

## Power-Conscious Multi-Frequency Modular Testing of SoCs with Dynamic Reconfiguration of Multi-Port ATE\*

Dan Zhao and Ronghua Huang  
Center for Advanced Computer Studies  
University at Louisiana at Lafayette  
{dzhao,rhx2888}@cacs.louisiana.edu  
Tel: 337-4826875, Fax: 337-4825791

Hideo Fujiwara  
Graduate School of Information Science  
Nara Institute of Science and Technology  
fujiwara@is.naist.jp  
Tel: +81-743-725220, Fax: +81-743-725229

**Abstract**— With the debut of a new class of multi-port ATE (e.g., Agilent 93000 series), there is a pressing need for test planning methods to fully adapting SoC test framework design to the new concurrent test capabilities and fulfil emerging demands of high-speed testing. In this paper, we propose a new test planning strategy that addresses multi-frequency SoC testing by dynamically reconfiguring ATE ports. The system integrators on-the-fly group pins into virtual ports while ATE ports simultaneously drive the testing of a set of cores at multiple independent clock domains. An effective and efficient system optimization technique is developed to manage test resources and improve test efficiency for modern complex SoC designs.

**keywords:** Concurrent Test, Multi-Frequency SoC Test, Test Resource Partitioning, ATE Port Reconfiguration, Constrained Scheduling

### I. INTRODUCTION

State-of-the-art Systems-on-Chip (SoCs) embed pre-designed and pre-verified functional modules from various intellectual-property (IP) providers, and integrate them within a single silicon system to provide complex functionality, high performance, low power and small form factor. As the complexity, heterogeneity and speed of SoCs continue to rise rapidly, the test cost-per-transistor, unlike manufacturing cost-per-transistor, has not tracked Moore's Law. The SoC trends that increase test cost are: (1) the number of cores and their terminals are increasing much faster than the chip pins that limits ATE access to the IP cores; (2) An increase in speed-related defects acquires at-speed fault detection; (3) the increasing heterogeneous and hierarchical integration pushes the need for different test speeds to address multi-clock domain issues among different cores; (4) different core requires different test method and test application time, and thus requires tremendous flexibility in test architecture; (5) longer testing time accounts for increasing complex SoC designs.

To ensure that test cost scales with Moore's Law, it calls for a new design-for-concurrent-test strategy where an SoC is viewed as a collection of embedded heterogeneous cores with different testing requirements, tested concurrently. The

move from traditional shared-resource ATE architecture to new multi-port ATE architecture (such as Agilent 93000 series [1]) allows that multiple pins are grouped into virtual ports to test individual cores in parallel. The ports will be immediately re-configured upon completion of one concurrent test session and initiate the next set of concurrent core testing until the SoC is completely tested. Moreover, multiple ports can independently operate multiple tests at different test speeds. Such a flexible per-port architecture allows different test pins to operate in different modes to fulfil comprehensive SoC test requirements thus improving test efficiency. To adapt an SoC design to ATE concurrent test capabilities, the system integrator needs to design appropriate test access architecture (TAM) to transport test data to the embedded core-under-test (CUT). The CUT needs to be completely isolated (from the rest of the chip) and independently accessed via core test wrapper (e.g. IEEE std. 1500 wrapper).

A significant amount of research has been conducted [2, 3, 4, 5, 6, 7] as shown in the literature to design and optimize core test wrappers and/or TAMs. Consequently, constraint-driven test scheduling problems have been studied to minimize SoC test cost in terms of test application time. Such optimization problems have been proven *NP*-hard by reducing into any instance of Bin Design and Multi-Processor Scheduling problems and thus fast heuristics have been developed to solve it. However, all these approaches are only applicable to single frequency modular SoC testing where all cores are tested at a single low ATE clock. Modern SoCs are typically embedded with modular IP cores operated in multiple clock domains and moreover, multiple internally generated clocks. To improve test efficiency, using multiple frequencies is a benefit over single frequency testing due to the ability to offer comprehensive fault detection.

Recently, a few initial attempts [8, 9, 10, 11] have been made to address multi-frequency wrapper/TAM design and optimization for multi-clock domain SoC testing. However, these approaches assume that the ATE delivers test data at a single data rate. With the debut of a new class of multi-port ATE, there is a pressing need for test planning methods to fully exploit the new concurrent test capabilities of these ATEs and fulfil emerging demands of high-speed testing. A recent work [12] on TAM optimization has introduced the use of dual-

\*This research is supported in part by LA BORSF Research Competitive Subprogram.

speed ATEs that drives the channels at two different speed. The test scheduling is performed by packing rectangle tests into pre-partitioned high-speed and low-speed bins. Although this approach doesn't fully utilize the flexible per-pin architecture that supports tremendous capability to dynamically reconfigure ports, this work is a promising first step in this direction.

We propose a new test planning strategy that addresses multi-frequency SoC testing by fully exploiting multi-port ATE concurrent test capabilities. As the per-pin architecture provides the flexibility to dynamically match ATE ports to SoC's pin-out, the system integrator can on-the-fly group pins into virtual ports to test individual cores that have various testing requirements as illustrated in Figure 1. These virtual ports simultaneously drive the testing of a set of cores at multiple independent clock domains. Upon completion of any test application, the freed-up pins are dynamically reconfigured to initiate other tests immediately. Multi-frequency interface (MFI) is properly designed and inserted when there is a mismatch between the ATE port capability and core test data rate. Virtual test buses are connected to the core terminals via MFI. The bandwidth matches at both sides of MFI. An efficient test resource management technique should be developed, that involves various aspects, such as dynamic ATE port reconfiguration, the routing of TAMs, the design of multi-frequency test interface, the configuration of core wrapper scan architecture, and the distribution of bandwidth and power budget. In this paper, we propose a system level optimization technique to ease test integration, to facilitate concurrent test to the maximum extent, and to efficiently partition test resources to fulfil demanding performance and cost challenges.

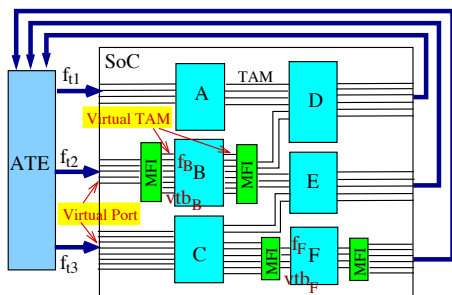


Fig. 1. Hypothetical illustration of concurrent test planning with multi-port ATE.

The rest of the paper is organized as follows. In Sec. IIA, we design a multi-frequency interface for virtual port configuration and virtual TAM assignment. The core wrapper configuration is studied in Sec. IIB. Then we formulate the system optimization problem into 3-D bin packing in Sec. IIC and propose an efficient heuristic algorithm to design and optimize the test framework with resource partitioning and test scheduling to minimize test cost in Sec. IID. The promising of the proposed optimization technique is confirmed in Sec. IIE where we run extensive simulation and comparison. Finally we conclude the paper in Sec. III.

## II. MULTI-SPEED TAM DESIGN WITH PORT RECONFIGURATION

### A. Design of Multi-frequency Interface

The multi-frequency interface design problem is basically to determine virtual port configuration on one hand and core wrapper configuration on the other hand.

Due to the heterogeneity of SoC integration, different cores may have different test requirement, and thus be tested at different clock rate. Some cores require specific test data rate and are driven by certain ATE ports. While the others do not set restriction on scan speed thus have the flexibility to operate under various clock domains and can be assigned to any port. A set of distinct frequencies derived from ATE clocks (using hardware division logic) forms the candidate frequency set for the cores. To provide a wide range of frequency selection improves test efficiency from two aspects. On one hand, it can save power dissipation when setting core test data rate lower than the ATE port. On the other hand, it can reduce the test time by running at higher clock rate. However, multi-frequency testing results in a frequency gap at the core interface between core wrapper scan architecture and core-external test access mechanism (TAM), and accordingly low utilization of bandwidth (defined as the product of data frequency and data transportation width). In order to resolve the mismatch between ATE capability and core test speed, a multi-frequency test interface (MFI) is designed that synchronizes input/output data and transfers test patterns/responses into/out of the corresponding scan-enabled core.

Assume the ATE supports  $N_{pt}$  ports at various clock domains  $f_{t_i}$ ,  $i \in N_{pt}$ . Using port  $pt_i$  as an example that consists of a set of ATE channels at clock domain  $f_{t_i}$ . The test data is transported from/to certain ATE channels along parallel TAMs with width of  $W_{tam_i}$  and frequency of  $f_{t_i}$  (where  $W_{tam_i}$  is the width of port  $pt_i$ ). The bandwidth  $W_{tam_i} \times f_{t_i}$  is distributed to one or multiple cores at distinct frequency via pairs of multi-frequency interface where it is necessary. A ATE port is thus divided into several virtual ports, each connecting to a core. Virtual TAMs connect core (say  $B$ ) terminals to MFI at chosen width and core test frequency of  $w_{vtb_B}$  and  $f_B$  respectively. The dedicated TAM width for a core (or the virtual port width) is determined by  $w_{vp_B} = \frac{w_{vtb_B} \times f_B}{f_{t_i}}$ . As multiple cores can be assigned to the same ATE port and tested concurrently, it should satisfy that  $\sum_{j=1}^m w_{vp_j} \leq W_{tam_i}$ . Note that, as the chip level TAM width  $W_{ext}$  is constrained by the SoC pin count,  $\sum_{i=1}^{N_{pt}} W_{tam_i} \leq W_{ext}$ .

The bandwidth matching is performed in consequence via multi-frequency interface (e.g. *MUX-DeMUX*). If test data rate  $f_B$  for core  $B$  is higher than  $f_{t_i}$ , we insert a MUX before core input terminals, and multiplex  $\lceil \frac{f_B}{f_{t_i}} \rceil \times w_{vtb_B}$  bits test data at  $f_{t_i}$  into  $w_{vtb_B}$  bits test data at  $f_B$ . On the other hand, if  $f_B$  for core  $B$  is lower than  $f_{t_i}$ , we insert a DeMUX instead, and de-multiplex  $w_{tam_B}$  bits test data at  $f_{t_i}$  into  $w_{tam_B} \times \lfloor \frac{f_{t_i}}{f_B} \rfloor$  bits test data at  $f_B$ . To observe test responses, we insert DeMUX/MUX accordingly after core output terminals. Multi-frequency interface design facilitates SoC test cost reduction in

a way that co-optimizes both core wrapper configuration, core test frequency selection and virtual port partitioning to achieve the best tradeoff among them.

### B. Core Wrapper Configuration

Core wrapper architecture is configured to minimize the test time by constructing wrapper scan chains in a way that their length are well balanced as the longest wrapper scan chain dominates the test time. The wrapper scan width of a scan-testable core is adapted to the core-external TAM width (i.e.  $w_{vtb}$ ) by serially connecting core inputs/output terminals to the internal scan chains. In order to balance the wrapper scan chain lengths so as to minimize the maximum wrapper scan chain length ( $L_{max}$ ), bin design based fast heuristics such as FFD and BFD [2, 6] are proposed to solve it. Thus we obtain a finite set of wrapper configuration candidate set for each core with decreasing  $L_{max}$  at increasing  $w_{vtb}$ , i.e.  $R_i(w_{vtb_i}, L_{max_i})$  (where  $R_i$  denotes a distinct wrapper configuration). As the test time for a core given test pattern is a function of the longest wrapper scan chain length and the scan frequency, *the increase of wrapper scan chain width itself decreases the test time, while the test time reduces with the increase of frequency*. With the flexibility to choose from a candidate set, it facilitates efficient test scheduling where the most suitable wrapper configuration will be selected for a core to fit into the available idle space.

An important observation is made from the wrapper candidate set of a core. That is doubling the width of a core, the rectangle area obtained by multiplying  $L_{max}$  and  $w_{vtb}$  increases or remains the same. This observation has been confirmed with all scan-testable cores in ITC SoC benchmarks [13]. No doubt, a fact arises that *by halving the shift frequency the shift time may increase or remain the same when matching the bandwidth*. With this feature, more flexible scheduling may be achieved by relieving tight power budget without increasing the test time as explained next in the optimization approach.

### C. Problem Formulation

In this section, we define the power-conscious multi-speed TAM design (*PMFT*) problem and formulate it into 3-D bin packing.

Without loss of generality, we assume an SoC model  $S$  embedded with  $N_c$  IP cores  $C = \{c_i | i = 1 \dots N_c\}$ . Each core is given a set of test parameters including the number of core input/output/bidirectional terminals, the number of test patterns, the number of internal scan chains and their lengths, the test power obtained at maximum allowable frequency, and the functional frequency. The cores are given a set of test frequency candidates derived from the ATE clocks. Each core can select a proper wrapper design at certain test data rate, thus a core  $c_i$  is expressed as a three-tuple  $c_i = \{t_i, p_i, w_{vp_i}\}$ , where  $t_i$  is the test time obtained at width of  $w_{vtb_i}$  and test frequency  $f_{c_i}$ ,  $p_i$  is the test power of  $c_i$  dissipated at frequency  $f_{c_i}$ ,  $w_{vp_i}$  is the virtual port width assigned to core  $c_i$ . In addition, the SoC is given a chip level TAM width of  $W_{ext}$  and a

power budget of  $P_{ave}$ . Assume a multi-port ATE supports up to  $N_{pt}$  ports, each corresponding to a independent clock domain  $f_{t_i}$ ,  $i \in N_{pt}$ . The port width is dynamically assigned during scheduling. But the total port width should not exceed  $W_{ext}$ . The optimization problem is stated as follows:

***PMFT Problem***: Given an SoC model  $S$  with  $N_c$  IP cores, a chip level TAM width  $W_{ext}$ , and the maximum average power allowance  $P_{ave}$ , and given a multi-port ATE with up to  $N_{pt}$  distinct clock domains, determine (1) dynamic grouping of ATE channels into several ports to deliver test data at different test speed, (2) the multi-frequency interface, test data rate and wrapper configuration of each core, and (3) a constrained test scheduling that parallel routes multiple cores on TAMs such that the overall SoC test time is minimized while satisfying power and chip level TAM constraints at any time.

We define a 3-D bin with the height of the overall SoC test time, and its length and width bounded by the power  $P_{ave}$  and TAM width  $W_{ext}$  constraints respectively. The cubes of cores may overlap in time dimension for concurrent testing but not the power and TAM width dimensions. The *PMFT* problem is thus reduced into 3-D bin packing: Given a set of core tests represented in cubes, find a way to packing the cubes into a 3-D bin bounded by the bottom, so as to minimize the height to which the cubes fill the bin. It is easy to reduce such bin packing problem into any instance of the partitioning problem [14], thus is *NP*-hard. An effective and efficient heuristic algorithm is developed thereafter.

### D. Advanced Shelf Packing Based Optimization Algorithm

The proposed algorithm has four major steps, namely, initial packing in shelf  $S_1$ , ceiling packing in idle bins, floor packing in shelf  $S_2$ , and halved floor packing. In this section, we give an intuitive description of the steps and illustrate the approach with a hypothetical example.

A pre-processing step is performed first to obtain the candidate wrapper set for each core  $R_i^k = \{w_{vtb_i}^k, L_{max_i}^k\}$  by running best-fit decreasing heuristic. Different combinations of  $w_{vtb_i}^k$  and  $L_{max_i}^k$  provides the flexibility to make the trade-off between several interdependent design items, such as test power, test time and core-external TAM width, thus results in the best selection possible in terms of the configuration of the three dimension of a cube.

#### Step 1: Initial Packing in Shelf $S_1$

We first obtain an ordered cube list to initiate packing. We try to find the maximum possible test data rate  $f_{c_i-max}$  for core  $c_i$  from a range of frequency selection at which testing is performed without exceeding the power constraint, i.e.,  $p_i(f_{c_i-max}) = \frac{f_{c_i-max} \times Pow_i}{F_{max}} \leq P_{ave}$ . Then we find a proper wrapper design via bandwidth matching, i.e.,  $w_{vtb_i} \leq \frac{\max\{f_{t_i}\} \times W_{ext}}{N_{pt} \times f_{c_i-max}}$ , which results in the initial test time  $t_{i-init}$  and virtual port width  $w_{vp_i}$  for  $c_i$ , respectively. After we obtain the initial setting of a cube, an additional checking process is applied. We reduce the width of those cubes whose initial width exceeds  $\beta W_{ext}$  (where  $0.5 < \beta < 1$ ) such that their

newly selected widths mostly close to  $\beta W_{ext}$  without exceeding it. Their initial test times are updated accordingly. We may further reduce the test power for a core without affecting its initial test time by employing the observation as discussed in Sec. IIB. If the initial test time remains the same each time when halving the frequency, the test power and virtual port width will be updated as well. Note that, the new width should not exceed  $\beta W_{ext}$ . By freeing up more idle power, we may accommodate more cubes at one time without exceeding tight power budget. Two ordered lists are obtained, one is in the descending order of  $t_{i\_ini}$ , i.e.  $L_t = \{t_{i\_ini}\}$  and the other in the descending order of  $w_{vp_i}$ , i.e.  $L_w = \{w_{vp_i}\}$ .

We start packing in a way that picks the cubes one by one from  $L_w$  whose width falls within the range of  $0.5W_{ext}$  to  $\beta W_{ext}$  and stack at the bottom of the bin one on top of the other in time dimension. A hypothetical example in Figure 2 illustrate the schedule after the first step.

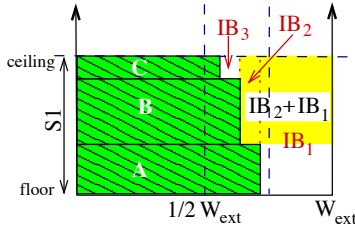


Fig. 2. Hypothetical illustration of initial packing.

### Step 2: Ceiling Packing in Idle Bins

After the first step, we say the allocated cubes are in the first shelf of the bin and we name the bottom edge, *the shelf floor* and the top edge, *the shelf ceiling*. In order to efficiently utilize the idle space left in this shelf, we divide the idle space into several idle bins  $IB[1..N_{ib}]$  from right to left as shown in Figure 2. The height, length and width of each idle bin is determined by the size of the associated cubes. For instance, cores  $A$ ,  $B$  and  $C$  are packed into the shelf from floor to ceiling, where  $t_B > t_A > t_C$ ,  $w_{tp_A} > w_{tp_B} > w_{tp_C}$  and  $p_B > p_C > p_A$ . The height of  $IB_1$  is the height of the shelf, i.e. the sum of the test time of these cores  $H(IB_1) = t_A + t_B + t_C$ . Its width is determined by the longest cube width  $W(IB_1) = W_{ext} - w_{vp_A}$ , and its length is determined by the most power-consuming core  $L(IB_1) = P_{ave} - p_B$ . Similarly, we can calculate the size of other idle bins. In order to best-fit unscheduled cubes into these idle bins, we consider a total of  $N_{ib}$  combinations of idle bins in the order of  $\{IB_1\}$ ,  $\{IB_2 + IB_1\}$ ,  $\{IB_3 + IB_2 + IB_1\}$ , ...,  $\{IB_{N_{ib}} + IB_{N_{ib}-1} + \dots + IB_1\}$ . The size of each combination of the idle bins is computed accordingly. For a particular combination, its height and length are determined by the smallest height idle bin in the group and its width is the sum of the width of these idle bins as shown in Figure 2.

We start allocating unscheduled cubes in the order of  $L_t$  and try to fit one at a time in the sequence of all combinations of idle bins. The idea is that we pick the first available combination of idle bins,  $CIB_j$   $j \in N_{ib}$ , and search through the core

list to find the best-fit cube that can utilize the idle bin to the maximum extent. If no cube fits, we try the next combination, so on and so forth.

More specifically, for a certain combination of idle bins, say  $CIB_j$ , we pick the first unscheduled core in  $L_t$  whose height fits  $CIB_j$ . Its width will be checked next. If the initial cube width exceeds the width of  $CIB_j$ , we try to reduce the wrapper scan chain width of the core in a way that the induced increase in test time won't lead to an excess of the height of  $CIB_j$ . If no suitable width can be found when we search through the candidate wrapper configuration list, we move on to the next unscheduled core in  $L_t$ , otherwise, we further check the power consumption. If the initial test power exceeds the length of  $CIB_j$ , we try to reduce the power by reducing its frequency following the frequency list. Again, the induced increase in test time should meet the height limit of  $CIB_j$ . If no proper frequency is found, we move on to the next core. Otherwise, the cube will be allocated in  $CIB_j$  justified from right to left with its top edge below the shelf ceiling as shown in Figure 3. If no suitable cube can be allocated in  $CIB_j$  after we search through list  $L_t$ , a new search of cube will be initiated to fit in the next combination of idle bins,  $CIB_{j+1}$ . After a cube is allocated, some idle bins may be eliminated while the remaining are updated with their length and width shortened. For example as shown in Figure 3, the previous  $IB_1$  is eliminated while  $IB_2$  and  $IB_3$  are updated by shortening its length and width. All available combinations of idle bins are updated by excluding  $IB_1$  and updating  $IB_2$  and  $IB_3$  as well.

We will repeat the above process until no unscheduled cores can be fit into the remaining idle space and we will move on to the third step.

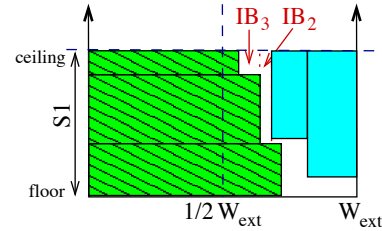


Fig. 3. Hypothetical illustration of ceiling packing.

### Step 3: Floor Packing in Shelf $S_2$

In this step, we pack the unscheduled cubes on top of the first shelf using a floor-packing approach. The basic idea is that a picked cube is packed left on floor justified on the second shelf  $S_2$  where it fits, and each subsequent ones adjacent to the one just packed as illustrated in Figure 4. More importantly, we need to find a way to pack into  $S_2$  as many cubes as possible while satisfying tight power and TAM width constraints.

Following the order in  $L_t$ , we start allocating unscheduled cubes in shelf  $S_2$  from the highest cube. The height of  $S_2$ ,  $H(S_2)$ , is determined by the test time of the first allocated cube. Then we try to pack the next highest possible unscheduled cubes to use up the remaining idle TAM width and power. A fast process to check if a cube can be contained in the shelf

is developed. For an incoming cube, we first check if it satisfies the power constraint at the minimum available frequency,  $\min\{f_i\}$ . If it satisfies power limit, we further check if it can meet the height allowance at the initial frequency,  $f_{c_i-max}$ . If it satisfies again, we will pick the best frequency within the range of  $(\min\{f_i\}, f_{c_i-max})$  such that its test time at certain wrapper configuration is closest to  $H(S_N)$  but not exceeding it. The reason is simply to free up more TAM width and power so as to contain more cores. The above process continues until there is no more idle space available or there are no cores left for packing.

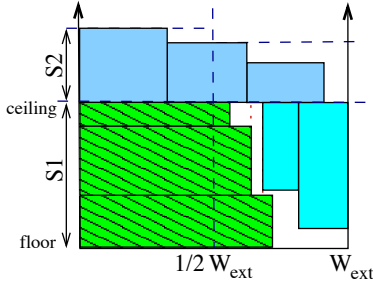


Fig. 4. Hypothetical illustration of floor packing.

#### Step 4: Halved Floor Packing

After we finish scheduling in shelf  $S_2$ , we partition the 3-D bin into two halves. Let  $S_{2L}$  denote the left half shelf with height of  $H(S_2)$ . Let  $S_{2R}$  denote the right half, and its height is determined by the highest cube in the right half (or partially in the right half) as shown in Figure 5. All subsequent packing in the left and right halves will occur above the ceiling of these half shelves. Each time we choose the half whose ceiling is lower and create a new half shelf on top of it. We pack the cubes horizontally from left to right into this half bin by applying the floor packing approach as described in Step 3. If no cubes can be fit into this half shelf, we create a new half shelf on top of the half bin whose ceiling is lower. We repeat these process until there are no more cubes.

The overall SoC test time is determined by the higher half bin. For example in Figure 5, the SoC test time is determined by the height of left half bin, i.e.,  $T_{SoC} = H(S_1) + H(S_{2L}) + H(S_4)$ .

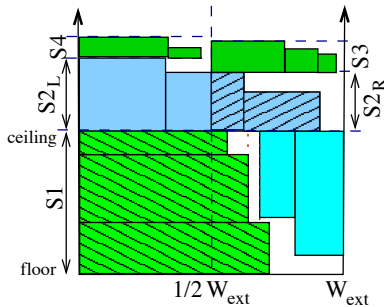


Fig. 5. Hypothetical illustration of halved floor packing.

#### E. Simulation Study

We evaluate the proposed algorithm by running simulations on ITC'02 SoC test benchmarks d281 and d695 where the test power parameters are provided [13]. Each SoC model is embedded with a set of IP cores. Each core is provided a set of test parameters, including the number of inputs, outputs and bidirectional terminals, the number of test patterns, the number of scan chains and their length, the test power and the associated test frequency. Some cores have a particular requirement on test data rate while the others set no restriction thus can perform testing at various clock domains. A range of candidate frequencies are provides for selection. The example ATE drives test data at three distinct clock domains, 50MHz, 120MHz and 250MHz. We run experiments with chip level TAM width  $W_{ext}$  changing from 16 to 64 (pin count from 32 to 128) while average power budget at 1500, 1800, 2000 and 2500 respectively. The simulation results are listed in Tables I, II.

From the simulation results, we can see that overall SoC testing time reduces when relaxing the power budget or SoC pin-count constraint. When comparing the test time of our approach to the one reported in [10], our approach outperforms this best-fit decreasing based heuristic, that could be reduced to an instance of 3-D bin packing problem. The improvement can reach as high as 44.15%. The reason is simply because we fully utilize the new concurrent test capabilities of multi-port ATE where multiple ports may deliver test data at distinct clock domains. While the approach in [10] assumes that the ATE delivers test data at a single data rate. Our proposed algorithm requires a negligible amount of computation time (in *ms*) and therefore is suitable for more complex SoC designs. This is especially an improvement over the CPU-intensive ILP-based method [12].

### III. CONCLUSION

We have presented in this paper a novel power-aware multi-speed TAM design by dynamically reconfiguring ATE ports. We have formulated the power-constrained multi-frequency TAM optimization problem into a 3-D bin packing problem. We have further proposed an efficient advanced shelf-packing based heuristic algorithm to manage the test resource partitioning from various aspects, such as dynamic ATE port reconfiguration, the routing of TAMs, the design of multi-frequency interface, the configuration of core wrapper scan architecture, and distribution of bandwidth and power budget. By fully exploiting multi-port ATE concurrent test capabilities, SoC test efficiency is significantly improved and test cost is minimized.

### REFERENCES

- [1] "Agilent debuts multi-clock domain test capabilities for 93000 soc." [www.agilent.com/see/semitestnews](http://www.agilent.com/see/semitestnews).
- [2] E. J. Marinissen, S. K. Goel, and M. Lousberg, "Wrapper design for embedded core test," in *Proc. of ITC*, pp. 911–920, october 2000.

TABLE I  
OVERALL TEST TIME FOR SoC *d281*

$P_{ave}$	$W_{ext}$						
	16	24	32	40	48	56	64
1000	283.343	128.858	91.410	91.225	83.587	78.209	72.332
1500	279.336	121.116	80.270	77.752	77.304	48.576	41.920
1800	257.964	121.116	80.244	65.464	65.016	44.328	37.672
2000	176.444	120.516	80.244	57.070	38.240	35.800	31.042
2500	176.444	90.118	69.447	56.596	34.754	35.800	31.042

TABLE II  
OVERALL TEST TIME FOR SoC *d695*

$P_{ave}$		$W_{ext}$						
		16	24	32	40	48	56	64
1000	$t_{MPFT}$	498.364	314.060	270.768	266.232	174.868	152.164	151.36
	$t_{[10]}$	440.00	356.80	273.67	-	264.96	-	247.07
	$\delta T$ (%)	-13.26	11.98	1.06	-	34.00	-	38.41
1500	$t_{MPFT}$	402.700	263.520	222.137	218.921	131.824	123.432	104.304
	$t_{[10]}$	436.05	300.01	222.79	-	180.28	-	162.39
	$\delta T$ (%)	7.65	12.16	0.29	-	26.88	-	35.77
1800	$t_{MPFT}$	398.788	253.472	190.240	141.104	120.260	116.052	77.724
2000	$t_{MPFT}$	364.254	214.504	167.330	101.672	86.288	81.672	75.207
	$t_{[10]}$	433.87	304.90	217.31	-	151.37	-	123.76
	$\delta T$ (%)	16.05	29.65	22.30	-	42.99	-	39.23
2500	$t_{MPFT}$	363.588	214.504	154.488	91.056	84.540	81.672	63.352
	$t_{[10]}$	433.87	299.18	216.77	-	151.37	-	112.93
	$\delta T$ (%)	16.20	28.30	28.73	-	44.15	-	43.90

- [3] K. Chakrabarty, "Optimal test access architectures for system-on-a-chip," *ACM Trans. on Design Automation of Electronic Systems*, vol. 6, pp. 26–49, January 2001.
- [4] Y. Huang, S. Reddy, W. Cheng, P. Reuter, N. Mukherjee, C.-C. Tsai, O. Samman, and Y. Zaidan, "Optimal core wrapper width selection and SOC test scheduling based on 3-D bin packing algorithm," in *Proc. International Test Conf.*, pp. 74–81, October 2002.
- [5] E. Larsson and Z. Peng, "Test scheduling and scan-chain division under power constraint," in *Proc. Asian Test Symp.*, pp. 259–264, november 2001.
- [6] V. Iyengar, K. Chakrabarty, and E. J. Marinissen, "Co-optimization of test wrapper and test access architecture for embedded cores," *Journal of Electronic Testing: Theory and Applications*, vol. 18, pp. 213–230, April 2002.
- [7] D. Zhao and S. Upadhyaya, "Power constrained test scheduling with dynamically varied TAM," in *Proc. of VTS*, pp. 273–278, April 2003.
- [8] Q. Xu and N. Nicolici, "Wrapper design for testing IP cores with multiple clock domains," in *Proc. Design, Automation and Test in Europe Conf.*, pp. 416–421, February 2004.
- [9] A. Sehgal, V. Iyengar, M. D. Krasniewski, and K. Chakrabarty, "Test cost reduction for SOCs using virtual TAMs and lagrange multipliers," in *Proc. IEEE/ACM Design Automation Conf.*, pp. 738–743, June 2003.
- [10] T. Yoneda, K. Masuda, and H. Fujiwara, "Power-constrained test scheduling for multi-clock domain SoCs," in *Proc. Design, Automation and Test in Europe Conf.*, pp. 297–302, March 2006.
- [11] D. Zhao, U. Chandran, and H. Fujiwara, "Design and optimization of a power-aware multi-frequency wrapper architecture for modular IP cores," in *IEEE Asia and South Pacific Design Automation Conference*, pp. 714–719, January 2007.
- [12] A. Sehgal and K. Chakrabarty, "Efficient modular testing of socs using dual-speed tam architectures," in *Proc. IEEE/ACM Design, Automation and Test in Europe Conf.*, pp. 422–427, February 2004.
- [13] E. J. Marinissen, V. Iyengar, and K. Chakrabarty, "ITC'02 SOC test benchmarks." <http://www.hitech-projects.com/itc02socbenchm/>.
- [14] M. Garey and D. Johnson, *A Guide to the Theory of NP-Completeness*. Freeman, San Fransisco, CA.