International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9, Issue-7, May 2020

TSV Optimized Test Wrapper Design for Fine Grain Partitioned 3D System on Chip

Harpreet Vohra, Ashima Singh

Abstract*: The 3D System-on-chip (SoC) technology supports the vertical interconnectivity required for the purpose of functional, supply and test access purposes through the use of Through Silicon Vias (TSVs). Little number of available TSVs for test purpose necessitates the optimization of test infrastructure. This paper proposes an algorithm to design the test wrapper for the 3D cores such that the number of the TSVs used per TAM chain are minimized. Test time optimization is done by balancing the lengths of the individual Wrapper chain inside the core. The proposed heuristic firstly distributes the different core elements on the given TAM chains and then uses a diagraph for their insertion ordering to get minimum possible TSV utilization. Simulation results are presented for the different cores of the ITC'02 SoC benchmark circuits. Results show that TSVs can be reduced to 20-30 percent with around 60-70 percent reduction in CPU time utilization for heavy SoCs in comparison to the other proposed techniques.

Keywords* : Optimal test scheduling, Test architecture, SoC Test, Embedded core testing, Wrapper cell design.

I. INTRODUCTION

System-on-chip (SoC) designs are common these days wherein the system integrator embeds the heterogeneous cores bought either from within the company or third party vendors on a single chip [1]. Increasing on chip complexity operating at very high frequency led to number of design and performance. For example: the long interconnects that span the complete integrated circuits (IC) periphery started posing capacitive and inductive effects which in turn produced delay and power concerns [2]. Developments of the semiconductor fabrication technology led to the expansion of ICs in the vertical dimension. This technology drift brings in solution for the long interconnects to a much reduced sizes through the use of the through silicon vias (TSV) in the vertical dimensions [3]. Using TSV technology, 3D ICs are created by placing multiple device dies together through wafer or die stacking, which are then connected using vertical TSVs [4].

Revised Manuscript Received on May 30, 2020.
***** Correspondence Author
Harpreet Vohra*, Assistant professor , Department of Electronics and Communication Engineering , Thapar Institute of Engineering & Technology, Patiala,India.
Ashima Singh, Assistant Professor, Computer Science and Engineering Department, Thapar Institute of Engineering & Technology, Patiala,India.

Such ICs have brought the various advantages like smaller footprint, reduced interconnect latency, less power requirements etc. Based on the style of the circuit design, the 3D SoCs can either be coarse grained partitioned or circuit partitioned [5]. In the former, each embedded core in the SoC is still a 2D design while their placement can be done on different dies while in later, each core may spread over multiple dies. As per the ITRS roadmap, test cost has become the bottleneck in reducing the overall cost of the system [6]. Test time which is an integral component of the overall cost of an IC, is dependent on the time required to transport the test data between its source, IC periphery and different core components. Out of these, the former can be reduced by minimizing the amount of test data by using techniques like test data compression [7,8] etc. whereas, the later needs efficient test infrastructure design which includes optimized core test wrappers and test architecture [9,10]. In the last three decades, lot of research focus has been given to reduce the test cost of 2D SoCs through the optimization of: core test Wrapper [11-14], Test Access Mechanism (TAM) [15,16] and Test Scheduling [12,17,19].

In case of the coarse grain partitioned SoCs, though the individual cores are 2D but, they may be placed on different dies. In order to efficiently deliver the test data to various cores, test wires need to span/ routed between various dies. Similarly, to save the test time, multiple cores need to be tested in parallel. Optimization techniques of TAM architecture and test schedules have been presented in [14-16]. In case of circuit partitioned SoC, the functional components like input/ output cells and the internal scan chains of individual cores may lie on different dies. Core components needs to be optimally stitched together to form the wrapper chains so that the test time of the core can be reduced. While doing so, TSVs are used wherever the chain needs to be routed from one layer to another. Therefore, optimal wrapper design should keep the number of TSVs used to be less than or at most equal to the global limit. In this paper, a solution for the wrapper optimization for 3 D SoCs under tight TSV constraint is proposed.

The paper is organized as follows: Section 2 presents the work done on system on chip test along with wrapper design problem for 2D SoCs. Section 3 presents the problem formulation for solving the wrapper optimization problem. Section 4 describes the proposed wrapper design solution and Section 5 gives simulation results for several cores from the ITC '02 SoC test benchmarks. Finally, Section 6 presents conclusions drawn from this work.

II. RELATED WORK IN 2D AND 3D SOC TEST

The core test wrapper forms the interface between the embedded core and its environment and helps match the available external TAM to the internal
wrapper chains [10-11]. It helps perform the test expansion as well as core isolation to facilitate the modular test. In case the number of TAM wires is less than the number of functional input/ output elements, the wrapper chains help in serialization of the test vectors for further application to the respective core elements. Out of all core elements, the length of the scan chain going to be larger so it is preferable to insert them first which helps balance the wrapper chains in a massive way. [13]. A P1500 core wrapper supports both core-internal and core external testing through the use of INTEST and EXTEST modes [13], [14]. Another wrapper approach, called Test Collar was proposed in [17]. The wrapper design optimization has been defined to be an NP hard [10]. Authors have proposed the wrappers for the 2D hierarchical cores in [18]. Authors in [19], have explored the TSV and test time co-optimization problem for a single circuit-partitioned 3D IP core. 3D test wrappers design with TAM bandwidth and TSV constraints have been proposed in [26].For a 2D IP core, the problem can be stated as: Given a core having its functional detail as follows: number of inputs I_{core}, number of outputs O_{core}, number of bidirectional pins B_{core}, number of scan chains SC_{core} with their respective lengths $L.SC_{core}(i)$ where $1 \le i \le SC_{core}$, test vectors TV_{core} and TAM bus width TAM_{core}, determine the optimal placement of core elements into wrapper chains such that the length of the longest wrapper chain is minimized. The testing time of an individual core j for available TAM width can be calculated as per the eqns. 1 - 3.

Test time: $T_{core}(j) = (1+ \max(SI_{core}(j), SO_{core}(j)))\ TV_{core}(j) + Min(SI_{core}(j), SO_{core}(j))$ (1)

Where $SI_{core}(j)$ and $SO_{core}(j)$ represents the scan-in and scan-out lengths of the longest wrapper chain k of core j Their values can be extracted as follows:

$$SI_{core}(j) = I_{core}(j,k) + B_{core}(j,k) + \sum_{i=1}^{sc(j,k)} L.SC_{core}(j,k,i) \quad (2)$$

$$SO_{core}(j) = O_{core}(j,k) + B_{core}(j,k) + \sum_{i=1}^{sc(j,k)} L.SC_{core}(j,k,i) \quad (3)$$

Since, the test wrapper is being designed for an individual core so, for here on $j=1$ and is neglected from core parameters. With reference to a realistic model of the 3D core as shown in the Fig. 1 which shows the possibility of occurrence of the wrapper cells and internal scan chains spanning various layers. Each wrapper chain so formed may contain wrapper input cells, internal scan chains and wrapper output cells in succession so as to load and unload the test data.

III. PROBLEM FORMULATION

The objective of an optimized test wrapper design for 3D core can be described as follows: Given a 3D core with its functional details (available TAM width C_{tam}, number of dies N_{dies} and maximum allowable TSV $C.TSV_{max}$), determine its test wrapper design such that the test time of the core can be optimized while keeping the total $C.TSV_{used}$ for all the Wrapper chains to be bounded by the TSV limit. As per the wrapper design [8], the insertion of the functional elements in each wrapper chains needs to be done in following order: input cells, internal scan chains and then output cells.

The core having number of functional input n, functional output m and internal scan chains SC. Placement of Individual components on various dies ranging between 1 and N_{dies} can be described as: $I.Core_i$ signifies the die number of functional input cell i where $1 \le i \le n$; . $O.Core_i$ signifies the die number of functional output cell I where $1 \le i \le m$; $I\_SC.Core_i$ signifies the die number of input cells of the scan chain i, $O\_SC.Core_i$ signifies the die number of output cells of the scan chain i. Lengths of various functional elements can be described as: $Len\_I.Core_i$ and $Len\_O.Core_i$ represents the lengths of the functional input and output cells respectively which are set to one since, both of them are composed of only one flip flop. The length $Len\_SC.Core_i$ of the scan chains i can have any value based on the core's details. The proposed algorithm performs a wrapper chains design of the 3D core such that:

The length of the longest wrapper chain WC is minimized. This is done to minimize the test time of the core (as calculated by eq.1).

The number of internal wrapper chains WC is equal to the external TAM i.e.:

$$\sum_{j=1}^{WC_{total}} WC(j) = C_{tam} \quad (4)$$

The total TSV_{used} should be less than or at max equal to the $C.TSVmax$. i.e.:

$$\sum_{j=1}^{WC_{total}} TSV_{WC}(j) \le C.TSVmax \quad (5)$$

Total number of TSVs used by any wrapper chain j (as given in eq. 5) can be calculated by adding the TSVs used in connecting the functional elements inserted in it i.e.

$$TSVwc(j) = S.TSV(j) + I.TSV(j) + O.TSV(j) \quad (6)$$

where, $S.TSV(j)$, $I.TSV(j)$ and $O.TSV(j)$ represents the TSVs used in insertion of the internal scan chains, core input and output elements respectively. The calculations of each of these can be done using the eqns. 7-9.

Theorem 1: The Minimum number of TSVs, $S.TSV(j)$ required to implement a wrapper scan chain j after insertion of the scan chains is given by:

$$S.TSV(j) = \sum_{s=1}^{sc(j)} \left| \begin{pmatrix} I.SC & (j,s+1) \\ -O.SC & (j,s) \end{pmatrix} \right| \quad (7)$$

where $SC(j)$ denotes the number of scan chains inserted in test wrapper chain j.

Proof: Case 1: When consecutive scan chains of same wrapper chain j reside on same die then no TSV would be required to insert them and hence $S.TSV(j)=0$.

Case 2: If the scan chains lie on different dies, or there is a difference between the output and input scan cell of the scan chains being connected then, TSVs number will increased. Herein, $S.TSV(j)$ will be equal to the difference between the die numbers of the output and input cells of the consecutively scan chains. It may be noted that while doing the calculation of $S.TSV(j)$, the TSVs used as a part of the scan chain formation are not considered.

Theorem 2: The value of the minimum number of TSVs $I.TSV(j)$ needed to insert the functional inputs *n(j)* in the wrapper chain *j* is given by :

$$I.TSV(j) = |I(j,1) - (N_{die=0})| + \sum_{i=1}^{n-1(j)} |I(j,i+1) - I(j,i)| \quad (8)$$

Proof: Case 1: If all the functional input cells and the input scan cell of the first internal scan chain inserted in the wrapper chain *j* all lie on the same die; then, the only $I.TSV(j)$ needed would be the ones that connects the first input cell to the bottom layer. This is reflected by taking N_{die}=0 in the first part of the eq. 8.

Case 2: If there is a difference between the die numbers on which the consecutively occurring input cells of wrapper chain *j* lie then, the difference between their die numbers contribute to the overall $I.TSV(j)$. Same is reflected by the second part of eq. 8.

Case 3: Finally, the difference between the last functional input cells and the input scan cell of the first internal scan chain inserted in the wrapper chain *j* need to be added (as shown in part three of eq. 8) to calculate the $I.TSV(j)$.

Theorem 3: The Minimum number of TSVs, $O.TSV(j)$. required to insert the functional outputs *m* in the wrapper chain *i* is given by:

$$O.TSV(j) = |O(j,m) - (N_{die} = 0)| + \sum_{i=1}^{m-1(j)} |O(j,i+1) - O(j,i)| \quad (9)$$

Proof: *Case 1:* If all the functional output cells $O(j,m)$ and the output scan cell of the last internal scan chain *s* inserted in the wrapper chain *j* all lie on the same die; then, the only $O.TSV(j)$ needed would be the ones that connects the last output scan cell $O(j,m)$ to the bottom layer. This is reflected by taking N_{die}=0 in the first part of the eq. 9.

Case 2: If there is a difference between the placement of consecutive output cells inserted in wrapper chain *j* then, the difference between their die number contribute to the overall $O.TSV(j)$. Same is described by the second part of eq. 9.

Case 3: The difference between the last functional output cells and the output scan cell of the last internal scan chain inserted in the wrapper chain *j* need to be added (as shown in part three of eq. 9) to calculate the $O.TSV(j)$.

The calculation of the $TSVwc(j)$ can become more by considering an illustrative example shown in FIG1. A hypothetical model of the 3D core consisting of 7 functional inputs, 6 functional outputs, 6 scan chains (with length equal to 8,6,5,5,5 and 4 scan cells respectively) is given in FIG 1 a). Out of the various internal scan chains: 3 scan chains lie on same die while rest 3 span different dies (i.e. die number of their input and output cells are different). FIG 1 b) represents a possible style of insertion of these elements in the wrapper chains. In this example, the number of wrapper chains or TAM wires is taken to be three. As can be seen in this case the lengths of different wrapper chains have been balanced out to minimize difference between the minimum and maximum scan lengths (as per eq. 1). Meanwhile, the TSV utilization is minimized by inserting the components lying on one layer to same wrapper chain. The $TSVwc$ as calculated using eq. 5 is 8 for all the wrapper chains. It may be noted that TSVs that cater to the vertical interconnect between the scan chains which span on different layers have not been included in the total TSV utilization since they are a part of the core's functional design itself.

IV. PROPOSED TEST WRAPPER DESIGN

The complete task of insertion of the scan chains and functional input/output elements can be described as follows:

A. Scan Chain insertion and reordering

This step is very important as it helps in defining the approximate length of the various wrapper chains. Each scan chain *SC (In, Type, Len, Flag, Out)* is associated with four parameters: *In (*Input die), *Type* (for scan chains its value is equal to 3), *len (*to signify the length of scan chain), *flag* (set equal to *1* if both input and output dies are same else *0), Out*

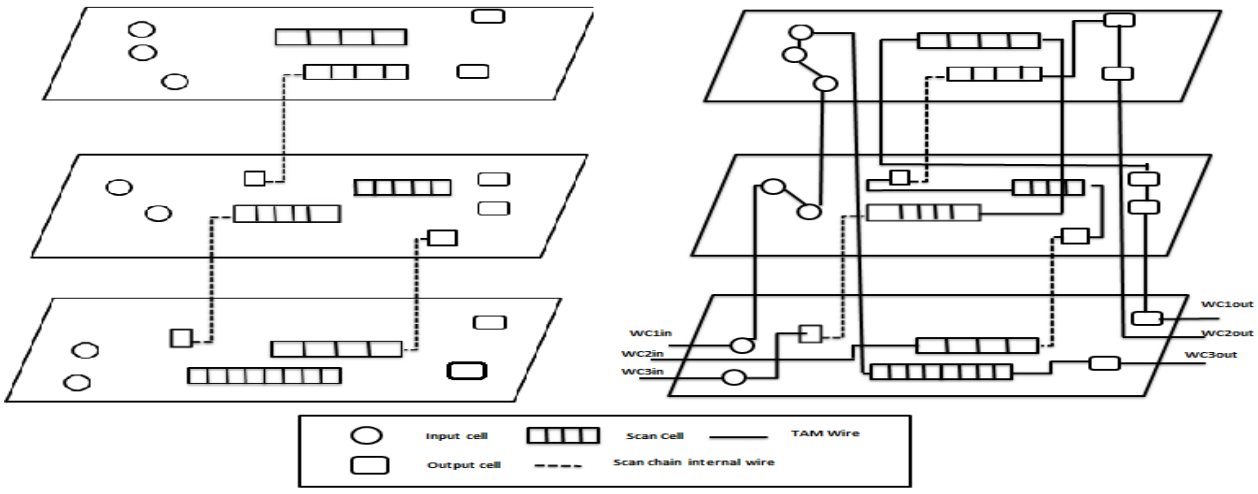

Fig. 1. (a) 3D model of a circuit partitioned IP core b) possible formation of the wrapper chain

(output die number). The pseudo code for the scan chain insertion is as shown in the FIG 2.

It initializes by arranging the internal scan chains in descending order of their lengths (shown in step 1). All the wrapper chains parameters like TSV_{usage}, scan chains in all the wrapper chains (as shown in step 2), die number of the input and output cells are initialized to be zero. The algorithm starts with an initial solution (as shown in step 3) by inserting the scan chains in the wrapper chains. The list of the scan chains is updated by deleting the scan chains which have been inserted in the wrapper chains.

B. Formation of Di Graphs and scan chain insertions

For the objective of the TSV minimization scan chains are placed in different wrapper chains such that the lengths of the wrapper chains are more or less balanced. This makes a heterogeneous mix of scan chains with same or different start and end layers. This is done by insertion of all the scan chains with flag bit equal to 1 in various wrapper chains followed by others as per lengths.

The TSV calculation is done as per given in the theorems 1 to 3. Once the scan chains insertion is decided, in between the wrapper chains reordering is done. For this the Di graphs are prepared which keep track of the TSV used in traversing between different scan chains as shown in FIG 3(a). The scan chains are shown as nodes with different parameters and the edges showing the possible connectivity between them. The numbers written adjacent to each edge reflects the number of TSVs used if one traverses from starting node of the edge to ending node. With 3 different layers, each element can have 1 out of 2^3 different possible combinations corresponding to their starting and ending layer. Initially, all the elements are sorted, counted for flag=1 and then distributed between the available number of wrapper chains. Thereafter similarly the scan chains with flag=0 are sorted and counted as per their I_SC.Core and O_SC.Core. Distribution into different bins is done so as to obtain a heterogeneous mix of the scan chains in the different wrapper chains. For Individual wrapper chains construct the diagraphs as shown in FIG 3 (b) with layers as the nodes and scan chains as the edges. here are edges between 1 to 3 for scan chains 1, 3, 6, and between 3 to 1 for Scan chain 8 etc. The connectivity of the scan chains should be done such that if a scan chain takes the wrapper chain from Node 1 to 2nd then the next scan chain should take it either back from layer 2 to 1 or from 2 to 3.

Pseudo code for scan chain insertion

```
Given Total scan chains SC and TAM wires Ctam
Step 1: /*initialization of scan chain list
For ( i =0; i<SC, i ++){
• Sort all the Scan chains in decreasing order of
  their length
• Initialize the Scan chain with their appropriate
  flag value;
  }
Step2: /*Initialize different wrapper chains
For (i=1 to Ctam) {
    WC.length ( i )= WC.TSVused( i )=
    WC.start( i )= WC.end( i )=0;
  }
Step3 /* Insert scan chain
    For (i=0; i<Ctam; i++){
    Insert SC (0) to WC (i)
    Update WC.length (i), WC.start (i),
           WC.end (i);
    Delete SC (0) from SC list;
  }
Step 4: /*while (SC! =0) {
    Find the WC with Smax
    Find WC with Smin
    ch_SC =Search (SC, WC [Ctam], Smax,
                   Smin)
    Insert ch_SC to WC and update all
        parameters of WC;
    Delete ch_SC from SC
}
Step 4: /* reordering of the scan chains in wrapper
chains
    For (i=1 to Wmax){
    Reorder all the Scan chains so as to
        minimize WC.TSVused;
    Update the WC parameters;
}
```

Pseudo code for selecting the scan chain for insertion

```
//Find a Scan chain such that its insertion
doesn't not increase the length of the
wrapper chain and TSV used by large
extent

Function Search (SC, WC[Ctam], Smax,
Smin){

If (length (Smax- (len.SC+Smin) !>Smax)){
  If (flag==1){
     If((O.WCmin=I.SC||
         I.WCmin==0.SC)
      Return SC;
  Else if ((O.WCmin! =I.SC && I.
          WCmin==0.SC){
       Find SC with min ( ( O.WCmin-I.S),
                         ( I. WCmin-0.SC))
       Return SC;
  }
  Else {
     SC with minimum length
     Return SC
     }
  }
```

Fig. 2. Pseudo code for scan chain insertion

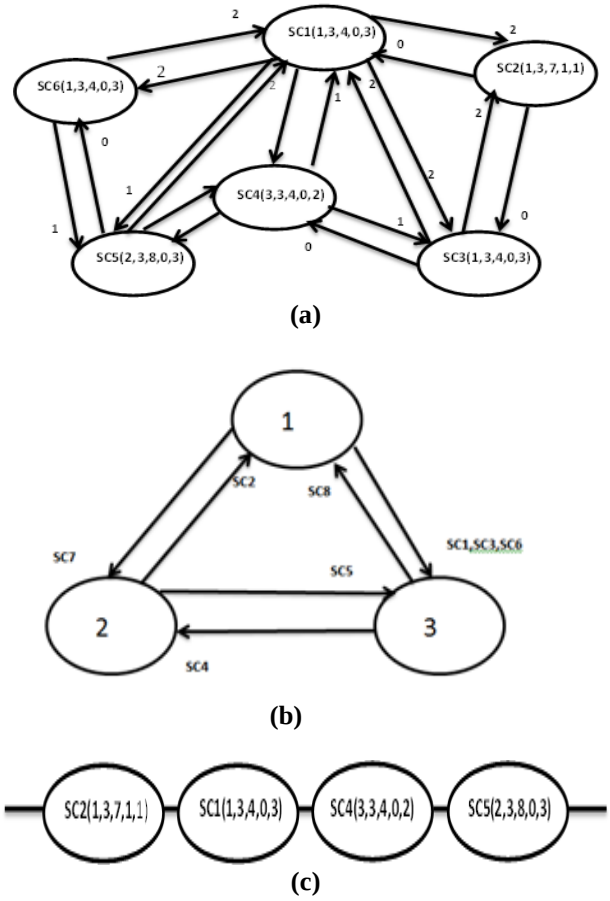

Fig. 3. a) Di-graph for various scan chains b) Diagraph to represent the scan chain and layer connectivity c) Possible stitching of the scan chains

Thereafter the insertion is done in the successive wrapper chains on the basis of the route with minimum TSV utilization in an order as shown in figure 3(c). Insertion of the functional input and output cells is done as described with the help of the pseudo algorithm shown in FIG 4 (a) and (b). Initially only those input and output elements are inserted that lie at the same layer as the starting layer of first scan chain and the output layer of the last scan chain of the individual wrapper chains respectively. Remaining I/O cells are tried to be added to the wrapper chains such that their lengths are balanced and TSV_{usage} does not exceed the maximum allowable limit. The benefit of starting the wrapper chains with different layers provide the advantage of adding as many input cells as possible that lie on the same layer.

Pseudo code for input cell insertion

// for adding functional input cells to the wrapper chains created in Part I
For all input cells {
  For (i=0; i< *Ctam*; i++) {
  - For each input cell, calculate $O.TSV(j)$ using equation 3.8
  - Insert input cells to *WC (i)* such that $O.TSV(j)$ is minimum and *Smax* is not exceeded.
  - Delete the used Output cells
  }

$WC.TSVused + = I.TSV(j) + |I(j,1) - number\ of\ the\ bottom\ die|$
Update length of each wrapper chain.
}

Fig. 4a. Pseudo code for input scan cell insertion

Pseudo code for output cell insertion

// for adding functional outputs to the wrapper chains created in Part I
For all output cells {
For (i=0; i< *Ctam*; i++) {
- For output cells, calculate $O.TSV(j)$ using equation 3.9
- Insert output cells to *WC (i)* such that $O.TSV(j)$ is minimum and *Smax* is not exceeded.
- Delete the used Output cells

}

$WC.TSVused\ (i) + = O.TSV(j) + |O(j,1) - number\ of\ the\ bottom\ die|$
Update length of each wrapper chain.
}

Fig. 4.b. output scan cell insertion

V. RESULTS AND ANALYSIS

The following section presents the results obtained for test wrapper design for fine grain partitioned SoC. To show the effectiveness of the proposed techniques, different ITC'02SoC benchmark circuits [18] have been utilized due to unavailability of benchmarks for 3D SoCs.Various SoC cores that have chosen for experimentation include: core7 of d281, core 4 and core13 of p93791. Out of the SoC benchmarks so selected for the experimentation d281 is developed by Duke University and has got 9 modules 2931 number of functional inputs and 882 functional outputs while p93791 and p22810 are developed by Philips research and Philips semiconductors. The detailed information of every benchmark is available in [18]. The algorithm is implemented in a high level language. Since the benchmark circuits are in 2D form so in order to make them compatible with the 3D circuit design, the placement of all the components of various cores is done using random enumeration. For every random assignment of the components, different results are obtained for test time and TSV utilization. Therefore, minimum of fifty iterations are utilized to normalize the results obtained. Tables 1 and 2 present the results as obtained for core 7 of d281. To show the effectiveness of the work, the comparisons have been made with the results presented in [18] and [19].

TABLE I.
TEST WRAPPER RESULTS FOR CORE 7 OF SOC D281 IN COMPARISON TO [18]

WC	TSV_{max}	Shortest Wrapper [18]	Proposed Shortest WC	TSV_{used} proposed	CPU Time (secs)
	12	NA	965	12	.02
2	14	NA	977	11	.05
	16	1633	967	10	.04
	18	1064	967	12	.07
	12	NA	592	11	.1
3	14	NA	624	12	.08
	16	1347	624	14	.09
	18	752	624	8	.04
	12	NA	500	12	.1
4	14	NA	492	13	.07
	16	1347	522	13	.1
	18	611	522	9	.12

5	12	NA	480	11	.02
	14	NA	456	14	.05
	16	1347	456	15	.01
	18	486	450	15	.08

TABLE II.
TEST WRAPPER RESULTS FOR CORE 7 OF SOC D281 IN COMPARISON TO [19]

WC	TSVmax	Longest Wrapper [19]	Proposed longest WC	TSVused [19]	TSV used proposed	CPU Time (in secs)
	12	1129	1163	10	12	.02
2	14	1223	1151	11	11	.05
	16	1223	1161	10	10	.04
	18	1223	1161	10	11	.07
	12	710	770	12	11	.1
3	14	710	755	14	12	.08
	16	710	755	14	14	.09
	18	816	755	16	8	.04
	12	532	532	12	12	.1
4	14	532	565	14	13	.07
	16	532	565	16	13	.1
	18	532	565	18	9	.12
5	12	426	430	12	12	.02
	14	426	430	14	14	.05
	16	426	432	16	15	.1
	18	426	432	18	15	.08

TABLE III.
TEST WRAPPER RESULTS FOR CORE 4 OF SOC P93791 IN COMPARISON TO [18]

WC	TSV_{max}	Shortest Wrapper [18]	Proposed Shortest WC	TSV_{used} proposed	CPU Time (secs)
	12	NA	68	10	.05
2	14	132	62	12	.05
	16	87	60	15	.07
	18	79	64	16	.04
	12	NA	43	10	.1
3	14	69	40	13	.08
	16	51	38	15	.09
	18	51	38	15	.04
	12	NA	35	11	.05
4	14	59	35	13	.07
	16	59	30	15	.1
	18	43	30	15	.12
5	12	NA	34	10	.04
	14	59	32	10	.03
	16	59	32	12	.08
	18	43	30	11	.09

TABLE IV.
TEST WRAPPER RESULTS FOR CORE 4 OF SOC P93791 IN COMPARISON TO [19].

WC	TSV_{max}	Longest Wrapper [19]	Proposed longest WC	TSV-used [19]	TSV used proposed	CPU Time (secs)
	12	85	85	11	10	.05
2	14	88	91	8	12	.05
	16	91	93	12	15	.04
	18	80	89	11	16	.07
	12	53	58	12	12	.02
3	14	53	59	13	14	.08
	16	52	58	14	15	.09
	18	73	56	15	15	.04
	12	40	44	12	11	.02
4	14	39	44	13	13	.07
	16	40	44	14	15	.1
	18	46	42	18	15	.12
5	12	31	42	11	12	.02
	14	33	40	12	14	.03
	16	32	40	14	14	.07
	18	36	38	15	11	.01

In all the tables presented above, column one represented the number of wrapper chains that have been formed as a part of the wrapper design. Column 2 represents the maximum limit for TSVs that can be utilized in each wrapper design. It may be concluded that where the algorithm in [18] is unable to form the wrapper chains (as shown in few of the cases) on being bounded by the TSV constraint, the solution can be obtained using the proposed algorithm. As shown in the tables 1- 4, the results obtained seem to be quite motivating for the adoption of the proposed approach. It may be noted that the values of the TSV used and lengths of the wrapper chains can vary when used for practical scenario wherein actual placement of individual components is given.

VI. CONCLUSION

3D technology has made the implementation of complex SoC using much reduced footprint. The vertical interconnects have led to the reduced delay and power issues which were posed by long interconnects. Different fabrication technology requires the test infrastructure design to be optimized with additional constraints on top of what existed for 2D counterparts. A method is proposed in this paper to guide the designer to derive an optimal test wrapper design for fine grain partitioned 3D SoCs using much lesser TSVs. Experimental results show that CPU time utilization for heavy SoCs is also reduced compared to the other proposed techniques.

REFERENCES

1 R.R Tummala, V.K. Madisetti, "System on chip or system on package?," IEEE Des. Test Comput., vol. 16, no. 2, pp. 48-56, Apr. 1999.

2 Davis, A. Jeffrey, J. Shukri, C. Krishna Saraswat, "Interconnect limits on gigascale integration (GSI) in the 21st century." P. IEEE, vol. 89, no. 3, pp. 305-324, Mar. 2001.

3 K. Banerjee, S.J. Souri, P. Kapur, and K.C. Saraswat, "3-D ICs: A novel chip design for improving deep-submicrometer interconnect performance and systems-on-chip integration," P. IEEE, vol. 89, no.5, pp.602-633, May 2001.

4 P. Ramm, A. Klumpp, J. Weber, "3D System-on-Chip technologies for More than Moore system," Microsys. Tec., vol. 16, no. 7, pp. 1051-1055, Jul. 2010.

5 Y. Xie, G.H. Loh, B. Black and K. Bernstein, "Design space exploration for 3D architectures," J. Emer. Tech. Comput. Sys. (JETC), vol. 2, no. 2, pp. 65-103, Apr. 2006.

6 "International technology roadmap for semiconductors," Available: http://www.itrs.net/links/2009ITRS/Home2009.htm, Jul. 13.

7 T.B. Wu, H.Z. Liu and P.-X. Liu, "Efficient test compression technique for SoC based on block merging and eight coding," J. Electro. Test., vol. 29, no. 6, pp. 849-859, Dec. 2013.

8 H. Vohra and A. Singh, "Optimal Selective count compatible Run length Encoding for SOC Test data compression," J. Electro. Test., vol. 32, no. 6, pp. 735-747, Dec. 2016

9 A. Sehgal, V. Iyengar and K. Chakrabarty, "SOC test planning using virtual test access architectures," IEEE Trans. Very Large Scale Inte. (VLSI) Systems, vol. 12, no. 12, pp. 1263-1276, Dec. 2004.

10 V. Iyengar, K. Chakrabarty, and E. Marinissen, "Efficient wrapper/TAM co-optimization for large SOCs," IEEE Comp. Soc. Confer. Design, Auto. Test. Europe, p. 491, 2002.
11 E. J. Marinissen, S. K. Goel, and M. Lousberg, "Wrapper design for embedded core test," IEEE Int. Test conf., Aug. 2002, pp. 911-920.
12 T Mak., "Test Challenges for 3D Circuits," in IEEE Int. Sym. on On-Line Test. (IOLTS'06), Jul. 2006, pp. 79-85 [Dig. 12th Int. Conf. Sym. Italy, 2006].
13 H.-H. Lee and K. Chakrabarty, "Test challenges for 3D integrated circuits," IEEE Trans. Des. & Test Comp., vol. 26, no. 5, pp. 26-35, Jan. 2009.
14 K. Shen, D. Xiang, and Z. Jiang, "Dual-speed tam optimization of 3d socs for mid-bond and post-bond testing", Asian Test Sym. (ATS), Nov. 2014, pp. 7-12 [Dig. 23 IEEE conf. China,2014].
15 B. Sen Gupta, U. Ingelsson, E. Larsson, "Scheduling tests for 3D stacked chips under power constraints.," J. elec. Test. , vol. 28, no. 1, pp. 121-135, Feb. 2012.
16 B. Noia,K. Chakrabarty and E. J. Marinissen, "Optimization methods for post-bond testing of 3D stacked ICs," J.Elec.Test., vol. 28, no. 1, pp. 103-120, Feb. 2012.
17 A. Sehgal, S.K Goel and K. Chakrabarty, "Testing of SOCs with hierarchical cores: common fallacies, test access optimization, and test scheduling," IEEE Trans. Computers, vol. 41, no. 10, pp. 409-423, Sep. 2009.
18 Brandon Noia and Y. Xie, "Test-wrapper optimization for embedded cores in TSV-based three-dimensional SOCs test-wrapper optimization for embedded cores in TSV-based three-dimensional SOCs", IEEE Comput. Des. Oct. 2009, pp. 70-78 [Dig. 27th Int. Conf. USA, 2009].
19 S. K. Roy, S. Ghosh, H. Rahaman, and C. Giri, "Test wrapper designfor 3d system-on-chip using optimized number of TSVs", in IEEE International Symposium on Electronic System Design (ISED), pp.197-202, 2010.

AUTHORS PROFILE

Dr. Harpreet Vohra has a rich experience of teaching Electronics and VLSI courses. She joined Thapar Institute of Engineering & Technology, Patiala as Assistant professor in Electronics and Communication Engineering Department. She has more than 12 years of experience in teaching and research both at undergraduate and post graduate levels She received her Ph.D. in test solution development for System on chip. She has many research publications in reputed peer reviewed journals and conferences. Her areas of interest include machine learning and its applications in the domain of VLSI test.

Dr. Ashima Singh is Assistant Professor in Computer Science and Engineering Department, at Thapar Institute of Engineering & Technology, Patiala since 2006. She has earned Ph.D. in Computer Engineering and holds a Master Degree with distinction in Computer Science. She has more than twelve years of experience in teaching and research both at UG/PG levels. She Has guided many Theses in M.E. Software Engineering and M.E. Computer Science and Engineering. She has more than 45 research publications in reputed peer reviewed journals and conferences. She is a professional member IEEE and Branch Counsellor, IEEE Student Chapter at TIET, Patiala, Punjab. Her research interest includes machine learning and data analytics, computational bioinformatics, cloud computing and IoT applications