

# Migração de Circuitos VLSI 2D para 3D Dirigida a Redução de Interconexões Verticais (TSV)

Sandro Sawicki<sup>1,2</sup>  
sawicki@inf.ufrgs.br

Marcelo Johann<sup>1</sup>  
johann@inf.ufrgs.br

Ricardo Reis<sup>1</sup>  
reis@inf.ufrgs.br

<sup>1</sup>PPGC/PGMicro - Instituto de Informática  
UFRGS - Universidade Federal do Rio Grande do Sul  
Porto Alegre, Brazil.

<sup>2</sup>DETEC – Departamento de Tecnologia  
UNIJUI – Universidade Regional do Noroeste do Estado do Rio Grande do Sul  
Campus Santa Rosa, RS, Brasil.

**Abstract** — Este trabalho apresenta um conjunto de algoritmos desenvolvidos para reduzir o número de vias em circuitos VLSI 3D. A otimização é obtida pelo uso de duas estratégias distintas: a análise prévia da estrutura interna do circuito e a redução do número de conexões verticais não-adjacentes. Os algoritmos propostos, além de reduzir o número de vias-3D, adaptam a lógica dos circuitos 2D para os 3D mantendo o balanceamento de área e dos pinos de I/O entre as diferentes camadas. Os resultados experimentais mostram que essas técnicas reduzem o número total de vias-3D em 19%, 18%, 12% e 16% em duas, três, quatro e cinco tiers, respectivamente, comparados com os resultados das abordagens atuais.

## I. INTRODUÇÃO

Os circuitos em três dimensões (3D) surgem como uma opção, uma nova forma de reduzir as interconexões. Teórica [14] e empiricamente [1, 2, 3 e 5], circuitos 3D podem reduzir o tamanho das interconexões. Com o tamanho das conexões reduzido, diminuem-se os focos de congestionamento, pois aumenta o número de conexões curtas. Conseqüentemente, com os caminhos mais curtos, a média da capacitância e da resistência são reduzidas, assim como o número de repetidores para os caminhos mais longos. O trabalho de Obenaus e Szymanski [15] mostra que essa redução é proporcional ao tamanho do circuito. Além disso, é razoável considerar que a implementação de um projeto 3D possa melhorar o *timing*, se comparada com uma implementação 2D. Para isso, novas teorias e algoritmos devem ser desenvolvidos.

Grandes empresas, como IBM, Intel, AMD, Samsung, Micron, ST, Cadence, Infineon, e startups, como Ziptronix, Xanoptix, ZyCube e Tezzaron, estão investindo em soluções relacionadas a essa área. Na academia, são destacadas iniciativas do MIT, Cornell University, University of Minnesota, Stanford University, University of Hannover

Leibniz, IMEC, Purdue University, Tohoku University entre outras.

Além da diferença topológica entre as estruturas 2D e 3D, é necessária a atenção especial para dois tópicos: questões térmicas [3, 5 e 17] e interconexões verticais (TSV – Through-Silicon Via). Problemas de aquecimento podem ser tratados com a adição de vias térmicas [3] e com os posicionadores e roteadores desenvolvidos especificamente para esse propósito [2]. Já o mecanismo de comunicação intercamadas, chamado de TSV ou via-3D, impõe inúmeras limitações para o projeto VLSI 3D por quatro razões principais (ainda não abordadas satisfatoriamente na literatura): (1) todos os elementos de comunicação que implicam os níveis de integração face-to-back e back-to-back ocupam espaço na camada ativa, limitando o posicionamento de células e blocos [1]; (2) a dimensão física da via-3D é consideravelmente grande se comparada com as vias normais, de modo que sua superpopulação pode inviabilizar um projeto 3D [16]; (3) as vias-3D têm características elétricas diferentes das conexões normais [18], especialmente se considerarmos que as conexões verticais precisam atravessar todas as camadas de metal; (4) vias-3D impõem problemas elétricos e de yield tanto por seus elementos serem difíceis de fabricar quanto por consumirem muitos recursos de roteamento [16 e 17].

É importante entender que o valor dos recursos usados pelas vias-3D é muito grande. Considerando-se que o caminho de uma tier para outra adjacente atravessa todas as camadas de metal, é sensato dizer que qualquer ligação vertical superior à adjacente pode ser muito mais custosa para o roteamento, sem mencionar o atraso provocado e quanta área ativa é ocupada.

Em vista disso, é proposto um fluxo para o projeto de circuitos 3D que considera a área do circuito, conexões verticais longas (aquelas que cruzam camadas não

adjacentes), número de tiers, número máximo de vias-3D, balanceamento de áreas e pinos de I/O. Estuda-se o problema pela perspectiva do particionamento, explorando-se métodos que auxiliam na compreensão da estrutura interna do circuito. O conjunto de algoritmos (fluxo completo desenvolvido) está incorporado a uma ferramenta chamada de LH-3D, descrita com detalhes ao longo do artigo. Métodos e contribuições são apresentados e validados por meio de experimentos práticos.

## II. TRABALHOS RELACIONADOS

Na literatura que aborda circuitos 3D não se tem encontrado trabalhos dedicados aos problemas causados pelas vias-3D. Contudo, é fácil enumerar diversas questões sobre este assunto que demandam cuidados, por exemplo: (1) a superpopulação de vias-3D gera o aumento de área ativa, ocupando o lugar dos transistores e, conseqüentemente, aumentando a área do circuito; (2) no momento em que uma via-3D cruza duas camadas adjacentes, surge a necessidade de posicionar e legalizar a sua estrutura (podem ser tratados como obstáculos móveis); (3) pela diferença física estrutural, as vias-3D têm características elétricas diferentes das conexões normais (o problema ocorre quando a via-3D faz parte do caminho crítico); (4) as vias-3D são obstáculos e, como tal, utilizam recursos de roteamento; (5) o pitch de uma via-3D é muito grande se comparado com o de uma conexão normal, isso consome bastante área do circuito. Com base nessas informações e em pesquisas específicas sobre o assunto descritas ao longo de todo este trabalho, nota-se que grande parte dos estudos explora a divisão de células e blocos entre camadas sob a perspectiva da redução do número de conexões entre partições, ou seja, uma abordagem que adapta de forma simplista as estratégias dos circuitos 2D para resolver problemas em três dimensões. Nessas soluções, o problema ocorre no momento em que a célula ou partição é atribuída à camada, pois os algoritmos tradicionais que atuam no particionamento de hipergrafos não percebem o arranjo proposto pela tecnologia 3D (arranjo em linha).

Assim, diversos trabalhos [1, 2, 3, 4 e 19], utilizam o particionador de hipergrafos hMetis [8]. Outros autores, como Lee e Lim [20], utilizam min-cut por meio do algoritmo de Fiduccia-Matheyses [21]. Entretanto, como mencionado anteriormente, para problemas específicos de circuitos 3D (como é o caso da redução de vias-3D), tanto hMetis quanto FM não conseguem obter resultados satisfatórios. Outros estudos [12 e 22] utilizaram algoritmos direcionados a forças para realizar o particionamento do circuito. Nesse contexto, as células e blocos sofrem a ação de forças que os deslocam para outras camadas. Também esses trabalhos não abordam pontualmente formas de redução total de vias-3D.

Apesar de os estudos citados acima relatarem os problemas causados pelo excesso de vias-3D no projeto dos

circuitos, nenhum deles apresenta especificamente uma solução para tal.

## III. FLUXO LH-3D PROPOSTO

Os algoritmos do fluxo LH-3D estão divididos em três grupos: LHp-3D, LHu-3D e LHR-3D. Os pertencentes ao grupo LHp-3D atuam de forma diferente das ferramentas de particionamento atuais, pois apresentam uma metodologia de análise prévia da estrutura interna do circuito que é executada antes do particionamento das células. Essa análise consiste em encontrar o menor caminho lógico entre pares de pinos de I/O a fim de mapear as informações de conectividade das células. Com base nos experimentos realizados pelo LHp-3D, criou-se o grupo LH-3D. Esse grupo visa reduzir o número de vias-3D por meio do aumento do desvio padrão do número de pinos de I/O que estão distribuídos pelas tiers. Por último, o grupo LHR-3D que tem por objetivo convergir para a redução do número total de vias-3D por meio da minimização das vias-3D que cruzam mais de duas tiers adjacentes. Nas seções seguintes, o fluxo de projeto e os demais algoritmos utilizados pelo LH-3D são descritos e discutidos com maiores detalhes.

### A. Algoritmos LHp-3D e LH-3D

Antes do posicionamento, uma netlist 2D NL é composta por um conjunto de células  $G = \{g_1, g_2, g_3, \dots, g_n\}$ , um conjunto de pinos de I/O  $P = \{p_1, p_2, p_3, \dots, p_m\}$  e um conjunto de redes  $N = \{n_1, n_2, n_3, \dots, n_m\}$ . Um hipergrafo HG representa a netlist, onde  $G \cup P$  é o conjunto de nodos e  $N$  é o conjunto de hipergrafos. A posição fixa de cada pino de I/O  $p_i$  é representada por  $X[i]$  e  $Y[i]$  ( $i \leq m$ ) e sua orientação por  $OR(p_i) \in \{\text{north, south, east, west}\}$ . A área  $A$  (altura  $H$  largura  $W$ ) tem seu canto inferior esquerdo na coordenada  $(x_{ini}, y_{ini})$ . Usualmente, os pinos de I/O cobrem toda a borda do circuito. A relação de espaços em branco  $S$  na área do posicionamento é alcançada pela subtração da área total de células ( $GA$ ) pela área disponível dentro dos pinos de I/O. A relação de aspecto  $AR$  é computada pela divisão de  $W$  por  $H$ .

Se  $Z$  é o conjunto dos números que representam as tiers  $\{1, 2, \dots, z\}$ . O problema pode ser definido como: dado uma netlist 2D NL com pinos de I/O fixos, encontre o conjunto de tiers  $T = \{t_1, t_2, \dots, t_z\}$  e seus correspondentes  $A_i, AR_i, GA_i, W_i, H_i, P_i, S_i, OR_i, X_i$  e  $Y_i$  ( $i \leq z$ ) tal que:

$$P_1 \cup P_2 \cup \dots \cup P_z = P \quad (1)$$

$$\forall (a, b \in Z) (a \neq b \rightarrow P_a \cap P_b = \emptyset) \quad (2)$$

$$\forall (i \in Z) S_i \approx S \quad (3)$$

$$\forall (i \in Z) \forall (j \in Z) W_i = W_j \wedge H_i = H_j \quad (4)$$

$$\forall (i \in Z) AR_i \approx AR \quad (5)$$

$$\forall (i \in Z) (\forall a \in P_i (OR_i(a) = OR(a))) \quad (6)$$

$$\forall(t \in Z)(\forall a, b \in P_i(OR(a)=OR(b) \wedge X_i[a] < X_i[b] \rightarrow X[a] < X[b])) \quad (7)$$

$$\forall(t \in Z)(\forall a, b \in P_i(OR(a)=OR(b) \wedge Y_i[a] < Y_i[b] \rightarrow Y[a] < Y[b])) \quad (8)$$

Em outras palavras, cada tier terá seu próprio conjunto de pinos de I/O (equações 1 e 2); os espaços em branco (whitespace) e a relação de aspecto (aspect ratio) preservados na mesma proporção do circuito 2D original. (equações 3, 4 e 5); a orientação e a ordem dos pinos também são preservadas (equações 6, 7, e 8).

Seja  $LD(p_i, p_j)$  o tamanho do menor caminho em HG de  $p_i$  para  $p_j$  (por exemplo, a distância lógica entre  $p_i$  e  $p_j$ ). Pode-se descrever o algoritmo de particionamento de pinos de I/O de forma detalhada abaixo:

- 1 Computar  $LD(i, j) \forall i, j \in P$
- 2 Criar um grafo completo PG tal que P seja o conjunto de nodos e  $LD(i, j) (i, j \in P)$  seja o custo das aresta conectando os nodos  $i$  e  $j$ .
- 3 Executar o particionamento de PG em  $P_1, P_2, \dots, P_z$  buscando a minimização do corte (min-cut) e o balanceamento de pinos entre as partições.
- 4  $\forall(i \in Z)GA_i \approx \frac{GA}{z}$  (9)
- 5  $\forall(i \in Z)A_i = GA_i \times (1 + S_i)$  (10)
- 6  $\forall(i \in Z)W_i = \sqrt{A_i} \times AR_i; H_i = \frac{\sqrt{A_i}}{AR_i}$  (11)
- 7  $\forall(i \in Z)\forall(p \in P_i)X_i[p] = x_{mi} + \frac{(X[p] - x_{mi}) \times W_i}{W}$  (12)
- 8  $\forall(i \in Z)\forall(p \in P_i)Y_i[p] = y_{mi} + \frac{(Y[p] - y_{mi}) \times H_i}{H}$  (13)
- 9 Legalizar os pinos de I/O . (14)

O algoritmo LHp-3D inicia com a extração de um subgrafo de pares de pinos de I/O da *netlist* original. As arestas correspondem ao menor número de células que separa os pinos (distância lógica). Com essa informação, executa-se o particionamento dos pinos por meio da ferramenta hMetis. Considera-se a distância lógica nessa etapa. Os pinos com menor distância lógica ficam na mesma partição, isso potencializa a migração de todo o caminho lógico para a partição de seus respectivos pinos, o que reduz, consequentemente, o corte entre as elas.

Logo após o particionamento de pinos, as células são particionadas e os pinos pré-fixados servem de guias para a migração das células. O algoritmo LHu-3D aumenta o desbalanceamento do número de pinos de I/O entre as partições em 10% e 25%. A redução foi, em média, 3% comparado com a solução proposta pelo LHp-3D o que reforça a idéia da influência dos pinos de I/O na redução das vias (TSV). Como os resultados obtidos pelo LHu-3D não foram significantes em comparação com o LHp-3D, optou-se por comparar o LHp-3D com as demais abordagens.

## B. Algoritmo LHR-3D

O algoritmo LHR-3D reduz as vias que cruzam camadas não-adjacentes. Para isso, cria uma função de custo que penaliza o surgimento de vias que atravessam mais de duas tiers. O LHR-3D inicia com a perturbação de células e pinos, com trocas duplas ou simples. Ao final de cada perturbação, avalia-se o resultado. Caso o custo do resultado tenha sido menor que o anterior, a troca é aceita, caso contrário, rejeitada.

Os valores de  $v$ ,  $a$  e  $p$  são computados da seguinte forma:

- Para cada rede, calcula-se o quadrado do número de vias; adiciona-se o número computado para cada rede para que seja obtido em  $v$ . Aplica-se o quadrado do valor para punir as redes que tenham conexões longas e incentivar o aumento de redes curtas.
- Para computar  $a$ , calcula-se primeiro a área de células de todas as *tiers*; o custo do desbalanceamento é a subtração da maior área pela menor área.
- O valor de  $p$  é calculado da mesma maneira que o valor de  $a$ .

$$custo = \frac{(w_v \times v)}{v_1} + \frac{(w_a \times a)}{a_1} + \frac{(w_p \times p)}{p_1}$$

## Algoritmo proposto para redução de conexões longas (LHR-3D)

- |         |                                                                                                                                                                               |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Passo 1 | Computar <i>netlist</i> mantendo orientação das <i>tiers</i><br>Calcular número inicial de:                                                                                   |
|         | <ul style="list-style-type: none"> <li>• vias-3D</li> <li>• desbalanceamento da área de células;</li> <li>• pinos de I/O</li> </ul>                                           |
| Passo 2 |                                                                                                                                                                               |
| Passo 3 | Executar <b>procedimento de perturbação</b>                                                                                                                                   |
| Passo 4 | Efetuar a troca:<br>$\{(pino, pino)   (pino, célula)   (célula, pino)   (célula, célula)\}$                                                                                   |
| Passo 5 | Calcular o custo da troca:<br>$\Delta \text{Custo} = \text{Custo (Nova Solução)} - \text{Custo (Solução)}$ ;<br>Se $(\Delta \text{Custo} < 0)$<br>Aceita Troca ();<br>Passo 3 |
| Passo 6 | Senão<br>Rejeita Troca ();<br>Desfaz Troca ();<br>Passo 3                                                                                                                     |
|         | Se o custo não mudou (0,7%) em $n$ iterações seguidas, fim.                                                                                                                   |
| Passo 7 | Senão<br>Passo 3                                                                                                                                                              |

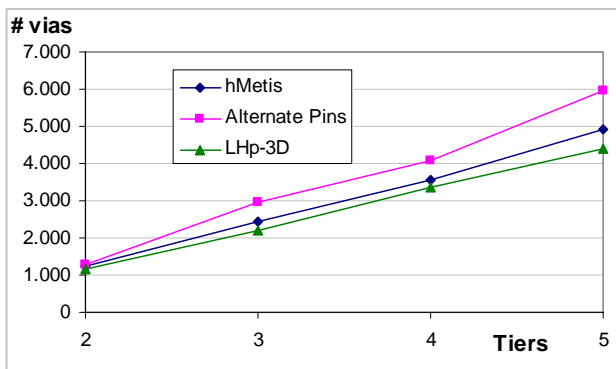
## IV. RESULTADOS EXPERIMENTAIS

Os resultados deste trabalho foram gerados a partir do conjunto de benchmarks ISPD 2004 (ISPD 2004 - IBM

STANDARD CELL BENCHMARKS WITH PADS, 2004) resumidos na Tabela 1, abaixo. Esse conjunto de circuitos benchmarks 2D foram otimizados e convertidos em circuitos 3D e vêm a complementar o conjunto de benchmarks disponíveis no meio acadêmico. Foram criados circuitos com duas, três, quatro e cinco tiers de todo o conjunto de benchmarks.

**Tabela 1: Benchmarks IBM ISPD 2004 com pads (ISPD IBM BENCHMARK WITH PADS, 2004)**

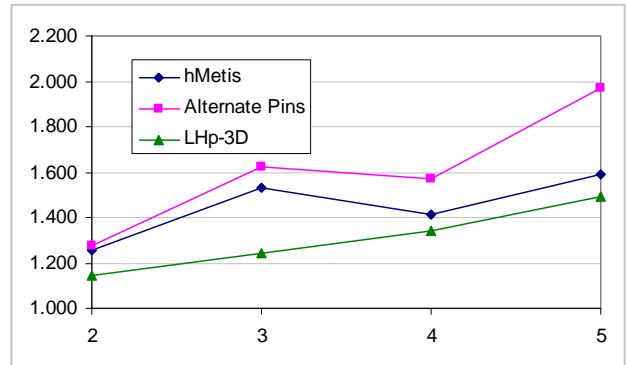
Bench	#Cells	#Pads	#Nets
ibm01	12.506	246	14.111
ibm02	19.342	259	19.584
ibm03	22.853	283	27.401
ibm04	27.220	287	31.970
ibm05	28.146	1201	28.446
ibm06	32.332	166	34.826
ibm07	45.639	287	48.117
ibm08	51.023	286	50.513
ibm09	53.110	285	60.902
ibm10	68.685	744	75.196
ibm11	70.152	406	81.454
ibm12	70.439	637	77.240
ibm13	83.709	490	99.666
ibm14	147.088	517	152.772
ibm15	161.187	383	186.608
ibm16	182.980	504	190.048



**Figura 1: Número Total de Vias (TSV)**

Os algoritmos desenvolvidos neste trabalho foram comparados com o particionador de hipergrafos mais utilizado no domínio VLSI 3D e que mostra os melhores resultados na redução de conexões que cruzam partições, hMetis [8]; outras comparações foram realizadas com o algoritmo Alternate Pins, uma estratégia que fixa os pinos de I/O alternadamente entre as partições; e também com o algoritmo ZPlace [12] uma estratégia direcionada a forças que usa o método quadratic placement para distribuir as células entre as camadas. Utilizou-se para o ZPlace, a informação dos pinos de I/O oferecidas pelo hMetis e pelo LHp-3D (menor caminho lógico).

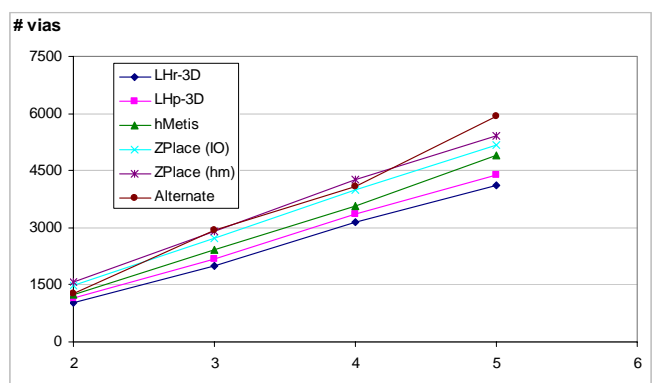
A Figura 1 mostra o número total de vias (TSV) em comparação com a ferramenta hMetis (10%, 10%, 6% e 11%) e o método Alternate Pins (10%, 26%, 18% e 26%), respectivamente para duas, três, quatro e cinco tiers.



**Figura 2: Número Máximo de Vias entre Tiers Adjacentes**

A Figura 2 mostra o número máximo de vias entre tiers adjacentes. Esse valor tem influência direta na área da tier. Percebe-se que a redução foi de 10%, 19%, 5% e 6% comparado com o algoritmo hMetis e 10%, 23%, 15% e 24% comparado com o algoritmo Alternate Pins, para duas, três, quatro e cinco tiers.

A Figura 3 mostra o número total de vias comparando os algoritmos, hMetis, Alternate Pins, ZPlace (I/O), ZPlace (hMetis), LHp-3D e LHp-3D. Percebe-se que os algoritmos LHp-3D e LHp-3D oferecem melhores resultados. Isso se deve a característica dos algoritmos, que penalizam o surgimento de novas vias, convergindo para a redução do seu número total.



**Figura 3: Número Total de Vias (TSV)**

O mesmo acontece na Figura 4, pois a redução do número total de vias influencia diretamente na redução do número máximo de vias. Já a Figura 5 e a Figura 6 descrevem o número de redes que cruzam 4 e 5 tiers respectivamente. Verifica-se que as conexões que não

cruzam as camadas foram aumentadas e as que cruzam as camadas diminuídas. Isso ocorreu, pois a função de custo penalizou o surgimento de conexões verticais longas (TSV que cruzam mas de duas camadas).

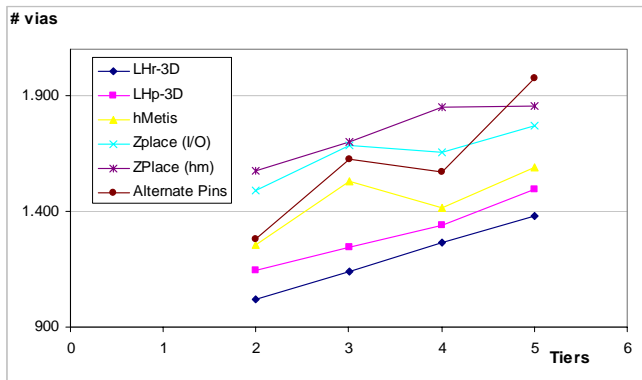


Figura 4: Número máximo de Vias entre tiers adjacentes

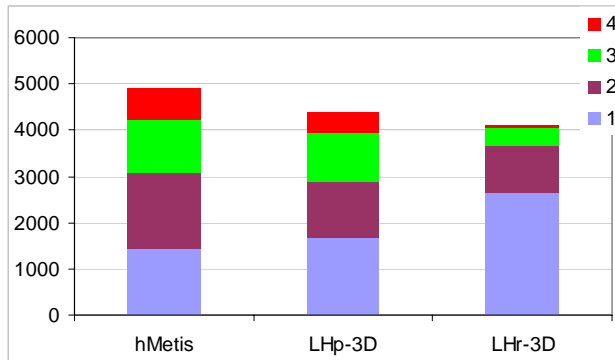


Figura 5: Número total de redes entre tiers adjacentes e não adjacentes em um projeto de 5 tiers.

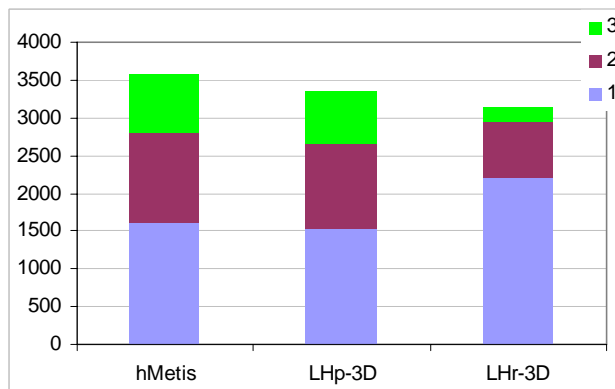


Figura 6: Número total de redes entre tiers adjacentes e não adjacentes em um projeto de 4 tiers.

## V. CONCLUSÕES

Este trabalho apresentou duas estratégias eficientes para reduzir o número total de vias em circuitos VLSI 3D. A

primeira trabalha com a análise prévia da estrutura do circuito e a segunda reduz o número de conexões verticais não adjacentes. Essa pesquisa resultou no desenvolvimento da ferramenta LH-3D, um fluxo de projeto que auxilia o particionamento de células e pinos em circuitos VLSI 3D (standard cells). Sua estrutura é composta por um conjunto de algoritmos de otimização que adaptam a lógica dos circuitos 2D para os 3D

O LHp-3D reduziu o número de vias, em média 11%, em comparação com o hMetis e 21% em comparação com o método Alternate Pins. Já o LHr-3D reduziu significativamente o número de conexões que cruzam tiers não-adjacentes e reduziu o número total de vias, em média de 7%, 16% e 29% comparado com os algoritmos hMetis, LHp-3D e ZPlace.

## REFERÊNCIAS

- [1] 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. *IEEE Design and Test of Computers – special issue on 3D integration*; pp 498-510, Nov.-Dec. 2005.
- [2] C. Ababei, Y. Feng, B. Goplen, H. Mogal, T. Zhang, K. Bazargan and S. Sapatnekar. Placement and Routing in 3D Integrated Circuits. *IEEE Design and Test of Computers – special issue on 3D integration*; pp 520-531, Nov.-Dec. 2005.
- [3] B. Goplen; S. Sapatnekar; Efficient Thermal Placement of Standard Cells in 3D ICs using Forced Directed Approach. *In: International Conference on Computer Aided Design, ICCAD'03*, November, San Jose, California, USA, 2003.
- [4] E. Wong; S. Lim. 3D Floorplanning with Thermal Vias. *In: DATE '06: Proceedings of the Conference on Design, Automation and Test in Europe*, 2006. p.878–883.
- [5] Das, S.; Fan, A.; Chen, K.-N.; Tan, C. S.; Checka, N.; Reif, R. Technology, performance, and computer-aided design of three-dimensional integrated circuits. *In: ISPD'04: Proceedings Of The 2004 International Symposium On 59 Physical Design*, 2004, New York, NY, USA. Anais. . . ACM Press, 2004. p.108–115.
- [6] Patti, R. Three-dimensional integrated circuits and the future of system-on-chip designs. *Proceedings of IEEE*, [S.l.], v.94, p.1214–1224, 2006.
- [7] Hentschke, R. et al. 3D-Vias Aware Quadratic Placement for 3D VLSI Circuits. *In: IEEE Computer Society Anual Symposium on VLSI, ISVLSI, 2007*, Porto Alegre, RS, Brazil. Proceedings. . . Los Alamitos: IEEE Computer Society, 2007. p.67–72.
- [8] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar. Multilevel Hypergraph Partitioning: Application in VLSI domain. *In Proceedings of 34th Annual Conference on Design Automation, DAC 1997*, pages 526–529, 1997.
- [9] Sawicki, Sandro ; Hentschke, Renato Fernandes ; Johann, M. O. ; Reis, Ricardo Augusto da Luz . An Algorithm for I/O Pins Partitioning Targeting 3D VLSI Integrated Circuits. *In: IEEE International Midwest Symposium on Circuits and Systems*, 2006, San Juan. Proceedings, 2006.
- [10] K. Bernstein; P. Andry; J. Cann; P. Emma; D. Greenberg; W. Haensch; M. Ignatowski; S. Koester; J. Magerlein; R. Puri; A. Young. Interconnects in the Third Dimension: Design Challenges for 3D ICs. *In: Design Automation Conference*, 2007. DAC'07. 44th ACM/IEEE. 2007 Page(s):562 - 567
- [11] ISPD 2004 - IBM Standard Cell Benchmarks with Pads. [http://www.public.iastate.edu/~nataraj/ISPD04\\_Bench.html#Benchmark\\_Description](http://www.public.iastate.edu/~nataraj/ISPD04_Bench.html#Benchmark_Description). Access on Mar 2008.

- [12] R. Hentschke, G. Flach, F. Pinto, and R. Reis, "Quadratic Placement for 3D Circuits Using Z-Cell Shifting, 3D Iterative Refinement and Simulated Annealing," Proc. Symp. on Integrated Circuits and Syst. Des. '06, 220-225.
- [13] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, Optimization by simulated annealing, Science, 1983, 220, pages 671--680
- [14] BANERJEE, K.; SOURI, S.; KAPUR, P.; SARASWAT, K. 3D-ICs: A novel chip design for improving deep submicrometer interconnect performance and systems on-chip integration. Proceedings of IEEE, New York, v.89, p.602-633, 2001.
- [15] OBENAU, S.; SZYMANSKI, T. Gravity: fast placement for 3-d vlsi. ACM Transactions on Design Automation of Electronic Systems (TODAES), Volume 8, Issue 3, Pages: 298 - 315, 2003.
- [16] PAVLIDIS F.; FRIEDMAN, E. G., Three-dimensional Integrated Circuit Design; Morgan Kaufmann, 304 pgs, 2008.
- [17] KOYANAGI, M.; FUKUSHIMA, T.; TANAKA, T.; High-Density Through Silicon Vias for 3-D LSIs;. In: Proceedings of the IEEE, Volume: 97, Issue: 1, On page(s): 49-59, 2009.
- [18] RYU, C. et al., High frequency electrical circuit model of chip-to-chip vertical via interconnection for 3-D chip stacking package, In: Proceedings of the IEEE Topical Meeting Electr. Performance, pp. 151-154, 2005.
- [19] TSAI, Y.; WANG, F.; XIE, Y.; VIJAYKRISHNAN, N.; IRWIN, M.; Design Space Exploration for 3-D Cache; IEEE Transaction on Very Scale Integration (VLSI) Systems, Vol. 16, no 4, April, 2008.
- [20] LEE, Y.; KIM; HUANG, G. BAKIR, M.; JOSHI, T.; LIM, S.; Co-Design of Signal, Power, and Thermal Distribution Networks for 3D ICs. In: DATE 2009.
- [21] FIDUCCIA, C. M., MATTHEYSES. A Linear Time Heuristic for improving network partitions. In. Proceedings 19th IEEE Design Automation Conference. Pages 175-181, 1982.
- [22] KAYA, I.; SALEWSKI, S.; OLBRICH, M.; BARKE, E. Wirelength Reduction Using 3- D Physical Design. In: Integrated Circuit And System Design – Power And Timing Modeling, Optimization And Simulation; Proceedings Of 14th International Workshop, PATMOS 2004, 2004. Anais. . . [S.l.: s.n.], 2004.