

GERAÇÃO AUTOMÁTICA DE REDES DE TRANSISTORES DEDICADO A DISPOSITIVOS FINFET UTILIZANDO COMPOSIÇÃO FUNCIONAL

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 evolução no desenvolvimento dos circuitos VLSI basicamente ocorre devido ao avanço tecnológico relacionado com a miniaturização da tecnologia CMOS. No entanto, esta continua miniaturização na tecnologia CMOS, especialmente para valores abaixo dos 22 nanômetros, vêm se tornando um processo complexo. Isto, basicamente, se deve aos limites físicos das estruturas e dos processos utilizados na fabricação. Um dos problemas apontados, relacionados aos limites físicos das estruturas criadas, é o aumento do consumo de energia do circuito, mesmo esse estando em estado de equilíbrio. Este consumo é ocasionado pela redução do canal do transistor, conhecido por corrente de fuga. Diversas alternativas foram sugeridas com o intuito de resolver este problema.

Neste contexto surgiu o transistor FinFET, o qual consiste em uma nova estrutura, um transistor em três dimensões. No transistor FinFET, o gate do transistor mantém contato com três faces do canal, proporcionando um controle mais efetivo do fluxo de elétrons no canal. Estudos mostram que o transistor FinFET apresenta vantagens significativas em termos de desempenho e eficiência energética quando comparado ao transistor MOSFET(SKOTNICKI, HUTCHBY, et al., 2005)(NOWAK, RAINY, et al., 2002). Além disso, o processo de fabricação do transistor FinFET é compatível com o processo de fabricação do transistor MOSFET, característica importante que o tornou a solução adotada pela indústria de microeletrônica para a fabricação das novas gerações de circuitos integrados.

A estrutura padrão de um transistor FinFET é conhecida como transistor Single Gate (SG) FinFET, ilustrado pela Fig. 1(a).Porém é possível encontrar na literatura trabalhos que apresentam algumas variações desta estrutura de transistor, conhecidas como Independent Gate (IG) FinFET ou Double Gate (DG) FinFET, representado pela Fig. 1(b)(WANG, 2010)(ROSTAMI e MOHANRAM, 2011)(MISHRA, MUTTREJA e JHA, 2011).

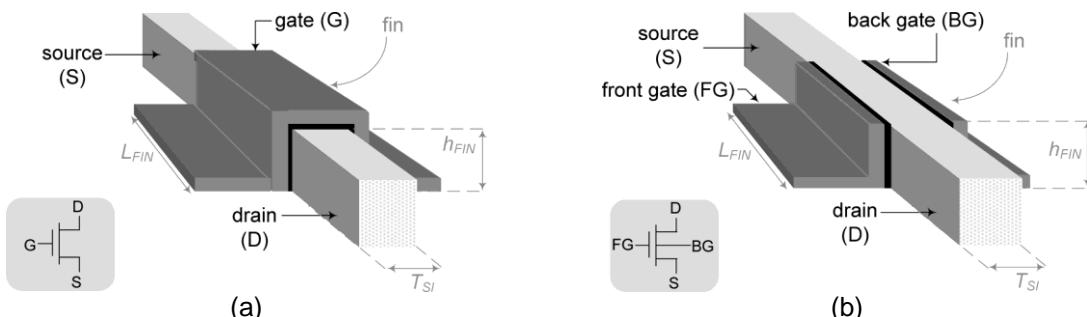


Figura 1 - Estrutura (SG) FinFET em (a) e estrutura (IG) FinFET em (b).

Estudos mostram que, através da estrutura (IG) FinFET, é possível utilizar cada gate do transistor com diferentes propósitos de acordo com as restrições de projeto no desenvolvimento de um circuito integrado. Assim, pode-se obter reduções em área, potência dinâmica e capacidade do circuito.

Um dos propósitos possíveis de se utilizar a estrutura (IG) FinFET, é conectar cada um dos gates em um sinal diferente, podendo assim, implementar tanto um arranjo de dois sinais em paralelo ($a+b$) quanto um arranjo em série ($a \cdot b$)(MUTTREJA, AGARWAL e JHA, 2007)(WANG, 2010). Através da união de transistores em série ou em paralelo é possível minimizar o número total de transistores de uma determinada função lógica, obtendo uma redução em área (CHIANG, KIM, et al., 2006)(CHIANG, KIM, et al., 2005)(WANG, 2010)(ROSTAMI e MOHANRAM, 2011)(POSSANI, 2015). Portanto explorar os agrupamentos em um transistor (IG) FinFET se tornar um meio interessante para reduzir o número de transistores em um circuito.

Neste sentido, este trabalho propõe um método dedicado para geração automática de redes de transistores do tipo (IG) FinFET *double gate*. O método proposto é baseado em uma técnica de fatoração, conhecida como Composição Funcional (*Functional Composition - FC*) (MARTINS, DA ROSA JUNIOR, et al., 2010). Através do uso da técnica de Composição Funcional é possível ter o controle de critérios específicos durante a otimização. Um destes critérios é o possível controle do número de transistores associados em paralelo ou em série, tornando assim, um bom indício para realizar os agrupamentos dos transistores (IG) FinFET.

2. METODOLOGIA

2.1 COMPOSIÇÃO FUNCIONAL

A composição Funcional (FC) utiliza uma abordagem *bottom-up* para gerar funções Booleanas, diferente de outras abordagens. A FC baseia-se nos seguintes princípios: (1) a representação da função lógica por um par funcional/estrutural; (2) inicia com um conjunto de funções com custo conhecido; (3) utiliza funções mais simples para produzir funções mais complexas; (4) um conjunto de funções “permitidas” é utilizado para reduzir o consumo de tempo e memória. A FC inicia com um conjunto de sub-funções de um único literal, extraídos da função de entrada do algoritmo (a ser otimizada) utilizando o conceito de co-fatores (MARTINS, DA ROSA JUNIOR, et al., 2010). O método 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-literals é apresentado na Equação 1. A Fig. 2 ilustra o processo para criar todos os baldes de até 5 literais.

A cada preenchimento de um balde, o algoritmo verifica se a sub-função gerada é uma possível solução. Caso a sub-função seja uma possível solução, o algoritmo encerra o processo. Caso contrário, continua preenchendo os baldes até encontrar uma possível solução ou até que um limite de baldes preenchidos seja alcançado.

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

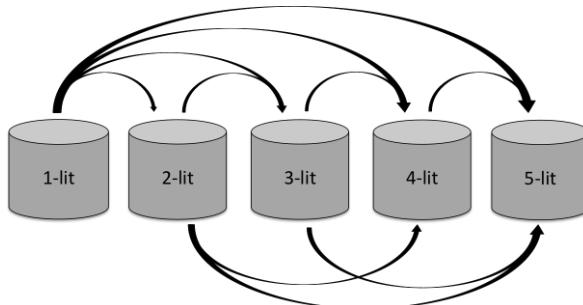


Figura 2 – Geração de sub-funções até o 5º balde.

2.2 MÉTODO PROPOSTO

O método proposto neste trabalho utiliza basicamente a mesma ideia de fatoração da Composição Funcional para gerar uma rede de transistores do tipo (IG) FinFET. O método recebe uma função como entrada e vai fatorando esta função preenchendo os baldes até encontrar uma solução. Porém, existe uma pequena 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 série e paralelo. O balde 1 (B1) presente na Fig. 3 exemplifica este procedimento. A Fig. 3 ilustra os 4 baldes que foram necessários para encontrar uma solução para a função apresentada em Exp. 1. Cada balde mostra uma pequena parte do conjunto de sub-funções que foram criadas para gerar na solução.

$$(!a * c * d) + (!a * b * d) + (a * !b * !c) + (a * !b * !d) \quad \text{Exp. 1}$$

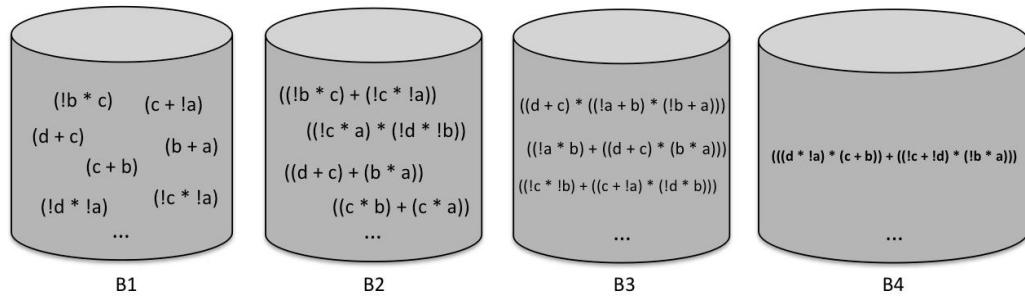


Figura 3 – Baldes resultantes para o exemplo apresentado por Exp. 1.

3. RESULTADOS E DISCUSSÃO

Para avaliar o método proposto foi utilizado um conjunto de funções Booleanas conhecidos pela comunidade científica, o qual é composto por todas as funções *p-class* de quatro entradas. Este conjunto de funções contém 3982 funções Booleanas. Esse conjunto foi aplicado ao método proposto e também a outro método de geração de redes de transistores do tipo (IG) FinFET (POSSANI, 2015). A Tabela I apresenta os resultados obtidos através de cada método para o experimento realizado. Os resultados apresentados pela Tabela I mostram uma redução de 1.302 no número total de dispositivos (IG) FinFET, quando utilizado o

método proposto em comparação ao método de defatoração proposto por Possani (POSSANI, 2015). Contabilizando uma redução em torno de 6%.

Tabela I. Número total de dispositivos (IG) FinFET para o conjunto da *p*-class 4 entradas.

Benchmark	P-Class 4 entradas	
Métodos	(POSSANI, 2015)	Método Proposto
Número total de dispositivos (IG) FinFET	21.764	20.462

4. CONCLUSÕES

Este trabalho apresentou um novo método de geração de redes de transistores, dedicado a dispositivos (IG) FinFET. O método proposto é baseado no paradigma de fatoração, a Composição Funcional. Os resultados obtidos neste primeiro momento mostram que o método é promissor para as novas tecnologia de transistores como os dispositivos (IG) FinFETs. A característica *bottom-up* do algoritmo mostrou-se bastante eficiente.

Como trabalhos futuros pretende-se realizar novos experimentos utilizando outros *benchmarks* e também utilizar algumas heurísticas para evitar que se excedam valores aceitáveis de tempo de processamento e uso de memória para funções com um número grande literais.

5. REFERÊNCIAS BIBLIOGRÁFICAS

- CHIANG, M. et al. Novel High-Density Low-Power Logic Circuit Techniques Using DG Devices. **IEEE TRANSACTIONS ON ELECTRON DEVICES**, v. 52, n. 10, p. 2339-2342, Oct 2005.
- CHIANG, M. et al. High-Density Reduced-Stack Logic Circuit Techniques Using Independent-Gate Controlled Double-Gate Devices. **IEEE TRANSACTIONS ON ELECTRON DEVICES**, v. 53, n. 9, p. 2370- 2377, Sep 2006.
- MARTINS, M. et al. **Boolean Factoring with Multi-Objective Goals**. IEEE Int. Conf. on Computer Design. Amsterdam: [s.n.]. 2010. p. 229-234.
- MISHRA, P.; MUTTREJA, A.; JHA, N. FinFET Circuit Design. **Springer Science+Business Media, LLC**, p. 23-54, 2011.
- MUTTREJA, A.; AGARWAL, N.; JHA, N. **CMOS Logic Design with Independent-gate FinFETs**. 25th International Conference on Computer Design. Lake Tahoe, CA: [s.n.]. 2007. p. 560 - 567.
- NOWAK, E. et al. **A functional FinFET-DGCMOS SRAM cell**. Tech. Dig. IEDM. [S.I.]: [s.n.]. 2002. p. 411–414.
- 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.
- SKOTNICKI, T. et al. The end of CMOS scaling. **IEEE Circuits Devices Mag.**, v. 21, p. 16-26, Jan-Feb 2005.
- WANG, M. Independent-Gate FinFET Circuit Design Methodology. **IAENG International Journal of Computer Science**, v. 37, n. 1, p. 50, Feb 2010.