electronics

MDPI

Article

Multi-Tier 3D IC Physical Design with Analytical Quadratic Partitioning Algorithm Using 2D P&R Tool

Azwad Tamir [1], Milad Salem [1], Jie Lin [1], Qutaiba Alasad [2] and Jiann-shiun Yuan [1,*]

[1] Department of Electrical and Computer Engineering, University of Central Florida, Orlando, FL 32816, USA; a.tamir@knights.ucf.edu (A.T.); miladsalem@knights.ucf.edu (M.S.); ljie@knights.ucf.edu (J.L.)
[2] Department of Petroleum Systems and Control Engineering, Tikrit University, Al-Qadissiya, Tikrit P.O. Box 42, Iraq; quitaibaeng@knights.ucf.edu
* Correspondence: Jiann-Shiun.Yuan@ucf.edu

Abstract: In this study, we developed a complete flow for the design of monolithic 3D ICs. We have taken the register-transfer level netlist of a circuit as the input and synthesized it to construct the gate-level netlist. Next, we partitioned the circuit using custom-made partitioning algorithms and implemented the place and route flow of the entire 3D IC by repurposing 2D electronic design automation tools. We implemented two different partitioning algorithms, namely the min-cut and the analytical quadratic (AQ) algorithms, to assign the cells in different tiers. We applied our flow on three different benchmark circuits and compared the total power dissipation of the 3D designs with their 2D counterparts. We also compared our results with that of similar works and obtained significantly better performance. Our two-tier 3D flow with AQ partitioner obtained 37.69%, 35.06%, and 12.15% power reduction compared to its 2D counterparts on the advanced encryption standard, floating-point unit, and fast Fourier transform benchmark circuits, respectively. Finally, we analyzed the type of circuits that are more applicable for a 3D layout and the impact of increasing the number of tiers of the 3D design on total power dissipation.

Keywords: 3D integrated circuit; analytical quadratic; minimum cut; MIV; monolithic 3D IC; physical design; placement; tier partitioning

Citation: Tamir, A.; Salem, M.; Lin, J.; Alasad, Q.; Yuan, J.-s. Multi-Tier 3D IC Physical Design with Analytical Quadratic Partitioning Algorithm Using 2D P&R Tool. *Electronics* **2021**, *10*, 1930. https://doi.org/10.3390/electronics10161930

Academic Editor: Esteban Tlelo-Cuautle

Received: 31 March 2021
Accepted: 8 August 2021
Published: 11 August 2021

1. Introduction

The continuing trend of rapid reduction in the transistor's size is causing a large increase in the density of the digital IC, resulting in severe problems in placing low parasitic interconnects within the standard cells. This reduction of transistor size is essential as it would allow more transistors to be fit on a chip of the same dimensionality and help manufacture smaller electronics. Moreover, we are nearing the limit of transistor dimensionality reduction, making it increasingly difficult to keep up with the golden rule of the industry, i.e., Moore's law [1]. One potential solution to these problems is the three-dimensional IC layout structure, which can contribute to ultra-high-density ICs, leading to higher performance and lower overall power dissipation.

The evolution of the IC physical design first produced the 2.5D structure, which consists of an interposer layer, with individual die placed on top of it [2]. This served as a steppingstone for the design of the 3D IC, which consisted of silicon die placed vertically on top of each other. The individual dies are referred to as tiers and they are made up of a silicon substrate, followed by a device layer with an interconnect layer on the top [2–4]. The early 3D designs involved the tiers being fabricated separately and placed on top of each other and connected with the help of vertical interconnects called through-silicon vias (TSVs) [5–11]. However, there were several disadvantages to the usage of TSVs compared to the utilization of the more recent technology of monolithic inter-tier vias (MIVs). The large pitch and size of the TSV results in high parasitic capacitance and resistance, and limits the density with which they could be placed. Conversely, MIVs are

orders of magnitude smaller in comparison and comparable in dimensions to a regular inter-tier via, which is less than 100 nm in diameter [12]. The monolithic IC is made by fabricating each tier sequentially instead of putting them together after fabrication, as done with TSV-based designs.

Significant work has been done on the monolithic 3D IC design in recent times, most of which contributed to developing various design concepts of 3D ICs. The studies in this field include developing a face-to-face stacked heterogeneous 3D IC structure [13], exploring various microfluidic cooling mechanisms for 3D ICs [14], studying recent developments in monolithic 3D ICs and the high density and performance benefits [15], designing a logic-on-memory processor using monolithic 3D (M3D) IC techniques [16], developing a design and testing system for M3D ICs [17], comparing the TSV-based 3D structure with M3Ds [18,19], studying the effect of process variation on the performance of M3D ICs [20,21], and developing effective gate-sizing methods to boost circuit speed while considering intra-die process variation [22]. Other related studies include repurposing various components of commercial 2D P&R tools to implement 3D ICs [23–26]. However, there are not many works related to the development of a complete RTL-to-GDS physical design flow for the M3D structure due to the unavailability of commercial EDA tools for M3D P&R implementation. The limited study in this area includes the design of an overall flow developed by H. Park et al. and the development of M3D logic by Y. Lee et al. [27–29]. Our work involves an easy-to-use RTL-to-GDS design flow with custom-made tier partitioning algorithms. We analyzed the results obtained and compared them with that of related works and found a significant performance increase.

The major contributions of our work are provided below:

1. We developed two different tier partitioning systems named the minimum cut (min-cut) and analytical quadratic (AQ) algorithms and compared their results with related works. We also compared the results obtained on the same circuit with the AQ and min-cut partitioning algorithms.
2. We designed a complete RTL-to-GDS design flow that is easily implementable using existing EDA tools. We used the Design Compiler (DC) and the IC Compiler II (ICC2) tool from Synopsys® for performing the logic synthesis and P&R implementations, respectively.
3. We evaluated power dissipation results for two-to-five-tier designs for three benchmark circuits using two different tier partitioning algorithms and compared the results with a 2D layout. Finally, we examined the power dissipation reductions of increasing the number of tiers for different circuit types.

We presented the details of our overall design flow, our tier partitioning algorithms, and the P&R flow system in Section 2. We reported and analyzed our results in Section 3. Finally, we discussed our findings in Section 4, and followed up with concluding remarks in Section 5.

2. Methodology

2.1. Overall Design Flow

The pipeline for the overall design is shown in Figure 1. The input of the system is a register-transfer level (RTL) netlist of the circuit. The Design Compiler tool from Synopsys® was used to synthesize the circuit into a gate-level netlist, which was then passed into the tier-partitioning algorithm, which provides the individual tier netlists. The details of the tier-partitioning algorithm are discussed in Section 2.2. The individual tier netlists are then passed to the Synopsys® ICC2 tool for the placement and routing (P&R) flow. The power dissipation results of each tier are generated from the P&R design and added together to get the overall result of the 3D IC. Each tier in the design is of the same size and is reduced depending on the number of tiers in the 3D design. For example, if it is a two-tier design, then the total core area of each tier is half the size of the original 2D design.

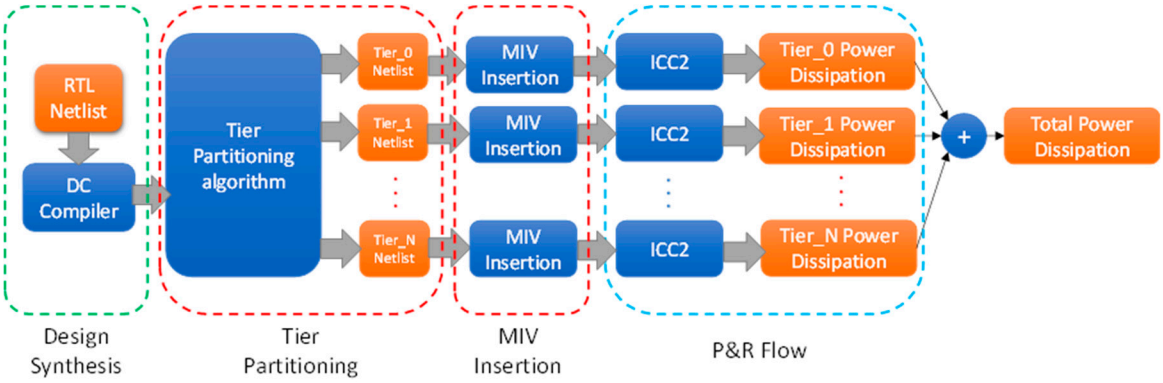


Figure 1. Overall design flow chart.

2.2. Tier Partitioning

One of the main challenges of 3D IC design is partitioning the cells between the tiers. Intuitively, this step has the most impact on the final performance of the physical design since it explicitly affects the contents of each tier. A subpar tier partitioning, such as a random tier partitioning, would spread cells between arbitrary tiers and would disregard four main considerations: (1) their connections to different nets, (2) their location after placement, (3) the number of MIVs, and (4) the total area of the tiers after partitioning. In order to add these considerations to the tier partitioning step, this work implemented two approaches for tier partitioning; first, a minimum-cut algorithm, which has traditionally been used for partitioning the graph of the netlist, and second, an analytical quadratic algorithm, a novel algorithm that uses placement results in order to decide which cell belongs to which tier.

Figure 2 compares the two methods for tier partitioning and depicts that the main differences between the two algorithms are two-fold and are the type of the algorithm and the optimization goal of the algorithm. The min-cut algorithm directly performs on the netlist graph and optimizes the number of connections that would be severed due to partitioning. On the other hand, the placement algorithm works with netlist placement and optimizes the wire length between the placed cells. The implications of these two optimization strategies are discussed in the following subsections.

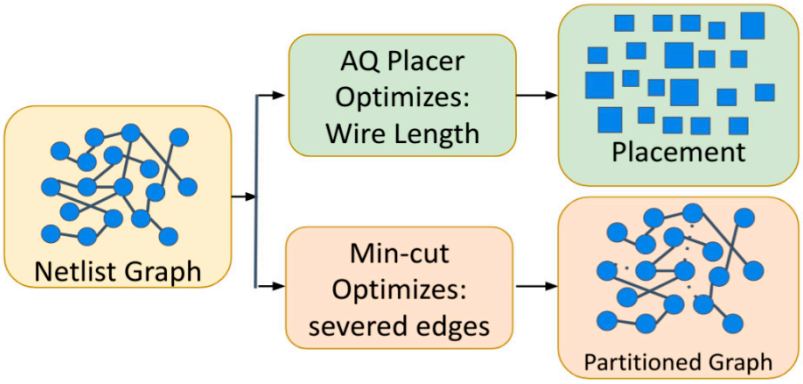


Figure 2. Comparison of the two implemented tier partitioning algorithms.

2.2.1. Partitioning Using the Minimum-Cut Algorithm

The min-cut algorithm was derived from the multilevel k-way partitioning graph (MKWPG) algorithm, which comprises three phases: coarsening, initial partitioning, and uncoarsening and refinement [30,31]. The process begins by coarsening the graph recursively until a graph with a small enough size is derived. More precisely, the input graph is replaced with a smaller graph version that is similar to the original graph but has fewer edges and nodes compared to the former. The process is repeated until a sufficiently small

version of the graph is obtained that could be partitioned easily and quickly. After the coarsening phase is achieved, the initial partitioning begins to partition this small version of the graph to k parts. As this coarsest partitioned graph is somehow like the parent graph, it could be partitioned in the same manner. Lastly, in the uncoarsening step the original graph is partitioned via reversely repeating this process. Figure 3 illustrates the three main steps of the MKWPG algorithm.

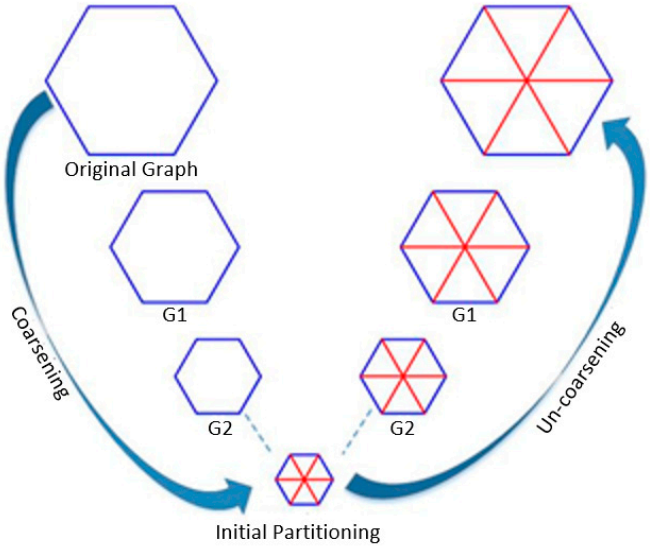


Figure 3. Multilevel k-way partitioning graph algorithm steps.

Inherently, a netlist is similar to a graph with nodes representing the gates, and edges representing the connections within the nets. Therefore, an algorithm such as min-cut can be leveraged to modify a netlist. Since the netlist is represented as a graph, the algorithm would directly work with the connections between the cells, which are represented as the edges of the graph. Therefore, using the min-cut algorithm would allow the tier partitioner to explicitly take into account the connections between the cells and the number of MIVs, which would replace the cut connections, heeding two of the main four considerations.

To implement the algorithm using a high-level language, the netlist graph needs to be represented by an adjacency matrix. The rows and columns represent vertices and the nonzero values in the adjacency matrix are edges where there are connections within the netlist. The breadth-first search (BFS) algorithm was used to compute the graph in the adjacency matrix. The matrix was then divided into k parts in a row-wise manner. These row blocks (k parts) were then processed where each row block representing a partition of the graph. The nodes were exchanged among the partitions repeatedly to balance the number of nodes between the parts and minimize the number of cut edges.

The min-cut partitioner starts by using Python to parse the gate-level Verilog circuit and convert it into a graph. The wires and the gates are represented as edges and nodes, respectively. Note that, since the sizes of the gate types are different, we weighed the converted gates based on the number of required transistors as described: inverter, buffer/NAND/NOR, AND/OR, XOR, and XNOR as 2, 4, 6, 12, and 14, respectively. Currently, we are focused on static CMOS logic, so the partitioner only works on static logic. This process is vital as the circuit size depends on the total number of transistors and not the total number of gates. We first convert/parse the Verilog circuit (including gates and wires) into a graph (including nodes and edges). During the conversion, we created a dictionary that has information about the Verilog circuit, e.g., gate type, gate name, gate location, wire direction, and number of transistors. We get the exact number of transistors for each gate type from the library since the library has all the gate information, including the number of transistors, which is very important to partition the circuit in an efficient way. Once the partitioning is done, we convert the partitioned parts back to partitioned circuits.

To clearly explain this, we use the small circuit shown in Figure 4, which has six gates. If we partition the circuit into two parts without weighing the gates based on the gate type, then the algorithm will partition the circuit, as shown with the red dotted line, in which each part will have three gates with two cut edges (wires) as a minimum. This makes the size of the right part (part 2) larger than the size of the left part (part 1) by about 2.7 times since the number of transistors in part 1 is 14, while it is 38 in part 2. However, if we weigh each gate type before converting to the graph, then part 1 will have 4 gates (26 transistors) and part 2 will have 2 gates (26 transistors), as shown by the dotted green line. Hence, the process allows us to partition the circuit into k parts with equal size and a minimum number of cut edges.

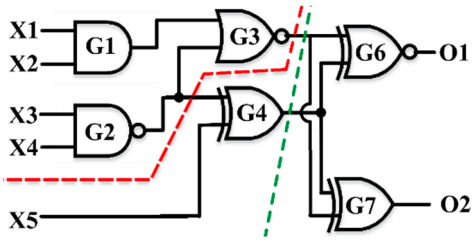


Figure 4. Demonstration of the min-cut partitioning algorithm using a simple example circuit.

2.2.2. Partitioning Using Initial Analytical Quadratic Placement

While the min-cut algorithm considers the connections between the cells, it fails to include any information regarding how the cells would be placed in the physical design. This information can be derived from the output of the placement algorithms that usually use wire length as a constraint to optimize the placement of the cells. To this end, this work proposes to use the output of an analytical quadratic placer as a discriminating feature for partitioning. Using the AQ algorithm would allow the tier partitioning to take into account the location of the cells after placement, the total area of the tiers after partitioning connections between the cells, and, implicitly, the connections between the cells, heeding three of the main four considerations.

This section describes the details of the AQ partitioner, which divides the netlist into N equal parts. The AQ algorithm works by assigning each standard cell in the gate-level netlist into one specific tier out of the overall N tiers. The principal steps of the AQ partitioning system are outlined in Figure 5. With the help of a placement algorithm, placed cells' physical locations could be used as a discriminative metric for tier assignment.

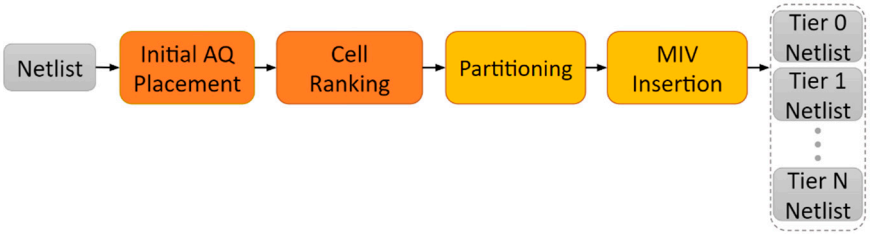


Figure 5. AQ partitioning pipeline.

As shown in Figure 5, the tier assignment starts with performing an initial AQ placement on the given netlist. This 2D placement finds the location (x, y) of each cell to minimize the total wire length of the placed netlist. In its standard implementation, the initial step of AQ is followed by iterative partitioning, placement, and legalization [32]. However, in this work, this initial step is taken as a representation of the placed netlist since it can be performed fast and is efficient in terms of spreading the cells. In order to implement the initial AQ placement, a new temporary net is inserted between each pair of connected cells, creating the fully connected clique model of the netlist. The connection matrix of all cells and pins are then extracted using the weights of each connection within the clique model.

The optimum location for each cell placement is then calculated by setting the derivative of the total quadratic wire length to zero.

As seen in Figure 6, having performed the initial AQ, the placed cells are ranked based on their location on the *x*-axis, with the cells on the left side of the placed netlist having higher ranks than the cells on the right side. This partitioning approach is inspired by the recursive partitioning step of the AQ placement algorithm. To partition the netlist, cells in the ranking queue are assigned a specific tier (starting at the lowest tier) until the tier's area limitation is reached. At this point, the tier is changed to the next higher one.

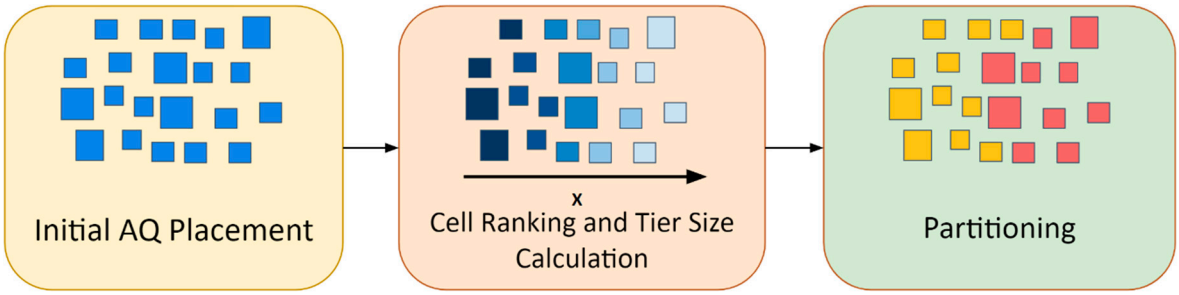


Figure 6. Example of tier partitioning. The red and yellow rectangles in the third window represent cells assigned to different tiers.

After the partitioning is performed and each cell is assigned a tier, there exist nets that have been spread across multiple tiers. In order to keep these connections intact, MIVs are inserted in the severed nets as an extra cell. The details of the MIV insertion process are described in detail in the next section.

2.3. MIV Insertion

Vertical interconnects such as MIVs are used to carry the signal from one tier to the next. The general representation of an MIV is shown in Figure 7, along with its equivalent circuit model. It consists of two contact resistances and a wire resistance in series with capacitor components with the substrate. Therefore, the MIV could be modelled as a resistor in series with a capacitance in parallel. The size of an MIV is negligible compared to the size of typical standard cells, so there is no density cap placed on the MIV placement. The overall resistance and capacitance of an MIV could be assumed to be 2 Ω and 0.1 fF, respectively, with an approximate diameter of 100 nm [11]. The approximate size and parasitic capacitance of the MIVs are comparable to the antennae diode standard cell, which is already present in the technology library. Therefore, the antennae diode standard cell is used to simulate MIV cells. These cells are placed at the end of each of the nets, which are cut off by the tier-partitioning algorithm, as shown in Figure 8. This insertion is also iteratively performed if the nets spread in more than two tiers, connecting the lowest tier to the highest tier that the net occupies.

Once the netlist is partitioned and MIVs are connected to the severed nets, all tiers are complete and can be saved as a netlist. Therefore, the outputs of the partitioning algorithms are N separate netlists, which are passed to physical design software for the P&R flow.

2.4. P&R Flow Using ICC2

The MIV insertion step is followed by the placement and routing implementation carried out with the ICC2 EDA tool. The detailed steps of this process are shown in Figure 9. The first step of the flow is the design initialization, which sets up the design environment and loads the technology library into the system. A 32 nm standard cell library was used to implement the flow, which was obtained from Synopsys®. The cell density of the different tiers of the 3D design are matched with that of the 2D design to aid in consistency and to achieve a fair result comparison. The design initialization is followed by the floor planning,

which builds the PG grid and the core area. Each tier of the design consists of a total of nine metal layers to accommodate the interconnects.

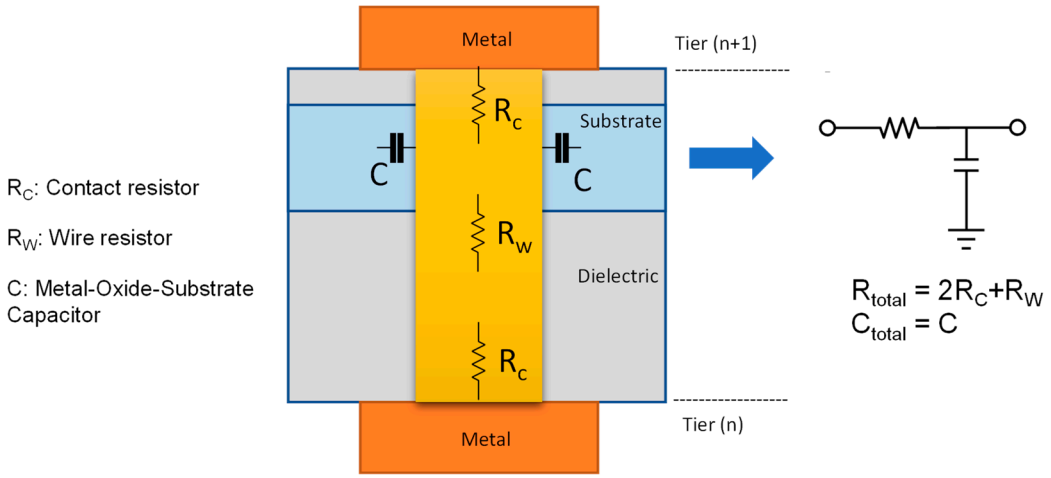


Figure 7. MIV model with equivalent resistance and capacitor.

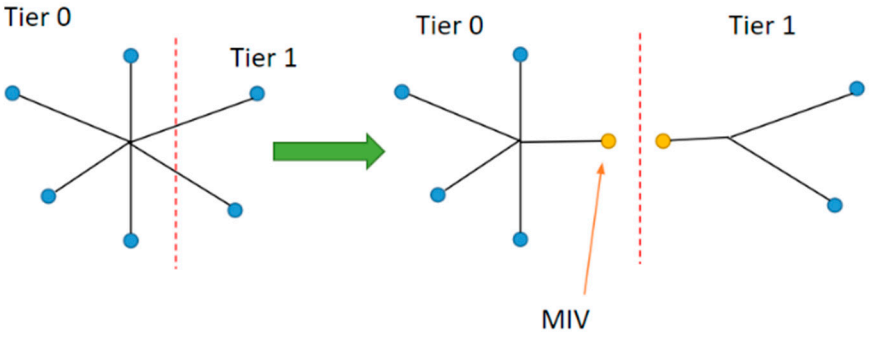


Figure 8. MIV insertion.

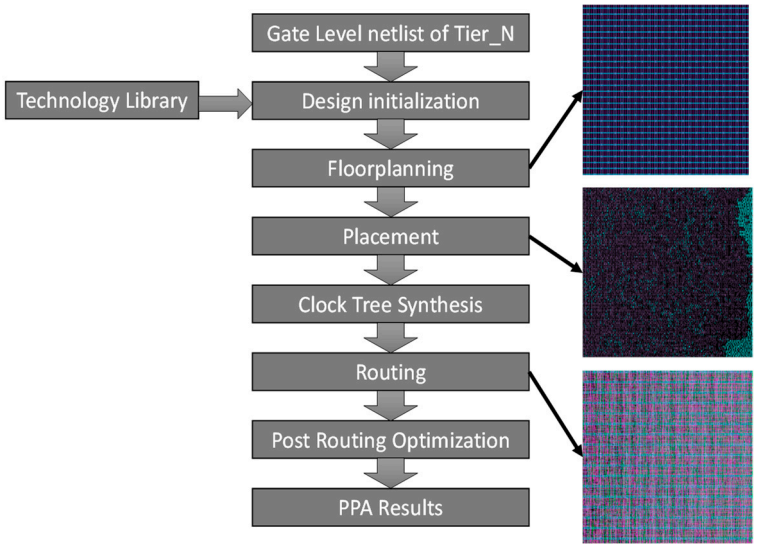


Figure 9. P&R design flow.

This is followed by the placement of the cells and the clock tree synthesis to remove timing violations. Finally, the cells are connected in the routing step, followed by some postrouting optimizations. These optimizations are handled automatically by the P&R tool and include such things as fixing timing violations within tiers, making small changes in routing to optimize wire lengths, power optimizations, postroute buffer removal, etc.

The power, performance, and area results are obtained for the individual tiers and added together to give the overall power dissipation of the entire 3D design.

3. Results

The entire design flow was executed on three different benchmark circuits: advanced encryption standard (AES), floating-point unit (FPU), and fast Fourier transform (FFT) circuits obtained from the opencores.org website (accessed on 5 June 2020). These circuits differ in size and density and are used to demonstrate the applicability of 3D designs on different circuit types. The properties of the benchmark circuits are given in Table 1.

Table 1. Benchmark circuits.

Benchmark Circuits	Total Number of Cells	Core Area at 70% Utilization
Advanced encryption standard (AES)	13,921	48,241.360
Floating-point unit (FPU)	37,242	112,944.389
Fast Fourier transform (FFT)	105,790	581,766.113

Power dissipation results have been studied for the 2D and 3D layouts for the benchmark circuits. Two-tier and three-tier models using the minimum-cut and analytical quadratic partitioning algorithms were developed and the reductions in power dissipation were compared with those from other similar studies [28,29]. Moreover, the four-tier and five-tier results were generated using the AQ partitioning for the benchmark circuits to show the trend in power dissipation as the total number of tiers in the 3D design were increased. The AQ algorithm was chosen for this analysis as it produces the best results.

Table 2 shows the power dissipation results and the percentage decrease in total power normalized to the 2D design for different 3D layouts and tier partitioning algorithms. The results are visualized in Figure 10 and show that the AQ algorithm's overall performance was better than that of the min-cut partitioning. The 3D results compared to the 2D counterparts were first analyzed. They show that the 3D implementation with the min-cut partitioning algorithm produced lower power dissipation for the AES and the FFT circuits while showing an increase in power dissipation for the FPU circuit. On the other hand, the results obtained with the AQ algorithm show a significant reduction in power for all the three benchmark circuits, with the AES circuit showing the most reduction in power dissipation for the two-tier models, while the FPU circuit showed the most power reduction for the higher-tier models. We also compared our results with those from other similar works [28,29], as shown in Figure 10. The FFT 3D comparison results were obtained from [28], where the 3TM metal layer structure was chosen as it produces the best results with no reduction in the metal dimensions, while the AES and the FPU comparison results were drawn from [29], using the T-MI 3D design with the 45 nm node. Only two-tier 3D models were available in these studies. Our two-tier 3D layout developed using the min-cut partitioner produced better results than the related works [28,29] for the AES and FFT circuits, while our model comprising the AQ partitioner did significantly better in all three benchmark circuits. The most likely cause for the better power results of our implementation compared to [28,29] was better partitioning algorithms. However, the exact technology libraries and layout configurations of [28,29] differ from our implementation. As a result, the comparison between the different methods applied in this work alone are more valid compared to those between our implementation and other related studies [28,29].

Table 2. Power dissipation results of benchmark circuits. The negative values in the "% Reduction in Power normalized to 2D" column represent an increase in the power dissipation.

Circuit	Structure	Partitioning Algorithm	Total Power Dissipation (mW)	% Reduction in Power Normalized to 2D
AES	2D	-	1.95	-
	3D (two tiers)	Min-cut	1.615	17.18%
		AQ	1.215	37.69%
	3D (three tiers)	Min-cut	1.552	20.41%
		AQ	1.162	40.41%
	3D (four tiers)	AQ	1.142	41.44%
	3D(five tiers)	AQ	1.073	44.9%
FPU	2D	-	5.79	-
	3D (two tiers)	Min-cut	7.11	−22.80%
		AQ	3.76	35.06%
	3D (three tiers)	Min-cut	7.31	−26.25%
		AQ	3.04	47.50%
	3D (four tiers)	AQ	2.76	52.33%
	3D(five tiers)	AQ	2.485	57.08%
FFT	2D	-	49.4	-
	3D (two tiers)	Min-cut	39.3	20.45%
		AQ	43.4	12.15%
	3D (three tiers)	Min-cut	39	21.05%
		AQ	42.7	13.56%
	3D (four tiers)	AQ	41.59	15.81%
	3D(five tiers)	AQ	43.6	11.71%

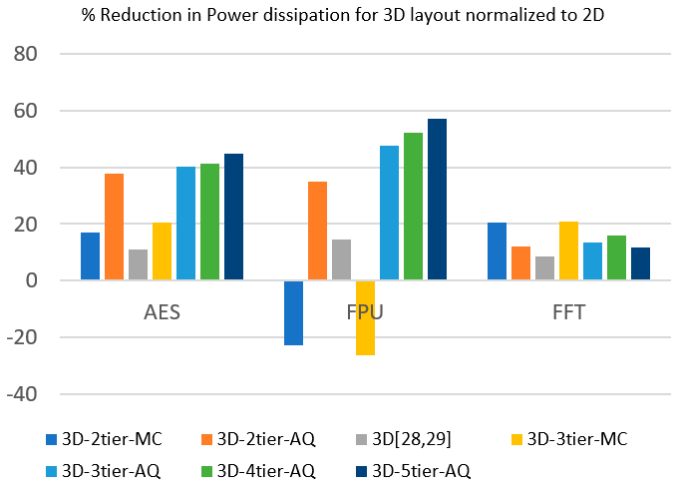


Figure 10. Reduction in power dissipation for different 3D layouts normalized to the 2D values.

We next analyzed the reduction in power dissipation obtained by increasing the number of tiers of the 3D design using the AQ partitioner, as shown in Figure 11. The results show a sharp decrease in power dissipation as we moved from the 2D layout to the two-tier 3D model, with smaller incremental decreases as we moved to higher tiers. The advantage of the 3D design in terms of reducing power dissipation is more prominent for circuits that are more wire dominant in nature. This makes the shift from 2D to 3D layouts

more applicable for wire-intensive circuits than those that are more cell dominant. To analyze this fact further, the cell density and wire utilization plots for the three benchmark circuits were generated and are shown in Figure 12. The cell density plots show the concentration of cells to be the highest in the FFT circuit, with those of AES and FPU being considerably lower. On the other hand, the wire utilization plots show the AES and the FPU circuits to have a more concentrated wire density than the FFT. This shows the FFT circuit to be much more cell dominant than the AES and the FPU, which are more wire dominant. Hence, the AES and FPU circuits are more suited for a 3D layout compared to the FFT circuit.

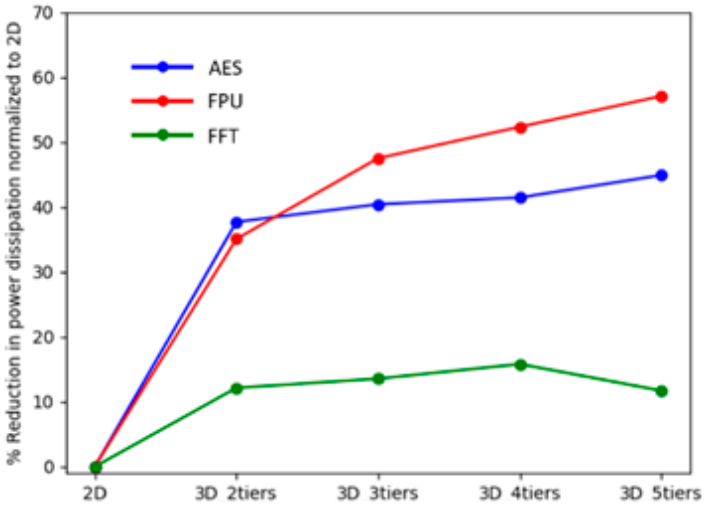


Figure 11. Power dissipation advantage for different tiers.

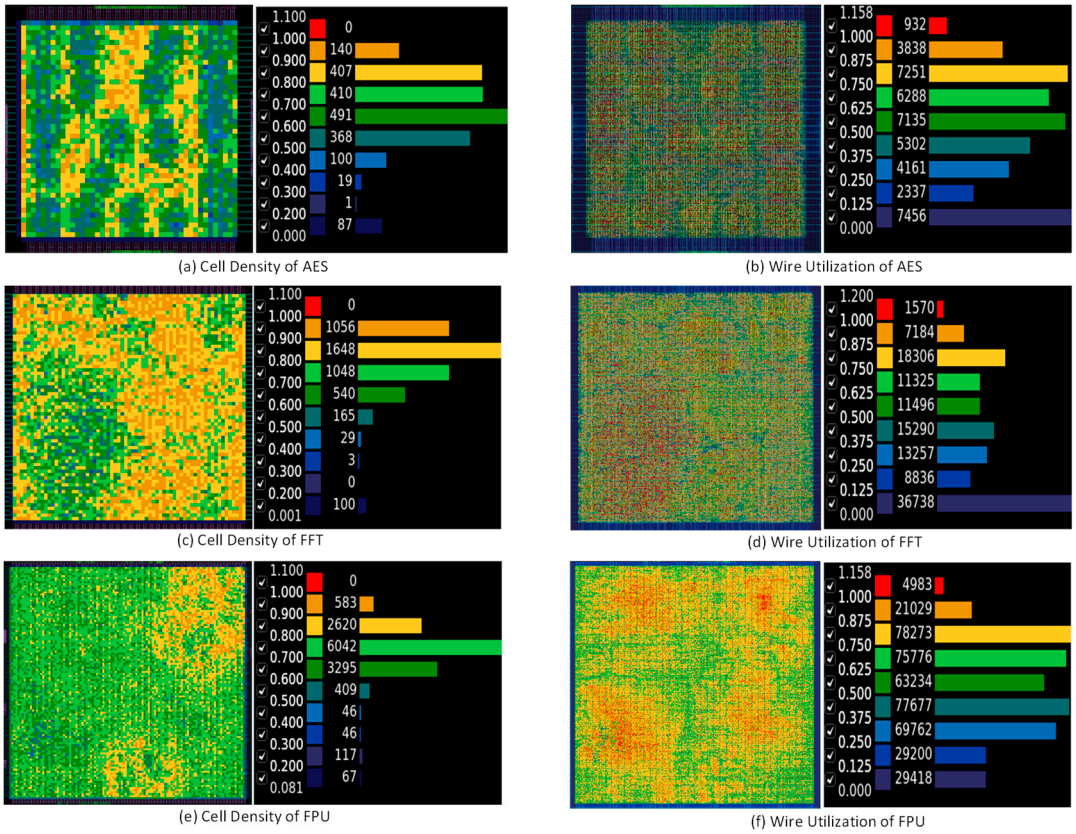
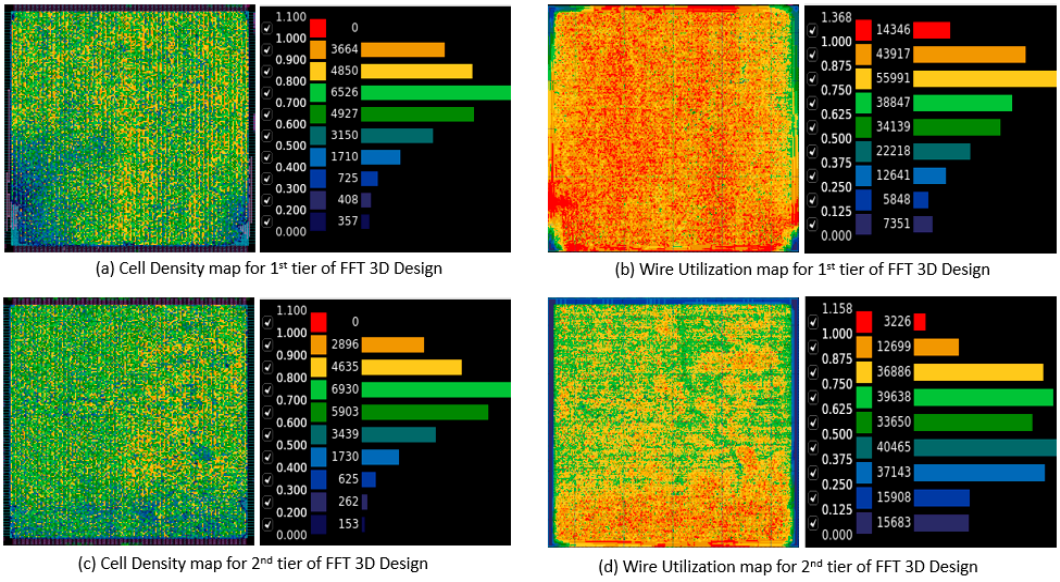


Figure 12. Cell density and wire utilization plots for the different benchmark circuits.

Moreover, the FFT circuit has a large clump of wires localized in the lower-left corner, making it more unsuitable for 3D implementation. These trends are confirmed by the 3D results that show the AES and the FPU's overall 3D performance to be much better than the FFT circuit. The FFT circuit also saturated faster and even experienced an increase in power dissipation as we expanded it to a five-tier design. In contrast, the other two circuits showed a steady increase up to this point.

The cell density and wire utilization plots for the two-tier 3D design of the FFT and the AES benchmark circuits using the AQ partitioner are given in Figures 13 and 14, respectively. The FPU benchmark circuit showed a similar trend, so the FFT and AES circuits are shown as examples. The plot shows even splitting of the cells among the two tiers. The plots show greater routing congestion for the FFT circuit compared to the AES circuit. Of note, the first tier of the FFT circuit showed very large wire utilization compared to the 2D design. This is confirmed by the power dissipation results, which show better 3D results for the AES circuit compared to the FFT circuit. The AES circuit, on the other hand, shows better wire utilization plots with very low wire utilization for the first tier and only two small hot spots in the second tier as well. This bolsters our earlier conclusion that more wire-dominant circuits like the AES have better 3D performance compared to more cell-dominant circuits like the FFT. However, in cell-dominant circuits like the FFT, there is significant difference between the wire utilization of the two tiers. This suggests that a partitioner that can better predict routing congestion would perform better for cell-dominant circuits compared to our proposed partitioning algorithms that are placement-based.



(a) Cell Density map for 1st tier of FFT 3D Design

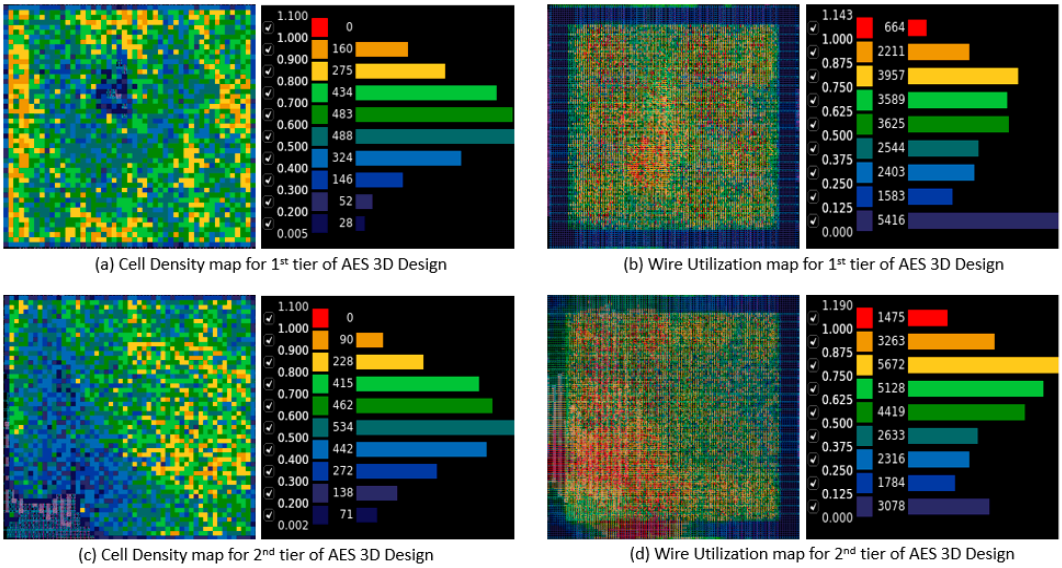
(b) Wire Utilization map for 1st tier of FFT 3D Design

(c) Cell Density map for 2nd tier of FFT 3D Design

(d) Wire Utilization map for 2nd tier of FFT 3D Design

Figure 13. Cell density and wire utilization plots for two-tier FFT 3D designs using the AQ partitioner.

The timing analysis data for the 2D and two-tier 3D designs of the AES benchmark circuit are shown in Table 3. Four different corner configurations were considered in the design. They are fast-fast at 125 °C, fast-fast at −40 °C, slow-slow at 125 °C, and slow-slow at −40 °C. The critical path length, slack, and clock period for the different corners are given in Table 3 along with the number of violating paths and the total negative slack for the 2D and two-tier 3D AES circuits. The timing violation statistics for the AES 2D and two-tier 3D designs are shown in Table 4. Both the 2D and 3D designs showed negligible violating nets compared to the total number of nets. The data shows similar timing performance for the 2D and the 3D circuits, so both the 2D and the two-tier 3D designs could be realized at a similar frequency.



(a) Cell Density map for 1st tier of AES 3D Design

(b) Wire Utilization map for 1st tier of AES 3D Design

(c) Cell Density map for 2nd tier of AES 3D Design

(d) Wire Utilization map for 2nd tier of AES 3D Design

Figure 14. Cell density and wire utilization plots for two-tier AES 3D designs using the AQ partitioner.

Table 3. Timing analysis results for 2D and two-tier 3D AES benchmark circuit.

Corners	Critical Path Length	Critical Path Slack	Critical Path Clock Period	Number of Violating Paths	Total Negative Slack
AES_2D					
ff_125c	0.74	3.19	4	0	0
ff_m40c	0.49	3.39	4	0	0
ss_125c	3.75	0	4	0	0
ss_m40c	3.77	−0.02	4	4	−0.03
AES_3D_Tier1					
ff_125c	0.66	3.31	4	0	0
ff_m40c	0.48	3.49	4	0	0
ss_125c	3.88	0.01	4	0	0
ss_m40c	3.68	0	4	1	0
AES_3D_Tier0					
ff_125c	0.71	3.26	4	0	0
ff_m40c	0.5	3.48	4	0	0
ss_125c	3.75	0	4	0	0
ss_m40c	3.81	−0.01	4	1	−0.01

Table 4. Timing violation information for the 2D and two-tier 3D benchmark circuits.

Circuit	Total Nets	Nets with Violations	Max Trans Violations	Max Cap Violations
AES_2D	14,299	33	13	32
AES_Tier1	7790	7	1	7
AES_Tier0	8194	10	2	10
FPU_2D	39,256	20	8	20
FPU_Tier1	20,944	62	27	57
FPU_TIer0	20,422	68	33	65
FFT_2D	114,197	242	121	240
FFT_Tier1	65,568	202	134	202
FFT_Tier0	52,658	61	9	61

However, our flow could only perform timing analysis on the nets that were confined to a single tier in the case of the 3D design and did not consider nets that travel between tiers. But a small portion of the overall nets travel in between tiers (less than 20% for the benchmark circuits), so the timing analysis of the 3D design could be considered as an approximation to the real results.

4. Discussion

Several further modifications could be made to increase the accuracy of the 3D model. These include implementing P&R of all the tiers together in the EDA tool instead of doing them separately. This would result in more accurate timing analysis and clock tree synthesis of the circuits. Another improvement to our current method could be achieved by implementing more detailed MIV modeling. In addition, different MIV models could be applied, and the results compared to investigate the effect of different types of MIVs on the performance of 3D ICs. Furthermore, the P&R flow could be modified to make sure that the connecting MIV cells in different tiers line up in the exact same physical locations in each tier to achieve more accurate results. The alignment of MIVs would ensure that cells on the partitioning border that were previously adjacent remain in close vicinity after tier partitioning, alleviating the problem of long routes between such cells. Moreover, the relationship between the MIV parasitic and the fan-in and fan-out of the logic gates should be analyzed in detail. These improvements and modifications are left as further work.

The time period of the 2D design and the individual tiers of the 3D design are set to be equal and only a small fraction of the nets go through the MIVs (about 18% for the two-tier AES design), so the 2D and the 3D designs are approximately iso performance. The implemented flow ensures that there are no timing violations among the nets that travel within tiers, but the flow does not have any mechanism to ensure that there are no timing violations in the fraction of the nets that travel in between tiers, and this accounts for a limitation in the study. However, the partitioner puts nets that are closely connected together in the same tier, so there are limited chances of timing violations being present in the inter-tier nets, and we leave the modification of the flow to include timing analysis for the inter-tier nets as future work.

The AQ partitioning runtime on the three benchmark circuits took less than 5 min each on a computer running on a standard Intel core i7 processor with 128GB of RAM. This shows that the AQ partitioning algorithm scales well and could be implemented easily on large industrial circuits as well.

5. Conclusions

The continuing trend in the reduction of the size of ICs is causing significant problems in the interconnect parasitic capacitance and resistance, which could be alleviated by moving to the 3D layout architecture. A complete design flow for the physical design of 3D ICs has been constructed in this study, which involves synthesizing the gate-level netlist from the RTL netlist, followed by tier partitioning and finally implementing the P&R flow.

The complete flow has been implemented on three benchmark circuits, which show significant reduction in the power dissipation of the 3D design compared to the 2D counterparts. Two different partitioning algorithms were implemented to assign the standard cells into different tiers. The AQ partitioning algorithm performed better on average compared to the min-cut algorithm on the three benchmark circuits that were used in the study.

The two-tier 3D flow power reduction compared to the 2D designs for the AES, FPU, and FFT benchmark circuits were 37.69%, 35.06%, and 12.15%, respectively, using the AQ partitioning algorithm. The power reduction steadily increased as the number of tiers for the 3D design was incremented. The steady decrease in the power dissipation was the most prominent for the FPU circuit and the least pronounced in the FFT circuit. Detailed analysis about the power reduction for the 3D architecture revealed that wire-dominant circuits like the AES are more applicable for 3D implementation than cell-dominant circuits like the FFT. The cell density and wire utilization heatmaps for the two-tier 3D designs for the FFT

and the AES circuits using the AQ partitioner show that the cells are evenly distributed among the tiers for both the AES and the FFT circuits, whereas the wire utilization plot for the FFT circuit shows more hot spots compared to the AES circuit. This observation is supported by the power dissipation results, which show better 3D performance for the AES circuit compared to the FFT circuit.

Finally, the timing analysis and violation data for the 2D and two-tier 3D designs show comparable timing performance between the 2D and 3D designs, which ensures the designs are iso performance. Of note, the 3D flow implemented in this work could only perform timing analysis for the nets that were restricted within a single tier of the 3D design, but only a small fraction of the nets travel in between the tiers so the reported timing data are very close to the real values.

Author Contributions: A.T. and J.L. designed the overall P&R flow and wrote the manuscript; M.S. implemented the AQ partitioning algorithm; Q.A. implemented the min-cut algorithm; J.-s.Y. edited and reviewed the manuscript and supervised the work. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by Tokyo Electron.

Acknowledgments: Special thanks to Tokyo Electron for providing constructive insight and discussions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sun, Y.; Agostini, N.B.; Dong, S.; Kaeli, D. Summarizing cpu and gpu design trends with product data. *arXiv* **2019**, arXiv:1911.11313.
2. Knickerbocker, J.U.; Andry, P.S.; Colgan, E.; Dang, B.; Dickson, T.; Gu, X.; Haymes, C.; Jahnes, C.; Liu, Y.; Maria, J.; et al. 2.5D and 3D technology challenges and test vehicle demonstrations. In Proceedings of the 2012 IEEE 62nd Electronic Components and Technology Conference, San Diego, CA, USA, 29 May–1 June 2012; pp. 1068–1076. [CrossRef]
3. Yoon, S.W.; Hoon, K.J.; Carson, F. Challenges and opportunity in 3D integration packaging. In Proceedings of the SEMICON Korea 2010 SEMI Technology Symposium (STS), Seoul, Korea, 3–5 February 2010.
4. Lancaster, A.; Keswani, M. Integrated circuit packaging review with an emphasis on 3D packaging. *Integration* **2018**, *60*, 204–212. [CrossRef]
5. Weerasekera, R.; Grange, M.; Pamunuwa, D.; Tenhunen, H.; Zheng, L.-R. Compact modelling of Through-Silicon Vias (TSVs) in three-dimensional (3-D) integrated circuits. In Proceedings of the 2009 IEEE International Conference on 3D System Integration, San Francisco, CA, USA, 28–30 September 2009; pp. 1–8.
6. Kim, D.H.; Topaloglu, R.O.; Lim, S.K. TSV density-driven global placement for 3D stacked ICs. In Proceedings of the 2011 International SoC Design Conference, Jeju, Korea, 17–18 November 2011; pp. 135–138. [CrossRef]
7. Ye, T.; Hou, L.; Zhang, S.; Wang, J.; Peng, X. TSV modelling in 3D IC thermoelectric simulation. In Proceedings of the 2017 IEEE 12th International Conference on ASIC (ASICON), Guiyang, China, 25–28 October 2017; pp. 678–681. [CrossRef]
8. Athikulwongse, K.; Kim, D.H.; Jung, M.; Lim, S.K. Block-level designs of die-to-wafer bonded 3D ICs and their design quality tradeoffs. In Proceedings of the 2013 18th Asia and South Pacific Design Automation Conference (ASP-DAC), Yokohama, Japan, 22–25 January 2013; pp. 687–692. [CrossRef]
9. Kim, D.H.; Athikulwongse, K.; Healy, M.B.; Hossain, M.M.; Jung, M.; Khorosh, I.; Kumar, G.; Lee, Y.-J.; Lewis, D.L.; Lin, T.-W.; et al. Design and Analysis of 3D-MAPS (3D Massively Parallel Processor with Stacked Memory). *IEEE Trans. Comput.* **2015**, *64*, 112–125. [CrossRef]
10. Jung, M.; Song, T.; Peng, Y.; Lim, S.K. Design Methodologies for Low-Power 3-D ICs With Advanced Tier Partitioning. *IEEE Trans. Very Large Scale Integr. Syst.* **2017**, *25*, 2109–2117. [CrossRef]
11. Song, T.; Panth, S.; Chae, Y.; Lim, S.K. More Power Reduction With 3-Tier Logic-on-Logic 3-D ICs. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **2016**, *35*, 2056–2067. [CrossRef]
12. Panth, S.; Samadi, K.; Du, Y.; Lim, S.K. Placement-Driven Partitioning for Congestion Mitigation in Monolithic 3D IC Designs. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **2015**, *34*, 540–553. [CrossRef]
13. Bamberg, L.; García-Ortiz, A.; Zhu, L.; Pentapati, S.; Shim, D.E.; Lim, S.K. Macro-3D: A Physical Design Methodology for Face-to-Face-Stacked Heterogeneous 3D ICs. In Proceedings of the 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, France, 9–13 March 2020; pp. 37–42. [CrossRef]
14. Wang, S.; Yin, Y.; Hu, C.; Rezai, P. 3D Integrated Circuit Cooling with Microfluidics. *Micromachines* **2018**, *9*, 287. [CrossRef] [PubMed]
15. Yang, C.-C.; Hsieh, T.Y.; Huang, W.H.; Shen, C.H.; Shieh, J.M.; Yeh, W.K.; Wu, M.C. Recent progress in low-temperature-process monolithic three dimension technology. *Jpn. J. Appl. Phys.* **2018**, *57*, 04FA06. [CrossRef]

16. Pentapati, S.; Zhu, L.; Bamberg, L.; Shim, D.E.; García-Ortiz, A.; Lim, S.K. A Logic-on-Memory Processor-System Design With Monolithic 3-D Technology. *IEEE Micro* **2019**, *39*, 38–45. [CrossRef]
17. Chaudhuri, A.; Banerjee, S.; Park, H.; Kim, J.; Murali, G.; Lee, E.; Kim, D.; Lim, S.K.; Mukhopadhyay, S.; Chakrabarty, K. Advances in Design and Test of Monolithic 3-D ICs. *IEEE Des. Test* **2020**, *37*, 92–100. [CrossRef]
18. Samal, S.K.; Nayak, D.; Ichihashi, M.; Banna, S.; Lim, S.K. Monolithic 3D IC vs. TSV-based 3D IC in 14nm FinFET technology. In Proceedings of the 2016 IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S), Burlingame, CA, USA, 10–13 October 2016; pp. 1–2. [CrossRef]
19. Nayak, D.K.; Banna, S.; Samal, S.K.; Lim, S.K. Power, performance, and cost comparisons of monolithic 3D ICs and TSV-based 3D ICs. In Proceedings of the 2015 IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S), Rohnert Park, CA, USA, 5–8 October 2015; pp. 1–2. [CrossRef]
20. Panth, S.; Samadi, K.; Du, Y.; Lim, S.K. Shrunk-2-D: A Physical Design Methodology to Build Commercial-Quality Monolithic 3-D ICs. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **2017**, *36*, 1716–1724. [CrossRef]
21. Panth, S.; Samadi, K.; Du, Y.; Lim, S.K. Power-performance study of block-level monolithic 3D-ICs considering inter-tier performance variations. In Proceedings of the 2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 1–5 June 2014; pp. 1–6. [CrossRef]
22. Perez-Rivera, Z.; Tlelo-Cuautle, E.; Champac, V. Gate Sizing Methodology with a Novel Accurate Metric to Improve Circuit Timing Performance under Process Variations. Available online: https://www.mdpi.com/2227-7080/8/2/25 (accessed on 8 April 2021).
23. Koneru, A.; Chakrabarty, K. Test and Design-for-Testability Solutions for Monolithic 3D Integrated Circuits. In Proceedings of the 2019 on Great Lakes Symposium on VLSI, Tysons Corner, VA, USA, 9–11 May 2019. [CrossRef]
24. Zanelli, J.C.; Metzler, C.; Reis, R. Gate Sizing for Power-Delay Optimization at Transistor-level Monolithic 3D-Integrated Circuits. In Proceedings of the 2020 IEEE 11th Latin American Symposium on Circuits & Systems (LASCAS), San Jose, Costa Rica, 25–28 February 2020; pp. 1–4. [CrossRef]
25. Chang, K.; Sinha, S.; Cline, B.; Southerland, R.; Doherty, M.; Yeric, G.; Lim, S.K. Cascade2D: A design-aware partitioning approach to monolithic 3D IC with 2D commercial tools. In Proceedings of the 2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), Austin, TX, USA, 7–10 November 2016; pp. 1–8. [CrossRef]
26. Jiang, J.; Parto, K.; Cao, W.; Banerjee, K. Ultimate Monolithic-3D Integration With 2D Materials: Rationale, Prospects, and Challenges. *IEEE J. Electron Devices Soc.* **2019**, *7*, 878–887. [CrossRef]
27. Park, H.; Chang, K.; Ku, B.W.; Kim, J.; Lee, E.; Kim, D.; Chaudhuri, A.; Banerjee, S.; Mukhopadhyay, S.; Chakrabarty, K.; et al. INVITED: RTL-to-GDS Tool Flow and Design-for-Test Solutions for Monolithic 3D ICs. In Proceedings of the 2019 56th ACM/IEEE Design Automation Conference (DAC), Las Vegas, NV, USA, 2–6 June 2019; pp. 1–4.
28. Lee, Y.; Lim, S.K. Ultrahigh Density Logic Designs Using Monolithic 3-D Integration. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **2013**, *32*, 1892–1905. [CrossRef]
29. Lee, Y.; Limbrick, D.; Lim, S.K. Power benefit study for ultra-high density transistor-level monolithic 3D ICs. In Proceedings of the 2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC), Austin, TX, USA, 29 May–7 June 2013; pp. 1–10.
30. Gilbert, J.R.; Zmijewski, E. A Parallel Graph Partitioning Algorithm for a Message-Passing Multiprocessor. *Int. J. Parallel Program.* **1987**, *16*, 427–449. [CrossRef]
31. Karypis, G.; Kumar, V. Parallel multilevel k-way partitioning scheme for irregular graphs. *Siam Rev.* **1999**, *41*, 278–300. [CrossRef]
32. Sigl, G.; Doll, K.; Johannes, F.M. Analytical placement: A linear or a quadratic objective function? In Proceedings of the 28th ACM/IEEE Design Automation Conference, San Francisco, CA, USA, 17–22 June 1991. [CrossRef]