

## ESTRATÉGIAS DE POSICIONAMENTO PARA PORTAS LÓGICAS COMPLEXAS NÃO-SÉRIE-PARALELO NA TECNOLOGIA CMOS

**MAICON S. CARDOSO<sup>1</sup>; GUSTAVO H. SMANIOTTO<sup>1</sup>; MATHEUS T. MOREIRA<sup>2</sup>;**  
**LEOMAR S. DA ROSA JUNIOR<sup>1</sup>; FELIPE DE S. MARQUES<sup>1</sup>**

<sup>1</sup> Universidade Federal de Pelotas – {mscardoso, ghsmaniotto, leomarjr, felipem}@inf.ufpel.edu.br

<sup>2</sup> Pontifícia Universidade Católica do Rio Grande do Sul – matheus.moreira@acad.pucrs.br

### 1. INTRODUÇÃO

O projeto de circuitos digitais sob demanda vem sofrendo profundas transformações nos últimos anos. Com o crescente desenvolvimento de métodos e ferramentas de apoio ao projeto eletrônico, a necessidade do projetista manual perde força na concepção de tais dispositivos.

Nesse cenário, o projeto baseado em portas lógicas complexas (*static CMOS complex gates*, SCCGs) geradas sob demanda constitui um dos paradigmas mais promissores para a geração da rede de portas lógicas e de transistores, haja visto a recorrente otimização no número de componentes atingida por essa metodologia (REIS, 2011). Para exemplificar esse aspecto, considere a função Booleana descrita pela Equação 1. Através do modelo tradicional – baseado em bibliotecas de células –, obteve-se o circuito esquematizado na Figura 1 (a), contendo 26 transistores, enquanto (b) ilustra o circuito equivalente obtido através de uma abordagem utilizando SCCGs, composto por 18 transistores. Nota-se que a quantidade de transistores – dispositivos básicos utilizados na computação da função Booleana objetivo – teve uma drástica queda, o que tende a melhorar características físicas do circuito digital, tais como sua área, energia dissipada e atraso da propagação do sinal.

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

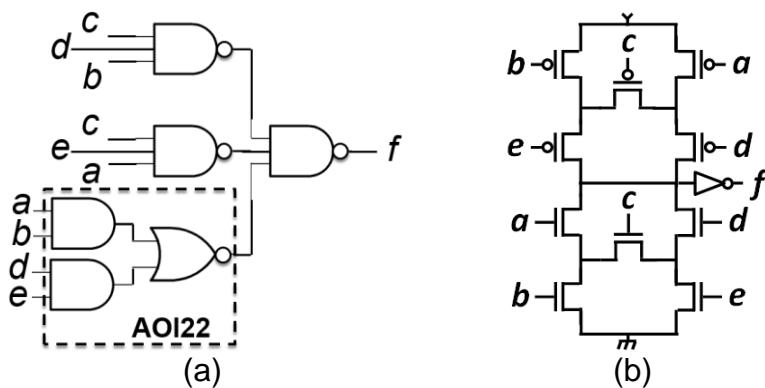


Figura 1. Implementações lógicas da função descrita em (1). (a) Através da metodologia *standard cell*. (b) Abordagem via SCCG.

Ainda que o projeto via SCCGs seja viável do ponto de vista lógico, como apresentado na Figura 1, existem diversas lacunas abertas considerando o projeto físico das células em si, especialmente se tratando de células não-série-paralelo. Tal topologia tem como característica apresentar componentes em

conexões do tipo *bridge*, diferentemente das tradicionais ligações série-paralelo. Esse aspecto, aliado a novos métodos de geração lógica, tal como (POSSANI, 2015), possibilita a aparição de soluções sem regularidade topológica, como, por exemplo, arranjos não-duais ou não-planares.

Um aspecto particular dessas soluções é observada nesse trabalho, definida aqui como *mismatching* intrínseco entre os planos lógicos da rede CMOS. Na seção seguinte, será apresentado o método de detecção de *mismatching*, bem como sua implicação na etapa de posicionamento do leiaute, estágio essencial da síntese física de células.

## 2. METODOLOGIA

Considerando a rede de transistores apresentada na Figura 1 (b) – e reproduzida na Figura 2 (a) –, consideraremos o posicionamento seguindo duas estratégias: a primeira – Figura 2 (b) – visa a minimização de quebras na área de difusão (c) (*Continuous Active Area*, CAA), enquanto a segunda (d) intenta minimizar o número de quebras nos polisilícios de *gates* (e) (*Continuous Polysilicon Gates*, CPG). Como ilustrado na Figura 2 para o arranjo anteriormente apresentado, a partir dessas estratégias de posicionamento é possível obter duas diferentes versões de leiaute para a SCCG (a).

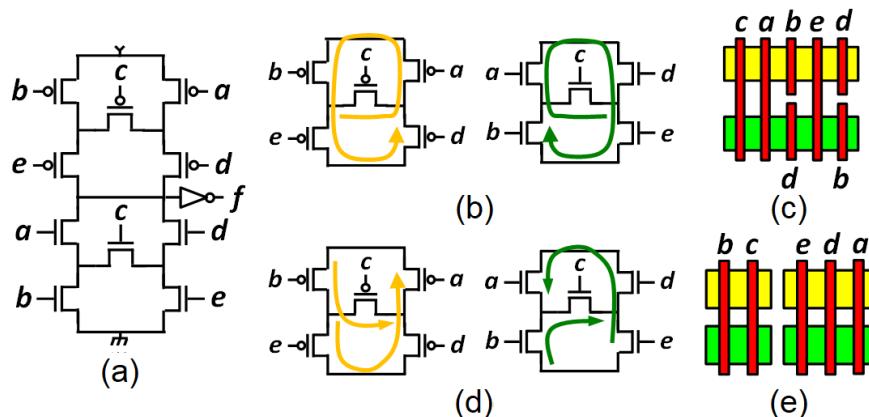


Figura 2. Estratégias de posicionamento para células com *mismatching* intrínseco. (a) Célula não-série-paralelo com *mismatching* intrínseco. (b) Posicionamento dos transistores seguindo a estratégia CAA. (c) Leiaute correspondente a estratégia CAA. (d) Posicionamento dos transistores seguindo a estratégia CPG. (e) Leiaute correspondente a estratégia CPG.

É importante notar que, independente da estratégia adotada para o posicionamento, é impossível alcançar uma solução sem quebras na área ativa e sem quebras nos polisilícios ao mesmo tempo. Essa característica é definida aqui como *mismatching* intrínseco e ocorre apenas em determinadas células não-série-paralelo. O Algoritmo 1 apresentado abaixo tem como objetivo detectar esse atributo a partir da busca de todos os caminhos de Euler possíveis considerando os planos *pull-up* (PU) e *pull-down* (PD) da célula.

A metodologia proposta consiste em identificar *mismatchings* intrínsecos em SCCGs não-série-paralelo (através do Algoritmo 1) para avaliar e comparar as estratégias de posicionamento quanto ao seu impacto no leiaute final (ou seja, levando em conta características físicas como área, potência e atraso).

### Algoritmo 1 Detecção de *Mismatching* Intrínseco

```

1: detectaMismatching ( PU, PD )
2:   caminhosPU ← CaminhosEuler ( PU )
3:   caminhosPD ← CaminhosEuler ( PD )
4:   mismatching ← verdadeiro
5:   para i=1 até pathsPU.size faça
6:     para j=1 até pathsPD.size faça
7:       se ( pathsPU[i] == pathsPD[j] ) então
8:         mismatching ← falso
9:       fim se
10:    fim para
11:   fim para
12:   retorna mismatching
13: fim

```

### 3. RESULTADOS E DISCUSSÃO

Para os experimentos propostos fora utilizado o catálogo de função (LOGICS, 2012), composto por 53 funções Booleanas. A partir dessa entrada, o método apresentado em (POSSANI, 2015) foi adotado para a concepção das redes de transistores, resultando em 53 redes não-série-paralelo. Desses, a partir do Algoritmo 1, identificaram-se 7 ocorrências de *mismatching* intrínseco. Os dados apresentados na Figura 3 são referentes a essas SCCGs, onde ilustram-se os ganhos e perdas da estratégia CPG em relação à CAA (ou seja, barras verticais azuis representam otimizações da estratégia CPG enquanto barras vermelhas correspondem a pioras nas características analisadas).

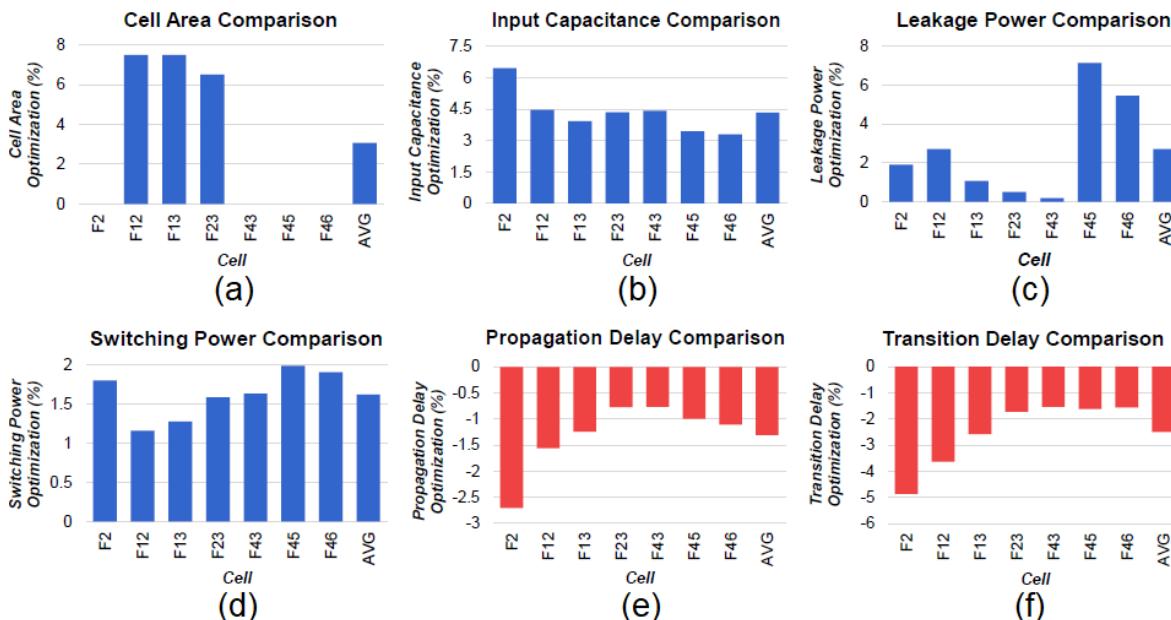


Figura 3. Comparativo entre as estratégias de posicionamento (ganhos e perdas da estratégia CPG em relação à CAA). (a) Área. (b) Capacitância de entrada. (c) *Leakage*. (d) Potência dinâmica. (e) Atraso de propagação. (f) Atraso de transição.

Como apresentado acima, a estratégia CPG apresentou ganhos em área (a) – média de 3,07% –, capacidade de entrada (b) – 4,33% –, *leakage* – 2,70% – e potência dinâmica (d) – 1,61%. A causa dessa otimização deve-se principalmente a uma diminuição significativa no comprimento de fio (*wirelength*) ligado as

entradas da célula, ou seja, há um menor congestionamento na área da interface da célula, haja visto que não há necessidade de se conectar polisilícios.

Por outro lado, a estratégia CPG apresenta perdas em relação à CAA quanto ao atraso de propagação (e) – 1,31% – e ao atraso de transição (f) – 2,50%. Isso é causado pela não ocorrência de quebras na difusão, o que levaria à necessidade de roteamento dessa área, atrasando o sinal devido a resistências e capacidades inerentes a colocação de contatos para o roteamento, principalmente.

#### 4. CONCLUSÕES

Esse trabalho tem como objetivos identificar uma característica topológica particular a SCCGs não-série-paralelo – o *mismatching* intrínseco entre planos –, haja visto que o desenvolvimento via portas complexas torna-se cada vez mais uma abordagem promissora para o desenvolvimento de circuitos e sistemas digitais. Como vista, tal característica topológica tem impacto direto na síntese física da célula, fazendo-se necessário identificá-la a priori para que se possa definir a estratégia de posicionamento que se adeque aos requisitos de projeto.

Nesse cenário, um algoritmo de detecção de *mismatching* intrínseco foi proposto, bem como duas estratégias para o posicionamento: a primeira, CAA, visa minimizar o número de quebras na área ativa da célula, enquanto a segunda, CPG, tem como objetivo a diminuição no número de quebras nos polisilícios de gate da célula.

Definidas as estratégias, os experimentos foram realizados implementando-se as células que apresentaram *mismatching* intrínseco de um catálogo de funções comumente utilizado na literatura. Como resultados, as células posicionadas seguindo a política CPG apresentaram ganhos em área, capacidade de entrada, *leakage* e potência dinâmica, enquanto a estratégia de posicionamento CAA levou as células a um ganho em atraso de propagação e transição. Tais resultados são úteis no sentido do projetista (ou da ferramenta de projeto utilizada) determinar qual estratégia seguir a partir de um requisito de entrada: por exemplo, para projetos com restrições temporais, torna-se mais adequado o posicionamento via CAA, haja visto os ganhos em tempo de propagação e transição alcançados se comparado a estratégia CPG.

#### 5. REFERÊNCIAS BIBLIOGRÁFICAS

REIS, R. Design Automation of Transistor Networks, a new Challenge. **IEEE International Symposium on Circuits and Systems (ISCAS)**, Rio de Janeiro, p. 2485-2488, 2011.

POSSANI, V., CALLEGARO, V., REIS A., RIBAS, R., MARQUES, F., DA ROSA, L. Graph-based transistor network generation method for supergate design. **IEEE Transactions on Very Large Scale Integration (VLSI) Systems**, vol. 24, no. 2, p. 488-492, 2015.

LOGICS (UFRGS). **Catalog of Handmade Optimum Switch Networks**. Outubro de 2012. Acessado em 11 de out. de 2017. Disponível em: [http://www.inf.ufrgs.br/logics/docman/53\\_NSP\\_Catalog.pdf](http://www.inf.ufrgs.br/logics/docman/53_NSP_Catalog.pdf).