

Redução de Consumo pela Otimização de Componentes

Ricardo Reis

PGMicro/PPGC, Instituto de Informática – Univ. Federal do Rio Grande do Sul

Porto Alegre - Brasil

reis@inf.ufrgs.br

Abstract. *The evolution of Computer Systems that allows the integration of systems with hundred of millions components is demanding more and more the reduction of the power consumption. This paper is a contribution to the establishment of a Grand Challenge research agenda that addresses fundamental challenges in the field of nano- and microelectronics that definitely will impact Computing. This work addresses some challenges for hardware technology evolution and challenges for the next 20 years, to cope with the goal of having a more Green Computing. The grand challenges from the Computer Science point of view are refined into a more realistic scenario than the stated in the first text of SBC Grand Challenges in Computer Science Research. CMOS will still be the main IT technology for at least 20 years). In this paper the computer-systems relevant issues of systems power dissipation is addressed and several issues related to power reduction by an optimization of systems in all level of abstraction are presented. The optimization of systems in all level of abstraction should consider that at the same time that it is needed a reduction in the amount of transitions in a systems, it is more and more important a reduction on the amount of components (transistors). The number of components is responsible for the increasing static consumption in nano CMOS technologies, due to the leakage current present in new devices using transistors with nano dimensions.*

Resumo. *A evolução dos sistemas computacionais, que permite a integração de sistemas com centenas de milhões de componentes, tem demandado mais e mais a redução do consumo de potência. Este artigo é uma contribuição ao estabelecimento de uma agenda de pesquisa dos Grandes Desafios que está relacionado aos desafios fundamentais no campo da nano e microeletrônica que irão definitivamente impactar a Computação. Este trabalho trata de alguns desafios para a evolução da tecnologia de hardware e desafios para os próximos 20 anos, para lidar com o objetivo de ter uma Computação mais Verde. Os Grandes Desafios sob o ponto de vista da Ciência da Computação estão associados a um cenário mais realista do que o cenário apresentado no primeiro texto dos Grandes Desafios da SBC na Pesquisa em Ciência da Computação. CMOS será ainda a principal tecnologia de TI ao menos por mais 20 anos. Neste artigo são tratadas questões relevantes à dissipação de consumo em sistemas computacionais e são apresentadas diversas questões relacionadas à redução de consumo pela otimização de sistemas em todos os níveis de abstração. A otimização de sistemas em todos os níveis de administração deve considerar que ao mesmo tempo em que é necessário uma redução no número de transições em um sistema, é cada vez mais importante uma redução no número de componentes (transistores). O número de componentes é responsável pelo crescente consumo estático em tecnologias nano CMOS, devido à corrente de fuga presente nos novos dispositivos usando transistores com nano dimensões.*

1. Introdução

Em 2006 a SBC publicou um texto indicando os 5 grandes desafios para a pesquisa avançada em Computação até 2016 [SBC 2006]. Este artigo está relacionado ao desafio “Impactos para a área de computação da transição do silício para novas tecnologias”

mas considerando o dito na publicação que fizemos em [Bampi 2009] de que esta transição não deverá ocorrer nas próximas décadas e que a tecnologia CMOS ainda possui ao menos muitas décadas de vida. Esta afirmação também está amparada no ITRS (International Technology Roadmap for Semiconductors) *roadmap* [ITRS2009], que claramente mostra que a tecnologia CMOS ainda será a tecnologia principal para a implementação de sistemas integrados durante várias décadas. Cabe observar que este *roadmap* é elaborada por especialistas de empresas, centros de pesquisa e universidades que estão na ponta dos avanços tecnológicos do setor de semicondutores. Estes *roadmaps* vêm sendo desenvolvidos desde 1992, tendo sido demonstrado ao longo do tempo uma significativa precisão, mas os avanços da tecnologia CMOS tem sido mais significativos do que os previstos nos primeiros *roadmaps*. Enquanto hoje existem tecnologias comerciais na faixa de 30 nm de tamanho da grade de transistores, o ITRS 2009 prevê para 2024 tecnologias com grade de transistores de 7,4 nm, e não coloca isto como um limite para a tecnologia CMOS.

Quanto objetivamos uma “Computação Verde” (Green Computing), devemos considerar técnicas de redução de consumo a serem aplicadas em todos os níveis de abstração utilizados no projeto de um sistemas computacional ou eletrônico. Somente a observação deste objetivo (redução de consumo) em todos os níveis de abstração é que permitirá maximizar a redução de consumo de um sistema. Deve também ser considerado em todos os níveis de abstração, em qual tecnologia o sistema deverá ser implementado. Em função da tecnologia de fabricação a ser utilizada, deve-se observar estratégias diferentes nas etapas de projeto em nível de abstração superior. Um exemplo claro disto é quanto aos mecanismos de estimativa de consumo em descrições de sistemas em alto nível. Ao usar-se na implementação do sistema uma tecnologia CMOS clássica, por exemplo, 0,35 μm , pode-se fazer uma estimativa de consumo considerando-se que o consumo será devido a taxa de transições (“1” lógico para “0” lógico, ou “0” para “1”) e com isto adotar-se técnicas que visam reduzir o numero de transições para a execução de uma função. Mas se usarmos uma tecnologia CMOS de última geração (45 nm, por exemplo), temos de considerar o consumo estático, que é proporcional ao número de componentes (transistores). Isto demanda portanto, que mesmo em níveis mais alto de abstração do projeto, deve-se objetivar uma redução do número de componentes de forma a reduzir o consumo estático.

É evidente também que a busca de uma “Computação Verde” que realmente procure minimizar o consumo de energia, demanda o uso de sistemas em chip (SoC), onde todo o sistema é implementado na forma de apenas um chip. A redução de consumo passa também pelo projeto de sistemas em chip dedicado à aplicação a que se destina, de forma que seja otimizado em função da aplicação visada.

2. O Consumo em Tecnologia MOS

Desde o início dos anos oitenta, a implementação da grande maioria dos sistemas computacionais e eletrônicos tem sido efetuada utilizando circuitos integrados fabricados em tecnologia CMOS (Complementary Metal-Oxide-Semiconductors). Até então era amplamente utilizada a tecnologia NMOS, que era mais veloz e utilizava um número menor de componentes, pois entre a saída de uma porta lógica e a alimentação VDD necessitava de apenas um transistor, denominado transistor de carga, que por ser um transistor funcionando em regime de depleção estava sempre conduzindo e funcionava como sendo uma resistência variável. Na figura 1 é mostrado uma porta

NOR NMOS onde pode-se observar a rede de transistores que compõem a porta lógica. Quando a saída da porta lógica estava com o valor zero, havia uma grande queda de tensão entre os extremos deste transistor de carga, provocando um consumo significativo [reis 2008].

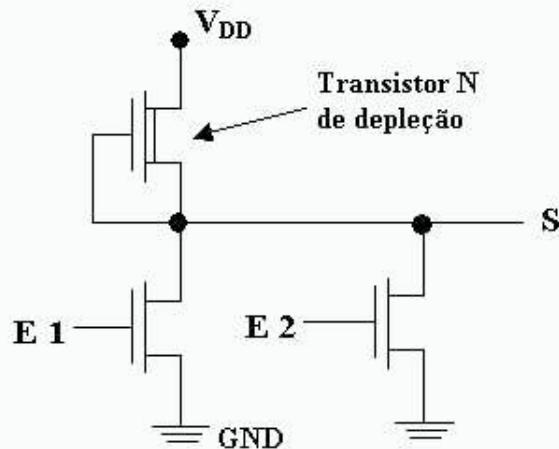


Figura 1. Esquema elétrico de uma porta NOR NMOS

No caso de circuitos CMOS, temos entre a saída da porta lógica e a alimentação VDD uma rede de transistores que é dual a rede de transistores que definem a função lógica da porta (rede *pull-down*) e que fica entre a saída e a massa (GND) (Figura 2). Desta forma, em uma porta lógica CMOS, temos sempre um caminho da saída ativo para a massa (GND) ou para VDD, ou seja, não existe um caminho para a massa e outro para VDD ativos ao mesmo tempo. Isto faz com que não haja um consumo estático em circuitos CMOS tradicionais, e o consumo pode ser computado considerando apenas o consumo dinâmico. Isto já não é mais possível nas tecnologias CMOS nanométricas (como as tecnologias de 65 nm ou menores), pois o consumo estático devido às correntes de fuga (*leakage current*) passou a ser da mesma ordem de grandeza ou maior do que o consumo dinâmico.

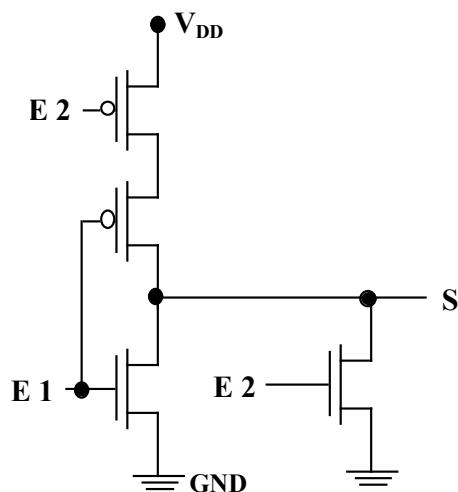


Figura 2. Esquema elétrico de uma porta NOR CMOS

3. Minimização do Número de Transistores

Atualmente, um grande número de projetos são implementados usando tecnologias do tipo Células-Padrão (Standard Cell). Ou seja, o leiaute do circuito é efetuado a partir de uma montagem de células pré-projetadas e existentes em uma biblioteca de células [reis 2009]. Um problema do uso de uma biblioteca de células é que o número de funções lógicas existentes nas bibliotecas comerciais é limitado, sendo em geral na ordem de 150 a 200 funções, com 3 opções de dimensionamento (visando desempenho, área ou consumo). Se considerarmos, por exemplo, o número de funções possíveis utilizando no máximo 4 transistores N e 4 transistores em série, podemos ter 3503 funções lógicas diferentes [DET87].

Tabela 1. Número de funções possíveis considerando um limite de transistores em P e N em série.

NUMBER OF STACKED PMOS TRANSISTORS					
	1	2	3	4	5
NUMBER OF STACKED NMOS TRANSISTORS	1	2	3	4	5
1	1	2	3	4	5
2	2	7	18	42	90
3	3	18	87	396	1677
4	4	42	396	3503	28435
5	5	90	1677	28435	125803

Na figura 3 podemos observar duas soluções para a implementação de uma mesma equação. A primeira solução faz uso de 4 portas lógicas básicas (3 portas NOR de 2 entradas e um inversor CMOS), usando um total de 14 transistores. A segunda solução faz uso de apenas uma porta lógica, que executa a mesma função, mas com apenas 8 transistores. Ou seja, a segunda solução, por ter uma redução do número de transistores, também terá um consumo estático proporcionalmente menor.

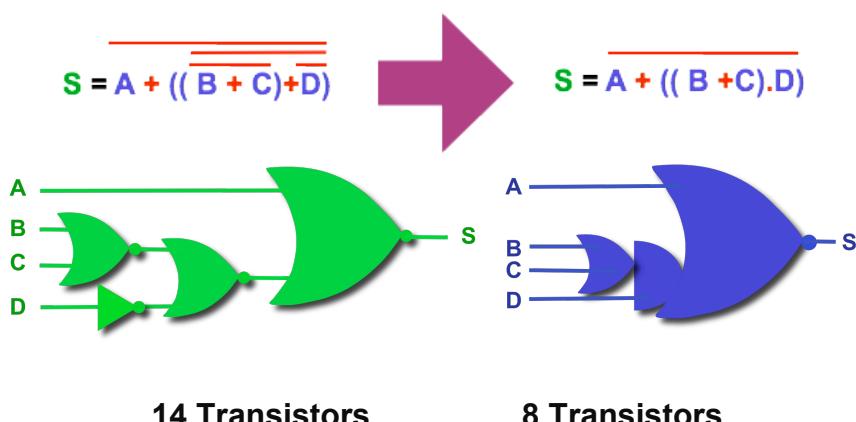


Figura 3. Duas opções para a implementação de uma mesma função

Portanto, a redução do número de transistores passa pela utilização de ferramentas de EDA (*Electronic Design Automation*) que efetuam a transformação das equações lógicas de um sistema de forma que além de corresponderem a equações mapeáveis em portas CMOS, façam um uso otimizado de portas lógicas complexas. Além disto é fundamental ter-se uma ferramenta de síntese de leiaute que consiga efetuar a realização física de qualquer função lógica. Não adianta efetuar uma otimização lógica, se depois deve-se mapear (transformar) as equações em função das portas lógicas disponíveis em uma biblioteca de células tradicional, como ainda é efetuado nos sistemas comerciais de EDA.

4. Síntese Automática do Leiaute

A construção e manutenção de bibliotecas de células tem um custo expressivo e esta é uma razões importantes por as bibliotecas de células terem um número relativamente pequeno de células com funções lógicas diferentes. Considerando isto, um de nossos projetos trata da síntese automática do leiaute de circuitos integrados. Neste caso consideramos a implementação do leiaute como a implementação de uma rede de transistores. Ou seja, a geração do leiaute corresponde a um nível de abstração menor do que o nível de abstração referente ao projeto de leiaute a partir do posicionamento e roteamento de células lógicas retiradas de uma biblioteca de células.

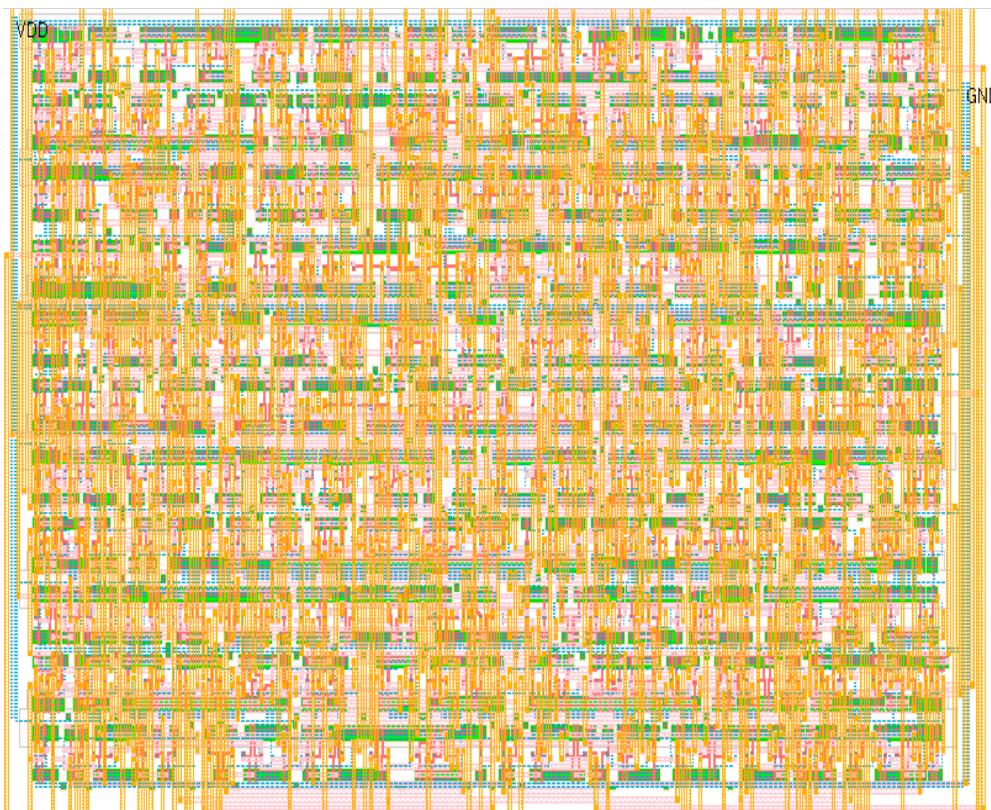


Figura 4. Leiaute gerado automaticamente com a ferramenta Parrot [Lazzari 2006]

A figura 4 mostra o leiaute de um circuito gerado automaticamente com a ferramenta Parrot, desenvolvida na UFRGS [Lazzari 2006]. Ou seja, o leiaute de todo o circuito foi gerado automaticamente, inclusive o leiaute das células lógicas (síntese do leiaute de células “on-the-fly”).

Na figura 5, é apresentado o leiaute de um FF JK gerado automaticamente com a ferramenta ASTRAN (Automatic Synthesis of Transistor Networks) [Adriel 2007] [Reis 2008], ferramenta de EDA em desenvolvimento na UFRGS, e que permite a geração automática do leiaute de células MOS correspondentes a qualquer função lógica, ou seja, a qualquer rede de transistores. Pode-se observar que o leiaute é bastante compacto, incluindo roteamentos internos em polisilício e metal com quebras de direção (*doglegs*). Além de ser possível a síntese automática de qualquer rede de transistores, é possível também a síntese do leiaute de transistores de qualquer tamanho. Quando a largura do transistor for maior do que a altura da banda de difusão P ou N, o transistor pode ser dividido em duas ou mais partes (*folding*).

Em função da diferença nas regras de projeto de diferentes tecnologias, a ferramenta de síntese de leiaute deve ser adequada às possíveis variações destas regras de projeto. Dependendo da evolução tecnológica, as diferenças de regras de projeto podem ser de maior ou menor quantidade.

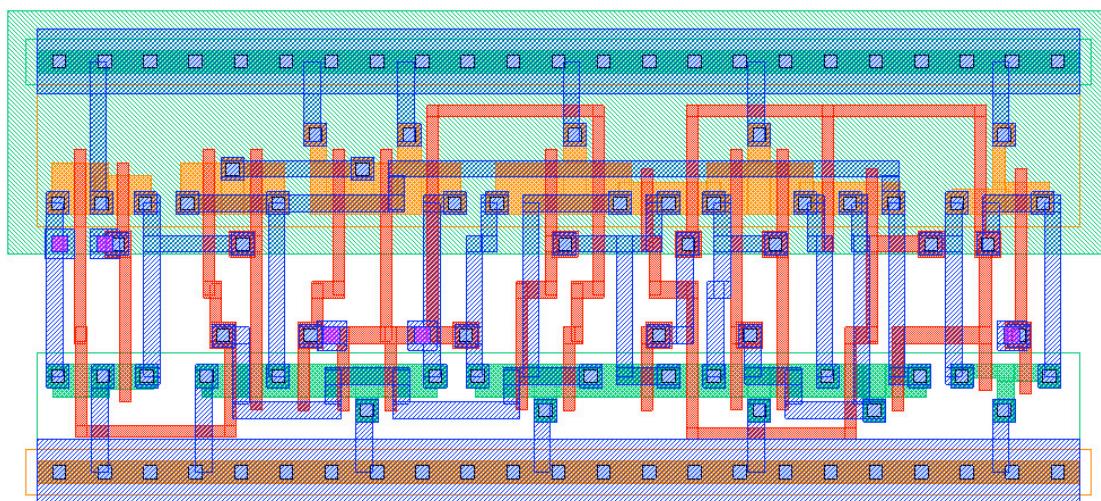


Figura 5. Leiaute um FF JK gerado automaticamente com a ferramenta ASTRAN
[Adriel 2007] [Reis 2008]

A Figura 6 mostra uma comparação entre o resultado obtido na geração de um multiplicador 4x4 usando um fluxo Standard Cell comercial e o resultado obtido utilizando o nosso fluxo de síntese baseado na geração automática das células com a ferramenta ASTRAN. Visualmente já é possível observar que a solução obtida com o ASTRAN apresenta uma área bem menor.

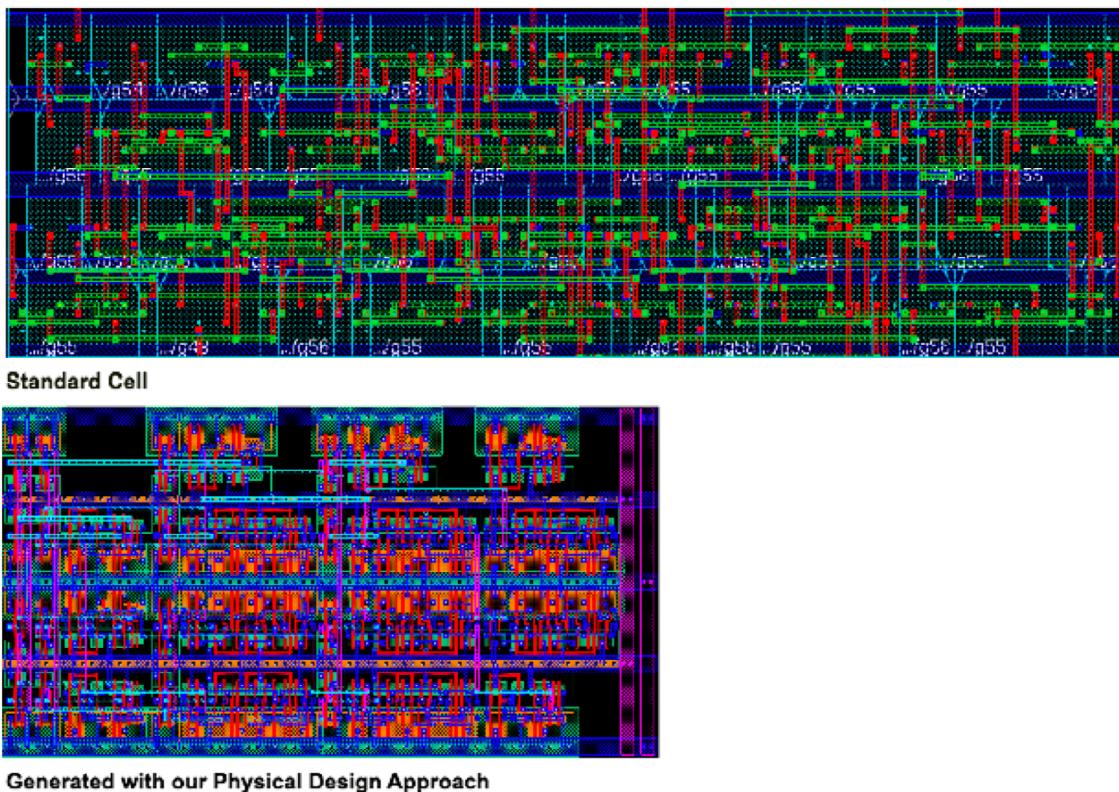


Figura 6. Leiaute um multiplicador 4x4 gerado com ferramentas comerciais e o leiaute do mesmo circuito gerado automaticamente com a ferramenta ASTRAN [Reis 2008].

Tabela 2. Dados numéricos comparando a síntese do multiplicador 4x4 com gerado em Standard Cell e com o fluxo baseada na ferramenta ASTRAN.

	Standard Cell	Cell Compiler	Gain (%)
Number of Cells	52	28	46
Number of Transistors	634	376	59.3
Area (μm^2)	6716	5070	24.50
Delay (ps)	2174	1896	12.8
Power (mW)	6.45	3.97	61.55

Os dados da tabela 2 mostram uma redução significativa do número de transistores, de 634 a 376 transistores, quando efetuado a geração automática. Portanto a solução Standard Cell possui mais de 65% a mais de transistores do que na solução gerada com o ASTRAN. A solução gerada automaticamente com o ASTRAN também apresenta um ganho em redução do atraso (maior desempenho), redução em área e uma significativa redução do consumo, acima de 60%. A redução do número de transistores é

efetivamente a principal causa da redução do consumo, mas a redução do comprimento médio das conexões também contribui. Por sua vez o comprimento médio das conexões é reduzido em função da redução de área do chip, sendo que a área é reduzida principalmente em função da redução do número de transistores. A redução do número de transistores naturalmente também conduz a uma redução do número de conexões. Devemos gerar outros exemplos, mas temos uma previsão de manter ganhos da mesma ordem, pois em qualquer caso, a utilização de portas complexas conduz a uma redução do numero de transistores como visto na figura 3.

5. Conclusões

O projeto de sistemas computacionais visando uma computação verde (Green Computing), considerando que ainda por muitas décadas as tecnologias em silício serão dominantes, passa por um processo de síntese que deve considerar a redução do número de componentes desde a especificação do sistema, no maior nível de abstração possível, até a síntese física. Foi mostrado que o uso de portas complexas (não disponíveis normalmente em biblioteca de células) e da possibilidade de implementar qualquer função lógica permite uma redução expressiva do número de transistores. Foi também mostrado que a redução do número de componentes (transistores) provoca uma redução do consumo estático, presente nas nanotecnologias CMOS atuais. Foi demonstrado também que para evitar as limitações do uso de uma biblioteca de células (em termos de funções lógicas e dimensionamentos disponíveis), devemos efetuar a geração automática do leiaute, tratando como sendo a geração do leiaute de uma rede de transistores. Foram apresentados alguns resultados obtidos com as ferramentas de síntese física construídas por nossa equipe, mostrando um exemplo comparativo com o resultado obtido através do uso de um fluxo de síntese Standard Cell efetuado com o uso de ferramentas comerciais. Portanto, um dos grandes desafios da computação verde é pesquisar e desenvolver métodos de otimização de componentes de um sistema em todos os níveis de abstração e etapas de síntese deste sistema.

6. Agradecimentos

Agradecemos o apoio do CNPq, FINEP e CAPES pelo apoio financeiro ao desenvolvimento dos trabalhos de nossa equipe, assim como aos alunos de mestrado e doutorado do PGMICRO e PPGC e alunos de Iniciação Científica que tem contribuído com os trabalhos de pesquisa que serviram de base para este artigo. Este trabalho também se insere no contexto do INCT NAMITEC.

7. Referências

SBC (2006) Brazilian Computer Society. **Grand Challenges in Computer Science Research in Brazil 2006-2016**, 25pgs. In: <http://www.sistemas.sbc.org.br>. Último acesso em 22 de março de 2010.

BAMPI, S., SUSIN, A., REIS, R., **Systems Architectural Challenges for Transitional and Compatible to CMOS Technologies in Giga-Scale Hardware Integration**,

SEMISH 2009, Anais do 36º Seminário Integrado de Software e Hardware, Bento Gonçalves, 21 a 22 de Julho de 2009, p. 281-292, ISSN: 2175-2761.

ITRS (2009), International Roadmap Committee. “**The International Technology Roadmap for Semiconductors - 2009**”. In <http://www.itrs.net/home.html>. Último acesso 26 de março de 2010.

REIS, R. e Cols., **Concepção de Circuitos Integrados**, 2ª Edição. Série Livros Didáticos do Instituto de Informática, Editora Bookmann, Porto Alegre, 2009, 258 páginas. ISBN 9788577803477

DETJENS, E. et al. **Technology Mapping in MIS**, IEEE ICCAD, proceedings, pp. 116-119, 1987.

LAZZARI, C., SANTOS, C., REIS, R., **A New Transistor-Level Layout Generation Strategy for Static CMOS Circuits**, 13th IEEE International Conference on Electronics, Circuits and Systems – ICECS2006, Nice, France, December 10 - 13, 2006, p. 660-663, ISBN: 1-4244-0395-2.

ZIESEMER, A.; LAZZARI, C., REIS, R., **Transistor Level Automatic Layout Generator for non-Complementary CMOS Cells**, In: IFIP/CEDA VLSI-SoC2007, International Conference on Very Large Scale Integration, Atlanta, USA, October 15-17, 2007, pp. 116-121, ISBN: 978-1-4244-1710-0.

REIS, R., **Physical Design Automation at Transistor Level**, IEEE NORCHIP 2008, Tallin, Estonia, November 17-18, 2008 (INVITED TALK), 5 p., ISBN 978-1-4244-2493-1