CASCADE: A Standard Super-Cell Design Methodology with Congestion-Driven Placement for Three-Dimensional Interconnect-Heavy Very Large Scale Integrated Circuits

Lili Zhou, Student Member, IEEE, Cherry Wakayama, Student Member, IEEE, and C.-J. Richard Shi, Fellow, IEEE

Abstract—In this paper, CASCADE—a standard super-cell based design methodology, its supporting automated design flow, and associated design tools, are presented for three-dimensional (3D) implementations of a class of interconnect-heavy application-specific very large scale integrated (VLSI) circuits. In CASCADE, a system is first partitioned and synthesized using standard 2D design tools to a set of super-cells with the same height and varying width. With this, the 3D design is reduced to 3D super-cell placement and 3D via assignment. A congestion-driven simulated annealing method is used to find a 3D placement of super-cells to minimize the total wire length, the longest wire, the number of 3D vias and routing density. To efficiently estimate the routing density of a 3D grid space within the optimization loop, a simple probabilistic congestion model with an incremental congestion computation has been developed. Once the super-cell placement is fixed, the problem of assigning 3D-vias to accomplish minimal 2D routing densities and uniform 3D via distribution is solved by an efficient min-cost max-flow method. The proposed methods have been implemented and tested on a set of ISPD98 circuit benchmarks. Experimental results have shown that the proposed congestion-driven 3D super-cell placement and flow-based 3D via assignment tools have yielded satisfactory placement with small area, low-congestion, short wire length, few and uniformly distributed 3D vias. Further, an excellent correlation between routing density estimation by our model and the actual routing performed by a commercial router has been observed. We have applied the proposed 3D design methodology, tools, and flows to tape out an over 4-million-gate Low-Density Parity-Check (LDPC) decoder in a 3-tier 0.18µm FDSOI 3D CMOS process manufactured by MIT Lincoln Laboratory. The post-layout simulation of this DRC-clean layout design showed an about 10x improvement on the power-delay-area product compared to a 2D implementation in the same process.

Index Terms—Three-Dimensional Integrated Circuits, Congestion-driven Placement, 3D-via Assignment, Low-Density Parity-Check (LDPC) Decoder, Interconnect-heavy Integrated Circuits.

I. INTRODUCTION

Three dimensional integrated circuits (3D-ICs) are integrated circuits consisting of active devices not confined to a single planar layer [1][2][3][4][5]. Generally, 3D technology can be described as stacking several integrated circuits vertically with fine-grade third dimensional interconnects, called 3D-vias. Fig. 1 is an illustration of the cross section of three tiers that form a 3D integrated circuit, where each tier consists of a layer of active devices connected with multi-layers of metal layers (as in a conventional two-dimensional integrated circuit).

Fig. 1. Cross-section of three tiers integrated to form a 3D IC.

The revolution of micro fabrication technologies in the past forty years has lead to a constant reduction of transistor feature sizes over time, and provided ever greater densities of transistors and faster clock rates (“Moore’s Law”). It is now reaching to a degree where the spatial domain of signal propagation between clock edges is smaller than the total chip area. This, combined with un-manageable power dissipation, lead to a “scaling wall”, which prohibits further integration of system functionalities to silicon.

Considering for most such very large scale integrated circuits, half or more of silicon area, power dissipation, and delays are used by interconnects rather than devices, placing and wiring devices in the third dimension thus promises higher clock rates, less power dissipation, and higher integration density. Additional benefits of 3D integration include (1) easy integration of circuit intellectual properties (IPs) fabricated with different technologies, and (2) better overall system-on-chip (SOC) performance and less cost, where analog and radio frequency (RF) functionality can be integrated on one tier, where logic and memory circuitries on other tiers.

Despite several theoretical studies of the benefits of 3D integration for high-performance application-specific integrated circuits (ASICs), existing exploration of 3D
integration has been limited to applications with regular structures such as memory, imagers, and field-programmable gate arrays (FPGAs). This is primarily due to the lack of practical 3D computer-aided design (CAD) tools and complete design flows that allow the design of complex 3D ASICs.

Research effort has been dedicated to the development of 3D IC design tools. Ababei et al. developed physical tools for 3D FPGA design [10]. Das et al. implemented a 3D standard-cell placement and global routing tool [11]. Optimal placement was based on recursive min-cut partitioning of a circuit represented as a hypergraph. Inter-tier via (3D-via) minimization was sought by min-cut partitioning for tier assignment. During the partitioning phase, the aspect ratio was used to determine the wire-length minimization. Cong et al. proposed thermal-driven 3D floor-planning and routing algorithms [12][13]. Boplen and Sapatnekar presented thermal-driven force-directed standard-cell placement and thermal-via placement based on finite-element analysis [14][15].

In this paper, we present a standard super-cell design methodology and supporting CAD tools, referred to as CASCADE, for 3D-ICs. CASCADE is designed specifically to target at 3D implementation of very large scale ASICs consisting of millions of gates with a lot of interconnects among gates. An example of such a system is high-throughput Low-Density Parity-Check (LDPC) code decoders used in wireless communication and disk storage applications. Blanksby and Howland reported that more than half of wires are longer than half of the chip diameter, and occupy more than half of silicon area [16]. Furthermore, even the 2D implementation of this interconnect-heavy system broke down existing 2D design tools [16]. Our emphasis is thus to explore fine-grain third dimensional interconnects available in the latest 3D integration technologies to address this massive interconnect challenge. For example, one 3D-via can be placed every 1.8µm to 5µm in the MIT-Lincoln Lab’s 3-tier 0.18µm FDSOI 3D process [6]. It is predicted that the 3D-via density will increase with the progress in 3D integration technologies.

Three key techniques have been proposed and implemented in CASCADE to cope with both the complexity of interconnect and the complexity of functionality (millions of gates). First, a standard super-cell design and layout style has been introduced so that the number of super-cells at the top-level design is in the range of thousands, and the complexity of 3D physical design can be handled. Second, a congestion-driven 3D super-cell placement and 3D-via assignment method has been developed. A two-phase simulated annealing algorithm is adopted for 3D super-cell placement, which not only allows easy tradeoff of multiple objectives (area, wire length, 3D-via, and routing density), but also enables easy incorporation of additional design constraints (thermal and reliability). Finally, a fast probability based method for estimating the routing density during simulated-annealing based placement has been implemented, and it has been verified to correlate well with the routing density obtained from an actual commercial router. All these techniques have been implemented and integrated with existing commercial 2D design tool sets, leading to a complete 3D design flow. With CASCADE, we have successfully designed and taped out a 1024-bit ½-rate fully-parallel over 4-million-gate LDPC decoder using MIT-LL’s 3-tier 0.18µm FDSOI 3D CMOS process.

This paper is organized as follows. Section II introduces the proposed standard super-cell 3D ASIC layout scheme, design methodology and design flow. Section III presents the details of the two-phase congestion-driven 3D super-cell placement method. Section IV describes the 3D-via assignment algorithm based on the min-cost max-flow formulation. Experimental results from a set of benchmark circuits are presented in Section V. Section VI describes the implementation results of a 1024-bit ½ rate fully parallel LDPC decoder using the proposed design flow and CAD tools. Section VII concludes the paper.

II. STANDARD SUPER-CELL BASED 3D ASIC DESIGN METHODOLOGY AND DESIGN FLOW

A 3D IC is any integrated circuit in which active devices are not confined to a single plane [17]. We may consider such a circuit to be a collection of distinct 2D (conventional) ICs, each of which individually is called a tier or layer of wafers. These conventional layers, together with a means of interconnecting devices on separate layers, make up a 3D-IC.

The individual wafers in such a 3D IC are fabricated using conventional means and fused together with inter-wafer electrical and mechanical interconnects. This technology is called “wafer bonding”. Wafer bonding methods differ in terms of the bonding material and the order of fabrication operations. For instance, the MIT-LL uses parallel oxide bonding approach [18]. The individual wafers are processed before bonding. Formation of inter-wafer interconnects is the remaining back-end step. Inter-wafer interconnects, referred to as “3D-vias” in this paper, are etched through the entire metallization stack on the top of each wafer. The size and density of 3D-vias are determined mainly by the alignment requirements and via formation process. For MIT-LL’s 3-tier 0.18µm FDSOI 3D process, each 3D-via occupies 1.75µm × 1.75µm, and the minimum distance between any two 3D-vias is from 1.8µm to 5µm depending on the 3D-via density required to meet the stringent alignment constraints. Compared to traditional multi-chip module (MCM) and system-in-package (SiP) technologies, 3D IC offers much finer-grain 3D interconnects, and thus offers much greater potential for improving system performance and integration density.

On the other hand, a 3D-via still uses much bigger silicon area than a conventional inter-metal via, introduces more delays, and causes the fabrication and reliability problem. Specific 3D-via placement strategy should be performed not only to satisfy stringent alignment constraints between neighboring tiers but also to reduce the routing density and achieve more uniform 3D-via distribution. Furthermore, whenever possible, the number of 3D-vias used should be reduced.

With these observations and with the target to implement interconnect-heavy very large scale integrated circuits, we
propose the following “standard super-cell” based layout scheme. As illustrated in Fig. 2, standard super-cells are layout macros with the same height and varying width. Standard super-cells are placed in rows. Spaces are reserved between standard super-cell rows for 3D-vias and buffers. Using this scheme, the 3D-via alignment constraint can be easily satisfied and does not affect super-cell placement. Buffer insertion is also less constrained with respect to the silicon resource, therefore both the 2D long wires and 3D long wires crossing different tiers can be handled easily. To accommodate routing congestion for interconnect-heavy systems and especially for the MIT-LL’s 3-tier 0.18µm FDSOI 3D CMOS process where only 3 metal layers are available per tier, vertical routing channels are also reserved.

With this proposed 3D standard super-cell layout scheme, 3D design can be accomplished with a flow similar to typical 2D design, as shown in Fig. 3. For 2D design, the process starts with a high-level VHDL or Verilog description of the system to be designed typically at the register-transfer level (RTL). Then, RTL and logic synthesis tools are used to generate a gate-level circuit netlist. A floorplan is developed, given the netlist and physical parameters for individual logic gates, which are placed in optimum locations on the chip. The resulting placement is wired or routed and buffers are inserted to break long lines during this procedure. The placed-and-routed circuit layout is then verified so that it meets the specifications [17].

For 3D design, a design is first partitioned into thousands of blocks. Then, for each block, traditional 2D logic and physical design tools are used to synthesize super-cells with the same height and varying width. The super-cell height is determined heuristically with consideration of chip aspect ratio and area constraints. Next, a 3D placer finds an optimal 3D super-cell placement. Once 3D placement of super-cells is fixed, the number of 3D-vias to be used is determined. Then 3D-via assignment is performed to assign 3D-vias to the reserved 3D-via locations (as shown in Fig. 2) with the objectives to reduce the 3D-via distribution density and the routing density on each tier. After 3D-via assignment, commercial 2D detailed routing tools are used for routing on each tier. Since super-cells are distributed over several tiers, tools for 3D buffer insertion and clock distribution are developed to handle long interconnects across tiers. Finally, 3D layout-vs-schematic (LVS) verification is performed to ensure the correctness between the final physical design and the original circuit netlist.

3D-specific tools have been developed including a 3D placer, a 3D-via assignment tool, a 3D buffer insertion and clock distribution tool, and finally a 3D LVS tool. In CASCADE, we focus on the development of an effective 3D placement and 3D-via assignment solutions, while 3D buffer insertion, clock distribution and LVS are accomplished by scripting existing commercial 2D tools.

III. 3D STANDARD SUPER-CELL PLACEMENT

A. Problem Formulation and 3D Grid Model

We are given a collection of super-cells with the same height and varying width. Each super-cell is a finished 2D layout macro with ports (input, output, power supply and ground pins) on the boundaries, and is described in the standard LEF format [26]. We are also given a collection of nets, each specifying a set of super-cell ports to be wired together. The entire 3D layout space is considered as a stack of multiple layers (called tiers) of 2D layout space. The problem of 3D placement is to find suitable physical locations for each super-cell on the entire 3D layout space to optimize given objective functions (such as area, total wire length, the number of 3D-vias etc.), subject to certain constraints. An example of constraints is the avoidance of super-cell overlap. The output of 3D super-cell placement is in the standard DEF format [26].

To describe the entire 3D layout space, we extend the conventional 2D grid model to 3D grid model. Fig. 4(a) shows the 2D grid model used for a 2D (one tier) physical design. Fig. 4(b) shows the 3D grid model. Each super-cell is considered to
be a point to be placed in a 3D location denoted by \((x, y, z)\). A 3D location is a cubic region (also called a bin or a region). The number of super-cells that can be placed in a 3D location legally (without overlap) is called the area-density (bin-density) capacity. In the grid model, we also refer to a cubic region (bin) as a node, the borders of a region as edges or links. Associated with each boundary of a cubic region is a routin-density capacity, which denotes, respectively, the number of wires can be placed in each of the \(x\), \(y\), \(z\) directions. The \(z\) direction capacity is the number of 3D-vias that can be allowed, i.e., 3d-via density capacity.

Similar to 2D placement, we use the 3D Manhattan geometry to estimate the wire length required to connect two super-cells located on two 3D locations as the number of grids spanned, and use Steiner trees and minimum-spanning trees for two-terminal and multiple-terminal net wire length estimation [24].

For the proposed super-cell scheme, each row of the super-cell placement has the same height and also has reserved area for 3D-via and buffer placement above or below the super-cell row. By limiting that each placement region to hold only one super-cell row on each tier, the non-overlap placement in the \(y\) and \(z\)-direction can be guaranteed.

If the area-density capacity for each location is 1, then the resulting placement is legal, i.e., no super-cell overlap, and is final. However, if the area-density capacity for each location exceeds 1, 3D-placement will consists of two steps: coarse global placement and detailed local placement. Coarse placement is to find super-cell placement to satisfy the area capacity constraint, and detailed local placement is performed at each location to obtain the final placement of super-cells without any overlap.

B. Objective Function

The objective for 3D standard super-cell placement is to place all the given super-cells on 3D locations with the total wire length, longest wire length, 3D-via number, and routing density minimized. We use the objective function (1)

\[
F = F_{\text{length}} + F_{3D-via} + F_{\text{density}}.
\]  

The first term in the objective function (1) is the weighted total wire length \(F_{\text{length}}\). The length cost of each net is the summation of horizontal \(x\)-direction span times the horizontal weighting factor, and the vertical \(y\)-direction span of the net times the vertical weighting factor. The sum of all the nets gives the wire length cost. Let \(M\) be the total number of nets, \(X_i\) be the span of net \(i\) in the \(x\) direction, \(Y_i\) be the span of net in the \(y\) direction, \(W_x\) be the weighting factor for net \(i\) in the \(x\) direction, and \(W_y\) be the weighting factor for net \(i\) in the \(y\) direction. Then we have

\[
F_{\text{length}} = \sum_{i=1}^{M} (W_x \cdot X_i + W_y \cdot Y_i).
\]

The weighting factors \(W_x\) and \(W_y\) are determined by (3) and (4), respectively, to minimize the total wire length, the number of long wires, and the length of the longest wires.

\[
W_x = \begin{cases} 
1, & X_i \leq X_{th} \\
\alpha, & X_i > X_{th}
\end{cases}
\]

\[
W_y = \begin{cases} 
1, & Y_i \leq Y_{th} \\
\beta, & Y_i > Y_{th}
\end{cases}
\]

\(X_{th}\) and \(Y_{th}\) are user-defined threshold values for \(x\)-length and \(y\)-length, respectively, where \(\alpha\) and \(\beta\) are penalty factors to penalize long wires whose lengths are larger than threshold values. Penalty values of \(\alpha\) and \(\beta\) are selected to make a tradeoff between the area, delay, and the number of 3D-vias.

The second term in the objective function (1) is the 3D-via cost \(F_{3D-via}\). The function \(F_{3D-via}\) is given in (5).

\[
F_{3D-via} = \sum_{i=1}^{M} W_{via} \cdot Z_i
\]

where \(W_{via}\) is the penalty weight for 3D-vias, \(Z_i\) is the number of tiers that net \(i\) will go between, and \(M\) is the total number of nets.

The last term in the objective function (1) is the routing-density cost \(F_{\text{density}}\). The routing-density cost is the weighted sum of the estimated routing densities of all wires. Let \(K\) be the total number of super-cells, \(D_x(i), D_y(i)\) and \(D_z(i)\) be \(x\), \(y\), and \(z\) direction routing densities associated, respectively, with super-cell \(i\). Let \(W_{dx}, W_{dy}\) and \(W_{dz}\) be, respectively, the weighting factors for \(x\), \(y\), and \(z\) routing densities. Then we have

\[
F_{\text{density}} = \sum_{i=1}^{K} \left( W_{dx} \cdot D_x(i) + W_{dy} \cdot D_y(i) + W_{dz} \cdot D_z(i) \right).
\]

The weighting factors \(W_{dx}, W_{dy}\), and \(W_{dz}\) are determined by (7), (8), and (9), respectively, to minimize routing densities of congested regions, which are defined as regions with routing densities larger than some threshold values. Let user-defined threshold values for the \(x\)-density, \(y\)-density, and \(z\)-density be \(D_{x,th}, D_{y,th}\), and \(D_{z,th}\), respectively. Then we have

\[
W_{dx} = \begin{cases} 
1, & D_x(i) \leq D_{x,th} \\
\frac{1}{c}, & D_x(i) > D_{x,th}
\end{cases}
\]

\[
W_{dy} = \begin{cases} 
1, & D_y(i) \leq D_{y,th} \\
\frac{1}{d}, & D_y(i) > D_{y,th}
\end{cases}
\]

\[
W_{dz} = \begin{cases} 
1, & D_z(i) \leq D_{z,th} \\
\frac{1}{e}, & D_z(i) > D_{z,th}
\end{cases}
\]

The penalty values of \(c\), \(d\), and \(e\) are determined by balancing the requirements on wire length, 3D-via usage, and routing-density minimization.

The values of weighting factors were assigned to penalize long wires and dense regions. In practice, the threshold values are determined by performing a set of test runs to study the distributions of wire lengths and routing densities. From the test runs, the average and standard deviation values of wire lengths and routing densities were computed. The threshold
value is then set to be the sum of the average value and the standard deviation.

Among the three terms in the objective function (1), $F_{\text{density}}$ is most costly to compute. Section III.C and Section III.D will present an efficient routing density calculation scheme. Section III.E will describe a two-phase congestion-driven optimization strategy.

### C. Probabilistic Congestion Estimation

We adopted a probabilistic model similar to [20][21]. However, to include congestion estimation in the placement stage, a much faster computation method is needed.

The problem of routing-density estimation can be described as follows: given a 3D placement grid $G$ and a set of two-terminal nets $N_j$, $1 \leq j \leq M$, where $M$ is the total number of nets, calculate the usage probability $P(l_i)$ of each link $l_i$, $1 \leq i \leq K$. Multi-terminal nets can be decomposed into a set of two-terminal nets by using Minimum Spanning Tree (MST) [20].

For each net, we compute the usage probability $P_{N_j}(l_i)$ of all the links in the entire region $R$. $P_{N_j}(l_i)$ is computed as (10):

$$P_{N_j}(l_i) = \frac{\text{Number of routes in } R \text{ that use the link } l_i}{\text{Total number of routes in } R}. \quad (10)$$

In defining a route, we make the following assumptions:

(i) All nets are routed along the links.

(ii) All nets are routed with shortest lengths.

(iii) All possible routes for each net have equal usage probability, which is independent of the number of direction changes (from a horizontal link to a vertical link and vice versa) that the net makes.

The expected usage or the routing density $P(l_i)$ for each link $l_i$, $1 \leq i \leq K$, in the 3D placement grid is obtained by summing corresponding routing densities contributed by all the nets. It is calculated as

$$P(l_i) = \sum_{j=1}^{M} P_{N_j}(l_i). \quad (11)$$

A simple example of a two-pin net AB that starts from A and ends at B on a 2 x 2 2D grid is illustrated in Fig. 5. There are a total of six possible routes in this region. Consider the lower middle vertical link, which is shaded in the figure. Two out of six possible routes go through this link (labeled as route 2 and 3, respectively). Therefore the usage density of the link for net AB is 1/3. Similarly, we can compute the probabilistic usages of all the links on the entire region $R$ for the net. Fig. 6 summarizes the probabilistic routing densities of this net in region $R$ using a $3 \times 3$ matrix, where the two numbers in an entry are the horizontal and vertical routing densities corresponding to a vertex in $x$ and $y$ directions.

### D. Incremental Congestion Computation

Different from [20] and [21], a fast incremental procedure to compute the routing densities has been developed. It consists of three steps:

**(i) Compute the route matrix for each net:** A route matrix stores the number of possible routes to reach one node from adjacent $x$, $y$ and $z$ nodes. It is a 3D matrix and each entry corresponds to the route count of the node at $(x, y, z)$ location. The entries in the route matrix are computed from the destination toward the source. The procedure for a two-tier case is demonstrated in Fig. 7. The arrows represent the directions along which the route matrix is computed. Note that the route count for the destination node is set to 1.

**(ii) Compute the 3D routing-density matrix for each net:** A routing-density matrix stores $x$, $y$ and $z$ densities associated with each node. It is a 3D matrix and each entry in the matrix has three sub-entries corresponding to $x$, $y$, and $z$ direction routing densities associated with the node at the $(x, y, z)$ location. We define the $x$-density associated with the node to be the density of the link represented by the outgoing arrow in the $x$-direction. The $y$-density and $z$-density are defined similarly. We compute the routing-density matrix by propagating the routing densities from the source toward the destination. This is illustrated in Fig. 8. The arrows indicate the directions along which routing densities are propagated. The $x$, $y$ and $z$ direction routing densities associated with each node are computed as follows:
The total routing density is the summation of the $x$, $y$, and $z$ routing densities entering the node, which is given in (15). Note that the total entering routing density of the starting node is set to 1. Each routing-density value is equal to $P_{xy}(l_i)$ in (10).

The description of simulated annealing applied to the general placement problem can be found in [22]. Some specific implementation considerations are summarized as follows.

1. Generating new states: The implementation begins with a random initial placement that satisfies all the area capacity constraints, the routing capacity constraints, and the 3D-via density constraints. A new state or candidate is generated by exchanging the super-cells of two locations subject to both the area capacity constraints (the maximal number of super-cells can be placed in one location) and the routing capacity and 3D via capacity associated with each cubic boundary (the maximal number of wires crossing in the x and y direction boundaries and the number of 3D-vias in the z direction boundary).

2. Objective function: Whenever a new state is generated, wire lengths and routing densities are updated only for the nets associated with the super-cells being moved. Since the routing-density calculation is costly and the wire length and routing density are correlated to the first order, annealing is performed in a two-phase procedure. In Phase 1, the first two terms, $F_{\text{length}}$ and $F_{3D-via}$ in the objective function are used. It was observed that minimizing total wire length also reduces the total routing density. However, it may not achieve the balanced routing-density distribution. Therefore in Phase 2, the routing-density cost is added to the objective function.

3. Cooling schedule: In this implementation, an exponential cooling schedule $(T_k = T_0 \beta^k)$ is used, where $T_0$ is the initial temperature, $\beta$ is a factor by which multiplicative steps the temperature is lowered and $T_k$ is the $k^{th}$ temperature step. In our implementation, the $\beta$ value is chosen to be 0.95.

4. Inner loop criterion: The inner loop criterion is implemented by specifying the number of new states generated for each temperature step and by recording the number of successful moves at each temperature step. The number of new states is specified as the square of the number of super-cells and the maximum number of successful moves is defined as 10% of the number of new states. The inner loop is escaped when the number of inner loop iterations reaches the specified number of new states or when the number of successful moves is greater than the specified maximum number of successful moves, whichever comes first.

5. Outer loop criterion (stopping criterion): The stopping criterion is implemented by recording the value of the cost function at the end of each stage of the annealing process (each temperature step). The program is stopped when the cost function value remains unchanged for three consecutive stages or when the pre-determined maximum number of temperature steps is reached.

6. Super-cell overlap consideration: If the area bound in each placement location is strictly enforced, a placement result can be legalized to an overlap-free detailed placement without moving any super-cell out of the placement location assigned. However, in practice, we allow certain overflow as some percentage of the area capacity constraint. Similar to 2D placement as in [22], some overlap is allowed in the early steps of simulated annealing, but a large penalty is enforced at the late stage. In practice, some relaxation in the horizontal direction is
typically acceptable, and the detailed overlap-free placement can be obtained without significant super-cell movement.

IV. 3D-VIA ASSIGNMENT ALGORITHM

Once the 3D placement of super-cells is obtained, the physical positions of super-cells are fixed, therefore the total number of 3D vias is also determined. However, the exact physical positions of the 3D vias are to be determined. The wire to connect two super-cells on different tiers can go through all possible reserved 3D-via locations while both the resulting routing density on each tier and 3D-via density can be different. Therefore the second step in our flow after 3D placement is to assign 3D-vias to proper locations to minimize the individual-tier routing density and 3D-via density. In our implementation, once we know where to place the 3D-vias, we generate separate DEF files for each tier with the physical information about super-cells and 3D-vias, as well as the connectivity information. Then any commercial 2D detailed router could be used to complete the final layout for each tier.

The 3D-via assignment problem can be formulated as follows:

**Problem of 3D-via Assignment:** Given a set of \( n \) 3D-vias \( V_{3D} = \{ v_1, v_2, ... , v_n \} \) and the corresponding \( m \) possible locations \( L = \{ l_1, l_2, ... , l_m \} \), the problem of 3D-via assignment is to find a set of \( n \) locations \( P = \{ p_1, p_2, ... , p_n \} \), where \( P \subseteq L \), such that \( p_i \) is the final location for 3D-via \( v_i \). If the cost of assigning \( v_i \) at \( l_j \) is \( c_{ij} \) \((j=1,2,...,m)\), which represents the resulted routing-density change, we also would like to minimize the total cost of assigning the \( n \) 3D-vias \( \sum_{i,j,p \in P} c_{ij} \).

To solve the 3D-via assignment problem, we first construct a network graph. Then run a min-cost max-flow algorithm [23],[24]. Fig. 9 shows the network flow graph \( G=(V,E) \) constructed for the problem.

In the graph,
\[
V = \{ S, D \} \cup V_{3D} \cup L.
\]
\[
E = \{(S,v_i) | v_i \in V_{3D}\} \cup \{(v_i,l_j) | v_i \in V_{3D}, l_j \in L\} \cup \{(l_j,D) | l_j \in L\}
\]
Edge Capacity: \( U(S,v_i) = 1, U(v_i,l_j) = 1, U(l_j,D) = K_j \).

Cost Function: \( C(v_i,l_j) = C_{ij} \), for other edge \( e \in E \), \( C(e) = 0 \).

The graph consists of a source vertex \( S \) supplying 3D-vias, a set of \( n \) 3D-via vertices \( V_{3D} = \{ v_1, v_2, ... , v_n \} \), a set of \( m \) location vertices \( L = \{ l_1, l_2, ... , l_m \} \) and a destination vertex \( D \) collecting all the 3D-vias. The edge capacities \( U(l_j,D) = K_j \) represent that a corresponding bin \( j \) can hold \( K_j \) 3D-vias. The cost of assigning a 3D-via \( v_i \) to location \( l_j \) is \( C_{ij} \), where \( C_{ij} \) is the resulted change in the routing density cost when assigning a 3D-via \( v_i \) to location \( l_j \).

The incremental routing-density calculation method described in Section III.C and Section III.D can be used here. During the super-cell placement stage, we estimate the routing density for each net going from \( v_i \) to \( v_j \) by a probabilistic measure. In the current stage, assigning a 3D-via \( v_i \) to location \( l_j \) actually means to cut the original net from \( v_i \) to \( v_j \) into two parts: \( v_i \) to \( l_j \) and \( l_j \) to \( v_j \). Similarly, we assume that all the nets are routed by shortest paths, i.e., the 3D-via \( l_j \) can only locate within the boundary box of net \( v_i \) to \( v_j \). Using network-flow computation on \( G \), we assign 3D-vias to preserved feasible locations that minimize the routing-density cost introduced. The 3D-via assignment algorithm is summarized as follows:

<table>
<thead>
<tr>
<th>Algorithm 3D-via Assignment by Network Flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Construct the network graph ( G(V,E) )</td>
</tr>
<tr>
<td>2. Assign the edge capacity for each edge</td>
</tr>
<tr>
<td>3. Calculate the cost for each ( C_{ij} )</td>
</tr>
<tr>
<td>4. Run min-cost max-flow algorithm for ( G(V,E) )</td>
</tr>
</tbody>
</table>

Finding a min-cost max-flow for a network is a classical problem for which several optimal polynomial-time algorithms are available, such as the flow augmentation method [23],[24].

V. EXPERIMENTAL RESULTS

All the proposed algorithms have been implemented in C++ and tested on ISPD98 benchmark circuits. All the experiments reported here are conducted using a SUN Fire V480 server with 900 MHz UltraSparc-III processors and 16GB RAM.

We first test the effectiveness of our algorithms by prioritizing different objectives by varying the penalty weights. We performed optimization in three configurations: in the first the total wire length is considered, in the second both the wire length and 3D-via usage is considered, and in the third, the wire length, 3D-via usage, and routing density are jointly minimized. The best tradeoff is achieved when the wire length, routing densities, and the number of 3D-vias are optimized together.

Experimental results are summarized in Table I. In the table, \( \text{obj (0)} \), \( \text{obj (1)} \), \( \text{obj (2)} \), and \( \text{obj (3)} \) represent \( F_{\text{length}} \) considering the total wire length only, \( F_{\text{length}} \) considering both the total wire length and longest wire length, \( F_{3D-via} \), and \( F_{\text{density}} \) in (1), respectively. For circuits ibm01, ibm05, and ibm18, we first apply the hMetis [19] to partition the circuits into, respectively,
200, 500, and 1000 super-cells. Column 3 in Table I lists the number of global nets of the hMetis partition. Then 8×9×3, 12×14×3 and 18×19×3 grid models are used, respectively, in 3D congestion analysis for each case. These grid sizes are adjusted corresponding to the number of super-cells and the W/L ratio of the chip. It is observed that optimizing the total wire length alone results in a number of longer wires, although the total length is minimized. Comparable wire length distributions are observed from optimizations with and without the routing-density objective. By including the 3D-via count in the objective function, the number of 3D-viias is reduced by 65% – 79%. Finally, after the routing-density objective is considered, a 13% – 22% reduction in the maximum routing density is achieved while maintaining all other objectives.

After 3D placement, we further perform 3D-via assignment to reduce the 3D-via density and the routing density of each individual tier. Table II shows the results of example ibm01. We can see that the 3D-via density can be further reduced by about 48%, while achieving almost the same routing density on each tier.

VI. REAL-WORLD DESIGN CASE: 1024-BIT ½-RATE LDPC DECODER

We have applied the developed tools, proposed design flow and methodology to the design of a high-throughput low-density parity-check (LDPC) code decoder. With the error correction capacity approaching the Shannon limit, the LDPC has been selected as the core of several emerging wireless communication standards. LDPC also has applications in data storage, optical communication, and quantum computing. However, VLSI implementation of high-throughput LDPC decoders presents a challenge due to the fact that half of the total wires traverse the chip from one side to the other.
reported by a design from Bell Labs [16], more than half of the silicon area and power consumption are used by interconnects.

We have implemented the fully parallel architecture of a 512 × 1024 LDPC decoder [16] using a MIT Lincoln Lab’s 3-tier 0.18μm FDSOI 3D CMOS process. This LDPC decoder is partitioned into 1,536 super-cells of two types: check nodes and variable nodes. A check-node super-cell is implemented by 812 (509 ~ 1121) standard cells, and a variable-node super-cell is implemented by 339 standard cells. There are 24,576 global nets connecting these super-cells. The optimal 3D/3-tier placement of these 1536 super-cells is obtained using the proposed two-phase congestion-driven 3D super-cell placement algorithm. Then 3D-via assignment is optimized to reduce further the routing congestion on each tier. Routing is performed on each individual tier using Cadence’s Silicon Ensemble (SE) [27], followed by 3D buffer insertion and LVS using in-house scripts.

Fig. 10(a) and Fig. 10(b) show the two routing-density maps, where Fig. 10(a) is obtained from running Cadence’s SE router on a placement described in [16], and Fig. 10(b) is obtained from the placement obtained using the proposed method. Clearly the routing congestion obtained by our tool has been reduced significantly and is more uniform.

We further evaluate the quality of our routing-density estimation method by comparing our estimation with the actual routing results from a commercial router, Cadence SE. As shown in Table III, six cases were tested – three 2D placements and three tiers of the optimized 3D placement. 2D placements include initial random placement, wire length optimized placement, and both wire length and density optimized placement. 3D optimized placement is achieved by balancing all the objectives in (1). Each tier of the 3D structure is treated as a traditional 2D placement. The routing density is measured by the number of routing tracks used. These results show that our wire-density estimation results are well correlated with the commercial tool actual routing results.

Table IV shows the CPU time comparison between using our incremental density computation with in the simulated annealing density-driven placement loop and using non-incremental method. Our incremental placement method performs up to 100 times faster than the non-incremental method.

Table V summarizes the layout statistics of the 2D and 3D implementations. The 2D design is accomplished by placing all
We can see that the 3D implementation achieves a significant advantage over the same 2D implementation in terms of wire length, area, clock skew, and buffer used. The improvement in terms of area-delay-power product is about an order of magnitude (2.5 \times 1.75 \times 2.5 = 10.9).

The final layout of the three-tier design, connected by the densely distributed 10,631 3D-vias, is shown in Fig. 12. To verify the functional correctness and error-correction capability of the LDPC decoder, the relationships of the BER (Block Error Rate) vs. SNR (Signal to Noise Ratio), as well as the decoder convergence iteration vs. SNR, were simulated using Cadence’s Verilog-XL simulator. Using a LDPC encoding and channel transmission software package [25], different levels of noise were added into the original code and transmitted through an Additive White Gaussian Noise (AWGN) channel, the number of information bits that are not corrected through our LDPC decoder is counted. The results are shown in Fig. 13.

Fig. 14 plots the power consumption results of post-layout netlist, obtained by running Cadence power analyzer [28]. It shows a significant power reduction with increasing switching activities (required for the low SNR channel decoding).

VII. CONCLUSION

We have developed a 3D design methodology and supporting tools, and explored the benefits of 3D technology for very large scale interconnect-heavy ASIC design. Our contribution includes: (1) a congestion-driven simulated annealing based 3D placement algorithm, which allows us to target multiple design objectives, (2) an efficient and effective 3D congestion analysis method and the inclusion of congestion metrics within the 3D placement, and (3) a super-cell based design methodology and supporting CAD tools that enables the first practical 3D design flow for tape-out using commercial 2D design environments. The proposed methodology and design flow enable us to successfully tape out the first-known multi-million gate ASIC in a 3D process. Post-silicon simulation of this specific 3D implementation demonstrated an order of magnitude improvement in terms of power-delay-product over the corresponding 2D implementation.

To the best of our knowledge, this is for the first time that a large ASIC has been implemented in silicon using both 2D and 3D technologies. The 10x improvement in the power-delay-area product comes from the implementation of interconnect-heavy circuits implemented with fine-grain 3D integration (only 3 tiers in this case). We note the rapid and exciting progresses in the development of fine-grain 3D integration technologies to provide even finer 3D interconnects and many more tiers, as for example, reported recently by IBM in [5]. The proposed CASCASDE standard super-cell design methodology provides a way to cope with the complexity of functionality and interconnect in 3D design, and would enable the further exploration of the potential of fine-grain multi-tier 3D-interconnect technologies in extending the life-time of Moore’s law.
A commonly cited roadblock for the wide use of 3D integration is the thermal issue. This is due to the poor thermal conductivity of the silicon insulation material used between the tiers, as well as the potentially high power density due to the increase of integration density. However, we believe that the thermal issue can be coped with effectively by innovations in fabrication, CAD tool development and design methodologies. There are intensive research efforts directed to the development of various micro cooling schemes. The use of thermal vias can help heat dissipation. Some recent work has shown that some thermal problems can be solved by incorporating thermal modeling as part of 3D physical design optimization objectives [12][13][14]. Careful 3D placement of circuit modules on tiers has been reported to produce a design that yields more-than-10-degree less thermal gradient than an arbitrary placement [29]. Furthermore, a coupled circuit-thermal design verification and optimization flow has been developed and demonstrated with existing commercial 2D IC design tools and standards [29]. Finally, innovation in circuit and architecture design methodologies can reduce the thermal stress on the circuit performance and reliability. Some techniques include thermal-averaging for current steering in high-speed ADC design [30] and low-power pipelining and dynamic voltage and frequency scaling develop for 2D low-power CMOS digital design [31].

ACKNOWLEDGMENTS

The authors thank Dr. Nuttorn Jangkrajparn for his help in developing the simulated-annealing based 3D placer. Dr. Bo Hu, Dr. Nuttorn Jangkrajparn, and Dr. Guoyong Shi have helped the tape out of the 3D LDPC decoder using the proposed CAD flow. The authors wish to thank Dr. Craig Keast, Dr. Peter Wyatt, and Dr. James Burns from MIT Lincoln Lab for their contribution in 3D fabrication, Prof. Jason Cong and Prof. Sachin Sapatnekar and their collaboration on 3D physical design. The authors are grateful to Dr. Daniel Radack from DARPA/MTO for his vision and support of this research.

REFERENCES