On - Line Path Delay Fault Testing of Omega MINs

M. Bellos[1], E. Kalligeros[1], D. Nikolos[1,2] & H. T. Vergos[1,2]

[1]*Dept. of Computer Engineering and Informatics*
University of Patras, 26 500, Rio, Greece

[2]*Computer Technology Institute*
3, Kolokotroni Str., 262 61 Patras, Greece

e-mail : {tarka, kalliger}@ceid.upatras.gr, {nikolosd, vergos}@cti.gr

Abstract

This paper studies path delay fault testing of Omega Multistage Interconnection Networks (MINs). Taking advantage of the MIN's parallelism, we show that although the number of physical paths is $O(n^2)$, all these paths can be tested optimally for path delay faults applying O(n) test vector pairs. Having derived the test set, we also show how the test set application as well as the response verification can be done either by a dedicated low – cost tester or on-line using the system resources.

I. Introduction

Many kinds of MINs have been proposed and built for use in massively parallel computers [1, 2]. In this paper we consider Omega MINs with centralized control.

Testing of MINs has been widely considered with respect to the state stuck-at fault model, the link fault model and the switch fault model, for example [3, 4]. However, physical defects in integrated circuits can degrade circuit performance without altering their logic functionality. Apart from this, increasing performance requirements of the contemporary VLSI circuits makes it difficult to design them with large timing margins. Thus imprecise delay modeling and the statistical variations of the parameters during the manufacturing process may result in circuits with longer than the expected delays. The change in the timing behavior of the circuit in this paper is modeled according to the path delay fault model under which a path is declared faulty if it fails to propagate a transition from the path input to the path output within a specified time interval [5]. The path delay fault model, since it captures the cumulative effect of small delay variations in gates along a path as well as the faults caused by a single gate, is deemed to be the most general among all delay fault models.

Two major problems discourage the use of the path delay fault model. The first is that only a very small subset of the paths of a design can be robustly tested for path delay faults (under either the stringent Single Path Propagating Hazard Free or the more relaxed Validatable Robustly Testable limitations). In this paper since we consider that each switch of the MIN is implemented by multiplexers, a robust two pattern test for each path can be derived [6]. The second problem is that the number of physical paths in a contemporary IC is excessively large for testing every path. Usually, a subset of the physical paths is selected for path delay fault testing based either on their propagation delay or on functional approaches [7]. However the number of paths selected by both methods is so large in general that all the selected paths cannot be tested, especially in performance optimized circuits [8]. This is especially true in the case of Omega MINs where most of the physical paths are of critical delay. To this end, by taking advantage of the parallelism of the MIN, we show that although the number of physical paths is $O(n^2)$, all these paths can be tested optimally for path delay faults applying O(n) test vector pairs.

In this paper we consider nxn Omega MINs, with $n=2^k$, implemented as a set of b/M M-bit slices [9], where b is the size of the bus of each source and destination connected by the network, $1 \leq M \leq b$ and each slice has been implemented as a VLSI chip. For M = b the network has been implemented on a single chip (probably on a single wafer). In this paper, apart from deriving an optimal test set, we will also present : a) a low-cost tester for production testing of the MIN and b) the way that this test set can be applied to the MIN and how its responses are verified on-line.

II. Preliminaries

A physical path of a circuit is an alternating sequence of gates and lines leading from a primary input to a primary output of the circuit. In delay fault test generation we associate two logical paths with each physical path. A logical path is a pair (T, p) with $T = \overline{x} \rightarrow x$ and $x \in \{0, 1\}$, being a transition at the input of p. In the case of delay fault testing the test set consists of pairs of vectors. The cardinality of the test set, that is, the number of pairs of vectors depends on the number of the paths that must be tested and the percentage of the paths that can be tested in parallel. Throughout the paper the term test session is used to denote the application of a test vector pair.

An Omega MIN is constructed from $N=\log_2 n$ stages of switches, where each of the stages has (n/2) 2x2 switches. The switch stages are labeled from 1 to N. We consider that the source and destination nodes constitute the stages 0 and N+1, respectively. The interconnection pattern between adjacent stages is the perfect shuffle permutation [10]. This holds for all pairs of stages except N and N+1. Figure 1 shows an 8x8 Omega MIN. A conflict of the MIN appears when any two sources are trying to set a switch of

the network in complementary states.

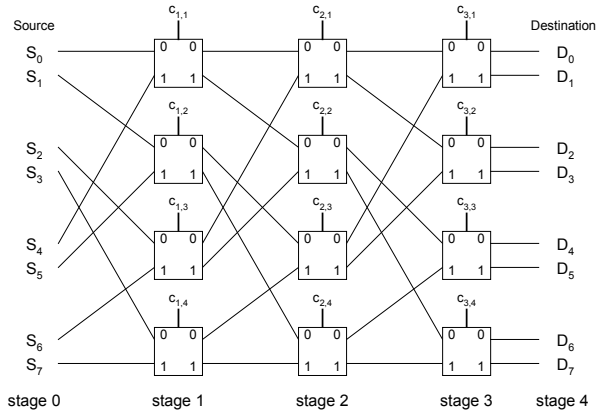

Figure 1. 8x8 Omega Network

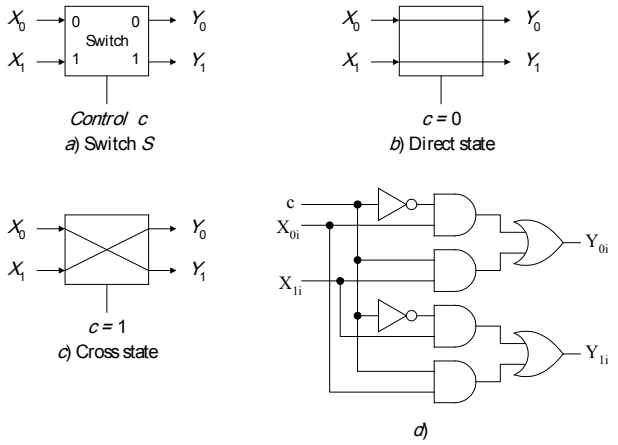

Figure 2.

Each switch *S*, (Figure 2.*a*) has a pair of input data buses X_0, X_1, a pair of output data buses Y_0, Y_1 and a control signal *c*. All four buses are unidirectional and identical in size. The two states of the switch *S* are determined by the control line *c* as follows: the *direct* state (Figure 2.*b*), where the values of X_0, X_1 are propagated to Y_0, Y_1 respectively, and a *cross* state (Figure 2.*c*), where the values of X_0, X_1 are propagated to Y_1, Y_0 respectively. The upper input and output are labeled with 0 while the lower are labeled with 1. Each switch is constructed from 2M 2->1 multiplexers, where M is the size of the buses. Each pair of multiplexers, (Figure 2.*d*), accepts two lines of X_0, X_1 buses, the control signal *c* and drives the corresponding lines of buses Y_0, Y_1.

Without loss of generality we consider that each link is a single virtual line, that may actually represent either one physical line or a physical bus. For testing purposes any value of the virtual line is always applied to every line of the physical bus that it may represent. Paths along the MIN are formed by concatenation of subpaths along links and subpaths through switches of the MIN as well as by subpaths sourcing from the control signals. We will hereafter present the analysis based on virtual lines (or simply lines). The analysis will be valid for all M lines of the bus. We note that to every virtual path or line correspond two logical virtual paths.

In an nxn Omega MIN we distinguish the paths in two sets: those not including subpaths sourcing from a control input and those which do. Since the connections of sources to destinations change dynamically during system operation, delays that stem from the control signals are also significant. Let P be the set consisting of all virtual paths starting from any source and ending at any destination. Since there are n sources and there is only one path from one of them to all the destinations, the number of all possible virtual paths is n^2. That is the cardinality of P, denoted $|P|$, is $|P| = n^2$.

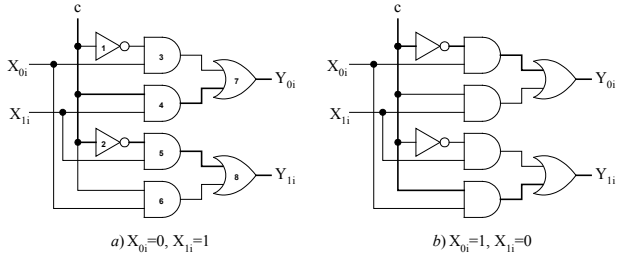

Figure 3.

Let L be the set consisting of all paths starting from the control input of a switch and ending at any destination. Figures 3.*a* and 3.*b* present the subpaths from the control input of a switch through two of its 2M multiplexers for X_{0i}=0 and X_{1i}=1 and for X_{0i}=1 and X_{1i}=0. Since the transition, during delay testing, propagates from the control input through 2M multiplexers and along the lines of the bus, in this case we refer also to virtual paths. For computing the number of the virtual paths starting from a control input we observe that at every stage the control input of a switch can be seen as the root of two full binary trees having the destinations as leaves. Each such tree has a depth of N-*i*+1, where *i* is the number of the stage and $i \in \{1, 2, ..., N\}$. The latter means that every such tree has 2^{N-i+1} leaves which is also the number of virtual paths. Since each stage has n/2 switches and each switch is the root of two trees, there are $2*(n/2)*2^{N-i+1}$ virtual paths starting from each stage's control inputs. Thus the cardinality of L is equal to the sum of the virtual paths starting from every control input, which is:

$$|L| = \sum_{i=1}^{N} 2\frac{n}{2}2^{N-i+1} = 2\frac{n}{2}2(2^N-1) = 2n(n-1)$$

Therefore the total number of virtual paths is equal to: $|P| + |L| = 3n^2 - 2n$ and the number of all logical virtual paths is: $|LV| = 2(3n^2 - 2n)$. The number of physical lines is equal to $M(3n^2 - 2n)$, however the propagation delays along the M lines of the bus are measured in parallel. Note that every physical path is sensitizable.

Although the number of the virtual paths of an nxn Omega network is $O(n^2)$ we will show in the next section

that due to the inherent parallelism of the network testing along various paths can be done in parallel.

III. Parallel path delay fault testing

It is well known that in an Omega network when the control inputs get a set of values each source is connected to a different destination. Throughout this section we consider that all control inputs of stage i, for i=1, 2, ..., N, of the Omega network take the same value c_i. Changing the value of the control inputs of a stage i the switches of this stage alternate between the two states of Figures 2.b and 2.c, and all paths from sources to destinations change. Then taking into account that in the Omega network only one path exists from a specific source to a specific destination we conclude that for the two sets of values $c_1c_2...c_N$ and of $c_1'c_2'...c_N'$, with $c_1c_2...c_N \neq c_1'c_2'...c_N'$, a common path from the source to destination in both configurations of the network does not exist. Therefore, applying to $c_1c_2...c_N$ all possible values, that is 2^N different values, we ensure that each source has been connected to each destination. A feature of the Omega network is that every source can be connected to every destination. Since we have n destinations we conclude that at least n value combinations of the control signals are necessary so that each source gets connected to each destination. From the above discussion we conclude that 2^N is the least number of configurations that ensures that all possible paths from sources to destinations have been established. For each configuration, that is a value of $c_1c_2...c_N$, the delays along n virtual paths can be measured, hence $2*2^N$ test sessions are required in order to measure the delays along all logical virtual paths from sources to destinations. Let T_p denote the number of test sessions required to measure the delays along all paths from sources to destinations. Then :

$$T_p = 2*2^N = 2n \quad (1)$$

At every stage i a control input can be seen as a root of two full binary trees having the destinations as leaves. Every such tree has 2^{N-i+1} virtual paths from the root to the leaves, as discussed earlier. For any value of $c_{i+1}c_{i+2}...c_N$ two paths starting from the control input of each switch of the stage i get established, which means that 2*n/2 virtual paths starting from the control inputs can be tested in parallel. Since two trees each one with 2^{N-i+1} virtual paths correspond to each switch of the stage i, we conclude that $2 * 2 * 2^{N-i+1} / 2$ test sessions are required for testing the logical virtual paths starting from a control input of stage i. Taking into account that i=1, 2, ..., N we get that the total number of test sessions T_L for measuring the propagation delays along the virtual paths starting from control inputs are:

$$T_L = 2 * \sum_{i=1}^{N} 2^{N-i+1} = 4(2^N-1) = 4(n-1) \quad (2)$$

From relations (1) and (2) we get: $T_p + T_L = 2(3n-2)$.

Therefore, while the number of virtual paths of a nxn Omega network is $O(n^2)$ we have shown that the number of the required test sessions is O(n). In a circuit with n outputs the maximum number of paths that can be tested in parallel is equal to n. Then taking into account that the number of all logical virtual paths of the nxn Omega network is equal to $2(3n^2-2n)$ and the fact that 2(3n-2) test sessions are required we conclude that this is the optimal number of test sessions required to test all possible paths.

Table 1. Test vectors for the P paths of the 8x8 Omega network.

	$S_0S_1S_2S_3S_4S_5S_6S_7$	$c_1c_2c_3$	$S_0S_1S_2S_3S_4S_5S_6S_7$	$c_1c_2c_3$
P	T T T T T T T T*	0 0 0	T T T T T T T T	1 0 0
	T T T T T T T T	0 0 1	T T T T T T T T	1 0 1
	T T T T T T T T	0 1 0	T T T T T T T T	1 1 0
	T T T T T T T T	0 1 1	T T T T T T T T	1 1 1

*T denotes a 0->1 and a 1->0 transition.

Table 2. Test set for the L paths of the 8x8 Omega network.

	$S_0S_1S_2S_3S_4S_5S_6S_7$	$c_1c_2c_3$	$S_0S_1S_2S_3S_4S_5S_6S_7$	$c_1c_2c_3$
L	0 1 0 1 0 1 0 1	0 0 T	1 0 1 0 1 0 1 0	0 0 T
	0 0 1 1 0 0 1 1	0 T 0	1 1 0 0 1 1 0 0	0 T 0
	0 0 1 1 0 0 1 1	0 T 1	1 1 0 0 1 1 0 0	0 T 1
	0 0 0 0 1 1 1 1	T 0 0	1 1 1 1 0 0 0 0	T 0 0
	0 0 0 0 1 1 1 1	T 0 1	1 1 1 1 0 0 0 0	T 0 1
	0 0 0 0 1 1 1 1	T 1 0	1 1 1 1 0 0 0 0	T 1 0
	0 0 0 0 1 1 1 1	T 1 1	1 1 1 1 0 0 0 0	T 1 1

Table 1 presents a test set suitable for parallel path delay fault testing of P paths for the MIN of Figure 1. As can been seen from this table, for testing the paths of P, we need to apply both 0->1 and 1->0 transitions at every source for every value of the control inputs. For the paths of set L, we need to apply complementary vectors at the sources, for every transition required at the control inputs. There exist many distinct sets of test vector pairs that can be used for path delay testing of the paths belonging to L (such a possible test set is presented in Table 2). From Table 2 we can easily see that the number of distinct test vectors that must be applied to the source inputs of the network is six. For large networks this number can be very large. Reducing the number of distinct source inputs, during the testing of L paths simplifies both the on-line testing of the network using resources of the system as well as the application of the test set during production testing.

To this end, we introduce a test set consisting only of two complementary vectors for the source inputs no matter what the size of the MIN is. The derivation of one of the vectors of the proposed test set can be done according to the following rules:

Rule 1. For the 4x4 Omega MIN the test vector is 0 1 1 0.

Rule 2. Recursively calculate the test vector v of the nxn Omega MIN from the test vector u of the (n/2) x (n/2) MIN by using the following : $v_i = u_i$ and $v_{i+(n/2)} = \overline{u_i}$, for i = 0, 1, 2, …, (n/2) – 1.

Table 3 presents the outcome of the application of these rules to an 8x8 MIN.

By applying the proposed test vectors at the inputs of the MIN we assure that each stage of switches receives at

its inputs, ordered from top to bottom, either the applied test vector or its complementary (the proof is omitted due to space limitations).

Table 3. Test set for the L paths of the 8x8 Omega network.

	$S_0S_1S_2S_3S_4S_5S_6S_7$	$c_1c_2c_3$	$S_0S_1S_2S_3S_4S_5S_6S_7$	$c_1c_2c_3$
L	0 1 1 0 1 0 0 1	0 1 T	1 0 0 1 0 1 1 0	0 1 T
	0 1 1 0 1 0 0 1	1 T 0	1 0 0 1 0 1 1 0	1 T 0
	0 1 1 0 1 0 0 1	1 T 1	1 0 0 1 0 1 1 0	1 T 1
	0 1 1 0 1 0 0 1	T 0 0	1 0 0 1 0 1 1 0	T 0 0
	0 1 1 0 1 0 0 1	T 0 1	1 0 0 1 0 1 1 0	T 0 1
	0 1 1 0 1 0 0 1	T 1 0	1 0 0 1 0 1 1 0	T 1 0
	0 1 1 0 1 0 0 1	T 1 1	1 0 0 1 0 1 1 0	T 1 1

IV. Low-cost Tester.

For production testing of the MIN we introduce a low-cost tester. This will be comprised of : a) an up-down $\log_2 n$ bit Gray counter, b) n*M T flip-flops, c) 2*n*M D flip-flops, d) a 2*n*M input 2-rail checker tree and e) some additional logic for controlling the components of the tester. The tester is fed with a clock frequency f.

The block diagram of the tester is depicted in Figure 4. The outputs of the Gray counter drive the control signals of the MIN. Each MIN source is driven by one of the T flip-flops. The T flip-flops are used in order to produce one of the two complementary test vectors presented in Section III. Each output of the MIN drives two D flip-flops as shown in Figure 5. The flip-flop outputs are driven to a tree of 2-rail checkers. When the MIN operates correctly at speed, the outputs of this checker are at complementary values.

For testing the paths of set P, the Gray counter and the D flip-flops are reset. The T flip-flops are driven by f, while the Gray counter is driven by a clock with frequency f/5. We assume that each new cycle begins with every rising edge of f. During the 5 cycles that the Gray counter is stable the following take place (see Figure 6) : the first cycle is used to set the MIN to a new configuration. At the next cycle the first test vector is applied to the inputs of the MIN. The complementary and the original test vector are applied at the start of the third and the fourth cycles respectively and the responses of the MIN are captured by the D flip-flops. The fifth cycle is utilized for capturing the response of the MIN to the test vector applied during the fourth cycle. The 2-rail checker is enabled at the middle of the fourth and the fifth cycles. The procedure is repeated until the Gray counter passes from all its states counting upwards.

For testing the paths of set L, the first of the complementary vectors is applied to the MIN inputs. The Gray counter changes values every two cycles (suppose first and second cycle). The counter is permitted to go through all of its states in both up and down direction. Then the second vector is applied and the counting procedure is repeated. The responses are captured at the start of each second cycle and verified at the middle of it by the 2-rail checker as shown in Figure 7.

The above described low-cost tester was designed and its correct function was verified by simulation.

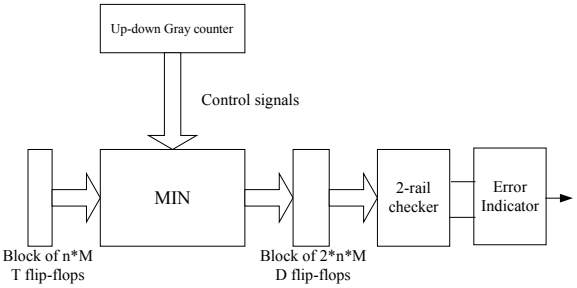

Figure 4. The dedicated tester

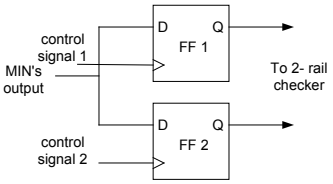

Figure 5. D flip-flops at MIN outputs

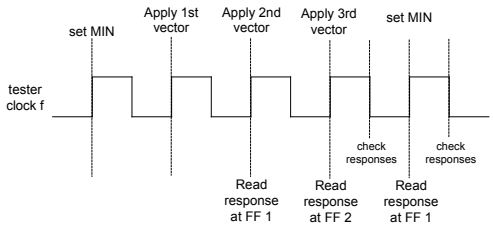

Figure 6. Test scheme for testing the paths of set P

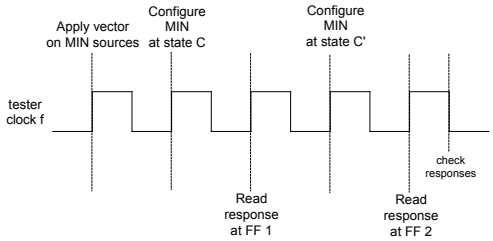

Figure 7. Test scheme for testing the paths of set L

V. On line testing.

We consider that the MIN is used for interconnecting processors that are synchronized by a common clock. The communication among the processors is established after requests towards the centralized control. Once the centralized control configures the MIN, the processors can exchange data. If the centralized control circuit can not establish a requested configuration (for example when two sources request communication with the same destination) it will use some priority scheme for serving some of the requests and it will signal inability to the rest sources. During a data transfer all the control signals are stable (i.e. the MIN remains at a fixed configuration). We further assume that the configuration is stable when there are no

new requests. The processors have different input and output buffered ports.

We assume that the communication scheme is performed by:

Step 1. The start of the communication is indicated by a rising edge of the clock. At that moment each processor places its data on the output buffer and sends a request to the centralized control.

Step 2. At the next rising edge the centralized control sets the MIN to the appropriate configuration and the processors are notified if they should proceed with transmission or reception of data.

Step 3. At each subsequent rising edge of the clock, each processor can send or receive a word or both, by enabling its corresponding buffers.

We assume that all buffers are of equal size and a communication session terminates when all buffers are empty.

Under these assumptions the application of the derived test set and the verification of the responses can be done in system environment as follows :

a) For the pairs of vectors that test the paths of set P each source's transmission buffer is filled with the test vectors that will activate the two desired transitions.
b) The processors send requests that will not cause any conflict in the MIN. We will present a procedure that ensures conflict free configurations of the MIN below.
c) The paths are considered fault free when the transition is identified one clock period after the data are placed on the inputs of the MIN. This check is performed by verifying the contents of the input buffer of each processor after the end of a communication session.
d) In order to test the paths of set L we start a communication session which will transmit one of the test vectors mentioned above over the MIN. This communication session will set the control signals of the MIN to a known state which will be stable until the next requests issued by the processors.
e) At the next communication session the requests issued by the processors cause the required transition of the control signals of the considered stage. The transition is observed at the outputs of the MIN at the rising edge of the clock one clock period after the MIN is set. Again the path is declared fault free after the verification of the contents of each processors' input buffer.

For achieving conflict free configurations of the MIN, all sources simulate an up-down Gray counter. The outputs of the simulated counters are used by each source for computing the address of the desired destination in order to issue a request to the centralized control unit. All the simulated counters must be reset and advanced to a new value simultaneously. Every time the simulated counter produces a new value, each source executes a procedure with inputs the new value of the counter and the source's label. Such a procedure based on the perfect shuffle permutation is presented below. Let src, dest denote the binary representations of the labels of the source and the targeted destination processors respectively. Let c_i be the i-th bit of the simulated Gray counter The proposed procedure is composed of the following steps :

Step 1 : Set dest = src.
Step 2 : Set i = 1.
Step 3 : Perform the perfect shuffle permutation on dest (binary rotation).
Step 4 : If $c_i = 0$ go to Step 6.
Step 5 : If the LSB of dest is 0 then set dest = dest + 1 else set dest = dest - 1.
Step 6 : Set i = i + 1.
Step 7 : If $i \leq \log_2 n$ then go to Step 3 else end.

VI. Conclusions

In this work we studied path delay fault testing of Omega MINs. The contributions of this work are :

a) We have shown, by taking advantage of the MIN parallelism, that the $O(n^2)$ physical paths of the MIN can be tested optimally by O(n) test vector pairs.
b) We have derived an optimal test set for parallel path delay fault testing of the MIN.
c) We have shown how the application of the test set as well as the verification of the MIN can be done either off-line by a dedicated low – cost tester or on-line using the system resources.

References

[1] H. J. Siegel, "Interconnection Networks for Large - Scale Parallel Processing", 2nd ed., McGraw-Hill, 1990.
[2] T. Feng, "A Survey of Interconnection Networks", Computer, December 1981, pp. 12 - 27.
[3] D. P. Agrawal, " Testing and Fault Tolerance of Multistage Interconnection Networks", Computer, April 1982, pp. 41 - 53.
[4] V. P. Kumar and S. M. Reddy, "Augmented Shuffle-Exchange Multistage Interconnection Networks". Computer, June 1987, pp. 30 - 40.
[5] G. L. Smith, "Model for Delay Faults Based upon Paths", Proc. of ITC-85, pp. 342 - 349.
[6] P. Ashar, et. al., "Path-delay-fault testability properties of multiplexer - based networks", Integration, The VLSI Journal, vol.15, (no.1), July 1993. pp. 1-23
[7] S. Tani, et. al, "Efficient Path Selection for Delay Testing Based on Partial Path Evaluation", Proc. of VTS '98, pp. 188 – 193.
[8] T. W. Williams et. al., "The Interdependence between Delay-Optimization of Synthesized Networks and Testing", Proc. of 28th DAC, pp. 119-125, ACM / IEEE 1995.
[9] M. A. Franklin, et. al, "Pin Limitations and Partitioning of VLSI Interconnection Networks", IEEE Trans. on Comp., C-31, November 1982, pp. 1109 - 1116.
[10] H. S. Stone, "Parallel Processing with the Perfect Shuffle", IEEE Trans. on Comp., C-20, pp. 153 - 161, 1971.