Scheduling of Core for Reduction of Test Time in SOC

[1]Trivedi Shivangi G, [2]HareKrishna Parmar, [3] Sneha Shah

[1]M.E. Scholar, Electronics and Coimmunictaion, [1] L.J Institute of Technology, Ahmedabad, India
[2]Assistant Proffesor in C. K. Pithawala College of Eng. & Tech., Surat,India..
[3] Assistant Proffesor in L.J Institute of Technology, Ahmedabad, India

***Abstract*— Test Time minimization is crucial problem for a System on Chip (SOC). Test Time Minimization could be obtained by arranging the cores in appropriate manner. The System-on-A-Chip (SOC) revolution challenges both design and test engineers in the area of power dissipation as well as in the area of test time. Test time could be minimized through the test scheduling technique. Test scheduling is crucially important for the optimal SOC. The minimization of test time of SOC is done through Test Scheduling. This work focuses the arrangement of the External cores as well as BIST cores of SOC such that the reduction in test time could be obtained. The arrangement of core takes care of the implicit dead time as well as explicit dead time. Efficient test schedules minimize the overall system test application time and limit power dissipation during test mode. Minimization of test time is becoming increasingly important in today's systems design and is a major goal in the future development of new designs. This Paper shows how the core should be arranged to have reduction of test time in SOC.**

Keywords—SOC, BIST, External Core, Test time.

I. INTRODUCTION

Pre-designed and pre-verified intellectual property (IP) cores are being increasingly used in multifaceted system-on a-chip (SOC) design. IP cores lead to short design cycle times since a plug-and-play approach can be used to build an entire system consisting of processors, memories, and peripherals. However, testing these systems is difficult, and SOC manufacturing test is widely recognized as being a major bottleneck in the SOC design process. SOCs are constructed by system designers who purchase intellectual property (IP) circuits, known as embedded cores from core vendors and integrate them into large designs. Testing SOCs is equally challenging in the absence of standardized test structures. Hence, a number of SOC and core development working groups have been formed. The use of IP cores has led to a new paradigm in IC design. Embedded cores can be easily purchased from vendors and stitched into entire systems, thus providing rich functionality in a short design cycle time. However, because ICs are now composed of processors, memories, other IP cores, and interface logic implemented in a variety of design styles, testing each core and the system as a whole is highly challenging. Testing the cores on an SOC has added considerable complexity to the role of the system integrator, since cores are imported as "software layouts" and do not come pre-fabricated and pre-tested. An SOC test is essentially a single composite test comprised of the individual tests for every core, the user defined logic (UDL) tests, and interconnect tests. Each individual core or UDL test may involve surrounding components and may imply operational constraints (e.g., safe mode, low power mode, bypass mode) which necessitate special isolation modes. [1].

The problem with increasing test application time for testing core-based system-on-chip (SOC) designs is addressed with test architecture design and test scheduling. [2] The test power consumption is important to consider since exceeding the system's power edge might harm the system. [3] Test scheduling is tightly integrated with TAM optimization and it incorporates precedence and power constraints in the test schedule, while allowing the SOC integrator to allocate a group of tests as preempt able. [4] Test scheduling is a major problem in system-on-a-chip (SOC) test automation. We present an integrated framework that addresses several important test scheduling problems. [5]

II. SOC AND ITS TEST CHALLENGES

System-on-a-chip integrated circuits that include processors, memories, and peripheral interface devices have propelled ICs to the domain of system-level functionality. The benefits of integrating an entire system on a chip include lower cost, faster time-to-market, higher performance, and reduced power consumption. Since the single SOC requires fewer board installation steps, the cost of the SOC is lower than the total cost of the components. Increased integration reduces the number of off-chip interconnects, thereby eliminating a number of 110 peripherals and board-level buses that need to be driven. The integration of all of the system components on a single IC reduces the interconnect distance between components, which in turn decreases power dissipation and timing delays, thus contributing to higher levels of performance. Figure 1 shows a generic SOC. [1]

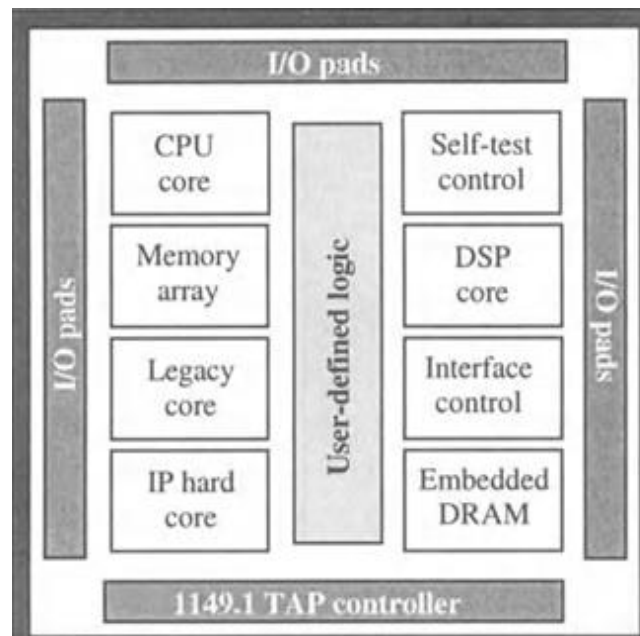

Figure 1 A Generic SOC [1]

A combination of BIST and external testing is often used so as resource conflict doesn't occur. Effective test scheduling for SOCs is challenging because it must address several conflicting goals: (a) SOC testing time minimization, (b) resource conflicts between cores arising from the use of shared TAMs and on-chip BIST engines, and (c) power dissipation constraints.

III. LITERATURE SURVEY

(a) Test Resource Partitioning:

Test Resource Partitioning for System-on-a-Chip is about test resource partitioning and optimization techniques for plug-and-play SOC test automation. Plug and play refers to the paradigm in which core to core interfaces as well as core to SOC logic interfaces are standardized. It presents new techniques for the partitioning and optimization of the three major SOC test resources: test hardware, testing time and test data volume. [6] Test resource partitioning (TRP) refers to the process of partitioning monolithic test resources, such as the test data set or the top-level TAM into subcomponents that can be optimized to achieve significant gains in test resource utilization. For example, large test data sets can be partitioned into subsets, some of which can be generated by on-chip hardware, thus reducing ATE complexity and cost. The top-level TAM can be partitioned into several sub-TAMs that fork out to test cores in parallel, thus increasing test concurrency and reducing testing time. [1]

(b) Test Scheduling:

Test scheduling is an important TRP technique that enhances the utilization of the testing time resource. This directly impact product quality and time-to-market, and is directly related to the manufacturer's economic performance. Scheduling refers to the process of determining the sequence wherein tests are to be useful to the SOC. This includes minimizing testing time by increased test parallelism, aborting the test of SOCs as soon as the first failing pattern is detected, and ensuring that power dissipation constraints are not violated during test. Test scheduling is the process to assign test resources (i.e. TAM wires) to cores at different time in order to minimize the overall test application time, while at the same time satisfying the given constraints. [7]

(c) Polynomial Time Algorithm:

The External and BIST data set are given, which are used to arrange the external cores as well as BIST cores. The external test schedule is denoted by the list s1, while s2 denotes the schedule for BIST. The symbol "+" is used to denote the concatenation operation.

The algorithm proceeds by dividing the jobs into two groups, say *A* and *B*, and it is adapted for test scheduling as follows. The jobs in *A* have $ei > bi$ while those in *B* have $ei < bi$. The schedule is built from the "middle" with jobs from *A* added on at the right and those from *B* added on at the left. (s denotes a "composite schedule" which is used later to derive schedules for external testing and BIST.) Finally, some finishing touches involving only the left-most and right-most jobs (pointed to by *l* and r, respectively) are made, and this guarantees an optimal schedule.

Considering an example having external and BIST data sets as follows:

Table 1: Test Data

Core, i	Task	
	External test data set, ei	BIST test data set, bi
1	125	100
2	200	250
3	300	200
4	200	150

The algorithm works by iterations as follows:
In the first iteration, since e1 (125) exceeds both b1 (100) and b0 (0), we (implicitly) add job 1, i.e., core 1, commencing A to the right of s (r = 1). In the second iteration, b2 (200) exceeds both e2 (200) and b0 (0), hence job 2 from B gets (implicitly) added to the left of s (l = 2). Next, in the 3rd iteration, the 1st t job is (explicitly) added to the right of s. These processes continue until all four iterations are concluded. Finally, the 0s in s are deleted and the tasks denote by l and r are added to the BIST and exterior test schedules. The schedules for exterior test and BIST are given by s1: 2134 and s2: 4213, respectively.

Table 2: Iterations of Schedule

End of Iteration	s	l	R	t1	t2
0	0	0	0	0	0
1	00	0	1	125	100
2	000	2	1	325	350
3	0001	2	3	625	550
4	00013	2	4	825	700
Finishing Touches	t1-e2>t2-b4 → s1=2+s+4=2134(delete zeros) s2=4+2+s=4213(delete zeros				

IV. RESULTS

The optimal solution can be obtained from the polynomial time algorithm. The obtained solution is as follows:

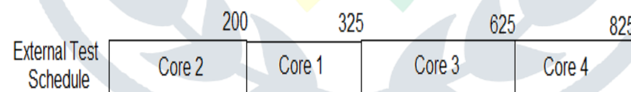

Figure 2:External Test Schedule

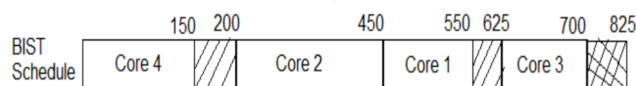

Figure 3: BIST Test Schedule

From the above references we have presented an overview that how cores of SOC could be arranged for reduction of time in SOC. Every core of SOC could be arranged using this algorithm and test time could be reduced.

V. SUMMARY AND FUTURE WORK PLAN

We have done the arrangement of core by taking an example. Every SOC cores could be arranged in an optimum schedule as an above example. Further, it needed to be in general for every SOC that is to be tested.

The next step could be performed to make the general code useful for the cores so as to have minimization of test time. In future, the coding of minimization of test time in general can be done which can give test time reduction for different SOC's. The coding could be done in MATLAB.

Acknowledgment

VI. ACKNOWLEDGEMENT

I am thankful to Mr. Harekrishna Parmar, Assistant Professor in C. K. Pithawalla college of Engineering and Technology for sharing his great knowledge with me and guiding me. I also wish to express sincere thanks and deep sense of gratitude to respected Asst. Prof. Sneha Shah, in Electronics and Communication Engineering department of L.J. Institute of Engineering and technology for her support. I am thankful to my family for their continuous encouragement to pursue higher studies and my friends for the help and support.

REFERENCES

[1] Krishnendu Chakrabarty, Vikram Iyengar, Anshuman Chandra, Department of Electrical and Computer Enginering, Duke University, Durham, NC 27708, “TEST RESOURCE PARTITIONING FOR SYSTEM-ON-A-CH I P”.

[2] Larsson, Erik, and Hideo Fujiwara. "System-on-chip test scheduling with reconfigurable core wrappers." Very Large Scale Integration (VLSI) Systems, IEEE Transactions on 14, no. 3, pp. 305-309, 2006.

[3] Pouget, Julien, Erik Larsson, and Zebo Peng. "SOC test time minimization under multiple constraints." In Test Symposium, 2003. ATS 2003. 12th Asian, pp. 312-317. IEEE, 2003.

[4] Chakrabarty, K., and E. J. Marinissen. "Test access mechanism optimization, test scheduling, and tester data volume reduction for system-on-chip."Computers, IEEE Transactions on 52, no. 12, 1619-1632, 2003.

[5] Iyengar, Vikram, and Krishnendu Chakrabarty. "Precedence-based, preemptive, and power-constrained test scheduling for system-on-a-chip." In VLSI Test Symposium, 19th IEEE Proceedings on. VTS 2001, pp. 368-374. IEEE, 2001.

[6] http://www.springer.com/engineering/circuits+%26+systems/book/978-1-4020-7119-5