Placement and Routing in 3D Integrated Circuits

Cristinel Ababei, Yan Feng, Brent Goplen, Hushrav Mogal,
Tianpei Zhang, Kia Bazargan, and Sachin S. Sapatnekar
ECE Department, Univ. of Minnesota
200 Union St. SE, Minneapolis, MN 55455
(kia,sachin)@umn.edu

***Abstract*— Advances in the chip fabrication technology have begun to make manufacturing 3D chips a reality. For 3D designs to achieve their full potential, it is imperative to develop effective physical design strategies that handle the complexities and new objectives specific to 3D designs. We present two frameworks of placement and routing techniques, for 3D FPGA and for 3D standard cell based designs, respectively. Our method addresses wire length, delay and area minimization, as well as thermal optimization during placement and routing phases. These two flows have been used to obtain optimized layouts for benchmarks with upto 8000 FPGA blocks and tens of thousands of standard cells, respectively.**

I. Introduction

In the natural world, high-rise buildings are the solution to the problem of accommodating large populations in areas that are considered prime real estate. In addition to permitting large population densities, such an arrangement also reduces the "interconnect bottleneck" that would come with the road network associated with an equivalent set of low-rises, which would have to be distributed over a larger area. The silicon world is not much different, and the need to densely pack circuits, and locate critical blocks as close as possible to each other, has led to the advent of three-dimensional (3D) technologies [1], with multiple tiers of devices stacked atop each other. The increased packing density improves the computation per unit volume, and results in diminished on-chip interconnect problems due to reduced parasitics [2]. This curtailment in the parasitics is achieved by reductions in the average interconnect lengths (in comparison with 2D implementations, for the same circuit size), as well as by denser integration, which results in the replacement of chip-to-chip interconnections by intra-chip connections. Consequently, 3D integration can be an enabler for enhancements in system performance, power, reliability, and portability. Advances in industrial [3], government [4] and academic [5] research laboratories have demonstrated 3D designs with inter-tier separations of the order of a few microns. Recently, MIT Lincoln Laboratories has offered a MOSIS-like 3D integration program under the auspices of DARPA.

Fundamentally, the problem of 3D design is related to topological arrangements of blocks, and therefore, physical design plays a natural role in determining the success of 3D design strategies. Physical design in the 3D realm requires a fresh approach, as new cost functions become important, and new design structures must be devised, and ordinary extensions of 2D approaches are unequal to the task of solving these problems.

This paper describes computer-aided design techniques for placement and routing in 3D integrated circuits, developed under our **3D-ADOpt** (**3D-A**nalysis and **D**esign **Opt**imization) framework. The approaches herein address a dichotomy of design styles: FPGA-style designs and ASIC-style designs. The factors that are important in each style are different, so that a "one-size-fits-all" approach is impractical, and therefore, we present separate approaches for 3D physical design for each of these technologies. For example, thermal issues are much more important in ASIC designs than in FPGA architectures since the power densities in the former are higher. This is because both the operating clock frequencies, and the density of utilized logic, are much higher in ASICs than in FPGAs. Therefore, our ASIC tool tightly integrates thermal issues in the placement and routing algorithms. Another example that highlights the differences between ASICs and FPGA fabrics is that the cost of higher connectivities in FPGAs is greater. This can be attributed to the fact that a larger number of possible connections must be facilitated (in the x, y and z dimensions) in FPGAs, and this entails an overhead of silicon real-estate that must be used to implement pass transistor switches, buffers and SRAMs that implement this capability; in ASICs, on the other hand, all we need to add is an inter-tier via that connects one active device tier (layer) to another one[1]. Hence, our FPGA placement method uses a two-step optimization process in which inter-tier vias are minimized first, followed by further optimization within and across tiers, while the ASIC flow uses cost function weighting to discourage, but not minimize, inter-tier crossings.

II. FPGA-style Designs

While there has been some previous work proposing 3D FPGA architectures, most of it falls short of proposing a complete 3D-specific system. Alexander *et al.* borrowed ideas from multi-chip module (MCM) techniques, and proposed to build a 3D FPGA by stacking together a number of 2D FPGA bare dies [6], with electrical contacts between different dies being made using solder bumps or vias passing through the die. The number of solder bumps that can fit on a die determines

This research was supported in part by DARPA under grant N66001-04-1-8909.

[1]It should be emphasized inter-tier vias are valuable resources in the 3D ASIC context too, but to a lesser degree than in 3D FPGAs

the width and separation of vertical channels between FPGA tiers (layers). Chiricescu *et al.* advocated placing the routing in one tier and the logic on another for more efficient tier utilization [7]. Universal switch boxes for 3D FPGA design were analyzed in [8]. It is important to point out that all previous FPGA works assume that the inter-tier connectivity is provided by vertical wire segments that connect each tier to its adjacent tiers only. With respect to developing CAD tools for 3D FPGA integration, Alexander *et al.* proposed 3D placement and routing algorithms for their architecture in [6].

A. FPGA Architecture Exploration

There are several considerations that must be taken into account while developing the architecture of a 3D FPGA. Designers must strike a balance between fabrication cost, area overhead, routability and speed. Architectural evaluation should be performed in the context of the circuits that will run on the FPGA chip, and the CAD tools that map such circuits to the FPGA device. An important factor affecting the performance and area efficiency of the 3D FPGA is the routing architecture. Switchboxes with too much connectivity will excessively waste area, and meager inter-tier via counts will hurt the performance of the design.

Figure 1(a) shows an example of a 3D FPGA where multiple 2D FPGAs are stacked, and a subset of the switches in the switchbox provide connections between tiers. Figure 1(b) shows such a switchbox. As can be seen from the figure, a switch that connects wire segments in all three x, y and z dimensions will have a connectivity $F_s = 5$[2], which translates to 15 pass transistors (and buffers) as opposed to a 2D connectivity of $F_s = 3$ which requires 6 pass transistors. As a result, the number of high connectivity switches must be minimized, without sacrificing routability. The routing architecture used in this work utilizes multi-segment routing with inter-tier wire segments of lengths 1, 2, and 6.

To fully evaluate the effect of architectural choices, designers need flexible physical design tools that can take architectural parameters as input, and report wire length, channel width, area and delay of benchmark circuits. We have developed a placement and routing tool called TPR (Three-dimensional Place and Route) for this purpose. Sections II-B and II-C describe the algorithms used in the placement and routing steps of TPR.

B. Placement Algorithms

The philosophy of our tool follows that of its 2D counterpart, VPR [9]. The flow of the TPR placement and routing CAD tool is shown in Figure II-B. The placement algorithm first employs a *partitioning* step using the hMetis algorithm [10] to divide the circuit into a number of balanced partitions, equal to the number of tiers for 3D integration. The goal of this first min-cut partitioning is to minimize the connections between tiers, which translates into reducing the number of vertical (i.e., inter-tier) wires and decreasing the area overhead associated with 3D switches as discussed in the previous section. After dividing the netlist into tiers, TPR continues with the *placement* of each tier using a hybrid approach that combines top-down partitioning and simulated annealing. The annealing step moves cells mostly within tiers. Finally, the cells are routed to obtain a placed and routed solution. The following sections describe these steps in more detail.

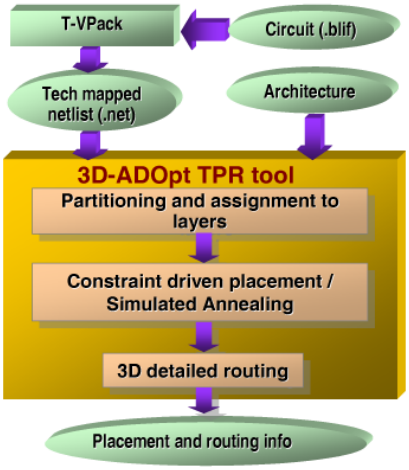


Fig. 2. Flow of our 3D-ADOpt TPR tool.

1) Partitioning the Circuit Between Tiers: The TPR step that performs partitioning and tier assignment of the circuit is shown conceptually in Figure 3. After the netlist is partitioned using hMetis, a novel linear placement approach is used to arrange the tiers such that wire length and the maximum cutsize between adjacent tiers is minimized. This is achieved by mapping this problem to that of minimizing the bandwidth of a matrix[3], using an efficient matrix bandwidth minimization heuristic.

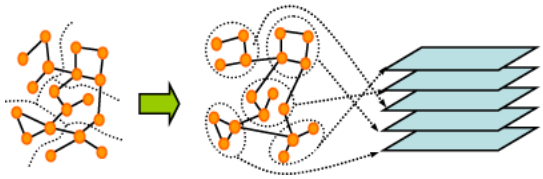


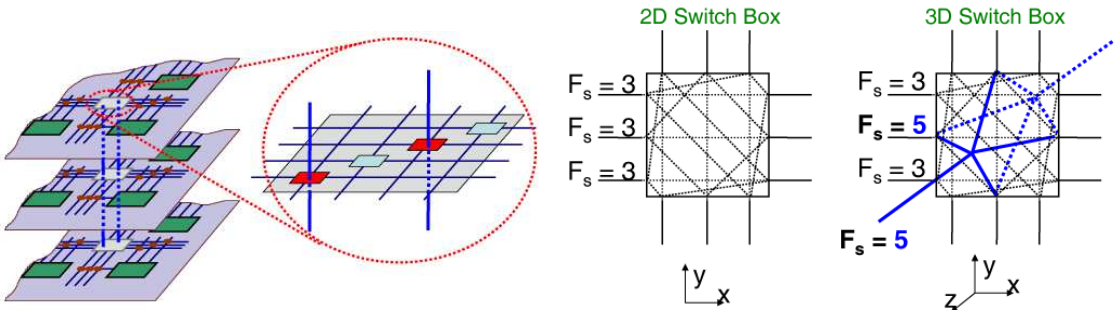
Fig. 3. Partitioning of the netlist into tiers

Figure 4 shows a graph in which each node corresponds to a cluster from the graph in Figure 3. An E-V matrix is formed in which each row corresponds to an edge, and the columns correspond to vertices. An entry a_{ij} in the matrix is non-zero if vertex j is incident to edge i, and zero otherwise, and the bandwidth of this matrix is sought to be minimized by choosing an optimal ordering of the vertices.

Intuitively, we would like to minimize the bandwidth of every row, because the bandwidth of a row represents how many tiers the net corresponding to that row spans. Furthermore, it

[2] F_s of a switchbox is defined as the number of outgoing tracks an incoming track is connected to.

[3] The bandwidth of a matrix is defined as the maximum bandwidth of all its rows. The bandwidth of a row is defined as the distance between the first and last non-zero entries.



(a) 3D FPGA

(b) Example 3D FPGA switch and its connectivity

Fig. 1. 3D FPGA and swtich example.

is desirable to distribute the bands of different rows among all columns, because the number of bands enclosing a particular column translates into the number of vertical vias that have to pass through the tier corresponding to that column. Minimizing the matrix band-width achieves both goals: it minimizes the span of every row, and distributes the bands across columns.

The bandwidth minimization problem is known to be NP-complete [11] and a solution for the tier assignment problem may not be optimal in terms of both objectives of wire-length and maximum cut between adjacent tiers. Therefore, for this step, we use an efficient heuristic [12] that is able to find solutions with very good trade-off between wire-length and maximum cut. This technique is briefly described in what follows using the graph example of Figure 4.

The procedure to solve the bandwidth minimization problem uses row (column) swaps in order to sort rows (columns) such that non-zero elements are moved towards the main diagonal. For example, for the matrix of Figure 4, in order to drift non-zero elements from the upper half towards the main diagonal (i.e., from right to left) column swaps are performed between columns 2 and 3 and then column 6 is moved between columns 2 and 4. This technique is repeated on rows and columns to move non-zero elements closer to the diagonal. When the above procedure is run on the example on the left of Figure 4, the linear arrangement on the right is created. The goal of getting the matrix to a band-form (which translates into a best linear ordering) serves two objectives:

> *Cutsize minimization:* by having all "1"s in the matrix clustered along the main diagonal, the cutsize (the number of nets cut by a dummy plane parallel to the tiers) is minimized everywhere in the linear arrangement.
>
> *Wirelength minimization:* by minimizing the band-width (maximum distance spanned by any of the nets) of the EV-matrix, the total wire-length of all nets is minimized.

The pseudo-code of the procedure used for EV-matrix bandwidth minimization is shown below. The "Left" array of the leftmost matrix in Figure 4 would be 3,6,4,6,6, since the rightmost "1" elements in the rows are located in columns 3, 6, 4, 6 and 6. Sorting this array requires swapping the second and third elements, which translates into swapping the second and third rows of the EV-matrix.

```
Algorithm Band-width minimization:
1. Build EV-matrix
2. Array Left = indices of right-most
     non-zero elements in rows.
   Sort Left swapping rows.
3. Array Top = indices of bottom-most
     non-zero elements in columns.
   Sort Top, swapping columns.
4. Array Right = indices of left-most
     non-zero elements in rows.
   Sort Right, swapping rows.
5. Array Bottom = indices of top-most
     non-zero elements in columns.
   Sort Bottom, swapping columns.
```

2) Partitioning-based Placement Within Tiers: After the initial tier assignment, placement is performed on each tier starting with the top tier, proceeding tier after tier. The placement of every tier is based on edge-weighted quad-partitioning using the hMetis partitioning algorithm, and is similar to the approach in [13], which has the same quality as VPR but at 3-4 times shorter run times. Edge weights are usually computed inversely proportional to the timing slack of the corresponding nets. However, we also selectively bias weights of the most critical nets. The set of critical nets is comprised of edges on the current k-most critical paths. In order to improve timing, the bounding box of the terminals of a critical net placed on a tier is projected to the lower tiers and used as a placement constraint for other terminals. More details of the partitioning-based placement phase can be found in [14].

3) Simulated Annealing Placement Phase: Following the partitioning-based placement step, a 3D-adapted version of VPR [9] is used in the low-temperature annealing phase to further improve wire length and routability. We use the following cost function for each net.

$$Cost_{3D}(e) = q.Cost_{2D}(e) + \alpha.Span_z(e) + \beta.numTiers(e) \quad (1)$$

where $Cost_{2D}$ is the half-perimeter size of the 2D projec-

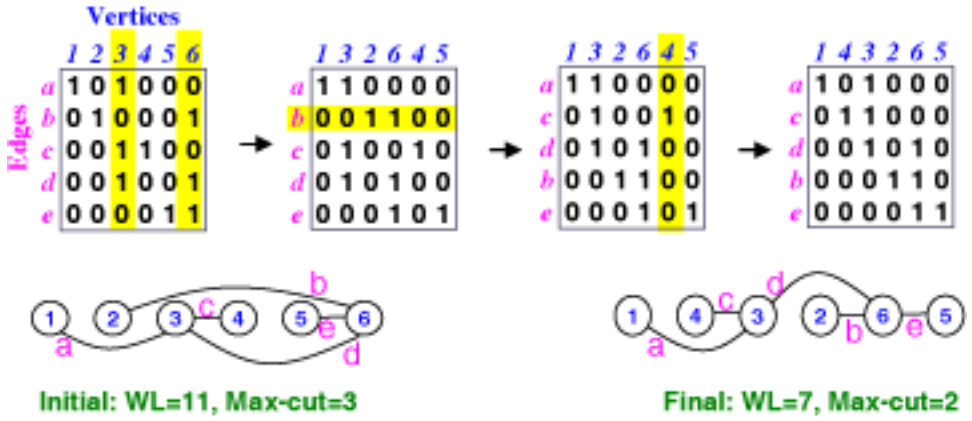


Fig. 4. The E-V matrix and steps to minimize both wire length and cutsize

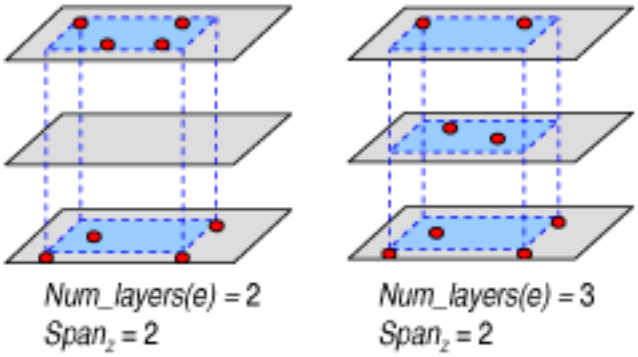


Fig. 5. An example showing the difference between a net's span and number of tiers.

tion of the bounding box of net e, $Span_z(e)$ is the total span of the net between tiers, and $numTiers(e)$ is the number of tiers on which the terminals of the net are distributed. Parameters q, α and β are tuning parameters (q has the same role as in VPR). Figure 5 shows an example to illustrate why we use the two components $Span_z$ and $numTiers$. In a 3D routing structure that employs multi-segment inter-tier connections, the left figure is more likely to use fewer vertical connections (of length 2) to connect the terminals on the first and the third tiers.

C. Routing Algorithms

Our routing algorithm is an extension of VPR's routing engine. The 3D FPGA architecture (see Figure 1(a)) described in the architecture file is represented as a routing resource graph. Each node of the routing resource graph represents a wire (horizontal tracks in the x and y channels of all tiers and vertical vias in the z channels) or a logic block input or output pin. A directed edge represents a unidirectional switch (such as a tri-state buffer). A pair of directed edges represents a bi-directional switch (such as a pass transistor). We add extra penalties to bends of a route created by a horizontal track and a vertical via as well as to vertical vias themselves in order to discourage the routing engine to prefer vertical vias and therefore to avoid a net placed totally in one tier to be routed using tracks in different tiers.

D. FPGA Results

In our experiments we used the 3D FPGA architecture of Figure 1(a) where segment lengths of 1, 2, and "long"[4] form the inter-tier routing structure, and segment lengths of 1, 2, 6, and "long" are used within tiers. The delay of an inter-tier segment is assumed to be equal to that of an intra-tier segment of the same length. This is justified by the relatively short length of inter-tier vias in the emerging 3D technologies, and the fact that the dominating factor in the delay of an FPGA routing segment is the pass transistor and buffer delays. Our architecture definition file can be modified to reflect any parasitics on the vertical connections, though.

Figure 6 shows the results of our algorithm on the MCNC benchmarks. The right graph compares the quality of 5-tier 3D circuits placed and routed using TPR to 2D circuits placed and routed by VPR. The bars show the ratio of the averages (over all the MCNC benchmark circuits) of metrics such as area, and delay to those of the 2D couterparts. Total area is the footprint area multiplied by number of tiers. It can be seen that delay and wire length can be improved by about 20%, whereas total area increases due to the extra area used by 3D switches and whitespace created on some tiers.

The right graph of Figure 6 shows the improvement in delay of all MCNC benchmarks as we increase the number of tiers. It can be observed that for this size of circuits, going up to 5-6 tiers has great benefits but beyond that, there are diminishing returns.

III. ASIC-style Designs

For standard cell based 3D designs, we describe a flow, illustrated in Figure 7, for performing placement and routing with built-in techniques for thermal mitigation. The input to the system is a technology-mapped netlist and a description of the library (these could be, for example, LEF/DEF and .lib descriptions), and the physical design process consists of several steps. Temperature is treated as a first-class citizen during this optimization, in addition to other conventional metrics, and intertier via reduction is also considered to be a desirable goal. In the *placement* step, the standard cells

[4] A "long" segment spans all tiers

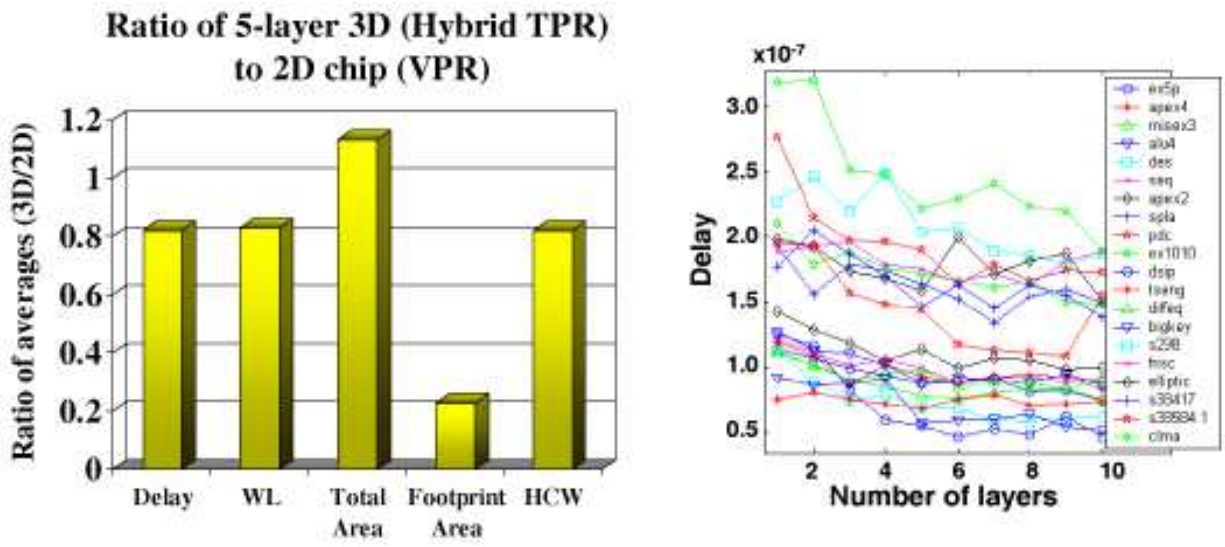


Fig. 6. Delay, wire length and area results of TPR compared to 2D annealing (i.e., VPR)

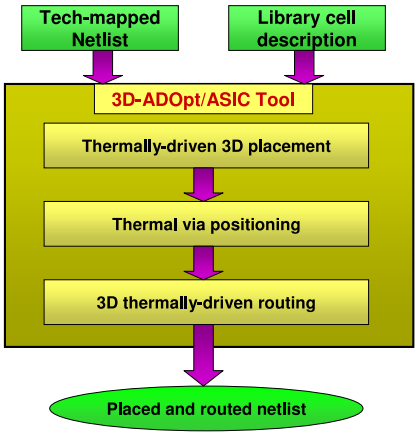


Fig. 7. Physical design flow for 3D ASIC-style implementations.

A. 3D thermally-driven placement

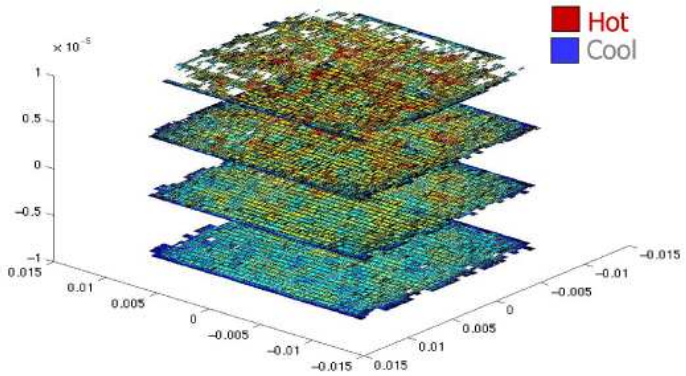


Fig. 8. A placement for the benchmark ibm01 in a four-tier 3D technology.

are arranged in rows within the tiers of the three-dimensional circuit. Since thermal considerations are particularly important in 3D ASIC-like circuits, this procedure must spread the cells to achieve a reasonable temperature distribution, while also capturing traditional placement requirements. In the second step, the temperature distribution is made more uniform by the judicious positioning of *thermal vias* within the placement, which achieves improved heat removal. These vias correspond to inter-tier metal connections that have no electrical function, but instead, constitute a passive cooling technology that draws heat from the problem areas to the heat sink. Finally, the placement goes through a *routing* step to obtain a completed layout. During routing, several objectives and constraints must be taken into consideration, including avoiding blockages due to areas occupied by thermal vias, incorporating the effect of temperature on the delays of the routed wires, and of course, traditional objectives such as wire length, timing, congestion and routing completion. We will now describe each of these steps in further detail.

Before describing the innards of the placer, it is illustrative to view the result of a typical 3D placement obtained using the 3D-ADOpt placer: a layout for the benchmark circuit, ibm01, in a four-tier 3D process, is displayed in Figure 8. The cells are positioned in ordered rows on each tier, and the layout in each individual tier looks similar to a 2D standard cell layout. The heat sink is placed at the bottom of the 3D chip, and the red regions are hotter than the blue regions. It is clear that the coolest cells are those in the bottom tier, next to the heat sink, and the temperature increases as we move to higher tiers. The thermal placement method consciously mitigates the temperature by making the upper tiers sparser, in terms of the percentage of area populated by the cells, than the lower tiers. In the subsequent description, we will provide an overview of the algorithms that are used within the placement engine, showing how it directly incorporates thermal objectives into placement.

1) Fast Thermal Analysis of 3D Integrated Circuits: An essential ingredient of a thermally-driven placement engine is a fast temperature analyzer. At the placement stage, it is

adequate to consider the steady-state case, where heat conduction within the chip substrate is described by the following differential equation:

$$K_x \frac{\partial^2 T}{\partial x}^2 + K_y \frac{\partial^2 T}{\partial y}^2 + K_z \frac{\partial^2 T}{\partial z}^2 + Q(x, y, z) = 0, \quad (2)$$

where T is the temperature, K_x, K_y, and K_z are, respectively, the thermal conductivities along the three coordinate directions, and Q is the heat generated per unit volume. A unique solution exists when convective, isothermal, and/or insulated boundary conditions are appropriately applied, and these are determined by the nature of the packaging and the heat sink.

The above partial differential equation can be solved numerically using finite element analysis (FEA) [15], which discretizes the design space into regions known as elements. For rectangular structures of the type encountered in integrated circuits, a rectangular cuboidal element can simulate heat conduction in the lateral directions without aberrations in the prime directions.

In FEA, the temperatures are calculated at discrete points (in this case, the nodes of the rectangular cuboid), and the temperatures within the elements are interpolated using a weighted average of the temperatures at the nodes. In deriving the finite element equations, the differential equation (2) is approximated within the elements using this interpolation. For a specific element type, such as a rectangular prism, one may derive "element stamps" that are similar in character to the element stamps for electrical elements in modified nodal analysis [16]. The heat conduction stamp for the eight-vertex rectangular prism can be derived as an 8×8 matrix.

These stamps are added to a global matrix to set up the global system of linear equations,

$$K\mathbf{T} = \mathbf{P}, \quad (3)$$

where $\mathbf{T}$ is the vector of nodal temperatures and $\mathbf{P}$ the vector of node powers. In the FEA parlance, the left hand side matrix, K, is referred to as the global stiffness matrix. Stamps for boundary conditions can similarly be derived. Conductive boundary conditions simply correspond to fixed temperatures; since these parameters are no longer variables, they can be eliminated and the quantities moved to the right hand side so that K is nonsingular. The FEA equations may be solved rapidly using an iterative linear solver, with clever adjustments of the convergence criteria to achieve greater or lesser accuracy, as required at different stages of the iterative placement process.

2) The force-directed paradigm: For 3D designs, placement must be carried out in not just the xy-plane, but the entire xyz-space in three dimensions. In current technologies, in the z dimension, the number of tiers is restricted to a small number.

The placement engine is based on a force-directed approach, where an analogy to Hooke's law is used by representing nets as springs and finding the cell positions that correspond to the minimum energy state of the system. Attractive forces, illustrated in Figure 9(a) and (b), are created between interconnected cells, and these are proportional to the quadratic function of the cell coordinates that represents the Euclidean distance between the blocks. The constants of proportionality are chosen to be higher in the z direction to discourage inter-tier vias. Fixed locations such as input/output pads, or fixed blocks, are easily incorporated into this formulation

Other design criteria such as cell overlap, timing, and congestion are used to derived the repulsive forces. In the 3D context, thermal criteria are used to generate repulsive forces, in order to prevent hot spots. The temperature gradient (which itself can be related to the stiffness matrix and its derivative) is used to determine the magnitudes and directions of these forces, as illustrated in Figure 9(c).

Once the entire system of attractive and repulsive forces is generated, repulsive forces are added, the system is solved for the minimum energy state, i.e., the equilibrium location. Ideally, this minimizes the wire lengths while at the same time satisfying the other design criteria such as the temperature distribution. The iterative force-directed approach follows the following steps in the main loop. Initially, forces are updated based on the previous placement. Using these new forces, the cell positions are then calculated. These two steps of calculating forces and finding cell positions are repeated until the exit criteria are satisfied. The specifics of the force-directed approach to thermal placement, including the mathematical details, are presented in [17].

Once the iterations converge, a final postprocessing step is used to legalize the placement. Even though forces have been added to discourage overlaps, the force-directed engine solves the problem in the continuous domain, and the task of legalization is to align cells to tiers, and to rows within each tier. The technique has been demonstrated on benchmark circuits with over 50,000 cells, and shows an approximately linear run-time trends as the circuit size increases. Placement results show average improvements of 17% in the average thermal gradient, as shown in Figure III-A.2, and 17% in the maximum temperature, for a nominal increase in wirelength as compared to non-thermal placement.

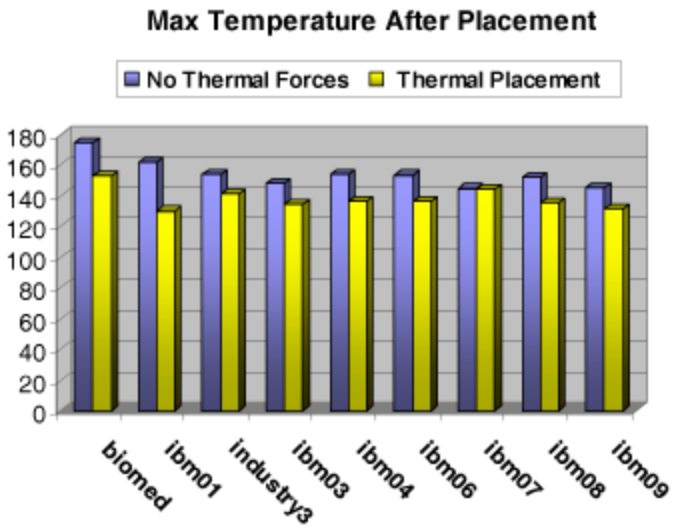


Fig. 10. Temperature gradient improvements with thermally-driven placement.

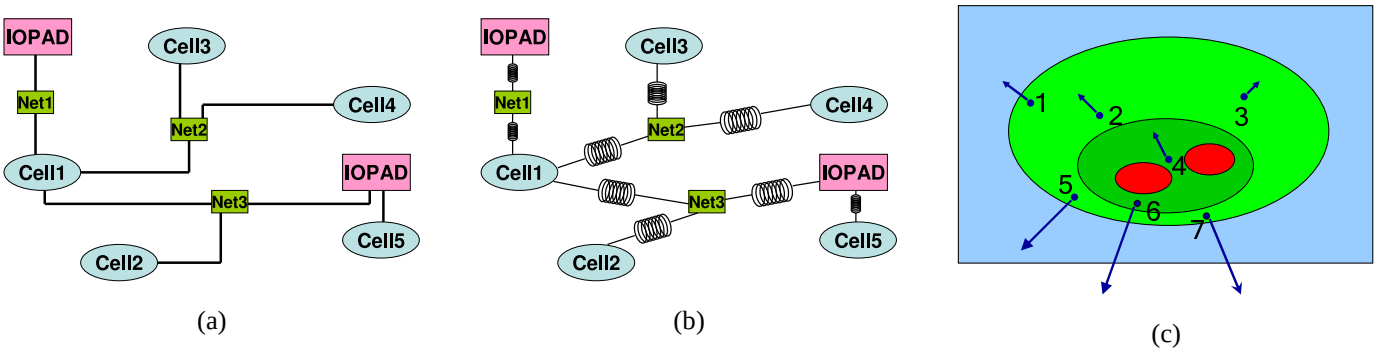


Fig. 9. (a) A sample netlist (b) Attractive forces corresponding to wire connections (c) Repulsive thermal force vectors

B. *Thermal via positioning*

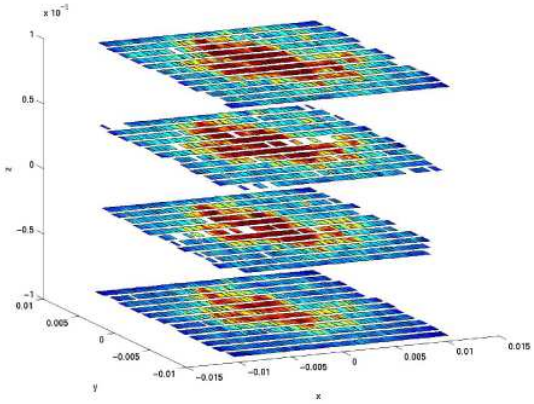


Fig. 11. Thermal profile of `struct` without thermal vias.

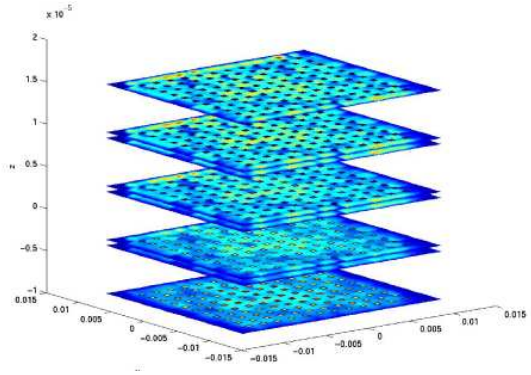


Fig. 12. Thermal profile of `struct` after thermal via insertion.

While silicon is a good thermal conductor, with half or more of the conductivity of typical metals, many of the materials used in 3D technologies are strong insulators that place severe restrictions on the amount of heat that can be removed, even under the best placement solution. The materials include epoxy bonding materials used to attach 3D tiers, or field oxide, or the insulator in an SOI technology. Therefore, the use of deliberate metal lines that serve as heat removing channels, called "thermal vias," are an important ingredient of the total thermal solution. The second step in the flow determines the optimal positions of thermal vias in the placement that provides an overall improvement in the temperature distribution. In realistic 3D technologies, the dimensions of these inter-tier vias are of the order of 5μm$\times 5\mu$m,

In principle, the problem of placing thermal vias can be viewed as one of determining one of two conductivities (corresponding to the presence or absence of metal) at every candidate point where a thermal via may be placed in the chip. However, in practice, it is easy to see that such an approach could lead to an extremely large search space that is exponential in the number of possible positions; note that the set of possible positions in itself is extremely large. Quite apart from the size of the search space, such an approach is unrealistic for several other reasons. First, the wanton addition of thermal vias in any arbitrary region of the layout would lead to nightmares for a router, which would have to navigate around these blockages. Second, from a practical standpoint, it is unreasonable to perform full-chip thermal analysis, particularly in the inner loop of an optimizer, at the granularity of individual thermal vias. At this level of detail, individual elements would have to correspond to the size of a thermal via, and the size of the FEA stiffness matrix would become extremely large.

Fortunately, there are reasonable ways to overcome each of these issues. The blockage problem may be controlled by enforcing discipline within the design, designating a specific set of areas within the chip as potential thermal via sites. These could be chosen as specific inter-row regions in the cell-based layout, and the optimizer would determine the density with which these are filled with thermal vias. The advantage to the router is obvious, since only these regions are potential blockages, which is much easier to handle. To control the FEA stiffness matrix size, one could work with a two-level scheme with relatively large elements, where the average thermal conductivity of each region is a design variable. Once this average conductivity is chosen, it could be translated back into a precise distribution of thermal vias within the element that achieves that average conductivity.

An algorithm to solving this problem is described in [18]. The technique has been applied to a range of benchmark circuits, with over 158,000 cells, and the insertion of thermal vias shows an improvement in the average temperature of about 30% [18], with runtimes of a couple of minutes. Therefore, thermal via addition has a more dramatic effect on temperature reduction than thermal placement.

Figures 11 and 12 show the 3D layout of the benchmark

`struct`, before and after the addition of thermal vias, respectively. As before, red and blue regions in the thermal map represent hot and cool regions, respectively. Remarkably, the greatest concentration of thermal vias is *not* in the hottest regions, as one might expect at first. The intuition behind this is as follows: if we consider the center of the uppermost tier, a major reason why it is hot is because the tier below it is at an elevated temperature. Adding thermal vias to remove heat from the second tier, therefore, effectively also significantly reduces the temperature of the top tier. For this reason, the regions where the insertion of thermal vias is most effective are those that have high thermal gradients.

C. Routing algorithms

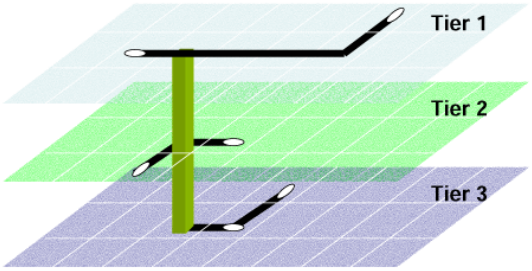


Fig. 13. An example route for a net in a three-tier 3D technology.

Once the cells have been placed and the locations of the thermal vias determined, the routing stage finds the optimal interconnections between the wires. As in 2D routing, it is important to optimize the wire length, the delay, and the congestion. In addition, several 3D-specific issues come into play. Firstly, the delay of a wire increases with its temperature, so that more critical wires should avoid the hottest regions, as far as possible. Secondly, inter-tier vias are a valuable resource that must be optimally allocated among the nets. Thirdly, congestion management and blockage avoidance is more complex with the addition of a third dimension. For instance, a signal via or thermal via that spans two or more tiers constitutes a blockage that wires must navigate around.

Each of the above issues can be managed through exploiting the flexibilities available in determining the precise route within the bounding box of a net, or perhaps even considering slight detours outside the bounding box, when an increase in the wire length may improve the delay or congestion or may provide further flexibility for inter-tier via assignment.

Consider the problem of routing in a 3-tier technology, as illustrated in Figure 13. The layout is gridded into rectangular tiles, each with a horizontal and vertical capacity that determines the number of wires that can traverse the tile, and an inter-tier via capacity that determines the number of free vias available in that tile. These capacities account for the resources allocated for non-signal wires (e.g., power and clock wires) as well as the resources used by thermal vias. For a single net, as shown in the figure, the degrees of freedom that are available are in choosing the locations of the inter-tier vias, and selecting the precise routes within each tier. The locations of inter-tier vias will depend on the resource contention for vias within each grid. Moreover, critical wires should avoid the high-temperature tiles, as far as possible.

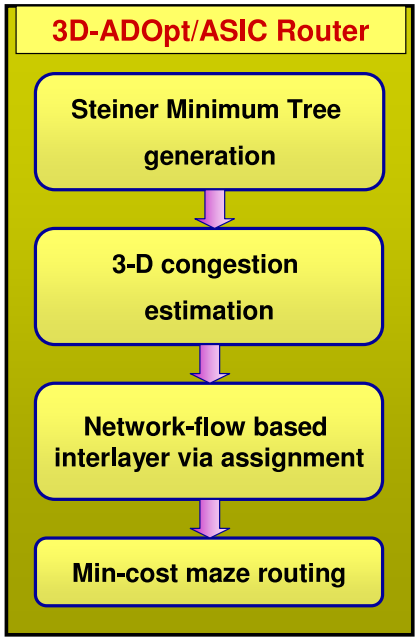


Fig. 14. Overall router algorithm.

The overall flow of the solution technique is shown in Figure 14. In the first step, a Steiner minimum tree is built for each net, with a cost that depends on the length of the net (with a penalty that discourages, but does not prohibit, the use of more than the minimum number of inter-tier vias). This Steiner structure still affords considerable flexibility in the routing through the availability of soft edges [19], and in each layer, assuming L and Z shaped routes, the distribution of wire congestion is determined in the second step. Next, a hierarchical procedure is followed for the precise assignment of inter-tier via locations: this corresponds to to a sequence of assignment problems (assigning nets to vias) that are solved using network-flow techniques. Once the inter-tier via locations are determined, the final step performs a minimum-cost maze routing in each layer, with a cost function that is based on the wire length, the temperature, and the congestion, to yield the global routing solution. Finally, any standard 2D detailed router may be used for detailed routing.

Figure 15 shows the average delay improvements for the critical sinks for a set of benchmark circuits, as compared to a router that ignores thermal effects. It can be seen that the range of improvement is between 12% an 30%, and the total wire length remains nearly unchanged from the non-thermal case.

IV. Conclusion

3D technologies offer great promise in providing improvements in the overall circuit performance. Physical design plays a major role in being able to exploit the flexibilities offered in the third dimension, and this paper has overviewed methods for placement and routing for both FPGA-style and ASIC-style designs.

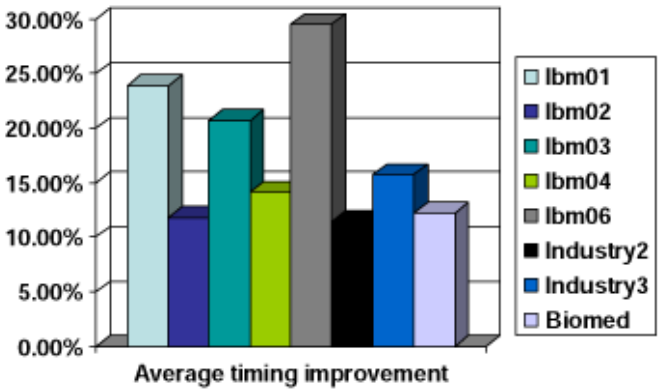


Fig. 15. Average delay improvement: thermal vs. nonthermal 3D routing.

There are several promising directions that remain to be explored, since 3D design enables other significant technologies: for example, 3D permits mixed-signal designs to isolate analog functionalities from digital blocks by placing them on different layers and/or using isolation ground planes between layers; heteregeneous integration is made possible, using dissimilar technologies in each tier (e.g., CMOS in one tier and GaAs in another); and so on. Each of these offer further challenges in the placement-and-routing arena, and are topics for further research.

REFERENCES

[1] K. Banerjee, S. J. Souri, P. Kapur, and K. C. Saraswat, "3-D ICs: A novel chip design for improving deep submicron interconnect performance and systems-on-chip integration," *Proceedings of the IEEE*, vol. 89, pp. 602–633, May 2001.

[2] J. Davis, R. Venkatesan, A. Kaloyeros, M. Beylansky, S. Souri, K. Banerjee, K. Saraswat, A. Rahman, R. Reif, and J. Meindl, "Interconnect limits on gigascale integration (GSI) in the 21st century," *Proceedings of the IEEE*, vol. 89, pp. 305–324, March 2001.

[3] K. W. Guarini, A. W. Topol, M. Leong, R. Yu, L. Shi, M. R. Newport, D. J. Frank, D. V. Singh, G. M. Cohen, S. V. Nitta, D. C. Boyd, P. A. O'Neil, S. L. Tempest, H. B. Pogpe, S. Purushotharnan, and W. E. Haensch, "Electrical integrity of state-of-the-art 0.13μm SOI CMOS devices and circuits transferred for three-dimensional (3D) integrated circuit (IC) fabrication," in *Technical Digest of the IEEE International Electron Devices Meeting*, pp. 943–945, 2002.

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

[5] R. Reif, A. Fan, K. N. Chen, and S. Das, "Fabrication technologies for three-dimensional integrated circuits," in *Proceedings of the International Symposium on Quality Electronic Design (ISQED)*, 2002.

[6] A. J. Alexander, J. P. Cohoon, J. L. Colflesh, J. Karro, E. L. Peters, and G. Robins, "Placement and routing for three-dimensional FPGAs," in *Fourth Canadian Workshop on Field-Programmable Devices*, pp. 11–18, 1996.

[7] S. Chiricescu, M. Leeser, and M. M. Vai, "Design and analysis of a dynamically reconfigurable three-dimensional FPGA," *IEEE Transactions on VLSI Systems*, vol. 9, pp. 186–196, Feb 2001.

[8] G.-M. Wu, M. Shyu, and Y.-W. Chang, "Universal switch blocks for three-dimensional FPGA design," in *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, pp. 254–259, 1999.

[9] V. Betz and J. Rose, "VPR: A new packing placement and routing tool for FPGA research," in *Field-Programmable Logic and Applications*, pp. 213–222, 1997.

[10] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar, "Multi-level hypergraph partitioning: Applications in VLSI design," in *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 526–529, 1997.

[11] J. Daz, J. Petit, and M. Serna, "A survey of graph layout problems," *ACM Computing Surveys Journal*, pp. 313–356, 2002.

[12] C. Ababei and K. Bazargan, "Non-contiguous linear placement for reconfigurable fabrics," in *Proceedings of the Reconfigurable Architectures Workshop*, 2004.

[13] P. Maidee, C. Ababei, and K. Bazargan, "Fast timing-driven partitioning-based placement for island style FPGAs," in *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 598–603, 2003.

[14] C. Ababei, H. Mogal, and K. Bazargan, "Three-dimensional place and route for FPGAs," in *Proceedings of the Asia-South Pacific Design Automation Conference*, 2005.

[15] D. L. Logan, *A First Course in the Finite Element Method*. Brooks/Cole Pub. Co., 3rd ed., 2002.

[16] C. Ho, A. E. Ruehli, and P. Brennan, "The modified nodal approach to network analysis," *IEEE Transactions on Circuits and Systems*, vol. 22, pp. 504–509, June 1975.

[17] B. Goplen and S. S. Sapatnekar, "Efficient thermal placement of standard cells in 3D ICs using a force directed approach," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 86–89, 2003.

[18] B. Goplen and S. S. Sapatnekar, "Thermal via placement in 3D ICs," in *Proceedings of the ACM International Symposium on Physical Design*, 2005.

[19] J. Hu and S. S. Sapatnekar, "A timing-constrained simultaneous global routing algorithm," *IEEE Transactions on Computer-Aided Design*, vol. 21, pp. 1025–1036, Sept. 2002.

Cristinel Ababei received the Ph.D. degree in electrical engineering from the University of Minnesota in 2004 and the B.S. degree in microelectronics from the Technical University of Iasi, Romania, in 1996. He joined Magma Design Automation in 2004. His research interests include CAD for layout and logic synthesis for robust VLSI circuits and FPGAs.

Yan Feng received his B.S. degree from University of Science and Technology, Beijing, in 1995. He received his M.S. degree and Ph.D. degree in computer science from the Colorado School of Mines in 2001 and 2004, respectively. Dr. Feng is currently a Post Doctoral researcher in the Electrical and Computer Engineering Department at University of Minnesota. His research interests include algorithms and CAD of VLSI circuits.

Brent Goplen received his B.A. degree in Chemistry and Biochemistry from Gustavus Adolphus College in 1997 and his M.S. Degree in Computer Engineering with a minor in Mechanical Engineering from the University of Minnesota in 2002. He is currently pursuing the Ph.D. degree in Electrical Engineering at the University of Minnesota. His research interests include CAD and physical design of next-generation circuits. Of particular interest are thermal issues, placement, and 3D ICs.

Hushrav Mogal received the B.E. degree in electronics engineering from the University of Mumbai (Bombay) in 2001. He is currently pursuing the Ph.D. degree at the University of Minnesota. His research interests are in computer-aided design tools and thermal issues for FPGAs and ASIC designs.

Tianpei Zhang received his Bachelor's degree in Applied Physics from University of Science and Technology of China in 1997, and Master's degree in Electrical Engineering from Purdue University in 2000. Currently he is pursuing Ph.D. degree in Electrical Engineering at the University of Minnesota. His research interest includes VLSI physical design, and design for manufacturability. He is a student member of ACM SIGDA.

Kia Bazargan received his Bachelors degree in Computer Science from Sharif University in Tehran, Iran, and his M.S. and PhD in Electrical and Computer Engineering from Northwestern University in Evanston, IL in 1998 and 2000 respectively. He is currently an Assistant Professor in the Electrical and Computer Engineering at the University of Minnesota. He has served on the technical program committee of a number of IEEE sponsored conferences (e.g., ISPD, ICCAD, ASPDAC, GLSVLSI). He was a guest co-editor of ACM Transactions on Embedded Computing Systems (ACM TECS), Special Issue on Dynamically Adaptable Embedded Systems in 2003. He is an Associate Editor of IEEE Transaction on Computer- Aided Design of Integrated Circuits and Systems. He was a recipient of NSF CAREER award in 2004. His research interests are computer-aided design, FPGAs and reconfigurable computing.

Sachin Sapatnekar received the B.Tech. degree from the Indian Institute of Technology, Bombay in 1987, the M.S. degree from Syracuse University in 1989, and the Ph.D. degree from the University of Illinois at Urbana-Champaign in 1992. From 1992 to 1997, he was an assistant professor in the Department of Electrical and Computer Engineering at Iowa State University. He is currently the Robert and Marjorie Henle Professor in the Department of Electrical and Computer Engineering at the University of Minnesota.

He has authored several books and papers in the areas of timing and layout. He has held positions on the editorial board of the IEEE Transactions on VLSI Systems, and the IEEE Transactions on Circuits and Systems II, IEEE Design and Test, and the IEEE Transactions on CAD. He has served on the Technical Program Committee for various conferences, and as Technical Program and General Chair for Tau and ISPD, and Techical Program co-chair for DAC. He has been a Distinguished Visitor for the IEEE Computer Society and a Distinguished Lecturer for the IEEE Circuits and Systems Society. He is a recipient of the NSF Career Award, three best paper awards at DAC and one at ICCD, and the SRC Technical Excellence award. He is a fellow of the IEEE.