Reconfigurable Multiple Scan-Chains for Reducing Test Application Time of SOCs

Jiann-Chyi Rau, Chih-Lung Chien, and Jia-Shing Ma
Department of Electrical Engineering, Tamkang University
151, Ying-Chuan Rd. Tamsui, Taipei Hsien 251, Taiwan, R.O.C
{jcrau, clchien, jsma}@ee.tku.edu.tw

Abstract

We propose an algorithm based on a framework of reconfigurable multiple scan chains for system-on-chip to minimize test application time. For the framework, the control signal combination causes the computing time increasing exponentially. The algorithm we proposed introduces a heuristic control signal selecting method to solve this problem. We also minimize the test application time by using the balancing method to assign registers into multiple scan chains. It could show significant reductions in test application times and computing times.

I. Introduction

For testing a System-on-Chip (SOC), it requires a test wrapper for each core, internal scan registers within each core, and a test access mechanism (TAM) [1]. The test wrapper is comprised of a standard cell at each core input and output that enables isolation of the core from the SOC for testing independently. The internal scan registers are designed for the necessary Design-For-Testablity (DFT) by the core providers. TAM is a mechanism to transport test data (test patterns as well as responses) and test control signals between SOC pins and core I/O and internal scan chains. The scan-based testing methodology needs high test application time because scan requires test data to be shifted in and out by one or more scan chains. The recent approaches to minimize test application time include [2], [3], and [4].

Our algorithm is based on a framework for scan chain design proposed in [5]. For the framework of Reconfigurable Multiple Scan Chains, the computing time is increasing exponentially with the number of control signals. So we propose an algorithm for the control signal selection to reduce the control signal space. Further the computing time can be reduced. We also modified the registers assignment to more balancing way to reduce the test application time.

The rest of the paper is organized as follows. In Section 2 we introduce the reconfigurable scan chain model and define the problem. The algorithm of control signal selection is presented in Section 3 and the modified registers assignment is presented in Section 4. Section 5 is the experimental results and Section 6 is Conclusion.

II. Model of Reconfigurable Scan Chain

Cores from providers are included necessary DFT to be integrated as a SOC by system integrators. The cores are prepared with internal scan chains and test vectors for each different core. The SOC integrators just saw the terminals of the I/O and the internal scan chains. That allowed integrators to insert a wrapper cell to each input and output. Further more, all the wrappers and internal scan chains would be assigned into one or several scan chains of TAM and the test vectors needed to be recombined based on the assignment. Reconfigurable Multiple Scan Chains are one kind of architectures to construct the scan chains. The following figure 1 is an example of SOC using Reconfigurable Multiple Scan Chains.

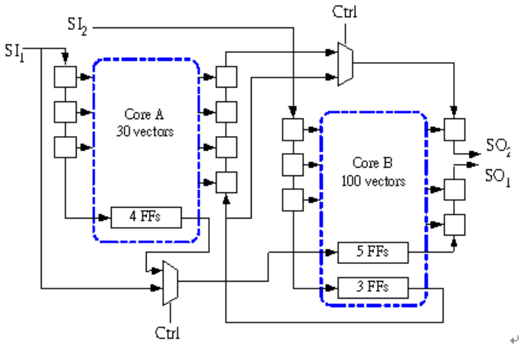


Figure 1: An example of reconfigurable scan chain design

In Figure 1, the SOC contents two cores, Core A and Core B. There are two scan chains. SC_1 contains 3 input wrappers, the internal scan chain of 4 flip-flops of Core A, the internal scan chain of 5 flip-flops and 2 output wrappers of Core B. SC_2 contains 3 input wrappers, 3 flip-flops internal scan chain of Core B, 4 output wrappers of Core A and a output wrapper of Core B. Both scan chains are reconfigurable by using the 2-to-1 multiplexers controlled by signal *Ctrl*. Two cores are tested concurrently. The scan chains are reconfigurable by the *Ctrl* signal and the multiplexers that are capable to bypass Core A.

A SOC contents many cores. Let n denote the number of cores in the SOC, each with a distinct test length, and let C = (C_1, C_2, …, C_n) denote the cores ordered in terms of strictly increasing test lengths. If two cores have the equal test lengths, they can be treated as a single core. Let L = (L_1, L_2, …, L_n) denote the test lengths in the set C. By the definition, $L_1 < L_2 < \ldots < L_n$.

In overlapped test application scheme, the test for a SOC consists a sequence of test sessions. In each session, test patterns are simultaneously applied to a subset of cores in the SOC until the test set for one core is exhausted. For an example in Figure 1, C = (C_1=core A, C_2=Core B), L = (30, 100). In the first test session, L_1 = 30 test patterns are applied to both cores. The test set for C_1 is exhausted at the end of TS_1. In the next test session, there are only L_2 — L_1 = 70 test patterns are applied to C_2. So, if there are n cores in the SOC, there are n test sessions as a test schedule (TS_1, TS_2, …, TS_n).

Let CC_i denote the chain cycle under the test session TS_i which is the minimum number of clocks required to shift in bits of a test vector in and to shift out test responses captured in the chains. Because of the control signals *Ctrl*s and the MUX, the every shift cycle CC_i for test session TS_i may not be the same. For an example in Figure 1, two cores mean two test sessions. For the TS_1, CC_1=12. After applying 30 test patterns, core A is exhausted, so next 70 test patterns would content the don't-care bits for core A if we ignore the MUXs. It would increase the test application time. If we active the Ctrl with the MUXs at the end of TS_1, all the wrappers and internal scan chains for the core A are bypass. The CC_2 would change to 6 for TS_2. That would decrease the test application time.

In the reconfigurable multiple scan chains, the control signals are defined that once a control signal is activated it remains active until the last test session and the signals could be activated at the end of test sessions only. Let $Ctrl_i$ denote the control signal activated at the end of TS_i. Once $Ctrl_i$ is activated, it is possible to bypass the registers in core C_1, C_2, …, C_i. The ideal number of control signals is n-1 which means there is a control signal activated at the end of every test sessions besides the last test session. But the more *Ctrl*s would increase the routing area since the MUXs is small. And replacing two *Ctrl*s could have one more scan chain. So the number of control signals, say t, must be limited.

The total test time τ for a given multiple scan chain configuration is the sum of each test session. The total test time is given by

$$\tau = \sum_{i=2}^{n} (L_i - L_{i-1})(CC_i + 1) + CC_1 \quad ,$$

where L0 = 0.

CC_i means the shift cycle of TS_i. If there is not having a control signal activated at the end of TS_{i-1}, CC_i for TS_i is equal to CC_{i-1} for TS_{i-1}. Since the scan chains are not reconfigured.

III. Control Signal Selection

The number of control signals, say t, is limited so that we must choose which *Ctrl* would active to make total test time minimum. Trying each choice needs a lot of computing time and computing time would increase exponentially by t. We propose an algorithm for selecting the control signals.

First we initial the Control Signal Selected Table (CSST). CSST = (CS_1=0, CS_2=0, …, CS_{n-1}=0), 0 denote not selected. For an example, the SOC with 4 cores would initial CSST = (0, 0, 0). Second we build a 1 × n matrix, named TSP, each element represent the number of test patterns for each test session which is $L_i - L_{i-1}$. Then we build another n × 1 matrix, named CSC, each element represent the minimum shift cycle for the single core with the TAM width w for the SOC. For the example, $TSP = 15 \quad 20 \quad 8 \quad 10$,

$$CSC = \begin{matrix} 12 \\ 8 \\ 16 \\ 20 \end{matrix}$$

We multiply the two matrixes as a data matrix, say M, for calculating which control signal would be chosen. In the data matrix, each element means the cycles for the session. For the example, CSC × TSP is showing in Figure 2. Based on the data matrix, we can build an array, named S, represent the cycles decreased if the *Ctrl* is chose. S = (S_1, S_2, …, S_t). The element in S is calculated as following:

$$S_i = \sum_{j=1}^{i} \sum_{k=i+1}^{n} M_{(j,k)}$$

S_m is the maximum number in S and represent for choosing $Ctrl_m$. Then we update the elements summed by S_m in M to 0 and set the $CSST_m$ to 1. After choosing the first signal, we can repeat calculating S and updating M for choosing next signal until t signals are chose. For the example in Figure 2, we assume t=2. S = (456, 360, 360). So we set $CSST_1$=1 and updating M. M after updating is showing in Figure 3. Based M after updating, S is calculated again as S = (0, 144, 240). So we set $CSST_3$=1. Two control signals are chose.

$$M = \begin{array}{c|cccc} & TS_1 & TS_2 & TS_3 & TS_4 \\ Core_1 & 180 & 240 & 96 & 120 \\ Core_2 & 120 & 160 & 64 & 80 \\ Core_3 & 240 & 320 & 128 & 160 \\ Core_4 & 300 & 400 & 160 & 200 \end{array}$$

Figure 2, the example of a data matrix.

$$M = \begin{array}{c|cccc} & TS_1 & TS_2 & TS_3 & TS_4 \\ Core_1 & 180 & 0 & 0 & 0 \\ Core_2 & 120 & 160 & 64 & 80 \\ Core_3 & 240 & 320 & 128 & 160 \\ Core_4 & 300 & 400 & 160 & 200 \end{array}$$

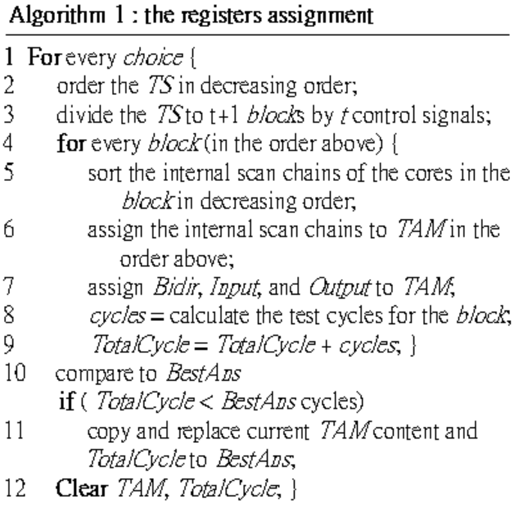
Figure 3, the updating after choosing $Ctrl_1$.

The algorithm above for control signal selection is roughly approaching the best choice. Here we propose a parameter to increasing the accuracy, say p. we would choose t + p control signals as the new control signal space. Because the number of elements in control signal space is decreasing, the computing time for all the choices is decreasing by the user defined parameter p. The table contented every choice with t control signals in the new space is built for the registers assignment. Each choice would go though the registers assignment once to find the best solution. The registers assignment is presented in next Section.

IV. Registers Assignment

The registers would be reassigned and computed test cycles for each control signal choices. For the registers assignment, the cores in the later test session would be assigned first. Because the registers in the later test session would not be bypass by Ctrl early and the patterns would be applied to the registers to the end. So the test session order set for register assignment is (TS_n, TS_{n-1}, …, TS_1). Considering the numbers of control signals, t controls would divide the test sessions into t+1 blocks. Each block may contain one or several test sessions. There is no control signal would be activated during the same block. In other words, the register assignment would not change for each test session in the same block. So we can treat the cores in the same block as a single core and assign the registers to the minimum shift cycles.

For each block, first we assign the internal scan chains into the given TAM width as decreasing orders. Next we assign bi-direct registers, inputs and output. The purpose is making the test process during the same block would be balanced. That makes the shift cycles for the block minimum and the total test application time can be decreased. The algorithm is presented in Algorithm 1. After calculate every choice, the solution for t control signals would be recorded in *BestAns*.



Algorithm 1 : the registers assignment

```
For every choice {
    order the TS in decreasing order;
    divide the TS to t+1 blocks by t control signals;
    for every block (in the order above) {
        sort the internal scan chains of the cores in the
            block in decreasing order;
        assign the internal scan chains to TAM in the
            order above;
        assign Bidir, Input, and Output to TAM;
        cycles = calculate the test cycles for the block;
        TotalCycle = TotalCycle + cycles; }
    compare to BestAns
    if ( TotalCycle < BestAns cycles)
        copy and replace current TAM content and
        TotalCycle to BestAns;
    Clear TAM, TotalCycle; }
```

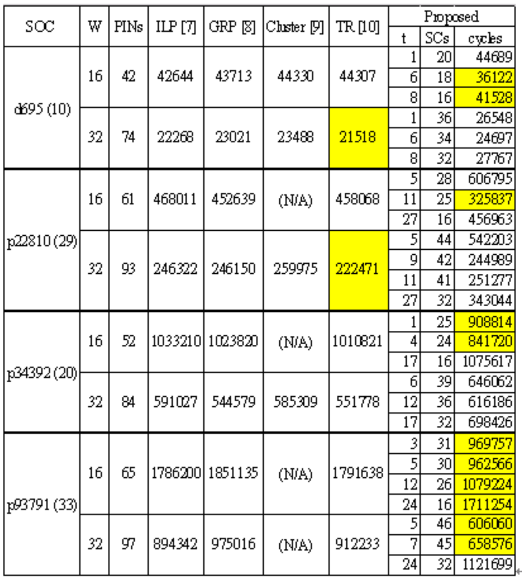
V. Experimental Results

For serial test schedules, each Core in a SOC is needed a control signal for switching TAM to each different Cores. For TAM width is 16 to a SOC with 10 cores, it needs 16×2+10=42 pins for the testing process. Based on reconfigurable multiple scan chains of the parallel test schedules, the number of control signals is limited as a constraint. For the same SOC with 10 cores, set t, the number of control signals, to 6. Then the TAM width could increase to 18. It can be said as a trade-off between the TAM width and the number of control signals.

To evaluate the proposed method we have simulated the ITC02 SOC test benchmarks [6].

In Table 1 we compare the test times of four SOC benchmarks using different test scheduling approaches: (1) the Test Bus Architecture optimization method base on ILP and exhaustive enumeration in [7], (2) the generalized rectangle-packing-based optimization (GRP) in [8], (3) the cluster-based TestRail Architecture optimization in [9], (4) a test time reduction algorithm for TestRail Architecture in [10]. The numbers after the SOC names represent the number of cores each SOC included. For example, d695(10) means there are 10 cores in SOC d695. W represents TAM width and PINs represents the total pins for the test scheduling comparing to W. For the proposed part, t is the number of control signals that is used, SCs means the number of scan chains is used after t is decided and cycles represents the test application time for the choice of t and SCs.

In the experimental results for the four SOCs, we can find a common characteristic. Our method is performed well for the situations when the SOCs are tested with the few TAM width. With the less TAM width, our method could save the control signals and changed into more scan chains. As more cores embedded in the SOC, the better performance for our method.



SOC	W	PINs	ILP [7]	GRP [8]	Cluster [9]	TR [10]	Proposed		
							t	SCs	cycles
d695 (10)	16	42	42644	43713	44330	44307	1	20	44689
							6	18	36122
							8	16	41528
	32	74	22268	23021	23488	21518	1	36	26548
							6	34	24697
							8	32	27767
p22810 (29)	16	61	468011	452639	(N/A)	458068	5	28	606795
							11	25	325837
							27	16	456963
	32	93	246322	246150	259975	222471	5	44	542203
							9	42	244989
							11	41	251277
							27	32	343044
p34392 (20)	16	52	1033210	1023820	(N/A)	1010821	1	25	908814
							4	24	841720
							17	16	1075617
	32	84	591027	544579	585309	551778	6	39	646062
							12	36	616186
							17	32	698426
p93791 (33)	16	65	1786200	1851135	(N/A)	1791638	3	31	969757
							5	30	962566
							12	26	1079224
							24	16	1711254
	32	97	894342	975016	(N/A)	912233	5	46	606060
							7	45	658576
							24	32	1121699

Table 1, Comparison of test time among different test scheduling methods

VI. Conclusions

In the paper, we have proposed an effective and efficient algorithm based on the framework of Reconfigurable Multiple Scan Chains to solve core-based SOC schedule problem. In our algorithm, the computing time is decreased by the Control Signal Selection and the Registers Assignment is simplified by the blocks divided by the control signals. The algorithm is performed well for the SOC with a large number of cores embedded and tested by few pins.

References

[1] Yervant Zorian, Erik J. Marinissen, and Sujit Dey, "Testing Embedded-Core Based System Chips", *In proceedings IEEE International Test Conference*, pp 130-134, 1998

[2] Vikram Iyegnar, Krishnendu Chakrabarty, and Erik Jan Marinissen, "Test Wrapper and Test Access Mechanism Co-Optimization for System-on-Chip", *In Proceedings IEEE International Test Conference*, pp 1023-1032, 2001.

[3] Vikram Iyegnar, Krishnendu Chakrabarty, and Erik Jan Marinissen, "Test Access Mechanism Optimization, Test Scheduling, and Tester Data Volume Reduction for System-on-Chip", *IEEE Transaction on Computers*, 52(12), pp. 1619-1631, 2003.

[4] Chih-pin Su, and Cheng-wen Wu, "A Graph-Based Approach to Power-Constrained SOC Test Scheduling", *Journal of Electronic Testing: Theory and Application* 20, 45-60, 2004.

[5] Md. Saffat Quasem, and Sandeep Gupta, "Designing Reconfigurable Multiple Scan Chains for System-on-Chip", *Proceeding of the 22nd IEEE VLSI Test Symposium (VTS 2004)*.

[6] Erik J. Marinissen, Vikram Iyegnar, and Krishnendu Chakrabarty, "ITC2002 SOC benchmarking initiative", http://www.extra.research.philips.com/itc02socbenchm.

[7] V. Iyengar, K. Chakrabarty, and E.J. Marinissen, "Efficient Wrapper/TAM Co-Optimization for Large SOCs", *in Proc. Design, Automation and Test in Europe (DATE)*, Paris, 2002, pp. 491-498.

[8] V. Iyengar, K. Chakrabarty et al. "On Using Rectangle Packing for SOC Wrapper/TAM Co-Optimization," *In Proceedings IEEE VLSI Test Symposium (VTS)*, 2002, pp. 253-258.

[9] S.K. Goel and E.J. Marinissen, "Cluster-based Test Architecture Design for System-on-Chip," *In Proceedings IEEE VLSI Test Symposium (VTS)*, 2002, pp. 259-264.

[10] S.K. Goel and E.J. Marinissen, "A Test Time Reduction Algorithm for Test Architecture Design for Core-Based System Chips," *Journal of Electronic Testing: Theory and Applications*, 2003, pp. 425-435.