

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTOS DE INFORMÁTICA, QUÍMICA, FÍSICA
PROGRAMA DE PÓS-GRADUAÇÃO EM MICROELETRÔNICA

LUCAS BERNARDO ZILCH

**Geração Automática de Padrões para Teste Estrutural de
Circuitos Analógicos**

Dissertação apresentada como requisito parcial
para a obtenção de grau de Mestre em
Microeletrônica.

Orientador: Prof. Dr. Tiago Roberto Balen

Co-orientador: Prof. Dr. Marcelo Soares
Lubaszewski

Porto Alegre
Março de 2021

CIP - Catalogação na Publicação

Zilch, Lucas Bernardo
Geração Automática de Padrões para Teste Estrutural
de Circuitos Analógicos / Lucas Bernardo Zilch. --
2021.
85 f.
Orientador: Tiago Roberto Balen.

Coorientador: Marcelo Soares Lubaszewski.

Dissertação (Mestrado) -- Universidade Federal do
Rio Grande do Sul, Instituto de Informática, Programa
de Pós-Graduação em Microeletrônica, Porto Alegre,
BR-RS, 2021.

1. Circuitos integrados analógicos. 2. Geração
automática de vetores de teste. 3. Dicionário de
falhas. 4. Cobertura de falhas. I. Balen, Tiago
Roberto, orient. II. Lubaszewski, Marcelo Soares,
coorient. III. Título.

"Each of us must work for his own improvement, and at the same time share a general responsibility for all humanity"

-MARIE CURIE

AGRADECIMENTOS

Agradeço à toda a minha família, em especial à minha mãe Regina, ao meu pai Antenor (em memória) e à minha irmã Daniela, por todo incentivo que me foi dado e pelos exemplos que sempre representaram para mim.

Agradeço ao Prof. Dr. Tiago Roberto Balen e ao Prof. Dr. Marcelo Soares Lubaszewski por todas as orientações dadas ao desenvolvimento do presente trabalho e, principalmente, por se demonstrarem bastante prestativos em sanar os meus questionamentos.

RESUMO

Os vultuosos incrementos de complexidade e de funcionalidade que os circuitos eletrônicos atuais apresentam em relação aos seus antecessores deram-se através da miniaturização dos componentes. Essa redução das dimensões incrementou os desafios impostos pelas etapas de projeto e fabricação, aumentando, por consequência, a importância da etapa de teste. Ao mesmo tempo em que os testes precisam ter boa qualidade, apresentando elevadas coberturas de falhas e baixa *yield loss*, o custo também é um fator primordial a ser levado em consideração. A metodologia de teste baseado em especificação se mostra como uma opção mais cara que a de teste baseado em defeitos. Entretanto, o ônus da opção mais barata está no possível decréscimo da qualidade do teste e da maior dificuldade na determinação das melhores configurações de testes a serem utilizadas. Nesse contexto, nota-se que os circuitos analógicos estão atrás dos digitais quando o assunto é a determinação dessas configurações de teste, pois, enquanto os circuitos digitais possuem, já há bastante tempo, ferramentas para determinação de vetores de teste otimizados, essa determinação, para testes analógicos, ainda não é totalmente automatizada (KABISATPATHY; BARUA; SINHA, 2005). De posse dessas informações, o presente trabalho consiste no desenvolvimento de uma ferramenta que vise amenizar essa discrepância entre circuitos analógicos e digitais. A ferramenta foi desenvolvida em MATLAB de modo a automatizar simulações SPICE de circuitos e, por fim, também de forma automatizada, analisar todos os resultados e chegar na conclusão de quais configurações de testes correspondem ao conjunto mais otimizado para aquele circuito dentro das condições simuladas. Nessa ferramenta, deve-se entrar com a descrição SPICE do circuito e do modelo de falhas que se deseja adotar, sendo esse composto por falhas paramétricas e catastróficas, cujas respectivas impedâncias e desvios são escolhidas pelo usuário. A ferramenta desconsidera que uma falha possa mascarar a outra e, por isso, cria uma descrição SPICE do circuito para cada uma das falhas de maneira individual. Através da simulação de todos os circuitos com falha e do circuito *fault free*, juntamente com a posterior comparação dos resultados, a ferramenta cria o dicionário de falhas. Esse dicionário contém as informações de quais falhas podem ser detectadas em cada um dos nós e para cada um dos possíveis sinais de entrada. Com ele, a ferramenta determina quais são as coberturas de falhas nos nós e qual o melhor conjunto testes. As funcionalidades da ferramenta foram avaliadas através de estudos de caso que consistiram na determinação de configurações de teste otimizadas para um amplificador totalmente diferencial de dois estágios e um filtro passa-baixas de segunda ordem composto pela conexão de dois estágios de primeira ordem em cascata. No primeiro caso, o amplificador apresentou cobertura de falhas máxima de 63,88%, porém com auxílio de funcionalidades da ferramenta, observou-se, que utilizando um nó interno do circuito como nó de teste, essa cobertura de falhas é aumentada para 73,22%. No segundo caso, do filtro passa-baixas de segunda ordem, a cobertura de falhas alcançada foi de 87,15%. Subsequentemente, foram investigadas as possibilidades de execução de um teste transiente adicional e da execução de testes nos amplificadores do circuito em malha aberta, ambas as análises buscavam o aumento da cobertura de falhas. O teste transiente resultou na detecção de uma única falha adicional, porém o teste dos amplificadores operando em malha aberta resultaram em um aumento significativo na cobertura de falhas, que chegou ao valor de 92,36%.

Palavras Chave: Circuitos integrados analógicos, teste, dicionário de falhas, vetores de teste, cobertura de falhas, geração automática de vetores.

ABSTRACT

The great increases in complexity and functionality that modern electronic circuits present in relation to their past generations happened through the miniaturization of their components. This dimension reduction has increased the challenges posed by project and fabrication phases, increasing therefore, the importance of the test phase. While the tests must be of good quality, presenting high fault coverage and low yield loss, the cost must also be a prime factor to be considered. The methodology of specification-based test presents itself as a more expensive option than the defect-based test. However, the burden of the cheapest option is the possible decrease of test quality and the greater difficulty in determining the best test configuration to be used. In this context, it is noted that determination of test configurations for analog circuits, is a step behind of the digital counterparts. This is because, while for digital circuits there are tools for optimized test vector determination, this determination, for analog tests, is still not totally automated (KABISATPATHY; BARUA; SINHA, 2005). Given this information, this work consists in the development of a tool that aims to soften this discrepancy between analog and digital circuits. A tool was developed in MATLAB in a way to automate SPICE circuit simulations and, finally, also in automated form, analyze the results and arrive at a conclusion about what test configurations correspond to the best optimized set for that circuit within the simulated conditions. In this tool, one must enter the SPICE circuit description and the fault model that one wishes to adopt, being the model composed of parametric and catastrophic faults, and whose respective impedances and deviations are chosen by the user. The tool disregards that one fault might mask the other and, therefore, creates an individual fault SPICE circuit description for each one. Through the simulation of all faulty circuits and the fault free circuit, along with a posterior result comparison, the tool creates the fault dictionary. This dictionary contains the information of which faults may be detected in each of the nodes (used as test point) and for each of the possible input signals. Finally, with the dictionary, the tool determines the fault coverage in each node and which are the best tests sets. The functionalities of the tool were evaluated through case studies that consisted in the determination of the optimized test configuration for a two-stage fully differential amplifier and a second order low pass filter. In the first case, the amplifier presented an initial fault coverage of 63,88%. However, with the assistance of the tool, it was observed that, using an internal node of the circuit as test node, this fault coverage increases to 73,22%. In the second case study (second order low pass filter), the fault coverage reached 87,15%. Subsequently, the execution of an additional transient test and the execution of tests in the amplifiers with open loop circuit were investigated; both the analysis searched for an increase in fault coverage. The transient test resulted in the detection of a single additional fault, however the test of the amplifiers operating in open loop resulted in a significant increase in fault coverage, reaching 92,36%.

KEYWORDS: Analog integrated circuits, test, fault dictionary, test vectors, fault coverage, automated vector generation.

LISTA DE FIGURAS

Figura 1 - Estimativa de custo por falha detectada em cada etapa de produção.	19
Figura 2 - <i>Corners</i> de uma tecnologia.	26
Figura 3 - Diagrama em blocos do verificador proposto.	28
Figura 4 - Bloco conceitual do circuito CMFB do FDA.	30
Figura 5 - Fluxograma geral da toolbox para análise de configurações de teste em circuitos analógicos.	32
Figura 6 - Fluxograma de funcionamento da janela "AnalogCircuitsTest".	33
Figura 7 - Primeira etapa da Janela "AnalogCircuitsTest".	34
Figura 8 - Inserção de parâmetros de simulação.	35
Figura 9 - Exemplo de arquivo de descrição de falhas.	38
Figura 10 - Escolha do arquivo de descrição de falhas.	39
Figura 11 - Descrição do modelo de falhas.	40
Figura 12 - Seleção dos nós.	41
Figura 13 - Exemplo de matriz de falhas.	43
Figura 14 - Representação do algoritmo para determinação do melhor conjunto de testes.	48
Figura 15 - Janela "Results" para o caso de teste DC e AC de um amplificador.	50
Figura 16 - Janela para consulta dos modelos de falhas utilizados.	51
Figura 17 - Gráfico com a resposta dos <i>corners</i> aberto através do botão "View Corners" para nó "vop".	52
Figura 18 - Janela para alteração dos nós de teste.	53
Figura 19 - Janela para execução de análise transiente.	54
Figura 20 - Resultados da análise transiente.	55
Figura 21 - Configuração do amplificador totalmente diferencial de dois estágios em malha fechada.	57
Figura 22 - Diagrama esquemático dos estágios principais e de compensação em avanço	58
Figura 23 - Diagrama esquemático do bloco de CMFB projetado em tecnologia IBM 130nm.	59
Figura 24 - Diagrama esquemático do bloco de polarização do FDA projetado em tecnologia IBM 130nm.	59
Figura 25 - Diagrama esquemático do filtro passa-baixas de dois estágios.	61
Figura 26 - Esquemático do amplificador totalmente diferencial de um estágio.	62
Figura 27 - Esquemático elétrico do circuito CMFB.	63

Figura 28 - Diagrama esquemático do FDA de um estágio em malha aberta.	66
Figura 29 - Conjunto dos melhores testes considerando apenas nós de saída para amplificador totalmente diferencial de dois estágios.	69
Figura 30 - Conjunto dos melhores testes considerando nó interno para amplificador totalmente diferencial de dois estágios.	70
Figura 31 - Conjunto dos melhores testes considerando apenas nós de saída para filtro passa-baixas de dois estágios.	71
Figura 32 - Resposta transiente do nó de saída negativa do circuito com falha no transistor X6.	72
Figura 33 - Representação gráfica dos tempos de acomodação de todos os circuitos com falha.	73
Figura 34 - Conjunto dos melhores testes considerando apenas nós de saída para amplificador totalmente diferencial de um estágio.	74

LISTA DE TABELAS

Tabela 1 – Dimensões dos transistores do FDA projetado em tecnologia IBM 130nm.	60
Tabela 2 - Especificações do projeto do circuito FDA.....	64
Tabela 3 - Tempos de execução do algoritmo.....	68

LISTAS DE ABREVIATURAS E SIGLAS

AC	Alternating Current
ADC	Analog to digital converter
ATE	Automatic Test Equipment
ATPG	Automatic Test-Pattern Generation
CF	Cobertura de Falhas
CI	Circuito Integrado
CMFB	Common-Mode Feed-Back
CMOS	Complementary Metal Oxide Semiconductor
CUT	Circuit Under Test
DA	Differential Amplifier
DC	Direct Current
DFT	Design For Test
FDA	Fully Differential Amplifier
FF	Fast-Fast
FS	Fast-Slow
GBW	Gain-Bandwidth product
NMOS	N-channel Metal Oxide Semicondutor
PMOS	P-channel Metal Oxide Semicondutor
RC	Resistivo Capacitivo
RF	Rádio Frequência
SF	Slow-Fast
SPICE	Simulation Program with Integrated Circuits Emphasis
SS	Slow-Slow
TRAM	Transient Response Analysis Method
TT	Typical-Typical
WO	Worst Case One Condition
WP	Worst Case Power Condition
WS	Worst Case Speed Condition
WZ	Worst Case Zero Condition

SUMÁRIO

1	INTRODUÇÃO	13
2	REVISÃO BIBLIOGRÁFICA	17
2.1	TESTES DE CIRCUITOS ANALÓGICOS.....	17
2.1.1	Teste em Produção	18
2.1.2	Teste Funcional versus Teste Estrutural	20
2.1.3	Geração de Teste	21
2.1.4	Modelo de Falhas Para Teste Analógico	22
2.1.5	Dicionário de Falhas	23
2.2	CARACTERIZAÇÃO DE VARIABILIDADES DE PROCESSO	24
2.2.1	Análise de Corners	25
2.2.2	Análise de Monte Carlo	26
2.3	METODOLOGIAS DE TESTE UTILIZADAS NESSE TRABALHO	27
2.3.1	Teste Transiente	27
2.3.2	Teste DC	27
2.3.3	Teste AC	28
2.4	AMPLIFICADORES TOTALMENTE DIFERENCIAIS	29
3	DESCRIÇÃO DA FERRAMENTA	31
3.1	JANELA “ANALOGCIRCUITTEST”	33
3.1.1	Etapa 1: Inserção de Arquivos para Simulação	33
3.1.2	Etapa 2: Parâmetros de Simulação	34
3.1.3	Etapa 3: Escolha das Falhas que Irão Compor o Modelo	37
3.1.4	Etapa 4: Escolha do Modelo Elétrico para as Falhas Catastróficas	39
3.1.5	Etapa 5: Determinação do Dicionário de Falhas e Resultados	41
3.2	JANELA “RESULTS”	44
3.2.1	Determinação da Máxima Cobertura de Falhas	45
3.2.2	Determinação do Melhor Conjunto de Testes	46
3.2.3	Exibição dos Resultados	49
3.2.4	Possibilidades de Alterações das Análises	52
3.2.5	Análise Transiente para Possíveis Otimizações de Execução de Testes DC	53
4	METODOLOGIA EXPERIMENTAL – ESTUDOS DE CASO	56
4.1	AMPLIFICADOR TOTALMENTE DIFERENCIAL DE DOIS ESTÁGIOS COM ANÁLISE DE MONTE CARLO	57
4.2	FILTRO PASSA-BAIXAS DE DOIS ESTÁGIOS.....	61

4.2.1	Análises AC e DC do Filtro Passa-Baixas Completo	64
4.2.2	Análise Transiente do Filtro Passa-Baixas Completo	66
4.2.3	Amplificador Totalmente Diferencial em Malha Aberta	66
5	RESULTADOS E DISCUSSÕES	68
5.1	AMPLIFICADOR TOTALMENTE DIFERENCIAL DE DOIS ESTÁGIOS COM ANÁLISE DE MONTE CARLO PARA DEFINIÇÃO DOS LIMITES DE TESTE	68
5.2	FILTRO PASSA-BAIXAS DE DOIS ESTÁGIOS	70
5.2.1	Análises AC e DC do Filtro Passa-Baixas Completo	70
5.2.2	Análise Transiente do Filtro Passa-Baixas Completo	71
5.2.3	Amplificador Totalmente Diferencial em Malha Aberta	73
6	CONCLUSÃO	76
7	SUGESTÕES PARA TRABALHOS FUTUROS	78
	APÊNDICE A – DESCRIÇÃO SPICE DO FDA DE UM ESTÁGIO EM MALHA ABERTA	84

1 INTRODUÇÃO

Um importante marco para o desenvolvimento tecnológico que presenciamos hoje foi a criação do primeiro transistor há quase 70 anos. Esse período de apenas sete décadas foi suficiente para promover a evolução dos primeiros transistores, que eram encapsulados de forma individual, até os CIs (circuitos integrados) de hoje em dia, que chegam a conter bilhões desses componentes interligados em um mesmo chip. Essa rápida evolução foi possibilitada pela profunda diminuição das dimensões dos transistores que ocorreu de acordo com a lei de Moore (SEGURA; HAWKINS, 2004; ARSLAN; ORAILOGLU, 2013), a qual propõe que a capacidade de integração dos transistores em um circuito integrado digital dobra a cada 18 meses (MOORE, 1965). O que presenciamos nos dias atuais é que a indústria da microeletrônica alcançou patamares gigantescos e que os circuitos eletrônicos se tornaram muito mais complexos, também em termos funcionais, do que os de antigamente, além de se tornarem mais eficientes em consumo energético. Hoje, com um *smartphone* suficientemente pequeno para caber no bolso, consegue-se executar tarefas como: vídeo-chamadas, acessar a internet, fazer compras, consultas bancárias, entre outras funções.

Entretanto, ao mesmo tempo que as funcionalidades dos circuitos evoluíram, a complexidade de projetar e fabricar esses circuitos também seguiu o mesmo caminho, tornando essas tarefas cada vez mais desafiadoras. Processos de fabricação atuais trabalham com dimensões tão pequenas que controlá-los tornou-se uma tarefa altamente complexa (QUIAN, 2015) o que vem requerendo cada vez mais o estudo de suas variabilidades (MEZZOMO; BAJOLET; CATHIGNOL; DI FRENZA; GHIBAUDO, 2011). Assim como as dimensões dos transistores diminuem, crescem as variabilidades de processo (SHIN, 2016), aumentam os efeitos de canal curto, nos quais há uma diminuição da tensão de *threshold*, e também aumentam exponencialmente as correntes de fuga, devido à diminuição da espessura do óxido (TAUR et al., 1996).

Todos esses aspectos apontam para a importância de outra etapa da concepção de circuitos eletrônicos, a etapa de teste. Em meio às crescentes dificuldades em projetar e fabricar CIs, garantir o correto funcionamento desses circuitos é essencial e, ao mesmo tempo, desafiador. Sistemas de alta segurança – como, por exemplo, de automóveis, aviões, trens de alta velocidade e plantas nucleares – não podem tolerar falhas e, por isso, requerem procedimentos altamente confiáveis para detecção de falhas nos chips (LUBASZEWSKI; MIR; KOLARIK; NIELSEN; COURTOIS, 2000). Nesse contexto, a busca por uma constante potencialização da qualidade dos testes a serem executados nos circuitos recém fabricados vem se tornando cada vez mais difícil (KABISATPATHY; BARUA; SINHA, 2005). Essa qualidade

é atrelada aos conceitos de cobertura de falhas e *yield lost*, bem como a minimização do preço de execução do teste, dado que esse custo será adicionado ao preço do produto e pode representar uma perda de competitividade no mercado.

A implementação de circuitos digitais, na maioria dos casos, foi preferida em relação a de analógicos. Isso porque, em muitas aplicações, o projeto de um circuito digital se mostra muito mais simples que o do seu equivalente analógico, devido ao fato de que os digitais possuem finitos possíveis estados, diferentemente dos analógicos que, por sua vez, apresentam infinitos possíveis valores de saída. Entretanto, muitas aplicações exigem a utilização de circuitos analógicos, ou seja, não existem equivalentes digitais. O interfaceamento dos circuitos eletrônicos com as variáveis físicas é um bom exemplo disso, pois o mundo físico é, por natureza, analógico. Nesse contexto, os circuitos amplificadores são de fundamental importância, pois são utilizados tanto nos estágios de condicionamento dos sinais analógicos, quanto na implementação de filtros ativos, indispensáveis em quase todas as situações em que um sinal analógico é amostrado por um ADC (*Analog to digital converter*) de modo a tornar-se digital.

Naturalmente, circuitos digitais e analógicos também diferem-se com relação aos testes. Devido ao fato de que os circuitos analógicos possuem infinitos possíveis sinais de saída, a comparação do resultado dos seus testes com o valor esperado não é tão direta quanto o seu equivalente digital. De um circuito digital, espera-se, para cada sinal de entrada, um vetor de saída específico, no qual a não conformidade de um único bit é suficiente para classificar o circuito como falho. Já um circuito analógico, mesmo não sendo falho, não apresentará resposta idêntica a uma resposta esperada, mas sim uma resposta dentro de uma faixa de tolerância.

Devido a essa diferença básica, é comum que circuitos analógicos sejam testados de acordo com a metodologia de teste baseado em especificações, onde o objeto de análise do teste não é diretamente o valor de tensão da saída, mas sim as especificações do *datasheet*. Segundo essa metodologia, todas as especificações do *datasheet* são testadas, uma a uma, até que se garanta que todas estão dentro da faixa de tolerância. Embora essa metodologia de teste apresente excelente qualidade (STRATIGOPOULOS; SUNTER, 2014), o custo de implementação é consideravelmente maior que o custo do teste de um circuito digital que ocupe a mesma área no chip (POEHL; DEMMERLE; ALT; OBERMEIR, 2010).

Buscando testes de boa qualidade e baixo custo para circuitos integrados (CIs), a metodologia mais indicada é a do teste estrutural (RENOVELL, 2004). Segundo ela, é proposto um modelo de falhas para o circuito em análise, com o qual, através de simulações, levanta-se o dicionário de falhas. Nesse contexto, a determinação dos vetores de teste para circuitos

analógicos é feita, em geral, por tentativa e erro. Entretanto, embora ferramentas para determinação de vetores de teste otimizados para circuitos digitais já existam há mais de três décadas, essa determinação, para testes analógicos, ainda não é totalmente automatizada (KABISATPATHY; BARUA; SINHA, 2005). Dessa forma, a carência por uma ferramenta de automatização pode inviabilizar a análise de um grande número de possíveis vetores de teste para circuitos analógicos.

Dessa questão, resultou a proposta do presente trabalho, que visou o desenvolvimento de um a ferramenta automatizada, capaz de analisar um número elevado de possíveis configurações de teste, e, através dessa análise, determinar quais as melhores configurações de teste de modo a otimizar o custo, preocupando-se em atingir a máxima qualidade possível dentro das condições estipuladas. Além disso, a ferramenta visou possibilitar a reanálise rápida dos dados já calculados, bem como possibilitar a alteração, por parte do usuário, das condições estabelecidas, resultando assim em novos resultados.

Dentro desse contexto, o desenvolvimento dos códigos de automatização iniciou com o trabalho de Chinazzo (2016). Em seu trabalho, o autor desenvolveu uma primeira versão de códigos em MATLAB para execução de simulações SPICE (*Simulation Program with Integrated Circuits Emphasis*) automatizadas, através das quais era possível gerar e simular a descrição de circuitos contendo falhas pré-estabelecidas para, então, analisar quais são as melhores configurações de teste. Entretanto, todas as análises eram feitas sem nenhuma interface gráfica para interação do usuário com o algoritmo, de modo que os parâmetros de simulação eram definidos ao longo o próprio código, e os resultados eram expostos na janela de comando.

Utilizando esses códigos, a proposta para esse trabalho, bem como o que foi iniciado em Zilch (2017), foi incrementar esses códigos em suas funcionalidades, além de criar uma toolbox em MATLAB que possibilitasse uma interface amigável entre o usuário e o algoritmo de automatização. Além disso, novas funções foram inseridas aos códigos, como, por exemplo, adicionar a possibilidade de analisar mais do que um sinal de entrada no caso de tensões DC (*Direct Current*), criar novas lógicas para determinação dos limites de detecção das falhas, destacando-se aqui a inserção do método de Monte Carlo às opções, e possibilitar uma rápida releitura dos resultados determinados anteriormente. Por fim, em relação aos resultados, foi criada uma lógica para determinação do melhor conjunto de testes, além de tornar possível a mudança das condições de teste, como, por exemplo, seleção de novos nós, alteração dos limites de detecção e restrição dos possíveis sinais de entrada, fazendo com que o algoritmo recalcule os resultados para as novas condições.

Os resultados mostraram que a ferramenta apresentou bons tempos de execução e se mostrou prática para a reanálise dos resultados como novas condições. Para o caso de um amplificador totalmente diferencial de dois estágios, a cobertura de falhas pode ser acrescida em quase 10% apenas por considerar a possibilidade de incluir um nó interno do circuito no conjunto de testes. Já no caso de um filtro passa-baixas de dois estágios em cascata, observou-se que a cobertura de falha de 87,15% poderia ser elevada a até 92,36% se os amplificadores do circuito pudessem ser testados separadamente e em malha aberta.

O presente trabalho divide-se em sete capítulos. No capítulo 2, é feita uma revisão bibliográfica sobre testes em circuitos integrados, dando-se ênfase aos analógicos, que são os objetos de estudo. A descrição geral da *toolbox* desenvolvida se encontra no capítulo 3, onde são detalhados tanto aspectos operacionais de utilização da ferramenta, quanto aspectos de programação, como a lógica utilizada na determinação dos resultados. No capítulo 4, são detalhadas as metodologias experimentais adotadas no trabalho. Os resultados desses estudos, juntamente com as considerações feitas, são encontrados no capítulo 5. Após a exposição dos resultados, são feitas as conclusões acerca de todo o trabalho executado, as quais se encontram no capítulo 6. Por fim, no capítulo 7, são abordadas sugestões de melhorias para a *toolbox* desenvolvida, juntamente com possíveis sugestões para continuação desse trabalho.

2 REVISÃO BIBLIOGRÁFICA

No presente capítulo, será feita uma revisão bibliográfica sobre conceitos importantes ao trabalho. Em um primeiro momento, serão descritos conceitos sobre teste de circuitos analógicos, principal objeto de estudo do trabalho. Uma vez que os circuitos analógicos não apresentam uma diferenciação tão clara quanto os digitais entre um sinal de saída esperado e outro fora do que é aceitável, é de fundamental importância que as variabilidades de processo sejam discutidas, bem como as suas metodologias de caracterização. Será através disso que os limites de aceitação para as variáveis medidas no teste analógico serão determinados.

2.1 TESTES DE CIRCUITOS ANALÓGICOS

Na indústria de semicondutores, a utilização de circuitos integrados que incorporam tanto funções digitais quanto analógicas está se tornando cada vez mais comum. Entretanto, enquanto as metodologias de testes para circuitos digitais já estão bem estabelecidas, as técnicas de teste para circuitos integrados analógicos ainda são pouco padronizadas. Ferramentas computacionais para geração de testes automatizados em circuitos integrados digitais já existem há mais de três décadas, ao mesmo tempo que a determinação de sinais otimizados para testes em circuitos analógicos ainda não é totalmente automatizada, normalmente dependendo da experiência do projetista e de especificações sobre a funcionalidade do circuito (KABISATPATHY; BARUA; SINHA, 2005; MILOR, 1998). Isso se deve ao fato de que os circuitos digitais, por possuírem finitos possíveis estados, apresentam uma definição clara da distinção entre um circuito falho e outro funcional, enquanto que os circuitos analógicos não possuem esse limite bem definido. Sendo assim, os testes com circuitos analógicos determinam, na verdade, o quão funcional o circuito se apresenta, onde a distinção entre circuitos falhos ou funcionais é feita através de limites pré-estabelecidos (KABISATPATHY; BARUA; SINHA, 2005).

Dessa maneira, um importante passo no teste analógico corresponde à determinação do limite que distinguirá um comportamento aceitável de um não aceitável para o circuito sob teste. Grande parte dos testes de circuitos integrados analógicos acontece de acordo com a metodologia de teste baseado em especificações, na qual os limites de aceitação são as próprias especificações do *datasheet* (STRATIGOPOULOS; SUNTER, 2014). Essa metodologia, além de requerer um grande conhecimento prévio do circuito, faz com que os tempos de execução dos testes sejam longos, o que, por consequência, torna-os consideravelmente caros se comparados ao teste de circuitos digitais que ocupem a mesma área no chip (POEHL; DEMMERLE; ALT; OBERMEIR, 2010). Circuitos analógicos de RF (Rádio Frequência), por exemplo, apresentam

uma grande quantidade de especificações a serem verificadas o que pode requerer um ATE (*Automatic Test Equipment*) bastante caro com instrumentos analógicos de alta resolução, elevada capacidade de processamento de sinais e uma significativa quantidade de memória para guardar as respostas do teste (BUSHNEL; AGRAWAL, 2002).

Buscando uma alternativa mais barata do que o teste baseado em especificações, porém mantendo uma boa qualidade, a metodologia mais indicada é a de teste estrutural (RENOVELL, 2004; SACHDEV; DE GYVEZ, 2007). Segundo ela, é proposto um modelo de falhas para o circuito em análise, com o qual, através de simulações, levanta-se o dicionário de falhas. Nesse contexto, a determinação dos vetores de teste para circuitos analógicos é feita, em geral, por tentativa e erro, ainda não sendo totalmente automatizada como no caso dos circuitos digitais (KABISATPATHY; BARUA; SINHA, 2005). Mesmo com o teste de circuitos integrados analógicos tendo recebido bastante atenção nos últimos anos e tendo apresentando avanços, os desafios de conduzir esses testes de maneira automatizada continuam existindo (COYETTE; ESEN; DOBBELAERE; VANHOOREN; GIELEN, 2016). Sendo assim, sem uma ferramenta de automatização que viabilize a análise de um grande número de possíveis vetores de teste, a metodologia de tentativa e erro por vias manuais pode não ser efetiva na busca por vetores de teste otimizados.

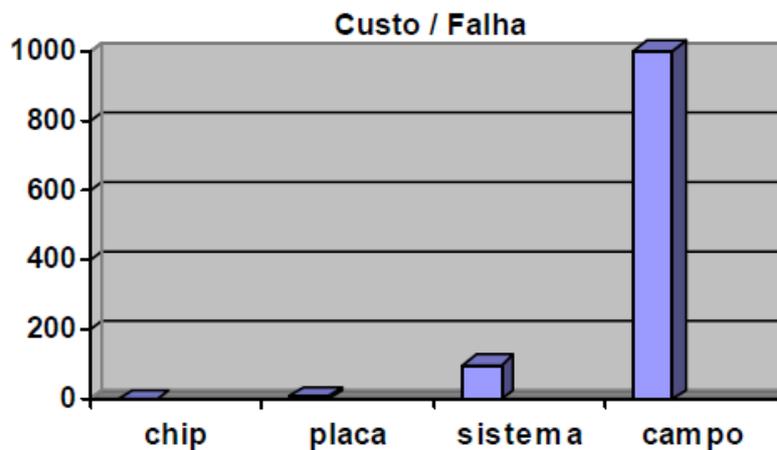
Para que a utilização de novas metodologias de teste que visam a redução do preço dos tradicionais testes baseados em especificações seja efetiva, é de fundamental importância avaliar a qualidade do teste, que está relacionada a dois parâmetros, a cobertura de falhas e o *yield loss* (STRATIGOPOULOS; SUNTER, 2014). Para um bom teste, a cobertura de falhas, ou seja, o percentual de falhas que podem ser detectadas, deve ser a maior possível, enquanto que o *yield loss*, correspondente ao percentual de circuitos “bons” que são incorretamente rejeitados pelo teste, deve ser o menor possível. No entanto, escolher, na prática, a metodologia de teste a ser adotada em um determinado circuito se torna uma questão complexa, haja vista a interdependência existente entre a qualidade do teste e o seu custo. Testes melhores tendem a ser mais caros, assim como testes mais baratos tendem a não apresentar a mesma qualidade.

2.1.1 Teste em Produção

No processo de fabricação de circuitos eletrônicos, é altamente desejável detectar falhas o quanto antes. Uma teoria bastante aceita na indústria, chamada de *rule of ten*, determina que o custo para reparo de uma falha é multiplicado por dez a cada etapa do processo de produção em que ela passa despercebida, assim como ilustra a Figura 1. Essa teoria é tão aceita que, inclusive, cogita-se a hipótese de alterá-la para *rule of twenty*, haja vista o enorme aumento de

complexidade dos chips e sistemas atuais em relação aos antigos, dos quais, de forma empírica, derivou-se essa teoria (DAVIS, 1982). Por isso, a primeira etapa de teste, conhecida em inglês como *wafer probe*, é feita sobre o próprio *wafer* de silício, antes mesmo do encapsulamento. Nesse teste, através da comparação entre o real desempenho e as especificações do circuito, são discriminados aqueles que possuem falhas, os quais são, em seguida, descartados (MILOR; VISVANATHAN, 1989).

Figura 1 - Estimativa de custo por falha detectada em cada etapa de produção.



Fonte: Balen (2006, p. 19)

Na etapa de produção, os testes são realizados por um equipamento conhecido por ATE, cujo preço ultrapassa facilmente a casa dos milhões de dólares (CHINAZZO, 2016). Dessa forma, o custo do teste de um CI é diretamente proporcional ao custo do equipamento e ao tempo gasto (*throughput*) (SOUDERS; STENBAKKEN, 1990), uma vez que o preço do ATE é amortizado entre todos os CIs nele testados, ou seja, quanto menor o tempo de teste, mais CIs poderão ser testados, e a parcela de custo atribuída a cada um será menor (BALEN, 2006). Como a cobertura de falhas possui uma relação direta com o tempo de execução, ela também influencia no custo. Em outras palavras, para atingir uma cobertura de falhas maior, é necessário utilizar um teste mais demorado, o que eleva os custos (ARSLAN; ORAILOGLU, 2013). Nesse contexto, buscando uma relação ótima entre o custo e a qualidade do teste, ressalta-se a necessidade de determinar testes otimizados, nos quais, dependendo da aplicação, seja alcançada a maior cobertura de falhas possível em um período de tempo pré-determinado, ou uma cobertura de falhas pré-determinada no menor período de tempo possível.

Na indústria, tanto o teste DC, quanto o teste AC (*Alternating Current*) são considerados confiáveis e efetivos para detecção de falhas em circuitos analógicos. Entretanto, quando se necessita uma maior cobertura de falhas, o teste mais indicado é o AC, pois esse é capaz de

transparecer mais informações sobre o circuito do que o teste DC para um número limitado de nós de teste (KABISATPATHY; BARUA; SINHA, 2005). Obviamente, para que seja válida essa superioridade do teste AC frente ao DC na cobertura de falhas, é necessário que a frequência do sinal de estímulo seja escolhida corretamente, para que se potencialize as diferenças de respostas dos circuitos com e sem falhas.

2.1.2 Teste Funcional *versus* Teste Estrutural

Em relação aos objetivos dos testes, é possível classificá-los em dois grupos: o dos testes funcionais e o dos testes estruturais. O primeiro deles, como o próprio nome já diz, busca a validação do funcionamento do circuito fabricado através de comparações com as especificações (teste orientado às especificações). Enquanto que os testes estruturais buscam a identificação de diferenças na própria estrutura do CUT (*Circuit Under Test*) em relação ao projeto (teste orientado aos defeitos).

Sendo assim, um circuito falha no teste funcional se algum parâmetro do seu funcionamento estiver diferente do que foi especificado em projeto. Por exemplo, um circuito de áudio falha no teste funcional se a potência do sinal de saída estiver abaixo dos limites especificados. Um problema para essa metodologia é que, para circuitos complexos, pode existir um número muito elevado de especificações, o que pode tornar esse teste demorado e, consequentemente, caro.

Por outro lado, o teste estrutural visa à detecção de defeitos. Esse método se baseia em modelar um grande número de falhas que possam vir a ocorrer no circuito e, então, utilizar essas informações para determinar o dicionário de falhas, que representa as informações sobre quais dessas falhas poderão ser detectadas em quais testes. Como empecilho à utilização dessa técnica, pode-se citar a necessidade de conhecer o modelo de falhas de maneira precisa (ARSLAN; ORAILOGLU, 2013).

A metodologia tradicionalmente utilizada pela indústria para teste de circuitos analógicos é de teste funcional. Todos os parâmetros são checados individualmente e comparados com os ranges especificados no *datasheet*. Nesse contexto, se qualquer um dos parâmetros estiver fora dos limites de tolerância, o circuito é considerado falho e, consequentemente, é descartado (SRIMANI; GHOSH; RAHAMAN, 2017). Assim como mencionado anteriormente, essa técnica requer tempos longos para a execução do teste, o que implica em elevados custos de execução, custos que por sua vez são adicionados ao preço final do produto.

Para tentar diminuir o custo dos testes funcionais, os próprios vetores utilizados para validação do circuito nas etapas de simulação do projeto podem ser utilizados no teste. Dessa forma, esses vetores já existem e não impõem nenhum custo adicional para sua geração. O que não é possível para o caso dos testes estruturais, cujos custos devem levar em consideração tanto a determinação dos vetores de teste, quanto a execução do teste propriamente dito (RENOVELL, 2004).

Segundo Renovell (2004), os testes estruturais se mostram soluções mais viáveis, sendo aplicados em situações onde se busca alta qualidade. Entretanto, o aumento da complexidade dos chips criou novos desafios a esta técnica.

2.1.3 Geração de Teste

A geração automática de padrões de teste, chamada de ATPG (*Automatic test-pattern generation*) consiste em um algoritmo que injeta falhas dentro do circuito (por simulação) e então usa uma variedade de mecanismos que ativem essas falhas, causando um efeito que se propague pelo circuito e se manifeste na saída do mesmo (BUSHNEL; AGRAWAL, 2002). Em circuitos digitais, por exemplo, o efeito de uma falha é propagado de uma entrada de uma porta AND/NAND para a sua saída apenas se a outra entrada estiver com nível lógico “1”. Isso porque, se o efeito da falha chegar a uma entrada de uma porta AND cuja outra entrada possui nível lógico “0”, a saída será determinada por esse sinal “0” e o efeito da falha não se propaga. De forma análoga, o efeito de uma falha só se propaga por uma porta OR/NOR se a outra entrada possuir nível lógico “0”, pois um nível lógico “1” já seria suficiente para deixar a saída em “1” independente do sinal da falha.

Sendo assim, a ATPG busca maneiras de fazer com que a saída do circuito apresente um valor diferente do esperado quando o circuito possuir falha. Nesse contexto, através da injeção de falhas, e posterior análise dos efeitos, o algoritmo procura determinar qual são os sinais de entrada que levam a essa manifestação da falha no sinal de saída. No caso dos circuitos digitais citados no exemplo do parágrafo anterior, o algoritmo busca quais são os sinais de entrada que imponham respectivos sinais de “1” e “0” nas portas AND/NAND e OR/NOR que estejam no caminho de propagação do efeito da falha. Em circuitos combinacionais mais complexos, contendo várias portas lógicas interconectadas, o algoritmo pode precisar encontrar mais de um vetor de entrada, pois nem todas as falhas terão os seus efeitos propagados até a saída pelo mesmo vetor de entrada.

Para o caso dos circuitos analógico, a geração de teste também busca sinais de entrada que façam o efeito da falha se propagar pelo circuito. Entretanto, nesse caso, uma manifestação

perceptível da falha no sinal de saída do circuito é caracterizada pela excursão desse sinal até um valor fora dos limites de tolerância. Além disso, existem outros aspectos que também podem indicar a existência de uma falha, por exemplo, o nível de ruído, fase, distorção harmônica, consumo de potência e impedâncias de entrada ou saída.

2.1.4 Modelo de Falhas Para Teste Analógico

Falha corresponde a um comportamento que cause a não conformidade de pelo menos uma especificação do *datasheet*, enquanto que um defeito é a diferença física indesejada entre o circuito fabricado e o projetado, podendo o defeito resultar ou não em uma falha (SUNTER; JURGA; DINGENEN; VANHOOREN, 2014). A modelagem de uma falha analógica corresponde a uma descrição matemática de como ela afeta o comportamento do circuito (FRACCAROLI; FUMMI, 2017). A determinação de um bom modelo de falhas, que atente para todos os possíveis mecanismos de falha de um processo de fabricação, é imprescindível para a determinação da cobertura de falhas, uma das grandezas mais importantes a serem levadas em consideração na escolha de metodologias de teste (STRATIGOPOULOS; SUNTER, 2014).

Para testes em circuitos digitais, existe um modelo chamado de *Stuck-at*, que se aplica a quase todos os circuitos. De acordo com esse modelo, é considerado que apenas um nó do circuito é falho, estando esse “preso” no nível lógico alto ou baixo, independente do sinal de entrada. Com isso, o número total de falhas em um circuito digital cresce linearmente com o número de nós, sendo igual ao dobro deste número (HUGHES, 1988).

Segundo Kondagunturi, Bradley, Maggard e Stroud (1999), as falhas em circuitos analógicos podem ser divididas em duas categorias: falhas catastróficas e paramétricas. Onde as falhas catastróficas, analogamente ao modelo *Stuck-at*, são curtos-circuitos (*Stuck-short*) ou circuitos abertos (*Stuck-open*). No primeiro caso, há o contato elétrico indesejado entre dois terminais de um componente, enquanto que, no segundo, há a perda de contato entre o terminal de um componente e o resto do circuito. Já as falhas paramétricas são desvios de parâmetros dos componentes que gerem um funcionamento fora dos limites especificados.

As falhas catastróficas são decorrentes de partículas indesejadas que se depositam sobre a máscara durante o processo de fabricação do CI (RENOVELL, 2004). Para simulação dessas falhas, vários autores propõem a utilização de resistores para modelar esses defeitos de circuito aberto e curto-circuito. Entretanto, os autores divergem quanto aos valores dessas resistências: Kondagunturi et al. (1999) sugerem que o circuito-aberto e o curto-circuito sejam modelados por resistências de $100\text{M}\Omega$ e de 1Ω , respectivamente; Deng, Shi e Zhang (2012) utilizam, em

suas análises, resistências de $1M\Omega$ e de 100Ω ; Petrashin, Dualibe, Lancioni e Toledo (2013) propõem que essas resistências sejam de $10M\Omega$ e de 10Ω .

Por outro lado, há um consenso entre alguns autores com relação à criação de modelos para as falhas paramétricas. Sugere-se que essas falhas sejam modeladas por variações nos parâmetros dos componentes do circuito que estejam entre três e seis desvios padrões em torno dos seus valores nominais (KONDAGUNTURI et al., 1999; DENG et al., 2012). Sendo assim, esses valores dependem da variabilidade da tecnologia utilizada.

2.1.5 Dicionário de Falhas

O dicionário de falhas para um circuito analógico traz informações referentes à detecção de falhas em determinados nós do circuito para determinados sinais de entrada. Sendo assim, o primeiro passo para sua definição é a escolha de um modelo de falhas adequado. Essa é a principal razão pela qual os circuitos analógicos estão muito atrasados em relação aos digitais na definição de metodologias para testes, pois, para os analógicos, ainda há uma carência por modelos de falhas mais adequados (KABISATPATHY; BARUA; SINHA, 2005).

A segunda etapa para criação do dicionário de falhas consiste em escolher quais nós do circuito serão possíveis nós de teste. Inicialmente, pode-se levar em consideração todos os nós do circuito, de modo a determinar quais são os melhores na detecção de falhas. No entanto, nem todos são acessíveis às medições dos testes, por isso, é dada preferência aos nós de saída, pois esses podem ser monitorados sem nenhuma alteração no circuito. Caso algum nó interno apresente resultados muito melhores na detecção de falhas se comparado ao de saída, pode se tornar interessante inserir, no circuito, uma estrutura adicional de teste que possibilite o monitoramento da tensão desse nó. Entretanto, a inserção dessa nova estrutura pode impactar no funcionamento do circuito, demandando análises adicionais, assim como feito por Bender (2015).

Por fim, para criação do dicionário de falhas, também é necessário determinar quais sinais de entrada serão considerados. Essa decisão parte de aspectos mais gerais, como, por exemplo, se o teste utilizará valores de tensão DC ou AC, e se estende até a definição de vetores de teste otimizados, ou seja, que maximizam a detecção de falhas nos nós de teste. Segundo Kabisatpathy et al. (2005), até então, a determinação de vetores de teste para circuitos analógicos não era feita de forma totalmente automatizada. Normalmente é utilizada a metodologia de tentativa e erro, assim como fez Bender (2015).

Uma vez definidos todos esses aspectos, a metodologia para determinação do dicionário de falhas consiste em simular o circuito sem falhas (*fault free*) para todos os sinais de entrada

estipulados, juntamente com os circuitos contendo cada uma das falhas modeladas. Posteriormente, deve-se analisar as tensões em cada nó de teste, de modo a levantar as informações sobre quais falhas são detectadas com cada um dos sinais de entrada.

Obviamente, é necessário definir limites a partir dos quais as respostas do CUT serão consideradas falhas. Nesse trabalho, para determinação do dicionário de falha, foram utilizados os códigos em MATLAB desenvolvidos por Chinazzo (2016), no quais é utilizada uma faixa de tolerância, de modo que, se a resposta do circuito estiver fora desse limite, o mesmo é considerado falho. Como o próprio autor justifica, essa proposta foi derivada do conceito de teste *go/no-go* (MILOR; VISVANATHAN, 1987). Com isso, a chamada matriz de falhas, na qual ficam armazenadas as informações sobre o dicionário de falhas, possui apenas valores “1” ou “0”, representando, respectivamente, se a falha é ou não detectada naquela condição.

2.2 CARACTERIZAÇÃO DE VARIABILIDADES DE PROCESSO

Durante a etapa de projeto de circuitos eletrônicos, fazer simulações computacionais vem se tornando cada vez mais importante. Atualmente, tecnologias CMOS (*Complementary Metal Oxide Semiconductor*) avançadas trabalham com dimensões de transistores tão pequenas que controlar o processo de produção vem se tornando cada vez mais difícil (QUIAN, 2015). Nesse contexto, para garantir o correto funcionamento dos circuitos fabricados, bem como evitar designs desnecessariamente pessimistas, é importantíssimo investigar o comportamento da variabilidade do processo (RAPPITSCH; SEEBACHER; KOCHER; STADLOBER, 2004; MEZZOMO; BAJOLET; CATHIGNOL; DI FRENZA; GHIBAUDO et al, 2011).

Existem diversos fatores que influenciam na variabilidade dos circuitos fabricados, fazendo com que seja impossível fabricar dois circuitos exatamente iguais quanto aos seus comportamentos elétricos. O primeiro fator é a granularidade da matéria que implica em flutuações aleatórias na quantidade de dopantes nos canais dos transistores e causa, por exemplo, variações na tensão de *threshold* que, por sua vez, alteram as correntes de fuga no canal e a potência dissipada (ROGENMOSER; CLARK, 2013). Outra fonte de variabilidade é o processo de fotolitografia. A intensidade de luz pode sofrer variações, principalmente nas bordas dos padrões impressos, devido à rugosidade dos materiais ou até mesmo pela quantização da luz (SHIN, 2016). Também pode ser citado o efeito de proximidade segundo o qual os íons colidem nas bordas da foto resina e são aleatoriamente refletidos para outras regiões da área ativa gerando uma dopagem extra nas proximidades das bordas (LI; WU, 2016). Por fim, existem as variabilidades causadas por aspectos ambientais, por exemplo, temperatura, tensão de alimentação, radiações, etc.

Essa variabilidade pode ser caracterizada através de testes aplicados a estruturas específicas que possibilitam a medição individual de amostras de transistores, cada um com conexões acessíveis em seus quatro terminais (CHEN; LIM; TEH; OMAR; LER; WATT, 2015). Nesse contexto, é possível estimar o desvio padrão de cada parâmetro do transistor e, assim, determinar modelos estatísticos para representar as variabilidades impostas pelo próprio processo de produção nos circuitos finais.

Existem três maneiras mais comuns de efetuar medições individuais em transistores de modo a caracterizá-los. O método mais comum é o de pares casados (TUINHOUT, 2005), que consiste em estruturas de dois transistores com o mesmo *layout*, onde as variabilidades podem ser calculadas pelo desvio padrão dos descasamentos existentes entre parâmetros de transistores de um mesmo par (CHEN; LI; LIM; TEH; OMAR; LER; WATT, 2015). O segundo método corresponde ao *simple array*, segundo o qual, um *array* de transistores é construído através da conexão de cada terminal de transistor no mesmo terminal dos outros, resultando em um circuito com apenas quatro terminais compartilhados por um grande número de transistores. E o último método corresponde à estrutura de *array* endereçado, onde é construído um *array* de transistores conectados a um *data bus* e um circuito decodificador que é responsável por selecionar, em cada medida, qual é o componente que está sendo medido através do *data bus* (MIZUTANI; KUMAR; HIRAMOTO, 2011).

O maior desafio dentro da caracterização de um circuito consiste em como utilizar as informações de caracterizações individuais de componentes para então determinar o impacto dessas variações no circuito como um todo. Para esse fim, as metodologias mais comuns são as análises de *corners* e Monte Carlo.

2.2.1 Análise de *Corners*

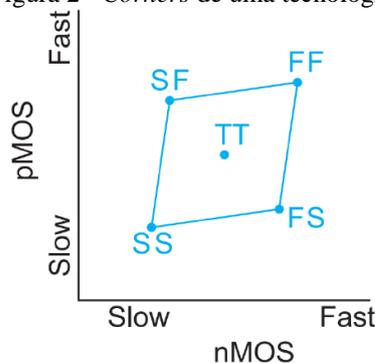
Uma das maneiras de modelar o impacto das variabilidades de processo nos circuitos é através da análise de *corners*. Segundo esse método, são considerados apenas quatro limites para variações toleráveis nos parâmetros dos transistores, os chamados “piores casos”. Esses extremos são: o pior caso no quesito velocidade, o pior caso no quesito consumo energético, o pior caso um e o pior caso zero.

Quando se fala em pior caso no quesito velocidade, tanto os transistores NMOS (*N-channel Metal Oxide Semiconductor*), quanto os PMOS (*P-channel Metal Oxide Semiconductor*) assumem parâmetros que os fazem ser considerados “lentos”. De forma análoga, quando se fala em consumo energético, o pior caso acontece quando os transistores são ditos “rápidos”. No pior caso um, os transistores NMOS e PMOS são, respectivamente,

“rápidos” e “lentos”, ao mesmo tempo em que, no pior caso zero, acontece o inverso, onde os NMOS são “lentos” e os PMOS são “rápidos”. Dessa forma, existem quatro *corners*, os quais são representados pelas siglas: SS (*Slow-Slow*); SF (*Slow-Fast*); FS (*Fast-Slow*); e FF (*Fast-Fast*), onde a primeira letra representa a condição de “rápido” ou “lento” aplicada ao transistor NMOS, e a segunda, ao PMOS (RAPPITSCH et al., 2004).

Na Figura 2, são mostrados os quatro *corners* de um processo de fabricação fictício. Nela, é possível observar linhas que interligam esses *corners* e representam os limites de tolerância para os parâmetros dos componentes. No centro da área delimitada por essas linhas, tem-se a condição típica, representada pela sigla TT (*Typical-Typical*).

Figura 2 - *Corners* de uma tecnologia.



Fonte: Chinazzo (2016, p.24) adaptado de Weste e Harris (2010, p. 245).

Analisar a variabilidade de um circuito integrado através da metodologia de *corners* traz vantagens quanto ao custo computacional. A determinação de limites toleráveis nas respostas dos circuitos fabricados, segundo essa técnica, requer menos simulações se comparada à técnica de Monte Carlo. Contudo, a análise de *corners* aplicada a circuitos analógicos normalmente leva a previsões excessivamente otimistas ou pessimistas quanto às variações dos circuitos (BENSCHWARTZ; SAKTHIVEL, 2016). Isso porque, esse tipo de análise foi criada mais direcionada a circuitos digitais. Sendo assim, para avaliar circuitos que envolvam um elevado número de componentes, a técnica de Monte Carlo se mostra mais eficiente (RUBINSTEIN; 1981), pois se baseia no conceito de aplicar variações individuais nos dispositivos, conforme métricas de incerteza caracterizadas para aquele processo de fabricação específico, considerando, inclusive, as variações de outros componentes além dos transistores.

2.2.2 Análise de Monte Carlo

O método de Monte Carlo para avaliação de variabilidades consiste em sucessivas simulações de amostras do circuito, onde cada uma delas é composta por componentes que

possuem parâmetros escolhidos aleatoriamente de acordo com os seus comportamentos estatísticos já caracterizados de maneira individual (MASON; HILL; MÖNCH; ROSE; JEFFERSON; FOWLER, 2008). Em outras palavras, esse método simula a fabricação de amostras do circuito, cada uma delas apresentando variações nos parâmetros individuais de componentes de acordo com a caracterização prévia, o que possibilita, através da simulação de um número suficientemente grande dessas amostras do circuito, caracterizar a variação geral imposta pelo processo de fabricação no circuito como um todo.

A análise de Monte Carlo pode levar a caracterizações mais realistas se comparada à análise de *corners*, pois representa uma metodologia mais completa que leva em consideração a interação das variações individuais dos componentes e não apenas os seus limites. Entretanto, o custo computacional de sua implementação se mostra muito mais elevado, pois requer muito mais do que quatro simulações, como é o caso da análise de *corners*.

2.3 METODOLOGIAS DE TESTE UTILIZADAS NESSE TRABALHO

2.3.1 Teste Transiente

O método de teste baseado em análise de resposta transiente (TRAM: *Transient Response Analysis Method*) (CALVANO; ALVES; LUBASZEWSKI, 1999, 2000) foi proposto como uma alternativa para teste de circuitos analógicos, em especial, filtros. De acordo com esse método, o circuito é dividido em blocos de primeira e segunda ordem que então são submetidos a um estímulo transiente de degrau (*step*), parábola ou rampa e, por fim, têm as suas respostas transientes analisadas. Os parâmetros analisados nesse trabalho foram o tempo de acomodação e o *overshoot*, que foram utilizados na detecção de falhas através de comparações simples entre os valores esperados e os obtidos para o CUT.

2.3.2 Teste DC

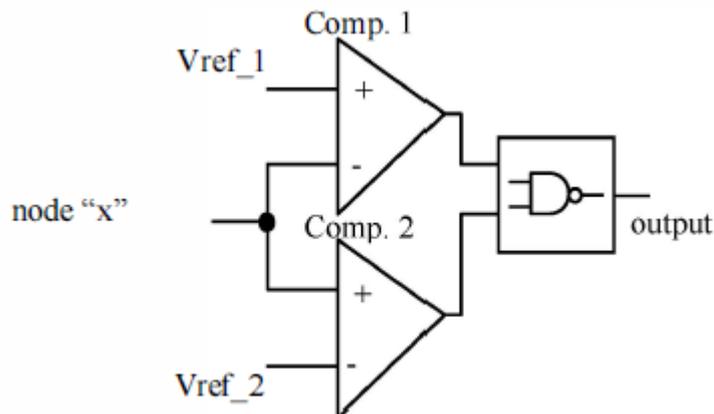
Uma vez que exista relações entre as tensões nodais do circuito e a presença de falhas, a técnica de teste DC corresponde a avaliar as tensões DC dos nós do circuito em condições com e sem injeção de falhas, para então estabelecer limites de tolerância que servirão para classificar as tensões nodais do CUT como aceitáveis ou não, fazendo o mesmo passar ou não no teste, respectivamente.

Diversos trabalhos utilizaram essa técnica para avaliar condições de teste, por exemplo, Petrashin, Dualibe, Lancioni e Toledo (2013) que aplicaram em um amplificador de transcondutância analógico, projetado em uma tecnologia de 65nm. Segundo o autor, a

metodologia de teste DC não é cara se comparada às outras técnicas, ao mesmo tempo em que é particularmente indicada para a detecção de falhas catastróficas. Outro exemplo corresponde ao trabalho de Bender (2015), onde a autora injetou falhas em amplificadores diferenciais projetados na tecnologia de 500nm e também obteve boas coberturas de falhas.

Além da técnica de teste DC apresentar baixo custo, a simplicidade que uma medida de tensão DC apresenta facilita a implementação, também a baixo custo, de estratégias de BIST. Na Figura 3, é mostrado o verificador proposto por Petrashin et al. (2013), onde dois comparadores e uma porta lógica são capazes de executar o teste de uma tensão do circuito, aprovando-o ou não. As tensões de referência (V_{ref_1} e V_{ref_2}) correspondem aos limites de aceitação pré-determinados e o nó sob teste corresponde ao nó “x”. Caso a tensão do nó sob teste esteja dentro da faixa de aceitação, as saídas dos dois comparadores estarão em nível lógico alto, única condição que faz com que a saída da porta lógica seja nível lógico baixo, indicando que o circuito passou no teste. Qualquer tensão nodal fora dos limites de aceitação faz um dos comparadores assumir nível lógico baixo e impõe um nível lógico alto na saída da porta lógica, indicando que o circuito possui falha.

Figura 3 - Diagrama em blocos do verificador proposto.



Fonte: Petrashin et al. (2013).

2.3.3 Teste AC

A metodologia de teste AC consiste em analisar as tensões nodais do circuito com relação aos seus espectros. Sendo assim, para uma certa excitação multitone na entrada, as tensões nodais do circuito apresentarão uma certa assinatura resultante dos diferentes ganhos apresentados para cada frequência. Essa análise espectral é feita normalmente através do algoritmo de transformada rápida de Fourier, pois esse apresenta maior eficiência com relação ao uso computacional (CHAUHAN; CHOI; ONABAJÓ; JUNG; KIM, 2014). Além do elevado

custo computacional que uma análise espectral impõe, essa metodologia também requer normalmente boas resoluções por parte do testador, de modo a alcançar uma precisão adequada.

Pensando em metodologias de teste AC mais simples, alguns trabalhos sugerem a utilização de sinais multitone compostos por harmônicos específicos que tornam a determinação do transformada rápida de Fourier mais simples (CHAUHAN et al, 2014) ou até mesmo a metodologia de teste serial, onde os sinais de entrada utilizados no teste são sinais com apenas uma frequência (CHINAZZO, 2016). Nesse último caso, a execução do teste se torna muito mais simples, apenas a amplitude de um sinal harmônico é o interesse de teste, sendo assim, um simples detector de pico poderia servir para executar o teste.

2.4 AMPLIFICADORES TOTALMENTE DIFERENCIAIS

Dentro de toda gama de circuitos analógicos, os circuitos amplificadores têm uma importância considerável. Eles geralmente representam a parte mais importante dos circuitos condicionadores de sinal, que são responsáveis, juntamente com conversores analógicos digitais e digitais analógicos, pelo interfaceamento de circuitos digitais com o mundo exterior que é, por natureza, analógico. Sendo assim, devido a esse papel de destaque que os circuitos amplificadores possuem, esses circuitos foram escolhidos para execução dos estudos de caso do presente trabalho, o que, por sua vez, justifica a existência desse capítulo específico.

Diferente de um DA (*Differential Amplifier*) com saída simples, um FDA (*Fully Differential Amplifier*) apresenta suas tensões de saída também na forma diferencial. Atualmente, a maior aplicação dos FDAs é no condicionamento de sinais para a entrada de ADCs (KARKI, 2002).

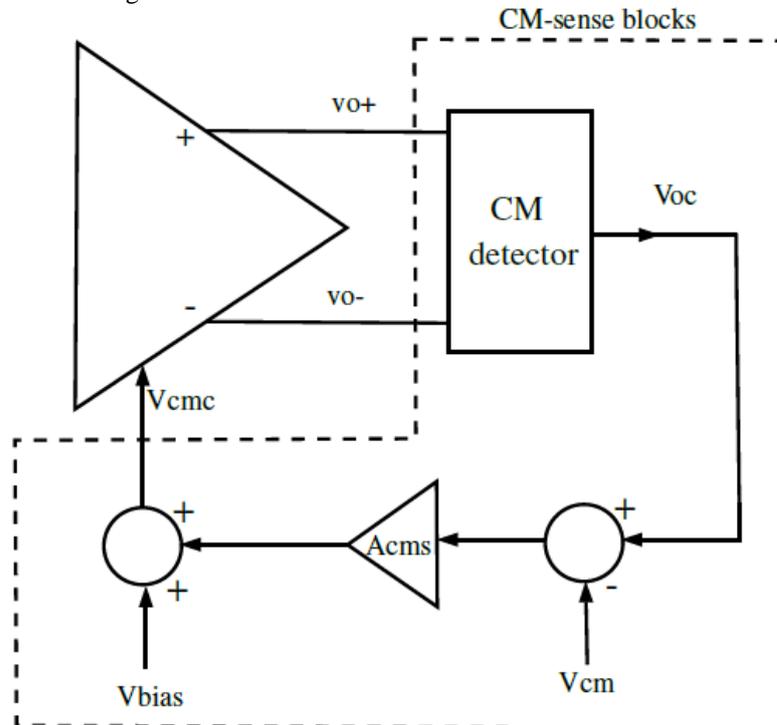
De acordo com Karki (2002), os FDAs apresentam diversas vantagens se comparados aos outros amplificadores. Em primeiro lugar, por rejeitarem a tensão de modo comum, são sistemas menos suscetíveis aos efeitos do ruído. Além disso, apresentam menores distorções harmônicas de ordem par em virtude da simetria dessas distorções nas saídas diferenciais. Por fim, devido à diferença de fase entre as duas saídas diferenciais, os FDAs também têm as suas excursões dinâmicas duplicadas.

Como desvantagem ao uso dos FDAs, pode-se citar a necessidade que esses amplificadores têm de possuírem um circuito adicional chamado de CMFB (*Common-Mode Feed-Back*). Esse circuito serve para estabilizar, em um valor desejado, a tensão de modo comum nas saídas. Sendo assim, na ausência ou mal funcionamento do CMFB, o FDA pode se tornar instável (OLIVEIRA; SEVERO; GIRARDI, 2014). O CMFB deve fornecer tensões de modo comum estáveis em toda a faixa de frequências em que o FDA opere e deve operar

independentemente do nível DC do sinal de entrada (XU; EMBABI, 2000; ALZAHER; ELWAN; ISMAIL, 2003).

Na Figura 4, é possível observar o bloco conceitual do circuito CMFB do FDA. Nele, as duas tensões de saída do amplificador (“vo+” e “vo-”) são lidas por um bloco detector de modo comum, cuja saída será “Voc”, correspondente à média entre “vo+” e “vo-”. “Vcm” corresponde à saída de modo comum desejada, ela é subtraída da tensão de modo comum do circuito e, então, amplificada pelo bloco “Acms”. Essa tensão amplificada é adicionada a “Vbias” e, em seguida, conectada à malha de polarização do OpAmp (*Operational Amplifier*).

Figura 4 - Bloco conceitual do circuito CMFB do FDA.



Fonte: Oliveira; Severo; Girardi (2014).

3 DESCRIÇÃO DA FERRAMENTA

Nesse capítulo, será descrito todo o funcionamento da toolbox criada em MATLAB R2015a para análise de configurações de teste em circuitos analógicos. A descrição vai desde aspectos de mais alto nível, como, por exemplo, quais dados o usuário precisa inserir, até aspectos de programação, como, por exemplo, a lógica utilizada na determinação dos resultados.

Buscando uma alternativa mais barata do que o teste baseado em especificações, porém mantendo uma boa qualidade, a metodologia escolhida para implementação da ferramenta foi a de teste estrutural (RENOVELL, 2004; SACHDEV; DE GYVEZ, 2007). Uma vez que a determinação dos vetores otimizados para execução desses testes em circuito analógicos ainda não é totalmente automatizada (KABISATPATHY; BARUA; SINHA, 2005), o intuito principal da criação dessa ferramenta foi, justamente, possibilitar essa automatização. Sendo assim, buscou-se a possibilidade de analisar um grande número de vetores de teste em um tempo extremamente menor que o da análise manual.

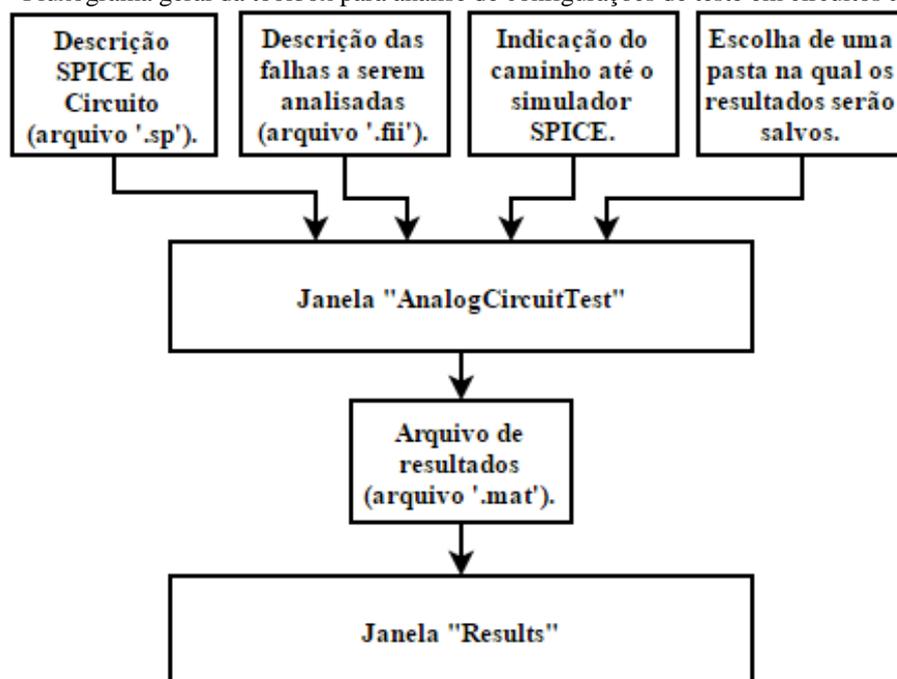
Para executar a análise de configurações de teste de um circuito analógico qualquer utilizando essa toolbox, é necessário dispor da sua descrição em linguagem SPICE (extensão ‘.sp’) e do arquivo com a descrição das falhas a serem analisadas (extensão ‘.fii’). Além disso, também é necessário dispor de um simulador SPICE – nesse trabalho, foi utilizado o HSPICE™ da empresa Synopsys.

Na definição da metodologia, foi considerado que a probabilidade de uma falha mascarar os efeitos de outra é muito baixa, o que torna possível a análise individual de cada falha. Sendo assim, seguindo a metodologia de teste estrutural, a ferramenta insere individualmente, na descrição SPICE do circuito sob análise, cada uma das falhas listadas no arquivo de descrição de falhas (extensão ‘.fii’). Dessa forma, são geradas, automaticamente, uma descrição SPICE do circuito para cada uma das falhas modeladas. Posteriormente, também de forma automatizada, todos esses circuitos com falhas, juntamente com o circuito *fault free*, são simulados pelo simulador SPICE. Os resultados deste processo de simulação de falhas são utilizados pela ferramenta para determinação do dicionário de falhas, nesse caso, sendo uma matriz de zeros e uns que carrega a informação de quais falhas podem ser detectadas, através das medições de quais tensões nodais do circuito e com quais sinais de entrada. Além disso, o algoritmo se encarrega de determinar quais são as melhores configurações de teste para esse circuito, de modo a maximizar a cobertura de falhas e otimizar o custo de execução. Em outras palavras, o resultado da aplicação da ferramenta é uma lista com a indicação de uma sequência de teste, contendo os tipos de estímulos a serem aplicados às entradas primárias, os nós a serem

observados e a cobertura de falhas esperada, segundo o modelo de falhas estrutural considerado. Ao levar em consideração também nós internos do circuito para a observação da resposta de teste a ferramenta pode ser utilizada para a decisão sobre a inclusão de barramentos de acesso, em tempo de projeto, para aplicação estratégias de DFT (*Design For Test*).

A execução completa da ferramenta requer a utilização de duas janelas, assim como mostra a Figura 5. A primeira janela de execução é chamada de “AnalogCircuitsTest” e é descrita em maiores detalhes na seção 3.1. Ela é responsável por criar e simular todos as descrições SPICE necessárias à análise, ler os arquivos resultantes dessas simulações e determinar a matriz de falhas. Por fim, essa janela de execução também se encarrega de salvar os resultados das simulações SPICE, juntamente com a matriz de falhas, em um arquivo no formato ‘.mat’ (específico do MATLAB). A segunda janela de execução é chamada de “Results” e é descrita pormenorizadamente na seção 3.2. Ela se encarrega de ler o arquivo ‘.mat’ de resultados que foi criado pela janela “AnalogCircuitsTest”, para, então, calcular a cobertura de falhas e determinar o conjunto de testes otimizados para o circuito, considerando a observação de diferentes nós. Essa janela também é responsável pela interface gráfica que possibilita a visualização das tensões nodais resultantes de todas as simulações SPICE (com e sem falhas), bem como permite a modificação de parâmetros que resultaram em novos conjuntos de testes otimizados.

Figura 5 - Fluxograma geral da toolbox para análise de configurações de teste em circuitos analógicos.



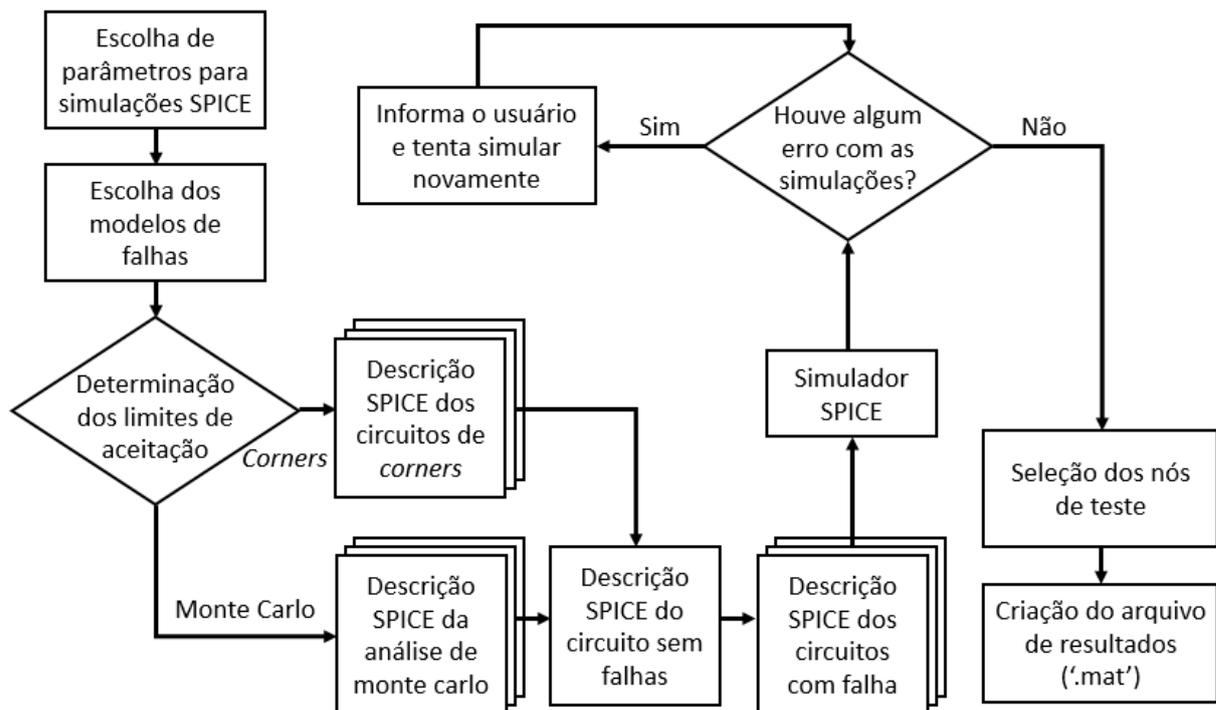
Fonte: elaborado pelo autor.

Durante a descrição do funcionamento da toolbox, serão mostradas imagens das suas janelas em execução. Para esses exemplos, meramente ilustrativos, foi utilizado o amplificador totalmente diferencial de um estágio, que será descrito em maiores detalhes na seção 4.2. Sendo assim, para essas figuras, contidas na seção 3.1 e na seção 3.2, o intuito principal não é demonstrar os resultados contidos nelas e, sim, ilustrar as funções que as janelas oferecem.

3.1 JANELA “ANALOGCIRCUITTEST”

A primeira janela que deve ser aberta para analisar as possíveis configurações de teste em um circuito frente às falhas selecionadas é a chamada “AnalogCircuitsTest”. Nas subseções 3.1.1 até 3.1.5, são descritas cada uma das etapas de execução dessa janela, enquanto o seu fluxograma geral de funcionamento é mostrado na Figura 6.

Figura 6 - Fluxograma de funcionamento da janela "AnalogCircuitsTest".



Fonte: elaborado pelo autor.

3.1.1 Etapa 1: Inserção de Arquivos para Simulação

Quando o comando “AnalogCircuitsTest” é chamado através da linha de comando do MATLAB, aparecerá a janela mostrada na Figura 7. Nela, é necessário a inserção de três caminhos para arquivos, sendo eles: o arquivo de descrição SPICE do circuito sem falhas (extensão ‘.sp’), o caminho para o simulador SPICE (arquivo com extensão ‘.bat’) e a pasta na qual os arquivos gerados devem ser salvos.

O arquivo com a descrição SPICE do circuito sem falhas (extensão ‘.sp’) deve conter a descrição do caminho até a biblioteca da tecnologia. Isso porque, na futura criação dos arquivos de descrição SPICE dos circuitos com falhas, será considerado esse mesmo caminho até a biblioteca. Então, se houver algum problema com essa descrição, além de haver problema na simulação do circuito sem falhas, também ocorrerá erro nas simulações de todos os arquivos com falhas gerados de forma automática pelo programa.

Figura 7 - Primeira etapa da Janela "AnalogCircuitsTest".

The screenshot shows a software window titled "AnalogCircuitsTest" with a subtitle "Fault Coverage Calculator" and a progress indicator "1 / 4". The window is organized into two main sections. The first section, "Input Files", contains two rows of input fields and buttons. The first row is for the "Netlist (.sp)" file, with a text box labeled "Enter Netlist File" and a "Find File" button. The second row is for the "Fault Description (.fi)" file, with a text box labeled "Enter Fault Description File" and a "Find File" button. The second section, "HSPICE Path and Folder to Save Files", also contains two rows. The first row is for the "HSPICE (.bat)" file, with a text box labeled "Enter The HSPICE.bat File" and a "Find File" button. The second row is for the "Folder to Save Simulation Files", with a text box labeled "Select a Folder To Save The Simulation Files" and a "Select Folder" button. At the bottom center of the window, there is a "Next" button.

Fonte: elaborado pelo autor.

Após efetuar a correta seleção de todos os arquivos e caminhos, deve-se clicar no botão “Next” para que o programa avance até a etapa 2, onde são inseridos os parâmetros de simulação.

3.1.2 Etapa 2: Parâmetros de Simulação

Nessa etapa, são inseridos os parâmetros que dizem respeito aos tipos de simulação SPICE que se deseja executar, podendo ser AC, DC ou TRAN. A Figura 8 mostra um exemplo dessa janela de execução para o caso de todos os três tipos de simulação terem sido escolhidos. Para as simulações DC, é necessário escolher o nó do circuito que será o terminal de entrada,

no qual serão aplicadas as tensões contínuas do teste. Além disso, os valores de tensão também precisarão ser especificados. No caso, será necessário inserir um valor inicial (*start*), final (*stop*) e o incremento entre cada um dos valores simulados (*step*). Já para as simulações AC, é necessário informar qual será a faixa de frequência da análise, juntamente com quantos pontos por década serão analisados. Para o caso da análise TRAN, deve-se informar os parâmetros temporais da análise. Nos casos AC e TRAN, a descrição de qual nó do circuito será o terminal de entrada deverá ser feita na própria descrição SPICE do circuito sem falhas, pois a sintaxe da linguagem SPICE não permite que essa informação seja inserida na linha onde se descreve o comando de controle (no caso, “.ac” e “.tran”), diferentemente do caso DC, que permite a inserção dessa informação. Além disso, para o caso TRAN, as definições referentes ao degrau de tensão na entrada também devem ser especificadas na descrição SPICE do circuito sem falhas.

Figura 8 - Inserção de parâmetros de simulação.

Fonte: elaborado pelo autor.

Outro parâmetro a ser escolhido corresponde à metodologia para determinação dos limites de aceitação. Para a toolbox, esses limites de aceitação correspondem diretamente a tensões nodais para o caso de teste DC; amplitudes de tensões nodais no caso AC; e tempo de acomodação e *overshoot* para o caso TRAN. Assim como comentado, a metodologia de teste

estrutural baseado em defeitos foi utilizada no desenvolvimento da ferramenta por ser uma alternativa para redução dos custos em relação aos testes baseados em especificações. Além disso, uma segunda vantagem é que os limites de detecção no teste baseado em defeitos podem ser determinados de maneira automática em função da variabilidade do processo. Diferentemente do teste baseado em especificações, no qual os limites de detecção dependem necessariamente de um conhecimento específico sobre o funcionamento do circuito e suas especificações.

Nesse contexto, as principais metodologias de determinação dos limites de aceitação que a ferramenta oferece são as análises de *corners* e Monte Carlo. De acordo com a metodologia de *corners*, o algoritmo simula automaticamente o circuito sem falhas nos quatro *corners* da tecnologia, de modo a determinar quais são os piores casos dentre essas respostas. Sendo assim, os valores mínimo e máximo de tensões nodais, que serão considerados como os limites de aceitação, ficam sendo as próprias respostas mínima e máxima entre os circuitos com parâmetros referentes aos *corners* da tecnologia. Entretanto, se a metodologia escolhida for a de Monte Carlo, o algoritmo realizará automaticamente simulações de Monte Carlo do circuito sem falhas (tantas quantas forem escolhidas pelo usuário da ferramenta). Para cada tensão nodal serão calculados a média e o desvio padrão das respostas, para que, por fim, os limites de aceitação sejam definidos como uma faixa de desvios padrões em torno da média, onde esse número de desvios padrões também é escolhido pelo usuário.

Assim como comentado anteriormente na seção 2.2, analisar a variabilidade de um circuito integrado através da metodologia de *corners* traz vantagens quanto ao custo computacional. A determinação de limites de tolerância nas respostas dos circuitos fabricados, segundo essa técnica, requer menos simulações se comparada à técnica de Monte Carlo. Entretanto, Monte Carlo pode ser mais eficiente em descrever as variabilidades do processo, pois aplica variações individuais nos dispositivos, conforme métricas de incerteza caracterizadas para o processo de fabricação, e assim determina o impacto dessas incertezas no funcionamento do circuito como um todo.

Um detalhe adicional na determinação do limite de aceitação consiste na diferença mínima que duas tensões devem ter para que possam ser distinguidas corretamente pelo testador. Isso porque, a diferença entre o limite de aceitação e um comportamento aceitável qualquer não pode ser menor do que isso, de modo a evitar *yield loss*. Essa diferença mínima de tensão é basicamente imposta pela precisão do equipamento de teste e do nível de ruído. Para levar essa questão em consideração, o algoritmo permite a escolha desse valor mínimo de

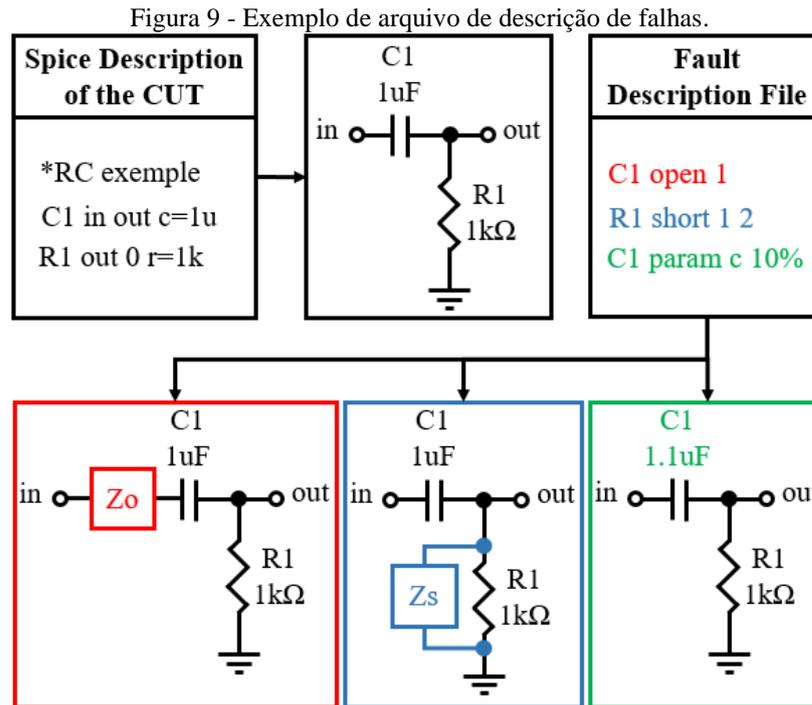
tensão capaz de ser medido pelo testador e então utiliza esse valor como diferença mínima que deve existir entre o comportamento sem falhas e os limites de tolerância.

Ao final dessa etapa, com todos os parâmetros de simulação devidamente escolhidos, deve-se clicar novamente no botão “Next”. Então, o programa seguirá para a etapa 3, referente à escolha do modelo de falhas.

3.1.3 Etapa 3: Escolha das Falhas que Irão Compôr o Modelo.

A sintaxe do arquivo de descrição de falhas (extensão ‘.fii’) foi estabelecida por Chinazzo (2016), e contempla a ocorrência de falhas catastróficas (em terminais de um mesmo componente) e paramétricas. Nesse arquivo, para descrever uma falha de circuito aberto, é escrito o nome do componente em questão (utilizado na descrição SPICE), a palavra “*open*” e o índice do terminal do componente no qual se deseja inserir a falha. Esse índice se refere à ordem com que os nós do componente são descritos em SPICE, ou seja, um índice “1” representa o primeiro nó descrito, que, se tratando de um transistor, seria o terminal de dreno. Analogamente, para descrever uma falha de curto-circuito, é escrito o nome do componente, a palavra “*short*” e os dois índices dos terminais que serão curto-circuitados. Para descrever uma falha paramétrica, é necessário escrever o nome do componente, a palavra “*param*”, o parâmetro a ser alterado e o seu desvio. Segundo o autor, a sintaxe foi escolhida de modo a possibilitar uma fácil compreensão das falhas descritas.

Na Figura 9, é mostrado um exemplo de arquivo de descrição de falhas, onde o CUT corresponde a um filtro RC (Resistivo Capacitivo) passa-baixas de primeira ordem. Nesse exemplo, são descritas três falhas: circuito aberto do terminal 1 do capacitor, curto-circuito entre os dois terminais do resistor e uma variação de 10% no valor da capacitância. Nessa mesma figura, são mostradas ilustrações de como ficam os circuitos com falha criados automaticamente pela ferramenta, onde se nota que as falhas catastróficas são descritas como caixas pretas contendo uma certa impedância interna. Isso porque, de fato, a ferramenta descreve curtos-circuitos e circuitos abertos na descrição SPICE como sub circuitos cujas impedâncias internas são escolhidas na hora da simulação de acordo com o modelo de falhas adotado para aquele caso específico, haja vista a variação que essa impedância pode apresentar em cada um dos processos de fabricação, incluindo a possibilidade de simulação de defeitos capacitivos, por exemplo. Sendo assim, cabe ao usuário da ferramenta escolher o melhor modelo de falhas a ser adotado e a ferramenta se encarregará de considerá-lo durante as simulações.



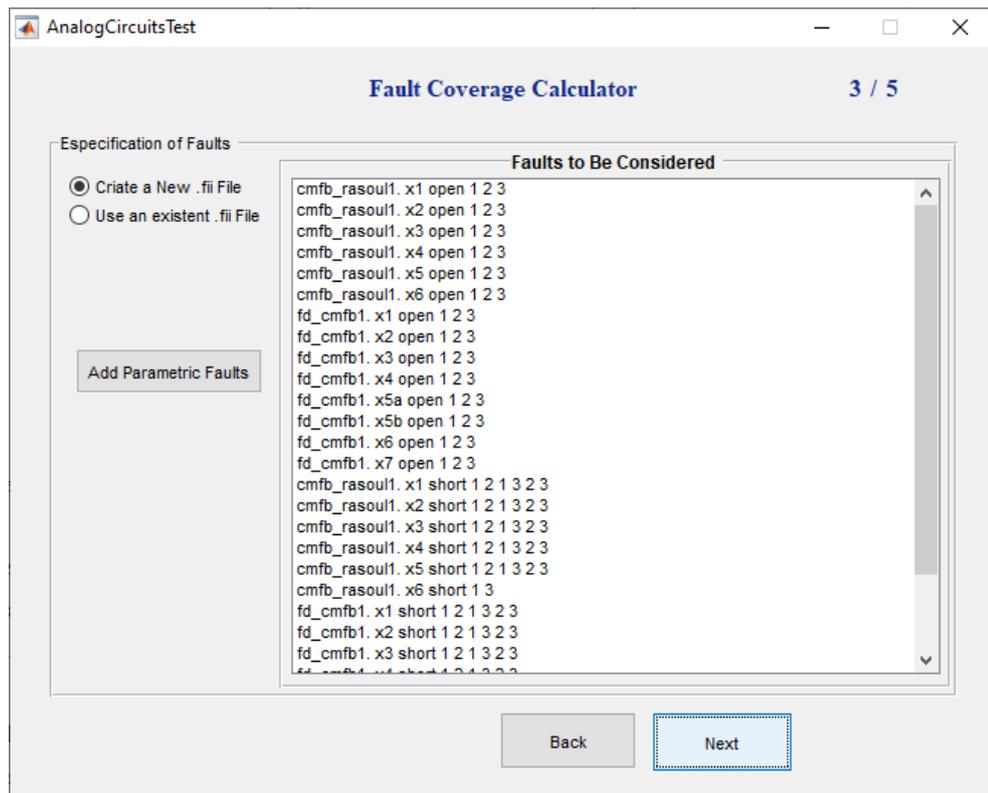
Fonte: elaborado pelo autor.

Visando a praticidade e um maior grau de automatização do processo, a ferramenta possibilita a criação automática do arquivo de descrição de falhas. Nesse caso, o algoritmo se encarrega de ler todos os componentes que compõem o circuito *fault free* (inclusive componentes internos de possíveis sub circuitos) para então adicionar no arquivo de descrição de falhas, todas as falhas catastróficas que são possíveis de acontecer nos terminais de cada um dos componentes analisados separadamente. Ou seja, o algoritmo cria esse arquivo de descrição de falhas considerando a possibilidade de falha de circuito aberto em todos os terminais de todos os componentes, bem como falhas de curto-circuito contemplando todas as combinações possíveis de interconexão de terminais de um mesmo componente, excluindo do modelo as falhas de curto-circuito que interconectam terminais que já estão interconectados pela própria topologia do circuito sob análise.

Na Figura 10, é mostrada a janela na qual é feita a escolha do arquivo de descrição de falhas, que pode ser a inserção de um caminho até um arquivo pré-existente, ou então deixar a própria ferramenta criá-lo. Nesse último caso, a ferramenta insere previamente todas as possíveis falhas catastróficas, possibilitando, ao usuário, editar essa seleção adicionando ou retirando falhas do modelo através da caixa de texto editável mostrada também na Figura 10. Um botão para inserção de falhas paramétricas também pode ser visto nessa mesma figura, e auxilia na inserção desse tipo de falha. Uma vez que a magnitude dos desvios paramétricos, em modelos de falhas analógicos, pode ser considerada a mesma para um dado tipo de componente,

esse botão possibilita a seleção do tipo de componente e a magnitude dos desvios a serem simulados. Havendo essas informações, a ferramenta se encarrega de escrever automaticamente no arquivo de extensão ‘.fii’ falhas paramétricas dessa magnitude em todos os componentes desse referido tipo.

Figura 10 - Escolha do arquivo de descrição de falhas.

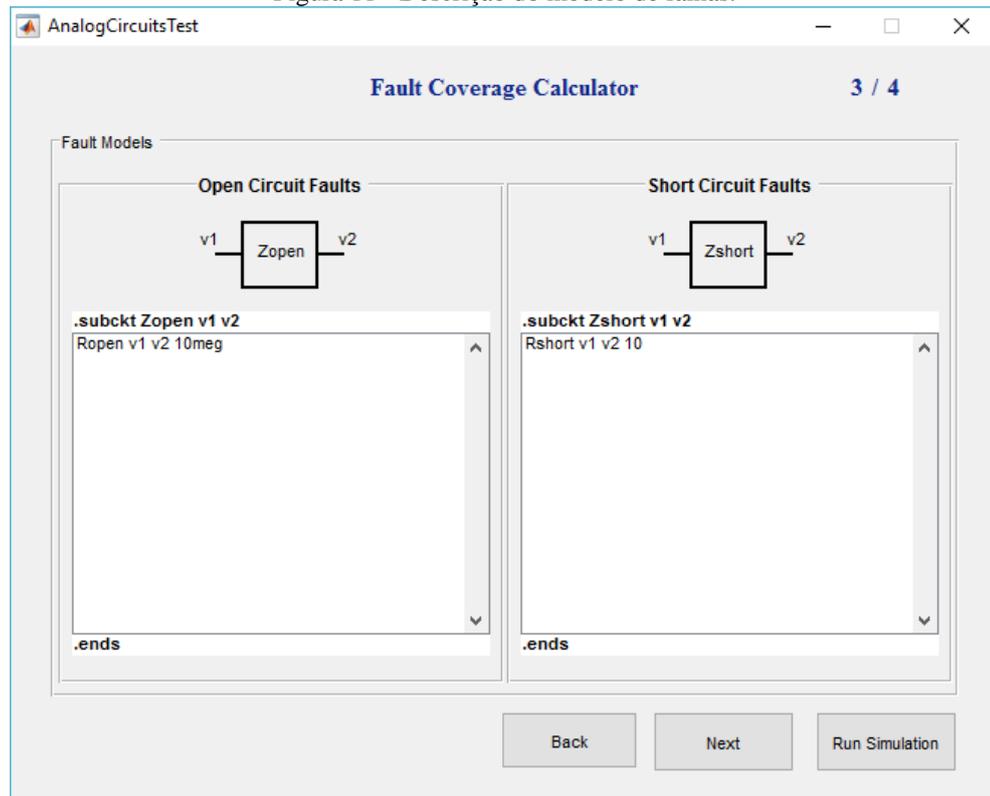


Fonte: elaborado pelo autor.

3.1.4 Etapa 4: Escolha do Modelo Elétrico para as Falhas Catastróficas.

Nessa etapa, são escolhidos os modelos elétricos que representarão as falhas de circuito aberto e de curto-circuito nas simulações. Para possibilitar essa escolha, assim como mostra a Figura 11, o programa disponibiliza duas caixas de texto para que sejam escritos esses dois modelos em linguagem SPICE. Inicialmente, o programa pré-determina esses modelos como resistências com valores de $10M\Omega$, para o caso do circuito aberto, e 10Ω , para o caso de curto-circuito, que são os valores utilizados por Petrashin et al. (2013).

Figura 11 - Descrição do modelo de falhas.



Fonte: elaborado pelo autor.

Depois de finalizar a descrição dos modelos de falhas, deve-se clicar no botão “*Run Simulation*”. Nesse momento, o programa criará um novo arquivo SPICE para representar o circuito sem falhas. Esse arquivo será muito semelhante ao arquivo de entrada inserido na etapa 1 (seção 3.1.1). A única diferença será a linha onde são descritos os parâmetros de simulação, que, nesse caso, corresponderão ao que foi escolhido pelo usuário na etapa 2 (seção 3.1.2).

Posteriormente, o programa roda os códigos desenvolvidos inicialmente por Chinazzo (2016), porém modificados no presente trabalho, para gerar e simular todos os circuitos com falhas. Além disso, nessa mesma etapa, são gerados e simulados os arquivos que servirão de base para a determinação dos limites de aceitação, que podem corresponder ao circuito *fault free* nas condições de *corners* da tecnologia ou simulações Monte Carlo desse mesmo circuito, dependendo do que foi escolhido na etapa 2 (seção 3.1.2). Para criar os arquivos de *corners*, o algoritmo altera a linha onde está descrita a biblioteca de modelos da tecnologia de fabricação, colocando a sigla do *corner* em questão. Já para o caso da simulação de Monte Carlo, o algoritmo insere, na descrição SPICE, todos os parâmetros necessários à esta análise.

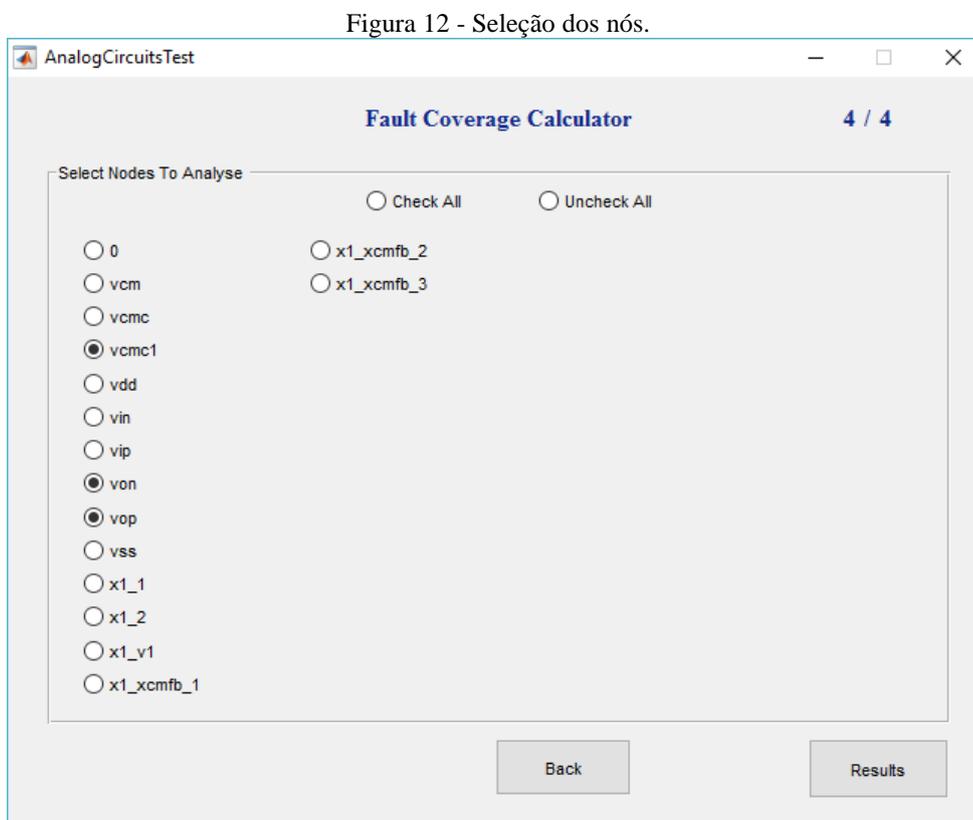
Segundo Chinazzo (2016), como as chamadas ao simulador são externas ao ambiente MATLAB, via sistema operacional, é possível que algumas não funcionem corretamente, fazendo com que alguns circuitos não sejam simulados. Por isso, quando o programa termina

de rodar as simulações SPICE, é feita a verificação de todos os arquivos gerados para identificação de possíveis erros. Caso tenha ocorrido algum problema, o programa informa ao usuário e roda novamente as simulações que falharam. Dessa forma, garante-se que nenhum circuito deixará de ser simulado por problemas nas chamadas do simulador via sistema operacional.

Ao alcançar o êxito em todas as simulações, a janela “AnalogCircuitsTest” segue para a etapa de escolha dos nós que serão possíveis pontos de teste para o circuito e que, por isso, terão as suas coberturas de falhas calculadas posteriormente pela janela “Results”. Sendo essa etapa de escolha dos nós descrita na seção 3.1.5.

3.1.5 Etapa 5: Determinação do Dicionário de Falhas e Resultados.

Na quinta e última etapa da janela “AnalogCircuitsTest”, utilizando a “HSPICE Toolbox for MATLAB” (PERROTT, 2011), o programa lê, no arquivo de resultado do circuito *fault free*, os nomes de todos os nós do circuito sob análise, trazendo-os para o ambiente MATLAB. Com isso, fica disponível, ao usuário, selecionar quais destes nós serão pontos de teste do circuito, assim como no exemplo ilustrativo da Figura 12.



Fonte: elaborado pelo autor.

Quando o botão “Results” é pressionado, são utilizados novamente os códigos modificados de Chinazzo (2016) e Perrott (2011) para ler os arquivos resultantes das simulações SPICE e importar, para o ambiente MATLAB, todos esses resultados. Já no ambiente MATLAB, são determinados os limites de aceitação para cada uma das tensões nodais através dos resultados das simulações de *corners* ou Monte Carlo, dependendo de qual dessas metodologias tenha sido escolhida na etapa 2 (seção 3.1.2). De posse desses limites, o algoritmo se encarrega de analisar sequencialmente cada uma das respostas de circuitos com falha, analisando, para cada sinal de entrada simulado, quais são as tensões nodais que estarão dentro ou fora desses limites. Dessa maneira, o programa constrói uma matriz de falhas, a qual carrega as informações de quais configurações de teste¹ serão capazes de detectar cada uma das falhas, em outras palavras, quais configurações de teste fazem os circuitos com falhas apresentarem um comportamento fora dos limites de aceitação.

A matriz de falhas é importantíssima para os cálculos executados pela janela “Results”, descrita na seção 3.2, por isso é necessário explicar a sua estrutura em maiores detalhes. Essa matriz possui 3 dimensões. Os índices da primeira delas correspondem a cada um dos sinais de entrada simulados. Os índices da segunda são relacionados a cada um dos nós do circuito, considerando todos eles, ou seja, tanto dos de saída, quanto nós internos. Por fim, a última dimensão possui índices correspondentes a cada uma das falhas modeladas. Dessa maneira, cada posição da matriz é preenchida com uma informação binária que explicita se a falha correspondente ao índice em questão é ou não detectada por um teste que leve em consideração a configuração de teste referente a essa posição da matriz.

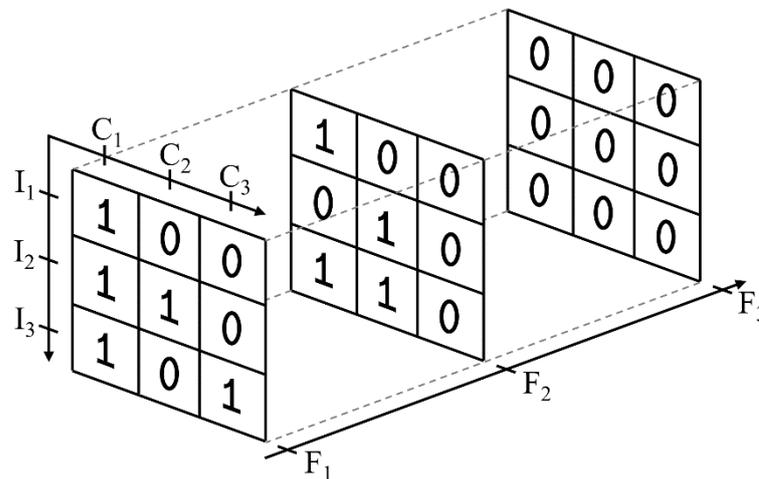
Definindo que o vetor $I = [I_1, I_2, \dots, I_S]$ seja aquele que contém todos os sinais de entrada simulados, ou seja, podendo ser composto por valores de tensão contínua no caso de análise DC, valores de frequências no caso da análise AC, ou ambos em uma análise mais ampla que considere análises no domínio tempo e no domínio frequência. Sendo assim “S” corresponde ao número total de pontos simulados em SPICE, o que depende da amplitude do passo adotado nas simulações. Além disso, pode-se definir os vetores $C = [C_1, C_2, \dots, C_N]$ e $F = [F_1, F_2, \dots, F_M]$ como sendo os vetores que denotam os nós do circuito e as falhas do modelo, respectivamente. O que implica em dizer que “N” corresponde ao número total de nós do circuito, bem como “M” representa o número total de falhas do modelo.

¹ Configuração de teste corresponde à combinação de nó (no qual será executada a medida de tensão) e sinal de entrada utilizados em um teste.

Chamando a matriz de falhas de MF , pode-se dizer que $MF(x, y, z) = 1$, significa que uma falha hipotética F_z é detectada através da medição do nó C_y quando a entrada do circuito estiver recebendo o sinal I_x . Analogamente, se $MF(x, y, z) = 0$, então essa mesma falha não é detectada através da medição desse nó do circuito com esse sinal de entrada, em outras palavras, esse nó C_y apresentará uma resposta dentro dos limites de aceitação quando o sinal de entrada for I_x , mesmo que a falha F_z esteja presente no circuito.

Na Figura 13, é ilustrado um exemplo de matriz de falhas que é resultante de uma simulação hipotética de um circuito contendo apenas três nós que levou em consideração somente três possíveis sinais de entrada e um modelo de falhas também contendo três falhas. Analisando a distribuição de valores “1” ao longo da matriz, fica fácil perceber quais falhas são detectadas em quais configurações de teste. A falha F_1 é detectada através da medição do nó C_1 para qualquer um dos três sinais de entrada, enquanto que a detecção da mesma falha através de medições em outros nós requer um sinal de entrada específico. A falha F_2 não é detectada através da medição do nó C_3 , porém essa mesma falha pode ser detectada em algumas configurações de teste relacionadas aos outros dois nós. Por fim, pode-se perceber também que a falha F_3 não é detectada por nenhuma configuração de teste analisada.

Figura 13 - Exemplo de matriz de falhas.



Fonte: elaborado pelo autor.

Após determinar a matriz de falhas, o algoritmo se encarrega de salvar, na pasta escolhida pelo usuário na etapa 1 (seção 3.1.1), um arquivo na extensão ‘.mat’ (específico do MATLAB), contendo todos os resultados obtidos até então. Esses resultados correspondem a todas as tensões nodais resultantes das simulações SPICE, juntamente com os limites de aceitação, a matriz de falhas e outras variáveis que dizem respeito aos parâmetros utilizados nas

simulações. Dessa forma, esse arquivo contém todas as informações necessárias para a análise dos resultados, determinação da cobertura de falhas e das melhores configurações de teste para o circuito em questão, operações que serão feitas por outra janela da toolbox, chamada de “Results”, que será descrita na seção 3.2.

Vale a pena ressaltar que, embora sejam selecionados nós para serem possíveis pontos de teste do circuito, o programa armazena no arquivo de resultados as tensões de todos os nós e a matriz de falhas também considerando todos os nós. A informação de quais são os possíveis nós de teste é armazenada como um vetor, no qual cada posição possui um valor referente ao índice de um nó escolhido. Com isso, a janela “Results”, que faz a análise dos resultados, leva em consideração apenas os nós que possuem os índices que constam nesse vetor, e, caso se deseje analisar outros nós, apenas esse vetor de índices é alterado, sem necessitar a releitura dos arquivos resultantes das simulações SPICE, operação que demandaria muito mais tempo de processamento.

Por fim, a janela “AnalogCircuitsTest” chama a execução da janela “Results” e, então, encerra sua execução automaticamente.

3.2 JANELA “RESULTS”

De uma maneira geral, a janela “Results” faz a leitura do arquivo ‘.mat’, que possui os resultados das simulações gerados pela janela “AnalogCircuitsTest” (descrita na seção 3.1), e determina quais são as melhores configurações de teste dentro das condições estabelecidas pelo usuário. Nela, são mostradas informações sobre os parâmetros utilizados nas simulações SPICE, a máxima cobertura de falhas que pode ser atingida dentre todas as condições simuladas, as máximas coberturas de falhas em cada nó e o conjunto de testes considerado como o melhor possível. Essa janela pode ser chamada de duas maneiras diferentes. A primeira delas ao final da execução da própria janela “AnalogCircuitsTest”, onde ela é chamada de forma automática, e o arquivo de resultados já é carregado automaticamente. A outra forma é através da linha de comando, mas, nesse caso, ao abri-la, ainda será necessário carregar o arquivo ‘.mat’ contendo os resultados que se deseja analisar.

Visando a melhor explicação do funcionamento da janela “Results”, foi escolhida a seguinte divisão dessa seção: a maneira como é determinada a máxima cobertura de falhas possível é descrita na seção 3.2.1; na seção 3.2.2, é descrito o método de escolha do melhor conjunto de testes; uma visão geral do visual da janela “Results” é mostrada na seção 3.2.3; na seção 3.2.4 são detalhados quais parâmetros são interativos, ou seja, que possibilitam alterações por parte do usuário, resultando em novos cálculos de coberturas de falhas e uma nova escolha

do melhor conjunto de teste; por fim, na seção 3.2.5 é descrita uma funcionalidade adicional da ferramenta que visa a otimização do conjunto final de teste através da análise da possibilidade de execução de um teste transiente entre dois testes DC pré-estabelecidos no conjunto.

3.2.1 Determinação da Máxima Cobertura de Falhas

A determinação da cobertura máxima de falhas se dá através da classificação prévia das falhas entre as detectáveis e as não detectáveis. Nessa classificação, uma falha é dita detectável se pelo menos uma configuração de teste, dentre as simuladas, é capaz de detectá-la, caso contrário, será dita não detectável.

Falando em uma linguagem mais lógica, o algoritmo faz essa classificação através da definição de um vetor chamada NCT , que corresponde ao número de configurações de teste que são capazes de detectar cada uma das falhas, ou seja, $NCT(F_z)$ corresponde ao número de configurações de teste que detectam a falha hipotética F_z . Essa variável é calculada através da matriz de falhas de acordo com a seguinte forma:

$$NCT(F_z) = \sum_{s=1}^S \sum_{n=1}^N MF(s, n, z) \quad (1)$$

onde “S” corresponde ao número total de pontos simulados no SPICE, o que depende do passo de simulação utilizado, “N” corresponde ao número total de nós do circuito; e MF é o valor binário da posição correspondente da matriz de falhas, como já mencionado.

Basicamente, o índice da terceira dimensão da matriz de falhas é fixado no índice da falha em questão, para que então NCT seja igual à soma de todos os valores da sub matriz resultante. Isso porque, como já dito anteriormente, a matriz de falhas é composta apenas por zeros e uns, onde o valor um está presente justamente nas posições referentes às configurações de teste que possibilitam a detecção da falha em questão. Dessa forma, a soma de todos os valores da sub matriz resultante corresponde ao número de configurações de teste que detectam essa falha específica.

Sendo assim, pode-se redefinir a classificação de falhas detectáveis ou não detectáveis como sendo aquelas que possuem $NCT \geq 1$ e aquelas que possuem $NCT = 0$, respectivamente. Com isso, a máxima cobertura de falhas corresponde ao número de posições do vetor NCT que possuem valores diferentes de zero, dividido pelo número total de falhas do modelo, uma vez que a cobertura de falhas (CF) seja uma grandeza percentual (BUSNELL; AGRAWAL, 2002).

$$CF = \frac{\text{Número de falhas detectáveis}}{\text{Número total de falhas do modelo}} \cdot 100\% \quad (2)$$

Além disso, vale a pena ressaltar que falhas que possuam $NCT > 0$, porém com valor muito baixo, podem ser consideradas difíceis de serem detectadas, pois são detectadas por poucas das configurações de teste simuladas ou até mesmo podem requerer uma configuração de teste específica, caso $NCT = 1$.

3.2.2 Determinação do Melhor Conjunto de Testes

A determinação do melhor conjunto de testes é feita de maneira sequencial, onde o algoritmo escolhe um a um todos os testes que irão compor o conjunto. O objetivo final é alcançar a maior cobertura de falhas possível com o menor número de testes, ou seja, escolher o menor conjunto de teste possível que seja capaz de detectar todas as falhas ditas detectáveis.

Quando se pensa em critérios de seleção para os testes que irão compor o conjunto, o mais simples deles corresponde ao número de falhas que cada candidato (teste) é capaz de detectar, uma vez que aqueles que detectam mais falhas, podem ser bons candidatos. Entretanto, esse não deve ser o principal critério, pois um cuidado especial deve ser adotado com as falhas detectáveis que possuam NCT muito baixo, pois elas, assim como dito na seção 3.2.1, podem restringir muito a escolha, ou até mesmo impor a necessidade de escolha de um teste específico.

Em um exemplo hipotético, pode-se considerar o caso em que uma falha tenha $NCT = 1$. Nesse caso, apenas uma das configurações de teste é capaz de detectar essa falha, o que impõem a necessidade de que esse teste esteja presente no conjunto. Esse teste pode não ser o que detecta mais falhas, embora, muito provavelmente, seja capaz de detectar mais do que essa falha específica. Sendo assim, uma vez que não há necessidade de que uma falha seja detectada por mais do que um teste do conjunto, saber previamente quais falhas esse teste específico detecta é primordial para a escolha dos outros, uma vez que a sua presença no conjunto é indispensável e influencia na escolha dos próximos, pois torna desnecessária a detecção das falhas que ele já detecte. Isso implica que o primeiro critério de escolha deve ser a condição de detecção da falha mais difícil de ser detectada, ou seja, aquela que possua o menor NCT não nulo do modelo.

Em diversas situações, existirão mais do que um teste que atenda a essa primeira condição, sendo assim, o algoritmo utiliza mais três critérios de seleção: número de novas falhas detectadas (considerando só aquelas que não são detectadas pelos testes já presentes no conjunto), número total de falhas detectadas e extensão da faixa de possíveis sinais de entrada

a ser utilizada no teste. Todos eles funcionam como critérios de desempate, uma vez que, dentro da hierarquia, o próximo critério de escolha só é acionado quando há pelo menos dois testes empatados nos critérios anteriores.

Para determinar o número de falhas que cada configuração de teste é capaz de detectar, o algoritmo define uma nova matriz, chamada *NFD* (número de falhas detectadas por cada configuração de teste). Uma vez que os dois primeiros índices da matriz de falhas (*MF*) correspondam aos sinais de entrada simulados e aos nós do circuito, respectivamente, basta fixá-los nos valores específicos de cada configuração de teste e somar os valores dessa sub matriz para obter o número de falhas que essa configuração de teste é capaz de detectar. Dessa maneira, o cálculo de uma posição da matriz *NFD* referente à configuração de teste hipotética constituída por um sinal de entrada I_x e a medição de um nó C_y é obtido da seguinte forma:

$$NFD(I_x, C_y) = \sum_{m=1}^M MF(x, y, m) \quad (3)$$

O valor *NFD* configura o terceiro critério de escolha. Para definir o segundo, que é apenas uma variação deste, onde são desconsideradas as falhas que já são detectadas pelos testes escolhidos previamente, o algoritmo define uma nova matriz de falhas, chamada *MF'*. Inicialmente, *MF'* é exatamente igual a *MF*, porém, após cada escolha de teste, o algoritmo zera todas as posições da matriz *MF'* referentes às falhas que são detectadas pelo conjunto definido até então. Assim, o segundo critério de escolha, chamado de *NFD'* pode ser calculado por:

$$NFD'(I_x, C_y) = \sum_{m=1}^M MF'(x, y, m) \quad (4)$$

Esta Equação (4) é semelhante à Equação (3), porém leva em consideração a variante da matriz de falhas *MF'* que possui, zeradas, todas as posições referentes a falhas já detectadas pelo conjunto de testes estabelecido até então.

O quarto e último critério de desempate, assim como mencionado, se refere à amplitude de sinais de entrada possíveis de serem utilizados no teste. Por se tratar de sinais analógicos, cada teste requer um sinal de entrada dentro de uma faixa de possíveis valores, diferentemente do caso dos circuitos digitais, onde cada teste requer um vetor de entrada específico. Sendo

assim, considerando que dois testes empatem nos três primeiros critérios de escolha, ou seja, ambos detectem a falha mais difícil, e exista a igualdade, tanto do número de novas falhas detectadas por eles, quanto do número de falhas totais, é desejável que seja escolhido aquele que possibilite uma maior amplitude de possíveis sinais de entrada em sua execução. Para avaliar esse critério, a ferramenta conta o número de *steps* de simulação do sinal de entrada que correspondam à detecção das mesmas falhas, ou seja, levam os mesmos resultados e, por isso, são considerados como um único candidato a teste. Se um teste contabilizar mais *steps* de simulação do que o outro, esse será preferido.

A Figura 14 mostra o algoritmo em pseudo-código utilizado para determinação sequencial do melhor conjunto de testes seguindo os quatro critérios de escolha já mencionados anteriormente.

Figura 14 - Representação do algoritmo para determinação do melhor conjunto de testes.

```

1 Conjunto de Testes =  $\emptyset$ ;
2 CF do conjunto = 0;
3  $MF' = MF$ ;
4 while CF do conjunto < CF máxima do
5   Melhor CF = 0;
6   Melhor Configuração de Teste =  $\emptyset$ ;
7   IPF = índice da falha com o pior NCT
   (baseado na  $MF'$ );
8   for n = 1 to N do
9     for s = 1 to S do
10      if  $MF'(s,n,IPF) = 1$  then
11        CF = NFD'(s,n);
12        if CF > Melhor CF then
13          Melhor CF = CF;
14          Melhor configuração de teste =
            ( $I_s, N_n$ );
15          Melhor CF total = NFD( $I_s, N_n$ );
16        else if CF == Melhor CF then
17          CF total = NFD(s,n);
18          if CF total > Melhor CF total then
19            Melhor CF = CF;
20            Melhor Configuração de
            Teste = ( $I_s, N_n$ );
21            Melhor CF total = CF total;
22          else if CF total == Melhor CF
            total then
23            Contabiliza os ranges de
            possíveis sinais de entrada;
24            if Range da configuração de
            teste ( $I_s, N_n$ ) > Range da
            melhor configuração de teste
            then
25              Melhor CF = CF;
26              Melhor configuração de
            teste = ( $I_s, N_n$ );
27              Melhor CF total = CF
            total;
28          Conjunto de Testes = Conjunto de Testes  $\cup$ 
            Melhor Configuração de Teste;
29          Atualiza  $MF'$  zerando todas as posições
            referentes as falhas detectadas pela melhor
            configuração de teste;
30          Atualiza CF do conjunto;

```

Fonte: elaborado pelo autor.

No início, não há nenhum teste na variável que armazena o conjunto. Sendo assim, a matriz MF' é exatamente igual a MF . Cada repetição do laço *while* é responsável pela adição de uma configuração de teste ao conjunto final e esse mesmo laço busca alcançar a máxima cobertura de falhas possível, ou seja, a sua execução só termina quando o conjunto de testes for capaz de detectar todas as falhas ditas detectáveis. Os dois laços de repetição do tipo *for* são responsáveis por varrer todas as condições de teste simuladas. Cada condição presente no interior desses laços representam um critério de escolha. Pode-se perceber que o primeiro deles

corresponde à detecção da falha mais difícil de ser detectada, sendo que, se uma configuração de teste não é capaz de detectar essa falha, ela será descartada logo nessa etapa. Entretanto, havendo o atendimento dessa condição, o programa calcula o NFD' dessa configuração de teste, pois esse valor representa o número de novas falhas que essa configuração é capaz de detectar. Comparando esse valor com o NFD' do teste que está sendo escolhido como melhor até então, o programa define se torna esse candidato a teste como melhor até então (caso seu NFD' seja maior), avalia outros critérios de escolha (caso haja igualdade entre os valores de NFD') ou se descarta esse teste em questão (caso seu NFD' seja menor que o candidato escolhido anteriormente). Havendo a igualdade e consequente necessidade de avaliação de outro critério de escolha, o algoritmo faz uma análise exatamente igual, porém baseada no valor NFD , que leva em consideração o número total de falhas detectadas, sem desconsiderar as que já são detectadas por testes que já foram escolhidos para compor o conjunto final. Por fim, como resultado de uma nova igualdade, há a avaliação da amplitude da faixa dos possíveis valores de entrada como último critério. Após varrer todas as configurações de teste e a execução do algoritmo sair dos laços *for*, o melhor teste para as condições em questão terá sido selecionado e, portanto, será adicionado ao conjunto dos melhores. Nesse momento, a próxima execução do laço *while* é preparada, através da atualização da matriz MF' , e é feito o cálculo da cobertura de falhas que o conjunto de testes já atingiu. Portanto, na próxima execução do laço *while*, haverá a mudanças nos dois primeiros critérios de seleção devido a uma nova falha ser considerada como a mais difícil de ser detectada e, também, a mudança nos valores de NFD'

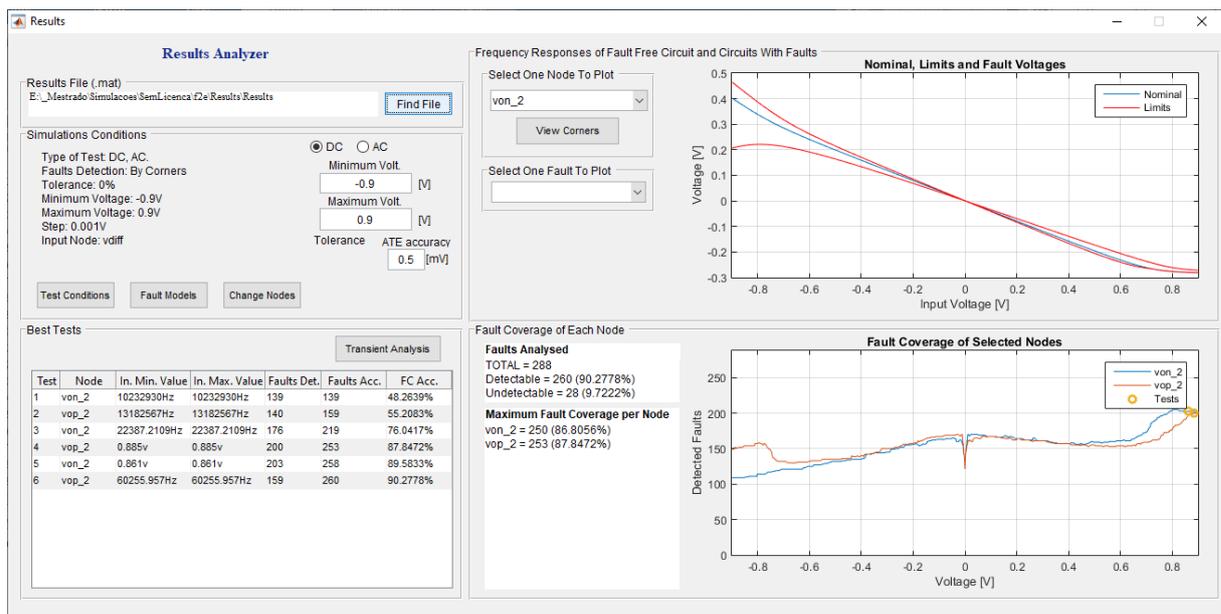
3.2.3 Exibição dos Resultados

O formato gráfico da janela “Results” é mostrado na Figura 15, onde, para fins de ilustração do visual da ferramenta, foi escolhido um exemplo que contempla teste DC e AC, por isso que o melhor conjunto de testes possui esses dois tipos de sinais de entrada. Entretanto, os gráficos mostrados nessa figura são todos referentes a sinais DC. Na Figura 15, é possível perceber quatro quadrantes. O primeiro deles, correspondente ao superior esquerdo, é responsável por mostrar informações a respeito das condições simuladas na etapa anterior, referentes à janela “AnalogCircuitTest” (seção 3.1). Nesse quadrante, é possível visualizar o nome e caminho do arquivo de resultados que está sendo lido no momento, além dos parâmetros utilizados nas simulações SPICE. O segundo quadrante, posicionado no canto superior direito, é onde se pode observar as respostas nodais das simulações SPICE. Através do *pop-up menu* com indicação “*Select One Node To Plot*”, é selecionado o nó que se deseja observar, assim, no gráfico chamado “*Nominal, Limits and Fault Voltages*”, é mostrada a tensão em função do

sinal de entrada desse nó do circuito sem falhas. Nesse mesmo gráfico, também são mostrados os limites de aceitação utilizados nas análises de detecção de falhas, juntamente com a tensão nesse mesmo nó de um circuito com falha, que pode ser selecionado no *pop-up menu* com indicação “*Select One Fault To Plot*”.

Os dois quadrantes inferiores mostram os resultados de coberturas de falhas e conjunto de teste. No quadrante da direita, são mostrados aspectos gerais, como o número total de falhas do modelo, máxima cobertura de falhas geral e as máximas coberturas de falhas em cada nó de teste. Além disso, no gráfico chamado “*Fault Coverage of Selected Nodes*”, são mostradas as coberturas de falhas em cada um dos nós de teste em função do sinal de entrada, marcando com círculos os pontos que tenham sido escolhidos para compor o conjunto dos melhores testes. Por fim, no quadrante inferior esquerdo, é mostrado o conjunto dos melhores testes através de uma tabela.

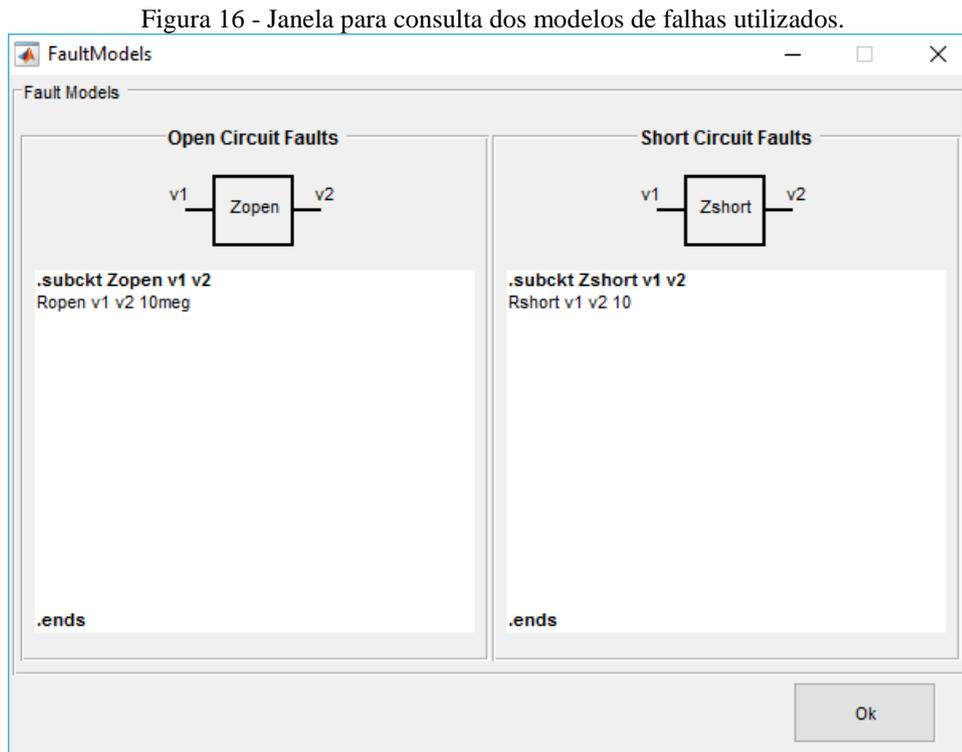
Figura 15 - Janela "Results" para o caso de teste DC e AC de um amplificador.



Fonte: elaborado pelo autor.

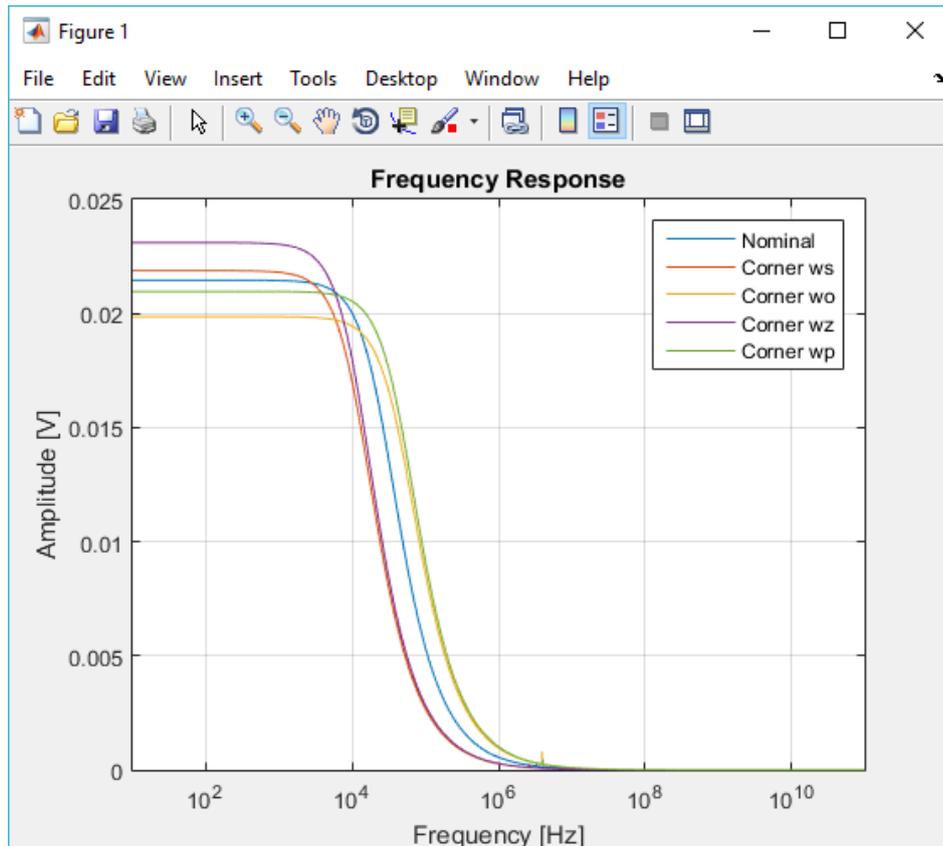
Assim como pode ser visto na Figura 15, existem alguns botões possíveis de serem pressionados. Alguns deles se referem apenas a consultas a informações, como por exemplo, o botão chamado de “*Fault Models*”, que, ao ser pressionado, abre uma janela semelhante à da Figura 16, onde pode-se visualizar os modelos de falhas que foram utilizados nas simulações SPICE. Além disso, através do botão “*View Corners*”, é possível visualizar as respostas dos circuitos com *corners* referentes ao nó plotado no gráfico “*Nominal, Limits and Fault Voltages*”. Como exemplo, é mostrada, na Figura 17, a resposta dos circuitos com *corners* no

nó “vop” para sinais de entrada AC. O nó em questão deve estar selecionado no *pop-up menu* “*Select One Node To Plot*” da Figura 15. Nesse exemplo hipotético, os limites de detecção foram determinados através da análise por *corners*, mas o botão “*View Corners*” funciona de maneira análoga em uma análise por Monte Carlo, sua indicação muda para “*View Monte Carlo*” e a sua resposta se torna a visualização das simulações de Monte Carlo.



Fonte: elaborado pelo autor.

Figura 17 - Gráfico com a resposta dos *corners* aberto através do botão "View Corners" para nó "vop".



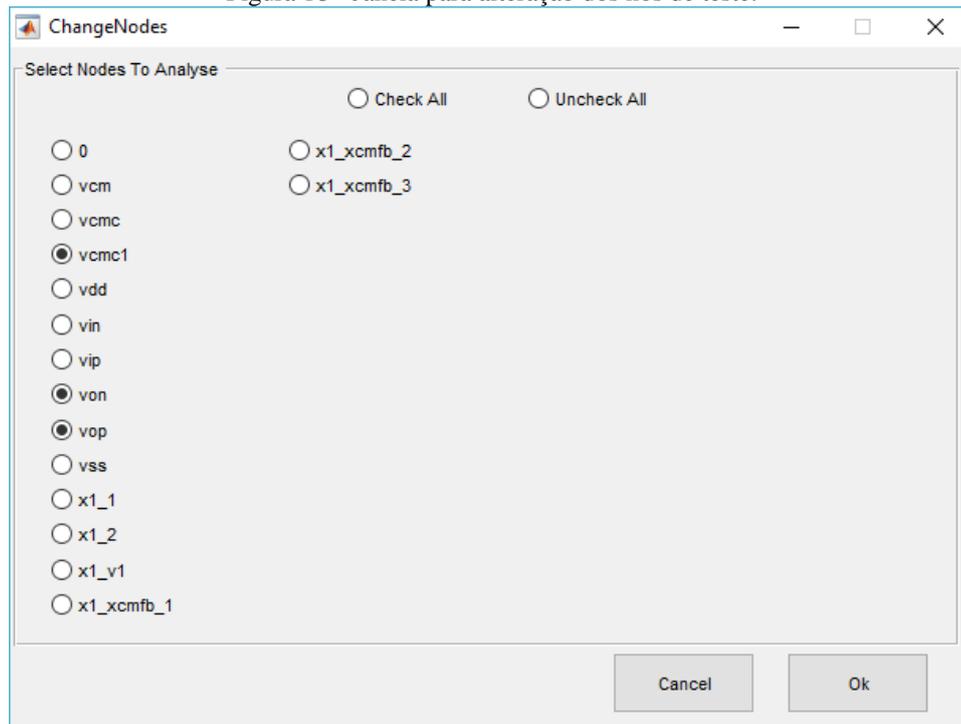
Fonte: elaborado pelo autor.

3.2.4 Possibilidades de Alterações das Análises

Um dos botões que permitem alterações das condições de teste e, conseqüentemente, dos resultados, é o chamado "Change Nodes" que serve para trocar os nós de teste. Para isso, ao ser pressionado, ele abre uma janela semelhante à da Figura 18. Selecionando novos nós e clicando no botão "Ok", o programa se encarrega de recalculer os resultados para os novos nós de teste.

Também é possível efetuar a alteração do percentual de tolerância, que determina a faixa de tensões em que as falhas não são detectadas, além da faixa dos possíveis sinais de entrada, ou seja, frequências (análise AC) ou tensões (análise DC) de entrada máxima e mínima que compõem o intervalo de análise. Qualquer uma dessas alterações faz com que o programa recalcule os resultados para a nova condição. É claro que as frequências ou tensões de entrada máximas e mínimas não podem extrapolar o intervalo utilizado nas simulações SPICE, pois a única maneira de analisar valores fora desse intervalo é utilizar a janela "AnalogCircuitsTest", descrita na seção 3.1, para rodar novamente as simulações SPICE com um novo intervalo.

Figura 18 - Janela para alteração dos nós de teste.



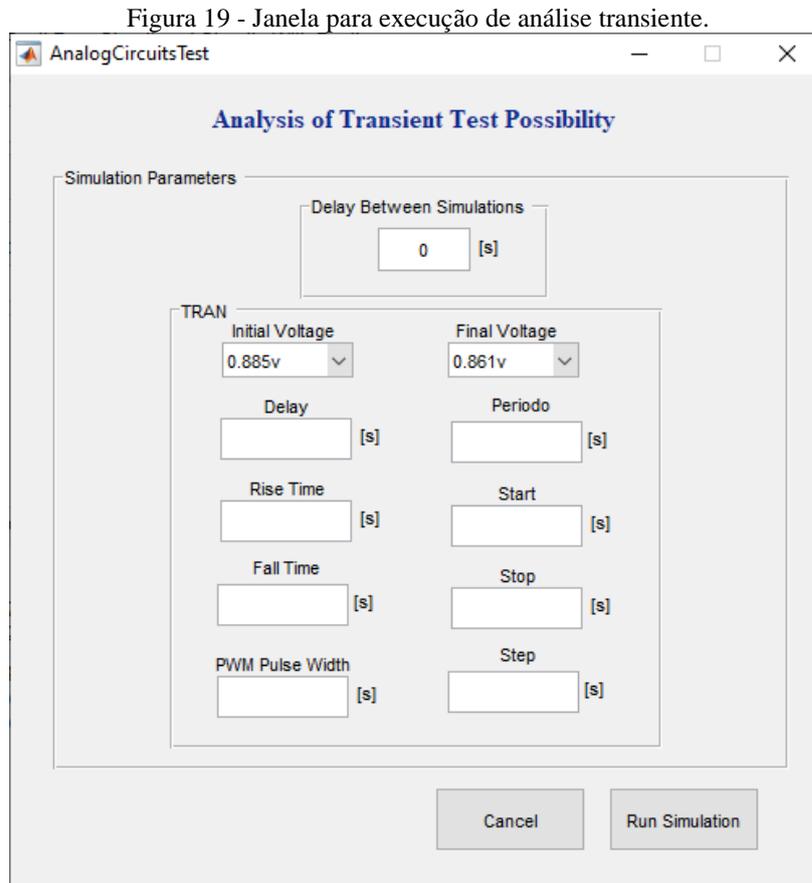
Fonte: elaborado pelo autor.

3.2.5 Análise Transiente para Possíveis Otimizações de Execução de Testes DC

Dependendo das condições simuladas, o conjunto final de testes determinado pela ferramenta pode conter testes de mais de um tipo: testes DC, AC ou transientes. Como nos testes transientes são analisados os comportamentos do tempo de acomodação do sinal e do *overshoot*, as suas execuções implicam na imposição de um degrau de tensão na entrada, sendo esse composto por duas tensões DC, a inicial e a final. Dessa forma, é possível combinar testes DC com transientes, pois aplicando uma tensão DC na entrada e esperando o regime permanente da tensão de saída, pode-se executar um teste DC, aplicando um degrau nessa tensão pode-se executar um teste transiente, e, por fim, através da nova acomodação do sinal de saída, pode-se executar um segundo teste DC. Em outras palavras, uma vez que se aplique um teste transiente no circuito, dois testes DC adicionais podem ser feitos quase que sem acréscimo no tempo de execução. Da mesma forma, a transição de tensão entre dois testes DC pode ser aproveitada para execução de um teste transiente.

Sendo assim, uma funcionalidade adicional da ferramenta busca a otimização do conjunto de teste através do aproveitamento de dois testes DC para execução de um teste transiente. Essa funcionalidade é implementada por um botão na janela "Results", chamado "*Transient Analysis*", que só está disponível quando o conjunto de teste escolhido pelo algoritmo descrito na seção 3.2.2 prevê a execução de pelo menos dois testes DC. Nesse caso,

o programa possibilita, através da janela da Figura 19, a escolha de quais das tensões de testes DC preestabelecidos no conjunto serão utilizadas como inicial e final na simulação do degrau e também possibilita a inserção de outros parâmetros que serão importantes para a execução das simulações SPICE no modo transiente.

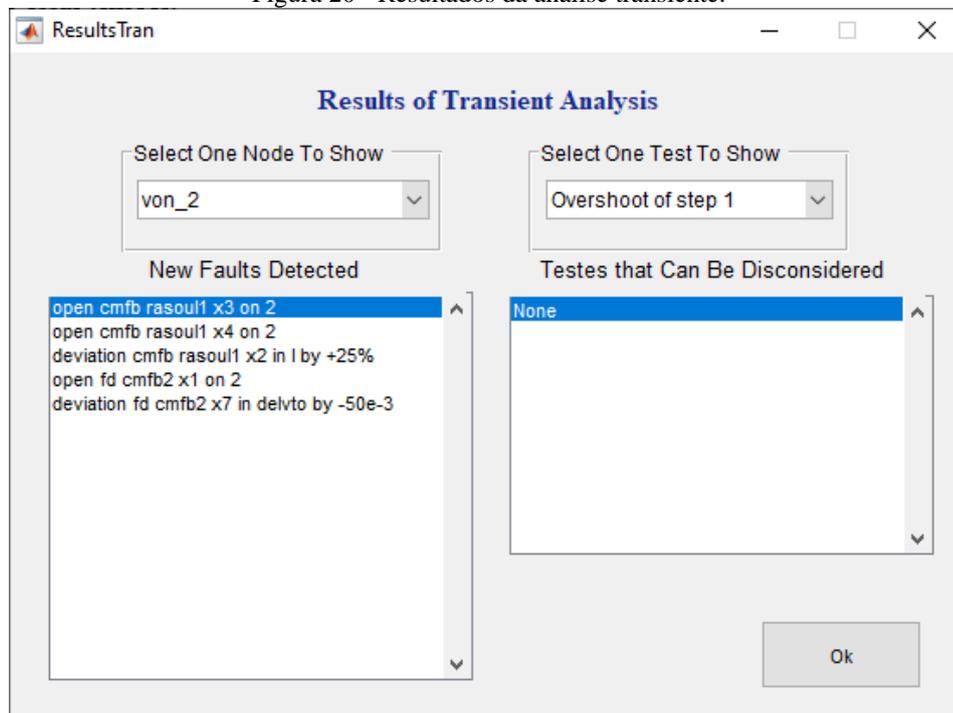


Fonte: elaborado pelo autor.

Ao pressionar o botão “*Run Simulation*” na janela da Figura 19, o algoritmo irá criar as descrições SPICE do circuito contendo cada uma das falhas do modelo e simulá-las uma a uma nas condições transientes especificadas. Ao final de tudo isso, o programa levanta o dicionário de falhas dessa condição transiente e então o compara com as falhas que já são detectadas pelo conjunto de testes pré-determinado (antes dessa análise transiente), de modo a observar quais falhas adicionais a análise transiente é capaz de detectar, bem como observar se algum dos testes do conjunto pode deixar de ser feito, condição que ocorre quando o teste transiente é capaz de detectar todas as falhas que um teste do conjunto já detecta. Nesse último caso, a substituição de um teste do conjunto pelo teste transiente pode ser vantajosa uma vez que esse teste transiente será executado na transição de dois testes DC o que diminui o tempo de execução do conjunto.

A Figura 20 mostra a janela de resultados da análise transiente. Nela, através do seletor chamado “*Select One Node To Show*”, pode-se escolher de qual nó do circuito se quer ver os resultados. Além disso, um segundo seletor, com indicação “*Select One Test To Show*”, implementa a escolha de qual dos possíveis testes analógicos se deseja observar os resultados, havendo quatro possibilidades: *Overshoot* do degrau 1, *overshoot* do degrau 2, tempo de acomodação do degrau 1 e tempo de acomodação do degrau 2. Existem dois possíveis testes para cada parâmetro (degraus 1 e 2), porque a ferramenta analisa ambas as possibilidades de degrau entre duas tensões DC, a primeira que parte da tensão inicial para a final e a outra invertendo essas tensões.

Figura 20 - Resultados da análise transiente.



Fonte: elaborado pelo autor.

4 METODOLOGIA EXPERIMENTAL – ESTUDOS DE CASO

Nesse capítulo, serão detalhadas as metodologias utilizadas para realização dos estudos de configurações de teste feitos através da ferramenta desenvolvida. Em um primeiro momento, foram analisadas as configurações de teste de um amplificador totalmente diferencial de dois estágios operando em malha fechada, onde os limites de detecção foram determinados através de Monte Carlo. Em um segundo momento, utilizando a análise de *corners* para determinação dos limites de detecção, a ferramenta foi utilizada para determinação da máxima cobertura de falhas e do conjunto dos melhores testes para o caso de um filtro passa-baixas de dois estágios. Esse filtro é composto por dois estágios iguais cascadeados, cada um implementado com a utilização de um amplificador totalmente diferencial de um estágio. Primeiramente, as configurações de teste foram avaliadas através do circuito como um todo e considerando que apenas os nós de saída é que são acessíveis às ponteiros do testador. Em um segundo momento, utilizando a opção de análise transiente que foi descrita na seção 3.2.5, analisou-se a possibilidade de incremento no número de falhas detectáveis através da execução de um teste transiente adicional, cujos valores inicial e final de tensão no degrau sejam duas tensões DC que já correspondam a testes DC presentes no conjunto dos melhores testes. Por fim, analisou-se as coberturas de falhas alcançadas através do teste de cada um dos amplificadores que compõem o circuito operando em malha aberta, com o intuito de analisar possíveis ganhos em termos de qualidade do teste para o caso de implementação de técnicas de DTF que possibilitem esse teste dos amplificadores em malha aberta no chip. Os resultados da aplicação da ferramenta aos estudos de caso selecionados são mostrados no capítulo 5.

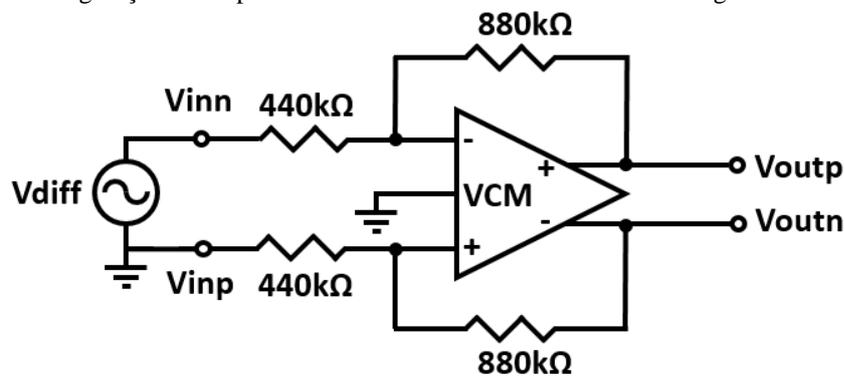
As metodologias escolhidas para a execução de cada um dos estudos de caso, levaram em consideração a possibilidade de explorar várias funcionalidades da ferramenta, sem transformar as análises em algo repetitivo. Dessa forma, devido à existência de duas possibilidades para a determinação dos limites de aceitação das respostas nodais, o primeiro circuito utilizou uma delas (Monte Carlo) e, o segundo, a outra (*corners*). Além disso, para o primeiro circuito, escolheu-se explorar a possibilidade de aumentar a cobertura de falhas através da utilização de nós internos do circuito nos testes, uma vez que a ferramenta possibilita analisar rapidamente todos eles. Essa mesma análise poderia ter sido feita no segundo circuito, porém, pelo critério já mencionado de não tornar as análises repetitivas, escolheu-se utilizar outras funcionalidades da ferramenta, que foi a análise transiente utilizando um degrau que utilize tensões DC já utilizadas por outros testes do conjunto, bem como a possibilidade de testar blocos do circuito separadamente, no caso, os amplificadores.

Vale a pena salientar que, para os estudos de caso, foram escolhidos circuitos projetados pelos próprios grupos de pesquisa parceiros especializados em projeto de circuitos analógicos. Além disso, esses circuitos foram escolhidos por utilizar tecnologias as quais os membros do PGMICRO têm acesso para fabricação, possibilidade que pode ser cogitada futuramente em novos trabalhos.

4.1 AMPLIFICADOR TOTALMENTE DIFERENCIAL DE DOIS ESTÁGIOS COM ANÁLISE DE MONTE CARLO

A fim de utilizar a análise de Monte Carlo da ferramenta, optou-se pela execução do estudo de caso de um amplificador totalmente diferencial, ou *Fully Differential Amplifier* (FDA) projetado na tecnologia IBM (atualmente Global Foundries) 8RF-DM de 130nm, composto por dois estágios de ganho diferencial e um estágio de compensação em avanço. Se comparada à compensação Miller, a técnica de compensação de fase por realimentação em avanço se mostrou vantajosa por apresentar um aumento no GBW (*Gain-Bandwidth product*) do FDA, com o ônus de requerer um amplificador a mais. Esse FDA foi analisado na configuração de malha fechada, com ganho igual a 2, assim como mostra a Figura 21.

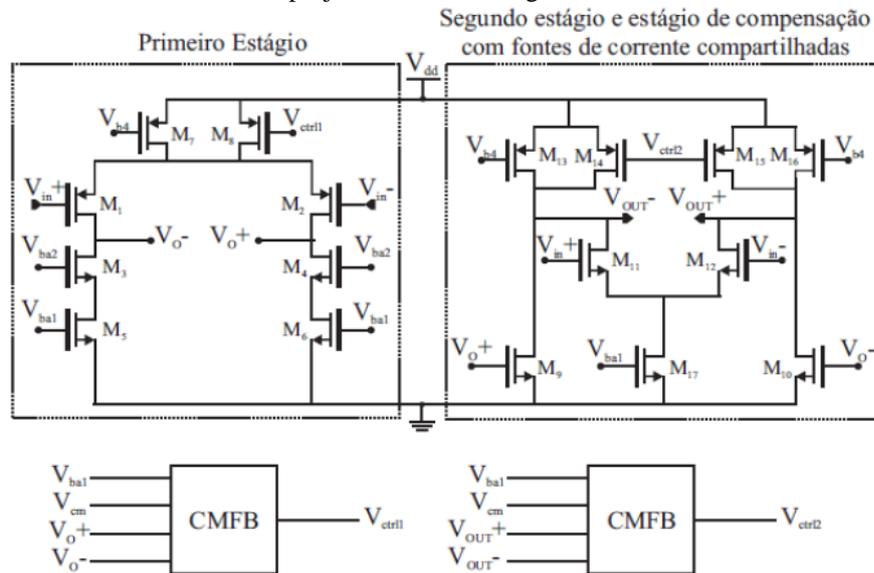
Figura 21 - Configuração do amplificador totalmente diferencial de dois estágios em malha fechada.



Fonte: elaborado pelo autor.

Na Figura 22, é possível observar que o FDA é composto por dois estágios amplificadores, um estágio de compensação e dois blocos de realimentação de modo comum, ou em inglês, *Common Mode Feedback* (CMFB). Devido ao fato de que os dois estágios amplificadores possuem correntes de polarização diferentes, há a divisão onde o primeiro estágio amplificador possui um bloco CMFB dedicado exclusivamente a si, enquanto que o segundo estágio amplificador compartilha o outro bloco CMFB com o estágio de compensação.

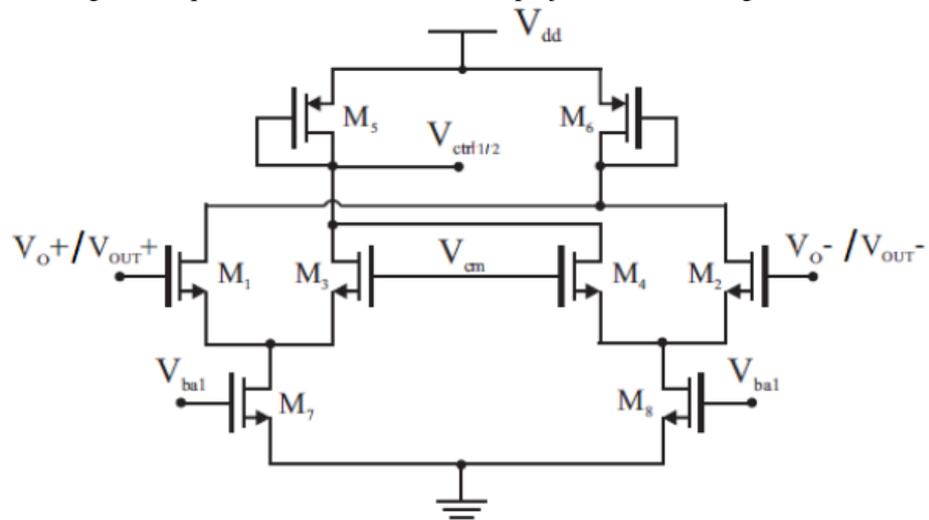
Figura 22 - Diagrama esquemático dos estágios principais e de compensação em avanço do FDA projetados em tecnologia IBM 130nm.



Fonte: Chinazzo (2016, p. 37) adaptado de Aguirre (2014).

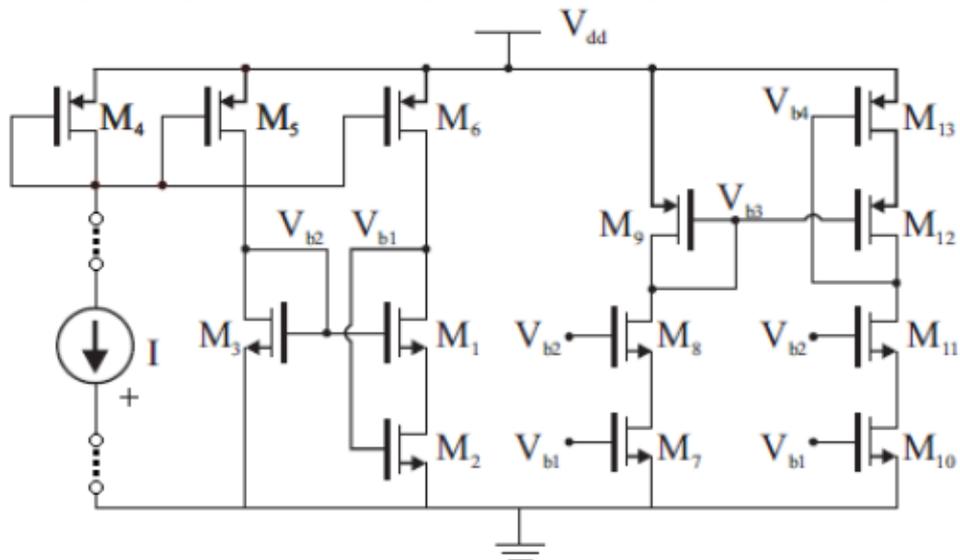
A Figura 23 mostra a composição interna dos blocos de CMFB. É possível perceber que o circuito funciona com dois pares diferenciais que fazem a comparação das tensões de saída do amplificador com a tensão V_{cm} , correspondente a 0,6V, uma vez que a tensão de alimentação seja 1,2V. Já a Figura 24 ilustra a composição do circuito de polarização, cuja topologia é apresentada por Baker (2010), onde todos os transistores foram projetados de modo a operarem saturados. A fonte I é definida em $20\mu A$ e seu valor é espelhado pelos transistores M_4 , M_5 e M_6 . Por fim, na Tabela 1 são mostradas todas as dimensões dos transistores que compõem o FDA como um todo.

Figura 23 - Diagrama esquemático do bloco de CMFB projetado em tecnologia IBM 130nm.



Fonte: Chinazzo (2016, p. 38) adaptado de Aguirre (2014).

Figura 24 - Diagrama esquemático do bloco de polarização do FDA projetado em tecnologia IBM 130nm.



Fonte: Chinazzo (2016, p. 39) adaptado de Aguirre (2014).

Tabela 1 – Dimensões dos transistores do FDA projetado em tecnologia IBM 130nm.

Parâmetro	Bloco		
	OPAMP	CMFB	Polarização
W ₁ /L ₁ [μm/μm]	6,0/0,6	96,0/0,6	10,0/0,6
W ₂ /L ₂ [μm/μm]	6,0/0,6	96,0/0,6	10,0/1,2
W ₃ /L ₃ [μm/μm]	10,0/0,6	96,0/0,6	2,0/3,6
W ₄ /L ₄ [μm/μm]	10,0/0,6	96,0/0,6	10,0/1,2
W ₅ /L ₅ [μm/μm]	10,0/1,2	80,0/1,2	10,0/1,2
W ₆ /L ₆ [μm/μm]	10,0/1,2	80,0/1,2	10,0/1,2
W ₇ /L ₇ [μm/μm]	10,0/1,2	80,0/1,2	10,0/1,2
W ₈ /L ₈ [μm/μm]	10,0/1,2	80,0/1,2	10,0/0,6
W ₉ /L ₉ [μm/μm]	5,0/2,4	-	2,4/1,2
W ₁₀ /L ₁₀ [μm/μm]	5,0/2,4	-	10,0/1,2
W ₁₁ /L ₁₁ [μm/μm]	12,0/0,6	-	10,0/0,6
W ₁₂ /L ₁₂ [μm/μm]	12,0/0,6	-	8,0/0,6
W ₁₃ /L ₁₃ [μm/μm]	10,0/1,2	-	10,0/1,2
W ₁₄ /L ₁₄ [μm/μm]	10,0/1,2	-	-
W ₁₅ /L ₁₅ [μm/μm]	10,0/1,2	-	-
W ₁₆ /L ₁₆ [μm/μm]	10,0/1,2	-	-
W ₁₇ /L ₁₇ [μm/μm]	20,0/1,2	-	-

Fonte: Adaptado de Aguirre (2014).

O modelo de falhas adotado para esse amplificador foi o mesmo de Chinazzo (2016). O autor considerou 6 falhas catastróficas em cada transistor, sendo três delas de circuito aberto (na porta, dreno e fonte) e outras três de curto-circuito (entre porta e dreno, porta e fonte, e dreno e fonte). Como falhas paramétricas, foram consideradas variações de 25% de largura de canal e de comprimento do transistor (SEGURA; HAWKINS, 2004), além de variações na tensão de *threshold* de 90mV. Esse valor de tensão foi escolhido por corresponder a 6 desvios padrões em torno da média para essa tecnologia (DENG; SHI; ZHANG, 2012), uma vez que, segundo Greer, Korkin e Labanowski (2003), um desvio padrão da tensão de *threshold* no nó tecnológico de 130nm corresponde a 15mV. As variações de largura e comprimento não representam necessariamente a variabilidade geométrica na fabricação dos transistores, mas são uma maneira de injetar, por simulação, falhas que impactem na capacidade de corrente do transistor. Estas falhas podem ter origem, por exemplo, em defeitos do tipo “*spot*” (imperfeições na litografia causadas por impurezas) que alterem a geometria da região ativa do transistor.

O circuito possui um total de 46 transistores, o que corresponderia a 414 falhas. Porém, foram desconsideradas 7 falhas de curto-circuito entre terminais de dreno e porta de transistores

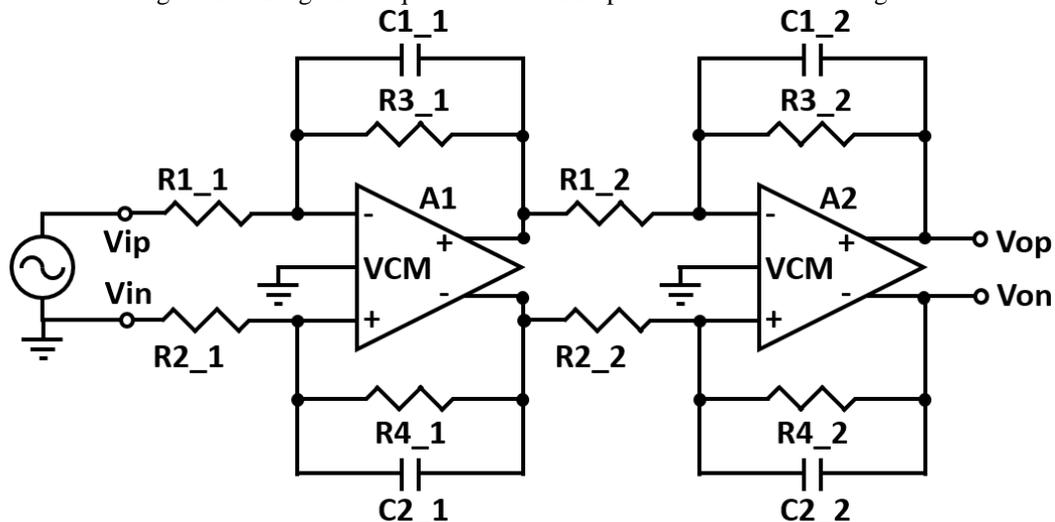
que possuíam esse curto na própria topologia do circuito, são eles: M_5 e M_6 de cada um dos dois blocos CMFB e M_3 , M_4 e M_9 do bloco de polarização. Dessa forma, o modelo final considerou 407 falhas.

Os limites de tensão que discriminam as falhas detectáveis das não-detectáveis foram baseados em simulações de Monte Carlo das quais foram simuladas 100 condições para cada circuito e o limite de aceitação foi estipulado em dois desvios padrões em torno da média. Como sinais de entrada do circuito, foram considerados sinais AC com frequências variando entre 10Hz e 100GHz com 100 pontos simulados em cada década. Vale a pena ressaltar que no momento em que foram simulados os testes, com possíveis sinais de entrada em frequências tão altas de até 100GHz, não se tinha conhecimento acerca da limitação de banda do instrumento de medida. Sendo assim, esses valores podem não ser muito práticos, porém servem para ilustrar a funcionalidade da ferramenta, que, por sua vez, possibilita inclusive que o usuário estipule posteriormente um limite superior em frequências menores que 100GHz se necessário.

4.2 FILTRO PASSA-BAIXAS DE DOIS ESTÁGIOS

O segundo teste da ferramenta desenvolvida, baseado em estudos de caso, consistiu em aplicar a análise de *corners* à um filtro passa baixas de dois estágios para determinar os limites de teste. Esse filtro é composto por dois estágios iguais conectados em cascata, cada um deles apresentando ganho unitário na banda de passagem e frequência de corte igual a 10kHz. O diagrama esquemático desse filtro é mostrado na Figura 25. Todos os resistores desse circuito possuem resistências de 570k Ω , e todos os capacitores possuem capacitâncias de 28pF.

Figura 25 - Diagrama esquemático do filtro passa-baixas de dois estágios.

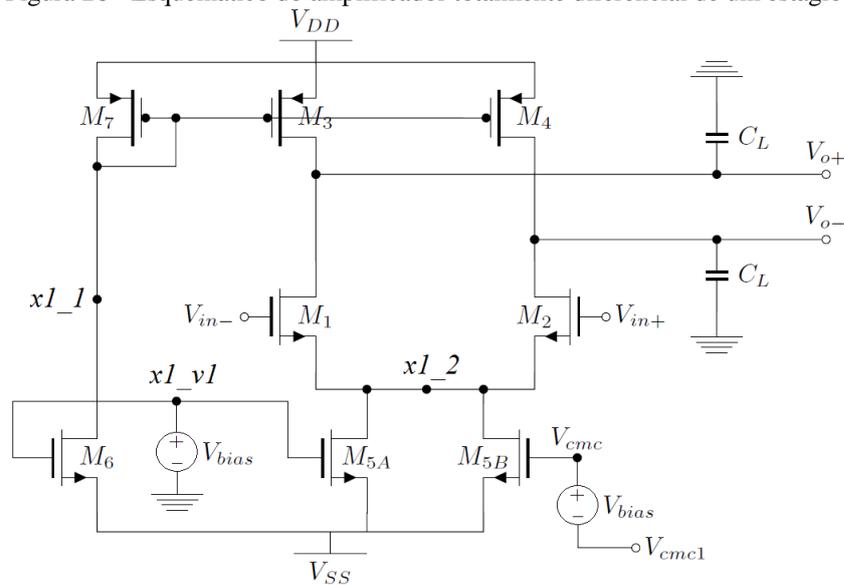


Fonte: elaborado pelo autor.

Para composição do circuito, os FDAs utilizados foram do modelo projetado por Oliveira, Severo e Girardi (2014), na tecnologia XFAB 180nm, cujo modelo esquemático pode ser observado na Figura 26. Nele, os transistores M1 e M2 formam o par diferencial de entrada e são os responsáveis pelo ganho do amplificador. Esse par diferencial tem, como carga ativa, os transistores M3 e M4, que estão conectados em espelho de corrente com o dispositivo M7. Esse espelho de corrente e o par diferencial de entrada são polarizados pelas respectivas correntes dos transistores M6 e M5.

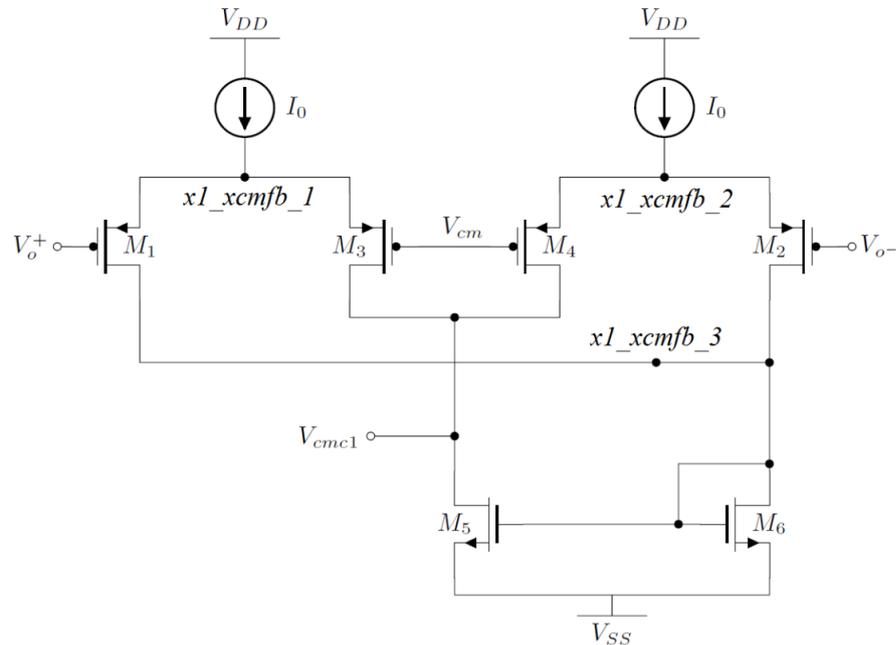
O circuito CMFB é mostrado na Figura 27, onde os transistores M1 e M2 são casados e recebem, respectivamente, a saída positiva e negativa do amplificador. Os dispositivos M3 e M4 são os responsáveis pela comparação do sinal, proveniente do amplificador, com a tensão de modo comum desejada (“ V_{cm} ”). Os transistores M5 e M6, conectados em espelho de corrente, juntamente com as fontes de corrente, são os responsáveis pela polarização do circuito (BENDER, 2015).

Figura 26 - Esquemático do amplificador totalmente diferencial de um estágio.



Fonte: adaptado de Oliveira; Severo; Girardi (2014).

Figura 27 - Esquemático elétrico do circuito CMFB.



Fonte: adaptado de Oliveira; Severo; Girardi (2014).

Na descrição SPICE, tanto o FDA (Figura 26), quanto o circuito de CMFB (Figura 27) foram descritos como subcircuitos. Nessa descrição, dentro do subcircuito do FDA, já é feita a adição do bloco CMFB, através da adição desse subcircuito com o nome “xcmfb”. Dessa forma, para descrever o circuito como um todo, foi necessário apenas declarar esse subcircuito do FDA, chamando-o de “x1”, e interligar as tensões: de alimentação, de modo comum desejado e de entrada.

Na toolbox, os nós internos de um subcircuito recebem um prefixo, referente ao nome dado a esse subcircuito na sua declaração, seguido de um “_”. Dessa forma, como pode ser visto na Figura 26, o nó chamado na descrição SPICE de “1”, que pertence ao subcircuito chamado “x1”, será chamado, pela toolbox, de “x1_1”. Na Figura 27, é possível identificar um caso onde há um subcircuito dentro de outro. Seguindo a mesma lógica, o nó chamado de “1”, que pertence ao subcircuito “xcmfb”, que, por sua vez, pertence ao subcircuito “x1”, será chamado de “x1_xcmfb_1”.

A topologia de descrição SPICE permite que um subcircuito seja declarado uma vez e depois possa ser chamado quantas vezes forem necessárias. Por exemplo, no caso do filtro da Figura 25, que possui dois FDAs iguais, o circuito do FDA pode ser descrito como um subcircuito, e então, na descrição geral do filtro, esse subcircuito será chamado duas vezes recebendo nomenclaturas diferentes para o primeiro e o segundo FDA. Entretanto, se os dois FDAs estiverem referenciados ao mesmo subcircuito, a ferramenta não conseguirá inserir falhas

em componentes internos de um FDA de maneira individual. Isso porque, para inserir essa falha, a ferramenta precisa alterar o subcircuito que compõe o FDA, o que, nesse caso, alteraria ambos os FDAs. A solução para esse problema consiste em não utilizar um subcircuito mais de uma vez na linguagem SPICE, ou seja, nesse exemplo do filtro, duplicar os subcircuitos que compõem os FDAs, de modo que cada FDA tenha o seus subcircuitos exclusivos. Dentro do subcircuito que foi duplicado, o nome dos componentes pode ser exatamente igual ao do original, porém deve haver diferenciação no nome de cada subcircuito.

Na Tabela 2 são mostrados os parâmetros dos componentes do circuito FDA. Vale a pena ressaltar que a tensão de polarização “Vbias” e a corrente de polarização I_0 foram consideradas ideais.

Tabela 2 - Especificações do projeto do circuito FDA.

Parâmetro	Amp. Diferencial	CMFB
W_1/L_1 [$\mu\text{m}/\mu\text{m}$]	39,29/0,197	35,91/0,8918
W_2/L_2 [$\mu\text{m}/\mu\text{m}$]	39,29/0,197	35,91/0,8918
W_3/L_3 [$\mu\text{m}/\mu\text{m}$]	27,41/6,83	35,91/0,8918
W_4/L_4 [$\mu\text{m}/\mu\text{m}$]	27,41/6,83	35,91/0,8918
W_5/L_5 [$\mu\text{m}/\mu\text{m}$]	15,3/9,75	7,19/0,4628
W_6/L_6 [$\mu\text{m}/\mu\text{m}$]	15,3/9,75	7,19/0,4628
W_7/L_7 [$\mu\text{m}/\mu\text{m}$]	27,41/6,83	-
Vbias [V]	-167,045	-
I_0 [μA]	-	15,19

Fonte: Chinazzo (2016, p. 34) adaptado de Oliveira, Severo e Girardi (2014).

4.2.1 Análises AC e DC do Filtro Passa-Baixas Completo

Em um primeiro momento, o filtro da Figura 25 teve as suas configurações de teste analisadas como um todo. Nessa condição, foi considerado que apenas os nós de saída do circuito serão utilizados como nós de teste, ou seja, desconsiderando a possibilidade de acesso a nós internos por parte das ponteiros do testador.

O modelo de falhas utilizado levou em consideração as seis falhas catastróficas por transistor consideradas na seção 4.1. No entanto, foram excluídas do modelo as falhas de curto-circuito entre os terminais de gate e dreno dos transmissores M6 do CMFB e M7 do bloco principal, haja vista a existência desses curtos-circuitos no próprio projeto do circuito. Além disso, uma vez que as tensões de polarização “Vbias” sejam consideradas ideais, as falhas de

curto-circuito entre os terminais de gate e fonte dos transistores M5A e M6 do bloco principal também foram excluídas do modelo.

Com relação às falhas paramétricas, Bender (2015) levou em consideração variações do comprimento do canal e da tensão de *threshold* em cada transistor. As variações de comprimento de canal foram de $\pm 25\%$, escolha que a autora justificou com base em Brosa e Figueiras (2000). Já as variações da tensão de *threshold* foram definidas em $\pm 50\text{mV}$. Neste caso, a autora justificou que – segundo Borkar, Karnik, Narendra, Tschanz, Keshavarzi e De (2003) – a variação de três desvios padrões da tensão de *threshold* na tecnologia de 180nm é 30mV, ao mesmo tempo em que Deng, Shi e Zhang (2012) definem que as falhas paramétricas devem estar dentro do intervalo de três a seis desvios padrões, ou seja, de $\pm 30\text{mV}$ até $\pm 60\text{mV}$.

As falhas paramétricas de redução de 25% do comprimento de canal dos transistores M1 e M2 do bloco principal foram excluídas do modelo, assim como fez Bender (2015), uma vez que os comprimentos de canal nesses casos se tornariam inferiores ao limite da tecnologia (180nm). Com isso, se chegou a um total de 120 falhas analisadas em cada um dos FDAs (três falhas paramétricas e seis catastróficas por transistor, com exclusão das seis mencionadas anteriormente).

Além disso, também foram consideradas falhas catastróficas e paramétricas nos componentes externos aos amplificadores. Sendo assim, o modelo de falhas também considerou duas falhas catastróficas em cada resistor e capacitor, uma de curto-circuito e outra de circuito aberto, bem como duas falhas paramétricas em cada um desses componentes, correspondentes as variações de $\pm 25\%$ dos seus valores de resistência e de capacitância. Com isso, o modelo completo de falhas para o filtro passa-baixas de dois estágios considerou 288 falhas. Sendo 120 em cada amplificador e 48 nos componentes externos.

Os limites de tensão que foram utilizados para discriminação das falhas detectáveis das não-detectáveis se basearam nas respostas dos circuitos simulados considerando os *corners* da tecnologia. As siglas dos quatro *corners* para esta tecnologia são: wo (*Worst Case One Condition*), wp (*Worst Case Power Condition*), ws (*Worst Case Speed Condition*) e wz (*Worst Case Zero Condition*).

Por fim, foram considerados dois tipos de teste: o DC e o AC. Para o teste DC, foi considerada toda a faixa de tensão entre a alimentação simétrica do circuito que corresponde a $\pm 0,9\text{V}$ com passos de 1mV. Para o teste AC foram consideradas frequências de 10Hz até 1MHz com 100 pontos simulados em cada década. A fim de obter resultados mais realistas, o limite para a menor diferença de tensão que pode ser detectada pelo testador foi escolhido como 0,5mV. Novamente, os resultados são mostrados no capítulo 5.

4.2.2 Análise Transiente do Filtro Passa-Baixas Completo

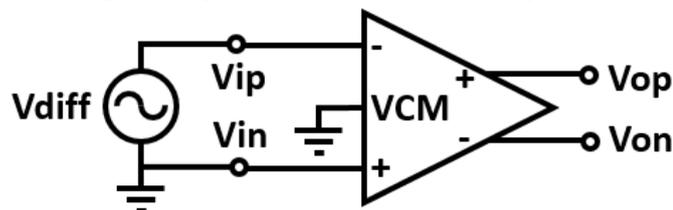
Uma vez que a análise DC do circuito completo tenha sido executada conforme descrita na seção 4.2.1, a proposta da presente seção é apresentar o recurso de análise transiente entre dois testes DC pré-estabelecidos no conjunto. Esse recurso foi descrito na seção 3.2.5 e, como dito, busca a otimização do conjunto final de configurações de teste, através da análise de possíveis testes transientes que possam ser executados entre dois testes DC que já seriam executados de qualquer maneira, podendo assim, aumentar a cobertura de falhas sem interferir significativamente no tempo de execução do conjunto e, conseqüentemente, no custo do teste.

Para isso, o modelo de falhas continua sendo o mesmo da seção 4.2.1, os limites de aceitação continuam se baseando nas respostas dos 4 *corners*. Os valores de tensão inicial e final do degrau que irá compor esse teste transiente dependerá de quais valores de tensão DC irão compor as configurações de teste que a ferramenta escolher para compor o conjunto mais otimizado ao final das análises dessa mesma seção 4.2.1.

4.2.3 Amplificador Totalmente Diferencial em Malha Aberta

Como outra possibilidade, foram analisadas as configurações de teste do FDA projetado por Oliveira, Severo e Girardi (2014) que compõem o filtro passa baixas da seção 4.2, mas em malha aberta (esquemático da Figura 28). O objetivo dessa análise foi avaliar a possibilidade de implementação de técnicas de DFT no circuito do filtro de modo a possibilitar o acesso a nós internos do circuito que permitam testar individualmente cada um dos dois amplificadores do filtro. Vale a pena destacar que o ganho do circuito em malha aberta é limitado, porém muito elevado. Por isso, para o teste AC, é importante utilizar sinais de entrada de baixa amplitudes de modo que esses não causem a saturação do sinal de saída.

Figura 28 - Diagrama esquemático do FDA de um estágio em malha aberta.



Fonte: elaborado pelo autor.

Através da análise em malha aberta, será possível avaliar qual o ganho que se pode ter em relação à cobertura de falhas se, além de se testar o filtro como um todo, também for possível desconectar seus amplificadores e testá-los separadamente. No modelo utilizado de 288 falhas,

mais de 80% delas estão relacionadas a componentes internos desses amplificadores, o que serve de indicativo que esses circuitos são peças-chaves a serem levados em consideração na determinação de configurações de teste.

Sendo assim, para possibilitar a comparação, o modelo de falhas para cada amplificador em malha aberta foi o mesmo da seção 4.2.1, bem como os possíveis sinais de entrada para o teste, que também foram iguais aos descritos nessa seção.

5 RESULTADOS E DISCUSSÕES

Nesse capítulo são mostrados os resultados para os estudos descritos no capítulo 4. Na Tabela 3 são mostrados os tempos de execução do algoritmo para cada um dos resultados presentes nesse capítulo. Na Tabela, “Tempo 1” se refere ao tempo para criar e simular todas as descrições SPICE, e “Tempo 2” se refere ao tempo para ler os arquivos resultantes, calcular a matriz de falhas, salvar o arquivo de resultados, abrir a janela “Results”, calcular as coberturas de falhas e determinar o conjunto dos melhores testes. Todas essas análises foram executadas em um notebook com 64bits de barramento de dados, processador Intel® Core™ i-5 de 2,5GHz e memória RAM de 6GB.

Tabela 3 - Tempos de execução do algoritmo.

Seção	Circuito	Tempo 1	Tempo 2	Nº de Falhas Analisadas
5.1	FDA 130nm	43min	1min 47s	407
5.2.1	Filtro Passa Baixas (AC e DC)	32min e 35s	1min e 23s	288
5.2.2	Filtro Passa Baixas (Transiente)	21min e 07s	43s	288
5.2.3	FDA 180nm	14min e 25s	32s	120

5.1 AMPLIFICADOR TOTALMENTE DIFERENCIAL DE DOIS ESTÁGIOS COM ANÁLISE DE MONTE CARLO PARA DEFINIÇÃO DOS LIMITES DE TESTE

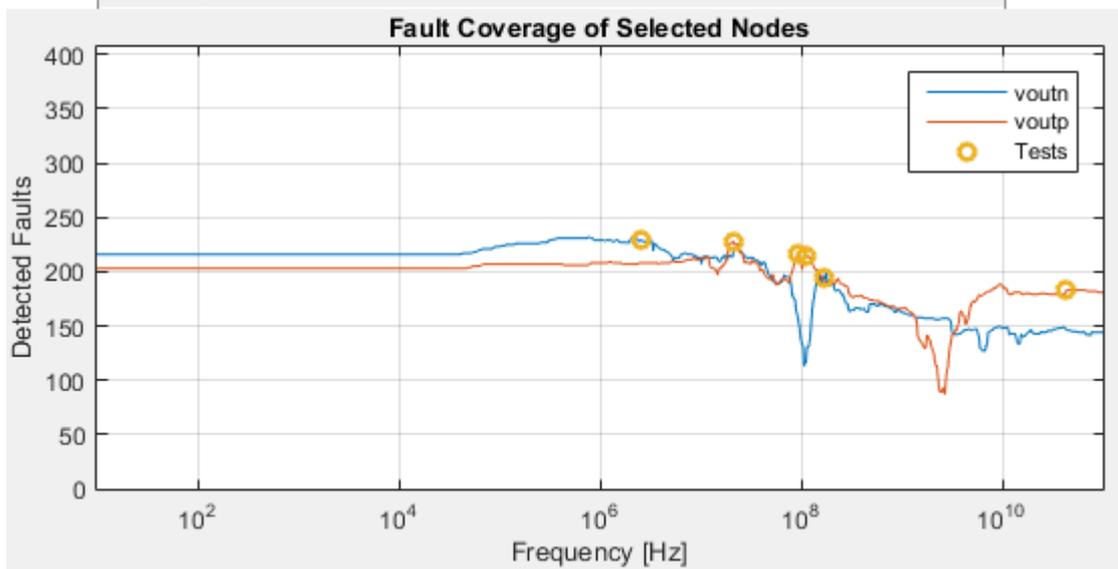
Em um primeiro momento, utilizando apenas os nós de saída do circuito como possíveis nós de teste, a máxima cobertura de falhas ficou em 63,88%. Esse número é alcançado por um conjunto de 6 testes que se distribuíram mais pelas frequências mais elevadas, acima dos 10MHz. Na Figura 29, são mostrados recortes da ferramenta que mostram o conjunto dos melhores testes, bem como o número de falhas detectadas em cada um dos nós em função dos sinais de entrada.

A Figura 29 (e outras figuras com esse mesmo formato que aparecerão ao longo da exposição dos resultados) é dividida em duas partes: a parte superior e a inferior. Na parte superior, é mostrado o melhor conjunto de testes em um formato de tabela. As duas primeiras colunas representam o índice do teste e o nó do circuito que será medido. A terceira e a quarta coluna mostram quais são os respectivos mínimo e máximo sinal de entrada a ser utilizado na execução do teste. Isso porque, dada a natureza analógica do sinal, um teste DC não aceitará apenas uma tensão específica, mas sim uma faixa de valores, bem como um teste AC, que

também aceitará frequências dentro de uma certa faixa. Por fim, as três últimas colunas representam respectivamente o número de falhas que cada um dos testes detecta de maneira individual, o número acumulado de falhas detectadas (teste em questão juntamente com os seus antecessores) e a cobertura de falhas acumulada. Sendo assim, a última linha dessa tabela mostra a cobertura de falhas referente à execução do conjunto completo de teste. Na parte inferior da figura, é mostrado o número de falhas que cada nó de teste detecta em função dos sinais de teste utilizados.

Figura 29 - Conjunto dos melhores testes considerando apenas nós de saída para amplificador totalmente diferencial de dois estágios.

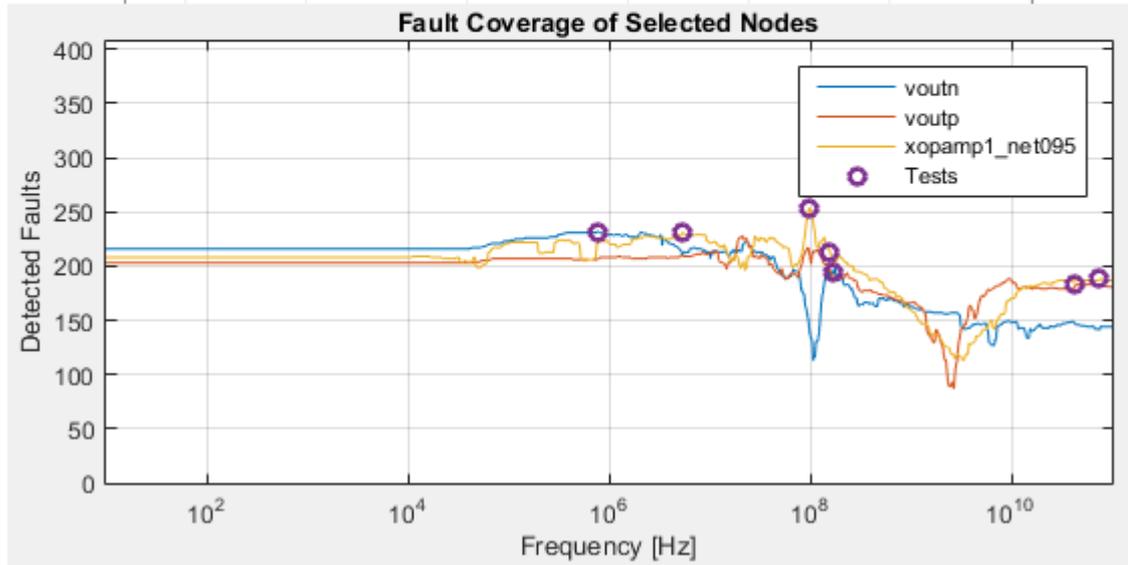
Test	Node	In. Min. Value	In. Max. Value	Faults Det.	Faults Acc.	FC Acc.
1	voutp	93325432Hz	93325432Hz	217	217	53.317%
2	voutn	2570395.75Hz	2570395.75Hz	229	248	60.9337%
3	voutp	114815360Hz	114815360Hz	215	250	61.4251%
4	voutn	165958688Hz	165958688Hz	195	253	62.1622%
5	voutp	20892962Hz	20892962Hz	228	257	63.145%
6	voutp	43651584000...	43651584000...	183	260	63.8821%



Buscando melhores coberturas de falhas, os comportamentos dos nós internos do circuito foram observados e constatou-se que a cobertura de falhas pode ser incrementada para 73,22% se for possível acessar um nó interno correspondente ao fio que conecta o transistor M4 com o transistor M6 na Figura 22. Assim como pode ser visto na Figura 30, essa cobertura de falhas é alcançada por um conjunto composto por 7 testes, ou seja, um teste a mais do que o conjunto anterior, porém com quase 10% de cobertura de falhas a mais.

Figura 30 - Conjunto dos melhores testes considerando nó interno para amplificador totalmente diferencial de dois estágios.

Test	Node	In. Min. Value	In. Max. Value	Faults Det.	Faults Acc.	FC Acc.
1	xopamp1...	97723720Hz	97723720Hz	254	254	62.4079%
2	xopamp1...	158489312Hz	158489312Hz	213	272	66.8305%
3	voutn	165958688Hz	165958688Hz	195	287	70.516%
4	voutp	43651584000...	43651584000...	183	291	71.4988%
5	xopamp1...	5370318Hz	5370318Hz	231	294	72.2359%
6	xopamp1...	75857756160...	75857756160...	188	297	72.973%
7	voutn	776247.125Hz	776247.125Hz	232	298	73.2187%



Fonte: elaborado pelo autor.

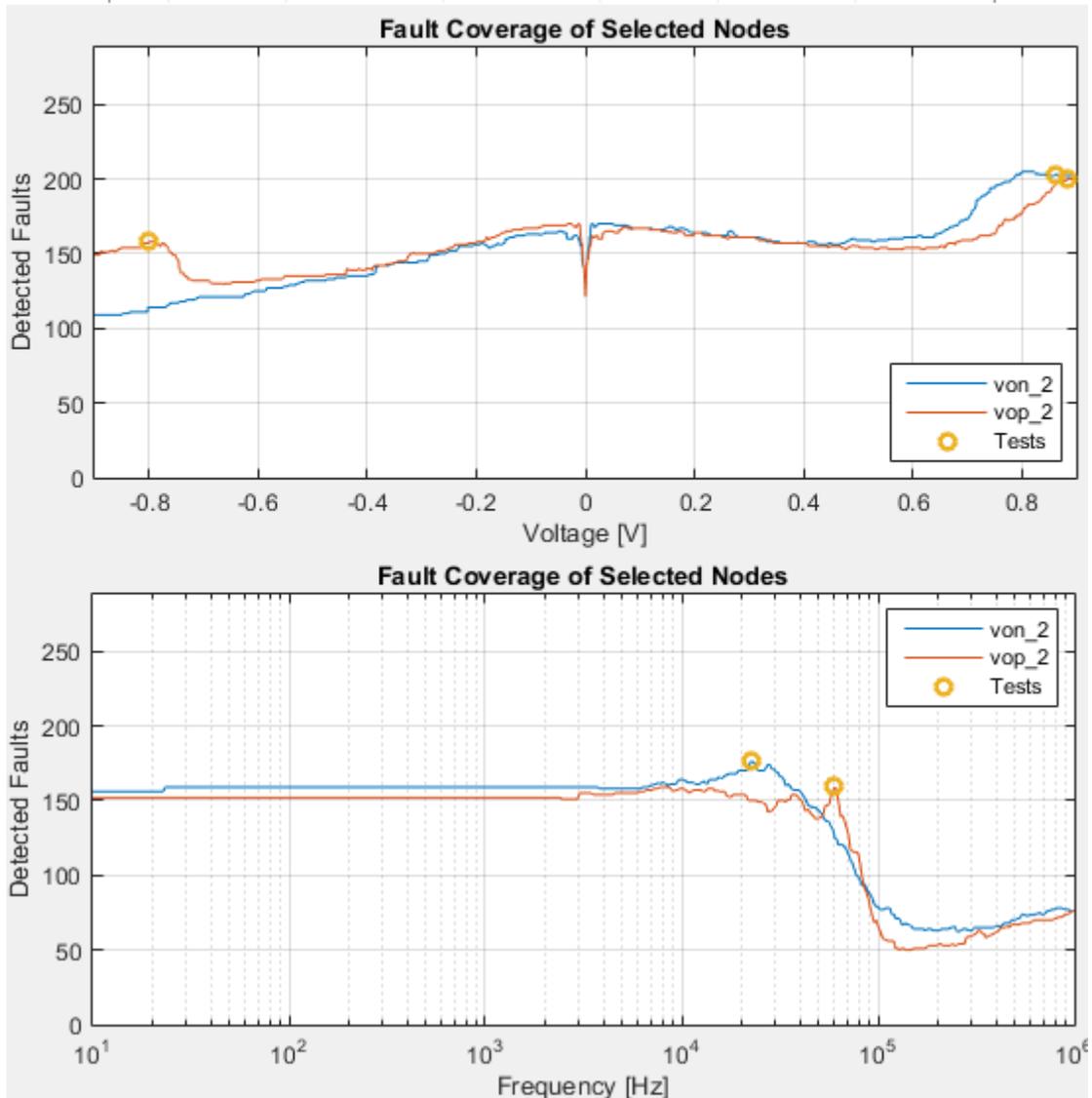
5.2 FILTRO PASSA-BAIXAS DE DOIS ESTÁGIOS

5.2.1 Análises AC e DC do Filtro Passa-Baixas Completo

Analisando o circuito completo do filtro passa baixas e considerando que apenas as suas saídas serão nós acessíveis para execução dos testes, a máxima cobertura de falhas foi de 87,15%. Observa-se, através da Figura 31, que são necessários apenas 5 testes para que essa cobertura de falhas seja alcançada. Importante lembrar que essa análise mescla testes DC com testes AC, o que justifica uma cobertura de falhas significativamente maior do que a obtida com o amplificador totalmente diferencial de dois estágios da seção 5.1.

Figura 31 - Conjunto dos melhores testes considerando apenas nós de saída para filtro passa-baixas de dois estágios.

Test	Node	In. Min. Value	In. Max. Value	Faults Det.	Faults Acc.	FC Acc.
1	von_2	22387.2109Hz	22387.2109Hz	176	176	61.1111%
2	vop_2	0.885v	0.885v	200	239	82.9861%
3	von_2	0.861v	0.861v	203	244	84.7222%
4	vop_2	60255.957Hz	60255.957Hz	159	248	86.1111%
5	vop_2	-0.8v	-0.8v	158	251	87.1528%



Fonte: elaborado pelo autor.

5.2.2 Análise Transiente do Filtro Passa-Baixas Completo

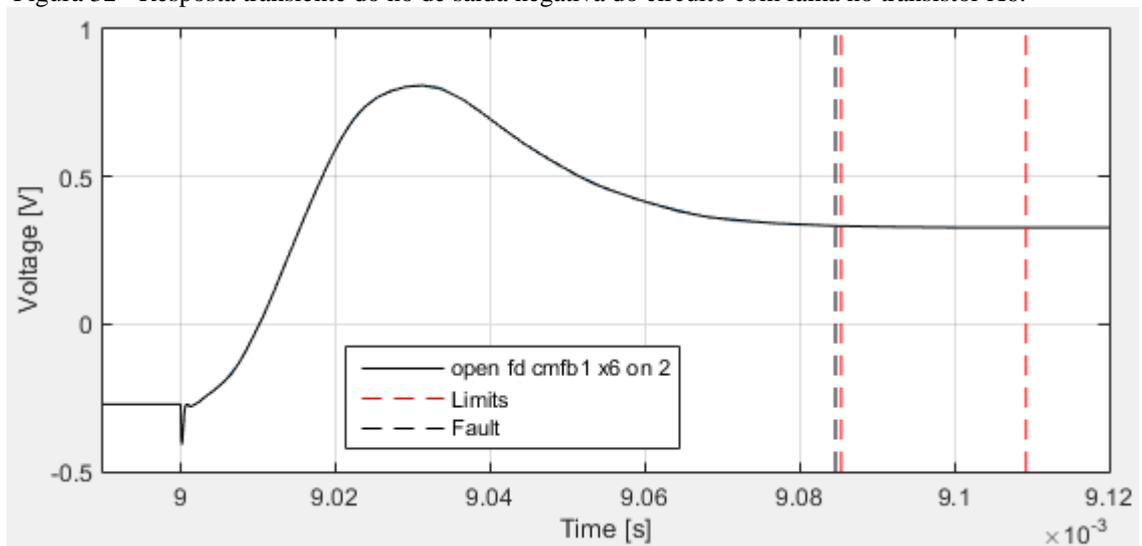
Assim como mostra a Figura 31, o conjunto final de testes para o filtro passa-baixas de dois estágios apresentou três testes DC, sendo dois deles com tensões parecidas, diferindo-se em apenas 24mV. Sendo assim, considerando essas três tensões, o teste transiente que se mostrou mais eficiente considera um degrau entre duas das tensões DC mais discrepantes entre si, sendo elas: a de 861mV e a de -800mV. Então, utilizando o recurso de análise transiente que

a ferramenta oferece, descrito na seção 3.2.5, foram simuladas as condições de teste para essa condição de degrau na entrada.

A ferramenta analisou os quatro possíveis testes transientes: o que considera a medição do tempo de acomodação do sinal para condição de degrau de subida e também de descida, além de considerar a mensuração de *overshoot* também para esses dois degraus. Embora a cobertura de falhas alcançada na seção 5.2.1 já tenha sido elevada, a ferramenta constatou que aferição do tempo de acomodação do sinal na saída negativa do circuito é capaz de detectar uma falha adicional, elevando um pouco a cobertura de falhas que passa a ser de 87,50%.

A falha que passou a ser detectada corresponde a de circuito aberto no terminal 2 do transistor M6 do bloco amplificador do FDA, cujo esquemático encontra-se na Figura 26. A tensão no nó de saída negativa desse circuito com falha em questão, bem como os limites de aceitação para o tempo de acomodação são mostrados na Figura 32, onde as linhas pontilhadas definem os tempos de acomodação desse circuito com falha (linha preta), além dos limites de aceitação (linhas vermelhas).

Figura 32 - Resposta transiente do nó de saída negativa do circuito com falha no transistor X6.

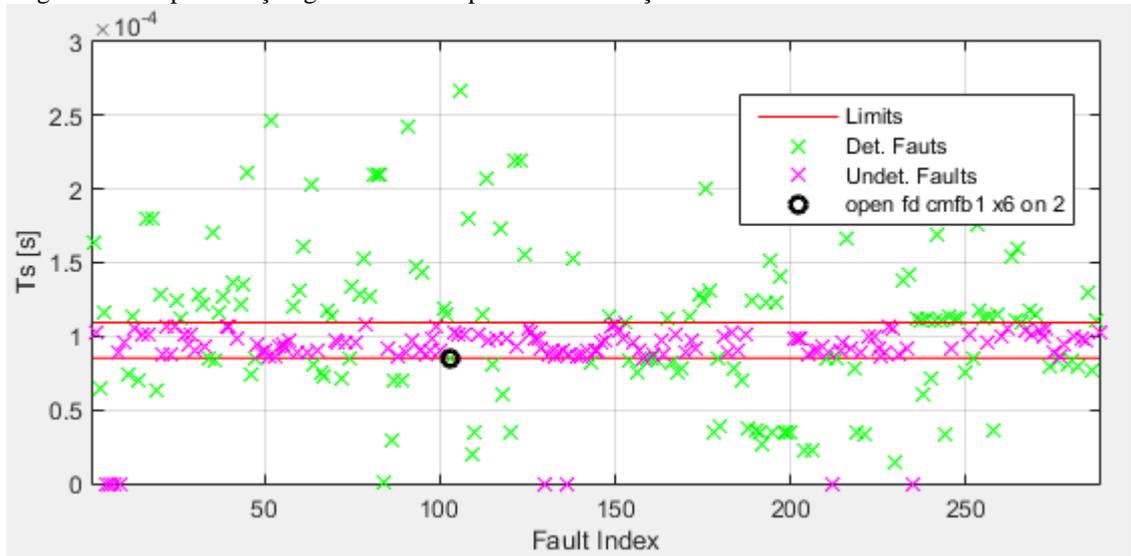


Fonte: elaborado pelo autor.

Ademais, visando uma melhor visualização dos resultados transientes, a ferramenta dispõe do gráfico ilustrado na Figura 33. Nele, cada tempo de acomodação do sinal de saída negativa de um dos circuitos com falha do modelo é representado por um “x”. As linhas vermelhas correspondem aos limites de aceitação. Dessa maneira, são discriminadas através das cores verde e rosa as falhas que são, respectivamente, detectáveis e não detectáveis por esse teste em questão. Além de que, está sinalizado por um círculo preto, a falha de circuito aberto

no terminal 2 do transistor X6 que foi a falha adicional que esse teste transiente foi capaz de detectar.

Figura 33 - Representação gráfica dos tempos de acomodação de todos os circuitos com falha.



Fonte: elaborado pelo autor.

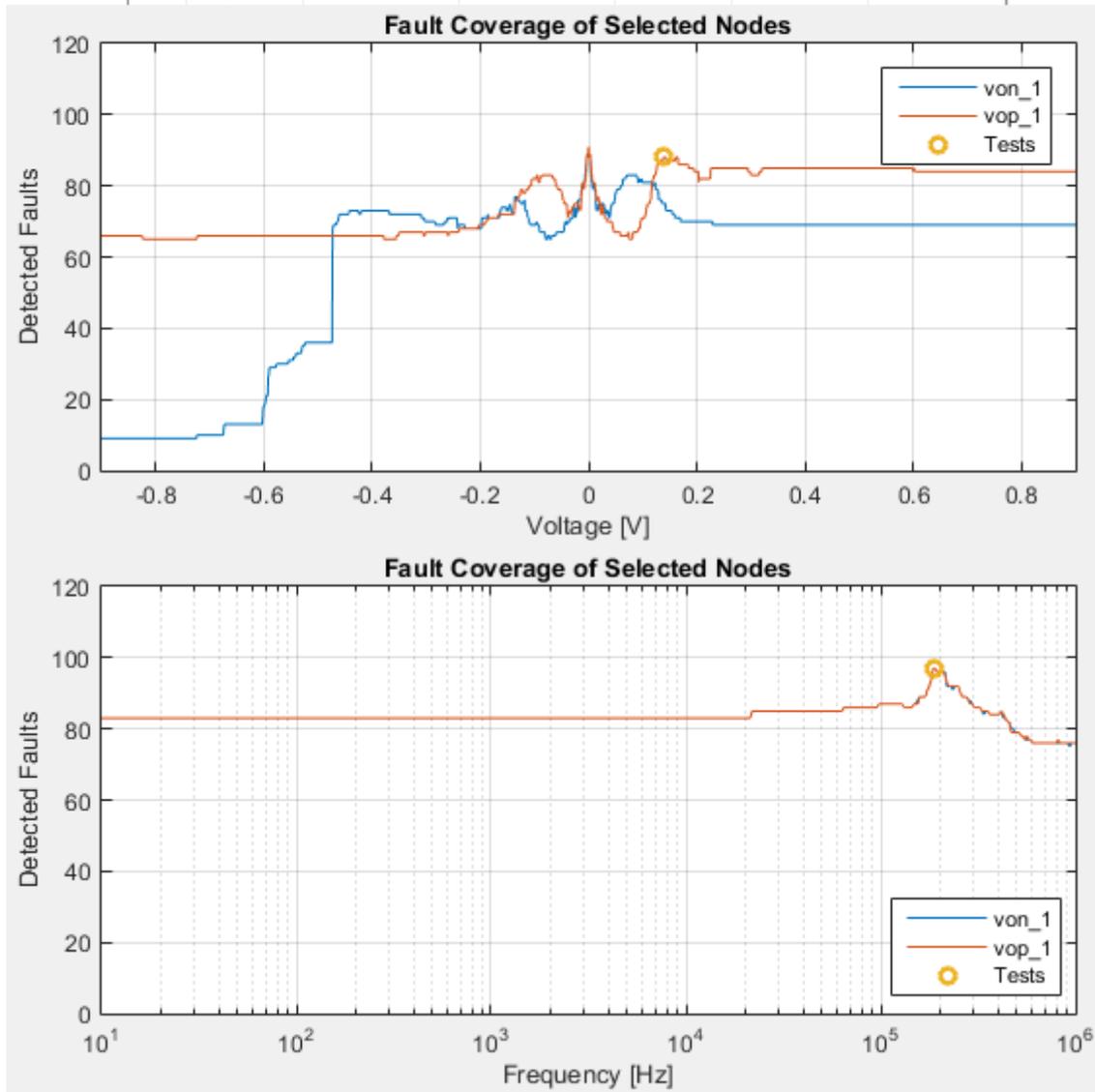
Vale a pena destacar então que todos os pontos verdes da Figura 33 representam falhas detectáveis pelo teste de tempo de acomodação. Entretanto, o único que causa algum acréscimo na cobertura de falhas é o marcado pelo círculo preto, pois todos os outros já são detectados pelo conjunto de testes DC e AC determinado na seção 5.2.1. Pode-se concluir acerca desse teste transiente que a detecção de apenas uma falha adicional ficou abaixo das expectativas, pois outros casos podem apresentar números maiores de detecção de falhas adicionais. Entretanto, se o testador for habilitado a executar testes transientes, o custo de execução desse se torna praticamente zero, pois de qualquer maneira, o conjunto já previa a execução de dois testes DC com tensões de entrada que compõem o degrau em questão, fazendo com que essa transição no sinal de entrada seja executada de qualquer maneira.

5.2.3 Amplificador Totalmente Diferencial em Malha Aberta

As simulações do amplificador totalmente diferencial de um estágio em malha aberta demonstraram que, considerando apenas os nós de saída, é possível alcançar a cobertura de falhas de 93,33% com apenas 3 testes, assim como mostra a Figura 34. Em outras palavras, apenas 8 das 120 falhas não puderam ser detectadas.

Figura 34 - Conjunto dos melhores testes considerando apenas nós de saída para amplificador totalmente diferencial de um estágio.

Test	Node	In. Min. Value	In. Max. Value	Faults Det.	Faults Acc.	FC Acc.
1	von_1	186208.7188Hz	186208.7188Hz	97	97	80.8333%
2	vop_1	186208.7188Hz	186208.7188Hz	97	99	82.5%
3	vop_1	0.138v	0.138v	88	112	93.3333%



Fonte: elaborado pelo autor.

Na seção 5.2.1, onde foi considerado o teste do filtro como um todo, a cobertura de falhas alcançada foi de 87,15%, o que quer dizer que 37 das 288 falhas do modelo não puderam ser detectadas. Usando a ferramenta para identificar quais eram essas falhas não detectáveis, constatou-se que apenas duas delas correspondiam a componentes externos aos amplificadores operacionais, sendo elas, os desvios de +25% no valor da resistência R2_1 e da capacitância C1_1 (Figura 25). Dessa forma, 35 falhas não detectáveis são relacionadas a componentes internos dos amplificadores.

Em um primeiro momento, pode-se imaginar que todas as 8 falhas que não são detectáveis nos testes com o amplificador em malha aberta também não seriam detectadas quando esse amplificador estiver sendo testado no filtro como um todo. Isso se mostrou verdade para o amplificador do primeiro estágio (A1). Porém, 1 falha, mais especificamente o desvio de -25% no comprimento do canal do transistor X5b do bloco amplificador do FDA do segundo estágio do filtro, pode ser detectada no teste do filtro como um todo, na seção 5.2.1, mesmo não sendo detectada no teste do amplificador em malha aberta dessa seção.

Sendo assim, através de todas essas análises, pode-se dizer que a inclusão de técnicas de DFT no circuito do filtro passa-baixas de dois estágios que possibilitem o teste individual de cada um dos amplificadores em malha aberta possibilita a detecção de 10 e 9 falhas adicionais nos componentes internos dos respectivos amplificadores do primeiro e do segundo estágio. Com isso, a cobertura de falhas total seria de 92,36%.

6 CONCLUSÃO

Nesse trabalho, partindo dos códigos desenvolvidos por Chinazzo (2016) para automatização das simulações SPICE, foi desenvolvida uma ferramenta (*toolbox*) em MATLAB para análise de configurações de teste em circuitos analógicos. Essa ferramenta é capaz de determinar automaticamente um conjunto otimizado de testes para um circuito analógico qualquer. Ela parte da descrição SPICE do circuito sobre análise e de um modelo de falhas pré-estabelecido para então gerar, simular e analisar o funcionamento de novas descrições SPICE do circuito, cada uma contendo uma falha do modelo, tudo isso de maneira automática. Por fim, também de maneira automática, a ferramenta determina o melhor conjunto de testes dentro das condições simuladas. Diversas funcionalidades foram adicionadas à essa ferramenta e cada uma delas foi utilizada em pelo menos um estudo de caso, que utilizaram circuitos amplificadores projetados nas tecnologias IBM (atualmente Global Foundries) 8RF-DM de 130nm e XFAB 180 μ m.

A ferramenta se mostrou capaz de rodar e analisar um elevado número de simulações, executando, em um tempo relativamente baixo, um trabalho que, se fosse feito manualmente, seria extremamente demorado e maçante. Além disso, as possibilidades de reavaliação dos resultados se mostraram importantíssimas na busca por aumentos nas coberturas de falhas. Isso se deve à metodologia implementada na ferramenta que lê os arquivos resultantes das simulações SPICE uma única vez, determina os resultados e armazena todas essas informações em um arquivo “.mat”, formato específico do MATLAB, que facilita a releitura. Dessa forma, é possível mudar diversas condições que levam a ferramenta a recalcular os resultados. Alguns exemplos dessas possibilidades foram as buscas por aumentos nas coberturas de falhas através: da execução de testes em nós internos do circuito na seção 5.1; da execução de um teste transiente adicional que utilizava a transição entre dois testes DC já pertencentes ao conjunto final; e da análise de configurações de testes do amplificador que implementava o filtro passa-baixas operando em malha aberta. O usuário da ferramenta pode facilmente avaliar cada uma dessas possibilidades de modo a encontrar o melhor custo benefício, uma vez que, à exceção dos testes transientes, as outras possibilidades envolvem técnicas de DFT que possibilitem a medição de tensão em nós internos do circuito, ou até mesmo o completo isolamento de um bloco do circuito para que seja testado separadamente, o que acarreta em aumento do custo do projeto com o bônus de incrementar a qualidade do teste.

Com relação ao primeiro estudo de caso, observou-se que o amplificador totalmente diferencial de dois estágios apresentou a menor cobertura de falhas presente nesse trabalho, que foi de 63,88%. Esse valor era atingido através da execução de 6 testes, e considerava apenas os

nós de saída do circuito como nós de teste. A ferramenta se mostrou bastante eficiente na busca por um nó interno do circuito que, se fosse adicionado ao conjunto dos nós aos quais as ponteiros do testador têm acesso, acarretariam na melhora significativa da cobertura de falhas. Todos os nós internos foram avaliados um a um em uma metodologia de tentativa e erro, até que se chegasse a conclusão de que a medição de tensão de um nó específico aumentaria a cobertura de falhas em quase 10%, chegando a 73,22%.

O segundo estudo de caso, considerou a possibilidade de execução de testes, tanto AC, quando DC, diferentemente do caso anterior que considerou apenas testes AC. Em virtude disso, a máxima cobertura de falhas se mostrou significativamente maior, alcançando 87,15% considerando apenas os acessos aos nós de saída do circuito. Como o conjunto final de testes continha 3 testes DC em um total de 5 testes, avaliou-se a possibilidade de execução de um teste transiente entre a troca de tensão que ocorreria de qualquer maneira quando o testador terminasse um teste DC e fosse iniciar o outro. Concluiu-se que esse teste é capaz de detectar uma única falha adicional, o que eleva a cobertura de falhas para 87,50%. Esse aumento não é tão significativo, porém, por esse teste adicional apresentar custo quase zero de execução, vale a pena adicioná-lo ao conjunto. Por fim, os resultados dos testes dos amplificadores em malha aberta foram bastante significativos, pois demonstraram a capacidade de elevar a cobertura de falhas do circuito como um todo até a marca de 92,36%. A ferramenta também foi utilizada para identificar as falhas que podem ou não ser detectadas em cada um dos casos, e concluiu-se que uma das falhas de componentes internos do amplificador não pode ser detectada pelo testes deste em malha aberta, porém é detectada pelo teste do circuito como um todo, o que é um detalhe curioso.

O presente trabalho apresentou duas contribuições principais. A primeira delas, foi a criação de uma toolbox em MATLAB que possibilita, de forma automatizada, a análise de configurações de teste para circuitos analógicos, bem como o manuseio dos resultados. Além disso, foram feitos estudos de caso que demonstraram que técnicas de DFT e estratégias de testes adicionais podem incrementar significativamente a cobertura de falhas e, conseqüentemente, a qualidade do teste. Por fim, conclui-se que a ferramenta desenvolvida possibilita a investigação rápida e eficiente dos efeitos de cada uma dessas técnicas ou estratégias de testes adicionais.

7 SUGESTÕES PARA TRABALHOS FUTUROS

Diversas oportunidades de melhoria surgiram ao longo do desenvolvimento dessa ferramenta, criando um potencial de torná-la mais completa. Concluiu-se que é possível aprimorá-la através do aumento dos seus recursos, aplicá-la em circuitos que sejam relevantes ao grupo de pesquisa, além de utilizar *Benchmarks*. De maneira mais específica, lista-se abaixo possibilidades de continuidade ao trabalho aqui apresentado:

- Migrar o código para outra linguagem de programação (*python*, por exemplo), tirando a ferramenta do ambiente MATLAB de desenvolvimento e transformando-a em um executável independente;
- Tornar o layout do circuito como um arquivo de entrada da ferramenta, de modo a possibilitar que o próprio algoritmo escolha automaticamente, pela proximidade dos componentes e linhas de metais, quais falhas de curto-circuito devem pertencer ao modelo;
- Implementar o diagnóstico de falhas;
- Fazer a ferramenta simular o nível de ruído nas tensões nodais, em prol de utilizar essa grandeza na determinação do valor mínimo de diferença de tensão que deve existir entre a resposta nominal do circuito e os limites de aceitação;
- Tornar possível a análise de configurações de teste que levem em consideração a medição de corrente elétrica ao invés de tensão;
- Analisar a variabilidade dos circuitos com falha, para tornar a detecção mais realista.
- Realizar uma extensiva avaliação da ferramenta através de estudos de caso reais e *benchmarks*, bem como comparar, em profundidade, os resultados aqui obtidos com outros métodos existentes;
- Testar a ferramenta através de circuitos reais montados pela conexão de componentes discretos. Isso possibilitaria a injeção de falhas reais (através de componentes) e a execução dos testes na prática;
- Classificar as falhas de acordo com as suas probabilidades de ocorrência, pois falhas mais prováveis de acontecerem são mais importantes de serem detectadas;
- Quantificar o custo do teste em função do seu próprio tipo, para que essa informação seja levada em consideração na hora de escolhê-los.

REFERÊNCIAS

- ALZAKER, H. A.; ELWAN, H.; ISMAIL, M. A CMOS Fully Balanced Second-Generation Current Conveyor. **IEEE Transactions on Circuits and Systems: Analog and Digital Signal Processing**, v.50, n. 6, p. 278 – 287, jun. 2003.
- ARSLAN, Baris; ORAILOGLU, Alex. **Tracing the Best Test Mix through Multi-Variate Quality Tracking**. Berkeley: Ieee 31st Vlsi Test Symposium (vts), 2013. 6 p.
- BAKER, R. J. CMOS Circuit Design, Layout, and Simulation. 3. ed. [S.l.]: Wiley-IEEE Press, 2010. ISBN 978-0-470-88132-3.
- BALEN, T. R. **Teste de Dispositivos Analógicos Programáveis (FPAAS)**. 127 p. Dissertação (Mestrado em Engenharia Elétrica) - Programa de Pós-Graduação em Engenharia Elétrica. Universidade Federal do Rio Grande do Sul, Porto Alegre, 2006.
- BENDER, I. D. **Teste de Amplificadores Diferenciais através de Medida DC e Transiente de Tensões Internas de Polarização**. 93 p. Dissertação (Mestrado em Microeletrônica) – Programa de Pós-Graduação em Microeletrônica. Universidade Federal do Rio Grande do Sul, Porto Alegre, 2015.
- BENSCHWARTZ, R.; SAKTHIVEL, P. "A process variation tolerant OTA design for low power ASIC design," May. 2016.
- BORKAR, S. *et al.* Parameter Variations and Impact in Circuits and Microarchitecture. **Proceedings of the 40th Annual Design Automation Conference - DAC**. New York, USA: ACM, p. 338 – 342, 2003.
- BROSA, A.M.; FIGUERAS, J. On Maximizing the Coverage of Catastrophic and Parametric Faults, **Journal of Electronic Testing: Theory and Applications** 16, 251–258, 2000.
- BUSHNEL, M.L.; AGRAWAL, V. D. **Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits**. New York, USA: Kluwer Academic Publishers, 2002.
- CALVANO, J. V.; ALVES, V. C.; LUBASZEWSKI, M. S. Fault Detection in Systems With 2nd Order Dynamics Using Transient Analysis. In: SYMPOSIUM ON INTEGRATED CIRCUITS AND SYSTEMS DESIGN, 1999, Natal, Brasil. Proceedings...[S. l.: S. n.], 1999, p.110 – 114.
- 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: VLSI

TEST SYMPOSIUM, 18., 2000, Montreal, Canada. Proceedings... Los Alamitos, USA: IEEE Computer Society Press, 2000, p. 319-324.

CHAUHAN H.; CHOI Y.; ONABAJO M.; JUNG I. S.; KIM Y. B. Accurate and Efficient On Chip Spectral Analysis for Built-In Testing and Calibration Approaches. **IEEE Transactions on Very Large Scale Integration**, v. 22 n. 3, mar 2014, p 497 – 506.

CHEN, C. S.; LI, L.; LIM, Q.; TEH, H. H.; OMAR, N. F. B.; LER, C. L.; WATT, J. T. “A compact test structure for characterizing transistor variability beyond 3σ ,” **IEEE Trans. On Semiconductor Manufacturing**, vol. 28, no. 3, pp. 329–336, Aug 2015.

CHINAZZO, André Lucas. **Desenvolvimento de Teste de Amplificadores Diferenciais utilizando Simulações SPICE Automatizadas**. 2016. 97 f. TCC (Graduação) - Curso de Engenharia Elétrica, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2016.

COYETTE A.; ESEN B.; DOBBELAERE W.; VANHOOREN R.; GIELEN G. Automatic Generation of Test Infrastructures for Analog Integrated Circuits by Controllability and Observability Co-Optimization. **INTEGRATION, the VLSI jornal**, may 2016.

DAVIS, B. **The Economics of Automatic Testing**. [S.l.]: McGraw Hill, 1982.

DENG, Y.; SHI, Y.; ZHANG, W. An Approach to Locate Parametric Faults in Nonlinear Analog Circuits. **IEEE Transactions on Instrumentation and Measurement**, v. 61, n. 2, p. 358 – 367, feb. 2012.

FRACCAROLI E.; FUMMI F. Analog Fault Testing Through Abstraction, **Design, Automation and Test in Europe**, 2017 p 270 - 273.

GREER, J.; KORKIN, A.; LABANOWSKI, J. Nano and Giga Challenges in Microelectronics. 1. ed. [S.l.]: Elsevier B.V., 2003. ISBN 0-444-51494-5.

HUGHES, J. L. A. Multiple Fault Detection Using Single Fault Test Sets. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, New York, USA: IEEE, v. 7, n. 1, p.100 – 108, jan. 1988.

KABISATPATHY, Prithviraj; BARUA, Alok; SINHA, Satyabroto. **FAULT DIAGNOSIS OF ANALOG INTEGRATED CIRCUITS**. Dordrecht: Springer, 2005. 182 p.

KARKI, J. Fully-Differential Amplifiers. **Texas Instruments Incorporated: Application Report**, p. 1 – 27, jan. 2002.

KONDAGUNTURI, R. et al. Benchmark circuits for analog and mixed-signal testing. In: Southeastcon '99. **Proceedings**. IEEE. [S.l.: s.n.], 1999. p. 217–220.

LI, R.; WU, H. "A study of narrow transistor layout proximity effects for 28nm poly/sion logic technology," **China Semiconductor Technology International Conference**, Mar. 2016.

LUBASZEWSKI, M. *et al.* **Design of Self-Checking Fully Differential Circuits and Boards**. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, New York, USA: IEEE, v.2, n. 2, p. 113, apr. 2000.

MASON S. J.; HILL R. R.; MÖNCH L.; ROSE O.; JEFFERSON T.; FOWLER J. W. Introduction to Monte Carlo Simulation. **IEEE Proceedings of the 2008 Winter Simulation Conference**. 2008, p 91 – 100.

MEZZOMO, C.; BAJOLET, A.; CATHIGNOL, A.; DI FRENZA, R.; GHIBAUDO, G. et al., "Characterization and modeling of transistor variability in advanced CMOS technologies," IEEE Trans. Electron Devices, vol. 58, no. 8, pp. 2235–2248, Aug. 2011.

MILOR, L. S. A tutorial introduction to research on analog and mixed-signal circuit testing. **IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing**, v. 45, n. 10, p. 1389–1407, Oct 1998. ISSN 1057-7130.

MILOR, L.; VISVANATHAN, V. Detection of catastrophic fault in analog integrated circuits. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, p. 114-130, 1989.

MILOR, L.; VISVANATHAN, V. Efficient Go/No Go Testing of Analog Circuits. In: INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS, 1987, [S.1]. **Proceedings...** [S.1.: S.n.], 1987, p. 414 – 417.

MIZUTANI, T.; KUMAR, A.; HIRAMOTO, T. "Measuring threshold voltage variability of 10G transistors," in Proc. IEEE IEDM, Washington, DC, USA, pp. 25.2.1–25.2.4, 2011.

MOORE, G. E. **Cramming More Components Onto Integrated Circuits**. Electronics, p. 114–117, 1965.

NAGEL, L. W.; PEDERSON, D. **SPICE (Simulation Program with Integrated Circuit Emphasis)**. [S.1.], 1973.

OLIVEIRA, A. C. de; SEVERO, L. C.; GIRARDI, A. G. A Two-Step Methodology for Automatic Design of Fully Differential Amplifiers With Output Balance, **XXIX South Symposium on Microelectronics (SIM)**, Alegrete, Brazil, mai. 2014.

PETRASHIN, P.; DUALIBE, C.; LANCIONI, W.; TOLEDO, L. Low-Cost DC BIST for Analog Circuits: A Case Study. **Test Workshop (LATW)**, Latin American, p. 1–4, apr. 2013.

PERROTT, M. H. **HSPICE Toolbox for Matlab and Octave (also for use with Ngspice)**. 2011. Disponível em: <http://www.cppsims.com/download_hspice_tools.html>. Acesso em: 27/03/2017.

POEHL, F.; DEMMERLE F.; ALT J.; OBERMEIR J. “Production test challenges for highly integrated mobile phone SoCs—A case study,” in Proc. IEEE Eur. Test Symp., Praha, Czech Republic, 2010, pp. 17–22.

QUIAN, K. “Variability modeling and statistical parameter extraction for CMOS devices,” Doctorate dissertation, University of California, Berkeley, pp. 158, Jun. 2015.

RAPPITSCH, G.; SEEBACHER, E.; KOCHER, M.; STADLOBER, E. SPICE Modeling of Process Variation Using Location Depth Corner Models. **IEEE Transactions on Semiconductor Manufacturing**. v. 17, n. 2, p. 201 – 213, mai. 2004.

RENOVELL, M. Digital and Analog System Testing: fundamentals and new challenges In: The 16th INTERNATIONAL CONFERENCE ON MICROELECTRONICS, 2004, Tunis, Tunísia. **Proceedings...** [S. l.: S. n.], 2004, p. 8 – 10.

ROGENMOSER, R.; CLARK, L. T. “Reducing transistor variability for higherperformance, lower-power chips,” **IEEE Micro**, vol. 33, no. 2, pp 18–26, Mar. 2013.

RUBINSTEIN, R.Y. Simulation and the Monte-Carlo Method, John Wiley & Sons, New York, NY, 1981.

SACHDEV, M.; DE GYVEZ, J. P. **Defect-Oriented Testing for Nano-Metric CMOS VLSI Circuits**. Dordrecht: Springer, 2007. 343 p.

SEGURA, Jaume; HAWKINS, Charles F.. **CMOS Electronics: How It Works, How It Fails**. Piscataway: Wiley-ieee Press, 2004.

SHIN, C, "Variation-aware advanced CMOS devices and SRAM," Springer, p. 140, 2016.

SOUDERS, T. M.; STENBAKKEN, G. N. A Comprehensive Approach for Modeling and Testing Analog and Mixed-Signal Devices. In: INTERNATIONAL TEST CONFERENCE, 1990, Washington DC, USA. **Proceedings...** Washington DC, USA: International Test Conference Press, 1990, p. 169 – 176.

SRIMANI S.; GHOSH K.; RAHAMAN H. Parametric Fault Detection in Analog Circuits : A Statistical Approach. Nov 2016.

STRATIGOPOULOS, H. G.; SUNTER, S, Fast Monte Carlo-Based Estimation of Analog Parametric Test Metrics **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, v. 33, n. 12, dec. 2014, p. 1977-1990.

SUNTER S.; JURGA K.; DINGENEN P.; VANHOOREN R. Practical random sampling of potential defects for analog fault simulation. **IEEE International Test Conference**, 2014.

TAUR, Y. *et al.* CMOS Scaling into the Nanometer Regime. **Proceedings of the IEEE**. New York, USA: IEEE, v. 85, n.4, apr. 1997, p. 486 – 504.

TUINHOUT, H. P. “Electrical characterization of matched pairs for evaluation of integrated circuit technologies,” Ph.D. dissertation, Dept. Math. Comput. Sci., Delft Univ. Technol., Delft, The Netherlands, 2005.

WESTE, N. H.; HARRIS, D. M. CMOS VLSI Design: A Circuits and Systems Perspective. 4. ed. [S.l.]: Addison-Wesley, 2010. ISBN 0-321-54774-8.

XU, G.; EMBABI, S. H. K. A Systematic Approach in Constructing Fully Differential Amplifiers. **IEEE Transactions on Circuits and Systems: Analog and Digital Signal Processing**, v. 47, n. 11, p. 1343 – 1347, nov. 2000.

ZILCH, Lucas Bernardo. **Desenvolvimento de uma Toolbox em MATLAB para Determinação de Configurações Otimizadas para Testes de Circuitos Analógicos Através de Simulações SPICE Automatizadas**. 86 f. TCC (Graduação) - Curso de Engenharia Elétrica, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2017.

APÊNDICE A – DESCRIÇÃO SPICE DO FDA DE UM ESTÁGIO EM MALHA ABERTA

```
.lib '<path><to>\xh018\lp3mos\xh018.lib' tm
.lib '<path><to>\xh018\lp3mos\param.lib' 3s
***** CMFB Block *****

.subckt cmfb_rasoul vop von vcm vdd vss vout
x1 3 vop 1 1 pe l=0.8918u w=35.91u delvto=0
x2 3 von 2 2 pe l=0.8918u w=35.91u delvto=0
x3 vout vcm 1 1 pe l=0.8918u w=35.91u delvto=0
x4 vout vcm 2 2 pe l=0.8918u w=35.91u delvto=0
x5 vout 3 vss vss ne l=0.4628u w=7.19u delvto=0
x6 3 3 vss vss ne l=0.4628u w=7.19u delvto=0
I1 vdd 1 15.19u
I2 vdd 2 15.19u
.ends

***** FDA Block Including CMFB *****

.subckt fd_cmfb vip vin vop von vdd vss vcm vcmc vcmc1
x1 von vip 1 1 ne w=36.29u l=0.197u delvto=0
x2 vop vin 1 1 ne w=36.29u l=0.197u delvto=0
x3 von 2 vdd vdd pe w=27.41u l=6.83u delvto=0
x4 vop 2 vdd vdd pe w=27.41u l=6.83u delvto=0
x5a 1 v1 vss vss ne w=15.3u l=9.75u delvto=0
x5b 1 vcmc vss vss ne w=15.3u l=9.75u delvto=0
x6 2 v1 vss vss ne w=15.3u l=9.75u delvto=0
x7 2 2 vdd vdd pe w=27.41u l=6.83u delvto=0
V1 v1 0 -167.045m
V2 vcmc vcmc1 -167.045m
Xcmfb vop von vcm vdd vss vcmc1 cmfb_rasoul
.ends

***** Test Setup *****

vdd vdd 0 0.9
```

```
vss vss 0 -0.9
vcm vcm 0 0
x1 vip vin vop von vdd vss vcm vcmc vcml fd_cmfb
vdiff vip vin dc 0 ac 1m
vmean vin 0 0
c1 von 0 10p
c2 vop 0 10p
.ac dec 400 1e1 1e11
.options ingold=2 abstol=1E-12 runlvl=6 numdgt=10 post=1
.end
```