Design Methodology using Congestion-Driven Placement Algorithms for Three-Dimensional ASIC Designs

ABSTRACT

We have developed an automated design flow and associated CAD tools for three-dimensional (3D) ASIC designs. The embedded two-phase congestion-driven 3D placement enables this design flow, leveraging the benefits of the 3D technology to general large-scale interconnect-complex applications. We have implemented our placement algorithms on the ISPD98 circuit benchmark suite. Furthermore, we have applied our 3D design methodology on an interconnect-congested fully parallel Low Density Parity Check (LDPC) decoder. This ASIC chip design was the first project attempted using 3D technology on the scale of millions of gate-level components. This resulted in a thorough evaluation of our design methodology and also displayed the significance of our work.

Keywords

Congestion, 3-Dimensional ASIC Design, LDPC Decoder

1. INTRODUCTION

As transistor feature size continues to shrink, and integration densities and chip areas continue to rise, the interconnection lengths and delays become critical parameters in determining system performance. Three-dimensional (3D) integration technology has been proposed as an alternative to address the interconnection issues [1,9-12] with its potential to reduce wire length and overall congestion. Generally, 3D technology could be described as stacking several integrated circuits vertically with third dimensional interconnects, called 3D-vias. A cross-section of three tiers that form a 3D integrated circuit is shown in Figure 1.

Significant challenges associated with efficient circuit design methodology and tools for 3D integration have hampered the further development of this technology. Although several regular circuit structures such as memories, FPGAs and imagers have been successfully explored in 3D technology, high demand exists for efficient 3D design methodologies and CAD tools that can apply 3D technology to larger ASIC design spaces.

Conference'04, Month 1–2, 2004, City, State, Country.

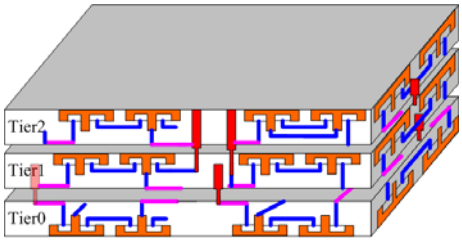

Figure 1. Cross-section of three tiers integrated to form a 3D circuit.

Recent work on design tools for 3D integrated circuits has included the following: Das et. al. implemented a 3D standard-cell placement and global routing tool [2]. The placement algorithm was based on a recursive min-cut partitioning of the circuit represented as a hypergraph. Inter-tier via (3D-via) minimization was sought by min-cut partitioning for tier assignment. During the partitioning phase, the aspect ratio was used to determine the wire-length minimization. Cong et. al. proposed thermal-driven 3D floor-planning and routing algorithms [3,14]. Boplen and Sapatnekar presented a thermal-driven force directed standard-cell placement and a thermal via placement method based on finite element analysis [4,15].

We propose a 3D methodology, with supporting CAD tools, for the physical design phase of ASICs. One of the most important steps in our methodology is the two-phase placement of components and 3D-vias. An efficient probabilistic grid model is used to predict the routing density within the placement process. We report a series of benchmark experiments, comparing the wire length and routing density distributions for placements optimized using various objectives. In particular, to evaluate proposed flow and algorithms in real circuit design, we use our 3D design flow and CAD tools to design and tape out a 3D fully parallel Low-Density Parity-Check (LDPC) decoder, which has severe interconnect-congestion issues with current 2D ASIC designs [7].

The organization of the paper is as follows. Section 2 first introduces our 3D ASIC design methodology. Section 3 gives details of our two-phase congestion-driven 3D placement algorithm with a congestion analysis model used in the placement algorithm. Experimental results of benchmark circuits are presented in section 4. In section 5, an LDPC is physically

realized using our design flow and CAD tools. The conclusion follows in section 6.

2. DESIGN FLOW FOR 3D ASIC

A typical 2D design flow of a digital integrated circuit is shown in Figure 2. This 2D ASIC design flow has been widely accepted by designers, and its corresponding CAD tools have been well developed commercially. In our approach to 3D design, we desired to utilize as much of this well-established 2D design flow as possible. Hence, we tailored the respective components of the design flow to 3D integrated circuits.

Due to the existence of several tiers and the alignment constraints of 3D-vias on neighboring tiers, new 3D placement and global routing tools were developed based on our "standard-macro" floor-planning scheme. In this scheme, general ASIC designs on the scale of millions of gate-level components are first hierarchically partitioned into thousands of macros of similar size by the Min-CUT partition algorithm [2,13]. Each macro contains hundreds of standard-cells, with the connections between macros minimized by the Min-CUT algorithm. Then commercial synthesis tools were used to make "standard-macros," where each macro was configured to the same height, with varying widths. Using the same strategy in 2D ASIC design, we placed the standard-macros in rows. Between standard-macro rows, we reserved sufficient space for 3D-vias and buffer banks. With this scheme, 3D-via alignment issues were avoided. And also with this standard-macro design scheme, the very large-scale interconnect-complex 3D design is simplified to a standard 2D ASIC design case.

In the next section, a two-phase congestion-driven 3D cell and 3D-via placement algorithm is further developed to optimize one or more of design objectives, which could exploit the benefits of 3D technology to solve the interconnect issues and provide high quality designer-specific solutions.

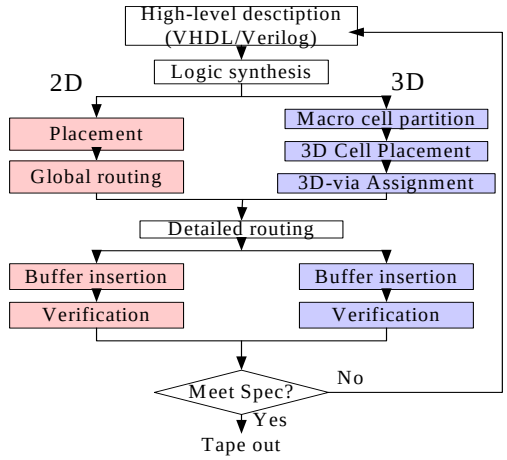

Figure 2. Simplified flowchart for the automated design of 2D and 3D ASIC.

Besides the placement step, other phases are also modified to accommodate the 3D design flow. After 3D cell and 3D-via placement is complete, the 3D design is converted to several 2D designs. Commercial detailed routing tools are used for routing on each tier. A 3D buffer insertion program was developed to handle long interconnects. As a final check, a verification program was designed to check the final physical design against the original netlist to ensure consistency.

3. 3D PLACEMENT ALGORITHM

3.1 Description of the 3D Placement Problem

The main goal of this section is to describe a fast framework for the 3D macro and 3D-via placement. A common practice for cell placement algorithms has been to minimize the total wire length of the chip. For interconnect complex designs, minimizing only total length may introduce routing problem and also timing problems. Thus, our objectives are to minimize the total wire length, the longest wire, 3D-via usage, and routing density. 3D-vias are costly due to resistance, capacitance, size and fabrication cost. Large routing densities consume large area and degrade performance due to coupling and uneven heat dissipation.

An objective function considering all the objectives mentioned above is described in equation (1).

$$F = F_{length} + F_{3D-via} + F_{density} \quad (1)$$

In the following section, the use of congestion analysis to estimate $F_{density}$ is discussed.

3.2 Probabilistic Congestion Analysis Model and Congestion Estimation

In equation (1), the $F_{density}$ is the most complex term to calculate. Therefore, a very fast and efficient routing density estimation method is required in our placement algorithm to reduce the routing density while placing the macros and 3D-vias.

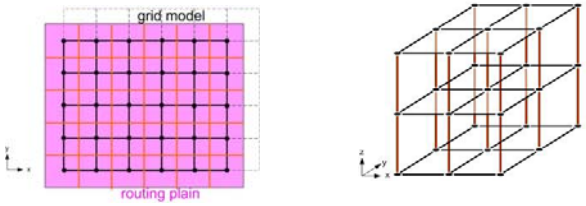

Figure 3. Congestion analysis grid models. (a) 2D grid model. (b) 3D grid model.

The routing problem is normally modeled as a graph $G(V,E)$, where edge e_{ij} connects vertices v_i and v_j. The edge e_{ij} has a capacity interval of $[0, c_{ij}]$. We must form Steiner trees to connect subsets of vertices, which satisfy capacity constraints and minimize total tree length. Normally, rectangular regions are defined over the circuit area, and the vertices of the graph define these regions. The borders between regions are represented by edges. Figure 3a shows the grid model used for congestion analysis on each 2D tier.

Similar 2D probabilistic models are proposed by Lou et al. [5] and Cheng et al. [6]. In [5], it is assumed that each net uses the shortest route and each possible route has the same usage probability. In [6], limited detouring is also considered. We will not consider detouring in our model. The inclusion of detouring at this stage is not considered to be particularly effective due to limited gains.

Extending this scheme to the 3D probabilistic model, the congestion problem is modeled by placing all reference points (pin locations) into a 3D-grid G of N_x by N_y by N_z points. Each point is treated as the source and the destination of all edges. This model is sketched in Figure 3b. The vertical links in the 3D grid model represent 3D-vias. The routing density of a region is defined as the estimated number of wires crossing a link representing the routing region. The 3D-via density is defined as the estimated number of 3D-vias on each vertical link.

The problem for the density estimation can be described as follows: calculate the usage probability $P(l_i)$ of each link l_i, $1 \le i \le K$, given a 3D placement grid G and a set of two-terminal nets N_i, $1 \le i \le m$, where m is a total number of nets. Multi-terminal nets can be decomposed into a set of two-terminal nets.

For each net connecting (x_s, y_s, z_s) and (x_d, y_d, z_d), where s and d represent source and destination respectively, we define a routing region R from (0, 0, 0) to (a, b, c), where $(a, b, c) = (|x_d - x_s|, |y_d - y_s|, |z_d - z_s|)$. For each net, we are to compute the usage probability $P_{Ni}(l_i)$ of all the links in the entire region R. $P_{Ni}(l_i)$ is computed as follows:

$$P_{N_i}(l_i) = \frac{Number\ of\ routes\ in\ R\ that\ use\ the\ link\ l_i}{Total\ number\ of\ routes\ in\ R} \tag{2}$$

In defining a route, we make the following assumptions. (1) All nets are routed along the links. (2) All nets are routed with shortest lengths. (3) All possible routes for each net have equal usage probability independent of the number of direction changes (from horizontal link to vertical link and vice versa) the net makes.

The expected usage or the routing density $P(l_i)$ for each link l_i, $1 \le i \le K$, in the 3D placement grid is obtained by summing corresponding densities contributed by all the nets. It is calculated as equation (3)

$$P(l_i) = \sum_{\substack{all\ nets, N_j \\ 1 \le j \le m}} P_{N_j}(l_i) \tag{3}$$

Our density estimation method is different from [5] and [6]. Instead of using combinatory computation, a simple incremental procedure to compute the number of routes and densities is implemented. Our algorithm consists of three steps:

(1) Computing route matrix for each net: Route matrix stores the number of possible routes to reach one node from adjacent x, y and z nodes.

(2) Computing 3D density matrix for each net: Density matrix stores x, y and z densities associated with each node. The density computation procedure is demonstrated in Figure 4. The x, y and z densities associated with each node is computed as Eqs. (4-6).

$$Density[x][y][z].x = total\ density \times \frac{route[x+1][y][z]}{route[x][y][z]} \tag{4}$$

$$Density[x][y][z].y = total\ density \times \frac{route[x][y+1][z]}{route[x][y][z]} \tag{5}$$

$$Density[x][y][z].z = total\ density \times \frac{route[x][y][z+1]}{route[x][y][z]} \tag{6}$$

The total density is the summation of the x, y, and z densities entering the node and it is given in Equation (7). Note that the total entering density of the starting node is set to 1. Each density value is equivalent to $P_{Nj}(l_i)$ given in Equation (3).

$$total\ density(x,y,z) = \begin{cases} density[x-1][y][z].x + \\ density[x][y-1][z].y + \\ density[x][y][z-1].z \end{cases} \tag{7}$$

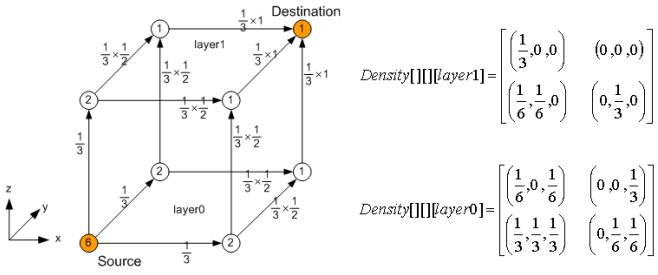

Figure 4. Density matrix computation.

(3) Computing 3D global density matrix for all nets: The global density matrix stores the total x, y and z densities associated with each node by adding the corresponding entries of the density matrixes defined by all the nets as described in Equation (3). Each global density value represents the estimated routing density of each link for a given placement.

3.3 Two-Phase Congestion-Driven 3D Cell Placement and 3D-Via Assignment Algorithm

We present a congestion-driven simulated annealing-based 3D placement algorithm, which uses the congestion estimation addressed above in a subroutine. In the first phase of the algorithm, the macros are placed on different tiers without considering the 3D-vias. In the second phase, the 3D-vias are incorporated into the results of the first phase.

3.3.1 Phase 1: Macro-cell Placement

We use simulated annealing (SA) techniques [8] in our placement algorithm to balance the routing densities while minimizing wire length and 3D-via usage. In the phase 1 of the placement, annealing is performed following a three-step procedure:

Step 0: We begin by randomly placing cells on a 3D grid G. Initial cost value is computed by summing the first two terms in the objective function, wire length and 3D-via costs.

Step 1: We anneal from a random cell placement configuration (i.e., temperature=∞). The move set consists of swapping the contents of two locations. The cost value is updated incrementally at each iteration. Since density computation is computationally expensive, we optimize using only wire length objective and 3D-via objective at this step. We note that minimizing total wire length also reduces the total wire density of the channels. However, it may not achieve the balanced density distribution.

Step 2: The temperature=0 configuration obtained in step (2) is again annealed, accepting only improving moves, using density objective while maintaining all the other objectives.

3.3.2 Phase 2: 3D-via Assignment

After phase 1 procedure is completed, the placement of macro-cells is fixed. The 3D-via assignment problem is to find an optimal location for each 3D-via, which minimizes the routing

density on each tier and generates even distributions of 3D-vias between tiers. This means we seek the best possible uniform distributions of wires as well as 3D-vias for a given placement. An objective function is the density objective as Equation (8).

$$F_{density} = \sum_{n=1}^{N_k} \left(W_{dx}.D_x(n) + W_{dy}.D_y(n) + W_{dz}.D_z(n) \right) \quad (8)$$

Where $D_x(n)$, $D_y(n)$ and $D_z(n)$ are x, y and z densities associated with each node n, W_{dx}, W_{dy} and W_{dz} are the weighting factors for x, y and z densities.

We define a 3D-net as a net connecting two cells on different tiers. For each 3D-net connecting (x_s, y_s, z_s) and (x_d, y_d, z_d), where s and d represent source and destination and $z_s \neq z_d$, we define a rectangular 3D-via insertion region V bounded by $(x_{min}, y_{min}, z_{min})$ and $(x_{max}, y_{max}, z_{max})$, where $\{x_{min}, y_{min}, z_{min}\} = \{min(x_s, x_d), min(y_s, y_d), min(z_s, z_d)\}$ and $\{x_{max}, y_{max}, z_{max}\} = \{max(x_s, x_d), max(x_s, x_d), max(x_s, x_d)\}$. The pseudo-codes for 3D-via insertion algorithm are given in Figure 5.

```
Via Insertion Algorithm
Input: 3D placement
Output: 3D-via locations
-------------------------------------------------------------------------
Initialization();
    Assign initial placement for each 3-D via.  Generate a Via_List.
    oldcost = Fdensity();
while ( i < iterations) {
   Choose a 3D-via randomly from Via_List.
   Relocate within the 3D-via insertion region Vi.
   newcost = Fdensity();
   if ( newcost < oldcost ) {
       oldcost = newcost;
       update_Via_List();
   }
   else  restore_Via_List();
}
```

Figure 5. Psedo-codes for 3D-via assignment algorithm.

Simulated annealing technique is used with the temperature fixed at zero. Initially each 3D-via is assigned a random location within its region Vi and a data structure, Via_List, is created to store via information. Initial objective function value is calculated as in Equation (8). A via is randomly chosen and assigned a different location within its via insertion region V_i. The cost value is updated by incremental computation. Only improving moves are accepted.

4. EXPERIMENTAL RESULTS

To evaluate the effectiveness of our algorithms, we test the placement algorithm on ISPD98 benchmark circuits, prioritizing different objectives by assigning different penalty weights. We perform the optimization in three configurations: in the first we consider total wire length, in the second we consider both wire length and 3D-via usage and in the third, we optimize for wire length, 3D-via usage and routing density. The best tradeoff is achieved between wire length, routing densities and the number of 3D-vias required. A summary of our experimental results for the resulting wire lengths and routing densities is shown in Table 1 and Table 2. In the tables, the following terminology is used: objective 1 represents total length objective, objectives 2, 3 and 4 represent maximum length, 3D-via count and maximum density objectives respectively. For ibm01, ibm05 and ibm18 benchmark circuits, we first use the MIN-CUT partition [13] to cut the circuits into 200, 500 and 1000 macros, respectively. Due to the MIN-CUT partition, the global nets between these "standard-macros" are minimized as listed. And then 8×9×3, 12×14×3 and 18×19×3 congestion analysis grid models are used for each case respectively. These grid sizes are adjusted corresponding to the macro numbers. From the statistical results, it is observed that optimizing total wire length alone results in a number of longer wires, although total length is minimal. Comparable wire length distributions are observed from optimizations with and without the density objective. Also by including the 3D-via count as one optimization objective, the number of 3D-vias is reduced by 65%-79%. Finally, after the density objective is considered, a 13%-22% reduction in maximum routing density is achieved while maintaining all other performance parameters constant.

Table 1. Wire length statistics of 3D placements with different optimization objectives

Circuits	no.of macros	no.of global nets (multi-pin)	2-pin nets after decomposition	Obj (1)	Obj (1,2,3)	Obj (1,2,3,4)
ibm01	200	3361	total length	13497	18729	18916
			avg length	1.78	2.46	2.49
			max length	12	8	9
			3D-via count	4122	872	872
ibm05	500	7327	total length	50530	62203	62794
			avg length	3.42	4.21	4.25
			max length	22	18	20
			3D-via count	13845	4959	4959
ibm18	1000	42985	total length	315957	431211	433905
			avg length	2.34	3.2	3.22
			max length	28	20	23
			3D-via count	80670	24135	24140

Table 2. Wire length statistics of 3D placements with different optimization objectives

Circuits		Obj(1)	Obj (1,2,3)	Obj(1,2,3,4)
ibm01	max density	99.90	97.44	79.11
	avg density	35.43	49.16	49.65
ibm05	max density	190.82	193.34	164.24
	avg density	54.33	66.88	67.52
ibm18	max density	436.10	510.96	381.33
	avg density	162.78	222.16	223.55

5. DESIGN CASE: 3D 1024-BIT 1/2–RATE FULLY PARALLEL LDPC DECODER

To evaluate our design flow and algorithms by a million-gate level design case, we present a particularly interconnect complex design case utilizing a Low-Density Parity-Check (LDPC) code decoder. The LDPC block-parallel message passing decoding algorithm and its fully-parallel implementation architecture yield the high-throughput error-correction capacity necessary for large-volume communication and data storage applications. However, this implementation leads to design challenges since the area used by the long interconnects rivals the space utilized by the logic due to routing congestion [7].

To address this interconnect design challenge, we explore the use of MIT Lincoln Lab's 3D process. This process stacks three wafers; each composing of a single layer of transistors with three

layers of metal wires (called one-tier) formed on fully-depleted silicon-on-insulator (FDSOI) substrates [9].

The fully parallel architecture and 512 x 1024 H matrix from [7] was utilized in our design. We first partition the LDPC decoder's 1,536 macros into two types of nodes: check nodes and variable nodes. Inside each variable node or check node, hundreds of standard-cells are used to realize the complicated operations. There are more than 25,000 signal global nets between the macros. The placement optimization is performed on the 3D structure by our two-phase congestion-driven 3D cell and 3D-via placement algorithm. After placing the 3D-vias, we perform routing on individual tiers using Cadence SE. Then the in-house buffer insertion and verification tools are applied. The layout of final three-tier design, connected by the densely distributed 10,631 3D-vias, is shown in Figure 7a. And the BER vs. SNR and iteration convergence vs. SNR simulation result of our decoder are shown in Figure 7b.

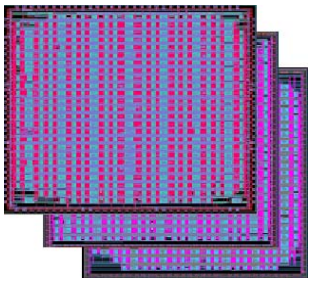

Figure7a. Final layout of 3D LDPC structure.

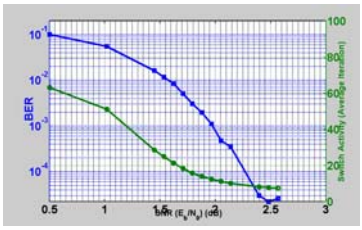

Figure7b. The simulated LDPC decoder performance.

To demonstrate the accuracy of our density estimation method for predicting routing congestion and placement, we compare the routing density maps from a commercial routing tool for the placement present in paper [7] in Figure 13. It is clearly seen that routing congestion is reduced with density optimization. And also our wire length and density estimation results are correlated well with the results of routing by commercial routing tools.

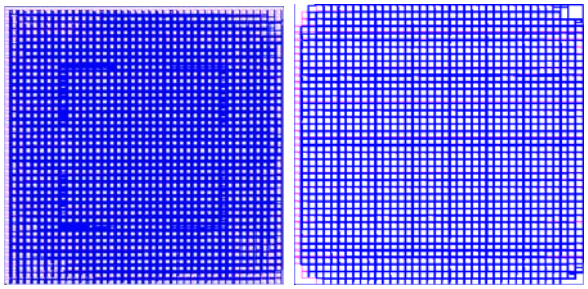

Figure 8. Density map of routing result by SE for the placement in [10] and our congestion-driven optimized placement.

Statistical analysis for the placement results of 3D architectures was performed using different optimization objectives. The wire length and density histograms are shown in Figures 9. In the figures, the following terminology is used: objective 1 represents total length objective, objectives 2, 3 and 4 represent maximum length, 3D-via count and maximum density objectives, respectively. Similar results as shown in ISPD benchmark circuits are achieved. Optimizing total wire length alone results in a number of longer wires, although total length is minimal. The longest wire length of 32 is more than double the results obtained from other optimizations, which consider maximum lengths. Comparable wire length distributions are observed from optimizations with and without the density objective. However, the density objective achieves a 22% reduction in maximum routing density.

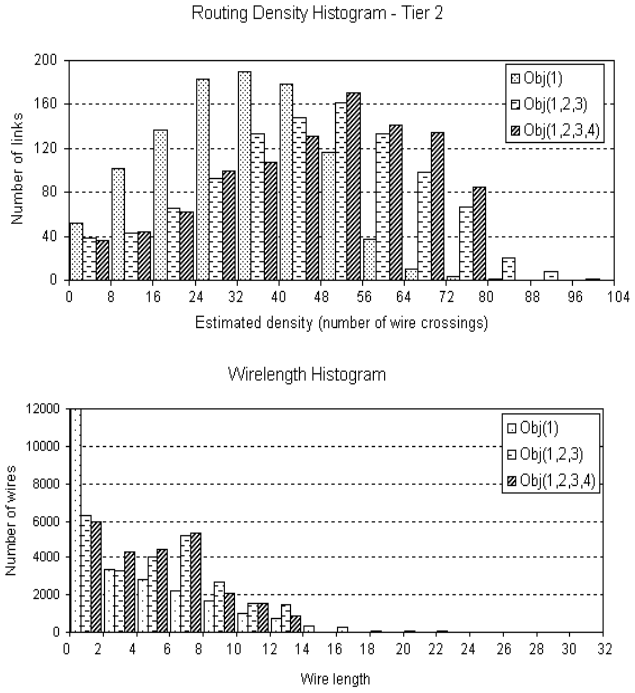

Figure 9. Routing density and Wire length comparison histograms for 3D placements.

To further demonstrate the advantage of the 3D technology over the 2D on this interconnect-complex application, Table 3 summarizes the design characteristics of the physical implementation, and a comparison to a 2D design. The 2D design was accomplished by putting all devices on one tier with the same technology and the same number of standard cells as 3D technology. We can see that the 3D implementation achieved a significant advantage over the same 2D implementation in terms of wire length, area, clock skew and buffer sizing. The improvement in terms of area-delay-power product is about an order of magnitude ($2.5 \times 3.0 \times 1.75 = 13.125$).

Table 3. The comparison between 3D and 2D designs.

	2D design	3D design
area (mm*mm)	18.238*15.92=**290.35**	(6.4*6.227)*3= **119.56**
total wire length (m)	**182.42**	22.39+22.57+22.46=**67.42**
max. WL before buffer insertion (mm)	**13.82**	**8.68**
max. WL after buffer insertion (mm)	4	4.17
buffer used	**32900**	**24636**
clock skew (ns)	**2.33**	**1**
power dissipation(mw)	**750**	**430**

6. CONCLUSIONS

We have explored the benefits of 3D technology in a large ASIC design space. Our contribution to 3D technology includes: (1) developing a design methodology and supporting CAD tools for use in real, large-scale ASIC designs; (2) a simple and efficient 3D congestion analysis procedure and the inclusion of congestion metrics in the cost functions of placement; (3) a congestion-driven simulated annealing based 3D placement algorithm, which allows us to target multiple design objectives to provide quality designer-specific results; (4) the first ASIC chip design project produced using 3D technology on the scale of millions of gate level components. And it was shown to yield an order of magnitude improvement over the corresponding 2D process, in terms of power-delay-area product. Future work will address thermal and reliability issues in 3D design.

7. REFERENCES

[1] K. Banerjee, S. Souri, P. Kapur, and K. Saraswat, “3-D ICs: A Novel Chip Design for Improving Deep-Submicro- meter Interconnect Performance and Systems-on-Chip Integration,” Proceedings of the IEEE, vol. 89, pp. 602-633, May 2001.

[2] S. Das, A. Chandrakasan, and R. Reif, “Design Tools for 3D Integrated Circuits”, Proceedings of the Asian and Sout Pacific Design Automation Conference, pp. 53-56, Jan 2003.

[3] J. Cong, J. Wei, and Y. Zhang, “A Thermal-Driven Floorplanning Algorithm for 3D ICs,” Proceedings of International Conference on Computer-Aided Design,” pp. 306-313, Nov 2004.

[4] B. Goplen and S. Sapatnekar, “Efficient Thermal Placement of Standard Cells in 3D ICs Using a Force Directed Approach,” International Conference on Computer-Aided Design, pp. 86-89, Nov 2003.

[5] J. Lou, S. Thakur, S. Krishnamoorthy, and H. S. Sheng. “Estimating Routing Congestion Using Probabilistic Analysis”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 21, pp. 32-41, Jan 2002.

[6] L. Cheng, W. N. N. Hung, G. Yang, and X. Song, “Congestion Estimation for 3-D Circuit Architectures”, IEEE Trans. Circuits and Sysstems-II: Express Briefs, vol. 51, pp. 655-659, Dec 2004.

[7] A. Blanksby, and C. J. Howland, “A 690-mW 1-Gb/s 1024-b, Rate ½ Low-Density Parity-Check Code Decoder,” IEEE Journal of Solid State Circuits, vol. 37, no. 3, pp. 404-412, Mar 2002.

[8] C. Sechen, VLSI Placement and Global Routing Using Simulated Annealing, Kluwer Academic Publishers, Boston, 1988.

[9] J. Burns, L. McIlrath, J. Hopwood, C. Keast, D. P. Vu, K.Warner, and P. Wyatt, "An soi-based three dimensional integrated circuit technology", IEEE International SOI Conference, pp. 20 - 21, Oct. 2000.

[10] K. W. Lee, et. al., "Three-dimensional shared memory fabricated using wafer stacking technology", Technical Digest of the International Electron Devices Meeting, pp. 165 - 168, 2000.

[11] J. Burns et. al., “Three-dimensional integrated circuits for low-power, high bandwidth systems on a chip”, ISSCC, 2001.

[12] W. Meleis, M. Leeser, P. Zavracky, and M. Vai, “Architectural design of a three dimensional FPGA”, Proceedings of the 17th Conference on Advanced Research in VLSI (ARVLSI), pages 256–268, September 1997.

[13] George Karypis, Rajat Aggarwal, Vipin Kumar, and Shashi Shekhar.Hmetis, “Multilevel Hypergraph Partitioning: Applications in VLSI Domain.” [KAKS97].

[14] J. Cong, and Y. Zhang, "Thermal-Driven Multilevel Routing for 3-D ICs," Proceedings of the Asia South Pacific Design Automation Conference, Jan 2005.

[15] B. Goplen and S. Sapatnekar, “Thermal Via Placement in 3D ICs,” Poceedings of the ACM International Symposium on Physical Design, pp. 167-174, 2005.