SCHOOL OF ENGINEERING
JÖNKÖPING UNIVERSITY

An Improved Method for Delay Fault Testing of NoC Interconnections

T. Bengtsson[1], S. Kumar[1], A. Jutman[2], R. Ubar[2]

[1]Jönköping University, Sweden
[2]Tallinn University of Technology, Estonia

TTÜ 1918
TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Abstract

Testing of chips built using deep sub-micron technology is becoming harder since crosstalk and small variations in the fabrication processes are adversely affecting circuit dynamic behaviour. The resulting logic and delay faults can only be detected using at-speed testing of the circuits. Faults resulting from crosstalk become more prominent among long links connecting two clock domains in a system designed using Globally Asynchronous Locally Synchronous (GALS) principles. We propose an efficient method for at-speed testing of delay faults in links used for handshaking based communication among GALS domains. The proposed method is conservative in the sense that it can detect all delay faults in links but may also label some good chips as faulty with a small probability. It is possible to extend the proposed method to combine it with functional testing of the link and adapt it for on-line testing

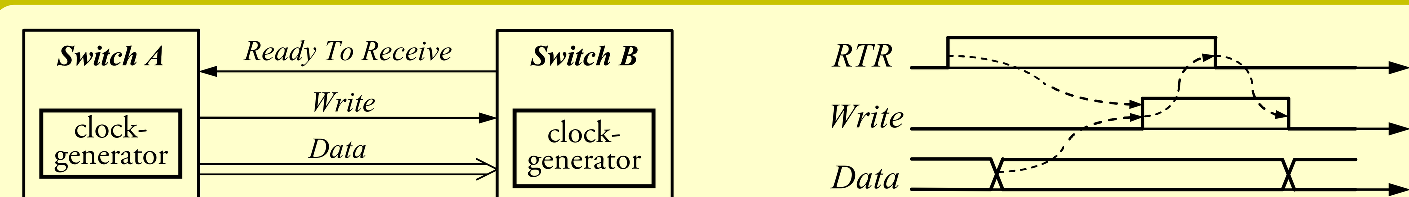
Introduction and Background

NoC circuits are likely to use Globally Asynchronous Locally Synchronous (GALS) approach in order to avoid clock skew and clock distribution problems. In a design using GALS approach, a system is partitioned into multiple domains, each having a different clock.

Testing of NoC circuits involves testing of the infrastructure's functional elements (routers/switches) and conductive interconnect lines. This work concentrates on interconnect testing, which becomes an extremely important research challenge due to the gigahertz operating frequencies and small gaps in the wires of high density chips. In such conditions, the crosstalk among interconnections is becoming a critical issue. Because of small dimensions, variations in the fabrication process are also affecting signal integrity and delays. These variations can lead to faulty behaviour. The faults are more likely to occur in the high speed links connecting two GALS domains. These links consist of closely packed data and address buses along with control and handshaking signals. Many of these faults may not be detected if the chip is tested at lower than the operating frequency.

Interconnection testing in SoCs has many things in common with interconnection testing in PCBs, which has been extensively studied [1].

Handshake Communication and Effects of Delay Faults

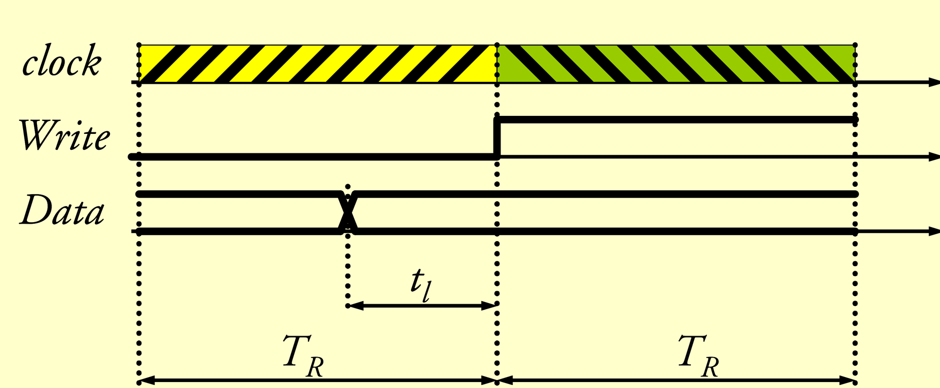


A typical handshaking protocol uses two control lines, *Write* and *Ready To Receive* (*RTR*).

	Effects of Delay Faults	
	Too slow signal	Too fast signal
Write	Degradation of throughput	Risk for data errors
RTR	Degradation of throughput	No problem
Data	Risk for data errors	No problem

The more difficult to detect and in many cases more dangerous fault is if a defect makes a data line more delayed than the signal *Write*. We present a method for detection of such delay faults.

Method for Detection of Delay Fault



This figure shows the clock at the receiver along with signal *Write* and the data lines. T_R is the clock period time at the receiver and t_l is the time from the data arrives until *Write* arrives at the receiver. If t_l is negative a delay fault exists that might result in erroneous data transfer. In each of the striped areas in the figure, one active edge of the clock signal occurs.

The Basic Idea

To detect if a delay fault is present, test data is sent by the transmitter. The receiver reads the data both on its active clock edge just before and on its active clock edge just after signal *Write* has arrived. If the data is correct at the first read it implies that $t_l > 0$ and then the delay fault we are looking for is not present. If the data is erroneous at the second read, there is a fault. If non of these two cases occurs, then this instance of the test could not judge if the fault is present or not. In the presented algorithm this test experiment is repeated until it can be judged if the fault is present or not, or until a certain number of repetitions is made. Appropriate test patterns are put on the data line to cause worst case delay. To achieve this, only one or a few data lines can be tested simultaneously [2].

Algorithmic description of receiver during test

```
Algorithm_Receiver (result, max_no_experiments)
i ← 0;
result ← MIGHT_BE_FAULTY;
WHILE (result = MIGHT_BE_FAULTY) AND
      (i < max_no_experiments) {
  RTR ← 1;
  WAIT ON active clock-edge;
  WHILE (Write = 0) {
    IF Write = 0 THEN
      Read data into register data_first_read;
    WAIT ON active clock-edge;
  }
  Read data into register data_second_read;
  IF data_first_read = Expeceted_data THEN
    result ← FAULT_IS_ABSENT;
  ELSE IF data_second_read ≠ Expeceted_data THEN
    result ← FAULT_IS_PRESENT;
  ELSE {
    RTR ← 0;
    WAIT UNTIL Write = 0;
    i ← i + 1;
  }
}
IF result = MIGHT_BE_FAULTY THEN
  result ← FAULT_IS_PRESENT;
```

Analysis and Results

To analyze the performance of this methodology, we consider one data wire and the signal *Write*. We assume that t_l has a normal distribution. Given a probability for fault and given a nominal value on t_l, the variance of t_l can be computed. Let $f(x)$ denote the density function of t_l. Let $g(t_l)$ denote the probability that one test can detect if the fault is present or not, given a specific t_l.

$$g(t_l) = \begin{cases} 1 & ;|t_l| \ge T_P \\ \frac{|t_l|}{T_P} & ;|t_l| < T_P \end{cases}$$

Let r_n be the percentage of chips for which judgment cannot be made. Let n_t be number of tests required on average. It can then be proved that $\lim_{r_n \to 0} n_t(r_n) = \infty$. Given r_n, the maximum and the average number of tests can be computed. Let p be the probability that one experiment detects if the fault is present or not. This probability is in itself a stochastic variable and the first step in the calculations is to compute its density function $r(x)$.

$$\int_a^b r(x)dx = \int_{a<g(t)<b} f(t)dt \qquad \forall(a,b) \in R \qquad ,a<b$$

Let $p_k(n)$ be the probability that n instances of the test can judge if the fault is present or not. For a given probability p, the probability $p_k(n)$ is $1 - (1 - p)^n$.

$$p_k(n) = \int_{-\infty}^{+\infty} r(x) \cdot \left(1-(1-x)^n\right)dx$$

This formula can be used to decide after how many instances of the test to give up. Let l denote after how many experiments to give up. Let $p_l(n)$ be the probability that an experiment is made exactly n times. For $n < l$ this means that $p_l(n)$ is the probability that the presence of the fault can be judged at the n:th experiment. For the case $n = l$, $p_l(l)$ is the probability that l-1 experiments cannot judged if the fault is present. Let $h_l(p)$ be the expected value of $p_l(n)$ given an l and a p. Let n_t be the average number of instances of the test given l.

$$p_l(n) = \begin{cases} (1-p)^{l-1} \cdot p & ;1<l<n \\ \sum_{k=l}^{\infty}(1-p)^{k-1} \cdot p = (1-p)^{l-1} & ;l=n \\ 0 & ;l>n;l \le 1 \end{cases}$$

$$h_l(p) = \sum_{i=-\infty}^{+\infty} i \cdot p_l(i) = \left(\sum_{i=1}^{l-1} i \cdot (1-p)^{i-1} p\right) + l \cdot (1-p)^{i-1}$$

So average number of instances of at test $n_t = \int_{-\infty}^{+\infty} r(x) \cdot h_l(x)dx$

Results

t_l / T_r \ P_f	Average no tests required n_t			Max number of tests l		
	0.1	0.01	0.001	0.1	0.01	0.001
1	2.0 (2.9)	1.8 (2.8)	1.4 (1.7)	13	124	273
0.5	4.8 (13)	3.3 (5.2)	2.6 (3.3)	91	223	619
0.1	24 (62)	16 (27)	13 (15)	448	1178	2244

The table shows sample results from the computation described above. This is made for different fault probabilities P_f and for different expected t_l. Parameter r_n is chosen to be one tenth of P_f. Figures in brackets shows number of tests needed if method in [3] is used. The new method on average terminates faster for bad chips. The improvement is larger for higher fault probabilities. The average number of tests is much smaller than the maximum number.

Conclusions

In this work we have presented a simple method to detect delay faults in links connecting two GALS domains in NoC architectures. The method can however lead to marking some good circuits as faulty with a certain probability. It is on the other hand possible to reduce this probability to an acceptable level by keeping the max-limit of number of times the experiments is repeated, sufficiently high. The method has potential of getting combined with at-speed functional testing of interconnections as well as for on-line testing.

References

[1] A. Jutman, "At-Speed On-Chip Diagnosis of Board-Level Interconnect Faults", in *Formal Proceedings of 9th European Test Symposium*, France, May 2004, pp. 2 – 7

[2] M. Cuviello, S. Dey, X. Bai and Y Zhao "Fault Modeling and Simulation for Crosstalk" *in System,-on-Chip Interconnects* in Proc. of ICCAD, Nov. 1999, pp 297 – 303

[3] T. Bengtsson, A. Jutman, S. Kumar and R. Ubar, "Delay testing of asynchronous NoC interconnects", *12th International Conference Mixed Design of Integrated Circuits and Systems*, June 2005.