

Melhorando o Desempenho de uma Ferramenta de Síntese de Circuitos Integrados em MATLAB®

Augusto Vargas, Márcia C. Cera, Lucas C. Severo

¹ Universidade Federal do Pampa (UNIPAMPA) - Campus Alegrete
Av. Tiarajú, 810 - CEP 97546-550 – Alegrete – RS – Brasil

gutorsscl@gmail.com, {marciacera, lucassevero}@unipampa.edu.br

Resumo. *Este artigo aborda a paralelização de uma ferramenta de síntese automática de circuitos integrados analógicos em MATLAB®. Ela usa a meta-heurística Simulated Annealing para encontrar um melhor dimensionamento dos transistores nos circuitos. Entretanto, testar diversas combinações demanda um grande tempo de processamento. Nossa implementação paralela manteve a qualidade das soluções, reduzindo o tempo de execução em até 63%.*

1. Introdução

Na área de Microeletrônica, a maior parte dos circuitos integrados são digitais e a menor parte são analógicos. Embora os circuitos digitais sejam predominantes, os analógicos sempre serão necessários, seja para geração de sinais para os circuitos digitais ou fazer a interface entre o circuito e a aplicação. Os circuitos integrados são compostos por transistores que devem ser dimensionados. O dimensionamento dos transistores de circuitos integrado analógicos é mais complexo do que os de circuitos digitais [Severo 2012]. Nestes últimos, todos os transistores possuem tamanhos pré-definidos, já em circuitos analógicos cada transistor deve ser dimensionado individualmente.

Para facilitar o processo de dimensionamento dos transistores em circuitos analógicos, Severo [Severo 2012] desenvolveu a ferramenta UCAF. Esta ferramenta foi implementada em MATLAB® e ela permite a automação do projeto de blocos analógicos integrados básicos voltada à exploração do espaço de projeto circuito através de meta-heurísticas. Assim, é possível encontrar soluções otimizadas que satisfaçam as especificações impostas ao circuito. Contudo, a ferramenta demanda um grande tempo de processamento para analisar várias combinações no espaço de projeto, o que agrava-se para configurações mais complexas.

Este artigo objetiva paralelizar a execução da ferramenta a fim de reduzir seu tempo de processamento e analisar se a paralelização influencia na qualidade das soluções encontradas. A paralelização foi implementada utilizando funcionalidades da *Parallel Computing Toolbox™* - PCT do próprio MATLAB®. O restante do texto está organizado da seguinte forma: a Seção 2 apresenta maiores detalhes sobre a ferramenta UCAF. A Seção 3 explica o PCT e suas funcionalidades que foram utilizadas neste trabalho. Os resultados obtidos em nossos experimentos são apresentados na Seção 4. Por fim, a Seção 5 conclui este artigo.

2. Ferramenta UCAF

A ferramenta de síntese automática de circuitos integrados analógicos (UCAF) permite a exploração do espaço de projeto de dimensionamento dos transistores de um circuito.

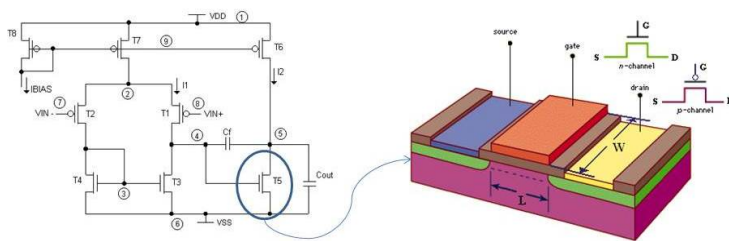


Figura 1. Exemplo de um circuito integrado analógico, em tecnologia CMOS, com 8 transístores e a ilustração dos parâmetros L e W de um dispositivo físico.

Como ilustra a Figura 1, um circuito analógico é composto por alguns transístores CMOS (Metal Óxido Semicondutor Complementar), para cada transistor é necessário obter as dimensões físicas do canal dadas pela largura W e o comprimento L .

Com a exploração do projeto, é possível buscar valores de W e L de um circuito de forma que suas especificações atendam aos valores desejados em um projeto e, ao mesmo tempo, sejam obtidas soluções otimizadas em termos de potência dissipada pelo circuito, fator muito importante em um projeto. A exploração do espaço de projeto é feita utilizando a meta-heurística Simulated Annealing (SA) [Severo et al. 2012]. O fluxo de projeto da ferramenta UCAF é mostrado na Figura 2a. São recebidos como entradas as configurações do SA, os requisitos de projeto e os dados da tecnologia de fabricação. Após isso, inicia-se a execução do SA que cria uma solução inicial aleatória. Como solução entende-se um conjunto de valores para as variáveis do circuito. Para cada solução gerada, é feita uma avaliação através de uma função custo. Após isso, novas soluções são geradas e o processo é repetido até que seja encontrada uma solução otimizada que atenda aos requisitos de projeto.

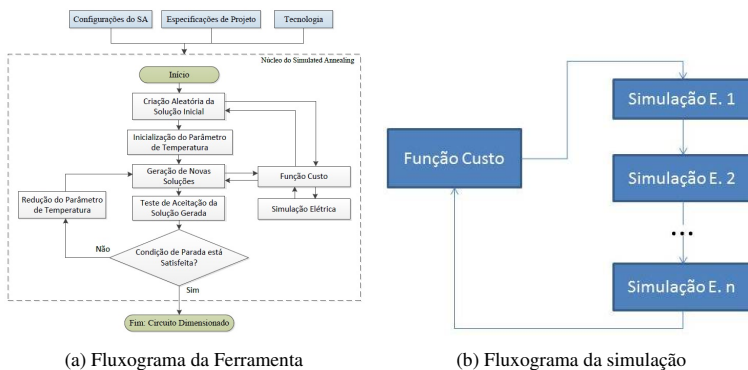


Figura 2. Fluxogramas representando: (a) o funcionamento da ferramenta e (b) a execução das simulações elétricas para o cálculo da função custo e avaliação de uma solução.

Para o cálculo da função custo, como a mesma depende das especificações de projeto, é necessário estimar o valor de cada uma das especificações de um circuito. Na ferramenta UCAF esta estimativa é feita através de simuladores elétricos comerciais do tipo *SPICE* (Simulador Elétrico com Ênfase em Circuitos Integrados), utilizando os dados de uma tecnologia de fabricação. Como para cada solução gerada é necessário executar o simulador elétrico várias vezes, conforme mostra a Figura 2b. Cada simulação é independente uma da outra e elas tendem a consumir um grande tempo computacional.

3. Paralelização da ferramenta UCAF

Como a ferramenta UCAF foi desenvolvida em MATLAB®, optamos por utilizar o *Parallel Computing Toolbox™* - PCT. Ele disponibiliza um conjunto de funções que ao serem incluídas em códigos MATLAB® permite executá-los em clusters, computadores multicore ou GPU's [The Mathworks Inc 2013]. Neste trabalho utilizamos o PCT em uma máquina com arquitetura multicore.

Neste trabalho utilizamos a primitiva `parfor` que permitiu executar as iterações de laços do tipo `for` em paralelo, similarmente a outras ferramentas de programação paralela, como o OpenMP [Barbara Chapman and Pass 2007]. Após a análise do código da ferramenta, decidiu-se paralelizar as execuções das simulações elétricas na função custo, já que estas tendem a consumir um grande tempo computacional. As simulações são repetidas através de um laço `for`, possibilitando a aplicação do `parfor`.

4. Resultados

A ferramenta foi executada em um computador equipado com um processador Intel® Core™ 2 Quad com 4 núcleos físicos. O sistema operacional em uso foi o Microsoft® Windows® 7 Profissional 64 bits e a versão do MATLAB® foi a R2011a (7.12.0). O MATLAB® não permite que o número de trabalhadores exceda o número de núcleos disponíveis, por isso foram utilizados no máximo 4 trabalhadores em nossos testes.

Inicialmente, configuramos a ferramenta UCAF para sintetizar um circuito Amplificador Diferencial com 2 variáveis (W e L), realizando 2 simulações/iteração e utilizando 2 e 4 trabalhadores. No MATLAB®, os trabalhadores representam unidades de execução, ou seja, a quantidade de entidades que computam paralelamente. A Tabela 1 mostra os resultados obtidos para o Amplificador Diferencial. Nela percebemos que o maior *speedup* ocorreu nas execuções com 2 trabalhadores. Isto justifica-se porque nessa configuração ocorrem 2 simulações por iteração e não há trabalho suficiente para melhorar o desempenho da ferramenta quando utiliza-se mais de 2 trabalhadores.

Para testar nossa proposta de paralelização com um circuito mais complexo, configuramos a ferramenta UCAF para sintetizar um circuito Amplificador Operacional de

Tabela 1. Tempos de execução (em minutos) e *speedups* para um circuito Amplificador Diferencial com 2 variáveis, 2 simulações/iteração, utilizando 2 e 4 trabalhadores.

Número de Trabalhadores	Tempo (minutos)	Speedup
Sequencial	66,02	–
2	37,02	1,77
4	39,05	1,69

Tabela 2. Tempo de execução (em minutos) e *speedup* para um circuito OTA do tipo Miller com 11 variáveis, 4 simulações/iteração e utilizando 2, 3 e 4 trabalhadores.

Número de Trabalhadores	Tempo (minutos)	<i>Speedup</i>
Sequencial	1194,07	–
2	616,08	1,93
3	539,00	2,21
4	447,03	2,67

Transcondutância (OTA) de dois estágios do tipo Miller, com 11 variáveis (5 pares de W e L e uma corrente de polarização), realizando 4 simulações/iteração e utilizando 2, 3 e 4 trabalhadores. Na Tabela 2 observa-se que foi possível reduzir o tempo de processamento pela metade ao executar com 2 trabalhadores, obtendo um *speedup* próximo ao ideal. Embora os *speedups* com 3 e 4 trabalhadores tenham ficado abaixo do ideal, os resultados obtidos com a paralelização foram significativos. Isto porque, a paralelização permite que se tenha um circuito sintetizado, que respeita as especificações de projeto, em aproximadamente 13 horas de antecedência (em torno de 20 horas de processamento sequencial \times em torno de 7 h e 30 min utilizando-se 4 trabalhadores).

5. Conclusão

Neste trabalho buscou-se obter a redução do tempo de execução da ferramenta UCAF e a manutenção da qualidade das soluções encontradas. A redução no tempo de execução obtida foi significativa, porém nosso *speedup* ficou sempre abaixo do ideal. A melhora no desempenho da ferramenta UCAF, obtida pela paralelização, permite que se chegue ao projeto de circuitos em um tempo menor. Nos nossos testes, reduzimos o tempo de processamento de um circuito Amplificador Diferencial com 2 variáveis e 2 simulações/iteração em até 44%, aproximadamente; enquanto que para um Amplificador Operacional de Transcondutância de dois estágios do tipo Miller com 11 variáveis e 4 simulações/iteração, a redução foi de até 63%, aproximadamente.

Referências

- Barbara Chapman, G. J. and Pass, R. V. D. (2007). *Using OpenMP: portable shared memory parallel programming*. The MIT press.
- Severo, L. C. (2012). Uma ferramenta para o dimensionamento automático de circuitos integrados analógicos considerando análise de produtividade. Master's thesis, Universidade Federal do Pampa - UNIPAMPA, Alegrete - RS.
- Severo, L. C., Girardi, A. G., de Oliveira, A. B., Kepler, F. N., and Cera, M. C. (2012). Simulated annealing to improve analog integrated circuit design: Trade-offs and implementation issues. In Tsuzuki, M., editor, *Simulated Annealing / Book 1*, pages 261 – 284. InTech, Rijeka - Croácia.
- The Mathworks Inc (2013). *Parallel Computing Toolbox*. <http://www.mathworks.com/products/datasheets/pdf/parallel-computing-toolbox.pdf>. Último acesso em janeiro de 2013.