

# Wrapper Design of Embedded Cores for Three Dimensional System-on-Chips (SoC) using available TSVs

Surajit Kumar Roy<sup>1</sup>, Chandan Giri<sup>2</sup>, Sourav Ghosh and Hafizur Rahaman<sup>3</sup>

Dept. of Information Technology

Bengal Engineering & Science University

Shibpur, Howrah - 711103, India

Email:<sup>1</sup>suraroy@gmail.com,<sup>2</sup>chandangiri@gmail.com, <sup>3</sup>rahaman\_h@yahoo.co.in

**Abstract**—Core based three-dimensional(3D) integrated circuits (ICs) design is an emerging field of semiconductor industry that promises greater number of devices on chip, increased performance and reduced power consumption. But due to scaling in technology features these chips are more complex and hence testing of these 3D ICs is a challenging task. This paper follows a P1500-style wrapper design for 3D ICs using through silicon vias (TSVs) for testing purpose. It is assumed that the core elements are distributed over several layers of the ICs. As the number of available TSVs are limited due to small chip area, this work is intended to design balanced wrapper chains using available TSVs. In this work we have proposed a polynomial time algorithm of  $O(N)$  to design the test wrapper. The results are presented based on the ITC'02 SOC test benchmarks and compared with prior work. Obtained results show that our algorithm provides better utilization of TSVs compared to the work presented in [1].

**Keywords**—Scan chain; 3D integrated circuits; wrapper design; test access mechanism

## I. INTRODUCTION

System-on-Chip (SOC) is common for today's integrated circuits. SOC designers can integrate different types of embedded cores such as ROM, DSP processor, combinational logic blocks, finite state machines (FSM), ALU, multipliers, comparators etc. on a single chip. Each of these cores need to be tested individually after manufacturing of the chip. For this reason modular test approach has been adopted[2]. To test the core using modular testing approach, Test Access Mechanism (TAM) and wrapper are introduced. TAM carries test vectors from the test source to any module under test and transport module output responses to the test sink. The modular testing is being done by designing IEEE standard 1500 [3] wrapper. Basic elements for designing the wrapper of a core are functional inputs, functional outputs and internal scan chains. The wrapper design and TAM optimization have great impact on SOC testing, because they directly affect the core testing time.

Due to the increasing demand of high performance, high density portable hand held application, electronic system's design has shifted from 2D design space to 3D design space. The 3D ICs are built by stacking multiple device layers and interconnecting them using vertical interconnects known as

Through Silicon Vias(TSVs). TSVs offer the greatest vertical density and therefore is the most promising one among all the vertical interconnect technologies.

The design of a 3D IC can be done at two levels of granularity [1]: a) *Coarse-granularity partitioning*: Here the embedded cores in the SOC is like a 2D design space.

b) *Fine-granularity partitioning*: cores are partitioned in multiple layers that shows significant improvement in performance and overall frequencies. For example, it is shown in [4] that repartitioning of the Intel Core 2 processor across four die-stacked tiers can improve 47.9% increase in overall frequency and 47% performance.

Manufacturing of 3D ICs are now possible, but the support of CAD tools for design and testing are not available for commercial use as it needs to optimize several design related constraints. Hence in this work, we are trying to focus on wrapper design and optimization of cores with available number of TSVs for vertical interconnects taking as a constraint. It is known that the test application time of a core is dependent on longest wrapper chain because it determines the time needed to load and read out test data. So during the wrapper chain design it is also one of the consideration to design the balanced wrapper so that test time can be minimized.

The rest of the paper is organized as follows. Section II discusses about the previous works done related to testing of 3D SOC. Section III describes the problem formulation. Proposed algorithm for solving the problem is presented in Section IV. Section V presents the experimental results based on ITC'02 benchmark circuits. Finally, Section VI draws the conclusion.

## II. PREVIOUS WORKS ON WRAPPER DESIGN

Previous works on wrapper optimization and test infrastructure design for 2D SOC [2], [5], [6] have been proposed in last few years. Optimization methods include ILP [2], bin packing [7], [6], Genetic Algorithm[6] and heuristic method [5].

There are few limited work on 3D SOC testing. TAM architecture design for 3D SOC is provided in [8] where ILP technique is used to divide the TAM wires into several

test buses and assign the cores to test buses to minimize the test time under TSV constraint. Jiang et al. [9] proposed simulated annealing (SA) based algorithms to optimize the pre-bond and post-bond test time of 3D ICs. A heuristic method to reduce weighted test cost with constraints on test pin width in pre-bond and post bond are discussed in [10]. In [11] authors have presented an optimization method to minimize the test time for a 3D-SIC, either for the final stack test or for any number of multiple test insertions during bonding. In [12], the authors have made an attempt to design 1500 Std. based test wrapper for 3D SOC. The authors have modeled 1500-style wrapper optimization in 3D ICs using optimum number of TSVs available for testing.

### III. PROBLEM FORMULATION

The main problem of the wrapper optimization is to minimize the test time of each core. The problem can be elaborated as follows: *Given a core for 3D IC that consists of different functional parameter such as number of functional inputs, number of functional outputs, set of scan chains, length of each of the scan chains and TAM width, determine (a) the distribution of the core elements over several layers of 3D IC for a number of wrapper chains and (b) connects them through the maximum available number of TSVs ( $TSV_{max}$ ) for a core such that the length of the longest wrapper chain is minimized.*

To solve this problem which is *NP-Hard* in nature[2], we have proposed one heuristic approach that try to distribute the core wrapper elements optimally over several layers of the IC so that the length of the longest wrapper chains is minimized in polynomial time of  $O(N)$ , where  $N$  is the total number of core wrapper elements.

### IV. PROPOSED METHODOLOGY

The goal of this heuristic method is to distribute the core wrapper elements over different layers of IC into a number of wrapper chains using 3D design constraint as maximum number of available TSVs ( $TSV_{max}$ ). Scan chains can present in multiple layers and hence the scan-in and scan-out also can present at different layers. The lowest layer of the IC is layer number 1, followed by 2, 3 and so on up to  $L$ , the maximum number of layer present in the IC. So we are given the followings: a set of core wrapper elements  $E = \{E_1, E_2, \dots, E_{x+y+n}\}$ , TAM width  $W$  and maximum number of available TSVs ( $TSV_{max}$ ), which are used for routing of the wrapper chains. Here  $x$  is the number of functional inputs,  $y$  is the number of functional outputs and  $n$  is the number of internal scan chains of the core. During the design of the wrapper chain it is considered that TSVs internal to the scan chains are not counted against TSV constraint as assumed in [1].

#### A. Heuristic Algorithm

The algorithm **Createchain()** (as shown in Algorithm 1) starts execution by initializing *Element* and *Wrapper* data

structures. Data structure *Element* stores the information about each and every wrapper elements  $E_i$  of the list  $E$ . The information about the wrapper elements include the type of element (whether I/O wrapper cell or internal scan chain), layer number in which it is placed and the length of this wrapper element. These three different information are stored in the variables *Type*, *Layer* and *Length*. Data structure *Wrapper* maintains three different lists using three variables corresponding to each type of wrapper elements  $E_i$ . Information about the number of TSVs required at any instant of time of designing the wrapper is stored in the variable *No\_of\_tsv*. Two other variables named *wrapper\_length* and *No\_of\_element* are used to hold the length of the wrapper chain and number of different types of wrapper elements contain by the wrapper chain. The length of wrapper chain before starting of the algorithm are initialized with 0.

The algorithm tries to pick up one core wrapper element and layer number of the selected element is considered randomly. But for every wrapper chain we have calculated the required number wrapper I/O cells where both the wrapper input and output cells are distributed almost equally among the wrapper chains. This is done because the assignment of wrapper I/O cells directly affect the number of TSVs required for a chain. When we have added an wrapper input or output cell in a wrapper chain during design we have checked with respect to the a priori calculated I/O wrapper cells. If the I/O wrapper cell requirement is satisfied for the current chain then only selected wrapper element is assigned otherwise we have found out another wrapper chain where the wrapper I/O cell requirement is satisfied. When we assign first wrapper element to any one of the wrapper chain, it is assumed that this must be placed at any layer except the lowest. Wrapper elements are assigned to the wrapper chain in clockwise fashion. For example, if four wrapper chains are to be designed then order will be  $W_1 \rightarrow W_2 \rightarrow W_3 \rightarrow W_4$ . Each time we have inserted an wrapper element into a wrapper chain, number of TSVs required for that wrapper chain is calculated. We keep on inserting element into the same wrapper chain until no extra TSV is required to connect this new element. When all the available TSVs are utilized we have assumed that rest of the elements will be placed in the lowest layer. This method will be continued until all the elements are assigned.

#### B. Analysis of the Algorithm

The **Createchain()** algorithm will attempt to place all  $N$  wrapper elements in  $W$  wrapper chains. According to the algorithm, for every wrapper elements to be placed in a wrapper chain we have checked whether the wrapper chain satisfies its wrapper I/O cell requirement or not. In the best-case scenario, every time the wrapper I/O cell requirement of the wrapper chain will satisfy and the element is placed in the current chain. Thus the best case complexity of the

---

**Algorithm 1:** CreateChain

---

**Input :** Test set  $E$ , Maximum available TSV ( $TSV_{max}$ ), Number of layers ( $L$ ), TAM\_width ( $W$ )

**Output:** Designed wrapper chain

```
1 begin
2   Initialize  $k = 0$ ,  $TSV\_count = 0$ ,  $tflag = 0$ ,
    $pretsv = 0$ ,  $posttsv = 0$ , set  $V = \phi$ ,
    $layer\_number(E_i \in E, \text{ for all } i) = -1$ ;
3   For each wrapper chain  $W_k$  determine wrapper I/O cells
   requirements  $R_k$ , where  $k = 1, 2, \dots, W$ ;
4   while  $E \neq \phi$  do
5     begin
6       if ( $pretsv \neq posttsv || tflag == 1$ ) then
7          $k = (k + 1) \% W$ ;
8         Pick up an element  $E_i \in (E - V)$  if available;
9         if ( $TSV\_count == TSV_{max}$ ) then
10           $tflag = 1$ ;
11          Set  $layer\_number(E_i \in (E - V)) = 1$ ;
12          if ( $length(W_k) == 0$ ) then
13            set  $layer\_number(E_i) \in \{2, 3, \dots, L\}$ ;
14          else set  $layer\_number(E_i) \in \{1, 2, 3, \dots, L\}$ ;
15          if ( $type\_of(E_i) == I/O \text{ wrapper cell}$ ) then
16            begin
17              if ( $no\_of\_I/O\_wrapper\_cell_k + 1 \leq R_k$ ) then
18                set  $r = k$ ;
19              else
20                begin
21                  Find next wrapper chain  $W_r (r \neq k)$  that has
22                   $No\_of\_I/O\_wrapper\_cell_k + 1 \leq R_k$ ;
23                end
24               $No\_of\_I/O\_wrapper\_cell_r += 1$ ;
25            end
26          if ( $type\_of(E_i) = scan \text{ chain}$ ) then
27            set  $r = k$ ;
28            Add  $E_i$  to  $W_k$ ;
29             $No\_of\_elements += 1$ ;
30             $pretsv = No\_of\_TSV$ ;
31             $posttsv = Required \text{ TSV for } W_r \text{ after insertion of } E_i$ ;
32             $No\_of\_tsv_r = posttsv$ ;
33             $wrapper\_length(W_r) += length(E_i)$ ;
34             $V = V \cup E_i$ ;
35             $E = E - E_i$ ;
36             $k = r$ ;
37          end
38        end
39      end
40    end
```

---

algorithm is  $O(N)$ .

If we pick up an I/O wrapper cell and the I/O wrapper cell requirement of the current wrapper chain is not satisfied, then we have to find out the another wrapper chain that satisfies the required condition. In the worst case, we have to search  $(W - 1)$  wrapper chains to find out the desired wrapper chain where the currently picked wrapper I/O cell has to be assigned. According to the algorithm, for the scan chains no condition checking has been done. Thus for remaining  $(N - n)$  wrapper I/O cells required condition has to be checked. Hence, the worst case time complexity of the algorithm is  $O(N + (W - 1) * (N - n))$ . But the number

of elements in a core is much higher than the number of wrapper chains to be designed. Hence  $W \ll N$  and the algorithm has worst case complexity  $O(N)$ .

## V. EXPERIMENTAL RESULTS

The proposed algorithm is implemented in C++ language and executed on a Intel Core 2 Duo processor having 1GB RAM. The simulation results are provided based on the cores from ITC'02 SOC test benchmarks. Number of layers for the SOC chip considered for each case of the simulation is 3. The simulation results for the core 7 of SOC d281 is presented in Table I for the range of TAM width between 2 to 6. Table II represents the results for core 4 of SOC p93791 for the range of TAM width between 2 to 8. Column 1 of each table lists the TAM width, Column 2 represents the maximum number of available TSVs ( $TSV_{max}$ ). Column 3 shows the total number of TSVs utilized for wrapper design using the proposed algorithm. Length of the longest wrapper chain is shown in Column number 5 obtained according to our algorithm. Column number 4 compares with the length of the longest wrapper length as given in [1]. Column number 6 and 7 compares the CPU running time of the proposed algorithm and the algorithm presented in [1]. It is seen from the Table I, Table II that for lower number of wrapper chains some TSVs are still unused. Hence, for lower number of wrapper chains where more number of I/O wrapper cells has to be connected we can design that wrapper chain even with less number of TSVs. In [1] authors have proposed polynomial time ( $O(N^3)$  and  $O(N^2)$ ) heuristic algorithms to design the wrapper for ITC'02 SOC cores that span over multi-layer. Comparison is made with respect to the longest wrapper chain length obtained by us. Again, it is seen from the Table I, Table II that for most of the cases when the number of available TSVs are less the algorithm proposed in [1] is not able to design the wrapper chain which is not the case arise according to our proposed algorithm. So, we can conclude that, the proposed algorithm is better than [1] with respect to the following reasons: (a) complexity is very less, (b) can design the wrapper for even less number of available TSVs.

## VI. CONCLUSION

In this work we have proposed an polynomial time ( $O(N)$ ) heuristic algorithm to minimize the test time by reducing the wrapper chain length for 3D core-based SOC. We have distributed the core wrapper elements over different wrapper chains and calculated the required number of TSVs. Our goal was to design the balanced wrapper chains so that the length of the longest wrapper chain is minimum using the available number of TSVs for a particular core. Results are presented on the cores of ITC'02 SOC benchmarks and it shows that the proposed algorithm design the wrapper chain within considerable time which is much less than the algorithm presented in [1] and efficiently utilize the available

TAM Width	Max TSV	Req TSV	Longest Wrapper Length [1]	Longest Wrapper Length	CPU Time (Min) (Our)	CPU Time (Min) [1]
2	12	10	N/A	1159	0.11	≈ 0.00
	14	10	N/A	1319	0.11	≈ 0.00
	16	10	1633	1223	0.11	0.22
	18	10	1064	1223	0.15	1.42
	22	12	1064	1095	0.13	2.03
3	12	12	N/A	784	0.13	≈ 0.00
	14	14	N/A	846	0.11	≈ 0.00
	16	16	1633	814	0.13	0.25
	18	15	710	816	0.13	3.67
	22	15	710	878	0.13	6.73
4	12	12	N/A	629	0.11	≈ 0.00
	14	14	N/A	626	0.1	0.02
	16	16	1633	596	0.16	0.30
	18	18	532	724	0.11	7.05
	22	20	532	789	0.08	16.17
5	12	12	N/A	458	0.13	0.02
	14	14	N/A	458	0.16	0.05
	16	16	1633	458	0.16	0.37
	18	18	470	490	.11	11.98
	22	20	426	618	0.1	32.28
6	12	12	N/A	409	0.1	0.02
	14	14	N/A	409	0.1	0.07
	16	16	1633	409	0.13	0.47
	18	18	459	441	0.13	16.75
	22	20	355	473	0.13	53.83

Table I  
RESULTS FOR CORE 7 OF D281

TSVs. Future work includes the modification of the proposed algorithm to design more balanced wrapper chains so that testing time can be reduced further.

#### ACKNOWLEDGEMENT

Part of this work is sponsored by the DST, Govt. of West Bengal, India under the grant of VLSI Design Project.

#### REFERENCES

- [1] B. Noia, K. Chakrabarty, and Y. Xie, "Test-Wrapper Optimization for Embedded Cores in TSV-Based Three-Dimensional SOCs," in *IEEE International Conference on Computer Design*, 2009, pp. 70–77.
- [2] V. Iyengar, K. Chakrabarty, and E. J. Marinissen, "Test Wrapper and Test Access Mechanism Co-Optimization for System-on-Chip," *Journal of Electronic Testing: Theory and Applications (JETTA)*, vol. 18, pp. 213–230, 2002.
- [3] "IEEE Std. 1500: IEEE Standard Testability method for Embedded Core-based Integrated Circuits," IEEE Press, 2005.
- [4] K. Puttaswamy and G. H. Loh, "Thermal herding: Microarchitecture techniques for controlling hotspots in high performance 3d integrated processors," in *IEEE High Performance Computer Architecture*, 2007, pp. 193–204.
- [5] S. K. Goel and E. J. Marinissen, "SOC Test Architecture Design for Efficient Utilization of Test Bandwidth," *ACM Trans. Design Automation of Electronic Systems*, vol. 8, no. 4, pp. 399–429, 2003.
- [6] C. Giri, S. Sarkar, and S. Chattopadhyaya, "A Genetic Algorithm Based Heuristic Technique for Power Constrained Test Scheduling in Core-based SOCs," in *Proc. IEEE Intl. Conf. on IFIP VLSI-SOC*, 2007, pp. 320–323.

TAM Width	Max TSV	Req TSV	Longest Wrapper Length [1]	Longest Wrapper Length	CPU Time (Min) (Our)	CPU Time (Min) [1]
2	12	10	N/A	81	0.1	≈ 0.00
	13	11	N/A	94	0.13	≈ 0.00
	14	10	89	97	0.1	≈ 0.00
	18	11	77	77	0.15	≈ 0.00
	20	11	77	86	0.15	≈ 0.00
3	12	12	N/A	58	0.11	≈ 0.00
	13	13	N/A	69	0.1	≈ 0.00
	14	14	84	52	0.08	≈ 0.00
	18	18	51	55	0.15	≈ 0.03
	20	19	51	52	0.08	≈ 0.03
4	12	12	N/A	60	0.11	≈ 0.00
	13	13	N/A	55	0.1	≈ 0.00
	14	14	79	45	0.1	0.02
	18	18	39	56	0.06	0.07
	20	19	39	63	0.15	0.08
5	12	12	N/A	48	0.1	≈ 0.00
	13	13	N/A	32	0.08	≈ 0.00
	14	14	79	42	0.06	0.02
	18	18	37	56	0.06	0.12
	20	20	35	51	0.15	0.12
6	12	12	N/A	33	0.08	≈ 0.00
	13	13	N/A	37	0.13	≈ 0.00
	14	14	79	37	0.1	0.03
	18	18	31	38	0.08	0.18
	20	20	28	40	0.1	0.32
7	12	12	N/A	27	0.15	≈ 0.00
	13	13	N/A	32	0.13	0.02
	14	14	79	35	0.11	0.07
	18	18	30	35	0.06	0.43
	20	20	30	35	0.06	0.87
8	12	12	N/A	25	0.1	≈ 0.00
	13	13	N/A	25	0.1	0.08
	14	14	79	29	0.11	0.18
	18	18	30	40	0.1	1.37
	20	20	25	30	0.15	3.32

Table II  
RESULTS FOR CORE 4 OF P93791

- [7] Y. Huang, S. M. Reddy, W.-T. Cheng, P. Reuter, N. Mukherjee, C.-C. Tsai, O. Samman, and Y. Zaidan, "Optimal Core Wrapper width selection and SOC Test Scheduling based on 3-D Bin Packing Algorithm," in *Proc. International Test Conference*, 2002, pp. 74–82.
- [8] X. Wu, Y. Chen, K. Chakrabarty, and Y. Xie, "Test Access Mechanism Optimization for Core-based Three-dimensional SOCs," in *IEEE International Conference on Computer Design*, 2008, pp. 212–218.
- [9] L. Jiang, L. Huang, and Q. Xu, "Test Architecture Design and Optimization for Three-dimensional SOCs," in *Proc. Design, Automation and Test in Europe*, 2009, pp. 220–225.
- [10] L. Jiang, Q. Xu, K. Chakrabarty, and T. Mak, "Layout-driven test-architecture design and optimization for 3d socs under pre-bond test-pin-count constraint," in *IEEE International Conference on Computer Design*, 2009, pp. 191–196.
- [11] B. Noia, K. Chakrabarty, , and E. J. Marinissen, "Optimization methods for post-bond die-internal/external testing in 3d stacked ics," in *Proc. International Test Conference*, 2010.
- [12] S. K. Roy, S. Ghosh, H. Rahaman, and C. Giri, "Test wrapper design for 3d system-on-chip using optimized number of tsvs," in *IEEE Intl. Symposium on Electronic System Design*, Orissa, Bhubaneswar, India, December 2010.