Wrapper Design for Multifrequency IP Cores

Qiang Xu, *Student Member, IEEE,* and Nicola Nicolici, *Member, IEEE*

***Abstract*—This paper addresses the testability problems raised by intellectual property cores with multiple clock domains. The proposed solution is based on a novel core wrapper architecture and a new wrapper design algorithm. It is shown how multifrequency at-speed test response capture can be achieved via the design of capture windows without any structural modifications to the logic within the embedded core. The new features in the core wrapper architecture, which introduce limited hardware overhead, can also synchronize the external tester channels with the core's internal scan chains in the shift mode. Thus, the wrapper implementation space can be explored in order to efficiently utilize the available tester bandwidth while meeting the constraints on the maximum internal shift frequency that guarantees low testing time within the given power ratings. Using experimental data, the benefits of the proposed solution are demonstrated by analyzing the tradeoffs between the number of tester channels, testing time, area overhead, and power dissipation.**

***Index Terms*—Core, multifrequency, system-on-a-chip (SOC), wrapper.**

I. Introduction

SYSTEM-ON-A-CHIP (SOC) design using reusable intellectual property (IP) blocks is an evolving implementation paradigm that has triggered novel business models, based on core providers and system integrators. While many new design methodologies have been proposed to tackle the SOC complexity, an emerging problem is whether the test technology can keep the pace with the design chain process, transistor-per-pin ratio and elevated operating frequencies. In addition, excessive test power may lead to either destructive or erroneous test [18]. Therefore, test planning and development is becoming an essential implementation step [26].

Although core-based SOC testing is a fast developing research area [3], there are several open problems which require further investigation. For example, it is common that SOC designs in telecommunications, networking and digital signal processing applications employ IP cores operating at different clock rates. In addition, many embedded cores operate internally using multiple frequencies. For instance, for the design reported in [25] all the cores have more than three clock domains. To illustrate a multiple-frequency core-based SOC, Fig. 1 shows a simple hypothetical design that comprises three cores with three different physical clocks. In addition, Core 2 consists of modules (M1,M2,M3) operating at different frequencies (f1,f2,f3). A *physical clock* is a chip-level clock, e.g., it can come from an external oscillator or an on-chip phase-locked loop (PLL). All the internal clocks generated from the same physical clock are considered to be a part of the same physical clock domain. The multifrequency modules communicate with each other through asynchronous handshake signals, synchronization logic or first in first out (FIFO) memory blocks [4]. Although multifrequency embedded cores present advantages, such as reduced power and silicon area, because of the clock skew and synchronization problems they require special attention during test.

The objective of this paper is to provide a solution for *at-speed stuck-at testing of multifrequency IP cores when test data is transferred using a low speed external testers.* The key to the proposed solution is a novel core wrapper architecture used 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. To the best of authors' knowledge this problem has not been discussed in public literature, and consequently the relevant related work is overviewed in the next section.

A. Related Work

Embedded cores in a SOC are not directly accessible through the primary inputs, and consequently dedicated test access mechanisms (TAMs) are required to facilitate SOC testing. These TAMs are connected to the embedded cores using special interfaces called *core wrappers* [26]. Recently a number of approaches have addressed core wrapper design (e.g., [6], [10], [12], [15], [16], [23], [24] only to name a few relevant ones). The work in [24] proposes a "test collar" as a test wrapper for SOC test. The method is based on two on-chip variable-width busses, one for test data and one for test control. Marinissen *et al.* [15], proposed a TestShell wrapper which forms the basis for the IEEE P1500 [1] core wrapper. The TestShell is scalable and supports the operating modes required by the IEEE P1500. Partial isolation rings [23] take into account critical timing paths and reduce area overhead of the wrapper. Wrapper design optimization to reduce testing time was first addressed in [16]. [10] introduced an algorithm (based on a best fit decreasing heuristic) to minimize, at the same time, the core testing time and the required TAM width. A similar algorithm was also described in [20]. In [12] a reconfigurable core wrapper which allows for a dynamic change in TAM width was proposed. Later a power-conscious reconfigurable wrapper design was presented in [13]. In [6] wrapper design algorithms take into account and tackle the volume of useless test data.

Since the existing core wrapper design approaches are applicable to single-frequency embedded core test, in the following an overview of the proposed approaches for multifrequency

Manuscript received December 11, 2003; revised December 11, 2004. This work was supported by Micronet and Gennum Corporation. This paper was presented in part at the IEEE/ACM Design, Automation, and Test in Europe Conference (DATE) 2004, Paris, France.

The authors are with the Computer-Aided Design and Test Group, Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON L8S 4K1, Canada (e-mail: xuqiang@grads.ece.mcmaster.ca; nicola@ece.mcmaster.ca).

Digital Object Identifier 10.1109/TVLSI.2005.848811

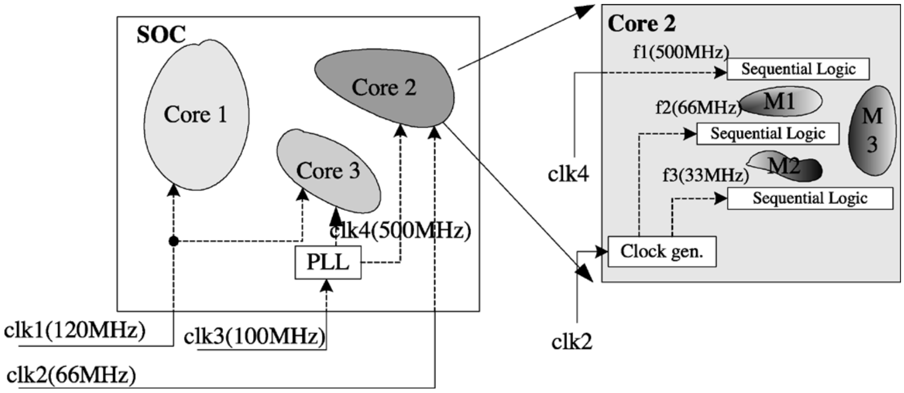

Fig. 1. Multifrequency SOC example.

testing is given. The main solutions are based on built-in self-test (BIST) [2], [5], [8], [17]. The solution presented in [5] used a dedicated clock generator to shift the multifrequency scan chains at their corresponding clock frequencies. To avoid clock skew during capture, retiming latches or dedicated two-phase/two-edge clocking schemes were used between different clock domains. In [17] the test data was shifted in/out at-speed by reusing the existing clock tree on chip and a programmable *scan mode* signal unit was used to control the capture of the circuit responses. The solutions proposed in [2], [8] employ a rather different approach that separates the clocking for scan and capture in two phases, by multiplexing the clock signals for each phase. The main difference between these two approaches lies in the design of the *capture window*. In [2], in the capture mode, all the rated clocks are applied iteratively in a number of intervals equal to the number of clock domains. In each iteration the selected clock signal will propagate to a scan chain only if its value is lower than the rated clock of the respective scan chain (see [2] for detailed operation). In [8] a more flexible *capture window* was used, which consists of captures in different clock domains and some shift operations to create inter-domain at-speed capture. The functional clock of each of the domains is used to obtain a shift followed by a capture. In addition, the shift operation for all the scan chains can work at any of the on-chip frequencies. Four types of elements, composed of a scan flip-flop and an extra latch or flip-flop, are introduced in between transition-hazard clock domains[1] in [22]. The extra latch or flip-flop is controlled by an external test signal and hence a two-phase clocking scheme is applied in the test mode. The technique can be effectively applied in low frequency scan test. However, since the relation between the two clocks is dependent on the maximum clock skew and the longest propagation delay, it is difficult to ensure a nonoverlapping clocking scheme for at-speed test.

B. Motivation and Objectives

Despite the fact that BIST is the primary solution for at-speed multifrequency testing [2], [5], [8], [17] there are a number of issues which arise in the SOC paradigm from the core provider and system integrator inter-operability perspective. There are four main cases: the system integrator will receive a (*i*) BIST-ed core, a (*ii*) BIST-ready core, a (*iii*) scan-testable core, or a (iv) functional-testable core. With the exception of the BIST-ed multifrequency cores, in order to deliver the patterns (using *low-speed testers*) to the IP-protected cores and to perform rapid at-speed test (using *high-speed on-chip generated clocks*) without exceeding the power ratings or maximum shift frequency, the system integrator is constrained to design a multifrequency core wrapper, which is the very aim of this paper. Unlike the existing approaches which assume that designs/cores can be BIST-ed for multifrequency test, i.e., the structural netlist can be modified with extra hardware to guarantee valid multifrequency capture, our approach is suitable for IP-protected cores where the SOC integrator is in charge of developing the multifrequency test strategy.

The system integrator can use test wrapper design to tackle the multifrequency test problem, however the existing single-frequency wrapper designs (e.g., [6], [10], [12], [13], [15], [16], [23], [24]) are not directly adaptable to at-speed multifrequency testing (if memory elements from different clock domains, such as core's internal latches/flip-flops and wrapper boundary registers, are captured at the same time, then clock skew may occur and corrupt test data). While in the functional mode, signals crossing different clock domains are made skew-tolerant through special handshake hardware (e.g, brute-force synchronizers or FIFOs) or protocols, avoiding clock skew during test is a major challenge. Although by grouping the flip-flops triggered by the same clock together and adding lock-up latches in between different clock domains, clock skew problem during shift can be solved (for mux-based scan approach), clock skew during at-speed capture still might occur and corrupt the test response. Therefore, careful attention must be paid to the design of the capture window and a special support must be provided within the core wrapper architecture in order to guarantee that IP protection of embedded cores is not violated. In addition to the capture window design, the frequency at which the data is fed to the core is of great importance, since it determines the tradeoff between test application time and power dissipation. On the one hand, if no attention is paid to the multifrequency core wrapper design and the test data is fed into scan chains at high shift frequency then the outcome of the testing process may lead to the permanent damage of the SOC due to the excessive test power. On the other hand, if the test data is fed into the scan chains at low

[1] If there are no data transfers between two clock domains or data transfers between two clock domains are safe during capture, then they are called *transition-free*. Otherwise, they are called *transition-hazard* clock domains.

shift frequency then large test application time may be required, thus leading to an increased cost of test execution. New features for scaling the shift frequency should be provided *within the core wrapper architecture* in order to efficiently utilize the available tester bandwidth while meeting the constraints on the maximum internal shift frequency that guarantees low testing time within the given power ratings.

To solve the above emerging problems for the non-BISTed IP-protected embedded cores (i.e., determining a reliable and cost-efficient shift frequency and designing at-speed multifrequency capture windows without any structural modifications to the core) we propose a novel core wrapper architecture and its associated new multifrequency wrapper design algorithm. The proposed architecture, using limited hardware overhead, can tradeoff the number of tester channels, scan time and test power and, at the same time, it implements a capture window for at-speed stuck-at fault testing that can efficiently work with cores for which automatic test pattern generation (ATPG) techniques described in [11], [14] have been applied. It is also important to note that, since clock skew problem during capture are avoided by making transition-hazard clock domains capture at different time, the proposed approach is not affected by different synchronizer designs [4].

II. Multifrequency Core Wrapper

This section presents the multifrequency core wrapper architecture and design algorithm, by focusing on the case when all the internal clocks are generated from a single physical clock. As summarized at the end of this section, the proposed solution can easily be extended to the general case when a core has multiple physical clocks. In order to describe the proposed architecture and algorithm, the single-frequency core wrapper (**SFCW**) design problem is extended to the multifrequency core wrapper (**MFCW**) design problem, which is formulated as follows:

MFCW problem: Given a core with its test set parameters, i.e., the number of clock domains N_c, each clock domain g comprising N_{in}^g primary inputs, N_{out}^g primary outputs, N_{bi}^g bidirectional I/Os, and N_{sc}^g scan chains with given scan chain lengths for fixed-length scan chains (or the number of scan cells when scan chains are flexible), determine a wrapper design (including the architecture of the wrapper, the shift frequency and the capture window) with minimum testing time for the given TAM width constraint.

To perform at-speed multifrequency testing, although it is not necessary to load/unload data at functional frequencies, the last launch and capture must be done at-speed [8]. To facilitate this, the physical test clock needs to function at the highest frequency f_h during launch/capture. Since most of the available testers cannot provide test data at the highest on-chip frequency, we need to provide a mechanism which can use low-speed testers to transfer test data (shift phase), yet perform test application using high-speed on-chip functional multiple frequencies (capture phase). To achieve this, the tester must synchronize with an on-chip low frequency f_l, derived from the on-chip high functional frequency f_h (e.g., coming from a PLL), which is used to shift in/out test data from/to the automatic test equipment (ATE). In the following the details of the proposed wrapper architecture and design algorithm are given.

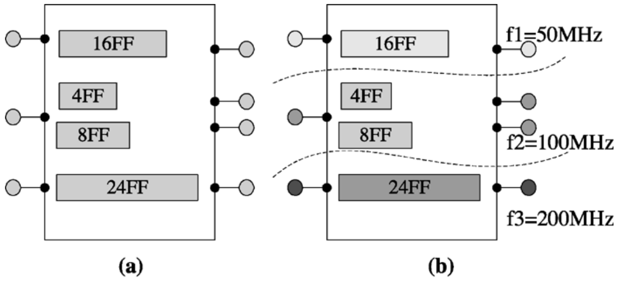

Fig. 2. (a) Single-frequency core. (b) Multifrequency core.

Wrapper Scan Chains in Multiple Frequency Groups: A multiple frequency core is shown in Fig. 2(b), where in addition to labeling the scan chains with their length, the associated normal operating frequency is also provided. Core wrapper design is mainly concerned with the construction of wrapper scan chains (*WSCs*) such that the testing time is minimized. The WSCs are composed from the input/output wrapper cells and the internal scan chains. When using single-frequency core wrappers, the testing time (in seconds) will be a function of the longest WSC [16] and the single shift frequency f, given by the following equation:

$$\tau(C) = \frac{\{(1 + \max(wsc_i, wsc_o)) \times v + \min(wsc_i, wsc_o)\}}{f} \quad (1)$$

where wsc_i/wsc_o are the lengths of the maximum input/output WSC, respectively, and v is the number of test vectors in the test set for core C. For multifrequency core wrappers, care is taken not to combine scan chains belonging to different frequency groups, and consequently the testing time of the core is not only dependent on the lengths of the WSCs but also on the shift frequency f^g for different frequency groups (note, f^g is not necessarily the same as its functional frequency). For multifrequency core test, since cores with different clock domain configuration will have different capture window design, hence different length of capture cycles, the testing time of the multifrequency core can be formulated by the following equation:

$$\tau(C) = \frac{\max\limits_{g} \{\max(wsc_i^g, wso_o^g) \times v + \min(wsc_i^g, wso_o^g)\}}{f^g} + t_c \times v \quad (2)$$

where the wsc_i^g/wsc_o^g are the lengths of the maximum input/output WSCs of clock domain g (with shift frequency f^g) from core C, t_c is the time spent on capture phase for each pattern and v is the number of test vectors. To compute the WSCs for the multifrequency core, a single-frequency algorithm can be employed and adapted for each frequency group, as shown later in this section. It should be note that for the architecture proposed in this paper, which is described in the following paragraphs, the shift frequency f^g is the same for all the clock domains and it will be referred to as f_s in the rest of the paper.

Core Wrapper Interface: A multifrequency core wrapper for the example core shown in Fig. 2(b) is shown in Fig. 3 (INTEST mode is illustrated). When compared to a P1500 single-frequency core wrapper, the proposed multifrequency core

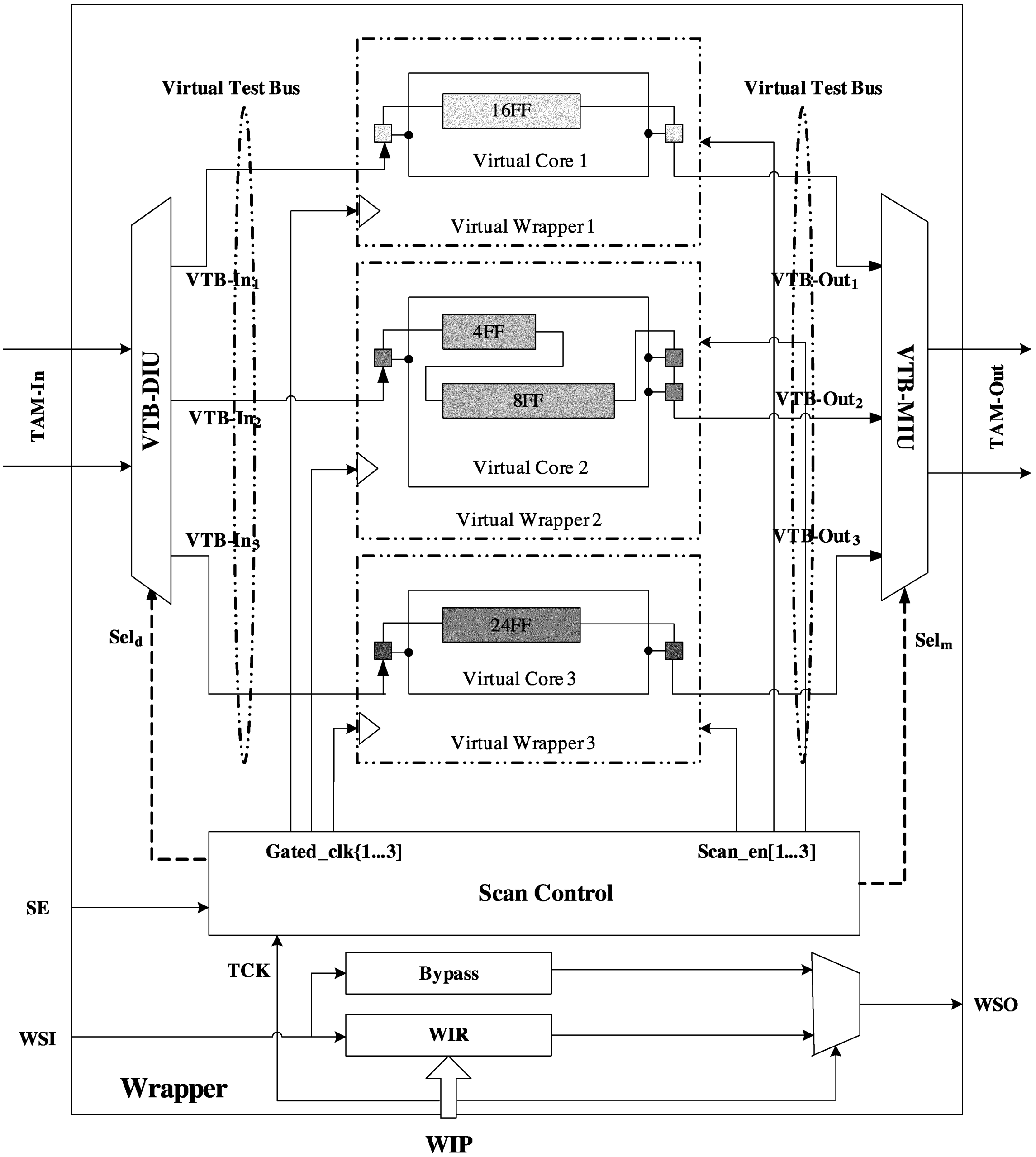

Fig. 3. Example multifrequency core wrapper.

wrapper includes the same interface signals: serial input/output (WSI/WSO), parallel input/output (TAM-In/TAM-Out), and the wrapper interface port (WIP). The WIP provides test control and test clock for the core under test. Additional off-chip test clocks (i.e., ATE supplied) or on-chip generated test clocks can also be included in the multifrequency interface.

Core Wrapper Architecture: Although the MFCW interface is similar to the SFCW one, the architecture of the proposed MFCW differs significantly from the standard SFCW. Logic blocks belonging to different frequency domains are grouped and marked in the figure as *virtual cores*, and for each virtual core (VC) a *virtual wrapper* (single-frequency core wrapper), containing the WSCs for the respective group, is assigned. The virtual wrapper is connected to the interface through virtual test bus (VTB) lines. We assume that the system integrator uses parallel TAM-In/TAM-Out lines for MFCWs case (this is optional for SFCW). The TAM-In is connected to a *virtual test bus de-multiplexing interface unit (VTB-DIU)*, which drives data in the *virtual test bus lines*. Similarly, TAM-Out is connected to a *virtual test bus multiplexing interface unit (VTB-MIU)*, which collects the data from the virtual test buses (the operation of VTB-DIU and VTB-MIU are explained later on this section).

Scan Control Block: Scan Control block is a key part of the proposed MFCW, since it is used to generate the gated clocks (*Gated_clk*) necessary for shift and capture phases and scan enable signals (*Scan_en*) required by each virtual core in the capture phase. Since for at-speed testing, it is not necessary to load/unload test data at the rated frequencies, the shift frequency is used to tradeoff testing time against power dissipation. Unlike [7], we do not speed up the test data loading/unloading to

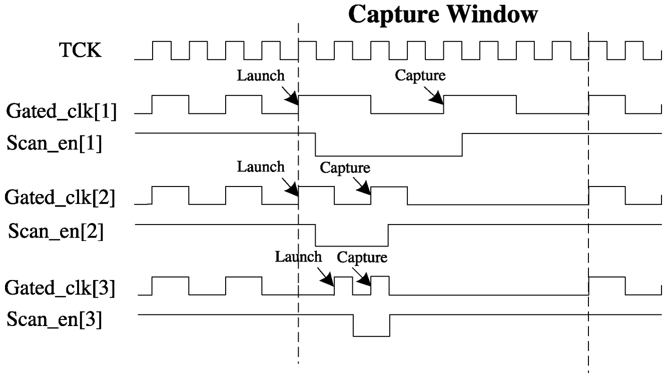

Fig. 4. At-speed multifrequency testing timing diagram with a single physical clock.

its functional frequency through serializing/de-serializing technique since this will reduce the tester channel capacity and increase test power. Rather, we load/unload test data from/to the ATE at a slower frequency f_t and distribute it to multiple scan chains at the speed of shift frequency f_s using the proposed wrapper architecture. For each virtual core, we use the same frequency f_s during shift, which is switched to the functional frequency in the capture window (see Fig. 4). However, the shift frequency f_s is not necessarily the same as the tester frequency f_t. For the example from Fig. 2(b), if we assume the maximum tester frequency is 120 MHz, we will synchronize the tester with the on-chip clock with the maximum functional frequency $(f_3 = 200 \text{ MHz})$. Therefore, the tester frequency is selected as $f_t = 100$ MHz, assuming core test power using f_t is within its power rating. The value of f_s and the number of the virtual test bus lines N_{vtb} depend not only on tester frequency f_t and the available TAM width W_{tam}, but also on the number of the clock domains N_c, such that $N_{\text{vtb}} \geq N_c$ and $N_{\text{vtb}} \times f_s = f_t \times W_{\text{tam}}$ are satisfied. To decrease the power consumption during test, the objective is to find the lowest possible shift frequency f_s without affecting testing time. To simplify the hardware implementation we select the ratio of f_t/f_s as two's exponent. For example, in Fig. 3, the tester frequency f_t is selected to be 100 MHz and if the available TAM width $W_{\text{tam}} = 2$, which is less than $N_c = 3$, then we will select f_s at 50 MHz and the total number of virtual core test bus lines will be $N_{\text{vtb}} = ((100 \times 2)/50) = 4$. Although decreasing f_s to 25 MHz, thus leading to $N_{\text{vtb}} = 8$, will reduce power consumption during test it will increase testing time, since having 8 VTB lines will be more than necessary for the 4 scan chains of the core from Fig. 3. It is also important to note that, due to the nature of launch from last shift at-speed testing strategy used in this paper, scan enable signal must be able to switch at-speed (see signal *Scan_en* [3] in Fig. 4), which requires either routing this signal as a clock tree or pipelining it to distribute the delay across several clock cycles [19]. In addition, since transition-hazard VCs are captured at different time in the capture window, the previous-captured VC will pass data to the later-captured VC and overwrite part of its shifted data. We assume in this paper that advanced ATPG tools, as described in [11], [14] are used to take care of such situations.

By grouping flip-flops from the same clock domain into separate scan chains we eliminate the problem of clock skew during shift. However, to avoid the clock skew problem during capture, we employ a capture window. In the capture window separate capture clocks are generated for each VC. The clock switching between shift clock and capture clock is made glitch-free in the Mux using techniques described in [21]. When compared to [8], to adapt capture window to the core provider/system integrator model, the proposed solution is not programmable. However, its control is *embedded in the core wrapper architecture*. In addition, based on core provider's information, the transition-free clock domains are captured at the same time (this decreases also test generation complexity), while the transition-hazard clock domains are captured at different times to avoid the inter-domain clock skew. This is achieved through carefully controlling the *Gated_clk* and *Scan_en* signals using the Capture FSM shown in Fig. 5. The generation of these two signals justifies the use of the highest frequency physical clock instead of the slow tester clock as the core wrapper TCK signal. If the shift frequency f_s is lower than the tester frequency f_t then an internal control finite state machine (VTB FSM) is used to generate the mux select signal for VTB-DIU and VTB-MIU (see Fig. 5). The testing timing diagram for the example from Fig. 2(b) is shown in Fig. 4. It should be noted that clock domains 2 and 3 are transition-free and hence they can be safely captured simultaneously, while the clock domain 1 will capture data at a different time to eliminate the test invalidation problem arising from clock skew during capture.

VTB-DIU and VTB-MIU Blocks: VTB-DIU block is used to synchronize the input test data and to transfer the test vectors into the corresponding virtual cores. If the shift frequency f_s is selected to be the same as the tester frequency f_t, then we can simply connect each TAM line to its corresponding VTB line through a flip-flop (used to register the last shift launch bit). However, if a lower shift frequency is used, then VTB FSM is needed to control the de-multiplexing unit. Again, flip-flops are used in the block to register the last shift launch bits. It is obvious that by changing the Sel_d at f_t, the test data from the TAM lines are loaded into the corresponding internal flip-flops in an interleaved mode first and then into all VTB lines at f_s with a latency of one clock cycle. It is important to note that the last shift launch bit registered in VTB-DIU block is shifted in at the correct time decided by the Scan Control block, which is an essential feature required by at-speed test through last shift launch. The multiplexing unit is the opposite of the de-multiplexing unit, i.e., it is used to synchronize the output test data and transfer the test responses to the corresponding TAM lines. Note that, both VTB-DIU and VTB-MIU are active only in the test mode and hence do not infer any additional performance penalty when compared to the standard P1500 wrapper.

Multifrequency Core Wrapper Design (MFCWD): The wrapper design algorithm (shown in Fig. 6) takes as inputs the tester frequency (f_t), the TAM width (W_{tam}), the number of clock domains N_c and the test parameters of core (C), including the number of primary inputs, primary outputs, and bidirectional I/Os, the number of scan chains and scan chain lengths for fixed-length scan chains (or the number of scan cells when scan chains are flexible), for each clock domain (virtual core), and it outputs the shift frequency (f_s) and the final wrapper design, including the virtual core wrapper VC and the wrapper scan chains SC. The algorithm initializes virtual cores (VC^g),

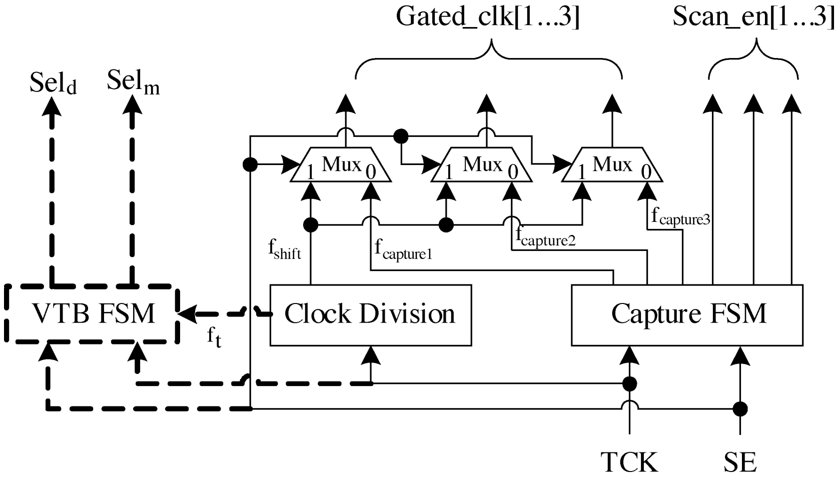

Fig. 5. Scan control block.

```
Algorithm: MFCWD
INPUT: $C$, $W_{tam}$, $f_t$, $N_c$
OUTPUT: $f_s$, $VC = \{VC^g | g = 1 \ldots N_c\}$, $SC$
1. Initialize $VC^g$;
2. Initialize $n$ to make $N_{vtb} \geq N_c$;
3. while (true) {
4.     Assign $f_s = f_t \div n$, $N_{vtb} = W_{tam} \times n$;
5.     Assign $N_{assigned\_vtb} = 0$;
6.     for i from 1 to $N_c$ {
7.         Assign $VTB_{vc^i} = 1$;
8.         $N_{assigned\_vtb}$++;
9.         do SFCWD;
.      }
10.    while (true) {
11.        find $g_{tat}^{\max} = \max\{\tau^g\}$;
12.        $VTB_g$++; $N_{assigned\_vtb}$++;
13.        do SFCWD;
14.        if($N_{assigned\_vtb} == N_{vtb}$) break;
15.        if no further reduction possible break;
.      }
16.    if ($T_{shift}$ increase) break;
17.    Assign $n = n \times 2$;
.  }
18. done
```

Fig. 6. Pseudocode for multifrequency core wrapper design.

by assigning to each virtual core the inputs, scan chains, outputs, and bidirectional I/Os which operate in its clock domain (line 1). In line 2 the ratio n (f_t/f_s) is initialized such that the number of virtual test bus lines N_{vtb} exceeds N_c. The algorithm loops through different configurations of f_s and N_{vtb} in order to reach the minimum shift time T_{shift} for a pattern. In line 5 the total number of assigned virtual test bus lines $N_{\text{assigned\_vtb}}$ is initialized as zero. Each virtual core VC_g is first allocated one virtual test bus line and then the single frequency core wrapper design (SFCWD) is performed to get an initial testing time (lines 6–9) to be used as the starting point for virtual test bus line allocation (lines 10–15). Depending on N_{vtb}, the algorithm proceeds as follows. First, all the virtual cores are sorted based on their testing time (τ^g) and the virtual core which needs the longest testing time is identified (line 11). Then the following steps will iteratively assign virtual test bus lines to virtual cores. The basic idea is to assign more virtual test bus lines to the cores with longer testing time (line 12). Whenever a virtual test bus line is assigned, SFCWD is performed again to get the new testing time (line 13). There are two exit points for the inner loop: one when all the virtual test bus lines are assigned and all the virtual cores have been connected (line 14); the other exit point is when there is no further testing time reduction possible (line 15). Note, that the SFCWD algorithm used in our implementation is based on *Design_wrapper* algorithm in [9]. When the shift time for the core with the new f_s has increased, the algorithm halts (line 16) since further growth in N_{vtb} will only increase testing time (because the frequency is lowered, while the number of clock cycles required for shifting the longest scan chain will stay the same). In addition, as long as the SFCWD algorithm returns the optimal time for each virtual core (and the capture clock cycles are ignored), the proposed MFCWD is optimal by construction.

The worst case complexity of algorithm *SFCWD* using *Design_wrapper* is shown to be $O(sc \log sc + sc \cdot W_{\text{vtb}})$ in [9], where sc is the number of scan chains in the virtual core and W_{vtb} is its VTB width. The worst case complexity of the proposed algorithm *MFCWD* is $O(\sum_i^{N_c} sc_i \log sc_i + W_{\text{tam}} \cdot sc_{\max} \log sc_{\max} + W_{\text{tam}}^2 \cdot sc_{\max})$, where sc_i and $sc_{\max}$ are the number of internal scan chains for virtual core i and the maximum number of scan chains of all virtual cores, respectively. As a result, for a core with a given number of clock domains and fixed-length scan chain designs, the computational complexity of algorithm *MFCWD* is quadratic in the number of external TAM wires.

MFCW With Multiple Physical Clocks: So far it was shown how a *new core wrapper architecture* can address at-speed multifrequency core test with only one physical clock domain. For the general problem, where the core comprises several physical clock domains, we still divide the core under test into virtual cores belonging to different clock domains and we can still use the same shift frequency for all the virtual cores. The main difference, however, lies in the design of the capture window. To reach at-speed testing without corrupting test data, we propose to separate the capture window into several separate sub-capture windows, corresponding to each physical clock domain.

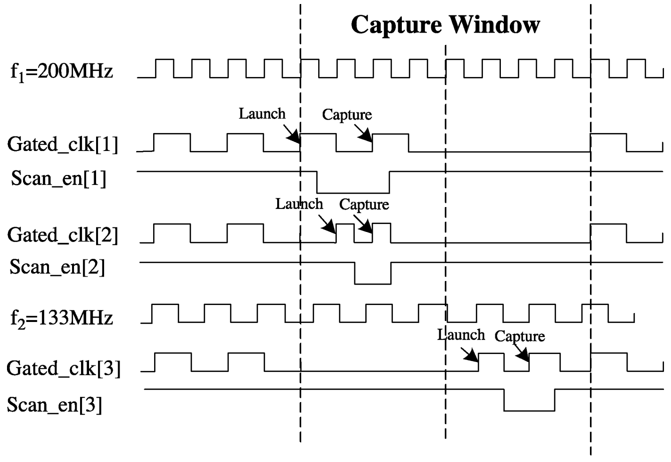

Fig. 7. At-speed multifrequency testing timing diagram with multiphysical clocks.

This is achieved by letting all the physical clocks connect to the Scan Control block and utilizing counters to control the interval between different sub-capture windows. By separating the launch/capture for transition-hazard physical clock domains in the capture window, we guarantee a risk-free at-speed test for each virtual core. For example, consider an embedded core with three virtual cores VC_1, VC_2, VC_3, that operate at 100, 200, and 133 MHz separately. Suppose $f_s = f_t = 100$ MHz (division of $f_{VC_2} = f_1 = 200$ MHz), then the timing diagram is shown in Fig. 7, which shows two separate sub-capture windows: one for clock domains 1 and 2, which are transition-free, and one for clock domain 3.

III. Experimental Results

Since no existing approaches have tackled the multifrequency embedded core testing problem, it is difficult to provide a one to one comparison to previous work. We have decided to analyze the tradeoffs of the proposed solution, in terms of the number of tester channels, testing time, area overhead and power dissipation. Therefore, in this section, we present experimental results for a hypothetical multifrequency core hCADT00. This core has four clock domains inside: the clock domain information is shown in Table I, in which f denotes the functional frequency; N_{in}, N_{out}, N_{bi} and N_{sc} are the number of inputs, outputs, bidirectionals, and scan chains in the specific clock domain, respectively; and the length of each scan chain is shown in column SC_{length}. Note, this is a hard core, i.e., the internal scan chains cannot be divided, and the wrapper scan chains contain also the I/O boundary cells.

For different TAM widths W_{tam}, the shift frequency of the core f_s, virtual test bus lines assigned to each VC ($VTB[1\ldots4]$), the necessary shift clock cycles C_{shift} and time T_{shift} for each pattern, and the additional area overhead Num_{gates} introduced by the new MFCW are shown in Table II. In this experiment, we assume that the maximum frequency of the tester is 120 MHz and, since we synchronize the tester with a division of the maximum internal frequency (200 MHz), the tester will shift test data at $f_t = 100$ MHz. From the experimental results shown in Table II we can observe that the shifting times for $W_{\text{tam}} = 24$ and $W_{\text{tam}} = 16$ are the same. This is because when the available TAM width exceeds 16 (which is the maximum number of wrapper scan chains), the shift time for virtual core VC_1 has already achieved its lowest value (100 clock cycles) and assigning more VTB lines to it will not lead to any improvements. It can also be seen that when the available TAM width is small (≤ 4 in Table II), a lower shift frequency is selected. For example, when the available TAM width is 3, f_s is selected to be 25 MHz and the total available VTB lines are $((f_t \times W_{\text{tam}})/f_s) = 12$.

TABLE I
hCADT00 Clock Domain Information

$f(MHz)$	N_{in}	N_{out}	N_{bi}	N_{sc}	SC_{length}
200	38	42	0	5	100 100 100 98 98
100	24	29	32	3	88 88 87
133	34	10	0	1	76
50	42	62	0	4	96 96 64 62

TABLE II
hCADT00 Wrapper Design With Different TAM Width

W_{tam}	f_s (MHz)	$VTB[1...4]$	C_{shift} (cc)	T_{shift} (μs)	Num_{gates}
24	100	[6 4 2 4]	100	1	333
16	100	[6 4 2 4]	100	1	333
8	100	[3 2 1 2]	198	1.98	241
4	50	[3 2 1 2]	198	3.96	246
3	25	[5 3 1 3]	127	5.08	293
2	25	[3 2 1 2]	198	7.92	248
1	12.5	[3 2 1 2]	198	15.84	271

TABLE III
hCADT00 Wrapper Design ($W_{\text{tam}} = 4$)

f_s (MHz)	$VTB[1...4]$	C_{shift} (cc)	T_{shift} (μs)	$Power$ $(\%)$	Num_{gates}
100	[1 1 1 1]	538	5.38	100	190
50	[3 2 1 2]	198	3.96	50	246
25	[6 4 2 4]	100	4	25	338
12.5	[6 4 2 4]	100	8	12.5	344

It is interesting to note that by using lower shift frequency f_s not only the power consumption during test is reduced, but the testing time can also be decreased in several cases. In Table III power denotes the percentage of power consumption for the case when $f_t = f_s$ (note, only the dynamic power component is accounted for). When the available TAM width for hCADT00 is 4, if we select the shift frequency $f_s = f_t = 100$ MHz, the shift time for each pattern will be 5.38 μs, which is larger than the shift time of 3.96 μs for $f_s = 50$ MHz. This is because, the latter case allows for a better distribution of VTB lines to different clock domains. In addition, if a small increase in testing time is acceptable, the test power can be further reduced. For example, if we select the shift frequency $f_s = 25$ MHz, the testing time will be 4 μs rather than 3.96 μs with $f_s = 50$ MHz, however, the test power will be decreased by an additional factor of 2.

In terms of area overhead, VTB-DIU and VTB-MIU blocks need one flip-flop for each virtual test bus line and additional logic for multiplexing/de-multiplexing if the shift frequency is lower than the tester frequency. The capture window size and the number of clock domains decide the hardware overhead of the scan control block. As it can be seen in the last column of Tables II and III, the new MFCW will introduce an additional area overhead (for hCADT00) in the range of 190 to 344 equivalent two-input NAND gates (2NANDs). This data is compiled using a 0.18-process technology where our results indicate that the overhead (for hCADT00) introduced by scan only is 1463 2NANDs and by scan and P1500 logic is 4512 2NANDs (this value is rather large since the number of I/Os is 313). Even if the number of I/Os will be lower, we believe that the added overhead to scan and P1500 logic will be around 10%. For complex cores with hundreds of thousands of gates this is insignificant, when compared to the benefits of facilitating at-speed multifrequency test of IP-protected cores.

IV. Conclusion

This paper proposed a new core wrapper design for IP cores, which, by means of a capture window, facilitates multifrequency at-speed testing, while accepting data from a low-speed tester, at the expense of small on-chip area overhead. In addition, the power consumption during test is decreased by shifting data with lower frequency without penalizing the testing time. The proposed architecture provides a P1500 compatible solution for multifrequency IP core test.

Acknowledgment

The authors wish to thank P. T. Gonciari from Southampton University for motivating discussions and his insightful suggestions. Thanks are also due to the anonymous reviewers for their constructive comments.

References

[1] IEEE P1500 Web Site [Online]. Available: http://grouper.ieee.org/groups/1500/

[2] S. Bhawmik, “Method and apparatus for built-in self-test with multiple clock circuits,” U.S. Patent 5,680,543, Oct. 21, 1997.

[3] K. Chakrabarty, Ed., *SOC (System-on-a-Chip) Testing for Plug and Play Test Automation*. ser. Frontiers in Electronic Testing (FRET). Boston, MA: Kluwer, 2002.

[4] W. J. Dally and J. W. Poulton, *Digital Systems Engineering*. Cambridge, U.K.: Cambridge Univ. Press, 1998.

[5] B. N. Dostie, A. S. Hassan, D. M. Burek, and S. K. Sunter, “Multiple clock rate test apparatus for testing digital systems,” U.S. Patent 5,349,587, Sep. 20, 1994.

[6] P. T. Gonciari, B. M. Al-Hashimi, and N. Nicolici, “Addressing useless test data in core-based system-on-a-chip test,” *IEEE Trans. Computer-Aided Design Integrat. Circuits Syst.*, vol. 22, no. 11, pp. 1568–1580, Nov. 2003.

[7] D. Heidel, S. Dhong, P. Hofstee, M. Immediato, K. Nowka, J. Silberman, and K. Stawiasz, “High speed serializing/de-serializing design-for-test method for evaluating a 1 GHz microprocessor,” in *Proc. 16th IEEE VLSI Test Symp. (VTS)*, 1998, pp. 234–238.

[8] G. Hetherington, T. Fryars, N. Tamarapalli, M. Kassab, A. Hassan, and J. Rajski, “Logic BIST for large industrial designs: Real issues and case studies,” in *Proc. IEEE Int. Test Conf. (ITC)*, 1999, pp. 358–367.

[9] V. Iyengar, K. Chakrabarty, and E. J. Marinissen, “Co-optimization of test wrapper and test access architecture for embedded cores,” *J. Electron. Testing: Theory Applicat.*, vol. 18, no. 2, pp. 213–230, Apr. 2002.

[10] ——, “Integrated wrapper/TAM co-optimization, constraint-driven test scheduling, and tester data volume reduction for SOCs,” in *Proc. 39th Design Automation Conf. (DAC)*, New Orleans, LA, Jun. 2002, pp. 685–690.

[11] V. Jain and J. Waicukauski, “Scan test data volume reduction in multiclocked designs with safe capture technique,” in *Proc. IEEE Int. Test Conf. (ITC)*, Oct. 2002, pp. 148–153.

[12] S. Koranne, “A novel reconfigurable wrapper for testing of embedded core-based SOC’s and its associated scheduling algorithm,” *J. Electron. Testing: Theory Applicat.*, vol. 18, no. 4/5, pp. 415–434, Aug. 2002.

[13] E. Larsson and Z. Peng, “A reconfigurable power-conscious core wrapper and its application to SOC test scheduling,” in *Proc. IEEE Int. Test Conf. (ITC)*, Charlotte, NC, Sep. 2003, pp. 1135–1144.

[14] X. Lin and R. Thompson, “Test generation for designs with multiple clocks,” in *Proc. 40th Design Automation Conf. (DAC)*, Jun. 2003, pp. 662–667.

[15] 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. IEEE Int. Test Conf. (ITC)*, 1998, pp. 284–293.

[16] E. J. Marinissen, S. K. Goel, and M. Lousberg, “Wrapper design for embedded core test,” in *Proc. IEEE Int. Test Conf. (ITC)*, Atlantic City, NJ, Oct. 2000, pp. 911–920.

[17] B. Nadeau-Dostie, D. Burek, and A. S. M. Hassan, “ ScanBist: a multifrequency scan-based BIST method,” *IEEE Des. Test Comput.*, vol. 11, no. 1, pp. 7–17, Spring 1994.

[18] N. Nicolici and B. M. Al-Hashimi, *Power-Constrained Testing of VLSI Circuits*, ser. Frontiers in Electronic Testing (FRET). Boston, MA: Kluwer, 2003.

[19] S. Pateras, “Achieving at-speed structural test,” *IEEE Des. Test Comput.*, vol. 20, no. 5, pp. 26–33, Sep.–Oct. 2003.

[20] J. Pouget, E. Larsson, Z. Peng, M.-L. Flottes, and B. Rouzeyre, “An efficient approach to SoC wrapper design, TAM configuration and test scheduling,” in *Proc. IEEE Eur. Test Workshop (ETW)*, Maastricht, The Netherlands, May 2003, pp. 51–56.

[21] R. Mahmud. Techniques to make clock switching glitch free. [Online]. Available: http://www.eetimes.com/story/OEG20030626S0035

[22] J. Schmid and J. Knablein, “Advanced synchronous scan test methodology for multi clock domain ASICs,” in *Proc. IEEE VLSI Test Symp. (VTS)*, Dana Point, CA, 1999, pp. 106–113.

[23] N. Touba and B. Pouya, “Testing embedded cores using partial isolation rings,” in *Proc. IEEE VLSI Test Symp. (VTS)*, Monterey, CA, Apr. 1997, pp. 10–16.

[24] P. Varma and S. Bhatia, “A structured test re-use methodology for core-based system chips,” in *Proc. IEEE Int. Test Conf. (ITC)*, Washington, DC, Oct. 1998, pp. 294–302.

[25] B. Vermeulen, S. Oostdijk, and F. Bouwman, “Test and debug strategy of the PNX8525 Nexperia digital video platform system chip,” in *Proc. IEEE Int. Test Conf. (ITC)*, Oct. 2001, pp. 121–130.

[26] Y. Zorian, S. Dey, and M. Rodgers, “Test of future system-on-chips,” in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design (ICCAD)*, Nov. 2000, pp. 392–398.

Qiang Xu (S’03) received the B.E. and M.E. degrees from Beijing University of Posts and Telecommunications, China, in 1997 and 2000, respectively. He is currently pursuing the Ph.D degree in electrical and computer engineering at McMaster University, Hamilton, ON, Canada.

His research interests are in the area of system-on-a-chip design and test.

Mr. Xu is a recipient of a Best Paper Award at the 2004 IEEE/ACM Design, Automation, and Test in Europe (DATE) Conference.

Nicola Nicolici (S’99–M’00) received the Dipl. Ing. degree in computer engineering from the University of Timisoara, Romania, in 1997, and the Ph.D. degree in electronics and computer science from the University of Southampton, U.K., in 2000.

He is an Assistant Professor of computer engineering at McMaster University, Hamilton, ON, Canada. His research interests are in the area of computer-aided design and test, and he has authored a number of papers in this area.

Dr. Nicolici received the IEEE TTTC Beausang Award for the Best Student Paper at the International Test Conference (ITC 2000) and the Best Paper Award at the IEEE/ACM Design Automation and Test in Europe Conference (DATE 2004). He is a member of the ACM SIGDA and the IEEE Computer and Circuits and Systems Societies and he serves on the editorial board of *IEE Proceedings—Computers and Digital Techniques*.