

# Exploração de paralelismo na etapa de legalização de circuitos digitais através do uso de estruturas de dados geométricas

Sheiny Fabre<sup>1</sup>, Laércio Pilla<sup>1</sup>, José Luís Güntzel<sup>1</sup>

<sup>1</sup>Laboratório de Computação Embarcada  
Florianópolis – Universidade Federal de Santa Catarina – SC – Brasil

sheiny.fabre@posgrad.ufsc.br

***Resumo.** Após o posicionamento inicial dos elementos de um circuito integrado, estes precisam ser legalizados para considerar regras de fabricação. Algoritmos de legalização devem tratar grandes quantidades de dados, produzindo uma solução determinística, com a menor perturbação possível do posicionamento. Este trabalho propõe o uso da estrutura de dados KD-tree para particionar o circuito viabilizando a legalização em paralelo das partições.*

## 1. Introdução

A evolução do processo de fabricação de *Circuito Integrados* (CIs) juntamente com as ferramentas de automação de projetos eletrônicos viabilizaram o projeto de CIs cada vez mais complexos, os quais possuem um maior número de transistores e regras de projeto [Flynn et al. 2007]. O projeto de CIs requer uma sequência de etapas, dentre as quais a síntese física é a responsável por gerar a descrição fabricável. A síntese física busca posicionar e rotear os *layouts* das portas lógicas, *flip-flops*, (referenciados por “células”) sobre uma região 2-D [Kahng et al. 2011].

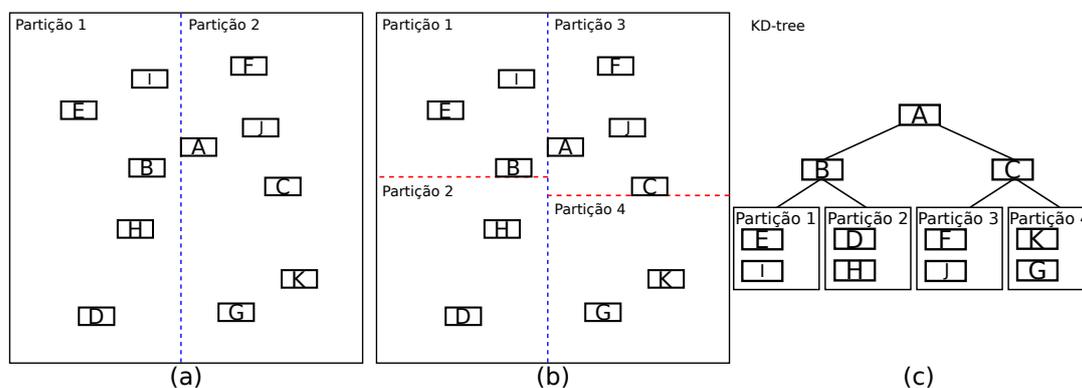
Primeiramente é necessário encontrar um posicionamento inicial para todos elementos de forma a otimizar métricas como por exemplo: redução do comprimento das interconexões [Alpert et al. 2012]. Para tratar circuitos contemporâneos complexos, isto é feito ignorando algumas restrições de legalidade do circuito, portanto células podem estar sobrepostas e desalinhadas em relação às linhas e colunas predeterminadas pelas regras de desenho. Para possibilitar a fabricação do circuito, a legalização deve remover as violações ignoradas pelo posicionamento, realizando a menor quantidade possível de movimentos para preservar a qualidade da solução [Darav et al. 2017].

Algoritmos de legalização precisam tratar grandes quantidades de dados de forma rápida e determinística, para preservar a qualidade da solução. O uso de uma estrutura de dados inapropriada para trabalhar com objetos geométricos, tais como os *layouts* das células, exige manipulações ou adaptações, aumentando o tempo de execução da técnica. Portanto este trabalho propõe o uso da estrutura de dados KD-tree para particionar o circuito, possibilitando o uso de algoritmos paralelos de legalização [Bentley 1975].

## 2. Uso de KD-tree na legalização

A estrutura da KD-tree é representada por uma árvore binária onde cada nó representa um ponto em um espaço  $k$  dimensional. A KD-tree 2-D é construída recursivamente onde a cada iteração par a árvore é dividida verticalmente, e nas interações ímpares horizontalmente. Para determinar onde a árvore será dividida utiliza-se o ponto mediano em relação

ao eixo da respectiva iteração. Na Figura 1, por exemplo: na primeira iteração a árvore é dividida verticalmente em relação ao eixo x onde está a célula A, e na iteração seguinte a divisão será horizontal em relação ao eixo y pelas células B e C, e assim por diante. A árvore naturalmente representa um particionamento, pois cada nó possui uma área de alcance e todos nós abrangidos podem ser obtidos visitando os descendentes do nó.



**Figura 1. (a): está ilustrado a primeira iteração, (b): recursivamente, as partições estão divididas horizontalmente, (c): a estrutura final da KD-tree.**

Esta estrutura de dados viabiliza a legalização em paralelo, pois cada subárvore (partição) possui uma área de alcance e um conjunto independente de células. Sendo assim, é necessário somente legalizar e fixar os nós até um determinado nível da árvore e conseqüentemente as subárvores desses nós serão as partições a serem legalizadas em paralelo. Por exemplo, na Figura 1 as células A, B e C seriam legalizadas e fixadas sequencialmente, e as quatro subárvores podem ser legalizadas em paralelo.

### 3. Trabalhos futuros

A validação técnica proposta será feita utilizando o algoritmo Abacus [Spindler et al. 2008] juntamente com os circuitos fornecidos pela competição ICCAD2017 [Darav et al. 2017], onde serão avaliados os tempos de execução, número de partições e a perturbação na qualidade da solução.

### Referências

- Alpert, C., Li, Z., Nam, G.-J., Sze, C. N., Viswanathan, N., and Ward, S. I. (2012). Placement: Hot or not? In *Proc. of ISPD*, pages 283–290. ACM.
- Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517.
- Darav, N. K., Bustany, I. S., Kennings, A., and Mamidi, R. (2017). Iccad-2017 cad contest in multi-deck standard cell legalization and benchmarks.
- Flynn, D., Aitken, R., Gibbons, A., and Shi, K. (2007). *Low power methodology manual: for system-on-chip design*. Springer Science & Business Media.
- Kahng, A. B., Lienig, J., Markov, I. L., and Hu, J. (2011). *VLSI physical design: from graph partitioning to timing closure*. Springer Science & Business Media.
- Spindler, P., Schlichtmann, U., and Johannes, F. M. (2008). Abacus: fast legalization of standard cell circuits with minimal movement. In *Proc. of ISPD*, pages 47–53. ACM.