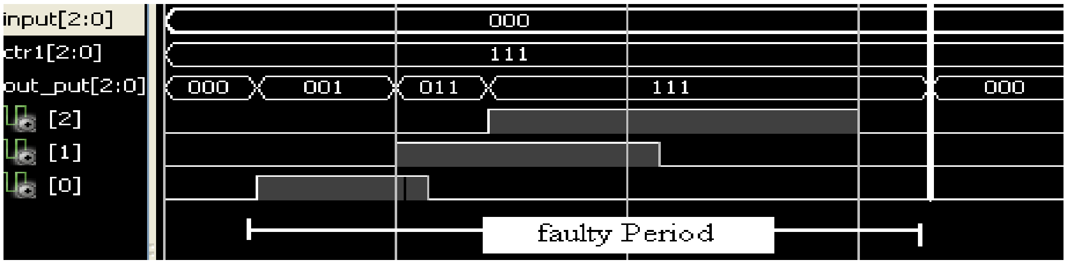


Figure 5.simulation of three overlapping faults.

Theoretically, an arbitrary number of faults can exist at any time. The analysis of M/M/∞shows that the condition to have an equilibrium solution is simply$\lambda/\mu < \infty$. Given than an environmental disturbance exists, this model provides

Probability that kfaults exist in the system:

$$\mathrm{P(k)} \;= (\frac{\lambda}{\mu})^{k} \frac{\exp(-\lambda/\mu)}{k!} \tag{1}$$

Our goal from this analysis is to find an optimum faulty period T, which will minimize the probability of false decision, both false alarm and a miss. A ***false alarm*** is the decision by the checking mechanism that a transient fault is permanent. A ***miss*** is the decision by the checking mechanism that the system is fault free, while one or more faults exist.

4. MATHEMATICAL DERIVATION

A.*False Alarms*

Based on the model under discussion the transient faults may overlap creating time periods where the system is faulty. A faulty period extends from the time of occurrence of the first fault until its decay together with the faults that overlapped. The CDF of the transient faulty periods is D (t),where the origin of the time scale (0, ***t)*** starts at the time of occurrence of the first fault

$$\mathrm{D(t)= Pr\{A\ faulty\ period} \leq \mathbf{t}\ \}\tag{2}$$

This function leads to the probability of a false alarm, F(T),for a certain retry period, T.

F(T)= 1- D(t) (3)

So to minimize false alarm, the faulty period is so chosen so that all transient faults are detected. The mean length of a faulty periods, l/μ', can be derived as follows.

The probability with which the system is fault free can be obtained from the above analysis of M/M/∞by setting k= 0 in the expression for P (k)**:**

$$\mathrm{P}(0) = \exp(-\lambda/\mu) \tag{4}$$

$$\mathrm{AlsoP}(0) = \frac{\mathrm{E\{NF\}}}{\mathrm{E\{NF\}+E\{FA\}}} \tag{5}$$

Where E{FA} is the average time of a faulty period, l/μ'.E{NF} is the average time of being not faulty or until a fault arrives. Given that there is no fault, the time for a new fault to arrive is exponentially distributed with mean E{NF} equal to 1/λ. Consequently, the probability that no fault exists is:

$$\exp(-\lambda/\mu) = \frac{\left({}^{1}/_{\lambda}\right)}{\left({}^{1}/_{\lambda}\ \right)+\left({}^{1}/_{\mu'}\right)} \tag{6}$$

$$\frac{1}{\mu'} = \frac{\exp(-\lambda/\mu)-1}{\lambda} \tag{7}$$

Using the assumption that the faulty periods are exponentially distributed, we obtain:

$$\mathrm{D(t)} = 1 - \exp(-\mu'\mathrm{t}) \tag{8}$$

Therefore, the probability of a false alarm for a certain faulty period T, is:

$$\mathrm{F(T)=exp(}-\mu'\mathrm{T)} \tag{9}$$

The CDF represented by F(t)is a monotonically decreasing function in *t.* This result implies that in order to minimize the probability of false alarm, the faulty period should be increased indefinitely. This observation is justified by the fact that with the longer faulty periods, transient faults have ample time to disappear, thus reducing the probability of misjudging them as being permanent.

B. A Miss

To analyze the CDF of a miss, G(T),fault overlaps analyzed considering the fact that multiple transient faults that could coexist. Most checkers are designed for single error detection, but they have the capability of detecting some double errors. Let the probability that a checker will miss the detection of double errors be P_f.Further assume that the checkers do not detect the existence of three or more coexisting errors. To evaluate the probability of a miss, the probability that the checker fails, in the time period (0, t), define the following distributions:

G´(t)=Pr{2 or more faults overlap in the time interval (0, t)}
G´´(t) =Pr{ **3** or more faults overlap in the time interval (0, t)}

where the time scale (0, t)starts at the occurrence of the first fault and ends at the end of the faulty period. The probability of a miss is:

$$\mathrm{G(t)} = \mathrm{P}\ _{f}[\mathrm{G'(t)} - \mathrm{G''(t)}] + \mathrm{G''(t)} \tag{10}$$

G ´(t)can be derived directly using the characteristics of Markov processes and exponential distribution. Since the faults are transient, there is a possibility that the faults do not overlap and the faulty period consists of only one fault. Let P_2be the probability that a second fault occurs while the first one is transient and still active. This probability can be obtained from a state transition diagram, where the initial state is 1and the absorbing states are 0 and 2as shown in Figure 6**.** P2is the probability with which the system starting at state 1**,**gets absorbed by state 2[8].

$$P_2 = \frac{\lambda}{\lambda+\mu} \tag{11}$$

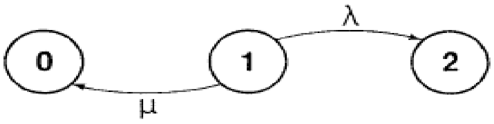


*Figure 6***.***State-transition-rate diagram for 2 or more overlapping faults*.

Assuming that state 2is reached (two faults overlap) the average time between the occurrence of these two faults, **$1/\lambda_2$,** is equal to the average time the system stays in state 1**.** This average can also be obtained from the same state transition diagram[8].

$$\frac{1}{\lambda_2} = \frac{1}{\lambda+\mu} \tag{12}$$

From these derivations we can approximate G‘(t),for 2 or more faults overlapping within time t,using an exponential function:

$$G'(t) = P_2(1 - \exp(-\lambda_2 t)) \tag{13}$$

To derive G´´(t),representing the occurrence of three or more coexisting faults within time *t*, a state transition diagram could be used where the initial state is state 1and the absorbing states are state 0 and state 3as shown in Figure 7**.** The probability that three faults overlap, P_3, is equal to the probability with which the system, starting at state one, gets absorbed by state three. This probability can be obtained by analyzing the jump chain derived from the state transition diagram in Figure 7 **[**8**].**

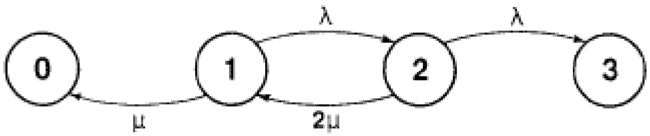


Figure 7.State-transition-rate diagram for 3 or more overlapping faults.

$$P_3 = \frac{\lambda^2}{\lambda^2 + 2\mu^2 + \lambda\mu} \tag{14}$$

Under the condition that state 3is reached, let l/$\lambda_3$be the average time between the occurrence of the first fault and the occurrence of the third overlapping fault. This average can be obtained by analyzing the jump chains derived from the state transition diagram in Figure 7[8].

$$\frac{1}{\lambda} = \frac{2\lambda + 3\mu}{\lambda^2 + 2\mu^2 + \lambda\mu} \tag{15}$$

From these derivations we can approximate G´´(t), for three or more transient faults overlapping within time t, using an exponential function:

$$G''(t) = P_3(1 - \exp(-\lambda_3 t)) \tag{16}$$

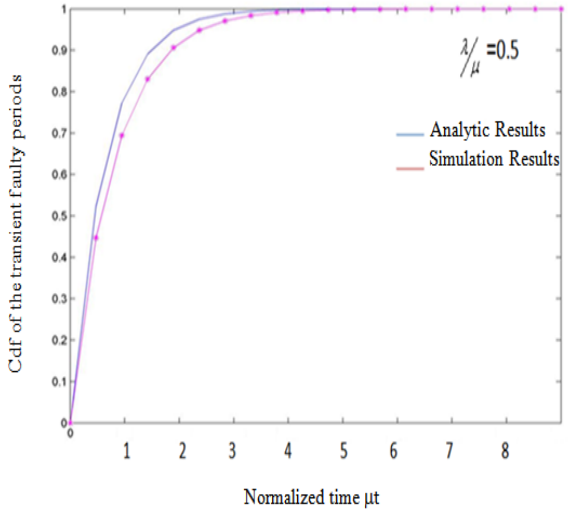
This derivation states that the probability that three faults overlap within time t, is equal to the probability of such an event occurring, times the probability that it will occur within time t,given that it is going to occur. After deriving all components of G (t)we realize that it is a monotonically increasing function in t.This result requires that in order to minimize the probability of a miss, the faulty period should be reduced indefinitely. So when the faulty period is minimal, there will be lower chances for faults to overlap.

5. SIMULATION

The analytic derivations in the previous section show that D(t), G'(t), and G"(t) in equation 2,13,16, respectively are represented by exponential functions. To validate these approximations, a simulation program was written representing the incidence of faults as described above. The sample size is 10^6 transient faults. The inter-arrival time and the lifetime of the faults are generated as exponentially distributed random variables. The average lifetime of a transient fault, 1/μ, is 1.0. The program was executed for three values of average fault interarrival rate such that λ/μ = 0.1, 0.5.

TABLE 1
Simulated and Analytic Values for the CDF of the Length of Transient Faulty Periods, D(t) for $\frac{\lambda}{\mu}$ = 0.1

Normalized time $t \cdot \mu$	Simulation values	Analytic values
0.0	0.0	0.0
1.0	0.5654	0.6136
2.0	0.8111	0.8507
3.0	0.9179	0.9423
4.0	0.9643	0.9777
5.0	0.9845	0.9914
6.0	0.9933	0.9967
7.0	0.9971	0.9987



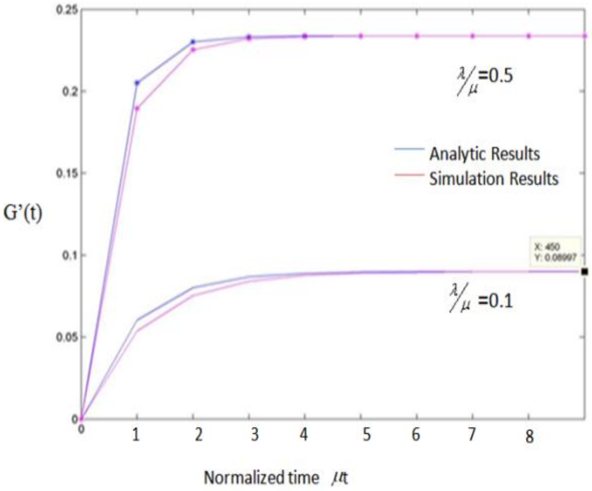


Figure 8.Simulated and analytic values for the CDF of the length of transient faulty periods.

Figure 9.Simulated and analytical values for the probability distribution function of ***2*** *or more transient faults overlapping.*

TABLE 2

Simulated and Analytical Values for the Probability Distribution Function, G'(t), of 2 or more transient faults overlapping,for$\frac{\lambda}{\mu}$ = 0.1

Normalized time $t \cdot \mu$	Simulation values	Analytic values
0.0	0.0	0.0
1.0	0.00210	0.00228
2.0	0.00327	0.00346
3.0	0.00392	0.00407
4.0	0.00428	0.00439
5.0	0.00448	0.00455
6.0	0.00459	0.00463
7.0	0.00465	0.00468
8.0	0.00468	0.00470
9.0	0.00470	0.00471

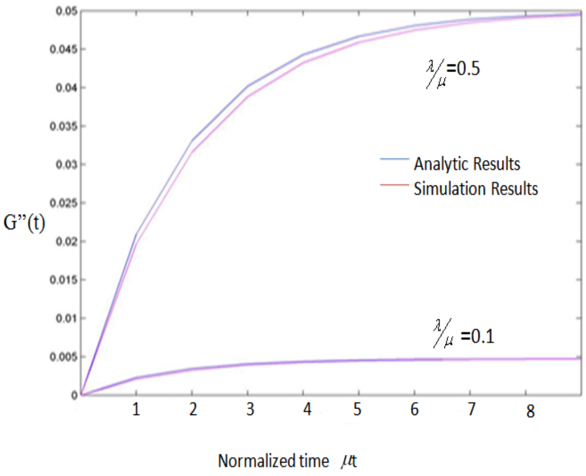


Figure 10.Simulated and analytical values for the probability distribution function of 2 or more transient faults overlapping.

TABLE 3
Simulated and Analytical Values for the Probability Distribution Function, G''(t), of 3 or more transient faults overlapping,for $\frac{\lambda}{\mu}$*= 0.1*

Normalized time $t \cdot \mu$	Simulation values	Analytic values
0.0	0.0	0.0
1.0	0.05374	0.06037
2.0	0.07539	0.08025
3.0	0.08411	0.08679
4.0	0.08763	0.08894
5.0	0.08899	0.08904
6.0	0.08944	0.08962
7.0	0.08970	0.08984
8.0	0.08989	0.08994

This simulation program monitors the length of the faulty periods, the length of the time between the occurrence of the first fault and the occurrence of the second overlapping fault, and the occurrence of the fault that causes three fault overlaps, whenever these cases occur. The data are processed to calculate the probabilities and the average times until the occurrence of two fault overlaps and three fault overlaps.

D(t)is plotted for several values of λ/μ; the time reference is normalized to 1/μ, ie, expressed as multiples of the average fault lifetime, 1/λ. Therefore, the time axis represents the value of the product (t.μ). Table 1 and Figure 8compare the simulated and approximated values of D(T) for values of λ/μ= 0.1, 0.5 respectively. Figure 11 gives the simulated values of the probability of false alarm, F(T),that a transient error is flagged as permanent using a faulty period of T.

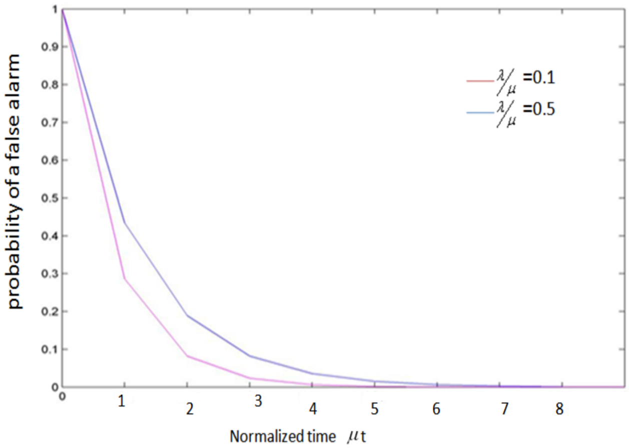


Figure 11.Simulated values for the probability distribution function of a false alert.

The CDF'S of the time periods, starting at the occurrence of the first fault until the occurrence of the second fault given that it overlaps, have been obtained by simulation. This distribution multiplied by the probability of such an event happening, P_2, is plotted and tabulated for several values of λ/ μ; it represents G'(t).Figure 9 and Table 2 compare the simulated and approximated values of G'(t).Similarly, plots are made for *G"(t)* and presented in Figure 10and Table 3.In general, the state which will cause a checker to miss can be any number of kcoexisting faults. The simulation program can be adjusted to accommodate for koverlapping faults and the analytic derivation can be generalized to (k+ 1) Markov states. The plots ofG'(t)and G"(t)show that for larger values of λ**,** the probability of a miss increases. This is mainly due to the fact that with larger λ/μ **,** faults will overlap more often since they occur at a faster rate relative to their lifetime.

6. OPTIMUM FULTY PERIOD

An infinite retry period will minimize the probability of false alarm, while an infinitesimal retry period will minimize the probability of a miss. Therefore, an optimum retry period ought to exist as a compromise which will satisfy both parameters. A CDF, Z(t),that combines the probability of false alarm for a given retry period T**,** F(T),and the probability of missing the detection of errors in this retry period, G(T),has been derived. Z(T)represents the probability of a false decisionbe it false alarm or a miss in detection. Figure 12 show the simulated Z(t)for λ/μ= 0.5, respectively, for the boundary values of P_f, the probability with which a checker fails in the presence of double errors. Z(T)can be viewed as a penalty function for a given retry period, T.Assuming that the false alarm and the miss of an error are equally damaging, i.e, considered with equal weight, Z(t)can be evaluated as follows.

$$Z(t) = 1 - (1 - F(t))(1 - G(t)) \quad (17)$$

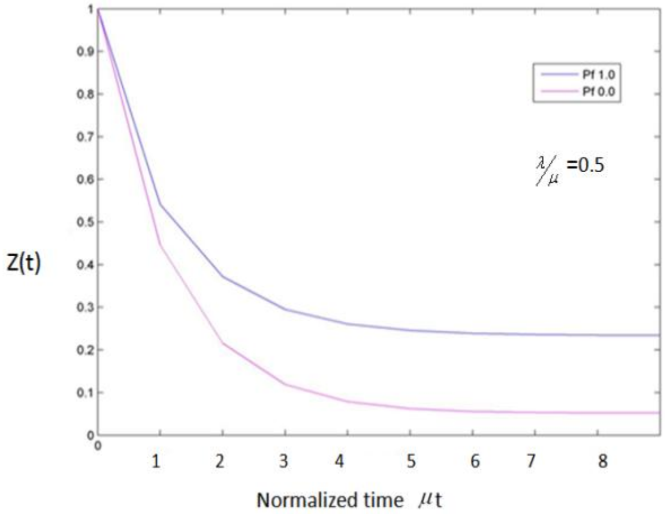


Figure 12.Simulated values for the distribution function of false decision, Z(t).

Figure 12 show that Z(T)is continuously decreasing as the retry period, T**,** is increased until a certain value where Z(T)becomes insensitive to the choice of T**.** This is not always true if F(t)and G(t)are given different weights. For example, in some applications, missing the detection of overlapping errors can be much more serious than declaring a transient fault as permanent. Thus, the penalty G(t)can be weighted more than the penalty of F(t)**.** In these cases, the function Z(T)is not necessarily monotonically decreasing when increasing the retry period, T.Consequently, an optimum retry period could be obtained which minimizes Z(T).

Figure 13 shows the simulation results of the 8-bit comparator operating at 100 MHz with fault injection. The 3-bit comparator output is taken as input to the fault injection block. Fault existing status shown by checker as shown in status(0:1).Figure 13 shows simulation results with the fault injection block turned on; the control signal is set at 001i.e. the rate of fault appearance is 50%. Seed for the shift register set as 001. At 23 ns PN sequences generated by two LFSR matched single bit, at this moment fault injector block inject a transient fault on in input data corresponding to bit'1' at the shift register output. As can be seen in the diagram for certain input combinations status(0:1) is either 00 or 11. After a few clock cycles, fault Insertion stops briefly to allow normal operation. This makes status(0:1) to become 01 or 10.If status(0:1) = "01" or "10" → Circuit is behaving normally. If status(0:1) = "00" or "11" → Circuit is operating with a fault.

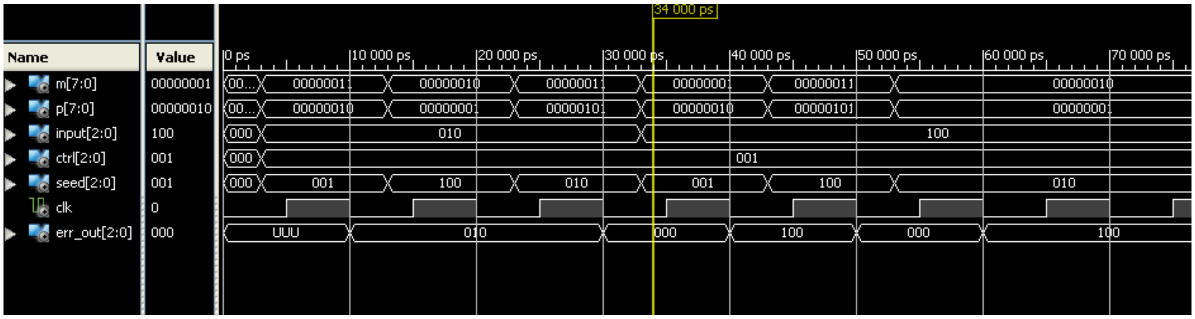


Figure 13.Simulation for offline testing with fault injection

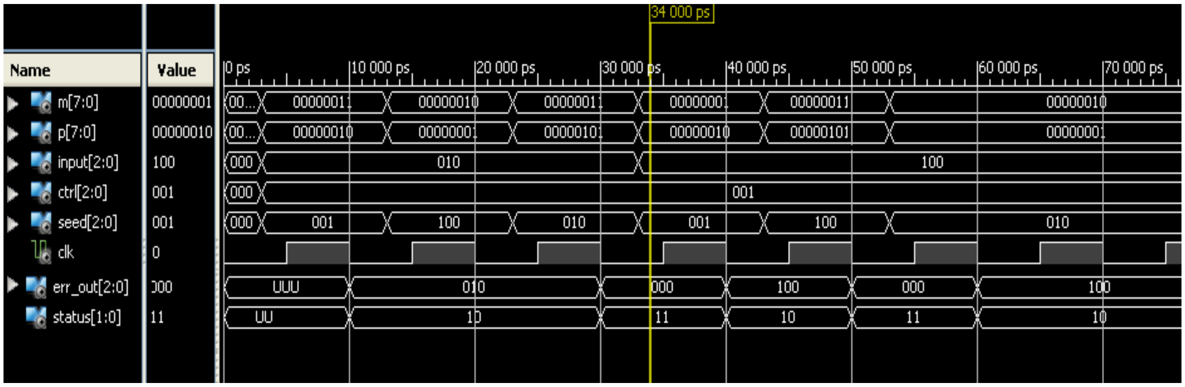


Figure 14.Simulation for offline testing with fault detection status

7. CONCLUSIONS

The proposed injection technique use VHDL to insert a fault on any signal within the block of a VHDL code. It allows the injection of transient faults randomly across a data word.By this technique fault insertion time, can be randomized. A probabilistic model of faulty periods ,the time period where at least one fault exists and a fault analysis to derive the optimum faulty period has been presented. Distribution functions are derived to represent the case of false alarm, where a transient fault is flagged as permanent, and the case of a miss, where too many faults coexist thus overcoming the checker's capability to detect them. These derivations are compared with the results of a simulation program representing the model.

8. REFERENCES

[1]. T. Delong et al., "A Fault Injection Technique for VHDL Behavioral-Level Models", IEEE Design & Test of Computers, 1996, pp. 24-33.

[2]. B .Parrotta et al., "New Techniques for Accelerating Fault Injection in VHDL Descriptions", IEEE 2000 Online Testing Workshop, pp. 61-66.

[3]. R.J. Hayne and B.W. Johnson, "Behavioral Fault Modeling in a VHDL Synthesis Environment, "Proceedings VLSI Test Symposium, April 1999, pp.333-340.

[4]. S. Kamal, C. V**.** Page, "Intermittent faults: A model and a detection procedure", IEEE Trans. Computers,vol C-23, 1974 Jul, pp 713-719.

[5]. ParagK.Lala, Self-Checking and Fault Tolerant Digital Design, Morgan Kaufmann Publishers,2001.

[6]. F. Vargas et al., "Estimating Circuit Fault-Tolerance by Means of Transient-FaultInjection in VHDL", IEEE 2000 Online Testing Workshop, pp. 67.

[7]. N.Z.Basturkmen et al., "A Low Power Pseudo-Random BIST Technique", IEEE 2002 Online Testing Workshop, pp. 140.

[8]. F. P. Kelly, Reversibility and Stochastic Networks**,** John Wiley &Sons, 1979, p 3.

[9]. S. Kamal, "An approach to the diagnosis of intermittent faults",IEEE Trans. Computers,vol C-24, 1975 May, pp 461-467.

[10]. D. C. Bossen, M. **Y.** Hsiao, "Model for transient and permanent error-detection and fault-isolation coverage", IBM J. Research & Development**,** vol 26, 1982 Jan, pp 67-77.