The Impact of Delay Optimization on Delay Fault Testing Quality

Young-Ho Park and Eun-Sei Park

Abstract

In delay-optimized designs, timing failures due to manufacturing delay defects are more likely to occur because the average timing slacks of paths decrease and the system becomes more sensitive to smaller delay defect sizes. In this paper, the impact of delay optimized logic circuits on delay fault testing will be discussed and compared to the case for non-optimized designs. First, we provide a timing optimization procedure and show that the resultant density function of path delays is a delta function. Next we also discuss the impact of timing optimization on the yield of a manufacturing process and the defect level for delay faults. Finally, we will give some recommendations on the determination of the system clock time so that the delay-optimized design will have the same manufacturing yield as the non-optimized design and on the determination of delay fault coverage in the delay-optimized design in order to have the same defect-level for delay faults as the non-optimized design, while the system clock time is the same for both designs.

I. Introduction

There are many paths through combinational blocks within logic networks and, historically, the delays along those paths have differed widely. Sophisticated logic synthesis tools are capable of reducing the worst-case path delay by increasing delays along other paths [1]. Path delay optimization will have a significant impact on delay fault testing and this work is about the trend toward more equal path delays and its resulting impact on delay testing methods and quality. In short, the results of delay optimization can skew the path delay distribution such that a large percentage of paths are very close to the maximum delay path [2]. This results in much higher sensitivity to delay defects. In particular, as path delays become more nearly equal, the average timing slack will decrease, and the system will be more sensitive to smaller delay defect sizes[3, 4].

The above description is a simple qualitative description of the impact of delay-optimized designs on delay testing and test quality. These simple concepts are developed in a more formal and quantitative manner in this paper. We are concerned with two attributes of delay-optimized designs -- (1) the adjustment of path delays through the combinational blocks of logic networks, and (2) testing for delay faults introduced in the manufacture of integrated circuits.

This paper is organized as follows. In Section II, we discuss the adjustment of path delays through the combinational blocks of logic networks as shown in Fig.1. We also present a delay-optimization method and show that the resultant density function of path delays in a delay-optimized network will converge toward a delta function or toward a density function that includes a delta function. In Section III, we discuss the impact of this type of synthesized network on the yield of a manufacturing process and the defect level for delay faults. Finally, some implications of the inter-dependence between delay optimization and delay testing will be given in Section IV.

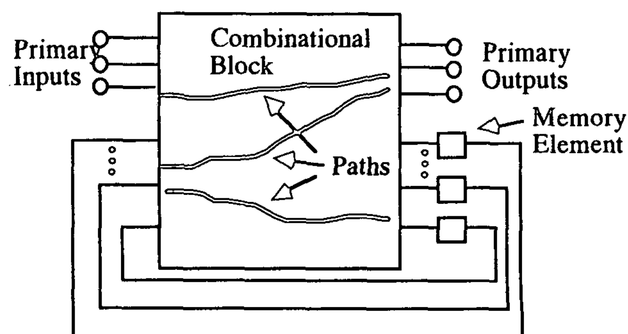


Fig. 1. A logic network model.

Manuscript received May 3, 1996; accepted May 26, 1997.

Y. H. Park is with HW Environment Section, Electronics and Telecommunications Research Institute P.O. Box 106, Yusung, Taejon.

E. S. Park is with Department of Electronics Engineering, Hanyang University Sa-1 dong 1271, Ansan-si kyunggi-do, 425-791 Korea.

III. Delay Optimization of a Logic Network

1. Basic Ideas

We now analyze the nature of path delay distributions in a network that has undergone some form of optimization for timing. Fishburn and Dunlop[5] investigated three different types of delay optimization at the device level,

(1) minimizing A subject to the constraint $T < K$.
(2) minimizing T subject to the constraint $A < K$,
(3) minimizing AT^K,

where A is chip or network area, T is clock time, and K is an arbitrary constant.

While the following discussions could focus on any or all of these optimization problems, without loss of generality we will demonstrate them only for case 1, the minimization of area for a fixed clock time. In this paper, we will use the concept of silicon area as a resource that may be used to adjust path delay. We assume that the adjustment of path delays can be done through the adjustment of the dimensions of either the transistors or the interconnection wires. In the following discussions, we will make some simplifying assumptions. The first is that the optimization process will not change the topological structure of the original network, that is, the same logic gates are present and they are connected in the same fashion. The second assumption is that area reduction is a continuous rather than a quantized process, so that there is never a minimum area by which a network can be reduced. The third assumption, made only for simplicity of explanation, is that each gate has equal rise and fall delays. Although the above assumptions are valid under ideal cases, recent delay optimization techniques for gate and interconnect sizing in circuit-level designs often employ similar assumptions for practical applications [6, 7].

We first define some terminologies used throughout this paper.

Definition : A *segment* of a logic network is a signal line between two directly connected logic elements and primary inputs and primary outputs. A *subpath* of a network is a collection of one or more segments that are consecutively connected. A *path* of a network is a subpath from a primary input to a primary output in a network. Note that no path will ever be a perfect subpath of any other path because at least the primary inputs or the primary outputs must differ.

Lemma 1 : In a network with two paths P_1 and P_2, through the combinational logic having a common subpath S_c, a subpath S_1 unique to P_1, and a subpath S_2 unique to P_2, either the paths P_1 and P_2 have equal path delay or the overall area can be reduced without increasing the clock time. This operation will equalize delay along the two paths[2].

2. A Delay Optimization Procedure

In any network with N paths $P_1, P_2, ..., P_N$ from the primary inputs to the primary outputs of combinational blocks, we will perform the following optimization to reduce the area for a fixed clock time. Assume that P_N is the path with the largest delay D_{max}. Then the clock time C is at least as large as D_{max}. Let us define a set S_{max} , which includes the paths of delay D_{max}, and a set S_0 that is initially empty. During the delay optimization process, we will partition all the N paths into the sets S_0 and S_{max}. Fig. 2 shows a timing optimization procedure.

After this iterative process in Fig. 2, we have partitioned all the N paths into the sets S_0 and S_{max}. Every path in S_{max} is guaranteed to have the same delay D_{max}. If the set S_0 is indeed empty, the distribution of path delays is a unit step function with step at D_{max} and the density function for this distribution is $\delta(D_{max})$. If S_0 is not empty, the density function of path delays includes a delta function $(k/N) \cdot \delta(D_{max})$, where k is the cardinality of S_{max}. If the adjustment of path delays is done through the adjustment of transistor sizes only as in [2], the resultant path delay density function of a delay-optimized design will converge toward a delta function or a density function that includes a delta function as shown in Fig. 3. However, if we assume that the adjustment of path delays can be done through the adjustment of either the transistor sizes or the interconnection wire lengths, then the set S_0 is always empty and the density function of path delays for the delay-optimized network is a delta function. A proof will be presented in a later section.

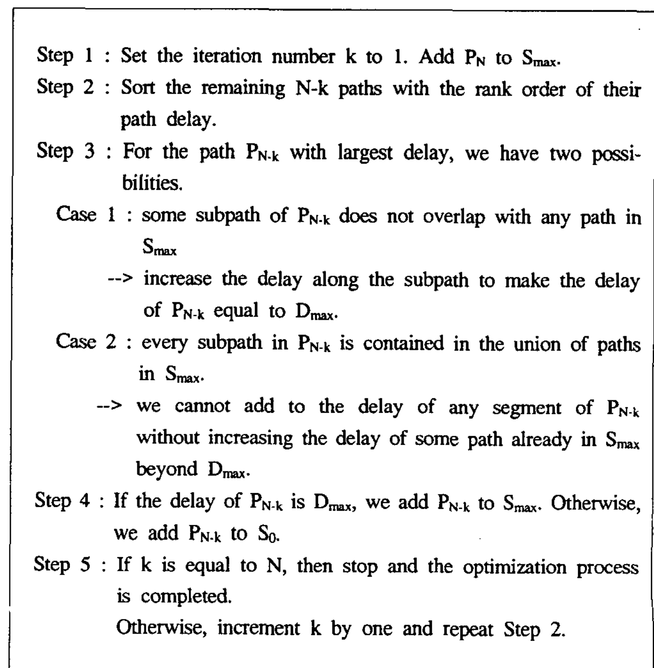

Step 1 : Set the iteration number k to 1. Add P_N to S_{max}.
Step 2 : Sort the remaining N-k paths with the rank order of their path delay.
Step 3 : For the path P_{N-k} with largest delay, we have two possibilities.
Case 1 : some subpath of P_{N-k} does not overlap with any path in S_{max}
--> increase the delay along the subpath to make the delay of P_{N-k} equal to D_{max}.
Case 2 : every subpath in P_{N-k} is contained in the union of paths in S_{max}.
--> we cannot add to the delay of any segment of P_{N-k} without increasing the delay of some path already in S_{max} beyond D_{max}.
Step 4 : If the delay of P_{N-k} is D_{max}, we add P_{N-k} to S_{max}. Otherwise, we add P_{N-k} to S_0.
Step 5 : If k is equal to N, then stop and the optimization process is completed.
Otherwise, increment k by one and repeat Step 2.

Fig. 2. A procedure for timing optimization.

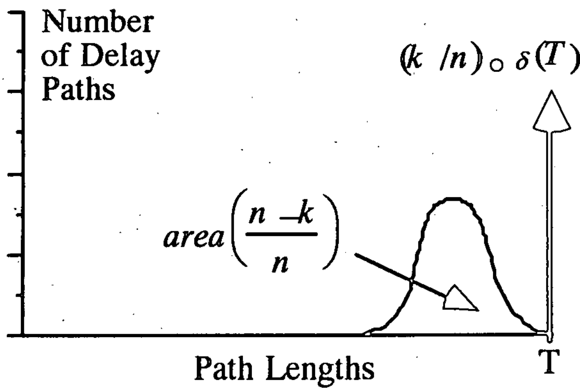


Fig. 3. Histogram of delay-optimized networks.

3. An Idealized Density Function of Path Delays in a Delay-Optimized Design

To represent the connectivity of a set of paths in a logic network, we define a graph G(V, E(w)) as a weighted acyclic directed graph, where V is the set of vertices, E is the set of directed edges, and w is a weight associated with each edge of the graph. Each vertex in V represents a logic element, a primary input, or a primary output. Each edge in E represents one line segment between two adjacent logic elements along a path. The weight of an edge is the number of paths that share the edge and thus paths can be enumerated by using the edges associated with it. For example, the graph for the paths A-C-D-E and B-C-D-F-G in the circuit of Fig. 4.(a) is shown as in Fig. 4.(b). Note that the weight of the edge between C and D in the graph is two because the segment C-D in the circuit diagram is contained in the two paths.

When we add an edge into a graph, if the edge already exists

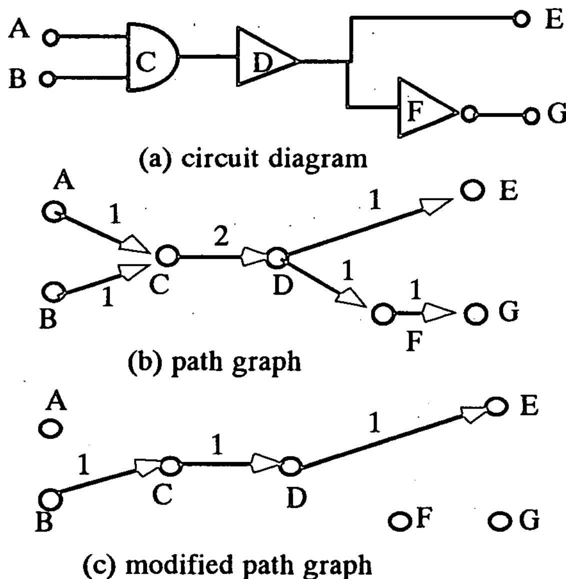


Fig. 4. A graph for representing the connectivity of a set of paths.

in the graph, we increment the weight of the edge by one. If the edge does not exist in the graph, we create the edge with weight one. Also, removing an edge from a graph is the reverse of adding an edge into a graph. Notice that there is no incoming (outgoing) edge for a primary input (output).

Lemma 2 : For any internal vertex in a graph, the sum of weights for its incoming edges is equal to the sum of weights for its outgoing edges. Also, the sum of weights for the incoming (or outgoing) edges of an internal vertex is equal to the number of paths which pass through the vertex.

Lemma 3 : The sum of weights for the incoming edges of all the primary outputs (sink vertices) in a path graph is same as the sum of weights for the outgoing edges of all the primary inputs (source vertices), and this sum is the total number of paths represented in the graph.

Lemma 4 : From a source vertex whose sum of weights for the outgoing edges is not zero, we can construct a path from that source vertex to a sink vertex.

Proof : Since the sum of weights for the outgoing edges of the source vertex is not zero, there exists an edge with nonzero weight from the source vertex to a vertex v. If the vertex v is not a sink vertex, we know that there exist an edge with nonzero weight from the vertex v to another vertex because the sum of weights for the incoming edges of the vertex is same as the sum of weights for its outgoing edges from Lemma 3. This procedure can be repeated until a sink vertex is encountered.

Q.E.D.

Lemma 5 : If every segment in a candidate path P is already included in S_{max}, then we can construct a new set S of paths such that (1) P is a member of S, (2) cardinality of S = cardinality of S_{max}, and (3) all the edges in the graph associated with S_{max} are contained in the graph associated with S, with the same weights.

Proof : First we partition the paths in S_{max} into two sets. The set A is the set of paths in S_{max} which do not intersect with the path P. The set B is the set of paths in S_{max} which intersect with the path P. Clearly, the sets A and B are disjoint. The union of paths in the set B completely include the path P and each of the paths has at least one subpath which intersects with the path P.

Let the cardinality of the set B be *N*. For the paths in the set B, we construct the graph. From Lemma 2, we know that the sum of weights for the incoming edges of any internal vertex in the graph is equal to the sum of weights for its outgoing edges. Now, we construct *N* different paths whose graph is the same as the graph of the paths in the set B. Let the set S be the set of paths which is initially empty. First, we add the path P to the set S and remove one edge for each corresponding segment of P from the graph. Note that the sum of weights for the incoming edges of any internal vertex in the graph is still equal to the sum of weights for its outgoing edges because the weights of the edges along the path P are equally decreased by one. And the

sum of weights for the incoming edges of all the primary outputs in a graph is also equal to the sum of weights for the outgoing edges of all the primary inputs. Next, from Lemma 4, we construct one path from one source vertex whose sum of weights for the outgoing edges is not zero, and add that path to the set S and decrement the weights of the edges along that path by one. This procedure is repeated until the graph becomes an empty graph. From Lemma 3, we know that the number of paths in the set S is also N. Finally, we know that the sum of the numbers of paths in the sets A and S is the same as the sum of the numbers of paths in the sets A and B. And, the path P is one of the paths in the set S.

Q.E.D.

Theorem 1 : The distribution of path delays for the delay-optimized network is a delta function, if the adjustment of path delays can be done through the adjustment of either the transistor sizes or the interconnection wire lengths,

Proof : From Lemma 5, we can construct the same number of different paths from the paths currently in S_{max}. Suppose that the number of paths in S_{max} is N and the delay of a path in S_{max} is D_{max}. Then, the sum of delays for the paths in S_{max} is $N \cdot D_{max}$, which is also the sum of delays for the newly constructed N paths. Here, we know that the delay of the candidate path P is not greater than D_{max}. Suppose that the delay of path P is less than D_{max}. Then, the sum of the (N-1) paths except the candidate path P from the newly constructed paths is greater than (N-1) $\cdot$ D_{max}. This implies that there is at least one path in the newly constructed paths whose delay is greater than D_{max}. This is a contradiction to the assumption that no path delay is greater than D_{max}. Therefore, the delay of the candidate path P is D_{max}.

Thus if every segment of a path such as P is completely contained in S_{max}, the path delay of P will itself be equal to D_{max} and the path can be added to the set S_{max} without further alteration to its delay. Therefore, the set S_0 is empty and the distribution of a delay-optimized network is a delta function at D_{max}.

Q.E.D.

III. Manufacturing Yield and Delay Testing Quality in a Delay-Optimized Design

In this section, we concern with the impact of differences in path delays on the likelihood of delay faults affecting the performance of the circuit.

1. Clock Rate and Path Delay Sensitivity

The actual manufactured path delay may be different from the designed path delay. There are two additional sources of delay in the manufactured path delay. One is the delay due to the manufacturing process variations, and the other source is the delay defect size due to the random manufacturing defects. We will use the term *good-circuit path delay* to refer to designed path delay plus delay due to manufacturing process variations. Fig. 5 shows the relationship between the good-circuit path delay in a delay-optimized design, designed path delay, and delay due to the manufacturing process variations.

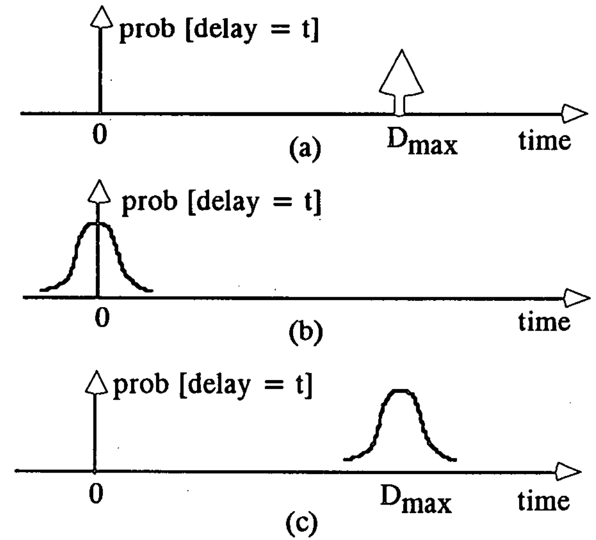


Fig. 5. Relationship between (a) designed path delay, (b) delay defect size, and (c) good-circuit path delay.

Let the random variable X and Y be defined as the good-circuit path delay in a delay-optimized design and the delay defect size, respectively. Then, the actual manufactured path delay, indicated by the random variable W, is the sum of the good-circuit path delay and the delay defect size. Since the random variables X and Y are independent, the random variable W can be determined as the sum of the random variables X and Y, i.e.,

$$W = X + Y \tag{1}$$

The density function $f_W(t)$ of the random variable W can be calculated by applying the standard convolution, that is

$$f_W(t) = f_X(t)^* f_Y(t) = \int_{-\infty}^{\infty}[f_X(s) \cdot f_Y(t-s)]ds \tag{2}$$

where $f_X(t)$ and $f_Y(t)$ are density functions of X and Y, and * is the operator for standard convolution[8]. The density function of X, $f_X(t)$, can be assumed as normal with the mean D_{max} (designed path delay) and the standard deviation σ_X[3]. The standard deviation σ_X of a path can be assumed approximately constant in large circuits[9]. The density function of Y, $f_Y(t)$, is the conditional probability that when there is a delay fault in a circuit, the delay defect size is t time units.

The system clock time, C, is usually determined by taking the longest delay path of the circuit perturbed by manufacturing process variations which has a mean μ_M (= D_{max}) and a standard

deviation σ_M such that

$$C = \mu_M + k \cdot \sigma_M, \quad (3)$$

where k is the confidence level provided by designers under the assumption that the distribution is normal[3]. Assume that we use the same system clock time for both the original design and the delay-optimized design. In a delay-optimized design, the slack of a path, which is the difference between the clock time and the path delay, will decrease. Consequently, the delay-optimized design will be more sensitive to manufacturing delay defects.

To specifically describe the above observation, we define the sensitivity of a path to a delay fault as the conditional probability that when a path contains a delay fault, a system timing failure occurs[3]. The sensitivity of the path under test can be represented as

$$S = \Pr[W > C] = \int_{t=C}^{\infty} f_W(t)dt \quad (4)$$

The sensitivity of a path to a delay fault is directly related to the good-circuit path delay. In a non-optimized network, the sensitivity of a path to a delay fault varies from zero to one. In a delay-optimized design, every path delay has the same distribution function with the same mean delay and standard deviation since the sensitivity of each path is exactly the same.

2. *Delay Test Quality in a Delay-Optimized Design*

In stuck-at fault testing, the ability to detect a fault does not depend on the path of the error indication. However, detection of delay faults depends heavily on the propagation delay of the path under test. Thus, in order to quantify this concept, we define the individual test effectiveness, E, of the path under test as the ratio between the system sensitivity of the path under test, S', and the system sensitivity of the longest delay path passing through the same delay fault site, S[3]. That is,

$$E = S' / S \quad (5)$$

When we apply the same pattern to a non-optimized design and its optimized design, the individual test effectiveness for the non-optimized design will be some value between zero and one depending the propagation delay of the path under test. However, in the delay-optimized design, if a given delay fault is tested, the individual test effectiveness is one because the sensitivities of all paths are the same in a delay-optimized design. Otherwise, its individual test effectiveness is zero,

A figure of merit for delay fault testing can be obtained by aggregation of the individual test effectivenesses over all target delay faults. The result is called statistical delay fault coverage (SDFC)[3]. SDFC is the ratio of the sum of the cumulative distributions of the delay defect size detected by testing the paths under test to those of the delay defect size detected by testing the longest paths. That is,

$$\text{SDFC} = (\sum_{i=1}^{N} S_i E_i) / \sum_{i=1}^{N} S_i = (\sum_{i=1}^{N} S'_i) / \sum_{i=1}^{N} S_i \quad (6)$$

where N is the number of target delay faults, S_i is the sensitivity of the longest path passing through the i-th delay fault site, S'_i is the sensitivity of the path under test, and E_i is the individual test effectiveness of a delay test for the i-th delay fault.

In a delay-optimized design, the statistical delay fault coverage can be determined in a similar way to the stuck-at fault coverage which is the fraction of tested faults out of the total number of faults assumed, i.e.,

$$\text{SDFC} = K / N, \quad (7)$$

where K is the number of tested faults and N is the total number of faults assumed.

The implication of Eq. (7) is that test generation algorithms targeted for "transition faults"[10] may be sufficient because any path is about as good as any other in terms of delay test quality. This type of testing is not totally for free in that an optimized network has a different effect on overal yield. The following section will deal with this aspect.

3. *Manufacturing Yield and Defect Level in a Delay-Optimized Design*

Assume that delay faults are independent of one another and that all delay faults are equally likely with probability p. Since the value p is generally low, the probability that the path under test has more than one delay defect is extremely low [9]. Thus, it can be further assumed that every path can have at most one delay defect. Under these assumptions, the probability that a system is free of system timing failures - the yield - is

$$Y = \prod_{i=1}^{N}(1 - pS_i) \quad (8)$$

where N is the number of delay faults and S_i is the sensitivity to a delay fault for the longest path passing through i-th delay fault site[3]. In a delay-optimized design, since the sensitivity is identical, the yield can be represented as

$$Y = (1 - pS)^N, \quad (9)$$

where S is the sensitivity of each path in a delay-optimized design.

Defect level (DL) is the probability that a defective system is accepted as a good one after delay testing. A relationship between the defect level, the yield, and the fault coverage has been established in delay fault testing [3]. That is, $\text{DL} = 1 - Y^{1-\text{SDFC}}$. From Eq. (7), we have the result that, in a delay-optimized design,

$$DL = 1 - Y^{1-K/N} \quad (10)$$

which is similar to the one in[11]. A detailed derivation of Eq. (10) can be found in[3]. From Eq. (10), we can determine the amount of testing to ship a product of a certain quality level.

From the theoretical background that we have discussed so far, we will provide the answers for the following two questions.

(1) How can we determine the system clock time so that the delay-optimized design will have the same manufacturing yield as the non-optimized design?

Let Y^A and Y^B be the yields of the non-optimized design and its delay-optimized equivalent, respectively. Since we desire that $Y^A = Y^B$, $Y^A = (1-pS_B)^N$, where S^B is the sensitivity of a path in the delay-optimized design, p is the fault probability, and N is the number of delay faults. Therefore, the sensitivity of any path in a delay-optimized design can be represented as

$$S_B = \frac{1-Y_A^{\frac{1}{N}}}{p} \tag{11}$$

From Eqs. (4) and (11), we have

$$S_B = \int_{t=C}^{\infty} f_W(t)dt = \frac{1-Y_A^{\frac{1}{N}}}{p} \tag{12}$$

If we know the density function $f_W(t)$ of delay defect size, the yield of a non-optimized design, the delay fault probability, and the number of possible delay faults, we can use Eq. (12) to determine the slack of the delay-optimized design and the new system clock time.

Example 1.

Assume that the density function of the longest good-circuit path delay in a circuit is a normal distribution with the mean μ_M (= D_{max}) and the standard deviation σ_M. Then, from Eq. (3), the system clock time for the non-optimized design, C_{old}, can be represented as

$$C_{old} = D_{max} + k_{old} \cdot \sigma_M, \tag{13}$$

and the system clock time for the delay-optimized design, C_{new}, can be represented as

$$C_{new} = D_{max} + k_{new} \cdot \sigma_M, \tag{14}$$

where k_{old} and k_{new} are the confidence levels for the non-optimized design and the delay-optimized equivalent, respectively. Also, assume that the density function of delay defect size is a normal distribution with the mean μ_Y and the standard deviation σ_Y. Then, the density function of the actual manufactured path delay (faulty-circuit path delay) in a delay-optimized design can be determined as

$$f_W(t) = N(D_{max} + \mu_Y, \sqrt{\sigma_M^2 + \sigma_Y^2}) \tag{15}$$

We can convert the normal variable t in Eq. (15) into a standard normal variable tn defined by

$$t_n = \frac{t - D_{max} - \mu_Y}{\sqrt{\sigma_M^2 + \sigma_Y^2}} \tag{16}$$

Then, from Eqs. (12), (14), and (16), the sensitivity of a path in the delay-optimized design can be represented as

$$S_B = 1 - \Phi\left(\frac{k_{new}\sigma_M - \mu_Y}{\sqrt{\sigma_M^2 + \sigma_Y^2}}\right) = \frac{1-Y_A^{\frac{1}{N}}}{p} \tag{17}$$

where $\Phi(.)$ is the cumulative distribution function of a standard normal distribution. From Eq. (17), the new confidence level, k_{new}, of the system clock time can be determined as

$$k_{new} = \frac{\mu_Y + \Phi^{-1}\left(1 - \frac{1-Y_A^{\frac{1}{N}}}{p}\right)\sqrt{\sigma_M^2 + \sigma_Y^2}}{\sigma_M} \tag{18}$$

where $\Phi^{-1}(.)$ is an inverse function of $\Phi(.)$. And the new system clock is determined as

$$\begin{aligned} C_{new} &= D_{max} + k_{new}\sigma_M \\ &= D_{max} + \mu_Y + \Phi^{-1}\left(1 - \frac{1-Y_A^{\frac{1}{N}}}{p}\right)\sqrt{\sigma_M^2 + \sigma_Y^2} \end{aligned} \tag{19}$$

From, Eq. (19), a system clock time will be calculated so that the delay-optimized design can have the same manufacturing yield as the non-optimized design. Fig. 6 shows the ratio of C_{new} to C_{old} for various values of the yield of the non-optimized design when k_{old} = 3, D_{max} = 400, σ_M = 20, μ_Y = 50, σ_Y = 30, N = 10^5, and p = 10^{-4}.

(2) How much more delay fault coverage is required in the delay-optimized design in order to have the same defect-level for delay faults as the non-optimized design, while

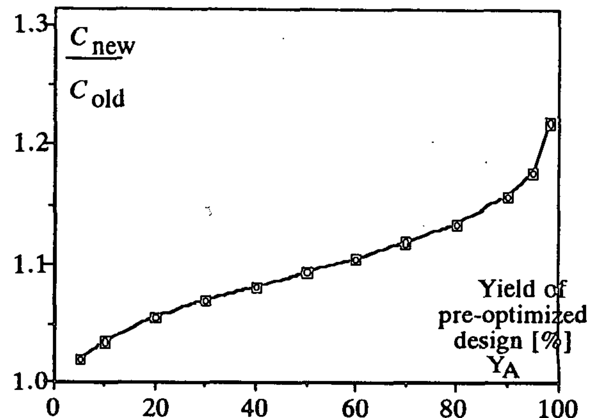


Fig. 6. The ratio of C_{new} to C_{old} for various yield of non-optimized design.

the system clock time is the same for both designs?

Since the system clock times are the same, the yield of the delay-optimized design will be smaller than that of the non-optimized design. In this case, we need more thorough testing to guarantee the same defect level for both designs.

Let DL_A and DL_B be the defect levels of the optimized design and the non-optimized design, respectively. Then, DL_A and DL_B are represented as

$$DL_A = 1 - Y_A (1-T_A) \tag{20}$$

$$DL_B = 1 - Y_B (1-T_B) \tag{21}$$

where Y_A, Y_B are the yields and T_A, T_B are the fault coverages of a non-optimized design and its delay-optimized design, respectively. For $DL_A = DL_B$, the required fault coverage T_B is determined as

$$\begin{aligned} T_B &= 1-(1-T_A)^{\frac{\ln Y_A}{\ln Y_B}} \\ &= \frac{\ln Y_A}{\ln Y_B} T_A + \left(1 - \frac{\ln Y_A}{\ln Y_B}\right) \end{aligned} \tag{22}$$

From Eq. (22), the required delay fault coverage will be determined so that the delay-optimized design can have the same defect-level for delay faults as the non-optimized design, while the system clock time is the same for both designs. Fig. 7 shows the above relationship between T_A and T_B, when we know the yields of both the non-optimized design and the optimized design.

IV. Implications and Conclusions

Path delay timing optimization has both positive and negative implications with respect to performance and test quality. This form of optimization will make more efficient use of silicon resources so that optimal performance is obtained for some set of hardware resources (such as silicon area). In this paper, the impact of delay optimized logic circuits on delay fault testing has been discussed and compared to the case for non-optimized designs. First, we have provided a timing optimization procedure and showed that the resultant density function of path delays is a delta function. Next we also discussed the impact of timing optimization on the yield of a manufacturing process and the defect level for delay faults. Finally, we have given the analytical results (Eq. (19) and (22)) on the determination of the system clock time so that the delay-optimized design will have the same manufacturing yield as the non-optimized design and on the determination of delay fault coverage in the delay-optimized design in order to have the same defect-level for delay faults as the non-optimized design, while the system clock time is the same for both designs.

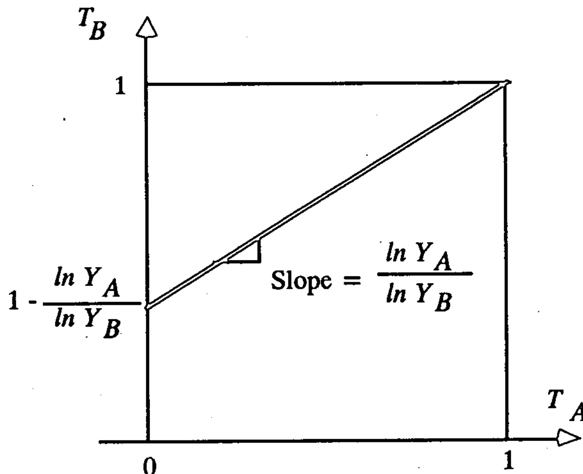


Fig. 7. Fault coverage T_B for a delay-optimized design as a function of T_A, Y_A, and Y_B.

Cautions must be exercised in the selection of the operating system clock rate because a clock rate too close to Dmax will result in unacceptably low yields. Similarly, the amount of effective delay fault coverage must, in general, also be increased in order to obtain the same defect level. While this might seem to be a negative development, there are positive results as well. As path delays become more equal and timing slacks decrease, detection of delay defects will become less dependent on the path along which the testing is done. Therefore, automatic test pattern generation algorithms for transition faults [10] may be sufficient, and more complex and computationally expensive algorithms targeted for small delay faults [12], [13] may not be required. Because any path is about as good as any other path, random or pseudo-random testing techniques in built-in self test (BIST) will be more effective in delay-optimized designs than those in non-optimized designs, which is a very positive result. However, this BIST approach will be even more efficient only when we are able to design a logic circuit structure that is highly random-pattern testable. Therefore, it is clear that in the future the strong interaction between synthesis and testing will work together to further improve designer productivity and product quality.

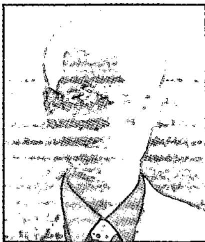
Acknowledgement

This work was supported in part by KOSEF under the contract 961-0919-097-2.

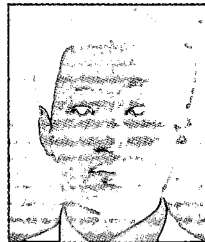
References

[1] J. Darringer, D. Brand, J. Gerbi, W. Joyner, and L. Trevilyan, "LSS: A System for Production Logic Synthesis," IBM Journal of Research and Development, Vol. 28, No. 5, pp. 326-328, 1984.

[2] T. W. Williams, B. Underwood, and M. R Mercer, "The Interdependence between Delay Optimization of Synthesized Networks and Testing," ACM/IEEE Proc. 28th Design Automation Conf., pp. 87-92, 1991.
[3] E. S. Park, M. R Mercer, and T. W. Williams, "The Total Delay Fault Model and Statistical Delay Fault Coverage," IEEE Trans. on Comput., Vol. C-41, No. 6, pp. 688-698, June 1992.
[4] D. M. Wu, C. E. Radke, and J. P. Roth, "Statistical AC Test Coverage", IEEE Proc. Int. Test Conf., pp. 538-541, 1986.
[5] J. P. Fishburn and A. E. Dunlop, "TILOS: A Posynomial Programming Approach to Transistor Sizing," IEEE Proc. Int. Conf. on CAD, pp. 326-328, 1985.
[6] J. J. Cong and K. S. Leung, Optimal Wiresizing Under Elmore Delay Model, IEEE Trans. On VAD, Vol. 14, No. 3, pp. 321-336, 1995.
[7] N. Menezes, S. Pullela, and L. T. Pileggi, Simultaneous Gate and Interconnect Sizing for Circuit-Level Delay Optimization, ACM/IEEE Proc. 32nd Design Automation Conference, pp. 690-695, 1995.
[8] A. Papoulis, Probability, Random Variables, and Stochastic Processes, McGraw-Hill Book Co., Inc., 1965.
[9] N. N. Tendolkar, "Analysis of Timing Failures due to Random AC Defects in VLSI Modules," ACM/IEEE Proc. 22nd Design Automation Conf., pp. 709-714, 1985.
[10] J. A. Waicukauski, E. Lindbloom, B. K. Rosen, and V. S. Iyengar, "Transition Fault Simulation", IEEE Design and Test, pp. 32-38, 1987.
[11] T. W. Williams and N. C. Brown, "Defect Level as a Function of Fault Coverage," IEEE Trans. on Comput., Vol. C-32, No. 12, pp. 987-988, 1981.
[12] E. S. Park and M. R. Mercer, "An Efficient Delay Test Generation System for Combinational Logic Circuits," IEEE Trans. on Computer-Aided Design, Vol. 11, No. 7, pp. 926-938, July 1992.
[13] B. K. Rosen, V. S. Iyengar, and I. Spillinger, "Delay Test Generation 2 -- Algebra and Algorithms," IEEE Proc. Int. Test Conf., pp. 867-874, 1988.



Young-Ho Park was born in Jumunjin, Korea, in 1956. He received the B.S. degree in computer science from Taejon Industrial University in 1986. Since 1983, he has been working for ETRI in the areas of hardware design environment development especially for TDX and ATM electronic switching system as a senior member of technical staff. His research interests include CAD, MCM design, and computer network.



Eun-Sei Park was born in Seoul, Korea, in 1957. He received the B.S. degree in electrical engineering from Seoul National University in 1980, the M.S degree in electrical and electronic engineering from the Korea Advanced Institute of Science and Technology in 1982, and the Ph.D degree in electrical and computer engineering from the University of Texas at Austin, Texas, USA, in 1989. Currently, he is an assistant professor in the department of electronic engineering at the Hanyang University. Previously, he worked for ETRI, Taejon, Korea, as a principal member of research staff, and for Motorola, Austin, Texas, as a research staff. His research interests include VLSI testing, CAD, computer architecture, and high speed network computing.