Efficient Modular Testing of SOCs Using Dual-Speed TAM Architectures[1]

Anuja Sehgal and Krishnendu Chakrabarty
Department of Electrical & Computer Engineering
Duke University, Durham, NC 27708, USA
{as,krish}@ee.duke.edu

Abstract

The increasing complexity of system-on-chip (SOC) integrated circuits has spurred the development of versatile automatic test equipment (ATE) that can simultaneously drive different channels at different data rates. Examples of such ATEs include the Agilent 93000 series tester based on port scalability and the test processor-per-pin architecture, and the Tiger system from Teradyne. The number of tester channels with high data rates may be constrained in practice however due to ATE resource limitations, the power rating of the SOC, and scan frequency limits for the embedded cores. Therefore, we formulate the following optimization problem: given two available data rates for the tester channels, an SOC-level test access mechanism (TAM) width W, V ($V < W$) channels that can transport test data at the higher data rate, determine an SOC TAM architecture that minimizes the testing time. We present an efficient heuristic algorithm for TAM optimization that exploits port scalability of ATEs to reduce SOC testing time and test cost. We present experimental results on dual-speed TAM optimization for the ITC'2002 SOC test benchmarks.

1 Introduction

Recent advances in CMOS technology have led to a significant increase in the complexity of system-on-chip (SOC) integrated circuits. Today's SOCs consist of embedded cores in multiple clock domains that can often be tested at different scan clock frequencies. In order to test some of the cores at higher clock frequencies, the test data needs to be transported at a higher data rate on some selected tester channels. In order to address this problem, automatic test equipment (ATE) vendors have recently announced a new class of testers that can simultaneously drive a limited number of channels at different data rates [4]. Examples of such ATEs include the Agilent 93000 series tester based on port scalability and the test processor-per-pin architecture [1], and the Tiger system from Teradyne [2] in which the data rate can be increased for selected pin groups to match SOC test requirements. However, the number of tester channels with high data rates are constrained in practice due to ATE resource limitations, the power rating of the SOC, and scan frequency limits for the embedded cores. Optimization techniques are therefore needed to ensure that the high data-rate tester channels are efficiently used during SOC testing. In this way, high-frequency ATE channels, typically used for at-speed functional testing, can also be used to reduce the time needed for scan testing.

Modular testing of embedded cores in an SOC can simplify the complex problems of test access and application [13]. For modular testing, an embedded core is isolated from surrounding logic using a test wrapper, and a test access mechanism (TAM) is designed to deliver test data from the I/O pins of the SOC. This facilitates the reuse of precomputed tests for individual cores and partitions the SOC for test.

The problem of designing a TAM architecture and determine a test schedule to minimize the SOC testing time has been shown in the literature to be $\mathcal{NP}$-hard [7]. Therefore, a number of efficient heuristic techniques have been developed for TAM optimization [5, 6, 8, 9]. However, in all these methods, it is assumed that at any instant in time, the ATE provides test stimuli to the SOC at a single data rate. As a result, existing optimization techniques cannot readily exploit the availability of simultaneous multiple data transfer speeds from the ATE to the SOC. In this work, we focus on the problem of designing an optimized TAM architecture that can benefit from the availability of port scalability in ATEs. As in [5, 8], we base our TAM design on a Test Bus model.

We extend the heuristic approach based on rectangle packing that was presented in [8]. The use of rectangles to model core tests was described in [6, 8]. The testing times for a core in the SOC can be represented using a set of rectangles. A set $\mathbf{R}_i$ of rectangles for Core i ($1 \leq i \leq N$, where N is the number of cores in the SOC) is determined such that the height and width of each rectangle correspond to a TAM width and the corresponding test application time for the core, respectively. The TAM optimization problem can now be formulated in terms of rectangle packing as follows: Select one rectangle from each set $\mathbf{R}_i$, $1 \leq i \leq N$, and pack the selected rectangles into a bin of fixed height, such that no two rectangles overlap, and the width to which the bin is filled is minimized. The problem formulation and heuristic solution in [8] are based on a single-speed TAM architecture; therefore they are not directly applicable to the problem of optimizing dual-speed TAM architectures being studied in this paper.

The availability of dual-speed ATEs was recently ex-

[1]This research was supported in part by the National Science Foundation under grants CCR-9875324 and CCR-0204077.

ploited in [12], where a technique was presented to match ATE channels with high data rates to core scan chain frequencies using virtual TAMs. Virtual TAMs operate at scan-chain frequencies; however, they interface with the higher-frequency ATE channels using bandwidth matching. Moreover, since the virtual TAM width is not limited by the ATE pin-count, a larger number of TAM wires can be used on the SOC, thereby leading to lower testing times. A drawback of virtual TAMs however is the need for additional TAM wires on the SOC, as well as frequency division hardware for bandwidth matching. In this paper, we reduce the hardware overhead compared to [12] by using a smaller number of on-chip TAM wires. We also use ATE channels with high data rates to directly drive SOC TAM wires, thereby obviating the need for frequency division hardware.

The rest of this paper is organized as follows. In Section 2, we define the dual-speed TAM optimization problem and formulate it in terms of rectangle packing. In Section 3, we present an efficient algorithm to optimize a dual-speed TAM architecture and to derive a test schedule that minimizes the testing time. In Section 4 we present the experimental results for three ITC'02 benchmark SOCs. Finally, we present conclusion and directions for future work in Section 5.

2 Dual-speed TAM optimization

In this section we define the dual-speed TAM optimization problem and formulate it in terms of rectangle packing.

Problem $P_{\mathrm{dual-speed}}$: Given the test data parameters for the embedded cores, total SOC-level TAM width W, a total of V available high-speed ATE channels ($V < W$), and the ratio f of the high-speed data transfer rate to the low-speed data transfer rate, determine (i) the wrapper design, TAM width and test data rate for each core, and the SOC test schedule such that (i) the total number of TAM wires utilized at any moment does not exceed W, (ii) the number of TAM wires driven at the high data rate does not exceed V, and (iii) the SOC testing time is minimized.

The test set parameters for each core include the number of primary inputs, primary outputs, bidirectional I/Os, test patterns, scan chains, and scan chain lengths. The cores are assumed to be hard cores, i.e, the number and length of scan chains are fixed.

For a given TAM width and wrapper design for a core, we assume that its testing time at the high data rate is f times less than its testing time at the low data rate. In other words, if it takes $T_{ih}(w_i)$ seconds to test Core i at the high data rate with a TAM width w_i, the time taken to test it at the low data rate is $T_{il}(w_i)$ seconds, where $T_{il}(w_i) = f \times T_{ih}(w_i)$. A core vendor can mandate an upper limit on the scan test frequency for a core, and if this upper limit is lower than the higher data rate, the core can only be tested at the lower data rate. Otherwise, a core can be tested using either the high-speed data channels or the low-speed data channels. We assume that a core is not connected to the ATE by both high-speed and low-speed data channels during scan testing, i.e., it is not possible to assign both high-speed and low-speed TAM wires to a core. While a higher data rate for a given TAM width always leads to reduced testing time for a core, higher data rate for a subset of TAM wires can lead to a smaller TAM width that is available for a core. The reduced TAM width for the core can lead to an increase in its testing time, despite the faster test data rate. As a result, an optimization procedure as described in this paper is needed to either select appropriate values of f and V, or determine an efficient TAM architecture for given values of f and V.

Recall that the height of a rectangle for a core represents the TAM width assigned to that core and the width of the rectangle denotes the testing time of the core for the corresponding value of the TAM width. The $Design\_Wrapper$ algorithm from [7] is used to design a wrapper and determine the testing time for a core for several possible TAM widths. These pre-calculated testing times are subsequently used in the TAM optimization procedure. Based on the heuristic approach of [8], we formulate the dual-speed TAM optimization problem as follows: Given a collection of two sets of rectangles for each core, one representing the testing times for the high data rate and the other representing the testing times for the low data rate, and two bins of fixed heights $W - V$ and V, respectively, denoting the two data transfer rates, select a rectangle for each core and pack it in the appropriate bin, such that no two rectangles overlap and the maximum of the widths of the two bins is minimized.

Let m be the number of TAM widths of interest for Core i. Let $Rl_i = \{Rl_{i1}, \ldots, Rl_{im}\}$ be the set of rectangles for Core i for the low data rate and $Rh_i = \{Rh_{i1}, \ldots, Rh_{im}\}$ be the set of rectangles for the high data rate. Let $R_i = Rl_i \cup Rh_i$. Consider two bins of unbounded width stacked on top each other; the height V of the top bin represents the V high-speed TAM wires, and the height $W - V$ of the lower bin represents the low-speed TAM wires. The optimization problem can now be formally stated as follows:

Problem P_{GRP2}: Given the collection of set of rectangles $R_1, R_2, \ldots, R_N$ for an SOC with N cores, select one rectangle $R_i^\star$ from each set R_i, pack $R_i^\star$ in the bin for high-speed (low-speed) TAM wires if $R_i^\star \in Rh_i$ ($R_i^\star \in Rl_i$), such that no two rectangles overlap and the maximum of the widths of the two bins is minimized.

P_{GRP2} reduces to the problem P_{GRP} described in [8] if $Rh_i = \emptyset, 1 \leq i \leq N$, and $V = 0$. Since P_{GRP} was shown to be $\mathcal{NP}$-hard in [8], we conclude that P_{GRP2} is also $\mathcal{NP}$-hard.

In [7], the staircase nature of testing time variation with TAM width for cores is exploited to reduce the TAM width assigned to cores to the minimal value required to achieve a specific testing time. The TAM width values for which the

testing time decreases are called the Pareto-optimal points of the core and only rectangles corresponding to the Pareto-optimal TAM width values are considered. In the TAM optimization problem addressed here, each core has a set of Pareto-optimal points for low-speed test application and the same number of Pareto-optimal points for high-speed test application.

Lower bound on testing time: In order to evaluate our heuristic approach, we determine a lower bound on the testing time for a dual-speed TAM architecture. In our lower bound and in the test scheduling approach, we do not allow the overlap of the scan-out operation for the last test pattern of a core with the scan-in operation for the first pattern of the next core on the same TAM wire. While this is feasible for a fixed-width TestRail architecture as in [5], it is difficult to implement for a flexible-width Test Bus architecture.

For a single-speed TAM architecture, the area of a bin, with the width representing total testing time T and the height representing total TAM width W, is given by $T \times W$. Each core yields a set of rectangles of different areas. The area of a rectangle representing core i being tested at TAM width w is given by $R_i(w) = T_i(w) \times w$, where $T_i(w)$ is the testing time of Core i on TAM width w. Let the area of the minimum-area rectangle for Core i be ${R_i}^{min}$, where ${R_i}^{min} = \min_i\{R_i(w)\}, 1 \leq w \leq W$. A lower bound on the testing time of a core on a TAM width w can be expressed as: $T_i(w) = \lceil(\max(ts_i, tr_i) \cdot p_i + \min(ts_i, tr_i))/w\rceil + p_i$, where ts_i is the number of test bits to be scanned into core i, tr_i is the number of test bits to be scanned out of core i, and p_i is the number of test patterns to be applied to Core i.

Now, we know that the total area of the bin cannot be less than the sum of the minimum-area rectangles of all the cores in the SOC. Thus for any bin, $T \times W \geq {R_1}^{min} + {R_2}^{min} + \ldots {R_i}^{min}$, which implies that $T \geq \sum_{i=1}^{N} {R_i}^{min}/W$. In dual-speed TAM optimization, two bins are stacked on top of each other. A core can only be assigned to one of the two bins. Let $x_{i1} = 1$ ($x_{i2} = 1$) if Core i is assigned to the upper (lower) bin. Clearly $x_{i1} + x_{i2} = 1, 1 \leq i \leq N$. In order to determine a lower bound on the SOC testing time, we define the following integer linear programming (ILP) problem: **Objective:** Minimize C subject to

(i) $C \geq T_h$; (ii) $C \geq T_l$; (iii) $T_h \geq (\sum_{i=1}^{N} {R_{ih}}^{min} \cdot x_{i1})/V$;
(iv) $T_l \geq (\sum_{i=1}^{N} {R_{il}}^{min} \cdot x_{i2})/(W - V)$;
(v) $x_{i1} + x_{i2} = 1, \quad 1 \leq i \leq N$.

The above ILP model can be solved easily to determine the lower bounds. It takes less than a second of CPU time for the benchmark SOCs. Compared to the lower bound based on the notion of a "bottleneck core" derived in [3], the above lower bound is more accurate for smaller TAM widths. However, for larger values of W, [3] provides a tighter bound in many cases. Hence, we take the maximum of the lower bound obtained from the ILP model and lower bound from [3]. Table 1 in Section 4 shows the lower bounds for various TAM widths.

Data structure *Schedule*

1. $width_l(i)$ /* preferred TAM width for Core i on the low-freq lines*/
2. $width_h(i)$ /* preferred TAM width for Core i on the high-freq lines*/
3. $width(i)$ /* TAM width assigned to Core i */
5. $begin(i)$ /* start time of Core i */
6. $end(i)$ /* end time of Core i */
7. $freq(i)$ /* boolean indicates Core$i$ is scheduled on high-freq or low-freq TAM*/
8. $scheduled(i)$ /* boolean indicates Core i has been scheduled */
9. $complete(i)$ /* boolean indicates test for Core i has completed */

Figure 1. Data structure for the test schedule.

3 Optimization procedure

In this section, we explain the heuristic procedure used to solve the $P_{\text{dual-speed}}$ problem, which was modeled as P_{GRP2} in Section 2. We extend the $TAM\_optimizer$ procedure from [8] to solve the rectangle packing problem concurrently over two bins. In the $TAM\_optimizer$ procedure, tests are scheduled depending on certain preferred TAM widths. When a core completes its test, the TAM wires being used by it are freed and are available for assignment to other cores. The goal is to assign a preferred TAM width to each core, as long as there are enough TAM lines available. We highlight the details of our algorithm that distinguish it from the $TAM\_optimizer$ procedure in the following paragraphs.

Data Structure. The TAM width and the testing time of each core are stored in a data structure, which contains information about the start time, end time, preferred TAM widths, and TAM width frequency assigned. The data structure is presented in Figure 1. This data structure is updated as the SOC test schedule is developed.

Preferred TAM widths. We first compute a collection of Pareto-optimal rectangles for both the bins. Each core has a "preferred low-frequency TAM width" and a "preferred high-frequency TAM width". The preferred TAM widths are computed as a small percentage of the maximum allowable TAM widths W_{max} and V_{max} respectively. In our algorithm we choose $W_{max} = 64$ and $V_{max} = 64$. In the $TAM\_optimizer$ procedure from [8], an input parameter p determines the preferred TAM width for each core. It is the TAM width at which a testing time of the core reaches within $p\%$ of its testing time at the maximum allowable TAM width W_{max}. This input parameter is varied from 1 to 10 and the value that results in the best solution is chosen. To account for the requirements of bottleneck cores, a input difference parameter d is chosen. If the testing time can be improved by adding a few TAM wires ($\leq d$) to the core, then the preferred TAM width is increased. In our procedure, we use the same value of p to determine the preferred TAM width for both the bins.

Assigning preferred TAM widths to cores. In the case

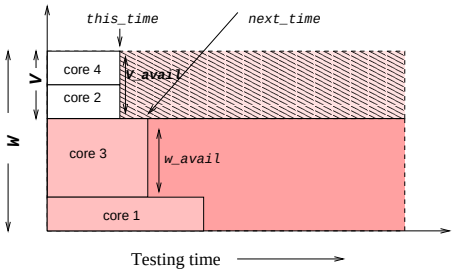

Figure 2. Test scheduling using high-speed and low-speed bins.

when both the high-speed and low-speed TAM wires are available, a core that has the highest testing time and whose preferred TAM width is less than the available TAM width is found for both the high-speed and low-speed bins. Of the two assignments, the assignment that yields a smaller end time is chosen. If only one type of TAM wires (low-speed or high-speed) is available, the core with the largest testing time and whose preferred TAM width is less than or equal to the available TAM width is chosen. On assigning a core to one of the two bins, the data structure for the core is updated. Figure 2 illustrated some of the variables used in our procedures. Two pointers, namely the parameter $this\_time$ and $next\_time$, are jointly maintained for the two bins. The parameter $this\_time$ is used to keep track of the earliest time at which TAM lines are available in either of the two bins and $next\_time$ is the minimum width to which either of the two bins are filled. These two pointers are updated as the tests for the cores are scheduled.

Minimizing idle time. The minimization of idle time is done in the same manner as in the $TAM\_optimizer$ procedure. If the available TAM widths are less than the preferred TAM width of all the cores, there might be idle spaces in the schedule since the next assignment can be made only when more TAM lines have been freed. These idle spaces may appear in both the high-speed and the low-speed bin. These spaces are minimized by assigning the freed TAM lines to a unscheduled core that will finish its test before more TAM lines are freed and will use up most of the idle time. However, the idle spaces in the two bins have to be minimized independently, since a core cannot be assigned both high-speed and low-speed TAM wires.

Redistribution of lines to fill idle time. If there does not exist a core that can fill up the idle time before the freeing up of more TAM lines, then the idle TAM lines can be redistributed among the cores that began their testing at the start of the idle time. The idle TAM lines in the low-speed bin are redistributed to the cores in that bin only and similarly idle lines in the high-speed bin can be redistributed among cores in the high-speed bin only.

In summary, the proposed heuristic procedure extends the $TAM\_optimizer$ by allowing more decisions to be made due to the availability of two bins. The time complexity of the procedure is found via experiments to be similar to that of $TAM\_optimizer$. For all the experiments, the CPU time was less than a minute.

4 Experimental results

In this section, we present experimental results on test scheduling and dual-speed TAM optimization for the three largest SOCs (in terms of the number of cores) from the *ITC'02 SOC Test Benchmarks* [11]. We first study the testing time reduction obtained with different sized high-speed bins. The testing time is calculated in μs, the low-speed TAM lines are assumed to be driven at 20MHz and the high-speed TAM lines are assumed to be driven at $20f$ MHz for different values of f. In Table 1, we present the testing time and lower bounds for various values of TAM widths W and a range of values for V, the number of TAM wires driven by the high-speed ATE channels. We also assume in this set of experiments that the high-speed data rate is twice that of the low-speed data rate, i.e., $f = 2$. The percentage change in testing time $\Delta T_n(\%)$ is calculated as follows: $\frac{T_n - T_0}{T_n} \times 100$, where T_n represents the testing time when $n\%$ ($n = V/W \times 100$) of the TAM width is made up of high-speed channels, and T_0 represents the base case when no high-speed channels are used. We vary W from 16 to 64 in steps of 8, and we also consider five different values of n.

As expected, the reduction in the testing time is in many cases proportional to the number of high-speed TAM wires. In addition, the testing time is often close to the lower bound derived in Section 2. Note however that since an increase in the number of high-speed TAM wires leads to a reduction in the number of TAM wires available per core, the decrease in testing time obtained with the dual-speed TAM architecture is not always proportional to the fraction of TAM lines that transport test data at the higher rate. A dual-speed TAM optimization procedure as presented in this paper helps the system integrator determine values of f and n for which the testing time reduction is especially noteworthy. For example, for p93791, the testing time for Core 5 with $f = 1$ and available TAM width of 23 bits is 11398.9 μs. With $f = 2$ but available TAM width of 10 bits (due to a smaller bin), the testing time increases to 14026.9 μs.

We make the unexpected observation that for smaller values of W and n, the testing time is sometimes higher for the dual-speed TAM architecture. The smaller sizes of the bins constrains the heuristic procedure to select small TAM widths for the cores; this leads to higher testing time for the SOC. The reduction in the testing time due to the higher data rate is not sufficient to outweigh the increase in testing time due to smaller TAM widths for the cores.

The results for SOC p34392 in Table 1 are especially interesting. This SOC is known to have a bottleneck core, due to which the testing time levels out at 27228.95 μs for $W \geq 32$ using a single speed TAM architecture [8]. The dual-speed architecture allows us to overcome this "lower

bound". For $n = 25\%$, the testing time drops below 27228.95 μs at $W = 29$ and levels out at 14703.20 μs, at $W = 56$. For $n = 50\%$, the testing time levels out at 13614.47 μs, (a 50 % reduction compared to 27228.95 μs) at $W = 56$.

Next, we study the effect of varying f, while keeping n constant. In Figures 3 and 4 we vary the frequency factor f from 2 to 5 for p22810 and p34392, keeping n fixed at 50%. It can be seen that the testing times for the higher speed ratios ($f = 4, 5$) for some TAM widths are close to each other. This is because while the testing time for the high-speed bin tends to decrease with an increase in the speed ratio, the testing time for the low-speed bin tends to remain the same. In such situations, the lower-speed bin dominates the overall testing time.

An advantage of the dual-speed TAM architecture is that compared to a single-speed TAM architecture, a desired testing time for the SOC can be achieved with a number of TAM wires. Let W ($W^\star$) be the SOC-level TAM width for the single-speed (dual-speed) TAM architecture that is required to achieve a desired testing time $\mathcal{T}$. Figures 5 shows the ratio $W^\star/W$ versus $\mathcal{T}$ for p22810, for two values of n and for $f = 2$. For $n = 25\%$, the ratio $W^\star/W$ is sometimes larger than one, which implies that no benefit is obtained with the dual-speed architecture for these cases. Similar results are obtained for the other SOCs.

Finally, in Table 2, we compare the testing time obtained with the dual-speed architecture to that obtained with the virtual TAM architecture of [12] for $n = 50\%$, $n = 25\%$, and $f = 4$. We find that for a given number of ATE channels, the dual-speed TAM architecture tends to outperform the virtual-TAM architecture for p22810 and p34392. The improvement in the virtual-TAM architecture is more pronounced for larger values of W. Note that for p34392, the testing time is determined by the bottleneck core for the virtual-TAM architecture, and it levels off at 27228.95 μs. However, the use of a higher data rate for a subset of TAM widths allows us to reduce the testing time further, without resorting to additional on-chip interconnect for the virtual TAM.

5 Conclusion

We have shown how SOC testing time can be reduced through a dual-speed TAM architecture that is optimized using rectangle packing. We have also presented a lower bound on the testing time that can be achieved by using a dual-speed architecture. The testing time for the optimized TAM architecture is often close to the predicted lower bound, and the testing time is reduced significantly compared to a single-speed TAM architecture. We are currently extending this work to TAM architectures with more than two data rates. We are also studying the impact of

p22810					
	$n = 100\%$	$n = 75\%$	$n = 50\%$	$n = 25\%$	$n = 0\%$
T_n $(W = 16)$	11315.97	14652.25	17236.85	25329.25	22631.95
LB	10536.20	12041.60	14048.65	16858.50	21072.95
ΔT_n (%)	−50.00	−35.25	−23.83	11.91	0
T_n $(W = 24)$	7694.45	9987.25	10838.75	14807.95	15389.00
LB	7023.90	8027.50	9365.70	11238.80	14048.55
ΔT_n (%)	−50.00	−35.10	−29.56	−3.77	0
T_n $(W = 32)$	6153.75	6966.00	8641.20	10437.85	12307.50
LB	5267.75	6020.55	7024.15	8429.05	10536.20
ΔT_n (%)	−50.00	−43.40	−29.78	−15.19	0
T_n $(W = 40)$	4932.37	5890.00	6966.00	7914.70	9864.65
LB	4213.95	4816.20	5619.15	6743.10	8428.75
ΔT_n (%)	−50.00	−40.29	−29.38	−19.76	0
T_n $(W = 48)$	4181.35	5159.30	6069.95	6691.10	8362.80
LB	3511.55	4013.40	4682.60	5619.20	7023.90
ΔT_n (%)	−50.00	−38.30	−27.41	−19.98	0
T_n $(W = 56)$	3635.40	4393.70	5150.55	6424.55	7270.85
LB	3009.8	3009.90	4013.50	4816.35	6020.25
ΔT_n (%)	−50.00	−39.57	−29.16	−11.63	0
T_n $(W = 64)$	3423.45	3985.75	4915.35	5531.70	6847.05
LB	2633.60	3511.80	3511.80	4214.30	5267.75
ΔT_n (%)	−50.00	−41.78	−28.21	−19.21	0

p34392					
	$n = 100\%$	$n = 75\%$	$n = 50\%$	$n = 25\%$	$n = 0\%$
T_n $(W = 16)$	25595.45	33798.55	42318.60	51660.45	51191.00
LB	23322.90	15550.75	23322.90	31095.10	46639.40
ΔT_n (%)	−50.00	−33.97	−17.33	0.91	0
T_n $(W = 24)$	18021.35	22469.20	25830.20	31902.15	37971.35
LB	15550.75	13614.45	15550.75	20732.20	31095.10
ΔT_n (%)	−52.53	−40.82	−31.97	−15.98	0
T_n $(W = 32)$	13614.45	18985.65	18404.10	21813.80	27228.95
LB	13614.45	13614.45	13614.45	15550.75	27228.95
ΔT_n (%)	−50.00	−30.27	−32.40	−19.88	0
T_n $(W = 40)$	13614.45	13614.45	14703.2	17008.75	27228.95
LB	13614.45	13614.45	13614.45	13614.45	27228.95
ΔT_n (%)	−50.00	−50.00	−46.00	−37.53	0
T_n $(W = 48)$	13614.45	13614.45	14703.20	14889.85	27228.95
LB	13614.45	13614.45	13614.45	13614.45	27228.95
ΔT_n (%)	−50.00	−50.00	−50.00	−46.00	0
T_n $(W = 56)$	13614.45	13614.45	13614.45	14703.20	27228.95
LB	13614.45	13614.45	13614.45	13614.45	27228.95
ΔT_n (%)	−50.00	−50.00	−50.00	−46.00	0
T_n $(W = 64)$	13614.45	13614.45	13614.45	14703.20	27228.95
LB	13614.45	13614.45	13614.45	13614.45	27228.95
ΔT_n (%)	−50.00	−50.00	−50.00	−46.00	0

p93791					
	$n = 100\%$	$n = 75\%$	$n = 50\%$	$n = 25\%$	$n = 0\%$
T_n $(W = 16)$	46278.15	60845.35	85952.05	102033.4	92556.75
LB	43734.00	49981.85	58312.50	69975.00	81124.00
ΔT_n (%)	−50.00	−34.26	−7.13	10.23	0
T_n $(W = 24)$	31135.45	43795.65	60795.50	76184.00	62439.75
LB	29155.60	33321.05	38874.75	46649.80	54082.50
ΔT_n (%)	−50.13	−29.85	−2.63	22.01	0
T_n $(W = 32)$	24375.40	31135.45	37761.05	59911.05	48750.80
LB	21866.70	24990.55	29156.00	34987.30	40561.75
ΔT_n (%)	−50.00	−36.13	−22.54	22.89	0
T_n $(W = 40)$	19850.50	23184.00	28588.90	33035.45	39701.00
LB	17493.20	19992.30	23324.65	27989.70	32449.20
ΔT_n (%)	−50.00	−41.60	−27.98	−16.78	0
T_n $(W = 48)$	15698.10	20691.35	23665.90	28436.15	31396.70
LB	14577.55	16660.15	19437.05	23324.65	27040.85
ΔT_n (%)	−50.00	−34.09	−24.62	−9.42	0
T_n $(W = 56)$	14210.90	17915.55	20383.35	24223.55	28421.80
LB	12494.90	12495.10	16660.30	19992.45	23177.75
ΔT_n (%)	−50.00	−36.96	−28.28	−14.77	0
T_n $(W = 64)$	12782.15	15727.15	16806.55	22794.75	25564.30
LB	10932.95	14577.80	14577.80	17493.30	20280.60
ΔT_n (%)	−50.00	−36.96	−28.28	−14.77	0

$\Delta T_n : \frac{T_n - T_0}{T_0}$; LB: Lower bound;

Table 1. Testing time results for p22810, p34392 and p93791 ($f = 2$).

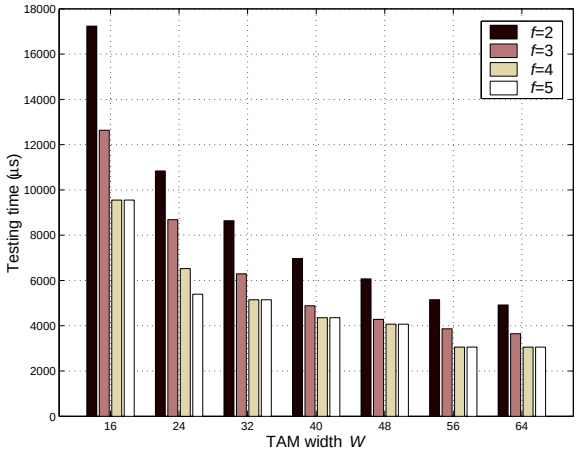

Figure 3. Effect of variation of frequency factor f on the testing time of SOC p22810 ($n = 50\%$).

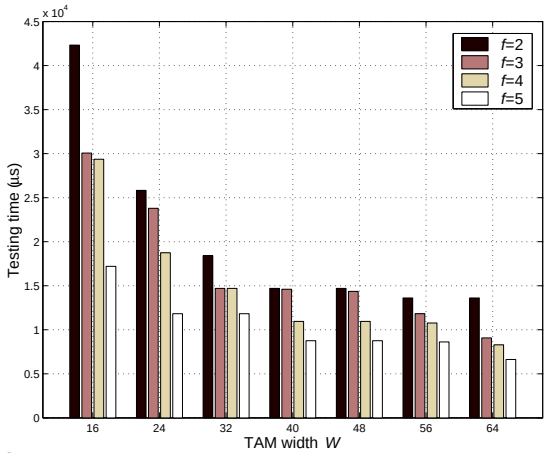

Figure 4. Effect of variation of frequency factor f on the testing time of SOC p34392 ($n = 50\%$).

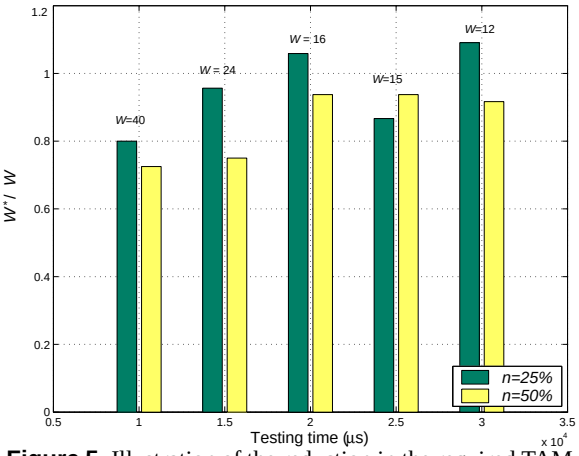

Figure 5. Illustration of the reduction in the required TAM width due to the availability of high-speed TAM wires for SOC p22810.

the higher data rate on SOC test power, and developing a power-constrained multi-speed TAM optimization technique.

SOC	W [1]	$V = W/2$	$W_{\rm virt}$ [2]	T_{virt} [3]	$T_{\rm dual-speed}$ [4]	ΔT(%)
p22810	16	8	40	9709.65	9549.75	−1.64
	32	16	80	6681.40	5148.25	−22.94
	48	24	120	5477.75	4069.95	−25.70
	64	32	160	5477.75	3055.00	−44.22
p34392	16	8	40	27228.95	29363.30	7.83
	32	16	80	27228.95	14703.20	−46.00
	48	24	120	27228.95	10950.15	−59.78
	64	32	160	27228.95	8289.90	−69.55
p93791	16	8	40	36704.25	45250.10	23.28
	32	16	80	23619.40	23583.15	−0.15
	48	24	120	13170.20	15210.95	15.49
	56	28	140	12858.65	12217.70	−4.98
	64	32	160	11179.90	12217.70	9.28

(a)

SOC	W [1]	$V = W/4$	$W_{\rm virt}$ [2]	T_{virt} [3]	$T_{\rm dual-speed}$ [4]	ΔT(%)
p22810	16	4	28	14272.50	17252.10	20.87
	32	8	56	7270.85	9426.70	29.65
	48	12	84	6601.25	5469.00	−17.15
	64	16	112	6069.65	5148.25	−15.18
p34392	16	4	28	32757.20	37887.70	15.66
	32	8	56	27228.95	15879.20	−41.68
	48	12	84	27228.95	14703.20	−46.00
	64	16	112	27228.95	14703.20	−46.00
p93791	16	4	28	52292.00	80235.15	53.43
	32	8	56	25741.25	45769.05	77.80
	40	10	70	25741.25	24183.90	−6.05
	48	12	84	20593.00	22794.75	10.69
	56	14	98	20560.25	19494.55	−5.18
	64	16	112	17428.35	17078.70	−2.00

(b)

[1] W: No. of ATE channels; [2] $W_{\rm virt}$: Virtual TAM width; [3] T_{virt}: Testing time using virtual TAMs; [4] $T_{\rm dual-speed}$: Testing time for dual-speed architecture; [5] $\Delta T = \frac{T_{\rm dual-speed} - T_{\rm virt}}{T_{\rm virt}}$.

Table 2. Comparison between virtual TAMs and the proposed dual-speed TAM architecture; (a) $n = 50\%$ (b) $n = 25\%$.

References

[1] Agilent Technologies. Winning in the SOC market, available online at: *http://cp.literature.agilent.com/litweb/pdf/5988-7344EN.pdf*

[2] Teradyne Technologies. Tiger: Advanced digital with silicon germanium technology. *http://www.teradyne.com/tiger/digital.html*

[3] K. Chakrabarty. Optimal test access architectures for system-on-a-chip. *ACM Trans. Design Automation of Electronic Systems*, vol. 6, pp. 26–49, January 2001.

[4] R. Dorsch et al. Adapting an SoC to ATE concurrent test capabilities. *Proc. Int. Test Conf.*, pp. 1169–1175, 2002.

[5] S. K. Goel and E. J. Marinissen. Effective and efficient test architecture design for SOCs. *Proc. Int. Test Conf.*, pp. 529–538, 2002.

[6] Y. Huang et al. Optimal core wrapper width selection and SOC test scheduling based on 3-D bin packing algorithm. *Proc. Int. Test Conf.*, pp. 74–82, 2002.

[7] V. Iyengar, K. Chakrabarty and E. J. Marinissen. Test wrapper and test access mechanism co-optimization for system-on-chip. *JETTA*, vol. 18, pp. 213–230, April 2002.

[8] V. Iyengar, K. Chakrabarty and E.J. Marinissen. On using rectangle packing for SOC wrapper/TAM co-optimization. *Proc. VLSI Test Symp.*, pp. 253–258, 2002.

[9] E. Larsson and Z. Peng. An integrated framework for the design and optimization of SOC test solutions. *JETTA*, vol. 18, pp. 385–400, Aug/Oct. 2002.

[10] E. J. Marinissen et al. A structured and scalable mechanism for test access to embedded reusable cores. *Proc. Int. Test Conf.*, pp. 284–293, 1998.

[11] E. J. Marinissen, V. Iyengar and K. Chakrabarty. A set of benchmarks for modular testing of SOCs. *Proc. Int. Test Conf.*, pp. 519–528, 2002.

[12] A. Sehgal et al. Test cost reduction for SOCs using virtual TAMs and Lagrange multipliers. *Proc. DAC.*, pp. 738–743, 2003.

[13] Y. Zorian, E. J. Marinissen and S. Dey. Testing embedded-core-based system chips. *IEEE Computer*, vol. 32, pp. 52–60, June 1999.