Journal of Computer Science 6 (1): 12-17, 2010
ISSN 1549-3636

Optimal Test Time for System-on-Chip Designs using Fuzzy Logic and Process Algebra

[1]G. Rohini and [2]S. Salivahanan
[1]Department of Electronics and Instrumentation Engineering,
St. Joseph's College of Engineering, Anna University, Chennai-600 119
[2]Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam-603 110,
Anna University, Chennai

Abstract: Problem statement: Test scheduling is crucially important for optimal SoC test automation to allocate the limited available test resources. In this study we introduced a fuzzy based engine to allocate test resources. The minimized test application time can be achieved by test pipelining. However the test power consumption incurred during test procedure must be controlled in order not to offend the allowed maximal power dissipation thus avoiding damaging the system under test. **Approach:** Process algebra is the adept to deal with concurrent behaviors, based on this, the test scheduling scheme for SoC cores concurrent test is outlined by mapping the parallel test actions into concurrent processes. The algorithm for SoC test scheduling based on process algebra under multiple constraints (test power dissipation, test resources and test priorities) and apply a fuzzy based optimum search for a solution to the scheduling problem. **Results:** The test application time was calculated for three ITC-02 SOC benchmark circuits and the results were compared with other approaches. **Conclusion:** The results showed for ITC-02 benchmarks circuits prove the effectiveness of our proposed method.

Key words: System-on-chip, test access mechanism, process algebra, fuzzy logic

INTRODUCTION

In many of the earlier research studies of the test scheduling for the SoC(System-on-Chip) benchmark circuits, scheduling was done using functional bus as the medium for test vector transportation and buffers are inserted between each core to store the test vectors and applying it to each core as per the given constraints and obtained schedule. The buffer size is the hardware overhead and considered as a constraint in the test scheduling.

In this research study, the hardware overhead is not considered as a constraint. Since cores in a SoC are not directly accessible via chip inputs and outputs, special access mechanisms are required to test them at the system level. For each core in the SoC, a Test Access Mechanism (TAM) is built around each core and test vectors are applied through these TAMs. There is conceptual test access architecture for embedded cores with the source, sink and test access mechanism. The TAM is used to deliver test vector from the source to the cores and also to deliver responses from cores to the sink. Test scheduling for various widths of TAM and various number of partitions are carried out.

The general problem of SoC test integration includes the design and optimization of wrapper and TAM architectures and test scheduling. Test wrappers form the interface between cores and TAM. TAM transport test data between SoC pins and test wrappers. Test scheduling determines the order in which tests are applied. The focus is on wrapper and TAM co-design to minimize testing time under TAM width constraints.

Core based SoC reuses the pre-designed and pre-verified Intellectual Property (IP) cores to shorten design cycle and reduce design costs. In the traditional way, the designers have no access to the cores that were deeply embedded into the SoC. While the dedicated test wrappers which serve as the interfaces between the cores and SoC, facilitate the data exchange between them. In this way, complex systems can be efficiently developed. However, the complexity in the system leads to high-test data volumes. So, the design and optimization of test solution are very much important for any test. Hence the following two independent problems are considered:

- Design of an infrastructure for the transportation of test data in the system

Corresponding Author: G. Rohini, Department of Electronics and Instrumentation Engineering,
St. Joseph's College of Engineering, Anna University, Chennai-600 119

- Design of a test schedule to minimize test time

The testable units in an SoC design are the cores, the User Defined Logic (UDL) and the interconnections. The cores are usually delivered with predefined test methods and test sets, while the test sets for UDL and interconnections are to be generated prior to test scheduling and TAM Design. The workflow when developing an SoC test solution can mainly be divided into two consecutive parts namely (i) an early design space exploration and (ii) an extensive optimization of the final solution. During the process, conflicts and limitations must be carefully considered. For instance, tests may be in conflict with each other due to the sharing of test resources and power consumption. Otherwise the system may be damaged during test. Further, test resources such as external testers support a limited number of scan-chains and limited memory.

Research has been going on in developing techniques for test scheduling, TAM design and testability analysis. Larsson *et al*. (2004) has studied wrapper design or TAM optimization as independent problems. They have not addressed the issue of sizing the TAM to minimize SoC testing time. Ravi *et al*. (2001); Sehgal *et al*. (2004) proposed an approach that combine TAM design with test scheduling do not address the problem of wrapper design and its relationship to TAM optimization. Genetic algorithm for SoC test scheduling and wrapper design is addressed in (Giri *et al*., 2007). Zhao *et al*. (2004; 2005; 2007) proposed different methods to optimize SoC testing methods. Shao *et al*. (2008) proposed process algebra based SoC test scheduling does not achieve the least test application time for SoC p34932, when test TAM width is 56 or 64.

In this study we introduce scheduling optimization using fuzzy logic with process algebra based engine to achieve least test application time. Here resource allocation is performed using fuzzy logic. Process algebra is a well known tool for dealing with concurrent system. Therefore it is utilized to describe parallel core test and all the core-under-tests are mapped into the concurrent processes to achieve minimized test application time using the reconfigurable wrapper as proposed in (Koranne, 2002), with power consumption not exceeding the allowed maximal value and test conflicts being avoided.

The research related to our approach and various issues related to SoC testing and test scheduling techniques, test vector optimization and test scheduling framework are based on fuzzy logic with process algebra algorithm, the results for SoC benchmark circuits of ITC-02 are presented.

MATERIALS AND METHODS

Basics of process algebra: The term process is used to refer to behavior of a system. A system is anything showing behavior, in particular the execution of a system, the actions of a machine or even the actions of a human being. Behavior is the total of events or actions that a system can perform, the order in which they can be executed and maybe other aspects of this execution such as timing or probabilities.

Process algebra describes concurrent and distributed system algebraically. It offers a new way to handle concurrent processes. Up to date, process algebra has had many models developed. Here we discuss Communicating Sequential Processes (CSP), for SoC test scheduling, which tests more than two cores at a time compared to Calculus of Communicating Systems (CCS) which tests only two cores at a time (Baeten, 2005).

Since a core may interact with another core, parallel or distributed systems, or so-called reactive systems have to be described called concurrency theory. When dealing about process algebra, we usually consider it as an approach to concurrency theory, so process algebra will usually (but not necessarily) have parallel composition as a basic operator.

Thus, we can say that process algebra is the study of the behavior of parallel or distributed systems by algebraic means. It describes about basic operator || for parallel composition, + for alternative composition (choice) and; for sequential composition (sequencing):

- P1; P2 denotes the sequential composition of process P2 and process P1, process P2 cannot be executed until process P1 terminates successfully
- P1+P2 denotes alternative composition of process P1 and process P2, either process P1 is executed, or process P2 is executed
- P1||P2 denotes the parallel composition of process P1and process P2

In this study, we investigate the SoC test scheduling for concurrently processed cores, since the scheduling problem and process algebra have concurrency in common, process algebra is employed to deal with SoC test scheduling for minimized test application time under multiple constraints.

Fuzzy based process algebra model for soc test scheduling: Assume that the given SoC has n cores embedded into it and each core is scan structured, there are a certain number of resources available for each one. And each core accesses the TAM via individual test wrapper. The test time TestTime of a certain core is determined by the length of scan input chain of its test wrapper Si and scan output chain with the greatest length So, where:

TestTime = [1+max(Si,So)] V+max(Si,So)

where, V denotes the number of test vectors of the core-under-test. Thus the test time of each core is known and our task is to schedule n cores that are to be tested concurrently under priorities and test power constraints. For process algebra, Labeled Transition System (LTS) is widely used for description of concurrently executed processes and our semantic model described in LTS is well elaborated on how concurrent SoC core test works.

Background and notations of process algebra: LTS for SoC test scheduling can be represented as a six-tuple <S,T,F,W,s,t>, Where S = $\{s_{ij}\}(i,j\epsilon\{0,1,..., n-1\})$ is a finite set of states, where s_{ij} is the jth state of the ith core-under-test.

T = $\{t_i\}(i \epsilon \{0,1,...,n-1\})$, is a finite set of actions, $t_i \epsilon\{TI_i^j,TT_i^j,TC_i^j\}(i,j\epsilon\{0,1,...,n-1\})$, t_i can be refined into three sub-actions, respectively: inputting the jth test pattern of the ith core-under-test, using such test pattern to test the ith core-under-test and capturing the test responses from the ith core-under-test. Each action corresponds to a weight w_{ij} denoting the time required for the action t_i to work on the ith core-under-test.

$F \subseteq S\times T\times S$ is state transition function, each action in set T proceeds according to state transition function F to transfer from one state to another.

W = $\{w_{ij}\}(i,j\epsilon\{0,1,...,n-1\})$ is a finite set of weights, with weight w_{ij} that the relative edges associate denoting the test time for the jth test pattern of the ith core-under-test, s represents the initial state, t is the final state. An LTS model for concurrent n-core SoC test scheduling is shown in Fig. 1.

Optimization with fuzzy rule base: Fuzzy based test scheduling is presented here. We introduce the terms fuzzy constraints, fuzzy objectives and fuzzy decision in contrast to the conventional definitions (Bellman and Zadeh, 1970; Klein and Langholz, 1998). We also introduce the framework of decision making in a fuzzy environment. Let X be a given set of possible alternatives, which contains the solution of a decision making problem under considerations. A fuzzy goal (objective) G is a fuzzy set on X characterized by:

$$\mu_G:X\rightarrow[0,1] \quad (1)$$

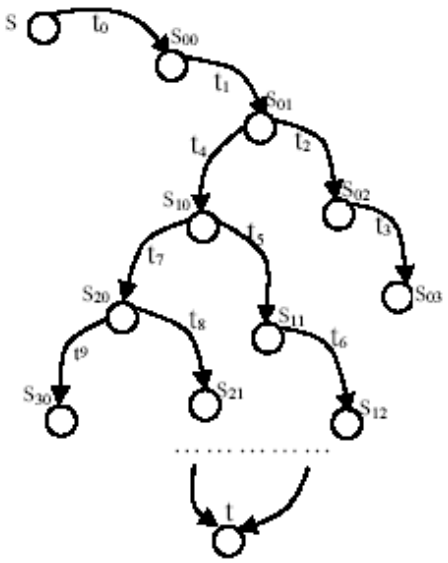

Fig. 1: LTS model for SoC concurrent test scheduling

A fuzzy constraint C is a fuzzy set on X characterized by the membership function:

$$\mu_C : X \rightarrow [0,1] \quad (2)$$

In other words, in definition of the fuzzy decision, there is no difference between the fuzzy goals and the fuzzy constraints. The introduction of fuzzy constraints requires the use of mathematical operations such as aggregation and weighting. However, the weights produced are passed on to the search engine with the assumption that the same units are used. This is not the case in normal scheduling problems. We introduce a search engine that receives the pairwise comparisons in a fuzzy way and produces a solution based only on the comparisons.

Fuzzy comparison of objectives (resource allocation): The resources available for system under test are not infinite, moreover during the test procedure two or more cores under test might contend for the same resource, or more than two tests are used to test the same core simultaneously, thus resulting in test conflicts. To make full use of the test resources available, allocate appropriate resources to each core rationally, assignment of cores based on area constraints and power constraints and to avoid test conflicts, the optimal resource allocation is done using fuzzy terminology as shown in Fig. 2.

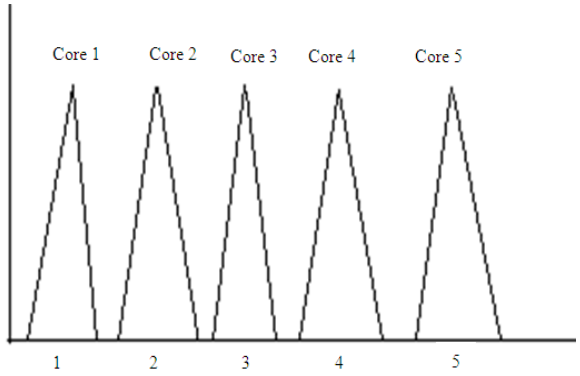

Fig. 2: Fuzzy terminology for pairwise comparison

Fuzzy based process algebra model for SoC test scheduling: Algebraically, n tasks can be viewed as n processes which are executed concurrently. P_{ij}^{m} (i.e., Fig. 2: Fuzzy terminology for pairwise comparison {0,1,...,n-1}, jϵ{0,1,..., n_j-1}, mϵT) is the process that corresponds to the behavior of the jth test pattern of the ith core-under-test, where each process is divided into three sub-actions: TI_i^j, TT_i^j, TC_i^j respectively. For instance, if process P_{12}^{1} is launched, then the second test vector of the first core, action TI_1^1 is input in a certain time and then when this action terminates, action TT_1^1 starts, if action TT_1^1 is accomplished, action TC_1^1 is carried out. In this way, three actions of one process are executed in a row.

Test pipelining can be implemented in due course to reduce test application time. Or rather, the three sub-actions of one process are executed consecutively and so are those of other processes. Sub-actions of different processes can be executed in parallel, e.g., when action TC_1^2 terminates, action TI_m^n can be launched in the meantime the next test vector j+1 of core i can be applied as long as there no test conflicts arise. Note that it is not the case that the more test vectors are pipelined, the better, for the test power incurred by concurrent operation might destroy the system under test.

The refined representation of LTS with fuzzy rule for SoC test scheduling from Fig. 1 is detailed in Fig. 3. One circle in the graph represents a state and an arc corresponds to each action and the time duration for completion of the action. The arrow head indicates the state arrived after action transition. The direct succession of each node means the concurrent execution of these actions. The weight that each arc associates is the test time that the corresponding action consumes.

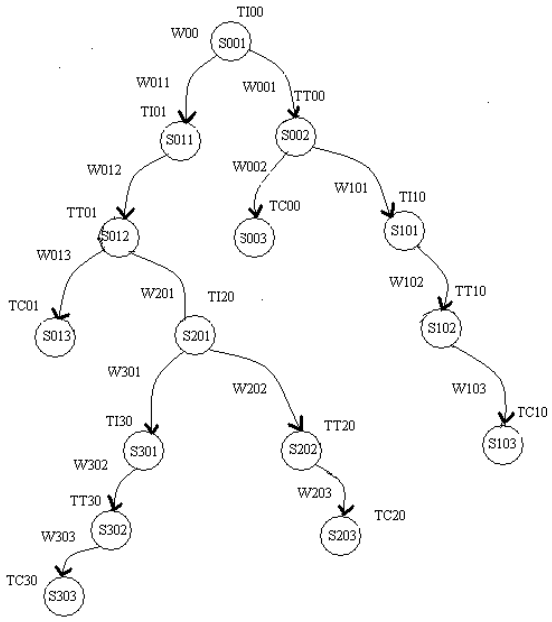

Fig. 3: Refinement LTS model for SoC parallel optimized test scheduling

Proposed algorithm for soc test scheduling: The basic problem in test scheduling is to assign a start time for all tests. In order to minimize the test application time, tests are scheduled as concurrent as possible. However, various types of constraints must be considered. A test to be scheduled consists of a set of test vectors produced or stored at a test source. The test response from the test is evaluated at a test sink. When applying a test, a test conflict may occur, which must be considered during the scheduling process. For instance, often a testable unit is tested by several test sets. If several tests are used for a testable unit, only one test can be applied to the testable unit at a time.

The test-application time can be minimized by scheduling the execution of the test sets as concurrently as possible. The basic idea in test scheduling is to determine when each test set should be executed. The main objective is to minimize the test application time.

The Process (P) is formulated in such a way that the fuzzy logic algorithm is used to optimize the solution. In the formulation of P, Number of cores (N) in SoC and number of test Buses (B) of TAM of time w_1, w_2, w_3…w_n are considered. The main objective is to determine the assignment of cores to test buses of TAM such that the assignment is used for test application for SoC and the total testing time is minimized.

RESULTS AND DISCUSSION

Experimental results for the SoC benchmark circuits of ITC-02: The experiments were conducted for the ITC-02 SoC benchmark circuits. The comparison of experimental results is illustrated as Table 1. Three types of SoCs from Philip IC are (p22810, p34932, p93791) utilized for our experiment, with each core having 30cores, 21cores and 18 cores on it respectively. Column two denotes the approach under which the experiment was performed and the test time unit is cycle. Column three gives the test application times comparison of different SoCs under four methods (in[1,8,13,14] and the proposed approach) subject to maximal test TAM and maximal power values (W_{max}/P_{max}^{-1}). Our proposed method differs from the other four in that we solved the SoC test scheduling problem shown in Fig. 4 algebraically integrated with fuzzy rule. It can be figured out from Table 1 the proposed method outperforms the classical approaches in (Larsson *et al*., 2004; Shao *et al*., 2008; Iyengar and Chakrabarty, 2002; Yoneda *et al*., 2007) on the whole. The approach given in (Shao *et al*., 2008) does not achieve the least test application time for SoC p34932, when test TAM is 56 or 64.

```
INPUT: Process P0
OUTPUT: Sequential Process with multiple
constraints considered
1. Obtain the Random Process P0
2. Set Pi = P0
3. Set Pbest = P0;
4. Find the N Feasible process (P0, P1,
   P2….PN)
5. Sort the Process such the Pij
6. If (Resource Ri is available)
     If (Priority P, Process is not violated)
     {
        Break;
        return (power exceeds the upper bound!)
     }
     Else
     {
        Execute(Pi)
     }
7. Pjn = Pi
8. Sort the new group of solution such that
   Pij.
9. Repeat the step from 6
```

Fig. 4: Algorithm for SoC test scheduling based on PA and fuzzy logic

Table 1: Test application times (#cycles) comparison with the previous proposed methods

		$W_{max}P_{max}^{-1}$						
SoC (#cores)	Different approaches	16	24	32	40	48	56	64
P22810 (30)	Present approach	1325121	1031024	781470	636510	576592	507972	413672
	Larsson *et al*. (2004)	1765647	1160128	867934	694017	592341	524183	448316
	Shao *et al*. (2008)	1725123	1131024	861470	687442	591827	511043	447250
	Iyengar and Chakrabarty, (2002)	1739018	1142471	869021	689213	592341	512462	447312
	Yoneda *et al*. (2007)	1740179	1154326	863201	688472	593149	527361	449231
P34932 (21)	Present approach	894613	646379	636531	501947	432793	406320	319786
	Larsson *et al*. (2004)	915834	656920	637126	562681	454923	422791	346105
	Shao *et al*. (2008)	915734	656379	636834	562437	454708	426579	353668
	Iyengar and Chakrabarty, (2002)	916271	657092	637014	563124	454813	437410	369483
	Yoneda *et al*. (2007)	916043	656441	636952	563024	454827	430479	346721
P93791 (18)	Present approach	416596	276569	195124	165478	154789	112457	96258
	Larsson *et al*. (2004)	432249	293928	213705	178996	152943	123824	115012
	Shao *et al*. (2008)	431389	293486	213536	178980	152847	123649	114810
	Iyengar and Chakrabarty, (2002)	432137	293567	213664	179024	152937	123731	114953
	Yoneda *et al*. (2007)	432068	293624	213894	179142	152861	124019	115301

Table 2: d695 characteristics

Core	No. of input	No. of output	Internal scan chain	Min chain length	Max chain length	No. of test pattern	Power (mW)
Model 1	32	32	0	0	0	12	14.70
Model 2	207	108	0	0	0	73	15.90
Model 3	34	1	1	32	32	75	16.60
Model 4	36	39	4	52	54	105	16.70
Model 5	38	304	32	44	45	110	16.90
Model 6	62	152	16	39	41	234	17.51
Model 7	77	150	16	33	34	95	21.30
Model 8	35	48	4	44	46	97	17.30
Model 9	35	320	32	54	54	12	15.40
Model 10	28	106	32	51	55	68	17.80

Our proposed method obtains the least test time instead, compared to other methods. Table 2 shows characteristics of d695 SoC.

CONCLUSION

We have presented an integrated approach for concurrent core test of SoC with minimized test application time. Process algebra model for SoC test scheduling is constructed and LTS gives the view of parallel core testing, refinement of actions on LTS is formulated for minimization of test application time. The pseudo-code of the algorithms for the test scheduling problem is described. The resource allocation for the scheduling is performed using fuzzy based rule. Experimental results using our method are analyzed in comparison with those previously published approaches. It demonstrates the efficacy of PA and fuzzy logic in dealing with SoC test scheduling under multiple constraints. Simulations using a real world scheduling problem indicate that the fuzzy approach is not only more intuitive but produces scheduling scenarios that are more appropriate.

REFERENCES

Baeten, J.C.M., 2005. A brief history of process algebra. Theor. Comput. Sci., 335: 131-146. http://portal.acm.org/citation.cfm?id=1085669

Bellman, R.E. and L.A. Zadeh, 1970. Decision making in a fuzzy environment. Manage. Sci., 17: 141-164. DOI: 10.1287/mnsc.17.4.B141

Giri, C., S. Sarkar and S. Chattopadhyay, 2007. Test scheduling for core-based SoCs using genetic algorithm based heuristic approach. Lecture Notes Comput. Sci., 4682: 1032-1041. DOI: 10.1007/978-3-540-74205-0_107

Iyengar, V. and K. Chakrabarty, 2002. System-on-a-chip test scheduling with precedence relationships, preemption and power constraints. IEEE Trans. Comput. Aid. Des. Integrat. Circ. Syst., 21: 1088-1094. DOI: 10.1109/TCAD.2002.801102

Klein, Y. and G. Langholz, 1998. Multi-criteria scheduling optimization using fuzzy logic. Proceeding of the IEEE International Conference on System, Man, Cybernetics, Oct. 11-14, IEEE Xplore Press, USA., pp: 445-450. DOI: 10.1109/ICSMC.1998.725452

Koranne, S., 2002. Design of reconfigurable access wrappers for embedded core based SoC test. Proceeding of the International Symposium on Quality Electronic Design, (QED'02), IEEE Xplore Press, USA., pp: 106-111. DOI: 10.1109/ISQED.2002.996707

Larsson, E., K. Arvidsson, H. Fujiwara and Z. Peng, 2004. Efficient test solutions for core based designs. IEEE Trans. Comput. Aid. Des. Integrat. Circ. Syst., 23: 758-774. DOI: 10.1109/TCAD.2004.826560

Ravi, S., L. Ganesh and N.K. Jha, 2001. Testing of core-based systems-on-a-chip. IEEE Trans. Comput. Aid. Des. Integrat. Circ. Syst., 20: 426-439. Digital Object Identifier 10.1109/43.913760

Sehgal, A., V. Iyengar and K. Chakrabarty, 2004. SoC test planning using virtual test access architectures. IEEE Trans. Very Large Scale Integrat. Syst., 12: 1263-1275. DOI: 10.1109/TVLSI.2004.834228

Shao, J., G. Ma, Z. Yang and R. Zhang, 2008. Process algebra based soc test scheduling for test time minimization. Proceeding of the IEEE Computer Society Annual Symposium on VLSI, Apr. 7-9, IEEE Xplore Press, Montpellier, pp: 134-138. DOI: 10.1109/ISVLSI.2008.88

Yoneda, T., M. Imanishi and H. Fujiwara, 2007. An SoC test scheduling algorithm using reconfigurable union wrappers. Proceeding of the Design, Automation Test Europe Conference and Exhibition, Apr. 16-20, IEEE Xplore Press, Nice, pp: 1-6. DOI: 10.1109/DATE.2007.364596

Zhao, D. and S.J. Upadhyaya, 2004. A generic resource distribution and test scheduling scheme for embedded core-based SoCs. IEEE Trans. Instru. Measur., 53: 318-329. DOI: 10.1109/TIM.2003.822712

Zhao, D. and S.J. Upadhyaya, 2005. Dynamically partitioned test scheduling with adaptive TAM configuration for power-constrained SoC testing. IEEE Trans. Comput. Aided Des. Integrat. Circ. Syst., 24: 956-965. DOI: 10.1109/TCAD.2005.847893

Zhao, D., U. Chandran and H. Fujiwara, 2007. Shelf packing to the design and optimization of a power-aware multi-frequency wrapper architecture for modular IP cores. Proceeding of the Asia and South Pacific Design Automation Conference, Jan. 23-26, IEEE Xplore Press, Yokohama, Yokohama, Japan, pp: 714-719. DOI: 10.1109/ASPDAC.2007.358071