

# Multikriterielle Layoutoptimierung durch TSV- und Deadspace-Planung für den 3D-IC-Entwurf

Johann Knechtel, Matthias Thiele, Jens Lienig

johann.knechtel@ifte.de, matthias.thiele@ifte.de, jens.lienig@ifte.de

Institut für Feinwerktechnik und Elektronik-Design, Technische Universität Dresden

## KURZFASSUNG

Bei der Optimierung von dreidimensional integrierten Schaltkreisen (3D-ICs) sind sog. *Deadspace-Regionen* zwischen den Schaltungsblöcken im Floorplan von großer Wichtigkeit. Deadspace ist für die Platzierung von Through-Silicon Vias (TSVs) und andere Entwurfsaufgaben erforderlich. Aufgrund der aus Kostengründen bewusst niedrig gehaltenen Verfügbarkeit von Deadspace ist eine gezielte Layoutoptimierung zur effektiven Nutzung dieser Ressource notwendig. Während bisherige Arbeiten meistens auf einzelne Entwurfsprobleme fokussieren, stellt dieser Beitrag eine *multikriterielle* Methodik zur Layoutoptimierung vor. Insbesondere durch TSV-Planung und Deadspace-Optimierung wird eine simultane Verbesserung der Verdrahtungslängen, des statischen Spannungsabfalls, der Taktsignalbaumgröße und der Maximaltemperatur angestrebt. Dabei wird die Entwurfsqualität entsprechend den genannten Kriterien iterativ geprüft und die Methodik anhand dieser Rückkopplung gesteuert.

## I. EINLEITUNG

Um die Anforderungen an Funktionalität, Leistung und Kosten für den modernen Schaltkreisentwurf auszubalancieren zu können, ist die dreidimensionale Integration (3D-ICs) ein attraktiver Entwurfsansatz. Die *Integration auf Chip-Ebene* (Abbildung 1a) wird durch *Through-Silicon Vias (TSVs)* ermöglicht und verspricht kürzere und sparsamere Verbindungen, verglichen mit traditionellen drahtgebundenen Systemen [3]. Diese kompakte Integration ist nicht nur wichtig für den kosteneffektiven Entwurf moderner Schaltkreise [6], sondern begünstigt außerdem die *Block-Level-Integration*, eine Integrationsvariante, bei welcher größere vorhandene 2D-Schaltungsblöcke in individuelle aktive Lagen eingebunden sind [10]. Dazu ist der Bereich auf den aktiven Lagen, der nicht durch Blöcke beansprucht ist, der sog. *Deadspace*, sowohl für das Floorplanning als auch für die Platzierung von verschiedenartigen TSVs zu nutzen. Blöcke und der zwischen ihnen entstehende Deadspace werden typischerweise während des Floorplannings so platziert, dass die Verdrahtungslänge minimiert und die Wärmeabführung unterstützt wird [5], [13], [17]. Weitere Entwurfskriterien sind entweder während des Floorplanning oder durch nachträgliche Verschiebung von Blöcken umsetzbar. Letzteres nutzt man insbesondere für die Platzierung verschiedener Arten von TSVs und verwandter Optimierungsaufgaben. Zum Beispiel werden Power/Ground-TSVs (PG-TSVs) bevorzugt zueinander ausgerichtet, um Elektromigration, Spannungsabfall und überhöhte Verdrahtungsdichten zu verringern.

Bisherige Arbeiten fokussieren meist auf einzelne Kriterien während solch einer Optimierung und vernachlässigen dabei das globale Entwurfsergebnis [5], [7], [10]. Die hier vorgestellte Optimierungsmethodik strebt dagegen eine Verbesserung der Platzierung von Blöcken und TSVs unter Einbeziehung von TSV-Kosten sowie weiterer, gleichzeitig betrachteter Entwurfskriterien und Optimierungszielen an.

## II. EINE METHODIK ZUR MULTIKRITERIELLEN LAYOUTOPTIMIERUNG UND TSV-PLANUNG

Um die gewünschte multikriterielle Optimierung praxisnah umzusetzen, wird im Folgenden eine modulare Methodik vorgestellt, welche sich in existierende 3D-IC-Entwurfsflüsse integrieren lässt. Die entsprechende Erweiterung eines Entwurfsflusses ist in Abbildung 1b dargestellt. Die Methodik beruht im Wesentlichen auf einem inkrementellen Prozess, der einen optimierten Floorplan inklusive platzierter TSVs anstrebt. Wie üblich in modularen 3D-IC-Entwurfsflüssen, lässt sich die TSV-Planung vom Floorplanning und/oder der Platzierung getrennt behandeln. Daher ist der Hauptprozess, welcher TSV-Planung und Deadspace-Optimierung umfasst, bemüht, zuerst (i) geeignete Positionen für TSVs zu lokalisieren. Dies erfordert meist eine Umverteilung und/oder Neuausrichtung der Deadspace-Regionen. Anschließend ist (ii) ein TSV auf diese

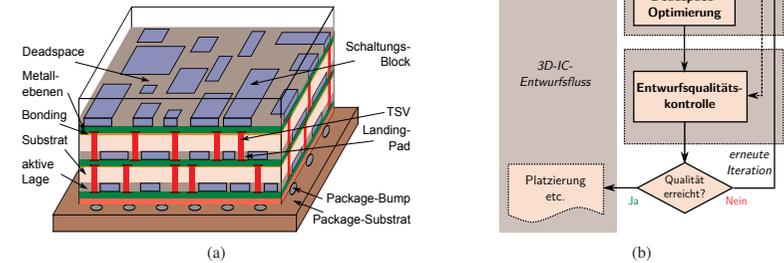


Abbildung 1. (a) Ein 3D-IC mit drei aktiven Lagen, hergestellt mittels Face-To-Back-Technologie. Zur besseren Darstellbarkeit sind die oberste Substratschicht und der Kühlkörper nicht dargestellt und die Chipvorderseiten sind geschnitten. TSVs dürfen Blöcke nicht durchdringen und sind deshalb dazwischen platziert. TSVs sind teilweise über benachbarte aktive Lagen hinweg ausgerichtet. (b) Prozessdiagramm der vorgestellten Methodik, mit schematischer Integration in einen 3D-IC-Entwurfsfluss.

Position (oder in der Nähe) zu platzieren und (iii) eine Deadspace-Optimierung auf Basis der veränderten TSV-Platzierung durchzuführen. Nach Konvergenz des Hauptprozesses ist die allgemeine Entwurfsqualität zu bewerten. Falls notwendig, kann danach erneut eine angepasste Iteration der Optimierung ausgelöst werden. Nachfolgend sind die Algorithmen und die Methodik im Detail dargestellt.

### A. Deadspace-Optimierung

Der Hauptprozess, bestehend aus Deadspace-Optimierung und TSV-Planung (Abschnitt II-B), bildet die Schlüsselkomponente der multikriteriellen Optimierungsmethodik. Dabei erfolgt die Steuerung bzw. Rückkopplung der Optimierung von Deadspace mittels TSV-Planung, um folgende Anliegen zu lösen:

- **Management der Deadspace-Auslastung:** Verschiedene Arten von TSVs "konkurrieren" um den vorhandenen Deadspace. Die Kontrolle der (lokalen) Auslastung wirkt sich auf die Platzierbarkeit von TSVs und damit direkt auf die Entwurfsqualität aus.
- **Eine Abstimmung bei der Anordnung von Deadspace** ist erforderlich, um über mehrere aktive Lagen hinweg ausgerichtete TSVs platzieren zu können.

Zur Verbesserung der Block- und TSV-Platzierung, welche gemeinsam die Entwurfsqualität maßgeblich beeinflussen, ist (i) die Neuordnung und Ausrichtung von gegebenem Deadspace sowie (ii) die Verschiebung von Blöcken und TSVs einzuführen. Das Einfügen von zusätzlichem Deadspace ist aufgrund der damit in Kauf zu nehmenden Erhöhung des Flächenverbrauchs und der Verdrahtungslänge nur begrenzt anwendbar [10].

Der zur Lösung dieser Probleme entwickelte Algorithmus zur Verschiebung beruht auf den Konzepten der *Constraint-Graphen (CG)* [9, Kapitel 3], der *Bereichs-Constraints* [15] und des *räumlichen Slacks* [1]. Sie erlauben zum Ersten die Repräsentation des Floorplans mittels eines CG-Paars. Dieses besteht aus zwei Graphen zur Beschreibung der horizontalen und vertikalen Blockanordnungen, der Einhaltung einer gültigen Platzierung sowie einer effizienten Handhabung der relativen Blockanordnungen. Zweitens dienen räumliche Slacks der Beschreibung der maximal erlaubten Verschiebungsdistanzen. Zum Dritten werden Bereichs-Constraints zur Begrenzung solcher Verschiebungen innerhalb bestimmter Bereiche genutzt, wie nachfolgend erläutert ist. Die Koordinaten  $(x, y)$  von Block- und TSV-Eckpunkten lassen sich dabei mittels

Bereichs-Constraints in *Verschiebungsfenster*  $[x - \delta, x + \delta], [y - \delta, y + \delta]$  umwandeln (Abbildung 2a). Diese dienen der Wahrung der initialen (optimierten) Blockanordnung in einem gewissen Rahmen bei gleichzeitiger Zulassung notwendiger Verschiebungen. Um die Platzierbarkeit von TSVs hinsichtlich der Möglichkeit für entsprechende lokale Verschiebungen von Blöcken und TSVs zu beurteilen, sind die räumlichen Slacks innerhalb der Graphen annotiert. Wie in Abbildung 2b dargestellt, betrachtet man während der Slack-Bestimmung die eingeführten Verschiebungsfenster als Grenzen.

Für den (inkrementellen) Hauptprozess ist zu Beginn das CG-Paar für jede aktive Lage einzeln zu bestimmen. Im Laufe der Optimierung, konkret nach der Platzierung von TSVs, sind die Graphen zu aktualisieren. Diese Aktualisierung ist trivial für die Fälle, in denen sich das geplante TSV direkt in Deadspace platzieren lässt. In den anderen Fällen, d.h. bei der Überlappung des geplanten TSVs mit Blöcken, sind die entsprechend verfügbaren Slacks während der Verschiebung zu berücksichtigen. Zur Begrenzung der durchzuführenden Verschiebungen sind TSVs dann außerhalb der nächstliegenden Block-Grenze zu platzieren und die relativen Blockanordnungen der Graphen entsprechend anzupassen.

Um den beschriebenen Verschiebungsprozess zu realisieren, wurden weiterhin simple und effektive Algorithmen zur Slack-Bestimmung sowie zur Generierung von Floorplans aus den CG-Paaren, und umgekehrt, implementiert [9, Kapitel 3]. Anzumerken ist, dass die Anwendung von Verschiebungsfenstern nicht kompaktierte Floorplans voraussetzt. Daher müssen für die Constraint-Graphen jeweils zusätzliche Kanten eingefügt werden, welche Blöcke und TSVs innerhalb von deren Fenstern einbinden.

## B. TSV-Planung

Um die Deadspace-Optimierung zu vereinfachen, sind die verschiedenen Arten von TSVs in folgender Reihenfolge zu planen: 1) PG-TSVs, 2) Taktsignal-TSVs, 3) Signal-TSVs und 4) thermische TSVs. Dies Priorisierung ist entsprechend der notwendigen Ausrichtung von TSVs, des Einflusses der TSV-Art auf die gesamte Entwurfsqualität, und der Flexibilität bei der Platzierung der TSV-Art gestaltet. Anzumerken ist, dass die Schritte der TSV-Planung und der anschließenden Deadspace-Optimierung die Reduzierung des initialen Wertes  $\Gamma_\gamma^0$  der jeweiligen Kosten  $\gamma$  (z.B. der maximale Spannungsabfall  $\Gamma_{\gamma_{IR}}^0$ ) auf  $\Gamma_\gamma^{opt} = w_\gamma^{opt} * \Gamma_\gamma^0$  anstreben. In den folgenden Unterabschnitten sind die Techniken zur TSV-Planung im Detail beschrieben.

1) *Planung von PG-TSVs*: Die gezielte, irreguläre Platzierung von ausgerichteten TSVs in Regionen der höchsten Leistungsabnahme stellt einen effektiven Ansatz zur Minimierung der TSVs selbst und des Spannungsabfalls dar [8]. Ausgehend von SPICE-Simulationen in klassischen PG-Versorgungsrastern wird in dieser Arbeit eine simple Diagnostik zur Bestimmung von belasteten Rasterknoten vorgestellt, d.h. zur gezielten Platzierung von PG-TSVs. Diese Diagnostik ist als *qualitative Spannungsabfall-Verteilung* bezeichnet und wie folgt anzuwenden. Als **Erstes** ist ein 2D-Raster entsprechend der Struktur des PG-Versorgungsrasters des 3D-ICs zu erstellen. **Zweitens** erfolgt die Aufsummierung des Leistungsverbrauches aller Blöcke

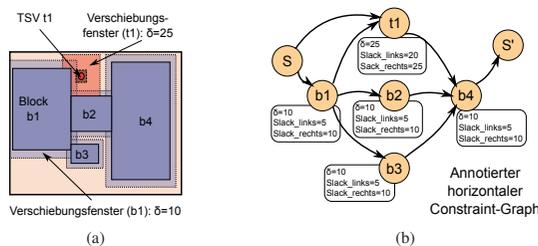


Abbildung 2. (a) Block- und TSV-Begrenzungen sind mittels Bereichs-Constraints um sog. Verschiebungsfenster erweitert (gepunktete Vierecke). (b) Der (a) entsprechende horizontale CG mit annotierten Slacks. Während der Slack-Bestimmung sind die Verschiebungsfenster als fixe Begrenzungen zu interpretieren.

welche sich jeweils im 3D-IC vertikal überlagern, sowie die Zuweisung der resultierenden Werte  $P(n)$  zu den zugehörigen Knoten im 2D-Raster. Im **dritten** Schritt sind jedem Knoten  $n$  vier sog. *Spannungsabfall-Streufaktoren* entsprechend den vier Gitter-Verlaufsrichtungen zuzuteilen. Jeder Faktor  $a_{links}, a_{rechts}, a_{oben}$  und  $a_{unten}$  ist mittels  $a = -\ln(a_{min})/d_{max}$  zu berechnen, wobei  $a_{min}$  als minimaler Spannungsabfall anzunehmen ist, welcher in einem Abstand  $d_{max}$  von  $n$  gegeben sei. Weiterhin ist  $d_{max}$  definiert als Abstand zum nächstliegenden TSV bzw. der Chipbegrenzung in der entsprechenden Gitter-Verlaufsrichtung. Als **Viertes** bestimmt man die Überlagerung der Streufaktoren für alle Knoten  $n$  einzeln, was dem *qualitativen Spannungsabfall*  $IR'(n)$  entspricht. Die Berechnung ist definiert als  $IR'(n) = s_{TSV} * (P(n) + \sum_{n',a} P(n') * \exp(-a * \text{dist}(n, n')))$ , wobei  $s_{TSV} \leq 1$  einen Skalierungsfaktor für Knoten mit platzierten TSVs bezeichnet. Für jeden Knoten  $n$  sind weitere, im selben Quadranten umliegende Knoten  $n' \neq n$  bei der Berechnung von  $IR'(n)$  zu berücksichtigen. Dabei sind jeweils die beiden Faktoren  $a(n')$  zu verwenden, welche von  $n'$  ausgehend geradlinig Richtung  $n$  verlaufen. Anzumerken ist, dass der initiale Kostenterm definiert ist als  $\Gamma_{\gamma_{IR}}^0 = \max_n(IR'(n))$ .

2) *Planung von Taktsignal-TSVs*: Im Gegensatz zur Verwendung einzelner TSVs und den einhergehenden globalen, ausgeweiteten Taktsignal-Bäumen, begünstigt die Einbindung mehrerer TSVs die Generierung von lokalen, kleineren Verdrahtungsbäumen [16]. Damit lässt sich eine maßgebliche Reduktion der Leistungsaufnahme des gesamten 3D-ICs erreichen [16]. Um die Synthese der Taktsignal-Bäume dahingehend zu unterstützen, wird der folgende Algorithmus zur Planung von Taktsignal-TSVs vorgestellt. Auf jeder aktiven Lage ist der Clustering-Algorithmus *k-means++* [2] für die Taktsignalsenken durchzuführen. Die jeweiligen Cluster-Zentren definieren geeignete TSV-Positionen und bieten eine Orientierung zur Verdrahtung der lokalen Taktbäume. Die Cluster-Anzahl  $k$  wird schrittweise erhöht, bis die gewünschten Kosten  $\Gamma_{\gamma_{CP}}^{opt}$  erreichbar sind. Der zugrunde liegende Kostenterm ist definiert als  $\Gamma_{\gamma_{CP}}(k) = \sum_{c \in C} \max(\text{dist}(c.\text{center}, \text{sink} \in c)) * |\text{sink} \in c|$ . Dieser beschreibt die Summe der maximalen Distanzen zwischen Cluster-Zentren und allen jeweils zugewiesenen Taktsignalsenken und dient damit der Abschätzung der Verdrahtungslänge in Abhängigkeit vom Parameter  $k$ . Es sei angemerkt, dass dieser Algorithmus weitere Parameter wie den Clock-Skew nicht berücksichtigen kann. Eine nachfolgende Synthese der Taktbäume kann jedoch durch gezielte Platzierung von Puffern und einer Optimierung der Baumstruktur diese Vernachlässigung kompensieren [12].

3) *Planung von Signal-TSVs*: Als **Erstes** ist für jedes Netz  $n$  ein *projiziertes umspannendes Rechteck*  $bb_n^p$  zu erstellen, welches alle Pins der betroffenen aktiven Lagen einschließt. Der **zweite** Schritt bestimmt die Deadspace-Regionen, welche die Rechtecke  $bb_n^p$  auf den einzelnen aktiven Lagen einschließen. **Drittens** werden für jede aktive Lage  $d$  die zugewiesenen Netze in aufsteigender Reihenfolge nach der jeweiligen Fläche  $bb_n^p$  und den Flächen der Deadspace-Regionen sortiert. Somit sind kleine (Teil)-Netze mit geringer Überschneidung zu verfügbaren Deadspace-Regionen priorisiert. Als **Viertes** werden TSVs innerhalb der Deadspace-Regionen der jeweiligen Rechtecke  $bb_n^p$  aller Netze platziert, unter Anwendung einer lokalen Suche im Falle von ausgelasteten Deadspace-Regionen. Dieser Prozess ist schrittweise für die aktiven Lagen durchzuführen, beginnend mit der untersten Lage des 3D-ICs, welche an die Package-Bumps angeschlossen ist. Die Platzierung strebt eine dichte Anordnung von gruppierten TSVs an. Dies erlaubt eine Reduktion der Keep-Out-Zone (KOZ) ohne einen stressinduzierten negativen Einfluss auf umliegende Schaltungsblöcke zu erhöhen [14].

4) *Planung von thermischen TSVs*: Für diesen Planungsschritt findet eine Studie von Cong et al. [4] Anwendung. **Zu Beginn** sind 2D-Raster zur Beschreibung der Leistungsaufnahme sich vertikal überlappender Blöcke (analog zu Abschnitt II-B1) für alle geordnete Teilmengen  $\{d_1\}, \{d_1, d_2\}, \dots, \{d_1, \dots, d_{|D|}\}$  der aktiven Lagen zu erstellen, wobei  $d_1$  die unterste Lage beschreibt. Die folgenden Schritte sind jeweils einzeln für die 2D-Raster  $g$  der Teilmengen der aktiven Lagen durchzuführen, wobei  $d_{top}$  die jeweils oberste Lage bezeichnet. **Erstens** ist die Anzahl der bereits platzierten TSVs  $T_{curr}(b)$  für jede Kachel  $b \in g$  innerhalb von  $d_{top}$  zu ermitteln. **Zweitens** wird das Verhältnis  $r = \sum_b T_{curr}(b) / \sum_b lp(b)$  von  $d_{top}$ 's Anzahl aller TSVs und  $g$ 's gesamter Leistungsaufnahme bestimmt. **Drittens** ist die benötigte Anzahl von TSVs

$T_{des}(b) = [0.5 + r * lp(b)]$  für jede Kachel  $b \in d_{top}$  abzuleiten, um die Maximaltemperatur zu reduzieren [4]. **Viertens** erfolgt die Planung von weiteren TSVs innerhalb  $b$  falls  $T_{curr}(b) < T_{des}(b)$ , unter Nutzung einer lokalen Suche. Anzumerken ist, dass der initiale Kostenterm für die Qualitätskontrolle definiert ist als  $\Gamma_{\gamma T}^0 = \sum_b T_{des}(b) - T_{curr}(b)$ .

### C. Analyse der Entwurfsqualität

Die Optimierungsmethodik sieht eine kontinuierliche Beurteilung der Entwurfsqualität vor, basierend auf den vorgestellten Metriken zur Steuerung der TSV-Planung. Wie bereits erwähnt dient diese Qualitätskontrolle vor allem der Steuerung des inkrementellen Optimierungsprozesses. Nach Konvergenz des Hauptprozesses erfolgt eine Bewertung des globalen Entwurfsergebnis und, falls die gewünschten Kostenreduktionen nicht erreicht wurden, ein erneuter Optimierungsdurchlauf. Beispielsweise ist für den Fall, dass die Kosten  $\Gamma_{\gamma IR}^0$  nicht auf  $\Gamma_{\gamma IR}^{opt} = w_{\gamma IR}^{opt} * \Gamma_{\gamma IR}^0$  gesenkt sind, die gewünschte Entwurfsqualität in Bezug auf die Reduktion des Spannungsabfalls nicht gesichert. Es besteht daher die Notwendigkeit, weitere PG-TSVs zu platzieren, was eine erneute, angepasste Optimierungssiteration erfordert.

## III. EXPERIMENTELLE UNTERSUCHUNG

### A. Konfiguration

Die initialen Floorplans werden mittels einer akademischen Software [17] generiert, welche die Parameter Verdrahtungslänge, Flächenverbrauch und Wärmeverteilung gleich gewichtet und optimiert. Die Experimente beruhen auf repräsentativen und für den 3D-IC-Entwurf modifizierten GSRC-Benchmarks.

Die Untersuchung der vorgestellten Methodik ist derart konzipiert, dass für eine Bewertung einzelner Planungsschritte die Ergebnisse mit den nachfolgenden, nicht optimierten *Grundkonfigurationen* zu vergleichen sind. Für die Spannungsabfall-Optimierung sind PG-TSVs nur auf den Außenringen der PG-Versorgungsraster platziert. Für die Taktsignal-Verdrahtungsoptimierung dienen einzelne globale Cluster der Platzierung globaler TSVs und der entsprechenden Verdrahtungsabschätzung. Zur Planung von Signal-TSVs erfolgt eine Platzierung ohne Gruppierung. Für die Wärmeabfuhr sind keine zusätzlichen TSVs zu platzieren, sondern nur anderweitig platzierte TSVs anzunehmen.

### B. Resultate

Die durchgeführten Experimente führen zu folgenden Schlussfolgerungen. **Erstens** ermöglicht die Methodik eine merkliche Steigerung der Deadspace-Auslastung und damit einhergehend eine verbesserte Platzierung verschiedenartiger TSVs. In einigen Experimenten verbleiben weniger als 5% der Deadspace-Regionen ungenutzt. **Zweitens** können multiple Anforderungen an die Verteilung von Deadspace in frühen Entwurfsphasen erfüllt werden. Dies zeigte sich in den Experimenten anhand einer überwiegend ausgeglichen Reduktion aller Kostenterme (um durchschnittlich 30-40 %) sowie der geschätzten Signalverdrahtungslänge bei verschiedenen Optimierungskonfigurationen. Gleichwohl besteht für die Optimierung eine Abhängigkeit von den initialen Floorplans. Eine große Menge an Deadspace-Flächen kann *per se* ungünstig sein, da ebenfalls die relativen Blockanordnungen und die daraus resultierenden räumlichen Slacks maßgeblichen Einfluß auf die Flexibilität der Optimierung haben. Beispielsweise ist der Benchmark *n100* durch wenige, große Blöcke und damit einhergehend großen räumlichen Slacks gekennzeichnet. Im Vergleich mit den Ergebnissen für andere Benchmarks lässt sich feststellen, dass trotz eines geringeren Anteils von Deadspace-Flächen an der gesamten Chipfläche die Optimierung aufgrund der ausreichend großen Slacks qualitativ hochwertige Ergebnisse erreichen kann. **Drittens** beeinflusst die Anzahl der aktiven Lagen die Optimierungsergebnisse. Die besten Ergebnisse werden bei der Verwendung von drei Lagen erreicht, welche typischerweise durch großzügige Deadspace-Regionen und ausreichender räumlicher Slack gekennzeichnet sind. Die Nutzung von vier Lagen erlaubt insgesamt mehr Deadspace-Regionen und erhöht damit die Flexibilität der Optimierung. Gleichzeitig jedoch tritt aufgrund einer notwendigerweise erhöhten

Anzahl an TSVs ein Anstieg der Optimierungskomplexität und der Herstellungskosten auf. Die Verwendung von lediglich zwei aktiven Lagen bewirkt typischerweise eine derartige Limitierung der räumlichen Slacks, dass ebenfalls die Optimierungsmethodik nur begrenzt wirken kann.

## IV. ZUSAMMENFASSUNG

Die hier vorgestellte Methodik dient der multikriteriellen Layoutoptimierung für den 3D-IC-Entwurf. Die Ressource der Deadspace-Regionen, d.h. der Freiflächen zwischen Schaltungsblöcken, ist aus Sicht der Herstellungskosten generell niedrig zu halten, was im Widerspruch zu ihrer Einbeziehung und Beanspruchung bei verschiedenen Entwurfsaufgaben steht. Dies erfordert einen einheitlichen Optimierungsansatz für die Deadspace-Verteilung, um die verschiedenen, teilweise konträren Herausforderungen an die 3D-IC-Integration von 2D-Schaltungsblöcken zu meistern. Um dies zu erreichen, wurde eine Entwurfsflussenerweiterung entwickelt, welche verschiedene algorithmische Optimierungen für die TSV-Planung, Deadspace-Optimierung und auch die Bewertung der Entwurfsqualität einschließt. Eine experimentelle Untersuchung zeigt, dass sich so eine Optimierung der Deadspace-Verteilung und damit einhergehend eine verbesserte TSV- und Blockplatzierung erreichen lässt, was wiederum die Verdrahtungslänge, die Maximaltemperatur, den geschätzten Spannungsabfall als auch die Taktsignalbaumgrößen von 3D-ICs optimiert. Weiterhin ist festzustellen, dass die Anzahl der aktiven Lagen und damit der benötigten TSVs die erreichbare Entwurfsqualität beeinflusst. Dies legt die Limitierung der Anzahl der Lagen für eine 3D-IC-Integration von 2D-Schaltungsblöcken nahe.

### ANMERKUNGEN

Gefördert durch die DFG (Projekt 1401/1). Inhalte dieses Beitrages wurden unter Zusammenarbeit mit Igor L. Markov (Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, USA) erstellt und sind in [11] einsehbar.

### LITERATUR

- [1] S. N. Adya and I. L. Markov. Fixed-outline floorplanning: enabling hierarchical design. *Trans. VLSI Syst.*, 11(6):1120–1135, 2003.
- [2] D. Arthur and S. Vassilvitskii. k-means++: the advantages of careful seeding. *Proc. Symp. Discr. Alg.*, pp. 1027–1035, 2007.
- [3] K. Banerjee, S. J. Souri, P. Kapur, and K. C. Saraswat. 3-D ICs: A novel chip design for improving deep-submicrometer interconnect performance and systems-on-chip integration. *Proc. IEEE*, 89(5):602–633, 2001.
- [4] J. Cong, G. Luo, and Y. Shi. Thermal-aware cell and through-silicon-via co-placement for 3D ICs. *Proc. Des. Autom. Conf.*, pp. 670–675, 2011.
- [5] J. Cong, J. Wei, and Y. Zhang. A thermal-driven floorplanning algorithm for 3D ICs. *Proc. Int. Conf. Comput.-Aided Des.*, pp. 306–313, 2004.
- [6] X. Dong, J. Zhao, and Y. Xie. Fabrication cost analysis and cost-aware design space exploration for 3-D ICs. *Trans. Comput.-Aided Des. Integr. Circuits Sys.*, 29(12):1959–1972, 2010.
- [7] X. He, S. Dong, Y. Ma, and X. Hong. Simultaneous buffer and interlayer via planning for 3D floorplanning. *Proc. Int. Symp. Quality Elec. Des.*, pp. 740–745, 2009.
- [8] M. B. Healy and S. K. Lim. Distributed TSV topology for 3-D power-supply networks. *Trans. VLSI Syst.*, 20(11):2066–2079, 2012.
- [9] A. B. Kahng, J. Lienig, I. L. Markov, and J. Hu. *VLSI Physical Design: From Graph Partitioning to Timing Closure*. Springer, 2011.
- [10] J. Knechtel, I. L. Markov, and J. Lienig. Assembling 2-D blocks into 3-D chips. *Trans. Comput.-Aided Des. Integr. Circuits Sys.*, 31(2):228–241, 2012.
- [11] J. Knechtel, I. L. Markov, J. Lienig, and M. Thiele. Multiobjective optimization of deadspace, a critical resource for 3D-IC integration. *Proc. Int. Conf. Comput.-Aided Des.*, pp. 705–712, 2012.
- [12] D.-J. Lee and I. L. Markov. Obstacle-aware clock-tree shaping during placement. *Trans. Comput.-Aided Des. Integr. Circuits Sys.*, 31(2):205–216, 2012.
- [13] Z. Li, X. Hong, Q. Zhou, S. Zeng, J. Bian, H. Yang, V. Pitchumani, and C.-K. Cheng. Integrating dynamic thermal via planning with 3D floorplanning algorithm. *Proc. Int. Symp. Phys. Des.*, pp. 178–185, 2006.
- [14] K. H. Lu, X. Zhang, S.-K. Ryu, J. Im, R. Huang, and P. S. Ho. Thermo-mechanical reliability of 3-D ICs containing through silicon vias. *Proc. Elec. Compon. Technol. Conf.*, pp. 630–634, 2009.
- [15] E. F. Y. Young, C. C. N. Chu, and M. L. Ho. Placement constraints in floorplan design. *Trans. VLSI Syst.*, 12(7):735–745, 2004.
- [16] X. Zhao, J. Minz, and S. K. Lim. Low-power and reliable clock network design for through-silicon via (TSV) based 3D ICs. *Trans. Compon., Packag., Manuf. Technol.*, 1(2):247–259, 2011.
- [17] P. Zhou, Y. Ma, Z. Li, R. P. Dick, L. Shang, H. Zhou, X. Hong, and Q. Zhou. 3D-STAF: scalable temperature and leakage aware floorplanning for three-dimensional integrated circuits. *Proc. Int. Conf. Comput.-Aided Des.*, pp. 590–597, 2007.