Novel NoC Mapping Scheme Optimized for Testing Time

Ying Zhang, Ning Wu, Fen Ge and Xin Chen

***Abstract*—NoC architecture has been increasingly applied to complex SoC chips and there is an important topic urgently needed to study, which is how to map the specific application to the NoC infrastructure efficiently. On the other side, embedded IP cores testing of NoC is also facing many challenges. This paper proposes a novel NoC mapping algorithm optimized for NoC IP cores testing. According to the pre-designed test structure, firstly adapt the Partition Algorithm to arrange IP cores into parallel testing groups and so that the testing time will be minimized. Then, accomplish the NoC mapping based on genetic algorithm with the traffic information between IP cores. The experiment results on ITC'02 benchmark circuits showed that the testing time can be reduced by 12.67% on average and the mapping costs decreased by 24.5% on average compared with the random mapping.**

***Index Terms*—mapping, testing optimization, NoC, genetic algorithm**

I. Introduction

With the increasing development of the semiconductor technology and chip integration, NoC (Network on Chip) becomes an important solution to complex SoC (System on Chip) architecture[1]. NoC mapping is how to assign a specific application on the NoC platform in accordance with certain rules, which will greatly influence the system performance. Mapping is one of the hot issues in NoC research [2]. The existing mapping schemes are mainly based on various optimization algorithms to minimize the energy consumption or traffic [3-4], which only consider functional requirements. However, the difficulty of complex system chip testing increases dramatically [5], which brings the requirement of considering test optimization while mapping.

Embedded IP core testing for SoC needs test data transmission channel (TAM, Test Access Machine). NoC-based SoCs often reuse NoC communication architecture as TAM. The ATE (Automatic Test Equipment) can simultaneously access the NUT (NoC Under Test) through multiple ports to achieve parallel testing.

Manuscript received June 11, 2014; revised August 05, 2014. This work was supported in part by the National Natural Science Foundation of China (No. 61376025, 61106018 and 61106029), the Defense Industrial Technology Development Program under contract (No. B2520110008), Jiangsu Province Key Technology R&D Program (BY2013003-11).

Ying Zhang is with the College of Information Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China (e-mail: tracy403@nuaa.edu.cn).

Ning Wu, Fen Ge and Xin Chen are with the same place of Ying Zhang. (e-mail:wunee@nuaa.edu.cn,gefen@nuaa.edu.cn, xin_chen@nuaa.edu.cn).

There are some research [6,7] discussing on the multiple access ATE schemes to optimize test time along with different schedule plans. Authors of [6] proposed a DFT scheme reusing the NoC as TAM and described the associated testing structure. Although it focuses on the design of ATE interface and test wrapper, the multi-ATE input/output and test scheduling are involved. The diagram of NoC partition and three accessible ATEs is provided and the aim of test partition is minimizing the testing time. However, it is worth noting that the partition is only applied to manufactured chips. M. Agrawal *et al.* [7] proposed the test data delivery optimization for NoC. It co-optimizes number and position of the access points and the assignment of cores to access points. Testing time minimization for NOCs is modeled as an NOC partitioning problem and solved with some optimized algorithm. And yet the scheme is also only concerned with the manufactured chips. Our proposed solution is to take into account the needs of the parallel testing with test strategies on NoC mapping process. According to the optimized test access structure, partition IP cores to testing groups in advance, which effectively improves the efficiency of SoC embedded IP cores testing.

We propose a novel NoC mapping scheme which is optimized for testability. Firstly, uses P_A algorithm to assign IP cores into groups to minimize testing time; then maps the specific application to NoC structure; finally optimizes the NoC mapping based on genetic algorithm in order to minimize mapping overhead.

Hereafter, Section II gives a brief introduction of the NUT and the associated problem. Section III presents the NoC mapping scheme with the optimization of embedded IP cores testing. Section IV explains the results of testing time and mapping performance evaluation and compares with other schemes. Section V draws the conclusions.

II. NoC Under Test and the Problem Description

NoCs can be defined as a set of structured routers and point-to-point channels interconnecting IP cores (Resources). The topology of NoC can be represented as an undirected connected graph $G(N,L)$, where $N=\{n_1,n_2,\ldots n_i\}$ is the set of nodes and $L=\{l_1,l_2,\ \ldots l_j\}$ is the set of links in the corresponding network. The widely applied topologies are Mesh, Torus, Ring, Hypercube, and Fat-tree [8]. For regularity of its structure, Mesh network are easy to implement and have good scalability. Each node in Mesh network is connected to neighbors through regular grid point-to-point links. Fig. 1 shows a 4×4 2D-Mesh network.

ISBN: 978-988-19252-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

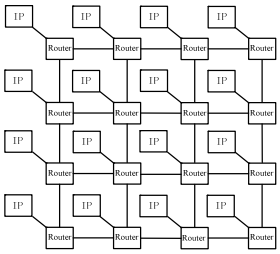

Fig. 1. 4×4 Mesh network

Assuming the core structure and IP cores traffic information are provided, each IP core is allocated to corresponding NoC resources node. This process is called NoC architecture mapping for specific applications. The 2D NoC mapping process is shown in Fig. 2. The mapping performance can be evaluated by the objective function.

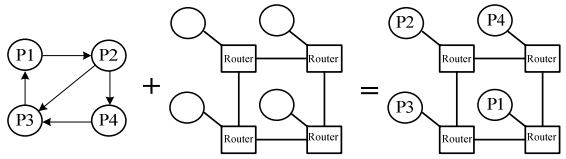

Fig. 2. 2D Mesh NoC mapping process

The description of 2D Mesh NoC mapping problem needs to provide three definitions as follows firstly.

Definition 1: Given directed acyclic weighted graph $G(V,E)$ as application characteristics chart, each vertex $v_i \in V$ shows the associated IP core, directed arc $e_{i,j} \in E$ shows the communication relationship between v_i and v_j, coefficient $\omega_{i,j}$ shows the traffic between v_i and v_j.

Definition 2: Given directed graph $P(R,P)$ as NoC structure feature graph, each vertex $r_i \in R$ shows resource node, directed arc $p_{i,j} \in P$ shows the router between v_i and v_j, $E(p_{i,j})$ shows the power consumption of one bit transfer between v_i and v_j.

Definition 3: power consumption functions are defined as (1) and (2).

$$Energy = \sum_{\forall e_{i,j}} \omega_{i,j} \times E(p_{i,j}) \tag{1}$$

$$E(p_{i,j}) = n_{router} \times E_{Sbit} + (n_{router} - 1) \times E_{Lbit} \tag{2}$$

n_{router} is the number of routers, E_{Sbit} and E_{Lbit} are separately power consumption of routers and interconnections. Considering the 2D Mesh architecture and XY routing algorithm, $E(p_{i,j})$ is actually determined by the hamming distance between v_i and v_j, so that it can be induced as (3).

$$Cost = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \omega_{i,j} * distance_{i,j} \tag{3}$$

Among them, *Cost* is the mapping overhead, $\omega_{i,j}$ is the traffic between v_i and v_j, $distance_{i,j}$ is the hamming distance between v_i and v_j, and the NoC is N×M 2D Mesh architecture.

NoC mapping problem: given *G* and *P*, search for mapping function $map()$, for minimizing the *Cost* and satisfying the following constraints.

$$\forall v_i \in V \Rightarrow map(v_i) \in R \tag{4}$$

$$\forall v_i \neq v_j \Leftrightarrow map(v_i) \neq map(v_j) \tag{5}$$

$$size(G) \leq size(P) \tag{6}$$

When applying parallel testing on NoC-based SoC, ATE can access the NUT by various ports. Ref.[9] provided performance evaluation of several ATE access modes and proposed the best solution as shown in Fig. 3. The ATEs access from the East, South, West, and North corners of Mesh. So that four sub-blocks testing can be executed simultaneously. The test architecture will significantly optimize the testing time and has relatively low testing cost.

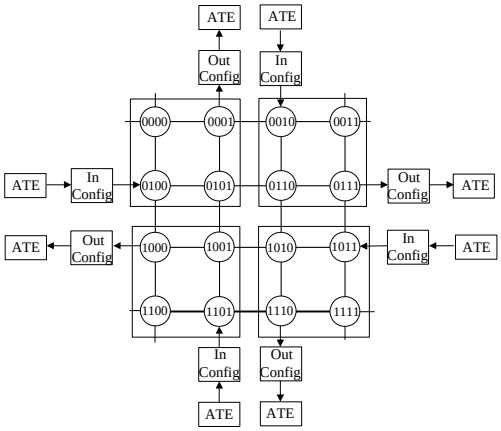

Fig. 3. Optimized ATE access mode

According to the test access mode, IP cores of NoC need to be pre-grouped and then mapped to a 2D Mesh architecture. By that means, the efficiency of NoC embedded IP core testing will be significantly improved. At the same time, NoC IP cores testing optimization has converted into the IP cores partition problems. It can be summarized as follows.

Given the set of IP cores $C = \{c_1, c_2, ..., c_N\}$, assume the width of NoC transfer channel is W, then the individual testing time of each IP core T_i can be induced. Partition C into four groups P_i ($i = 0,1,2,3$), the core number and testing time of P_i is separately Num_i and TP_i, so that the grouping optimization objective is as (7).

$$Minimize(TP_i) \quad i = 0,1,2,3 \tag{7}$$

And the following constraints should be satisfied in the meantime.

$$\sum_{i=0}^{3} Num_i = N \tag{8}$$

$$Num_i \leq \left\lceil \frac{N}{4} \right\rceil \tag{9}$$

III. Testing Optimized NoC Mapping

We proposed testing optimized NoC mapping on ITC'02 benchmark circuits [10], which contains 12 typical SoC circuits provided by the research institutions and corporations. The benchmarks provide test structure description, such as the number of scan chains and test patterns. They can be used to evaluate the performance of test access architecture and scheduling algorithms.

ISBN: 978-988-19252-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

The optimization objective of IP cores grouping algorithm is testing time (the number of test cycles), so the independent testing time of each core needs to be firstly determined. Circuits in benchmark are made up with cores (modules). For each module, number of test patterns is p_m, number of input signals is i_m, number of output signals is o_m, number of bidirectional signals is b_m, number of scan chains is s_m, length of scan chain k is $l_{m,k}$, total number of scan FFs is f_m and $f_m = \sum_{k=1}^{s_m} l_{m,k}$. The number of test stimulus and response for each test vector is s_{im} and s_{om}, so (10) and (11) can be obtained.

$$s_{im} = i_m + b_m + f_m \tag{10}$$

$$s_{om} = o_m + b_m + f_m \tag{11}$$

The input data and output data of each test vector can be transmitted simultaneously in the scan test except the last one. Moreover, there is an extra cycle needed for function. Therefore, the independent testing time for an IP core T_c without considering the TAM width is as (12).

$$T_c = \{1 + \max(s_{im}, s_{om})\} \times p_m + \min(s_{im}, s_{om}) \tag{12}$$

Assume the TAM width is w, T_c will be as (13).

$$T_c = \left\lceil \frac{\max(s_{im}, s_{om}) \times p_m + \min(s_{im}, s_{om})}{w} \right\rceil + p_m \tag{13}$$

For NoC-based SoCs, TAM is usually the NoC communication channel and the width is typically 16 or 32 bits. Test data are transmitted in the form of packets, which are composed of flits in wormhole routing. One flit is transferred in one clock cycle. The testing time will be different when there is scan chain or not in IP core and is various when the number of the scan chain less than, equal to or greater than the transmission bandwidth. Therefore, we analyze the testing time of embedded IP cores under various circumstances and the results are summarized as follows.

1. IP core has no scan chain

Flits number of each packet is $N_f = \frac{\max(s_i, s_o)}{w}$, $s_i = i_m + b_m$, $s_o = o_m + b_m$, testing time is as (14).

$$T_c = \max\left\{\left\lceil \frac{s_i}{w} \right\rceil, \left\lceil \frac{s_o}{w} \right\rceil\right\} \times p_m + \min\left\{\left\lceil \frac{s_i}{w} \right\rceil, \left\lceil \frac{s_o}{w} \right\rceil\right\} + p_m \tag{14}$$

2. IP core has scan chains

A. the number of scan chain is much smaller than current bandwidth

Since the input and output can be transmitted together in the scan chain, so that $N_f = max(l_{m,k}) \quad k = 1, ..., s_m$, testing time is as (15).

$$T_c = N_f \times p_m + p_m + N_f \tag{15}$$

B. the number of scan chain is close or equal to the bandwidth

The input and output will not be transmitted together in the scan chain, so $N_f = N_{f1} + N_{f2}$, $N_{f1} = avg(l_{m,k}) \quad k = 1, ..., s_m$, $N_{f2} = \max\left\{\left\lceil \frac{s_i}{w} \right\rceil, \left\lceil \frac{s_o}{w} \right\rceil\right\}$, and $T_c = T_{c1} + T_{c2}$

$$T_{c1} = N_{f1} \times p_m + N_{f1} + p_m \tag{16}$$

$$T_{c2} = N_{f2} \times p_m + \min\left\{\left\lceil \frac{s_i}{w} \right\rceil, \left\lceil \frac{s_o}{w} \right\rceil\right\} \tag{17}$$

C. the number of scan chain is greater than the bandwidth so $N_f = \max\left\{\left\lceil \frac{f_m + s_i}{w} \right\rceil, \left\lceil \frac{f_m + s_o}{w} \right\rceil\right\}$, testing time is as (18).

$$T_c = N_f \times p_m + p_m + \min\left\{\left\lceil \frac{f_m + s_i}{w} \right\rceil, \left\lceil \frac{f_m + s_o}{w} \right\rceil\right\} \tag{18}$$

According to the above analysis, the testing time of each IP core for ITC'02 circuits will be obtained. For example, IP core testing times of g1023 circuit are shown in Table I.

TABLE I
TESTING TIME OF G1023 CIRCUIT

Core	W=16		W=32	
	flits/packet	T_c	Flits/packet	T_c
1	43	5939	43	5939
2	84	6374	84	6374
3	53	3131	53	3131
4	54	14794	54	14794
5	32	1715	32	1715
6	47	1775	47	1775
7	47	1679	47	1679
8	52	1695	52	1695
9	64	4484	64	4484
10	13	419	13	419
11	9	159	9	159
12	13	237	13	237
13	4	2564	2	1528
14	9	10248	5	6148

After testing time of each core are determined, P_A algorithm will apply for grouping IP cores so that the parallel testing time of NoC is minimized. The detail of the P_A algorithm process is shown in Fig. 4.

Procedure P_A algorithm

1. Define N_p, which is the upper limit of cores number in each partition based on the total cores number;
2. Sort cores in descending order of test cycles;
3. Add the first core (whose test cycles is maximum) to P_0;
4. Add the successive three cores to P_1, P_2, P_3 respectively;
5. For each of other core c_i
   i. Find partition P_{max} with current maximum test cycles;
   ii. Find partition P_{min} with current minimum test time and cores number is less than N_p;
   iii. Assign c_i to partition p, such that {test time(P_{max})−(test time(p)+test time(c_i))} is minimum and cores number is less than N_p;
6. If there is no such partition p, then assign c_i to P_{min}.

Fig. 4. Pseudo-code of P_A algorithm

Firstly, sorts all IP cores in descending order according to their independent testing time. Each IP core is then successively assigned to the group, whose length of testing time after this assignment is closest to, but not exceeding the current maximum test time. That means, each IP core is

ISBN: 978-988-19252-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

assigned to the group in which it achieves the best fit. If there is no such a group available, the IP core will be assigned to the current minimum testing time group.

According to IP cores grouping results and traffic information, the initial layout of IP cores in NoC may be determined. The specific mapping algorithm needs to be applied for further optimizing mapping performance.

Genetic Algorithm is one of evolutionary algorithms, which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover. When using genetic algorithms to solve practical problems, optimization parameters or variables need to be firstly translated into code string (usually in binary), these are so-called chromosomes and the process is encoding. After encoding is finished, construct the initial possible solution set, which provides the primary population size and randomly generates the chromosome set. In each generation, selects individual from problem domain based on the size of fitness, and implements the crossover and mutation by the genetic operator, resulting in a population representative of new solution set. This process will obtain the best individual in the offsprings, which is the same as the natural evolution. Finally, the last individual will be decoded and can be approximated as the optimal solution.

As far as the NoC mapping problem is concerned, we adopt the Genetic Algorithm as the optimization mapping algorithm. The communication power consumptions are calculated as the fitness. The appropriate genetic operations (selection, crossover and mutation) execute successively to produce a new generation of NoC mapping. This process will be iterated until the optimal mapping solution will be produced. The mapping scheme based on GA algorithm is shown in Fig.5.

```
Procedure GA algorithm
1. Initialize the population with segmented
     sequences;
2. Read the communication data;
3. For i=1 to max generation
        For each solution
            Generate placement;
            Calculate fitness;
        End for
4. Select;
5. Crossover;
6. Mutate;
7. i=i+1;
8. End for
```

Fig. 5. Mapping scheme based on GA algorithm

The GA algorithm steps are described in detail as follows.

A. Initialization

Based on the grouping results, IP cores are assigned to four ports. So the initialization of the Genetic Algorithm is divided into four sections. Among them, the range of chromosome segment is determined by grouping results. An example of chromosome coding is shown in Fig. 6.

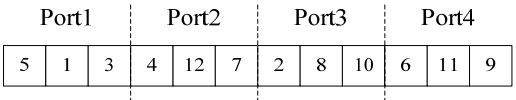

Fig. 6. Example of chromosome encoding

B. Calculating fitness

Fitness is the communication power consumption, whose value is related to the communication distance and the traffic between cores and can be obtained by (3).

C. Selection

Roulette selection method is applied to select individuals in the populations, in which the probability of an individual being selected is inversely proportional to its fitness. It reflects the proportion of the individual fitness to the sum of the entire population fitness.

D. Crossover

Crossover operation is to reconstruct parts of two parent individuals to generate new individuals. We adopt single-point crossover, a random number is generated within the length of the chromosome as a scheduled crossing point. Then two individual chromosomes exchange the order on this crossover point. Fig. 7 shows the improved sequence exchange example.

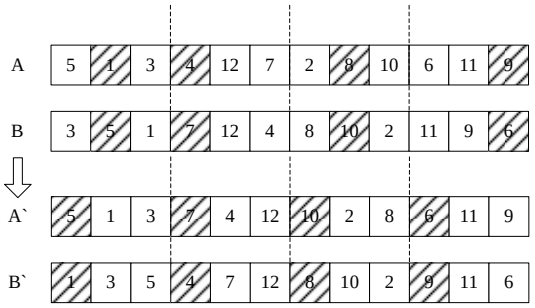

Fig. 7. Example of sequence exchange

E. Mutation

Mutation operation is to change the gene value of the individual, in order to ensure that the algorithm has the ability of random search and maintain the local population diversity. Usually selects one or more genes by random in the individuals and alters them at the preset probability.

IV. Performance Evaluation of Mapping Schemes

In order to evaluate the proposed testing optimization mapping scheme, some experiments are applied to four circuits of ITC'02 benchmark and they are d695, g1023, p22810, p93791. Firstly, P_A algorithm is implemented and the comparison of testing time with the Ref. [11] is shown in Table II.

TABLE II
Comparison of testing time

Circuit	W=16		W=32	
	Base Case[11]	P_A case	Base Case[11]	P_A case
d695	16197	13462 (-16.9%)	10705	9869 (-7.8%)
g1023	17925	14953 (-16.6%)	16489	14953 (-9.3%)
p22810	166800	154310 (-7.5%)	150921	135909 (-9.9%)
p93791	502876	453923 (-9.7%)	333091	228287 (-31.5%)

ISBN: 978-988-19252-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

The results in Table II showed that the P_A algorithm significantly optimized testing time. When data transfer width *W* is 16, the testing time is reduced by 12.67% on average; while *W* is 32, the average reduction is 14.63%.

NoC mapping needs the determined traffic information between various IP cores. Since g1023 has the same number of IP cores as H.263, it adopts traffic diagram of H.263 (H263.enc.mp3.enc) [12], while other three ITC'02 circuits adopt TGFF randomly generated traffic [13]. Fig. 8 shows the traffic diagram of d695 circuit as an example.

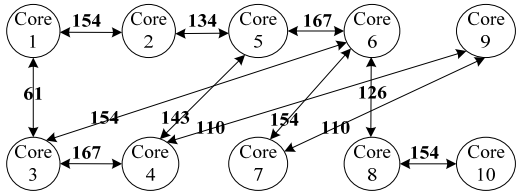

Fig. 8. Traffic diagram of d695 circuit

According to the known traffic information and partition results, the layout of the NoC may be initially determined; then the actual communication power consumption(fitness) will be calculated based on the cores distance; and the appropriate genetic operations will execute(such as selection, crossover and mutation) to produce a new generation of IP core layout. The process will be repeated until achieving the best optimal layout.

For example, testing optimized grouping of d695 is (5), (6,2), (10,7,9), (4,8,1,3). Applying the random mapping and the mapping result is showed as Fig. 9. The mapping cost of Fig. 9 is 3353 according to (3).

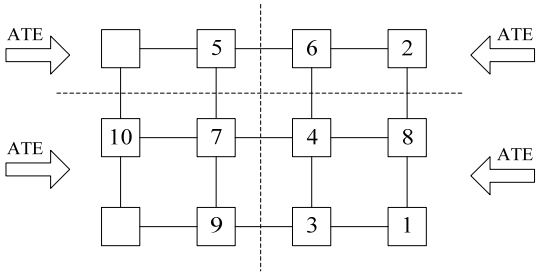

Fig. 9. Random mapping of d695 on NoC

After adopting genetic algorithms, the mapping result of d695 is shown in Fig. 10 and its mapping cost is 2516.

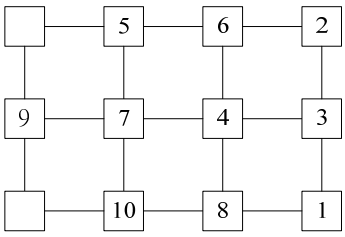

Fig. 10. Optimized mapping of d695 on NoC

Applying the genetic algorithm to d695, g1023, p22810, p93791 circuits and the transfer width is 16 and 32. The mapping cost comparisons with random mapping are shown in Table III. It can be concluded that the proposed mapping scheme is superior to the random mapping. When transfer width *W* is 16, the average reduction of mapping cost is 24.5%; while *W* is 32, the average reduction is 33.75%. It is obvious that the proposed mapping scheme optimizes testing efficiency while greatly reducing the mapping overhead.

TABLE III
COMPARISON OF MAPPING COST

Circuit		optimal cost	random cost	cost saving
W=16	d695	2516	3353	25%
	g1023	3214	4970	35%
	p22810	4178	5283	21%
	p93791	4409	5307	17%
W=32	d695	2892	3733	23%
	g1023	3526	4900	28%
	p22810	3019	4960	39%
	p93791	3365	6161	45%

V. CONCLUSION

This paper described the testing time of each IP core on the condition that NoC scan test structure and transmission bandwidth are provided. Then it introduced a partition algorithm cooperating with optimized test structure, which minimizes testing time. Finally, it proposed a NoC mapping scheme based on genetic algorithm with minimum overhead. The experiment results on ITC'02 benchmark circuits demonstrated the effectiveness of the testing optimization mapping scheme. Moreover, the testing optimized NoC mapping for 3D and many-core SoC is currently on the research.

REFERENCES

[1] T. Bjerregaard, S.A. Mahadevan. "Survey of Research and Practices of Network-on-Chip," *ACM Computing Surveys*, vol. 38, no. 1, 2006, pp. 1–51.

[2] S. Yang, L. Li, M. Gao. et al. "An Energy- and Delay-aware Mapping Method of NoC," *Acta Electronica Sinica*, vol.36, no. 5, 2008, pp. 937–942.

[3] J. C. Hu, R. Marculescu. "Energy-and performance-aware Mapping for Regular NoC Architectures," *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 2005, vol. 24, no. 4, pp. 551–562.

[4] L. Wang, X. Ling. "Energy-and latency-aware NoC Mapping Based on Chaos Discrete Particle Swarm Optimization," in *Proc. Communications and Mobile Computing Conf.*, Los Alamitos, 2010, pp. 263–268.

[5] K. Chakrabarty, "Optimal Test Access Architectures for System-on-a-Chip," *ACM T Des Automat EL*, 2001, vol. 6, no. 1, pp. 26–49.

[6] A.M. Amory, F. Ferlini, M. Lubaszewski et al. "DfT for the Reuse of Networks-on-Chip as Test Access Mechanism," in *Proc. 25th IEEE VLSI Test Symposium.*, Berkeley, CA, 2007, pp. 435–440.

[7] M. Agrawal, M. Richter, K. Chakrabarty. "A Dynamic Programming Solution for Optimizing Test Delivery in Multicore SOCs," in *Proc. ITC'12 Conf.*, Anaheim, CA, 2012, pp. 1–10.

[8] Tayan, O. "Networks-on-Chip: Challenges, trends and mechanisms for enhancements," in *Proc. ICICT'09 Conf. Information and Communication Technologies*, Karachi, 2009, pp. 57–62.

[9] Y. Zhang, N. Wu, F. Ge. "The Co-design of Test Structure and Test Data Transfer Mode for 2D-Mesh NoC," I*AENG Trans. on Engineering Technologies*, 2013, pp. 171–184.

[10] E.J. Marinissen, V. Iyengar, K. Chakrabarty. "A Set of Benchmarks for Modular Testing of SOCs," in *Proc. ITC'02 Conf.*, 2002, pp. 519–528.

[11] C. Liu, J. Shi, , E. Cota et al. "Power-Aware Test Scheduling in Network-on-Chip Using Variable-Rate on-Chip Clocking," in *Proc. 23th IEEE VLSI Test Symposium.* 2005, pp. 349–354.

[12] J.Hu, R. Marculescu. "Energy-Aware Mapping for Tile-based NoC Architectures under Performance Constraints," in *Proc. VLSI Test Asia and South Pacific Design Automat Conf.*, Kitakyushu, 2003, pp. 233–239.

[13] P. R. Dick, L. D. Rhodes, W. Wayne. "TGFF: Task Graphs for Free," in *Proc. 6th International Hardware/Software Codesign Workshop*, 1998, pp. 97–101.

ISBN: 978-988-19252-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)