

Yield Modeling and Analysis of a Clockless Asynchronous Wave Pipeline with Pulse Faults

T. Feng and N. Park
Dept. of Computer Science
Oklahoma State University
Stillwater, OK 74078-1053
npark@cs.okstate.edu

Y.B. Kim
Dept. of ECE
Northeastern University
Boston, MA 02115
ybk@ece.neu.edu

V. Piuri
Dept. of IT
University of Milan
Bramante 65, 26013, Crema, Italy
piuri@dti.unimi.it

Abstract

This paper proposes a new fault model and its modeling and analysis methods in a clockless asynchronous wave pipeline for extensive yield evaluation and assurance. It is highly desirable to have an adequate and specific pulse fault rate model for establishing a sound theoretical foundation for clockless wave pipeline design for reliability. The pulse fault model is thoroughly identified as the unique fault specifically in the clockless wave pipeline in comparison with conventional wave and wave delay faults. The pulse fault rate is statistically yet practically modeled, and extensively evaluated with respect to various design parameters, such as yield, fault coverage, defect-level, and request level length. An extensive numerical simulation is conducted to demonstrate the effect of the proposed pulse fault on the yield.

1. Introduction

Wave pipeline is a cutting-edge technology for extending the performance of modern microprocessors to their maximum. This technology has been extensively researched to achieve significant performance upgrade [4]. Wave pipeline can be implemented either synchronously or asynchronously. Clockless wave pipeline is one of the asynchronous implementations without internal clock-controlled-registers to further improve the performance of synchronous wave pipeline. Instruction pipeline is a realization of linear synchronous pipeline in which performance is improved through instruction-level parallelism by allowing to start execution of an instruction before the previous ones already in the pipeline are finished.

Recently, researches have been focused on clockless asynchronous wave pipeline (AWP), which replaces the clock with request and acknowledgement signal, or just request signal to realize clockless asynchronous wave pipeline operation. Basic architectures of AWP have been presented in [7], [8]. Since logic stages operate asynchronously in AWP, AWP is potentially faster than SWP, and thus more attention has been paid recently. AWP has two basic types. One uses both request and acknowledgement signals, i.e., handshaking protocol. This kind of AWP is able to buffer signals to impose

strict control by the clock, but extra delay is required due to the waiting period for acknowledge signal. The other kind of AWP uses request signal only without acknowledgement signaling but could run much faster than the former kind of AWP. This kind of AWP is to be investigated in this work. However, fault models, testing, fault tolerance and defect-level issues of such AWP have not been adequately and extensively addressed.

The objective of this paper is to identify an adequate and clockless AWP-specific fault model, and provide a sound theoretical yet practical foundation for AWP design for reliability. Specifically, this paper will address and propose a method for modeling and analysis of the effect of the new AWP-specific fault associated with AWP design parameters on yield, defect-level, thereby providing an effective and accurate model for AWP testing algorithms and ultimately fault-tolerance.

2. Preliminaries and Review

There are three major potential advantages of asynchronous wave pipeline (AWP) [9].

1. Large system can be easily built and modified in a plug and play fashion because it is not necessary to consider the global synchronization of different components.
2. Extra consideration for clock path design and extra power for clock signal can be avoided in AWP.
3. AWP may outperform SWP because fast data does not have to wait for a clock signal to be synchronized to slow data.

However, due to the clockless and asynchronous operation, clockless asynchronous pipeline can not be adequately facilitated by synchronous register-based control.

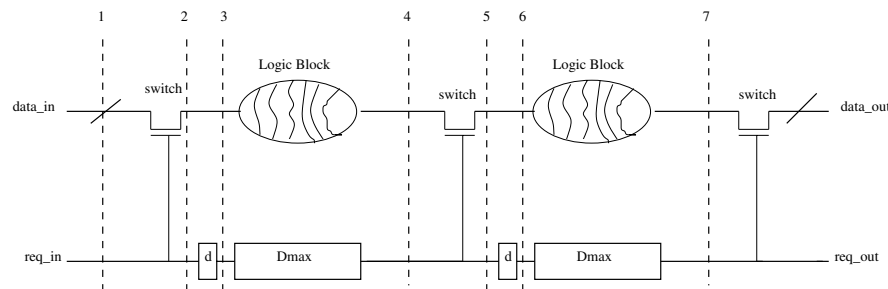


Figure 1. Two-phase asynchronous wave pipeline [7]

An asynchronous wave pipeline was proposed by Hauck [7], called *two-phase asynchronous wave pipeline*, which has a two-phase operation by alternating positive and negative level-sensitive switches. Instead of clock signal-based control AWP relies on a *request signal*. Although the duty of the request signal is quite similar to conventional clock signal and is generated by a clock at the primary input, the operation of switches and logic stages are asynchronous while a clock drives all registers in parallel in SWP.

The switches separate logic stages. A *request line* controls the switches behaving as a reference signal. Data waves and request pulses enter the AWP at the same time. They need to stay coherent until hit the primary output. *Delay elements* are added in the request line to emulate the propagation delay of the data waves. The delay element denoted by d represents the propagation delay of the switches. The other delay element denoted by D_{max} represents the worst-case propagation

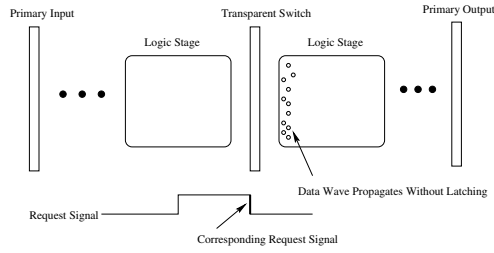


Figure 2. Transparent Switch

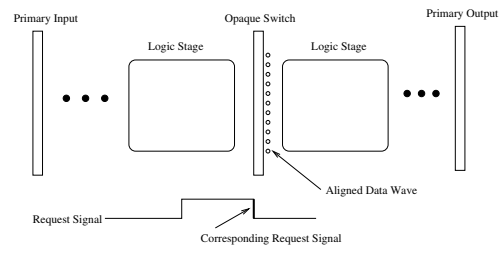


Figure 3. Opaque Switch

delay of the logic stage. Ideally, the request signal always get aligned to the last bit of the wave. Although the paths are equalized for delay-balancing, the path delay variations still exist. Hence, data propagating through logic stage may get skewed since some bits run faster or slower. As a result, the wave spreads wider as the wave propagates through a path for a longer period of time. If the shape (i.e., the width) of the waves is not controlled, then the wave pipeline's utilization will not realize its maximum high.

If a wave becomes exceedingly wide, then it may interfere its adjacent waves. Consequently, less number of waves can be accommodated in the AWP at the same time. A solution to this problem is using *positive and negative switches*, which can be inserted in between the logic stages. The switches can be *transparent* or *opaque* to the data wave. If transparent, the switch turns off for a data wave just to propagate through the switch without any latching as shown in Fig.2. Then, the data bits start to propagate on through the next stage before the arrival of the request signal. If opaque, the switch turns on to latch through-data wave as shown in Fig.3. The data bits are latched being aligned at the output line of the switch. They can not propagate on forward until the arrival of their corresponding request signal. Hence, the data wave gets aligned at the output line of the switch. Thus, more waves can be active within the AWP simultaneously, and AWP's throughput is increased. Specifically, positive switches are opaque to the data associated with low request signal and transparent to those associated with high request signal, and vice versa for negative switches.

The throughput of AWP is determined by the minimum length of the low or high level of the request signal. The following parameters will be used for the timing analysis.

- *Partial circuit*: any two adjacent logic stages within a AWP circuit and their corresponding switches.
- *Partial path*: a path within a partial circuit.
- d_{max}, d_{min} : maximum and minimum propagation delay of a partial path within any partial circuit.
- D_{max}, D_{min} : maximum d_{max} and minimum d_{min} within the AWP circuit, respectively.
- Δ : $d_{max} - d_{min}$
- Δ_{max} : maximum Δ
- Δ_{ps} : uncontrolled pulse skew.
- T_{su}, T_h : switch setup and hold time, respectively.
- d : propagation delay of a switch.
- L_{min} : minimum length of request level.

For simplicity, suppose a data wave from the primary input to the primary output is traced to calculate the L_{min} . Normally, all the bits of a data wave enter the AWP at the same time as aligned

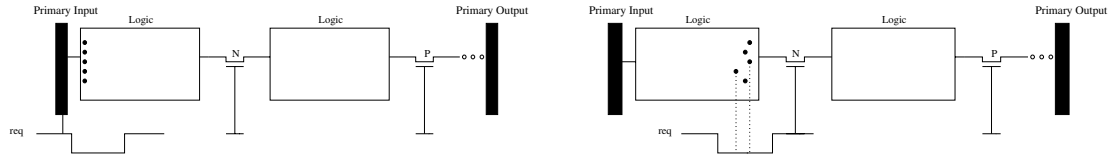


Figure 4. Beginning of a data wave entering a transparent switch **Figure 5. Data wave before a transparent switch**

as shown in Fig.4. Also as shown in Fig.4, suppose a low request signal is associated with the data wave. The first switch the data wave will hit is an N -switch, and the N -switch is transparent to low data. Therefore, all the bits pass on through the N -switch without being latched as shown in Fig.5. After propagating through the N -switch, the data wave continues to propagate through the next logic stage and hits the next switch, a P -switch, which is opaque to the low data. Therefore, the data bits gets latched and aligned at the output line of the switch. At this point, the width (i.e., the spread of the data bits in the data wave) may become the widest on this partial path. The Δ can be captured at this point. After passing through this switch, the width of the data wave returns to 0 (i.e., aligned). Then, the data wave will propagate on toward the next partial path to get aligned untill the next P -switch. Then at this point, another Δ can be captured. Continuing on this way, finally at the primary output, the maximum Δ can be captured as referred to as Δ_{max} .

In order to make the data wave successfully passing through the AWP without faults, the request signal level must always maintain the proper association with its corresponding data wave. The minimum length of request level is

$$L_{min} = \Delta_{max} + T_{su} + T_h + d + 2\Delta_{ps}. \quad (1)$$

The reason for T_{su} and T_h is that the data wave must stay stable before the rising edge and after the falling edge of the request signal. Otherwise, the input data will be observed as an invalid input data by the switch and correct output value cannot be guaranteed. The high-data is delayed by the N -switch and moves closer to the falling edge while it has to become stable before the falling edge [7]. Therefore, the request signal must operate with an adequate delay (i.e., d) to orchestrate with the switch delay.

3. Proposed Pulse Fault Rate

Path delay of a circuit with fabrication variations can be modeled by using an interval than a single value. Let x denote the path delay of a partial path. Suppose there are n partial paths in a partial circuit, e.g., $x_1, x_2, \dots, x_n$. All of the partial paths are assumed to be well balanced in the AWP under investigation. Then, without loss of generality, x can be modeled by using the normal distribution with a mean μ and a standard deviation σ as follows.

$$x \sim N(\mu, \sigma^2). \quad (2)$$

Let $\bar{x}$ be the sample mean, and s be the standard error, i.e. $\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ and $s = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n-1}}$.

Then μ and σ can be substituted with $\bar{x}$ and s , respectively, because $\bar{x}$ and s are unbiased estimator of μ and σ .

$$x \sim N(\bar{x}, s^2). \quad (3)$$

Suppose $f(x)$ is the *p.d.f.* (probability density function) of x as follows.

$$f(x) = \frac{1}{\sqrt{2\pi}s} e^{-\frac{(x-\bar{x})^2}{2s^2}} \quad (4)$$

Then the probability that some bits in the data wave proceed too fast and then overstep their associated request level (i.e., the probability of setup time pulse fault) or some bits in the data wave are too slow such that they lag behind the associated request level (i.e., the hold time pulse fault) is the probability that some bits are out of the range of the request level interval as shown in Fig.6 (where the shaded areas represent the probability of pulse fault).

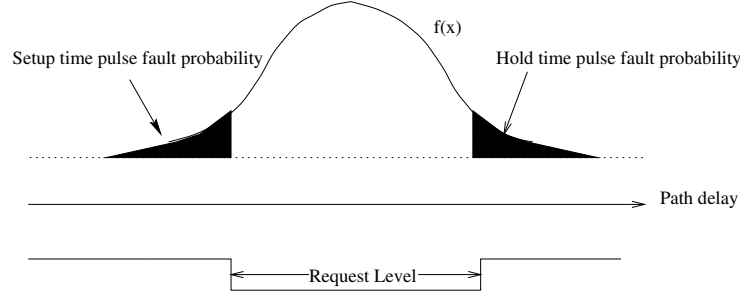


Figure 6. Pulse Fault Probability

The request signal and all the data bits enter the circuit at the same time, and for proper operation the request signal should be slower than the slowest bit of the data wave and reach the switch after the slowest bit. Likewise, the previous request signal should reach the switch before the fastest bit of the associated data wave. The data skew (represented by Δ in Fig.7) is supposed to be covered by the associated request level properly. Therefore, the coverage of Δ by the request level, i.e., the relative position between the data wave and its associated low or high request level may influence the pulse fault rate to a great extent.

Lets α refer to the difference of propagation time between the slowest bit of the data wave and the request level as shown in Fig.7. The associated request signal's propagation delay (denoted as d_s) of the data wave can be expressed as $d_s = d_{max} + \alpha$. The propagation delay of the request signal pulse through the switch (denoted as d_p) is $d_p = d_{min} - (L - \alpha - \Delta)$. Thus, $L = d_s - d_p$.

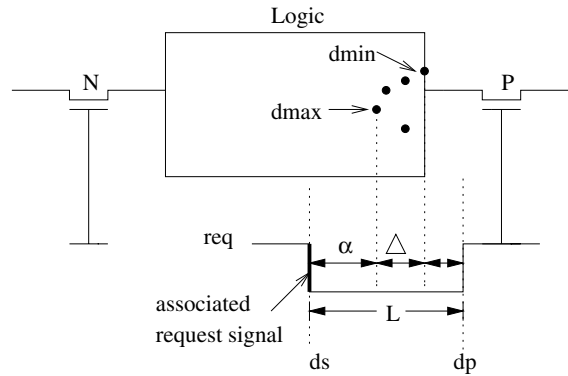


Figure 7. Relative position between the request signal and data wave

Table 1. Parameter Values Simulated

-	# path	testability	path 1	path 2	path 3	path 4	path 5	path 6
$S_0 - S_2$	4	0.78	27	24	26	22	-	-
$S_1 - S_3$	5	0.89	60	58	64	62	63	-
$S_2 - S_4$	6	0.95	37	34	33	35	32	30
$S_3 - S_5$	5	0.86	72	75	78	70	77	-
$S_4 - S_6$	6	0.91	40	43	42	44	41	46

The pulse fault rate at switch i is then:

$$P_i = 1 - \int_{d_{min_i} - (L - \alpha_i - \Delta_i)}^{d_{max_i} + \alpha_i} \frac{1}{\sqrt{2\pi}s} e^{-\frac{(x-\bar{x})^2}{2s^2}} dx \quad (5)$$

If we have n switches in the AWP and P_i is the pulse fault rate at each switch where $1 \leq i \leq n$, then the *total pulse fault rate* (P_{total}) of the AWP is as follows.

$$\begin{aligned} P_{total} &= 1 - \prod_{i=1}^n (1 - P_i) \\ &= 1 - \prod_{i=1}^n \left(1 - \int_{d_{min_i} - (L - \alpha_i - \Delta_i)}^{d_{max_i} + \alpha_i} \frac{1}{\sqrt{2\pi}s} e^{-\frac{(x-\bar{x})^2}{2s^2}} dx \right) \end{aligned} \quad (6)$$

For simulation, suppose there is a circuit with 7 switches (e.g., $S_0 - S_6$ as provided in Table 1). Also, suppose that the number of paths within any partial circuit (the # path in the table), and their path delays are known. Lastly, assume that the testability (i.e., the probability to detect the pulse faults) at each switch is known as provided in Table 1 with the unit ns .

From Table 1, $\Delta_{max} = 8 ns$ can be calculated. It is also assumed that $d = 1 ns$, $T_{su} = 0.6 ns$, $T_h = 0.2 ns$, and $\Delta_{ps} = 0.1 ns$. Then $L_{min} = \Delta_{max} + T_{su} + T_h + d + 2\Delta_{ps} = 10 ns$.

The length of the request level is set to L_{min} first, i.e., $10 ns$, and then to the larger values up to $15 ns$. It is shown that the total pulse fault rate at $15 ns$ becomes very low as shown in Fig.8. For this simulation, SAS (Statistical Analysis Software) was used to compute the standard deviations of those path delay samples in Table 1. Then, from the *c.d.f.* (cumulative density function) of Equation 5, the pulse fault probability were calculated.

By using Equation 6, the total pulse fault rate can be calculated. Then, the yield, the fault coverage and the defect level can be calculated as well. The simulation results are shown in Fig.8, Fig.9, Fig.10 and Fig.11.

In summary, the simulation results have demonstrated that the proposed pulse fault in association with the request level length has great effects on the integral pulse fault rate, as shown in Fig.8. All four possible P_i drop to less than 5% when the request level length extends from L_{min} (i.e., $10 ns$) up to $15 ns$. An increased request level length may significantly affect the pulse fault rate to decrease, while in consequence the yield is enhanced and defect level is reduced significantly. The fault coverage is evaluated with an arbitrary sample testability distribution as shown in Table I. Comparing the four strategies for P_i , it can also be observed that strategies 1 and 2 demonstrate similar performances, and strategies 3 and 4 demonstrate similar performances, in which the last two strategies 3 and 4 demonstrate less pulse fault rate. Thus strategies 3 and 4 can be suggested to be more suitable for implementation in practice.

4 Conclusion

This paper has proposed the pulse fault and yield model for the two-phase clockless asynchronous wave pipeline. and yield analysis. The pulse fault model has been thoroughly identified as the unique fault of the clockless wave pipeline in comparison with wave and wave delay faults of conventional wave pipelines. The pulse fault rate is statistically yet practically modeled, and extensively evaluated with respect to various design parameters, such as yield, fault coverage, defect-level, and request level length. The simulation results have revealed that the proposed pulse fault in association with the request level length has great effect on the integral pulse fault rate. Also, it has been demonstrated that an increased request level length may significantly affect the pulse fault rate to decrease, while in consequence the yield is enhanced and defect level is decreased significantly. Moreover, it has been observed and analyzed that placing Δ in the middle of L or dividing d and $\Delta_{max} - \Delta$ equally to stride at both sides of Δ can reduce the pulse fault rate drastically. In conclusion, the proposed modeling and analysis provides a theoretical yet practical foundation for a feasible clockless asynchronous wave pipeline design strategy for reliability.

References

- [1] Burleson.W.P , Ciesielski.M, Klass.F, Liu.W, “Wave-Pipelining: A Tutorial and Research Survey”, *IEEE Transactions on very large scale integrate (VLSI)system*, VOL.6, NO. 3, September 1998.
- [2] Cotton. L. W, “Maximum-rate pipeline system”, *AFIPS Proceedings Spring Joint Computer Conference*, vol. 34, Montvale, NJ: AFIPS Press, pp.581-586, May 1969
- [3] Fishburn.J, “Clock Skew Optimization”, *IEEE Trans. on Computers*, 1990
- [4] Gray.T, Hughes.T, Arora.S Liu.W, Cavin.R, “Theoretical and Practical Issues in CMOS Wave Pipelining” *VLSI Design*, pp, 9.2.1-9.2.5, 1991
- [5] Gray.C, Liu.W.T, Cavin.K.R, “Timing Constraints for Wave-Pipelined Systems”, *IEEE Transactions on computer-aided design of integrated circuits and systems*, vol., 13, NO, 8, August 1994.
- [6] Heald.R, Shin.K, Reddy.V, Lynch.W, “64-Kbyte-Sum-Addressed-memory Cache with 1.6-ns Cycle and 2.6-ns Latency”, *IEEE Journal of Solid-State Circuits*, vol. 33, no. 11, PP. 1682-1689, Nov. 1998
- [7] Hauck.O, Garg.M, Huss.A.S, “Two-phase asynchronous wave-pipelines and their application to a 2D-DCT”, *Proc., Int. Sym. on Advanced Research in Asynchronous Circuits and Systems*, , page(s): 219-228, 1999
- [8] Hauck.O, Huss.A.S, “Asynchronous wave pipelines for high throughput data paths”, *IEEE International Conference on Electronics, Circuits and Systems*, , Vol. 1, pp. 283 -286 vol.1, 1998
- [9] Hermanns.S, Huss.S.A, “Embedding of Asynchronous Wave Pipelines into Synchronous Data Processing” *SAME Conference 2001, 14., 15., Sophia Antipolis, Valbonne, France*, November 2001
- [10] Lein.H.W, Burleson.W, “Wave domino logic: theory and applications”, *Proc.Int.Symp.Circuits Syst.*, 1992

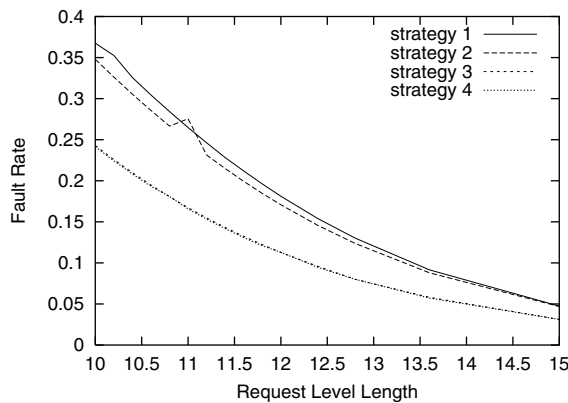


Figure 8. Fault Rate

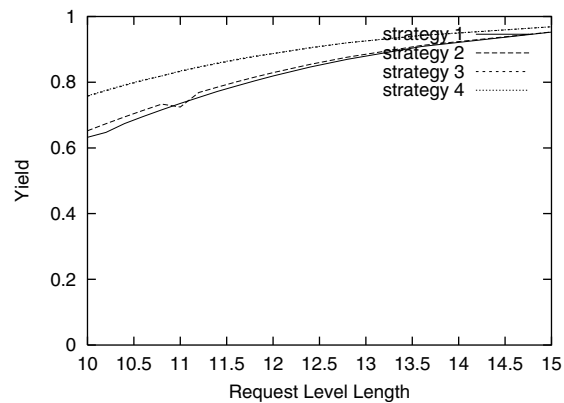


Figure 9. Yield

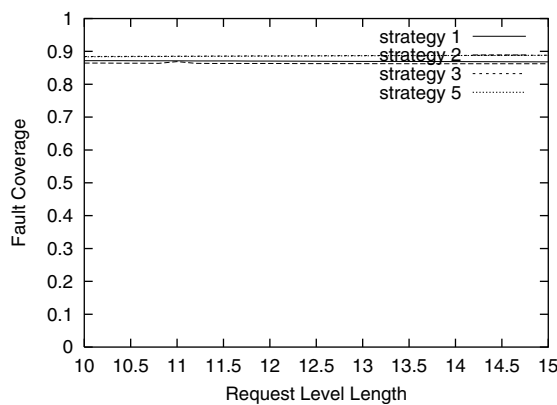


Figure 10. Fault Coverage

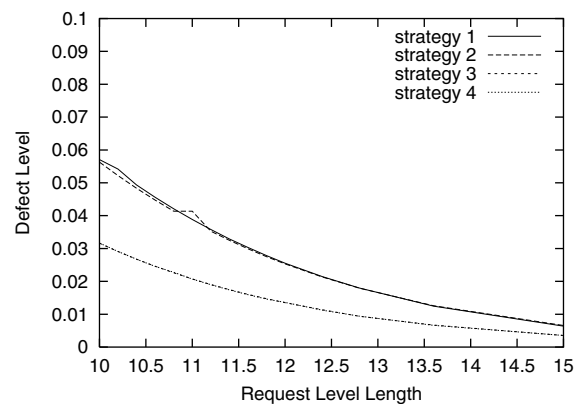


Figure 11. Defect Level

- [11] Shyur.J.C ,Chen. H.P, Parng.T.M, "On testing wave pipelined circuits", *Proceedings of the 31st annual conference on Design automation conference* ACM Press, Series-Proceeding-Article, pp. 370-374, 1994
- [12] Sakallah.A.K, Mudge.N.T, Burks.T.M, Davidson.S.E, "Synchronization of pipeline", *IEEE Trans on Computer-Aided Design*, vol.12, 1993.
- [13] Shenoy.V.N, Brayton.K.R, Sangiovanni-Vincentelli.L.A, "Minimum padding to satisfy short path constraints", *proc. Int. Wrokshop logic Synthesis*, 1993
- [14] Wu.W.C, Lee.C.Len, "A probabilistic testability measure for delay faults", *Proceedings of the 28th conference on ACM/IEEE design automation conference*, p.440-445, June 17-22, 1991
- [15] Venkataraman.A , Koren.I "Determination of yield bounds prior to routing", *Proc. of the 1999 IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, pp. 4-13, November 1999.
- [16] Wong.D, Micheli.G.De , Flynn.M, "Inserting active delay elements to achieve wave pipelining", *Proc. Int. Conf. Computer-Aided Design*, pp. 270-273, Nov. 1989
- [17] Williams.T.W, Brawn.N.C, "Defect level as a function of fault coverage", *IEEE Trans. on computers*, C-30, 1981.