Publication of Little Lion Scientific R&D, Islamabad PAKISTAN
Journal of Theoretical and Applied Information Technology
15th March 2011. Vol. 25 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

OPTIMIZED TEST SCHEDULING WITH REDUCED WRAPPER CELL FOR EMBEDDED CORE TESTING

K. CHAKRAPANI [1] DR. P. NEELAMEGAM [2]
[1]School of Computing, SASTRA University
[2]School of EEE, SASTRA University

E-mail: kcp@core.sastra.edu, neelkeer@eie.sastra.edu

ABSTRACT

The increasing Design for Test (DfT) area overhead and potential performance degradation is caused due to wrapping all the embedded cores for modular System-on-Chip (SoC) testing. This paper proposes a solution for reducing the number of Wrapper Boundary Register (WBR) cells. By utilizing the WBRs of the surrounding cores to transfer test stimuli and responses, the WBRs of some cores can be removed without affecting the testability of the SoC. We denote the cores without WBRs as light-wrapped cores and present a new modular SoC test architecture for concurrently testing both the wrapped and the light-wrapped logic cores. Since the WBRs of cores that transfer test stimuli and test responses for light-wrapped cores become shared resources during test, conflicts arise during test scheduling that will negatively impact the test application time. The algorithm for SoC test scheduling and light-wrapped logic cores works under multiple constraints (test power dissipation, test resources and test priorities) and applies a Power Swarm optimization based optimum search for a solution to the scheduling problem. We consider the experiments on several SoC benchmark circuits and demonstrate that, with an acceptable increase in test application time, the number of WBRs can be significantly decreased.

Keywords - *System-on-Chip, Test Access Mechanism, Process Algebra, Fuzzy Logic.*

1. INTRODUCTION

Large-scale integration has added enormous complexity to the process of testing modern digital circuits. Besides, during the past several years, Integrated Circuit technology evolved from chip-set philosophy to embedded cores based *system-on-a-chip (SoC)* concept[1], which simply refers to an IC, designed by stitching together multiple stand-alone *VLSI* designs to provide full functionality for an application.
These innovations are already on their way to the next generation of cell phones, multimedia devices, and PC graphics chipsets. The core-based design, justified by the necessity to decrease time-to-market, has created a host of challenges for the design and test community[2]. The core test integration is a complex problem – the chip integrator can modify the test and add *Design for Test (DfT) and Built-In Self-Test (BIST)* features, if necessary. Specifically, in the context of embedded cores-based system testing, electrical isolation involving the input and output ports of the core from the chip or other cores is a necessity. The fundamental items of interest in a core test are access, control, and isolation, and these are the issues which were addressed by the *IEEE Technical Council on Test Technology Working Group 1500* which has been entrusted with the responsibility of developing standard architecture for their solution. The embedded core test requires hardware components like wrapper around the core, a *source* and a *sink* for test patterns (*on-chip* or *off- chip*) and an on-chip *Test Access Mechanism (TAM)* to connect the wrapper to the source or sink[3]. The cores could be without boundary scan or with boundary scan. For design and test reuse, *ASIC* manufacturers have suggested certain characteristics. In general, different *DfT* and *BIST* schemes like scan, partial scan, *logic BIST* and *scan-based BIST* are used to test various logic blocks within a *SoC* like microprocessor or microcontroller. However, the main problem is still the resulting area overhead and performance penalties. Structural test methods like scan and *BIST* are desirable for test reuse, portability, and test integration into the *SoC* test set. The *TAM* includes on-chip test generation logic for cores with *BIST*[3]. The *DfT* techniques involve adding optimized test logic

Publication of Little Lion Scientific R&D, Islamabad PAKISTAN
Journal of Theoretical and Applied Information Technology
15th March 2011. Vol. 25 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

within cores and at the chip level to enhance testability and *DfT* logic helps in test pattern generation and application, and assist in the support test environment. In this paper, test methodologies are proposed for embedded core-based *SoC* digital systems comprising of wrapper and *TAM*. The fault model used is the conventional single *stuck-fault* model. The nature of faults is single stuck faults. Thus each line can have only two types of stuck faults: *stuck-at-1* and *stuck-at-0*. The IEEE 1500-compliant wrapper separates the core under test from other cores. The *TAM* plays a vital role in transporting the test patterns to the desired core and the core responses to the output pin of the *SoC*. The *TAM* is implemented as a signal transport medium, which is shared by all the cores in the *SoC*. Once the compilation of the cores is done, the fault simulation is carried out with the test patterns feeding the cores through the *TAM*. The selection of the appropriate core is taken care by the daemon program running in the background[4]. The simulation process is completely automatic, and requires no intervention from the designer during the test generation process. This paper describes the architecture of the wrapper and test access mechanism, together with models of the *SoCs* being used, based on application environment.

2. 500-BASED INTEGRATION ARCHITECTURE

The *IEEE 1500* wrapper has various modes of operation. There are modes for functional (non test) operation, *inward facing* (*IF*) test operation, and *outward facing* (*OF*) test operation. Different test modes determine whether the serial test data mechanism (*WSI–WSO*) or the parallel test data mechanism (*WPI–WPO*), if present, is being utilized. Instructions loaded into the *Wrapper Instruction Register (WIR)*, together with the *IEEE 1500* wrapper signals, determine the mode of operation of the wrapper and possibly the core itself. There is a minimum set of instructions and corresponding operations that shall be supplied. Optional instructions and their corresponding behavior are also defined, together with the requirements for extension of the instruction set. All instructions that establish test modes and that utilize the parallel port *WPI* and *WPO* are optional, as the presence of this port is optional. Furthermore, *IEEE 1500* allows user-defined instructions as well.

IEEE 1500 has a set of instructions that are defined to use only the *serial interface (WSP)* and a corresponding set of instructions that are defined for the *parallel interface*. *IEEE 1500* allows accessibility to test the core.

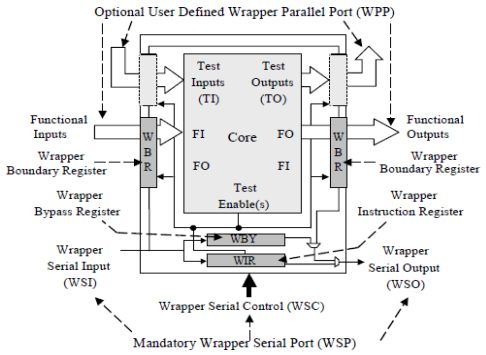

Fig.1: Standard IEEE 1500 wrapper Components

There is one main core test instruction—*Wx_INTEST* (user-specified core-test instruction)—that is flexible enough to allow any core test to execute. There are two other instructions that are mandatory: an instruction for functional mode (*WS_BYPASS*) and an instruction for external test mode (*WS_EXTEST*). *WS_BYPASS* puts the wrapper into the bypass configuration and allows access to all functional terminals of the core shown in Fig 1. *WS_EXTEST* is the serial *EXTEST* configuration of the wrapper. Even if there is a *WP_EXTEST* mode (for parallel access), there must still be a *WS_EXTEST* instruction capability. The signal connected to the *WRCK* terminal is a dedicated clock used to operate *IEEE 1500* functions.

3. LIGHT WRAPPERS FOR EMBEDDED CORES

From the system integrator's standpoint, to test the embedded cores and their *interconnects*, full *controllability* and *observability* need to be provided at the inputs and outputs of each core. To ensure the modularity and scalability of an *SOC* test methodology, the controllability and observability of each embedded core should be

Publication of Little Lion Scientific R&D, Islamabad PAKISTAN
Journal of Theoretical and Applied Information Technology
15th March 2011. Vol. 25 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

test set independent. To achieve this, it is not necessary to wrap all the core's terminals with *WBR* cells, since the system integrator can also exploit the functional interconnect between cores to transfer the test data. To illustrate this observation, producers and consumers are introduced. For a given *Core-i*, its producers are the cores that feed its *PIs* and its consumers are the cores that capture its *POs* in the normal (functional) mode. Fig. 2 shows a part of an *SOC* where *Core3* is not wrapped with *WBR* cells; however, all its producers (*Core1*, *Core2*) and its consumer (*Core4*) are *P1500-wrapped*. For *INTEST* of *Core3*, the controllability of its input terminals is provided through its producers' output *WBR* cells while the observability of its output terminals is provided through its consumer's input *WBR* cells. In other words, we can shift in its test stimuli through the output *WBR* cells of *Core1* and *Core2*, feed in the test stimuli into *Core3* through its normal functional path, and then capture its test response and shift it out through the input *WBR* cells of *Core4*

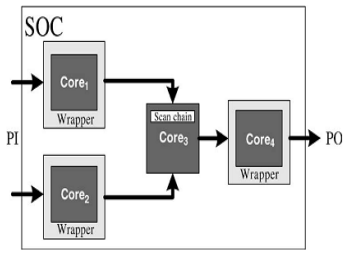

Fig 2. Full controllability and observability for Core3 without WBR cells.

4. SWARM OPTIMIZATION BASED OPTIMUM SEARCH

The *PSO* model is a new population based optimization strategy introduced by J. Kennedy et al. in 1995. It has already shown to be comparable in performance with traditional optimization algorithms such as simulated annealing and the genetic algorithm[16]. However, *PSO* does exhibit some disadvantages: it sometimes is easy to be trapped in local optima, and the convergence rate decreased considerably in the later period of evolution; when reaching a near optimal solution, the algorithm stops optimizing, and thus the accuracy that the algorithm can achieve is limited[19]. Theoretical results have shown that the particle positions in standard *PSO* oscillate in damped sinusoidal waves until they converge to points in between their previous best positions and the global best positions discovered by all particles so far. If some point visited by a particle during this oscillation has better fitness than its previous best position, then particle movement continues, generally converging to the global best position discovered so far. All particles follow the same behavior, quickly converging to a good local optimum of the problem. However, if the global optimum for the problem does not lie on a path between original particle positions and such a local optimum, then this convergence behavior prevents effective search for the global optimum. It may be argued that many of the particles are wasting computational effort in seeking to move in the same direction towards the local optimum already discovered, whereas better results may be obtained if various particles explore other possible search directions[18].

Each particle *Pi* consists of a position vector x_i which represents the candidate solution of the optimization problem, a velocity vector v_i and a memory vector *pibest* of the best candidate solution encountered by the particle. Each particle flies in the dimensional problem space with a velocity which is dynamically adjusted according to the flying experiences of its own and its colleagues. The position of a particle is updated by

$$x_i(t1) = x_i(t)v_i(t1) - - - 1$$

and its velocity according to

$$v_i(t+1) = wv_i(t) + c_1 r_1\left(P_{ibest}(t) - x_i(t)\right)$$

$$+c_2 r_2\left(p_{gbest}(t) - x_i(t)\right) - - - -2$$

Algorithm **Particle Swarm optimization**

Create and initialize a n-dimensional swarm S and set *t = 0* ;

repeat
for *each particle i=1,…,s*
if *fi <fibest*
S.pibest = S.xi ; fibest = fi ;
else if *fi >fiworst*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

S.piworst = S.xi ; fiworst = fi ;
endif
if *fibest < fgbest*
S.pgbest = S.pibest ; fgbest = fibest ;
endif
endfor
for *each particle i=1,…,s*
compute its current activity *activity(Pi(t))* and inertia weight *wi(t)*;
if *activity(Pi(t))<a(t)*
update its velocity *vi(t)* using Eq.(1);
else
update its velocity using Eq.(2);
endif
update its current position *xi(t)* using Eq.(1);
endfor
t++;
until *stopping condition is true*;
return *S.pibest* ;

The *scheduling algorithm* selects wrapper design for each core, assigns a start time, an end time and which *TAM* wires to use for each core in such a way that the test application time is minimized while all constraints are satisfied. The *Optimal Time* is a lower bound that represents the *"ideal"* situation, but, due to the TAM structure and the wrapper design, this limit is almost never reached. The *Optimal Time* is calculated using the formula:

$$OptimalTime = \left| \frac{\sum_i bestPareto}{W_{max}} \right|$$

where W_{max} is the number of available pins for test access (the TAM bandwidth).
The *Optimal Time* gives the lower bound of the total test time of the system when no constraints but *TAM* width limitations are considered. In the ideal case, the schedule does not contain any idle times (i.e. there is no cost loss between tests in the test schedule), and it is therefore the best test application time that can ever be achieved.

5. EXPERIMENTAL RESULTS AND DISCUSSIONS

We have applied our wrapper design, *TAM* design and test scheduling algorithms to the *ITC'02 benchmarks.* The results for system *d695* are given in Table 1, respectively. For each *TAM* width, the first group of columns gives the results of the *ITC Benchmark*, in terms of the test time indicating the quality of the test schedule, and the time used to generate the solution. The last group of columns gives the corresponding results of our efficient approach. The Pareto-optimal points are chosen as close as possible to the *TAM* width limit. The experimental results show clearly that our approach outperforms the multiplexed approach in terms of test scheduling lengths.

Core Ci	Time Ti	Power Pi	Width Wi
1	216	14.7mW	2
2	578	16.6mW	2
3	428	15.9mW	2
4	912	16.7mW	3
5	814	16.9mW	3
6	879	17.51mW	4
7	1072	21.3mW	6
8	378	17.3mW	3
9	256	15.4mW	3
10	329	17.8mW	3

Table 1: Time, power, width relationship for embedded cores.

6. CONCLUSION

The Recent technology development has made it possible to design and manufacture extremely complex systems[6]. In this paper we have proposed a particle *swarm optimization* with light wrapper core for embedded core based test scheduling technique that minimizes the test application time by allowing tests to be applied as concurrently as possible. The technique takes *test power consumption* and *test conflicts* into account when minimizing the test application time. The test conflicts we consider are due to unit testing with multiple test sets, hierarchical *SOCs* where cores are embedded in cores, and the sharing of TAM wires.

7. REFERENCE

[1]. Larson, E., K. Arvidsson, H. Fujiwara and Z. Peng, 2004. *"Efficient test solutions for core based designs."* IEEE Trans. Comput. Aid. Des. Integrat. Circ. Syst., 23: 758-774. DOI: 10.1109/TCAD.2004.826560

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[2]. Ravi, S., L. Ganesh and N.K. Jha, 2001 " *Testing of core-based systems-on-a-chip"*. IEEE Trans. Comput. Aid. Des. Integrat. Circ. Syst., 20: 426-439. Digital Object Identifier 10.1109/43.913760

[3]. Sehgal, A., V. Iyengar and K. Chakrabarty, 2004*" SoC test planning using virtual test access architectures"* IEEE Trans. Very Large Scale Integrat. Syst., 12: 1263-1275. DOI: 10.1109/TVLSI.2004.834228

[4]. Shao, J., G. Ma, Z. Yang and R. Zhang, 2008 " *Process algebra based soc test scheduling for test time*

[5]. *Minimization"* Proceeding of the IEEE Computer Society Annual Symposium on VLSI, Apr. 7-9, IEEE Xplore Press, Montpellier, pp: 134-138. DOI: 10.1109/ISVLSI.2008.88

[6]. Yoneda, T., M. Imanishi and H. Fujiwara, 2007 " *An SoC test scheduling algorithm using reconfigurable union wrappers"* Proceeding of the Design, Automation Test Europe Conference and Exhibition, Apr. 16-20, IEEE Xplore Press, Nice, pp: 1-6. DOI: 10.1109/DATE.2007.364596

[7]. Shaer, B., D. Landis, and S. Al-Arian, *"Partitioning algorithm to enhance pseudoexhaustive testing of digital VLSI circuits,"* IEEE TVLSI, Volume: 8 Issue 6, Dec. 2000 pp. 750 -754.

[8]. Brglez, F., and H. Tujiwara, *"A Neutral Netlist of 10 Combinational Benchmark Circuits and A Target*

[9]. *Translator in Fortran,"* IEEE Int'l Symp. On Circuits and Systems, 1985.

[10]. Al-Arian, S. and R. Bolling, *"Improving the Testability of VLSI Circuits through Partitioning"* IEEE Sym. on Circuits and Systems, Vol. 4, 1994, pp. 199-202.

[11]. Dr. S., Venayagamoorthy, G.K., Gudise, V.G., 2004 *"Optimal PSO for collective robotic search applications"* IEEE Congress on Evolutionary Computation, June 20–23, pp. 1390–1395.

[12]. Gudise, V.G., Venayagamoorthy, G.K., 2004 " *FPGA placement and routing using particle swarm optimization"* IEEE Computer Society Annual Symposium on VLSI, February 19–20, pp. 307–308.

[13]. Hu Xiaohui, A., Eberhart, R., 2002 " *Multi objective optimization using dynamic neighborhood particle swarm optimization"* In: Proceedings of the 2002 Congress on Evolutionary Computation, vol. 2, May 12–17, pp. 1677–1681.

[14]. Kennedy, J., Eberhart, R., 1995 *"Particle swarm optimization"* In: Proceedings IEEE International Conference on Neural Networks, vol. IV, Perth, Australia, pp. 1942–1948. ISBN 1558605959.

[15]. Shaer, B., Al-Arian, S., Landis, D., 2000b " *Partitioning sequential circuits for pseudoxhaustive testing"*. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 8 (5), 534–541.

[16]. Shi, Y., Eberhart, R.C., 1998 *"A modified particle swarm optimizer"* In: Proceedings of the IEEE Congress on Evolutionary Computation, pp. 69–73.

[17]. D. K. Agrafiotis, and W. Cedeno. Feature Selection for Structure-Activity Correlation Using Binary Paricle Swarms. *J. Med. Chem*., 2002, 45, 1098.

[18]. Nadia Nedjah and Luiza de Macedo Mourelle," Systems Engineering using Particle Swarm Optimization", Series:Intelligent Systems Engineering Book Series - Nadia Nedjah and Luiza de Macedo Mourelle (State Univ. of Rio de Janeiro) - Series Editors Binding: Hardcover Pub. Date: 2007 ISBN: 1-60021-119-4

[19]. Konstantinos E. Parsopoulos (University of Ioannina, Greece); Michael N. Vrahatis (University of Patras, Greece) ," Particle Swarm Optimization and Intelligence: Advances and Applications" , DOI: 10.4018/978-1-61520-666-7 ,ISBN13: 9781615206667

[20]. Panda, S., & Padhy, N. P. (2007). Comparison of particle swarm optimization and genetic algorithm for tcsc-based controller design. *International Journal of Computer Science and Engineering, 1*, 1–49.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AUTHOR PROFILES

Prof *K. Chakrapani* obtained his B.E in Electronics and Communication from Regional Engineering College, Tiruchirapalli, M.Tech in Computer Science from SASTRA University, Thanjavur and MS in Information Technology from Bharathidasan University, Tiruchirapalli. Currently he is working as an Assistant Professor at TIFAC CORE, SASTRA University, Thanjavur. He is a life member of ISTE. He can be contacted at *kcp@core.sastra.edu* and his hand phone number is +919362637203

Dr. P.Neelamegam is currently working as a professor at SASTRA University, Thanjavur. He can be contacted at *neelkeer@eie.sastra.edu* and his hand phone number is +919443077327.