Shelf Packing to the Design and Optimization of A Power-Aware Multi-Frequency Wrapper Architecture for Modular IP Cores*

Dan Zhao and Unni Chandran

Center for Advanced Computer Studies
University at Louisiana at Lafayette
Lafayette, LA 70504-4330, USA
Tel: +1-337-4826875
Fax: +1-337-4825791
e-mail: {dzhao,uxc0983}@cacs.louisiana.edu

Hideo Fujiwara

Graduate School of Information Science
Nara Institute of Science and Technology
Ikoma, Nara 630-0192, JAPAN
Tel: +81-743-725220
Fax: +81-743-725229
e-mail: fujiwara@is.naist.jp

Abstract— This paper proposes a novel power-aware multi-frequency wrapper architecture design to achieve at-speed testability. The trade-offs between power dissipation, scan time and bandwidth are well handled by gating off certain virtual cores at a time while parallelizing the remaining. A shelf packing based optimization algorithm is proposed to design and optimize the wrapper architecture while minimizing the test time under power and bandwidth constraints.

I. Introduction

Today's System-on-Chips (SoCs) embed hundreds of memories, different types of logic, and dozens of functional blocks obtained from various vendors, and moreover multiple clocks operating at multiple frequencies. This brings in its wake problem of defining a proper test strategy and optimizing test cost. Modular testing approach becomes attractive where IP modules are tested as stand-alone units, because its "divide-n-conquer" test development at core level helps reduce the test generation time and associated data volume [1]. It is even mandatory for non-logic and black-box third party cores [2].

Modular test of SoCs requires that the IP cores are surrounded by core test wrappers to facilitate core isolation and to ease test access. The wrappers support various configuration of wrapper cells, that allow core internal or external tests to be carried out via test access mechanisms (TAMs). Both the core test wrappers and TAMs form the on-chip test access infrastructure. The design of wrappers and TAMs has a large impact on SoC test cost specially the test application time. A significant amount of research [3, 4, 5, 6, 7] has been conducted in the design and optimization of core test wrappers and/or TAMs. However, most of these approaches address single frequency modular SoC testing irrespective of the fact that modern SoCs are embedded with modular IP cores operated (internally) in multiple clock domains [8]. To improve test cost, using multiple frequencies is a benefit over single frequency testing due to the ability to offer comprehensive fault detection when testing SoCs with multiple clocks and multiple frequencies. Support for multi-frequency testing requires significant improvements to the existing approaches.

*This research is supported in part by LA BORSF Research Competitive Subprogram, and Japan Society for Promotion of Science (JSPS) under grant S06089.

The move to nanoscale SoCs is expected to yield a higher percentage of speed-related defects, and scan-based (e.g., MUX-D) at-speed test or even beyond at-speed test [9] is desired as a cost-effective way to maintain test quality. Shifting the scan chains, however, may present the highest level of switching activities and hence the highest level of power consumption. A slow-shift and fast-functional operation [10] turns out to be the most practical method, where one can load/unload test data at a rate much slower than the launch and capture clock. In order to facilitate at-speed testability for modular SoC testing, DFT techniques are required to synchronize the external tester channels with the core's internal scan chains in the shift mode, and provide at-speed test control in the capture mode. As at-speed scan testing is becoming an increasingly critical component of the test framework, we will focus on the design of a multi-frequency wrapper architecture for IP modules to achieve at-speed testability.

The rest of the paper is organized as follows. The related work is discussed in Sec. II. We describe in Sec. III a novel power-aware multi-frequency wrapper architecture. In Sec. IV, we formulate the power-constrained multi-frequency wrapper design into a shelf packing problem and propose an efficient heuristic algorithm to optimize the wrapper scan architecture and minimize the test time of a core. The performance of the proposed algorithm is evaluated in Sec. VI with the experimental results and the comparison with the best existing approaches. We finally conclude the paper in Sec. VII.

II. Related Work

There are several types of wrapper structures, such as TestShell [3] and TestCollar [11]. IEEE Std. 1500 provides a standard but scalable and configurable wrapper [12] that is very similar to TestShell and TestCollar. The 1500 wrapper architecture comprises Wrapper Instruction Register (WIR), Wrapper Bypass Register (WBY) and Wrapper Boundary Register (WBR). It contains a mandatory single-bit wrapper serial port, and user-defined scalable multi-bit wrapper parallel port. Recently, a reconfigurable wrapper [4] and a hierarchical wrapper [7] have been presented to address various SoC test environment. A large number of approaches have been proposed in the literature [13, 14, 5] to design and optimize a single frequency core wrapper architecture. The problem is mainly con-

1-4244-0630-7/07/$20.00 ©2007 IEEE.

cerned with the construction of wrapper scan chains (WSC) for minimal test time. The wrapper scan chains are composed from the input/output wrapper cells and core-internal scan chains. When using single frequency core wrappers, the test time is a function of the longest WSC and the single frequency. It has been shown in [14] that the wrapper design problem for hard cores is equivalent to the well-known NP-hard problems of Bin Design and Multi-Processor Scheduling and various heuristics such as BFD can be used to solve it.

Only limited work has been done in multi-clock domain testing. They are focused on either BIST architecture [15, 16] or SoC level TAM design [17, 18, 19, 20]. Recently, a few initial attempts have been made to address the wrapper design for multi-clock domain SoC testing [21] that allow multiple clock domains to perform shift operation in parallel. However, high switching activity may result by shifting scan cells simultaneously and accordingly high average power consumption. Although power dissipation can be reduced by shifting at a lower frequency, such a parallel architecture restricts the trade-off between test power and scan time. As a result, it causes excessive increase in test time when the power constraints become tight, especially when the IP cores become more complex while the test pattern volumes grow even larger. In addition, no high-speed clock generation technique is proposed to achieve at-speed or beyond at-speed testability.

III. Design of A Power-Oriented Serial-Parallel Wrapper Architecture

In this section, we design a IEEE 1500-compliant multi-frequency wrapper architecture. For a given IP core, all scan chains should be ordered to minimize clock skew during shifting. All scan cells within the same clock domain are grouped into virtual cores. Each virtual core is assigned with a single frequency virtual wrapper, consisting of a set of wrapper scan chains configured into virtual TAMs. The virtual cores are connected via these virtual TAMs to the external TAM. The test data is transported from/to the external ATE along the SoC level TAMs with width of W_{tam} and at ATE frequency of f_t. In order to bridge the frequency gap at wrapper interface ports, the bandwidth (defined as the shift frequency times the data transportation width) is matched by *MUX-DeMUX* interface which synchronizes the input data and transfers the test patterns into the corresponding virtual core. Such amount of bandwidth $W_{tam} \times f_t$ is distributed to the virtual cores shifted at distinct frequencies. A scan control logic is required to configure a wrapper architecture under multiple frequencies and provide proper at-speed test controlling.

Gating off portions of scan chains during shift [22, 23] is a useful technique to power reduction. Special care should be taken to design power reduction circuitry at a minimal hardware overhead. Aiming at minimizing the test time of a IP module at power constraint, we propose the idea to gate off certain virtual cores, and allow serial shifting of the others. By serial shifting, we can save the scan time of a virtual core by increasing its shift rate to compensate for the test power consumption. With clock gating, the virtual cores are separated into several groups. Only one group will be activated in shift phase at a time while the virtual cores within the same group are shifted in parallel. Such a serial-parallel shifting scheme is realized by a two-level *MUX-DeMUX* pairs. The outer level *SMUX-SDeMUX* pair is for serial shifting among groups of virtual cores while the inner level *PMUX-PDeMUX* pair for parallel shifting within a particular group. Figure 1 illustrates such a power-aware multi-frequency serial-parallel wrapper architecture (*MFSP* for short). The example IP core contains five virtual cores. They are organized into three groups, each consisting of one or several virtual cores. These three groups will enter shift mode in sequence.

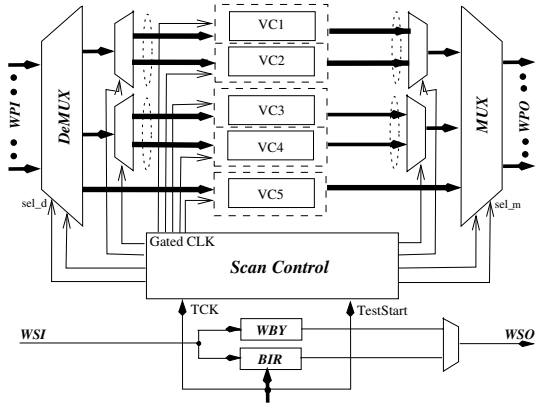


Fig. 1. Power-aware serial-parallel wrapper architecture.

Under *MFSP* architecture, the scan control block is designed that consists of a Capture Finite State Machine, the Clock Division Logic, the on-chip PLL, and a decoder to serialize the shift operation as shown in Figure 2. With reasonable hardware overhead, we realize at-speed test controlling solely on-chip.

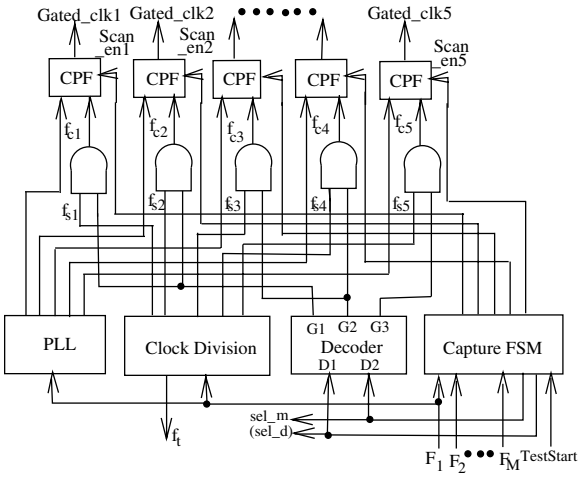


Fig. 2. Gated multi-frequency shifting.

The Capture FSM is used to generate $scan\_en$ signals, and control shifting sequence by feeding proper inputs to the decoder as well as the *DeMUX-MUX* interface selection signals. The Clock Division Logic is to generate a set of trial frequencies $F = \{F_1, F_2, ..., F_M\}$ which are provided for the virtual cores to be chosen as the shift frequency. In order to simplify the hardware implementation, the ratio of trial frequencies is set as two's exponent, i.e., $\frac{F_j}{F_{j+1}} = 2$. The decoder generates the gating signals that will gate off all other shift clocks except for those in the active group. We use a 2-of-3 decoder here as an example. By setting the combinations of the two inputs $(D2, D1)$, we may activate one group of virtual cores at a

TABLE I
TEST PARAMETERS FOR CORE $hCADT01$

num	f_{func} (MHz)	N_{in}	N_{out}	N_{bi}	Pow	N_{sc}	L_{sc_j}
1	200	109	32	72	2572	16	{168 168 166 166 163 163 163 163 162 162 162 162 151 151 151 151}
2	133	144	67	72	450	3	{150 150 150}
3	120	89	8	72	930	10	{93 93 93 93 93 93 93 93 93 93}
4	75	111	31	72	1314	6	{219 219 219 219 219 219}
5	50	117	224	72	2605	5	{521 521 521 521 521}
6	33	146	68	72	576	11	{82 82 82 81 81 81 18 18 17 17 17}
7	25	15	30	72	40	4	{10 10 10 10}

time. For example, when setting $(D2, D1)$=(0,1), the decoder outputs $(G3, G2, G1)$=(0,0,1), and only $Group-1$ activates and virtual cores VC_1 and VC_2 in $Group-1$ start shifting test data in/out of its wrapper scan chains pulsed at distinct $Gated\_clk1$ and $Gated\_clk2$ respectively. Similarly by setting $(D2, D1)$=(1,0) or (1,1), parallel scan-in and scan-out test data for the virtual cores in $Group-2$ and $Group-3$ respectively. When $(D2, D1)$ is set to (0,0), the at-speed clock pulses are applied and the responses are captured. The on-chip functional PLL is reused here to create independent high-speed clock signals used during the capture phase. After serial shifting has finished for the virtual cores, the $scan\_en$ signals are switched off, and at-speed clock pulses f_{c_i} are filtered out of $gated\_f_{s_i}$ using the clock pulse filters (CPF). The CPF ([24] describes one possible design) replaces clock multiplexer and is controlled simply by the $scan\_en$ signal.

IV. WRAPPER DESIGN PROBLEM FORMULATION

Without loss of generality, we assume that an embedded IP core C includes N_c virtual cores $VC = \{VC_i \,|i = 1...N_c\}$, each corresponding to one clock domain. Each virtual core VC_i is given a set of test parameters, e.g., the number of input N_{in}, output N_{out} and bidirectional N_{bi} terminals, the power consumption Pow obtained at the maximum allowable frequency F_{max}, the number of scan chains N_{sc} and their lengths L_{sc_j}. For instance, we show in Table I a representative IP core $hCADT01$ [21], which is used as a running example throughout this paper. It contains seven virtual cores, divided in terms of different functional frequency. A virtual core VC_i can select a wrapper design w_i at certain shift frequency f_{s_i}, thus VC_i is expressed as a three-tuple $VC_i = \{BW_i, p_i, t_i\}$. Here, $BW_i = w_i \times f_{s_i}$ is the bandwidth of VC_i. p_i is the power of VC_i dissipated at shift frequency f_{s_i}, which is computed by $p_i = \frac{Pow_i \times f_{s_i}}{F_{max}}$. t_i is the minimum test time obtained at width of w_i and shift frequency of f_{s_i}, which is calculated by $t_i = \frac{L^i_{max}(w_i)}{f_{s_i}} \times P$, P is the number of test patterns of core C.

The power-constrained multi-frequency wrapper design problem (namely, *PMWD*) can be stated as follows. Given an IP core model C, core bandwidth limitation BW_{ext}, and maximum power allowance P_{ave}, select a wrapper design for each virtual core VC_i and determine the corresponding shift frequency f_{s_i} such that we minimize the test time for core C while matching the core external bandwidth and satisfying the power constraint at any time.

We define a 3-D bin with the height of test time T_c for core C, while its length and width are constrained by the external bandwidth BW_{ext} and the maximum average power allowance P_{ave} of C respectively. We also define $S = \{S1, S2, ..., S_N\}$ (N is the number of shelves) as a set of shelves that divide the 3-D bin into N sub-bins. The height of S_j, $H(S_j)$, is defined as the maximum height (or time) among all cubes (or virtual cores) fitted into shelf S_i, i.e., $H(S_j) = max\{t_i\},\ VC_i \in S_j$. The virtual cores allocated in the same shelf will perform shift operation simultaneously, i.e., the cubes can overlap in time dimension. While the cubes within the same shelf cannot overlap to each other along the other two dimensions. It is because two virtual cores cannot share the bandwidth and power if they are tested in parallel. Thus, it is a restricted 3-D bin packing problem. The power dissipation for S_j, given by $P(S_j) = \Sigma_{i=1}^{M} p_i$, ($M$ is the number of $VC_i \in S_j$) should satisfy the power constraints, i.e., $P(S_j) \leq P_{ave}$. The bandwidth for S_j, given by $BW(S_j) = \Sigma_{i=1}^{M} BW_i$, should match the external bandwidth, i.e., $BW(S_j) \leq BW_{ext}$. Since the virtual cores within distinct shelves will perform shifting in sequential, the height of the bin, i.e., $\Sigma_{j=1}^{N} H(S_j)$ is the test time for core C.

Thus, the *PMWD* problem is deduced into a shelf-packing based 3-D bin design problem where we will minimize the height of the bin bounded by the restricted width and length.

V. PROPOSED SHELF PACKING ALGORITHM: *MWDSP*

There are three major steps of the proposed *MWDSP* heuristic, namely cube ordering, shelf division and cube packing, and cube merging and shelf elimination. In this section, we give an intuitive description of the steps and illustrate the approaches with the pseudo-code. We start with an initialization step to obtain all candidate rectangle set $R_i(w_i, L^i_{max})$ for each virtual core VC_i by running single frequency wrapper configuration (*SFWC*). The different combinations of virtual TAM width and maximum wrapper scan chain length, $pp_i[k] | k \in (1...num\_pp_i) = \{w_i, L^i_{max}\}$, are obtained to provide the flexibility to make the trade-off between scan time and test power, thus results in the best selection possible in terms of the configuration of the three dimensions of a cube. An important observation from the rectangle set is that doubling the width, the corresponding rectangle area ($A_i = w_i \times L^i_{max}$) increases or remains the same. This observation has been confirmed with all scan-testable cores in ITC SoC benchmarks [25]. This further leads to the fact that *by halving the shift frequency, the shift time may increase or remain the same when matching the bandwidth*. This feature will be employed to enhance the scheduling efficiency.

Initialization
i1: for each $VC_i, i \in (1...N_c)$
i2: run SFWC(VC_i);
i3: record $pp_i[k] := \{w_k, L^k_{max}\}$;

Step 1: Cube Ordering

The first step is to provide an ordered cube list to initiate shelf division. We try to find the maximum possible shift frequency f_{s_i} for VC_i at which the shift operation is performed without exceeding power constraint of core C, i.e., $p_i(f_{s_i}) = \frac{f_{s_i} \times Pow_i}{F_{max}} \leq P_{ave}$. Then we find the corresponding wrapper design which satisfies the external bandwidth limit while resulting in the minimum test time t_i at the maximum possible f_{s_i}. Furthermore, with the idea that reducing the shift frequency further (f_{s_i}/=2) does not lead to an increase in test time, we take the wrapper design at a lower shift frequency that maintains the same shifting time. In this way, we can save the power consumption significantly by halving the shift frequency, and even save the consumed bandwidth BW_i (or remain the same). Thus, we can handle the restrict power constraint more efficiently by distributing the freed-up power to other cubes so as to pack more cubes into the same shelf.

Cube Ordering

01: for i:=1 to N_c
02: find maximum possible f_{s_i} such that $\frac{f_{s_i} \times Pow_i}{F_{max}} \leq P_{ave}$;
03: $w_i := \frac{W_{tam} \times f_t}{f_{s_i}}$;
04: find closest Pareto point with $pp_i[j].w \leq w_i$;
05: $w_i := pp_i[j].w$;
06: $t_i := \frac{pp_i[j].L_{max}}{f_{s_i}}$;
07: find best wrapper design by f_{s_i}/=2 while t_i remains the same;
08: sort VC_i in descending order of t_i;
09: $L_{VC} = \{VC_i | i \in 1...N_c\}$;

For the given example, we build a list of virtual cores of core $hCADT01$ in the descending order of their t_{i_min}, i.e., $L_{vc} = \{VC_5, VC_1, VC_4, VC_3, VC_2, VC_6, VC_7, \}$, under the constraints of $W_{ext} = 9$ and $P_{ave} = 4500$. We assume that f_t=100*MHz* and a set of trial frequencies $F = \{100, 50, 25, 12.5\}$ *MHz*.

Step 2: Shelf Division & Cube Packing

Based on the initial ordering of cubes, we will divide the 3-D bin into several shelves or sub-bins. In this step, we attempt to pack into each shelf as many cubes as possible while satisfying power and bandwidth constraints. The shift operation is performed in parallel for the virtual cores in the same shelf while shifting in sequence if allocated within different shelf.

Shelf Division & Cube Packing

11: $Unsched := \{L_{VC}\}$;
12: N:=0;
13: while($|Unsched| \neq 0$)
14: N++;
15: $S_N := \{\emptyset\}$;
16: find VC_{max} with $t_{VC_{max}} := max_{i \in Unsched}\{t_i\}$;
17: $S_N := S_N \bigcup \{VC_{max}\}$;
18: $Unsched := Unsched \backslash \{VC_{max}\}$;
19: $H(S_N) := t_{VC_{max}}$;
20: compute $idle\_Pow$ & $idle\_BW$;
21: $VCskip := \{\emptyset\}$;
22: while(($|VCskip| \neq |Unsched|$) &&($idle\_BW \neq 0$)&&($idle\_P_{ave} \neq 0$))
23: find VC_{temp} with $t_{VC_{temp}} := max_{i \in Unsched}\{t_i\}$;
24: if ($p_{VC_{temp}} > idle\_Pow$)
25: $VCskip := VCskip \bigcup \{VC_{temp}\}$;
26: else
27: find maximum possible $f_{s_{VC_{temp}}}$;
28: if ($t_{VC_{temp}} > H(S_N)$)
29: $VCskip := VCskip \bigcup \{VC_{temp}\}$;
30: else
31: find best freq. and wrapper design such that $t_{VC_{temp}}$ closest to $H(S_N)$;
32: $S_N := S_N \bigcup \{VC_{temp}\}$;
33: $Unsched := Unsched \backslash \{VC_{temp}\}$;
34: update $idle\_Pow$ & $idle\_BW$;

We follow the order in L_{VC}, and start scheduling from the highest cube (i.e., largest in t_i), as their height will dominate the total height of the bin (or the test time for a core). We hope that the resources can be released earlier and more efficiently utilized by smaller virtual cores by scheduling the critical ones earlier. We then divide the bin into shelves by the largest cube in each shelf, i.e., $H(S_j) = max\{t_i\}$ (where VC_i is contained in S_j). After allocating the highest VC_{max} in an empty shelf S_N ($S_N = S_N \bigcup \{VC_{max}\}$ and $H(S_N) = t_{VC_{max}}$), we try to pack the next highest possible unscheduled cube VC_{temp} in the same shelf to use up the remaining bandwidth $idle\_BW$ and power $idle\_Pow$. If no suitable cube can be fitted into the idle space or the idle bandwidth/power is used up, a new shelf will be created.

The fast process of checking if a cube can be contained in the shelf is developed by employing the idea that halving the shift frequency may result in an increase in shifting time. Thereby, for the given $idle\_BW$, we first check if an incoming cube VC_{temp} can satisfy the power constraint at the minimum frequency ($f_{s_{VC_{temp}}} = F_{min}$), i.e., it cannot be allocated in the shelf if $p_{VC_{temp}}(f_{s_{VC_{temp}}}) > idle\_Pow$. If it satisfies power limit, we further check if VC_{temp} can meet the height allowance at the maximum frequency ($f_{s_{VC_{temp}}} = f_{s^{max}_{VC_{temp}}}$), i.e., it cannot be contained if $t_{VC_{temp}} > H(S_N)$, where $t_{VC_{temp}}$ is the time for VC_{temp} at $f_{s^{max}_{VC_{temp}}}$. If it satisfies, VC_{temp} will be allocated. Since the cube meets the bandwidth and power constraints, we will pick the best frequency within the range (F_{min}, $f_{s^{max}_{VC_{temp}}}$) such that its height is closest to $H(S_N)$ while $t_{VC_{temp}} \leq H(S_N)$. The reason is simply to free up more bandwidth and power so as to contain more smaller virtual cores next.

After step 2, we obtain the initial schedule of the virtual cores of core $hCADT01$ as shown in Figure 3 and the total test time is T_C=12.71.

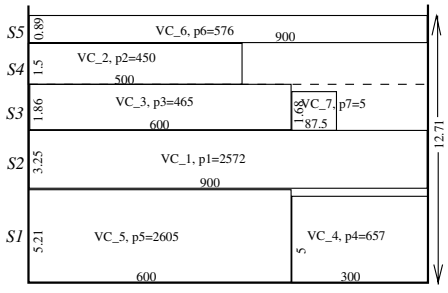


Fig. 3. The illustration of step 2 applied to $hCADT01$.

Step 3: Cube Merging & Shelf Elimination

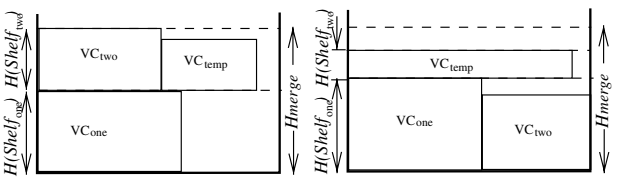
We have divided the bin into N shelves and sorted their heights in descending order after step 2. Each shelf contains one or more cubes in decreasing order of height. We further reduce the height of the bin in step 3 by merging the cubes in shelf S_j into S_i such that $H(S_i\_new) + H(S_i\_new) <$

$H(S_i) + H(S_j)$. If all cubes in S_j can be merged into S_i, S_j is eliminated.

We use the following example to illustrate the basic idea. Assuming a merging candidate $shelf_{two}$ contains two cubes, VC_{two} and VC_{temp} as shown in Figure 4(a). We try to merge the cubes into $shelf_{one}$ to further reduce the hight of bin. We first check if the power constraint can be satisfied when adding VC_{two} into $shelf_{one}$, i.e., $p_{VC_{two}} \leq rem\_Pow$. Again, the maximum possible frequency is obtained in order to achieve a minimum testing time under bandwidth constraint. If meeting the power constraint, we try to free up the width of VC_{one} while keeping its height $t_{VC_{one}} < H_{merge}$. In this way, more bandwidth could be distributed to VC_{two} with its height $t_{VC_{two}} < H_{merge}$. In the meantime, the freed-up bandwidth and power could be efficiently utilized by VC_{temp} in $shelf_{two}$ to further reduce the height of $shelf_{two}$. VC_{two} can be merged into $shelf_{one}$ only if the sum of the height of the two new shelves reduces, i.e., $H_{new}(Shelf_{one}) + H_{new}(Shelf_{two}) < H_{merge}$ as shown in Figure 4(b).

Cube Merging & Shelf Elimination

41: $Unmerged := \{S_i | i \in 2...N\}$;
42: $S_{mark} := S_{mark} \bigcup S_1$;
43: while($|S_{mark}| \neq |Unmerged| + 1$)
44: find $Shelf_{one}$ with $H(Shelf_{one}) := max_{i \in Unmerged}\{H(S_i)\}$;
45: find VC_{one} with $t_{VC_{one}} := max_{i \in Shelf_{one}}\{t_i\}$;
46: $Shelfskip := S_{mark} \bigcup \{Shelf_{one}\}$;
47: while($|Shelfskip| \neq |Unmerged| + 1$)
48: find $Shelf_{two}$ with $H(Shelf_{two}) := max_{i \in Unmerged \setminus Shelfskip}\{H(S_i)\}$;
49: find VC_{two} with $t_{VC_{two}} := max_{i \in Shelf_{two}}\{t_i\}$;
50: $H_{merge} := H(Shelf_{one}) + H(Shelftwo)$;
51: compute idle power rem_Pow in $Shelf_{one}$;
52: if ($p_{VC_{two}} > rem\_Pow$)
53: $Shelfskip := Shelfskip \bigcup \{Shelf_{two}\}$;
54: else
55: VC_{one}release rem_BW such that $t_{VC_{one}} < H_{merge}$ & VC_{two} fit into $Shelf_{one}$ with $t_{VC_{two}} < H_{merge}$;
56: findVC_{temp} with $t_{VC_{temp}} := max_{i \in Shelf_{two} \setminus VC_{two}}\{t_i\}$;
57: redistribute released try_BW and try_Pow to VC_{temp};
58: if ($H_{new}(Shelf_{one}) + H_{new}(Shelf_{two}) < H_{merge}$)
59: record VC_{one}, VC_{two}, and VC_{temp} wrapper design;
60: $Shelf_{one} := Shelf_{one} \bigcup \{VC_{two}\}$;
61: $Shelf_{two} := Shelf_{two} \setminus \{VC_{two}\}$;
62: if ($Shelftwo == \emptyset$)
63: $Unmerged := Unmerged \setminus \{Shelf_{two}\}$;
64: else
65: $Shelfskip := Shelfskip \bigcup \{Shelf_{two}\}$;
66: if $Shelf_{one}$ cannot be merged with any shelf
67: $S_{mark} := S_{mark} \bigcup \{Shelf_{one}\}$;



(a) Before merging (b) VC_{two} is merged into Shelfone

Fig. 4. An example on merging a cube into a shelf $hCADT01$.

The final packing of cubes is illustrated in Figure 5. As we can see, the total height of bin is reduced from 12.71 to 11.83 by applying step 3. With the schedule result, we obtain the minimum shifting time for core $hCADT01$ as well as the multiple frequency assignment among the virtual cores. The corresponding wrapper architecture design is thus determined in Figure 6.

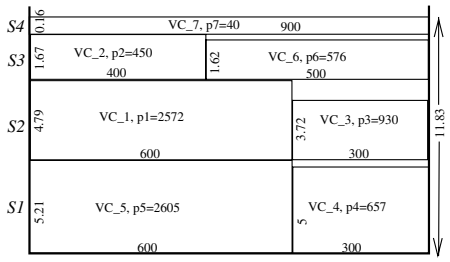


Fig. 5. The illustration of step 3 applied to $hCADT01$.

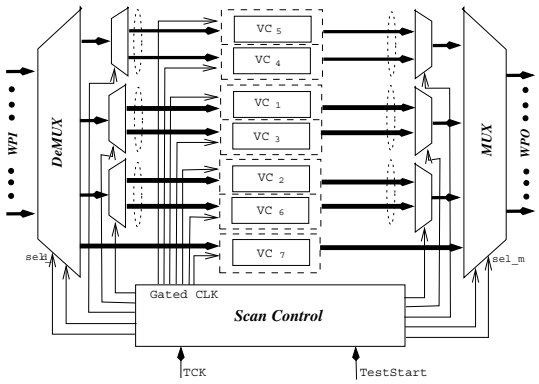


Fig. 6. Multi-frequency wrapper architecture for core $hCADT01$.

VI. Simulation Study

We evaluate the proposed multi-frequency wrapper design with shelf packing algorithm *MWDSP* via simulation and compare the test time with the best published approach. Since none of the public domain SoC benchmarks provides multiple frequency information of the embedded cores, we will use the same hypothetical but nontrivial multi-frequency IP core, $hCADT01$ (given in Table I), for simulation and comparison. Assuming that the ATE will shift test data at $f_t = 100HMz$ which is synchronized to a division of the maximum functional frequency of core $hCADT01$. Given a set of trial frequencies $F = \{100, 50, 25, 12.5\}MHz$ (use the same set for fair comparison to [21]).

Table II shows the test application time of $hCADT01$, T_{new} (in μsec) using *MWDSP* algorithm with the consideration of a wide range of external width W_{tam} and various power constraints P_{ave}. The test time reported in [21] is also listed for comparison. The percentage improvement of T_{new} over T [21] is calculated as $\delta T = \frac{T[10] - T_{new}}{T[10]}$. As observed from Table II, the *MWDSP* approach outperforms the existing best approach. The reduction in the overall shifting time can reach as high as 18.9%.

The advantage of our proposed multi-frequency serial-parallel shifting architecture (*MFSP*) is that we try to relax the power constraint by serializing shifting. If only some of the virtual cores will shift in parallel at a particular time slot, these virtual cores can shift at a higher frequency while satisfying power constraint. In the meantime, overall bandwidth could be distributed only among these cubes to achieve a minimum shifting time. From the comparison to [21], we observe that when the power constraint is quite tight, our approach achieves more improvement by serializing the shift operation. Even when there is no power constraint (i.e., $p_{ave} = \infty$), our approach can reduce the shift time further by parallelizing the shifting. The shelf packing based optimization technique provides the flexi-

TABLE II
COMPARISON RESULTS FOR CORE $hCADT01$

W_{tam}	$P_{ave} = 1500$			$P_{ave} = 3000$			$P_{ave} = 4500$			$P_{ave} = \infty$		
	T [21]	T_{new}	$\delta T(\%)$	T [21]	T_{new}	$\delta T(\%)$	T [21]	T_{new}	$\delta T(\%)$	T [21]	T_{new}	$\delta T(\%)$
16	20.84	16.90	18.90	10.42	9.13	12.38	7.44	6.94	6.72	7.44	7.53	-1.21
15	20.84	16.90	18.90	10.42	10.42	0	8.76	8.33	4.90	7.49	8.33	-11.21
14	20.84	16.90	18.90	10.42	10.42	0	8.88	8.79	1.01	8.88	8.79	1.01
13	20.84	16.90	18.90	10.42	10.53	-1.05	10.42	8.93	14.29	9.59	8.79	8.34
12	20.84	16.90	18.90	10.42	10.53	-1.05	10.42	9.75	6.42	10.42	9.15	12.18
11	20.84	16.97	18.57	11.62	11.12	4.30	10.42	9.75	6.42	10.42	9.75	6.42
10	20.84	17.07	18.09	12.08	11.24	6.95	11.62	10.58	8.95	11.62	10.58	8.95
9	20.84	17.46	16.20	13.00	12.56	3.38	12.78	11.83	7.43	12.78	11.83	7.43
8	20.84	17.57	15.69	14.48	13.50	6.76	14.88	13.50	9.27	14.88	13.50	9.27
7	20.84	19.29	7.40	17.76	16.57	6.70	15.63	15.43	1.27	15.63	15.43	1.27
6	20.84	19.60	5.95	20.84	17.19	17.51	19.20	17.19	10.46	19.18	17.19	10.38
5	25.04	23.38	6.60	23.24	21.81	6.15	23.24	21.81	6.15	23.24	21.81	6.15
4	29.76	26.16	12.09	29.76	24.84	16.53	29.01	24.84	14.37	29.01	24.84	14.37
3	41.68	34.06	18.28	38.36	34.06	11.21	38.36	34.06	11.21	38.36	34.06	11.21
2	59.88	51.61	13.80	58.02	50.36	13.2	58.02	50.36	13.2	58.02	50.36	13.20
1	116.04	100.70	13.22	116.04	98.44	15.16	116.04	98.44	15.16	116.04	98.44	15.16

bility to configure the wrapper into a serial architecture when the power and bandwidth constraints are tight, while transform to a parallel architecture when the constraints become loose. As the test time and test power of a scan-testable core are interdependent (e.g., the scan time is inversely proportional to the scan frequency, while the test power increases when increasing the scan frequency), we can achieve the best trade-off in a way that power-critical virtual cores will be serialized in shifting while others are accommodated in parallel with them to reduce the overall test time while meeting the bandwidth limit.

VII. CONCLUSION

We have presented in this paper a novel power-aware serial-parallel wrapper architecture for multi-frequency IP modules. The tight power budget is handled by gating off certain virtual cores at a time to gain the best trade-off between test power and scan time. We have formulated the power-constrained multi-frequency wrapper optimization problem into a 3-D Bin Packing problem and proposed an efficient shelf-packing heuristic algorithm to optimize the wrapper scan architecture and minimize the test time of a core. Through performance evaluation, our approach outperforms those previously published. The improvement can reach as high as 18.9%. The algorithm requires a negligible amount of computation time ($6ms$) and therefore is suitable for more complex cores. This is especially an improvement over the CPU-intensive ILP-based method in [21].

REFERENCES

[1] T. Waayers, E. J. Marinissen, and M. Lousberg, "IEEE std 1500 compliant infrastructure for modular SOC testing," in *Proc. of ATS*, p. 450, November 2005.

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

[3] E. J. Marinissen, S. K. Goel, and M. Lousberg, "Wrapper design for embedded core test," in *Proc. of ITC*, pp. 911–920, october 2000.

[4] S. Koranne, "A novel reconfi gurable wrapper for testing of embedded core-based SoCs and its associated scheduling algorithm," *Journal of Electronic Testing: Theory and Applications*, vol. 18, pp. 415–434, August 2002.

[5] 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.

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

[7] S. K. Goel, "An improved wrapper architecture for parallel testing of hierarchical cores," in *Proc. of ETS*, pp. 147–152, May 2004.

[8] B. Vermeulen, S. Oostdijk, and F. Bouwman, "Test and debug strategy of the PNX8525 NexperiaTM digital video platform system chip," in *Proc. of ITC*, pp. 121–130, October 2001.

[9] M. Amodeo and B. Cory, "Beyond at-speed," in *Test and Measurement World*, pp. 43–48, November 2005.

[10] N. Tendolkar and et.al, "Scan-based at-speed testing for the fastest chips," 2001. Mentor Graphics White Paper.

[11] P. Varma and S. Bhatia, "A structured test reuse methodology for core-based system chips," in *Proc. of ITC*, pp. 294–302, October 1998.

[12] A. Hales and E. Marinissen, "IEEE P1500 web site." http://grouper.ieee.org/groups/1500.

[13] J. Aerts and E. J. Marinissen, "Scan chain design for test time reduction in core-based ICs," in *Proc. of ITC*, pp. 448–457, 1998.

[14] E. J. Marinissen, R. Arendsen, G. Bos, H. Dingemanse, M. Lousberg, and C. Wouters, "A structured and scalable mechanism for test access to embedded reusable cores," in *Proc. of ITC*, pp. 284–293, 1998.

[15] G. Hetherington and et.al, "Logic BIST for large industrial designs: Real issues and case studies," in *Proc. of ITC*, pp. 358–367, 1999.

[16] L.-T. Wang, X. Wen, P.-C. Hsu, S. Wu, and J. Guo, "At-speed logic bist architecture for multi-clock designs," in *Proc. of ICCD*, pp. 475–478, 2005.

[17] A. Khoche, "Test resource partitioning for scan architecture using bandwidth matching," in *Digest of Int'l Workshop on Test Resource Partitioning*, pp. 1.4.1–1.4.8, 2001.

[18] A. Sehgal and et.al, "Test cost reduction for SoCs using virtual tams and lagrange multipliers," in *Proc. of DAC*, pp. 738–743, June 2003.

[19] Q. Xu and N. Nicolici, "Multi-frequency test access mechanism design for modular SOC testing," in *Proc. of ATS*, pp. 2–7, November 2004.

[20] T. Yoneda, K. Masuda, and H. Fujiwara, "Power-constrained test scheduling for multi-clock domain socs," in *proc. of DATE*, 2005.

[21] Q. Xu, N. Nicolici, and K. Chakrabarty, "Multi-frequency wrapper design and optimization for embedded cores under average power constraints," in *proc. of DAC*, June 2005.

[22] L. Whetsel, "Adapting scan architectures for low power operation," in *Proc. of ITC*, pp. 863–872, October 2000.

[23] J. Sexena, K. Butler, and L. Whetsel, "An analysis of power reduction techniques in scan testing," in *Proc. of ITC*, pp. 670–677, October 2001.

[24] M. Beck, O. Barondeau, M. Kaibel, F. Poehl, X. Lin, and R. Press, "Logic design for on-chip test clock generation - implementation details and impact on delay test quality," in *Proc. of DATE*, pp. 56–61, 2005.

[25] E. J. Marinissen, V. Iyengar, and K. Chakrabarty, "ITC'02 SOC test benchmarks." http://www.hitech-projects.com/itc02socbenchm/.