

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

ANTONIO DE QUADROS ANDRADE JÚNIOR

**PLANEJAMENTO DE TESTE DE SISTEMAS BASEADOS EM
NÚCLEOS DE HARDWARE DE SINAL MISTO USANDO BIST**

Porto Alegre

2005

ANTONIO DE QUADROS ANDRADE JÚNIOR

**PLANEJAMENTO DE TESTE DE SISTEMAS BASEADOS EM NÚCLEOS DE
HARDWARE DE SINAL MISTO USANDO BIST**

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica (PPGEE), da Universidade Federal do Rio Grande do Sul (UFRGS), como parte dos requisitos para a obtenção do título de Mestre em Engenharia Elétrica.

Área de concentração: Automação e Instrumentação Eletro-Eletrônica

ORIENTADOR: Marcelo Soares Lubaszewski

Porto Alegre

2005

ANTONIO DE QUADROS ANDRADE JÚNIOR

**PLANEJAMENTO DE TESTE DE SISTEMAS BASEADOS EM NÚCLEOS DE
HARDWARE DE SINAL MISTO USANDO BIST**

**Esta dissertação foi julgada adequada para a
obtenção do título de Mestre em Engenharia
Elétrica e aprovada em sua forma final pelo
Orientador e pela Banca Examinadora.**

Orientador: _____

Prof. Dr. Marcelo Soares Lubaszewski, UFRGS

Doutor pela Institut National Polytechnique de Grenoble, França

BANCA EXAMINADORA:

Prof. Dr. José Calvano Vicente, IPqM

Doutor pela Universidade Federal do Rio de Janeiro – Rio de Janeiro, Brasil

Prof. Dr. Luigi Carro, UFRGS

Doutor pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

Prof. Dra. Fernanda Gusmão de Lima Kastensfeld, UFRGS

Doutor pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

Coordenador do PPGEE: _____

Prof. Dr. Carlos Eduardo Pereira

Porto Alegre, maio de 2005

DEDICATÓRIA

Há algumas pessoas sem as quais este trabalho não se tornaria realidade.

Dedico este trabalho, em especial, aos meus pais, pela dedicação e apoio em todos os momentos difíceis, por não medirem esforços em propiciar-me um ambiente adequado para que o trabalho se desenrolasse a contento. A meus irmãos que nesses dois anos demonstraram um companheirismo que ainda não havia experimentado, e uma torcida e interesse pelo meu sucesso que me deixou feliz e ciente de estar fazendo a coisa certa.

Não posso me esquecer também dos amigos que deixei em Salvador e que me apoiavam com seus telefonemas, correio eletrônico e mensagens instantâneas. Não cito nomes, mas cada um que contribuiu sabe que nesse texto está um pedacinho seu.

Dedico também à Igreja Batista Central, que, embora fosse eu um desconhecido, me acolheu e cuidou como se fosse um da família – o que de certa forma é verdade. Em especial, algumas famílias se fizeram mais presentes no meu dia-a-dia: Ana, Jorge, e Jônatas Corrêa; Alice e Wilson Bonotto e Tiago Massulo; Neander, Regina, Renato e Rafael Winter; Lira, Valentim e Tiago Garros; e à minha “avó” baiana “Didi”, minhas segundas famílias, cujas participações foram cruciais para a conclusão deste trabalho. Também estendo minha inefável gratidão ao Clube Bíblico Jovem da mesma igreja, bem como a todos os companheiros e amigos que fiz durante estes dois anos.

Enfim, dedico este trabalho ao meu Deus, que tem cuidado de mim, apesar de meu relaxamento, e me presenteia com essa vitória pela Sua Graça.

AGRADECIMENTOS

Ao Departamento de Engenharia Elétrica da Universidade Federal da Bahia (UFBA), que me deu a formação técnica básica e abriu as oportunidades pelo Brasil para que pudesse contribuir positivamente para a sociedade brasileira em minha carreira. Em especial às professoras Lígia Palma e Ana Isabela Araújo por terem me orientado e incentivado a perseguir o sonho da pós-graduação em outro estado.

Ao Programa de Pós-graduação em Engenharia Elétrica da Universidade Federal do Rio Grande do Sul (UFRGS), pela oportunidade de realização de trabalhos em minha área de pesquisa, pela confiança em meu potencial e pelos frutos gerados. Por todos os professores que participaram deste processo, seja como ministrantes de disciplinas ou orientadores em trabalhos, em especial aqueles ligados à pesquisa em microeletrônica, professores Luigi Carro, Altamiro Susin e meu orientador, Marcelo Lubaszewski, pelo seu exemplo, dedicação e companheirismo.

Ao Grupo de Microeletrônica (GME) da UFRGS, que me mostrou ser a minha escolha por esta escola a mais acertada, além de tantas amizades criadas. Não posso deixar de citar a professora Erika Cota, que participou diretamente deste trabalho, seja orientando tecnicamente como também com uma boa conversa informal. Cito também aqueles que foram mais próximos a mim nestes dois anos, como os companheiros do Laboratório de Prototipação e Teste, Erik Schuler, Tiago Balen, Gustavo Pereira, Tiago Jost, e do Laboratório de Processamento de Sinais e Imagens, Josias Mainardi, Letícia Guimarães, Carla Diniz, Viviane Cordeiro, Emilena Specht, Thiago Figueiró, André Borin, Marcelo Negreiros.

Ao CNPq pela provisão da bolsa de mestrado, que serviu para minha manutenção financeira longe de minha família, bem como a toda a população brasileira, que voluntariamente ou não, contribui para a educação superior pública no Brasil. Apesar das dificuldades, este é um projeto que vale a pena e é o primeiro passo de qualquer caminho que o país tome rumo a uma maior autonomia tecnológica, econômica e política em um contexto global.

RESUMO

Atualmente, os sistemas eletrônicos integrados seguem o paradigma do projeto baseado em núcleos de hardware. Além de núcleos digitais, tais sistemas podem incluir núcleos analógicos, que, neste caso, dominam os requisitos de teste, como tempo de teste e número adicional de pinos. Conseqüentemente, há um aumento do custo total de manufatura do dispositivo. O presente trabalho propõe o uso de técnicas de autoteste integrado (BIST) analógico, baseado no reuso de núcleos digitais presentes no mesmo sistema, com objetivo de reduzir os custos relativos ao teste do sistema. Além disso, uma estratégia satisfatória requer um adequado planejamento de teste, de forma a melhor explorar as possibilidades de teste simultâneo de mais de um núcleo e o escalonamento do teste de cada um destes, diminuindo custos associados ao teste. Adaptando uma ferramenta computacional voltada ao planejamento de sistemas compostos exclusivamente de núcleos digitais para o universo dos sistemas mistos e considerando a possibilidade do uso de BIST, pode-se avaliar o impacto da estratégia proposta em termos de tempo de teste, acréscimo de área em virtude das estruturas de teste e pinos extras. Restrições de dissipação de potência também são consideradas. Para validação das hipóteses levantadas, sistemas mistos foram descritos a partir de *benchmarks* industriais e acadêmicos puramente digitais, através da inclusão de núcleos analógicos. Os resultados obtidos através de simulações com a ferramenta apontam para uma redução no tempo de teste e otimização de custos de pinos e área, além da redução no custo de equipamentos automatizados de teste (ATE), para o caso de teste de produção. Com isso, uma redução no custo total do procedimento de teste de tais sistemas pode ser alcançada.

Palavras-chaves: Planejamento de teste. Circuitos de sinal misto. BIST. Sistemas baseados em núcleos de hardware (SOC). Redução de custos de teste.

ABSTRACT

Currently, integrated electronic systems follow the core-based design paradigm. Such systems include not only digital circuits as internal blocks, but also analog circuits, which dominate test resources, such as testing time, extra pins and overhead area, thus increasing the total manufacture cost of these devices. The present work proposes the application of analog Built-in Self Test (BIST) techniques based on the reuse of available digital cores within the same integrated system, aiming to reduce the test costs of the analog cores. Moreover, a satisfactory strategy requires an adequate test planning, so that the design space is better explored. By adapting a software tool, which was originally designed for test planning of exclusively digital SOC, to consider analog cores, as well as the possibility of BIST, one can evaluate the impact of the proposed strategy in terms of test application time, area overhead due to test structures added and extra pins. Power dissipation restrictions may also be taken into account. In order to validate the hypotheses considered, mixed-signal systems are described from digital industrial and academic benchmarks, just adding analog cores. Through simulation with the adapted tool, the obtained results point to a decrease in the system test time, as well as a reduction in the cost of Automatic Test Equipment (ATE), in case of a production test. Thus, a reduction in the overall cost of the test procedure for such devices can be achieved.

Keywords: Test Planning, Mixed-Signal circuits, BIST, Systems-on-chip (SOC), Test cost reduction.

SUMÁRIO

1	INTRODUÇÃO	13
2	ESTADO-DA-ARTE EM TESTE DE CIRCUITOS ANALÓGICOS	19
2.1	AUTO-TESTE INTEGRADO (BIST)	22
2.1.1	BIST de ADCs usando histograma linear	24
2.1.2	BIST baseado em FFT.....	25
2.1.3	BIST baseado em ruído.....	27
2.1.4	Auto-teste baseado em oscilação (OBIST).....	28
2.1.5	BIST baseado na análise da resposta transiente.....	31
3	ESTADO-DA-ARTE EM TESTE DE SOC	35
3.1	PROJETO DE SISTEMAS BASEADOS NO REUSO DE NÚCLEOS DE HARDWARE (SOC) ...	35
3.2	TESTE DE SOCS	37
3.3	TESTE DE SOCS DE SINAL MISTO	46
4	PLANEJAMENTO DE TESTE DE SOCS E BIST BASEADO EM REUSO.....	51
4.1	REBATE - <i>REUSE-BASED TEST SCHEDULING</i>.....	52
4.1.1	Algoritmo de escalonamento e definição das TAMs	54
4.2	ADAPTANDO A FERRAMENTA PARA TESTE DE SOCS MISTOS	62
5	RESULTADOS EXPERIMENTAIS	68
5.1	RESUMO DESCRITIVO DOS <i>BENCHMARKS</i> UTILIZADOS	68
5.2	RESULTADOS DAS SIMULAÇÕES USANDO A FERRAMENTA	69
5.3	RESULTADOS DAS SIMULAÇÕES EXPLORANDO PARALELISMO DE TESTE	76
6	CONCLUSÕES E TRABALHOS FUTUROS.....	82
	REFERÊNCIAS	85
	APÊNDICE A : arquivo MATALB com descrição do sistema u226m.....	91

LISTA DE ILUSTRAÇÕES

Figura 1 - Diagrama de blocos simplificado de um SOC.....	14
Figura 2 - Evolução dos custos de fabricação e teste do dispositivo, para transistores dentro de um CI digital.....	15
Figura 3 - Diagrama de blocos de teste particionado (analógico e digital) de um sistema misto.....	20
Figura 4 - Caracterização de conversores AD e o tipo de teste associado.....	22
Figura 5 - Diagrama de blocos de BIST em um mesmo encapsulamento.....	23
Figura 6 - Diagrama de blocos do modelo de ADC proposto por Csizmadia.....	27
Figura 7 - Implementação da estratégia de teste baseada em oscilação, avaliando frequência e amplitude do sinal gerado.....	29
Figura 8 - Diagrama de blocos do circuito de codificação sigma-delta para a resposta ao teste baseado em oscilação.....	30
Figura 9 - Diagrama de blocos para BIST baseado em TRAM.....	32
Figura 10 - Classificação dos blocos IP quanto ao nível de abstração e flexibilidade.....	36
Figura 11 - Hierarquia entre núcleos e supernúcleos em um SOC.....	36
Figura 12 - Comparação do fluxo de projeto para: a) SOB; b) SOC.....	37
Figura 13 - Arquitetura e estruturas de teste de um SOC.....	40
Figura 14 - Topologias de acesso para o Modelo Multi-TAM: a) externo direto; b) via núcleo, sem reuso de sua funcionalidade; c) com reuso de conexões funcionais; d) via <i>bypass</i> paralelo; e) via <i>bypass</i> serial.....	42
Figura 15 - Exemplo de escalonamento de teste de núcleos para um SOC arbitrário.....	43
Figura 16 - Métodos de escalonamento: a) Não-particionado; b) Particionado com execução até o fim; c) Particionado com preemptividade.....	44
Figura 17 - a) Arquitetura simplificada do barramento de teste analógico segundo o padrão IEEE 1149.4; b) Arquitetura simplificada do Módulo Analógico de Periferia (ABM).....	47
Figura 18 - Arquitetura de teste de um SOC adotando <i>wrapper</i> analógico.....	48
Figura 19 - Arquitetura do SOC U226.....	55
Figura 20 - Otimização de recursos de teste de núcleos digitais (pinos e área).....	57

Figura 21 - Construção do escalonamento do teste dos núcleos: (a) construindo lista de conflitos de recursos baseado no escalonamento corrente; (b) identificação de conflitos e escalonamento do núcleo restante.....	58
Figura 22 - Busca dos acessos de teste e expansão da árvore até a interface do sistema: (a) Lista das possibilidades de conexão entre CUT e núcleos e interface; (b) seleção das TAMs menos caras para cada núcleo; (c) escolha do melhor caminho e expansão da TAM para o próximo nível	59
Figura 23 - Pseudo-código para escalonamento de núcleos e otimização de custos de teste da ferramenta REBATE	61
Figura 24 - Exemplo de conexão em TAM para circuitos analógicos	63
Figura 25 - Exemplo de TAMs para sistema misto hipotético e respectivo escalonamento, considerando métodos padrão de teste dos núcleos analógicos.....	65
Figura 26 - Exemplo de TAMs para sistema misto hipotético e respectivo escalonamento, considerando método de teste baseado em ruído para os núcleos analógicos	66
Figura 27 - Exemplo de TAMs para sistema misto hipotético e respectivo escalonamento, considerando método de teste baseado em oscilação para os núcleos analógicos ..	67
Figura 28 - Aumento do paralelismo de teste do sistema através do aumento do número de núcleos disponíveis para geração de padrões de teste e análise da resposta.....	77
Figura 29 - Redução no paralelismo devido ao escalonamento de teste com restrições de potência.....	78
Figura 30 - Ilustração da exploração do paralelismo no teste de núcleos analógicos usando o tempo ocioso do processador reusado	80

LISTA DE TABELAS

Tabela 1 - Comparação entre métodos de BIST para ADC e filtros	34
Tabela 2 - Características dos <i>benchmarks</i> utilizados nas simulações	68
Tabela 3 - Resultados da simulação do SOCs para BIST baseado em ruído, variando o número de núcleos com BIST	71
Tabela 4 - Resultados da simulação do SOCs para BIST baseado em ruído, variando o número de núcleos analógicos com BIST e limite de 30% na potência máxima durante o teste	73
Tabela 5 - Resultados da simulação do SOCs para BIST baseado em oscilação, variando o número de núcleos com BIST	74
Tabela 6 - Resultados de simulação com teste rodado a 20MHz e a 50MHz	76
Tabela 7 - Resultados experimentais para p93791m e p34392 com 3 processadores disponíveis	78
Tabela 8 - Resultados da simulação do p93791m2, com reuso de processadores, diferentes proporções de potência entre núcleos analógicos e digitais	79
Tabela 9 - Resultados da simulação para os sistemas P93791m e P34392m explorando o paralelismo com um único processador, usando intervalos de tempo entre os processamentos das amostras	80

LISTA DE ABREVIATURAS

ABM: Analog Boundary Module

ADC: Analog-to-Digital Converter

ATE: Automatic Test Equipment

ATPG: Automatic Test Pattern Generation

AWG: Arbitrary Waveform Generator

BIST: Built-in Self-Test

CI: Circuito Integrado

CMOS: Complementary Metal-Oxide-Semiconductor

CTL: Core Test Language

CUT: Circuit Under Test

DAC: Digital-to-Analog Converter

DfT: Design for Testability

DNL: Differential Non-Linearity

DSP: Digital Signal Processing

DTMF: Dual Tone Mode Frequency

ENoB: Effective Number of Bits

FFT: Fast Fourier Transform

IEEE: the Institute of Electrical and Electronic Engineers

INL: Integral Non-Linearity

IP: Intellectual Property

ITRS: International Technology Roadmap for Semiconductors

LSB: Least Significant Bit

OBIST: Oscillation Built-in Self Test

ORA: Output Response Analysis

PCI: Placa de Circuito Impresso

PDA: Personal Digital Assistant

PSD: Power Spectrum Density

RF: Rádio-frequência

RISC: Reduced Instruction Set Computer

RMS: Root Mean Square

RTL: Register Transfer Level

SFDR: Spurious Free Dynamic Range

SINAD: Signal-to-Noise-and-Distortion Ratio

SNR: Signal-to-Noise Ratio

SOB: System-on-a-Board

SOC: System-on-a-Chip

TAM: Test Access Mechanisms

THD: Total Harmonic Distortion

TPG: Test Pattern Generation

TRAM: Transient Response Analysis Method

UDL: User Defined Logic

1 INTRODUÇÃO

A indústria de semicondutores tem se caracterizado, desde seu início, por constantes inovações na fabricação de dispositivos e no desenvolvimento de sistemas eletrônicos, sempre visando um melhor desempenho e uma redução nos custos de manufatura.

A evolução da tecnologia de fabricação de circuitos integrados tem levado à construção de circuitos com dimensões cada vez menores, permitindo o projeto e manufatura de circuitos com maior número de componentes e de maior complexidade. Sistemas completos podem ser incluídos em uma única pastilha de silício, proporcionando vantagens em termos de redução de custos de projeto e acelerando o tempo de desenvolvimento do produto para o mercado, ou *time-to-market*. Este novo paradigma é popularmente conhecido como sistemas baseados em núcleos de *hardware*, ou Sistemas-em-um-chip, ou somente *System-on-a-chip* (SOC), contrapondo-se ao paradigma antigo de Sistema-em-uma-placa, ou *System-on-a-board* (SOB).

Um SOC é um circuito de alta escala de integração de transistores, em que blocos funcionais baseados em diferentes tecnologias (como lógica, circuitos analógicos, memórias, circuitos bipolares, entre outros), antes fabricados separadamente e integrados em uma Placa de Circuito Impresso (PCI), são fabricados em uma mesma pastilha de silício (Figura 1). Tal paradigma permite a redução de custos na fase de projeto do ciclo de produção de sistemas eletrônicos integrados. Por exemplo, o uso de blocos com propriedade intelectual, ou blocos IP simplifica o fluxo de projeto de tais sistemas, tornando-o mais rápido e mais barato. Esses blocos realizam uma função específica, mas comum a vários sistemas, de modo que podem ser integrados a diferentes tipos de sistema, como por exemplo, um processador, que pode ser utilizado tanto em uma máquina de lavar roupa como pode fazer parte do sistema de injeção de combustível de um automóvel. Eles podem ser projetados pelo próprio desenvolvedor do sistema ou por uma empresa especializada no projeto e descrição de tais blocos, que pode fornecer a mais de um cliente. Com isso, a responsabilidade do projetista do sistema é de escolher dentre vários fornecedores de blocos IP aquele cujo bloco oferece a melhor compatibilidade com o sistema a ser desenvolvido e melhor desempenho em termos de desempenho, área em silício ocupada, e potência dissipada.

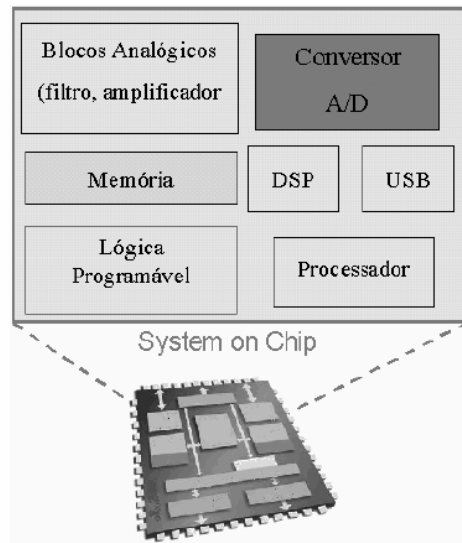


Figura 1 - Diagrama de blocos simplificado de um SOC

Atualmente, este paradigma inclui o uso de circuitos digitais, analógicos e memórias na mesma lâmina de silício, o que não realizado antes por limitação tecnológica dos equipamentos. Assim, tornou-se possível a criação de circuitos integrados com a capacidade de receber sinais em rádio-frequência (RF), trazendo-os para a banda base por meio de um demodulador analógico, efetuar o processamento da informação contida no sinal usando DSP ou qualquer outro processador e ainda reenviar informações ou comandos utilizando um transmissor de RF, por exemplo (VANA; BLAIR, 2003). Como já citado, isto tem enormes atrativos para a indústria, permitindo inclusive a diminuição das dimensões de dispositivos portáteis, como telefones móveis e *Personal Digital Assistants* (PDA).

Apesar destas vantagens, o teste destes sistemas representa ainda um desafio a ser vencido. Um gráfico retirado do *ITRS 1997 Roadmap* e disponível em (ZORIAN; DEY; RODGERS, 2000) retrata a diminuição ao longo do tempo dos custos referentes à fabricação de dispositivos semicondutores (Figura 2). Neste mesmo intervalo, os custos referentes ao teste não sofreram redução semelhante. Isso não quer dizer que a área de tecnologia de teste não tenha evoluído ao longo dos últimos anos. Pelo contrário, o desenvolvimento de técnicas como o *Boundary-Scan* e o Auto-Teste Integrado, ou *Built-in Self Test* (BIST), além de novos paradigmas como o Projeto visando o Teste e a Testabilidade (*Design for Test* e *Design for Testability*, ou DfT, respectivamente) permitiram que o custo associado ao teste não se tornasse ainda maior, mesmo com a crescente complexidade no projeto de novos sistemas.

Em certos casos, a inexistência de tais estratégias poderia inclusive inviabilizar a fase de verificação do projeto. Ainda assim, o mesmo gráfico aponta que os custos de teste chegam a aproximadamente 50% do custo total de manufatura do dispositivo. Apesar de ter sido confeccionado em 1997, as previsões da evolução dos custos têm se cumprido ao longo dos anos.

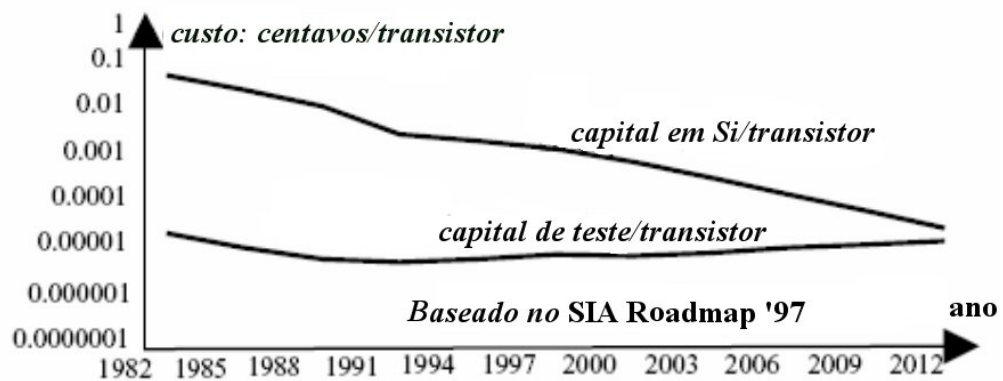


Figura 2 - Evolução dos custos de fabricação e teste do dispositivo, para transistores dentro de um CI digital

Fonte: Zorian; Dey; Rodgers (2000)

No contexto de SOCs, os núcleos fornecidos pelos provedores de blocos IP apresentam uma descrição de alto nível do seu funcionamento, a fim de proteger a propriedade intelectual, acompanhado de um conjunto de vetores de teste e respostas esperadas. Fica a cargo do projetista-integrador a definição do teste do sistema. Atividades associadas ao teste podem ser realizadas ao longo das fases de projeto, por exemplo, a verificação do funcionamento em nível de núcleos e em nível de sistema. Após a fabricação, o circuito será testado antes de ser encapsulado, garantindo seu correto funcionamento dentro de tolerâncias pré-estabelecidas. De uma maneira simplista, cada núcleo embutido no sistema poderia ter pinos exclusivos de acesso ao ambiente externo ao encapsulamento, para fins de inserção de estímulos e extração de respostas ao teste. Tal solução implica num custo muito alto em termos de número de pinos, preço este que nem sempre pode ser pago, ou até mesmo inviabilização por limitações tecnológicas. Outra possibilidade é o uso de um barramento exclusivo para condução de sinais de teste, a ser compartilhado por cada núcleo dentro do sistema. Mais uma vez, tal solução implica em um custo muito alto em termos de tempo de teste, já que o barramento só poderia ser usado por um núcleo de cada vez, o que implica na serialização do seu teste. Em outros casos, se necessário, o circuito será monitorado durante

sua operação normal em campo em busca de modos de falha e sua correção em funcionamento, levando ao acréscimo de área em silício consumida e degradação de desempenho devido às estruturas adicionais. Dessa forma, a responsabilidade do teste de um SOC caminha para se tornar um trabalho conjunto do integrador de sistemas e dos provedores de blocos IP, no sentido de diminuir os custos envolvidos.

Com o desenvolvimento de técnicas para o teste de circuitos digitais, o teste de SOCs constituídos apenas de núcleos digitais já é bastante explorado, apresentando técnicas que diminuem satisfatoriamente os custos de teste. Contudo, no caso de SOCs de sinal misto, é necessário também considerar o teste dos núcleos analógicos. Uma vez que, de forma geral, ainda não há procedimentos padrão ou com resultados satisfatórios para esses núcleos, cria-se um grande desafio para projetistas de circuitos e engenheiros de teste. As estratégias atualmente usadas não conseguem satisfazer o compromisso entre tempo de teste e custo do circuito de teste, por exemplo. A fim de diminuir tais custos, o gerador de estímulos, o analisador de resposta ou ambos podem ser integrados junto ao sistema sob teste, caracterizando uma estratégia de BIST. Com isso, os requisitos do testador externo, ou *Automatic Test Equipment (ATE)* podem ser relaxados, diminuindo seu custo, podendo em alguns casos até dispensar o seu uso. Entretanto, tal estratégia tem como desvantagem o consumo de área em silício, diminuindo o rendimento, ou seja, a produção líquida, tanto devido à área que deixou de ser usada para o circuito, quanto aumentando a área de cada circuito e, conseqüentemente, a probabilidade de ocorrência de defeitos de fabricação.

No intuito de diminuir ainda mais os custos de teste de circuitos analógicos, Negreiros (2002) lança a idéia de se usar um conversor *dithered*, isto é, usando ruído deliberadamente, de forma que o teste de blocos analógicos pode usar circuitos digitais, como *Digital Signal Processors (DSP)*, para processamento e avaliação da resposta ao teste. Contextualizando, seria possível então minimizar requisitos do testador externo, o acesso dos núcleos aos geradores de padrões de teste e analisadores de resposta, e inclusive o tempo de teste, fazendo com que blocos digitais já testados sejam usados para gerar vetores de teste ou avaliar a resposta de blocos analógicos sob teste.

Ainda no universo dos SOCs, uma alternativa para se embutir geradores de estímulos e analisadores de resposta com mínimo acréscimo de área se baseia na estratégia de reuso dos

recursos já disponíveis no sistema, isto é, outros núcleos presentes no mesmo chip. Com isso, o acréscimo de área em silício devido ao circuito de teste é diminuído.

No contexto de SOCs exclusivamente digitais, Cota (2003) desenvolveu um trabalho em que propõe uma ferramenta de apoio à decisão para estratégia de teste nesses circuitos, realizando uma exploração do espaço de projeto de modo que tais decisões sejam feitas ao longo da fase de projeto do sistema, e não apenas ao fim desta. Essa ferramenta se apóia numa exploração do espaço de projeto de modo que interconexões e infra-estrutura de teste são definidas de forma a encontrar a melhor solução em termos de consumo de área em silício, desempenho e potência dissipada. Além disso, explorando relações de precedência na ordem de teste dos núcleos do sistema, a mesma ferramenta busca a melhor solução de escalonamento do teste de cada núcleo e alocação de recursos de teste, sendo que restrições de dissipação de potência podem ser consideradas, o que reduziria o potencial de teste simultâneo.

A proposta deste trabalho consiste numa extensão do trabalho desenvolvido por Cota (2003), englobando agora SOCs compostos de núcleos digitais e analógicos. A inclusão de núcleos analógicos traz uma nova dimensão ao problema de teste destes sistemas, devido aos motivos já citados, principalmente os custos de um testador externo misto. Assim, deve-se levar em consideração a possibilidade de redução de custos do testador externo e a integração dos núcleos analógicos no algoritmo da ferramenta. Uma forma de relaxar os requisitos deste testador externo sem acréscimo significativo de área devido ao circuito de teste é aproveitar os blocos digitais presentes no sistema para testar os blocos analógicos, segundo a proposta de Negreiros (2002). Com isso, a sequência de blocos a serem testados e os mecanismos de acesso ao teste, *Test Access Mechanisms* (TAMs), usadas levam em consideração não só a possibilidade do teste simultâneo de mais de um bloco e a disputa destes pelos mesmos recursos, mas também as relações de interdependência entre o circuito analógico a ser testado e o núcleo digital já testado que pode ser reusado. A simultaneidade de teste dos núcleos do sistema faz com que o tempo de teste total do sistema seja reduzido, assim buscam-se maneiras de explorar esse paralelismo de maneira a conseguir tal redução sem incremento de outros custos. Além disso, a interdependência no caso de reuso implica em uma serialização do teste, que por sua vez aumenta o tempo de teste. A ferramenta proposta deve ser capaz também de abordar esse aspecto, de forma a superar esse obstáculo e prover uma solução satisfatória.

Outra decisão importante é a escolha do método de teste para os núcleos analógicos que permita uma maior redução dos custos associados. Para isso, diversos métodos foram estudados, comparados e inseridos na descrição dos núcleos dentro da ferramenta, de forma que o impacto de cada um pode ser avaliado, ajudando o projetista na sua decisão. Os resultados são avaliados em termos de tempo de teste, área em silício ocupada pela infraestrutura de teste e número de pinos adicionais necessários, observando se a adoção de BIST analógico de fato implica em redução do tempo de teste. Outras restrições importantes, como a potência consumida durante o procedimento de teste, também podem ser consideradas pela ferramenta em discussão, verificando qual a influência dessa restrição nos resultados.

O capítulo 2 deste documento traz a revisão do estado-da-arte na área de teste de circuitos analógicos e métodos BIST, enquanto que o capítulo 3 discorre sobre o estado-da-arte na área de teste de SOCs digitais e mistos. O capítulo 4 descreve a proposta de planejamento de teste combinado com BIST analógico e reuso dos blocos do sistema, apresentando a ferramenta de suporte ao planejamento, REBATE-MIX. Resultados obtidos usando a ferramenta aplicada a sistemas mistos baseados nos *benchmarks* ITC'02 são apresentados e discutidos no capítulo 5, enquanto que o capítulo 6 traz as conclusões finais e apresenta possíveis propostas para novos trabalhos e soluções no campo de teste de SOCs de sinal misto.

2 ESTADO-DA-ARTE EM TESTE DE CIRCUITOS ANALÓGICOS

O desenvolvimento de técnicas para o teste de circuitos analógicos não acompanhou o mesmo ritmo do teste de circuitos digitais, que já apresentam resultados consistentes e padronizados. Tal atraso deve-se a aspectos inerentes ao domínio analógico, dentre eles, o alto número de especificações de desempenho associadas aos circuitos analógicos, a natureza analógica dos sinais, que implica em um grande número de vetores de teste, e a dificuldade de se estabelecer um modelo de falhas suficientemente abrangente e simples, como as falhas do tipo *stuck at* em circuitos digitais.

Técnicas de DfT, incluindo BIST digital, apresentam características desejáveis para o estabelecimento de uma solução de teste de aplicação geral, como modelos de falhas padronizados, métodos de acesso de teste padronizados, suporte à geração automática de padrões de teste (ATPG) e uma emergente linguagem padrão para descrição da estrutura de teste. Tais características não são realidade no teste analógico, fazendo com que soluções nesse domínio sejam mais orientadas a aplicações específicas, além de requerer um alto grau de conhecimento e experiência não só em teste, mas também em arquitetura de circuitos analógicos, por parte dos engenheiros (SONG, 2002).

O teste de circuitos analógicos pode ser classificado quanto ao seu objetivo. Ele pode ser orientado ao diagnóstico das falhas, visando esclarecer o motivo da ocorrência de um determinado defeito, ou orientado à caracterização do dispositivo, isto é, se ele atende às especificações das aplicações a que ele é destinado, ou ainda orientado à produção em massa. Nesse último-caso, é também conhecido como teste *go/no-go*, indicando que o dispositivo tem ou não as condições mínimas de operação para a aplicação. A complexidade do teste aumenta na ordem em que os diversos tipos são apresentados nesse parágrafo, influenciando no custo do teste e no *time-to-market*.

Outro tipo de classificação diz respeito ao nível de abstração adotado no modelo de falhas adotado quando da elaboração da estratégia de teste. Caso a arquitetura do circuito sob teste, *Circuit Under Test* (CUT), seja conhecida, pode-se trabalhar com hipóteses de defeitos nesta, como variações nos valores dos componentes ou defeitos na topologia do mesmo, causadas por variações do processo e/ou impurezas presentes na fabricação. Nesse caso, diz-se ser um teste estrutural. Por outro lado, em outras situações, o modelo de falhas é baseado

nas especificações de desempenho do circuito, como ganho, largura de banda, relação sinal-ruído, para citar alguns. Nesse caso, deseja-se verificar se os circuitos atendem a requisitos de operação, ou seja, se seu desempenho está dentro de uma tolerância do valor nominal de projeto, sendo este tipo de teste também referido como teste funcional.

A Figura 3 mostra um diagrama de blocos simplificado para o teste de sistemas mistos, em que se utiliza a estratégia de particionar o teste digital e o analógico. Para o teste da parte analógica, o sinal de estímulo pode ser gerado por outro circuito analógico ou gerado digitalmente e convertido para o domínio analógico por um conversor analógico-para-digital (ADC). Na primeira situação, o circuito para geração de um sinal analógico de teste em geral ocupa grande área em silício e assim deve ser também testado. Além disso, requer esforço de projeto, já que o sinal gerado, na maioria das aplicações, deve apresentar baixos níveis de ruído e não-linearidade. No caso de um estímulo de teste digital, apesar da geração ser mais simples e implicar num circuito menor, o resultado do teste dependerá do desempenho do conversor, já que este pode interferir na característica do sinal.

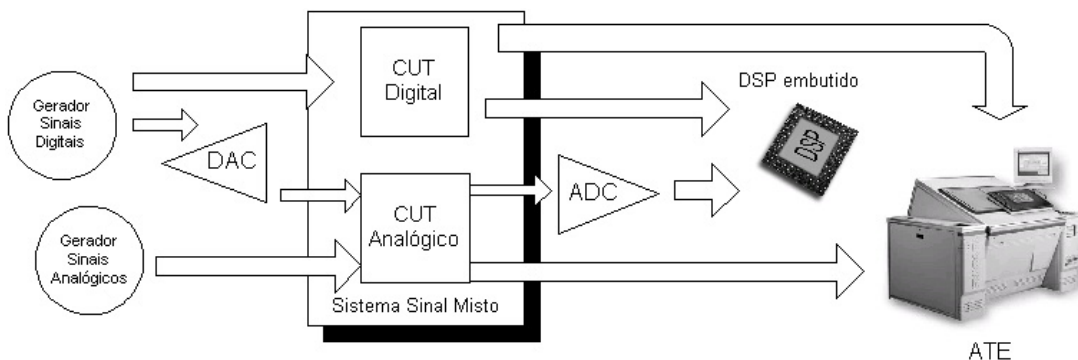


Figura 3 - Diagrama de blocos de teste particionado (analógico e digital) de um sistema misto
Fonte: adaptado de Milor (1998)

Quanto ao modelo de falhas para o teste analógico, existem duas classes de falhas, as catastróficas e as paramétricas (ou *soft*) (MILOR, 1998). As primeiras são aquelas cuja manifestação implica na descaracterização total da funcionalidade do circuito, inutilizando-o, e ocorrem principalmente por problemas na fabricação, devido a curtos-circuitos e circuitos abertos em partes do sistema por contaminação por impurezas ou rompimento de contatos, entre outros. Falhas paramétricas, por sua vez, correspondem a desvios nos valores dos componentes internos do circuito (no caso de um teste estrutural), ou das características de

seu desempenho (para o teste funcional). Elas ocorrem devido a variações de processo, por exemplo, devido a alterações na geometria de um transistor ou resistor, presença de impurezas, temperatura, implantação de íons e outros fatores que influem nas propriedades físico-químicas dos materiais que constituem o circuito. O sistema não perde sua funcionalidade, mas pode não satisfazer a requisitos de desempenho, como o ganho na faixa de passagem de um dado filtro. Segundo Stapper, Armstrong e Saji (1983), cerca de 90% das falhas reportadas na fabricação de circuitos analógicos são catastróficas. Entretanto, devido às variações de processo, nem todos os circuitos fabricados terão o mesmo desempenho, havendo assim a necessidade de se estabelecer limites de tolerância aceitáveis para variações nos componentes e parâmetros de desempenho, a fim de que a fabricação do circuito seja economicamente viável.

Como já citado anteriormente, há diversas classes diferentes de circuitos analógicos, com diferentes arquiteturas, o que dificulta a adoção de uma estratégia padrão, que seja suficientemente abrangente para atender todas estas classes. Com isso, existem técnicas que foram desenvolvidas especificamente para filtros analógicos e conversores de dados, para nomear algumas. No caso de filtros analógicos, devido às aplicações a que se destinam, é importante assegurar o desempenho satisfatório com relação a parâmetros estáticos e dinâmicos (isto é, DC e AC, respectivamente) como ganho na faixa de passagem, atenuação na faixa de rejeição, frequência de corte e fator de qualidade do filtro, justificando a escolha de métodos de teste funcional.

Apesar de haver um grande número de topologias implementando um mesmo tipo de filtro, considerações de facilidade de projeto e de testabilidade fazem com que os filtros projetados, em sua maioria, sejam compostos de estágios de 1ª ou 2ª ordem em cascata, em que cada um destes é testado e caracterizado separadamente. Com isso, torna-se possível também um teste estrutural, em que falhas são injetadas em cada componente, sendo a sua sensibilidade ao teste mais fácil de ser determinada.

No caso de conversores de dados, especificamente analógico-digitais, pode-se dividir os métodos de teste em dois grandes grupos: os métodos de Densidade de Transição de Código (*Code Density Transition*), cujo principal representante é o método do Histograma Linear (DOERNBERG; LEE; HODGES, 1984), que avaliam principalmente as características estáticas, e os métodos de Análise Espectral da saída do conversor, que avaliam

principalmente as características dinâmicas, dentre os quais se destacam os baseados na FFT, listados na Figura 4 (FLORES, 2003). A caracterização estática está associada aos parâmetros da função de transferência do conversor, como erros de não linearidade integral e diferencial (INL e DNL, respectivamente), ganho e *offset*, bem como sua operação em baixas frequências. Já a relação de energia entre o sinal de interesse e o ruído total (conhecida também como SINAD), distorção harmônica total (THD), a faixa dinâmica de operação e a faixa livre de componentes espúrias (SFDR) e a resolução efetiva (ENoB) são parâmetros importantes para avaliação do seu comportamento dinâmico.

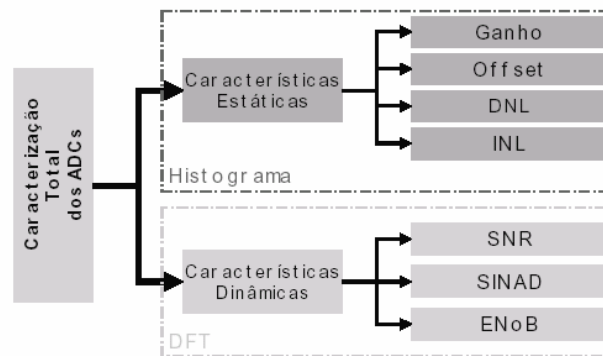


Figura 4 - Caracterização de conversores AD e o tipo de teste associado
Fonte: Flores (2003)

Deve-se ressaltar ainda que, apesar desses métodos serem aplicáveis para a maior parte dos ADC, conversores de alta resolução ou alta velocidade, que apresentam arquiteturas específicas (JOHNS, 1997), requerem métodos específicos (FLORES, 2003).

2.1 AUTO-TESTE INTEGRADO (BIST)

As demandas atuais no mercado de semicondutores incluem a redução dos custos do teste analógico através da integração de estruturas voltadas ao teste na mesma pastilha do sistema, explorando ao máximo a possibilidade de reuso de componentes disponíveis no mesmo sistema. Esta técnica de integração, inserida no contexto de DFT, é conhecida como Auto-Teste Integrado, ou *Built-In Self Test* (BIST) (STEININGER, 2000). Outro aspecto desejável é a possibilidade de teste em funcionamento (*on-line testing*), para utilização em aplicações críticas, como sistemas de controle e de apoio à vida, de modo que a correta

operação do sistema possa ser monitorada pelo próprio durante sua operação, sem necessidade de retirá-lo da aplicação.

O BIST consiste em integrar blocos funcionais de teste antes embutidos no ATE, como o gerador de vetores de teste e o analisador de vetores de resposta ao teste, na mesma pastilha em que se encontra o circuito a ser testado, como mostrado na Figura 5. Com isso, poucos pinos são necessários para observação e controle do procedimento de teste, uma vez que a saída pode ser um único bit, indicando sucesso ou falha. As estruturas de teste embutidas também tornam o ATE necessário para o teste mais simples, conseqüentemente, mais barato. Outra vantagem é o fato de que, uma vez que estas estruturas estão mais próximas do circuito a ser testado, diminui-se sensivelmente a degradação dos sinais de estímulo de teste e de resposta ao teste, o que é comum na transmissão dos sinais entre o CUT e o ATE (MILOR, 1998).

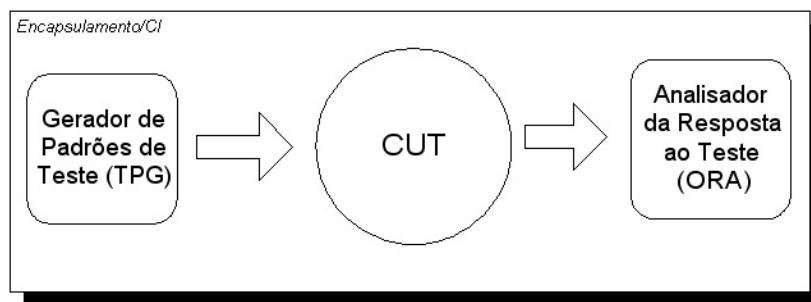


Figura 5 - Diagrama de blocos de BIST em um mesmo encapsulamento

Contudo, a inclusão das estruturas de BIST também implica num acréscimo de área ocupada em silício, em potência dissipada e, inclusive, em degradação do desempenho do circuito durante sua operação normal, a depender do grau de intrusividade das estruturas de teste no CUT. Conseqüentemente, a adoção de BIST deve se inserir num contexto mais abrangente de projeto e requisitos do sistema, a fim de concluir se de fato é vantajosa a inserção dessas estruturas no sistema.

Uma possibilidade investigada de se contornar o problema do espaço ocupado em silício é explorar o reuso de estruturas disponíveis no próprio sistema, a fim de efetuar as tarefas de geração de vetores de teste ou *Test Pattern Generation* (TPG) e análise da resposta ou *Output Response Analysis* (ORA). Este paradigma tem sido utilizado em exemplos com filtros (COTA et al., 2000) e com estruturas analógicas reprogramáveis (BALEN et al., 2004;

PEREIRA et al., 2005), de forma que o acréscimo de hardware se torna muito pequeno, devido a algumas estruturas de controle de teste.

A seguir, serão apresentadas algumas estratégias de BIST disponíveis na literatura e que podem ser aplicadas no contexto do teste de núcleos analógicos em SOCs. Dentre todas, algumas visam certas classes de circuitos analógicos, enquanto outras podem ser aplicadas à grande parte dos circuitos analógicos. Esta última propriedade é mais desejável, pois um método que sirva para testar mais de um tipo de núcleo analógico seria adequado para um ambiente de um SOC, sendo reusado entre os diversos núcleos existentes no sistema. Com isso, espera-se obter um conjunto de informações suficientes que contribua para decisão sobre qual estratégia, ou combinação destas, usar em um dado sistema com diversos núcleos analógicos a serem testados.

2.1.1 BIST de ADCs usando histograma linear

O método do Histograma linear é um método clássico adotado pela indústria, descrito em Doernberg, Lee e Hodges (1984). O procedimento consiste na aplicação de um sinal de estímulo de teste à entrada do conversor e conseqüente contagem do número de ocorrências de cada código de saída. O sinal de estímulo deve ser um cuja função densidade de probabilidade seja conhecida, de forma que a distribuição das ocorrências dos códigos de saída seja o mais próxima possível daquela, sendo que os eventuais desvios podem indicar falhas na operação do dispositivo.

Uma implementação viável deste método em BIST é descrita em Azaïs et al. (2001) e Renovell et al. (2000), propondo otimizações em hardware que afetam desde a escolha do tipo do sinal de entrada até o cálculo dos parâmetros. Um sinal de estímulo triangular é mais adequado que um sinal senoidal, pois permite que a assinatura dos circuitos sem falhas possa ser armazenada usando somente duas palavras de código, sendo uma para os códigos relativos aos pontos extremos do sinal e outra para os códigos restantes (AZAÏS et al., 2001; ALEGRIA et al., 2001). Outra implicação é a simplificação do cálculo dos parâmetros estáticos do conversor, usando apenas operações aritméticas de soma, subtração, multiplicação e divisão, o que dispensa o uso de processadores mais complexos. Ademais, também é realizada uma modificação no procedimento de cálculo do histograma medido,

segundo uma Técnica de Decomposição no Tempo, de modo que o histograma e os parâmetros são avaliados código a código e os recursos de armazenamento dos resultados do histograma e do processamento são economizados.

Enfim, o protótipo do circuito de teste, com todas as modificações citadas, ocupou uma área em silício de cerca de 7% do circuito sob teste. Entretanto, devido às mesmas modificações, o tempo de teste é prejudicado, chegando a atingir 20 minutos para o teste de um conversor de 14 bits a uma frequência de amostragem de 1MHz. Por isso, para um tempo de teste aceitável, esta técnica se limita para aplicações em conversores de baixa resolução, até 12 bits. Apesar de tudo, a técnica descrita acima se mostra atraente para o teste das características estáticas do conversor.

2.1.2 BIST baseado em FFT

A caracterização dinâmica do CUT é realizada, em geral, através de uma análise espectral (FFT) dos vetores de resposta ao teste, obtendo a resposta em frequência do CUT a um sinal de estímulo senoidal. Com isso, parâmetros de desempenho importantes como SINAD, ruído de fundo (*noise floor*), THD e SFDR podem ser determinados, como mostrado por Peetz (1983).

Os métodos baseados em FFT geralmente são mais rápidos do que os baseados em medidas estáticas. Entretanto, alguns tipos de erros, como códigos inexistentes (*missing codes*), críticos para aplicações de controle, são identificados apenas como ruído para testes de FFT, razão pela qual alguns conversores precisam dos dois procedimentos para serem testados.

Pode-se afirmar que, caso todas as componentes harmônicas e espúrias estejam $6N$ dB abaixo do nível relativo à frequência fundamental do sinal de teste, sendo N o número de bits do conversor, então o dispositivo conversor tem operação satisfatória, uma vez que cada componente de frequência errônea contribui com menos de 1 LSB para o erro total do dispositivo (LECHNER et al., 2004). Entretanto, caso este nível seja ultrapassado, há a necessidade de se realizar procedimentos mais específicos para determinação de erros de não linearidade, como o teste do histograma, o que implica na realização de dois procedimentos distintos e aumenta a área ocupada em silício das estruturas embutidas de teste, se ambos

forem realizados simultaneamente. Para eliminar esse inconveniente, buscaram-se maneiras de realizar ambos os testes através de um único experimento, de forma que características estáticas possam ser extraídas a partir do resultado da caracterização dinâmica, e vice-versa.

A primeira hipótese é verificada em Azaïs et al. (2002), aplicando um estímulo senoidal com uma única componente de frequência e observando os desvios nas estimativas de parâmetros dinâmicos, como SINAD e THD. Resultados experimentais permitem observar a influência de erros estáticos tão pequenos quanto 0,2 LSB na estimativa de parâmetros dinâmicos, caso a amplitude do sinal de estímulo seja maior do que a escala de entrada do dispositivo.

Entretanto, para uma integração total do teste ao sistema, é necessário um núcleo para geração do estímulo de teste, isto é, um sinal senoidal com a mínima distorção possível e com precisão suficiente para não comprometer a interpretação dos resultados do teste, o que tem sido alvo de outros estudos (DUFORT; ROBERTS, 1997; BERNARD et al., 2002), haja vista a dificuldade associada ao projeto de tais blocos. Ademais, precisa-se ainda de um circuito para a estimativa da FFT e outro para a extração dos parâmetros de teste desejados e sua avaliação, podendo ambos serem efetuados através de um processador disponível. Finalmente, é necessário mais um núcleo para o controle das fases do teste. O número de amostras deve ser suficiente para diminuir a incerteza dos parâmetros estimados, e o número de períodos deve respeitar a condição de amostragem coerente. Para um conversor de seis bits, foi adotado um número de 1024 amostras e 31 períodos de observação, de forma a garantir baixo custo em termos de tempo de teste. Ainda, o tamanho da memória para armazenar os resultados é também minimizado, diminuindo ainda mais o hardware necessário.

Entretanto, a maior dificuldade reside no gerador de estímulos, que tem requisitos de desempenho rígidos e que ocupa grande área em silício. Esta alternativa pode se tornar viável se houver um número maior de blocos que necessitam de vetores de teste analógicos e a implementação de um gerador de formas de onda arbitrárias (*Arbitrary Waveform Generator*, ou AWG) for possível e permitir compartilhamento entre todos os núcleos sob teste.

2.1.3 BIST baseado em ruído

Outra proposta que visa diminuir o custo de teste usando um único procedimento baseado em análise espectral para caracterização estática e dinâmica é descrito em Flores (2003). Ele é baseado em um modelo de alto nível para ADCs, idealizado por Czismadia e Janssen (1999), em que o conversor é representado por um bloco que realiza uma dada função de transferência $h(x)$, a cuja saída é adicionado um ruído de banda larga, inerente ao dispositivo, seguido da operação de quantização e codificação para o domínio digital. A Figura 6 ilustra esse modelo segundo um diagrama de blocos. Idealmente, $h(x)$ deve ser uma função linear com inclinação unitária, porém não-linearidades presentes podem ser modeladas como termos de maior ordem. A análise espectral do sinal de saída do conversor, estimada através da transformada de Fourier ou outro cálculo numérico, permite uma estimativa da curva $h(x)$, que pode ser escrita em função das componentes espectrais assim obtidas, conforme a Equação (1). Como estímulo, é usado um sinal senoidal com amplitude, frequência e *offset* conhecidos. T_n é o polinômio de Chebyshev do primeiro tipo, A é a amplitude do sinal de estímulo e L é o maior grau polinomial que se deseja modelar em $h(x)$.

$$\tilde{h}(x) = X(0) + 2 \sum_{n=1}^L i^n X(n) T_n(x/A) \quad (1)$$

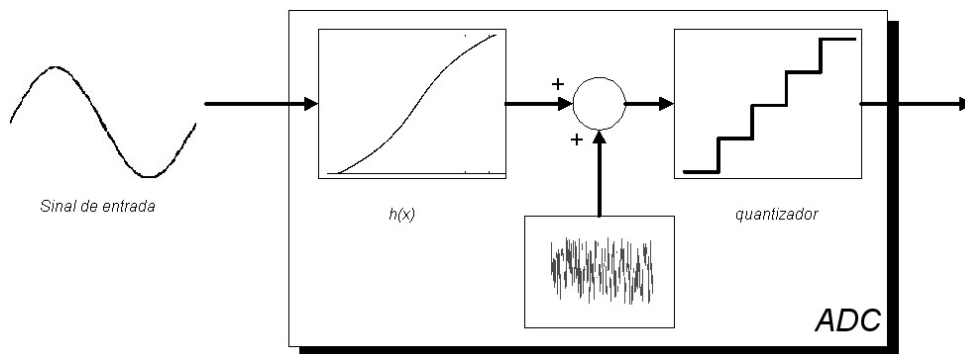


Figura 6 - Diagrama de blocos do modelo de ADC proposto por Czismadia, Janssen (1999)

Em outras palavras, a partir da análise espectral da saída do conversor excitada por um sinal de banda larga, pode-se obter uma aproximação da função de transferência real do conversor, da qual são facilmente extraídos os parâmetros estáticos de offset, ganho e não-linearidade. Os parâmetros dinâmicos são obtidos utilizando a análise tradicional por FFT.

A implementação BIST desta estratégia inclui um gerador de ruído de banda larga, através da amostragem pseudo-aleatória de uma curva de carga e descarga de um circuito RC (FLORES et al., 2002), e um analisador digital da resposta, que pode também ser realizado reusando processadores e DSPs embarcados no sistema (FLORES, 2003). O acréscimo em área reportado após a fabricação do circuito foi de 7% em relação ao conversor sob teste. O consumo de potência é dado pelo circuito RC e um *sample-and-hold*, mais circuito digital de controle e de análise da resposta ao teste. O tempo de teste diminui bastante em relação ao método do histograma convencional, pois o número de amostras necessárias cai de mais de um milhão (no caso de amostras aleatórias) e 64000 (para amostras determinísticas) para apenas 8192.

A idéia de usar ruído como estímulo de teste para caracterização de circuitos analógicos pode ser usada para outras classes, como filtros. Neste trabalho, a resposta do CUT ao teste é comparada com ruído de banda larga, obtendo assim um sinal que pode ser representado digitalmente. Essa comparação é realizada através de um comparador analógico com especificações de desempenho pouco restritivas, que é referido em Negreiros (2002) pelo nome *statistical sampler*. Papoulis (1991) mostra que, apesar do sinal comparado não poder ser recuperado, sua estatística se mantém na saída do comparador, de modo que grandezas como a autocorrelação e a Distribuição de Potência Espectral (PSD) do sinal obtido são proporcionais ao do sinal original, segundo a lei do arco-seno, descrita na Equação (2), e pode então ser facilmente processada.

$$R_y(\tau) = \frac{2}{\pi} \arcsin\left(\frac{R_x(\tau)}{R_x(0)}\right) \quad (2)$$

Alguns trabalhos mostram exemplos em que até um circuito digital dedicado de tamanho razoável pode ser empregado para estimar a função de transferência de um filtro (SOUZA JR; CARRO, 2003), além de alguns parâmetros dinâmicos específicos como a THD (ANDRADE JR.; CARRO; LUBA, 2004a).

2.1.4 Autoteste baseado em oscilação (OBIST)

Esta técnica foi introduzida inicialmente por Arabi e Kaminska (1997), e baseia-se na modificação do circuito sob teste, de forma que em sua nova configuração ele atue como um

oscilador. A partir da frequência de sua saída oscilatória, seria possível extrair conclusões sobre falhas em suas estruturas. A oscilação poderia ser obtida acrescentando um laço de realimentação no circuito de modo que, por exemplo, os pólos da função de transferência de um filtro sob teste fossem deslocados para o eixo imaginário, caracterizando um oscilador. Tal técnica foi demonstrada para circuitos como filtros e conversores, porém não apresentava cobertura de falhas satisfatória. Em seguida, Huertas et al. (2002) propõem que um aumento na cobertura de falhas pode ser alcançado se outros parâmetros da resposta oscilatória serem extraídos, como a amplitude da oscilação. Para isso, não basta apenas fazer o circuito oscilar, mas garantir que a oscilação seja controlada e sustentada (HUERTAS, 2004). Huertas ainda propõe a implementação de um laço de realimentação não-linear, de forma que a metodologia pode ser aplicada a um número maior de classes de circuitos. A Figura 7 resume os pontos discutidos neste parágrafo.

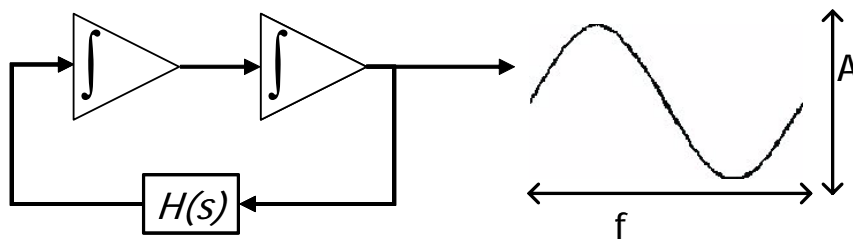


Figura 7 - Implementação da estratégia de teste baseada em oscilação, avaliando frequência e amplitude do sinal gerado

A análise da estratégia indica que a cobertura de falhas obtida então se torna satisfatória, e que o fato de não ser necessário um gerador de estímulos para aplicação do teste a tornam uma estratégia interessante para redução de custos de teste. Entretanto, como o circuito é reconfigurado na fase de teste, o que implica, no mínimo, em acrescentar algumas chaves, o desempenho do circuito é degradado. Outra consideração é a implementação de uma unidade de avaliação da resposta oscilatória que possa ser integrada ao sistema ocupando um mínimo de área em silício. A proposta de Huertas (2004) traz a idéia de se usar um modulador sigma-delta para transformar a resposta de teste em um *bitstream* que pode ser processado por um circuito digital extra (Figura 8), que consome cerca de 7% do circuito sob teste para o caso de um filtro *Dual-Tone Multi-Frequency* (DTMF), utilizado em aplicações de telecomunicações. Junto com as chaves e outras estruturas que implementam o laço de realimentação, o acréscimo total de área pode se tornar razoavelmente alto para integração em

sistemas. A cobertura de falhas fica entre 98 e 100%, bastante satisfatória, dado que sejam medidas tanto a amplitude como a frequência.

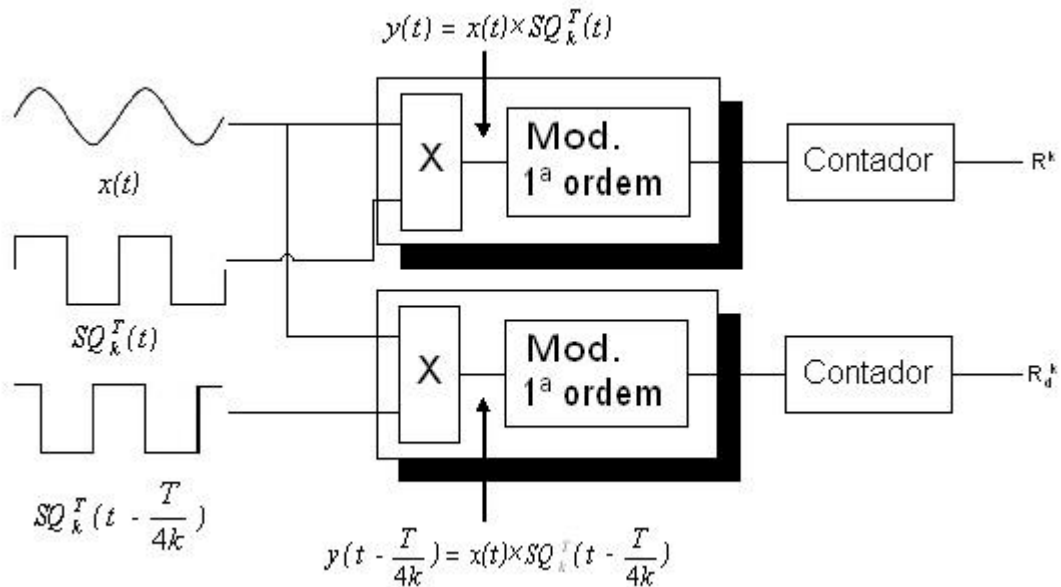


Figura 8 - Diagrama de blocos do circuito de codificação sigma-delta para a resposta ao teste baseado em oscilação

Fonte: Vazquez et al. (2003)

A estratégia OBIST modificada, então, consiste na implementação de um laço de realimentação através de um bloco não-linear, de forma que os pólos da função de transferência realizada pelo filtro sejam colocados no eixo imaginário, configurando um sistema oscilatório, e outras implicações práticas, como manutenção e inicialização rápida da oscilação, sejam garantidas. Ainda em Huertas (2004), o amplificador operacional é reprojetoado, a fim de satisfazer esses requisitos e promover a configurabilidade requerida para o método. Simulações realizadas mostram que a amplitude e a frequência da oscilação apresentam boa sensibilidade em relação a desvios nos valores dos componentes internos. Entretanto, apesar da sensibilidade indicar uma boa cobertura de falhas, tal estratégia depende da arquitetura do circuito analógico a ser testado. Além disso, a modificação e inserção de estruturas geralmente afetam o desempenho do circuito quando em operação normal. Os experimentos realizados consideraram um conversor baseado em modulador sigma-delta, pois tal arquitetura favorece o surgimento de oscilações.

Em termos de tempo de teste, pode-se dizer que quanto maior o número de ciclos de oscilação avaliados, maior a precisão do resultado obtido com trem de pulsos gerado pelo modulador sigma-delta. Em Vazquez et al. (2003), o sinal de interesse é multiplicado por duas ondas quadradas, defasadas de 90° , separadamente, de forma que dois números são obtidos. Através de operações matemáticas realizadas sobre estes resultados, indicados no mesmo trabalho, consegue-se uma boa estimativa não só da amplitude e frequência do sinal, mas também de componentes harmônicas de 2ª e 3ª ordem. Entretanto, na maioria das situações, a amplitude e frequência do sinal já são medidas satisfatórias. Assim, 30 ciclos de oscilação são necessários para uma medida com intervalo de tolerância adequado.

Em resumo, a presente estratégia persegue quatro objetivos: geração de estímulo no chip (na verdade, não há necessidade de vetores de estímulo), controle interno das fases de teste (um controlador que determine as configurações do circuito sob teste e em funcionamento normal, por exemplo), resposta ao teste codificada digitalmente junto com um sistema de interpretação de resposta também interno ao sistema.

A estratégia baseada em oscilação foi experimentada também em filtros implementados em capacitor chaveado. É considerado um modelo de falhas paramétricas nos componentes passivos do circuito, isto é, desvios percentuais de 5 a 50% nos valores dos capacitores e resistores que o compõem. Seja qual for a ordem do filtro em questão, este deve ser projetado de forma que seja composto de uma cascata de filtros de primeira e segunda ordem, que serão testados seqüencialmente. Em outras palavras, a estratégia baseada em oscilação requer também que algumas regras de DfT sejam adotadas. Embora o procedimento tenha sido validado para este tipo de implementação, obtendo alta cobertura de falhas, entre 98 e 100%, o mesmo pode ser estendido a filtros de tempo contínuo.

2.1.5 BIST baseado na análise da resposta transiente

Baseadas no fato de que técnicas estáticas são mais simples e menos caras do que as dinâmicas e podem ser tão ou mais eficientes na detecção de falhas catastróficas, alguns trabalhos, como Dufaza e His (1997) e Chatterjee, Kim e Nagi (1996), propõem uma mudança de paradigma, aplicando teste estático ao invés de um dinâmico. Uma variante desta técnica é a Análise da Resposta Transiente, ou TRAM (CALVANO; ALVES; LUBASZEWSKI,

2000), que consiste na excitação do CUT através de vetores compactados de teste, consistindo em formas de onda do tipo degrau, rampa ou parábola, e avaliação de parâmetros da resposta transiente, como o valor máximo atingido (*overshoot*) e tempo de pico, comparando-os com os valores associados a uma condição sem falhas. Os parâmetros medidos são o ganho na faixa de passagem, frequência de corte ω_C e fator de qualidade (Q) de filtros de segunda ordem segundo as aproximações de Chebyshev, Bessel e Butterworth, formando uma assinatura sem falhas. Dessa forma, falhas são detectadas quando os parâmetros medidos durante o teste diferem da assinatura sem falhas. Filtros dessa ordem têm expressões matemáticas analiticamente demonstradas em função destes mesmos parâmetros de desempenho. Apesar de basear-se em estruturas de filtros de 2ª ordem, a estratégia não é restritiva, tendo em vista que, atualmente, o projeto de filtros de altas ordens segue o modelo de cascata de filtros de segunda ordem. Além disso, mesmo sendo um teste funcional, mostra-se que falhas injetadas segundo o modelo estrutural também podem ser detectadas.

O circuito de BIST inclui um integrador, uma chave analógica (para geração dos sinais de estímulo), um detector de cruzamento por zero, e um contador para avaliação dos parâmetros da resposta transiente (Figura 9). Esta estrutura se baseia no fato do filtro ser um sistema linear e invariante no tempo, logo a derivada da resposta deste sistema a um dado estímulo é igual à resposta do mesmo sistema a um sinal equivalente à derivada do mesmo estímulo. Como consequência, um contador é usado para estimar o tempo que leva para o detector acusar o cruzamento por zero, indicando que a função original chegaria ao seu valor máximo. O integrador à entrada serve para gerar, em duas fases, o sinal original e o sinal de teste, resultado da integração do sinal original. Assim, este representa a derivada do sinal de teste, sendo usado para determinação do tempo de pico do filtro.

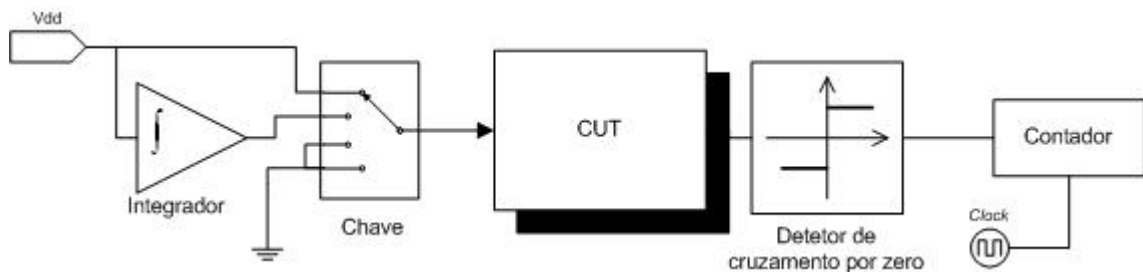


Figura 9 - Diagrama de blocos para BIST baseado em TRAM
Fonte: Calvano; Alves; Lubaszewski (2000)

Observando a metodologia, pode-se concluir que o acréscimo de área devido às estruturas tanto de avaliação da resposta como de geração de vetores é pequeno, já que a avaliação é realizada por um contador e um detector de cruzamento por zero. O tempo de teste pode ser avaliado como duas vezes o tempo que o circuito leva para alcançar seu valor máximo, já que num primeiro momento, o tempo de pico é avaliado para em seguida ser extraído o valor máximo. Segundo as equações descritas em Ogata (1982), o tempo de pico é função da frequência natural de oscilação do sistema, conforme a Equação (3). Através desta, pode-se observar que o tempo de pico depende do fator de amortecimento ζ do sistema em questão, mas que nunca pode ultrapassar metade do período de oscilação natural do mesmo. Conseqüentemente, o tempo de teste desta estratégia equivale a no máximo, um ciclo da oscilação natural do CUT. Além disso, deve-se levar em conta o tempo necessário para avaliação de ω_c e Q . Estas, por envolverem operações mais complexas, como raízes e logaritmos, podem requerer um maior tempo e poder de processamento.

$$T_p = \frac{\pi}{\sqrt{1-\zeta^2}\omega_n} = \frac{1}{\sqrt{1-\zeta^2}} \frac{T_n}{2} \quad (3)$$

Enfim, a Tabela 1 resume qualitativamente as técnicas discutidas acima, respectivamente, para ADCs e filtros analógicos. Para uma comparação mais acurada e quantitativa, pode-se efetuar simulações e outros experimentos de forma a calibrar uma futura ferramenta de apoio à decisão de testabilidade baseada em tais dados, por exemplo.

A potência consumida, por exemplo, é estimada a partir do acréscimo de estruturas de teste e do tipo, isto é, se é funcional ou estrutural. Em testes funcionais, a potência dissipada é praticamente a mesma que na operação normal do circuito, excluindo-se os circuitos adicionados, já que o procedimento de teste consiste em caracterizar o circuito durante seu funcionamento. Já no teste estrutural, toda a topologia do circuito é acionada em um curto espaço de tempo, e assim a potência dissipada é maior do que aquela durante o funcionamento do mesmo. Ademais, quanto maior o número de estruturas presentes, maior será a potência dissipada, razão pela qual a avaliação qualitativa também se baseia no acréscimo de área devido à estratégia.

Tabela 1 - Comparação entre métodos de BIST para ADC e filtros

	Histograma linear	Baseado em ruído	Baseado em oscilação	FFT	TRAM
Tipo	Funcional	Funcional	Estrutural	Funcional	Funcional
Arquiteturas-alvo	ADC de Baixa e Média Resolução	Independente	Conversores, OpAmps, Filtros	Independente	Filtros
Tempo de Teste	Alto	Baixo	Muito Baixo	Baixo	Baixo
Overhead de Área	Pequeno	Pequeno	Alto	Alto	Pequeno
Cobertura de Falhas	Média	Alta	Alta	Alta	Alta
Potência Consumida	Baixa	Baixa	Alta	Alta	Pequena
Estímulos de Teste	Onda Senoidal ou Triangular	Ruído	Sem Vetores de Teste	Sinal Senoidal de Precisão	Pulso e Rampa de Tensão
Interpretação da Resposta	Embutido	Externo/Reuso	Circuito Digital para Extração de Parâmetros	Externo/Reuso	Cruzamento por zero e contador externo
Teste <i>On-Line</i>	Não	Sim	Não	Sim	Não
Degradação de Desempenho	Não	Não	Sim	Não	Não

3 ESTADO-DA-ARTE EM TESTE DE SOC

3.1 PROJETO DE SISTEMAS BASEADOS NO REUSO DE NÚCLEOS DE HARDWARE (SOC)

Como já mencionado no capítulo 1, a evolução nos processos de fabricação de circuitos integrados permite a integração de sistemas completos em uma única pastilha. Dentre as mudanças no paradigma, pode-se citar a criação de dois grupos distintos na comunidade de projetos de sistemas baseados em núcleos de hardware: os provedores de blocos IP pré-projetados e os usuários destes núcleos, ou integradores de sistemas. Cabe então ao último buscar entre os provedores de tais blocos no mercado aquele que satisfaz suas necessidades de projeto e integração, e se concentrar nos aspectos de alto nível do sistema, como protocolos de comunicação, temporização e compatibilidade elétrica entre os núcleos.

Uma das vantagens associadas ao sucesso do paradigma do SOC é a possibilidade de reuso de blocos IP, bem como sua parametrização e padronização de interfaces. Com a crescente complexidade dos sistemas, contendo milhões de transistores, o projeto baseado no reuso de estruturas pré-projetadas e pré-verificadas permite superar as pressões do *time-to-market* ao mesmo tempo em que o compromisso com o desempenho dos sistemas é mantido (BERGAMASCHI; LEE, 2000). Além disso, há um grande número de sistemas que se utilizam dos mesmos circuitos em sua arquitetura, como conversores, microprocessadores, memórias, o que favorece a estratégia de reuso.

Os núcleos de hardware são disponibilizados pelos fornecedores de blocos IP em três níveis diferentes de abstração (KUCUKCAKAR, 1998), fazendo um compromisso entre flexibilidade de projeto durante a integração e previsibilidade do desempenho do circuito em termos de atraso de sinais, capacitâncias parasitas e outros parâmetros, obtidos a partir do leiaute fixo (Figura 10).

Os núcleos *soft* são fornecidos na forma de linguagens de descrição de alto nível, como *Register Transfer Level* (RTL), sendo o integrador do sistema responsável num estágio posterior pela síntese física e leiaute. Assim, lhe é permitido uma maior flexibilidade de integração do núcleo ao resto do sistema, mantendo a proteção à propriedade intelectual do fornecedor do núcleo. Os núcleos *firm* são disponibilizados em um estágio pós-síntese, isto é,

após síntese lógica e mapeamento de tecnologia, mas ainda sem leiaute físico. Finalmente, núcleos *hard* são fornecidos como caixas-pretas, em geral já com leiaute pré-definido.

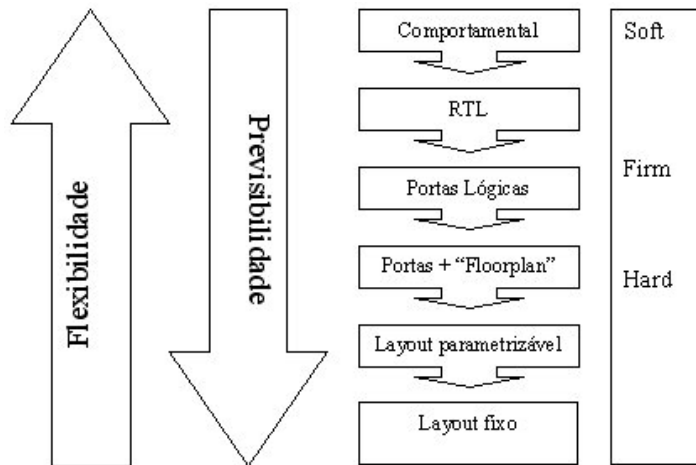


Figura 10 - Classificação dos blocos IP quanto ao nível de abstração e flexibilidade
Fonte: Kucukcakar (1998)

Adicionalmente, o integrador de sistemas pode incluir no SOC uma lógica personalizada para promover outras funcionalidades ao sistema, como compatibilidade entre núcleos e protocolos de comunicação. Tais circuitos são conhecidos como Lógica Definida pelo Usuário ou *User Defined Logic* (UDL). Ainda, núcleos mais complexos podem ser criados a partir de outros núcleos disponíveis e UDL, formando uma hierarquia, conforme ilustrado na Figura 11.

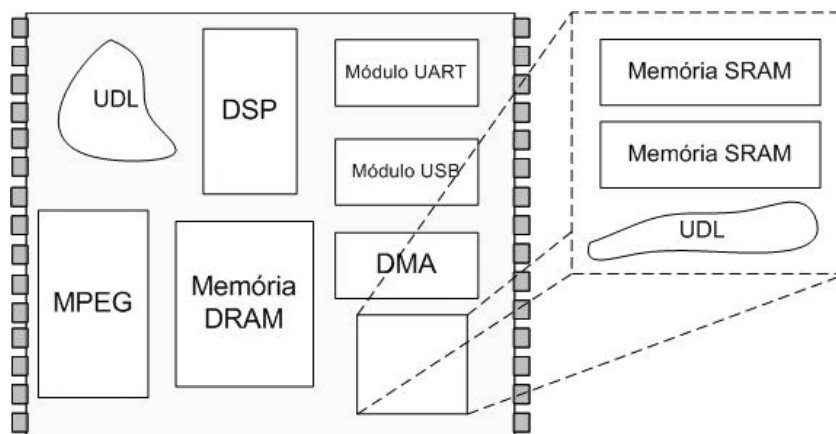


Figura 11 - Hierarquia entre núcleos e supernúcleos em um SOC
Fonte: Cota (2003)

Mesmo com os avanços alcançados com o advento do projeto baseado em reuso, segundo Bergamaschi e Lee (2000), este ainda é um processo que envolve muito esforço por parte do integrador, principalmente manual, e assim suscetível a erros e ocupando tempo que atrasa a disponibilização do produto no mercado. Por isso, muitas linhas de pesquisa são dedicadas ao desenvolvimento e aperfeiçoamento de metodologias de projeto e ferramentas de automação, com o intuito de reduzir tais custos associados ao projeto.

3.2 TESTE DE SOCS

Comparando o fluxo de projeto em alto nível para SOBs e SOCs, ilustrado na Figura 12, observa-se que, no primeiro, o projeto da Placa de Circuito Impresso (PCI) é precedido pela fabricação e teste dos componentes, principalmente CIs, que a compõem.

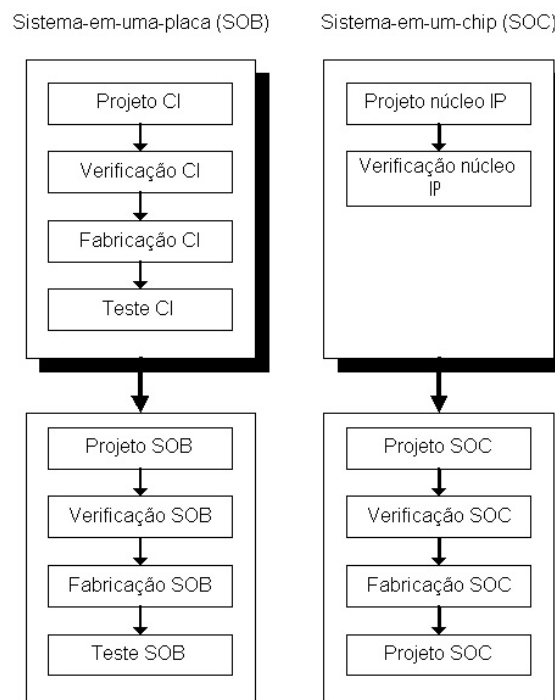


Figura 12 - Comparação do fluxo de projeto para: a) SOB; b) SOC

Por outro lado, em SOCs, os componentes do sistema, isto é, os núcleos, não estão fisicamente fabricados, apenas formalmente verificados. Com isso, o seu teste será realizado quando o sistema completo estiver manufaturado em silício. Isso traz dificuldades no acesso aos núcleos, uma vez que as PCIs disponibilizam pontos para inserção de pontas de prova

para observabilidade de sinais de teste, ao contrário dos núcleos embutidos em um SOC, que só podem ser acessados a partir dos pinos de entrada e saída do encapsulamento. Ainda, com o aumento da integração nestes sistemas, o número de núcleos tende a crescer, e mantendo a mesma limitação nos pinos, a acessibilidade aos núcleos tende a se tornar crítica. O custo de teste do sistema aumenta, seja pelo crescimento do tempo de teste do sistema como um todo, devido à serialização no acesso da interface para o teste, ou pelo aumento do número de pinos necessários para atender a todos os blocos.

Apesar das vantagens do ponto de vista de projeto, o teste destes sistemas representa ainda um desafio a ser vencido. Uma vez que a dificuldade e os custos do teste de SOCs têm crescido nos últimos anos, a estes tem sido reservada uma atenção especial por parte da indústria e da comunidade acadêmica. Os custos de teste de SOCs, especialmente os que contêm grande número de núcleos, podem ser divididos entre duas fontes principais (SEHGAL; IYENGAR; CHAKRABARTY, 2004):

- a) custos explícitos: referem-se aos custos de investimentos em novos Equipamentos de Teste Automáticos (ATE), também considerados como custos de capital. Eles se tornam maiores à medida que novas tecnologias e circuitos mais específicos são agregados ao paradigma dos sistemas baseados em núcleos, como circuitos analógicos e de RF. Conseqüentemente, fabricantes de circuitos integrados periodicamente têm de renovar seus ATEs, que são sistemas de alto custo para aquisição;
- b) custos implícitos: considerando que o ATE satisfaz as necessidades tecnológicas do teste, entram nessa classificação os recursos de tempo disponíveis para se assegurar uma cobertura de falhas satisfatória, o que aumenta o *time-to-market*, diminuindo a margem de lucro do fabricante.

A redução dos custos explícitos implica na adequação de equipamentos de teste já existentes aos SOCs mais novos e com mais funcionalidades, através de técnicas como compactação dos dados da resposta ao teste, compactação da própria resposta ao teste e redução do número de pinos para o teste. Contudo, os requisitos de teste e a funcionalidade dos sistemas têm apresentado cada vez mais complexidade, de modo que maiores esforços têm sido dispendidos na redução dos custos implícitos, através da utilização de forma eficiente dos recursos de teste empregados no ATE e no sistema a ser testado. Por exemplo, o

número de bits observados na resposta ao teste pode ser minimizado respeitando-se as restrições dadas pelo integrador do sistema de forma que um único ATE seja capaz de testar mais de um chip ao mesmo tempo, referido como *multi-site testing* (IYENGAR et al., 2002c). Ainda, conceitos como o projeto visando o teste e DfT podem ser aplicados aos SOCs e otimização para escolha de pontos de controle e observação específicos do circuito.

Neste contexto, um dos paradigmas mais aceitos atualmente é o teste modular (ZORIAN; MARINISSEN; DEY, 1999). Cada núcleo é testado isoladamente, e o sistema é considerado apto para funcionamento se todos os núcleos passarem no teste. Com isso, o acesso e a aplicação do teste tornam-se atrativamente mais simples. Entretanto, considerações sobre o sistema e sobre as interconexões devem ser levadas em conta, assim como seus requisitos.

Para o teste modular dos núcleos, é necessário definir, além do método de teste a ser aplicado, como será o acesso a cada núcleo e como isolá-lo do resto do sistema durante o teste. Além destes, uma solução completa requer a definição da utilização dos canais do ATE, associando-os aos pinos do sistema sob teste, levando em conta as limitações de memória daquele, bem como a definição do escalonamento de teste dos diversos núcleos. Neste caso, o paralelismo do teste dos núcleos internos é explorado, respeitando limitações de potência consumida e conflito de recursos de teste, por exemplo. Na literatura (IYENGAR; CHAKRABARTY; MARINISSEN, 2002a), estes problemas aparecem juntos numa estratégia de otimização de recursos de teste (*Test Resource Optimization*), sendo que grande parte são do tipo NP-completo (IYENGAR; CHAKRABARTY, 2002b; CHAKRABARTY, 1999). Assim, eles são computacionalmente inviáveis de serem solucionados através de algoritmos usando métodos determinísticos. Para obter soluções próximas de mínimos locais em um tempo razoável de simulação, várias heurísticas foram elaboradas, apresentando resultados satisfatórios (CHAKRABARTY, 1999; CHAKRABARTY, 2000; CHOU; SALUJA; AGRAWAL, 1997).

Para acesso dos estímulos de teste da interface ao CUT, e dos vetores de resposta deste à interface, é necessária a definição de caminhos entre estas duas partes, que são chamados de Mecanismos de Acesso de Teste, ou *Test Access Mechanisms* (TAM). Estas estruturas podem reutilizar conexões operacionais já existentes entre os núcleos e a interface ou utilizar conexões adicionadas durante o projeto. Ainda, o CUT deve ser isolado do resto do sistema

durante o seu teste, para que conexões entre as partes não invalidem o resultado do teste, sendo assim necessária também a definição de *wrappers*. Estes são circuitos que permitem a conexão do núcleo com o sistema durante a operação normal e isolamento elétrico durante o teste. Tais estruturas estão ilustradas na Figura 13.

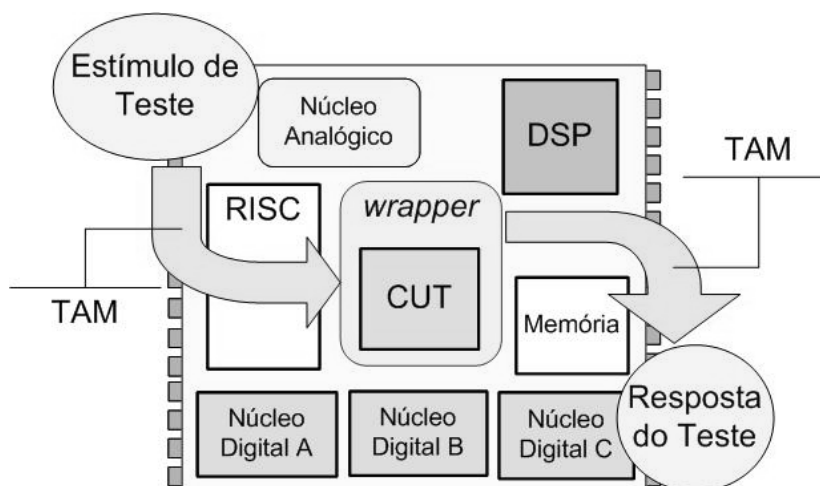


Figura 13 - Arquitetura e estruturas de teste de um SOC
 Fonte: adaptado de Zorian, Dey e Rodgers (2000)

Através da Figura 13, percebe-se, entre outras coisas, que tanto as TAMs quanto os *wrappers* são responsáveis por uma parcela do acréscimo em área de silício devido às estruturas de teste. Conseqüentemente, um dos objetivos de pesquisas recentes é a otimização do uso de tais recursos, explorando possíveis paralelismos no uso, de modo a diminuir o número de caminhos condutores acrescentados, sem implicar em aumento do tempo de teste ou volume de dados de teste.

A respeito das TAMs e dos *wrappers*, diversas arquiteturas foram propostas, sendo citadas em Cota et al. (2002) e mostradas na Figura 14, sendo referido como Modelo Multi-TAM, terminologia adotada ao longo deste trabalho. As conexões possíveis são, a saber:

- a) acesso externo direto dos controladores de teste (Figura 14a);
- b) uso de modos transparentes disponíveis no núcleo vizinho, ou seja, a funcionalidade do núcleo é utilizada apenas para transporte do sinal, sem efetuar qualquer espécie de alteração ou processamento (Figura 14b);

- c) reuso das conexões e pinos funcionais disponíveis no *wrapper* do núcleo seguinte no caminho da TAM (Figura 14c);
- d) uso de *bypass* paralelo (Figura 14d) e serial (Figura 14e) no próximo núcleo do caminho da TAM, segundo o padrão P1500 (IEEE STANDARDS BOARD, 2005);
- e) uso da funcionabilidade do núcleo vizinho como gerador de estímulo de teste ou analisador da resposta;
- f) finalmente, qualquer núcleo pode conectar-se diretamente ao controlador de teste, seja externo ou incluso, através de um barramento exclusivo.

A escolha do método de teste depende da topologia do circuito do núcleo, se o mesmo é digital ou analógico, bem como a função e a aplicação a que se destina (teste de produção, caracterização, entre outros). Além destes, tal definição também está ligada à tecnologia-alvo do sistema e aos recursos disponíveis. Devido à proteção intelectual dos blocos IP, as informações referentes à arquitetura e detalhes de funcionamento do circuito não são repassadas ao integrador, ficando a cargo do provedor de blocos IP a disponibilização de conjuntos de vetores de teste e respostas esperadas, além de estratégias de DfT voltadas ao núcleo, de forma que o integrador possa escolher o que melhor se adequa ao sistema. No caso dos requisitos de teste exigirem mais informações sobre o funcionamento do núcleo ou até mesmo alterações na sua arquitetura, é necessária uma comunicação eficiente e livre de ambigüidade entre o provedor de núcleos e o integrador de sistemas.

Uma vez definidos os métodos de teste a serem empregados e a estrutura física para aplicação dos mesmos, falta resolver o problema de escalonamento do teste dos núcleos, respeitando as condições de conflito de recursos usados e potência consumida durante o teste. Um exemplo de escalonamento pode ser visto na Figura 15. A potência dissipada em um SOC durante o teste pode ser muito maior do que a dissipada em condições normais de operação, já que núcleos que normalmente não operariam ao mesmo tempo podem ser testados concorrentemente devido ao escalonamento (ZORIAN, 1993).

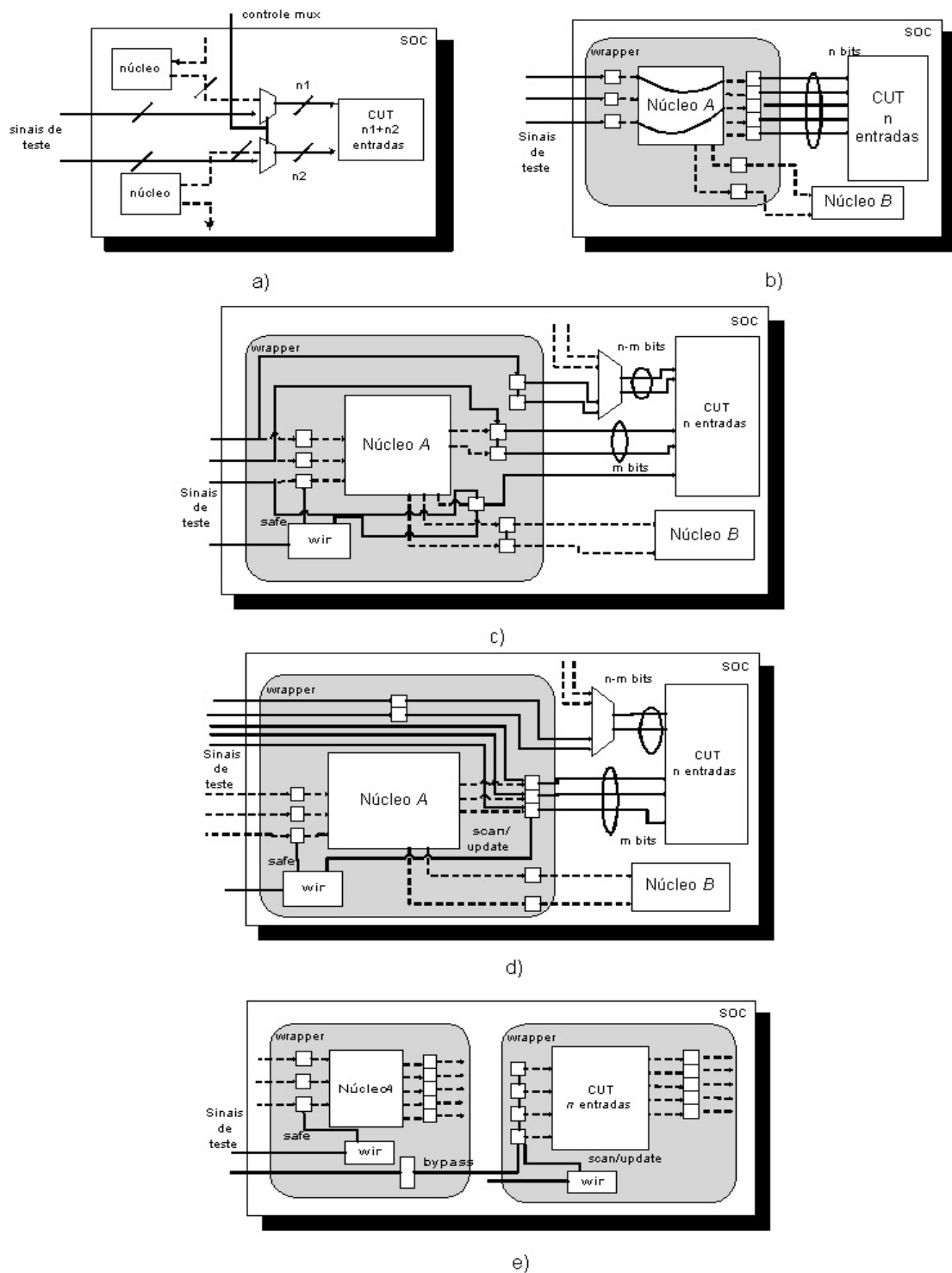


Figura 14 - Topologias de acesso para o Modelo Multi-TAM: a) externo direto; b) via núcleo, sem reuso de sua funcionalidade; c) com reuso de conexões funcionais; d) via *bypass* paralelo; e) via *bypass* serial
 Fonte: Cota et al. (2002)

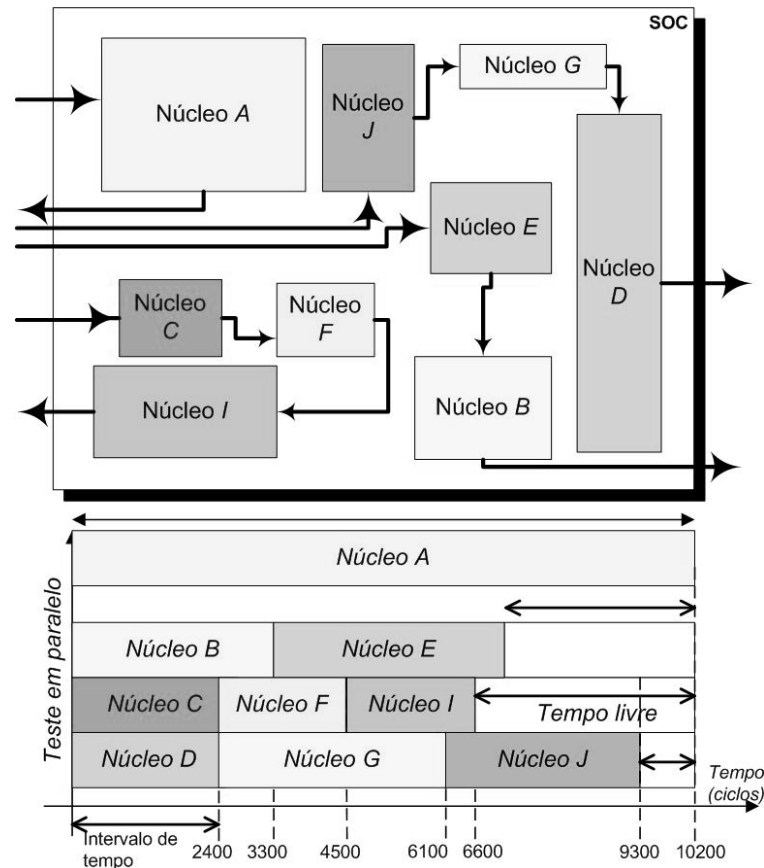


Figura 15 - Exemplo de escalonamento de teste de núcleos para um SOC arbitrário

Diversas técnicas na literatura são apontadas para obter escalonamentos explorando ao máximo o paralelismo, de modo a reduzir tanto quanto possível o tempo de teste do sistema, e estão ilustradas na Figura 16. Zorian (1993) e Chou, Saluja e Agrawal (1997) propuseram que os testes dos núcleos fossem agrupados em sessões de teste, intervalos de tempo específicos em que núcleos fisicamente próximos seriam testados, concorrentemente, com estruturas de teste e BIST dedicadas, ou ainda com conflitos de recursos, e são classificados como teste não-particionado, ilustrado na Figura 16a. Muresan, Wang e Vladutiu (2000) propõem que novos testes possam ser iniciados dentro de uma sessão independente de pertencer a outra, e tal método é classificado como teste particionado com execução até o fim (Figura 16b). Finalmente, buscando otimizar ainda mais os intervalos de tempo livre com restrições de potência e o paralelismo, um teste preemptivo é sugerido por Iyengar e Chakrabarty (2001), de modo que o teste de um dado núcleo pode ser interrompido e reiniciado mais tarde, conforme mostrado na Figura 16c.

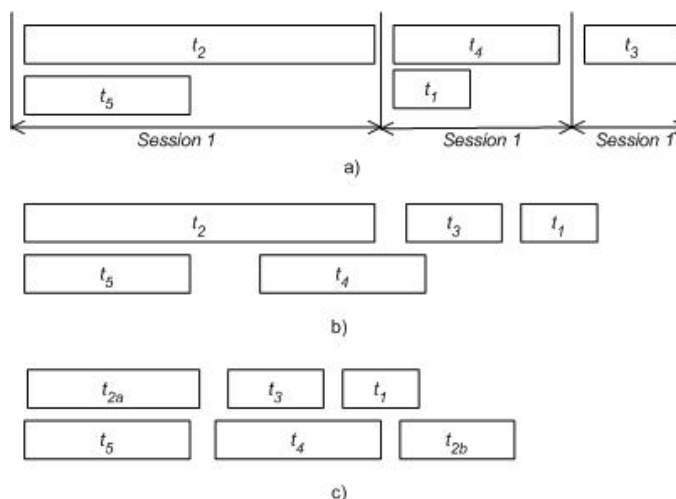


Figura 16 - Métodos de escalonamento: a) Não-particionado; b) Particionado com execução até o fim; c) Particionado com preemptividade
Fonte: Larsson; Pengo (2002)

O terceiro aspecto de otimização diz respeito à profundidade dos dados de teste e da memória disponível para armazená-los de uma vez, já que se a primeira for maior do que a segunda, então serão necessárias duas etapas de carga para o ATE, e o tempo de carga de dados nessa direção, diferente do tempo de aplicação de vetores de teste, é significativo (BARNHART, 2001). Por outro lado, o desejado é que a profundidade da memória do ATE seja maior, possibilitando, a depender da folga, que mais de um sistema encapsulado seja testado ao mesmo tempo (IYENGAR et al., 2002c).

Outro fator importante que implica em custo de projeto é a interação exigida entre o integrador do sistema e o desenvolvedor de teste do núcleo, uma vez que não necessariamente ambos trabalham juntos e têm à disposição todas as informações sobre o sistema e o núcleo a ser integrado. Por isso, uma mudança recente no paradigma de projeto de SOCs reside na integração do planejamento de teste nas fases iniciais do fluxo de projeto (LARSSON; PENGO, 2002), de modo que custos referentes ao roteamento funcional ou de teste dos núcleos sejam otimizados, assim como a inserção de pontos de observação e de outras estruturas de teste, como controladores ou *wrappers*. Além disso, muitos problemas referentes a esse planejamento, que antes eram tratados separadamente, passam a ser considerados em um procedimento integrado, uma vez que os resultados de um interferem em outros.

Uma outra linha de pesquisa no sentido de economia de recursos e simplificação na interação entre fornecedor de blocos IP e integrador de sistema é o padrão IEEE P1500

Standard for Embedded Core Test (SECT) (MARINISSEN et al., 1999; IEEE STANDARDS BOARD, 2005). Em linhas gerais, a idéia é padronizar e parametrizar as informações relativas ao teste de cada núcleo, bem como os *wrappers*, através de uma arquitetura simples e escalável e uma linguagem específica para descrição dos requisitos de teste e estruturas auxiliares (*Core Test Language* ou CTL), estendendo o conceito de reuso para o ambiente de teste de sistemas. Seguindo essa filosofia, outras iniciativas de padronização de infra-estrutura de teste propõem estruturas para ambos o barramento e o *wrapper*, como o *TestBus/TestCollar* (VARMA; BHATIA, 1998) e o *TestRail/TestShell* (MARINISSEN et al., 1998).

Enfim, considerando as principais vertentes no teste de SOCs digitais e mais esse aspecto relacional entre as partes, a ferramenta descrita em Cota (2003) serve de início para a investigação dessa dissertação, e propõe uma heurística de modo que:

- a) o planejamento de teste seja considerado juntamente com as fases iniciais de projeto, com isso, contribuindo para uma maior interação entre o integrador de sistema e o fornecedor de núcleos. O processo iterativo realizado pela ferramenta permite que o integrador aponte e sugira modificações no núcleo de forma que este se adapte melhor ao seu sistema, bem como reduzir seus custos de teste e de projeto;
- b) a otimização de recursos na definição das estruturas de TAMs e “wrappers” não seja feita para cada núcleo isoladamente, mas que se explore a diversidade de requisitos de teste entre eles, de forma que aqueles núcleos com teste mais caro terão acesso a mais recursos de teste;
- c) o Modelo Multi-TAM seja considerado na exploração do espaço de projeto, conforme já mencionado anteriormente.

Ao mesmo tempo em que a TAM global é definida, o escalonamento dos núcleos é realizado, obedecendo a critérios de sistema estabelecidos a priori. Dessa forma, uma exploração mais abrangente do espaço de projeto é possível, de modo que a síntese física do sistema seja otimizada em termos de teste e recursos como área em silício, por exemplo.

No capítulo 4, mais detalhes sobre a ferramenta serão dados, bem como os conceitos a partir dos quais ela foi elaborada.

3.3 TESTE DE SOCS DE SINAL MISTO

Devido ao grande número de aplicações em sistemas eletrônicos que envolvem sinais de RF ou alguma forma de interface com sinais analógicos, muitos projetos de sistemas embutidos têm incluído núcleos analógicos ou mistos, como conversores, filtros *anti-aliasing*, *mixers* para RF, entre outros. O teste de tais circuitos apresenta o inconveniente de não ter procedimentos padrão e bem difundidos na indústria, que reduzam o custo de teste tanto em termos de acréscimo de área em silício, tempo de teste e pinos extras. Circuitos operando a baixas frequências, como conversores e filtros em aplicações para áudio, apresentam tempo de teste muito longo. Limitações tecnológicas também incrementam o custo do testador. Por exemplo, circuitos de RF devem ser testados por circuitos com desempenho superior, com uma maior largura de banda, sendo assim, naturalmente, mais caros de se fabricar. Estratégias para tratar desse problema envolvem o desenvolvimento de técnicas de estabilidade como BIST, a ser tratado na próxima seção.

Uma possibilidade de se reduzir ainda mais o acréscimo em área é através do reuso de dispositivos já disponíveis dentro do sistema. Alguns testadores externos analógicos possuem um conversor analógico-digital interno que converte para o domínio digital a resposta do teste, que então pode ser avaliada pelo processador interno deste. Como muitos sistemas baseados em núcleos possuem também dispositivos com capacidade de processamento suficiente para análise, seria possível diminuir os requisitos do testador externo trazendo essas funções para dentro do sistema sob teste. Entretanto, é necessário que o conversor tenha resolução alta o suficiente para garantir a precisão desejada da medida. No entanto, conversores de média a alta resolução, se não estiverem disponíveis no sistema, representam um grande acréscimo de área, além de necessitarem também de teste, aumentando o tempo de teste do sistema todo.

No intuito de se reduzir o acréscimo de área devido à inserção de estruturas de BIST, alguns trabalhos, dentre eles o de Negreiros (2002), têm investigado a possibilidade de se efetuar a avaliação da resposta ao teste através de processamento digital, que pode ser tanto efetuado por processadores localizados no ATE ou embutidos no mesmo sistema que o CUT, no caso de SOCs.

Em nível de sistema, também há pendências na resolução de questões sobre a integração do teste com o projeto do sistema com núcleos mistos. Propostas na literatura incluem a adaptação do padrão IEEE 1149.1, estabelecido para teste de sistemas digitais, para o teste de núcleos mistos, sendo referenciado como o padrão IEEE 1149.4 (IEEE STANDARDS BOARD, 2000; BENNETTS, 2004). A adoção deste novo padrão não exclui o antigo, antes lhe dá mais funcionalidade, o que se mostra bem versátil para sistemas de sinal misto. Basicamente, propõe a adição de mais dois pinos para barramentos de sinais analógicos de teste, como se pode observar na arquitetura ilustrada na Figura 17, além de outras estruturas que permitem casamento de impedância, medidas em tensão ou corrente e de variáveis contínuas como resistências, capacitâncias, atrasos e observação de sinais diferenciais, entre outros (SUNTER, 1996). Uma dessas estruturas, o Módulo Analógico de Periferia, ou *Analog Boundary Module* (ABM), ilustrado na Figura 17b, tenta trazer para o domínio analógico as facilidades que o *Boundary Scan* trouxe para o domínio digital. Trabalhos recentes apontam um crescimento no interesse da validação deste padrão para projetos de circuitos mistos (ACEVEDO; RAMIREZ-ANGULO, 2002; CALVANO et al., 2002; KAC et al., 2003).

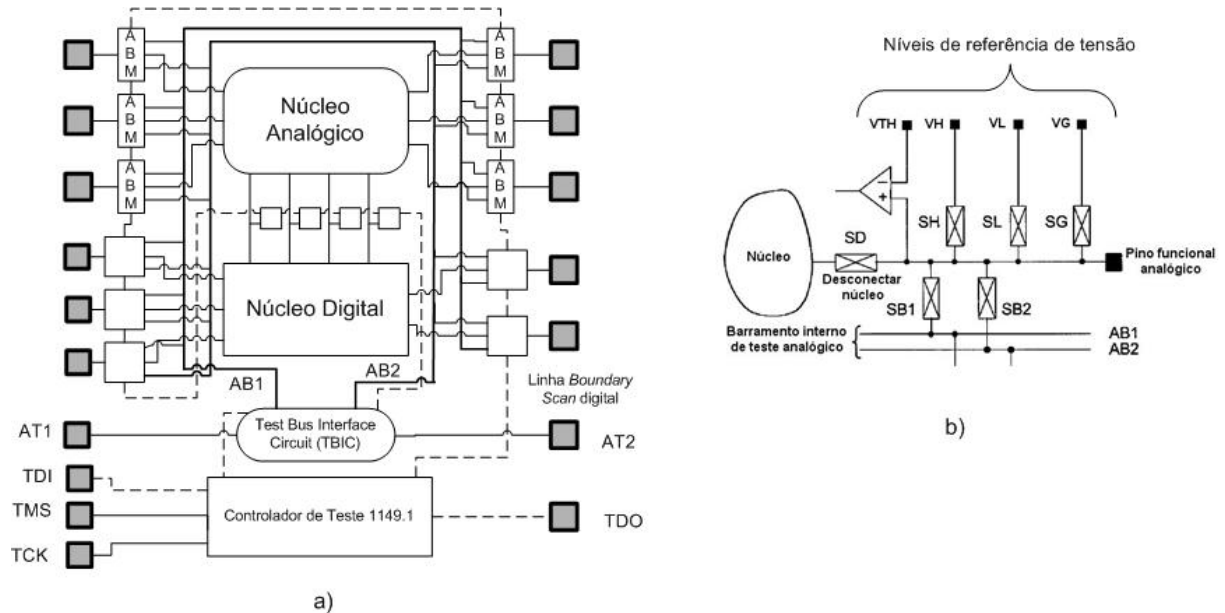


Figura 17 - a) Arquitetura simplificada do barramento de teste analógico segundo o padrão IEEE 1149.4; b) Arquitetura simplificada do Módulo Analógico de Periferia (ABM)
Fonte: IEEE Standards Board (2000)

Uma outra solução proposta na literatura, mas ainda em estágio inicial de investigação, é a adoção de *wrappers* analógicos para integração dos núcleos analógicos na estratégia de otimização da TAM Global (SEHGAL; OZEV; CHAKRABARTY, 2003). Em suma, o que se pretende é que, para efeito de otimização da estratégia de teste, os núcleos analógicos sejam considerados como digitais. Para que isso seja possível, deve-se considerar a existência de ADCs e DACs (conversores digital-para-analógico), de modo que o sinal de teste digital seja convertido para o domínio analógico, excitando adequadamente o núcleo analógico. Então, basta reconverter o sinal de resposta do núcleo para o domínio digital, de modo que seja adquirido pelo equipamento de teste externo. A Figura 18 mostra um SOC hipotético com essa arquitetura.

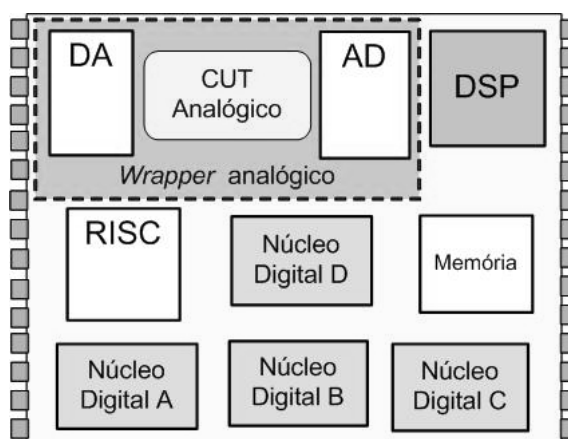


Figura 18 - Arquitetura de teste de um SOC adotando *wrapper* analógico
Fonte: Sehgal; Ozev; Chakrabarty (2003)

Ainda segundo Sehgal, Ozev e Chakrabarty (2003), os conversores devem ser projetados de forma que, a partir de sinais de controle, eles possam se adaptar às diversas condições de faixa de operação e velocidade dos diversos núcleos. Além disso, para cada sistema, há apenas um par de conversores ADC-DAC, que são compartilhados entre todos os núcleos analógicos presentes no sistema, já que o acréscimo em área devido a um par de ADC-DAC já é grande o suficiente para desestimular sua adoção. Em consequência, o teste destes núcleos será realizado de forma serial, o que impede redução significativa no tempo de teste usando teste simultâneo de núcleos. Além disso, nos resultados mostrados, não fica claro que o tempo de teste obtido inclui o teste dos *wrappers* analógicos também. Uma vez que estes têm uma área considerável, a hipótese da probabilidade de ocorrência de falhas ser tão

pequena que possa ser desprezada não é mais válida. Com isso, os *wrappers* devem também ser testados, o que representará um incremento considerável ao tempo de teste do sistema.

Finalmente, observa-se que os trabalhos anteriormente apresentados focam na integração de núcleos analógicos e digitais em uma única estratégia de otimização de recursos, seja através da otimização da definição de TAM ou pela adoção de um barramento de teste analógico e digital. A proposta a ser apresentada neste trabalho também busca a integração dos núcleos digitais e analógicos em uma única estratégia, sem, entretanto, apresentar as mesmas desvantagens em termos de acréscimo de área em silício ou de tempo de teste, devido à serialização do teste dos núcleos analógicos e do teste das estruturas dos *wrappers*. Tais estruturas representaram parte significativa da área do circuito e também estão sujeitas a falhas. A redução nos custos do testador externo se dá pelo reuso de recursos já disponíveis dentro do circuito para realizar as funções de geração de estímulos e análise da resposta de teste, antes executadas no testador. Por exemplo, essas funções podem ser realizadas por processadores digitais, que são peças presentes na maioria dos projetos de sistema de sinal misto. Uma vez que algumas técnicas de BIST baseiam-se no processamento de uma resposta de teste digitalmente compactada, trazer o processamento do ATE para dentro do sistema pode permitir que equipamentos de menor custo sejam necessários para se realizar o teste de tais sistemas, diminuindo os custos explícitos.

Este reuso também implica em teste serializado, por isso se verifica a possibilidade de reusar mais de um núcleo presente no sistema, ou a utilização de métodos de teste eficazes de forma a reduzir o tempo de teste dos núcleos analógicos e diminuir a influência negativa da serialização. No caso em estudo, serão considerados métodos de BIST analógico, e a sua escolha influi diretamente na cobertura de falhas, no tempo e na qualidade de teste.

Um terceiro aspecto a ser explorado é a possibilidade de exploração do espaço de projeto e ordenação do teste dos núcleos, de forma a encontrar uma solução otimizada em termos de outros parâmetros associados aos custos implícitos, como área das estruturas de teste e número de pinos adicionais na interface do circuito. Também se deseja verificar a influência da limitação de dissipação de potência do circuito, de modo a evitar danos provenientes da energia térmica excessiva, no escalonamento dos núcleos.

Considerando todos os aspectos citados, a ferramenta proposta por Cota (2003) deve ser adaptada de forma a incluir os núcleos analógicos no algoritmo de otimização. As modificações efetuadas serão discutidas em detalhe no próximo capítulo.

4 PLANEJAMENTO DE TESTE DE SOC E BIST BASEADO EM REUSO

Conforme mencionado no capítulo anterior, grande esforço tem sido envidado em estratégias de otimização do desempenho das soluções de teste, junto com sua integração ao fluxo de projeto, de forma que ambos sejam efetuados concomitantemente, em contraposição ao paradigma de se considerar o teste do sistema somente após sua concepção e projeto.

O escalonamento do teste de núcleos de um SOC visa obter o menor tempo de teste possível, alocando recursos de tempo aos núcleos sob testes de forma que uma maior número de núcleos sejam testados simultaneamente, o que implica na disponibilidade de recursos suficientes para tanto. Dessa forma, a necessidade de escalonamento para SOCs mistos é a mesma para SOCs digitais, com algumas peculiaridades, devidos aos núcleos analógicos. Uma dessas peculiaridades é que o tempo de teste de circuitos analógico, como já dito, é em geral muito maior do que o tempo de teste de circuitos digitais. Logo, os circuitos analógicos o tempo de teste do sistema. Ao longo desse tempo, o algoritmo de escalonamento alocará os recursos de tempo, área, TAM e pinos para os núcleos digitais. Tal escalonamento pode gerar intervalos de tempo ociosos, se o tempo de teste dos núcleos analógicos for razoavelmente maior do que o escalonamento de teste dos núcleos digitais. Nessa situação, o papel da ferramenta de planejamento de teste fica mais evidente, pois o escalonamento dos núcleos digitais pode ser refeito, de forma a otimizar outros recursos de teste, como número de pinos e área dos *wrappers* em troca do aumento do tempo de teste dos núcleos digitais, dentro do intervalo limitado pelos núcleos analógicos.

Neste capítulo, será abordada a ferramenta de planejamento de teste originalmente elaborada para o teste de SOCs exclusivamente digitais e os conceitos que permitirão adaptá-la para SOCs contendo núcleos de sinal analógico e com a possibilidade de realização de BIST, assim reduzindo os custos explícitos de teste. Os custos implícitos também podem ser reduzidos, dependendo da técnica de teste analógica utilizada.

Dentre as adaptações necessárias estão a criação de um tipo de acesso ao núcleo especial para os núcleos analógicos, a definição de quais núcleos podem enviar e receber sinais de teste do CUT analógico e quais serão os núcleos reusados, pois eles terão de ser testados antes do CUT. Este é o conceito de precedência, que será discutido em detalhes neste

capítulo. Essas adaptações serão necessárias para integrar os núcleos analógicos à estratégia de otimização das TAMs e do escalonamento de núcleos no SOC.

4.1 REBATE - REUSE-BASED TEST SCHEDULING

A ferramenta a ser usada é o REBATE, que é um acrônimo para *Reuse-based Test Scheduling*, e é descrita e elaborada em Cota (2003). Seu nome vem da possibilidade do reuso das conexões e dos núcleos no sistema para determinação e otimização das soluções de TAMs globais e escalonamento dos núcleos, sendo que estas duas tarefas são realizadas concomitantemente, privilegiando aqueles que apresentam maiores custos de teste.

A ferramenta precisa, como dados de entrada, de um modelo do sistema a ser testado, incluindo informações físicas, como posicionamento dos núcleos na pastilha de silício e a distância entre os mesmos, e requisitos de teste para cada núcleo, como número de vetores e de entradas e saídas dedicadas e potência consumida durante o teste. As conexões funcionais entre os núcleos são modeladas através de um grafo dirigido e ponderado, cujos vértices representam os núcleos e as arestas, as conexões. O peso associado a cada aresta é função do custo em termos de área, potência, pinos e ciclos de teste, de forma que essa associação depende do tipo de modelo de TAM usado e dos núcleos conectados. Finalmente, são informados os limites de consumo de potência e do número de pinos de todo o sistema.

Um conceito importante na concepção da reusabilidade presente no algoritmo é o de vizinhança de um núcleo. Diz-se que um núcleo é vizinho de outro núcleo quando este pode usar as conexões e funções daquele na definição da TAM, ou seja, *wrappers*, cadeias *scan*, *flip-flops*, *buffers* e outras estruturas do núcleo, além dele mesmo, podem servir como parte da TAM para o teste de outro núcleo. A rigor, inicialmente todos os núcleos podem ser vizinhos de todos, considerando que todos sejam digitais. No caso de sistemas mistos, será visto a seguir que alguns cuidados devem ser feitos no momento da modelagem do sistema. A interface é considerada como um núcleo do sistema, de modo que se pode determinar quais núcleos podem ou não reusar recursos da interface (por exemplo, conexão direta) no momento da definição da TAM.

A resposta esperada do algoritmo é composta de um conjunto de vetores contendo intervalos de tempo, ou *slots*, do teste de cada núcleo, e ainda suas TAMs de entrada e saída.

Em cada *slot*, é computada a potência dissipada pelos núcleos testados durante o intervalo. A solução deve respeitar os limites estipulados de potência e pinos, além de buscar otimização em termos do tempo de teste e do acréscimo de área, sendo que as prioridades de otimização podem ser definidas pelo usuário, dependendo dos requisitos do sistema. Além disso, baseado nas definições das TAMs, o acréscimo de área devido aos *wrappers* e o número de pinos extras necessários para transporte dos sinais de teste também são calculados e armazenados no arquivo de saída. Através dessas informações, é possível então avaliar a eficiência do planejamento de teste.

Informações relativas ao teste dos blocos estão disponíveis nos *benchmarks* divulgados por ocasião do ITC 2002 (MARINISSEN; IYENGAR; CHAKRABARTY, 2003). Por questões de proteção de propriedade intelectual, a funcionalidade dos blocos não é divulgada. Por exemplo, as dimensões dos núcleos são importantes para o posicionamento e avaliação dos custos de roteamento, e pode ser dada pela Equação (4) (COTA, 2003). Para estimativa da área ocupada por cada núcleo, alguns dados físicos do sistema podem ser retirados ou estimados a partir de informações dos *benchmarks*, como o número de *flip-flops scan* (n_{ff}) e o número de entradas e saídas do núcleo (respectivamente, *inputs* e *outputs*). Outros parâmetros como área de um *flip-flop* (ff_{area}) ou de uma porta lógica ($gate_{area}$) são dependentes da tecnologia de fabricação considerada, e também são definidos a priori, sendo que neste trabalho considera-se uma tecnologia de 0,8 μ m. Os valores α e β servem como fatores de escalamento para aumento ou redução da área.

$$nucleo_{area} = \alpha(n_{ff} \cdot ff_{area}) + \beta(inputs + outputs) \cdot gate_{area} \quad (4)$$

Dos modelos disponíveis, apenas um apresenta o consumo de potências dos seus núcleos. Assim, para os restantes, uma estimativa da potência consumida é feita baseada na Equação (5), em função do equivalente de portas lógicas do circuito (n_{gate}) e de *flip-flops* (n_{ff}), do índice de chaveamento de cada um desses dispositivos (sw_{gate} e sw_{ff}) da tensão de alimentação V_{DD} , da capacitância de porta dos dispositivos C_L e da frequência do sinal de entrada f . Observa-se, então, que neste caso há também uma dependência com a tecnologia de fabricação escolhida, uma vez que esta influi na tensão de alimentação do circuito e nas capacitâncias de entrada.

$$potencia_{nucleo} = C_L \cdot V_{DD} \cdot f \cdot [(sw_{ff} + 1) \cdot n_{ff} + sw_{gate} \cdot n_{gate}] \quad (5)$$

Já o posicionamento e roteamento dos blocos, ou *place-and-route*, pode ser realizado através de ferramentas comerciais disponíveis no mercado. Entretanto, para fins de validação da estratégia proposta, utiliza-se um algoritmo desenvolvido pelo Instituto de Informática da Universidade Federal do Rio Grande do Sul (UFRGS) baseado na técnica de *Simulated Annealing* com o objetivo de se obter uma solução que minimize a distância entre os núcleos, o que por sua vez minimiza os custos associados às interconexões.

Um exemplo de sistema, desenvolvido e fornecido pela UFRGS, é denotado pela sigla U226 (Figura 19), e consiste de um processador RISC, uma memória, um núcleo de controle do processador mais três pares de filtro analógico e ADC. O sistema recebe três sinais analógicos de entrada que são filtrados e convertidos e então alimentam o processador, que também recebe estímulos digitais, com largura de oito bits. No Apêndice, para efeito de ilustração, é apresentado um exemplo de arquivo de descrição de sistema usado nos experimentos, já considerando as adaptações para a proposta de teste misto e BIST, que serão apresentadas a seguir.

4.1.1 Algoritmo de escalonamento e definição das TAMs

As TAMs são consideradas como conexões em série entre núcleos de um sistema, o que implica que a otimização de custos não pode ser considerada um problema exclusivo de alocação de recursos de teste, como TAMs e *wrappers*, ou de escalonamento, no caso de determinação dos *slots* de teste para cada núcleo. O tempo total de teste de um dado núcleo depende da definição de sua TAM, incluindo os núcleos vizinhos por que os sinais de teste passam, até chegar à interface ou ao ORA embutido no sistema, estando este presente. Por sua vez, a definição dos recursos de teste individuais de cada núcleo só está disponível após a definição da TAM Global de todo o sistema, o que implica que o tempo total de teste do sistema já esteja definido. Assim, ambos os problemas estão intrinsecamente relacionados e não se tem a informação necessária a priori para usar métodos tradicionais de resolução computacional para cada um dos problemas isoladamente, o que leva a pensar na hipótese de se integrar as soluções aos dois problemas em um único procedimento.

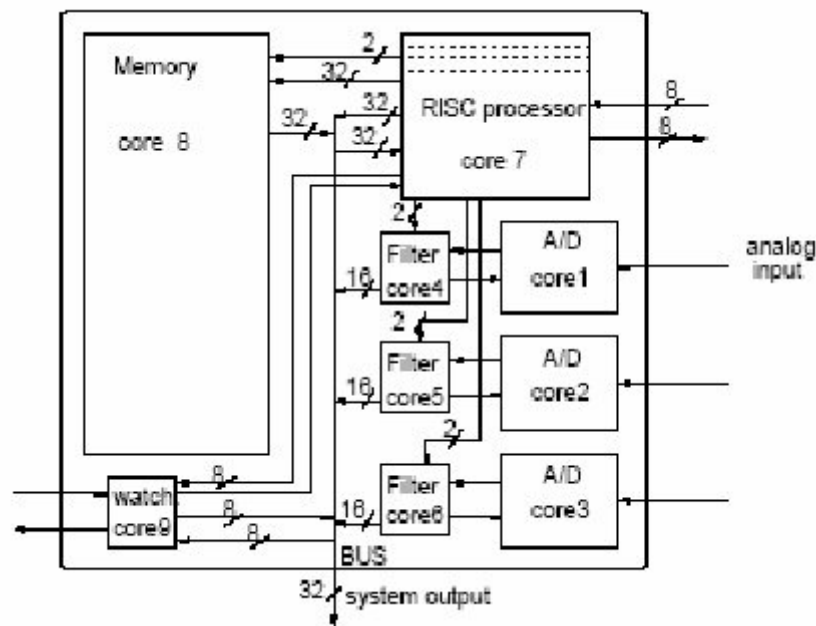


Figura 19 - Arquitetura do SOC U226
Fonte: Cota (2000)

O modelo escolhido para resolução do problema das TAMs usado em Cota (2003) é o algoritmo do menor caminho (*shortest path*). Neste, cada possibilidade de acesso entre núcleos é denotada como um caminho associado a um peso, que é função de seus custos. O conjunto de todas as possibilidades expandidas desde o CUT até a interface é chamado de árvore. Então, o objetivo se torna encontrar o caminho da periferia até o núcleo que corresponda ao menor custo otimizado em um ou mais fatores. De acordo com o mesmo modelo, para cada núcleo são definidas duas árvores, sendo uma para transporte dos sinais de teste de entrada e outra para os sinais de saída. Cada árvore lista as possíveis conexões desde a periferia do circuito ou TPG ou ORA, que seriam suas folhas, até os pinos do núcleo sob teste, que seria sua raiz, passando por outros núcleos do sistema representados pelos nós internos da árvore. Cada arco ligando dois nós em uma árvore é associado a um valor que é função de seus custos, e o número de arcos depende do número de modelos de TAMs possíveis para cada conexão, segundo o conceito de Multi-TAM.

Tal modelo, entretanto, apresenta dois defeitos: primeiro, o número de nós da árvore gerada a partir de um dado CUT aumenta com o fatorial do número total de núcleos do sistema N , segundo a Equação (6) abaixo (COTA, 2003).

$$total_nos = \sum_{1 \leq j \leq N-1} \frac{(N-1)!}{(N-j-1)!} + \sum_{1 \leq j \leq N} \frac{(N-1)!}{(N-j)!} \quad (6)$$

Mesmo restringindo alguns núcleos de serem reusados na definição das TAMs, de forma a diminuir a expansão da árvore, isso não traz efeito algum para simplificação do algoritmo se o número de núcleos do SOC cresce. O algoritmo *shortest path* possui complexidade da ordem $O(V^2)$, em que V é o número de nós da árvore, isto é, o número de operações aumenta de forma quadrática com relação ao número de nós. Enfim, a solução se torna impraticável.

A segunda desvantagem é que o modelo proposto ainda não consegue realizar as questões da otimização do escalonamento e da alocação de recursos de teste ao mesmo tempo. Para cada núcleo do sistema, os custos das TAM associadas são avaliados e com base nesses custos os caminhos são definidos. Só então o algoritmo otimiza o escalonamento, quando o conflito de recursos pode ser avaliado. Desse modo, embora tal procedimento atinja o objetivo de otimização na alocação dos recursos, não consegue realizar um escalonamento de forma a obter um tempo total ótimo de teste do sistema.

Para contornar essas dificuldades, o algoritmo é modificado de forma que começa a busca definindo um tempo mínimo de tempo total do sistema. Este tempo mínimo corresponde ao maior de todos os tempos de teste de cada núcleo separadamente, numa situação em que o acesso à interface do circuito é direto e com largura de bits completa, no caso de núcleos digitais. Logo, o tempo mínimo é função do tamanho das cadeias *scan*, pelo número de vetores de teste e o número de ciclos para aplicação de um vetor de teste. Uma vez que este tempo é definido, os núcleos restantes são escalonados concorrentemente dentro deste intervalo de tempo e respeitando os possíveis conflitos da solução encontrada.

Para melhor ilustração, considere-se a solução hipotética mostrada na Figura 15, em que o **Núcleo A** determina o tempo de teste do sistema. Observa-se que nesta solução há um intervalo de tempo disponível após o teste do **Núcleo J**. Supondo que não existam mais núcleos a serem testados, uma possibilidade de aproveitar melhor este intervalo seria, por exemplo, verificando se haveria outro acesso para o mesmo **Núcleo J** tal que seu tempo de teste aumentasse, mas permanecendo ainda dentro do limite do sistema e permitindo uma redução nos custos da TAM, através da diminuição do número de pinos ou da área do *wrapper*. Escrevendo formalmente, seja o tempo de teste do sistema **tempo_máximo**, e um

dado **Núcleo J** com tempo de teste T_J e custo de TAM C_J , devido a pinos extras e acréscimo de área, principalmente. O que se quer dizer é que pode ser possível encontrar outra solução (T'_J, C'_J) em que $T_J < T'_J < \text{tempo_máximo}$ e $C'_J < C_J$ (Figura 20), ou seja, pode-se reduzir alguns custos de teste sem afetar o tempo total de teste do sistema, desde que seja respeitado ainda o limite de potência dissipada, quando este for considerado. Se necessário for, o teste do núcleo pode também ser atrasado, mantendo-se dentro do tempo livre disponível, de forma que a potência dissipada em um certo momento esteja dentro dos limites tolerados.

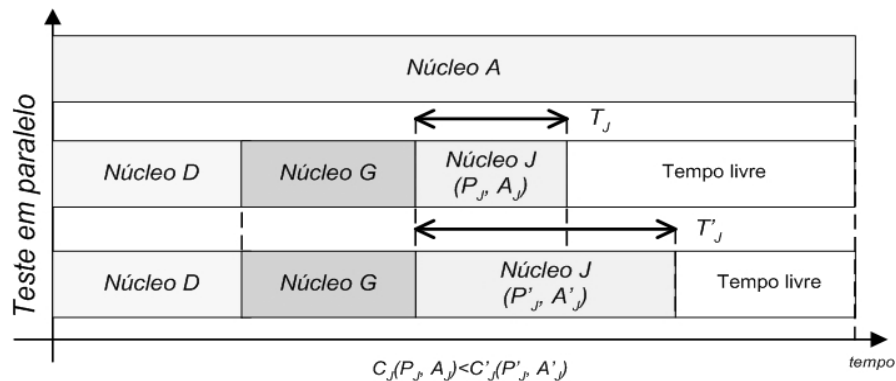


Figura 20 - Otimização de recursos de teste de núcleos digitais (pinos e área)

A aplicação deste procedimento sistematicamente promoveria, então, a exploração do espaço de projeto desejada durante o processo de otimização de recursos de teste. Entretanto, dois pontos ainda precisam ser discutidos:

- dado um intervalo de tempo ocioso, a escolha do núcleo que o utilizará na redefinição de seu acesso;
- caso o tempo de teste limite do sistema seja pequeno o suficiente para restringir a otimização dos recursos (o que é muito provável), uma estratégia para expansão de tal limite deve ser empregada.

O primeiro ponto é resolvido classificando todos os núcleos em ordem decrescente em relação aos custos de teste, de forma que aqueles que requerem mais recursos têm maior prioridade para redução dos custos, enfim, direito a mais recursos e mais tempo disponível para aplicação de seu teste. O segundo ponto pode ser resolvido através de um processo de incremento iterativo do tempo limite de teste e reavaliação dos custos do novo escalonamento

de forma que o novo limite seja estimado através da lista completa de conflitos e recursos de teste obtidos da solução anterior.

O funcionamento do algoritmo pode ser ilustrado pela Figura 21, com um exemplo de construção do escalonamento, e pela Figura 22. Considera-se inicialmente que dos cinco blocos presentes, quatro já foram escalonados, faltando a definição de escalonamento e da TAM do CUT 4. Na Figura 22a, os arcos representam TAMs que ligam o CUT 4 aos outros núcleos vizinhos ou à interface, num primeiro nível. A partir do escalonamento parcial, listas de conflito de recursos e de intervalos de tempo livre são elaboradas (Figura 21), seguindo o seguinte raciocínio: supondo que o núcleo 1 é usado como parte da TAM do CUT 2, então tem-se um conflito de recursos entre ambos, e o algoritmo não escala o teste de ambos ao mesmo tempo. Ainda, qualquer outro núcleo que incluir o núcleo 1 em sua TAM entrará em conflito não só com este, mas também com o núcleo 2 que já usa o núcleo 1 em sua TAM, e assim também não podem ser escalonados juntos. Os intervalos de tempo livre são aqueles em que um núcleo pode ser testado sem entrar em conflito com outros núcleos.

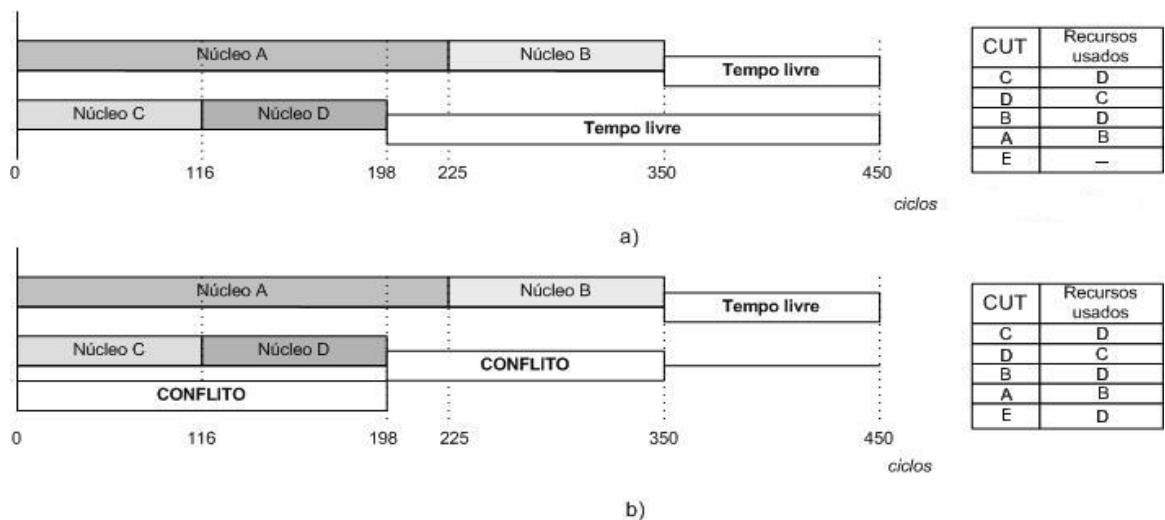


Figura 21 - Construção do escalonamento do teste dos núcleos: (a) construindo lista de conflitos de recursos baseado no escalonamento corrente; (b) identificação de conflitos e escalonamento do núcleo restante

Fonte: Cota (2003)

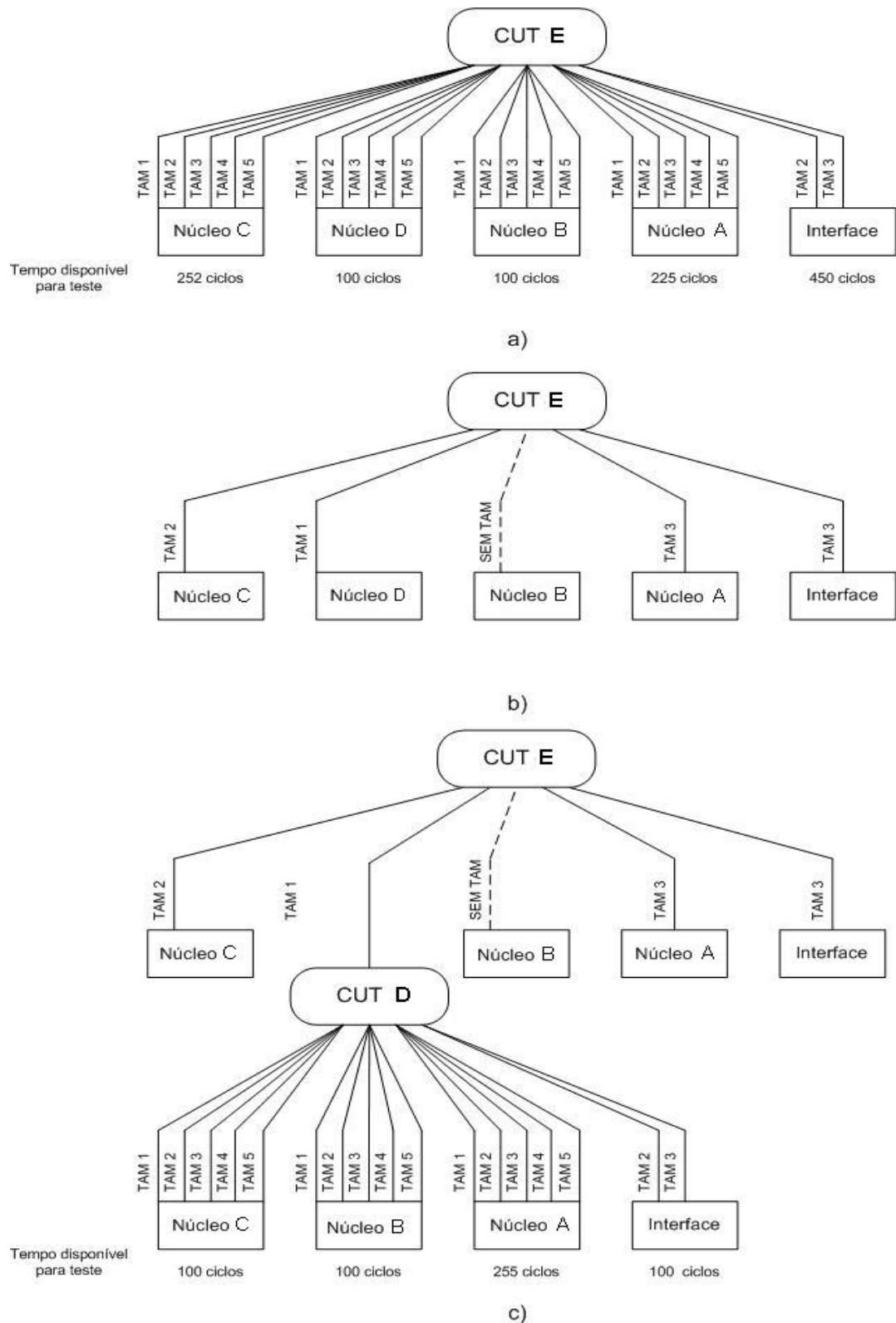


Figura 22 - Busca dos acessos de teste e expansão da árvore até a interface do sistema: (a) Lista das possibilidades de conexão entre CUT e núcleos e interface; (b) seleção das TAMs menos caras para cada núcleo; (c) escolha do melhor caminho e expansão da TAM para o próximo nível
 Fonte: Cota (2003)

Com base nessas listas, a TAM de menor custo para cada núcleo vizinho e que satisfaça a condição de tempo de teste dentro do intervalo disponível (caso exista alguma satisfatória) é previamente escolhida. Por exemplo, Se o núcleo vizinho B for parte da TAM, então o tempo de teste do CUT deve ser menor do que o disponível, no caso 100 ciclos. Suponha-se que este não é o caso, assim, dentre as possibilidades restantes e que satisfazem o intervalo de tempo disponível, o algoritmo escolhe a mais barata em termos de área e pinos. O procedimento é repetido no nível seguinte (Figura 22b), ainda seguindo as limitações de reuso de recursos e tempo de teste, até que chegar a uma folha da árvore, isto é, à interface do sistema (Figura 22c). Enfim, com a TAM já definida, o escalonamento dos núcleos é atualizado com base na nova TAM (Figura 21b).

Em suma, a construção da árvore e busca do caminho ótimo em termos de custos segue os seguintes passos:

- a) cada arco ligando dois nós da árvore é associado a uma função custo total com quatro variáveis: tempo de teste, área, pinos e potência;
- b) a construção da árvore é feita ao mesmo tempo em que o algoritmo *shortest path* é executado, de modo que apenas os nós mais promissores são expandidos, isto é, aqueles com maior chance de otimização.

Finalmente, a heurística proposta para resolver os problemas acima citados está resumida no pseudo-código da Figura 23. Segundo o mesmo, em primeiro lugar o tempo mínimo de teste do sistema é definido, e então o algoritmo é executado enquanto os requisitos do sistema não forem atingidos ou até um determinado número de operações (linha 2). O laço principal está entre a linha 3 e a linha 19. Dentro desse, entre as linhas 7 e 17, é efetuada a construção das árvores de entrada e saída a partir do CUT, uma de cada vez. Para evitar repetições indefinidas deste trecho, observa-se o uso da definição da vizinhança do núcleo (linha 8), de modo que um caminho do CUT até a interface sempre é encontrado.

O algoritmo de busca e seleção de TAMs desenvolvido é implementado entre as linhas 9 e 14, através da verificação dos requisitos de teste, avaliação dos custos e enfim seleção do melhor núcleo vizinho na TAM em direção à interface. A heurística foi desenvolvida de modo que a otimização realizada esteja focada naquilo que o integrador ou engenheiro de teste deseja otimizar, se tempo de teste, pinos, área ou potência. Essa otimização é permitida através da realimentação implementada entre as linhas 20 e 26, em que se verifica se a

solução encontrada no momento é satisfatória. Caso contrário, se o foco da otimização é o número de pinos ou área, é realizada a citada perturbação no tempo de teste do sistema, a fim de encontrar uma solução mais atraente em termos de recursos de teste. O núcleo mais crítico em termos destes recursos é escolhido e sua TAM é redefinida, usando o mesmo algoritmo especificado entre as linhas 7 e 17, sem a verificação de tempo livre no escalonamento e escolhendo o núcleo que permita uma maior redução no uso de recursos de teste. Para otimização do tempo de teste, o melhor vizinho em qualquer nível da árvore é escolhido de forma que implique no menor incremento possível no tempo de teste do sistema (linha 14). Finalmente, baseando-se na lista de conflitos elaborada durante a solução global anteriormente encontrada, calcula-se um novo limite de tempo (linha 25), de modo que otimizações já obtidas sejam mantidas se adequado. Em seguida, o escalonamento é iniciado novamente e a procura parte da linha 3.

```

1. Estabelecer o tempo de teste inicial sob restrições de potência
2. Enquanto (restrições do sistema não satisfeitas) E (iterações < máximo_iterações)
3.   Enquanto (houver núcleos não escalonados)
4.     Escolhe CUT = núcleo crítico em termos de custos
5.     Para cada (direção das portas de CUT)
6.       Nó = CUT
7.       Enquanto (não está na interface) E (custo acumulado <= custo atual)
8.         vizinhos = recupera (vizinhança CUT) E (vizinhança nó)
9.         Para cada (elemento em vizinhos)
10.          Intervalo_de_tempo = intervalo de tempo livre no escalonamento atual
11.          Para cada (Modelo de TAM)
12.            Avalia custos da conexão de TAM CUT-vizinho
13.            Escolhe melhor TAM CUT-vizinho
14.            Escolhe melhor vizinho para CUT
15.            Custo acumulado + = custo CUT-vizinho escolhido
16.            Nó = vizinho escolhido
17.            Vai para o passo 7
18.   Insere CUT no escalonamento
19.   Atualiza Lista de conflitos
20. Se (restrições não satisfeitas)
21.   Fixa limite de tempo de teste = infinito
22.   Escolhe CUT = núcleo crítico em termos de custos
23.   Achar uma TAM mais barata para CUT
24.   Escalonar CUT
25.   Fixar novo limite de tempo para o sistema
26.   Vai para o passo 5

```

Figura 23 - Pseudo-código para escalonamento de núcleos e otimização de custos de teste da ferramenta REBATE
Fonte: Cota (2003)

Considerando o número de laços de repetição presentes na heurística e a complexidade de cada laço, Cota (2003) demonstra que essa tem uma complexidade da ordem de $O(N^3)$, isto

é, o tempo de execução cresce com velocidade proporcional a uma função de terceiro grau, em que N representa o número de núcleos do sistema. Imediatamente pode-se concluir que para sistemas suficientemente grandes, o tempo de simulação pode se tornar proibitivamente alto. Como possibilidades de modelagem para diminuir este tempo está um número menor de núcleos na vizinhança de cada CUT, de modo a restringir a busca.

4.2 ADAPTANDO A FERRAMENTA PARA TESTE DE SOCS MISTOS

Para que a ferramenta REBATE possa também abranger o planejamento de teste de sistemas mistos, algumas modificações se fazem necessárias. De uma forma geral, as alterações a serem efetuadas no modelo do sistema na transição do teste analógico para o digital consistem principalmente no número de cadeias *scan* e número de amostras necessárias. No caso de circuitos analógicos, como não existem cadeias *scan*, esta informação não terá sentido, e pode ser descartada. Por outro lado, o número de amostras fornecido no modelo deve ser alterado, pois, no algoritmo, ele denota o número de ciclos digitais que o núcleo necessita para ser testado, assumindo que cada amostra leva um ciclo. Assim, na descrição do sistema, o número de amostras de teste de cada núcleo deve ser multiplicado por um fator que representa a razão entre a frequência de Nyquist para o núcleo analógico em questão e a frequência de operação do testador externo. O cumprimento do critério de Nyquist é importante para garantir que a informação de teste não seja corrompida ou invalidada. Desta forma, o resultado final será, em número de ciclos, o tempo gasto pelo núcleo analógico durante seu teste.

Por sua vez, o número de amostras necessárias para o teste analógico depende da estratégia de teste a ser adotada. Para uma estratégia convencional, esse número é de 120000 amostras; considerando o teste baseado em ruído de ADCs, o número de padrões de teste necessário cai para 8192 padrões de teste. Para filtros, essa estratégia utiliza o mesmo número de padrões de teste. Para o teste baseado em oscilação, a redução é ainda mais drástica, pois somente 30 ciclos do sinal de oscilação são necessários para se obter a amplitude e a frequência do mesmo. Entretanto, deve se ressaltar que esta última acarreta num grande acréscimo de área devido ao circuito interpretador da resposta. No caso do TRAM, por

exemplo, o tempo de teste depende principalmente da frequência de corte do filtro, já que esta influi no tempo de pico da resposta a um degrau, rampa ou parábola, conforme o caso.

Outra consideração importante é a inclusão de mais um tipo de acesso ao núcleo para o modelo Multi-TAM, entretanto, dessa vez restrito aos núcleos analógicos. Esse modelo está resumido na Figura 24, para entrada e saída, embora não restrita a essas posições. No primeiro caso, há conexão direta entre o núcleo e a interface do *chip*, semelhante aos núcleos digitais; à saída, observa-se que o núcleo analógico reusa a funcionalidade do núcleo vizinho, ou outro núcleo alcançável na TAM Global com o objetivo de processamento da sua resposta ao teste. Subentende-se que a saída do núcleo é digital (como no caso de ADCs) ou que há alguma conversão do sinal do domínio analógico para o digital (como no caso da estratégia usando *statistical sampler*, por exemplo).

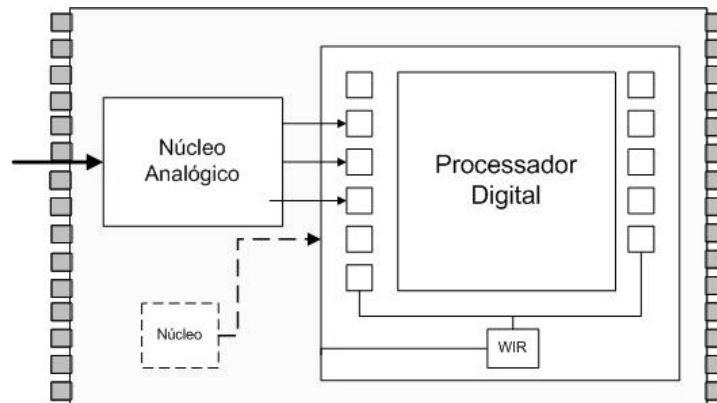


Figura 24 - Exemplo de conexão em TAM para circuitos analógicos

A necessidade de transporte de sinais analógicos na saída e/ou entrada restringe o conjunto de núcleos de um sistema candidatos à vizinhança de um núcleo analógico sob teste. Por exemplo, os sinais de estímulo para teste de um filtro devem passar por um circuito analógico, não sendo adequado usar o *wrapper* de um núcleo digital; para um ADC, a situação se repete, porém núcleos digitais podem ser vizinhos do lado de sua saída, uma vez que ela está no domínio digital. Dessa forma, é mister que núcleos analógicos tenham em sua vizinhança outros núcleos analógicos, ou então a interface do sistema, de modo que o transporte de sinais durante o teste não seja comprometido. Como exceção já citada, está o ADC, cuja saída é um sinal digital e sua TAM pode ser otimizada segundo o procedimento normalmente usado em núcleos digitais.

Finalmente, como a estratégia envolve o reuso de estruturas disponíveis no sistema, pode-se argumentar que falhas nessas estruturas podem fazer com que os resultados obtidos no teste sejam inconclusivos, mascarando falhas ou apontando como defeituoso um dispositivo sem falhas. Para evitar tal situação, é necessário garantir que os dispositivos reusados sejam testados antes dos núcleos analógicos, e assim cria-se o conceito de precedência, ou seja, os núcleos a serem reusados devem ter prioridade no escalonamento de teste, de forma que a solução encontrada não seja inválida. Esta informação também é fornecida na descrição do sistema.

Uma das vantagens a serem obtidas com a adoção da ferramenta para o planejamento de teste é a maior exploração do espaço de projeto. Conforme discutido na seção anterior, um dos limitantes para otimização dos recursos a alocar para o teste dos núcleos digitais é o tempo de teste. Entretanto, em se considerando teste misto, como o tempo de teste dos núcleos analógicos é em geral maior do que o dos núcleos digitais, esses dominarão o tempo de teste do sistema, provocando maiores intervalos de tempo ociosos. Com isso, é possível que a ferramenta, tendo como novo limite de tempo o estabelecido pelos núcleos analógicos, consiga otimizar outros recursos de teste, como número de pinos, área dos *wrappers*, como já ilustrado na Figura 20.

Usando como exemplo as estratégias de BIST citadas no capítulo 2 desta dissertação, pretende-se mostrar os benefícios advindos da aplicação da metodologia descrita neste trabalho no teste de sistemas mistos. Em geral, há uma redução no tempo de teste dos núcleos analógicos isoladamente. Uma das questões que se pretende discutir neste trabalho é se essa redução também implica numa redução do tempo de teste do sistema. Além disso, deseja-se saber se os novos limites de tempo permitem que essa redução seja feita sem prejuízo em termos de outros parâmetros, como número de pinos extras e área.

Independente do SOC considerado, supõe-se que exista um núcleo com capacidades de processamento que será reusado para avaliação da resposta de teste. Considera-se ainda que este tem uma memória interna capaz de armazenar a resposta do teste analógico para processamento a posteriori, ou que existe uma memória no sistema que também será reusada.

Uma situação hipotética de escalonamento e definição de caminhos para um sistema misto arbitrário, sem uso de BIST analógico, está à mostra na Figura 25. Observa-se que o núcleo analógico sob teste tem acesso direto e exclusivo à interface, uma vez que ele é o

único deste tipo no sistema. De fato, devido aos sinais analógicos com que ele trabalha, não há possibilidade dele reusar estruturas de outros núcleos digitais, nem destes usar daqueles. Ademais, considerando-se procedimentos padrão e a maioria dos circuitos analógicos em comparação com a maioria dos circuitos digitais, o tempo de teste analógico é ordens de grandeza maior do que o tempo de teste digital. Com isso, na etapa de planejamento de teste, soluções típicas envolvem o compartilhamento de recursos entre os núcleos digitais e o núcleo analógico sendo este quem limita o tempo de teste. Apesar de, em geral, se obter um bom resultado em termos de número de pinos extras e acréscimo de área, o tempo de teste se torna proibitivo para fabricação em volume, e por isso, técnicas devem ser desenvolvidas visando também a otimização deste recurso.

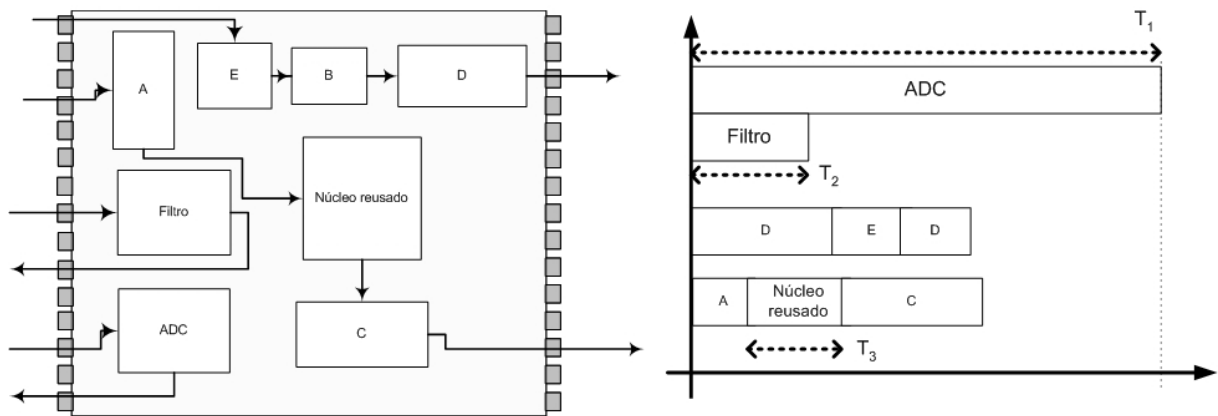


Figura 25 - Exemplo de TAMs para sistema misto hipotético e respectivo escalonamento, considerando métodos padrão de teste dos núcleos analógicos

No caso do método de BIST baseado em ruído, ilustrado na Figura 26, observa-se agora o efeito da serialização do teste devido ao reuso do núcleo para avaliação da resposta, bem como o acréscimo em área devido ao gerador de pseudo-ruído, funcionando como gerador de padrões de teste, e ao *statistical sampler*, no caso de teste de filtros. Analisando o escalonamento associado, pode-se concluir que pode haver ganho significativo em termos de tempo de teste caso a redução do tempo de teste dos núcleos isolados, promovida pelas estratégias de teste analógico, seja maior do que o aumento do teste devido à serialização. Outro ponto importante é que, uma vez que os tempos de teste analógicos são reduzidos, a otimização de outros recursos de teste, como pinos extras e acréscimo de área, pode não produzir resultados tão bons quanto no caso do procedimento padrão de teste, segundo o raciocínio delineado na Figura 20.

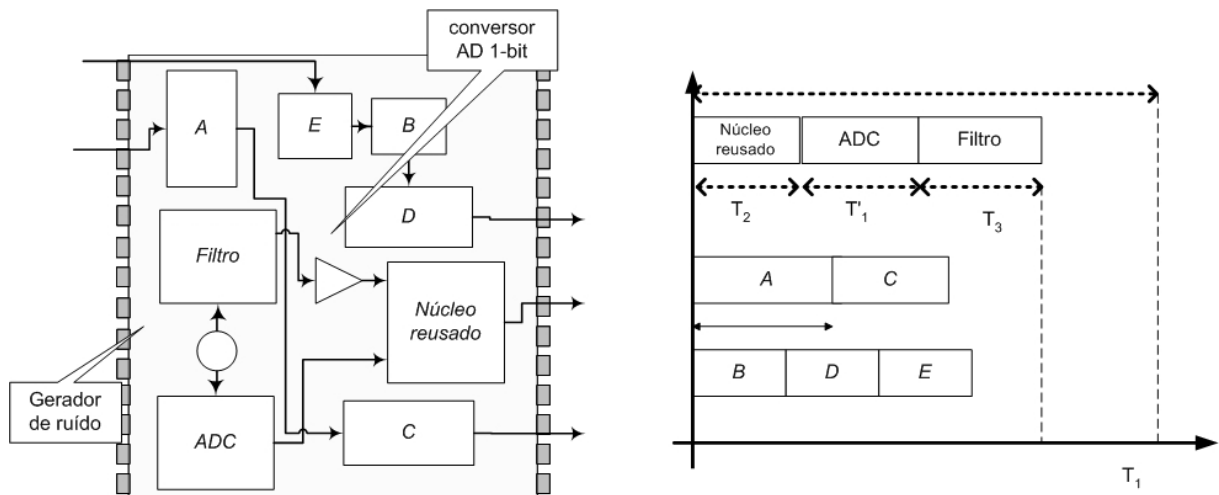


Figura 26 - Exemplo de TAMs para sistema misto hipotético e respectivo escalonamento, considerando método de teste baseado em ruído para os núcleos analógicos

Já no caso do BIST baseado em oscilação (Figura 27), a redução no tempo de teste dos núcleos analógicos pode ser ainda maior comparado ao caso do teste baseado em ruído, além de não precisar de vetores de teste. Entretanto, isto implica em um razoável acréscimo de área, devido às estruturas necessárias para se estabelecer a realimentação dos CUTs, além de um circuito extra para extração da amplitude da oscilação baseado em uma arquitetura sigma-delta. Cabe ressaltar que tal circuito é necessário para estimativa precisa da amplitude de oscilação, que por sua vez é necessária para se garantir uma cobertura de falhas satisfatória.

No escalonamento de núcleos associado a este método, o tempo de teste dos núcleos analógicos é reduzido ainda mais, restringindo o espaço de projeto e pesquisa da ferramenta. Assim, apesar de uma redução satisfatória no tempo de teste do sistema, outros custos relativos a outros recursos de teste podem inviabilizar a escolha deste método.

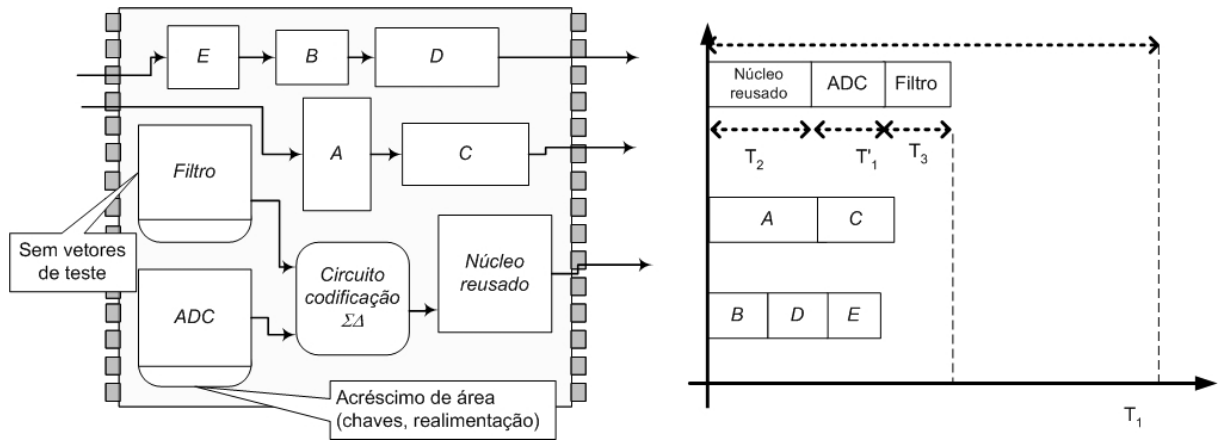


Figura 27 - Exemplo de TAMs para sistema misto hipotético e respectivo escalonamento, considerando método de teste baseado em oscilação para os núcleos analógicos

5 RESULTADOS EXPERIMENTAIS

Com o objetivo de validar o uso da ferramenta, a mesma é aplicada a exemplos de sistemas adaptados de alguns *benchmarks* disponíveis em Marinissen, Iyengar e Chakrabarty (2003), os quais incluem contribuições acadêmicas e industriais, visando assim mostrar que a metodologia proposta é abrangente o suficiente para ser utilizada por todos os segmentos de semicondutores. As adaptações consistem da inclusão de dois tipos de núcleos analógicos (ADCs e filtros), além de informações de precedência, indicando o núcleo a ser reusado durante o teste desses, e da vizinhança dos núcleos. Por simplicidade e sem perda de generalidade da metodologia, apenas os métodos de BIST baseado em oscilação e BIST baseado em ruído serão tratados.

5.1 RESUMO DESCRITIVO DOS *BENCHMARKS* UTILIZADOS

Alguns dos sistemas disponíveis em Marinissen, Iyengar e Chakrabarty (2003) foram adaptados através da adição de alguns núcleos de sinal misto, a saber, pares de filtros analógicos combinados com um conversor analógico-digital interpretado como um simples circuito codificador-decodificador. A nomenclatura dos sistemas é tal que o nome original é conservado e a letra **m** ao final indica a modificação. Na Tabela 2, estão resumidas algumas das especificações dos sistemas disponibilizadas nos *benchmarks* e alterados para o propósito deste trabalho.

Tabela 2 - Características dos *benchmarks* utilizados nas simulações

Sistemas	Número de Núcleos Digitais	Número de Núcleos Analógicos inseridos	No. E/S	Requisitos de Teste do processador reusado		
				Número de Cadeias <i>Scan</i>	Tamanho máximo de cadeias <i>scan</i>	Número de Padrões de teste
p93791m	32	6	248	48	168	409
p34392m	21	6	173	2	806	210
u226m	6	3	53	22	52	76
d695m	10	6	0	38	54	12

O sistema U226 já foi mencionado no capítulo anterior, e já apresenta núcleos mistos, de forma que é idêntico ao sistema U226m a ser considerado neste trabalho. O sistema D695 é

outra contribuição do âmbito acadêmico, da Universidade de Duke, Estados Unidos, enquanto que os outros sistemas, cujos nomes iniciam com a letra **P**, são contribuições do fabricante *Philips Semiconductors*. Como já foi salientado, devido à propriedade intelectual dos sistemas, não há informação sobre a funcionalidade dos núcleos, apenas sobre os seus requisitos de teste.

Outra diferença entre os sistemas mistos descritos é a razão entre a potência consumida pelos núcleos analógicos e a consumida pelos núcleos digitais. Ainda devido à falta de informações sobre o funcionamento dos blocos, foi necessário estimar o consumo dos blocos analógicos em função da quantidade de núcleos digitais do sistema original. Isto implica que, em sistemas como o P93791m, em que o número de núcleos digitais é muito maior do que o de núcleos analógicos, o consumo devido aos núcleos digitais é arbitrado maior do que o devido aos analógicos. Já no caso do U226m, o consumo dos circuitos analógicos é equivalente ao consumo dos digitais. Finalmente, no D695m, como há poucos núcleos digitais, o consumo devido aos analógicos é arbitrado maior do que o devido aos digitais. Com a descrição destes perfis de consumo de potência, pretende-se estudar a influência de tais perfis no escalonamento, quando restrições de dissipação são impostas.

5.2 RESULTADOS DAS SIMULAÇÕES USANDO A FERRAMENTA

O alvo dos experimentos é observar a evolução do custo de teste com a inclusão de núcleos analógicos no sistema, avaliando a eficácia do BIST baseado em reuso. Em uma primeira etapa, as simulações são realizadas variando o número de núcleos analógicos com BIST para cada sistema considerado, comparando os resultados obtidos em termos de tempo de teste do sistema, acréscimo em área devido à infra-estrutura de teste e número de pinos extras.

Ademais, o cálculo do tempo de teste resultante se baseia no número de ciclos de teste digital para um testador digital externo. Na ferramenta, o tempo de teste dos núcleos analógicos é expresso em termos de ciclos digitais, conforme já explicado no capítulo 4. Na situação em estudo, assume-se que as frequências de operação do bloco analógico e do processador digital reusado são 40kHz e 2MHz respectivamente, o que implica que um ciclo analógico corresponde a 50 ciclos digitais. Esta informação é usada na descrição do núcleo

analógico, expressando o número de padrões de teste analógico em termos de ciclos digitais. Assumindo, enfim, que o testador externo opera à mesma frequência do processador no sistema, pode-se calcular o tempo de teste total devido ao teste dos núcleos analógicos a partir do resultado obtido em número de ciclos.

A Tabela 3 a seguir mostra os resultados obtidos em simulação para os sistemas considerados, apresentados por Andrade Jr., Cota e Lubaszewski (2004b) e Andrade Jr. et al. (2004c). Em cada situação, um número diferente de pares filtro-ADC é adaptado para teste usando BIST baseado em ruído, desde zero até três pares, representando a totalidade dentro do sistema. No caso do U226m, como são apenas três núcleos analógicos, cada núcleo é considerado um par, para efeitos de simplificação na organização dos resultados na tabela.

Pelos resultados, pode-se observar alguns detalhes. Primeiro, a aplicação da estratégia de BIST parcialmente, isto é, a apenas alguns núcleos do sistema, sem redução no número de amostras, não se traduz em redução no tempo de teste do sistema todo. Isto porque o núcleo analógico que não teve seu tempo de teste reduzido permanecerá dominando o tempo de teste do sistema, como mostrado na Figura 25. Consequentemente, com mais intervalos de tempo disponíveis, a ferramenta foi capaz de reduzir significativamente a área acrescentada devido aos *wrappers*, embora dobrasse o número de pinos extras, no caso do U226m.

Segundo, a redução do número de padrões de teste permitida pelos métodos de BIST analógico só se traduz em redução do tempo de teste do sistema completo caso a técnica seja aplicada a todos os núcleos analógicos caros em termos de teste, como ocorre no caso dos sistemas U226m, D695m e P34392m. Para estes sistemas, ainda com a serialização devido ao reuso do mesmo processador por todos os blocos analógicos com BIST, o tempo de teste é menor do que o sistema original misto usando técnicas de teste convencional, com uma redução de 82,5%, 43,7% e 93,08%, respectivamente. O número de pinos extras requeridos se manteve o mesmo para os sistemas U226m e P34392m e diminuiu em 84,6% para o sistema D695m, enquanto que os resultados obtidos para a área extra também revelam uma conservação nos valores ou um pequeno aumento. Ainda sobre o D695m, a diminuição do número de pinos extras pode estar associada ao aumento no tempo de teste. Com mais tempo livre disponível para exploração do espaço de projeto, a ferramenta conseguiu uma solução com tal redução do número de pinos.

Entretanto, para o sistema P93791m, a serialização dos blocos analógicos com BIST e do processador reusado causam um aumento de 1,31% no tempo de teste em relação ao teste convencional. Analisando mais detalhadamente o escalonamento resultante, observa-se que o teste do processador reusado toma a maior parte do tempo de teste do sistema. Junto com a serialização do teste, isto impede uma maior redução no tempo de teste para o sistema P93791m.

Tabela 3 - Resultados da simulação do SOCs para BIST baseado em ruído, variando o número de núcleos com BIST

Nome do SOC	Recurso de teste	Sistema original	Sistema Misto			
			Sem BIST	1 par BIST	2 pares BIST	3 pares BIST
U226m	Tempo de teste (ciclos)	1.363.968	6.000.000	6.000.000	6.000.000	1.048.576
	Pinos extras	3	6	6	6	6
	Acréscimo de área (%)	0	0,1	0,2	0,2	0,2
D695m	Tempo de teste (ciclos)	37.089	12.409.599	6.000.000	6.000.000	6.989.904
	Pinos extras	3	13	24	12	2
	Acréscimo de área (%)	3	7	1,7	4,3	6,5
P93791m	Tempo de teste (ciclos)	939.145	6.000.000	6.000.000	6.000.000	6.078.966
	Pinos extras	35	43	44	43	49
	Acréscimo de área (%)	2	5	1,7	7,3	10,4
P34392m	Tempo de teste (ciclos)	814.378	42.170.698	25.395.134	14.214.334	2.917.097
	Pinos extras	19	6	6	6	7
	Acréscimo de área (%)	1	1,40	1,42	1,42	2,88

Segundo, a redução do número de padrões de teste permitida pelos métodos de BIST analógico só se traduz em redução do tempo de teste do sistema completo caso a técnica seja aplicada a todos os núcleos analógicos caros em termos de teste, como ocorre no caso dos sistemas U226m, D695m e P34392m. Para estes sistemas, ainda com a serialização devido ao reuso do mesmo processador por todos os blocos analógicos com BIST, o tempo de teste é menor do que o sistema original misto usando técnicas de teste convencional, com uma redução de 82,5%, 43,7% e 93,08%, respectivamente. O número de pinos extras requeridos se

manteve o mesmo para os sistemas U226m e P34392m e diminuiu em 84,6% para o sistema D695m, enquanto que os resultados obtidos para a área extra também revelam uma conservação nos valores ou um pequeno aumento.

Entretanto, para o sistema P93791m, a serialização dos blocos analógicos com BIST e do processador reusado causam um aumento de 1,31% no tempo de teste em relação ao teste convencional. Analisando mais detalhadamente estes resultados, observa-se que o teste do processador reusado toma a maior parte do tempo no escalonamento. Junto com a serialização do teste, isto impede uma maior redução no tempo de teste para o sistema P93791m.

Além disso, é importante também considerar a potência consumida durante o teste, já que uma dissipação excessiva de calor pode danificar o chip. A estratégia de reuso pode tornar esse problema mais crítico, já que mais de um bloco é ativado ao mesmo tempo durante o procedimento de teste. Outro aspecto é que SOCs estão cada vez mais presentes em equipamentos portáteis, e, caso o dispositivo inclua autoteste realizado na inicialização, a duração da bateria entre recargas é diminuída, o que não é desejável. Enfim, a mesma metodologia utilizada anteriormente é realizada, comparando os resultados para o caso em que a potência dissipada é limitada em 30% da soma das potências de todos os núcleos operando isoladamente. Estes resultados estão expostos na Tabela 4.

Assim como os resultados obtidos e expostos na Tabela 3, observa-se mais uma vez a tendência de redução do tempo de teste em relação ao teste convencional quando a todos os núcleos é aplicada a técnica de BIST. Além disso, uma das conseqüências imediatas do aumento do limite de potência é o aumento no tempo de teste dos sistemas, já que tal limite exclui um maior paralelismo no teste dos núcleos, fazendo com que menos núcleos sejam testados simultaneamente. Comparando a Tabela 3 com a Tabela 4, observa-se que, de uma forma geral, o tempo de teste resultante dos sistemas com limitação em potência é maior do que aqueles em que não há tal limitação. Nos casos em que tal diferença não é observada, como no sistema P34392m, uma breve análise do escalonamento sem restrições revela que a potência consumida durante o procedimento de teste não era elevada o suficiente para violar o limite arbitrado em 30% da potência total do sistema. Desse modo, a rigor, o algoritmo deveria encontrar a mesma solução para ambos os casos, entretanto, as restrições fixadas nas condições iniciais levaram o algoritmo a explorar o espaço de projeto de forma distinta, chegando a uma solução inesperadamente melhor.

Tabela 4 - Resultados da simulação do SOCs para BIST baseado em ruído, variando o número de núcleos analógicos com BIST e limite de 30% na potência máxima durante o teste

Nome do SOC	Recurso de teste	Sistema original	Sistema Misto			
			Sem BIST	1 par BIST	2 pares BIST	3 pares BIST
U226	Tempo de teste (ciclos)	1.363.968	12.112.446	7.161.022	7.161.022	1.570.622
	Pinos extras	3	6	6	6	6
	Acréscimo de área (%)	0	0	11,9	11,2	35,5
D695	Tempo de teste (ciclos)	25.147	30.409.597	11.999.999	6.000.000	7.783.844
	Pinos extras	94	10	17	12	0
	Acréscimo de área (%)	5,3	3,2	6,2	4,3	8,3
P93791	Tempo de teste (ciclos)	1.615.500	8.290.685	7.827.909	6.000.000	8.336.943
	Pinos extras	11	51	42	26	34
	Acréscimo de área (%)	23	4,6	6,3	5,6	11
P34392	Tempo de teste (ciclos)	1.368.792	37.164.578	30.989.898	19.978.742	2.784.670
	Pinos extras	49	6	6	7	5
	Acréscimo de área (%)	0,76	1,28	1,41	1,41	2,12

Deste exemplo, pode-se depreender que, dependendo do consumo de potência dos núcleos individualmente, pode não haver diferença prática entre um escalonamento sem ou com limites de potência de um dado valor. Obviamente, os efeitos esperados se tornam mais perceptíveis à medida que este limite se torna mais rigoroso, por exemplo, reduzindo-o para 10% do consumo total.

Ademais, a estratégia elaborada não é restrita apenas ao BIST baseado em ruído, mas, com uma modelagem adequada, pode abranger outros métodos de BIST disponíveis na literatura. Por exemplo, o BIST baseado em oscilação também requer um subsistema de processamento para avaliação da sua resposta, indicando a necessidade da precedência. Seu tempo de teste é dado em termos de ciclos analógicos, que, para determinação com baixa incerteza da amplitude e frequência, requer 30 ciclos analógicos. Embora ainda não tenha sido considerado no modelo, ao avaliar os resultados, é importante ter em mente que tal estratégia apresenta alto acréscimo em área devido ao circuito codificador delta-sigma usado para

transformar o sinal analógico oscilante num trem de pulsos digitais que será calculado pelo microprocessador ou DSP. Este acréscimo não é contabilizado na solução encontrada pela ferramenta.

Dessa forma, numa segunda etapa, outras simulações são realizadas, trocando o BIST baseado em ruído pelo baseado em oscilação, e os resultados obtidos para os sistemas com BIST para todos os núcleos e sem BIST são mostrados na Tabela 5.

Tabela 5 - Resultados da simulação do SOCs para BIST baseado em oscilação, variando o número de núcleos com BIST

Nome do SOC	Recurso de teste	Sistema original	Sistema Misto	
			Sem BIST	3 pares BIST
U226	Tempo de teste (ciclos)	1.363.968	6.000.000	1.048.576
	Pinos extras	3	6	6
	Acréscimo de área (%)	0	0,1	0,06
D695	Tempo de teste (ciclos)	37.089	12.409.599	85.182
	Pinos extras	3	13	3
	Acréscimo de área (%)	3	7	2,88
P93791	Tempo de teste (ciclos)	939.145	6.000.000	407.569
	Pinos extras	35	43	108
	Acréscimo de área (%)	2	5	6
P34392	Tempo de teste (ciclos)	814.378	42.170.698	2.917.097
	Pinos extras	19	6	7
	Acréscimo de área (%)	1	1,40	2,88

Devido à redução abrupta no número de ciclos analógicos necessários para extração e interpretação dos parâmetros de teste (de 120.000 para apenas 30), a redução no tempo de teste chega a 93,2% e 99,3% para os sistemas P93791m e D695m. Entretanto, observa-se no caso do P93791m, a redução drástica no tempo de teste em número de ciclos provoca um número alto também no número de pinos extras requeridos, o que pode ser proibitivo para a definição da estratégia de teste do sistema. Nesse caso, a estratégia de BIST baseado em

oscilação se torna desinteressante, apesar da redução no tempo de teste. Soluções intermediárias podem ser obtidas se outros métodos de teste analógicos forem aplicados. No caso do U226m, o número de ciclos para o teste não apresentou alteração, e para o P34392 aumentou, em relação à estratégia baseada em ruído. Nesses casos, a diferença do número de ciclos para o teste dos núcleos analógicos entre as duas estratégias, em relação ao número de padrões de teste para os núcleos digitais, não é grande o suficiente para causar ganhos em termos de tempo de teste na solução encontrada para cada um desses sistemas.

Também é importante ressaltar que a frequência do testador externo (ou do processador interno, se este for reusado para fins de teste) inicialmente usada de 2MHz é muito pequena e não condiz com a realidade da velocidade de processamento dos circuitos digitais modernos. Por isso, o valor da frequência do processador reusado, ou do testador externo, se o primeiro for testado externamente, foi alterada para 20MHz e 50MHz respectivamente, e mais simulações foram realizadas usando a ferramenta, com o intuito de observar o impacto destas modificações. Tal alteração não reduzirá o tempo de teste dos núcleos analógicos, uma vez que o processo que mais consome tempo é o de obtenção e aquisição das respostas analógicas de teste e estes independem da frequência do processador ou testador. Entretanto, o tempo de teste dos núcleos digitais é reduzido na mesma proporção, de modo que há um ganho devido à redução no tempo de teste do processador reusado, diminuindo os efeitos indesejados da serialização. Além disso, como o tempo mínimo de teste do sistema é afetado pela alteração do teste do processador reusado, uma pesquisa diferente pelo espaço de projeto será realizada, o que possibilitará uma melhor otimização dos recursos de teste. Os resultados obtidos para os diversos sistemas, com todos os núcleos com BIST e a nova frequência, estão dispostos na Tabela 6. Para uma melhor comparação entre os diferentes cenários, as unidades foram convertidas de número de ciclos para tempo em milissegundos.

Os resultados mostram que de fato há um ganho significativo em termos de tempo de teste quando a frequência é aumentada de 2MHz para 20MHz. Para o BIST baseado em oscilação, a redução foi de 94,5%, 91,7%, 90% e 75,8% para os sistemas P34392m, P93791m, U226m e D695m respectivamente. Entretanto, como o efeito principal da alteração da frequência de operação do testador externo é na redução do tempo de teste do processador reusado e na exploração diferente do espaço de projeto, observa-se que, da frequência de 20MHz para 50MHz, ainda há redução no tempo de teste do sistema, mas não na mesma

proporção que o cenário anterior. As reduções obtidas foram de 56,3%, 23,1%, 59,9% e 7,61% para os sistemas P34392m, P93791m, U226m e D695m respectivamente. A mesma tendência é observada considerando o BIST baseado em ruído. O que se pode concluir é que aumentar a frequência de teste indefinidamente pode não acarretar em ganhos significativos no tempo de teste, uma vez que o tempo de teste dos núcleos analógicos se torna muito maior do que o dos digitais.

Tabela 6 - Resultados de simulação com teste rodado a 20MHz e a 50MHz

Frequência		F=2MHz			f=20MHz			f=50MHz		
Sistema	Método	Sem BIST	Baseado Oscilação	Baseado em ruído	Sem BIST	Baseado Oscilação	Baseado em ruído	Sem BIST	Baseado Oscilação	Baseado em ruído
P34392m	Tempo (ms)	21085	1459	1459	6000	80,3	1266	6000	35,1	1437
	Pinos	19	7	7	29	7	7	29	6	6
	AO (%)	1,4	2,88	1,44	1	2,85	1,4	1,02	1,94	1,4
P93791m	Tempo (ms)	3000	203,8	4168	3000	16,9	1394	3000	13,0	1295
	Pinos	3	108	34	33	31	15	54	54	12
	AO (%)	5	6	11	11	4,8	4	2,69	6,56	5,08
U226m	Tempo (ms)	3000	524,3	524,3	3000	52,4	209,2	3000	21,0	206,6
	Pinos	3	6	6	6	6	6	6	6	6
	AO (%)	0,1	0,06	0,2	0,14	0,13	35,47	0,14	0,138	35,47
D695m	Tempo (ms)	6205	42,6	3495	9001	10,3	2459	9000	9,5	2458
	Pinos	35	3	2	8	4	2	8	2	2
	AO (%)	7	2,8	6,5	4,94	1,66	6,91	4,94	6,91	6,91

5.3 RESULTADOS DAS SIMULAÇÕES EXPLORANDO PARALELISMO DE TESTE

Apesar das vantagens apresentadas em termos de tempo de teste com a solução anterior, é desejável que a serialização do teste dos núcleos seja evitada, seja devido à possibilidade de uma maior redução no tempo de teste total do sistema, seja devido a fatores de dissipação de potência excessiva. Uma das possibilidades, que é razoável em alguns sistemas atualmente projetados, é a existência de mais de um processador ou núcleo complexo o suficiente que execute tarefas de computação ou avaliação de resposta adequadas ao teste de

núcleos analógicos. Com um maior número de núcleos reusados para escalonamento do teste analógico, obviamente a serialização será reduzida, e com isso o tempo de teste.

Tal situação foi considerada na quinta etapa dos experimentos realizados, de forma que outros dois núcleos digitais presentes no sistema foram assumidos como capazes de efetuar a avaliação da resposta ao teste, e cada par de filtro e ADC está associado a apenas um núcleo reusado, de forma a minimizar o efeito da serialização (Figura 28). Os resultados obtidos podem ser vistos na Tabela 7, comparando com os resultados obtidos para apenas um núcleo reusado, considerando, no momento, apenas a estratégia baseada em ruído. Resultados para a situação de restrição de potência também foram obtidos, uma vez que, com um maior paralelismo no teste, espera-se que tais restrições tenham um efeito maior do que o verificado na Tabela 3 e na Tabela 4, por exemplo.

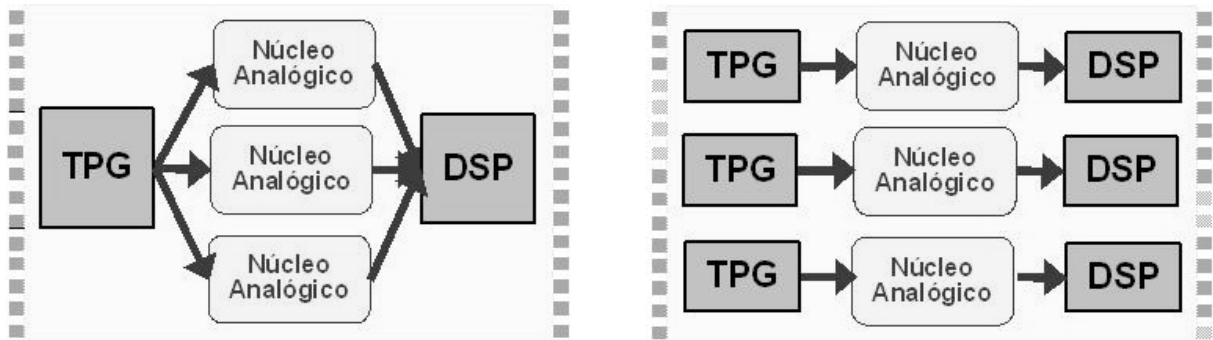


Figura 28 - Aumento do paralelismo de teste do sistema através do aumento do número de núcleos disponíveis para geração de padrões de teste e análise da resposta

Além disso, os diferentes perfis de consumo de potência entre núcleos analógicos e digitais de um SOC podem influenciar no escalonamento dos núcleos, limitando o paralelismo, como ilustrado na Figura 29. Por exemplo, se o consumo de potência devido aos núcleos analógicos crescer em relação aos núcleos digitais, como aqueles ainda requerem um maior tempo de teste, mesmo com o uso de BIST analógico, o paralelismo de teste entre os núcleos analógicos, conseqüentemente o tempo de teste do sistema, será comprometido.

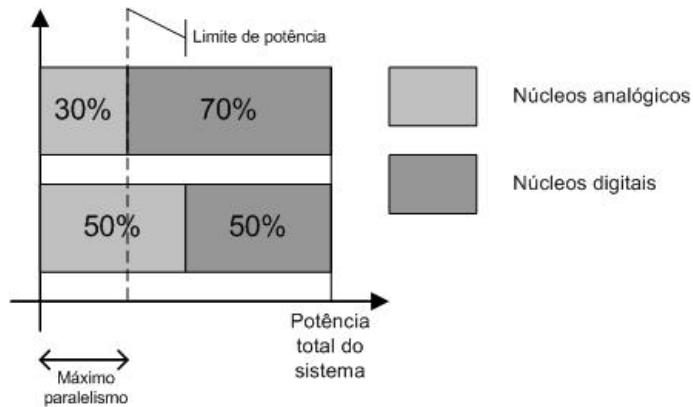


Figura 29 - Redução no paralelismo devido ao escalonamento de teste com restrições de potência

Tabela 7 - Resultados experimentais para p93791m e p34392 com 3 processadores disponíveis

Limite Potência	No. processadores	Recursos	p93791m	p34392m
Infinito	3	Tempo (ciclos)	4.142.123	1.504.507
		Tempo (ms)	2071	752
		pinos	49	15
		AO (%)	3,7	1,74
	1	Tempo (ciclos)	6.078.966	2.917.097
		Tempo (ms)	3039	1459
		pinos	19	7
		AO (%)	10,4	1,4
30%	3	Tempo (ciclos)	6.847.072	2.228.266
		Tempo (ms)	3424	1114
		pinos	44	18
		AO (%)	2,1	1,42
	1	Tempo (ciclos)	8.336.943	2.784.670
		Tempo (ms)	4168	1392
		pinos	34	5
		AO (%)	11	2,2

O sistema p93791m2 é idêntico ao p93791m, exceto pelo consumo de potência dos núcleos analógicos, que é igual ao consumo de potência dos núcleos digitais. Como não há informações detalhadas sobre o funcionamento dos blocos digitais nem foram requeridos detalhes sobre os analógicos, tal suposição pode ser feita. A Tabela 8 mostra os resultados obtidos da simulação do sistema modificado. Verifica-se que a redução no tempo de teste devido ao maior número de processadores disponíveis é maior no sistema original P93791m do que no sistema P93791m2, sendo a mesma de 17,8% para o primeiro e 4,7% para o último. Isto mostra que, de fato, o aumento do consumo de potência devido aos núcleos analógicos

provoca uma menor redução do tempo de teste do sistema, o que pode ser explicado pelo menor paralelismo de teste. O número de pinos extras não altera de um caso para o outro, enquanto que se obteve uma redução na área extra devido às estruturas de teste no P93791m, o mesmo não ocorrendo para o P93791m2. Esta diferença pode ser fruto dos valores absolutos do tempo de teste, já que no caso do P93791m2, o tempo de teste do sistema é menor, permitindo uma menor otimização dos recursos.

Tabela 8 - Resultados da simulação do p93791m2, com reuso de processadores, diferentes proporções de potência entre núcleos analógicos e digitais

Sistema	Perfil de potência	No. de Process.	Tempo de teste (ciclos)	Pinos extras	Acréscimo de área (%)
P93791m	30%/70%	3	6.847.072	43	3,4
		1	8.336.943	34	11
P93791m2	50%/50%	3	4.813.967	42	7,9
		1	5.053.363	34	4,4

Entretanto, o paralelismo efetuado por um número maior de núcleos digitais processados embarcados no sistema pode não ser a realidade de muitos projetos existentes no mercado. Com isso, outras formas de paralelismo no teste devem ser buscadas. Analisando o método do histograma linear, observamos que este apresenta a propriedade da preempção no momento do processamento dos dados de teste, isto é, uma vez garantida a aquisição coerente da resposta de teste, o processamento dos dados pode ser interrompido no processador, suas variáveis armazenadas e a operação retomada num momento posterior. Assumindo que o cálculo da FFT é feito ponto a ponto, e leve, em média, 50 ciclos de relógio para ser efetuado (SOUZA JR.; CARRO, 2003), é razoável supor que um processador operando a uma frequência 500 vezes maior do que a frequência de Nyquist dos núcleos analógicos (por exemplo, respectivamente, 20MHz e 40kHz) consegue gerenciar o teste de até 10 núcleos analógicos diferentes entre duas amostras do mesmo núcleo, sem se considerar o tempo necessário para chaveamento das cadeias de processamento interno do núcleo reusado. A Figura 30 ilustra como se dá esse paralelismo em um exemplo com seis amostras a serem processadas.

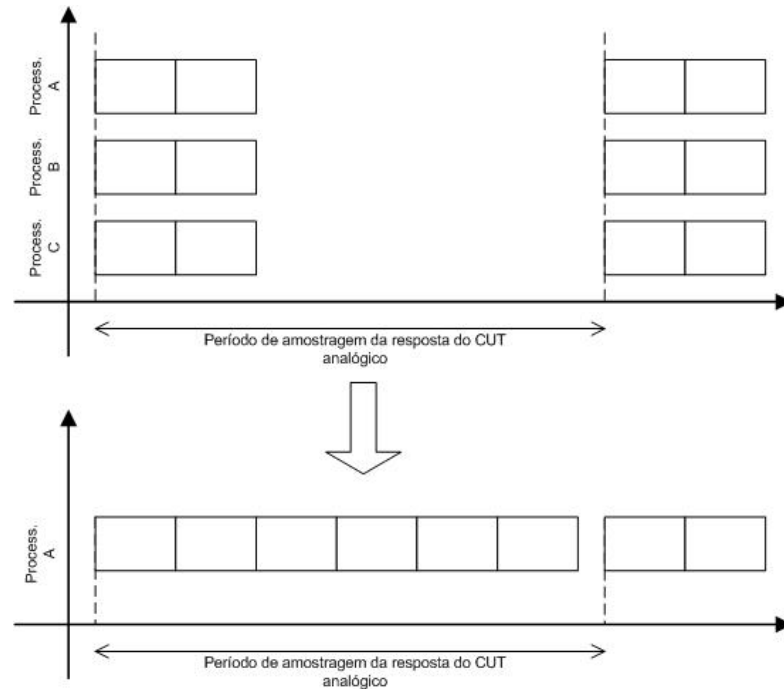


Figura 30 - Ilustração da exploração do paralelismo no teste de núcleos analógicos usando o tempo ocioso do processador reusado

Para se considerar essa hipótese, o algoritmo sofre mais uma alteração, tal que todos os núcleos analógicos podem reusar o processador ao mesmo tempo. A Tabela 9 mostra os resultados obtidos com os sistemas P93791m, P34392m e D695m, quando a estratégia de BIST permite o paralelismo de teste de núcleos analógicos com apenas um processador reusado disponível. Com isso, para este trabalho, apenas o método baseado em ruído pode ser adaptado.

Tabela 9 - Resultados da simulação para os sistemas P93791m e P34392m explorando o paralelismo com um único processador, usando intervalos de tempo entre os processamentos das amostras

Sistema	Limite Potência	f=2MHz			f=20MHz		
		Tempo (ms)	pinos	AO (%)	Tempo (ms)	pinos	AO (%)
P93791m	Infinito	444,86	51	7,11	279,74	15	5,2
	30%	415,01	64	7,54	370,13	11	5,47
P34392m	Infinito	723,98	6	5,11	213,35	28	1,42
	30%	1033,54	7	4,91	468,71	10	4,7
D695m	Infinito	212,55	22	6,79	205,57	22	6,79
	30%	819,93	22	8,55	819,27	22	8,55

Finalmente, comparando a Tabela 9 com a Tabela 7 e a Tabela 6, apenas para o método baseado em ruído, observa-se que o ganho em termos de tempo de teste para a última estratégia é superior no caso em que não haja limite de potência, já que todos os núcleos podem ser testados em paralelo. A redução para o sistema P34392m em relação ao BIST de todos os núcleos analógicos usando apenas um processador sem exploração do paralelismo no tempo foi de 50,3%, enquanto que, no caso do P93791m essa redução foi de 89,3%. Comparando com a solução explorando o paralelismo através do aumento do número de núcleos reusados, considerando a mesma frequência de 2MHz, a redução para o sistema P34392m foi de apenas 3,7%, enquanto que no P93791m essa redução chega a 78,5%.

A dissipação de potência durante o teste é proporcional à quantidade de processamento, que é maior neste caso uma vez que todos os núcleos analógicos estão sendo testados ao mesmo tempo. Por isso, observa-se uma diferença maior entre os resultados sem limite de potência e com limite de 30%, devido a este acréscimo na potência dissipada. É interessante notar que o efeito da limitação de potência é mais visível quando a frequência de teste aumenta. Tomando o exemplo do sistema P93791m, o tempo de teste para o caso em que a frequência é igual a 2MHz com limite de potência é menor do que sem esse limite, o que pode ser explicado pelo fato de que nesse sistema o consumo de potência pelos blocos analógicos corresponde a 30% da potência total do sistema, e assim o limite não interfere significativamente no paralelismo, mas permite à ferramenta explorar o espaço de projeto de forma diferente. Finalmente, só é necessário um processador passível de reuso de forma a paralelizar o teste dos núcleos analógicos.

Outro aspecto a ser depreendido da Tabela 9 é a maior redução no tempo de teste com o aumento da frequência de 2MHz para 20Mhz, como já havia sido observado nas situações anteriormente discutidas. Adicionalmente, como a dissipação de potência aumenta com o aumento da frequência, observa-se também que, no que tange ao tempo de teste, a diferença entre a solução sem limite de potência e com limite de potência aumenta, evidenciando mais uma vez como a busca pelo melhor escalonamento reage em casos de dissipação excessiva.

6 CONCLUSÕES E TRABALHOS FUTUROS

Esta dissertação apresenta uma nova proposta para o teste de SOCs de sinal misto, combinando as vantagens de uma estratégia de planejamento de teste no início do fluxo de projeto, proporcionando uma melhor exploração do espaço de projeto, com a redução de custos de teste obtidos através de técnicas de BIST.

Com o intuito de avaliar esta proposta, uma ferramenta utilizada para auxílio no planejamento de teste de SOCs digitais foi adaptada de forma a permitir a inclusão de núcleos analógicos no planejamento de teste, bem como o BIST como método de teste disponível para tais núcleos. Os sistemas utilizados pela ferramenta são adaptações dos sistemas disponíveis nos *benchmarks* ITC'02.

Os resultados das simulações mostram a TAM global obtida, utilizando heurísticas a fim de conseguir uma solução próxima da ótima, as soluções são avaliadas em termos de tempo de teste em número de ciclos, número de pinos extras e acréscimo de área requerido para inserção das estruturas de teste, como *wrappers*.

As poucas referências na literatura sobre o planejamento de teste de SOCs mistos são uma evidência de que esta é uma área nova e com amplas possibilidades de avanços e desenvolvimento de novos paradigmas de metodologias de teste. Neste trabalho, foi dada mais uma contribuição para o debate de soluções e os resultados alcançados e divulgados mostram que a idéia é promissora.

Os resultados obtidos com a ferramenta adaptada, batizada de REBATE-MIX, mostram, em primeiro lugar, que é possível obter uma sensível redução no tempo de teste do sistema em relação ao sistema com teste convencional com núcleos analógicos, graças à redução do tempo de teste destes. Para isso, no entanto, a estratégia de teste deve ser aplicada a todos os núcleos analógicos, pelo contrário, o núcleo testado da maneira convencional permanecerá dominando o tempo de teste, sem apresentar vantagem em relação ao cenário sem BIST. Mesmo quando essa redução não é possível, devido à serialização do teste, outros custos são reduzidos, como o número de pinos ou o acréscimo em área, uma vez que os núcleos analógicos dominam o tempo de teste, proporcionando intervalos de tempo livre para otimização desses parâmetros. Outro ponto interessante é que o reuso de núcleos do sistema para geração e avaliação de sinais de teste permite a redução dos custos do testador externo,

através do reaproveitamento de estruturas já disponíveis no sistema para executar tarefas antes realizadas pelo ATE.

Uma solução para superar a serialização é o reuso de um número maior de processadores internos, se disponíveis. Tal estratégia aumenta o paralelismo de teste, mas implica outros custos, como uma maior dissipação de potência, além de estar condicionada à existência de núcleos disponíveis no sistema para tal função. Outra forma de explorar o paralelismo é aproveitar o tempo livre do processador entre duas amostras, de forma que mais de um núcleo possa ser testado ao mesmo tempo sem necessidade de um número maior de processadores. Os ganhos em termos de tempo de teste se mostram adequados, mas para isso há o custo de dissipação excessiva de potência devido ao fato do processador trabalhar durante todo o tempo, o que pode diminuir o paralelismo no teste, caso limites de potência sejam considerados.

Outra vertente explorada para conseguir maior redução no tempo de teste do sistema é através da variação da frequência de operação do núcleo reusado. Os experimentos foram realizados para valores de frequência iguais a 2MHz, 20MHz e 50MHz, verificando que os ganhos em termos de redução de tempo de teste diminuem à medida que a frequência do testador aumenta, pois seu efeito maior é na redução do tempo de teste dos núcleos digitais. Entretanto, assim como na solução explorando o paralelismo, o preço a pagar é uma maior dissipação de potência, devido à maior frequência de operação, indicando assim que há um valor ótimo de frequência do testador ou do núcleo reusado que atende o compromisso entre esses dois objetivos.

Durante os experimentos realizados, foi verificada também a influência da limitação de dissipação de potência durante o teste, o que é uma situação mais próxima da real. Observa-se que tal restrição só se torna perceptível quando limita o paralelismo de teste, assim, depende do perfil de dissipação de potência dos núcleos analógicos e digitais. Observa-se que, à medida que a potência dissipada pelos circuitos analógicos aumenta em relação aos circuitos digitais, o paralelismo no teste é prejudicado e assim a redução do tempo de teste do sistema, quando se aumenta o número de núcleos reusados, é menor.

Além desses aspectos, foram comparadas duas metodologias de BIST analógico, a saber, BIST baseado em ruído e baseado em oscilação. Observa-se que as otimizações obtidas usando BIST baseado em oscilação proporcionam uma redução de tempo de teste da ordem

de 90%, ao custo do aumento do número de pinos extras, enquanto que os resultados dos sistemas que aplicam BIST baseado em ruído apresentam uma redução do tempo de teste menor, com custos em termos de número de pinos mais razoáveis. Dessa forma, o projetista pode escolher qual solução é mais adequada para o sistema em questão. Outros métodos, embora não tenham sido simulados, podem ser facilmente descritos e utilizados na ferramenta, aumentando a flexibilidade do projetista na definição da metodologia de teste. Aumentando o escopo de metodologias de teste, pode-se também implementar a possibilidade de adoção de um barramento de teste misto baseado no padrão IEEE 1149.4, de modo que a ferramenta otimizasse as TAMs e o escalonamento considerando a existência desse barramento.

Os resultados alcançados abrem possibilidades de novos trabalhos, como a extração dos resultados obtidos no escalonamento e da definição das TAMs para uma variedade de métodos de BIST analógico, cruzando com a avaliação qualitativa apresentada na Tabela 1. Com isso, a idéia é prosseguir com uma comparação quantitativa, em que o engenheiro de teste estabelece quais critérios são os mais críticos (se tempo de teste, pinos ou área), e então um algoritmo avaliaria e retornaria com uma pontuação dos métodos considerados em termos dos custos envolvidos, que serviria de ajuda ao integrador de sistema no momento do projeto.

Outras classes de circuitos analógicos, como *mixers*, são muito importantes em aplicações ligadas a comunicações sem fio, e por isso também devem ser contempladas nos próximos trabalhos, de forma a obter uma comparação direta com os resultados obtidos com os sistemas apresentados em Sehgal, Ozev e Chakrabarty, 2003.

Ainda não estão muito claros alguns efeitos encontrados, como o fato de soluções com restrições de potência serem melhores do que aquelas sem as mesmas restrições. Embora o motivo para tanto possa ser atribuído a uma exploração diferente no espaço de projeto, o mais adequado seria que a melhor solução fosse logo encontrada no caso sem restrições. Entretanto, há que se considerar que a ferramenta utilizada é uma adaptação de outra ferramenta desenvolvida para SOCs digitais, e assim, haveria espaços para aperfeiçoamentos que levem a resultados mais previsíveis.

REFERÊNCIAS

- ACEVEDO, G. O. D.; RAMIREZ-ANGULO, J. Built-in Self-Test Scheme for On-chip Diagnosis, Compliant with the IEEE 1149.4 Mixed-Signal Test Bus Standard. In: IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS (ISCAS) 2002, Scottsdale, AZ, EEUU. **Proceedings...** New York: IEEE, mai. 2002, p.149-152. v. 1
- ALEGRIA, F. et al. ADC Histogram Test by Triangular Small-Waves. In: IEEE INSTRUMENTATION AND MEASUREMENT TECHNOLOGY CONFERENCE (IMTC), 18., 2001, Budapeste, Hungria. **Proceedings...** New York: IEEE, 2001, p.1690-1695. v. 3
- ANDRADE JR., A.; CARRO, L.; LUBASZEWSKI, M. THD Testing of Analog Circuits Using Autocorrelation Sequence. In: IEEE LATIN AMERICAN TEST WORKSHOP, 5., 2004, Cartagena, Colômbia. **Digest of papers...** [S.l.: s.n.], mar. 2004a. p. 140-145.
- ANDRADE JR., A.; COTA, E.; LUBASZEWSKI, M. Improving Mixed-Signal SOC Testing: a power-aware reuse-based approach with analog BIST. In: SIMPÓSIO BRASILEIRO DE CONCEPÇÃO DE CIRCUITOS INTEGRADOS (SBCCI), 17., 2004, Porto de Galinhas, Brasil. **Proceedings...** NewYork: ACM, set. 2004b. p. 105-110.
- ANDRADE JR., A. et al. Test Planning for Mixed-Signal SoC and Analog BIST: a case study. In: DESIGN OF CIRCUITS AND INTEGRATED SYSTEMS (DCIS), 19., 2004, Bordeaux, França. **Proceedings...** [S.l.: s.n.], 2004c. p. 567-572.
- ARABI, K.; KAMINSKA, B. Testing Analog and Mixed-Signal Integrated Circuits Using Oscillation-Test Method. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, Nova York, v. 16, n. 7, p. 745-753. jul. 1997.
- AZAÏS, F. et al. Implementation of a Linear Histogram BIST for ADCs. In: DESIGN, AUTOMATION AND TEST IN EUROPE CONFERENCE AND EXHIBITION (DATE), 2001, Munique, Alemanha. **Proceedings...** Los Alamitos: IEEE Computer Society, 2001. p. 590-595.
- AZAÏS, F. et al. Estimating Static parameters of AD converters from Spectral Analysis. In: IEEE INTERNATIONAL MIXED-SIGNAL TESTING WORKSHOP (IMSTW), 8., 2002, Montreaux, Suíça. **Digest of Papers**, [S.l.:s.n.], 2002. p.165-169.
- BALEN, T. R. et al. Testing the Configurable Analog Blocks of Field Programmable Analog Arrays. In: INTERNATIONAL TEST CONFERENCE, 35., 2004, Charlotte, USA. **Proceedings...** Washington D.C.: International Test Conference, out. 2004. p. 893-902.
- BARNHART, C. et al. OPMISR: the foundation for compressed ATPG vectors. In: INTERNATIONAL TEST CONFERENCE (ITC), 32., 2001, Baltimore, USA. **Proceedings...** Los Alamitos: IEEE Computer Society, out. 2001. p. 748-757.
- BENNETTS, B. Status of IEEE Testability Standards 1149.4, 1532 and 1149.6. In: DESIGN, AUTOMATION AND TEST IN EUROPE CONFERENCE AND EXHIBITION (DATE),

2004, Paris, França. **Proceedings...** Los Alamitos: IEEE Computer Society 2004, fev. 2004, p.1184-1189. v. 2

BERGAMASCHI, R.; LEE, W. R.; Designing Systems-on-Chip Using Cores. In: ACM/IEEE DESIGN AUTOMATION CONFERENCE (DAC), 37., 2000, Los Angeles, USA. **Proceedings...** New York: ACM, jun. 2000. p. 420-425.

BERNARD, S. et al. A High Accuracy Triangle-Wave Signal Generator for On-Chip ADC Testing. IEEE EUROPEAN TEST WORKSHOP (ETW), 7., 2002, Corfu, Greece. **Digest of papers.** [S.l.: s.n.], mai. 2002. p. 365-370.

CALVANO, J. V.; ALVES, V. C.; LUBASZEWSKI, M. S. Fault Detection Methodology and BIST Method for 2nd Order Butterworth, Chebyshev and Bessel Filter Approximations. In: IEEE VLSI TEST SYMPOSIUM (VTS), 18., 2000, Montreal, Canada. **Proceedings...** Los Alamitos: IEEE Computer Society, abr. 2000. p. 319-324.

CALVANO, J.V. et al. Filters Designed for Testability Wrapped on the Mixed-Signal Test Bus. In: IEEE VLSI TEST SYMPOSIUM (VTS), 20., 2002, Moterey, USA, **Proceedings...** Los Alamitos: IEEE Computer Society, mai. 2002. p. 201-206.

CHAKRABARTY, K. Test Scheduling for Core-Based Systems. In: IEEE/ACM INTERNATIONAL CONFERENCE ON COMPUTER AIDED DESIGN (ICCAD), 1999, San Jose, CA, USA. **Proceedings...** New York: IEEE, nov. 1999. p. 391-394.

CHAKRABARTY, K. Test Scheduling for Core-Based Systems Using Mixed-Integer Linear Programming. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, USA, v. 19, n. 10, p. 1163-1174, out. 2000.

CHATTERJEE, A.; KIM, B.C.; NAGI, N. DC Built-In Self-Test for Linear Analog Circuits. **IEEE Design & Test of Computers**, Los Alamitos: IEEE Computer Society, v. 13, n. 2, p. 26-33, jun. 1996.

CHOU, R.; SALUJA, K; AGRAWAL, V. Scheduling Tests for VLSI Systems under Power Constraints. **IEEE Transactions on VLSI Systems**, EEUU, v. 5, n. 2, p. 175-185, jun. 1997.

COTA, E. et al. Reuse of Existing Resources for Analog BIST of a Switched Capacitor Filter. In: DESIGN, AUTOMATION AND TEST IN EUROPE CONFERENCE AND EXHIBITION (DATE), 2000, Paris, França. **Proceedings...** Los Alamitos: IEEE Computer Society, mar. 2000. p. 226-230.

COTA, E. et al. Test Planning and Design Space Exploration in a Core-based Environment. In: DESIGN, AUTOMATION AND TEST IN EUROPE CONFERENCE AND EXHIBITION (DATE), 2002, Paris, França. **Proceedings...** Los Alamitos: IEEE Computer Society, mar. 2002. p. 478-485.

COTA, E. **Reuse-based Test Planning for Core-based Systems-on-Chip**, 2003. 166f. Tese (Doutorado em Ciência da Computação), Programa de Pós-graduação em Ciência da Computação, Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2003.

CSIZMADIA, N.; JANSSEN, A. Estimating the Integral Non-Linearity of AD-Converters via the Frequency Domain. In: INTERNATIONAL TEST CONFERENCE (ITC), 30., 1999, Atlantic City, NJ, USA. **Proceedings...** Washington, D.C.: International Test Conference, set. 1999. p. 757-762.

DOERNBERG, J.; LEE, H.; HODGES, D. Full-Speed Testing of A/D Converters. **IEEE Journal of Solid-State Circuits**, USA, v. 19, n. 6, p. 820-827, dez. 1984.

DUFAZA, C.; HIS, H. Test Synthesis for DC Test and Maximal Diagnosis of Switched-Capacitor Circuits. In: IEEE VLSI TEST SYMPOSIUM (VTS), 15., 1997, Monterey, CA, USA. **Proceedings...** Los Alamitos: IEEE Computer Society, abr. 1997. p. 252-260.

DUFORT, B.; ROBERTS, G.W. Signal Generation Using Periodic Single and Multi-Bit Sigma-Delta Modulated Streams. In: INTERNATIONAL TEST CONFERENCE (ITC), 28., 1997, Washington, D.C., EEUU, **Proceedings...** Washington, D.C.: International Test Conference, nov. 1997. p. 396-405.

FLORES, M.G. et al. A Noise Generator for Analog-to-Digital Converter Testing. In: SIMPÓSIO BRASILEIRO DE CONCEPÇÃO DE CIRCUITOS INTEGRADOS (SBCCI), 15., 2002, Porto Alegre, Brasil. **Anais...** Porto Alegre: SBMICRO, SBC, set. 2002. p.135-140.

FLORES, M. G.; **Teste Embarcado de Conversores Analógico-Digitais**, 2003. 134f., Dissertação (Mestrado em Engenharia Elétrica), Programa de Pós-graduação em Engenharia Elétrica, Universidade Federal do Rio Grande do Sul, 2003.

FLORES, M. G., et al. Low Cost BIST for Static and Dynamic Testing of ADCs. **Journal of Electronic Testing: theory and applications (JETTA)**, Boston: Kluwer Academic Publisher, v. 21, n. 3, p. 283-290, jun. 2005.

HUERTAS, G. et al. Practical Oscillation-Based Test of Integrated Filters. **IEEE Design & Test of Computers**, Los Alamitos: IEEE Computer Society, v 19, n. 6, p. 64-72, nov. 2002.

HUERTAS, G. **Test Basado em Oscilación de Circuitos de Señal Mixta**, 2004, 460f. Tese (Doutorado em Ciências Físicas), Departamento de Electrónica y Eletromagnetismo, Universidad de Sevilla, Sevilla, Espanha, 2004.

IEEE STANDARDS BOARD. **IEEE Std 1149.4**: standard for a mixed-signal test bus, New York, 2000.

IEEE STANDARDS BOARD. **IEEE P1500**: standard for embedded core test (SECT), New York, 2005. Disponível em: <<http://grouper.ieee.org/groups/1500/>>. Acesso em: 9 mai. 2005.

IYENGAR, V.; CHAKRABARTY, K. Precedence-Based, Preemptive, and Power-Constrained Test Scheduling for System-on-a-chip. In: IEEE VLSI TEST SYMPOSIUM (VTS), 19., 2001, Marina del Rey, EEUU. **Proceedings...** Los Alamitos: IEEE Computer Society, 2001. p. 368-374.

IYENGAR, V.; CHAKRABARTY, K.; MARINISSEN, E. Test Wrapper and Test Access Mechanism Co-optimization for System-on-chip. **Journal of Electronic Testing: theory and applications (JETTA)**, Boston: Kluwer Academic Publisher, v.18, n.2, p.213-220, abr. 2002a.

IYENGAR, V.; CHAKRABARTY, K. System-on-chip Test Scheduling with Precedence Relationships, Preemption and Power Constraints. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, USA, v. 21, n. 9, p. 1088-1094, set. 2002b.

IYENGAR, V. et al. Test Resource Optimization for Multi-site Testing of SOCs, Under ATE Memory Depths Constraints. In: INTERNATIONAL TEST CONFERENCE (ITC), 33., 2002, Baltimore, MD, USA. **Proceedings...** Washington D.C.: International Test Conference, 2002c. p. 1159-1168.

JOHNS, D.; MARTIN, K. **Analog Integrated Circuit Design**. New York: John Wiley & Sons, 1997, 706 p. ISBN 0-471-14448-7.

KAC, U. et al. Extending IEEE Std. 1149.4 Analog Boundary Modules to Enhance Mixed-Signal Test. **IEEE Design & Test of Computers**, Los Alamitos: IEEE Computer Society Press, v. 20 , n. 2, mar. 2003, p. 32-39.

KUCUKCAKAR, K. Analysis of Emerging Core-based Design Lifecycle. In: IEEE/ACM INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN, 1998. **Digest of Technical Papers...** New York: IEEE, 1998. p. 445-449.

LARSSON, E.; PENGO, Z. An Integrated Framework for the Design and Optimization of SOC Test Solutions. **Journal of Electronic Testing: theory and applications (JETTA)**, Boston: Kluwer Academic Publisher, v. 18, n. 4-5, p. 385-400, ago. 2002.

LECHNER, A. et al. Review of Test Strategies and Resources used in High-Resolution Interface Testing, **Relatório de projeto TAMES: testability of analogue macrocells embedded in system-on-chips**, 2002, 65p. [S.l.] Disponível em: <http://www.imse.cnm.es/~tames2/Documents/Deliverables/TAMES2_D1.1_V1.2.pdf>. Acesso em: mar. 2005.

LIU, R.W. **Testing and Diagnosis of Analog Circuits and Systems**, New York: Van Nostrand Reinhold, 1991, 284 p. ISBN: 0-44-2259328.

MARINISSEN, E. J.; IYENGAR, V.; CHAKRABARTY, K. **ITC'02 SOC Test Benchmarks**. [S.l.: s.n.], 2003. Disponível em: <<http://www.extra.research.philips.com/itc02socbenchm/>>. Acesso em: 9 mai. 2005.

MARINISSEN, E. J. et al. Towards a Standard for Embedded Core Test: an example. In: INTERNATIONAL TEST CONFERENCE (ITC), 30., 1999, Atlantic City, NJ, USA. **Proceedings...** Washington, D.C.: International Test Conference, 1999. p. 616-627.

MARINISSEN, E. J. et al. A Structured and Scalable Mechanism for Test Access to Embedded Reusable Cores. In: INTERNATIONAL TEST CONFERENCE (ITC), 29., 1998, Washington, D.C., USA. **Proceedings...** Washington, D.C.: International Test Conference, 1998. p. 284-293.

MARINISSEN, E.J.; GOEL S.K.; LOUSBERG, M. Wrapper Design for Embedded Core Test. In: INTERNATIONAL TEST CONFERENCE (ITC), 31., 2000, Atlantic City, NJ, USA. **Proceedings...** Washington, D.C.: International Test Conference, 2000. p. 911–920.

MILOR, L. S. A Tutorial Introduction to Research on Analog and Mixed-Signal Circuits Testing. **IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing**, USA, v. 45, n. 10, p.1389-1407, out. 1998.

MURESAN, V.; WANG, X.; VLADUTIU, M. A Comparison of Classical Scheduling Approaches in Power-Constrained Block-Test Scheduling. In: INTERNATIONAL TEST CONFERENCE (ITC), 31., 2000, Atlantic City, NJ, USA. **Proceedings...** Washington, D.C.: International Test Conference, 2000. p. 882–891.

NAGI, N.; SUNTER, S. A Simplified Polynomial-fitting Algorithm for ADC and DAC Test. In: INTERNATIONAL TEST CONFERENCE (ITC), 28., 1997, Washington, D.C., USA. **Proceedings...** Washington, D.C.: International Test Conference, 1997. p. 389-395.

NEGREIROS, M. **Statistical Samplers (Dithered Quantizers) Applied To The Test Of Analog And Mixed Signal Circuits**. 2002, 51f. Proposta de Tese (Doutorado em Ciência da Computação), Programa de Pós-Graduação em Ciência da Computação, Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2002.

OGATA, K. Análise de Resposta Transitória. In: **Engenharia de Controle Moderno**. Rio de Janeiro: Prentice-Hall do Brasil, 1982, p. 267-279. ISBN: 8570540191.

PAPOULIS, A. **Probability, Random Variables and Stochastic Processes**. 3. ed., New York: McGraw-Hill, 1991, p. 307, Eq. 10-71. ISBN: 0071008705.

PEETZ, B. E. Dynamic Testing of Waveform Recorders. **IEEE Transactions on Instrumentation and Measurement**, EEUU, v. IM-32, n. 1, p. 12-16, 1983.

PEREIRA, G. et al. Testing the Interconnect Networks and I/O Resources of Field Programmable Analog Arrays, In: IEEE VLSI TEST SYMPOSIUM (VTS), 21., 2005, Palm Springs, CA, USA. **Proceedings...** Los Alamitos: IEEE Computer Society, mai. 2005. p. 389-394.

RENOVELL M. et al., Hardware Minimization for an Histogram-Based ADC-BIST. In: IEEE VLSI TEST SYMPOSIUM (VTS), 18., 2000, Montreal, Canada. **Proceedings...** Los Alamitos: IEEE Computer Society, 2000. p. 247-252.

SEHGAL, A.; IYENGAR, V.; CHAKRABARTY, K. SOC Test Planning using Virtual Test Access Architectures. **IEEE Transactions on Very Large Scale Integration (VLSI) Systems**, USA, v. 12, n. 12, p. 1263-1276, dez. 2004.

SEHGAL, A.; OZEV, S.; CHAKRABARTY K. TAM Optimization for Mixed-Signal SOCs Using Test Wrappers for Analog Cores. In: IEEE/ACM INTERNATIONAL CONFERENCE ON COMPUTER AIDED DESIGN (ICCAD), 2003, San Jose, CA, EEUU. **Proceedings...** New York: ACM, 2003. p. 95-99.

SONG, L. Y. Mixed-Signal BIST: fact or fiction. In: INTERNATIONAL TEST CONFERENCE (ITC), 33., 2002, Baltimore, MD, USA. **Proceedings...** Washington D.C.: International Test Conference, 2002. p. 1203.

SOUZA JR., A.; CARRO, L. A DFT Macrocell for Digital Test of Analog Blocks. In: IEEE LATIN AMERICAN TEST WORKSHOP, 4., 2003, Natal, Brasil. **Digest of papers...** New York: IEEE, 2003. p. 217-222.

STAPPER, C. H.; ARMSTRONG, F. M.; SAJI, K. Integrated Circuit Yield Statistics. In: **Proceedings of the IEEE**, USA, v. 71, p. 453-470, abr. 1983.

STEININGER, A. Testing and Built-in Self Test: a survey. **Journal of Systems Architecture**, Amsterdam: Elsevier, v. 46, n. 9, p. 721-747, jul. 2000.

SUNTER, S. K. Cost/Benefit Analysis of the P1149.4 Mixed-Signal Test Bus. In: CIRCUIT DEVICES SYSTEMS, UK: IEE, v. 143, n. 6, p. 393-398, dez. 1996.

VANA, T.; BLAIR, D. **Convergent SOCs challenge Mixed-Signal Test.** [S.l.], jan. 2003, hipertexto. Disponível em: <www.evaluationengineering.com/archive/articles/0103ate.htm>. Acesso em: 9 mai. 2005.

VARMA, P.; BHATIA, S. A Structured Test Reuse Methodology for Core-Based System Chips. In: INTERNATIONAL TEST CONFERENCE (ITC), 29., 1998, Washington, D.C., USA. **Proceedings...** Washington, D.C.: International Test Conference, 1998. p. 294-302.

VAZQUEZ, D. et al. Sine-Wave Signals Characterization Using First Order $\Sigma\Delta$ Modulators: Application to Mixed-Signal BIST. In: IEEE INTERNATIONAL MIXED-SIGNAL TESTING WORKSHOP (IMSTW), 9., 2003, Sevilha, Espanha. **Digest of papers...** [S.l.: s.n.], 2003. p. 171-176.

ZORIAN, Y. A Distributed BIST Control Scheme for Complex VLSI Devices. In: IEEE VLSI TEST SYMPOSIUM (VTS), 11., 1993, Atlantic City, NJ, USA. **Proceedings...** Los Alamitos: IEEE Computer Society, 1993. p. 4-9.

ZORIAN, Y.; DEY, S.; RODGERS, M. Test of Future System-on-Chips. In: IEEE/ACM INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN (ICCAD), 2000, San Jose, CA, USA. **Proceedings...** USA: IEEE, 2000. p.392-398.

ZORIAN, Y.; MARINISSEN E. J.; DEY, S. Testing Embedded Core-Based System Chips. **IEEE Computer**, Los Alamitos: IEEE Computer Society, v. 32, n. 6, p. 52-60, jun. 1999.

APÊNDICE A : arquivo MATLAB com descrição do sistema u226m

```
#####%
System description file
%
% Adaptado da descrição feita por Erika Cota, 25/11/02
% erika@inf.ufrgs.br
%
% Antonio de Quadros Andrade Júnior, 03/03/2005
% andradejr@eletro.ufrgs.br
#####

function [technology, cores_no, in_connections, sys_distances, features,
critical_path, total_pinos, limite_pinos, limite_tempo, vizinhanca_sc_in,
vizinhanca_sc_out, vizinhanca_ext_in, vizinhanca_ext_out,
criterio,limite_potencia, precedentes_in, precedentes_out, system_name] =
u226()

system_name = 'u226_SB_3ADF_30';

% indica se arquivo de entrada para posicionamento dos núcleos deve
% ser gerado
gera_placement = 0;

% o núcleo 7 eh o mais próximo de um processador, e eh escolhido como
% núcleo precedente
prec = 7;

% #####
% Definições da tecnologia
%
% alguns parametros de tecnologia estao definidos em features_list,
% para facilitar implementacao (capacitancia, Vdd, frequencia
% eventualmente.

technology = 0.8e-6;
largura_fio = 1.0e-6 + 1.2e-6; % espacamento + largura do fio

trans_n = 1.2e-6*(1.2e-6 + 1.2e-6 + 0.8e-6); % resultado em um2
trans_p = 2*trans_n;

fator_conexao = 1.5;

ff_area = fator_conexao*16 *(trans_n + trans_p) ;
% 16 transistores * 0.64 = um2
gate_area = fator_conexao*5*(trans_n + trans_p);
% 1 gate tem de 4 a 6 transistores. Cada transistor mede 2*technology.
% (estimative grosseira)
inv_area = fator_conexao*(trans_n + trans_p);
% 2 transistores * 0.64 = um2

% #####
% USER-DEFINED!!!
% Características e restrições do sistema, fatores de otimização...
```

```

% numero de nucleos no sistema (apenas no nivel mais alto da
% hierarquia)
no_busses = 0;
modules_no = 9;
% number of level 1 modules. Those will be considered for connections,
% sys_distances, etc. The cores described after this number are the
% virtual ones, representing BIST or hierarchical tests

cores_no = 10+ 1 + no_busses;
level_0 = 11;

% numero maximo de nucleos funcionalmente ligados entre a entrada e a
% saída do sistema
critical_path = 4;

% number of extra pins allowed in the system interface for testing.
% do not consider wrapper control pins.

porcentagem_pinos = 10;
% acrescimo de 10% na qtde de pinos da interface

% Tempo limite para o escalonamento; caso não exista limite pré-
% definido, deixar em zero.
limite_tempo = 0;

% POWER limit during testing.
% o teste no schedule sera, no entanto, feito em relacao a
% ENERGIA gasta em cada tempo. Este limite de potencia sera
% multiplicado pelo numero de ciclos total do schedule = maxtime.
% Cada nucleo tambem tem uma potencia durante o teste definida de
% forma normalizada (independe da tecnologia e da frequencia de
% teste). Ou seja, a potencia definida aqui para cada nucleo e' por
% ciclo e sera multiplicada
% pelo numero de ciclos de teste durante a definicao da solucao, de
% forma a se obter a energia usada por cada nucleo durante o teste.
% As formulas utilizadas para calculo da potencia sao:
% P_cc = C_lt * Vdd^2 * f * a (potencia do circuito combinacional)
% P_ff = C_ltf * Vdd^2 * f * a_d + C_ltf * Vdd2 * f * 1
% C_lt = f(# gates do nucleo)
% f = frequencia de chaveamento
% a = fator do numero de gates que chaveiam durante o teste
% C_ltf = f(#flip-flops do circuito)
% a_d = fator do numero de ffs que chaveiam
% 1 = chaveamento do relógio (clock aconado em todos os ciclos)
% Entao, para um circuito combinacional, potencia = P_cc. PARA o
% circuito sequencial, potencia = P_cc + P_ff.
% Normalizando, temos que:
% C_l * Vdd^2 => funcao da tecnologia
% e f tambem nao e' sabida. Assim, por exemplo, P_cc fica:
% P_cc = C_l * #gates * Vdd^2 * f * a_d
% = C_l * Vdd^2 * #gates * 1/T * a_d (potencia por ciclo)
% C_l * Vdd^2 e' uma constante para a tecnologia
% Energia = P_cc * numero de ciclos de teste, calculado no algoritmo.
% F * V^2 * Hz = Watts (para comparacao no algoritmo)

limite_potencia = inf; %1.5e-8*0.4; % watts => limite de energia sera
limite_potencia*maxtime.
% soma das potencias dos cores = 1.5e-8

```

```

% optimization criteria: pins, area, test time. The value 1 indicates
% that this cost must be optimized. See table below for algorithm
% behavior
criterio = cell(1);
criterio.pinos = 1;
criterio.area = 1;
criterio.tempo = 1;
criterio.potencia = 0;

% pins | area | tempo | limite_ | limite_ | Resultado do sistema
% pins | area | tempo | pins | tempo |
%-----|-----|-----|-----|-----|-----
% 1 | 0 | 0 | -1 | X | acha solucao com no maximo
% | | | | | | 4 pinos (2 para scan e 2
% | | | | | | funcional)
%-----|-----|-----|-----|-----|-----
% 1 | 0 | 0 | 0 | X | acha solucao com menor no.
% | | | | | | de pinos (maximo 20
% | | | | | | iterações para cada parte)
%-----|-----|-----|-----|-----|-----
% 1 | 0 | 0 | v,v > 0 | X | acha solucao com no maximo
% | | | | | | v pinos (ou mais próxima
% | | | | | | dentro das 20 iterações)
%-----|-----|-----|-----|-----|-----
% 1 | 1 | 0 | X | X | acha solucao com no de
% | | | | | | pinos de acordo com descr.
% | | | | | | anteriores e menor area)
%-----|-----|-----|-----|-----|-----
% 1 | 0 | 1 | X | X | acha solucao com no de
% | | | | | | pinos de acordo com descr.
% | | | | | | anteriores e menor tempo)
%-----|-----|-----|-----|-----|-----
% 0 | 1 | X | X | X | acha solucao com menor
% | | | | | | area no tempo dado por
% | | | | | | limite_tempo ou sem limite
% | | | | | | de tempo)
%-----|-----|-----|-----|-----|-----
% 0 | 0 | 1 | 0 | 0 | acha solucao com menor
% | | | | | | tempo possivel. Tempo_
% | | | | | | Maximo e' o tempo do core
% | | | | | | com maior tempo de teste)
%-----|-----|-----|-----|-----|-----
% 1 | 1 | 1 | 0 | 0 | Idem anterior. Se pinos e
% | | | | | | área estiverem setados, a
% | | | | | | solução com menores custos
% | | | | | | neste fatores que cabem no
% | | | | | | tempo e' encontrada. Tempo
% | | | | | | Maximo e'dado por limite_
% | | | | | | tempo ou core critico.
%-----|-----|-----|-----|-----|-----
% X | X | X | X | v, v>0 | Idem anteriores (de acordo
% | | | | | | com pinos e área) mas com
% | | | | | | tempo maximo dado por v.
%-----|-----|-----|-----|-----|-----

% #####
% Estrutura de Dados: alocação e definições básicas

```

```

features_list;
[in_connections, sys_distances, features] = soc_customize(cores_no);

% #####
% USER DEFINED!!!
% Características de cada núcleo no sistema
% apenas características válidas (diff. zero). Se não for mencionado,
% o valor é inicializado com zero. Para uma lista completa das
% características modeladas, referir-se a "features_list.m"
% Para cada núcleo, criar uma nova linha na matriz de características

% Núcleos 4 a 9: digitais

i = 4;
features(i,clength) = 1.0; % mil micra m
features(i,cheigth) = 1.0; % mil micra m
features(i,inverter_no) = 0;
features(i,gates_no) = 3640;
features(i,area_core) = 2*(features(1,gates_no)*gate_area +
features(1,ff_no)*ff_area + features(1,inverter_no)*inv_area);
features(i,inputs_no) = 3;
features(i,outputs_no) = 17;
features(i,bidirs_no) = 0;
features(i,ff_no) = 256;
features(i,extest_inputs) = 3;
features(i,extest_outputs) = 17;
features(i,extest_cycles) = 1;
features(i,extest_vector_no) = 2666;
features(i,sctest_inputs) = 0;
features(i,sctest_outputs) = 0;
features(i,sctest_maxchain) = 0;
features(i,sctest_vector_no) = 0;
features(i,bij_no) = 0;
features(i,bij_cost) = inf;
features(i,bij_cycles) = inf;
features(i,test_time) = features(i,extest_vector_no);
%(features(1,sctest_vector_no)+1)*features(1,sctest_maxchain) +
features(1,sctest_vector_no);
features(i,switching_gates) = 0.12;
features(i,switching_ff) = 0.12;
power_mux_wr =
capacitancia_l*Vdd^2*(2*features(i,extest_inputs)+2*features(i,extest_outpu
ts)+2*features(i,sctest_inputs)+2*features(i,sctest_outputs))*switching_mux
;
power_ff_wr =
(capacitancia_l*Vdd^2*(features(i,extest_inputs)+features(i,extest_outputs)
+features(i,sctest_inputs)+features(i,sctest_outputs)))*(switching_ff +1);
power_wr = capacitancia_l*Vdd^2*5*switching_ff + capacitancia_l*Vdd^2*5*1 +
capacitancia_l*Vdd^2*11*switching_mux;
features(i,cpower) = power_mux_wr + power_ff_wr + power_wr +
capacitancia_l*Vdd^2*features(i,gates_no)*features(i,switching_gates) +
capacitancia_l*Vdd^2*features(i,ff_no)*features(i,switching_ff) +
capacitancia_l*Vdd^2*features(i,ff_no)*1;
features(i,cenergia) = features(i,cpower)*features(i,test_time);

```

```

i = 5;
features(i,clength) = 1.0; %e3*1e-6;    % mil micra m
features(i,cheigth) = 1.0; %e3*16-6;    % mil micra m
features(i,inverter_no) = 0;
features(i,gates_no) = 3640;
features(i,area_core) = 2*(features(1,gates_no)*gate_area +
features(1,ff_no)*ff_area + features(1,inverter_no)*inv_area);
features(i,inputs_no) = 3;
features(i,outputs_no) = 17;
features(i,bidirs_no) = 0;
features(i,ff_no) = 256;
features(i,extest_inputs) = 3;
features(i,extest_outputs) = 17;
features(i,extest_cycles) = 1;
features(i,extest_vector_no) = 2666;
features(i,sctest_inputs) = 0;
features(i,sctest_outputs) = 0;
features(i,sctest_maxchain) = 0;
features(i,sctest_vector_no) = 0;
features(i,bij_no) = 0;
features(i,bij_cost) = inf;
features(i,bij_cycles) = inf;
features(i,test_time) = features(i,extest_vector_no);
%(features(1,sctest_vector_no)+1)*features(1,sctest_maxchain) +
features(1,sctest_vector_no);
features(i,switching_gates) = 0.12;
features(i,switching_ff) = 0.12;
power_mux_wr =
capacitancia_l*Vdd^2*(2*features(i,extest_inputs)+2*features(i,extest_outpu
ts)+2*features(i,sctest_inputs)+2*features(i,sctest_outputs))*switching_mux
;
power_ff_wr =
(capacitancia_l*Vdd^2*(features(i,extest_inputs)+features(i,extest_outputs)
+features(i,sctest_inputs)+features(i,sctest_outputs)))*(switching_ff +1);
power_wr = capacitancia_l*Vdd^2*5*switching_ff + capacitancia_l*Vdd^2*5*1 +
capacitancia_l*Vdd^2*11*switching_mux;
features(i,cpower) = power_mux_wr + power_ff_wr + power_wr +
capacitancia_l*Vdd^2*features(i,gates_no)*features(i,switching_gates) +
capacitancia_l*Vdd^2*features(i,ff_no)*features(i,switching_ff) +
capacitancia_l*Vdd^2*features(i,ff_no)*1;
features(i,cenergia) = features(i,cpower)*features(i,test_time);

i = 6;
features(i,clength) = 1.0; % mil micra m
features(i,cheigth) = 1.0; % mil micra m
features(i,inverter_no) = 0;
features(i,gates_no) = 3640;
features(i,area_core) = 2*(features(1,gates_no)*gate_area +
features(1,ff_no)*ff_area + features(1,inverter_no)*inv_area);
features(i,inputs_no) = 3;
features(i,outputs_no) = 17;
features(i,bidirs_no) = 0;
features(i,ff_no) = 256;
features(i,extest_inputs) = 3;
features(i,extest_outputs) = 17;
features(i,extest_cycles) = 1;
features(i,extest_vector_no) = 2666;

```



```

features(i,sctest_inputs) = 0;
features(i,sctest_outputs) = 0;
features(i,sctest_maxchain) = 0;
features(i,sctest_vector_no) = 0;
features(i,bij_no) = 0;
features(i,bij_cost) = inf;
features(i,bij_cycles) = inf;
features(i,test_time) = features(i,extest_vector_no);
%(features(1,sctest_vector_no)+1)*features(1,sctest_maxchain) +
features(1,sctest_vector_no);
features(i,switching_gates) = 0.12;
features(i,switching_ff) = 0.12;
power_mux_wr =
capacitancia_l*Vdd^2*(2*features(i,extest_inputs)+2*features(i,extest_outpu
ts)+2*features(i,sctest_inputs)+2*features(i,sctest_outputs))*switching_mux
;
power_ff_wr =
(capacitancia_l*Vdd^2*(features(i,extest_inputs)+features(i,extest_outputs)
+features(i,sctest_inputs)+features(i,sctest_outputs)))*(switching_ff +1);
power_wr = capacitancia_l*Vdd^2*5*switching_ff + capacitancia_l*Vdd^2*5*1 +
capacitancia_l*Vdd^2*11*switching_mux;
features(i,cpower) = power_mux_wr + power_ff_wr + power_wr +
capacitancia_l*Vdd^2*features(i,gates_no)*features(i,switching_gates) +
capacitancia_l*Vdd^2*features(i,ff_no)*features(i,switching_ff) +
capacitancia_l*Vdd^2*features(i,ff_no)*1;
features(i,cenergia) = features(i,cpower)*features(i,test_time);

i=7;
features(i,length) = 2.4; % mil micra m
features(i,cheigth) = 2; % mil micra m
features(i,inputs_no) = 41;
features(i,outputs_no) = 82;
features(i,bidirs_no) = 0;
features(i,ff_no) = 1024;
features(i,extest_inputs) = 0;
features(i,extest_outputs) = 0;
features(i,extest_cycles) = 1;
features(i,extest_vector_no) = 0;
features(i,sctest_inputs) = 20+2;
features(i,sctest_outputs) = 20+2;
features(i,sctest_maxchain) = 52;
%ceil(features(i,ff_no)/features(i,sctest_inputs));
features(i,sctest_vector_no) = 76;
features(i,bij_no) = 0;
features(i,bij_cost) = inf;
features(i,bij_cycles) = inf;
features(i,test_time) =
(features(7,sctest_vector_no)+1)*features(7,sctest_maxchain) +
features(7,sctest_vector_no);
features(i,inverter_no) = 0;
features(i,gates_no) = 22840;
features(i,switching_gates) = 0.12;
features(i,switching_ff) = 0.12;
features(i,area_core) = 2*(features(7,gates_no)*gate_area +
features(7,ff_no)*ff_area + features(7,inverter_no)*inv_area);
power_mux_wr =
capacitancia_l*Vdd^2*(2*features(i,extest_inputs)+2*features(i,extest_outpu

```

```

ts)+2*features(i,sctest_inputs)+2*features(i,sctest_outputs))*switching_mux
;
power_ff_wr =
(capacitancia_l*Vdd^2*(features(i,extest_inputs)+features(i,extest_outputs)
+features(i,sctest_inputs)+features(i,sctest_outputs)))*(switching_ff +1);
power_wr = capacitancia_l*Vdd^2*5*switching_ff + capacitancia_l*Vdd^2*5*1 +
capacitancia_l*Vdd^2*11*switching_mux;
features(i,cpower) = power_mux_wr + power_ff_wr + power_wr +
capacitancia_l*Vdd^2*features(i,gates_no)*features(i,switching_gates) +
capacitancia_l*Vdd^2*features(i,ff_no)*features(i,switching_ff) +
capacitancia_l*Vdd^2*features(i,ff_no)*1;
features(i,cenergia) = features(i,cpower)*features(i,test_time);
features(i,op_power) = 0.8*features(i,cpower);

i = 8;
features(i,clength) = 2.4; % mil micra m
features(i,cheigth) = 5.8; % mil micra m
features(i,inverter_no) = 0;
features(i,gates_no) = 14000;
features(i,area_core) = 2*(features(1,gates_no)*gate_area +
features(1,ff_no)*ff_area + features(1,inverter_no)*inv_area);
features(i,inputs_no) = 34;
features(i,outputs_no) = 32;
features(i,bidirs_no) = 0;
features(i,ff_no) = 0;
features(i,extest_inputs) = 0;
features(i,extest_outputs) = 0;
features(i,extest_cycles) = 1;
features(i,extest_vector_no) = 0;
features(i,sctest_inputs) = 0;
features(i,sctest_outputs) = 0;
features(i,sctest_maxchain) = 0;
features(i,sctest_vector_no) = 0;
features(i,bij_no) = 0;
features(i,bij_cost) = inf;
features(i,bij_cycles) = inf;
features(i,test_time) = 0; %features(i,extest_vector_no);
%(features(1,sctest_vector_no)+1)*features(1,sctest_maxchain) +
features(1,sctest_vector_no);
features(i,switching_gates) = 0.12;
features(i,switching_ff) = 0.12;
power_mux_wr = 0;
%capacitancia_l*Vdd^2*(2*features(i,extest_inputs)+2*features(i,extest_outp
uts)+2*features(i,sctest_inputs)+2*features(i,sctest_outputs))*switching_mu
x;
power_ff_wr = 0;
%(capacitancia_l*Vdd^2*(features(i,extest_inputs)+features(i,extest_outputs)
)+features(i,sctest_inputs)+features(i,sctest_outputs)))*(switching_ff +1);
power_wr = 0; %capacitancia_l*Vdd^2*5*switching_ff +
capacitancia_l*Vdd^2*5*1 + capacitancia_l*Vdd^2*11*switching_mux;
features(i,cpower) = 0; %power_mux_wr + power_ff_wr + power_wr +
capacitancia_l*Vdd^2*features(i,gates_no)*features(i,switching_gates) +
capacitancia_l*Vdd^2*features(i,ff_no)*features(i,switching_ff) +
capacitancia_l*Vdd^2*features(i,ff_no)*1;
features(i,cenergia) = 0; %features(i,cpower)*features(i,test_time);

i = 9;

```

```

features(i,clength) = 2; % mil micra m
features(i,cheigth) = 3; % mil micra m
features(i,inverter_no) = 0;
features(i,gates_no) = 2000;
features(i,area_core) = features(1,clength)*features(1,cheigth);
%2*(features(1,gates_no)*gate_area + features(1,ff_no)*ff_area +
features(1,inverter_no)*inv_area);
features(i,inputs_no) = 17;
features(i,outputs_no) = 10;
features(i,bidirs_no) = 0;
features(i,ff_no) = 0;
features(i,extest_inputs) = 17;
features(i,extest_outputs) = 10;
features(i,extest_cycles) = 1;
features(i,extest_vector_no) = 15;
features(i,sctest_inputs) = 0;
features(i,sctest_outputs) = 0;
features(i,sctest_maxchain) = 0;
features(i,bij_no) = 0;
features(i,bij_cost) = inf;
features(i,bij_cycles) = inf;
features(i,sctest_vector_no) = 0;
features(i,test_time) = features(i,extest_vector_no);
%(features(1,sctest_vector_no)+1)*features(1,sctest_maxchain) +
features(1,sctest_vector_no);
features(i,switching_gates) = 0.12;
features(i,switching_ff) = 0.12;
power_mux_wr =
capacitancia_l*Vdd^2*(2*features(i,extest_inputs)+2*features(i,extest_outpu
ts)+2*features(i,sctest_inputs)+2*features(i,sctest_outputs))*switching_mux
;
power_ff_wr =
(capacitancia_l*Vdd^2*(features(i,extest_inputs)+features(i,extest_outputs)
+features(i,sctest_inputs)+features(i,sctest_outputs)))*(switching_ff +1);
power_wr = capacitancia_l*Vdd^2*5*switching_ff + capacitancia_l*Vdd^2*5*1 +
capacitancia_l*Vdd^2*11*switching_mux;
features(i,cpower) = power_mux_wr + power_ff_wr + power_wr +
capacitancia_l*Vdd^2*features(i,gates_no)*features(i,switching_gates) +
capacitancia_l*Vdd^2*features(i,ff_no)*features(i,switching_ff) +
capacitancia_l*Vdd^2*features(i,ff_no)*1;
features(i,cenergia) = features(i,cpower)*features(i,test_time);

% Núcleos 1 a 3: analógicos

% Potência dos blocos analogicos equivale a 30% da potencia total de teste
potdig = sum(features(4:9,cpower));

i = 1;
features(i,clength) = 1.4; % mil micra m
features(i,cheigth) = 1.6; % mil micra m
features(i,inverter_no) = 0;
features(i,gates_no) = 10000;
features(i,area_core) = 2*(features(1,gates_no)*gate_area +
features(1,ff_no)*ff_area + features(1,inverter_no)*inv_area);
features(i,inputs_no) = 2;
features(i,outputs_no) = 1;
features(i,bidirs_no) = 0;

```

```

features(i,ff_no) = 0;
features(i,extest_inputs) = 1;
features(i,extest_outputs) = 1;
features(i,extest_cycles) = 1;
features(i,extest_vector_no) = 8192*50;
features(i,sctest_inputs) = 0;
features(i,sctest_outputs) = 0;
features(i,sctest_maxchain) = 0;
features(i,sctest_vector_no) = 0;
features(i,bij_no) = 0;
features(i,bij_cost) = inf;
features(i,bij_cycles) = inf;
features(i,test_time) = 0;
features(i,switching_gates) = 0.12;
features(i,switching_ff) = 0.12;

% Como é um circuito analógico, nao há consumo de potencia devido a
% estruturas como muxers e flip-flops
power_mux_wr = 0;
power_ff_wr = 0;
power_wr = 0;
features(i,cpower) = potdig/3; %power_mux_wr + power_ff_wr + power_wr +
capacitancia_l*Vdd^2*features(i,gates_no)*features(i,switching_gates) +
capacitancia_l*Vdd^2*features(i,ff_no)*features(i,switching_ff) +
capacitancia_l*Vdd^2*features(i,ff_no)*1;
features(i,cenergia) = features(i,cpower)*features(i,test_time);

i = 2;
features(i,clength) = 1.4; % mil micra m
features(i,cheigth) = 1.6; % mil micra m
features(i,inverter_no) = 0;
features(i,gates_no) = 10000;
features(i,area_core) = 2*(features(1,gates_no)*gate_area +
features(1,ff_no)*ff_area + features(1,inverter_no)*inv_area);
features(i,inputs_no) = 2;
features(i,outputs_no) = 1;
features(i,bidirs_no) = 0;
features(i,ff_no) = 0;
features(i,extest_inputs) = 1;
features(i,extest_outputs) = 1;
features(i,extest_cycles) = 1;
features(i,extest_vector_no) = 8192*50;
features(i,sctest_inputs) = 0;
features(i,sctest_outputs) = 0;
features(i,sctest_maxchain) = 0;
features(i,sctest_vector_no) = 0;
features(i,bij_no) = 0;
features(i,bij_cost) = inf;
features(i,bij_cycles) = inf;
features(i,test_time) = 0;
features(i,switching_gates) = 0.12;
features(i,switching_ff) = 0.12;
features(i,cpower) = potdig/3;
features(i,cenergia) = features(i,cpower)*features(i,test_time);

i = 3;
features(i,clength) = 1.4; % mil micra m

```

```

features(i,cheigth) = 1.6; % mil micra m
features(i,inverter_no) = 0;
features(i,gates_no) = 10000;
features(i,area_core) = 2*(features(1,gates_no)*gate_area +
features(1,ff_no)*ff_area + features(1,inverter_no)*inv_area);
features(i,inputs_no) = 2;
features(i,outputs_no) = 1;
features(i,bidirs_no) = 0;
features(i,ff_no) = 0;
features(i,extest_inputs) = 1;
features(i,extest_outputs) = 1;
features(i,extest_cycles) = 1;
features(i,extest_vector_no) = 8192*50;
features(i,sctest_inputs) = 0;
features(i,sctest_outputs) = 0;
features(i,sctest_maxchain) = 0;
features(i,sctest_vector_no) = 0;
features(i,bij_no) = 0;
features(i,bij_cost) = inf;
features(i,bij_cycles) = inf;
features(i,test_time) = 0;
features(i,switching_gates) = 0.12;
features(i,switching_ff) = 0.12;
features(i,cpower) = potdig/3;
features(i,cenergia) = features(i,cpower)*features(i,test_time);

% #####
% If there are hierarchical cores with test sets, they
% are defined here accordingly. Usually, all internal cores
% are defined together by adding up their test vectors, if
% all have only external test. These cores have no area, but
% have power.
%
% See that in_connections and sys_distances for these hierarchical
% tests are properly defined below.

% #####
% Havendo conjuntos de teste BIST, estes são definidos neste espaço, como
% núcleos virtuais sem area mas com potência

i = 10; % BIST memoria
features(i,clength) = 2.4; % mil micra m
features(i,cheigth) = 5.8; % mil micra m
features(i,inverter_no) = 0;
features(i,gates_no) = 14000;
features(i,area_core) = 0; %2*(features(1,gates_no)*gate_area +
features(1,ff_no)*ff_area + features(1,inverter_no)*inv_area);
features(i,inputs_no) = 34;
features(i,outputs_no) = 32;
features(i,bidirs_no) = 0;
features(i,ff_no) = 0;
features(i,bist_inputs) = 1;
features(i,bist_outputs) = 1;
features(i,extest_cycles) = 1;
features(i,bist_cycles) = 1048576;
features(i,sctest_inputs) = 0;
features(i,sctest_outputs) = 0;

```

```

features(i,sctest_maxchain) = 0;
features(i,sctest_vector_no) = 0;
features(i,bij_no) = 0;
features(i,bij_cost) = inf; %0.05;
features(i,bij_cycles) = inf; %6;
features(i,test_time) = features(i,bist_cycles);
%(features(1,sctest_vector_no)+1)*features(1,sctest_maxchain) +
features(1,sctest_vector_no);
features(i,switching_gates) = 0.12;
features(i,switching_ff) = 0.12;
power_mux_wr =
capacitancia_l*Vdd^2*(2*features(i,inputs_no)+2*features(i,outputs_no)+2*fe
atures(i,sctest_inputs)+2*features(i,sctest_outputs))*switching_mux;
power_ff_wr =
(capacitancia_l*Vdd^2*(features(i,inputs_no)+features(i,outputs_no)+feature
s(i,sctest_inputs)+features(i,sctest_outputs)))*(switching_ff +1);
power_wr = capacitancia_l*Vdd^2*5*switching_ff + capacitancia_l*Vdd^2*5*1 +
capacitancia_l*Vdd^2*11*switching_mux;
features(i,cpower) = power_mux_wr + power_ff_wr + power_wr +
capacitancia_l*Vdd^2*features(i,gates_no)*features(i,switching_gates) +
capacitancia_l*Vdd^2*features(i,ff_no)*features(i,switching_ff) +
capacitancia_l*Vdd^2*features(i,ff_no)*1;
features(i,cenergia) = features(i,cpower)*features(i,test_time);

% Núcleo nivel 0: interface
i = 11;
features(i,clength) = 0;
features(i,cheigth) = 0;
features(i,inputs_no) = 12;
features(i,outputs_no) = 41;
features(i,bidirs_no) = 0;
features(i,ff_no) = 0;
features(i,extest_inputs) = 0;
features(i,extest_outputs) = 0;
features(i,extest_cycles) = 0;
features(i,extest_vector_no) = 0;
features(i,sctest_inputs) = 0;
features(i,sctest_outputs) = 0;
features(i,sctest_maxchain) = 0;
%ceil(features(10,ff_no)/features(10,sctest_inputs));
features(i,sctest_vector_no) = 0;
features(i,bij_no) = 0;
features(i,bij_cost) = inf;
features(i,bij_cycles) = inf;
features(i,test_time) =
(features(11,sctest_vector_no)+1)*features(11,sctest_maxchain) +
features(11,sctest_vector_no);
features(i,inverter_no) = 0;
features(i,gates_no) = 0;
features(i,area_core) = 2*(features(11,gates_no)*gate_area +
features(11,ff_no)*ff_area + features(11,inverter_no)*inv_area);
features(i,switching_gates) = 0.12;
features(i,switching_ff) = 0.12;
power_mux_wr = 0;
power_ff_wr = 0;
features(i,cpower) = 0;
features(i,cenergia) = features(i,cpower)*features(i,test_time);

```

```

% #####
% Havendo BARRAMENTOS funcionais, os mesmos são definidos aqui como
% núcleos virtuais com área e potência

% #####
% Define o limite real de potência durante o teste como uma porcentagem da
% potência somada de todos os núcleos do sistema
limite_potencia = limite_potencia*sum(features(:,cpower));

% #####
% USER-DEFINED!!! Modificar aqui!
% Conexões funcionais do sistema
% estruturas são inicializadas com 0 (zero). Apenas as conexões existentes
são definidas aqui, seguindo a convenção:
% in_connections(a,b) = número de pinos do núcleo 'b' para núcleo 'a'

in_connections(1,5) = 2;
in_connections(2,5) = 2;
in_connections(3,8) = 2;
in_connections(4,7) = 2;
in_connections(5,8) = 3;
in_connections(7,4) = 8;
in_connections(7,8) = 25;
in_connections(8,1) = 1;
in_connections(8,4) = 16;
in_connections(8,6) = 17;
in_connections(9,7) = 17;
in_connections(cores_no+1,1) = 1;
in_connections(cores_no+1,2) = 1;
in_connections(cores_no+1,3) = 1;
in_connections(4,cores_no+1) = 1;
in_connections(cores_no+1,4) = 9;
in_connections(6,cores_no+1) = 3;
in_connections(7,cores_no+1) = 8;
in_connections(cores_no+1,7) = 19;
in_connections(cores_no+1,9) = 10;

% #####
% Verificando conexões. Nenhuma entrada pode ficar 'flutuando', nem
% qualquer núcleo pode receber mais sinais que o número de entradas

saidas = zeros(1,cores_no+1);
entradas = zeros(1,cores_no+1);

for i=1:modules_no,
    for j=1:modules_no,
        saidas(j) = saidas(j) + in_connections(i,j);
        entradas(i) = entradas(i) + in_connections(i,j);
    end;
    entradas(cores_no+1) = entradas(cores_no+1) +
in_connections(i,cores_no+1);
    saidas(cores_no+1) = saidas(cores_no+1) + in_connections(cores_no+1,i);
    entradas(i) = entradas(i) + in_connections(i,cores_no+1);
    saidas(i) = saidas(i) + in_connections(cores_no+1,i);
end;

```

```

for i=1:modules_no,
    if (entradas(i) > (features(i,inputs_no) + features(i,bidirs_no)))
        fprintf('core %d tem MAIS conexoes de entrada que inputs_no. entradas
= %d, inputs_no = %d
\n',i,entradas(i),features(i,inputs_no)+features(i,bidirs_no));
    end;
    if (entradas(i) < (features(i,inputs_no) + features(i,bidirs_no)))
        fprintf('core %d tem MENOS conexoes de entrada que inputs_no.
entradas = %d, inputs_no = %d
\n',i,entradas(i),features(i,inputs_no)+features(i,bidirs_no));
    end;
end;

% #####
% Havendo núcleos hierarquizados com 'setups' de teste, eles devem ser
% associados aqui com o núcleo de hierarquia superior, no qual está incluso
% à esquerda (a) fica o núcleo incluso, enquanto que à direita (b) está o
% módulo hierarquicamente superior
% in_connections(a,:) = in_connections(b,:);
% in_connections(:,a) = in_connections(:,b);

% #####3
% Define o limite de pinos extras como uma percentagem do número total de
% pinos do sistema

total_pinos = 0;
for i=1:cores_no-no_busses,
    total_pinos =
total_pinos+in_connections(i,cores_no+1)+in_connections(cores_no+1,i);
end;

for i=cores_no-no_busses+1:cores_no,
    total_pinos = total_pinos+in_connections(i,cores_no+1);
end;

total_pinos

limite_pinos = ceil(percentagem_pinos*total_pinos)

% #####
% Definições de posicionamento
% Baseados nas coordenadas dos núcleos, ou alternativamente, através de uma
% tabela de distâncias, como será no nosso caso

% #####
% Definição de distâncias
%
% Tabela de distâncias
%
% Esta tabela dá a distância relative entre núcleos e entre um núcleo e a
% interface do sistema
% As últimas duas colunas correspondem às interfaces de entrada e saída,
% respectivamente

sys_distances = zeros(cores_no+1, cores_no+1);

```



```

sys_distancesb = [
    0   800  1506   742   735   500   500  1012  1200  2001
    800   0  6369  8246  1514  6992  5929  2117  4463  4187
  1506  6369   0  1877  7309  2707  3770  5514  3430  2510
    742  8246  1877   0  9186  1738  2801  6129  5307  4059
    735  1514  7309  9186   0  7932  6869  3207  3879  5127
    500  6992  2707  1738  7932   0  1063  4875  4053  2805
    500  5929  3770  2801  6869  1063   0  3812  4412  1742
  1012  2117  5514  6129  3207  4875  3812   0  6156  3004
  1200  4463  3430  5307  3879  4053  4412  6156   0  3152
  2001  4187  2510  4059  5127  2805  1742  3004  3152   0
];

sys_distancesb = sys_distancesb*1e-6;

% Adaptação para ler a tabela gerada pelo placement com simulated
% annealing
for i=1:modules_no,
    sys_distances(cores_no+1,i) = sys_distancesb(1,i+1);
    sys_distances(i,cores_no+1) = sys_distancesb(i+1,1);
end;
for i=2:modules_no+1,
    for j=2:modules_no+1,
        sys_distances(i-1,j-1) = sys_distancesb(i,j);
    end;
end;

% Calcula tamanho medio das conexões
% tamanho de uma net é: no. fios*comprimento_fio
%
no_conexoes = 0;
total_conex = 0;

for i=1:cores_no+1,
    for j=1:cores_no+1,
        if (in_connections(i,j) > 0)
            no_conexoes = no_conexoes + 1;
            tamanho_conex = in_connections(i,j)*sys_distances(i,j);
            total_conex = total_conex + tamanho_conex;
        end;
    end;
end;

media_conex = total_conex/no_conexoes

% #####
% DEFINIÇÕES DE VIZINHANÇA
% 4 tipos de vizinhança são definidos para cada núcleo:
% entrada scan, saída scan, entrada funcional e saída funcional
%
% Vizinhança SCAN
%
% Inicialmente sao calculados os caminhos para scan apenas. Depois, com as
% restricoes impostas pelo scan sao calculados os caminhos das conexoes
% funcionais.

```

```

% Serão considerados como "vizinho habilitado para trazer dados scan", os
% vizinhos :
% a) que estao posicionados antes de cut em direcao a entrada
% (x_vizinho < x_cut) e que tbem usam scan.
% b) que estao posicionados na mesma coordenada x de cut e que sejam
% scan_based
%
% Serão considerados como "vizinho habilitado para levar dados scan para a
% saida do chip", os vizinhos :
% a) que estao posicionados depois de cut em direcao a saida
% (x_vizinho > x_cut) e que tbem sejam scan-based
% b) que estao posicionados na mesma coordenada x de cut e que sejam
% scan_based.

vizinhanca_sc_in = ones(cores_no, cores_no+1);
vizinhanca_sc_out = ones(cores_no, cores_no+1);

% o núcleo nível 0 (maior hierarquicamente) não é vizinho de ninguém
vizinhanca_sc_in(:,level_0) = 0;
vizinhanca_sc_out(:,level_0) = 0;
vizinhanca_sc_in(level_0,:) = 0;
vizinhanca_sc_out(level_0,:) = 0;
vizinhanca_sc_in(level_0,cores_no+1) = 1;
vizinhanca_sc_out(level_0,cores_no+1) = 1;

% Vizinhança FUNCIONAL
%
% Serão considerados como "vizinho habilitado para trazer dados", os
% vizinhos :
% a) que estao posicionados antes de cut em direcao a entrada
% (x_vizinho < x_cut)
% b) que estao posicionados na mesma coordenada x de cut e que estão
% conectados as entradas de cut.
% c) que tem conexao com a interface de entrada (para aumentar as chances
% de minimizar pinos).
%
% Serão considerados como "vizinho habilitado para levar dados para a saída
% do chip", os vizinhos :
% a) que estao posicionados depois de cut em direcao a saida
% (x_vizinho > x_cut)
% b) que estao posicionados na mesma coordenada x de cut e que estão
% conectados as saidas de cut.
% c) que tem conexao com a interface de saida (para aumentar as chances de
% minimizar pinos).
% Nesta primeira iteracao, o core mais caro em termos de tempo nao é
% vizinho de ninguem!

vizinhanca_ext_in = ones(cores_no, cores_no+1);
vizinhanca_ext_out = ones(cores_no, cores_no+1);

% O core de nivel 0 nao é vizinho de ninguem, pois já se consideram as
% conexoes com a interface diretamente
vizinhanca_ext_in(:,level_0) = 0;
vizinhanca_ext_out(:,level_0) = 0;
vizinhanca_ext_in(level_0,:) = 0;
vizinhanca_ext_out(level_0,:) = 0;
vizinhanca_ext_in(level_0,cores_no+1) = 1;

```

```

vizinhanca_ext_out(level_0,cores_no+1) = 1;

% Núcleos Virtuais (Embedded hierarchical e BISTed) não são vizinhos de
% ninguém
for i=1:cores_no,
    if ( (features(i,embedded) == 1) | (features(i,bist_inputs) > 0) )
        vizinhanca_sc_in(:,i) = 0;
        vizinhanca_ext_in(:,i) = 0;
        vizinhanca_sc_out(:,i) = 0;
        vizinhanca_ext_out(:,i) = 0;
    end;
end;

% vizinhanca especifica para nucleos analogicos e mistos
i=1;
vizinhanca_sc_in(i,:) = 0;
vizinhanca_ext_in(i,:) = 0;
vizinhanca_sc_in(i,cores_no+1) = 1;
vizinhanca_ext_in(i,cores_no+1) = 1;

vizinhanca_sc_out(i,:) = 0;
vizinhanca_ext_out(i,:) = 0;
% BISTed
vizinhanca_sc_out(i,prec) = 1;
vizinhanca_ext_out(i,prec) = 1;
% NoBISTed
% vizinhanca_sc_out(i,cores_no+1) = 1;
% vizinhanca_ext_out(i,cores_no+1) = 1;
% for j=4:9,
%     vizinhanca_sc_out(i,j) = 1;
%     vizinhanca_ext_out(i,j) = 1;
% end

i=2;
vizinhanca_sc_in(i,:) = 0;
vizinhanca_ext_in(i,:) = 0;
vizinhanca_sc_in(i,cores_no+1) = 1;
vizinhanca_ext_in(i,cores_no+1) = 1;

vizinhanca_sc_out(i,:) = 0;
vizinhanca_ext_out(i,:) = 0;
% BISTed
vizinhanca_sc_out(i,prec) = 1;
vizinhanca_ext_out(i,prec) = 1;
% NoBISTed
% vizinhanca_sc_out(i,cores_no+1) = 1;
% vizinhanca_ext_out(i,cores_no+1) = 1;
% for j=4:9,
%     vizinhanca_sc_out(i,j) = 1;
%     vizinhanca_ext_out(i,j) = 1;
% end

i=3;
vizinhanca_sc_in(i,:) = 0;
vizinhanca_ext_in(i,:) = 0;

```

```

vizinhanca_sc_in(i,cores_no+1) = 1;
vizinhanca_ext_in(i,cores_no+1) = 1;

vizinhanca_sc_out(i,:) = 0;
vizinhanca_ext_out(i,:) = 0;
% BISTed
vizinhanca_sc_out(i,prec) = 1;
vizinhanca_ext_out(i,prec) = 1;
% NoBISTed
% vizinhanca_sc_out(i,cores_no+1) = 1;
% vizinhanca_ext_out(i,cores_no+1) = 1;
% for j=4:9,
%     vizinhanca_sc_out(i,j) = 1;
%     vizinhanca_ext_out(i,j) = 1;
% end

% #####
% Informação de PRECEDENCIA, quando for o caso
% Para núcleos que usem recursos específicos (inclusive outros núcleos)
% durante o teste, como blocos analógicos sendo testados pelos digitais
%
% A precedência é definida por uma matriz: as linhas indicma os núcleos
% (usuários) e as olunas os recursos (nesse caso, outros núcleos presentes
% no sistema). O valor '1' numa posição (i,j) na matrix indica que o núcleo
% 'i' usa o núcleo 'j' durante o teste, implicando que o núcleo 'j' seja
% escalonado primeiro, antes de se iniciar o teste do núcleo 'i'. Além
% disso, quando acha uma TAM para o núcleo 'i', ela deve terminar no núcleo
% 'j', e a interface do sistema deixa de ser vizinha para a saída do núcleo
% 'i'.
%
% Precedência é definida e mambas as direções, isto é, geração de padrões
% de teste e análise da resposta.

precedentes_in = zeros(cores_no, cores_no+1);
precedentes_out = zeros(cores_no, cores_no+1);

% como exemplo, para teste
precedentes_out(1,prec) = 1;
precedentes_out(2,prec) = 1;
precedentes_out(3,prec) = 1;

% #####
% Adicionando area das trilhas para cada núcleo

for i=1:modules_no,
    for j=1:modules_no,
        features(j,area_fio) = features(j,area_fio) +
in_connections(i,j)*sys_distances(j,i)*largura_fio;
    end;
end;

% #####
% Gera arquivo de entrada para placement, se requerido

if (gera_placement == 1)
    nome = strcat(system_name, '_place.txt');
    arquivo = fopen(nome, 'w');

```

```

fprintf(arquivo, '%%SISTEMA %s \n\n',system_name);

fprintf(arquivo,'%%Numero de Cores\n%d\n', modules_no);
fprintf(arquivo,'\n%%Dimensoes dos cores: core W H\n\n');

for i=1:modules_no,
    fprintf(arquivo,'%d %4d %4d\n', i,fix(features(i,clength)/1e-
6),fix(features(i,cheigth)/1e-6));
end;

fprintf(arquivo,'\n%%Conexoes\n');

for i=1:modules_no,
    for j=i:modules_no,
        valor = in_connections(i,j) + in_connections(j,i);
        if (valor > 0)
            fprintf(arquivo,'%d %d %d\n',valor,i,j);
        end;
    end;
end;

fprintf(arquivo,'\n%%Interface\n');
for i=1:modules_no,
    valor = in_connections(i,cores_no+1) + in_connections(cores_no+1,i);
    if (valor > 0)
        fprintf(arquivo,'%d %d 0\n',valor,i);
    end;
end;

a = 1;b = 1; c= 1;
for i=level_0+1:cores_no,
    a = a+5;b=b+5;c=c+5;
    temp = find(in_connections(i,:));
    fprintf(arquivo,'\n%%barramento %d - %d
bits\n',i,features(i,bidirs_no));
    fprintf(arquivo,'\n(%d %d %d) %%muda a cor \n',a,b,c);
    fprintf(arquivo,'%d ',features(i,bidirs_no));
    for j=1:length(temp),
        if (temp(j) == cores_no+1)
            fprintf(arquivo,'0 ');
        elseif (temp(j) <= modules_no)
            fprintf(arquivo,'%d ',temp(j));
        end;
    end;
    fprintf(arquivo,'\n');
end;

fclose(arquivo);
end;

% FIM da definição da estrutura de dados

```