

Layer-Aware 3D Partitioning in VLSI Circuit using Cuckoo Search Meta-heuristic

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Technology

in

Computer Science and Applications

Submitted By

Ashwin Parashar

Roll No. 601303007

Under the supervision of:

Dr. Maninder Kaur

Assistant Professor, CSE Department



COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

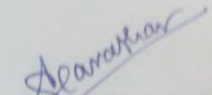
PATIALA – 147004

June 2015

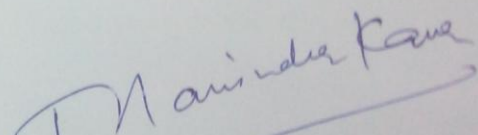
CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "Layer-Aware 3D Partitioning in VLSI Circuit using Cuckoo Search Meta-heuristics", in partial fulfillment of the requirements for the award of degree of Master of Technology in *Computer Science and Applications* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Dr. Maninder Kaur* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

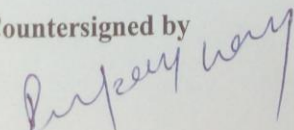

(Ashwin Parashar)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.


(Dr. Maninder Kaur)

Assistant Professor,
CSE Department

Countersigned by



(Dr. Deepak Garg)

Head

Computer Science and Engineering Department

Thapar University

Patiala


(Dr. S. S. Bhatia)

Dean (Academic Affairs)

Thapar University

Patiala

ACKNOWLEDGEMENT

First of all I would like to thank the Almighty, who has always guided me to work on the right path of the life.

This work would not have been possible without the encouragement and able guidance of my supervisor **Dr. Maninder Kaur**. I thank my supervisor for their time, patience, discussions and valuable comments. Their enthusiasm and optimism made this experience both rewarding and enjoyable.

I am equally grateful to **Dr. Deepak Garg**, Associate Professor and Head, Computer Science & Engineering Department, a nice person, an excellent teacher and a well – credited researcher, who always encouraged me to keep going with work and always advised me with his invaluable suggestions.

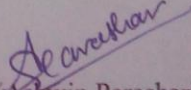
I will be failing in my duty if I don't express my gratitude to **Dr. S.S. Bhatia**, Senior Professor and Dean of Academic Affairs, Thapar University, for making provisions of infrastructure such as library facilities, computer labs equipped with net facilities, immensely useful for the learners to equip themselves with the latest in the field.

I am also thankful to the entire faculty and staff members of Computer Science and Engineering Department for their direct-indirect help, cooperation, love and affection, which made my stay at Thapar University memorable.

Last but not least, I would like to thank my family whom I dearly miss and without whose blessings none of this would have been possible. To my parents, I own thanks for their wonderful love and encouragement. I would also like to thank my brother, since he insisted that I should do so. I would also like to thank my close friends for their constant support.

Date: July, 2015

Place: Thapar University, Patiala


(Ashwin Parashar)

ABSTRACT

Three-Dimensional (3D) technology has great potential to improve the performance and ease of heterogeneity of the integrated system design. As the size of circuit increases day-by-day with the demand of more functionality and good performance requirements. To satisfy all these demands Two-Dimensional (2D) system design is not sufficient. This will enforce the researchers for the beginning of 3D-era. 3D Integrated Circuits (ICs) provide an appealing alternative to the 2D ICs. In simple term, stacking of 2D planar circuits form 3D circuit. By using 3D system design technology, one can integrate the several best technologies into the cube for the fulfillment of the requirements of an application. To achieve the high potential of the 3D ICs, it is more important to design strategies in such a way that can handle objectives and the complexness of the 3D design.

Partitioning is an initial and crucial phase of physical design cycle for a circuit. In partitioning, the netlist is divided into several parts by satisfying the constraints subjected to the system. But in the 3D partitioning, partitioning of netlist is not only the sufficient step. Although, it is required to assign partitioned blocks to the different layers of the 3D design of the system. Inter-block connections are responsible for the inter-layer communication of the design. These inter-connections are realized by using Through-Silicon-Via (TSV) technology on the circuit board. TSV has great impact on the performance of the circuit due to the reasons that it affects the global wire-length, signal delay of the circuit and many more. A TSV occupy more area as compared to other components of the circuit on die area. This fact cannot be ignored because numbers of TSVs may lead to the larger size of the chip. Hence, optimization of TVS becomes main objective in 3D-partitioning of physical design. In the work, an approach named TPCM (**T**hree-dimensional **P**artitioner based on **C**uckoo **M**eta-heuristic) based on the Cuckoo Search Meta-heuristic for the partitioning and layer assignment of the netlist in 3D system design, with the satisfaction of constraints like area balancing, inter-tier connections optimization etc is proposed. The proposed TPCM approach is tested against IBM benchmarks and the results are compared with the Meta-Genetic algorithm. The results show better performance of proposed approach for the 3D-partitioning in comparison with Meta-Genetic one.

Table of Content

S. No.	Topic Name	Page No.
	Certificate	i
	Acknowledgement	ii
	Abstract	iii
	Table of Content	iv
	List of Figures	v
	List of Table	vi
1	Introduction	1
1.1	3D VLSI Design	1
1.2	Partitioning: A Process of Physical Design	3
1.3	3D Partitioning	4
1.3.1	Layer Unaware Vs Layer Aware Partitioning.	6
1.4	Evolutionary Approaches	7
1.4.1	Cuckoo Search algorithm	9
2	Literature Review	11
2.1	3D Partitioning related Preliminary work	11
2.2	Gaps in Literature survey	20
3	Problem Formulation	22
4	Proposed work	23
4.1	Benchmarks Used	23
4.2	Solution Representation	25
4.3	Step Size	26
4.4	Up-gradation in Solution	27
4.5	Fitness Calculation for the Solution	28
4.6	Threshold Calculation	29
4.7	Selection of Best Solution	29
4.8	Proposed Algorithm	29
5	Simulation Results	32
6	Conclusion and Future Scope	37
	References	38
	List of Publications and Video Link	42

List of Figures

Figure No.	Figure Name	Page No.
1	Migration from 2D to 3D VLSI Integrated circuit	2
2	Flow for 3D ICs	5
3	A TSV based 3D structure	6
4	(a)A 4-way min cut partitioned designed (b) worst possible (c)best possible	7
5	Sample circuit with cells and pads	24
6	Solution representation	25
7	New technique for initial solution representation	26
8	Representation of n^{th} and m^{th} solutions	27
9	Proposed algorithm for circuit partitioning with min-cut	30
10	Shows the value of cuts for different IBM benchmarks at different level of partitions	34
11	Shows the average CPU time for different IBM benchmarks at different level of partitions	34
12	Correlation between P and CPU time	35
13	Comparison between number of TSVs generated by Cuckoo search and genetic algorithm for 4-layer partitioning.	35
14	Shows the relation between number of TSVs and number of Layers	36

List of Tables

Table No.	Table Name	Page No.
1	Detail of benchmarks used	32
2	Comparison of cuts and CPU time (in best, average and worst case) for proposed algorithm and meta genetic approach (pop_size is taken as 10)	33

1.1. 3D VLSI Design

The fast improvement in development of the computer and the information Technologies is demanding very large scale integrated (VLSI) circuits with high functionality and performance at power dissipation and minimum cost. To meet this demand, VLSI circuits are scaled. But due to large scaling of circuit some other issues come under the focus for semiconductor industry. For large size circuit, inter connection delays are increase and increasing in the interconnection loading influenced the consumption of power in high performance chips. Clock net of the circuit consume significant amount of power of the overall power consumption by the circuit. Hence it should be important to route clock net efficiently on the circuitry board but for the large sized circuit, cloak net has to be large enough. Additionally, scaling in inter-connect has significant effect on Computer Aided Design (CAD) methodologies and tools i.e. number of design cycles increases, and hence there is increment in the time and the cost per chip function. Integration of disparate signals like- digital, analog and technologies like- SOI, GaAs and so on is introducing for which existing two-dimensional (2D) technology may not be appropriate.

Nowadays, Three-dimensional (3D) integrated circuit (IC) is a providing better solution to the new generation of electronic designs. In 2D technology all devices are arranged on the single chip whereas in 3D ICs multiple chips are stack one on other having smaller size as compared to the 2D ICs. Hence in result, form factor of 3D ICs is reduced and packing of devices become closer. Due the compactness of 3D-IC, it helps in the reduction of interconnects and hence help to improves performance. Through 3D-ICs, the heterogeneous integration can be realized by fabricating chips having different functionality on single stack. With the help of Through Silicon Via (TSV), one can achieve inter layer communication throughout the chip stack. TSV is micro channel of metal passes vertically through the stack of chips by penetrating within silicon substrate. A TSV links net connecting different chip of 3D design, by placing it on any white space that can be viable on the chip. With the help of TSV there is significant reduction in signal delay and wire-length. And hence it becomes

important technique for the 3D physical design of the circuit. 3D IC become the main attention topic since past few years and has been observed for many perspectives like fabrication process [1], [2], cost of production [3], thermal issues [4] - [6]and yield rate [7]. In the initial research it was found that there was significantly reduction in interconnections with the presence of vertical connections. A high density of inter-tier connections on the chip with TSVs can be incorporated but the area occupied by TSV on the chip cannot be ignored. But in reality as the numbers of TSVs are increased, there is increase in size of chip and length of inter-connection. And hence, realization of large number of TSVs on the chips does not favor the better performance of the design. On the other hand, minimum number of TSVs unable to bring advantages incorporated by TSVs. Hence, to get complete benefit of TSVs for the 3D physical design of chip it is required to optimize the number of TSVs. M. Tsai et al. [24] have studied impact of number of TSVs and wire-length on the 3D-ICs. They have done this studied for varying size of TSVs, i.e. from 10 μ m to 200 μ m, and evaluate the optimum number of TSV such that performance of the chip became high [10], [11]. Therefore, it is important to find out that how the designs of 3D-ICs are affected by different technologies of the TSVs.

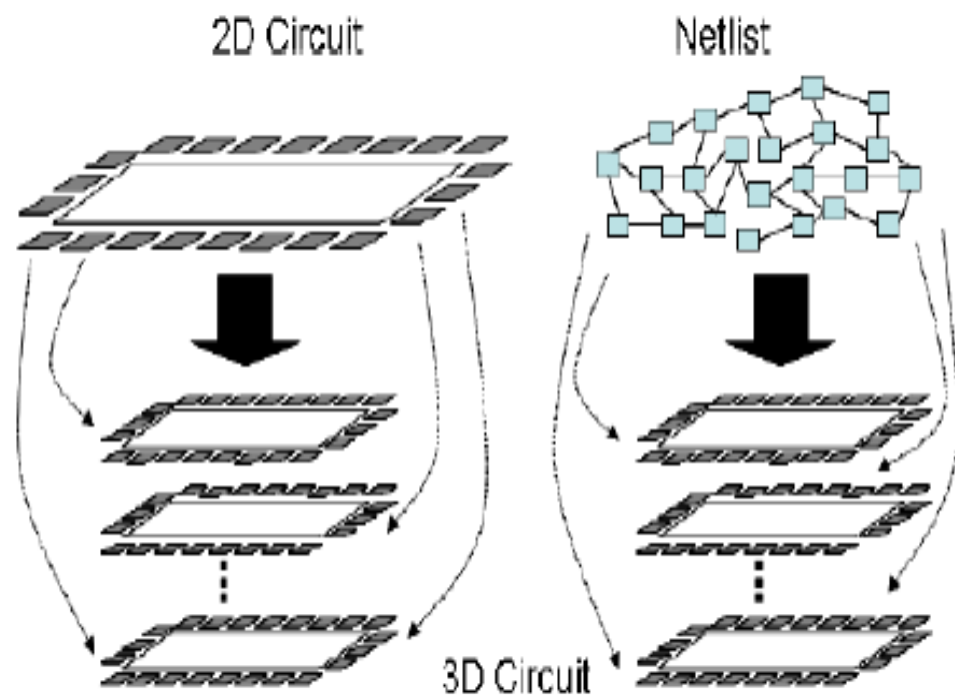


Fig.1. Migration from 2D to 3D VLSI Integrated circuit [21].

1.2. Partitioning: A Process of Physical Design

A famous methodology design to reduce the complexity of designing modern ICs is to divide them into smaller modules [8]. The size of designed module may vary from small electrical components set to completely functioning ICs. The large circuit is divided into the many smaller sub-circuits (also known as block, sub-module or partition) with the help of partitioner subjected to the constraint such that minimizing the number of inter-connections among the sub-circuit or different partitions [9].

Circuit contains several million numbers of transistors. In partitioning phase of circuit, large numbers of circuitry elements are grouped into smaller block or parts in such a way that there should be lesser number of inter block connections among the circuitry elements. Circuit partitioning comes under the NP-hard problems set. So, there is no such algorithm in existence to solve this problem in polynomial time.

A partition on the chip cannot be realized independently, i.e. without considering other partitions of the same circuit. Because it would lead to the reduction in performance of the design, increase the delay in the circuit and the circuit reliability also decreases. Inter-connections between blocks decide the dependency of one block to the other. If there is higher dependency between the blocks it will be more complex or hard to realize a single block on the chip. Therefore, the main aim for the circuit partitioning is to reduce number of inter-connection between the blocks. Each partition must fulfill the overall design constraints.

In the physical design of ICs, phase to partition the circuit has an important role [9]. In general, an IC is made up of millions of transistors. It has to be sub-divided into many smaller sub-modules for the ease of the design process. For the inter-connection among the block, there is a terminal available at the periphery of the block which are used to connect the blocks. Connections are mentioned by a netlist, netlist is the collection of nets.

There are various levels of partitioning in the physical design of the circuit. Some are-

- System Level, in this system is partitioned into group of PCBs.
- Board Level, in this PCB is partitioned into chips.
- Chip Level, in this chip is partitioned into smaller sub-circuits.

It is necessary to mention the set of constraints for every level of partitioning because they are different for every level of partitioning [12], [13]. It may be possible that at one level of partitioning, one of the constraints may have more priority than the others. Whereas, at another level of partitioning that constraint may have lesser priority than the other. So, every time it is required to mention all constraints in sequence according to their priority to be satisfied completely.

1.3 3D-partitioning

In the modern era, the technologies become advanced and come in compact size day by day. This compact size of electronic devices is only possible due to the introduction of System on Chip (SoC) ICs. On SoC, it is possible to integrate all the components of any electronic device on the single chip. It can support analog signals, digital signals, mixed signals and radio frequency functions simultaneously. Rather than creating specific purpose chips, it is more emphasized on developing general purpose chips. Due to this, one single chip has to hold the several numbers of the circuit elements like transistors etc. This would result in large size of the chip and this larger size will be responsible for the signal delay on the circuitry board specially for clock net delay, which is responsible for the synchronization of the processing on the chip. Nowadays efforts are made to realize heterogeneous technologies on the chip to provide multiple functionalities to them. This may lead to poor yield of the chip. To avoid such kind of problems, 3D-IC has been developed. With the help of 3D-ICs it is possible to reduce the global length of wire or net, power consumption of the chip and hence performance of chip is getting high.

Andrew E. Caldwell [26] described two reasons that cause performance deterioration and proposed some modifications to remedy this deterioration. They performed two or more calls to black-box partitioner which was iterative in nature. Every call used a small tolerance for partitioning; the first call tolerance was large enough for the every transferable node, whereas the last call had original requested or desirable tolerance value. Solutions generated by preceding partitioner are used by the next call. The illegal solution was 'greedily legalized' with respect to the smaller tolerance before the next partitioner call.

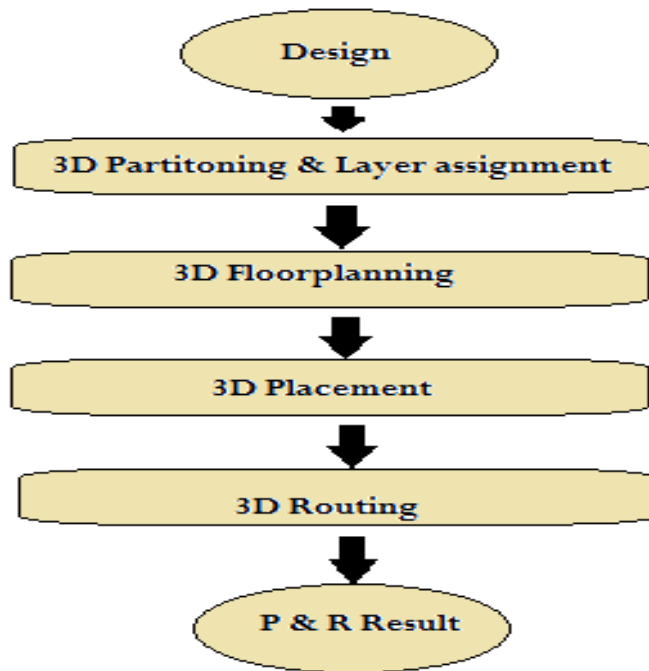


Fig.2. Flow for 3D ICs [19]

The problems of two dimensional space can be solved at some level but as going through the three dimensional space, new major issues are faced in 3D technologies. There is a need to redesign existing algorithms in such a way that they can be able to produce optimum results for the 3D ICs. There is a necessity to develop 3D physical design cycle for the circuit means development for 3D partitioning, 3D placement, 3D routing, etc. The current research work focuses on the 3D partitioning of the circuit. It is very crucial step in the physical designing of the circuit. Results produced by this step are fed to the next steps.

In 3D partitioning, the give circuit is firstly partitioned into parts such that the number of inter-connections among the parts remains small in number. After the partitioning of the whole circuit into parts, all different parts will be assigned to the different layers of the 3D-design, which are stacked one on another. To enable the communication among the different layers of the design, the concept of 3D-via is introduced which can be implemented by Through Silicon Via (TSV). By increasing in the number of TSVs, the global wire length of the circuit can reduced. But TSV occupy some space on the chip which is an overhead of the circuit design. So, larger the number of TSVs larger will be space occupied by them on the circuit and this would results in large size of the chip. So, the aim is to optimize the number of TSVs such that

it can help in reduction in of global wire length as well as in minimizing the area of chip with area balancing.

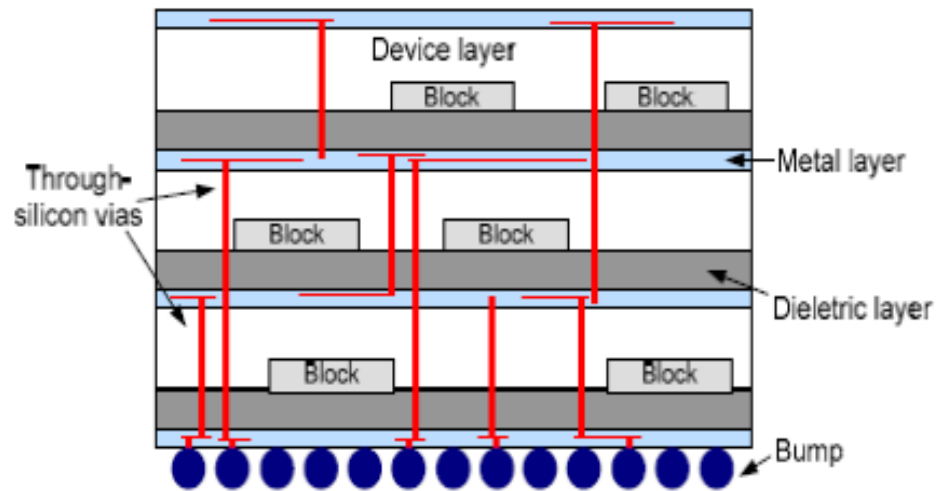


Fig.3. A TSV based 3D structure [19]

1.3.1. Layer Unaware Vs. Layer Aware Partitioning

The main aim of layer unaware as well as layer aware is to minimize the number of TSVs so that one can achieve smaller size of the chip. In layer unaware strategy, there is a need to just partition the circuit into smaller sub-circuits without specifying the layer position of the 3D-architecture. Sometimes this would be lead to worst solution after assigning the partitions to the different layers of the 3D-design although the results produced by partitioner are good. Whereas in case of layer aware partitioning, partitioning process is held layer by layer i.e. with the specification of the layer to the new partition generated. in layer aware strategy, min-cuts produced by I/O pads are also considered because they are also responsible for the inter-layer connections and the generated schema will be more closer to the circuit design in reality.

With the help of the following example this can be easily understand that how the layer specification of the partition vary the result for the same partitions produced by the partitioner.

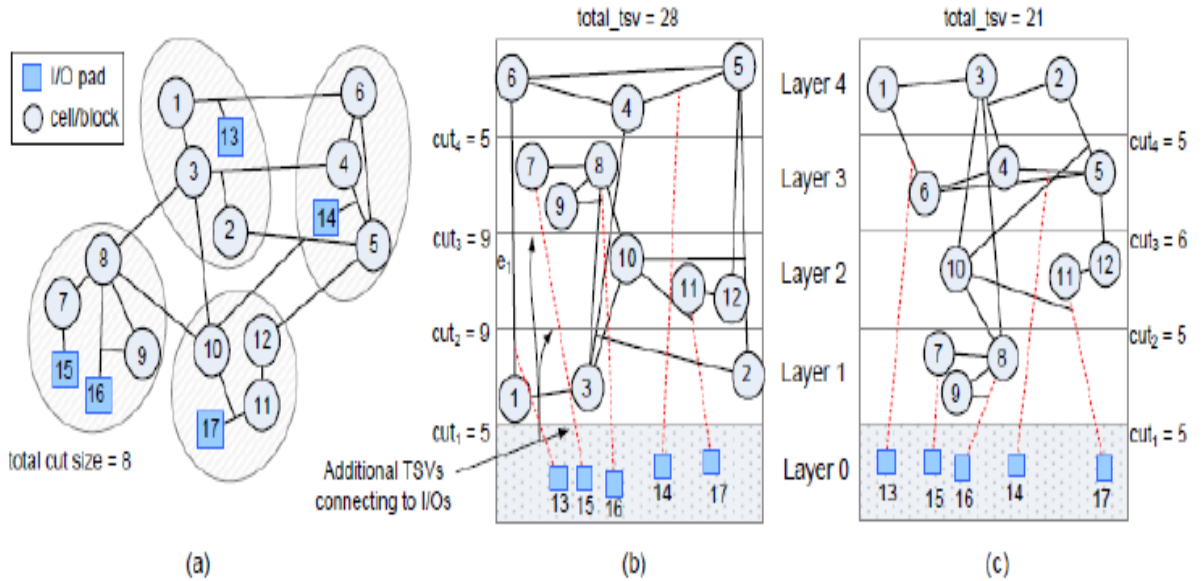


Fig.4. (a) A 4-way min-cut partitioned designed (b) worst possible (c) the best possible [19].

In fig.4 (a), it is shown that a given netlist is partitioned into 4 partitions. But as shown in fig.4 (b), it may lead to produce large number of TSVs due to inappropriate assignment of partitions to the layers and hence the number of total TSVs becomes 28. The number of TSVs can be reduced by placing same partitions to the appropriate layers with layer aware partitioning with the consideration of nets connecting to the I/O pads, as shown in fig.4 (c). With the layer aware partitioning number of TSVs reduced to 21.

1.4. Evolutionary Approaches: A brief introduction

Evolutionary algorithms (EA) are based upon the evolution in the population of solution achieving by repeated modification. The major difference between the evolutionary algorithms and traditional algorithms is that in evolutionary algorithm there is a set of solutions, known as population, which is used for simultaneous manipulation where in traditional algorithms start with single solution and manipulate it. In EA, operators are needed for the betterment or modification of the solutions, these operator are drawn from the evolution occurred in nature.

General evolutionary algorithm starts with the construction of initial set of solutions, i.e. population. This initial population set is created randomly selection of solution from the solution space of the specific problem. It results in the collection of

solutions with good diversity factor i.e. there is great chance of finding such solutions having good and bad characteristics (qualities). Now by applying a series of nature inspired operators on the solutions to modify or improve them. For example, in genetic algorithm (GA), there are two kinds of nature inspired operators named as crossover and mutation. In crossover, there is involvement of two parent solutions to generate new solution (offspring). In the case of mutation, there is requirement of single parent for the generation of new solution (offspring). By using crossover operation, it is tried to bring the good characteristics of parent to the child (offspring). It is tried to bring new characteristics in child by using mutation operation. The quality of solution can be measured by using fitness function i.e. either the maximization or minimization of something in the result according to the problem definition. Encoding of solutions and functioning of operators may differ for different problems. But the basic idea behind all encoding and functioning remains same, i.e. inspired by nature. Solutions with good characteristics propagate to the next level of generations which is more similar to the concept of survival of the fittest in the nature. In EA, a set of solutions used initially for the further manipulation that will avoid the convergence of the solution to the local optimum solution in the solution space and solution will converges toward best solution fast enough [14]. In EA, there is a need to define either some threshold value up to which solution produced can be accepted or number of iterations up to that times solution will evolved. Both of the things, i.e. threshold and number of iterations, can be specified together according to the problem statement.

Many of the work have been done in the field of VLSI physical design process using the nature inspired algorithms to solve the complex problem of the circuit design. Pradeep Kumar Sharma [28] presented a nature inspired meta-heuristic approach for the solution of circuit partitioning with the constraint that inter block cuts should be minimum. VLSI circuit design has two main objectives out of many objectives. First, to minimize the number of inter block connections and second, performance improvement with respect to time. R. Banos [30] described a parallel evolutionary algorithm (EA) for partitioning of the circuit. Due to parallelism, there was improvement in the solution as compared to the corresponding sequential algorithms. S. S. Gill [31] presented multi-way circuit partitioner which was based on the Genetic Algorithm. The main objectives of partitioning of the circuit were to minimize interconnection delay, to minimize number of interconnections and minimization of ratio-cut.

1.4.1. Cuckoo Search algorithm

Cuckoo search is a meta-heuristic method for the optimization problems. Cuckoo Search algorithm is based on the obligate brood parasitic nature of some species of cuckoos with combination of Levy flight nature of some fruit flies and birds.

Many of the nature inspired algorithms were proposed like PSO, which was inspired by the bird and fish swarm, firefly, which was inspired by the tropical fireflies. All these meta-heuristics [15] have efficiency to solve NP hard problems like Travelling Sales Man etc. All nature inspired algorithm have two characteristics i.e. selection of fittest and the adoption to environment. Modern meta-heuristics has two crucial characteristics i.e. Intensification and Diversification. Intensification means searching of best solution around current best solution. Whereas Diversification is responsible to ensure that nature inspired method will explore all search space efficiently. This will lead to the avoidance of local optimum results as output of the algorithm. The main reason behind the power of all nature inspired meta-heuristics algorithm is comes from the fact that it mimic the best characteristics in the nature.

Cuckoo is an attractive bird not just for its sweet voice, but also for the fact that some species of Cuckoo has an amazing quality that they laid their eggs into the other birds nest in such a way that sometimes it will be hard to find alien egg by host bird itself or sometimes after laying the egg it remove the eggs of the host bird to increase the probability of hatching of their own eggs. In addition, some species of cuckoo like Tapera have the characteristic to mimic their egg similar to the egg of host bird of some species in colour and pattern to make the lesser possibilities of recognition of their egg. Although, if alien egg is recognized by the host bird there may be two thing can be happen either it simply thrown away from the nest or the host bird searches new space for their nest.

To make cuckoo search algorithm simple and easy to implement, programmer or developers designed three basic rules to follow for the cuckoo search algorithm. That are -

- 1) One cuckoo can lay only single egg at a time and nest selected by cuckoo will be random.

- 2) The nest containing best egg would be selected as best nest and proceed for the next generations.
- 3) Fixed number of nest would be available and $P\alpha$ shows the probability of finding alien egg such that $P\alpha \in [0, 1]$.

Cuckoo Search was quite better as compared to other meta-heuristic[16], [17] methods like genetic and PSO for multi-objective problems, this may be because of the reason that in cuckoo search there was no need to tune or set many parameters like in PSO. There is a need to set only single parameter i.e. $P\alpha$ for the cuckoo search algorithm and it is required to change this parameter for every problem. This proposed algorithm can be extended for the multi-objectives optimization problems with many constraints, even for the NP-hard problems.

There are some work has been proposed in VLSI for the standard cell placement problem in two dimensional integrated circuit which based on the behavior of Cuckoo. Amanpreet Singh [18] presented cuckoo's behavior based meta-heuristic approach for the solution of standard cell placement problem, named as Discrete Cuckoo Search approach Standard Cell Placement (DCSSCP). Results are tested on the set of MCNC benchmarks circuit and compared with the pre-existing meta-heuristic approach.

Chapter 2

Literature Review

2.1 3D Partitioning related Preliminary work

Since past few years, valuable work has been done which can serve as well starter for the future researches on the three dimensional (3D) physical VLSI circuit design. Some related works are as follows -

Huang, Y. et al. [19] presented a layer aware iterative method named “iLap” for the minimization of TSV in 3D structures. iLap applied on given netlist in bottom-up pattern, iteratively on each layer of the partition. iLap also considered the I/O pads constraints raised by the 3D design of the die for the betterment of the further results. Experimental results generated by iLap showed that number of TSV reduced by 35% as compared with other pre-existing methods.

In the proposed method, hMetis is used as the core engine to generate partitions of the input circuit. They considered that all I/O pads must reside at bottom most layer of the 3D structure. They considered all I/O pads as single node (super-vertex) of the graph with negligible weight to avoid area balancing disturbance in further steps of partitioning. The iLap results were compared with the results generated by other hMetis based method like EV-matrix and EX-hMetis. iLap always produced better result in term of number of TSVs as compared to the other hMetis based methods when they tested proposed work against MCNC benchmarks, Altera and in-house design named fft128.

Ghosal, P. et al. [20] presented a strategy for the partitioning of netlist with the constraints of minimization of inter-tier interconnections as well as area balancing. Inter-tier connections are implemented through TSV which occupy lot of area. They assigned weight to each and every logical module according to the area occupied by the module on chip. This help in area balancing. Adjacency matrix of netlist and the weight of the module supplied as input for the partitioning with the assumption that each module had finite number of output lines.

The proposed algorithm generates the output where different logical modules are allocated to different tiers. The proposed strategy has two main parts: numbers of layers

prediction and assignment of modules to different layers. In the tier prediction phase, problem of number of tier for the given 2D netlist considered as mathematical optimization. As the numbers of tier are increases numbers of TSV are also increases. In the second part, different modules of the inputted netlist were assigned to different tiers as determined in first phase with the focus on area balancing constraints. The proposed work had different conditions of tie during the execution of algorithm. One of the special cases of tie was mentioned as when many modules were in competition to occupy common seed position of module. In that situation, the minimal path distance is calculated between the selected one and module in competition and their summation value was taken to resolve the tie. The suggested work was run against different fan out values with different number of modules and corresponding generated values of area variation, inter-tier connections and CPU time were noted.

Sawicki, S. et al. [21] proposed a new method for the partitioning of cells and I/O pads with the reduction in 3D vias and its effect. They presented two different strategies: one is based on analysis of circuit structure and second is based on minimization of number of inter connection between non adjacent layers.

In the proposed algorithm, firstly the distance between I/O pins evaluated. Then, taking logical distances as weights complete graph created. Finally, through hMetis partitioner graph is partitioned into parts. I/O pins were fixed to the partitions. At the end, to find the best arrangement simulated annealing applied. The shortest path values are used to design a complete graph connecting all pairs of I/O pins.

They have designed a function to transfer cells among the different partitions named as perturbation function. This function was random by nature. They used two different type perturbation functions. In one, only one cell was transferred to another partition. Whereas in other, swapping of cells from different partitions were took place. They have suggested a refinement process through that they made reduction in number of 3D vias in the connections which were long. This refinement step took place after the assignment of partitions to the different tiers.

Cheng Hu, Y. et al. [22] proposed multi-layer partitioner algorithm for 3D ICs. It is based on multi-level framework to partition the input netlist. On each level of partition partitioning method is applied during un-coarsening stage. The main aim of proposed

work is to minimize the total number of TSV with the consideration of area of each layer. Area of circuitry elements and area of TSVs collectively form the total area of the chip. The proposed algorithm can be customized according to the 3D ICs structure.

Proposed algorithm consists of four phases: a) Multi-level coarsening- In this phase, smaller graph is generated successively. This phase was running until the threshold value was achieved or no further partitioning was possible. b) Initial partitioning- In this phase, partitions were assigned to the different tiers. Cells connected to the I/O pads are placed at first level of 3D structure. Then, other cells which were connected to the placed cell would be placed until the area of the die was completely occupied. This process was running until all elements were placed. c) K-layer partitioning- In this phase, FM liked partitioner is applied. Main aim of this phase was to reduce the number of TSVs without disturbing the pre-defined constraints. As in result, reformation of K-level partitions took place. d) Un-coarsening and refining- In this step, previous step was reapplied on finer level. Netlist at finer level has more degree of freedom and this procedure was move to increase the quality of the resultant partitions.

Chen, Y. et al. [23] proposed a two-phased layer aware algorithm for the partitioning with the objective of minimization of TSVs in 3D design. In first step, they applied OpenMP to circuit to parallelize the bi-partitioning with minimum numbers of cuts. In second step, using simulated annealing method the result of first phase was improved. Proposed work can reduced the TSV's numbers by 39% in the comparison of the many pre-existing methods.

In the proposed work, they designed a 3D partitioning problem as multi-way as well as layer aware partitioning problem. Proposed algorithm partitioned the netlist into K sub-divisions in parallel and each sub-division is associated with a layer in such a manner that the total number of TSVs is minimized. The proposed work has two phases i.e. deterministic and stochastic phase. Deterministic phase further subdivided into divergent and convergent steps. In divergent step, min-cut bi-partitioner is applied on the netlist until the desirable number of partitions generated. This step was random in nature. In the convergent step, refining of partitioning took place by merging and splitting the adjacent layers until all the layers were merged once. In stochastic phase, they used power of GPUs for the further reduction in the TSV's count. They compared

their proposed work with the hMetis based algorithms like iLap, hMehtis and EX-hMetis for the four and eight layer architecture against MCNC and altera benchmark and noted the run time of the CPU, which quite better than the run time of the other algorithms. But the main disadvantage of the proposed work is that it only generate partitions for the 2ⁿ layers design.

Tsai, M. et al. [24] presented a study on the number of varying size TSV and wire-length. TSV can help in the minimization of wire-length but it occupy more area on the die which impact negatively on the designing of the chip and cancel out the benefits brought by TSVs. Experimental results explained that maximum area occupied by TSV was 25.3% of the area occupied by cells. Beyond this, no wire-length reduction can be obtained.

In proposed work area of die is calculated by adding area occupied by cells and area occupied by TSVs, and wire-length is measured with the help of die size. Since, wire-length and number of TSVs are depending on different net topologies. So in this paper, they discussed two wire-length analyses with different kind of net topologies. In the paper, they suggested two different kind of wirelength analysis because wirelength is varied with the variation in net topology. First was wirelength analysis of three dimensional spanning tree. Second was wirelength analysis of different three dimensional placements. The proposed work is tested against the IBM benchmarks with fixed size of TSV, variable size of TSV and the value of wirelength generated were noted for the all kind of cases. At the other hand, they noted the number of TSV for the different kind of the placements. And they plotted the variation in number of TSVs for the different value of α . And they also described that how the wirelength of circuit is changed with the variable value of α . At last they concluded that for the smaller size TSV, minimum spanning tree methodology was best. Whereas for the large size TSV, minimum spanning tree methodology with least value of TSVs was preferred to minimize the total wirelength and total number of TSVs.

According to the paper, TVS technology applied decided the best partition scheme. Firstly, it would be better to partition the netlist for the assignment of the modules to the different layers for the bigger sized TSV.

Ababei, C. et al. [27] presented 3D-Adopt for short, which is a framework in which they designed CAD methodology for the routing and placement in 3D ICs. For every style, it is not necessary that all the factors would be same. So they have designed separate approaches for the 3D structure for every technology. Because of the high clock frequencies and higher density in ASIC, ASICs have thermal issues which is more crucial as compared to the thermal issues in FPGAs. Whereas in FPGA for higher connectivity additional fabric cost is required. In 3D FPGA, number of inter tier via are more than the 3D ASIC. So, FPGA required optimization in two steps. One was the optimization of inter tier via and second for the optimization across and within the tiers.

Width and separation between vertical layers of FPGA can be determined by the amount of solder bumps which can be fitted on a die. It was assumed by previously work that for the inter-layer connectivity, vertical wire can connect only adjacency layers. For the partitioning of netlist they used hMetis and after that a novel method is applied for the placement in such a way to minimize the total wirelength and maximize the number of net cut between adjacency layers of 3D design. This can be achieved by the mapping this problem to the bandwidth minimizing heuristic of efficient matrix. Bandwidth minimization problem comes under the NP-complete problem. And the solution generated for the layer assignment problem might not be optimized in terms of wirelength minimization and cut size maximization between adjacency layers of 3D-structure. 3D technologies provide a better way for the improvement in the performance of the circuit. It provides flexibility in third dimension up to some extends. In this paper, they provided some view over the routing and placement methods for the ASIC and FPGA design style. Several directions still remain to be solved which were not solved in paper.

Hentschke, R. et al. [29] presented a methodology to minimize the number of 3D vias with the area balancing constraint during the partitioning phase of the circuit in the 3D physical design.

Introduction of 3D technologies helped to resolve some issues of 2D design like it is responsible for significant reduction in wirelength of the circuit as compared to 2D design. But 3D technologies have some own issues like power dissipation and how to make insertion of 3D-vias to enable the inter-tier communication. If there is net connecting top most and bottom most layer through the net. Then, the net has to be

passes through all intermediate layers of the 3D architecture. There are many granularity levels in the 3D integration varying from core level to transistor level. Sometimes, it would be hard to gain physical constraints imposed by finer granularity about the higher number of 3D-via demand. During the cell partitioning, layer assignment of cells determined the number of 3D-vias for the design. Usually cell partitioning, it done by hyper-graph partitioning tools like hMetis. Amount of resources are directly proportional to the vertical distance between the layers.

In the paper, they presented a refinement heuristic which reduced the number of 3D-vias in the results produced by the traditional hyper-graph partitioner with the area balance constraints. The main difference between proposed work and other hyper-graph partitioner was that they considered the partition locations. There were two kinds of partitions one is adjacent partition which was cheaper in terms of cut and second was distance partitions which was costly in terms of cut. They first employed hyper-graph partitioner like hMetis to make initial partitions and then they proposed a refinement strategy to minimize the number of cuts. In the proposed work there were two kind of perturbation function i.e. 1) single perturbation, in which there was a random selection of cell or I/O pin from any partition and then shifted to the different partition. 2) double perturbation, in which a pair of circuitry elements selected randomly from two different partitions and swapped together. For the quality measurement of the intermediate states cost function was defined.

The proposed algorithm performed better in partitioning than the hyper-graph partitioner because hyper-graph partitioner were unable to handled long vertical connections. In this way, they influenced the total number of 3D-vias and produced less realistic outcomes.

Ouyang, J. et al. [33] described the design for two three dimensional (3D) Arithmetic Units i.e. 3D Multiplier and 3D Adder. Both are implemented using three dimensional Through-Silicon-Via technologies. Results are compared with the two dimensional counter parts. 3D adder optimized 11.0% to 46.1% less energy and 10.6% to 34.3% less delay when they had increased width from 12 bits to 72 bits and 3D multiplier (32 X 32) incurred 6.8% less energy 14.4% less delay.

For the reduction in interconnection pressure and improvement in the intensive interconnect design, three dimensional integrated circuit were proposed. In three

dimensional integrated circuit, partitioning of two dimensional chips took place into the number of smaller dies and stacked them on the top of each other. To carry propagation of signals from one layer to other layer of the 3D design Through Silicon Via (TSV) were fabricated. There were many advantages of using 3DIC: 1) the device routing and density per unit area increased effectively with the number of layers or dies. 2) Distance among the components of the two dimensional chip reduced in the three dimensional chip if the component were aligned carefully. And hence this will result in reduction in global wire length of overall circuit. Advantages of three dimensional (3D) integrated circuit have been discovered and applied the different kind of integrated circuits. They had proposed three dimensional (3D) integrated circuits (ICs) for the arithmetic unit to improve the performance of 3D Kogge-Stone adder and 3D Wallace Tree multiplier. Bit-wise scheme for partitioning are used by the adder. Along with the last two level boundaries multiplier are partitioned.

Marinissen, E. J. et al [34] presented a DfT test for the three dimensional stacked integrated circuit for both post bound stack testing and pre bound stack testing. The architecture of DfT was based on modular test approach. They could be tested various dies and their IP cores, inter-die interconnections (TSVs), and the external IOs separately for the optimization of the flow of the test for three dimensional stacked integrated circuit. With reuse of pre-existing hardware of DfT at different level, i.e. core, die and product, the design or architecture of DfT was built on which was based on IEEE 1500. It had some characteristics like it contained probe pads on all dies or layer except bottom layer for the testing of pre bound die. It contained test elevators for the transportation of data and control signals up and down for the testing of post bound stack. And it had Wrapper Instruction Register (WIR) hierarchical chain. They also presented some hints regarding the standardization and optimization of proposed architecture.

In this paper, they focused on the design of scalable and structured architecture of test access. Proposed architecture testing of pre bound die, post bound stack and final packaged product. It was based on the modular testing of inter die inter-connections (TSVs) and intra-layer circuitry. Stacking of possibly heterogeneous and black box dies were also supported by it. The architecture support the reusability of design for the test structure encountered commonly within different dies as much as possible. Die-level wrapper is the main component of 3D-stacked integrated circuit test access architecture

which is based on the IEEE 1500. It was extended with the implementation of three characteristics i.e. Probe Pad, used for pre bound testing and situated at all layer except bottom layer. Test Elevators, used for the transportation of data and control signals among the inter layer of the three dimensional design. Hierarchical WIR chain to for the prevention of uncontrollable growth of length of it. The architecture proposed was structured in such a way that it provided a commonly used DfT template that satisfied all test access requirement of 3D stacked integrated circuit. The proposed design or architecture was in scalable in nature means it could be applied on any number of layers of the three dimensional design and variable bandwidth of user defined access test provided. It provided opportunity between test length and silicon area.

Ye, H.S. et al [35] presented an algorithm for the partitioning of an integrated circuit with the consideration of objectives to reduce the number of Through Silicon Via (TSV) as well as chip area. The main benefit of reducing the number of TSVs is not only the minimization in chip delay but also it helps to reduce the area of die. Proposed work partitioned the given circuit into sub-parts with minimum number of TSVs in short time period.

Three Dimensional (3D) Integrated Circuits (ICs) are based on technique in which multiple layers of device stacked one on other and form monolithic chip. 3DICs has many benefits over 2DICs. Some are listed as: 1) Density of integration is high: it can hold more components of circuit in per unit area as compared to the two dimensional (2D) ICs. 2) Heterogeneous integration: it can integrate the chip of variety of technologies, e.g. memory, mixed signal components and logic circuit. 3) Better performance: because of the significant reduction in wire-length, signal delay become minimum and hence result in better performance. 4) Power consumption reduces: large fraction of power consumption is consumed by the clock net. But due to reduction in the length of clock net power consumption get reduce. In this paper, they had proposed modified method for partitioning of 3DICs. Modified method was based on Fiduccia-Matteyses (FM) Heuristic. FM Heuristic is an efficient method which provides good solution for the problem of partitioning of circuit, i.e. network of vertices. It divides the network into two parts such that there will be lesser number of nets which has nodes in each parts after partitioning. FM heuristic was designed to handle a complex network which contains large number of hyper edges and cells of various weights. By using effective data structures and the idea of moving one cell at a time between the partitions

such that it helps to reduced the number of current cut nets, FM heuristic obtained linear time behaviour. FM algorithm is heuristic in nature hence the time to obtain good solution is based on the initial partitions. They had proposed heuristic based algorithm to solve the problem of partitioning of circuit with the constraints to minimize the number of TSVs and balancing the area of chip. Proposed algorithm had two parts, i.e. Initial partitioning and incremental refinement partitioning. In first, initial partitioning took place with minimum number of TSVs and balancing of different layers area. After that in second part of algorithm, they had found optimal results gradually by comparing the result of incremental partitioning with the initial level partitioning. If they have found better result in refinement partitioning. They continued the process until there was no more chance of the improvement in the incremental refinement partitioning. The proposed algorithm obtained very good solutions in both number of TSVs as well as in chip area. Area distribution was uniformed and the overhead were also reduced.

Lim, S.K. et al [36] presented many big challenges in the area of VLSI physical design for three dimensional (3D) integrated circuits (ICs) containing Through Silicon Vias (TSVs). Several issues were centered on Through Silicon Via (TSV), which were the new for the 3DIC design. For the successful and better design of 3DICs based on TSVs, fundamental knowledge of thermal, mechanical, and electrical properties of TSVs is necessary. It is crucial to investigate the impact of through silicon via on the qualities of overall layout, i.e. manufacturability, performance, reliability and power.

In this paper, Sung K. Lim presented many of the challenges for 3DICs in VLSI physical design. Some of them are illustrated as: 1) Pseudo tools for 3D and their limitations: there is lack of tools for the physical design of 3D ICs. Most of the tools are the only extensions of tools which are exist for 2D ICs. 2) Management of TSVs: it is crucial step in the designing of 3d ICs. Number of TSVs and position of TSVs has great impact on the performance and reliability of 3D ICs. 3) Thermal Management: Thermal problem is the one of the roadblock in the acceptance of 3D ICs in the mainstream. Due to the high element density on the chip several hotspot were created and hence background temperature rises. 4) Power Delivery: in 3D ICs, networks on each layer are connected with the help of Through Silicon Via (TSV). It was a big challenge in 3D ICs. 5) Clock Delivery: elements of circuit are placed on the different tiers of the 3D ICs. This incurs a big challenge in clock signal delivering to all layers with minimum power consumption slew, skew, and jitter. 6) Manufacturability Issues: TSVs occupy

large area on the die as compared with the other element. Hence, it may result in non uniform density distribution on the layout. 7) Interposer integration for 3D: first commercial product of 3D ICs was processor + memory stacking. The memory capacity of this product is depend on the number of tier available for the memory in 3D architecture. This paper discussed major challenges of TSV based 3D ICs like need of 3D IC tool, TSV based partitioning and placement, management of thermal power, delivery of clock and power, manufacturable design for TSV based ICs etc.

Kim, T.Y. et al. [37] proposed an algorithm for the implementation of clock tree, which was cost effective, in Through Silicon Via. They presented DLE-3D, stands for Deferred Layer Embedding for 3DICs, which find the optimal embedding layers for the nodes of the tree such that cost required by TSVs in tree topology would be minimum. They also presented DME-3D, stands for Deferred Merge Embedding for 3DICs, which was an extension of the algorithm used for merging of segments in 2D for the minimization of global wirelength of the design with the focus on the delay attached by the TSV. In addition, they proposed NN-3D, stands for Nearest Neighbour selection for 3DICs, which was able to built abstract tree topology which was cost effective for the 3DICs, when the abstract topology were not given.

2.2 Gaps in Literature survey

Partitioning of graph is a kind of combinatorial problems whose main aim is to find out a solution which can satisfy all constraints and objective of the problem optimizes. In addition, it is important to know that partitioning of graph comes under the category of NP-hard problems and decisional version of partitioning belongs to NP-complete problems set. Usually in several applications of partitioning, it is sufficient to have an optimal cost feasible solution. As in the result, effective heuristics and approximation algorithm are in practice preferably.

Lots of work has been done in the partitioning of the digital circuits and several algorithms were proposed for the partitioning in two dimensional structure. Day by day, the size and the complexity of circuit increases. And hence, it results in the large size of the chip (die) area. Moreover, generalization of the VLSI design is to be preferred, which can able to solve set of multiple problems by making the use of single chip. Hence, the heterogeneity of the chip comes in demand. To overcome the such kind of several problems of two dimensional circuit. The concept of three dimensional circuit

was introduced. In three dimensional structure, one die (layer) containing some part of whole circuitry will lay on the top of other die (layer) having other part of circuitry in such a way that will satisfy the set of constraints. It lead to the stacked structure of dies. As a consequence, 3D structure helps in the reduction in the area of die, minimization of global wirelength, optimizing the signal delay, reduction in the power consumption of the chip and support heterogeneity of the circuit.

From last few couples of years, researchers are highly interested in the development for the three dimensional design of the circuit. For the partitioning of the digital circuit in 3D space became the new objective for the researchers. For the 3D partitioning, researchers proposed either new algorithms or extend the algorithms which are existed for the partitioning in the 2D space. “hMetis” is the one of the new approaches used to partition the circuit. Many other algorithms for 3D partitioning were proposed which are based on hMetis for the generation of initial set of solutions with various kinds of refinement strategies. Additionally, hMetis is not a layer aware partitioning algorithm. On the other hand, some work made use of algorithms which were developed for 2D partitioning with the various layer assignment techniques. Hence on moving from 2D to 3D physical designing of the digital circuits, objectives of the designing, constraints to be satisfy during designing were changed. New technologies come with new solutions for the previous one but bring its own problems.

Problem Formulation

Consider a random collection of logical circuit netlist and a targeted 3D architecture with fix area and fixed number of tiers. As in figure shown below, generate the partitions such that the total number of 3D-via count is minimized with the area balancing constraint.

The problem can be stated more formally as: consider a graph $G(V, E)$, where V represents vertices of the graph and E represents edges of the graph. i.e. $V = \{v_1, v_2, \dots, v_n\}$ where from v_1 to v_{n-1} are the cells of the netlist having same weight and v_n is a node consisting all I/O pads in common, It is assumed that v_n always lies on bottom most layer and $E = \{e_1, e_2, \dots, e_m\}$ is set of hyper-nets or nets which make connection among the vertices of the graph. Let, K is the number of tiers, such that $K = 2^n$ where $n = 1, 2, 3, \dots$, on which it is required to place the graph after partitioning except most bottom layer which is reserved for I/O pads. The main objective is to minimize inter-tier connections i.e. TSVs as in result total wire-length is minimized without unbalancing the area of the circuit.

The current work proposes a new 3D via driven partitioning method based on the cuckoo search meta-heuristic. In the work, it is considered each egg laid by cuckoo is a solution with the assumption that a cuckoo can lay an egg at a time. The objective is to generate new and better solutions, which can remove old and inefficient solutions. The nest containing best quality of egg, i.e. solution, consider as best nest and selected for the further processing to generate next generation of solutions. A variable P_a is assumed which indicates the probability of finding foreign egg, i.e. cuckoo's egg, by the host bird. Significantly, it shows that solution is not good enough. There may be two cases, if host bird aware of cuckoo's egg, i.e. either it kills the egg means simply discard the solution from solution space or let down it simply and build new nest at some other place means start finding better solution at other location of solution space. Various parts of algorithm are explained as follows:

4.1. Benchmarks Used

The proposed methodology uses IBM ISPD 2004 benchmarks [32] for the testing Three-dimensional Partitioning based on Cuckoo Meta-heuristic (TPCM) algorithm. Each benchmark is consists of three files. That are- 1) a.net file, it contains the details of the all the hyper-edges of the circuit. 2) a.are file, it contains the details about the area occupied by the each element of the netlist. 3) a.netD, it contains the details of the hyper-edges of the netlist.

Header of each netlist contains five entries as follow:

Ignored, #Pins (number of I/O pins), #Nets (number of Nets), #Modules (number of modules), Pad offset.

Each net is connect to either the subset of cells or subset of pads or subset of both cells and pads. Cells are numbered from 0 to pad offset. Whereas pads are numbered from 1 to (number of modules – pad offset - 1). Cells are prefixed by “a”, pads are prefixed by “p”, and the beginning of each net is represented by “s”.

With the help of a sample circuit diagram the file structure of benchmarks can be easily understood.

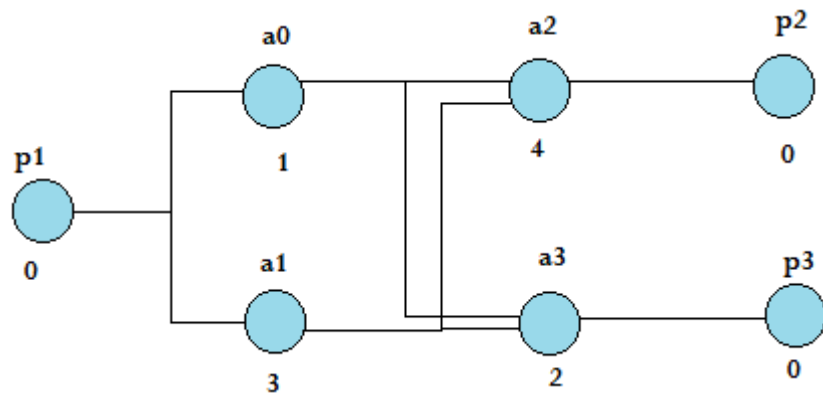


Fig.5. Sample circuit with cells and pads.

In fig 5, circuit contains 4 cells, 5 nets, 3 pads and 13 pins. The .net file for above example as:

```

0
13
5
7
3
p1 s 1
a0 1
a1 1
a0 s 1
a2 1
a3 1
a1 s 1
a2 1
a3 1
a2 s 1
p2 1
a3 s 1
p3 1

```

The .are file is given as:

```

a0 1
a1 3
a2 4
a3 2
p1 0
p2 0
p3 0

```

The .netD file is given as:

```

0
13
4
7
3
p1 s O
a0 1 I
a1 1 I
a0 s O
a2 1 I
a3 1 I
a1 s O
a2 1 I
a3 1 I
a1 s O
p2 1 I
a3 s O
p3 1 I

```

4.2. Solution Representation

Encoding of solution is the permutation of circuit elements contain in the input netlist file. One egg represents a solution that partitioned the given netlist into sub-parts. With help of array data structure the solution is represented as shown in figure 6 below. Array contains a permuted sequence of cells and I/O pad. Elements are assigned to one partition, say 0th, and rest are assigned to next partition, say 1th, with the fulfillment of balancing constraints.

7	3	10	6	1	5	8	2	9	4
0	0	0	0	0	1	1	1	1	1

Fig. 6. Solution representation.

For the validation of the work, the IBM benchmarks have been used to verify proposed algorithm. IBM benchmarks are very large in size. e.g. ibm05 has 28,146 cells which large to store. To reduce this space complexity, a technique is used in which only half of the total cell is stored that belong to one partition and as the remaining cells belong to the second partition and there is no need to store them. If further partitioning is required, remaining half set of elements can be evaluated using set theory operations like set difference operation. For example, if there are ten circuit elements numbered from 1 to 10 and let us suppose above figure shows a solution initially. It is required to

store either of one partitioned elements, not all. In result, it is required only half of the total space for storage of the initial solutions. Solution can be represented by new technique as shown in figure below.

7	3	10	6	1
---	---	----	---	---

Fig. 7. New technique for the initial solution representation

From the above diagram, circuit elements having number 7, 3, 6, 9 and 2 belongs to one partition and other circuit elements i.e. 1, 4, 5, 8 and 10 belongs to other partition and can be generated by set difference operation i.e. universal set minus known half set. For example

Let, there are 10 cells in a netlist numbered from 1 to 10.

After initial bi-partitioning one part may contain elements: 5, 2, 7, 3, 1 (say set A).

Hence, $A = \{5, 2, 7, 3, 1\}$

And there is universal set of 10 elements. i.e.

$U = \{10, 7, 3, 2, 8, 6, 7, 9, 1, 4\}$

So, other part of bi-partition can be evaluate as (say set B)

$B = U - A$. (set difference between U and A).

Note: this storage technique is used only for the initial bi-partitioning.

4.3. Step Size

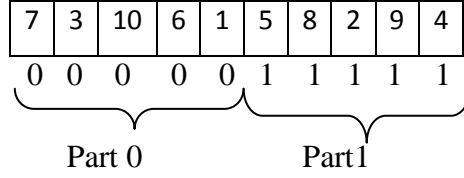
Step size (α) can be defined as distance between the two solutions. The main aim to calculate step size is to find the difference between best solution and current generated solution. Through the step size, current generated solution is moved towards the best solution if and only if the current solution is not better than previous existing best solution. The step size can be expressed as-

$$\alpha = X_j^{(i)} - X_{\text{best}}^{(i)} \quad (1)$$

The distance between two solutions, say m^{th} and n^{th} cuckoos, can be calculated as the difference in the number of elements between any pair of partitions of different

solutions. For the fast processing, the algorithm takes two parts from two solutions having lesser distance between them.

m^{th} cuckoo-



n^{th} cuckoo-

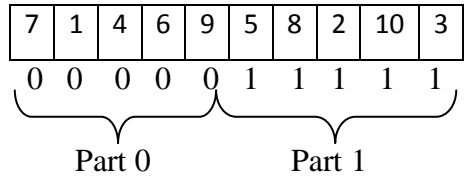


Fig. 8. Representation of n^{th} and m^{th} solutions.

From the above diagram, the distance between two different solutions can be calculated in four ways. i.e.-

- Distance (m^{th} part-0, n^{th} part-0) = 2.
- Distance (m^{th} part-0, n^{th} part-1) = 3.
- Distance (m^{th} part-1, n^{th} part-0) = 3.
- Distance (m^{th} part-1, n^{th} part-1) = 2.

Here, 2 is the minimum distance between two cuckoos or solutions. Any pair from above having distance equal to 2 can be chosen. Any pair having minimum distance value can be selected.

4.4. Updation in Solution

When current solution is updated towards best solution, the current solution is changed. The current solution is updated with α step size and the step length called Lévy flight, drawn from Levi's distribution. The value of Lévy flight equal to one is taken. Through below mentioned expression solution updating scheme can be formulated.

$$x_j^{(i+1)} = x_j^{(i)} + \alpha = x_j^{(i)} + (x_j^{(i)} - x_{\text{best}}^{(i)}) \quad (2)$$

For the random walk around the best solution is done as:

$$x_j^{(i+1)} = x_j^{(i)} + \varepsilon_i \quad (3)$$

Here, ε_i represent a random value within the range of $[0, n/2]$ (where n is the number of cells plus one node for I/O pads) and the value of ε_i is generated from uniform distribution.

4.5. Fitness Calculation for the Solution

Let, x_i represents a solution with n -tuple array of either 0 or 1 value, where n is the number of cells plus one (for the all I/O pads). And let, m_i is the n -tuple array of either 0 or 1 value, such that index containing 1 represent that $(index+1)^{th}$ element of the circuit is connected through that hyper-edge (assuming index starts from 0). Now, calculate P and Q such that

$$P = x_i \text{ bitwise AND } m_i$$

And

$$Q = x_i \text{ ' bitwise AND } m_i$$

If the value of both P and Q evaluated to one for any hyper-edge. It means that hyper-edge will participate for the cut and value of cut variable is increased by one. Otherwise, no increments in cut variable. By evaluating total number of cuts for a solution with all hyper-edges, the fitness value of the solution can be calculated. Here, lower the fitness value mean lesser number of cuts means better will be solution for the partitioning.

```
if (P == 1 and Q == 1)
```

```
    cut = cut + 1;
```

```
else
```

```
    Proceed for next hyper-edge without affecting the value of cut
    variable;
```

```
end_if.
```

4.6. Threshold Calculation

Threshold calculation function has two parameters as argument one is fitness value of best solution and other is value of P_a . It is required to calculate this threshold value for per iteration because fitness value of best pop may differ for each iteration. By using equation number 4, threshold value can be evaluated using following formula

$$\text{Threshold value} = \text{fitness_of_best_solution} [1 + .01 (1 - P_a)] \quad (4)$$

4.7. Selection of Best Solution

Initially, N numbers of solutions are produced by the proposed algorithm. On the basis of their fitness values, best solution among the N solution is selected. Then, steps are further repeated for next generation using the Lévy flight towards the best solution. In the next generation only N-1 new solutions are produced. So, among 2N-1, i.e. N (old solution) plus N-1 (new solution), total solutions N new solutions are selected for the further production of the solutions on the basis of their fitness value. To produce the next generation in the algorithm above explained procedure of best solution selection is repeated every time during each iteration.

4.8. Proposed Algorithm

The pseudo code of TPCM is given in figure 5. TPCM_3D read the input netlist and construct a matrix of size number of nets X number of cells+1 in the given netlist, where 1 is for all the I/O pads contain in the last column of the matrix. Value of probability factor is given by the user such $P_a \in [0, 1]$. Higher the value of P_a greater will be chance of finding foreign egg by host bird it lead to more chances of solution rejection according to their fitness value. Number of layers (say K) are also mentioned by the user such that $K = 2^n$ where $n = 1, 2, 3, \dots$. Using queue data structure, Cuckoo_partitioner is called recursively until the desired number of partitioned produced.

In TPCM, three array structures, pop_start, pop_update and pop_new are created having two fields bit and fit. Bit is used to store bit pattern generated for the solution

TPCM_3D (File_IO , Num_gates, Num_nets, Pop_size, Num_of_gen, P_α , No_of_Layers)**Input:** Read the Benchmark files in circuits, Q-Queue of partitions, A –partition**Output:** Layered Partitions of all the movable objects in netlist**Begin**

```
Set variable Layer:= 1
Netlist=CreateNetlist (File_IO, Num_gates, Num_nets)
ENQUEUE (Q, Netlist)
While (EMPTY(Q) != FALSE)
    (A)← DEQUEUE(Q)
    If (Layer > No.of_Layers) Then
        Output: Resultant Partitions
        Exit While loop
    Else
        (A1,A2)=Cuckoo_partitioner(Netlist, Num_gates, Num_nets, Pop_size,
Num_of_gen, Pα)
        ENQUEUE(Q,A1)
        ENQUEUE(Q,A2)
        Layer:= Layer + 1
    Endif
Endwhile
```

End**Cuckoo_partitioner (Netlist, Num_gates, Num_nets, Pop_size, Num_of_gen, P_α)****Input:** pop_start, pop_update are structure array of size (Num_gates X Pop_size)

pop_new is structure array of size (Num_gates X Pop_size-1)

best_pop is an individual of population of size (Num_gates)

Output: Two Partitions A1 and A2 having min-cut**Begin**

```
[pop_start] = initialise (Pop_size, Num_gates, Num_nets)
Find_Fitness (Netlist, pop_start, Num_gates, Num_nets, Pop_size)
Set iteration: =1
While (iteration <= Num_of_gen)
    [best_pop] = Find_Best (pop_start, Pop_size, Num_gates, Num_nets)
    TR=Calculate_Threshold(best_pop, Pα )
    Set j: = 1
    For i =1 to Pop_size
        If best_pop = pop_start(i)
            i=i+1;
        Endif
        Do
            Step_size = Arc_diff (best_pop, pop_start(i), Num_gates)
            pop_new(j) = Levy_flight (pop_start(i), Num_gates, Step_size )
            net_cut = Find_Fitness_Individual(pop_new(j), Num_gates, Num_nets)
            pop_new(j).fit = net_cut
            While (pop_new(j).fit > TR )
                Set j=j+1
            Endfor
        Endfor
        pop_update = Update_solution (Pop_size, pop_start, pop_new)
        pop_start = pop_update
        iteration = iteration +1
    Endwhile
```

End

Fig. 9. Proposed algorithm for circuit partitioning with min-cut.

and fit is used to store the fitness value of the corresponding solution. Using initialize function Pop_size solutions are generated randomly initially and with the help of fitness function, as explained in section 4.4, number of net cut are evaluated. For better results, a number of iterations are applied to refine the solution. This number of iterations is defined by user. Higher the number of iteration more will chance of getting better result but at the cost of long CPU time. One solution is selected as best solution among the set of initial populations stored in pop_start. Except the best solution, Arc_diff is calculated, as mentioned in section 4.2, to decide the step size for the betterment of corresponding solution. Value of Step_size passed as argument to Levy_flight function. In Levy_flight corresponding solution is moved towards the best solution present currently with the step length of one, step length one is assumed for the fast improvement in the solutions. After the movement in the solution new solution is generated and the corresponding fitness value will be evaluated if that fitness value satisfied the threshold criteria, as mentioned in section 4.5, then the solution will be accepted otherwise whole procedure will be repeated to generate better solution for the next generation. After the Pop_size-1 new solution generation, top Pop_size solutions are selected on the basis of their fitness values for the next iteration.

Finally, when required number of layers generated by the TPCM. Another important task is to assign layers to all partitions one by one. Starting from the sub-partition having last node entry will be assigned to the layer just above the layer containing all I/O pads (bottom most layer). After that sub-partition just separated from the assigned sub-partition will be assigned to the layer above the just assigned layer previously. This procedure will be follow until the all sub-partitions are assigned to all layers of 3D design. This will make proposed algorithm, i.e. TPCM, as layer aware partitioning algorithm and help to reduce the number of 3D vias generated by the cell to I/O pads interconnections. Subsequently, total numbers of TSVs are reduced.

Simulation Results

The proposed algorithm is implemented on Matlab (R2013a) and run on Intel core i5 processor, with 6 GB RAM and 64 bit Operating System. The suggested work is tested against IBM benchmarks and results are compared with meta-genetic method. Table 1 shows the characteristics of IBM benchmarks used during testing process.

Table 1. Characteristics of benchmarks used.

Benchmark Circuit	Cells	Nets	I/O Pads
ibm01	12506	14111	246
ibm02	19342	19584	259
ibm03	22853	27401	283
ibm04	27220	31970	287
ibm05	28146	28446	1201

As shown in table 2, Results generated from the proposed algorithm is better than the results produced by meta-genetic approach in both ways i.e. number of TSVs and CPU time required to calculate results. The average CPU time out of the 100 runs of the algorithm is taken. There is not much more difference in the number of TSVs between meta-genetic and Three –dimensional Partitioner based on Cuckoo Meta-heuristic (TPCM) for the two layer partitioning. But as increase in the number of layers of the 3D-design the difference between the numbers of TSVs increases significantly. This shows that for the large architectural design TPCM works better to generate min-cuts.

Table 2. Comparison of cuts and CPU time (in best, average and worst case) for proposed algorithm and meta-genetic approach (pop_size is taken as 10).

IBM Circuits	Number of layers	Meta -Genetic Algorithm		Cuckoo Algorithm			
		No. of TSVs	CPU time (in secs)	No. of TSVs	Best case	Average case	Worst case
lbm01	2	9061	438.38	8968	420.84	428.53	445.50
lbm02		13109	896.76	13105	781.409	802.9298	818.7835
lbm03		17217	1.5501e+03	17142	1.4986e+03	1.5401e+03	1.5753e+03
lbm04		20300	2.2089e+03	20257	2.1077e+03	2.1547e+03	2.2351e+03
lbm05		18500	1.8636e+03	18423	1.8071e+03	1.8323e+03	1.8435e+03
lbm01	4	24075	605.50	23942	424.7283	433.0015	447.4109
lbm02		36879	1.1809e+03	33058	842.6394	857.0507	867.8648
lbm03		47903	2.2874e+03	37011	1.5221e+03	1.6080e+03	1.660e+03
lbm04		57031	3.0625e+03	40247	2.0796e+03	2.16187e+03	2.2393e+03
lbm05		52134	3.0127e+03	38272	1.8153e+03	1.8419e+03	1.8960e+03
lbm01	8	50929	1.0514e+03	41936	428.1291	436.9978	450.5389
lbm02		80017	2.1366e+03	72048	848.5830	873.6172	924.1031
lbm03		92655	3.7941e+03	77117	1.5285e+03	1.6195e+03	1.6662e+03
lbm04		109834	4.8471e+03	80226	2.1630e+03	2.2412e+03	2.3175e+03
lbm05		103336	4.3732e+03	78370	1.8060e+03	1.8533e+03	1.9028e+03

In table 2, the algorithm generates different CPU time (in seconds) for every trial although for the same benchmark as well as same value of K (numbers of tiers) because of the reason that value of arc different is not same for every iteration. If arc difference has large value CPU time will be high and vice versa.

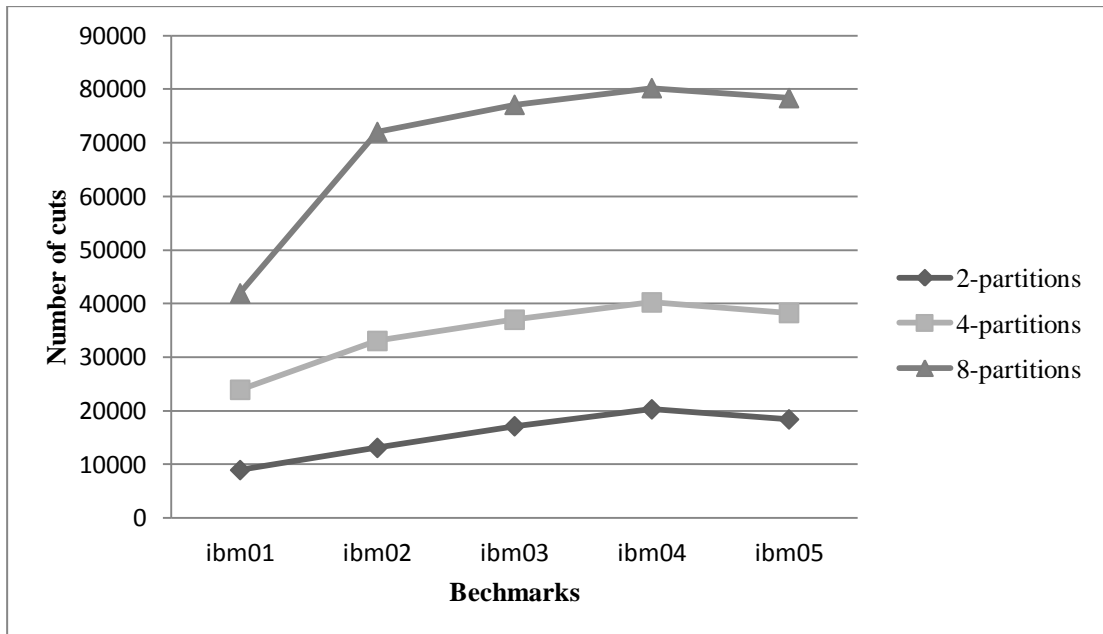


Fig. 10. Shows the value of cuts for different IBM benchmarks for different levels of partitions.

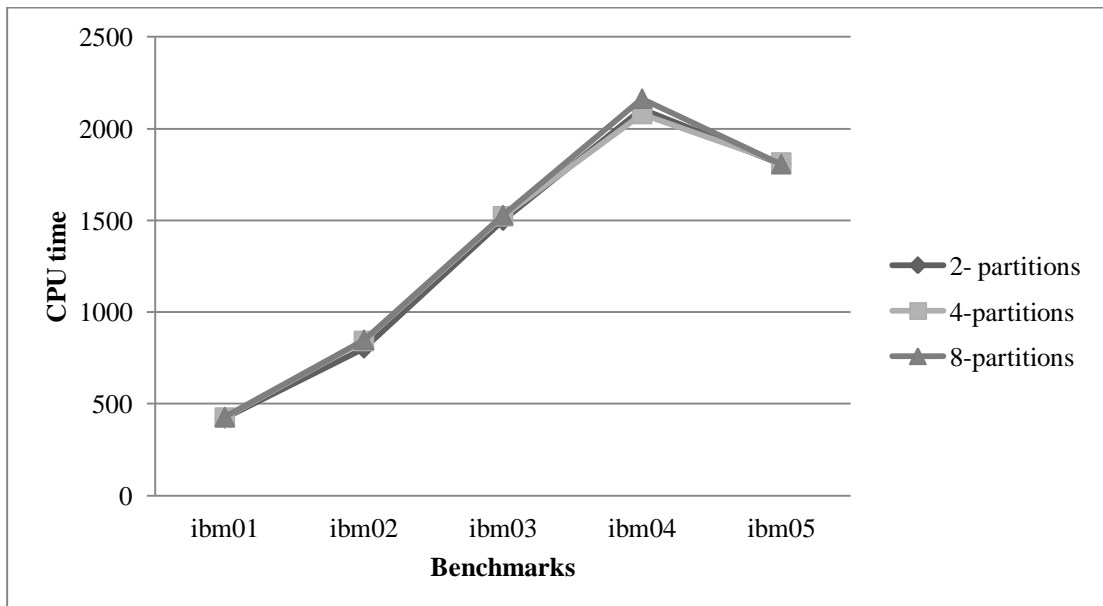


Fig. 11. Shows the average CPU time for different IBM benchmarks for different levels of partitioning.

Figure 12 demonstrate the correlation between the probability of finding foreign egg (P_{α}) and required CPU time to run at corresponding values of P_{α} . As the value of P_{α} increases CPU time is also increase because of the reason that at high value of P_{α} there is more chances of worst solution rejection. As in result, more solution generation is required in the place of rejected solutions.

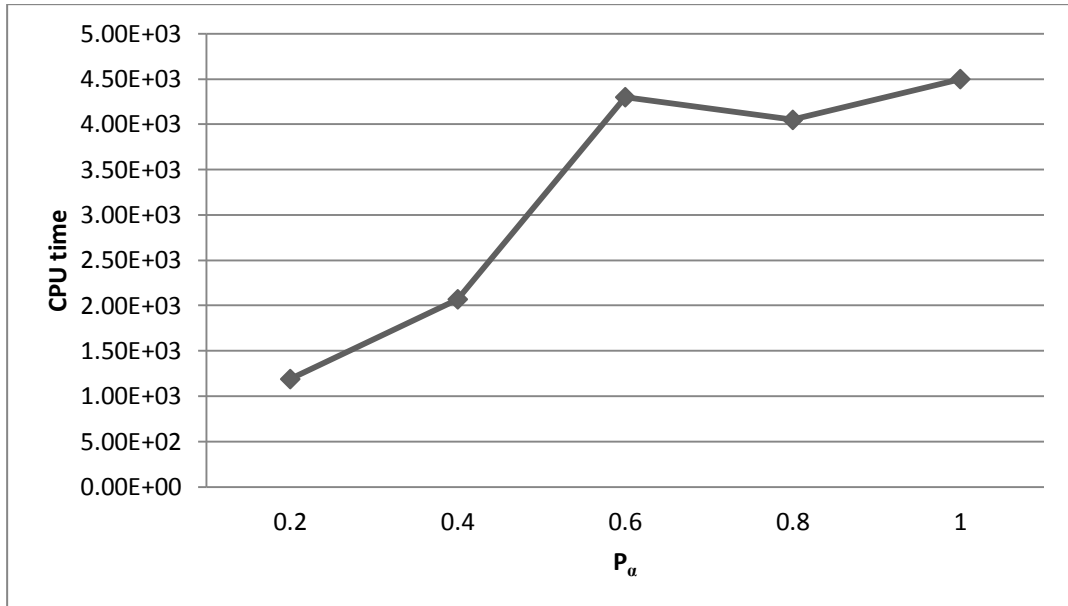


Fig. 12. Correlation between P_{α} and CPU time.

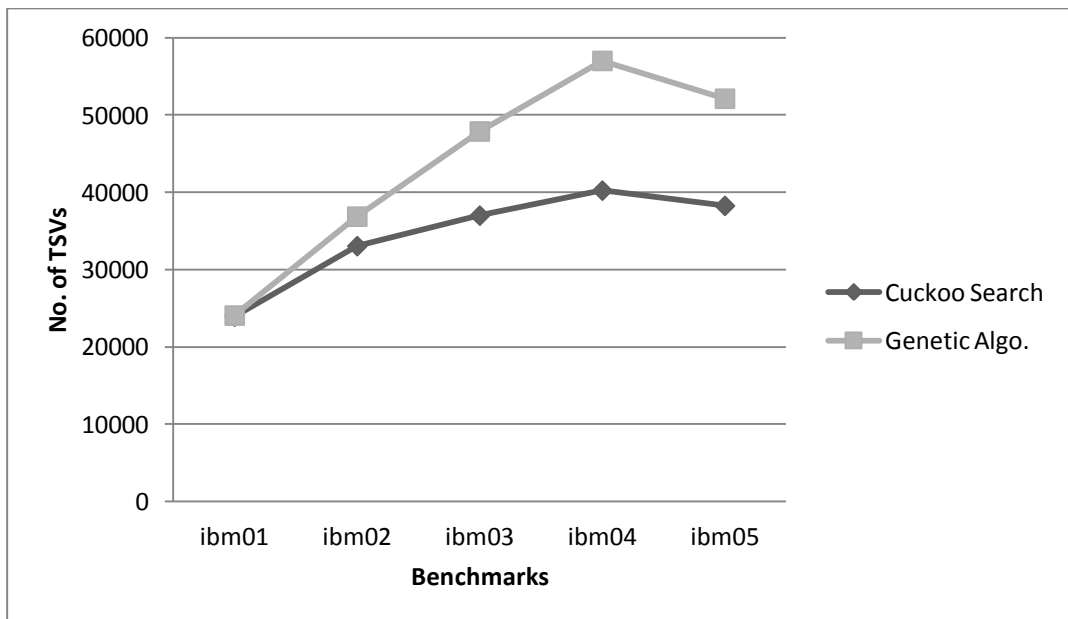


Fig. 13. Comparison between number of TSVs generated by Cuckoo Search and Genetic Algorithm for the 4 layer-partitioning.

Fig. 13 shows the comparative study for the number of TSVs produced by both TPCM and Meta-Genetic Algorithm. This shows that TPCM always produced lesser number of TSVs as compared to the Meta-Genetic Algorithm.

Hence, it can be easily seen that TPCM will produce better optimization results for the three dimensional circuit partitioning.

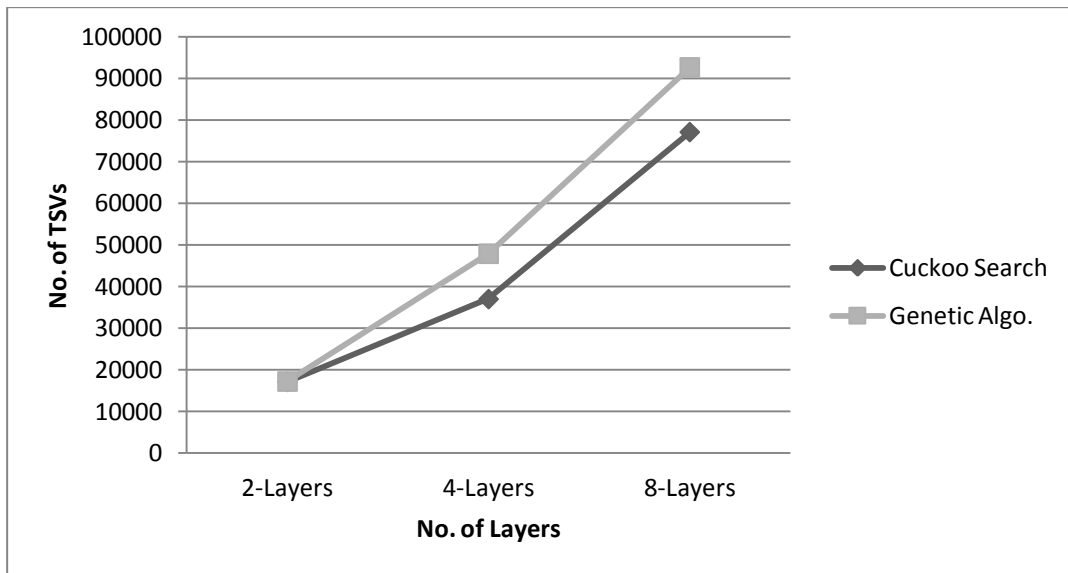


Fig. 14. Shows the relation between number of TSVs and Number of layers.

In above fig, it can be concluded that as the number of layers for the partitioning increases the result quality of TPCM will become far better as compared to the result quality of the Meta-genetic algorithm.

Conclusion and Future Work

Because of growth in the size of digital electronic circuit, 2D VLSI physical design process is moving towards 3D space. 3D technology helps to overcome the issues emerges under the 2D technology like increasing in the size of chip, uneasiness in the integration of heterogeneous technology, lager global wire-length, inefficient signal delay etc. In proposed work, a new algorithm named as TPCM is introduced which is based on the cuckoo search meta-heuristic, for the partitioning of netlist in such a way so that number of inter-partition connections (TSVs) remains minimum with the area balancing constraint. Results of TPCM are tested on some standard netlist of IBM benchmarks and further it was compared with the results of Meta-genetic algorithm. Comparison shows that TPCM always produced better output in terms of number of TSVs and CPU time. Finally, it can be concluded that the proposed algorithm is able to improve the results for the problem in hand and will be able to generate good results for the larger design in acceptable CPU run-time, which can serve for the further tradeoff as good starting point.

In future, the work may be extended with the consideration of TSV capacitance and timing characteristics. Work can be improved for the variable number of tiers of the 3D design. Different variants of hybrid evolutionary approaches can be applied on the 3D VLSI design problem to reduce the computation complexity and optimize the result.

References

- [1] P. S. Andry, et al., “A CMOS-compatible processor for fabricating electrical through-vias in silicon”, *Proceedings of the 56th Electronic Component and Technology Conference*, San Diego, 2006, pp. 831-837.
- [2] S. Spiesshoefer, Z. Rahman, G. Vangara, S. Polamreddy, S. Burkett, L. Schaper, “Process Integration for Through-Silicon Vias”, *Journal of Vacuum Science and Technology A*, vol 23, pp. 824-829, 2005.
- [3] K. Takahashi, M. Sekiguchi, “Through Silicon Via and 3-D Wafer/Chip Stacking Technology”, *Symposium on VLSI Circuit Digest of Technical Papers*, 2006, pp. 89-92.
- [4] P. Zhou, Y. Ma, Z. Li, R.P. Dick, L. Shang, H. Zhou, X. Hong, Q. Zhou, “3D-STAF: Scalable Temperature and Leakage Aware Floorplanning for Three-Dimensional Integrated Circuits ”, *International Conference on Computer-Aided Design*, 2007, pp. 590-597.
- [5] I. Loi, S. Mitra, T.H. Lee, S. Fujita, L. Benini, “A low-overhead fault tolerance scheme for TSV-based 3D network on chip links”, *International Conference on Computer-Aided Design*, 2008, pp. 598-602.
- [6] J. Cong, J. Wei, “A Thermal-Driven Floorplan for 3D-ICs”, *International Conference on Computer-Aided Design*, 2006, pp. 306-313.
- [7] Z. Li, X. Hong, Q. Zhou, S. Zeng, J. Bian, H. Yang, V. Pitchumani, C.K. Cheng, “Integrating dynamic thermal via planning with 3D floorplanning algorithm”, *International Symposium on Physical Design*, 2006, pp. 178-185.
- [8] N.A. Sherwani, “Partitioning”, in *Algorithm for VLSI physical design automation*, 3rd ed. New York, USA: Kluwer, 2002, ch. 5, pp. 157-187.
- [9] A.B. Kahng, J. Lienig, I.L. Markov, J. Hu, “Global and Detailed Placement”, in *VLSI Physical Design: From Graph Partitioning to Timing Closure*, Netherland: Springer, 2011, ch.4, pp. 95-126.

- [10] M. Pathak, Y.J. Lee, T. Moon, S.K. Lim, "Through-silicon-via management during 3D physical design: When to add and how many?", in *IEEE/ACM International conference on Computer-Aided Design (ICCAD)*, San Jose, 2010, pp. 387-394.
- [11] K. Banerjee, S.J. Souri, P. Kapur, K.C. Saraswat, "3-D ICs: a novel chip design for improving deep-submicrometer interconnect performance and systems-on-chip integration", in *Proceedings of the IEEE*, vol 89, issue 5, pp. 602-633, 2001.
- [12] S.M. Sait, H. Youssef, "Circuit Partitioning", in *VLSI Physical Design Automation: Theory and Practice*, McGraw Hill, 1995, ch. 2, pp. 41-90.
- [13] C.J. Alpert, J.H. Huang, A.B. Kahng, "Multilevel Circuit Partitioning", in *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems*, vol 17, pp. 655-667, 1998.
- [14] J. Cohoon, J. Karro, J. Lienig, "Evolutionary Algorithm for the Physical Design of VLSI Circuits", in *Journal of Springer on Advances in Evolutionary Computing*, Berlin, pp. 683-711, 2003.
- [15] B.W. Kernighan, S. Lin, "An Efficient Heuristic Procedure for Partitioning Graphs", in *Bell System Technical Journal*, vol 49, pp. 291-307, 1970.
- [16] X.S. Yang, S. Deb, "Engineering Optimisation by Cuckoo Search", in *International Journal Mathematical of Modelling and Numerical Optimisation*, pp. 330-343, 2010.
- [17] X. Yang, S. Deb, "Cuckoo search via Levy Flights", in *Proc. of World Congress on Nature and Biologically Inspired Computing*, India, 2009, pp. 210-214.
- [18] A. Singh, M. Kaur, "A Cuckoo based Approach for Solving Standard Cell Placement Problem", in *Journal of VLSI Design Tool and Technology*, vol 4, issue 2, 2014.
- [19] Y.S. Huang, Y.H. Liu, J.D. Huang, "Layer-Aware Design Partitioning for Vertical Interconnect Minimization", in *IEEE Computer Society Annual Symposium on VLSI*, Chennai, 2011, pp. 144-149.

- [20] P. Ghosal, S. Chatterjee, "Partitioning in 3D ICs: A TSV aware strategy with area balancing", in *International Conference on Devices, Circuit and Systems (ICDCS)*, Coimbatore, 2012, pp. 576-580.
- [21] S. Sawicki, G. Wilke, M. Johann, R. Reis, "3D-Via Driven Partitioning for 3D VLSI Integrated Circuits", in *CLEI Electronic Journal*, vol 13, 2010.
- [22] Y.C. Hu, Y.L. Chung, M.C. Chi, "A multilevel multilayer partitioning algorithm for three dimensional integrated circuits", in *11th International Symposium on Quality Electronic Design (ISQED)*, San Jose, 2010, pp. 483-487.
- [23] Y.H. Chen, Y.T. Chen, J.D. Huang, "Parallel Layer-Aware Partitioning for 3D Designs", in *Proceedings Synthesis And System Integration of Mixed Information technology (SASIMI)*, 2013. pp. 176-179.
- [24] M.C. Tsai, T.T. Hwang, "A Study on the Trade-off among Wirelength, Number of TSV and Placement with Different Size of TSV", in *International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, Hsinchu, 2011, pp. 1-4.
- [25] C. Sechen, D. Chen, "An Improved objective function for mincut circuit partitioning", in *IEEE International Conference on Computer-Aided Design (ICCAD)*, Santa Clara, 1988, pp. 502-505.
- [26] A.E. Caldwell, A.B. Kahng, I.L. Markov, "Iterative Partitioning with Varying Node Weights", *VLSI Design*, 2000, vol 11, Issue 3, pp. 249-258.
- [27] C. Ababei, Y. Feng, B. Goplen, H. Mogal, T. Zhang, K. Bazargan, S.S. Sapanekar, "Placement and Routing in 3D integrated Circuits," in *IEEE Design & Test of Computers*, 2005, Vol 22, No. 6, pp. 520-531.
- [28] P.K. Sharma, M. Kaur, "A Discrete FireFly Algorithm for VLSI Circuit Partitioning," in *International Conference on Electronics and Communication Systems (ICECS)*, Coimbatore, 2014, pp. 1-4.
- [29] R. Hentschke, S. Sawicki, M. Johann, R. Reis, "An Algorithm for I/O Partitioning Targeting 3D Circuits and its Impact on 3D-Vias," in *IFIP International Conference on Very Large Scale Integration*, Nice, 2006, pp. 128-133.

- [30] R. Banos, C. Gill, M.G. Montoya, J. Ortega, "A Parallel evolutionary algorithm for circuit partitioning," in *Proceedings Eleventh Euromicro on Parallel, Distributed and Network-Based Processing*, Genova, 2003, pp. 365-371.
- [31] S.S. Gill, R. Chandel, A. Chandel, "Genetic Algorithm Based Approach To Circuit Partitioning," in *International Journal of Computer and Electrical Engineering*, Vol 2, No. 2, pp. 196-202, 2010.
- [32] <http://vlsicad.ucsd.edu/UCLAWeb/cheese/ispd98.html>
- [33] J. Ouyang et al., "Arithmetic Unit Design Using 180nm TSV-based 3D Stacking Technology", in *IEEE International Conference on 3D System Integration*, San Francisco, 2009, pp. 1-4
- [34] E.J. Marinissen, J. Verbree, M. Konijnenburg, "A structured and scalable Test Access Architecture for TSV-Based 3D Stacked ICs", in *28th VLSI Test Symposium (VTS)*, Santa Cruz, 2010, pp. 269-0167.
- [35] H.S. Ye, M.C. Chi, S.H. Huang, "A Design Partitioning Algorithm for Three Dimensional Integrated Circuits", in *International Symposium on Computer Communication Control and Automation (3CA)*, Tainan, 2010, pp. 229-232.
- [36] S.K. Lim, "TSV-based 3D Physical Design Tool Needs for Faster Mainstream Acceptance of 3D ICs", in *Proc. 47th ACM/IEEE DAC Knowl. Center Article*, 2010, pp. 1-11.
- [37] T.Y. Kim, T. Kim, "Clock Tree Synthesis for TSV-Based 3D IC Designs", in *Proc. 47th ACM/IEEE Design Automation Conference*, Vol 16, Issue 4, pp. 723-728, 2010.

List of Publication and Video Link

- [1] Parashar, A., Kaur, M. (2015) Partitioning in Three Dimensional Integrated Circuits: A Cuckoo based Meta heuristic Approach. Paper (Communicated)
- [2] Parashar, A., Kaur, M. (2015) Various approaches for Three Dimensional Partitioning in VLSI physical design : A Glance. Paper (Communicated)
- [3] <https://www.youtube.com/watch?v=qypa0vItOXg> (Video Link)