

UMA METODOLOGIA PARA GERAÇÃO DE REDES DE TRANSISTORES FINFET DOUBLE GATE

RENATO SOUZA DE SOUZA; FELIPE DE SOUZA MARQUES;
LEOMAR SOARES DA ROSA JUNIOR

*Universidade Federal de Pelotas – Grupo de Arquiteturas e Circuitos Integrados
{rsdsouza; felipem; leomarjr}@inf.ufpel.edu.br}*

1. INTRODUÇÃO

A tecnologia FinFET é amplamente reconhecida como a principal alternativa para resolver os problemas ocasionados pela redução do canal do transistor. Esta tecnologia consiste em uma nova abordagem para a construção de um transistor em três dimensões. Assim, o *gate* do transistor mantém contato com três faces do canal, proporcionando um controle maior do fluxo dos elétrons no canal. A estrutura padrão de um transistor FinFET é conhecida como *Single Gate* (SG) FinFET. Contudo, algumas variações desta estrutura foram propostas. Uma destas variações estruturais é conhecida como *Independent Gate* (IG) FinFET, onde um transistor (IG) FinFET pode ser implementado com dois *gates* (WANG, 2010) (ROSTAMI e MOHANRAM, 2011) (MISHRA, MUTTREJA e JHA, 2011). Consequentemente, explorar os agrupamentos de um transistor (IG) FinFET *double gate* acaba por tornar-se um meio interessante para reduzir o número de transistores em um circuito.

Neste contexto, este trabalho propõem um método alternativo para a geração de redes de transistores dedicada a dispositivos (IG) FinFET *double gate*. O método baseia-se em uma metodologia, especialmente desenvolvida para síntese lógica, chamada de Composição Funcional (MARTINS, M., RIBAS, R., AND REIS, A, 2012). Utilizando-a, é possível controlar o número de transistores associados em paralelo ou em série. Sendo assim, simplifica a procura por padrões de arranjos promissores para explorar o potencial dos dispositivos (IG) FinFET *double gate*.

2. METODOLOGIA

A composição Funcional utiliza uma abordagem bottom-up para gerar funções Booleanas, diferente de outras abordagens. No qual, utiliza uma analogia de baldes que são preenchidos com funções, classificando-os pelo número de literais compostos na função. Neste caso, o primeiro balde é composto por funções que contém apenas um literal. Após, as funções do balde 1 são combinadas utilizando operações lógicas de AND/OR para criar o balde de 2 literais. De forma similar, a combinação das funções nos baldes de 1 e 2 literais preenchem o balde de 3 literais. O balde de 4 literais é composto por operações entre os baldes 1 e 3 e entre pares de elementos do balde 2. Para compor o balde 5, utilizam-se combinações entre os baldes 1 e 4 e os baldes 2 e 3. O conceito da criação do balde de n-literais é apresentado na Equação (1).

$$B_n = \bigcup_{i=1}^{\frac{n}{2}} \left((B_i * B_{n-1}) \cup (B_i + B_{n-1}) \right) \mid n \geq 2 \quad (1)$$

O método proposto utiliza basicamente a mesma ideia de fatoração da Composição Funcional para gerar uma rede de transistores do tipo (IG) FinFET *double gate*. O método recebe uma função como entrada e vai fatorando esta

função preenchendo os baldes até encontrar uma solução. Porém, existe uma diferença no preenchimento dos baldes. Neste caso, o primeiro balde ao invés de ser composto por sub-funções que contém apenas um literal, ele é constituído por todas as sub-funções possíveis com 2 literais, que resultam em arranjos (IG) FinFET *double gate* série e paralelo. Considere a função f apresentada pela Equação (2) como entrada para o método proposto. Essa função está descrita em uma forma de SOP (Soma de Produtos).

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

Primeiramente é realizado um pré-processamento para verificar a quantidade de variáveis presentes na função de entrada e, também, para analisar se todas as variáveis encontram-se diretas ou negadas. Esta etapa é realizada para coletar as informações para construir o balde 1. Portanto, analisando a função presente na Equação (2) é possível notar que ela é composta por quatro variáveis e todas as variáveis ocorrem em ambas às polaridades. Deste modo, o balde 1 será composto por todos os pares funcional/estrutural que representam as variáveis, e também pelos pares que representam as combinações dois a dois, *AND* e *OR*, entre estas variáveis.

A cada geração de um par, antes dele ser adicionado a um balde, é verificado se este par não é a função alvo. Caso a sub-função seja a função alvo, o método proposto retorna como resultado a sub-função. Caso contrário, o método proposto continua a operação de busca. Além disso, a cada geração de um par, este par é adicionado em uma tabela de funções permitidas. Entretanto, deve-se verificar se um par com a mesma funcionalidade encontra-se na tabela. Caso este par já esteja presente na tabela, o par não é adicionado no balde. Caso contrário, o par é adicionado na tabela e também no balde referente ao seu custo. A Figura 1 ilustra a representação do balde 1 para este exemplo.

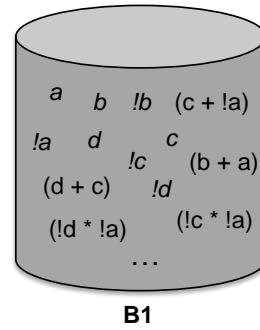


Figura 1 - Representação do balde 1, considerando a função apresentada pela Equação (2).

Após construído o balde 1, o método proposto segue o processo de busca gerando o próximo balde. O balde 2 é composto pelas sub-funções que apresentam custo dois, isto é, as sub-funções que contém apenas dois transistores (IG) FinFET *double gate*. Então, para gerar o balde 2, é usado os pares presente no balde 1. Assim, através das associações entre os próprios pares do balde 1, utilizando as operações *AND* e *OR*, são criados os pares que irão compor o balde 2. A Figura 2 ilustra este processo.

Caso, a função alvo não tenha sido alcançada, o método proposto continua o processo de geração dos baldes. O próximo balde a ser construído é o balde 3, que contém as sub-funções que apresentam custo três. Assim, o balde 3 é preenchido através das associações, *AND* e *OR*, entre os pares dos baldes 1 e 2.

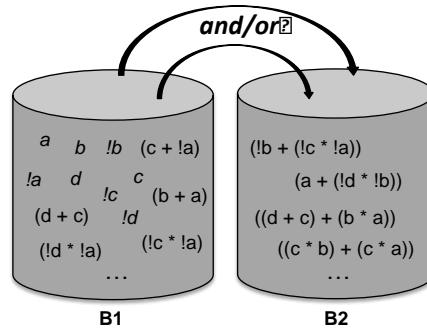


Figura 2 - Geração do balde 2

É importante lembrar que a cada geração de uma sub-função verifica-se se esta sub-função não é a função alvo. Neste caso específico, a função alvo ainda não foi encontrada. Logo, deve-se gerar o balde 4. A criação do balde 4 deve ser composta por pares que apresentam custo quatro. Assim, utiliza-se os pares do balde 1 com o balde 3. Caso não encontrado a função alvo, utiliza-se ainda, a associação entre os próprios pares contidos no balde 2, para completar o balde 4. A Figura 3 ilustra este processo. Neste caso acaba-se encontrando a função alvo.

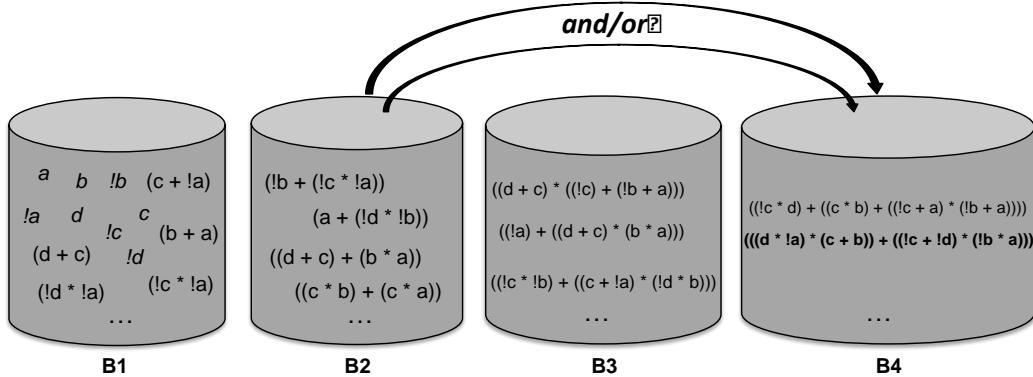


Figura 3 - Construção do balde 4, através da associação entre os pares presentes no balde 2.

Por fim, o método proposto, retorna a função alvo encontrada. Note que a função alvo é composta por quatro agrupamentos dois a dois. Deste modo ela pode ser construída por quatro transistores (IG) FinFET *double gate*.

3. RESULTADOS E DISCUSSÃO

Para avaliar o método proposto neste trabalho, foram realizados experimentos onde os resultados obtidos foram comparados com os métodos dedicados a dispositivos *double gate* propostos por Possani (POSSANI, 2015). O primeiro *benchmark* utilizado foi o conjunto *P-class* de 4 entradas, o qual é composta por 3.982 funções Booleanas. Também foi utilizado um subconjunto da classe NPN (negação-permutação-negação) de 5 entradas. Esta classe é composta por 616.125 funções. Porém, foram utilizadas apenas 413 funções dessa classe. A Tabela I apresenta os experimentos realizados. Os resultados apresentados pela Tabela I mostram que o método proposto apresentou uma redução no número de transistores, comparados aos métodos de grafos e o de defatoração, de 9,65% e 5,98%, respectivamente. No segundo experimento,

utilizando um subconjunto de funções da classe NPN de 5 entradas, o método proposto mostrou uma redução de 8,31% comparado ao método de grafos e 8,44% quando comparado ao método de defatoração.

Tabela 1 - Total de número de transistores (IG) FinFET considerando o conjunto de funções da *P-class* 4 entradas e subconjunto de 413 funções da classe NPN 5 entradas.

		Métodos		
	Benchmark	Baseado em Grafo (POSSANI, V. 2015)	Defatoração (POSSANI, V. 2015)	Método Proposto
Número total de dispositivos (IG) FinFET	P-Class 4 entradas	22.647	21.764	20.462
	Subconjunto de 413 funções da classe NPN-5	2.106	2.109	1.931

4. CONCLUSÕES

Este trabalho apresentou um método automático de geração de redes de transistores dedicado a dispositivos (IG) FinFET *double gate*. A motivação para a realização deste trabalho foi a mudança de paradigma relacionada à concepção da rede de transistores apresentada por Possani (POSSANI, 2015), onde ficou demonstrado que os métodos convencionais de geração de redes não apresentam bons resultados quando considerados os dispositivos *double gate*. A solução apresentada neste trabalho é baseada na metodologia de Composição Funcional. Através dos resultados obtidos, conclui-se que o método proposto pode ser considerado uma forma alternativa para gerar redes de transistores dedicadas a dispositivos (IG) FinFET *double gate*. Como trabalhos futuros pretende-se realizar avaliações elétricas para verificar o desempenho e o consumo de energia das redes geradas pelo método proposto.

5. REFERÊNCIAS BIBLIOGRÁFICAS

MARTINS, M., RIBAS, R., AND REIS, A. **Functional composition**: A new paradigm for performing logic synthesis. 13th International Symposium on Quality Electronic Design (ISQED). [S.I.]: [s.n.]. 2012. p. 236–242.

MISHRA, P.; MUTTREJA, A.; JHA, N. FinFET Circuit Design. **Springer Science+Business Media, LLC**, p. 23-54, 2011.

POSSANI, V. **Exploring Independent Gates in FinFET-Based Transistor Network Generation** [obj]. Dissertação do Programa de Pós-Graduação em Computação, UFPel. Pelotas: [s.n.]. 2015. p. 78.

ROSTAMI, M.; MOHANRAM, K. Dual-V_{th} Independent-Gate FinFETs for Low Power Logic Circuits. **IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS**, v. 30, n. 3, p. 337- 349, Mar 2011.

WANG, M. Independent-Gate FinFET Circuit Design Methodology. **IAENG International Journal of Computer Science**, v. 37, n. 1, p. 50, Feb 2010.