Design Tools for 3-D Integrated Circuits

Shamik Das
M.I.T., Department of E.E.C.S.
60 Vassar St. Room 39-625
Cambridge, MA 02139
e-mail: shamikd@mit.edu

Anantha Chandrakasan
M.I.T., Department of E.E.C.S.
50 Vassar St. Room 38-107
Cambridge, MA 02139
e-mail: anantha@mtl.mit.edu

Rafael Reif
M.I.T., Department of E.E.C.S.
50 Vassar St. Room 38-427
Cambridge, MA 02139
e-mail: reif@mtl.mit.edu

Abstract— We present a set of design tools for 3-D integration. Using these tools – a 3-D standard-cell placement tool, global routing tool, and layout editor – we have targeted existing standard-cell circuit netlists for fabrication using wafer bonding. We have analyzed the performance of several circuits using these tools and find that 3-D integration provides significant benefits. For example, relative to single-die placement, we observe on average 28% to 51% reduction in total wire length.

I. Introduction

As performance demands on integrated circuits increase, the interconnect structures on these chips consume more and more of the available power and delay budgets. Shrinking technology feature sizes coupled with growing overall chip dimensions lead to increasingly expensive global and semi-global wires, to the point that these wires dominate the delay and power budgets of circuits.

A potential means of relief is to increase the number of "nearest neighbors" seen by a transistor, gate, or module, by using **three-dimensional integration**. In a 3-D integrated circuit, transistors may be fabricated on top of other transistors, resulting in multiple planes of active components. These transistors may then be wired to other transistors on the same plane, to transistors on different planes, or both, depending on the process technology. Suggested fabrication approaches include multichip-module packaging, solid-phase recrystallization, and wafer bonding [1, 2, 3]. While we focus here on wafer-bonding technology, as shown in Fig. 1, the user may configure our tools for the other approaches.

While there is a long history of development of 3-D integration technology, only recently have such technologies entered the marketplace. Predictive analysis has been done to explore the possibilities afforded by 3-D integration [4], but only with actual circuits can we be certain of the benefits available. Conversely, these benefits cannot be realized without tools to allow designers to target large, sophisticated designs for 3-D integration.

We present here a set of physical design tools for 3-D integration. The tool chain consists of a standard-cell placement and global routing tool, as well as a 3-D layout editor based on the Berkeley tool MAGIC [5]. Together, these tools form a design flow for 3-D integrated circuits. Using these tools,
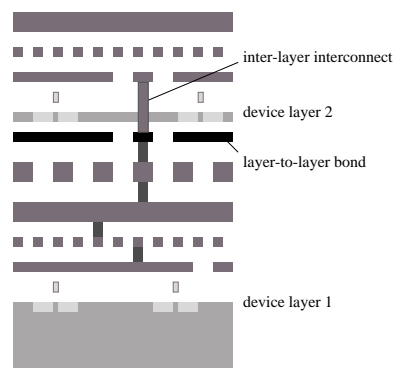

Fig. 1. Wafer-bonded structure with two device layers and copper interconnect interface.

we can perform interesting analyses of benchmark circuits to verify predictive models, as well as target actual circuits for fabrication.

II. 3-D Standard Cell Tool Design

A. 3-D Standard Cell Placement

Conventional standard-cell placers fall into several categories based on their core algorithms. Typical techniques include quadratic placement, simulated annealing, and placement by min-cut partitioning [6, 7, 8]. Placers incorporating each of these techniques have produced results of similar quality. Therefore, in selecting the core algorithm for our 3-D standard-cell placer, we are motivated by factors such as running time, and ease of extension to the 3-D case, in addition to placement quality.

With the advent of fast, high-quality partitioners [9, 10], placement by min-cut partitioning has become increasingly attractive. Recent results indicate that placers incorporating min-cut partitioning, at least at the global level, are capable of achieving high-quality placements in significantly less time than placers using other strategies [11, 12]. Furthermore, it is natural to think of a 3-D integrated circuit as being partitioned into device layers or planes.

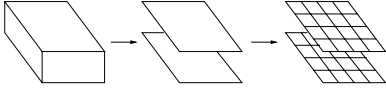Thus, our placement framework consists of embedding a hy-

Fig. 2. Partitioning strategy where plane assignment is done first in order to minimize the number of inter-plane vias.
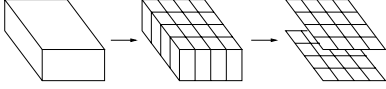

Fig. 3. Partitioning strategy where plane assignment is done by considering aspect ratio in order to minimize total wire length.

pergraph representation of a netlist into a rectangular block that represents the available die area. Each node of the hypergraph is a cell and the weight of each node is the cell area. Hyperedges correspond to electrical nets. We assume that the dimensions of the block (number of rows, width of each row) are fixed *a priori* (i.e. a fixed-die context). For the 3-D case, given a set number of device layers (specified at run-time by the user), we adjust the number of rows and widths of each row (prior to execution) such that the total area available for placement remains the same as in the 2-D case and the aspect ratio for each device layer is the same as in the 2-D case.

We then proceed by recursively partitioning the block roughly into halves, assigning nodes to each partition such that the capacity of each partition is not exceeded and such that the number of hyperedges spanning both partitions is minimized. Each partitioning step is permitted a tolerance varying from 2% to 10% depending on the discreteness of the partition: partitioning into wafers or parallel to rows, for example, must be done very precisely since the resulting partition sizes must be integral numbers of rows or wafers, but when partitioning perpendicular to rows, a higher tolerance will yield a better partitioning.

We note that min-cut partitioning along the 3rd dimension is equivalent to minimizing the number of inter-layer vias. Thus, in cases where such vias are costly (due to capacitance, pitch, or fabrication cost), we may trade off increased total wire length for fewer inter-plane vias by varying the point at which the design is partitioned into planes. For example, we may choose to partition into planes first (as shown in Fig. 2), or we may leave plane assignment until the detailed placement stage (Fig. 3). We find that the optimal wire length is obtained by using aspect ratio to determine the cut sequence – that is, a given partition is bisected perpendicular to the longest dimension of the partition. (For purposes of comparison, the length of the third dimension is scaled by the cost of inter-layer vias.) The user specifies at run-time whether to minimize total wire length or number of inter-layer vias, as well as the cost of these vias.

In partitioning a given block, we account for the presence of external nets using a terminal propagation scheme based on [8]. We extend this scheme to 3-D by expanding the "dummy terminals" used for propagation to include nodes that have been locked to planes above or below the partitioning point. At very detailed levels, we use branch-and-bound partitioning and placement [13]. Finally, wire lengths are determined using the half-perimeter metric, which in the 3-D case is the sum of the length, width, and height of the bounding box containing all terminals of a given net.

B. 3-D Global Routing

Global-routing algorithms generally may be categorized as sequential approaches (such as maze routing) or concurrent approaches [14]. We have chosen to implement a concurrent (hierarchical) global router for 3-D integration. Our approach is based on that of [15]. The use of hierarchy allows the router to avoid computational complexity.

Since modern technologies offer many levels of metal interconnect, we adopt an over-the-cell routing strategy. We assume that inter-cell wires may be routed without restriction on the upper levels of metal, whereas the lower levels are reserved for intra-cell wiring as well as power, ground, and other critical wires such as the clock tree. This uniformity in the routing substrate permits us to investigate hierarchical approaches based on concurrent methods.

In a 2-D hierarchical global router, the routing substrate (consisting of the wiring surface above the placed cells) is recursively bisected into routing subregions. Wires within a region may be either fully contained by the region or may terminate at a *pin* on one or more sides of the region. At each partitioning step, the existing pins on the sides of the routing region must be allocated to one of the two subregions. Then, those wires that are fully contained within the region must be allocated to one or both subregions. The remaining wires connect cells on both sides of the partition line; these are cut by the partition, and for each, a pin is inserted into the side between subregions. When complete, the resulting regions may be fed to a detailed router as formulations of channel or switchbox routing problems. A sample routing for a single net is shown in Fig. 4.

Our 3-D global router considers a routing region to be a set of aligned, congruent 2-D routing regions on one or more adjacent wafers. Wires may enter or exit the region through any of the sides of the 2-D regions as well as through the top and bottom of the set. The 3-D router must therefore determine the location and quantity of inter-wafer vias in addition to routing the wires on each wafer. In the 2-D case, it was assumed that cells would not interfere with the routing area; with inter-wafer vias this cannot be the case. However, we can assume that the terminals of cells are at the cell boundary or can be routed there. Thus, as long as space between the cell rows is provided for vias to pass through the wafer, wiring between adjacent wafers can be done by placing a via directly beneath the terminal on the upper wafer to which the wire is connected.
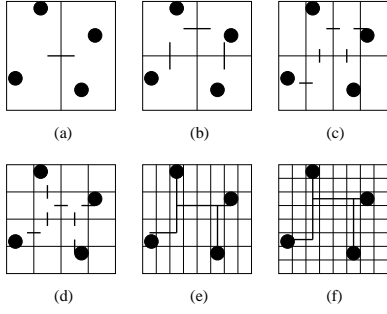
Fig. 4. Single-net example of the hierarchical routing procedure. Routing proceeds from stage (a) to (f) by recursive partitioning.
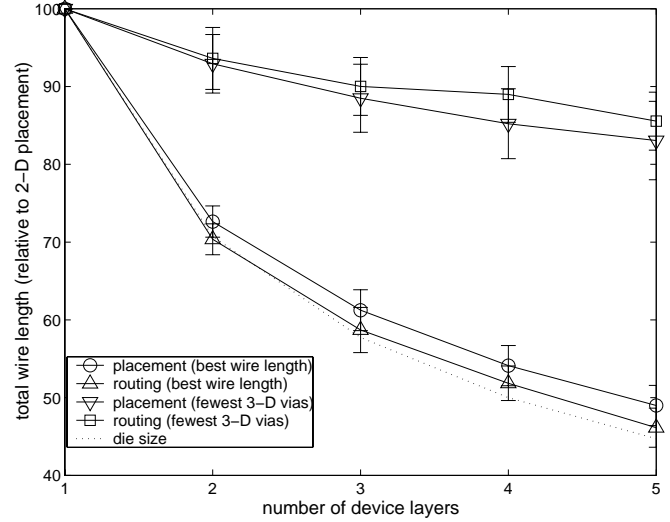

Fig. 5. Total wire length (as a function of number of device layers) obtained from placement and routing. The two pairs of data sets correspond to whether the tool is used to minimize total wire length or to minimize the number of inter-wafer (3-D) vias.

What remains is the two-dimensional problem of routing terminals on the lower wafer to that via. If, however, wires must span non-adjacent wafers without connecting to the wafers in between, a 3-D feedthrough must be placed on the internal wafers. Such vias present an obstacle for detailed routing. We therefore limit the total capacity for inter-wafer vias to number a single row's worth of vias per row of cells on a wafer (e.g. if a given wafer has ten rows of cells, and 50 vias can fit side-by-side within the width of a row, then the wafer has a inter-wafer via capacity of 500). With this capacity computed, 3-D global routing proceeds by hierarchical partitioning into wafers as well as along the horizontal and vertical dimensions of each wafer.

Once completed, the results of global routing are computed as the sum over all routing regions of the half-perimeter wire lengths of the wires contained within each region. This measurement should more closely reflect the final aggregate wire length.

III. Performance of the 3-D Standard Cell Place and Route Tools

Our standard-cell tool is implemented in about 17,000 lines of C code. For partitioning during placement, the user may select between hMetis [9] and PaToH [10]. Placement instances may be formatted in either GSRC [16] or Cadence LEF/DEF formats.

The 3-D standard-cell tool was tested on the ISPD '98 circuit benchmark suite, documented in [17]. This suite consists of 18 standard-cell designs from IBM with cell counts ranging from tens of thousands to hundreds of thousands of cells, and is available in the GSRC bookshelf format. We test the placer and router in two configurations: in the first we optimize for total wire length, whereas in the second we optimize for least number of inter-layer vias.

The performance of our place and route tool is shown in Fig. 5. Wire lengths listed are half-perimeter wire lengths (for the case of placement) and routed wire lengths, measured in microns (an inter-wafer is assumed here to be equivalent to one micron of wire; this value can be adjusted to match a given 3-

D technology) and averaged over the 18 circuits in the benchmark. For 2-D instances, the placer is competitive with the contemporary standard-cell placers shown in [18]. With 3-D integration, we note that a 28% reduction in total wire length is achievable with just two wafers, and that 51% reduction is possible with five wafers, assuming an optimal integration technology. The tools show us that if inter-layer via minimization is required, we may obtain 7% reduction in total wire length with two wafers and 17% reduction using five wafers.

IV. 3-D Circuit Layout Management

We have extended the Berkeley layout editor MAGIC [5] to facilitate the design of 3-D circuits. Called 3-D MAGIC, this tool allows designers to bond individual 2-D designs by issuing a ":bond" command to the interface. Once bonded, 3-D MAGIC treats the designs as a single entity. Following the methodology in [19], vias are added to the technology file and defined to be inter-layer vias. Our tool 3-D MAGIC automatically displays these vias in the relevant device-layer views of a 3-D stack. Electrical connectivity and wire selection is maintained across layers, allowing designers to examine and extract connectivity the same way as with 2-D nets. Finally, cell management has been extended to cover 3-D cells, so that 3-D circuit designs may be instantiated as subcells of larger designs.

A sample screen shot of 3-D MAGIC, where a wire that spans two layers has been highlighted, is shown in Fig. 6. In Fig. 7, a two-layer placement of a small benchmark circuit is shown.
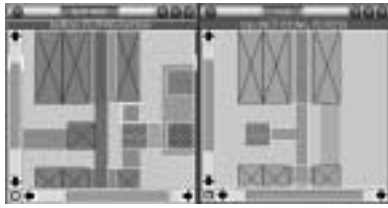
Fig. 6. Screen shot of 3-D MAGIC, with inter-layer interconnect highlighted.

Fig. 7. Screen shot of 3-D MAGIC with two-layer placement of a small benchmark circuit.

V. Conclusion

3-D integration as a technology has potential. As it matures, designers inevitably will want to know what benefits this technology offers and what means are available to use it. We have presented a set of tools that allows designers to exploit this emerging process. At the layout level, our 3-D MAGIC editor facilitates low-level design of multi-wafer integrated circuits as well as management of hierarchical 3-D designs. We have also developed a standard-cell place-and-route tool for 3-D integration. This tool can target netlists for a user-specifiable number of wafers with user-configurable 3-D technology parameters in terms of density and cost of inter-wafer vias.

Using this tool, we have obtained placement and global routing results on a variety of benchmarks, indicating that significant benefits may be obtained by targeting designs for 3-D integration. We have seen improvements of 28% to 51% in total wire length when targeting two to five wafers respectively. Clearly, there is some benefit to 3-D integration technology, and the tools we have presented give designers the capability to utilize these benefits in their designs.

VI. Acknowledgments

References

[1] C. W. Eichelberger. Three-dimensional multichip module system. United States Patent 5,111,278, May 1992.

[2] V. Subramanian, P. Dankoski, L. Degertekin, B. T. Khuri-Yakub, and K. C. Saraswat. Controlled two-step solid-phase crystallization for high-performance polysilicon TFTs. *IEEE Electron Device Letters*, 18:378–381, Aug. 1997.

[3] A. Fan, A. Rahman, and R. Reif. Copper wafer bonding. *Electrochemical and Solid-State Letters*, 2:534–536, 1999.

[4] A. Rahman and R. Reif. System-level performance evaluation of three-dimensional integrated circuits. *IEEE Trans. on VLSI Systems, Special Issue on System-Level Interconnect Prediction*, 8(6):671–678, 2000.

[5] J. Ousterhout et al. Magic: A VLSI layout system. In *Proceedings of the 21st Design Automation Conference*, pages 152–159, 1984.

[6] J. Kleinhans, G. Sigl, F. Johannes, and K. Antreich. GORDIAN: VLSI placement by quadratic programming and slicing optimization. *IEEE Transactions on Computer-Aided Design*, 10(3):356–365, 1991.

[7] C. Sechen. *VLSI Placement and Global Routing Using Simulated Annealing*. Kluwer Academic Publishers, Boston, 1988.

[8] A. E. Dunlop and B. W. Kernighan. A procedure for placement of standard cell VLSI circuits. In *IEEE Transactions on Computer-Aided Design*, pages 92–98, 1985.

[9] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar. Multilevel hypergraph partitioning: Applications in VLSI design. In *Proceedings of the 34th Design Automation Conference*, pages 526–529, 1997.

[10] U. V. Catalyurek and C. Aykanat. Hypergraph-partitioning-based decomposition for parallel sparse-matrix vector multiplication. *IEEE Transactions on Parallel and Distributed Systems*, 10(7):673–693, 1999.

[11] A. E. Caldwell, A. B. Kahng, and I. L. Markov. Can recursive bisection alone produce routable placements? In *Proceedings of the 37th Design Automation Conference*, pages 477–482, 2000.

[12] M. Wang, X. Yang, and M. Sarrafzadeh. Dragon2000: Fast standard-cell placement for large circuits. In *IEEE International Conference on Computer-Aided Design*, pages 260–263, 2000.

[13] A. B. Kahng, A. E. Caldwell, and I. L. Markov. Optimal partitioners and end-case placers for standard-cell layout. In *ACM International Symposium on Physical Design*, pages 90–96, April 1999.

[14] N. A. Sherwani. *Algorithms for VLSI Physical Design Automation*. Kluwer Academic Publishers, Boston, 1993.

[15] M. Burstein and R. Pelavin. Hierarchical wire routing. *IEEE Transactions on Computer-Aided Design*, CAD-2(4):223–234, 1983.

[16] http://vlsicad.cs.ucla.edu/ GSRC/ bookshelf/ Slots/ Placement/ plFormats.html.

[17] C. J. Alpert. The ISPD98 circuit benchmark suite. In *ACM International Symposium on Physical Design*, pages 80–85, April 1998.

[18] P. H. Madden. Reporting of standard cell placement results. In *ACM International Symposium on Physical Design*, pages 30–35, April 2001.

[19] S. M. Alam, D. E. Troxel, and C. V. Thompson. A comprehensive layout methodology and layout-specific circuit analyses for three-dimensional integrated circuits. In *Proc. ISQED*, March 2002.