Multi-Site Test Optimization for Multi-V_{dd} SoCs Using Space- and Time- Division Multiplexing

[1,2]Fotis Vartziotis
[1]Computer Engineering Technological Educational Institute of Epirus, Greece
fvartzi@teiep.gr

Xrysovalantis Kavousianos*
[2]Computer Science and Engineering University of Ioannina, Greece
kabousia@cs.uoi.gr

Krishnendu Chakrabarty*
Electrical and Computer Engineering Duke University, USA
krish@duke.edu

Rubin Parekhji, Arvind Jain
Texas Instruments Bangalore, India
parekhji@ti.com, a-jain@ti.com,

***Abstract*—Even though system-on-chip (SoC) testing at multiple voltage settings significantly increases test complexity, the use of a different shift frequency at each voltage setting offers parallelism that can be exploited by time-division multiplexing (TDM) to reduce test length. We show that TDM is especially effective for small-bitwidth and heavily loaded test-access mechanisms (TAMs), thereby tangibly increasing the effectiveness of multi-site testing. However, TDM suffers from some inherent limitations that do not allow the fullest possible exploitation of TAM bandwidth. To overcome these limitations, we propose space-division multiplexing (SDM), which complements TDM and offers higher multi-site test efficiency. We implement space- and time-division multiplexing (STDM) using a new, scalable test-time minimization method based on a combination of bin packing and simulated annealing. Results for industrial SoCs, highlight the advantages of the proposed optimization method.**

I. Introduction

Dynamic voltage scaling (DVS) reduces dynamic power consumption during periods of light workload [7]. Many state-of-the-art processors nowadays adopt this technique [1], [2], [3]. Further power savings are achieved by partitioning the system-on-chip (SoC) into voltage islands [9], [20] where separate power management policies are applied [6], [7], [13].

Multi-V_{dd} SoCs must be tested at multiple voltage-frequency settings [5], [16], [24]. Test scheduling for multi-core, multi-V_{dd} SoCs was recently addressed in [14]. In [15] a time division multiplexing (TDM) method was proposed, that exploits the difference in shift frequencies at different voltage settings to minimize test time. Even though they are very effective in single-site test platforms, these techniques do not consider multi-site optimization parameters. Most production lines employ multi-site test processes to increase the automatic test equipment (ATE) utilization and decrease the overall time for testing a large volume of chips [22], [23], [26].

The minimization of the time needed for testing a single chip is not the primary goal in multi-site testing [12]. Instead, efficient exploitation of the ATE channels to increase the number of dies that are tested in parallel is more beneficial [11]. The number of SoCs that can be tested in parallel depends, among other parameters, on the availability of ATE channels [22], [23], [26]. For a fixed number of ATE channels, a dual objective must be pursued to maximize the ATE channel utilization, namely the minimization of both the number of ATE channels needed per SoC and the test time per chip.

In order to facilitate efficient multi-site testing, various techniques focus on the optimization of Test Access Mechanism (TAM) width [10]–[12], [18]. Other techniques reduce the SoC test length through optimization test scheduling [4], [14], [17], [27]. Many test-resource-partitioning techniques minimize test data volume, test time and the number of ATE channels [25]. Any further reduction in the number of ATE channels to facilitate a higher multi-site efficiency would make the test length per chip to step up in a conversely proportional way, eliminating, thus, any gains due to increased parallelism [11].

In this paper, we first show that multi-core/multi-V_{dd} SoCs offer increased potential for parallelism, provided that TDM is employed, and a suitable TAM width is used. Specifically, we show that TDM is especially effective for small-bitwidth and heavily loaded test-access mechanisms (TAMs), thereby tangibly increasing the effectiveness of multi-site testing. However, inherent limitations of TDM do not allow the fullest possible exploitation of TAM bandwidth. To overcome these limitations, we propose space-division multiplexing (SDM), which complements TDM. Space- and time-division multiplexing (STDM) is a unified solution for multi-core/multi-V_{dd} SoCs that combines SDM and TDM and offers very high multi-site test efficiency. It is implemented using a new, scalable test-time minimization method based on a combination of bin packing and simulated annealing. Results for industrial SoCs highlight the advantages of the proposed optimization method.

II. Background & Motivation

Test scheduling for multi-core/multi-V_{dd} SoCs is considerably more challenging than test scheduling for single-V_{dd} designs [14]. Besides the repetitive testing at multiple voltage levels, DVS and voltage islands impose: (a) dependencies between voltage islands due to TAMs spanning these islands, (b) dependencies between cores of the same island that share the same power network, (c) the use of low shift frequency at lower power-supply voltages, and (d) increased difficulty

*This research has been co-financed by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: Thales. Investing in knowledge society through the European Social Fund.

978-3-9815370-2-4/DATE14/©2014 EDAA

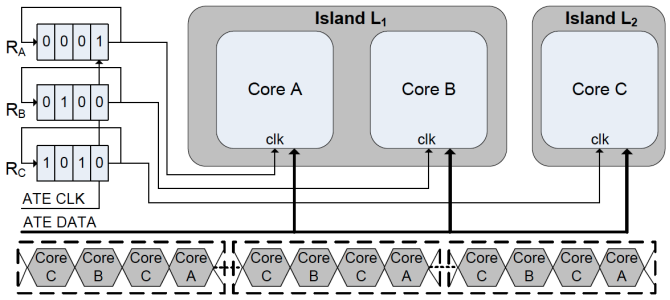

Fig. 1: The TDM scheme proposed in [15].

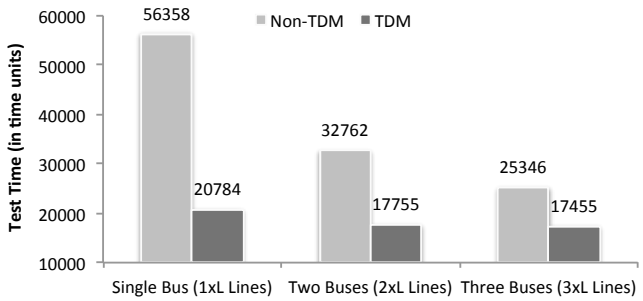

Fig. 2: Comparisons between TDM and non-TDM schemes.

to concurrently provide multiple clocks with different shift frequencies to test cores at different voltage levels [15].

To overcome these limitations the TDM approach was proposed in [15]. By using TDM, test data for different cores are multiplexed in time on the ATE channels, and they are transmitted by the ATE at the highest frequency supported by the SoC. At the SoC-end, test data are demultiplexed and shifted into the corresponding cores at the frequencies dictated by the voltage settings used. An example is shown in Fig. 1. Cores A, B, C share the bus, and the ATE clock has frequency F, which is divided using one circular register per core loaded with non-overlapping patterns. The test data of Cores A, B and C are transmitted on the bus aligned with the clocks generated by the registers, in a loop consisting of four time-slots as shown in Fig. 1. Core C is loaded from the ATE with frequency $F/2$, and Cores A, B are loaded with frequency $F/4$.

TDM is very effective for single-site testing as it minimizes the test time per chip. However, multi-site testing is widely employed in large production lines as it is more efficient than single-site testing [22], [23], [26]. For a given set of ATE channels, the degree of parallelism in multi-site testing is based on the number of ATE channels used per SoC. Therefore, minimization of the number of ATE channels required per SoC directly leads to the maximization of the number of sites tested under certain bounds set by the number of sockets available (in final testing) or by probe card limitations (in wafer testing) within the constraints of the tester load board.

Test data compression, TAM optimization, and test scheduling optimization are used synergistically to reduce both the test length and the required number of ATE channels. Any attempt to further reduce the size of the TAM beyond this point would only lengthen proportionally test time, thus cancelling any parallelization benefits [11]. TDM overcomes this limitation by exploiting the bandwidth of the ATE channels that is left unutilized due to the low shift frequencies used at the lower voltage settings. However, the efficiency of TDM depends on the availability of tests that can be concurrently executed using the same TAM resource. Note that multiple islands and voltage settings impose many constraints that restrict the set of tests that can be concurrently applied [14]. Therefore, in order to boost the performance of TDM, each TAM resource must be connected to a sufficiently large number of cores. Despite being counterintuitive, this unconventional technique is very effective in TDM, as it increases the likelihood that tests can be applied in parallel. Conventional test scheduling techniques, which apply tests in a sequential manner, avoid heavily loaded TAM resources, because they prolong test application time.

In order to highlight this trend, we run simulations using an industrial SoC described in Section V. We simulated three different bus architectures for this SoC with one, two and three identical (in size) buses. In the three-buses configuration, each bus was connected to an appropriately selected subset of cores such that the test-time-load for each bus was almost the same. The load of every bus was calculated as the aggregate time of the tests for the cores connected to the bus. The two-bus configuration was generated from the three-bus configuration by removing one randomly selected bus. The set of cores connected to this bus was partitioned into two subsets, such that the aggregate time of the tests for the cores in each subset was almost the same. The cores of one subset were connected to one of the two remaining buses, and the cores of the other subset were connected to the second bus.

Fig. 2 presents a comparison between TDM and the test-scheduling method of [14] that is also suitable for multi-core/multi-V_{dd} SoCs (denoted hereafter as non-TDM). The x-axis presents the three configurations, where each bus consists of L TAM lines, and the y-axis presents the test time per device (bus widths and test times are given in normalized units to conceal confidential information about this SoC). As we decrease the TAM width, the non-TDM approach worsens considerably in terms of test time. In particular, when one of the two buses is removed (all the cores are connected to the remaining bus), the test time almost doubles, thus offering no gain in multi-site efficiency. In contrast, the test time for the TDM approach increases only slightly as we reduce the number of buses —14.5% per device in the worst case. At the same time, TDM leads to a considerable increase in the number of sites that can be tested in parallel.

Unfortunately, much of the available TAM bandwidth cannot be exploited by TDM, due to the large number of constraints in multi-core/multi-V_{dd} SoCs that prevent tests from being applied in parallel. In addition, it is possible that different cores in an SoC will have different wrapper parallel port (WPPs) widths. This may happen when hard IP cores are embedded into the SoC or when test decompressors are employed that impose the use of a specific number of inputs channels to provide optimal results (e.g., linear based decompressors fed by external channels in a continuous flow

manner [21]). Any bus lines left unutilized by a core cannot be exploited unless fork and join techniques are employed, which introduce additional constraints and increased routing overhead. In order to overcome these limitations, we propose a new DFT architecture, which is referred to as Space-Division-Multiplexing (SDM). SDM complements TDM and offers a unified approach for testing multi-core/multi-V_{dd} SoCs.

III. Space-Division Multiplexing

The objective of SDM, is twofold: (a) enable the use of narrow TAMs to increase multi-site efficiency; (b) overcome mismatch between the WPP size and the size of the bus, and increase the parallelization efficiency of TDM. SDM inserts an interface between the bus and the wrapper whenever their widths do not match. Using SDM, a wide (narrow) bus can be efficiently used to serve a narrow (wide) wrapper interface.

The basic concept of SDM is as follows. Test data at the input of the wrapper are first latched and when they match the width of WPP, they are shifted into the wrapper. SDM considers two cases: when the bus is (a) wider, and (b) narrower than the WPP. In the first case, the test data are latched into an interface register in one clock cycle, and then they are disaggregated and loaded into the wrapper in multiple cycles. In the second case, the test data are latched into the register in multiple cycles, and when they match the width of the WPP, they are transferred to the core in a single cycle.

In Case (a) the bus transfers test data at a faster rate than the wrapper can consume. SDM eliminates this gap by enabling low frequency to transmit test data over the bus, and high frequency to transfer the data to the core. Therefore, TAM channels, which would otherwise be left unutilized, are used to reduce the frequency at which test data are transported through the TAM. If TDM is combined with SDM, it can exploit the released frequency to pack more tests in parallel. In other words, Space- and Time- Division Multiplexing (STDM) reduces test time further, by trading-off the number of ATE channels with the frequency at which test data is transferred. In Case (b), the bus transfers smaller amounts of data than the wrapper can consume. In this case, SDM offers the means to increase the frequency at which test data are transferred over the bus in order to shorten the test length.

Example 1. Fig. 3 presents the SoC of Example 1 with WPP sizes equal to 8, 4, 16 bits for Cores A, B, C respectively. Let the size of the bus be 8 bits and the maximum scan frequencies at voltage settings V_1, V_2, V_3 be $F, F/2$ and $F/4$, respectively. At V_2, all cores must be loaded using frequency $F/2$ or lower. Core A receives the full bandwidth at $F/2$ since the width of the WPP exactly matches the width of the bus. In the case of Core B, the bus is loaded with 8 bits at frequency $F/4$, and the WPP with 4 bits at frequency $F/2$. Therefore, instead of using the half of the bus at frequency $F/2$, SDM uses the complete bus at frequency $F/4$ and the test length remains the same. In case of Core C, the frequency for loading the bus increases to F, while the frequency for loading the wrapper remains at $F/2$. As a result, the test length is reduced by half. ■

SDM complements TDM, and the unified STDM scheme fully utilizes the available capacity of the ATE channels. As shown in Section V, STDM not only exploits unutilized TAM lines, but it also offers high multi-site efficiency even when there are no unutilized TAM lines. This is accomplished by enabling the use of buses that are narrower than the wrappers, with a small additional overhead in test time per device.

The block diagram of STDM for the example SoC is shown in Fig. 3. Every core is assigned a small set of registers and a small amount of control logic (not shown in Fig. 3 for simplicity and because the hardware overhead is negligible). Three different types of registers are involved:

Interface Register IR: It stores test data from the bus and loads test data into WPP (used only for cores B, C that need WPP width adjustment). For Core B, IR_B is 8-bits long and it is loaded in one clock cycle from the bus. The most and least significant nibbles of IR_B are multiplexed at the output of register IR_B and they load WPP in two clock cycles. Register IR_C is 16-bits long; it is loaded in two clock cycles from the bus, and it loads the 16-bit WPP of core C in a single cycle.

Bus Control Register BCR: This circular register divides the ATE clock according to a pre-loaded pattern. As the pattern rotates inside BCR, the rightmost cell receives periodically the value of '1' and triggers the loading of the data from the bus directly into WPP (Core A) or into IR (Cores B, C).

Wrapper Control Register WCR: This is similar to BCR and triggers the loading of IR into the wrapper.

Example 2. Let the frequency of the ATE clock for the SoC of Fig. 3 be F. Core A is shifted using frequency $F/2$, as the pattern loaded into BCR_A triggers CLK_A every two clock cycles. Register IR_B is loaded with frequency $F/4$ from the bus. Note that BCR_B triggers IR_B once every four cycles. The register IR_B transfers data to the WPP with frequency $F/2$. Note that WCR_B, which controls the loading of IR_B into the WPP, triggers the wrapper once every two cycles. Finally, register IR_C is loaded with frequency $F/4$ from the bus — BCR_C triggers IR_C once every four cycles. Register IR_C loads the WPP with frequency $F/8$. The patterns loaded into BCR_A, BCR_B, BCR_C have no-overlapping values of '1' to ensure mutually exclusive use of the bus. ■

IV. Test Scheduling Method

We consider a multi-core SoC with I islands $L_1, \ldots, L_I$, N wrapped cores $C_1, \ldots, C_N$ ($N \geq I$), and M voltage levels $V_1 > \cdots > V_M$. Each core (and island) may be tested at all or at a subset of these voltage levels, using one out of S independent buses. Each voltage level V_m is associated with one shift frequency F_m, which is the maximum rated frequency that can be used for shifting test data at V_m. Note that frequency F_m may differ from core to core, and it is usually quantized to a pre-specified set of frequencies. Overall we assume that M shift frequencies $F_1 > \cdots > F_M$ are supported, and every core with maximum (quantized) shift frequency equal to F_m at voltage V_m can use any of the frequencies $F_m, F_{m+1}, \ldots, F_M$ for loading test data.

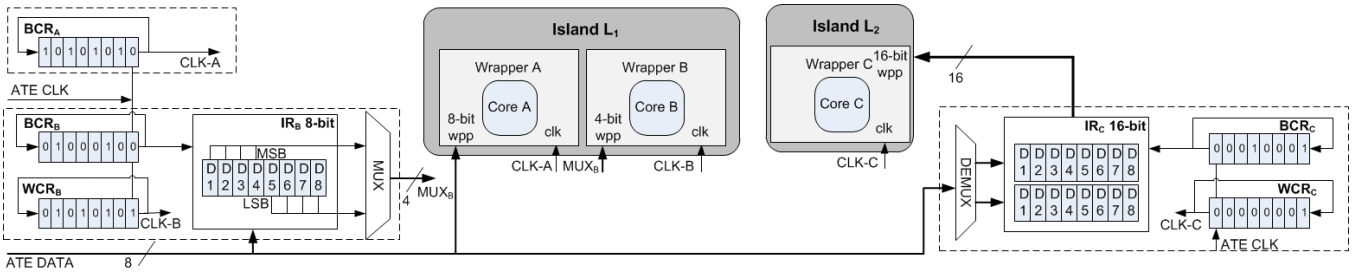

Fig. 3: Proposed STDM scheme.

Let T_{C_j,V_m,F_i} be the test time needed to test core C_j at voltage V_m using shift frequency F_i ($i \geq m$, where F_m is the highest shift frequency at V_m). T_{C_j,V_m,F_i}, depends on the wrapper depth $WD(C_j)$, on the number $Patts(C_j, V_m)$ of test patterns applied at V_m, and on F_i. When the WPP width W is equal to the size B of the bus, T_{C_j,V_m,F_i} is approximated using the formula $T_{C_j,V_m,F_i} = Patts(C_j, V_m) \cdot WD(C_j)/F_i$. When the ratio $k = B/W$ of core C_j is equal to $2, 3, \ldots$ or $1/2, 1/3, \ldots$, SDM is employed, and the test time when the bus is used at frequency F_i is equal to $T_{C_j,V_m,F_i} = Patts(C_j, V_m) \cdot WD(C_j)/(k \cdot F_i)$. The frequency for shifting test data into the core is equal to $k \cdot F_i$, therefore, only the values of i that fulfill the relation $k \cdot F_i \leq F_m$ are used.

Every frequency F_i used for shifting test data into core C_j at any voltage level V_m defines an alternative test, which has a unique length and frequency allocation on the bus. The TDM algorithm selects and schedules the best one for every core-voltage pair, based on the available resources and the inter-core constraints. We have developed a heuristic based on Bin Packing (BP) and Simulated Annealing (SA). Each test with test time T_{C_j,V_m,F_i} is modeled as a rectangle with height equal to T_{C_j,V_m,F_i} and width equal to F_i. Next, a sequence of such rectangles, one for every core-voltage pair, is packed into S virtual bins (S is the number of buses), so that the overall height, i.e. test schedule length TL, is minimum. Each virtual bin has width equal to F_1, which corresponds to the maximum supported scan frequency used for shifting test data.

The heuristic used to implement the packing is based on the Bottom-Left rule [8]. We place each rectangle to the position that (a) fulfills the Multi-V_{dd} testing constraints posed in [15], (b) the y-coordinate of the top side of the rectangle is the smallest. If there are several such valid positions, we select the one that has the smallest x-coordinate value. The BP method is combined with an SA heuristic to further optimize its performance. SA uses as energy function $E(s)$ the test schedule length produced by BP. To select the *next state* in the SA, n randomly selected tests from the current list of scheduled tests are substituted so that a new acceptable schedule of tests can be created. The SA acceptance probability function is derived from the Maxwell-Boltzmann distribution [19]. Both the initial SA temperature and cooling rate are empirically defined to be a constant value.

Example 3. Fig. 4 shows the TDM and STDM test schedules for the SoC shown in Fig. 3. The x-axis of each bin presents the maximum shift frequency of 200 MHz, and the y-axis

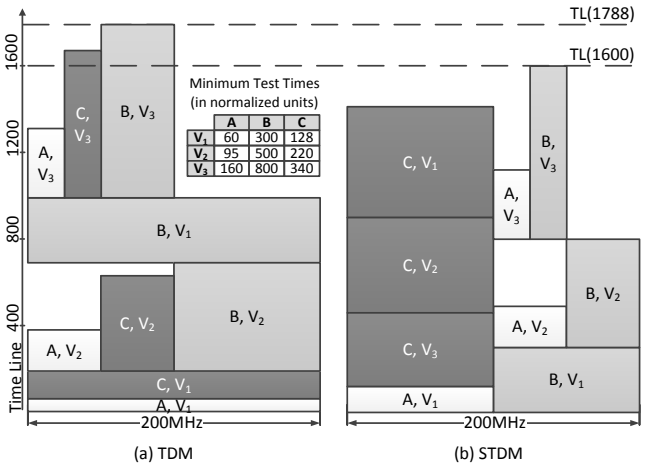

Fig. 4: TDM and STDM test schedules.

presents the test time. The maximum rated shift frequency at V_1, V_2, V_3 is equal to 200, 100 and 50 MHz respectively. The test times at V_1, V_2, V_3 using the maximum rated shift frequency at each voltage setting are shown in the embedded table. The test times at V_1, V_2, V_3 using lower shift frequencies can be derived directly from this table. For example, the test time of Core A at V_1 is 60×1, 60×2 and 60×4 units when 200, 100 and 50 MHz frequency is used respectively. In the TDM case all cores use 200 MHz at voltage level V_1 (all rectangles extend to the full width of the x-axis). At voltage levels V_2, V_3 Core B uses 100 MHz and 50 MHz respectively. At the same voltage levels Core C uses 50 MHz and 25 MHz respectively. In STDM, the tests for Core A retain the same characteristics at V_2 and V_3, while at V_1, the test data are shifted in half of the frequency that was used in TDM. The height of the rectangle is double and the width is half compared to that of TDM. In the case of Core B where width adjustment is used, all tests retain the same time with TDM, even at half the frequency (same height at half the width). In the case of Core C, the frequency used for shifting the test data into the WPP is equal to 50 MHz in all cases, and the frequency for loading the IR_C from the bus is equal to 100 MHz. It is obvious that STDM exploits many different options for sharing the bus and thus it better exploits the bandwidth provided by the ATE. ■

V. Experimental Results

For evaluation purposes, we used an industrial SoC from Texas Instruments that is targeted for portable wireless applica-

TABLE I: SoC Minimum Test Times (in Normalized Time Units) & Maximum Scan Frequencies F^m (in MHz).

V_{dd}	C_1	C_2	F^m	C_3	C_4	C_5	F^m	C_6	C_7	C_8	F^m	C_9	C_{10}	F^m	C_{11}	C_{12}	F^m	C_{13}	C_{14}	C_{15}
V_1	900	300	400	700	100	550	200	N/A	188	600	266	700	N/A	200	N/A	165	300	500	125	1300
V_2	1200	396	300	1400	200	1100	100	475	370	950	200	924	198	150	900	192	200	N/A	N/A	N/A
V_3	1350	450	266	2800	400	2200	50	700	500	1600	100	1050	225	133	N/A	250	150	N/A	N/A	N/A
V_4	1800	600	200	N/A	N/A	N/A	N/A	1400	1000	3200	50	1400	300	100	N/A	500	75	N/A	N/A	N/A
V_5	3600	N/A	100	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	2800	N/A	50	N/A	1000	38	N/A	N/A	N/A
V_6	7200	N/A	50	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	5600	N/A	25	N/A	N/A	N/A	N/A	N/A	N/A

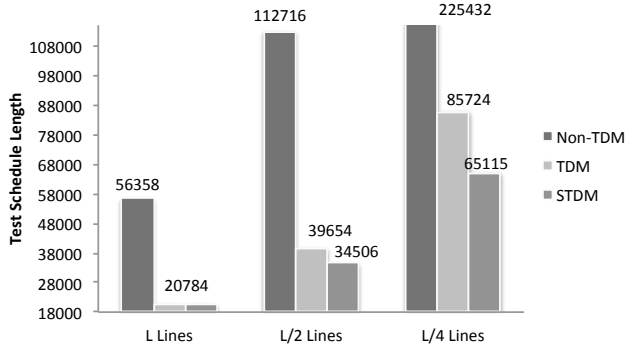

Fig. 5: Results comparing STDM with TDM and Non-TDM..

tions. The SoC has 7 islands $I_1, \ldots, I_7$, 15 cores $C_1, \ldots, C_{15}$ and 225 clock domains. It can be set to up to 6 voltage settings $V_1, V_2, \ldots, V_6$, each one supporting a different maximum shift frequency F^m. Table I presents the test times and the highest rated shift frequencies F^m for every pair C_i, V_j assuming that all WPPs are L-bits wide. In order to conceal confidential data, test times are presented in normalized units, while bus sizes and WPP sizes are presented as multiples of a basic unit L. The test data for the cores are grouped according to the island they belong to. Note that F^m is reported in the last column of each island, except of the last two islands, for which it is equal to 200 MHz. The entries denoted by "N/A" correspond to cores that are not tested at the corresponding voltage settings.

We assume that the tester provides a clock signal with 400 MHz frequency, and TDM supports frequencies of 25, 50, 100, 200 and 400 MHz. For every core-voltage pair, every frequency in this set that is lower or equal to the corresponding F^m reported in Table I can be used.

Both TDM and STDM were implemented using the C programming language. The CPU time for the SoC of Table I is only a few minutes. The hardware overhead of the STDM scheme for the SoC at hand is less than 300 gates.We consider WPP widths that are up to 4x smaller than the size of the bus. Since STDM reduces the frequency on the bus to compensate for wrappers that are narrower than the bus, the set of frequencies supported by STDM is extended to include also the values of 12.5 MHz and 6.25 MHz, which are 2x and 4x smaller than the lowest frequency supported by TDM. Finally, based on the observations in Section II, we used a single bus with size in the range $\{L, L/2, L/4\}$.

In the first experiment, we compare STDM against the TDM and non-TDM approaches. We consider first the case that wrappers are flexible and thus their parallel ports can be set to the same bitwidth with the bus. Since all WPPs were initially designed to be L bits wide, in order to consider buses of bitwidth smaller than L, the wrapper configuration of every core had to be adjusted in TDM and non-TDM schemes. In STDM, all wrappers retain their original width L. The reshaping of the wrapper affects the length of the tests loaded into the wrapper; they are proportionally increased (every 2x reduction of the WPP bitwidth corresponds to a 2x increase of test time for shifting test data into the core).

The results for bus sizes equal to $L, L/2, L/4$ are shown in Fig. 5. STDM outperforms both TDM and non-TDM methods by a considerable margin. Note that the test time for the non-TDM scheme extends above the chart in the case of $L/4$ bits. When the size of the bus is equal to L bits, STDM degenerates to TDM, since the bus and the wrappers have the same bitwidth. As the bus bitwidth decreases to $L/2$ and $L/4$, TDM cannot further exploit the released TAM lines and the test time doubles in both cases. In contrast, SDM exploits the released TAM lines and test time does not scale proportionally, thereby offering higher multi-site efficiency.

Adjustments in the WPP size require adjustments of the scan chains and thus they are not always possible, like for example in the case of hard IP-cores or when there are decompressor constraints. At the same time, TDM and non-TDM cannot be applied when the bus is smaller than the WPP width. To show the advantages of STDM in this case, we compare the overall test time needed for testing one million devices using STDM with bus bitwidth equal to $L/2$ and $L/4$ against the overall test time needed for the same number of devices using TDM and non-TDM with bus and WPP bitwidth equal to L. Hence, only the effect of ATE-channel count is considered to evaluate multi-cite efficiency. Fig. 6 shows the improvement offered by STDM over TDM for various ATE channel counts in the range $[2L, 8L]$. Note that this count includes ATE channels reserved for control signals. The improvement of STDM over non-TDM (not shown in Fig. 6) is in the range of $[65\%, 80\%]$. It is obvious that STDM offers considerable test time savings, especially when the number of available ATE channels is small, because it achieves higher parallelism than the other methods. In practical scenarios, ATE channels constitute an expensive resource and only a small number of channels is available per chip in multi-site testing. Hence, we conclude that STDM is a very efficient approach.

In the next experiment, we examine the case that the different cores have different WPP bitwidths, and they cannot all perfectly match the size of the shared bus. Specifically, we examine the case that the WPPs of cores $C_2, C_3, C_5, C_8, C_{12}, C_{15}$ are $L/2$ bits wide, the WPPs of cores

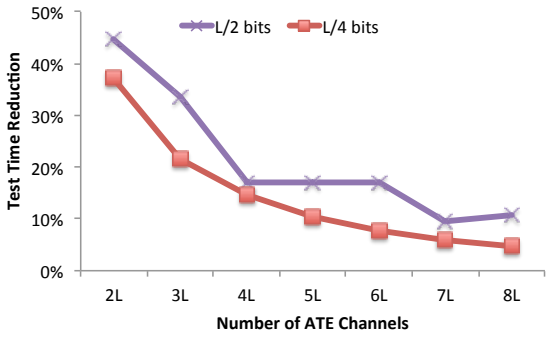

Fig. 6: Test-time reduction of STDM for multi-site testing.

C_1, C_9 are $L/4$ bits wide and the rest of the WPPs are L bits wide. In TDM, the bus is equal in bitwidth to the widest among the WPPs (i.e., L bits), and the smaller wrappers use only part of the bus. In the case of STDM, three different bus bitwidths were considered: L, $L/2$, $L/4$. The overall time improvements of STDM over TDM for testing 1 million devices were equal to 1.1x, 2.7x and 3.6x, respectively. STDM is better suited for buses of small bitwidth and outperforms TDM in this case. It is also important to note that when the bus width reduces by the factor of 2 from L to $L/2$ bits, STDM tests more than double the number of sites. Therefore, there are cases in which STDM not only increases the multi-site factor, but it also offers shorter test time per device than TDM.

Finally, we run experiments on a second SoC from Texas Instruments, which consists of 9 cores, 4 islands, 124 clock domains, and it can be set up to 4 voltage settings (details can be found in [15]). The maximum shift frequency is 200 MHz. All WPPs are L'-bit wide and a single test bus is used. In the case that the WPPs can be set to the same bitwidth as the bus, the TDM scheme gives normalized test times of 8191 units, 16808 units and 33604 units for bus widths equal to L', $L'/2$ and $L'/4$, respectively. The corresponding times for STDM are 8191 units, 12954 units and 24166 units (in STDM, all wrappers retain their original width L'). In the case that the WPPs cannot be set to the same bitwidth with the bus, we compare TDM with bus width equal to L', against the STDM scheme with bus width equal to $L'/2$ and $L'/4$. When the number of ATE channels varies in the range $[2L', 8L']$, STDM with bus width $L'/2$ ($L'/4$) reduces the time needed to test 1 million devices by a percentage in the range of 47% to 15% (41% to 10%) as compared to TDM. Therefore, we conclude that STDM scheme is also very effective for the second SoC.

VI. Conclusions

We have shown that STDM offers a highly efficient solution for multi-site test applications with a limited number of ATE channels. Space multiplexing permits the use of TAMs that are narrower than the wrappers and time multiplexing exploits the available frequency bandwidth to parallelize test application, thereby minimizing the additional time overhead. Experiments on industrial SoCs show that STDM is very effective for narrow TAMs and it significantly increases the number of the sites that can be tested in parallel.

References

[1] ARM 1176JZ(F)-S http://www.arm.com/products/CPUs/ARM1176.html.

[2] Intel Corp, "Intel XScale Core Developer's Manual, 2003", http://developer.intel.com/design/intelxscale/.

[3] *Intel: PXA270 Processor Datasheet, Nov. 2007. [Online]. Available: http://www.phytec.com/pdf/datasheets/.*

[4] A. Larsson, et. al., , "Test-Architecture Optimization and Test Scheduling for SoCs with Core-Level Expansion of Compressed Test Patterns," in *Proc. DATE*, March 2008, pp. 188–193.

[5] N. Ali, M. Zwolinski, B. Al-Hashimi, and P. Harrod, "Dynamic Voltage Scaling Aware Delay Fault Testing," in *IEEE ETS*, 2006, pp. 15–20.

[6] B. Amelifard and M. Pedram, "Optimal Design of the Power-Delivery Network for Multiple Voltage-Island System-on-Chips," *IEEE Trans. on CAD of Integr. Circ. and Systems*, vol. 28, no. 6, pp. 888–900, 2009.

[7] T. D. Burd and R. W. Brodersen, "Design Issues for Dynamic Voltage Scaling," in *ISLPED*, 2000, pp. 9–14.

[8] B. Chazelle, "The Bottomn-Left Bin-Packing Heuristic: An Efficient Implementation," *IEEE Trans. on Computers*, vol. C-32, no. 8, pp. 697–707, 1983.

[9] D. E. Lackey, et. al.,, "Managing Power and Performance for System-on-Chip Designs Using Voltage Islands," in *IEEE/ACM ICCAD*, Nov. 2002, pp. 195–202.

[10] S. K. Goel and E. J. Marinissen, "SoC Test Architecture Design for Efficient Utilization of Test Bandwidth," *ACM TODAES*, vol. 8, no. 4, pp. 399–429, 2003.

[11] S. Goel and E. Marinissen, "Optimisation of On-Chip Design-For-Test Infrastructure for Maximal Multi-Site Test Throughput," *IEE Proc. - Computers and Digital Techn.*, vol. 152, no. 3, pp. 442–456, 2005.

[12] V. Iyengar, S. Goel, E. Marinissen, and K. Chakrabarty, "Test Resource Optimization for Multi-Site Testing of SOCs under ATE Memory Depth Constraints," in *Proc. ITC*, 2002, pp. 1159–1168.

[13] K. J. Nowka, et. al., "A 32-bit PowerPC System-on-a-Chip with Support for Dynamic Voltage Scaling and Dynamic Frequency Scaling," *IEEE Journal of Solid-State Circ.*, vol. 37, no. 11, pp. 1441–1447, 2002.

[14] X. Kavousianos, K. Chakrabarty, A. Jain, and R. Parekhji, "Test Schedule Optimization for Multicore SoCs: Handling Dynamic Voltage Scaling and Multiple Voltage Islands," *IEEE Trans. on CAD of Integr. Circuits and Systems*, vol. 31, no. 11, pp. 1754–1766, 2012.

[15] ——, "Time-Division Multiplexing for Testing SoCs with DVS and Multiple Voltage Islands," in *17th IEEE ETS*, 2012, pp. 1–6.

[16] S. Khursheed, B. M. Al-Hashimi, S. M. Reddy, and P. Harrod, "Diagnosis of Multiple-Voltage Design With Bridge Defect," *IEEE Trans. on CAD of Integr. Circ. and Systems*, vol. 28, no. 3, pp. 406–416, 2009.

[17] E. Larsson, "Architecture for Integrated Test Data Compression and Abort-on-Fail Testing in a Multi-Site Environment," *IET Computers Digital Techniques*, vol. 2, no. 4, pp. 275–284, 2008.

[18] V. Iyengar, K. Chakrabarty, and E. J. Marinissen, "Test Access Mechanism Optimization, Test Scheduling, and Tester Data Volume Reduction for System-on-Chip," *IEEE Trans. on Computers*, vol. 52, no. 12, pp. 1619–1632, Dec. 2003.

[19] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes: The Art of Scientific Computing*, 3rd ed. New York: Cambridge University Press, 2007.

[20] R. Puri, et. al., , "Pushing ASIC Performance in a Power Envelope," in *Proc. DAC*, June 2003, pp. 788–793.

[21] J. Rajski, J. Tyszer, M. Kassab, and N. Mukherjee, "Embedded Deterministic Test," *IEEE Trans. on CAD of Integr. Circ. and Systems*, vol. 23, no. 5, pp. 776–792, 2004.

[22] J. Rivoir, "Lowering Cost of Test: Parallel Test or Low-Cost ATE?" in *12th IEEE ATS*, 2003, pp. 360–363.

[23] ——, "Parallel Test Reduces Cost of Test More Effectively than just a Cheap Tester," in *IEEE/CPMT/SEMI 29th Int. Electronics Manufacturing Technology Symp.*, 2004, pp. 263–272.

[24] S. Khursheed, et. al., "Bridging Fault Test Method With Adaptive Power Management Awareness," *IEEE Trans. on CAD of Integr. Circ. and Systems*, vol. 27, no. 6, pp. 1117–1127, June 2008.

[25] N. Touba, "Survey of Test Vector Compression Techniques," *IEEE Design Test of Computers*, vol. 23, no. 4, pp. 294–303, 2006.

[26] E. Volkerink, A. Khoche, J. Rivoir, and K.-D. Hilliges, "Test Economics for Multi-Site Test with Modern Cost Reduction Techniques," in *Proc. 20th IEEE VTS*, 2002, pp. 411–416.

[27] T. Yoneda, M. Imanishi, and H. Fujiwara, "An SoC Test Scheduling Algorithm using Reconfigurable Union Wrappers," in *DATE*, 2007.