Proceedings of National Conference on Computing, Electrical, Electronics and Sustainable Energy Systems

Implementation of Ultra Large-Scale Soc Test Control Architecture With Scan Test Bandwidth Management Using Multihop

P. Akhila[1] | K. Jayaram Kumar[2]

[1]PG Scholar, Department of ECE, Godavari Institute of Engineering and Technology, Rajahmundry, Andhra Pradesh, India.
[2]Assistant Professor, Department of ECE, Godavari Institute of Engineering and Technology, Rajahmundry, Andhra Pradesh, India.

ABSTRACT

With the increase in chip size and complexness, the direct or bus interconnects in standard SoC check management models square measure rather restricted. During this paper, we have a tendency to propose a replacement distributed multihop wireless check control network supported the recent development in "radio-on-chip" technology. The proposed design consists of 3 basic parts, the check computer hardware, the resource configurators and therefore the RF nodes that support the communication the check hardware and clusters of cores. Underneath the construction tree structure, the resources (including not solely the circuit blocks to perform checking, however conjointly the on-chip radio-frequency nodes for intra-chip communication) square measure properly distributed and system optimization is performed in terms of each check application time and check management value.

KEYWORDS: *Bandwidth management, check access mechanism, check application time, scheduling, check compression*

Copyright © 2016 International Journal for Modern Trends in Science and Technology
All rights reserved

I. Introduction

The integration of a whole system, that till recently consisted of multiple ICs on a PCB, onto one chip is termed as System-on-Chip (SoC) that uses embedded reusable cores. The technology of micro chip style and manufacture advances, additional and additional transistors are often placed on a semiconductor device. This is a continuous increase in the within style complexness posses variety of challenges to the system integrators while incorporating the check methodologies. Since cores in associate SoC don't seems to be directly accessible via chip inputs and outputs, special access mechanisms square measure needed to check them at system level, also known as Check Access Mechanisms (TAMs). It is used to deliver the check stimuli from the check source to cores and additionally to deliver responses from cores to the sink. The potency of a tam-o-shanter depends on to what extent it can cut back test time that is, time to check all cores in the SoC. Optimized architectures square measure required to check the System-on-Chip during cost-effective manner. Except for the testing of the cores, the interconnect between them additionally ought to be tested. This primarily means some input pattern to be applied at the origin of the interconnect and the worth be checked at the other end. A number of interconnects can be checked at the opposite finish variety of interconnects are often tested in parallel, if the check resources square measure on the market. Thus, to scale back the overall testing time for the chip, it is necessary that we have a tendency to contemplate the core checking associated interconnect testing in an integrated fashion. Another vital issue throughout checking is that test power consumption. Tho' affected planning to match the power budget has been planned, another vital part of power reduction is that of scan chains. a variety of cores measure placed on a specific tam-o'-shanter, their check patterns can suffer the wrappers scan cells. This can be significantly true for wrappers designed with none by-pass mechanism. Thus, the order within which the cores are square measure placed on a tam-o'-shanter determines the shift and the associated power consumptions.

II. Literature Survey

This chapter deals with the theory behind the topics analyzed in this thesis. The first section of this chapter is a study of two DFT techniques that are widely used. The second section discusses why high check application time is a problem worth solving. The third section explains the power considerations during check, which plays an important role in deciding the scan clock frequency.

Design For testability (DFT) Techniques

As discussed earlier, as the size of circuit's increases, their check complexity also increases. The internal nodes in the circuits become harder to check. Circuits are therefore modified so that they can be checked better. This section describes some of the techniques used to improve the quality of check.

Scan Design

A combinational circuit with n inputs has 2n possible input combinations. As n increases, the number of possible input vectors increases exponentially. It is therefore impossible to apply all possible input vectors to check the circuit. A subset is therefore chosen such that a sufficient percentage of the faults can be captured by the check. Sequential circuits are harder to check than combinational circuits. This is due to the presence of memory elements (shown in Figure 1) which create internal states during circuit operation. An exhaustive check would involve application of all possible input vectors at all possible states of the memory elements. This number becomes large even for small circuits.

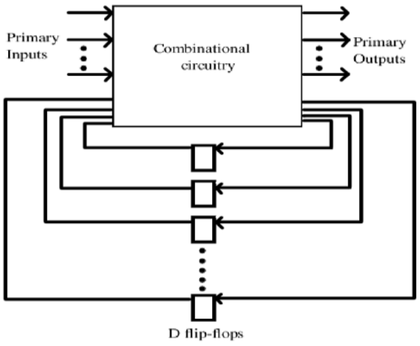


Figure 1: Sequential Circuit

In order to improve the testability of sequential circuits, they are enabled with a check mode. When the circuit is in the check mode, the flip-flops in the circuit are chained together to form one or more shift registers. Thus, the flip-flops can be sent to any state without depending on the values at the primary inputs. The flip-flops serve as points of controllability and observability and help in achieving better check coverages. There are two widely used types of scan designs - full scan and partial scan designs. Full scan designs utilize all flip-flops in the circuit to generate shift registers [11]. Partial scan designs [1] use a selective set of flip-flops to form shift registers. The flip-flops are chosen [18], [13] such that they minimize overhead without loss of coverage.

Built-In self test

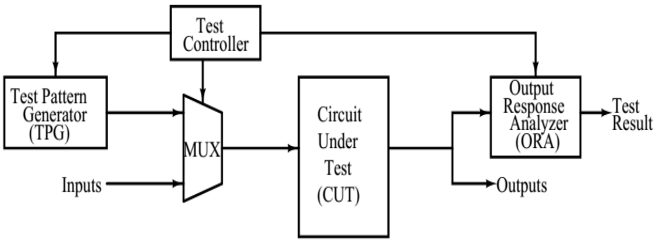


Figure 2: BIST Implementation

BIST is a DFT technique in which additional hardware is added to the circuit to be checked so that it can check itself. BIST is widely used since it makes the chip easier and faster to be checked. The basic circuitry required to implement BIST is shown in Figure 2. The patterns required for check are generated through a number of techniques [11]. One of them is to store the check patterns in a ROM on the chip. This method uses a lot of chip area and is hence not very widely used. Counters can be used to generate exhaustive check sequences. However, the number of exhaustive inputs is very high for any normal-sized circuit and hence the check time required is very high. A more commonly used technique is the use of a Linear Feedback Shift Register (LFSR) that generates pseudo-random pattern sets. A large number of check patterns are used in this method but the area overhead on-chip is very low. A large number of outputs are received from the circuit under check. Storing the correct values of all these bits would add a lot of extra hardware to the chip. The circuit responses are therefore reduced to a size that can be stored on the chip. This is done through a number of response compaction methods. A widely used method compacts responses with an LFSR [25]. Some other methods count number of transitions, or use parity information and so on.

III. Proposed System

Control data delivery

The approach summarized in Section II does not make any specific provisions for the way control data is delivered to SoC check logic in order to setup check configurations. It appears, however, that the number of check configurations, and hence the amount of control data one needs to employ and transfer between the ATE and DSR address registers, may visibly impact check scheduling and the resultant check time. Consequently, we begin this paper by analyzing three alternative schemes that can be used to upload control bits and show how they determine the final SoC check logic architecture.

A. Using IJTAG

The IEEE 1687 is a proposed standard for accessing on-chip check and debug features via the IEEE 1149.1 test access port (TAP). The purpose of this internal Joint test Action Group (IJTAG) standard is to automate the way one can manage on-chip instruments, and to describe a language for communicating with them via the IEEE 1149.1 check data registers (TDRs). If there is an IJTAG network available on the SoC, and the total number of check configurations is relatively small, one can use it to deliver the control data, as shown in Fig. 3. The SoC design of Fig. 3 has a single TAP and three different blocks: 1) two cores (C1 and C2) under check and 2) the DSR interfacing ATE withC1 and C2. TAP can be instructed to enable a check path via the IEEE 1687 segment insertion bits (SIBs). Every SIB is used to either enable or disable the inclusion of an instrument into the path from a check data In to a check data output. The TDR in C1 orC2 can be either bypassed or loaded with data putting both cores into specific check modes. The TDR in DSR receives the control data indicating which core and which of its check channels are connected to which ATE channels.

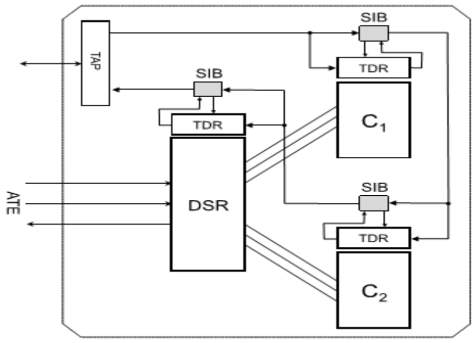


Figure 3: Using IJTAG network to transfer control data

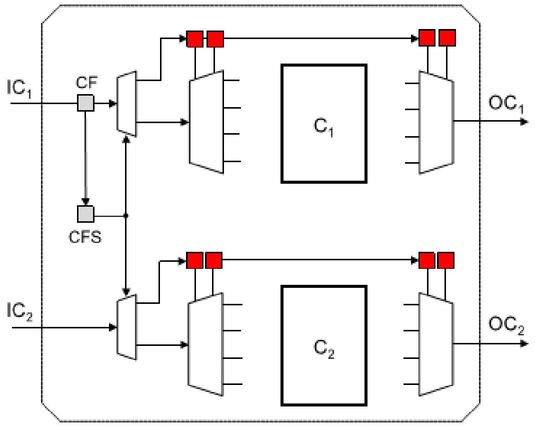


Figure 4: Dedicated control chain-based architecture.

The advantage of using the IJTAG network to deliver the DSR control data is a simple and easy way to implement flow as the network is frequently used to set the cores TDRs. However, such an approach can only support a limited number of configurations. This is because the IJTAG shift clock is typically 10 to 20 times slower than the scan shift clock. Delivering a large volume of control data can incur an unacceptable total check time overhead. Consequently, this architecture can be used for a relatively small number of check configurations. If the TDRs work with parallel update registers and many patterns use the same configuration, a low throughput IJTAG control can be mitigated. If one changes the control state rather seldom, the next configuration vector can be shifted in coincidently with application of check patterns, followed by updating TDR when ready to switch to the new configuration. It requires more DFT logic, though.

B. Check Control Architectures

In this section, we first introduce the basic network components. Then we present three proposed check control architectures.

Network Components

Three basic components are used in the proposed check control architectures: the check scheduler, the resource configurators and the RF (radio frequency) nodes dispersed on the SoC. The check scheduler is employed as a central controller, it (1) carries out the chip level check procedure, including the checking of the interconnects between the cores, the checking of the user-defined logics around the cores and the core checking, (2) communicates with the resource configurators and also with the chip external, such that no conflict arises during resource utilization and check application, (3) configures the routing of the check control path for each individual core, and (4) provides proper check control signals to carry out the check procedure of the selected core. The function of the resource configurator is to configure the check resources required for checking a particular core on command of the scheduler . A set of check resources (i.e., the circuit blocks required to perform checking) is distributed in the system for checking the cores. At any particular control step, each resource is configured into its appropriate operating mode by the control signals. In case when more than one checks share common check resources, the resource configurators are activated such that no conflicts result in the use of resources. The RF node is a radio-frequency interface for (two-way) communication between the scheduler and IP cores. Particularly, one RF node is dedicated to the scheduler. The distribution of RF nodes chip-wide provides the coverage of the entire on-chip wireless communication. To reduce the routing cost and area overhead, one RF node is shared by a cluster of cores which are hard-wired to it. For example, as shown in Figure 1,cores ✂✁ , ☎✄ , ✝✆ and ✞✟ are organized into one cluster and are wired to the RF node. In addition, the IP cores in the system are organized into clusters and each has the IEEE P1500 wrapper interface to switch between different modes according to the control signals received.

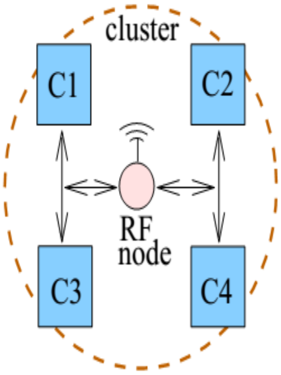


Figure 5: A RF node in a cluster of cores.

C. *Miniature Wireless LAN Based Check Control Network*

Our first proposal is a miniature wireless LAN (local area network) that works as the intra-chip check control network for system-chips, where the scheduler broadcasts control signals through the attached RF node as shown in Figure 2. A single wireless channel is shared by all RF nodes in the chip and the control signals sent from the scheduler will be received by all RF nodes. Each RF node has a unique ID and each control signal is attached with an ID field to specify the intended recipient. Upon receiving a signal, a node checks the ID field through its local decoder . If the signal is intended for the receiving node, the node processes the control signal, otherwise, it is just ignored. By specifically assigning the ID (for example, reserving one bit to indicate multicasting while the remaining bits are to hold a group number), we can also support multicasting to a subset of RF nodes and consequently a subset of cores can be checked concurrently

When a core finishes checking, the related RF node needs to notify the scheduler its completion. Since the schedule of the checks is predetermined, each RF node is given exclusive access to the network in a predetermined order. Permission to transmit signals to the scheduler is passed from one RF node to another using a special message called a poll and the polling order is maintained by the scheduler according to the schedule result. When the scheduler receives the completion signal from the RF node which holds the poll, it then forwards the poll to the next node in the polling sequence. This centralized polling scheme has its unique features as compared to the conventional polling network, which divides time into alternating types of intervals: polling intervals, during which the poll is transferred between stations, and transmission intervals, during which the station with the poll transmits packets. Our scheme is quite simplified due to the fact that the scheduler knows in advance the completion time of each check and the transmission time is quite short. Thus it's not necessary to maintain the polling and transmission intervals. By using the polling scheme, no collision occurs even when multiple checks finish checking at the same time.

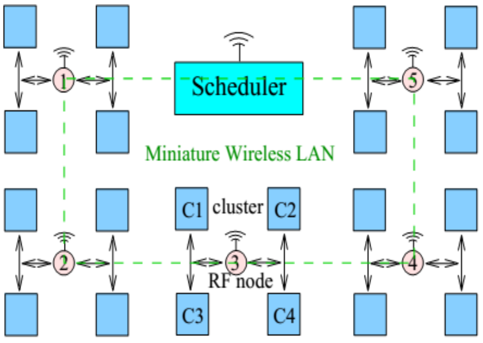


Figure 6: Miniature wireless LAN based architecture

D. Using Pipelines

One can also use the regular scan channels to deliver controls through pipelining stages, as shown in Fig. 7. For each channel, this approach concatenates n+m control bits, where n and mare the numbers of control bits used by the In demultiplexers and output multiplexers, respectively. Moreover, each control bit is shadowed to avoid distorting check configurations in the middle of check data shifting. The shadow registers are updated at the end of each pattern upload. Thus, when a check pattern launches a new check configuration, the corresponding control data need to be loaded with its predecessor. Clearly, the first vector is exclusively a setup one. The architecture of Fig. 7 supports as many check configurations as required. However, the control data is always uploaded through the ATE channels as an integral part of a check vector. Hence, given a check configuration, the same control data is repeated for all check patterns. The amount of control data is small, though, as the number of control bits per channel is typically a tiny fraction of the check pattern shift cycles.

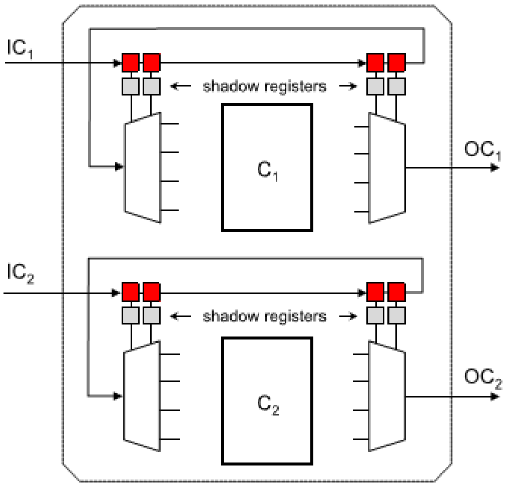


Figure 7: Pipeline architecture.

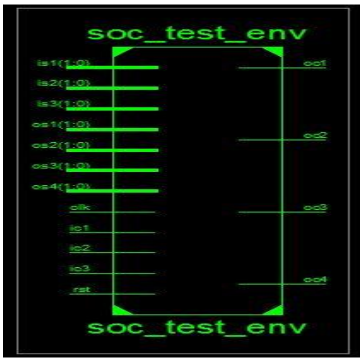
IV. Results and Examination

By making use of the waveforms, it is very easier to evaluate the effectiveness of the proposed architecture through simulation. Modelsim is used for simulation and Xilinx is used for evaluating the response for the existing and proposed architecture.

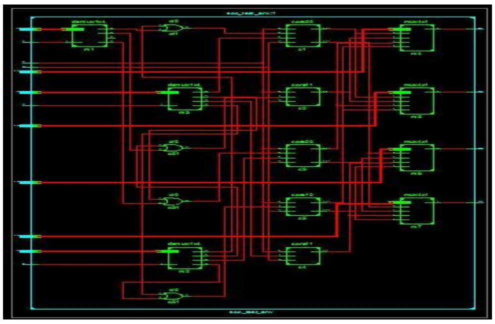
Proposed Method:

A. Soc Check environment:

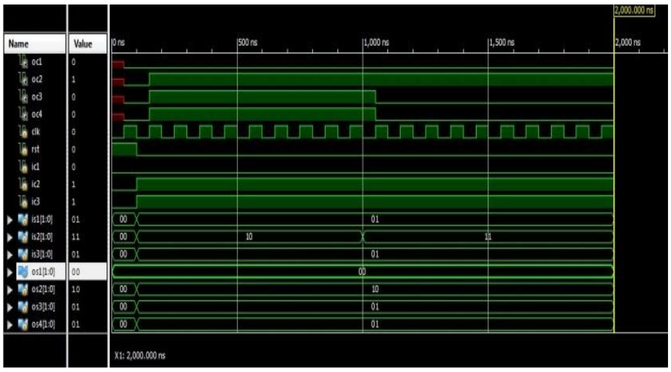
Block diagram:



Technology schematic:

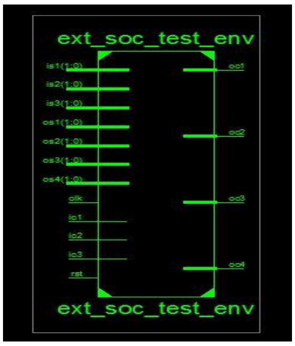


Simulation results:

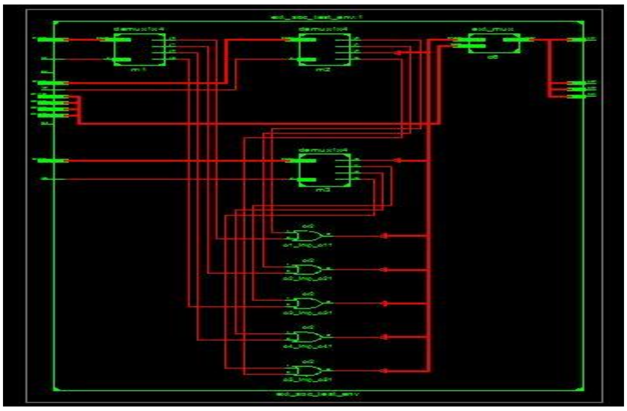


B. Single multiplexed SoC environment:

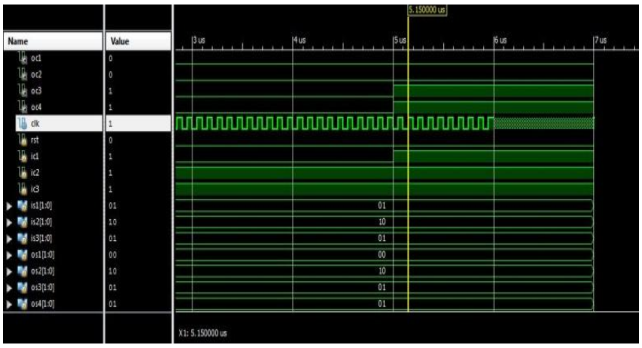
Block diagram:



Technology schematic:



Simulation results:



Name	Value
oc1	0
oc2	0
oc3	1
oc4	1
clk	1
rst	0
ic1	1
ic2	1
ic3	1
is1[1:0]	01
is2[1:0]	10
is3[1:0]	01
os1[1:0]	00
os2[1:0]	10
os3[1:0]	01
os4[1:0]	01

V. Performance Analysis

The performance simulations of implementation ultra large-scale soc check control architecture with scan check bandwidth management using multihop are carried out with MODELSIM ALTERA 6.5e Simulator. Synthesis has been carried out with XILINX ISE. The check frequency is 132.258MHz; the power supply voltage is 1.1 V and delay of 9.010ns (Levels of Logic = 6); (6.723ns logic, 2.287ns route). Number of slices used 11, Number of 4 inputs LUT's used 21, and Number of bonded IOB's used 21.

VI. Future Work

In future research, several system optimization issues such as RF nodes placement, the optimal number of RF nodes and routing problems will be addressed in detail under the multilevel tree structure. Techniques need to be presented for the integration of check resource distribution and system optimization among TAM design, check scheduling (concurrent core checking as well as interconnect checking) under power and cost constraints.

VII. Conclusion

In this paper, we have proposed a novel distributed wireless check control network using the "radio-on-chip" technology for future high density, high volume embedded system chips. Three types of control architectures, i.e., miniature WLAN, multihop scheme and distributed multihop scheme have been presented and the system optimization has been performed on control constrained check resource partitioning and distribution. Simulations using randomly generated check sets and experiments with benchmarks will be performed for evaluation and verification of the proposed check optimization algorithms.

Acknowledgment

I would like to thank my guide and all the reviewers for their constructive comments and active support to my work.

References

[1] Y. Zorian, "A distributed BIST control scheme for complex VLSI devices," in *Proc. IEEE VLSI Check Symposium*, pp. 4–9, April 1993.

[2] M.-C. F. Chang, V. Roychowdhury, L. Zhang, H. Shin, and Y. Qian, "RF/wireless interconnect for inter- and intra-chip communications," *Proc. of The IEEE*, vol. 89, pp. 456–466, April 2001.

[3] G. L. Craig, C. R. Kime, and K. K. Saluja, "Check scheduling and control for VLSI built-in self-check," *IEEE Trans. on Computers*, vol. 37, pp. 1099–1109, September 1998.

[4] K. Chakrabarty, "Check scheduling for core-based systems using mixedinteger linear programming," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 19, no. 10, pp. 1163–1174, Oct. 2000.

[5] K. Chakrabarty, V. Iyengar, and M. D. Krasniewski, "Check planning for modular checking of hierarchical SOCs," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 24, no. 3, pp. 435–448, Mar. 2005.

[6] S. K. Goel and E. J. Marinissen, "Effective and efficient check architecture design for SOCs," inProc. Int. Check Conf. (ITC), 2002, pp. 529–538.

[7] S. K. Goel, E. J. Marinissen, A. Sehgal, and K. Chakrabarty, "Checking of SoCs with hierarchical cores: Common fallacies, check access optimization, and check scheduling,"IEEE Trans. Comput., vol. 58, no. 3, pp. 409–423, Mar. 2009.

[8] . Huang et al., "Optimal core wrapper width selection and SOC check scheduling based on 3-D bin packing algorithm," inProc. Int. Check Conf. (ITC), 2002, pp. 74–82.

[9] P. T. Gonciari and B. M. Al-Hashimi, "A compression-driven check access mechanism design approach," in Proc. 9th IEEE Eur. Check Symp. (ETS), May 2004, pp. 100–105.

[10] V. Iyengar and K. Chakrabarty, "System-on-a-chip check scheduling with precedence relationships, preemption, and power constraints," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 21, no. 9, pp. 1088–1094, Sep. 2002.

[11] V. Iyengar, K. Chakrabarty, and E. J. Marinissen, "Check access mechanism optimization, check scheduling, and checker data volume reduction for system-on-chip, "IEEE Trans. Comput., vol. 52, no. 12, pp. 1619–1632,Dec. 2003.

[12] V. Iyengar, A. Chandra, S. Schweizer, and K. Chakrabarty, "A unified approach for SoC checking using check data compression and TAM optimization," inProc. Des., Autom. Check Eur. Conf. Exhibit. (DATE), 2003, pp. 1188–1189.

[13] B. A. Floyd, C.-M. Hung, and K. K. O, "Intra-chip wireless interconnect for clock distribution

implemented with integrated antennas, receivers, and transmitters," *IEEE Journal of Solid-State Circuits*, vol. 37, pp. 543–552, May 2002.

[14] K.-J. Lee and C.-I. Huang, "A hierarchical check control architecture for core based design," in *Asian Check Symp.*, pp. 248–253, December 2000.

[15] B. A. Floyd, C.-M. Hung, and K. K. O, "Intra-chip wireless interconnect for clock distribution im-plemented with integrated antennas, receivers, and transmitters," *IEEE Journal of Solid-State Circuits*, vol. 37, pp. 543–552, May 2002.

[16] B. Moore, D. Garvey, M. Margala, and C. Backhouse, "Wireless checking techniques and circuits for deep submicron VLSI circuits," in *Proc. of the 12th International Conference on Wireless Communications*, pp. 224–235, July 2000.