

POSICIONAMENTO DE TRANSISTORES EM REDES NÃO-SÉRIE-PARALELO

MAICON SCHNEIDER CARDOSO; LEOMAR SOARES DA ROSA JUNIOR,
FELIPE DE SOUZA MARQUES

Universidade Federal de Pelotas – Grupo de Arquitetura e Circuitos Integrados –
{mscardoso,leomarjr,felipem}@inf.ufpel.edu.br

1. INTRODUÇÃO

O posicionamento de componentes em redes de transistores é de extrema importância no modelo *Very-Large-Scale Integration* (VLSI) para a concepção de circuitos integrados (CIs). Essa técnica é uma das etapas iniciais da síntese física, e, devido a sua importância, tornou-se um tema amplamente pesquisado nas últimas décadas, levando a diferentes abordagens, métodos e algoritmos para a melhor solução desse problema.

O método clássico de posicionamento (UEHARA, 1981) foi pioneiro ao demonstrar o impacto que esse estágio tem na área do leiaute final do CI, ilustrado na Figura 1. A partir desse estudo, surgiram outras soluções com o objetivo de otimizar e estender o método para outras topologias e tecnologias de transistores (HWANG, 1990; SADAKANE, 1995; GUPTA, 2000; SERDAR, 2001; IIZUKA, 2005).

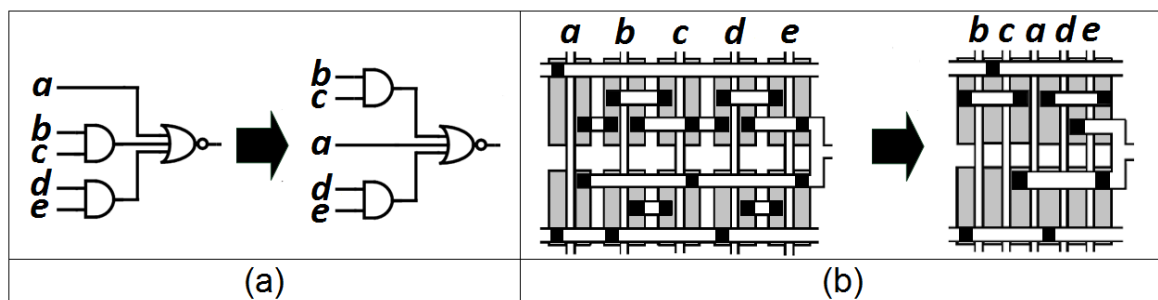


Figura 1. Impacto do ordenamento no leiaute. (a) Posicionamento das entradas do circuito lógico. (b) Leiautes sem otimização e com otimização.

Em trabalhos recentes (KAGARIS, 2007; POSSANI, 2011), observou-se um decréscimo recorrente no número de componentes necessários para implementar determinadas funções lógicas, obtido graças as redes não-série-paralelo (NSP), zalternativa às clássicas redes série-paralelo (SP). Entretanto, esses trabalhos abordam apenas a síntese lógica, não havendo investigação quanto ao impacto no leiaute do circuito integrado.

Nesse cenário, esse trabalho busca propor um modelo baseado no método clássico de posicionamento de transistores para redes NSP. A partir disso, poder-se-á comparar o leiaute do CI de redes NSP com as redes SP, avaliando os impactos (positivos e negativos) que a topologia NSP causa no projeto do circuito final (em termos de área, potência consumida, atraso de sinal e complexidade de roteamento, por exemplo), além de dar subsídio ao estudo e desenvolvimento de outras etapas da síntese física para redes NSP.

2. METODOLOGIA

Utilizando-se de representação em forma de grafos da rede de transistores e caminhos eulerianos como base do algoritmo, o modelo está dividido em dois

estágios principais: a busca das possíveis soluções para posicionamento e escolha das melhores soluções nos conjuntos resposta.

Primeiramente, a rede é representada em forma de grafo e todos os caminhos de Euler dos planos *pull-up* (PU) e *pull-down* (PD) são encontrados a partir do algoritmo de Fleury. A Figura 2 ilustra esse processo (apenas para o plano PU) para a função descrita abaixo (Equação 1).

$$f = !(a \cdot b + a \cdot c \cdot e + d \cdot e + b \cdot c \cdot d) \quad (1)$$

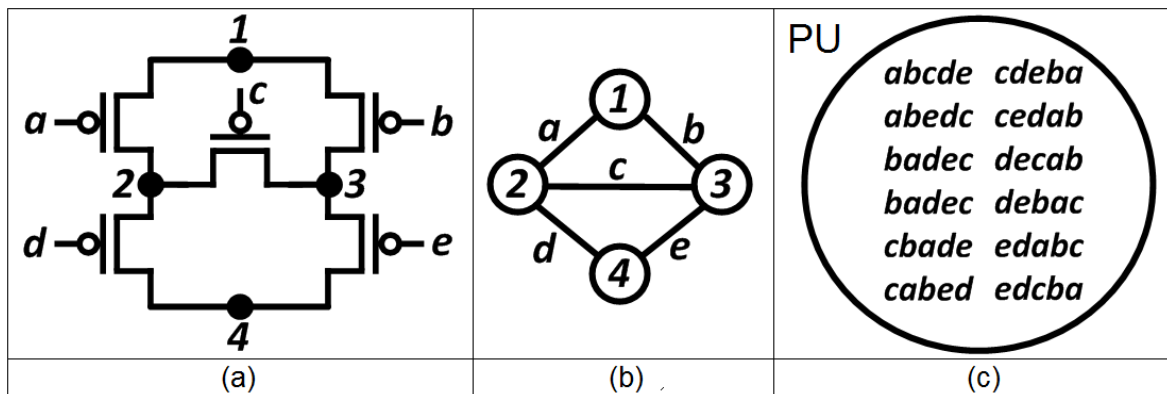


Figura 2. (a) Plano *pull-up* da rede de transistores NSP descrita na Equação 1. (b) Grafo equivalente ao plano. (c) Conjunto de caminhos de Euler do plano.

Posteriormente, a etapa de correspondência entre os planos busca, através de um algoritmo de *string matching*, quais são as cadeias de caracteres (representantes dos caminhos de Euler obtidos dos planos) com maior número de literais iguais na mesma posição. A Figura 3 (b) ilustra esse estágio, agora para a rede completa - planos PU e PD, como visto em (a) -, da rede descrita na Equação 1.

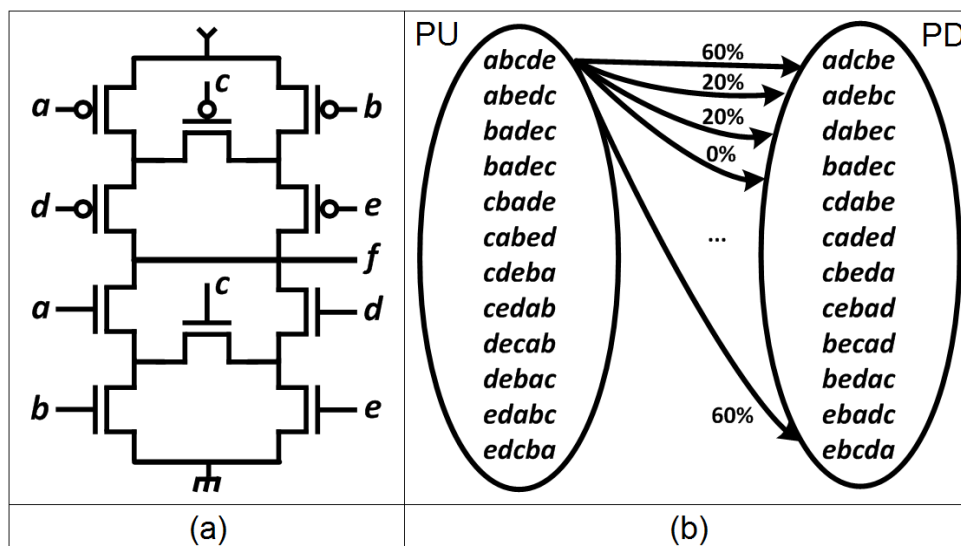


Figura 3. (a) Rede NSP completa. (b) Correspondência entre caminhos de Euler nos planos *pull-up* e *pull-down*.

Nota-se, na Figura 3 (b), que ao testar todas as possibilidades de casamentos entre os caminhos de Euler das redes PU e PD, a maior taxa de literais em mesma posição é de 60% (por exemplo, na correspondência “*abcde*” e “*adcbe*”). A partir desse resultado, posicionam-se os transistores nos respectivos

planos, adicionando quebras no polisilício de *gate* quando não houver correspondência (como a Figura 4 (a) ilustra). Após esse processo, o posicionamento da célula está completo e a célula está apta a receber o roteamento, próxima etapa do fluxo VLSI.

3. RESULTADOS E DISCUSSÃO

A Figura 4 (a) e a Figura 4 (b) são resultados da execução do método para a função descrita na Equação 1. Em (a), tem-se a versão do leiaute para a rede de transistores NSP – *match* de 60% –, enquanto que, em (b), tem-se a versão SP *match* de 100%. Percebe-se que, apesar do compartilhamento total de transistores na versão (b), a versão (a) apresenta ganho em área de 25% (causado pela menor quantidade de transistores, 10 da versão NSP contra 16 da versão SP). Observa-se que, apesar da solução mais otimizada contar com apenas 60% de alinhamento de *gates*, o ganho em área é expressivo, e, quanto maior for a taxa de compartilhamento, maior será essa otimização.

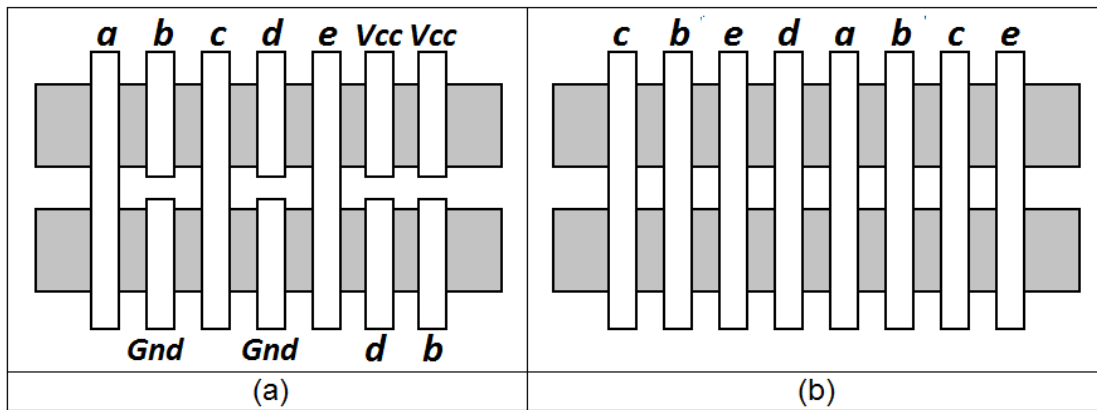


Figura 4. Resultados de execução do método proposto para a rede descrita na Equação 1. (a) Leiaute na versão NSP. (b) Leiaute na versão SP.

A Figura 5 (a) e (b) ilustram a rede de transistores NSP e o leiaute, respectivamente, que implementam a função descrita na Equação 2.

$$f = !((b \cdot !c \cdot !d) + (!a \cdot (!c \cdot d + !b))) \quad (2)$$

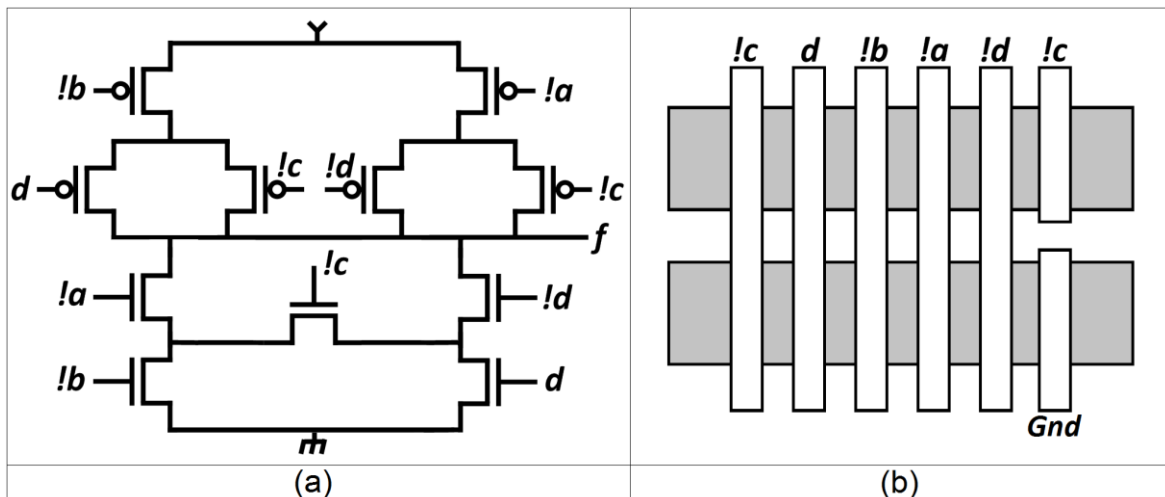


Figura 5. Resultados de execução do método proposto para a rede descrita na Equação 2. (a) Rede NSP. (b) Leiaute da rede.

Em (a), temos um caso específico de topologias NSP: a diferente quantidade de componentes nos planos (6 transistores no plano PU e 5 no plano PD). Nessa situação, é adicionado um polisilício ligado ao *ground* alinhado ao *gate* do transistor sem par, conforme ilustra a Figura 5 (b).

4. CONCLUSÕES E TRABALHOS FUTUROS

A pesquisa, ainda em estágio inicial, apresenta dois casos específicos de redes NSP: na primeira, o casamento parcial entre os transistores nos planos PU e PD e, na segunda, o diferente número de componentes nos planos. Pode-se perceber que, quanto ao posicionamento, redes NSP apresentam leiaute mais otimizado quanto a área, comparado a redes SP. No entanto, para conclusões mais precisas quanto a validade de redes NSP quanto a síntese física, faz-se necessário seguir investigando as outras etapas do fluxo VLSI, principalmente o roteamento *intra-cell*, que tende a ser mais complexo em redes NSP pelas suas características.

Como continuação desse trabalho, pretende-se aplicar o método aqui descrito, juntamente com o roteamento, para uma biblioteca de células (POSSANI, 2014), a qual engloba funções com topologias lógicas e elétricas diferenciadas, diversificando os casos de teste.

Finalmente, a partir desses estudos, busca-se prover uma ferramenta de apoio ao projeto (*Electronic Design Automation*, EDA) especializada para redes NSP, tornando, esta, uma alternativa viável para a concepção de CIs.

5. REFERÊNCIAS BIBLIOGRÁFICAS

- GUPTA, A., HAYES, P. CLIP: Integer-programming-based Optimal Layout Synthesis of 2D CMOS Cells. **ACM Transactions on Design Automation of Electronic Systems**, p.510-547, 2000
- HWANG, Y., HSIEH, C. A Fast Transistor-chaining Algorithm for CMOS Cell Layout. **IEEE Transactions on Computer-Aided Design**, p. 781-786, 1990.
- IIZUKA, T., IKEDA, M., ASADA, K. Exact Minimum-Width Transistor Placement for Dual and Non-dual CMOS Cells. **Proceedings of the 15th ACM Great Lakes Symposium on VLSI (GLSVLSI)**, p.74-77, 2005.
- KAGARIS, D., HANIOTAKIS, T. A Methodology for Transistor-Efficient Supergate Design. **IEEE Transactions on Very Large Scale Integration (VLSI) Systems**, p.488-492, 2007.
- POSSANI, V., TIMM, E., AGOSTINI, L. et al. A Graph-based Technique to Optimize Transistor Networks. **IEEE Symposium on Circuits and Systems (LASCAS)**, p.1-4, 2011.
- POSSANI, V. **Catalog of 53-NSP Networks**. Porto Alegre: Instituto de Informática, UFRGS. Acessado em 15 jul. 2014. Online. Disponível em: http://www.inf.ufrgs.br/logics/docman/53_NSP_Catalog.pdf
- SAKADANE, T., NAKAO, H., TERAJ, M. A New Hierarchical Algorithm for Transistor Placement in CMOS Macro Cell Design. **IEEE Custom Integrated Circuits Conference**, p.461-464, 1995.
- SERDAR, T., SECHEN, C. Automatic Datapath Tile Placement and Routing. **Conference on Design, Automation and Test in Europe**, p.552-559, 2001.
- UEHARA, T., VANCLEEMPUT, M. Optimal Layout of CMOS Functional Arrays. **IEEE Transactions on Computers**, p.305-312, 1981.