International Journal for Modern Trends in Science and Technology, 8(S09): 25-28. 2022

ISSN: 2455-3778 online
DOI: https://doi.org/10.46501/IJMTST08S0906
Available online at: https://www.ijmtst.com/vol8si09.html

IC Placement with Awareness using white space and wirelength

Radeep Krishna R[1] | Sreedivya K M[2] | Abraham C G [3] | Karthikumar S [4]

[1,3]Department of ECE, Mangalam College of Engineering, Ettumanoor, Kerala, India
[4]Department of ECE, Mangalam College of Engineering, Ettumanoor, Kerala, India
[2]Department of EEE, Carmel College of Engineering and Technology, Alappuzha, Kerala, India

To Cite this Article
Radeep Krishna R, Sreedivya K M, Abraham C G and Karthikumar S. IC Placement with Awareness using white space and wirelength. International Journal for Modern Trends in Science and Technology 2022, 8(S09), pp. 25-28. https://doi.org/10.46501/IJMTST08S0906

Article Info
Received: 26 May 2022; Accepted: 24 June 2022; Published: 30 June 2022.

ABSTRACT

When numerous layers of a 3D IC are developed, the most adaptive method for routability is the allotment of white space. In order to reduce density and improve wire length, white space allocation is used in the block placement zones of ICs. The hybrid technique is provided to handle legalization and minimization in order to resolve wire length and routability for various IBM-PLACE benchmark circuits.

KEYWORDS: ***Routability, white space allocation, wire-length, IBM-PLACE benchmarks.***

1. INTRODUCTION

Three-dimensional integrated circuits (ICs) have emerged as a feasible solution for addressing electronic device requirements such as greater performance, extended functionality, low power consumption, and shorter range. TSVs (Through Silicon Vias) are positioned between the blocks of each layer in a 3D IC to interconnect the stacked layers. TSVs are arranged in vertically stacked layers utilizing White Space Allocation (WSA) within the layers. This white space allocation in 3D IC was utilized for both placement and routing. Routability is a crucial placement stage. If there is a crowded region in an Integrated Circuit (IC), the WSA plays a role in reducing that region's congestion.The WSA consists of empty space in which any block may be inserted. The WSA is obtained by rearranging the benchmark's blocks. Because of the aspect ratio, the block arrangement of a standard die benchmark differs from that of a Mixed/Macro die benchmark. The aspect ratio is the proportion between the length and width of each block in a die. In contrast to mixed/macro die benchmark circuits, the standard die benchmark circuits have the same aspect ratio. White space allocation for a typical die benchmark is a difficult task.The most significant limitation of WSA is the cable length overhead. This can all be optimized using the Half Perimeter Wirelength (HPWL) method. For instance, the approximate length of a net can be computed using the formula (1).

EstimatedWire-length=Total numberofedges *2/n (1)

Where n represents the total of total nodes. The HPWL has a shorter wire-length estimation than the ST (Steiner tree) and RSMT (Rectilinear Minimum Steiner tree).

2. WIRELENGTH CALCULATION

Wire length plays a significant influence in vertically

stacked ICs. Typically, the wire-length of the two terminal nets can be calculated using equation (2). Speed and accuracy of estimation have a significant impact on the efficiency of placement algorithms. We can estimate 2-terminal nets using the Manhattan distance. If (x1, y1) and (x2, y2) are the end coordinates, then the wire length is

$$L= |x1 - x2| + |y1 - y2| \quad (2)$$

The half perimeter wire length (HPWL) is the ideal method for calculating wire length due to its efficiency and widespread application. The most important characteristic of HPWL is that it locates the smallest bouncing rectangle that encompasses all of the to-be-connected pins of the network. A significant amount of the perimeter of this rectangle, which is computed by equation, represents the assessed wire length (3)

$$L(x,y)=\sum(\min|xi - xj| + |yi - yj|) \quad (3)$$

3. ALLOCATION OF WHITE SPACE

The WSA is mostly employed for routing purposes. In other words, the higher density region estimate approach has a stronger effect on WSA's performance. There are numerous ways to compute the density for each bin, resulting in varying white space circulation. These varied routes are primarily determined by the wiring probabilities within the bounded box. The density of each container is determined by the horizontal and vertical wires running through each container.

The estimation of congestion locations causes a delay in timing. This WSA also increases the wire length, although the HPWL approach described in section II can optimize this wire-length overhead issue.

There are various White Space Allocation Methods (WSA). This includes (I) Direct allocation, (II) Two-step allocation, (III) Allocation with row white space control, and (IV) Aggressive allocation [2]. We pick the Aggressive allocation approach because even in densely populated places, a high-quality result can be obtained when compared to other ways of WSAs [2]; this is the case with the Aggressive allocation method.

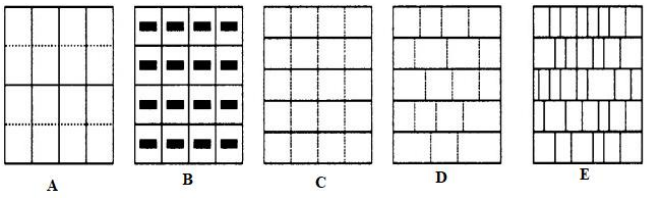

Fig 2. Design Flow

Here A represents Recursive bisection, B represents cluster annealing, C represents row adjustment white space allocation, D represents Cell annealing with white space in bins, E represents another allocation overlap removal local improvement.

Figure2 displays the white space allocation (WSA) in which the WSA has been repeated twice; both are in the detailed placement phase. The density region data is quite precise at this level of specificity. The primary assignment is determined following bin alteration. Currently, the number of columns in the bin equals the number of standard-cell lines for the design. The second allocation occurs following the annealing stage and prior to the cover expulsion (overlapping).

Steps for the design includes

- createm×ngridtopology
- nextanalyzethewhitespaceatfinerplacementlevel .
- next implementDijkstra's algorithmfor
- connectingwhite spaces.
- nextimplementwhitespacealgorithm-Slicing treeanditscorrespondingcutlinesandregionsslicing treeaftercongestion estimation andregionsaftercut linesadjustment.
- next implementLegalizationandLocal
- MinimizationbyPermutation
- placementflowgreatlyimproveroutability andreduceroutedwire- length.

A. Dijkstra's algorithm:

Widespread usage of the Dijkstra algorithm to determine the shortest path between white space nodes. This algorithm is utilized once the white space allocation has been computed.

B. Slicing Tree

The term slicing tree refers to the cutting of trees, which is the division of blocks. The tree's cut directions

are decided by comparing the aspect ratio values of certain parts to a die's preset value. Once the cut directions and cut sites have been determined, the cells/blocks to the left of the cut line when it is cut vertically or above the cut line when it is cut horizontally constitute the left child of the tree node.

In contrast, the remaining tree cells after the cut line constitute the right child of the tree node. These cuts are based entirely on two methodologies. The approaches are (I) top-down and (II) bottom-up. The total routing overflow of the grid cells contained within a leaf node can be used to determine its density level. Using a post-order traversal of the tree, the density level of an internal node can be determined by adding the densities of two child nodes.

During the legalizing phase, all the cells are aligned so that any overlap between them can be eliminated. The cells are arranged according to the ascending x-coordinates of their centers. By deciding one cell at a time from the leftmost cells of a row, the cost of relocation can be decreased. Here, the migration cost is defined as the cost of a cell moving from one region to another.

In the reduction process, the wire length is further reduced by seeking the local minimum wire length of a block having a small number of cells and potential white space. We slide the block from left to right and bottom to top, covering the entire chip with the previous block (overlapping). In other words, local minimization might lower the wire length estimation.

4. RESULTS

In this research, the routability-driven and wire-length optimization is performed using WSA and HPWL.

This experiment is conducted using the Windows operating system with MATLAB 2021a with 8GB of RAM. This input is the IBM-PLACE benchmark file.

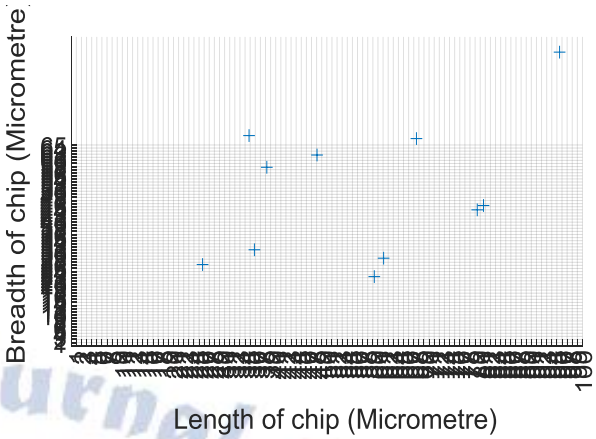

Fig 3. IC Placement after proper white space distribution for 100 micrometer Length and 100 micrometers in breadth of chip.

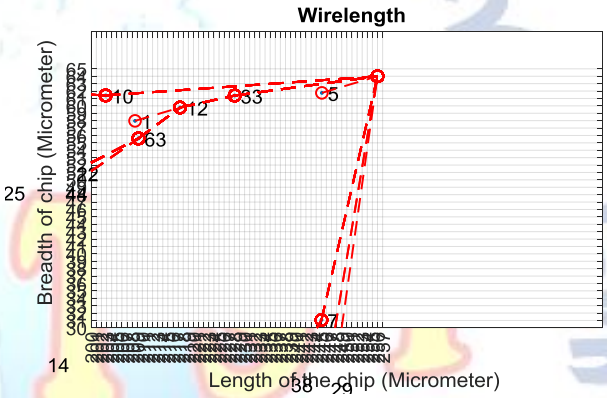

Fig 4, IC Wirelength

Table 1, Optimised Benchmarks

Benchmarks	#Cells	Grids	Optimized	Actual Wire-	White-Space
IBM01	12036	64*64	21733	22032	13
IBM02	19062	80*64	32034	32827	9.2
IBM03	21924	80*64	51212	51967	9.4
IBM04	26346	96*64	196672	199793	9.1
IBM05	32185	192*64	449170	450157	8.6

5. CONCLUSION

For a variety of IBM-PLACE benchmark circuits, MATLAB is used to determine the white space allocation and routability for the standard cell die. Using this WSA in the future, TSVs can be put between different layer blocks.

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

REFERENCES

[1] Chen Li, Min Xie, Cheng-Kok Koh, Jason Cong, Patrick H. Madden "Routability-Driven Placement and White Space Allocation" IEEE transactions on computer-aided design of integrated circuits and systems, vol. 26, no. 5, MAY 2007.

[2] Xiaojian Yang, Bo-Kyung Choi, and Majid Sarrafzadeh "Routability-Driven White Space Allocation for Fixed-Die Standard-Cell Placement" IEEE transactions on computer-aided design of integrated circuits and systems, vol. 22, no. 4, April 2003

[3] Sameer Pawanekar Gaurav Trivedi "TSV aware Standard Cell placement for 3D ICs" IEEE,

[4] S. Adya, I. Markov, and P. Villarrubia, "On whitespace and stability in mixed-size placement and physical synthesis," in Proc. Int. Conf. Comput.- Aided Des., 2003, pp. 311–318.

[5] S. N. Adya, S. Chaturvedi, J. A. Roy, D. A. Papa, and I. L. Markov, "Unification of partitioning, placement and floorplanning," in Proc. Int. Conf. Comput.-Aided Des., 2004, pp. 550–557.

[6] A. Agnihotri, M. C. Yildiz, A.Khatkhate, A. Mathur, S. Ono, and P. H. Madden, "Fractional cut: Improved recursive bisection placement," in Proc. Int. Conf. Comput.-Aided Des., 2003, pp. 307–310.

[7] C. J. Alpert, "The ISPD98 circuit benchmark suite," in Proc. Int. Symp. Phys. Design, 1998, pp. 80–85.

[8] C. J. Alpert, G.-J. Nam, and P. G. Villarrubia, "Effective free space management for cut-based placement via analytical constraint generation," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 22, no. 10, pp. 1343–1353, Oct. 2003.

[9] U. Brenner and A. Rohe, "An effective congestion driven placement framework," in Proc. Int. Symp. Phys. Design, 2002, pp. 6–11.

[10] A. E. Caldwell, A. B. Kahng, and I. L. Markov, "Relaxed partitioning balance constraints in top-down placement," in Proc. IEEE ASIC Conf., 1998, pp.229–232.

[11] A. E. Caldwell, A. B. Kahng, and I. L. Markov, "Can recursive bisectional one produce routable placements?" in Proc. Des. Autom. Conf., 2000, pp. 477–482.