Concept Paper

ILP Based Power-Aware Test Time Reduction Using On-Chip Clocking in NoC Based SoC

Harikrishna Parmar and Usha Mehta *

ECC Department, Nirma University, Ahmedabad 382481, India; hk_parmar@yahoo.co.in

* Correspondence: usha.mehta@nirmauni.ac.in; Tel.: +91-9898993232

Received: 22 February 2019; Accepted: 21 May 2019; Published: 17 June 2019

Abstract: Network-on-chip (NoC) based system-on-chips (SoC) has been a promising paradigm of core-based systems. It is difficult and challenging to test the individual Intellectual property IP cores of SoC with the constraints of test time and test power. By reusing the on-chip communication network of NoC for the testing of different cores in SoC, the test time and test cost can be reduced effectively. In this paper, we have proposed a power-aware test scheduling by reusing existing on-chip communication network. On-chip test clock frequencies are used for power efficient test scheduling. In this paper, an integer linear programming (ILP) model is proposed. This model assigns different frequencies to the NoC cores in such a way that it reduces the test time without crossing the power budget. Experimental results on the ITC'02 benchmark SoCs show that the proposed ILP method gives up to 50% reduction in test time compared to the existing method.

Keywords: NoC based SoC; test scheduling; test time reduction; test bus architecture; test power; test clock frequency; ILP

1. Introduction

Considering time-to-market, today's system-on-chips (SoCs) are becoming sea-of-cores! The overall architecture of SoC is becoming complex. In this scenario, testing of each IP core as stand-alone and as a part of system is difficult. Test access mechanism (TAM) architecture is designed to fetch the test data from the automatic test equipment (ATE) to the core and to transfer test response from the core to sink. The bus-based TAM was being widely used in SoC. The typical bus-based TAM architecture for SoC is having blems related to scalable global synchronous clock, communication time and performance issues [1]. To overcome these issues, NoC based SoCs are introduced. Use of routers, channels and packet switching interconnections reduces the problem of test data communication from source to IP core and IP core to sink [2].

For effective reduction in test time under the constraints of power budget, test scheduling is the key point [3]. Minimization of test time with NoC being used as TAM is a multivariable problem, including the co-optimization of the core assignment to TAM for test data transportation, active exploitation of the channel bandwidth, and the number and location of the test interface. Further, it is to be noted that the test time and test power are correlated issues. Increase in test frequency may reduce the test time but simultaneously increases the dynamic test power. In this paper, an ILP based method is proposed to reduce the test time in NoC with multiple test clock frequencies. Here, the test power and the test time are formulated as a function of the test clock frequency, and hence this method gets the test time reduction for the predefined power limit. In the proposed method, dynamic clock control based on the power dissipation of the test session is adopted.

The whole paper is organized as follows. Section 2 discusses the prior work on TAM in NoC. The motivation and background are described in Section 3. The problem statement is described in

Section 4. The proposed test methodology to establish the problem statement is shown in Section 5, whereas Section 6 discusses the experimental result analysis. Finally, Section 7 draws a conclusion.

2. Prior Work

TAM architecture is the mediator to transfer the test data from automatic test equipment to the core and the core to the sink. In NoC based SoC, the NoC fabric can be used as TAM replacing the dedicated TAM. Since no extra hardware is required to build TAM, it reduces the cost of NoC based SoC testing. The fundamental of reusing NoC as TAM is first introduced in [3]. Here, the core having a longer test time is given higher priority in scheduling to reduce testing time. This method was further developed in [4], with power constraint and increased test parallelism. In [5], the test time has been shown as a function of the TAM width and assignment of a core to the TAM width to minimize the test time. In [6], it is shown that the test time of the core varies in staircase pattern with TAM width.

For high speed data transport over the network, the time division multiplexing (TDM) approach was discussed in [7]. In [8,9], power-aware test scheduling is shown by effectively utilizing the on-chip network. Here the on-chip clocking is used in such a way that the faster clock is assigned to some cores and slower to remaining to limit the overall power consumption. In short, clock rate distribution is effectively designed in this methodology to have a lower test time. Test scheduling using rectangle packing solution and use of multiple test clocks for NoC test was proposed in [10]. Test scheduling with the different topology of the network was described in [11]. In [12], unicast-based multicast scheme is used for NoC based SoC testing, where different techniques like test data compression, power constraint scheduling, vector compactions are also combined to minimize test time. Power and thermal-aware NoC test scheduling with a multiple clock rate is proposed in [13]. The algorithm is designed based on Integer linear programming and simulated annealing technique. Co-optimization of pin assignment to the access point and NoC core test scheduling was proposed in [14]. Minimization of test time with the given pin count is well described here. In [15], test delivery optimization of many core systems is proposed. Here, NoC partitioning difficulty is formulated with dynamic programming. In [16], the hybrid test data transportation system for advanced NoC based SoC is described. As the schedule is affected by the location of the access point and the position of the embedded core, a new technique is developed here for concurrently testing several different cores.

3. Motivation

The ITC'02 benchmark NoC based SoC d695 was used as an example. The d695 has 10 IP cores with 2D mesh topology, as shown in Figure 1a [4]. The IP cores included in d695 are standard ISCAS benchmark circuits. The database for the SoC d695 is shown in Figure 1b, which includes number of I/Os, scan chain, number of test patterns, test power and test time [17,18].

Figure 1. Cont.

Core	No. of I/P	No. of O/P	No. of scan chain	No. of test patterns	Test time (Test clocks)	Test Power (mw)
C6288	32	32	0	12	25	660
C7552	207	108	0	73	584	602
S838	36	3	1	75	2475	823
S9234	40	43	4	105	5775	275
S38584	70	336	32	110	5843	690
S13207	78	168	16	234	9828	354
S15850	93	166	16	95	3325	530
S5378	39	53	4	97	4559	753
S35932	67	352	32	12	834	641
S38417	60	138	32	68	3859	1144

(b)

Figure 1. (a) ITC'02 Benchmark circuit for NoC based system-on-chips (SoC)—d695 [4] (b) SoC d695 database [17,18].

Here, there are 10 cores each with test time t_i and power dissipation P_i . The cores can be tested individually or in groups i.e., sessions. The length of each session i.e., test time for that session can be defined as the maximum of test time of individual core being tested in that session and total power dissipation for that session is the summation of individual test power of cores being tested in that session.

Generally, the available NoC channel width exceeds the TAM width of IP core test wrapper and hence some channel width may remain idle during testing. If we assign the cores to channel in such a way that, parallel testing of core is maximum and the idle channel width is minimum, then the test time is reduced effectively. In [19], we have presented the algorithm for the effective utilization of idle channel width to determine the width of the TAM bus assigned to the individual core and improved partition strategy for the TAM bus. The algorithm also focuses on improved core assignment to the partitioned TAM bus and test scheduling. The limitation of the algorithm in [19] is that it is exhaustive in nature and has a long execution time for big SoCs.

Here, in Figure 2, we show the effective TAM and IO assignment for the test scheduling of SoC d695 cores without power constraints. The number in each rectangle block represents the core number and corresponding test time in terms of test clock cycles for that core. The shaded region in this figure shows the idle time. Cores are mapped to any one of the three I/O pair. Results show that cores 1, 2, 7, 8 and 10 are assigned to IO1, cores 3 and 6 are assigned to IO2, and cores 4, 5 and 9 are assigned to IO3. Total test time of the cores assigned to IO1 is 12,352 clocks, total test time of the cores assigned to IO2 is 12,303 clocks and total test time of the cores assigned to IO3 is 12,452 clocks. In short, total test time of the SoC d695 is the maximum out of these three, which is 12,452 clocks.

Figure 2. Test scheduling with effective test access mechanism (TAM) width allocation.

With single frequency allocation, assuming a normalized test frequency of 1Hz in Figure 2a, maximum test time out of three I/O pair is 12,452 s.

In [20], we have proposed a method to minimize the test time in NoC based SoC by variable clock frequency to session based scheduling. Let's assume that the maximum allowable average power for SoC is P_{max} and test power for individual core is P_{core} , and test frequency is f_{test} . Here test time is inversely proportional to test frequency and dynamic test power is directly proportional to test frequency. Hence to reduce an individual core's test time, we have increased its test frequency by a power factor time. The power factor was considered based on the power limit constraint of SoC. $Power\ Factor = P_{max}/P_{core}$. After increasing the test frequency, the scheduling/rectangle packing of IP core is converted into serial testing of cores. With experimental results on different benchmark SoCs, we have shown that the overall test time was reduced for each SoC. But the limitation here is reliability issues. As we have made each of the core to run at higher frequencies, the thermal reliability issues were still unanswered.

Here, in this paper, to overcome the limitation of reliability issues, we have adopted a power constraint test scheduling where the maximum power limit was set as the percentage of the gross of total functional power consumption of all core i.e., 30% power means 30% of summation of total power consumed by each core. This value of 30% was chosen to stringent the power constraints for our experiments [9]. We have further constrained the frequency to increase by factor two for some of the cores with lower power budget and at the same time the frequency is slow down by factor $\frac{1}{2}$. With higher power budget cores. In general, we used f_{test} , $2f_{test}$ or $\frac{1}{2}f_{test}$ to individual cores using P_{max} and P_{core} as constraint values.

Figure 3a,b shows the test scheduling of SoC d695 cores with power constraints. Here, P_{max} value is taken as 30% of the total test power consumed by all cores in SoC. Number in each rectangle block represents core number and corresponding test clock cycles of that core. Cores are mapped to any one of the three I/O pair.

With single frequency allocation in Figure 3a, the maximum test time out of three I/O pair was 18,546 ns (considering a normalized test frequency of 1GHz) whereas, with multiple frequencies in Figure 3b, it was 12,439 ns considering the three clock frequencies $\frac{1}{2}$ GHz, 1 GHz and 2 GHz, which showed the reduction in test time.

(a)

Figure 3. Cont.

Figure 3. (a) Power aware test scheduling with single frequency allocation (b) Power aware test scheduling with multiple frequency allocation.

Further, we have used the TAM channel width and IP core's bus width as constraint for scheduling. It means, the test time reduction problem formation was done using the integer linear program where the TAM channel width, IP core's bus width, P_{max} and P_{core} were used as a constraint for scheduling and frequency allocation objective. The exact problem formation for ILP is described in the next section.

4. Problem Statement

The problem statement for the proposed method of test time minimization can be stated as below:

- Given the number of inputs, outputs and scan chains for the NoC based SoC;
- Given the maximum TAM width W_{max} to be used for data transfer between IP cores of SoC;
- Given the network topology for NoC to be used for IP cores of SoC;
- Given the multiple test clock frequency for the testing of various IP Cores in NoC based SoC;

Determine:

1. Optimum allocation of the core to the input-output pairs.
2. Optimum allocation of test clock to the cores.
3. Scheduling algorithm to minimize test time with given power constraints.

5. Proposed Methodology

Here, ILP based test scheduling algorithm is proposed for a dedicated routing path, which is implemented for the non-preemptive test.

The ILP based schedule will assign each core a dedicated routing path, input port, output port, test clock frequency, channel through which test vector will transport from input port to the core and output response from core to the output port. Once the core is scheduled in the dedicated routing path, all the resources are reserved until entire test is finished for that core.

Let us consider:

N_c = Number of Cores;

N_p = Number of input-output pairs available in NoC based SoC;

W_{max} = a maximum TAM width;

t_i = test time of core i ;

f_n = multiple test clock frequencies.

T = total test time

T_{ro} = time consumed in the router
 T_{chan} = time consumed in NoC channels
 N_{chan} = number of channels
 N_{ro} = the number of routers
 P_{ro} = power consumed in the router
 P_{chan} = power consumed in NoC channels
 N_{chan} = number of channels
 N_{ro} = number of routers
 S_i = length of wrapper-scan-in chain
 S_o = length of wrapper-scan-out chain
 TP = Test pattern count of the core

Here we are considering mesh network as the network topology for IP Cores of NoC based SoC. The test time t_i of the core i is the combination of the two entities.

(1) The time it takes to transmit data through a number of channels and routers, which is defined as T_{cri}

$$T_{cri} = N_{chan} * T_{chan} + N_{ro} * T_{ro} \quad (1)$$

(2) Test time of the core which is defined as T_{corei}

Core test time T_{ci} depends on the TAM width selection and arrangement of the scan chain with the best fit decreasing algorithm [17]. So total core test time in terms of clock cycles is given as

$$T_{corei} = (1 + \max(S_i, S_o)) * TP + \min(S_i, S_o) \quad (2)$$

The combined test time of the SoC can be defined as

In NoC, the testing time of core is considerably higher than the transmit time. So here, the transmit time is neglected as in contrast to core test time. Here, S_i and S_o is basically a wrapper scan in and scan out flip flops and it works on the edge of clocks, so the test time measured here is in the number of clock cycles it used.

$$t_i = \sum_{i=1}^{N_c} T_{corei} \quad (3)$$

The scheduling of the core will start after calculating test time of each core. Here, we are proposing two scheduling techniques: 1. test scheduling without power constraint; 2. test scheduling with power constraint. ILP formulation is developed to schedule the cores. We will discuss and analyze these two techniques here.

(3) ILP formulation for Test Scheduling with single frequency allocation and without power constraint:

The main motive behind this algorithm is to reduce test time of the SoC. So, the main objective for this ILP formulation becomes the minimization of test time.

Main Objective: Min T

Let X_{ij} is the 0-1 binary variable which is defined as

$$\begin{aligned}
 X_{ij} &= 1, & \text{if core } i \text{ is assigned to I/O pair } j. \\
 &= 0, & \text{otherwise}
 \end{aligned}$$

Each core is mapped to one and only one input-output pair. Mathematically, we can write this unity condition using the binary variable X_{ij} , which is defined as

$$\sum_{j=1}^{N_p} x_{ij} = 1, \forall i, 1 \leq i \leq N_c \quad (4)$$

Since test time of SoC is maximum of individual I/O pair's test time. The maximum test time among all I/O pairs can be given as

$$T = \max \sum_{i=1}^{N_c} t_i * x_{ij}, \forall j, 1 \leq j \leq N_p \quad (5)$$

(4) Pseudo Code for the ILP Problem formulation

The detail description of ILP formulation in form of pseudo code is shown in Figure 4.

- 1 Define number of cores, number of I/O pairs and test clock frequency.
- 2 Calculate test time of each core using Equations 1, 2 and 3. Store the test time in an array.
- 3 Make List l1, which contains all unscheduled cores along with test time
- 4 Make List l2, which contains all scheduled cores
- 5 **While** List l1 is not empty **do**
- 6 Find free I/O pair
- 7 **If** no free I/O pair
- 8 Update current test time of the cores with all I/Os
- 9 Repeat procedure from Step 5
- 10 **Else**
- 11 Check for routing path
- 12 **If** the core is attempted
- 13 Insert the core in List l2
- 14 Remove the core from list l1
- 15 Update current test time
- 16 Repeat procedure from Step 5
- 17 **Else**
- 18 Try next core from List l1
- 19 Repeat procedure from Step 5
- 20 **End if**
- 21 **End if**
- 22 **End while**
- 23 Find the maximum test time of cores assigned to different I/O pairs which is the final test time.

Figure 4. Pseudo code for ILP problem formulation.

(5) Power aware test scheduling with multiple frequency allocation.

In NoC, power consumed in routers and channels is given as

$$P_{cri} = N_{chan} * P_{chan} + N_{ro} * P_{ro} \quad (6)$$

Power consumption in each core can be calculated from information, like the number of input-outputs, a number of scan flip flops etc. So total power consumption can be given as,

$$P_{total} = \sum_{i=1}^{N_c} (P_{cri} + P_{corei}) \quad (7)$$

Power aware test scheduling starts with Equation (6) and (7). Here, Cores are scheduled in such a way that power consumption in a given session should not go beyond Pmax value. Mathematically it is defined as:

$$\sum_{j=1}^{N_p} \sum_{i=1}^{N_c} \sum_{k=1}^{f_n} P_i * X_{ijkl} \leq P_{max}, \forall 1, 1 \leq l \leq S \quad (8)$$

where, S is the number of sessions.

Since test time of SoC is the maximum of individual I/O pair's test time. The maximum test time among all i/o pairs can be given as

$$T = \max(\sum_{j=1}^{N_p} \sum_{i=1}^{N_c} \sum_{k=1}^{f_n} t_i * x_{ijk}), \forall 1, 1 \leq S \quad (9)$$

The ILP model of this whole system can be summarized as

Objective: Minimize 'T' Subject to:

$$\begin{aligned} \sum_{j=1}^{N_p} \sum_{k=1}^{f_n} x_{ijk} &= 1, \forall i, 1 \leq i \leq N_c \\ \sum_{i=1}^{N_c} x_{ijk} &\geq 1, \text{ for } j = 1, \forall k, 1 \leq k \leq f_k \text{ and } j = 1 \\ \sum_{j=1}^{N_p} \sum_{i=1}^{N_c} \sum_{k=1}^{f_n} P_i * X_{ijkl} &\leq P_{max}, \forall 1, 1 \leq l \leq S \\ T &= \max(\sum_{j=1}^{N_p} \sum_{i=1}^{N_c} \sum_{k=1}^{f_n} t_i * x_{ijk}), \forall 1, 1 \leq S \end{aligned}$$

The detail description of ILP formulation in form of pseudo code is shown in Figure 5.


```

1  Define number of cores, number of I/O pairs, test power of the cores  $P_i$ , maximum power budget
   of the SoC  $P_{\max}$  and
   test clock frequencies.
2  Calculate the test time of each core using Equations 1, 2 and 3. Store the test time in an array.
3  Define Array A, which takes the maximum test time of the core in a particular session.
4  Make List l1, which contains all unscheduled cores along with test time with different frequencies.
5  Make List l2, which contains all scheduled cores
6  Start a new session for scheduling
7      While List l1 is not empty do
8          Find free I/O pair
9          If no free I/O pair
10             Update current test time of the cores with all I/Os
11             Repeat procedure from Step 7
12         Else
13             If  $\sum P_i < P_{\max}$ 
14                 Check for routing path
15                 If the core is attempted
16                     Insert the core in List l2
17                     Remove the core from List l1
18                     Update current test time
19                     Repeat procedure from Step 7
20                 Else
21                     Remove recently added core from List l1
22                     Repeat procedure from Step 7
23                 End if
24             Else if no more core is suitable for schedule in the session
25                 Remove the recently added core from List l1
26                 Update Array A with the maximum test time of the core
27                 Start a new session and repeat procedure from Step 6
28             Else
29                 Remove recently added core from List l1
30                 Repeat procedure from Step 7
31             End if
32         End if
33     End while
34  Final test time is the sum of all test time stored in Array A.

```

Figure 5. Pseudo code for Proposed ILP Problem for Power aware test scheduling with Variable Test Frequency Allocation.

6. Experimental Results and Analysis

Here, the proposed algorithm was implemented on the windows core i3 processor with 2.00 GHz frequency and 4 GB RAM. The simulation was done on MATLAB 14 and LPSOLVE. The algorithm was applied to the widely used ITC'02 benchmark SoCs. Since the power consumption of each core is

not mentioned in the ITC'02 benchmark database, it was taken from [21]. For the proposed algorithm, it was assumed that SoCs have the similar NoC fabric as given in [9,15,17] including network topology, core placement etc.

The simulation results of the proposed algorithm for various SoCs were compared with [9,17]. Tables 1 and 2 show the detailed results for ITC'02 benchmark SoC d695. Columns 2–3 in Table 1 show the results in case of normal algorithm and ILP based Algorithm implementation with two cases:

1. Single frequency allocation: where the normal test frequency is applied to each core and then test scheduling is done.
2. Multiple frequency allocation: each core is mapped with any one of $f/2$, f , and $2f$ frequency and then power constrained test scheduling is done.

Table 1. Results of test scheduling with single frequency allocation and without power constraint for SoC d695 using integer linear programming (ILP).

Number of I/O Pairs	Test Time (nS) Required in Case of Single freq. Allocation [11]	Test Time (nS) Required in Case of Single freq. Allocation with Proposed ILP	Test Time (nS) Required in Case of Multiple freq. Allocation [11]	Test Time (nS) Required in Case of Multiple freq. Allocation with Proposed ILP
2/2	18,869	18,561	14,176	9457
3/3	13,412	12,452	9073	6454
4/4	10,705	9828	9073	4914

Table 2. Proposed clock allocation & core scheduling in case of multiple frequency allocation without power constraint using ILP.

No. of I/O Pairs	Pair No.	A Sequence of Core Allocation with Corresponding Frequency Allocation to Particular I/O Pair	Test Time nS
2/2	1	Core3-2f, Core4-2f, Core6-2f, Core9-2f	9457
	2	Core1-f/2, Core2-f, Core5-2f, Core7-2f, Core8-2f, Core10-2f	
3/3	1	Core2-f, Core7-2f, Core8-2f, Core10-2f	6454
	2	Core3-2f, Core6-2f, Core1-f/2	
	3	Core4-2f, Core5-2f, Core9-2f	
4/4	1	Core6-2f	4914
	2	Core1-f/2, Core2-f, Core7-2f, Core8-2f	
	3	Core3-2f, Core4-2f, Core9-2f	
	4	Core5-2f, Core10-2f	

Frequency allocation was done with the ILP model as described earlier. Results are shown in Table 2, which describes the allocation of frequencies to different cores and allocation of cores to I/O pairs. Column 1 shows the number of I/O pairs for each system. Column 2 shows the mapping of the core to different I/O pairs and allocation of frequency to the core. For example, for I/O pair 2/2, cores 3, 4, 6 and 9 are assigned to I/O pair 1 and cores 1, 2, 5, 7, 8 and 10 are assigned to I/O pair 2. It also shows the allocation of frequency to the cores. For example, Core1-f/2 means core1 is assigned with frequency $f/2$. Similarly, all the cores are assigned with a specific frequency. Optimized results of test time are shown in the last column of that table.

Tables 3–5 show the % reduction in test time for three different benchmark SoCs.

Table 3. Comparison of the proposed test scheduling results with preceding work [9] for SoC d695.

No. of I/O Pairs	Test Time in ns [9]		Test Time in ns [Proposed]		% Reduction in Time as Compared to [9]	
	50% of the Total Power Limit	30% of the Total Power Limit	50% of the Total Power Limit	30% of the Total Power Limit	50% of the Total Power Limit	30% of the Total Power Limit
2/2	15,576	21,029	9167	12,881	41.4%	38.7%
3/3	12,887	18,317	8706	12,326	32.4%	32.7%
4/4	13,696	18,317	8706	12,326	36.4%	32.7%

Table 4. Comparison of proposed test scheduling results with preceding work [9] for SoC p22810.

No. of I/O Pairs	Test Time in ns [9]		Test Time in ns [Proposed]		% Reduction in Time as Compared to [9]	
	50% of the Total Power Limit	30% of the Total Power Limit	50% of the Total Power Limit	30% of the Total Power Limit	50% of the Total Power Limit	30% of the Total Power Limit
2/2	177,915	211,505	152,671	153,111	14.4%	27.6%
3/3	154,478	199,151	106,560	107,926	31.0%	45.8%
4/4	143,044	195,122	76,546	95,718	46.4%	50.9%

Table 5. Comparison of proposed test scheduling results with preceding work [9] for SoC p93791.

No. of I/O Pairs	Test Time in ns [9]		Test Time in ns [Proposed]		% Reduction in Time as Compared to [9]	
	50% of the Total Power Limit	30% of the Total Power Limit	50% of the Total Power Limit	30% of the Total Power Limit	50% of the Total Power Limit	30% of the Total Power Limit
2/2	498,353	538,062	338,902	417,432	31.9%	22.41%
3/3	436,400	523,082	279,212	354,218	36.0%	32.2%
4/4	387,841	512,280	275,207	353,913	29.0%	30.9%

7. Conclusions

Here, it is shown that significant test time reduction is achieved by mapping the appropriate test clock frequency to the core with power constraint test scheduling. Here, the dedicated routing path algorithm was chosen to transport test data to the core. The ILP method was used to solve the scheduling of the core in NoC. Experimental results on the ITC'02 benchmark show that the ILP based method reduced test time up to 50%, which shows the effectiveness of this method.

Author Contributions: Research is carried out by the H.P. under the guidance of U.M. Both the authors contributed to make this manuscript and research.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Moreno, E.; Webber, T.; Marcon, C.; Moraes, F.; Calazans, N. MoNoC: A monitored network on chip with path adaptation mechanism. *J. Syst. Arch.* **2014**, *60*, 783–795. [\[CrossRef\]](#)
- Touzene, A. On All-to-All Broadcast in Dense Gaussian Network On-Chip. *IEEE Trans. Parallel Distrib. Syst.* **2015**, *26*, 1085–1095. [\[CrossRef\]](#)

3. Cota, E. The impact of NoC reuse on the testing of core-based systems. In Proceedings of the 21st VLSI Test Symposium, Napa, CA, USA, 1 May 2003; pp. 128–133.
4. Cota, E.; Liu, C. Constraint-driven test scheduling for NoC based systems. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2006**, *25*, 2465–2478. [[CrossRef](#)]
5. Larsson, E. *Introduction to Advanced System-on-Chip Test Design and Optimization*; Springer: Berlin, Germany, 2005.
6. Iyengar, V.; Chakrabarty, K.; Marinissen, E. Test wrapper and test access mechanism co-optimization for system-on-chip. *J. Electron. Test. Theory Appl.* **2002**, *18*, 213–230. [[CrossRef](#)]
7. Nolen, J.; Mahapatra, R. A TDM Test Scheduling Method for Network-on-Chip Systems. In Proceedings of the Seventh International Workshop on Microprocessor Test and Verification (MTV'06), Austin, TX, USA, 3–5 November 2005; pp. 90–98.
8. Liu, C.; Iyengar, V. Test scheduling with thermal optimization for network-on-chip systems using variable-rate on-chip clocking. In Proceedings of the conference on Design, automation and test in Europe, Munich, Germany, 6–10 March 2005; pp. 349–354.
9. Liu, C.; Iyengar, V.; Shi, J.; Cota, E. Power-Aware Test Scheduling in Network-on-Chip Using Variable-Rate On-Chip Clocking. In Proceedings of the 23rd IEEE VLSI Test Symposium (VTS'05), Palm Springs, CA, USA, 1–5 May 2005; pp. 349–354.
10. Ahn, J.; Kang, S. Test Scheduling of NoC-Based SoCs Using Multiple Test Clocks. *ETRI J.* **2006**, *28*, 475–485. [[CrossRef](#)]
11. Amory, A.M.; Lazzari, C.; Lubaszewski, M.S.; Moraes, F.G. A new test scheduling algorithm based on Networks-on-Chip as Test Access Mechanisms. *J. Parallel Distrib. Comput.* **2011**, *71*, 675–686. [[CrossRef](#)]
12. Xiang, D.; Zhang, Y. Cost-Effective Power-Aware Core Testing in NoCs Based on a New Unicast-Based Multicast Scheme. *IEEE Trans. Comput. Integr. Circuits Syst.* **2011**, *30*, 135–147. [[CrossRef](#)]
13. Aktouf, C. A complete strategy for testing an on-chip multiprocessor architecture. *IEEE Test. Comput.* **2002**, *19*, 18–28. [[CrossRef](#)]
14. Richter, M.; Chakrabarty, K. Optimization of Test Pin-Count, Test Scheduling, and Test Access for NoC-Based Multicore SoCs. *IEEE Trans. Comput.* **2014**, *63*, 691–702. [[CrossRef](#)]
15. Agrawal, M.; Richter, M.; Chakrabarty, K. Test-delivery optimization in many core SoC. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2014**, *33*, 1067–1080. [[CrossRef](#)]
16. Ansari, M.A.; Kim, D.; Jung, J.; Park, S. Hybrid Test Data Transportation Scheme for Advanced NoC-Based SoCs. *J. Semicond. Technol. Sci.* **2015**, *15*, 85–95. [[CrossRef](#)]
17. Hu, C.; Li, Z.; Xu, C.; Jia, M. Test Scheduling for Network-on-Chip Using XY-Direction Connected Subgraph Partition and Multiple Test Clocks. *J. Electron. Test.* **2016**, *32*, 31–42. [[CrossRef](#)]
18. Chakrabarty, K.; Iyengar, V.; Chandra, A. *Test Resource Partitioning for System-on-a-Chip (Frontiers in Electronic Testing)*; Springer: New York, NY, USA, 2002; Volume 20.
19. Parmar, H.; Mehta, U. An improved algorithm for TAM optimization to reduce test application time in core based SoC. In Proceedings of the IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE), Dhaka, Bangladesh, 19–20 December 2015.
20. Parmar, H.; Mehta, U. Power Aware Network on Chip Test Scheduling with Variable Test Clock Frequency. In *Ubiquitous Communications and Network Computing*; Springer: Cham, Switzerland, 2017; pp. 256–264.
21. Pouget, J.; Larsson, E.; Peng, Z. SOC test time minimization under multiple constraints. In Proceedings of the 12th Asian Test Symposium (ATS), Xi'an, China, 17–19 November 2003; pp. 312–317.