

Received: 3 July, 2020; Accepted: 26 Feb., 2021; Published: 9 April, 2021

# GPU Implementation of Thermal Aware 3D IC Floorplanning

JEYA PRAKASH KADAMBARAJAN<sup>1</sup>, SIVAKUMAR POTHIRAJ<sup>2</sup> and PANDIARAJ KADARKARAI<sup>3</sup>

<sup>1</sup> Department of Electronics and Communication Engineering, Kalasalingam Academy of Research and Education, Anand Nagar, Krishnankoil 626126, Tamilnadu, India.  
*jeyaprakash.kadambarajan@gmail.com*

<sup>2</sup> Department of Electronics and Communication Engineering, Kalasalingam Academy of Research and Education, Anand Nagar, Krishnankoil 626126, Tamilnadu, India.  
*siva@klu.ac.in*

<sup>3</sup> Department of Electronics and Communication Engineering, Kalasalingam Academy of Research and Education, Anand Nagar, Krishnankoil 626126, Tamilnadu, India.  
*pandian.raj.@gmail.com*

**Abstract:** Several academic/commercial types of research are being seen in a much-paced manner concerning the optimizing the 3-D ICs with the techniques of FP-Floor Planning by the ever-increasing human needs along with the advancements that we are exposed these days. In general, FP-Floor Planning was nothing but the process of optimizing any appropriate parameters in the taken 3-D ICs against the suitable applications. It was usually done whenever there were huge counts of spares inside the IC in every layer present interiorly. Thus, this paper deals with the challenges of optimization in 3-D ICs by optimizing the TSV count, cross-sectional area, and wire length and temperature with shuffled frog leaping, MFO-Moth Flame Optimization, and MVO-Multi-Verse Optimization FP-Floor Planning methodologies, respectively. For any optimizing need arising in ICs, there will be some parameters or additional features that will be added to the circuit to facilitate the optimization. However, there might be some delay in the tasks depending on which the floor planning kind of optimization is done by optimization of several attributes like space, temperature, etc. This delay could be caused primarily because of operating the simulation operation with the CPU-Central Processing Unit rather than with GPUs-Graphical Processing Units. So, the usage of GPUs in any computational process could put forth a drastic time consumption (i.e.) running any tasks in CPU alone consumes times but using the GPUs along with the CPU utilization makes a huge time-related contribution to the task which is being executed. So, furthermore, the run time will be reduced if the introduction of GPUs-Graphical Processing Units takes place. Performance evaluation is made for the temperature and run/execution time specifically to prove the effectiveness of this proposed method improvisation in 3-D ICs.

**Keywords:** 3-D ICs, Floor Planning, Temperature Optimization, Run time Optimization, GPU Programming, Moth-Flame Optimization, Hybrid Multiverse Optimizer.

## I. Introduction

Usage of 3D-ICs-Three-Dimensional Integrated kind of Circuits has increased day by day because of the technological advancements that we are encountering these

days. Concerning these ICs, many works have been reported in the literature which will be discussed in brief in this part. Monolithic kind of 3D ICs was employed and investigated by [1] to investigate the temperature based characteristics about the ICs. They primarily made use of 90° thermal attachments found in those ICs with the employment of floor planning base. Their outcomes were prominent by giving rise to reduced peak temperatures in the ICs when subjected to a variety of applications.

A TSV-Through-Silicon Via dependent 3D-ICs was used by [2] to analyze and improvise the performance of it in terms of an overall length of the wire that was enabling the interconnections between any two components, designated allowance made towards the failures, and thermal aware systemic approaches. This work also stands out by avoiding the usual application of duplicate TSVs to improvise the thermal characteristics and failures allowance made in the ICs.

The structure level planning of triple-layered 3-D ICs was done by [3] with the deployment of sizes of irregular dies. They were determined to work with top/bottom layers in 3-D ICs concerning the taken-up structure level planning. As far as the planning arrangement was concerned, they made use of splitting and layering methodologies. Basic form/arrangement of 3-D ICs with triple layers from [3] is indicated in the below figure 1.

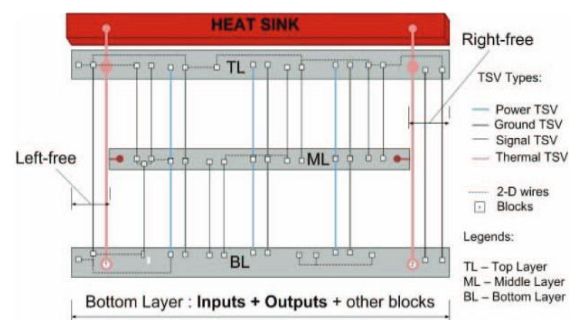


Figure 1. Typical arrangement of 3-D ICs with triple layers [3]

Another instant of work with FP was undertaken by [4] with the deployment of ICs for improvising the area of the chip, Number of TSV, and Overall length of the wire intermediary to the two different spares in the ICs. The changes in all three will put forth a change in the periods of the operations getting carried out in the CPU core.

An arbitrary styled bonding in 3-D ICs was investigated by [5] to make a successful floor planning. They were keen to designate the modules to the respective layers in the ICs against every variety of styled bonding. This devised methodology was formulated based on the cosine relation for the determination of bins, which made this methodology, an analytic method. They used datasets from IBM to cross verify the performance of the floor plan made.

To be specific more and more research has been reported these days because of the industrial revolution, in which ICs play a crucial and dominating role. IC's role was because many technological advancements command the needs of VLSI, optimization, and so on to improvise the performance of ICs by many adaptations in it like floor planning specifically. A few kinds of literature will be discussed now to compare and correlate the already reported works.

The impact of TSVs downsizing and later nanotechnology engineered CMOS scaling was investigated by [6] with factors like delays in the interconnects and power in 3-D ICs. Routing and TSV dependent floor panning operation was conducted by [7] with special reference to the optimizing the length of the wire in the ICs. Then, scale level temperature improvisation was taken up by [8] with the help of FP with the Harmony Exploration and PSO-Particle Swarm Optimizing methodologies. Similarly, [9] used hybridized PSO and CS-Cuckoo Search FP operation. PSO and Firefly methodologies were utilized in combination in [10] concerning the optimizing issues with the help of FP by VLSI of the non-slicing type. [11] utilized GA-Genetic Algorithm and PSO to optimize the temperature by VLSI FP of the non-slicing type. [12] made the extensive investigation of the 3-dimensional ICs by taking both the mechanical and thermal generated stresses in it for taking the sake of improvising the design of 3-dimensional ICs. MLC-Modified Corner List indications were used by [13] for the FP of ICs of the non-slicing type. [14] carried out the FP of ICs concerning the parameters like TSVs and delay period currently in the ICs under consideration. Three Dimensional ICs were appropriately assigned to layers by following the portioning operation while the designing of the ICs. This work made use of computed annealing and tabu explore. Partitioning of dies in ICs was done by [15] with the help of temperature optimized FP.

In recent years, the development of graphics hardware consists of three major factors based on commodity PCs. The computational power of GPUs was grown much faster compared to CPUs for commodity PC hardware. Then the higher performance presents in good cost and performance ratio. GPU is the parallel streaming processor that is used for the fast processing of large arrays. There are many research utilized this graphics processor for enhancing their work performances. Additionally, the 3D graphics processing and modern GPUs programmed for general computation. It consists of the highest number of "shader processors" and operates as a Single Instruction Multiple Data (SIMD) and Multiple Instruction Multiple Data (MIMD) processor. The modern GPU contains hundreds of stream processors with minimum cost [16].

The GPU involves faster graphics manipulations than the general purpose of CPU that the processor specially designed to handle specific primitive operations. This processor involves a parallel operation with each other. Here, the vertex processor estimates the 3D view and shader processor draw this model before it displayed. The number of instruction-presents based on the type of GPU's. To use the maximum number of operations, the program requires to break into suitably sized units, that may influence the performance.

## II Background

For any optimizing need arising in ICs, there will be some parameters or additional features that will be added to the circuit to facilitate the optimization. However, there might be some delay in the tasks depending on which the floor planning improvisations are done by optimization of several attributes like space, temperature, etc. This delay could be caused primarily because of operating the simulation operation with the CPU-Central Processing Unit rather than with GPUs-Graphical Processing Units. So, the usage of GPUs in any computational process could put forth a drastic time consumption(i.e.) running any tasks in CPU alone consumes times but using the GPUs along with the CPU utilization makes a huge time-related contribution to the particular task which is being executed. Along this way, by using the GPUs, [17] made efforts to determine the SER-Soft Error Rates in the VLSI-Very Large-Scale Integrated circuits for the sake of scaling as many as components compactly. SCEDs-Single-Chip Edge Devices have been suggested under the work of [18], which aimed to virtualize the SCEDs to have access over its abilities. For good control over those devices' abilities, they found using the DGIC-Domain General application Integrated Circuit. The outcomes were more fruitful than the methodologies making use of both the CPU and GPUs for the task execution. [19] found it useful for accelerating the SAT-Satisfiability encoder cum solver with the deployment of GPUs while the task execution. This work made use of one of the most common GPU 'NVIDIA'.

Likewise, larger scaled DC/AC systems combining both the LCC-Line Commutated Converter and Modular Multi-level Converter (MMC) was put frothed by [20] with the comparative investigation with GPUs and CPU. Many optimization issues were sorted out smoothly with the usage of GPUs integrated with the main CPU because of its ability to lessen the time. Thus, [21] tended to sort out the peak-cut complex improvisation issues with the help of computational methodology by using the GPUs in it.

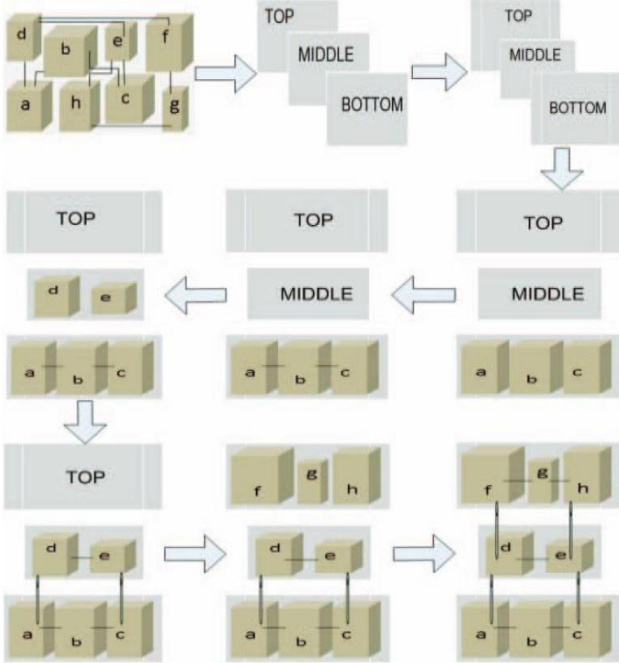
Though many kinds of literature have addressed many optimization challenges, it is still a question to execute those operations within the stipulated run time with utmost accuracy and contribution. Therefore, this work is being proposed to execute all those optimizations with minimum possible run time with the introduction of GPUs through which the implementation of optimization methodologies is further optimized to the core. The main aimed objectives of this work are as follows:

1. To optimize the counts of TSVs through frog leaping Algorithm
2. To optimize the area by using the MFO

3. To optimize the length of the wire and temperature conflicts in 3-D ICs by using Hybrid MVO dependent floor planning methodology
4. To reduce the run time with the introduction of GPUs.
5. To make the comprehensive comparative study with temperature and run time.

### III Proposed Work

For instance, the flow of the floor planning could be comprehended better by indicating the below figure 2.

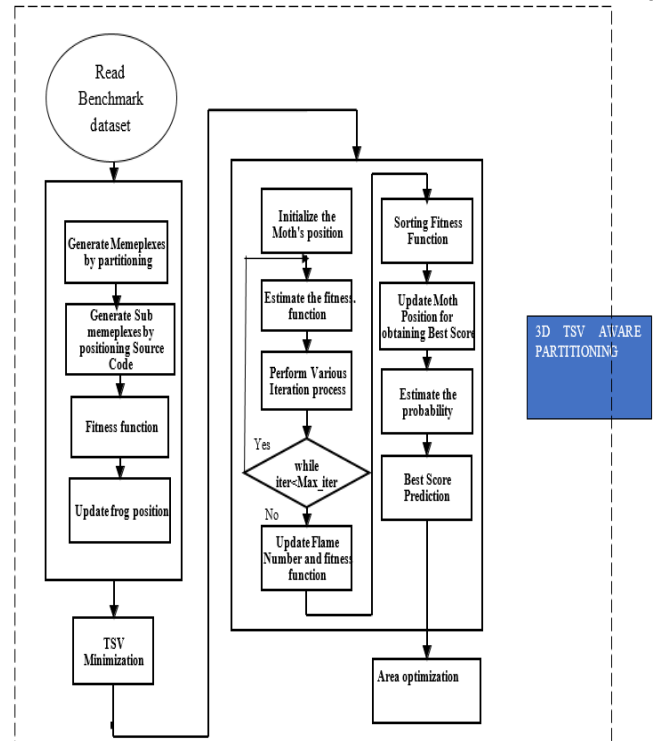
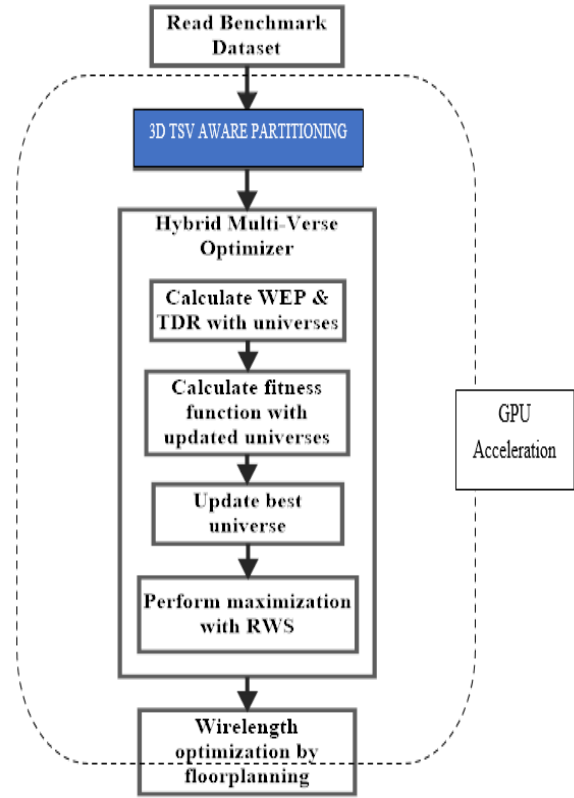


**Figure 2.** Principle of floor panning in ICs with 3 layers [3]

The overall flow diagram of the proposed way of floor panning the 3-D is shown in the below figure. 3. First, we select the benchmark dataset based on our needs. With partition, memplexes are produced and with the layer positioning of origin nodes, sub-memplexes are produced in each layer. Afterward, fitness relations will be estimated. Then, updating of frog location/position takes place after fitness relation estimation, thereby acquiring the reduced and improvised  $TSV_{optim}$ .

This reduced outcome is achieved with the utilization of Enhanced shuffled Frog Leaping Algorithm. As stated in [25] by Pothiraj S. et.al, the enhanced shuffled frog leaping algorithm optimizes the TSV count by means of local search and global information exchange.

Then, the area optimization will be taken up for minimizing the area to benefit from the fruitful results. Hence, MFO-Moth Flame Optimization is appropriately used to optimize the area by regular updates to the fitness relation  $A_{bestscore}$ . As the last step, we tend to optimize the temperature and length of the wire in the various components present inside the 3-D ICs by Hybrid Multi-Verse Optimizer enabled floor planning methodology. Here, the universes will be updated continuously until obtaining the best universe by shifting objects in every universe present in our consideration.



**Figure 3.** Overall Schema for the GPU implementation of 3D IC Floor Panning

The proposed flow is indicated in the above figure 3 which will be discussed in detail in the forthcoming section.

#### A. TSV based partitioning with frog leaping methodology:

A memetic meta-heuristic frog-leaping algorithm is recognized to solve combinatorial optimization problems. It is a population-based cooperative search metaphor enthused with the help of natural memetics. The local search and global information exchange are the elements present in the

algorithm and it contains a set of the interacting virtual population of frogs separated into diverse memplexes.

Based on the partition, the count of TSV is determined. The benefit of using this algorithm is to minimize the TSV count by reducing the components at the layer. If the TSV count is fewer than the previous one update the new one instead of the old one.  $TSV_{optim} = TSV_{cnt}$ . If TSV count is not satisfied with our condition the new position of the worst frog is calculated in each *submemplex (source node position)*.

$$TSV_{optim} = TSV_{cnt} \quad (1)$$

Calculate the new position of worst frog in each *submemplex (source node position)*

$$LP^i = \{S_N^j, \sum LP^i > Th\} \quad (2)$$

### B. Optimizing the area in 3-D ICs using MFO-Moth Flame Optimization

Moth-Flame Optimization (MFO) algorithm is proposed to optimize the area in partitioning. An important inspiration of this optimizer is the navigation method of moths in nature named transverse orientation. Concerning the moon, the fixed angle is maintained for the moth fly in the night, and an efficient approach to travel for a longer distance. However, these fancy insects are surrounded in an unusable/deadly spiral path around artificial lights.

In our proposed moth flame optimization method, the lower and upper bound is initialized for minimizing TSV count and area. This optimization algorithm is utilized for attaining an effective solution with the fitness relation,  $Area_{est} = \sum_{k=1}^3 \sum_{i=1}^c size(LP_k^i)$  and the probability is also estimated finally for the best score of area prediction. The moth flame optimization algorithm is utilized in our work to minimize the area.

*Best score prediction,*

$$A_{bestscore} = Fit_{val} \quad \text{if } Pr > 0.8$$

### C. Hybrid MVO-Multi-Verse Optimizer - Wirelength and temperature optimization-based floor planning

As stated in [25], the multi verse optimizer uses the black hole and white hole ideas to explore the search space for the optimized floorplanning solution. In the work, it has been shown the algorithm is efficient in floorplanning optimization of 3D ICs.

The common steps of MVO are given as,

1. Initialize the optimization process.
2. Generate a group of random variables for the optimization process.
3. Rise higher inflation rate of objects to shifting to universes with the lower inflation rate.
4. Each object in the universe moves arbitrarily to the best universe irrespective of the inflation rate.

The general working of MVO can be comprehended better by referring to the below-depicted figure 4.

The base equations corresponding to the optimization by MVO.

$$U(i, j) = BU + \left( TDR * \left( ((ub - lb) * ub) \right) \right) \quad \text{if } ran_3 < 0.5$$

$$U(i, j) = BU - \left( TDR * \left( ((ub - lb) * lb) \right) \right) \quad \text{if } ran_3 < 0.5$$

Where  $U(i, j)$  - the best universe in  $j$ th parameter recognized till now, TDR - travelling distance rate, BU – best universe and  $lb, ub$  - lower and upper bounds of  $j$ th variable.

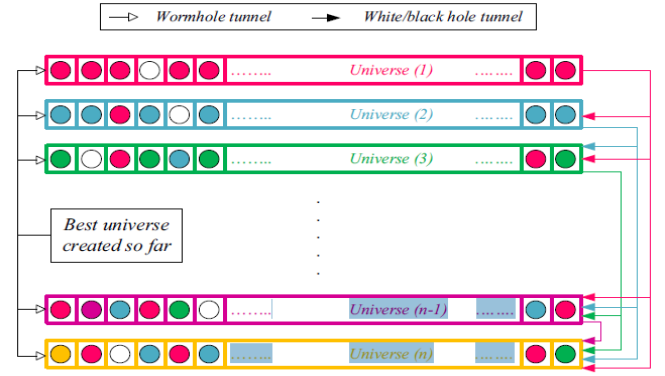


Figure 4. Working of MVO

$$T \rightarrow \sum_{b=0}^{n-1} \frac{P_a P_b}{(x_a - x_b)^2 + (y_a - y_b)^2 + (z_a - z_b)^2} \quad (3)$$

The smaller distance between two blocks provides the highest impact on temperature. Similarly, the highest power consumption of two adjacent blocks provides greater impacts on temperature. In MVO work, the objective function is directly based on temperature.

where  $P_a$  and  $P_b$  indicate the rate of power consumed by the assumed blocks, say block 'a' and  $b^{\text{th}}$  block that is nearer to the block 'a' correspondingly. These 2 blocks assumed will be indicating the influence of the thermal/temperature coordinates on the ICs. Coordinates of  $x$ ,  $y$ , and  $z$  concerning the block 'a' center can be represented by  $x_a$ ,  $x_b$ , and  $x_c$  correspondingly. Likewise,  $x_a$ ,  $x_b$ , and  $x_c$  indicate the  $x$ ,  $y$ , and  $z$  axes for the block 'b' center correspondingly. Each predefined block has higher or else a lesser influence over temperature.

### D. Acceleration of optimization operation with the introduction of GPU

All the optimization operation is fed to the simulation again but this time with the introduction of GPUs because of the flexibility that the GPUs impart towards any task by releasing the memory overloaded in the CPU by its introduction. The specifications of the PC concerning CPU and GPU are indicated in the below table 1. The optimizing operation starts with the CPU, which feeds the data arrangements and kernel code to the GPUs, initializing each simulation operation. Design data of the circuit comprises the type and source info for every gate and then, the gates in the circuit are arranged based on the order of topological type. Intermediary faults information-comprises feasibilities in terms of beginning period, pulse width, logical, and polarity offset in every SET-Single Event Transient.

Item	value
OS Type	Microsoft Windows 8.1 Pro N
Built number	6.3.9600 Build 9600
Manufacturer of OS	Microsoft Corporation
System kind	x64-based PC
Processor configuration	Intel(R) Core(TM) i3-7100 CPU @ 3.90GHz, 3900 Mhz, 2 Core(s), 4 Logical Processor(s)
BIOS edition with Date	American Megatrends Inc. 3805, 07-05-2018
SMBIOS	Version 3.0
Embedded	255.255

Controller edition	
BIOS Mode	Legacy
Manufacturer of Base Board	ASUSTeK COMPUTER INC.
Role of Platform	Desktop
State of Secure Boot	Unsupported
Location	India
Zone of Time	India Standard Time
Total Physical Memory	7.94 GB
Installed Physical Memory (RAM)	8.00 GB
Available Physical Memory	5.77 GB
Available Virtual Memory	22.6 GB
Total Virtual Memory	25.5 GB
Page File Space	17.6 GB
Hyper-V - VM Monitor Mode Extensions	Yes
Hyper-V - Data Execution Protection	Yes
Hyper-V - Virtualization Enabled in Firmware	Yes
Hyper-V - Second Level Address Translation Extensions	Yes

Table 1. Specification sheet for the PC used in simulation.

For enabling the operations to be fast in all phases of optimization, we make use of NVIDIA GT 710 dedicated GPU with integration with in-built intel graphics card for sharing the memory already occupied in the CPU to execute the designated task faster than usual. The RAM usage will not be free since the running of GPUs generally consumes a little bit of RAM usage in the CPU to enable the GPUs to withstand the subjected task load. The properties of the utilized GPU are indicated in the below figure 5.

In figure 5, the monitor comprises the name of BenQ G610HDAL on NVIDIA GeForce GT 710, 1366 x 768 pixels of current resolution and model of NVIDIA GeForce GT 710. It had the other components of work resolution, state, monitor width, monitor height, monitor BPP, and frequency. The common memory that is available globally for a GPU is a DRAM. The latency of accessing memory could become larger by the fewer sizes as cache. Likewise, loading the kernel relation/any task suddenly into GPU would be consuming a huge quantity of period. So, the efficiency of the GPUs should be used efficiently for the sake of reducing the period for any designated and operating tasks. So, GPU utilization is made productive by loading data in a discrete period of times, thereby making the ALU-Arithmetic Logic Units (integral parts/components of CPU which gives the GPU with the capability to perform faster along with within-built graphics in CPU) to improvise the total span time of a particular task.

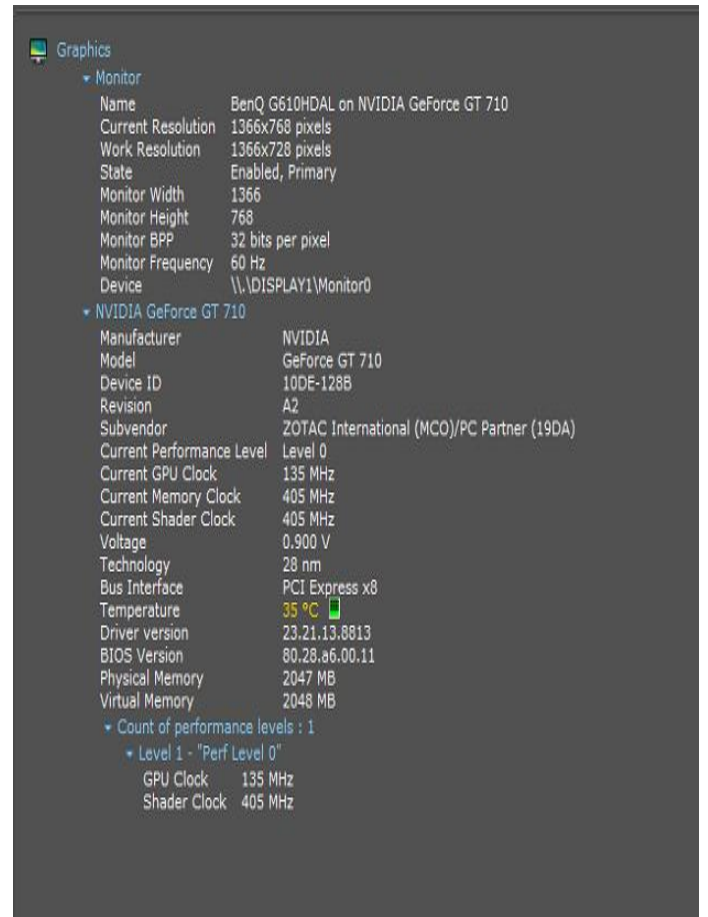


Figure 5. GPU Properties

#### IV Performance Analysis

The performance analysis part is proceeded with the comparative study, especially taking [22] and [23] datasets correspondingly as and when required. Temperature and execution/run time will be compared with the help of 2 sets of datasets mainly in the later section.

Table 2 shows the Hybrid multiverse optimizer is better in terms of execution time and other constraints in 3D IC floorplanning. The method for the benchmark circuit n300 has resulted in 970 TSVs within a time of 2.95 seconds.

Circuit	Lin et al[7]			Hybrid Multiverse optimizer		
	No. of TSV	WL ( $\mu\text{m}$ )	Time (s)	No. of TSV	WL ( $\mu\text{m}$ )	Time (s)
n 100	455	130145	4271	350	125504	2.90
n 200	969	241208	28.1	845	221809	3.20
n 300	1024	322329	49.76	970	290203	2.95

Table 2. Evaluation of our method with existing methods for soft module standards

First, we make a comparison with the temperature in ICs in table 3 concerning the executed optimization tasks. We make the correlation with the average temperature of CPU and GPU (supported with the operation of CPU in the background) to show the effectiveness of using the GPU to reduce the temperatures in ICs. The proposed method of comparison analysis made with [24] existing 3D global routing algorithm. This analysis proved that the proposed

method contains minimum temperature with CPU and GPU for IBM datasets compared to existing methods.

Circuit	Average Temperature (°C)	
	Proposed	Thermal Aware Method [24]
ibm01	33.98	61.7
ibm02	36.46	57.3
ibm03	71.49	68.1
ibm04	68.99	68.6
ibm06	43.76	63

Table 3. Average temperature evaluation with IBM sets

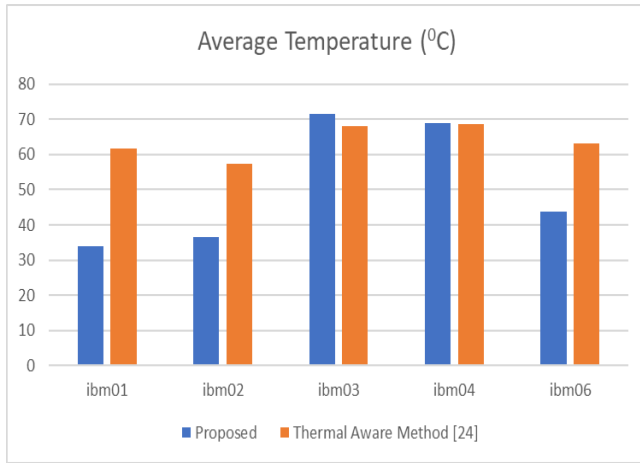


Figure 6. Temperature evaluation with IBM sets

From the above figure 6, we have compared both the CPU optimization and integrated optimization with GPU to realize the average temperature values during the task execution. The comparison was made with 18 IBM sets.

Then, we make the investigation with the task execution/run time for the designated task in Table 4 concerning the executed optimization tasks. We make the correlation with the CPU and GPU (supported with the operation of CPU in the background) to show the yielded less run time with GPU for the optimizations performed in ICs.

Dataset	Execution Time(sec)	
	CPU	GPU
ibm01	412.4968	321.3987
ibm02	686.2946	565.8517
ibm03	656.1899	576.3484
ibm04	762.2604	629.8746
ibm06	768.7134	661.0886
ibm07	743.743	652.7173
ibm08	900.392	780.0342
ibm09	571.655	491.4761
ibm10	711.171	570.7921
ibm11	958.706	857.1644
ibm13	980.277	886.1482
ibm14	828.046	716.1947

ibm15	517.097	419.7814
ibm16	925.823	800.644
ibm17	967.695	855.2158
ibm18	840.317	723.1714

Table 4. Run time evaluation with CPU and GPU using IBM sets.

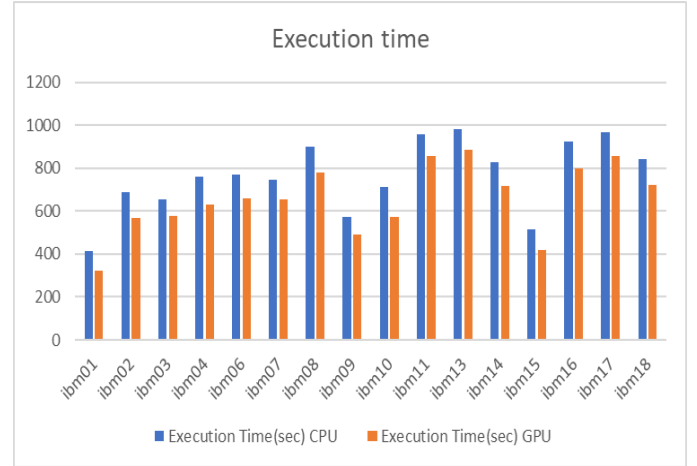


Figure 7. CPU / GPU Runtime evaluation with IBM sets  
The above figure 7 depicts the comparative study with the optimization by CPU and integrated optimization by GPU to indicate the optimized task run time. The correlation was made with 18 IBM sets and realized that the GPU contributed lesser running/operational run time outcomes when performing the optimization tasks towards the improvisations in the ICs.

Dataset	Average Temperature (°C)
cal_040	31.77
cal_098	38.51
cal_336	51.46
cal_353	66.42
cal_523	41.83
cal_542	79.49
cal_566	76.78
cal_583	54.12
cal_588	44.440
cal_643	69.83

Table 5. Temperature evaluation with Calypto sets.

Secondly, we make a comparison with the temperature in ICs in table 5 concerning the executed optimization tasks. We make a correlation with the CPU and GPU (supported with the operation of CPU in the background) for our investigation.

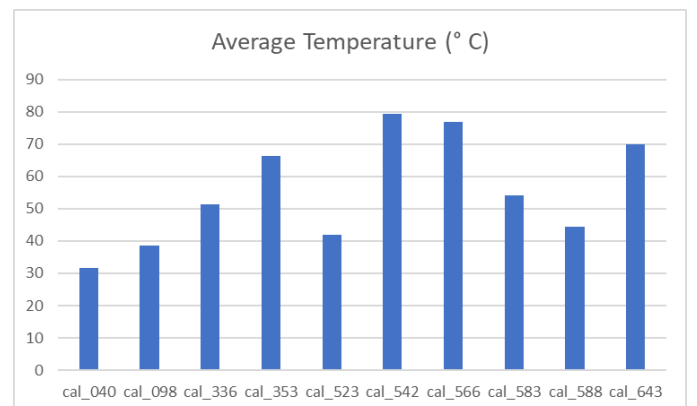
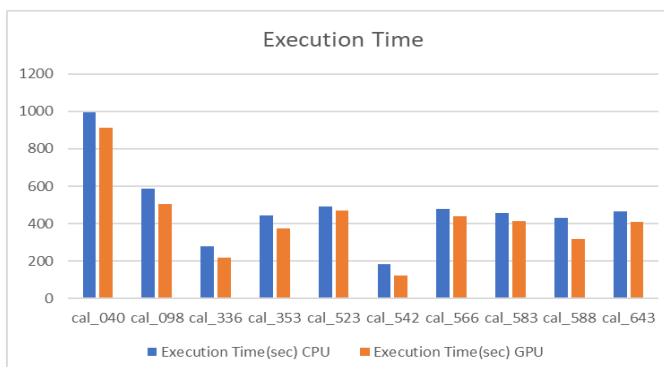


Figure 8. Temperature evaluation with Calypto sets.

From the above figure 8, we have compared both the CPU optimization and integrated optimization with GPU to indicate the average temperature of both CPU and GPU. The comparison was made with 10Cal sets and realized that the GPU gave rise to improvised run times as shown in the above figure 8. Finally, we make the performance evaluation with the task execution/run time for the designated task in Table 6 concerning the executed optimization tasks. We make the comparison with the CPU and GPU (supported with the operation of CPU in the background) to contribute lesser run time after GPU introduction for optimizing in ICs.

Dataset	Execution Time(sec)	
	CPU	GPU
cal_040	995.482	912.764
cal_098	588.7655	506.3816
cal_336	280.4898	218.4456
cal_353	444.5853	375.3404
cal_523	489.5472	470.9593
cal_542	182.9293	125.2435
cal_566	476.5678	438.0759
cal_583	457.0119	413.4694
cal_588	431.7482	316.9133
cal_643	465.9961	408.0782

Table 6. Run time evaluation with CPU and GPU with Calypto sets



**Figure 9.** CPU / GPU Runtime evaluation with Calypto sets. The above figure 9 depicts the comparative study with the CPU aided optimization and GPU aided optimization to indicate the improvised task run time. The correlation was made with 10 Calypto sets from which we can infer that the GPU resulted in economical operational/running run times when performing the optimization tasks. Thus, the performance evaluation of the intended parameters was made in this section successfully.

## V Conclusion

This paper proposed the TSV, Area, Wire length, and temperature floor planning optimization algorithms in 3-D ICs. First, we reduced the TSV counts based on the position designated by the Shuffled Frog Leaping Algorithm. Then, the space occupied by several interior/tiny components was improved by reducing the overall area occupied by them in every layer with the deployment of the Moth Flame Optimization Algorithm. Then, after optimizing the area in all possible ways, we tended to optimize the length of the wire intermediary to the components whose position was

altered to optimum feasible position/location and temperature of all the components/spares inside the ICs by using the methodology of Multi-Verse Optimizer. Also, our work tended to reduce the operating time of the optimization operation by introducing GPUs along with the primary CPU in the simulated system. When investigating the performance of our devised sequential optimizations in terms of TSV counts, cross-sectional area, wire length, and temperature, these methods outperformed all the state of the methodologies. The novelty of this work was to introduce GPU to reduce the simulation time taken for executing all kinds of optimizations FP problems that we tackled in this work.

## References

- [1] S. K. Samal, S. Panth, K. Samadi, M. Saeidi, Y. Du, and S. K. Lim, "Adaptive regression-based thermal modeling and optimization for monolithic 3-D ICs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, pp. 1707-1720, 2016.
- [2] Y. Zhao, S. Khursheed, B. M. Al-Hashimi, and Z. Zhao, "Co-optimization of fault tolerance, wire length and temperature mitigation in TSV-based 3D ICs," in *2016 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*, 2016, pp. 1-6.
- [3] A. Dutt, P. Roy, and H. Rahaman, "TSV-aware 3-D IC structural planning with irregular die-size," in *2016 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, 2016, pp. 713-716.
- [4] M. A. Ahmed, S. Mohapatra, and M. Chrzanowska-Jeske, "Dynamic nets-to-TSVs assignment in 3D floorplanning," in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2015, pp. 1870-1873.
- [5] J.-M. Lin and C.-Y. Huang, "General floorplanning methodology for 3D ICs with an arbitrary bonding style," in *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2018, pp. 1199-1202.
- [6] S. Mohapatra, S. K. Vendra, and M. Chrzanowska-Jeske, "Through Silicon Via-Aware Layout Design and Power Estimation in Sub-45 Nanometer 3D CMOS IC Technologies," in *2018 IEEE 13th Nanotechnology Materials and Devices Conference (NMDC)*, 2018, pp. 1-4.
- [7] J.-M. Lin and J.-A. Yang, "Routability-driven TSV-aware floorplanning methodology for fixed-outline 3-D ICs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, pp. 1856-1868, 2017.
- [8] S. Paramasivam, S. Athappan, E. D. Natrajan, and M. Shanmugam, "Optimization of Thermal Aware VLSI Non-Slicing Floorplanning Using Hybrid Particle Swarm Optimization Algorithm-Harmony Search Algorithm," *Circuits and Systems*, vol. 7, p. 562, 2016.
- [9] S. K. Reddy, "MINIMIZATION OF VLSI FLOORPLAN USING HYBRID CUCKOO SEARCH AND PSO," 2015.
- [10] P. Sivaranjani and A. Senthil Kumar, "Hybrid Particle Swarm Optimization-Firefly algorithm (HPSOFF) for combinatorial optimization of non-slicing VLSI floorplanning," *Journal of Intelligent & Fuzzy Systems*, vol. 32, pp. 661-669, 2017.

- [11] P. Sivaranjani and A. S. Kumar, "Thermal-aware non-slicing VLSI Floorplanning using a smart decision-making PSO-GA based hybrid algorithm," *Circuits, Systems, and Signal Processing*, vol. 34, pp. 3521-3542, 2015.
- [12] Q. Zou, E. Kursun, and Y. Xie, "Thermomechanical Stress-Aware Management for 3-D IC Designs," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, pp. 2678-2682, 2017.
- [13] P. Sivaranjani and A. Senthilkumar, "3D VLSI Non-Slicing Floorplanning using Modified Corner List Representation," *Indian Journal of Science and Technology*, vol. 8, p. 1, 2015.
- [14] M. A. Ahmed, S. Mohapatra, and M. Chrzanowska-Jeske, "TSV-and delay-aware 3D-IC floorplanning," *Analog Integrated Circuits and Signal Processing*, vol. 87, pp. 235-248, 2016.
- [15] C. Jang and J. w. Chong, "Thermal-Aware Floorplanning with Min-cut Die Partition for 3D ICs," *ETRI Journal*, vol. 36, pp. 635-642, 2014.
- [16] P. Subbaraj, P. Sivakumar, V. District, and V. District, "Parallel memetic algorithm for VLSI circuit partitioning problem using graphical processing units," *Journal of Computer Science*, vol. 8, p. 705, 2012.
- [17] M. A. Sabet, B. Ghavami, and M. Raji, "GPU-Accelerated Soft Error Rate Analysis of Large-Scale Integrated Circuits," *IEEE Design & Test*, vol. 35, pp. 78-85, 2018.
- [18] Z. Yunzhou, Z. Mo, L. Haoqi, and Z. Gang, "Innovative architecture of single chip edge device based on virtualization technology," *Pervasive and Mobile Computing*, vol. 52, pp. 100-112, 2019.
- [19] M. Osama, L. Gaber, A. I. Hussein, and H. Mahmoud, "An efficient SAT-based test generation algorithm with GPU accelerator," *Journal of Electronic Testing*, vol. 34, pp. 511-527, 2018.
- [20] D. Shu, Y. Wei, V. Dinavahi, K. Wang, Z. Yan, and X. Li, "Co-Simulation of Shifted-Frequency/Dynamic Phasor and Electromagnetic Transient Models of Hybrid LCC-MMC DC Grids on Integrated CPU-GPUs," *IEEE Transactions on Industrial Electronics*, 2019.
- [21] C. Cook, H. Zhao, T. Sato, M. Hiromoto, and S. X.-D. Tan, "GPU-based Ising computing for solving max-cut combinatorial optimization problems," *Integration*, vol. 69, pp. 335-344, 2019.
- [22] [http://vlsicad.eecs.umich.edu/BK/ISPD06bench/#IBM-HB+\\_Bench](http://vlsicad.eecs.umich.edu/BK/ISPD06bench/#IBM-HB+_Bench).
- [23] [http://vlsicad.eecs.umich.edu/BK/ISPD06bench/#Calypto\\_Bench](http://vlsicad.eecs.umich.edu/BK/ISPD06bench/#Calypto_Bench).
- [24] T. Zhang, Y. Zhan, and S. S. Sapatnekar, "Temperature-aware routing in 3D ICs," in *Asia and South Pacific Conference on Design Automation, 2006.*, 2006, p. 6 pp.
- [25] Pothiraj, S., Kadambarajan, J. & Kadarkarai, P. Floor Planning of 3D IC Design Using Hybrid Multi-verse Optimizer. *Wireless Pers Commun* (2021). <https://doi.org/10.1007/s11277-021-08166-z>

## Author Biography



**K. Jeya Prakash** is currently an Assistant Professor in the Department of Electronics and Communication Engineering at Kalasalingam Academy of Research and Education (Kalasalingam University), Krishnankoil, India since 2009. Prior to his appointment at Kalasalingam University, he worked as a lecturer in Electronics and Communication Engineering department at RMK Engineering College, Kavaraipettai from 2007 to 2009 and as a lecturer in Electronics and Communication Engineering Department at Sethu Institute of Technology, Puloor from 2003 to 2005. He completed his M.E. in VLSI Design from Anna University, Chennai and B.E. in Electronics and Communication Engineering from Bharathidasan University, Tiruchirappalli in the years 2007 and 2000, respectively. He is doing his Ph.D. in 3D IC Physical Design at Kalasalingam Academy of Research and Education (Deemed to be University). He is also interested in implementation of Problem based Learning and Outcome Based Education in Engineering. He has published 6 papers in Indexed Journals and nearly 24 papers in reputed national and international conferences.

**Dr. P. Sivakumar** is currently a Professor in the Department of Electronics and Communication Engineering at Kalasalingam Academy of Research and Education (Kalasalingam University), Krishnankoil, India since 2008. He graduated in Electronics and Communication Engineering from Anna University and post graduated in VLSI Design Sastra University, Thanjavur, India. He had 15 years' experience in Research and Development, Teaching. He has published 15 technical research papers in reputed National and International Journals. He had completed his Ph.D. in VLSI Physical Design at Anna University. He is also interested in Machine Intelligence research.



**K. Pandiaraj** is currently an Assistant Professor in the Department of Electronics and Communication Engineering at Kalasalingam Academy of Research and Education (Kalasalingam University), Krishnankoil, India since 2008. He graduated in Electronics and Communication Engineering from Anna University and post graduated in VLSI Design in 2006 from Anna University, Chennai, India. He has published 6 technical research papers in reputed International Journals and IEEE Conferences and published 23 papers in International and national Level Conferences. He is doing his Ph.D. in 3D IC Physical Design at Kalasalingam Academy of Research and Education.

