RESEARCH ARTICLE OPEN ACCESS

Test Scheduling of Core Based SOC Using Greedy Algorithm

Naveen Dewan, Harpreet Vohra
Department of Electronics and Communication Engineering, Thapar University, Patiala.

Abstract—
Escalating increase in the level of integration has led the design engineers to embed the pre-design and pre-verified logic blocks on chip to make a complete system on chip (SoC) technology. This advancing technology trend has led to new challenges for the design and test engineers. To ensure the testability of the entire system, the test planning needs to be done during design phase. To save the test cost, the test application time needs to be reduced which requires the test to be done concurrently. However the parallel running of test of multiple cores increases the power dissipation. This thereby leads to make test optimization to take care of time and power. This paper presents an approach for the scheduling the cores with the test time, power, test access mechanism and bandwidth constraint based on greedy algorithm. The TAM allotment to the various cores is done dynamically to save the test time and utilize the full bandwidth. Scheduling is done on ITC'02 benchmark circuits. Experiments on these ITC'02 benchmark circuits show that this algorithm offers lower test application time compared to the multiple constraint driven system-on-chip.

Keywords— SoC testing, test scheduling, test bandwidth, power constraint

I. INTRODUCTION

The Advancement in design methodologies and semiconductor process technologies has led to the development of systems with excessive functionality implemented on a single die, called system-on-chip. A set of predesigned and pre-verified design modules in the form of hard, soft or firm cores brought from either are integrated into a system using user-defined logic (UDL) and interconnects. We can implement complex systems having digital, analog and mixed signal components. The urgent time to market requirement poses many challenges for the design and test engineers. The associated test cost has become the major bottleneck in the reduction of overall cost of system[23]. Testing cost have made IC testing more difficult. ITRS semiconductor roadmap [17] represents that there will be a need of hundred of processors for the future generation of SoC designs which will further increase the test cost. Testing of SoC is costly due to large data volume introduced due to increase in the integration and interconnection intricacies, huge power dissipation during test, expensive test generation procedures , heterogeneous mix of cores and their long test application times. Many techniques have been proposed to reduce the cost by test scheduling, reducing test data volume and optimizing test design mechanism. Test generation can either be done off-chip by employing ATPG (Automatic test pattern generation) algorithms running on expensive automatic test equipments or on-chip using a built-in hardware called BIST (Built In Self Test) [15]. BIST offers the benefit in case if on-chip TAM availability is less. However BIST ready cores are not always available, also the multi site testing of SoC for test time reduction makes the ATE more promising. For the test access and application Zorian et al. [24][25] proposed a modular approach. It comprises of wrapper design [4][27][28][20], TAM [21][22][29][30]and test scheduling [2][18][19][26]. TAM optimization and test scheduling have been the integral part of the research and test optimization for past three decades. Test scheduling has been proved to be an NP-hard problem. This paper proposes a greedy algorithm based approach for test scheduling to reduce the test time subject to test power and bandwidth constraint. We can reduce the problem into a rectangle packing problem [3]. Experimental results for ITC'02 benchmark circuits show the optimal results achieved. Also a comparison with Pouget et al.[4] shows to be a better approach. This paper also includes the background of the SoC test scheduling based papers.

II. LITERATURE REVIEW

Concurrently testing a core based system accelerates the speed of testing. An efficient schedule can reduce the overall test time. Several works have been proposed on test scheduling using various algorithms. Pouget et al.[4] proposed a test scheduling technique with the objective to minimize the test application time while considering multiple resource conflicts. The conflicts are testing of interconnections between the cores, module testing with multiple test sets, sharing of the TAM and test power conflicts. Wrapper design algorithm and test scheduling heuristic algorithm is used to calculate the test time. Further, calculation of the all Pareto optimal points for each core and Optimal Time has been calculated for each core from these Pareto optimal points. Considering all conflicts and optimal points the scheduling is done. Goel et al. [5] have proposed two approaches for efficient testing of SoCs with hierarchical cores. In the first approach the problem is solved by wrapper design this approach leaves full flexibility for TAM optimization and test scheduling. The second approach is based on a modified wrapper design for parent cores that operate in two disjoint modes for testing of parent and child cores. The first approach gives lower test application times, while second approach offers less area costs. The ΔT is given as the change in total application time of modified wrapper cell with respect to flat core scheduling which is 0 to 2 percent. So with modified wrapper cells hierarchical cores can be tested with minimum test application time.

Power optimization is required in test scheduling so in Larsson et al. [6] the concurrent test application leads to higher activity during the testing, hence the power consumption is higher. The power consumed during concurrent testing is higher than normal operation in order to maximize the number of tested faults in a minimal time. A system under test can be damaged so the power constraint must be considered. In this paper three level power model is proposed i.e. system, power grid and core. The advantage is that the system level power budget is met and hotspots can be avoided at a specific core and at hotspot areas in the chip. The results from the experiment shows that by new design and test alternatives total test cost can be reduced. The proposed technique produces results that are near the ones produced by the pseudo-exhaustive technique at computational costs that are near the costs of the estimation based technique.

There are different types of algorithms and techniques used and some of them are explained in [7-10][12][13]. In Harmanani et al. [7] presented an power constrained efficient approach for the test scheduling problem of core-based systems based on genetic algorithm. The method minimizes the test application time through compact test schedules. In genetic core test scheduling formulation there is chromosomal representation, selection and reproduction, genetic operators (mutation, crossover and fill gap). During every generation, chromosomes are selected for reproduction, resulting in new test schedules. The mutation operator uses a constructive approach that minimizes the generation unfeasible test schedules. Ahn et al. [8] a SoC test scheduling method based on an ant colony optimization algorithm. The algorithm formulates the SoC test scheduling problem as a rectangle bin packing problem and uses ACO to cover more solution space to increase the probability of finding optimal solutions. Before beginning the scheduling there is need to design the test wrappers for embedded cores and found the Pareto-optimal. In [9] genetic algorithm based approach is considered for TAM optimization. Different data rates for ATE channels are used to reduce the test time. Ant colony optimization algorithm based approach is considered in [10]. This is a technique to find good paths through graphs. In [11] and [14] temperature constraint is considered for test scheduling.

III. PROPOSED ALGORITHM

The SoC has three types of cores are combinational cores, sequential cores and embedded memory cores. The core which, is Built in self tested assumed to have one unit time. The no of inputs of individual core is called the bandwidth of the core. The total bandwidth is the limited test access mechanism busses available. The total power is the power available for testing for the SoC.

Using greedy schedule we may not get the minimum test time so the schedule is heuristically improved for time minimization. The pictorial representation of a schedule is given in Fig. 1. The y-axis of each rectangle represents the bandwidth of a particular core and the x-axis represents the test time for that core. The maximum power and maximum bandwidth in Fig. 1 is 12 and 10 respectively. So at a particular instance the power (p) and bandwidth should not exceed the maximum value. The cores should be closely bounded to get the minimum test time. The cores with bandwidth and power lesser than the total bandwidth and total power can only be tested using this algorithm.

The algorithm greedily arranges the cores with respect to their bandwidth and schedules the cores with given total bandwidth and total power. The total test time (TTT) can be calculated and stored. Then another schedule is formed by re-arranging the cores e.g. with respect to test time of cores or power of the cores. The new TTT can be compared and the lowest TTT is considered to be the best schedule.

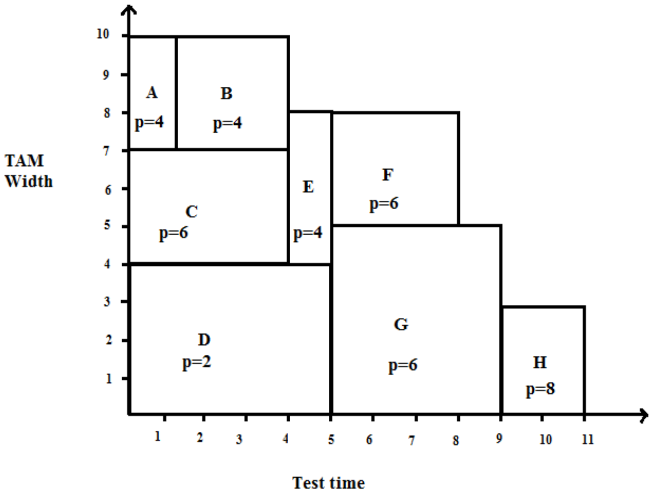

Figure 1: Representation of a schedule consisting of eight cores.

Algorithm:

INPUT:

1. N: total no of cores.

2. maxBW: total bandwidth available.

3. maxP: total power available.

4. Core set: a set of cores, for each core

(i) number of core,

(ii) bandwidth of the core,

(iii) max power consumption of the core,

(iv) test time of the core,

(v) integer u :- to check whether the core has been scheduled yet and to check whether this core has minimum time in the scheduled cores.

(vi) the start time of the core .

Output:

1. Core set: for each core

(i) the test end time of core.
(ii) Integer u value to check whether every core is tested or not.

2. total test time of the schedule.

```
BEGIN
Get the inputs N, Core set, maxBW, maxP.
Arrange the Core set with decreasing BW and decreasing P, decreasing Power and decreasing test time or any other arrangement possible.
Set the values of start time (ST) = 0; remBW = maxBW; remP = maxP; integer u = 0; integer temp = 1000000000; integer t=0;
for i=0 to N-1 do
{
  for i=0 to N-1 do
    {Select the core with integer u = 0, which has the required
    bandwidth and power from the arranged cores.
    Update the remBW, remP, integer u=1,
    start time of the core = ST and
    test time of core = ST + test time of core
    }
  for i=0 to N-1 do
    {if integer u = 1 and test time of that core < temp then
    temp = test time of the core
    integer t = number of core
    }
  for i=0 to N-1 do
   {if the remP and remBW <= power and bandwidth of the
    core and
    if integer u=1 and test time of core = temp then
   Update bandwidth and power of the core again
    Set the integer u = 2
    }
  Set integer u = 2
  Update the remBW and remP
}
 END
```

In this Algorithm the test time of the last core selected for scheduling will be updated as the total test time for the whole scheduling process.

IV. EXPERIMENTAL RESULTS

The proposed algorithm is applied to ITC'02 p93971 and p22810 which are ITC'02 benchmark SoCs. The test application time is calculated while dynamically varying the TAM sizes applied to different cores keeping the total test buswidth constant. The parameters given in Table 2 and 3 represents the module number, core bandwidth or TAM, power consumption of each core and test time of each core. Test time of the each core depends upon the scan chain width. Scan chain length is calculated by adding the number of functional inputs and the total scan chain length then dividing the TAM width required (64, 32 or 16). The test time can be calculated by using equation 1[16].

$$\text{Test Time} = (1+ \max(Si, Sout))\, TP + \min(Si, Sout) \quad (1)$$

S_i is the input scan chain length, Sout is the output scan chain length and TP is the number of test patterns. Calculating the test time of cores we can apply those values to the algorithm and the result are compared with [4]. Table 4 and Table 5 provides the results of p93971 and p22810 which represents the total test time at TAM widths 64, 80 and 128, power for p93971(P_{max}= 30000, 25000 and 10000) and power for p22810 (P_{max}= 10000, 6000 and 3000) using the algorithm. Table 6 and Table 7 represent the results for the same values in [4]. These values can be compared which gives heuristically optimal result.

Table 2: Test time calculation of p93791

Module	**Core Bandwidth (BW)**	**Power**	**Core test time**
1.	32	7014	91019
2.	16	16	768
3.	16	69	3893
4.	12	225	143
5.	32	248	42895
6.	64	6150	83219
7.	9	41	708
8.	9	41	708
9.	16	77	768
10.	32	395	13968
11.	16	862	8835
12.	32	4634	56447
13.	64	9741	29639
14.	64	9741	29639
15.	16	78	1152
16.	32	201	2376
17.	32	6674	44701
18.	16	113	294
19.	64	5252	16246
20	64	7670	50039
21.	16	113	294
22.	16	76	168
23.	64	7844	29374
24.	17	21	3072
25.	29	45	2688
26.	16	76	384
27.	64	3135	44932
28.	32	159	1584
29	64	6756	18164
30	16	77	768
31	32	218	1224
32	32	396	37008

Table 3: Test time calculation of p22810

Module	**Core Bandwidth (BW)**	**Power**	**Core test time**
1.	16	173	80
2.	16	173	445
3.	28	1238	33011
4.	16	80	61620
5.	16	64	12432
6.	32	112	666
7.	32	2489	15224
8.	32	144	2848
9.	32	148	10528
10.	16	52	7824
11.	64	2505	6687
12.	32	289	389
13.	16	739	3989
14.	32	848	2856
15.	32	487	23
16.	16	115	631
17.	32	580	645
18.	16	237	80
19.	32	442	311
20	32	441	8384
21.	32	167	412
22.	32	318	1385
23.	64	1309	9319
24.	32	260	539
25.	31	363	491
26.	32	311	279
27.	32	2512	15551
28.	64	2921	33123
29	32	413	32
30	32	508	431

Table 4: Scheduling on p93791 using proposed algorithm

TAM Width	**P_{max}= 30000 Test Time**	**P_{max}= 25000 Test Time**	**P_{max}= 10000 Test Time**
128	228718	228718	432241
80	449134	449134	493419
64	454711	454711	493419

Table 6: Scheduling on p93791 using [4].

TAM Width	**P_{max}= 30000 Test Time**	**P_{max}= 25000 Test Time**	**P_{max}= 10000 Test Time**
128	457862	493599	568734
80	787588	821475	1091210
64	945425	965383	1117385

Table 5: Scheduling on p22810 using proposed algorithm.

TAM Width	**P_{max}= 10000 Test Time**	**P_{max}= 6000 Test Time**	**P_{max}= 3000 Test Time**
128	61620	68307	96909
80	98133	98133	115194
64	127018	127018	127018

Table 7: Scheduling on p22810 using [4].

TAM Width	**P_{max}= 10000 Test Time**	**P_{max}= 5000 Test Time**	**P_{max}= 3000 Test Time**
128	128332	157568	293021
80	195733	209559	356215
64	236186	250487	309255

V. CONCLUSION AND FUTURE WORK

The proposed algorithm schedules the test of the various cores of SoC heuristically based on greedy algorithm and calculates the test time with to fixed TAM bandwidth and power constraints. The experimental results using ITC'02 SoC benchmarks show that the proposed work gives better results as compared to the one proposed by Pouget et al [4]. This algorithm can be used for large scale SoCs as it provides heuristic solution. This work can be extended for the hierarchical cores and 3D SOCs as well.

REFERENCES

[1] G. L. Craig, C. R. Kime, and K. K. Saluja, "*Test scheduling and control for VLSI built-in-self-test*", IEEE transactions on Computers, 37(9).1099-1109, September 1998.

[2] B. H. Fang, Q. Xu, and Nicola Nicolici, "*Hardware/Software Co-testing of embedded Memories in Complex SOCs*," Proceedings of the 2003 IEEE/ACM international conference on Computer-aided design, 599-605, November 2003.

[3] V. Iyenger, K. Chakrabarthy and E. J. Marinissen, "*On Using Rectangle Pack-ing for SOC Wrapper/TAM Co-optimization*," Proceedings of 20th IEEE VLSI Test Symposium (VTS'02), 253-258, May 2002.

[4] J. Pouget, E. Larsson and Z. Peng, "*Multiple-Constraint Driven System-on-Chip Test Time Optimization*," Springer Journal of Electronic Testing: Theory and Applications 21, 599-611, 2005.

[5] S. K. Goel, E. J. Marinissen, A. Sehgal and K. Chakrabarty, "*Testing of SoCs with Hierarchical Cores: Common Fallacies, Test Access Optimization and Test Scheduling*," IEEE Transactions on Computers, vol. 58, no. 3, March 2009.

[6] E. Larsson and Z. Peng, "*Power–Aware Test Planning in the Early System-on-Chip - Design Exploration Process*," IEEE Transactions on Computers, vol. 55, no. 2, February 2006.

[7] H.M. Harmanani and H.A. Salamy, "*Power-Constrained System-on-a-chip Test Scheduling using a Genetic algorithm*," Journal of Circuits, Systems, and Computers, World Scientific Publishing Company, vol.15, no. 3, (2006) 331-349.

[8] Jin-Ho Ahn and S. Kang, "*SOC Test Scheduling Algorithm Using ACO-Based Rectangle Packing*," © Springer-Verlag Berlin Heidelberg 2006, ICIC 2006, LNAI 4114, pp. 655-660, 2006.

[9] C.Giri, D.K.R. Tipparthi, S. Chattopadhyay, "*A Genetic Algorithm Based Approach for System-on-chip test Scheduling using Dual Speed TAM with Power Constraint*," WSEAS Transactions on Circuits and Systems, issue 5, volume 7, May 2008.

[10] Jin-Ho Ahn and S.Kang, "*NOC- Based SOC Test Scheduling Using Ant Colony Optimization*," ETRI Journal, vol 30, no.1, February 2008.

[11] C.Yao, K.K.Saluja, P. Ramanathan, "*Tempurature Dependent Test Scheduling for Multi-core System-on-Chip*," Asian Test Symposium , 2011.

[12] J.Shao,G.Ma,Z.Yang, "*Process Algebra Based SOC Test Scheduling for Test Time Minimization*," IEEE Computer Society Annual Symposium on VLSI, 2008.

[13] Q.Xu, N.Nicolici, "*Modular SoC Testing with Reduced Wrapper Count*," IEEE transactions on computer-aided design of integrated circuits and systems, vol. 24, no. 12, December 2005.

[14] C. Yao, K. K. Saluja, and P. Ramanathan, "*Power and Thermal Constrained Test Scheduling Under Deep Submicron Technologies*," IEEE transactions on computer-aided design of integrated circuits and systems, vol. 30, no. 2, February 2011.

[15] Y. Zorian, "*A Structured Testability Approach for Multi-Chip Modules Based on BIST and Boundary Scan*," IEEE Transactions on Components, Packaging and Manufacturing Technology – Part B, Vol-17, no. 3, August 1994.

[16] E. J. Marinissen, S. K. Goel, M. Lousberg, "*Wrapper design for Embedded Core Test*," ITC International Test Conference, 911-920, 2000.

[17] ITRS (International Technology Roadmap for Semiconductor) report 2009 Edition. http://www.itrs.net/Links/2009ITRS/Home2009.htm

[18] J. B. Im, S. Chun, G. Kim, J. H An and S. Kang, " *RAIN Scheduling Algorithm for SoC Test*,"13th Asian Test Symposium 2004.

[19] X. Chuan-pei, H. Hong-bo, N Jun-hao, "*Test Scheduling of SOC with Power Constraint Based on Particle Swarm Optimization Algorithm*," 2009 Third International Conference on Genetic and Evolutionary Computing.

[20] J. Pouget, E. Larsson, Z. Peng, M. Flottes, B. Rouzeyre, "*An Efficient Approach to SoC Wrapper Design, TAM Configuration and Test Scheduling*," Eighth IEEE European Test Workshop (ETW'03).

[21] H. M. Harmanani and R. Farah, "*Integrating Wrapper Design, TAM Assignment, and Test Scheduling for SOC Test Optimization*," 2008 IEEE.

[22] S. Koranne, "*Design of Reconfigurable Access Wrappers for Embedded Core Based SoC Test*," IEEE transactions on very large scale integretion (VLSI) systems, vol. 11, no. 5, October 2003.

[23] International Technology Roadmap for Semiconductors (ITRS), 2003, http://www.itrs.net/Links/2003ITRS/Home2003.htm

[24] Y. Zorian, "*A distributed BIST control scheme for complex VLSI devices*," VTS, pp. 6–11,1993.

[25] Y. Zorian, E. J. Marinissen and S. Dey,"*Testing Embedded-Core-Based System Chips*", IEEE Computer, 32(6),52-60, June 1999.

[26] C. Su And C. Wu," *A Graph-Based Approach to Power-Constrained SOC Test Scheduling*", Journal Of Electronic Testing: Theory And Applications 20, pp. 45–60, 2004.

[27] E. J. Marinissen, S. K. Goel, and M. Lousberg, "*Wrapper Design for Embedded Core Test*", Proceedings of International Test Conference (ITC), Atlantic City, NJ, USA, pp. 911-920. October 2000.

[28] K. Kim and K. K. Saluja," *Low-Area Wrapper Cell Design for Hierarchical SoC Testing*", Journal of Electron Test , pp. 347-352, 2009.

[29] X. Wu, Y. Chen, K. Chakrabarty , Y. Xie," *Test-access mechanism optimization for core-based three-dimensional SOCs*", Microelectronics Journal pp. 601–615, 2010.

[30] S. K. Goel, E. Marinissen," *SOC Test architecture design for efficient utilization of test bandwidth*", ACM transactions on design automation of electronic design, vol 8 ,no.4 , pp. 399-429, October 2003.

AUTHORS

First Author – **Naveen Dewan**, pursuing MTech VLSI, Department of Electronics and Communication Engineering, Thapar University, Patiala. nandydan@gmail.com

Second Author – **Harpreet Vohra**, Assistant Professor, Department of Electronics and Comminication Engineering, Thapar University, Patiala. hvohra@thapar.edu

Correspondence Author– **Naveen Dewan**, nandydan@gmail.com ,
Contact number (+918054121464)