

3D-Integration und 3D-Datenstrukturen – Eine Übersicht

Robert Fischbach, Jens Lienig
Technische Universität Dresden
Fakultät für Elektrotechnik und Informationstechnik
Dresden, Deutschland
Email: fischbach@ifte.de, lienig@ifte.de

Gefördert durch die DFG (Projekt 1401/1)

Kurzfassung

Neuartige Technologien wie die 3D-Integration schaffen die Voraussetzung, auch in der heutigen Nano-Ära die Erwartungen des Moore'schen Gesetzes zu erfüllen. Das Hinzufügen der dritten Dimension in modernen 2D-Schaltkreisen ermöglicht höhere Integrationsdichten, kürzere Verbindungsleitungen und den Aufbau heterogener Systeme in einer Baugruppe. Die Vorteile der 3D-Integration lassen sich jedoch nur nutzen, wenn Layoutdesigner und Werkzeugentwickler den raschen Fortschritt auf diesem Gebiet bewusst erkennen. Dieser Beitrag gibt zunächst einen Überblick über aktuelle 3D-Integrationstechnologien sowie zu 3D-Baugruppen und 3D-integrierten Schaltkreisen. Danach werden 3D-Datenstrukturen untersucht und verglichen, um Schlussfolgerungen über deren zukünftige Anwendbarkeit zu ziehen.

1 Einleitung

Wie durch das Moore'sche Gesetz vorhergesagt, erlauben moderne Technologien die Herstellung hochintegrierter Schaltkreise mit hunderten Millionen Transistoren. Dieser erstaunliche Fortschritt wurde bisher hauptsächlich durch die Verkleinerung der Halbleiterstrukturen erreicht. Zukünftig wird ein solches Verkleinern der lithografischen Strukturen zunehmend teurer. Eine andere Möglichkeit, die Vorgaben des Moore'schen Gesetzes zu erfüllen, ist das Einführen neuer Technologien, beispielsweise Verbindungen aus Kupfer, Silicon on Insulator, gestrecktes Silizium (strained silicon) oder die Nutzung der 3D-Integration. Der Einsatz von 3D-Integrationstechnologien ist dabei ein vielversprechender Ansatz. Hierbei werden die aktiven Bauelemente in mehreren Ebenen übereinander angeordnet. Obwohl 3D-Integration bereits seit mehreren Jahrzehnten diskutiert wird, erreicht sie erst jetzt praktische Relevanz für die Halbleiterindustrie.

Energieverbrauch und Leistungsfähigkeit eines Chips werden zunehmend durch seine internen Verbindungsleitungen (Interconnect) beeinflusst (Abb. 1). Globale Verbindungen skalieren dabei schlecht und verursachen eine wachsende Anzahl an Repeaterschaltungen. 3D-Integration lässt wesentlich kürzere Verbindungslängen zu und ermöglicht damit eine Leistungssteigerung und die Verringerung des Strombedarfs. Höhere Datenbandbreiten sind realisierbar und erlauben den Entwurf von hocheffizienten Cache-Strukturen. Kleinere Grundflächen verbessern die Ausbeute bei der Herstellung und erhöhen den Nutzen für mobile Anwendungen mit strengen Masseanforderungen. Damit bietet die 3D-Integration die Möglichkeiten, Gesamtverbindungslänge, Signalverzögerungen, Pufferanzahl und Stromverbrauch zu reduzieren. Außerdem ermöglicht die 3D-Integration den Aufbau von System in Package (SiP) und weiteren Entwurfsmethodiken auf Systemlevel, bei denen heterogene Technologien unabhängig in verschiedenen Systembestandteilen verwendet werden. So lassen sich unterschiedliche Schaltkreisarten,

z.B. Logik-, Speicher- und Analogschaltungen, in einer einzigen Baugruppe integrieren. Des Weiteren können völlig neuartige Bauelemente, wie etwa vertikal integrierte Transistoren, den Aufbau integrierter Schaltkreise grundlegend ändern.

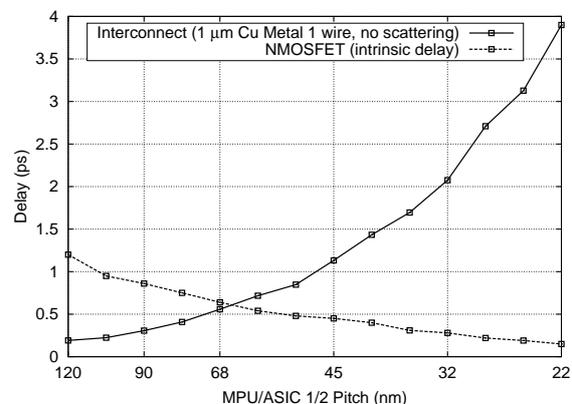


Abbildung 1: Wachsender Einfluss der Verbindungen (Interconnect) auf die Signalverzögerung für unterschiedliche Technologie-Generationen.

Wenn es so viele Vorteile gibt, erhebt sich die Frage, warum die 3D-Integration bisher noch keine breite Anwendung gefunden hat. Ein Grund dafür ist, dass das Skalieren der Halbleiterstrukturen bisher einfacher zu erreichen war, als in eine völlig neue Entwurfsmethodik zu investieren. Weiterhin bringt 3D-Integration einige große Herausforderungen mit sich, sowohl technologisch als auch beim Layoutentwurf. Typische Probleme der 3D-Integration sind unter anderem Zuverlässigkeit, Ausrichtungsgenauigkeit, Prüffreundlichkeit und thermische Gesichtspunkte. Die Komplexität des Layoutentwurfs von 3D-Strukturen übersteigt bei Weitem die des 2D-Falls, da u.a. die dritte Dimension den Lösungsraum erheblich vergrößert. Neben dieser höheren Komplexität müssen zusätzliche Randbedingungen,

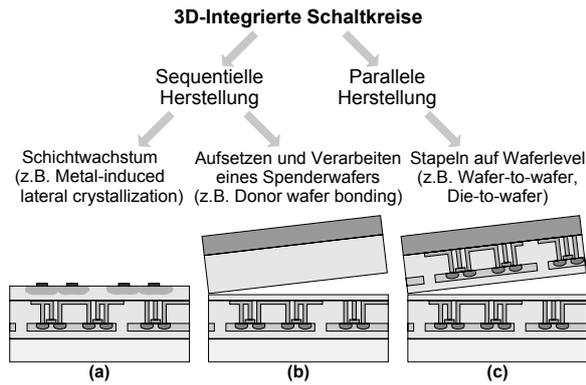


Abbildung 4: Klassifikation von 3D-ICs in Bezug auf ihre Technologie. Während Schichtwachstum (a) und das Aufsetzen eines unprozessierten Spenderwafers (b) die sequentielle Fertigung der einzelnen Schaltungsebenen bedingt, erlaubt das Stapeln auf Waferlevel (c) eine parallele Herstellung dieser Ebenen.

reicht eine hohe Integration durch das Stapeln verschiedener verpackter Schaltkreise (Abb. 3 d). Eine der am weitesten entwickelten 3D-Baugruppen ist das System on Package (SoP, Abb. 3 e) [4]. Es vereint die Vorteile unterschiedlicher Baugruppenttechnologien auf einer erweiterten multifunktionalen Leiterplatte. Viele verschiedenartige Komponenten können in solch einem System miteinander kombiniert werden, z.B. digitale und analoge Bestandteile, eingebettete passive Bauelemente, optische Module, Filter und Antennen. Die Evolution des SoP ist in Abb. 3 dargestellt. In diesem Beispiel besteht das SoP aus einem Multi-Chip-Modul (MCM), einem System in Package (SiP) und einem Package on Package (PoP). Aufgrund ihrer Heterogenität werden die einzelnen Module der 3D-Baugruppen oft unabhängig voneinander entworfen und gefertigt (Bottom-Up), bevor man sie in einer Baugruppe vereinigt. Dieser letzte Entwurfsschritt umfasst die Platzierung der Module, deren Verdrahtung untereinander und die Optimierung des Designs. Der Layoutentwurf eines System on Package ist in [5] beschrieben.

2.2 3D-integrierte Schaltkreise

3D-integrierte Schaltkreise bestehen aus mehreren Bauelemente-Ebenen, welche direkt auf Waferlevel integriert werden. Die typischen Technologien sind Schichtwachstum, Aufsetzen eines Spenderwafers und das Stapeln fertigprozessierter Wafer/Dies (Abb. 4). Im Vergleich zu den 3D-Baugruppen lassen sich mit 3D-integrierten Schaltkreisen wesentlich höhere Integrationsdichten erreichen. Schlüsseltechnologien sind Through Silicon Vias, Dünnen der Wafer, hochgenaues Ausrichten und geeignete Waferbonding-Verfahren. Ein Beispiel für das Schichtwachstum ist die Metall-induzierte laterale Kristallisation (engl. metal-induced lateral crystallization, Abb. 4 a). Nach der Abscheidung von amorphem Silizium leitet hierbei ein partiell aufgebracht Metall einen Rekristallisationsprozess bei hohen Temperaturen ein ($> 400\text{ }^{\circ}\text{C}$). Während diese Technologie große Viadichten ermöglicht, haben die hohen Temperaturen einen negativen Einfluss auf darunterliegende, bereits hergestellte Ebenen.

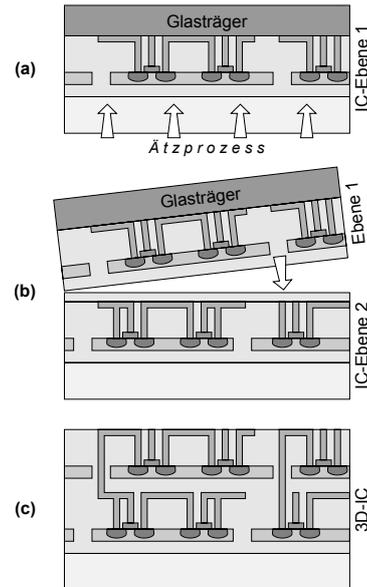


Abbildung 5: Verarbeitung eines Silicon on Insulator (SoI) Wafers zu einem 3D-IC: (a) IC-Ebene 1 wird an einen Glasträger angeheftet, rückseitiger Ätzprozess, vergrabene Oxidschicht als Ätzstopp, (b) Ausrichten und Bonden von IC-Ebene 1 auf Ebene 2, (c) Entfernen des Glasträgers und Herstellung der Through Silicon Vias für die Verbindung der IC-Ebenen.

Das Aufsetzen eines Spenderwafers basiert ebenfalls auf der sequentiellen Herstellung mehrerer Schaltungsebenen (Abb. 4 b). Dieses Verfahren vermeidet aber die temperaturbezogenen Nachteile für die darunterliegenden Ebenen, da kein Rekristallisationsprozess notwendig ist.

Beim Stapeln auf Waferlevel werden mehrere vorgefertigte Wafer (Schaltungsebenen) entweder Vorderseite an Vorderseite (engl. face-to-face, F2F) oder Vorderseite an Rückseite (engl. face-to-back, F2B) zusammengebracht (Abb. 4 c). Obwohl man den Prozess als Stapeln auf Waferlevel bezeichnet, können auch sogenannte die-to-wafer (D2W)-Methoden verwendet werden, um den Einfluss von Defekten einzelner Wafer zu verringern. Durch das Stapeln einzelner vorher getesteter Dies (anstatt ganzer Wafer) auf den Wafer ergibt sich eine höhere Fertigungsausbeute.

Der IBM 3D-IC-Herstellungsprozess (beschrieben in [6, 7]) ist ein vielversprechendes Beispiel für die 3D-Integration auf Waferlevel. Diese Methode ist in Abb. 5 dargestellt. Um möglichst kurze Verbindungen zwischen den Schaltungsebenen zu erzielen, sind diese dünn zu gestalten. Deshalb werden Silicon on Insulator (SoI) Wafer mit einer vergrabenen Oxidschicht (BOX) verwendet. Das Verbinden einer Schaltungsebene mit einer anderen wird entweder durch Verschmelzen von Oxidschichten, eutektisches Bonding oder die Verwendung von Klebstoffen erreicht. Die oberste Schaltungsebene ist meist durch einen Glasträger gestützt. Der hohe Grad wechselseitiger Beziehungen der einzelnen Module eines 3D-Schaltkreises verlangt nach einem hierarchischen Top-Down-Entwurf mit einem signifikanten Anteil an Abhängigkeiten zwischen den einzelnen Entwurfsschritten. Einen Einblick in den Layoutentwurf von 3D-ICs geben [8–10].

Tabelle 1: Vergleich von 3D-Integrationstechnologien.

Merkmale	3D-Baugruppen SiP/SoP/PoP	3D-Integrierte Schaltkreise		
		Schichtwachstum	Spenderwafer	Waferstapeln
Via-Dichte	niedrig	sehr hoch	sehr hoch	hoch
Herstellung	parallel	sequentiell	sequentiell	parallel
Abstand der Ebenen	hoch	sehr gering	sehr gering	gering
Heterogene Strukturen	ja	schwierig	möglich	ja
Thermische Abh. der Ebenen	niedrig	sehr hoch	hoch	mittel
Testbarkeit	Chiplevel	schwierig	schwierig	möglich

2.3 Herausforderungen und Vergleich von 3D-Technologien

3D-Integration ist die logische Konsequenz eines immer größer werdenden Engpasses der 2D-Integration, resultierend aus Interconnect-bedingten Problemen, welche weiteren Verbesserungen in Leistung und Funktionalität entgegenstehen. Allerdings bringt die 3D-Integration große Herausforderungen mit sich. Es sind zum Teil technologische Probleme, wie das hochgenaue Ausrichten oder Zuverlässigkeitsfragen, aber auch Entwurfsthematiken, z.B. das Berücksichtigen neuer thermischer Randbedingungen.

Einen Vergleich der 3D-Integrationstechnologien in Bezug auf deren Haupteigenschaften enthält Tabelle 1.

3 3D-Datenstrukturen

3.1 Übersicht

In der Entwurfsautomatisierung, speziell beim Layoutentwurf, werden Datenstrukturen verwendet, um Informationen über verschiedene Layoutelemente und deren Eigenschaften zu speichern. Solche Datenstrukturen sind ein abstraktes Modell des entsprechenden Entwurfsproblems.

Effiziente Datenstrukturen integrieren zusätzlich nützliche Eigenschaften über die Layoutelemente, etwa den Direktzugriff auf Nachbarn. Sie spannen einen Lösungsraum, welcher dann mit einem geeigneten Optimierungsverfahren, wie Simulated Annealing, durchsucht werden kann. Dieser Raum sollte eine geringe Redundanz besitzen, so klein wie möglich sein und trotzdem die besten Lösungen beinhalten. Eine effiziente Implementation einer Datenstruktur muss schnelle Operationen, z.B. Rotation und Vertauschen sowie die Transformation vom abstrakten Abbild in die reale Geometrie der Blöcke realisieren.

Die ersten Werkzeuge für den 3D-Layoutentwurf benutzten klassische, bewährte Datenstrukturen, z.B. einen Schnittbaum pro aktive Ebene. Diese sogenannten 2.5D-Ansätze haben den Nachteil, dass enge Beziehungen zwischen Modulen auf verschiedenen Ebenen vernachlässigt werden. Damit ist zum Beispiel ein erfolgreicher thermischer Entwurf nur schwer möglich. Es ist offensichtlich, dass moderne 3D-Datenstrukturen 3D-Layouts vollständig unterstützen sollten. Wie in Abb. 6 dargestellt, sind die Mehrzahl der derzeit entwickelten 3D-Datenstrukturen Weiterentwicklungen ihrer bekannten 2D-Vorgänger. Ein Beispiel ist der in [11] vorgestellte T-Tree. Bei diesem konnten die effizienten Eigenschaften seiner 2D-Version (B*-Tree) beibehalten werden, indem eine Ternärbaumstruktur (anstelle eines binären Baumes) verwendet wurde, welche die Abbildung der dritten Dimension erlaubt. Aber nicht alle

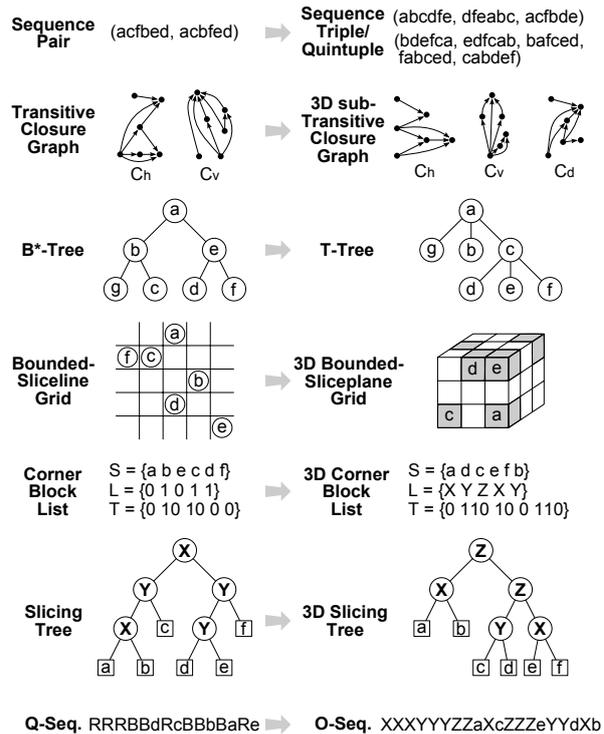


Abbildung 6: Klassische 2D-Floorplanning-Datenstrukturen (links) und ihre 3D-Weiterentwicklungen (rechts, siehe auch Tabelle 2).

3D-Erweiterungen der klassischen 2D-Datenstrukturen können ihre guten Eigenschaften beibehalten, die sie im 2D-Fall so effizient gestaltet haben. Ein Beispiel dafür ist das in [12] beschriebene Sequence Triple. Deshalb ist es notwendig, neue, speziell an die 3D-Integration angepasste Datenstrukturen zu finden, welche in der Lage sind, den veränderten Anforderungen gerecht zu werden. Nachfolgend wird der Fortschritt umrissen, welchen man bisher auf dem Gebiet des 3D-Floorplanning erreicht hat.

3.2 3D-Datenstrukturen für das Floorplanning

Während des Floorplanning werden Formen und Positionen der einzelnen Module (Schaltkreispartitionen) bestimmt. Bei vielen 3D-Technologien lassen sich bei Platzierung und Routing bisherige Datenstrukturen beibehalten, da diese Schritte innerhalb der einzelnen, hierbei oft getrennt betrachteten Module angewendet werden. Das Top-Level-Floorplanning hingegen muss die Beziehungen der Module berücksichtigen und be-

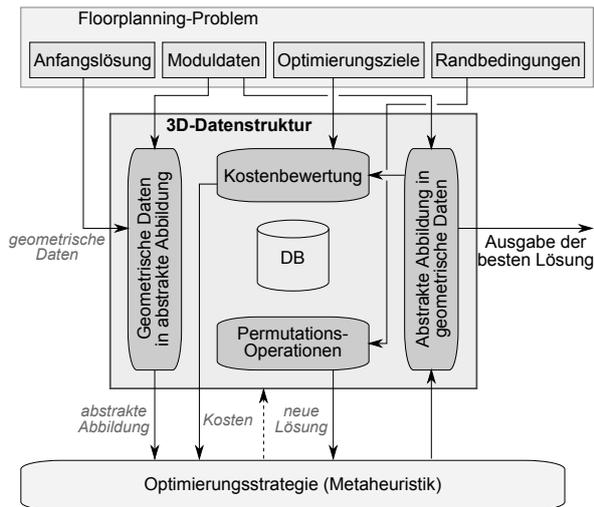


Abbildung 7: Bedeutung der 3D-Datenstruktur für das Floorplanning-Problem und die Optimierungsstrategie. Physische Informationen der Module, Randbedingungen und Optimierungsziele sind gegeben und werden von der Datenstruktur verwendet, um einen konkreten Floorplan bewerten sowie zwischen abstrakter und geometrischer Abbildung umwandeln zu können.

nötigt deshalb einen 3D-Ansatz. Aufgrund dessen sind Floorplanning-Algorithmen die ersten, welche sich den neuen Herausforderungen der 3D-Integration stellen müssen.

Herkömmliches Floorplanning geht von einer (einzelnen) zweidimensionalen Ebene aus, auf der mehrere Module anzuordnen sind. Eine Vielzahl verschiedenster algorithmischer Ansätze lassen sich einsetzen, um dieses Floorplanning-Problem erfolgreich zu bearbeiten. Analytische Methoden lösen eine Menge von mathematischen Gleichungen, in denen Randbedingungen und Optimierungsziele beschrieben sind. Heuristiken, wie das Floorplan Sizing Verfahren oder das Cluster Wachstum, sind angepasste problemspezifische Algorithmen. Im Gegensatz dazu sind Metaheuristiken problemunspezifisch und ermitteln Näherungslösungen für allgemeine kombinatorische Optimierungsprobleme. Eine solche Metaheuristik, z.B. Simulated Annealing, lässt sich auf das Floorplanning-Problem anwenden, wenn eine passende effiziente problemspezifische Abstraktion gefunden wurde. Die meisten Datenstrukturen für das Floorplanning basieren auf solchen Abstraktionen und ermöglichen deshalb die Verwendung von Metaheuristiken für die Optimierung des Floorplans.

Datenstrukturen bilden einen geometrischen Zusammenhang (physische Repräsentation) effizient durch ein abstraktes Modell ab, auf welchem dann verschiedene Operationen angewandt werden, um neue Lösungen zu finden. Die Lösungsbewertung wird unter Ausnutzung besonderer Eigenschaften einer Datenstruktur implementiert. Eine 3D-Datenstruktur kann man als Abstraktionsebene zwischen einem problemunspezifischen Optimierungsverfahren (Metaheuristik) und dem eigentlichen 3D-Floorplanning-Problem ansehen (Abb. 7).

Das 3D-Floorplanning-Problem weist neue, 3D-spezifische Merkmale auf, die mit in die verwendeten Datenstrukturen einbezogen werden. Zum Beispiel sind die vertikalen Abhängigkeiten der Module

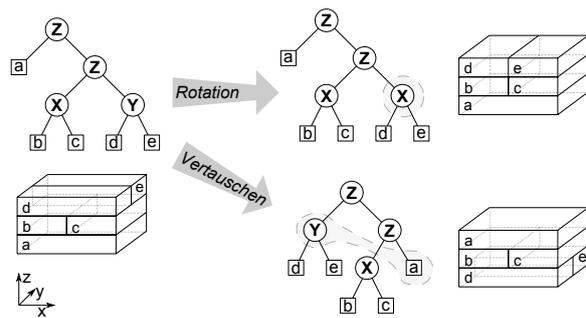


Abbildung 8: Illustration der Operationen eines 3D-Slicing Tree, um einen gegebenen 3D-Floorplan zu modifizieren. Eine Rotation verändert einen inneren Knoten (dieser stellt einen Schnitt durch eine Normalebene dar) und führt damit zur geometrischen Rotation der Module, welche im Unterbaum dieses Knotens enthalten sind. Das Vertauschen zweier Teilbäume führt zum geometrischen Austausch der Module dieser Teilbäume.

untereinander zusätzlich zu den horizontalen zu berücksichtigen.

Zwei Ansätze für vertikale Abhängigkeiten wurden bisher verfolgt. Der erste ist die mehrfache, ebenenweise Verwendung klassischer (2D-)Datenstrukturen, also der sogenannten 2.5D-Ansatz. Hierbei ist es jedoch notwendig, zusätzliche Mechanismen zu implementieren, um die vertikalen Abhängigkeiten (z.B. Überlappungsregeln) der Module unterschiedlicher Ebenen zu berücksichtigen. Solche Repräsentationen führen zu diskreten z-Koordinaten, z.B. beim Combined Bucket and 2D-Array (CBA)-Ansatz in [13].

Es wurde aber schnell offensichtlich, dass die vertikalen Abhängigkeiten zwischen den Modulen besser direkt in den Datenstrukturen eingebaut werden sollten. Aktuellere 3D-Datenstrukturen bilden deshalb Module über mehrere Ebenen direkt in allen drei Dimensionen ab. Ein Beispiel für eine solche 3D-Datenstruktur ist der 3D-Schnittbaum, welcher in [14] beschrieben ist. Wie in Abb. 8 dargestellt, werden verschiedene Operationen (z.B. Rotation oder Vertauschen) verwendet, um eine gegebene Lösung zu modifizieren. Eine wiederholte Ausführung solcher Operationen erlaubt es, jeden beliebigen Schnittbaum aus jedem gegebenen Schnittbaum zu erhalten. Jedoch sind die Lösungen, die man mit Hilfe von 3D-Schnittbäumen erhält, auf geschnittene Floorplans beschränkt.

Tabelle 2 fasst die bekanntesten Vertreter der 3D-Datenstrukturen zusammen, welche auf moderne Floorplanning-Probleme angewandt wurden [11, 12, 14–19]. Wie zu erkennen ist, lassen sich die Datenstrukturen in drei Kategorien einteilen: Sequenzen und/oder Listen, Gitterstrukturen sowie auf Graphen basierende Strukturen. Diese Datenstrukturen mit ihren jeweiligen Vorteilen, Grenzen und Möglichkeiten werden im Vortrag vorgestellt, ebenso ihre mögliche Verwendung in anderen 3D-Entwurfsschritten, wie etwa bei der 3D-Platzierung und -Verdrahtung.

4 Zusammenfassung und Ausblick

Zur Erfüllung des Moore'schen Gesetzes auch im heutigen Zeitalter von Nanostrukturen gewinnt die 3D-Integration zunehmend an Bedeutung. Eine Vielzahl von 3D-Datenstrukturen wurde in letzter Zeit entwi-

Tabelle 2: Typische 3D-Datenstrukturen bei 3D-Floorplanning-Problemen mit Angabe von: Publikationsdatum, Worst-Case-Komplexität der Operationen (Rechenkomplexität), Größe des Lösungsraumes und Hauptcharakteristika (n = Anzahl der Module, $n.g.$ = nicht gegeben). Bis auf die letzte Datenstruktur haben alle anderen einen direkten 2D-Vorgänger (siehe auch Abb. 6).

Datenstruktur	Jahr	Komplexität	Lösungsraum	Charakteristika
Sequence Triple/Quintuple	2000 [12]	$n.g.$	$O((n!)^3)/O((n!)^5)$	Sequenz: Drei bzw. fünf Sequenzen (Locii)
3D sub-Transitive Closure Graph	2004 [15]	$O(n^2)$	$O((n!)^3)$	Graph: Drei transitive Graphen
T-Tree	2004 [11]	$O(n)$	$O(n! 3^{3n}/2^{2n} n^{1.5})$	Graph: Ternärbaum; Knoten: Module; Äste: Nachbarschaft
3D Bounded-Sliceplane Grid	2005 [16]	$O(n^2)$	$n.g.$	Gitter: Dreidimensionale Gitterstruktur
3D Corner Block List	2005 [17]	$O(n)$	$O(n! 3^{n-1} 2^{4n-4})$	Sequenz/Liste: Modulsequenz, Ausrichtungsliste, T-Kreuzungsliste
3D Slicing Tree	2005 [14]	$n.g.$	$n.g.$	Graph: Binärbaum; innere Knoten: Schnitte; Blätter: Module
O-Sequence	2006 [18]	$O(n)$	$n.g.$	Sequenz: Modul- und Symbolsequenz
Labeled Tree and Dual Sequences	2008 [19]	$O(n^{4/3} \log n)$	$(n!)^2 (n+1)^{n-1}$	Sequenz/Graph: Modulsequenz, Zahlensequenz und Baum

ckelt, um 3D-Integrationstechnologien in einem effizienten und automatisierten Entwurf berücksichtigen zu können.

Um die Vorteile einer zusätzlichen Dimension in 3D-integrierten Schaltkreisen auszunutzen, müssen Layoutdesigner und Werkzeugentwickler die rasche Entwicklung im Bereich der 3D-Integration bewusst erkennen. Der Beitrag gibt eine Übersicht über neu entstehende 3D-Integrationstechnologien und bietet einen aktuellen Überblick über die effizientesten 3D-Datenstrukturen. Während 3D-Integrationstechnologien in zwei Gruppen eingeteilt werden können (3D-Baugruppen und 3D-integrierte Schaltkreise), lassen sich die zugehörigen Datenstrukturen in Kategorien unterteilen: Sequenzen und/oder Listen, Gitterstrukturen sowie auf Graphen basierende Strukturen. Derzeit finden alle drei Arten Anwendung. Außerdem ist ein Trend zu Hybridstrukturen zu erkennen. Aktuelle Fortschritte erlauben vielen Datenstrukturen das Erreichen (fast) linearer Rechenkomplexität.

Literatur

- [1] ITRS, "http://www.itrs.net/reports.html," tech. rep., ESIA, JEITA, KSIA, TSIA and SIA, 2007.
- [2] E. Beyne, "The rise of the 3rd dimension for system integration," in *Interconnect Technology Conference, 2006 International*, pp. 1–5, June 5–7, 2006.
- [3] W. R. Davis, J. Wilson, S. Mick, J. Xu, H. Hua, C. Mineo, A. M. Sule, M. Steer, and P. D. Franzon, "Demystifying 3D ICs: The pros and cons of going vertical," *Design & Test of Computers, IEEE*, vol. 22, pp. 498–510, Nov.–Dec. 2005.
- [4] R. Tummala, "SOP: What is it and why? A new microsystem-integration technology paradigm-moore's law for system integration of miniaturized convergent systems of the next decade," *Advanced Packaging, IEEE Transactions on*, vol. 27, pp. 241–249, May 2004.
- [5] S. Lim, "Physical design for 3D system on package," *Design & Test of Computers, IEEE*, vol. 22, pp. 532–539, Nov.–Dec. 2005.
- [6] K. W. Guarini, A. W. Topol, M. Jeong et. al, "Electrical integrity of state-of-the-art 0.13 μm SOI CMOS devices and circuits transferred for three-dimensional (3D) integrated circuit (IC) fabrication," in *Proc. Digest. International Electron Devices Meeting IEDM '02*, pp. 943–945, 2002.
- [7] A. W. Topol, J. D. C. La Tulipe, L. Shi, D. J. Frank, K. Bernstein, S. E. Steen, A. Kumar, G. U. Singco, A. M. Young, K. W. Guarini, and M. Jeong, "Three-dimensional integrated circuits," *IBM Journal of Research and Development*, vol. 50, pp. 491–506, July/Sept. 2006.
- [8] S. Das, *Design Automation and Analysis of Three-Dimensional Integrated Circuits*. PhD thesis, Massachusetts Institute of Technology, 2004.
- [9] G. H. Loh, Y. Xie, and B. Black, "Processor design in 3D die-stacking technologies," *Micro, IEEE*, vol. 27, pp. 31–48, May–June 2007.
- [10] Y. Xie, G. H. Loh, B. Black, and K. Bernstein, "Design space exploration for 3d architectures," *J. Emerg. Technol. Comput. Syst.*, vol. 2, no. 2, pp. 65–103, 2006.
- [11] P.-H. Yuh, C.-L. Yang, and Y.-W. Chang, "Temporal floorplanning using the T-Tree formulation," in *Proc. ICCAD-2004 Computer Aided Design IEEE/ACM International Conference on*, pp. 300–305, 2004.
- [12] H. Yamazaki, K. Sakanushi, S. Nakatake, and Y. Kajitani, "The 3D-packing by meta data structure and packing heuristics," *Fundamentals of Electronics, Communications and Computer, IEICE Transactions on*, vol. E83-A, pp. 639–645, April 2000.
- [13] J. Cong, J. Wei, and Y. Zhang, "A thermal-driven floorplanning algorithm for 3d ics," in *Computer Aided Design, 2004. ICCAD-2004. IEEE/ACM International Conference on*, pp. 306–313, 7–11 Nov. 2004.
- [14] L. Cheng, L. Deng, and M. D. F. Wong, "Floorplanning for 3-D VLSI design," in *ASP-DAC '05: Proceedings of the 2005 Conference on Asia South Pacific Design Automation*, pp. 405–411, ACM, 2005.
- [15] P.-H. Yuh, C.-L. Yang, Y.-W. Chang, and H.-L. Chen, "Temporal floorplanning using 3D-subTCG," in *Proc. Asia and South Pacific Design Automation Conference the ASP-DAC 2004* (C.-L. Yang, ed.), pp. 725–730, 2004.
- [16] H. Ninomiya and H. Asai, "Three dimensional module packing by simulated annealing," in *Evolutionary Computation, IEEE Congress on*, vol. 2, pp. 1069–1074, 2–5 Sept. 2005.
- [17] Y. Ma, X. Hong, S. Dong, and C. Cheng, "3D CBL: An efficient algorithm for general 3D packing problems," in *Proc. 48th Midwest Symposium on Circuits and Systems*, vol. 2, pp. 1079–1082, 2005.
- [18] H. Ohta, T. Yamada, C. Kodama, and K. Fujiyosi, "The O-Sequence: Representation of 3D-floorplan dissected by rectangular walls," in *Proc. of Research in Microelectronics and Electronics 2006, Ph.D.*, pp. 317–320, 2006.
- [19] R. Wang, E. F. Young, Y. Zhu, F. C. Graham, R. Graham, and C.-K. Cheng, "3-D floorplanning using labeled tree and dual sequences," in *ISPD '08: Proceedings of the 2008 International Symposium on Physical Design*, pp. 54–59, ACM, 2008.