Time-Division Multiplexing for Testing SoCs with DVS and Multiple Voltage Islands*

Xrysovalantis Kavousianos
Dept. of Computer Science
University of Ioannina, Greece
Email: kabousia@cs.uoi.gr

Krishnendu Chakrabarty
Dept. of Electrical & Computer Engineering
Duke University, NC, USA
Email: krish@duke.edu

Arvind Jain, Rubin Parekhji
Texas Instruments (India) Pvt. Ltd.
Bangalore, India
Email: a-jain@ti.com, parekhji@ti.com

***Abstract*—Dynamic voltage scaling (DVS) has been widely adopted in multicore SoCs for reducing dynamic power consumption. Despite its benefits, the use of DVS increases test time because high product quality can only be ensured by testing every core at multiple supported voltage settings; hence the repetitive application of the same or different tests at multiple voltage settings becomes necessary. In addition, testing at lower supply voltage settings increases considerably the length of each test because lower scan frequencies must be used for shifting test data using scan chains. Standard scheduling techniques fail to reduce the test time for DVS-based SoCs since they do not model testing at multiple voltage settings. In addition, they do not consider the practical aspects of tester overhead and the dependencies between core voltage settings due to the use of voltage islands. To alleviate the detrimental impact of DVS on test application time, we propose a time-division multiplexing (TDM) method and an integer linear programming-based test scheduling technique, which exploit high automatic test equipment (ATE) frequencies even when low shift frequencies must be used at low voltage settings. Experimental results on two industrial SoCs highlight the effectiveness of TDM and the associated scheduling method.**

I. Introduction

Dynamic voltage scaling (DVS) offers a good trade-off between power consumption and system performance as it adaptively adjusts power supply-voltage depending on the workload of the SoC [14]. Low workload permits the use of low power-supply voltage level as the SoC can still achieve performance goals without consuming an excessive amount of energy. High workload imposes the use of high supply voltage to meet performance objectives. In addition, SoCs consist of multiple voltage islands with separate supply rail and unique power characteristics [16]. Voltage islands permit the adaption of the DVS technique to the specific requirements of the cores of each island and maximize the power gains [14]. Several state-of-the-art processors adopt this technique [2], [8], [18].

Defect-free operation of an SoC that supports DVS can be only assured by testing the SoC at multiple voltage levels as various defects manifest themselves in different ways at various voltage levels [1], [7], [11]. Testing at different voltage settings increases test time because either test patterns have to be applied multiple times or additional patterns must be used for the other voltage settings. As a result, test cost is increased considerably due to the use of DVS for power management.

Even though many techniques reduce the test cost for single-V_{dd} SoCs [6], [9], [12], [13], [15], [20] they are not suitable for multi-V_{dd} designs. Multi-V_{dd} designs impose constraints that do not exist in single-V_{dd} designs. For example, cores residing at the same island cannot be tested in parallel at different voltage settings even if their corresponding test access mecahanisms (TAM) are available. In addition, lower supply voltages reduce the shift frequency for scan chains of cores which further increases test time. The test scheduling method proposed in [10] considers the additional constraints imposed by the use of multiple voltage settings, but it does not tackle the problem that low shift frequencies must be used at the lower voltage settings. Thus its effectiveness for test scheduling in DVS-based SoCs is limited.

The work presented in this paper alleviates the problem of low shift frequencies in DVS-based SoCs; it reduces the test time by means of an efficient time-division-multiplexing (TDM) architecture and an effective integer linear programming (ILP)-based test scheduling method. It even allows us to reduce the shift frequency for testing various cores at multiple voltage settings below the nominal value, whenever this choice of shift frequencies and TDM minimizes the total SoC test time. Experimental results for two representative industrial SoCs are presented, which highlight the clear benefits of applying the proposed technique on multi-V_{dd} designs. In particular, we highlight the benefits of the proposed method over [10] as well as over a baseline based on a conventional test scheduling that does not directly target multiple voltage settings. Although the proposed ILP method is efficient for contemporary industrial SoCs, it might not be computationally feasible for future generations of SoCs. For these cases, we present an LP-relaxation approach, which provide sub-optimal, but nevertheless good solutions to the test-scheduling problem.

II. Motivation

Testing of multi-V_{dd} SoCs requires long test application times due to the high volume of tests that must be repeatedly applied at the various voltage levels. Test time is dominated by the process of serially loading test data into the cores through scan chains. As scan chains are usually not designed to operate

*This research was supported in part by the National Science Foundation under grant no. CCF-0903392, and by the Semiconductor Research Corporation under contract no. 1992.

at the rated speed of the cores, an ATE transfers and loads test data to cores using a slow scan shift frequency. Consequently the time for testing each core is very long. Even in the case that the tester supports higher scan frequencies, this capability cannot be exploited, thus leaving tester potential underutilized.

The above problem is exacerbated when cores are tested at the lower power-supply voltages, as the reduction in supply voltage reduces the maximum alloweable scan frequency. In addition, testers usually conduct SoC testing using a single scan frequency over the duration of the test period. Thus, in order to avoid scan violations at any voltage setting, the lowest frequency for shifting test data has to be used, which corresponds to the lowest voltage level. Therefore, the application of tests at multiple voltage settings combined with the decreased scan frequency, which is imposed by the lower voltage settings, increases the test time of the SoC even more.

Despite these limitations, the partitioning of every SoC into multiple voltage islands permits the concurrent testing of different cores at different voltage settings when they belong to different islands. In addition, cores which are tested at lower shift frequencies and which share the same TAM resources may be tested concurrently by also sharing the high frequencies of the ATE channels. For example, two cores, which have to be loaded using low scan frequencies, can be loaded in parallel using the same TAM resource by multiplexing the test data in time on an ATE channel. By using time-division multiplexing (TDM), the test data can be transmitted by the tester at a higher frequency, while at the same time, they can be shifted into the scan chains of multiple cores at much lower frequencies which depend on the voltage setting used in each case. Note that this is a generalization of the TDM approach as it does not only exploit the gap between the shift frequency of the ATE and the cores but it also exploits the gap on the shift frequencies between cores of different islands which are concurrently tested at different voltage settings.

Example 1. Suppose that the maximum scan frequencies of the cores in an SoC at the voltage settings $V_1 > V_2 > V_3$ are 200, 100 and 50 MHz, respectively. Let the maximum scan frequency supported by the tester be 200 MHz. Then, a TAM resource can be used to test at any given time instance:

1) a single core at V_1,
2) two cores at V_2,
3) four cores at V_3,
4) one core at V_2 and two cores at V_3,
5) one core at V_1 at 100 MHz, one core at V_2 at 50 MHz (note that a slower shift frequency than the nominal one is used for these cores) and one core at V_3, etc. ■

Depending on the maximum tester frequency, as well as on the maximum scan frequency at each voltage setting, there exist many different scheduling scenarios that exploit the full capability of the tester and the TAM mechanism for increasing the parallelism in transporting and loading test data into the cores. In addition, there can be situations when counter-intuitively and in contrast to what we expect, shifting test data into the scan chains at a lower than the nominal frequency may be beneficial in terms of ATE-channel-frequency utilization and this strategy may reduce the overall SoC test time. Since the solution space is very large, it is computationally challenging to find a good solution, especially when lower than nominal scan frequencies are also explored for reducing the overall test time. Test scheduling using multiple scan frequencies is a generalization of the simple scheduling problem that assumes a single scan frequency for all cores; the latter problem is NP-complete, therefore the scheduling problem being considered is also at least NP-hard. To this end, an ILP-based test scheduling approach is proposed in this paper to provide optimal solutions, or near optimal solutions using LP-relaxation if exact solutions are not computationally feasible.

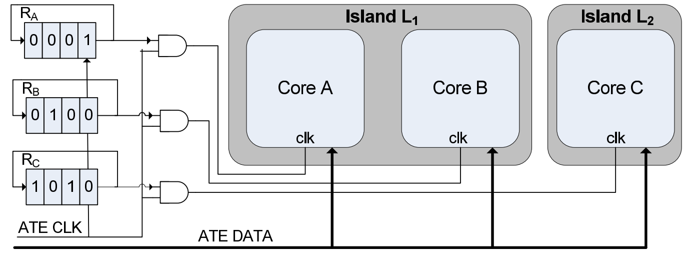

Fig. 1. Proposed TDM scheme.

III. TDM Scheme

Fig. 1 presents the proposed TDM scheme for an SoC consisting of cores A, B, C, and a test bus shared between them. Cores A, B belong to island L_1 and core C belongs to L_2. Both islands support voltage settings V_1, V_2, V_3 and the nominal scan frequencies at each voltage setting are $F, F/2$ and $F/4$, respectively. Let the tester provide the ATE_CLK signal with frequency F used as the generator clock signal for loading the scan chains. The tester also provides test data on the bus at frequency F. Each core is assigned one cyclical shift register with length equal to 4, which divides the scan frequency by a value equal to 1, 2 or 4. The scan frequency for each core is determined by loading the appropriate pattern to each register before the testing of the core begins. Every shift register is clocked with the fast ATE_CLK (frequency F) and in turn provides a clock signal with frequency equal or smaller to F. The following example illustrates this method.

Example 2. Let us assume that at a specific time instance cores A, B are tested at voltage V_3 and C is tested at voltage V_2. Then, the highest frequencies that can be used for A, B, C are $F/4, F/4, F/2$, respectively. In order to provide scan frequency of $F/4$ to core A, register R_A in Fig. 1 is loaded with the pattern “0001”. Then, during every 4 succesive cycles of ATE_CLK, the rightmost cell of R_A receives the value ‘1’ only once and permits the application of one out of the four active edges of signal ATE_CLK to the core A. In this case the scan clock frequency for this core is equal to $F(ATE_CLK)/4 = F/4$. Register R_B is loaded with the pattern “0100”, which sets the scan frequency of core B equal to $F/4$ too. However, note that a different pattern from core A is used in order to offer non-overlapping loading of the test data from the common bus. Register R_C is initialized with pattern “1010” and thus core C sees one active clock

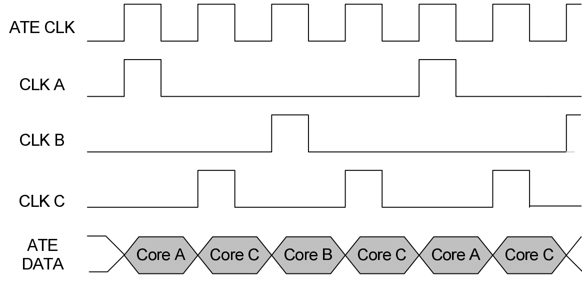

Fig. 2. Illustration of time-division-multiplexing of test data.

edge every two ATE_CLK cycles. Therefore, the scan clock frequency for core C is set to $F/2$. Similar to the previous case, the pattern loaded into R_C has non-overlapping '1' logic values with the patterns loaded into registers R_A, R_B. ■

From the above example we note that: a) all shift registers are concurrently shifted at every ATE_CLK cycle, and b) they are loaded with patterns that have no overlapping logic values of '1'. Therefore, at most one core at any ATE_CLK cycle receives the active edge of ATE_CLK. At the same time, at every ATE_CLK cycle, one test data vector is available at the bus and the core which receives the active clock edge at that time also loads the respective test vector from the bus.

The timing diagram for Example 2 is shown in Fig. 2. It is obvious that the tester channel utilization is increased without violating the timing specifications of the cores. Whenever a core is not tested, the corresponding shift register is loaded with the all-'0' pattern and it receives no active edges of the scan clock. Larger shift registers can be also used to allow various levels of TDM (e.g., 8-bit registers offer division by 2, 4 and 8). The selection of the register size depends on the specifications of each SoC and the amount of TDM required.

The various frequencies offered by the TDM technique cannot perfectly match the highest scan frequencies that can be used at the various voltage settings. However, this is not a limitation of the proposed technique as the optimization target is to exploit as much of the available ATE frequency as possible regardless of the shift frequency used for each core. This goal is achieved by the proposed test scheduling method, even when lower than the nominal scan frequencies are used at each voltage setting for shifting test data into the cores. For example, suppose that a tester supports a maximum shift frequency of 300 MHz, and that cores C_1, C_2 share the same TAM resource and are being tested using shift frequencies 150 and 200 MHz, respectively. Let the test time in each case be 10 and 5 ms, respectively. Then, these two tests must be applied serially and the total test time is equal to 15 ms. If the shift frequency for core C_2 is reduced to 150 MHz then the test time for this core increases to 7 ms. However, both tests can be applied concurrently so the total test time drops to 10 ms.

Similar to [19] the proposed TDM scheme can be combined with test data compression and BIST to derive additional benefits. Identical cores can be tested using a broadcast mode if identical patterns are loaded at the respective registers to concurrently shift test data from the bus. Power constraints can be considered to set an upper bound on the amount of test parallelism. For those cycles of ATE_CLK signal in which no core receives test data, the ATE repeat command can be used to avoid storing unnecessary test data in tester memory. Finally, testing of the logic in between the cores can be done by considering tests that involve multiple cores at a time.

IV. Test Scheduling Method

We consider a multi-core SoC with N_c cores $C_1, \ldots, C_{N_c}$ and N_I voltage-islands $L_1, \ldots, L_{N_I}$ ($N_c \geq N_I$). Each island includes a subset of the N_c cores. We consider a set of N_v voltage levels $V_1, \ldots, V_{N_v}$ sorted in descending order (i.e., $V_1 > \cdots > V_{N_v}$) and we associate each island to a subset of these voltage levels. We assume that the DFT infrastructure for the chip has been implemented, whereby TAMs do not have to be optimized. The DFT infrastructure imposes restrictions on how much parallelism can be achieved during the testing of the cores. For each core C_i and every voltage V_j, there is a minimum test time MT_{C_i,V_j} associated with testing core C_i at voltage V_j. The parameter MT_{C_i,V_j} is determined by the highest scan frequency $F^{max}_{C_i,V_j}$ that can be used for this case.

Let N_F frequencies $F_1 > F_2 > \cdots > F_{N_F}$ be supported by the TDM scheme. Testing a core at a particular voltage setting involves different completion time for each of the scan frequencies $F_1, F_2, \ldots, F_{N_F}$. Let $t_{C_iV_jF_k}$ correspond to the task of testing core C_i at voltage V_j using scan frequency F_k and $T_{C_iV_jF_k}$ be its completion time. It is obvious that task $t_{C_iV_jF_k}$ exists only when $F_k \leq F^{max}_{C_i,V_j}$ and also $T_{C_iV_jF_1} < T_{C_iV_jF_2} < \cdots < T_{C_iV_jF_{N_F}}$. In the general case, we note that

$$\text{if } F_k \leq F^{max}_{C_i,V_j} \text{ then } T_{C_iV_jF_k} \simeq MT_{C_i,V_j} \cdot \frac{F^{max}_{C_i,V_j}}{F_k} \quad (1)$$

(capture cycles constitute a negligible portion of the test period.)

For every core C_i, voltage V_j and frequency F_k a binary variable $S_{C_iV_jF_k}$ is assigned, which is equal to '1' whenever the test of C_i at V_j is applied using F_k. When $F_k > F^{max}_{C_i,V_j}$, F_k cannot be used for testing core C_i at V_j and thus we set $S_{C_iV_jF_k} = 0$ (higher frequencies are not supported at the lower voltage settings due to timing violations of the scan chains).

The first constraint that must be satisfied is that any test for core C_i at any voltage setting V_j is applied using a single scan frequency. This is modeled by the following relation:

$$\sum_{k=1}^{N_f} S_{C_iV_jF_k} = 1, \; \forall i \in \{1 \ldots N_c\}, \forall j \in \{1 \ldots N_v\} \quad (2)$$

For those voltage settings that a core is not tested the above constraint is omitted.

Let $ST_{C_iV_jF_k}$ denote the start time of task $t_{C_iV_jF_k}$. For testing core C_i at voltage setting V_j, only a single frequency can be used as indicated by (2). Therefore, the start and end

time for this test is given by the following relations

$$ST_{C_iV_j} = \sum_{k=1}^{N_f} S_{C_iV_jF_k} \cdot ST_{C_iV_jF_k} \quad (3)$$

$$END_{C_iV_j} = \sum_{k=1}^{N_f} S_{C_iV_jF_k} \cdot (ST_{C_iV_jF_k} + T_{C_iV_jF_k}) \quad (4)$$

The product $S_{C_iV_jF_k} \cdot ST_{C_iV_jF_k}$ is not linear so it is replaced by the variable $y_{C_iV_jF_k}$ and new constraints are introduced which can be found in the Appendix (Part 1). Thus (3), (4) become

$$ST_{C_iV_j} = \sum_{k=1}^{N_f} y_{C_iV_jF_k}, \quad (5)$$

$$END_{C_iV_j} = \sum_{k=1}^{N_f} (y_{C_iV_jF_k} + S_{C_iV_jF_k} \cdot T_{C_iV_jF_k}) \quad (6)$$

The second constraint is that any two cores in the same island cannot be concurrently tested at different voltage settings. This is equivalent to the following statement: for any two cores C_{i_1}, C_{i_2} located in the same island, and any two voltage settings V_{j_1}, V_{j_2} ($V_{j_1} \neq V_{j_2}$), either test for C_{i_1}, V_{j_1} begins after the test for C_{i_2}, V_{j_2} finishes or vice versa. Using relations (5), (6) it is written as: $\mathcal{C}_1$ or $\mathcal{C}_2$, where

$$\mathcal{C}_1 : \sum_{k=1}^{N_f} y_{C_{i_1}V_{j_1}F_k} \geq \sum_{k=1}^{N_f} (y_{C_{i_2}V_{j_2}F_k} + S_{C_{i_2}V_{j_2}F_k} T_{C_{i_2}V_{j_2}F_k})$$
$$\mathcal{C}_2 : \sum_{k=1}^{N_f} y_{C_{i_2}V_{j_2}F_k} \geq \sum_{k=1}^{N_f} (y_{C_{i_1}V_{j_1}F_k} + S_{C_{i_1}V_{j_1}F_k} T_{C_{i_1}V_{j_1}F_k}) \quad (7)$$

Constraint (7) is linearized as it is shown in the Appendix (Part 2).

In a similar way, we determine the concurrency between different tests: if for two cores C_{i_1}, C_{i_2} and any two voltage settings V_{j_1}, V_{j_2}, the test for $C_{i_1}V_{j_1}$ begins after the test for C_{i_2}, V_{j_2} finishes or the test for C_{i_2}, V_{j_2} begins after the test for C_{i_1}, V_{j_1} finishes, then the two tests are not concurrent. In every other case, they are concurrent. Concurrency is determined for all cores sharing a common bus, excluding those that are in the same island and correspond to $V_{j_1} \neq V_{j_2}$ as they are excluded by the second constraint. Let $Conc_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}}$ be a binary variable which is equal to '1' if the tests C_{i_1}, V_{j_1} and C_{i_2}, V_{j_2} overlap. Then we can formally write this constraint as follows:

If $ST_{C_{i_1}V_{j_1}} \geq END_{C_{i_2}V_{j_2}}$ or $ST_{C_{i_2}V_{j_2}} \geq END_{C_{i_1}V_{j_1}}$
then $Conc_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} = 0$ else $Conc_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} = 1$ (8)

The linearization of (8) is shown in the Appendix (Part 3).

The final constraint bounds the number and type of tests that can concurrently use the same TAM resource. Each scan frequency is assigned a weight proportional to the capacity (in time) of the resource consumed by the tests applied at this frequency. For example, consider a scan frequency F supported by the tester. Then, any core that is tested using scan frequency $F/2$, $F/4, \cdots$ consumes half, a quarter, etc., of the capacity of the TAM resource. Any concurrent combination of tests must not exceed the capacity of the TAM resource. Let $W(F_k)$ be the capacity consumed by any test using scan frequency F_k. Let also $W(TAM)$ be the capacity available on the TAM resource. Then, for any two cores C_{i_1}, C_{i_2} connected to the same TAM resource and tested concurrently at supply voltages V_{j_1}, V_{j_2} the sum of their weights must not exceed $W(TAM)$. This is modeled by the following constraint:

$$Conc_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} \cdot \sum_{k=1}^{N_f} (S_{C_{i_1}V_{j_1}F_k} + S_{C_{i_2}V_{j_2}F_k}) \cdot W(F_k) \leq W(TAM) \quad (9)$$

Again (9) is linearized as it is shown in the Appendix (Part 4). In the same way we construct the constraints for triplets or larger sets of tests depending on the maximum number of cores which are connected to any TAM resource (see Appendix (Part 5)).

Finally, if $TestLength$ is the total test time of the SoC, the optimization objective of the ILP model is the following:

Minimize: $TestLength$
Subject to: $TestLength \geq ST_{C_iV_j} + T_{C_iV_j}\ \forall i \in \{1 \cdots N_c\},\ j \in \{1 \cdots N_v\}$.

Even though ILP models can optimally solve test scheduling optimization problems, they are NP-hard [5] and they do not scale well for large SOC designs. ILP problems can be solved in polynomial time by using the method of LP-relaxation [3]. In LP relaxation, the binary variables are relaxed to real-valued variables such that the solution to the relaxed LP problem provides a lower bound for the cost function (test time in this case). However, real-valued variables are inadmissible in practice. A common technique to map these binary values to '0', '1' values is the weighted probabilistic technique of randomized rounding [17]. After the corresponding LP problem is solved, all binary variables that are assigned to fractional values are identified. One of them is randomly selected and it is set to '1' with a probability equal to the fractional value. At this point, we check this assignment for consistency with the constraints of the ILP model, and if it violates any constraints we reverse the assignment. Then, the LP problem is solved again, and the randomized rounding step is repeated until all variables are set to either 0 or 1.

V. Experimental Results

For evaluating the proposed method we used test data for two industrial SoCs, hereafter referred to as SoC-A, SoC-B, respectively, which are targeted for portable wireless applications. SoC-A has 4 voltage islands $I_1^A, I_2^A, I_3^A, I_4^A$, 9 cores $C_1^A, \ldots, C_9^A$ and 124 clock domains. SoC-B has 7 voltage islands $I_1^B, \ldots, I_7^B$, 15 cores $C_1^B, \ldots, C_{15}^B$ and 225 clock domains. SoC-A can be set to up to 4 voltage settings, while SoC-B can be set to up to 6 voltage settings.

Table I and Table II present the minimum testing times MT_{C_i,V_j} for all cores C_i of SoC-A and SoC-B, at the various

TABLE I
SoC-A MINIMUM TEST TIMES (IN NORMALIZED TIME UNITS) & MAXIMUM SCAN FREQUENCIES F^m (IN MHZ).

V_{dd}	I_1^A		I_2^A				I_3^A					I_4^A	
level	C_1^A	F^m	C_2^A	C_3^A	C_4^A	F^m	C_5^A	C_6^A	C_7^A	C_8^A	F^m	C_9^A	F^m
V_1	300	266	60	128	262	200	N/A	N/A	128	600	133	55	200
V_2	500	200	95	220	500	100	475	375	170	950	100	N/A	N/A
V_3	800	100	160	340	700	50	800	600	300	1600	50	N/A	N/A
V_4	1600	50	N/A	N/A	N/A	N/A	1600	1200	600	3200	25	N/A	N/A

TABLE II
SoC-B MINIMUM TEST TIMES (IN NORMALIZED TIME UNITS) & MAXIMUM SCAN FREQUENCIES F^m (IN MHZ).

V_{dd}	I_1^B			I_2^B				I_3^B				I_4^B			I_5^B			I_6^B		I_7^B
level	C_1^B	C_2^B	F^m	C_3^B	C_4^B	C_5^B	F^m	C_6^B	C_7^B	C_8^B	F^m	C_9^B	C_{10}^B	F^m	C_{11}^B	C_{12}^B	F^m	C_{13}^B	C_{14}^B	C_{15}^B
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

voltage levels V_j. The test data for the cores are grouped into columns according to the island they belong to. The test times are calculated using the maximum scan frequency, denoted as 'F^m' that does not cause scan chain timing violations at any core for shifting at the corresponding voltage level (they are presented in normalized time units, so as not to reveal confidential data). This frequency is presented in the last column of each island (reported in MHz and different for every voltage setting). For the last two islands I_6^B, I_7^B of SoC-B, the value of F^m is not included in Table II; it is equal to 200 MHz. The entries denoted as "N/A" correspond to cores that either do not operate at the respective voltage settings or they are not tested at these voltage settings.

We assume that the tester provides a clock signal with frequency that is close to the highest scan frequency of any core of the SoC (200 MHz for SoC-A and 400 MHz for SoC-B). For SoC-A this frequency is divided on-chip using the proposed TDM scheme by 8, 4, 2 and 1, which corresponds to scan frequencies 25, 50, 100 and 200 MHz. For SoC-B, it is divided by 16, 8, 4, 2 and 1, and the scan frequencies are 25, 50, 100, 200 and 400 MHz, respectively. Each core C_i is tested at a voltage V_j using any of the TDM frequencies that are smaller or equal to the corresponding highest nominal scan frequency of C_i at V_j. For example, for testing core C_8^A at V_1, the highest scan frequency that can be used is 133 MHz. If we consider a TDM scheme that provides frequencies of 200, 100, 50 and 25 MHz, then only the frequencies 100, 50 and 25 MHz can be used for testing C_8^A at V_1 and using equation (1) the respective test times are equal to 800, 1600 and 3200, respectively. Note that when the scan frequency is divided by two, the completion time for the task doubles. However, at the same time, the use of a smaller frequency permits a higher level of parallelization between the various tasks, which offsets the increase in task time due to the reduction in frequency.

First, we run various experiments for SoC-A where the ATE frequency is set between 25 MHz - 400 MHz. We considered as TAM two test buses that are shared between the cores of

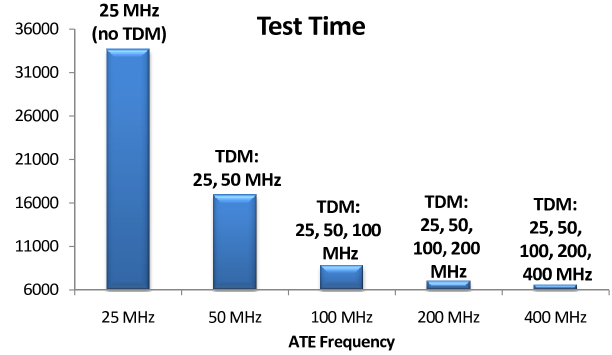

Fig. 3. Test time obtained using TDM for different ATE frequencies.

SoC-A. For each ATE frequency, we assume that the TDM scheme divides the frequency by 2, 4, 8, etc., until the lowest frequency provided is equal to 25 MHz (this is the lowest scan frequency for SoC-A as shown in Table I). The results are shown in Fig. 3. Above each bar we report the scan frequencies generated by the TDM scheme. It is obvious that, despite the low frequency used for shifting test data into scan chains, as the tester frequency increases, the test time decreases due to the higher parallelism achieved. Without TDM, testing has to be conducted at the lowest nominal scan frequency (i.e., 25 MHz in this case). The test time at this frequency is reported in the leftmost bar of Fig. 3 and it obviously represents the least effective solution for testing SoC-A.

Next we compare the proposed method against two non-TDM based methods. The first one is the Shortest-Job-First (SJF) approach proposed in [4], which is efficient for scheduling tests at single-V_{dd} designs. This method schedules the tasks using a priority order based on the length (in time) of each task (the next task selected to be scheduled each time is the task with the shortest test length). We adapted this approach to multi-V_{dd} designs by appending additional constraints. In addition, we run experiments using the multi-V_{dd} approach proposed in [10] for the test cases used in this paper.

TABLE III
SoC-A Comparisons

No of Buses	Cnf	SJF		[10]		Proposed	
		RL	LB	RL	LB	ILP	Rand
2	1	37248	14055	33704	12795	6548	19763
	2	28252	12828	27729	12200	10100	20453
	3	36884	14153	35848	13428	9368	20191
3	1	22456	9540	19872	7603	6548	15987
	2	26752	11040	25572	7525	6548	21451
	3	28252	12350	27729	12200	6720	18850

TABLE IV
SoC-B Comparisons

No of Buses	Cnf	SJF		[10]		Proposed TDM	
		RL	LB	RL	LB	ILP	Rand
3	1	123360	24994	119724	23358	17095	90896
	2	125149	25603	123364	24275	18589	77709
	3	135748	25848	129384	25532	18045	80150
4	1	129379	30447	129379	30447	25435	58207
	2	123304	20396	112000	21321	22937	58551
	3	137904	27246	137904	27246	23982	52774

For these baseline methods, we consider a realistic scenario (denoted as "RL") with respect to tester resources and TAM constraints, according to which there is only one ATE channel for providing the clock signal to every core. Note that usually testers in industrial multi-site testing environments do not provide separate clocks for different scan partitions; hence it is likely that all partitions may have to be shifted at the same rate. In addition, parallel testing of multiple cores at different voltage settings imposes the use of a lower frequency that does not violate the scan-chain timing for any of the cores being tested. Therefore, only one frequency is available for testing all cores, and it needs to equal the lowest shift frequency used for any of the cores and any voltage setting. For both SoCs this frequency was set equal to 25 MHz.

Even though the above baseline scenario is realistic for multisite testing in industry, we consider also a second hypothetical scenario for the baseline methods. Our goal is to show that the benefits of the proposed method are not limited to leveraging of higher scan frequencies for loading cores. Higher utilization of TAM bandwidth can also be achieved due to increased parallelism achieved by running different tasks (tests) at different voltage settings and frequencies at the same time. We assume that for each core, the highest possible scan frequency is used for loading the test data at any voltage setting. In other words, test scheduling is done using the minimum test times reported in Tables I-II. A separate ATE clock signal is required for every core, which increases the test cost as it increases both the number of ATE channels and the number of pins required for concurrently supporting these different scan frequencies. However, any non-TDM method cannot achieve better test time than this scenario, therefore it provides the lower bound of the test time for non-TDM methods. This baseline scenario is denoted as "LB".

We used a commercial solver for the proposed ILP model. The solver was allowed to run for a few hours and the best schedule provided during that time is reported. We also run the LP-relaxation technique (denoted as "Rand") for ten randomized experiments and we report the best result found. The CPU time for "Rand" is a few minutes for each experiment.

We used various TAM configurations, assuming a predetermined TAM architecture in each case (the problem of TAM optimization was not considered). For each volume of TAM resources, considered as shared buses in this work, we assumed three different random sharings of the buses between the various cores. The results for SoC-A are shown in Table III and for SoC-B in Table IV. The first two columns present the number of buses and the configuration index number while the next three pairs of columns present the results for SJF, the method proposed in [10] and the proposed method respectively. The proposed test scheduling method achieves remarkably high reduction in test time compared to all baseline methods. The test time is considerably lower than that of both realistic baseline approaches and it is also much lower in almost all cases than the lower bound for any non-TDM approach. The percentage reduction of the proposed ILP approach over [10] for the RL scenario approaches the value of 86.1% while for the LB scenario it approaches the value of 48.8%. The respective reduction percentages over the SJF approach are even higher. Therefore, we conclude that the proposed method offers considerable reductions in test time.

VI. Conclusions

We have introduced a test scheduling method which uses TDM and ILP-based optimization for multicore SoCs consisting of multiple islands that support multiple voltage settings. Experimental results on two industrial SoCs targeted for wireless applications have shown that the proposed method reduces test time by overcoming ATE bottlenecks and limitation due to DVS on loading test data into cores. Even though ILP-based methods are not scalable for general problem instances, the proposed method can be easily used for practical SoCs through careful model development and LP relaxation.

References

[1] N. B. Ali, M. Zwolinski, B. M. Al-Hashimi and Peter Harrod, "Dynamic Voltage Scaling Aware Delay Fault Testing," in *Proc. ETS*, 2006

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

[3] D. Bertsimas and J. Tsitsiklis, *Introduction to Linear Optimization*. Athena Scientific, 1997.

[4] K. Chakrabarty, "Test scheduling for core-based systems using mixed-integer linear programming", *IEEE Trans. CAD*, vol. 19, pp. 1163-1174, Oct. 2000

[5] M. R. Gary and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*. Freeman, 1979.

[6] S. Goel, et. al., "Testing of SoCs with Hierarchical Cores: Common Fallacies, Test Access Optimization, and Test Scheduling", *IEEE Trans. Computers*, vol. 58, No 3, March 2009, pp. 409-423.

[7] U. Ingelsson, et. al., "Resistive Bridging Faults DFT with Adaptive Power Management Awareness," in *Proc. IEEE ATS*, Oct. 2007, pp. 101-106.

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

[9] V. lyengar, K. Chakrabarty, and E. Marinissen, "Test access mechanism optimization, test scheduling, and tester data volume reduction for system-on-chip,", *IEEE Trans. on Comp.*, Vol. 52, no 12, pp. 1619-1632, 2003.

[10] X. Kavousianos, K. Chakrabarty, A. Jain and R. Parekhji, "Test Scheduling for Multicore SoCs with Dynamic Voltage Scaling and Multiple Voltage Islands", in *Proc. IEEE Asian Test Symposium*, pp. 33-39, 2011

[11] S. Khursheed, U. Ingelsson, P. Rosinger, B. M. Al-Hashimi, and P. Harrod, "Bridging Fault Test Method with Adaptive Power Management Awareness," *IEEE Trans. CAD*, vol. 27, pp. 1117-1127, Jun. 2008.

[12] E. Larsson and Z. Peng, "An integrated framework for the design and optimization of SOC test solutions," *J. Electron. Test.: Theory Appl. (JETTA)*, vol. 18, no. 4/5, pp. 385-400, 2002.

[13] A. Larsson et al., "Test-Architecture Optimization and Test Scheduling for SOCs with Core-Level Expansion of Compressed Test Patterns", in *Proc. DATE*, 2008

[14] G. Magklis et al., "Dynamic Frequency and Voltage Scaling for a Multiple-Clock-Domain Microprocessors," *IEEE J. Micro*, Nov-Dec 2003, pp. 62-68.

[15] E.J. Marinissen et al., "A Structured and Scalable Mechanism for Test Access to Embedded Reusable Cores," in*Proc.ITC*, pp. 284-293, 1998.

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

[17] P. Raghavan and C. Thompson, "Randomized rounding: A technique for provably good algorithms and algorithmic proofs,'' *Combinatorica*, vol. 7, no. 4, pp. 365374, 1987.

[18] TransMeta Crusoe TM5800. (2009 Oct.) [Online]. Available: http://www.cpu-world.com/CPUs/TM5800/

[19] L.-T. Wang, et al., "UltraScan: Using Time-Division Demultiplexing/Multiplexing (TDDM/TDM) with VirtualScan for Test Cost Reduction", *Proc. ITC*, p. 36.4, 2005.

[20] T. Yoneda, M. Imanishi, and H. Fujiwara, "An SoC Test Scheduling Algorithm Using Reconfigurable Union Wrappers," in *Proc. DATE*, pp. 231-236, Apr. 2007.

APPENDIX

Part 1

At first we show the linearization of the terms $S_{C_iV_jF_k} \cdot ST_{C_iV_jF_k}$. For every $i \in [1 \cdots N_c], j \in [1 \cdots N_v], k \in [1 \cdots N_f]$ we introduce an integer variable $y_{C_iV_jF_k}$ which is set equal to the above product. Let TB (TimeBound) be the maximum possible test time calculated as the summation of the time needed for every core and voltage setting when it is loaded using the minimum scan frequency F_{N_f}, that is

$$TB = \sum_{i \in [1 \cdots N_c], j \in [1 \cdots N_v]} T_{C_iV_jF_{N_F}}$$

i.e., we assume that the longest task is used for every core and no parallelism is allowed. Then we have the following three relations for each variable $y_{C_iV_jF_k}$:

$$\begin{aligned} y_{C_iV_jF_k} - TB \cdot S_{C_iV_jF_k} &\leq 0 \\ -ST_{C_iV_jF_k} + y_{C_iV_jF_k} &\leq 0 \\ ST_{C_iV_jF_k} - y_{C_iV_jF_k} + TB \cdot S_{C_iV_jF_k} &\leq TB \end{aligned}$$

Part 2

First we introduce two binary variables $\theta^1_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}}$ and $\theta^2_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}}$ which satisfy the equation $\theta^1_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} + \theta^2_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} = 1$. Then constraint $\mathcal{C}_1$ or $\mathcal{C}_2$ can be written as

$$\theta^1_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} \cdot \left(\sum_{k=1}^{N_f} y_{C_{i_1}V_{j_1}F_k} - \sum_{k=1}^{N_f} (y_{C_{i_2}V_{j_2}F_k} + S_{C_{i_2}V_{j_2}F_k} \cdot T_{C_{i_2}V_{j_2}F_k}) \right) \geq 0$$

$$\theta^2_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} \cdot \left(\sum_{k=1}^{N_f} y_{C_{i_2}V_{j_2}F_k} - \sum_{k_1=1}^{N_f} (y_{C_{i_1}V_{j_1}F_k} + S_{C_{i_1}V_{j_1}F_k} \cdot T_{C_{i_1}V_{j_1}F_k}) \right) \geq 0$$

Each of the terms

$$\begin{aligned} &\theta^1_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} \cdot y_{C_{i_1}V_{j_1}F_k} \\ &\theta^1_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} \cdot y_{C_{i_2}V_{j_2}F_k} \\ &\theta^1_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} \cdot S_{C_{i_2}V_{j_2}F_k} \\ &\theta^2_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} \cdot y_{C_{i_2}V_{j_2}F_k} \\ &\theta^2_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} \cdot y_{C_{i_1}V_{j_1}F_k} \\ &\theta^2_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} \cdot S_{C_{i_1}V_{j_1}F_k} \end{aligned}$$

needs further linearization. To this end we introduce 6 variables $L^1_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}k}$, $L^2_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}k}$, $L^3_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}k}$, $L^4_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}k}$, $L^5_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}k}$, $L^6_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}k}$ which are set equal to the above terms respectively. Note that L_1, L_2, L_4 and L_5 are integer variables while L_3, L_6 are binary variables. We will show the linearization for L_1, L_3 and the rest of the variables are linearized in a similar manner. For $L^1_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}k}$ we have

$$\begin{aligned} L^1_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}k} - TB \cdot \theta^1_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} &\leq 0 \\ -y_{C_{i_1}V_{j_1}F_k} + L^1_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}k} &\leq 0 \\ y_{C_{i_1}V_{j_1}F_k} - L^1_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}k} + TB \cdot \theta^1_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} &\leq TB \end{aligned}$$

For $L^3_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}k}$ we have

$$\begin{aligned} \theta^1_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} + S_{C_{i_2}V_{j_2}C_{i_2}V_{j_2}} &\leq L^3_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}k} + 1 \\ \theta^1_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} + S_{C_{i_2}V_{j_2}C_{i_2}V_{j_2}} &\geq 2 \cdot L^3_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}k} \end{aligned}$$

Then, constraint $\mathcal{C}_1$ or $\mathcal{C}_2$ becomes linear as follows:

$$\sum_{k=1}^{N_f} (L^1_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}k} + L^2_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}k} + L^3_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}k} + L^4_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}k} + L^5_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}k} + L^6_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}k}) \geq 0$$

Part 3

We introduce two binary variables $\delta^1_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}}$, $\delta^2_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}}$. Then the formula can be replaced by the

following relations:

$$
\begin{aligned}
ST_{C_{i_1}V_{j_1}} \geq END_{C_{i_2}V_{j_2}} - TB \cdot (1-\delta^1_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}}) \\
ST_{C_{i_1}V_{j_1}} \leq END_{C_{i_2}V_{j_2}} + TB \cdot \delta^1_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} \\
ST_{C_{i_2}V_{j_2}} \geq END_{C_{i_1}V_{j_1}} - TB \cdot (1-\delta^2_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}}) \\
ST_{C_{i_2}V_{j_2}} \leq END_{C_{i_1}V_{j_1}} + TB \cdot \delta^2_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} \\
1 - Conc_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} \geq \delta^1_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} \\
1 - Conc_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} \geq \delta^2_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} \\
1 - Conc_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} - \delta^1_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} - \delta^2_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} \leq 0
\end{aligned}
$$

By replacing $ST_{C_{i_1}V_{j_1}}$, $END_{C_{i_1}V_{j_1}}$, $ST_{C_{i_2}V_{j_2}}$ and $END_{C_{i_2}V_{j_2}}$ in the first four relations with their equivalent notation we get the following four relations which are all linear

$$
\begin{aligned}
\sum_{k=1}^{N_f} y_{C_{i_1}V_{j_1}F_k} \geq \sum_{k=1}^{N_f} (y_{C_{i_2}V_{j_2}F_k} + S_{C_{i_2}V_{j_2}F_k} T_{C_{i_2}V_{j_2}F_k}) - \\
-TB \cdot (1-\delta^1_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}}) \\
\sum_{k=1}^{N_f} y_{C_{i_1}V_{j_1}F_k} \leq \sum_{k=1}^{N_f} (y_{C_{i_2}V_{j_2}F_k} + S_{C_{i_2}V_{j_2}F_k} T_{C_{i_2}V_{j_2}F_k}) + \\
+TB \cdot \delta^1_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} \\
\sum_{k=1}^{N_f} y_{C_{i_2}V_{j_2}F_k} \geq \sum_{k=1}^{N_f} (y_{C_{i_1}V_{j_1}F_k} + S_{C_{i_1}V_{j_1}F_k} T_{C_{i_1}V_{j_1}F_k}) - \\
-TB \cdot (1-\delta^2_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}}) \\
\sum_{k=1}^{N_f} y_{C_{i_2}V_{j_2}F_k} \leq \sum_{k=1}^{N_f} (y_{C_{i_1}V_{j_1}F_k} + S_{C_{i_1}V_{j_1}F_k} T_{C_{i_1}V_{j_1}F_k}) + \\
+TB \cdot \delta^2_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}}
\end{aligned}
$$

Part 4

We introduce the binary variables $b^{1-1}_{C_{i_1}V_{j_1}C_{i_2}V_{j_2})}, b^{1-2}_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}}$. The product term $Conc_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} \cdot S_{C_{i_1}V_{j_1}F_k}$ is linearized as follows:

$$
\begin{aligned}
Conc_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} + S_{C_{i_1}V_{j_1}F_k} \leq \quad & b^{1-1}_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} + 1 \\
Conc_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} + S_{C_{i_1}V_{j_1}F_k} \geq \quad & 2 \cdot b^{1-1}_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}}
\end{aligned}
$$

In a similar way the term $Conc_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} \cdot S_{C_{i_2}V_{j_2}F_k}$ is linearized as follows:

$$
\begin{aligned}
Conc_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} + S_{C_{i_2}V_{j_2}F_k} \leq \quad & b^{1-2}_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} + 1 \\
Conc_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} + S_{C_{i_2}V_{j_2}F_k} \geq \quad & 2 \cdot b^{1-2}_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}}
\end{aligned}
$$

Consequently we get:

$$
\sum_{k=1}^{N_f} \left((b^{1-1}_{C_{i_1}V_{j_1}F_k} + b^{1-2}_{C_{i_2}V_{j_2}F_k}) \cdot W(F_k) \right) \leq W(TAM)
$$

Part 5

Let us see the case of three tests running concurrently. The constraint is re-written as follows: for any three cores $C_{i_1}, C_{i_2}, C_{i_3}$ which are tested concurrently at supply voltages $V_{j_1}, V_{j_2}, V_{j_3}$ and are connected to the same TAM resource the sum of their weights should not exceed the capacity of the TAM resource. Three different tests are executed concurrently (at least a common part of all of them) when every possible pair of them has a part which executes in parallel, i.e. when $Conc_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} \cdot Conc_{C_{i_1}V_{j_1}C_{i_3}V_{j_3}} \cdot Conc_{C_{i_2}V_{j_2}C_{i_3}V_{j_3}} = 1$. Then the constraint is written as follows:

$$
\begin{aligned}
Conc_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} \cdot Conc_{C_{i_1}V_{j_1}C_{i_3}V_{j_3}} \cdot Conc_{C_{i_2}V_{j_2}C_{i_3}V_{j_3}} \cdot \\
\left(\sum_{k=1}^{N_f} (S_{C_{i_1}V_{j_1}F_k} + S_{C_{i_2}V_{j_2}F_k} + S_{C_{i_3}V_{j_3}F_k}) W(F_k) \right) \leq \\
\leq W(TAM)
\end{aligned}
$$

At first we need to linearize the product term $Conc_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} \cdot Conc_{C_{i_1}V_{j_1}C_{i_3}V_{j_3}} \cdot Conc_{C_{i_2}V_{j_2}C_{i_3}V_{j_3}}$. To this end we introduce one binary variable $p_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}C_{i_3}V_{j_3}}$ and we set the following relations:

$$
\begin{aligned}
Conc_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} + Conc_{C_{i_1}V_{j_1}C_{i_3}V_{j_3}} + \\
+Conc_{C_{i_2}V_{j_2}C_{i_3}V_{j_3}} \leq p_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}C_{i_3}V_{j_3}} + 2 \\
Conc_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}} + Conc_{C_{i_1}V_{j_1}C_{i_3}V_{j_3}} + \\
+2Conc_{C_{i_2}V_{j_2}C_{i_3}V_{j_3}} \geq 4 \cdot p_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}C_{i_3}V_{j_3}}
\end{aligned}
$$

Using the above relations we have:

$$
\begin{aligned}
\sum_{k=1}^{N_f} \left((S_{C_{i_1}V_{j_1}F_k} + S_{C_{i_2}V_{j_2}F_k} + S_{C_{i_3}V_{j_3}F_k}) \cdot W(F_k) \right) \cdot \\
\cdot p_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}C_{i_3}V_{j_3}} \leq W(TAM)
\end{aligned}
$$

The product terms $p_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}C_{i_3}V_{j_3}} S_{C_{i_1}V_{j_1}F_k}$, $p_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}C_{i_3}V_{j_3}} S_{C_{i_2}V_{j_2}F_k}$, $p_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}C_{i_3}V_{j_3}} S_{C_{i_3}V_{j_3}F_k}$ need a final step of linearization. Therefore we introduce binary variables $b^{2-1}_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}C_{i_3}V_{j_3}k}$, $b^{2-2}_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}C_{i_3}V_{j_3}k}$, $b^{2-3}_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}C_{i_3}V_{j_3}k}$ for each scan frequency F_k and we have the following three pairs of relations

$$
\begin{aligned}
p_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}C_{i_3}V_{j_3}} + S_{C_{i_1}V_{j_1}F_k} \leq \quad & b^{2-1}_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}C_{i_3}V_{j_3}k} + 1 \\
p_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}C_{i_3}V_{j_3}} + S_{C_{i_1}V_{j_1}F_k} \geq \quad & 2b^{2-1}_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}C_{i_3}V_{j_3}k} \\
p_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}C_{i_3}V_{j_3}} + S_{C_{i_2}V_{j_2}F_k} \leq \quad & b^{2-2}_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}C_{i_3}V_{j_3}k} + 1 \\
p_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}C_{i_3}V_{j_3}} + S_{C_{i_2}V_{j_2}F_k} \geq \quad & 2b^{2-2}_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}C_{i_3}V_{j_3}k} \\
p_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}C_{i_3}V_{j_3}} + S_{C_{i_3}V_{j_3}F_k} \leq \quad & b^{2-3}_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}C_{i_3}V_{j_3}k} + 1 \\
p_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}C_{i_3}V_{j_3}} + S_{C_{i_3}V_{j_3}F_k} \geq \quad & 2b^{2-3}_{C_{i_1}V_{j_1}C_{i_2}V_{j_2}C_{i_3}V_{j_3}k}
\end{aligned}
$$

Finally we have:

$$\sum_{k=1}^{N_f} \Big((b^{2-1}_{C_{i_1} V_{j_1} C_{i_2} V_{j_2} C_{i_3} V_{j_3} k} + b^{2-2}_{C_{i_1} V_{j_1} C_{i_2} V_{j_2} C_{i_3} V_{j_3} k} + \\ + b^{2-3}_{C_{i_1} V_{j_1} C_{i_2} V_{j_2} C_{i_3} V_{j_3} k}) \cdot W(F_k) \Big) \leq W(TAM)$$

In a similar way we can bound the concurrency of any four, five, etc. number of tests, depending on the maximum number of cores which are connected to any TAM resource.