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

LUCAS BERNARDO ZILCH

Ferramenta Automática para Geração de Conjuntos de Testes e Avaliação de DfT em Circuitos Analógicos Integrados

Proposta de tese apresentada como requisito parcial para a obtenção de grau de Doutor em Microeletrônica.

Orientador: Prof. Dr. Tiago Roberto Balen

Co-orientador: Prof. Dr. Marcelo Soares Lubaszewski

Porto Alegre

Dezembro de 2025

CIP - Catalogação na Publicação

Zilch, Lucas Bernardo
Ferramenta Automática para Geração de Conjuntos de Testes e Avaliação de DfT em Circuitos Analógicos Integrados / Lucas Bernardo Zilch. -- 2025.
141 f.
Orientador: Tiago Roberto Balen.

Coorientador: Marcelo Soares Lubaszewski.

Tese (Doutorado) -- Universidade Federal do Rio Grande do Sul, Instituto de Informática, Programa de Pós-Graduação em Microeletrônica, Porto Alegre, BR-RS, 2025.

1. Teste de circuitos integrados analógicos. 2. Geração automática de vetores de teste. 3. Dicionário de falhas. 4. Cobertura de falhas. 5. Design for testability. I. Balen, Tiago Roberto, orient. II. Lubaszewski, Marcelo Soares, coorient. III. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da UFRGS com os dados fornecidos pelo(a) autor(a).

"Everything that man ignores does not exist for him. That is why the universe of each one is summed up to the size of their knowledge"

-ALBERT EINSTEIN

AGRADECIMENTOS

Agradeço à toda a minha família, em especial à minha mãe Regina, ao meu pai Antenor (em memória), à minha irmã Daniela e à minha namorada Karine 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.

Agradeço ao aluno de mestrado Luis Augusto Duarte pela colaboração para realização do estudo de caso descrito pela Seção 4.4, sendo ele o responsável pela montagem do circuito físico, emulação das falhas e medições experimentais descritas na seção 4.4.2.

RESUMO

Este trabalho apresenta o desenvolvimento, aprimoramento e validação de uma ferramenta computacional para geração automática e otimização de conjuntos de testes aplicados à detecção de falhas em circuitos integrados analógicos. A ferramenta, originalmente implementada em MATLAB, foi totalmente reestruturada em Python, incorporando novas funcionalidades que ampliam significativamente sua capacidade analítica. Dentre as melhorias implementadas destacam-se: (i) inclusão da medição de correntes em testes AC (*Alternating Current*), DC (*Direct Current*) e transientes; (ii) suporte simultâneo a módulo e fase em análises AC; (iii) mecanismos de combinação de testes complementares (AC módulo/fase e transientes de *overshoot*/tempo de acomodação); e (iv) avaliação automatizada de testes transientes derivados de dois testes DC previamente selecionados. O algoritmo de seleção foi também reformulado para reduzir o tempo de busca e melhorar sua escalabilidade. Vários estudos de caso foram conduzidos para validar e comparar as versões MATLAB e Python. O primeiro consiste em um filtro Butterworth de segunda ordem, implementado com um amplificador operacional totalmente diferencial em tecnologia CMOS (*Complementary Metal Oxide Semiconductor*) de 180 nm. Utilizando dois testes AC, dois DC e um teste transitório, a ferramenta obteve uma cobertura de falhas simulada de 96,87%. No segundo estudo, a campanha de injeção de falhas considerou uma abordagem baseada em testes funcionais, com cobertura de falhas de apenas 78,13%, demonstrando a superioridade da abordagem estrutural automatizada. O terceiro estudo avalia a aplicação de técnicas de DfT (*Design-for-Testability*) em um filtro de dois estágios, também baseado no mesmo amplificador operacional. Foi observado que, através de técnicas de DfT, a cobertura de falhas poderia ser aumentada de 87,15% para 95,83%. O quarto estudo de caso compara os resultados de simulação da ferramenta com um experimento físico de injeção de falhas em hardware, baseado na emulação do circuito, com resultados de injeção física de 48 falhas em hardware, obtendo 41 falhas detectadas experimentalmente, em comparação a 43 falhas previstas em simulação, demonstrando consistência entre o comportamento real e o simulado. Além disso, as novas funcionalidades da ferramenta desenvolvida em Python foram utilizadas em três circuitos do conjunto de benchmarks IEEE P2427. Para o amplificador operacional (OPAMP1), a cobertura de falhas passou de 77,53% (somente tensão de saída) para 96,63% com a inclusão de medições de corrente e testes transientes, utilizando apenas seis testes no conjunto final. Para o regulador de tesão (VREG1), o uso exclusivo de medições de tensão levou a uma cobertura de 97,09% com cinco testes; ao incluir medições de corrente, a ferramenta identificou um conjunto reduzido para apenas três testes, alcançando 100% de cobertura das 103 falhas avaliadas. Para o comparador de tensão (COMP1), a cobertura inicial de 72,36% (considerando apenas medições de tensão de saída) foi ampliada para 94,31% com três testes transientes baseados exclusivamente em medições de corrente. Uma verificação exaustiva demonstrou que nenhuma combinação de até quatro testes superou o conjunto determinado automaticamente pelo algoritmo, confirmando sua eficácia do algoritmo de escolha. Os resultados obtidos demonstram a robustez, a flexibilidade e a eficiência da nova versão da ferramenta, bem como seu potencial de aplicação em diferentes cenários de teste de circuitos integrados analógicos, contribuindo para o avanço das técnicas de detecção de falhas e para a automação do processo de geração de testes.

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

ABSTRACT

This work presents the development, enhancement, and validation of a computational tool for the automatic generation and optimization of test sets applied to fault detection in analog integrated circuits. The tool, originally implemented in MATLAB, was fully restructured in Python, incorporating new functionalities that significantly expand its analytical capabilities. The main improvements include: (i) support for current measurements in AC (Alternating Current), DC (Direct Current) and transient tests; (ii) simultaneous support for magnitude and phase in AC analyses; (iii) mechanisms for combining complementary tests (AC magnitude/phase and transient overshoot/settling time); and (iv) automated assessment of transient tests derived from pairs of previously selected DC tests. The test-selection algorithm was also redesigned to reduce search time and improve scalability. Several case studies were conducted to validate and compare the MATLAB and Python versions of the tool. The first involves a second-order Butterworth filter implemented with a fully differential operational amplifier in 180-nm CMOS (Complementary Metal Oxide Semiconductor) technology; using two AC tests, two DC tests, and one transient test, the tool achieved a simulated fault coverage of 96.87%. In the second case study, a functional-test-based campaign obtained only 78.13% fault coverage, demonstrating the superiority of the automated structural approach. The third study evaluates the application of DfT techniques to a two-stage filter based on the same operational amplifier, where fault coverage increased from 87.15% to 95.83% through DfT (Design-for-Testability) insertion. The fourth case study compares simulation results with a physical fault-injection experiment using hardware emulation: among 48 injected faults, 41 were detected experimentally versus 43 predicted in simulation, showing strong consistency between real and simulated behavior. In addition, the new functionalities implemented in Python were applied to three circuits from the IEEE P2427 benchmark set. For the operational amplifier (OPAMP1), fault coverage increased from 77.53% (using only output-voltage measurement) to 96.63% with the inclusion of current measurements and transient tests, using only six tests in the final set. For the voltage regulator (VREG1), exclusive use of voltage measurements yielded 97.09% coverage with five tests; by incorporating current measurements, the tool identified a reduced set of only three tests achieving 100% coverage of the 103 evaluated faults. For the voltage comparator (COMP1), the initial coverage of 72.36% (using only output-voltage monitoring) rose to 94.31% with three transient tests based solely on current measurements. An exhaustive verification demonstrated that no combination of up to four tests surpassed the set determined automatically by the algorithm, confirming the effectiveness of the test-selection algorithm. The results confirm the robustness, flexibility, and efficiency of the new Python-based version of the tool, as well as its potential applicability to a wide range of analog-circuit testing scenarios. The work contributes to advancing fault-detection techniques and to the automation of test-set generation for analog integrated circuits.

Keywords: Analog integrated circuits, testing, fault dictionary, test vectors, fault coverage, automatic test vector generation, design for testability.

LISTA DE FIGURAS

Figura 1 - Estimativa de custo por falha detectada em cada etapa de produção. 24
Figura 2 - *Corners* de uma tecnologia. 34
Figura 3 - Diagrama em blocos do verificador proposto. 37
Figura 4 - Bloco conceitual do circuito CMFB do FDA. 40
Figura 5 – Diagrama de blocos geral da ferramenta para análise de configurações de teste em circuitos analógicos. 43
Figura 6 - Fluxograma de funcionamento da janela "AnalogCircuitsTest". 44
Figura 7 - Primeira etapa da Janela "AnalogCircuitsTest". 45
Figura 8 - Exemplo de arquivo de descrição de falhas. 46
Figura 9 - Escolha do arquivo de descrição de falhas. 48
Figura 10 - Inserção de parâmetros de simulação: (a) Visão inicial antes da seleção dos tipos de simulação. (b) Visão após a seleção consecutiva dos três tipos de simulação (AC, DC e TRAN). 50
Figura 11 – Limites de Detecção. 51
Figura 12 - Exemplo de matriz de falhas. 56
Figura 13 - Representação do algoritmo para determinação do melhor conjunto de testes. 62
Figura 14 - Janela para execução de análise transiente. 64
Figura 15 - Janela "Results" para o caso dos testes DC e AC de um amplificador. 68
Figura 16 - Janela para consulta dos modelos de falhas utilizados. 69
Figura 17 - Gráfico com a resposta dos *corners* aberto através do botão "*View Corners*" para nó “Vout”. 70
Figura 18 - Janela para alteração dos nós de teste. 71
Figura 19 - Diagrama esquemático do filtro passa baixas com múltiplo *feedback*. 78
Figura 20 - Esquemático do amplificador totalmente diferencial de um estágio. 79
Figura 21 - Esquemático elétrico do circuito CMFB. 80
Figura 22 - Diagrama esquemático do filtro passa-baixas de dois estágios. 84
Figura 23 - Diagrama esquemático do modo de teste com os FDAs em malha abeta. 86
Figura 24 – Circuito interno do CD4007. 88
Figura 25 – Esquemático do amplificador operacional implementado. 89
Figura 26 – OPAMP1 amplificador operacional básico disponibilizado pelo conjunto de benchmarks do P2427. 93

Figura 27 – VREG1 regulador de tensão disponibilizado pelo conjunto de benchmarks do P2427. 95
Figura 28 – COMP1 Comparador de alta velocidade com pino de habilitação (pd) disponibilizado pelo conjunto de benchmarks do P2427. 97
Figura 29 – Degrau de tensão aplicado às entradas do comparador COMP1 para testes transientes. 98
Figura 30 – Parâmetros configurados para simulação transiente do COMP1. 99
Figura 31 – Resposta aos diferentes sinais DC de entrada para circuito sem falhas e para o caso de curto-circuito no resistor R2A. 103
Figura 32 – Número de falhas detectáveis em função da tensão DC aplicada na entrada. 104
Figura 33 - Resposta aos diferentes sinais AC de entrada para circuito sem falhas e para o caso de curto-circuito no resistor R2A. 105
Figura 34 – Número de falhas detectáveis em função da tensão AC aplicada na entrada. 105
Figura 35 – Falha adicional detectada pelo teste transiente. 106
Figura 36 – *Slew Rate* do nó Vo+. 107
Figura 37 - *Slew Rate* do nó Vo-. 107
Figura 38 – Ganho de tensão para nó Vo+. 108
Figura 39 - Ganho de tensão para nó Vo-. 108
Figura 40 – Frequência de -3dB para nó Vo+. 109
Figura 41 - Frequência de -3dB para nó Vo-. 109
Figura 42 – Números de falhas detectáveis nos nós de saída do filtro passa-baixas de dois estágios em função de sinais DC e AC aplicados na entrada. 111
Figura 43 – Número de falhas detectáveis nos nós de saída para amplificador totalmente diferencial em malha aberta em função de sinais DC e AC aplicados na entrada. 113
Figura 44 – Componentes internos do amplificador totalmente diferencial. 115
Figura 45 – Número de falhas detectáveis através da medição da amplitude de tensão no nó de saída em função da frequência. 120
Figura 46 – Número de falhas detectáveis através da medição das amplitudes das correntes de alimentação do circuito, de entrada e nos pinos pd e xpd em função da frequência. 120
Figura 47 – Número de falhas detectáveis através da medição das correntes de alimentação do circuito, entrada, pino pd e xpd em função de tensões DC na entrada. 121
Figura 48 – Parâmetros para simulação de teste transiente adicional. 122
Figura 49 – Número de falhas detectáveis através da medição da amplitude de tensão no nó de saída em função da frequência. 125

Figura 50 – Número de falhas detectáveis através da medição da amplitude da corrente de alimentação do circuito em função da frequência. 125

LISTA DE TABELAS

Tabela 1 - Especificações do projeto do circuito FDA. 81
Tabela 2 - Parâmetros de dimensionamento dos transistores. 89
Tabela 3 - Tempos de execução do algoritmo. 101
Tabela 4 – Melhor conjunto de testes. 103
Tabela 5 – Resultados testes funcionais. 106
Tabela 6 – Melhor conjunto de testes filtro completo. 110
Tabela 7 – Melhor conjunto de testes amplificador em malha aberta. 112
Tabela 8 – Melhor conjunto de testes amplificador com CD4007. 116
Tabela 9 – Conjunto de testes II OPAMP1 considerando a possibilidade de medição da tensão no nó de saída, corrente de alimentação, corrente de entrada e corrente nos pinos pd e xpd. 117
Tabela 10 – Conjunto de testes I do OPAMP1 considerando apenas medição da tensão no nó de saída. 119
Tabela 11 – Conjunto de testes II OPAMP1 considerando a possibilidade de medição da tensão no nó de saída, corrente de alimentação, corrente de entrada e corrente nos pinos pd e xpd. 119
Tabela 12 – Conjunto de testes II OPAMP1 considerando a execução de testes transientes adicionais. 123
Tabela 13 – Melhor conjunto de testes OPAMP1 considerando apenas medição da tensão no nó de saída. 123
Tabela 14 – Melhor conjunto de testes OPAMP1 considerando apenas medição da tensão no nó de saída e correntes de alimentação e de entrada. 124
Tabela 15 – Melhor conjunto de testes COMP1 considerando apenas medição da tensão no nó de saída. 126
Tabela 16 – Melhor conjunto de testes COMP1 considerando medição da tensão no nó de saída e corrente nos pinos de entrada, pd, xpd, vref e vdd. 127
Tabela 17 – Resultados das avaliações exaustivas de todas as possíveis combinações de 2, 3 e 4 testes. 128

LISTAS DE ABREVIATURAS E SIGLAS

AC	*Alternating Current*
ADC	*Analog to digital converter*
ATE	*Automatic Test Equipment*
ATPG	*Automatic Test-Pattern Generation*
BIST	*Built-In Self-Test*
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-Testability*
FDA	*Fully Differential Amplifier*
FF	*Fast-Fast*
FFT	*Fast Fourier Transform*
FS	*Fast-Slow*
GBW	*Gain-Bandwidth product*
MF	Matriz de Falhas
NMOSFET	*N-channel Metal-Oxide-Semiconductor Field-Effect Transistors*
PLL	*Phase-Locked Loops*
PMOSFET	*P-channel Metal-Oxide-Semiconductor Field-Effect Transistors*
preHRL	*Higher Relative Likelihood*
preLRL	*Lower Relative Likelihood*
RC	Resistivo Capacitivo
RF	Rádio Frequência
SF	*Slow-Fast*
SMS	*Short Message Service*
SNR	*Signal-to-Noise Ratio*
SPICE	*Simulation Program with Integrated Circuits Emphasis*
SS	*Slow-Slow*

TRAM	*Transient Response Analysis Method*
TT	*Typical-Typical*

SUMÁRIO

1 INTRODUÇÃO 15

2 REVISÃO BIBLIOGRÁFICA 21

2.1 TESTES DE CIRCUITOS ANALÓGICOS 21

2.1.1 Teste em Produção 23

2.1.2 Teste Funcional *versus* Teste Estrutural 25

2.1.3 Geração de Teste 26

2.1.4 Modelo de Falhas Para Teste de Circuitos Integrados Analógico 28

2.1.5 Dicionário de Falhas 29

2.1.6 Técnicas de *Design-for-Testability* (DfT) 31

2.2 CARACTERIZAÇÃO ELÉTRICA DAS VARIABILIDADES DO PROCESSO DE FABRICAÇÃO 32

2.2.1 Análise de *Corners* 34

2.2.2 Análise de Monte Carlo 35

2.3 METODOLOGIAS DE TESTE UTILIZADAS NESSE TRABALHO 36

2.3.1 Teste DC 36

2.3.2 Teste AC 37

2.3.3 Teste Transiente 38

2.4 AMPLIFICADORES TOTALMENTE DIFERENCIAIS 38

3 DESCRIÇÃO DA FERRAMENTA 41

3.1 JANELA “ANALOGCIRCUITTEST” 43

3.1.1 Etapa 1: Seleção de Arquivos para Simulação 44

3.1.2 Etapa 2: Escolha das Falhas que Irão Compor o Modelo. 45

3.1.3 Etapa 2: Parâmetros de Simulação 49

3.2 JANELA “RESULTS” 57

3.2.1 Determinação da Máxima Cobertura de Falhas 58

3.2.2 Determinação do Melhor Conjunto de Testes 59

3.2.3 Análises de Possíveis Combinações de Teste no Conjunto Final 63

3.2.4 Exibição dos Resultados 67

3.2.5 Possibilidades de Alterações das Análises 70

3.3 SIMULAÇÕES DE TESTES FUNCIONAIS 72

3.4 EXPORTAÇÃO DA FERRAMENTA DE MATLAB PARA PYTHON 73

4 METODOLOGIA EXPERIMENTAL – ESTUDOS DE CASO 77

4.1 FILTRO BUTTERWORTH DE UM ESTÁGIO 78

4.1.1 Testes AC e DC 82
4.1.2 Teste Transiente Adicional 83
4.2 DETECÇÃO DE FALHAS ATRAVÉS DE TESTES FUNCIONAIS 83
4.3 FILTRO PASSA-BAIXAS DE DOIS ESTÁGIOS 84
4.3.1 Análises AC e DC do Filtro Passa-Baixas Completo 85
4.3.2 Amplificador Totalmente Diferencial em Malha Aberta 86
4.3.3 Utilização Adicional de um Nó Interno Do Amplificador Como Nó de Teste 87
4.4 VALIDAÇÃO DA FERRAMENTA ATRAVÉS DE EMULAÇÃO DE FALHAS POR HARDWARE 87
4.4.1 Simulação de Configurações de Teste 88
4.4.2 Medição Experimental 90
4.5 GRUPO P2427 E BENCHMARK DE CIRCUITOS ANALÓGICOS 91
4.5.1 OPAMP1 92
4.5.2 VREG1 94
4.5.3 COMP1 96
5 RESULTADOS E DISCUSSÕES 101
5.1 FILTRO BUTTERWORTH DE UM ESTÁGIO 102
5.1.1 Testes AC e DC 102
5.1.2 Teste Transiente Adicional 105
5.2 DETECÇÃO DE FALHAS ATRAVÉS DE TESTES FUNCIONAIS 106
5.3 FILTRO PASSA-BAIXAS DE DOIS ESTÁGIOS 110
5.3.1 Análises AC e DC do Filtro Passa-Baixas Completo 110
5.3.2 Amplificador Totalmente Diferencial em Malha Aberta 111
5.3.3 Utilização Adicional de um Nó Interno do Amplificar Como Nó de Teste 114
5.4 VALIDAÇÃO DA FERRAMENTA ATRAVÉS DE EMULAÇÃO DE FALHAS POR HARDWARE 115
5.4.1 Simulação de Configurações de Teste 116
5.4.2 Medição Experimental 116
5.5 GRUPO P2427 E BENCHMARK DE CIRCUITOS ANALÓGICOS 118
5.5.1 OPAMP1 118
5.5.2 VREG1 123
5.5.3 COMP1 125
6 CONCLUSÕES 129
7 SUGESTÕES PARA TRABALHOS FUTUROS 132
REFERÊNCIAS 133

1 INTRODUÇÃO

Um marco fundamental para o atual panorama de desenvolvimento tecnológico é a invenção do primeiro transistor, ocorrida há aproximadamente 70 anos. Em um intervalo de apenas sete décadas, a humanidade testemunhou uma notável evolução, partindo dos transistores iniciais, encapsulados individualmente, até os Circuitos Integrados (CIs) contemporâneos, capazes de conter bilhões desses componentes interconectados em um único *chip*. Essa rápida progressão foi impulsionada pela significativa redução das dimensões dos transistores, conforme previsto pela Lei de Moore (SEGURA; HAWKINS, 2004; ARSLAN; ORAILOGLU, 2013), a qual postula que a capacidade de integração dos transistores em um circuito integrado digital duplica a cada 18 meses (MOORE, 1965). O cenário atual mostra que a indústria da microeletrônica alcançou patamares gigantescos, proporcionando circuitos eletrônicos extremamente complexos em termos funcionais e eficientes em termos energéticos, em comparação com seus equivalentes do passado (YURISH, 2025). Hoje, um *smartphone* compacto o suficiente para caber no bolso possibilita uma gama diversificada de atividades, que vão desde as antigas mensagens SMS (*Short Message Service*) até o acesso à internet para videochamadas, transações comerciais, consultas bancárias, entre outras funcionalidades.

Entretanto, à medida que as funcionalidades dos circuitos integrados avançam, o processo de projetá-los e fabricá-los torna-se proporcionalmente mais complexo, exigindo um nível crescente de precisão e controle. As tecnologias atuais de fabricação de CIs operam em escalas nanométricas, onde a minimização das dimensões dos componentes introduz desafios significativos no controle dos processos produtivos, tornando essencial o estudo detalhado de suas variabilidades (QUIAN, 2015). Com a redução expressiva do tamanho dos transistores, as variações nos processos de fabricação tornam-se mais relevantes (SHIN, 2016), o que requer que sejam estudadas cada vez mais (MEZZOMO *et al*., 2011). Variações essas que acabam agravando efeitos como o de canal curto, que resulta na diminuição da tensão de limiar (*threshold voltage*), além do aumento exponencial das correntes de fuga, consequência direta da redução da espessura do óxido (TAUR et al., 1997).

Todos esses fatores ressaltam a importância de uma etapa fundamental no desenvolvimento de circuitos eletrônicos: a fase de teste. Com os desafios cada vez maiores no projeto e na fabricação de CIs, garantir seu funcionamento correto torna-se não apenas essencial, mas também altamente desafiador. Em aplicações de alta segurança, como em automóveis, aviões, trens de alta velocidade e usinas nucleares, o nível de segurança funcional exigida é altíssimo. Portanto, métodos extremamente confiáveis para a detecção de defeitos nos *chips* (LUBASZEWSKI *et al*., 2000) e sistemas são necessários. Nesse contexto, a busca

contínua por aprimorar a qualidade dos testes aplicados a circuitos integrados recém-fabricados vem se tornando uma tarefa cada vez mais complexa (KABISATPATHY; BARUA; SINHA, 2005). Essa qualidade está diretamente relacionada aos conceitos de cobertura de falhas e *yield loss*, além da necessidade de minimizar os custos de execução dos testes, uma vez que esses custos são incorporados ao preço final do produto e podem afetar sua competitividade no mercado.

A implementação de circuitos integrados digitais é, na maioria dos casos, preferida em relação aos circuitos analógicos, principalmente devido à sua maior simplicidade de projeto e maior previsibilidade em diversas aplicações. Isso se deve ao fato de que os circuitos digitais operam com um número finito de estados bem definidos, enquanto os circuitos integrados analógicos apresentam uma gama infinita de valores de saída possíveis, tornando os seus comportamentos mais complexos e sujeitos a variações (SEDRA; SMITH, 2010). Além disso, os circuitos digitais oferecem maior imunidade a ruídos e menor susceptibilidade a variações de temperatura e tolerâncias dos componentes, características que contribuem para sua ampla adoção em sistemas elétricos modernos (BAKER, 2019).

No entanto, apesar dessas vantagens, existem aplicações onde o uso de circuitos analógicos é imprescindível, pois não há uma alternativa digital equivalente capaz de desempenhar a mesma função. Um exemplo clássico é o interfaceamento entre circuitos eletrônicos e variáveis físicas do ambiente, uma vez que o mundo real é, por natureza, analógico. Sensores de temperatura, pressão, luz e som, por exemplo, produzem sinais contínuos que precisam ser processados adequadamente antes de qualquer conversão digital.

Nesse contexto, os circuitos amplificadores desempenham um papel fundamental, sendo essenciais para o condicionamento de sinais analógicos antes de sua digitalização. Amplificadores operacionais, por exemplo, são amplamente utilizados na amplificação e filtragem de sinais provenientes de sensores, garantindo que a informação captada seja tratada de forma eficiente antes de ser amostrada por um conversor analógico-digital (*Analog to digital converter* - ADC) (HOROWITZ; HILL, 2015). Além disso, amplificadores são componentes indispensáveis na implementação de filtros ativos, que são cruciais para eliminar ruídos indesejados e garantir a integridade dos sinais processados. Como consequência, praticamente todas as aplicações que envolvem a conversão de sinais analógicos para digitais dependem diretamente de circuitos amplificadores para otimizar a qualidade do sinal antes da conversão.

Circuitos integrados digitais e analógicos também apresentam diferenças significativas no que diz respeito aos métodos de teste e validação. Nos testes de circuitos integrados digitais, cada conjunto específico de sinais de entrada deve gerar um vetor de saída correspondente e

bem definido, de modo que qualquer discrepância, mesmo que seja de um único bit, pode indicar a presença de um defeito no circuito. Esse tipo de teste permite a aplicação de métodos altamente automatizados e eficientes, como *scan testing* e *Built-In Self-Test* (BIST), que aumentam a confiabilidade da detecção de falhas em larga escala (ZHOU *et al*., 2024).

Por outro lado, o teste de circuitos integrados analógicos é consideravelmente mais desafiador, pois a resposta esperada para uma determinada entrada não é um valor exato, mas sim um intervalo de valores dentro de uma faixa de tolerância aceitável. Fatores como variações nos parâmetros dos componentes, ruído e temperatura podem impactar os resultados dos testes, tornando mais difícil distinguir um circuito funcional de um defeituoso. Além disso, enquanto nos circuitos digitais é possível utilizar métodos como *fault simulation* para prever falhas específicas, nos circuitos analógicos a modelagem de defeitos é mais complexa, pois envolve a degradação gradual de parâmetros elétricos em vez de falhas binárias absolutas (KARTHI; KAVITHA, 2024).

Outra grande diferença está nos equipamentos e metodologias utilizadas para testes. Em circuitos integrados digitais, testadores automáticos de alto desempenho podem rapidamente validar milhares de padrões de entrada e saída. Já no caso dos circuitos integrados analógicos, os testes costumam ser mais demorados e exigem equipamentos de medição de alta precisão, como analisadores de espectro, testadores de resposta em frequência e placas de aquisição com elevado poder de processamento digital de sinais, aumentando significativamente o custo e a complexidade do processo (IEEE, 2023).

Devido a essa diferença fundamental entre circuitos integrados digitais e analógicos, é comum que os circuitos integrados analógicos sejam testados por meio da metodologia de teste baseado em especificações. Nesse método, o objetivo principal não é verificar diretamente a tensão de saída em resposta a um conjunto fixo de estímulos, como ocorre nos circuitos integrados digitais, mas sim garantir que todas as especificações descritas no *datasheet* sejam atendidas dentro das faixas de tolerância aceitáveis. Cada especificação é analisada individualmente, avaliando parâmetros como ganho de tensão, resposta em frequência, distorção harmônica total e consumo de energia elétrica, garantindo que o circuito integrado funcione corretamente dentro das condições estabelecidas pelo fabricante. Embora essa abordagem proporcione testes de alta qualidade e confiabilidade (STRATIGOPOULOS; SUNTER, 2014), sua implementação tem um custo significativamente elevado se comparado ao teste de um circuito integrado digital que ocupe a mesma área no *chip* (POEHL *et al*., 2010).

Na busca por métodos de teste que combinem alta qualidade com menor custo para CIs, uma abordagem recomendada é a do teste estrutural (RENOVELL, 2004). Diferente do teste

baseado em especificações, que avalia diretamente o desempenho funcional do circuito integrado, o teste estrutural se baseia na criação de um modelo de falhas que representa possíveis defeitos no circuito em análise. A partir desse modelo, são realizadas simulações computacionais para gerar um dicionário de falhas, no qual se registra a relação entre determinados padrões de entrada e os comportamentos de saída esperados em caso de defeitos.

No contexto dos circuitos analógicos, a definição dos vetores de teste, isto é, os sinais de entrada utilizados para detectar falhas, é feita, de modo geral, de maneira empírica. 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. 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.

Foi dentro desse contexto que surgiu a proposta do presente trabalho, cujo principal objetivo foi desenvolver uma ferramenta automatizada capaz de analisar um grande número de possíveis configurações de teste para circuitos integrados analógicos. Através dessa análise, a ferramenta busca identificar as melhores configurações de teste, otimizando os custos e, ao mesmo tempo, garantindo um nível de qualidade elevado dentro das condições estipuladas. Além disso, a ferramenta foi projetada para permitir uma reanálise rápida dos dados já processados, oferecendo flexibilidade ao usuário para ajustar as condições de teste e explorar novos cenários de simulação.

O desenvolvimento dessa ferramenta de automação teve início com os trabalhos de Chinazzo (2016), que elaborou uma primeira versão de códigos em MATLAB para a realização de simulações automatizadas utilizando o software SPICE (*Simulation Program with Integrated Circuits Emphasis*). Essa versão inicial possibilitou a geração e simulação de circuitos com falhas pré-definidas, permitindo a avaliação de diferentes configurações de teste.

No entanto, essa primeira implementação apresentava algumas limitações. Todas as análises eram realizadas sem uma interface gráfica, o que dificultava a interação do usuário com o sistema. Os parâmetros de simulação precisavam ser configurados diretamente no código-fonte, sem uma interface intuitiva que permitisse ajustes dinâmicos. Além disso, os resultados eram apresentados apenas na janela de comando, sem a possibilidade de visualização gráfica ou exportação de dados para reanálise e manipulação de parâmetros.

Em outro trabalho (ZILCH, 2021), foram expandidas as funcionalidades existentes e foi criada uma *toolbox* em MATLAB, que proporciona uma interface amigável e intuitiva entre o usuário e o algoritmo de automatização. A nova versão da ferramenta possibilita um acesso

facilitado aos resultados de simulações já feitas, permitindo que os usuários interajam de maneira mais direta com o algoritmo sem a necessidade de modificações no código-fonte.

Além disso, novas funcionalidades foram integradas ao sistema, aprimorando a análise e o processamento dos testes. Um dos principais avanços foi a capacidade de analisar múltiplos sinais de entrada em casos de tensões DC (*Direct Current*), o que amplia a aplicabilidade da ferramenta a um maior número de cenários de teste. Também foram desenvolvidas novas lógicas para a determinação dos limites de detecção de falhas, com a incorporação do método de Monte Carlo entre as opções de simulação.

Outro avanço importante foi a implementação de uma função que permite a reanálise rápida dos resultados previamente gerados. Essa funcionalidade proporciona uma revisitação ágil dos dados sem a necessidade de refazer todo o processo de simulação, o que torna o ciclo de análise mais eficiente e menos custoso em termos de tempo de processamento. Com isso, o usuário pode ajustar facilmente os parâmetros de teste e obter novos resultados sem interrupções significativas no fluxo de trabalho.

Em relação aos resultados obtidos, uma lógica foi desenvolvida para a determinação do melhor conjunto de testes, levando em consideração não apenas os parâmetros de desempenho, mas também os custos associados e a abrangência da detecção de falhas. Essa estratégia assegura que os testes selecionados maximizem a qualidade dos resultados enquanto minimizam os custos e a complexidade do processo. Além disso, a ferramenta agora permite que o usuário modifique as condições de teste de forma dinâmica. A seleção de novos nós de teste, a alteração dos limites de detecção e a restrição dos sinais de entrada podem ser facilmente ajustadas. Quando essas mudanças são feitas, o algoritmo recalcula automaticamente os resultados para refletir as novas condições, garantindo que os testes sejam sempre adaptados às necessidades específicas de cada análise.

Essas melhorias não só otimizaram a funcionalidade do sistema, mas também o tornaram uma ferramenta mais poderosa e acessível para a realização de testes em circuitos analógicos. A flexibilidade no ajuste de parâmetros e a automatização do processo de análise são componentes-chave que tornam essa ferramenta uma solução robusta para enfrentar os desafios complexos do teste de circuitos analógicos, oferecendo um avanço significativo em relação às abordagens tradicionais de teste baseadas em especificações e tentativa e erro.

Utilizando os códigos desenvolvidos por Chinazzo (2016) como base, a proposta deste trabalho, assim como desenvolvido inicialmente em Zilch (2017) e Zilch (2021), foi melhorar a metodologia, expandir as funcionalidades e obter uma interface mais amigável entre o usuário e o algoritmo de automatização. Novas funções foram incorporadas, incluindo a capacidade de

analisar múltiplos sinais de entrada para tensões DC, a implementação de novas lógicas para a determinação dos limites de detecção de falhas, com destaque para a inclusão do método de Monte Carlo, e a possibilidade de uma rápida releitura dos resultados previamente determinados. Além disso, foi desenvolvida uma lógica para identificar o melhor conjunto de testes e possibilitar ajustes nas condições de teste, como a seleção de novos nós, alteração dos limites de detecção e restrição dos sinais de entrada possíveis. Isso permite que o algoritmo recalcule os resultados de acordo com as novas condições estabelecidas.

Dentre as contribuições deste trabalho, destacam-se o desenvolvimento de uma ferramenta automatizada em Python para a geração e seleção de testes estruturais aplicados a circuitos integrados analógicos, a incorporação de metodologias baseadas na variabilidade do processo de fabricação para a determinação automática dos limites de aceitação, a integração de medições de tensão e corrente em testes DC, AC (*Alternating Current*) e transientes, incluindo a possibilidade de análise simultânea da magnitude e da fase dos sinais, bem como o desenvolvimento de novas metodologias para a determinação do conjunto ótimo de testes, com foco na redução do tempo de execução e, consequentemente, do custo do teste. Adicionalmente, a ferramenta é validada por meio de estudos de caso simulados e experimentais, incluindo circuitos de referência amplamente utilizados na literatura. Essas contribuições visam avançar o estado da arte em teste de circuitos analógicos, ao fornecer uma abordagem mais sistemática, flexível e escalável para a geração e avaliação de conjuntos de teste.

O presente trabalho divide-se em sete capítulos. No Capítulo 2, é feita uma revisão bibliográfica sobre testes em circuitos integrados, dando ênfase aos analógicos, que são nosso objeto de estudo. A descrição geral da ferramenta 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.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo, será realizada uma revisão bibliográfica sobre conceitos essenciais para o entendimento deste trabalho. Inicialmente, serão abordados conceitos relacionados ao teste de circuitos analógicos, que é o principal foco do estudo. Dado que os circuitos integrados analógicos não apresentam uma distinção tão clara quanto os digitais entre um sinal de saída esperado e um sinal fora dos limites aceitáveis, é crucial discutir as variabilidades de processo e as metodologias de caracterização elétrica associadas. Esse exame permitirá a determinação dos limites de aceitação para as variáveis elétricas medidas durante o teste de circuitos integrados analógicos.

2.1 TESTES DE CIRCUITOS ANALÓGICOS

Na indústria de semicondutores, a integração de funções digitais e analógicas em circuitos integrados tem se tornado cada vez mais comum, refletindo as necessidades de sistemas eletrônicos modernos que exigem a combinação de ambos os tipos de circuitos para um desempenho otimizado. No entanto, enquanto as metodologias de teste para circuitos digitais são bem estabelecidas e amplamente utilizadas, as técnicas para testar circuitos integrados analógicos ainda estão em processo de evolução e carecem da adoção de uma padronização (IEEE STANDARDS ASSOCIATION, 2018; AZAM et al., 2024; GANJI et al., 2024).

As ferramentas computacionais para a geração de testes automatizados em circuitos integrados digitais estão em uso há mais de três décadas, possibilitando automação, otimização de vetores e redução substancial do custo de teste. No entanto, a otimização automática de sinais de teste para circuitos integrados analógicos ainda permanece limitada, frequentemente dependendo da experiência do projetista e de especificações funcionais do circuito, o que torna o processo mais subjetivo e suscetível a erros. Estudos recentes reforçam essa lacuna: trabalhos contemporâneos mostram que, apesar dos avanços em métodos assistidos por aprendizado de máquina, a definição de estímulos analógicos eficientes ainda não é totalmente automatizada e continua sendo um desafio aberto na área (WANG; TIAN; ZHANG, 2021).

A principal razão para essa diferença está relacionada à natureza dos próprios circuitos. Circuitos integrados digitais, com seu número finito de estados possíveis, permitem uma distinção clara entre circuitos falhos e funcionais, o que torna a detecção de falhas direta e fácil de implementar. Já os circuitos integrados analógicos não possuem essa definição precisa e têm uma gama de respostas contínuas, com variabilidade nos valores que podem ser aceitos, tornando o processo de diagnóstico muito mais complexo. Nos testes de circuitos analógicos, a

funcionalidade do circuito é avaliada com base em limites pré-estabelecidos, os quais determinam o quão bem o circuito opera dentro das especificações definidas, mas sem a clareza binária encontrada nos testes de circuitos digitais (KABISATPATHY; BARUA; SINHA, 2005).

Um dos passos cruciais no teste de circuitos integrados analógicos é a definição dos limites de aceitação, ou seja, os parâmetros que irão distinguir comportamentos aceitáveis dos não aceitáveis. A maioria dos testes de circuitos integrados analógicos segue a metodologia baseada em especificações definidas no *datasheet* do componente. Essa abordagem, embora amplamente adotada, exige um conhecimento profundo do circuito e pode resultar em tempos de execução prolongados, já que é necessário testar o circuito de forma complexa para garantir que todas as especificações estejam dentro das faixas de tolerância estabelecidas. Isso torna os testes de circuitos analógicos consideravelmente mais caros quando comparados aos testes de circuitos integrados digitais de mesma área no *chip* (STRATIGOPOULOS; SUNTER, 2014; POEHL *et al*., 2010). Um exemplo claro desse desafio pode ser observado em circuitos analógicos de RF (Rádio Frequência), que frequentemente exigem a verificação de uma vasta gama de especificações que não podem ser facilmente testadas com equipamentos básicos. Tais testes frequentemente necessitam de equipamentos automáticos de teste (*Automatic Test Equipment* -ATE) avançados, que são de alta resolução, com capacidade de processamento de sinais elevada, resposta para altas frequências e memória considerável para armazenar grandes volumes de dados de teste (BUSHNEL; AGRAWAL, 2002).

Para enfrentar a questão do alto custo e da complexidade envolvida no teste de circuitos integrados analógicos, uma alternativa recomendada é a metodologia de teste estrutural (RENOVELL, 2004; SACHDEV; DE GYVEZ, 2007). Esta abordagem propõe a criação de um modelo de falhas para o circuito em análise, a partir do qual se desenvolve um dicionário de falhas por meio de simulações. Esse modelo permite que os testes sejam conduzidos de maneira mais sistemática e menos dependente da intuição do projetista, otimizando o processo de verificação. No entanto, a determinação dos vetores de teste para os circuitos integrados analógicos ainda é um desafio, frequentemente realizada por tentativa e erro, o que impede a total automação do processo, como é possível nos circuitos digitais (KABISATPATHY; BARUA; SINHA, 2005). Mesmo com os avanços nas metodologias de teste para os circuitos integrados analógicos, os desafios em automatizar totalmente os testes persistem, e a necessidade de ferramentas que possam lidar com a complexidade dos circuitos integrados analógicos continua a ser uma questão para a indústria (COYETTE *et al*., 2016).

Encontrar vetores de teste otimizados sem uma ferramenta de automatização que possibilite a análise de um grande número de possíveis vetores de teste se mostra uma tarefa difícil e custosa no aspecto da demanda de tempo. Essa ineficiência pode resultar em falhas não detectadas, em custos elevados de teste e em um aumento significativo no tempo de desenvolvimento dos produtos, o que impacta diretamente na competitividade da indústria de semicondutores.

Para que novas metodologias de teste, que buscam reduzir os custos dos tradicionais testes baseados em especificações, sejam efetivas, é crucial avaliar com precisão a qualidade do teste. Essa qualidade está relacionada a dois parâmetros-chave: a cobertura de falhas e o *yield loss* (STRATIGOPOULOS; SUNTER, 2014). A cobertura de falhas refere-se ao percentual de falhas detectáveis, ou seja, a capacidade do teste de identificar falhas reais no circuito, enquanto o *yield loss* diz respeito ao percentual de circuitos funcionais que são erroneamente classificados como falhos. Uma boa metodologia de teste deve maximizar a cobertura de falhas e minimizar o *yield loss*, garantindo que a maioria das falhas sejam detectadas, mas sem rejeitar circuitos funcionais desnecessariamente. No entanto, a escolha da metodologia de teste apropriada é extremamente complexa, uma vez que existe uma interdependência entre a qualidade do teste e seu custo. Testes mais eficazes geralmente são mais caros, enquanto testes mais baratos frequentemente não oferecem a mesma qualidade, gerando um compromisso entre a eficácia do teste e o custo envolvido (AGRAWAL; SETH; AGRAWAL, 1982).

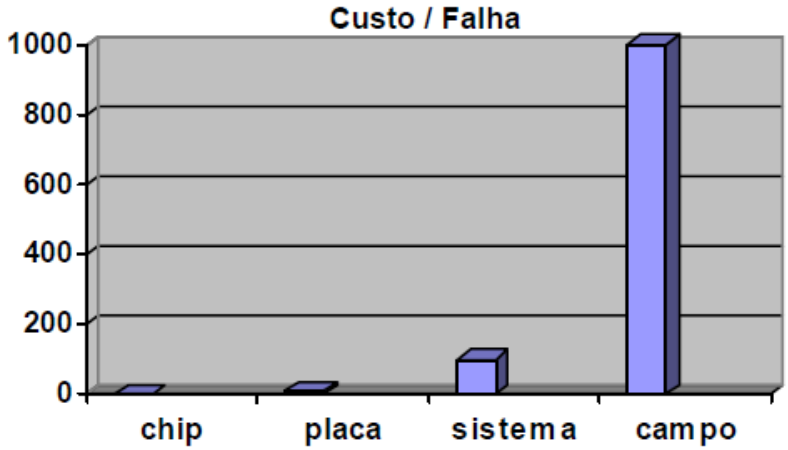
2.1.1 Teste em Produção

No processo de fabricação de circuitos eletrônicos, a detecção precoce de falhas é um aspecto crucial para garantir a qualidade e a confiabilidade do produto final. Uma teoria amplamente aceita na indústria, conhecida como "*rule of ten*", afirma que o custo de reparar uma falha aumenta em dez vezes a cada etapa do processo de produção em que ela passa despercebida, assim como ilustrado na Figura 1. Essa teoria é de grande relevância para a indústria, pois destaca a importância de detectar falhas no início do ciclo de produção. No entanto, conforme a tecnologia de circuitos integrados se torna cada vez mais avançada e complexa, cogita-se a hipótese de que a teoria possa ser ajustada para a "*rule of twenty*", levando em consideração a complexidade crescente dos *chips* e sistemas modernos em comparação com os sistemas mais simples nos quais a teoria original foi desenvolvida (DAVIS, 1982).

A primeira etapa de teste, conhecida como "*wafer probe*", ocorre diretamente no *wafer* de silício, antes de o circuito integrado ser encapsulado. Durante essa fase, o desempenho real

de cada circuito integrado é comparado com as especificações preestabelecidas, e os circuitos integrados com falhas são identificados e descartados, evitando que componentes defeituosos avancem para as etapas seguintes de produção (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 utilizando um equipamento especializado denominado ATE. O custo desses equipamentos pode facilmente ultrapassar milhões de dólares, o que torna o processo de teste uma parte significativa do custo de fabricação de circuitos integrados (CHINAZZO, 2016). O custo do teste de um CI está diretamente relacionado ao custo do ATE e ao tempo necessário para realizar os testes, o qual é conhecido como *throughput*. Quanto menor o tempo de teste, mais circuitos integrados podem ser testados em um mesmo intervalo de tempo, diluindo o custo do equipamento entre as unidades testadas e reduzindo o custo por unidade (SOUDERS; STENBAKKEN, 1990; BALEN, 2006).

A cobertura de falhas está diretamente relacionada ao tempo de execução dos testes. Para alcançar uma maior cobertura de falhas, é necessário realizar testes mais extensivos, o que inevitavelmente aumenta o custo do teste (ARSLAN; ORAILOGLU, 2013). Por outro lado, a cobertura de falhas também é um fator determinante na qualidade do teste, pois ela mede a eficácia dos testes na identificação de falhas no circuito. A cobertura de falhas ideal deve garantir que a maior parte das falhas possíveis sejam detectadas, enquanto o *yield loss* deve ser minimizado, evitando que circuitos funcionais sejam erroneamente rejeitados durante o teste. Nesse sentido, a escolha dos métodos e técnicas de teste deve ser planejada para encontrar um equilíbrio ótimo entre custo e qualidade. A ideia é determinar testes otimizados que, dependendo da aplicação, ofereçam a maior cobertura de falhas possível sem tornar os custos demasiadamente elevados.

No contexto de testes de circuitos integrados analógicos, tanto os testes DC quanto os testes AC são considerados eficazes para a detecção de falhas. No entanto, o teste AC é frequentemente preferido quando é necessário obter uma maior cobertura de falhas, uma vez que ele pode fornecer mais informações sobre o comportamento do circuito integrado em comparação com o teste DC. Isso ocorre porque, ao utilizar sinais alternados, o teste AC permite explorar o circuito integrado em uma gama mais ampla de condições operacionais, revelando comportamentos que podem não ser detectados com sinais de corrente contínua (DC). Além disso, o teste AC é capaz de detectar falhas que são difíceis de identificar em um teste DC, especialmente quando se trabalha com um número limitado de nós de teste (KABISATPATHY; BARUA; SINHA, 2005). No entanto, para garantir a eficácia do teste AC, é essencial escolher a frequência do sinal de estímulo de forma adequada, a fim de maximizar as diferenças nas respostas entre circuitos com e sem falhas.

A escolha da metodologia de teste, seja DC ou AC, está diretamente relacionada à natureza do circuito integrado a ser testado e aos objetivos de desempenho e qualidade. Testes mais extensos e complexos, como o teste AC, podem demandar mais tempo, mas oferecem uma cobertura de falhas superior, enquanto testes menos extensivos e mais rápidos, como o teste DC, podem ser mais econômicos, mas com menor eficiência na detecção de falhas. Dessa forma, a otimização do processo de teste continua sendo um desafio constante, à medida que os sistemas eletrônicos se tornam mais complexos e as exigências de desempenho aumentam. Combinações entre testes DC e AC também podem ser utilizadas a fim de elevar a qualidade dos testes.

2.1.2 Teste Funcional *versus* Teste Estrutural

Os testes de circuitos integrados podem ser classificados em dois grupos com base em seus objetivos: testes funcionais e testes estruturais. Os testes funcionais têm como objetivo validar o funcionamento do circuito fabricado, comparando-o com as especificações de projeto e verificando se o circuito opera dentro dos parâmetros definidos. Por exemplo, em um circuito de áudio, o teste funcional seria considerado falho se a potência elétrica do sinal de saída estivesse abaixo dos limites especificados. Já os testes estruturais, que são orientados a defeitos, concentram-se na identificação de discrepâncias na estrutura do *Circuit Under Test* (CUT) em relação ao seu projeto original. Essa abordagem baseia-se em modelar um grande número de falhas potenciais que podem ocorrer no circuito e, a partir dessas informações, construir um dicionário de falhas, que descreve quais falhas podem ser detectadas por cada teste realizado. No entanto, um dos principais desafios dessa técnica é a necessidade de se conhecer o modelo

de falhas de maneira precisa, o que nem sempre é uma tarefa simples (ARSLAN; ORAILOGLU, 2013).

A metodologia tradicionalmente empregada pela indústria para o teste de circuitos integrados analógicos é o teste funcional. Nesse método, cada parâmetro do CI é verificado individualmente e comparado com os intervalos de tolerância especificados no *datasheet* do componente. O objetivo é garantir que todos os parâmetros operacionais do circuito estejam dentro dos limites estabelecidos, assegurando seu desempenho adequado. Se algum parâmetro estiver fora dos limites de tolerância, o CI é considerado falho e descartado (SRIMANI; GHOSH; RAHAMAN, 2016). Assim como comentado anteriormente, um dos principais desafios dessa técnica é o tempo prolongado necessário para a execução dos testes, o que implica em elevados custos operacionais e impacta o preço final do produto, já que os custos com os testes são repassados ao preço de venda. Portanto, em resumo, embora o teste funcional seja eficaz na verificação da conformidade com as especificações, ele pode resultar em custos elevados devido ao seu longo tempo de execução.

Para tentar reduzir o custo associado aos testes funcionais, é comum utilizar os próprios vetores de teste empregados nas etapas de simulação do projeto para a validação do circuito integrado. Esses vetores, já gerados durante a fase de projeto, não acarretam custos adicionais para sua determinação, o que ajuda a diminuir o custo total dos testes. No entanto, essa abordagem não se aplica aos testes estruturais, os quais envolvem também a determinação dos vetores de teste. Sendo assim, embora os testes estruturais se mostrem como uma opção mais barata se for considerado o teste como um todo, nessa metodologia deve-se necessariamente considerar dois custos: o custo de determinação dos vetores de teste e o custo de execução do teste (RENOVELL, 2004).

Embora os testes estruturais sejam, segundo Renovell (2004), mais viáveis para alcançar boas relações entre qualidade e custo do teste, o aumento da complexidade dos *chips* introduziu novos desafios para essa técnica. Sendo assim, evidencia-se a necessidade de novas ferramentas para tornar essa técnica mais eficaz.

2.1.3 Geração de Teste

A geração automática de padrões de teste, conhecida como ATPG (*Automatic Test-Pattern Generation*), constitui uma etapa fundamental no processo de verificação de circuitos eletrônicos, uma vez que visa determinar vetores de entrada capazes de ativar e propagar falhas internas até pontos observáveis do circuito. Para isso, algoritmos especializados injetam falhas

simuladas e analisam as condições sob as quais seus efeitos podem se manifestar nas saídas, garantindo a possibilidade de detecção (BUSHNELL; AGRAWAL, 2002).

No contexto de circuitos integrados digitais, a propagação da falha depende diretamente dos níveis lógicos presentes nas portas que compõem o caminho de propagação. Em portas AND/NAND, por exemplo, uma falha somente se propaga quando todas as entradas estão em nível lógico “1”, enquanto em portas OR/NOR a propagação ocorre apenas quando todas as entradas permanecem em nível lógico “0”. Em arquiteturas combinacionais mais complexas, compostas por diversas portas interconectadas, o processo torna-se mais desafiador: um único vetor de teste provavelmente não é suficiente para garantir que todas as falhas potenciais produzam efeitos observáveis na saída (TOUBA, 2006). Assim, o ATPG deve explorar múltiplas combinações de entrada, identificando aquelas que, de forma eficaz, ativam e propagam cada falha ao longo das diferentes trajetórias do circuito, assegurando a cobertura de falhas desejada.

Quando se trata de circuitos integrados analógicos, a abordagem de geração de testes segue um princípio semelhante, mas com características distintas. A principal diferença é que, enquanto nos circuitos integrados digitais as falhas costumam se manifestar de forma clara e direta, nos circuitos integrados analógicos as falhas são caracterizadas pela variação do sinal de saída, que pode se desviar dos limites de tolerância estabelecidos para o CI. Em vez de um simples comportamento binário de falha ou não falha, os circuitos integrados analógicos apresentam uma gama de variações contínuas nos sinais de saída, que exigem uma análise mais detalhada para a classificação entre CI com falha ou sem falha (AZAM *et al*., 2024).

Além da simples excursão do sinal de saída, existem vários outros aspectos que podem indicar a presença de falhas em circuitos integrados analógicos. Alguns desses aspectos incluem o nível de ruído no sinal, a fase do sinal, a distorção harmônica, o consumo de energia elétrica e as impedâncias de entrada ou saída. Cada um desses fatores pode fornecer informações cruciais sobre o comportamento do CI e ajudar na identificação de falhas. Em circuitos analógicos, a detecção de falhas não se limita a observar se o sinal de saída está dentro dos limites, mas também envolve a análise da qualidade do sinal em termos de características mais complexas, como a distorção e o ruído, que podem ser indicativos de falhas mais sutis ou comprometimentos no desempenho do circuito.

Portanto, a geração de testes para circuitos integrados analógicos exige uma abordagem mais sofisticada e complexa em comparação com os circuitos integrados digitais, dado que é necessário considerar a natureza contínua e os diferentes parâmetros de operação dos sinais. A evolução das técnicas de ATPG para CIs analógicos tem como foco não apenas a propagação

de falhas, mas também a análise de como os diversos parâmetros do CI podem ser usados para diagnosticar falhas de forma eficaz (CHAUDHURI; BHATTACHARYA; CHAKRABARTY, 2024).

2.1.4 Modelo de Falhas Para Teste de Circuitos Integrados Analógico

No contexto de testes de circuitos integrados, uma falha é definida como um comportamento anômalo que causa a não conformidade de pelo menos uma especificação do *datasheet* do componente. Por outro lado, um defeito refere-se a uma diferença física indesejada entre o CI fabricado e o projeto original, podendo ou não resultar em uma falha funcional (SUNTER *et al*., 2014). A modelagem de falhas em circuitos integrados analógicos consiste em uma descrição matemática de como essas falhas impactam o comportamento do CI, permitindo prever seu efeito durante o teste (FRACCAROLI; FUMMI, 2017).

A determinação de um modelo de falhas adequado é fundamental para estimar com precisão a cobertura de falhas, uma métrica essencial para avaliar a eficácia das metodologias de teste. Quanto mais completo for o modelo de falhas, mais eficaz será a estratégia de teste empregada (STRATIGOPOULOS; SUNTER, 2014).

Para circuitos digitais, o modelo de falha *Stuck-at* é amplamente adotado devido à sua simplicidade e aplicabilidade. Esse modelo assume que um único nó do circuito está permanentemente preso a um nível lógico alto (*Stuck-at-1*) ou a um nível lógico baixo (*Stuck-at-0*), independentemente do sinal de entrada. Dessa forma, o número total de falhas em um circuito digital cresce linearmente com o número de nós, sendo igual ao dobro desse número (HUGHES, 1988).

Diferentemente dos circuitos digitais, os circuitos analógicos apresentam uma variedade mais ampla de falhas, o que torna sua modelagem mais complexa. Nesse contexto, as falhas em circuitos analógicos podem ser classificadas em duas categorias principais: falhas catastróficas e paramétricas (KONDAGUNTURI *et al*., 1999).

Falhas catastróficas correspondem à interrupção total ou parcial de conexões entre terminais de componentes e o restante do circuito, bem como o aparecimento de conexões indesejadas. Essas falhas podem ser comparadas ao modelo *Stuck-at* dos circuitos digitais, pois envolvem circuito aberto (*Stuck-open*) quando há perda de contato entre o terminal de um componente e o restante do CI, impedindo ou dificultando a condução da corrente elétrica. E também o curto-circuito (*Stuck-short*) quando dois ou mais nós do CI entram em contato indevido, alterando seu funcionamento normal (AZAM *et al*., 2024).

Falhas paramétricas não interrompem completamente o funcionamento do circuito, mas causam variações nos parâmetros elétricos dos componentes, fazendo com que operem fora dos limites especificados. Tais variações podem incluir alterações na resistência elétrica, capacitância, corrente elétrica de polarização, ganho de tensão de transistores e outros parâmetros críticos do circuito.

As falhas catastróficas geralmente resultam de contaminações durante o processo de fabricação do CI, como partículas indesejadas que se depositam sobre a máscara do *chip* (RENOVELL, 2004). Para simular essas falhas, diversos autores sugerem o uso de resistores para modelar curtos-circuitos e circuitos abertos. No entanto, há divergências sobre os valores ideais dessas resistências: Alguns autores sugerem a utilização de resistências muito elevadas, da ordem de 100 MΩ para representar circuitos abertos e 1Ω para modelar curtos-circuitos (KONDAGUNTURI et al., 1999). Outros trabalhos indicam valores mais moderados, como 1MΩ para circuitos abertos e 100Ω para curtos (DENG; SHI; ZHANG, 2012). Há ainda abordagens que adotam valores intermediários, como 10MΩ para simular aberturas e 10Ω para representar curtos (PETRASHIN *et al*., 2013)

Por outro lado, há um consenso maior sobre a modelagem de falhas paramétricas. Diversos estudos sugerem que essas falhas sejam modeladas por meio de variações nos parâmetros dos componentes, geralmente entre três e seis desvios padrão em relação aos seus valores nominais (KONDAGUNTURI et al., 1999; DENG et al., 2012). O impacto dessas variações depende diretamente da tecnologia de fabricação de CIs utilizada.

A modelagem de falhas em circuitos analógicos continua sendo um desafio devido à complexidade dos efeitos analógicos, que não podem ser representados apenas por estados binários. Assim, o desenvolvimento de modelos mais precisos e abrangentes é crucial para melhorar a eficácia das metodologias de teste e garantir circuitos mais confiáveis e robustos.

2.1.5 Dicionário de Falhas

O dicionário de falhas para um circuito analógico é o elemento fundamental na escolha das configurações de teste, pois ele fornece informações detalhadas sobre quais falhas são detectadas em determinados nós do circuito, considerando diferentes condições de entrada. Para sua construção, o primeiro passo consiste na seleção de um modelo de falhas adequado. Esse é um dos principais desafios enfrentados no teste de circuitos analógicos, uma vez que, diferentemente dos circuitos digitais, ainda não há um modelo de falhas amplamente padronizado para tais aplicações (KABISATPATHY; BARUA; SINHA, 2005).

A definição dos nós de teste representa a segunda etapa na criação do dicionário de falhas. Inicialmente, pode-se considerar todos os nós do circuito como candidatos para teste, permitindo a seleção daqueles que oferecem maior cobertura de falhas. No entanto, nem todos os nós são acessíveis às medições. Assim, costuma-se priorizar os nós de saída, pois podem ser monitorados sem alterações significativas no circuito. Caso um nó interno demonstre uma capacidade significativamente maior de detecção de falhas, pode-se considerar a inserção de uma estrutura adicional de teste para seu monitoramento. No entanto, essa alteração pode impactar o desempenho do circuito, exigindo estudos adicionais (BENDER, 2015).

Outro aspecto essencial na criação do dicionário de falhas é a seleção dos sinais de entrada a serem considerados. Essa escolha envolve os tipos de excitações a serem consideradas (DC, AC ou transiente), bem como a definição dos seus parâmetros. Entretanto, a determinação de vetores de teste para circuitos analógicos historicamente não tem sido totalmente automatizada (KABISATPATHY; BARUA; SINHA, 2005), sendo frequentemente baseada em metodologias de tentativa e erro (BENDER, 2015). Estudos mais recentes têm buscado algoritmos de otimização para a seleção de vetores de teste, visando maior eficiência na identificação de falhas.

A metodologia para a construção do dicionário de falhas envolve a simulação do circuito sem falhas (*fault-free*) para todos os sinais de entrada estabelecidos, em conjunto com a simulação do circuito contendo cada uma das falhas modeladas. Posteriormente, analisa-se a resposta do circuito nos nós de teste, registrando quais falhas podem ser detectadas para cada condição de entrada (YANG; TIAN; LONG, 2009).

Para a correta identificação das falhas, é essencial definir limites dentro dos quais a resposta do CUT seja considerada aceitável. No presente trabalho, a determinação do dicionário de falhas foi realizada com base nos códigos desenvolvidos por Chinazzo (2016) em MATLAB, nos quais se aplica uma faixa de tolerância. Se a resposta do circuito ultrapassar esse limite, ele é classificado como defeituoso. Esse conceito foi derivado do teste *go/no-go*, e tem sido amplamente adotado em metodologias de teste devido à sua simplicidade e eficiência na tomada de decisão (MILOR; VISVANATHAN, 1987).

A estrutura do dicionário de falhas é organizada em uma matriz de falhas, onde cada entrada assume um valor binário: "1" para indicar a detecção da falha e "0" para indicar que a falha não foi identificada naquelas condições (DEB *et al*., 1995).

2.1.6 Técnicas de *Design-for-Testability* (DfT)

A necessidade de estratégias eficientes de teste cresce à medida que os dispositivos se tornam mais complexos, exigindo abordagens que facilitem a detecção de falhas e otimizem o tempo de aplicação destes testes (NAGARAJ, 2025). Nesse cenário, o *Design-for-Testability* (DfT) surge como um conjunto de técnicas voltadas para o desenvolvimento de CIs que já possuem mecanismos embutidos para facilitar o processo de teste (WANG; WU; WEN, 2006). O DfT permite que a análise e teste dos circuitos integrados sejam realizados de maneira mais eficiente, reduzindo custos e aumentando a cobertura de falhas.

A aplicação de técnicas de DfT tem como foco principal melhorar a observabilidade e a controlabilidade dos circuitos integrados (NAGARAJ, 2025), possibilitando que eventuais falhas possam ser detectadas através de procedimentos mais baratos de teste. Para alcançar esse objetivo, são inseridos recursos no projeto que permitem a aplicação e a análise de sinais de teste de forma mais eficiente.

As técnicas de DfT podem apresentar diversos benefícios. Primeiramente, pode-se citar a redução do tempo de teste, pois através da inclusão de elementos que facilitam a aplicação de estímulos permite que o teste dos circuitos seja realizado em menos tempo, aumentando a produtividade e reduzindo o custo do processo. Além disso, há a melhora na qualidade, métodos como varredura (*scan*), inserção de pontos de controle e observação, além de técnicas de autocorreção, são capazes de aumentar a cobertura de falhas, conforme observado em arquiteturas modernas de alto desempenho que empregam DfT de maneira intensiva (NAGARAJ, 2025). Por fim, destaca-se a maior facilidade no diagnóstico, pois estruturas voltadas para a testabilidade tornam a análise de falhas mais ágil, auxiliando na identificação dos defeitos durante a fabricação.

Como exemplo de técnica de DfT, pode-se citar o *scan design*, que é amplamente adotado em circuitos integrados digitais complexos, permitindo que os estados internos sejam controlados e observados diretamente pelo testador. Para isso, são inseridos *flip-flops* de varredura no CI, possibilitando que os sinais de teste sejam aplicados e lidos de maneira simplificada, aumentando a detecção de falhas em elementos sequenciais (ARAMOON; QU; CUI, 2022).

Outro exemplo bastante utilizado é o BIST (*Built-in Self-Test*). Essa técnica consiste em adicionar mecanismos internos ao próprio circuito integrado que servem exclusivamente para realização de testes, eliminando a necessidade de um equipamento de teste externo (ATE). Essa abordagem possibilita verificações automáticas em diferentes momentos, inclusive durante a operação normal do dispositivo. Os principais componentes de um sistema BIST incluem:

gerador de estímulos de teste, bloco de análise de respostas e lógica de decisão de falha (DRI; PERETTI; ROMERO, 2022).

Outra técnica, chamada de *Boundary Scan* e padronizada pelo IEEE 1149.1, é utilizada para testar circuitos integrados sem a necessidade de acesso físico direto aos seus pinos. Esse método emprega registradores de varredura internos aos *chips* e conectados aos seus *pads*, permitindo a análise de falhas em interconexões do circuito e outros componentes montados em placas de circuito impresso (INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, 2013). Estudos mostram que o *Boundary Scan* tem sido amplamente empregado em placas de alta densidade por permitir a detecção eficiente de defeitos estruturais, reduzir o tempo de teste e eliminar a necessidade de grandes quantidades de pontos físicos de acesso (SHASHIDHARA; YELLAMPALLI; GOUDANAVAR, 2014).

Embora muitas das estratégias de DfT sejam direcionadas para circuitos integrados digitais, também existem técnicas voltadas para circuitos integrados analógicos e mistos. A mais simples delas pode ser a simples adição de um contato a mais no circuito que possibilite a medição das tensões de um nó interno do CI ou a possibilidade de dividir o circuito em blocos, desconectando-os do resto do circuito e possibilitando seus testes de maneira independente (ZAKIZADEH *et al*., 2005).

2.2 CARACTERIZAÇÃO ELÉTRICA DAS VARIABILIDADES DO PROCESSO DE FABRICAÇÃO

Durante a etapa de projeto de circuitos eletrônicos, fazer simulações computacionais vem se tornando cada vez mais importante. Com o avanço das tecnologias CMOS (*Complementary Metal-Oxide-Semiconductor*), os transistores atingiram dimensões extremamente reduzidas, tornando o controle do processo de fabricação um grande desafio (QUIAN, 2015). Nesse contexto, para garantir o correto funcionamento dos circuitos integrados fabricados e evitar *designs* excessivamente conservadores, é fundamental investigar e compreender a variabilidade do processo produtivo (RAPPITSCH *et al*., 2004; MEZZOMO *et al*., 2011).

A variabilidade em circuitos eletrônicos decorre de múltiplos fatores, tornando impossível a fabricação de dispositivos eletricamente idênticos. Entre as principais fontes de variação, destaca-se a granularidade da matéria, que causa flutuações aleatórias na concentração de dopantes nos canais dos transistores. Essas flutuações impactam diretamente a tensão de limiar (*threshold voltage*), afetando as correntes de fuga e o consumo energético dos dispositivos (ROGENMOSER; CLARK, 2013). Outro fator relevante é a fotolitografia, cujas

variações na intensidade luminosa, especialmente nas bordas dos padrões impressos, podem ser influenciadas pela rugosidade do material e efeitos quânticos da luz (SHIN, 2016). Além disso, o efeito de proximidade, em que íons colidem nas bordas da fotoresina e são refletidos aleatoriamente, gera uma dopagem indesejada na área ativa, contribuindo para a variabilidade do dispositivo (LI; WU, 2016). Aspectos ambientais, como temperatura, tensão de alimentação e radiação ionizante, também introduzem variações significativas no desempenho elétrico dos circuitos integrados.

A caracterização elétrica dessas variabilidades pode ser realizada por meio de testes aplicados a estruturas específicas que possibilitam a medição individual dos transistores. Tais testes permitem a estimativa do desvio padrão de parâmetros críticos e, consequentemente, a definição de modelos estatísticos para representar as variações impostas pelo processo de fabricação nos circuitos integrados finais (CHEN *et al*., 2015).

Três abordagens são amplamente utilizadas para a medição individual de transistores. O método mais comum é o de pares casados (*matched pairs*), que consiste em utilizar dois transistores idênticos em *layout* e comparar seus parâmetros, determinando a variabilidade com base no desvio padrão dos descasamentos (TUINHOUT, 2005; CHEN *et al*., 2015). A segunda abordagem é o *simple array*, no qual um grande número de transistores compartilha quatro terminais comuns, permitindo medições simultâneas e proporcionando um maior volume de dados estatísticos. Por fim, a técnica do *array* endereçado emprega um barramento de dados (*data bus*) e um circuito decodificador, possibilitando a seleção individual de cada transistor para medição, permitindo análises mais detalhadas (MIZUTANI; KUMAR; HIRAMOTO, 2011).

O grande desafio na caracterização elétrica da variabilidade dos circuitos integrados está na utilização das medições individuais de transistores para a análise do impacto global dessas variações no desempenho do circuito como um todo. Para esse fim, duas metodologias são amplamente empregadas para simulação de variabilidade: a análise de *corners* e as simulações de Monte Carlo. A análise de *corners* considera cenários extremos de variabilidade, permitindo verificar o desempenho do circuito em condições de pior caso. Já a simulação de Monte Carlo adota uma abordagem estatística baseada na geração de múltiplas amostras aleatórias dentro de uma distribuição específica, proporcionando uma visão probabilística do impacto das variações (CORTES *et al*., 2003).

Dessa forma, a correta caracterização elétrica da variabilidade e a aplicação de metodologias apropriadas são fundamentais para garantir a robustez e o desempenho confiável dos circuitos eletrônicos modernos.

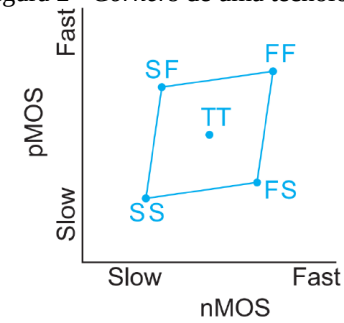
2.2.1 Análise de *Corners*

Uma das formas mais utilizadas para modelar o impacto das variabilidades de processo nos circuitos integrados é a análise de *corners*. Esse método considera quatro extremos para as variações toleráveis nos parâmetros dos transistores, denominados "piores casos". Esses limites englobam diferentes cenários operacionais: pior caso em relação à velocidade, pior caso quanto ao consumo energético, pior caso lógico "1" e pior caso lógico "0".

No pior caso relacionado à velocidade, tanto os transistores NMOSFET (*N-channel Metal-Oxide-Semiconductor Field-Effect Transistors*) quanto os PMOSFET (*P-channel Metal-Oxide-Semiconductor Field-Effect Transistors*) operam com os piores valores para os parâmetros que os tornam "lentos", porém ainda no limiar da aceitação. De maneira oposta, no pior caso de consumo energético, os transistores são considerados "rápidos", pois apresentam correntes de fuga maiores. O pior caso lógico "1" ocorre quando os transistores NMOSFET são "rápidos" e os PMOSFET são "lentos". No pior caso lógico "0", o inverso acontece: os NMOSFET são "lentos" e os PMOSFET, "rápidos". Esses quatro cenários resultam nos chamados *corners*, representados pelas siglas: SS (*Slow-Slow*), SF (*Slow-Fast*), FS (*Fast-Slow*) e FF (*Fast-Fast*), onde a primeira letra indica a condição do transistor NMOSFET, e a segunda, do PMOSFET (RAPPITSCH et al., 2004).

A Figura 2 ilustra os quatro *corners* de um processo de fabricação hipotético. As linhas interligando os corners representam os limites de tolerância dos parâmetros dos componentes. No centro da área demarcada, encontra-se a condição típica, representada pela sigla TT (*Typical-Typical*), que indica os valores médios esperados para os dispositivos em condições normais de operação.

Figura 2 - *Corners* de uma tecnologia.



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

A análise de variabilidade através das análises de *corners* apresenta vantagens significativas em termos de custo computacional se comparada as análises de Monte Carlo. A determinação dos limites toleráveis nas respostas dos circuitos requer um menor número de

simulações em comparação com a técnica de Monte Carlo (MERRETT *et al*., 2011). Por essa razão, é amplamente utilizada na fase inicial do projeto de circuitos integrados.

Entretanto, ao aplicar a análise de *corners* em circuitos integrados analógicos, observa-se que as previsões tendem a ser excessivamente otimistas ou pessimistas em relação às reais variações dos CIs (BENSCHWARTZ; SAKTHIVEL, 2016). Isso ocorre porque essa metodologia foi desenvolvida com foco em circuitos integrados digitais, nos quais os transistores são analisados em estados discretos de liga/desliga, enquanto circuitos analógicos são sensíveis a variações contínuas de seus parâmetros elétricos. Assim, para avaliações mais detalhadas e precisas de circuitos que contêm um grande número de componentes, a técnica de Monte Carlo se mostra mais eficaz (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

A análise de Monte Carlo para avaliação das variabilidades em circuitos eletrônicos baseia-se em simulações sucessivas de amostras do circuito. Cada amostra é composta por componentes cujos parâmetros são selecionados aleatoriamente, de acordo com os comportamentos estatísticos individualmente caracterizados desses componentes (MASON *et al*., 2008). Em outras palavras, o método simula a fabricação de múltiplas amostras do circuito, com cada uma apresentando variações nos parâmetros dos componentes, conforme a caracterização elétrica estatística prévia. Dessa forma, ao realizar um número suficientemente grande de simulações, é possível modelar as variações gerais impostas pelo processo de fabricação e analisar seu impacto no desempenho global do circuito.

Uma vantagem significativa da análise de Monte Carlo em comparação com a análise de *corners* é a capacidade de fornecer caracterizações mais realistas do comportamento do circuito. Isso ocorre porque a análise de Monte Carlo considera a interação entre as variações individuais dos componentes, ao passo que a análise de *corners* se limita a avaliar o impacto dos parâmetros nos extremos de suas faixas de variação. Essa abordagem mais abrangente permite uma avaliação mais precisa das variações no desempenho do circuito, refletindo de maneira mais fiel as condições reais de fabricação (MERRETT *et al*., 2011).

Entretanto, a análise de Monte Carlo envolve um custo computacional consideravelmente mais elevado, uma vez que exige um número muito maior de simulações do que a análise de *corners*, que, em geral, utiliza apenas quatro pontos de operação,

correspondendo aos limites superiores e inferiores dos parâmetros dos componentes. Essa demanda por recursos computacionais pode tornar a implementação da análise de Monte Carlo um desafio, especialmente em circuitos mais complexos ou quando a precisão das simulações requer um grande número de amostras.

2.3 METODOLOGIAS DE TESTE UTILIZADAS NESSE TRABALHO

Para as simulações SPICE, foram consideradas três metodologias de teste diferentes: análise em corrente contínua (DC), análise em corrente alternada (AC) e análise de resposta transiente. Essas três metodologias são descritas em maiores detalhes nas respectivas Seções 2.3.3, 2.3.1 e 2.3.2.

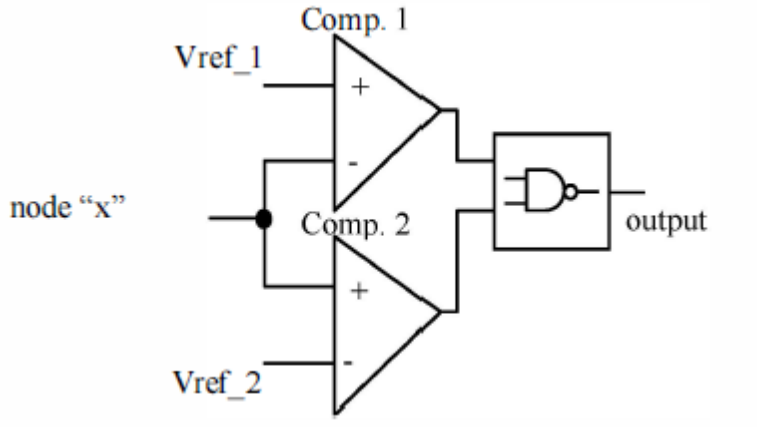
2.3.1 Teste DC

A técnica de teste em corrente contínua (DC) baseia-se na análise das tensões nos nós do circuito para identificar a presença de falhas. Nesse teste, são aplicadas tensões contínuas nas entradas do circuito para então avaliar as tensões nodais de regime permanente. Ao avaliar as tensões DC em condições normais e sob injeção de falhas, é possível estabelecer limites de tolerância que classificam as tensões nodais do CUT como aceitáveis ou não, determinando a aprovação ou reprovação no teste.

Diversos estudos aplicaram essa técnica para avaliar condições de teste. Petrashin, Dualibe, Lancioni e Toledo (2013) implementaram-na em um amplificador de transcondutância analógico projetado em tecnologia de 65 nm, destacando que a metodologia de teste DC é econômica em comparação com outras técnicas e particularmente eficaz na detecção de falhas catastróficas. De maneira semelhante, Bender (2015) investigou amplificadores diferenciais em tecnologia de 500 nm, obtendo boas coberturas de falhas.

Além do baixo custo, a simplicidade das medições de tensão DC facilita a implementação de estratégias de Autoteste Integrado (BIST) de forma econômica. A Figura 3 ilustra um verificador proposto, no qual dois comparadores e uma porta lógica são suficientes para testar a tensão em um nó específico do circuito, aprovando-o ou não (PETRASHIN et al., 2013). As tensões de referência (Vref_1 e Vref_2) definem os limites de aceitação, e o nó sob teste é denominado "x". Se a tensão no nó "x" estiver dentro da faixa aceitável, as saídas dos comparadores estarão em nível lógico alto, resultando em uma saída de nível lógico baixo na porta lógica, indicando aprovação no teste. Qualquer desvio além dos limites faz com que um dos comparadores assuma nível lógico baixo, levando a uma saída de nível lógico alto na porta lógica, sinalizando que o circuito possui falha e deve ser rejeitado.

Figura 3 - Diagrama em blocos do verificador proposto.



Fonte: Petrashin et al. (2013).

2.3.2 Teste AC

A metodologia de teste em corrente alternada (AC) envolve a análise espectral das tensões nos nós do circuito. Ao aplicar uma excitação multi senoidal na entrada, as tensões nodais exibem assinaturas específicas resultantes dos diferentes ganhos de tensão para cada frequência. Essa análise espectral é geralmente realizada por meio da Transformada Rápida de Fourier (*Fast Fourier Transform* - FFT), devido à sua eficiência computacional (CHAUHAN *et al.*, 2014).

Contudo, a análise espectral impõe um custo computacional elevado e requer alta resolução dos equipamentos de teste para alcançar a precisão desejada. Para simplificar a metodologia de teste AC, alguns estudos sugerem o uso de sinais multi senoidal compostos por harmônicos específicos, facilitando a aplicação da FFT (CHAUHAN et al., 2014). Outra abordagem é a metodologia de teste serial, na qual sinais de entrada compostos por uma única frequência são utilizados. Nesse caso, o teste torna-se mais simples, pois a amplitude de um único harmônico é o foco da análise, permitindo o uso de detectores de pico para a execução do teste ao invés de precisar usar analisadores de espectro e FFT (CHINAZZO, 2016).

No âmbito da ferramenta desenvolvida neste trabalho, a metodologia de testes AC considera a utilização de harmônicos como sinais de entrada do circuito e então possibilita que duas grandezas sejam avaliadas nos nós de teste: o módulo e a fase do sinal. No teste AC de módulo, analisa-se exclusivamente a amplitude do sinal nodal, seja tensão ou corrente elétrica, sem considerar a diferença de fase entre o sinal aplicado na entrada e o sinal observado no nó analisado.

Por outro lado, no teste AC de fase, a variável de interesse consiste na diferença de fase, expressa em radianos, entre o sinal nodal e o sinal de entrada, não sendo levado em conta o

módulo do sinal. No entanto, para que a medição de fase seja viável, o sinal deve apresentar amplitude mínima suficiente para permitir a correta identificação da defasagem. Essa amplitude mínima depende de características do equipamento de teste, tais como sua resolução, bem como dos níveis de ruído presentes no sinal medido.

Na ferramenta desenvolvida, as precisões do equipamento de teste são fornecidas como parâmetros de entrada e são empregadas na determinação do dicionário de falhas, de modo a incorporar condições mais realistas ao processo de caracterização. Esses aspectos e suas implicações na análise serão discutidos em maior profundidade na Seção 3.1.3.

A escolha entre as metodologias de teste AC depende das especificidades do circuito e dos recursos disponíveis, equilibrando a complexidade do teste com a precisão necessária para garantir o desempenho adequado do circuito analógico.

2.3.3 Teste Transiente

O método de teste baseado na Análise de Resposta Transiente (TRAM - *Transient Response Analysis Method*) foi proposto como uma alternativa para o teste de circuitos analógicos, especialmente filtros. Nesse método, CUT é decomposto em blocos de primeira e segunda ordem, os quais são submetidos a estímulos transientes específicos, como degrau (*step*), parábola ou rampa. A resposta transiente de cada bloco é então analisada, focando em parâmetros como o tempo de acomodação e o *overshoot*. Esses parâmetros são comparados aos valores esperados para identificar possíveis falhas no CUT (CALVANO; ALVES; LUBASZEWSKI, 1999, 2000).

A análise da resposta transiente permite detectar discrepâncias no comportamento dinâmico do circuito, que podem não ser evidentes em testes estáticos (VARIYAM; CHERUBAL; CHATTERJEE, 2002). Por exemplo, variações no tempo de acomodação ou *overshoot* podem indicar problemas de estabilidade ou desempenho inadequado dos componentes.

2.4 AMPLIFICADORES TOTALMENTE DIFERENCIAIS

Dentro do vasto universo dos circuitos integrados analógicos, os amplificadores desempenham um papel crucial, especialmente em aplicações de condicionamento de sinais. Eles são fundamentais para a interface entre circuitos integrados digitais e o mundo exterior, que, por natureza, opera de forma analógica. Em conjunto com conversores analógico-digitais (ADC) e digital-analógicos, os amplificadores garantem que sinais analógicos sejam corretamente amplificados, filtrados e ajustados antes da conversão para o domínio digital.

Dada a relevância desses circuitos no processamento de sinais, os amplificadores foram escolhidos como objeto de estudo deste trabalho, justificando a existência deste capítulo dedicado ao tema. Dentre os diversos tipos de amplificadores, os amplificadores diferenciais totalmente diferenciais (*Fully Differential Amplifiers* - FDAs) se destacam por suas vantagens em relação aos amplificadores diferenciais convencionais (*Differential Amplifiers – DA*).

Ao contrário de um amplificador diferencial tradicional, que possui uma única saída, os FDAs apresentam saídas diferenciais, ou seja, duas tensões de saída em oposição de fase. Essa característica faz com que eles sejam amplamente empregados no condicionamento de sinais para a entrada de conversores A/D, onde a simetria do sinal e a rejeição a ruídos são aspectos críticos (KARKI, 2002).

Os FDAs oferecem diversas vantagens quando comparados a amplificadores convencionais. A primeira delas é a rejeição de ruído em modo comum, como os FDAs operam de forma diferencial, eles são menos suscetíveis a interferências externas e ruídos de modo comum, melhorando significativamente a relação sinal-ruído (*Signal-to-Noise Ratio* - SNR). Além disso, também apresentam redução de distorções harmônicas de ordem par devido à simetria entre os sinais de saída diferenciais. Por fim, apresentam também uma maior excursão dinâmica: Devido à inversão de fase entre as duas saídas diferenciais, os FDAs dobram a excursão dinâmica do sinal de saída (KARKI, 2002).

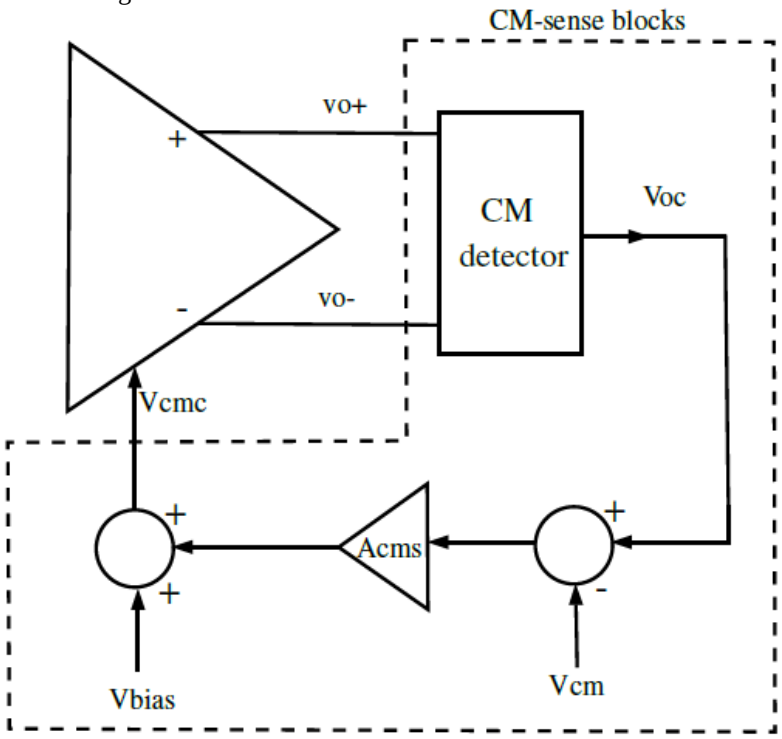
Apesar das vantagens, os FDAs possuem um desafio adicional: a necessidade de um circuito de realimentação de modo comum (*Common-Mode Feedback – CMFB*). Esse circuito é essencial para manter a tensão de modo comum das saídas do amplificador em um valor desejado, garantindo a estabilidade da operação.

Na ausência de um CMFB eficiente, o amplificador pode se tornar instável, comprometendo seu desempenho (OLIVEIRA; SEVERO; GIRARDI, 2014). O CMFB deve ser projetado para operar de maneira confiável em toda a faixa de frequências de interesse, além de ser independente do nível DC do sinal de entrada (XU; EMBABI, 2000; ALZAHER; ELWAN; ISMAIL, 2003).

A Figura 4 apresenta um diagrama conceitual do circuito CMFB utilizado em FDAs. Nele, as tensões de saída ("vo+" e "vo-") são lidas por um bloco detector de modo comum, que calcula a tensão média entre ambas e gera um sinal de realimentação ("Voc"). Esse sinal é então comparado à tensão de referência desejada ("Vcm"), e a diferença entre eles é amplificada pelo bloco "Acms". O resultado é uma tensão corrigida que ajusta a polarização do amplificador operacional (*OpAmp*), estabilizando o circuito. Essa realimentação é essencial para garantir que as saídas diferenciais do amplificador não derivem para valores indesejados, evitando

distorções e garantindo a qualidade do sinal amplificado (OLIVEIRA; SEVERO; GIRARDI, 2014).

Figura 4 - Bloco conceitual do circuito CMFB do FDA.



Fonte: Oliveira; Severo; Girardi (2014).

3 DESCRIÇÃO DA FERRAMENTA

Neste capítulo, será descrito todo o funcionamento da ferramenta criada em linguagem Python 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. Essa ferramenta carrega diversas funcionalidades e aspectos visuais semelhantes à *toolbox* criada em MATLAB em Zilch (2021), porém, com algumas novas funcionalidades que serão descritas no decorrer da seção.

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 circuitos 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 ferramenta, é necessário dispor da sua descrição em linguagem SPICE (extensão ‘.sp’ ou ‘.cir’) e um simulador SPICE, neste trabalho, foi utilizado o NGSPICE. Além disso, é opcional dispor de um arquivo com a descrição das falhas a serem analisadas (extensão ‘.fii’). Esse arquivo é opcional, pois a ferramenta disponibiliza um recurso que auxilia na criação desse arquivo, listando automaticamente todas as falhas catastróficas e fornecendo suporte à especificação das falhas paramétricas, reduzindo assim o esforço manual do usuário.

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 umas 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 a 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 ou correntes elétricas 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 os 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 de estratégias de DfT.

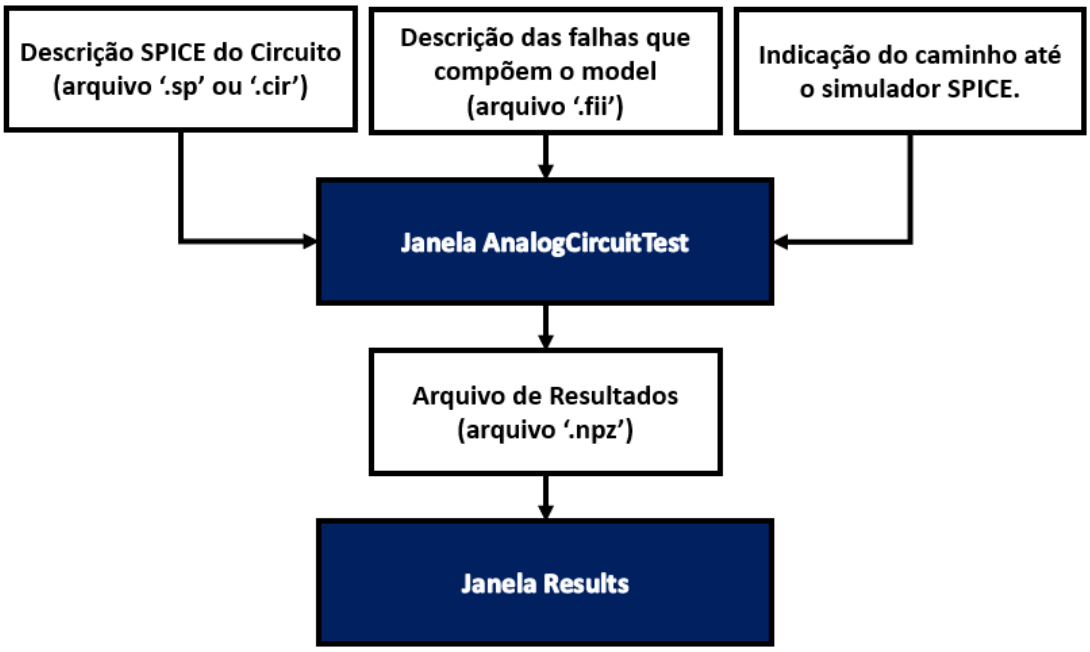
A execução completa da ferramenta requer a utilização de duas janelas, assim como mostra a Figura 5. A primeira janela de execução é chamada de “AnalogCircuitsTest” e é descrita em maiores detalhes na Seção 3.1. Ela é responsável por criar e simular todas as descrições SPICE necessárias à análise, ler 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 ‘.npz’. A segunda janela de execução é chamada de “Results” e é descrita pormenorizadamente na Seção 3.2. Ela se encarrega de ler o arquivo ‘.npz’ 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 resultarão em novos conjuntos de testes otimizados.

O formato ‘.npz’ é uma extensão nativa da biblioteca NumPy (*Numerical Python*), utilizada em aplicações desenvolvidas em Python**.** Esse formato permite o armazenamento eficiente de múltiplos vetores NumPy em um único arquivo comprimido**,** preservando integralmente as informações de cada estrutura de dados, como dimensões, tipo dos elementos e valores numéricos.

A biblioteca NumPy é uma das principais ferramentas para computação científica na linguagem Python, oferecendo estruturas de dados e operações vetorizadas altamente otimizadas para manipulação numérica e matricial. Desenvolvida inicialmente por Travis Oliphant em 2006, a biblioteca evoluiu a partir do projeto *Numeric* e tornou-se um dos pilares

do ecossistema científico em Python (OLIPHANT, 2007). Entre suas principais características, destacam-se o suporte a *arrays* multidimensionais (*ndarray*), a eficiência computacional em operações matemáticas e a integração com outras bibliotecas científicas, como SciPy, Pandas e Matplotlib (HARRIS et al., 2020).

Figura 5 – Diagrama de blocos geral da ferramenta para análise de configurações de teste em circuitos analógicos.

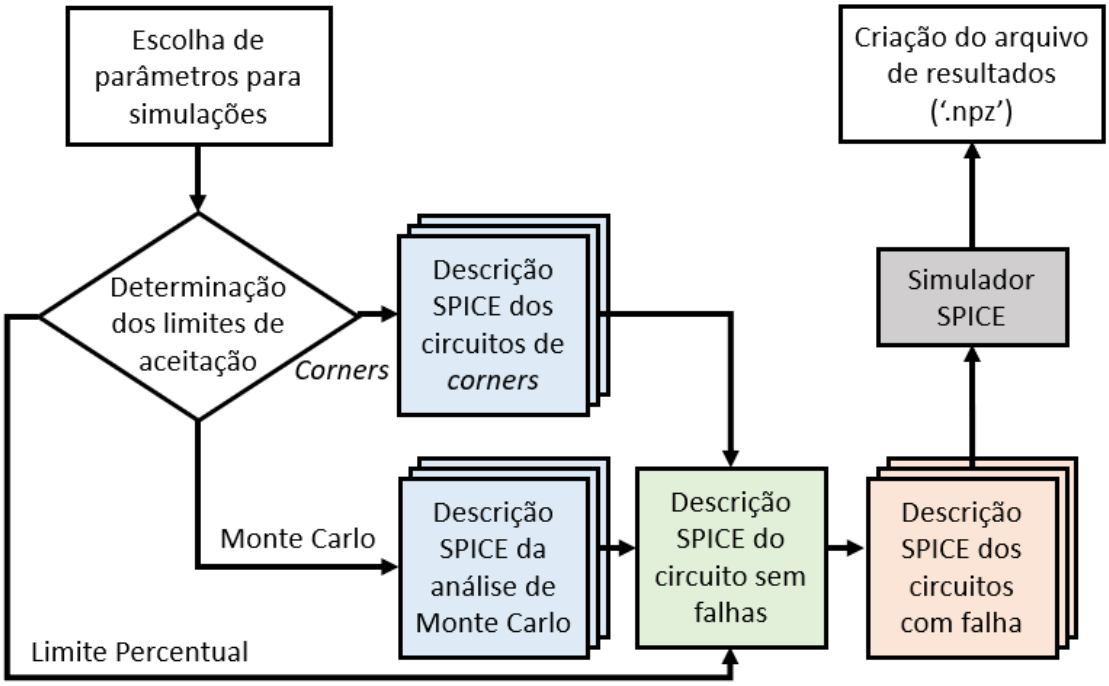


Fonte: elaborado pelo autor.

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 Seções 3.1.1 até 3.1.3, 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".



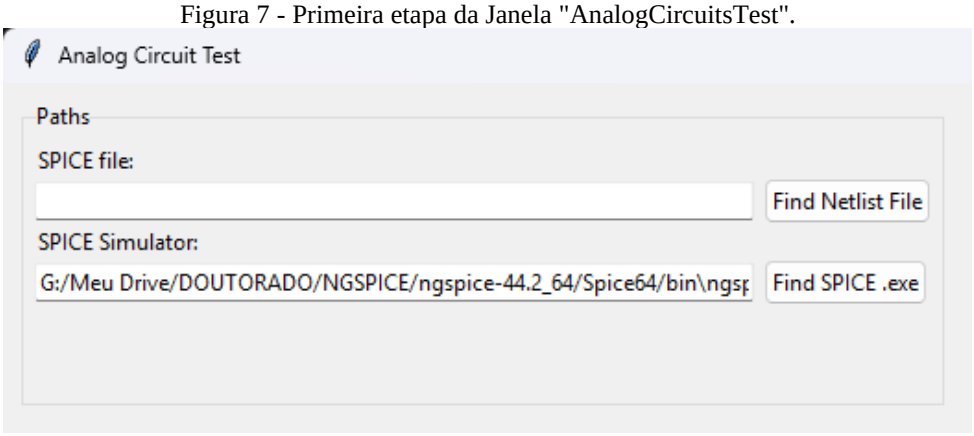
Fonte: elaborado pelo autor.

3.1.1 Etapa 1: Seleção de Arquivos para Simulação

Quando a janela "AnalogCircuitsTest" é executada, os primeiros elementos que aparecem são mostrados na Figura 7. Nesse momento, é necessário a inserção de dois caminhos para arquivos, sendo eles: o arquivo de descrição SPICE do circuito sem falhas (extensão '.sp' ou '.cir') e o caminho para o simulador SPICE (arquivo com extensão '.bat').

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".



Fonte: elaborado pelo autor.

Após a correta seleção do caminho do arquivo de descrição SPICE do circuito, novos elementos ficam disponíveis na janela para que seja possível a completa especificação do modelo de falhas. Essa nova etapa será descrita na Seção 3.1.2.

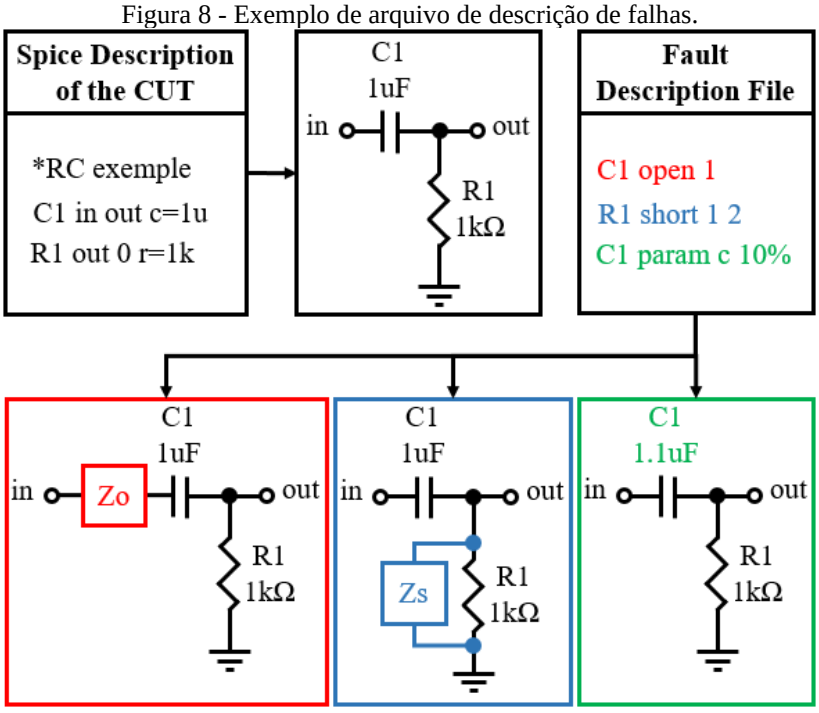
3.1.2 Etapa 2: Escolha das Falhas que Irão Compor o Modelo.

Para descrever o modelo de falhas nessa ferramenta, é necessário um arquivo de texto na extensão '.fii', que funciona como uma lista, na qual são especificadas todas as possíveis falhas que podem ocorrer no circuito. A sintaxe deste arquivo 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 8, 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.

Figura 8 - Exemplo de arquivo de descrição de falhas.



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.

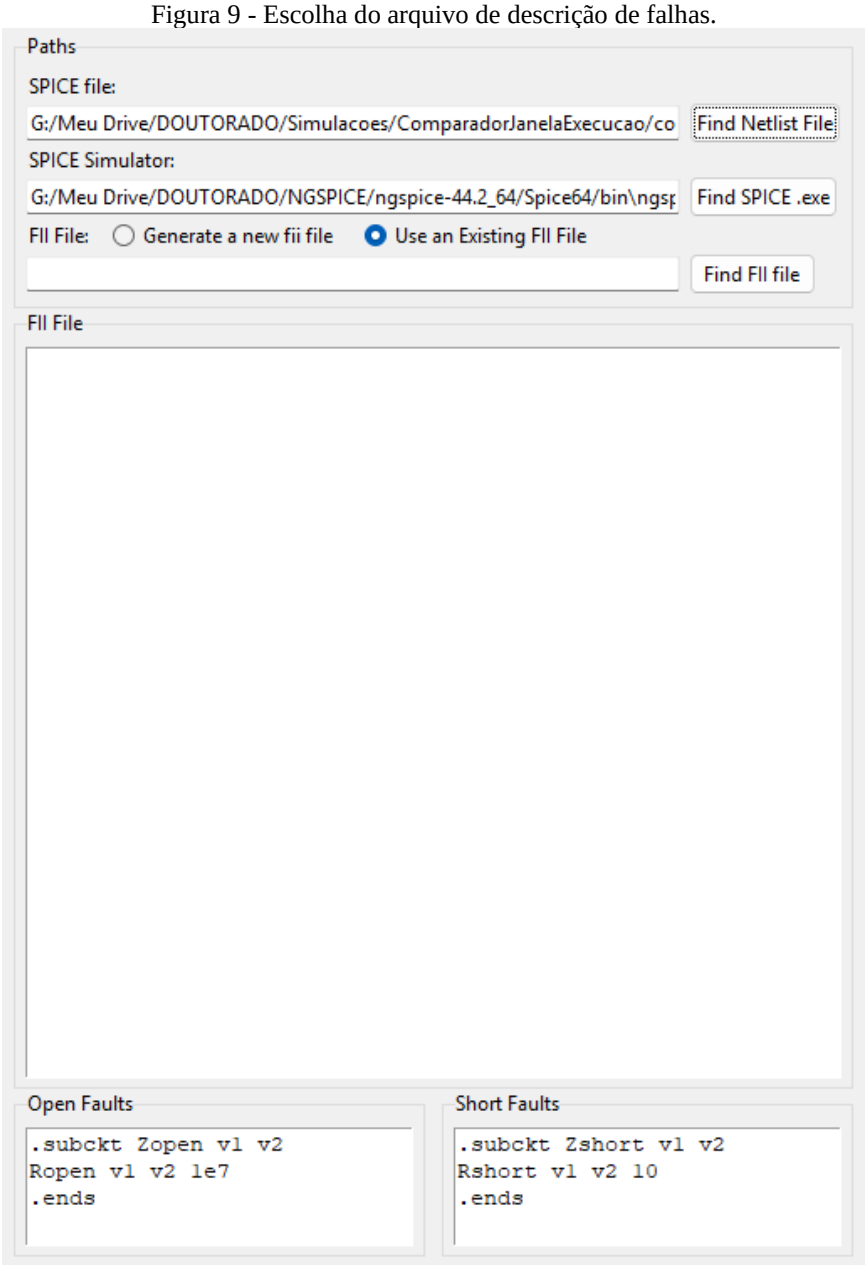
É importante destacar que a ferramenta desenvolvida interpreta os nomes dos componentes exatamente conforme declarados na descrição SPICE original. Quando um componente pertence a um subcircuito, seu nome completo é formado pela concatenação do nome do subcircuito com o nome interno do componente, separados por um ponto. Essa convenção assegura a identificação única de cada elemento dentro da hierarquia do circuito.

Para que o processo de injeção de falhas opere corretamente, é fundamental que cada subcircuito seja instanciado apenas uma vez no *netlist*. Caso haja a necessidade de utilizar o mesmo subcircuito em mais de uma parte do circuito, é necessário duplicar a declaração do subcircuito, atribuindo-lhe nomes distintos. Assim, cada cópia poderá ser instanciada uma única vez, evitando conflitos de identificação durante a análise.

Os nomes dos componentes internos dessas cópias podem permanecer inalterados, uma vez que a diferenciação entre eles ocorre por meio do prefixo associado ao nome do subcircuito ao qual pertencem. Dessa forma, apenas o nome do subcircuito precisa ser distinto em cada duplicata, garantindo a unicidade dos identificadores de componentes em toda a hierarquia do circuito.

Na Figura 9 é ilustrada a janela de execução da ferramenta, que permite ao usuário definir o arquivo de descrição de falhas. Essa definição pode ser feita de duas maneiras: informando o caminho para um arquivo de descrição de falhas já existente ou permitindo que a própria ferramenta gere um novo arquivo.

Figura 9 - Escolha do arquivo de descrição de falhas.



Fonte: elaborado pelo autor.

No primeiro caso, ao selecionar a opção de utilizar um arquivo de descrição de falhas previamente criado, o botão "*Find FII File*" possibilita a localização e seleção do arquivo desejado nos diretórios do computador. Após a escolha do arquivo .fii, o conteúdo é automaticamente carregado e exibido na caixa de texto "*FII file*", facilitando a visualização e a conferência das falhas que compõem o modelo.

Alternativamente, caso o usuário opte por gerar um novo arquivo de descrição de falhas, a ferramenta cria automaticamente um arquivo .fii no mesmo diretório em que se encontra a descrição SPICE do circuito *fault free*. Nesse processo, são inseridas previamente todas as falhas catastróficas possíveis, e o conteúdo do arquivo é carregado na janela de texto "*FII file*".

A partir daí, o usuário pode editar livremente a lista de falhas diretamente na janela de execução, removendo aquelas que não deseja incluir no modelo ou adicionando novas falhas, como as de natureza paramétrica.

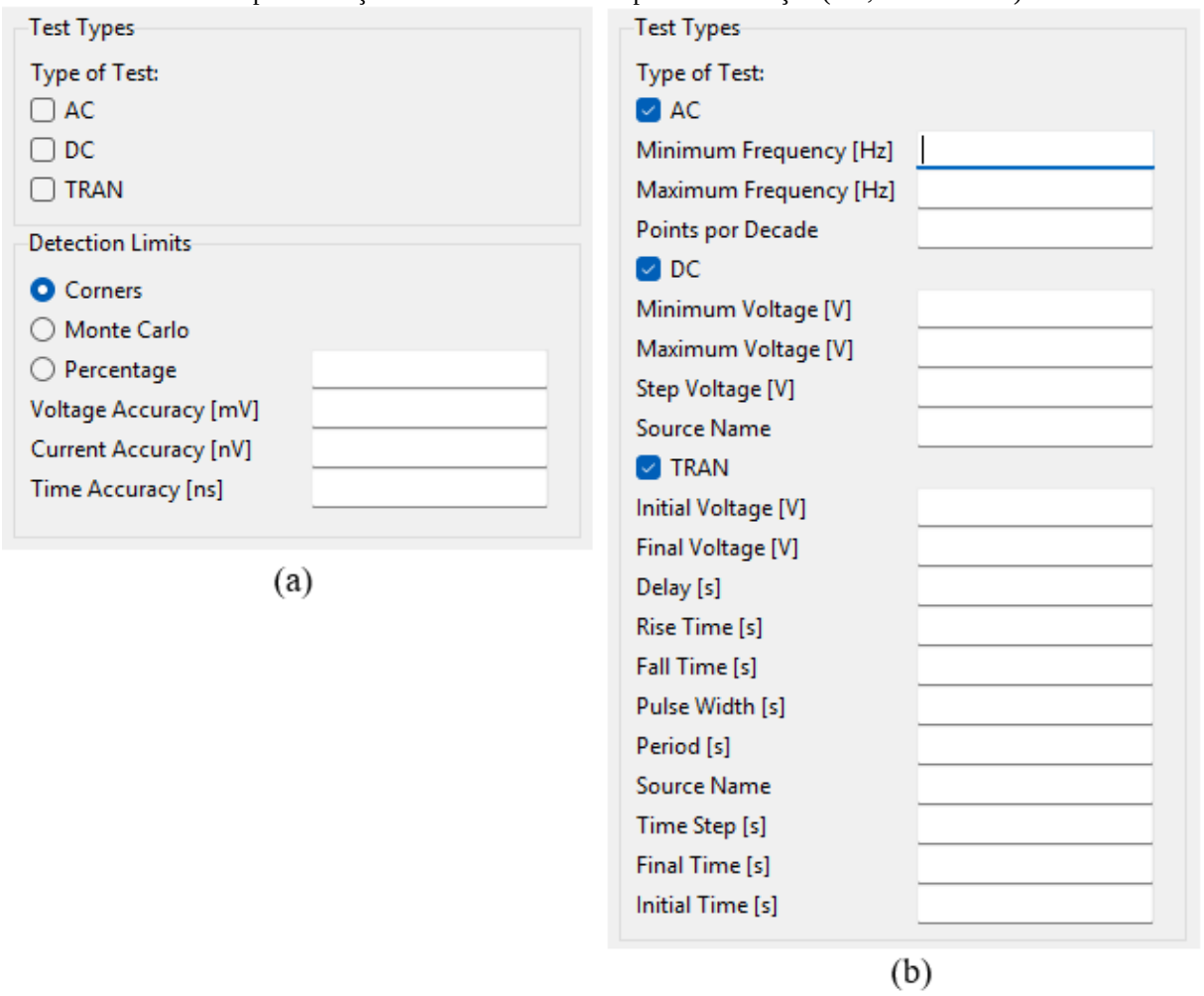
A determinação automática das falhas catastróficas é realizada por meio de um algoritmo que analisa a descrição SPICE do circuito e extrai todos os componentes que o constituem. A partir dessa identificação, são geradas automaticamente todas as combinações possíveis de falhas catastróficas nesses elementos. Para componentes de dois terminais — como resistores, capacitores, indutores e diodos — o algoritmo insere duas falhas básicas: um curto-circuito entre os dois terminais e um circuito aberto em apenas um deles. A inclusão de uma falha de circuito aberto no segundo terminal é considerada redundante, uma vez que resultaria no mesmo comportamento elétrico observado na abertura do primeiro terminal. No caso de componentes com mais de dois terminais, como transistores, o algoritmo insere uma falha de circuito aberto em cada terminal individualmente, além de todas as combinações possíveis de curtos-circuitos entre pares distintos de terminais. Entretanto, o algoritmo realiza uma verificação topológica do circuito, de modo a eliminar automaticamente as falhas de curto-circuito entre terminais que já se encontram conectados na própria descrição SPICE. Assim, evita-se a simulação de falhas sem significado prático, garantindo a geração de um modelo de falhas consistente e livre de redundâncias.

Na parte inferior da Figura 9, é apresentada a descrição SPICE dos subcircuitos responsáveis por representar as falhas catastróficas. Essas descrições são totalmente editáveis, permitindo que o usuário defina os valores das impedâncias utilizadas na modelagem das falhas. Por padrão, a ferramenta adota 10 Ω para representar falhas de curto-circuito e 10 MΩ para falhas de circuito aberto (PETRASHIN; DUALIBE; LANCIONI; TOLEDO, 2013), valores que podem ser ajustados conforme a necessidade do modelo ou a metodologia adotada pelo usuário.

3.1.3 Etapa 2: Parâmetros de Simulação

Nessa etapa, são definidos os parâmetros correspondentes aos tipos de simulação SPICE a serem executados, os quais podem ser AC, DC ou TRAN. A Figura 10 apresenta a interface de configuração inicial desses parâmetros. Para cada tipo de simulação selecionado, são exibidos campos adicionais destinados à configuração das variáveis específicas de cada análise, campos que só ficam visíveis caso o seu tipo de simulação tenha sido selecionado. Sendo assim, a Figura 10 apresenta a interface de configuração dos parâmetros de simulação SPICE, sendo mostradas as visões antes (a) e após a seleção dos tipos de simulação (b).

Figura 10 - Inserção de parâmetros de simulação: (a) Visão inicial antes da seleção dos tipos de simulação. (b) Visão após a seleção consecutiva dos três tipos de simulação (AC, DC e TRAN).



Fonte: elaborado pelo autor.

Nas simulações AC, deve ser delimitada a faixa de frequência de análise, por meio da especificação dos valores de frequência mínima (*Minimum Frequency*) e frequência máxima (*Maximum Frequency*), bem como do número de pontos por década (*Points per Decade*) a serem considerados. Nessa simulação, a amplitude do sinal AC já deve estar especificada na descrição do arquivo SPICE.

Nas simulações DC, são definidos os valores de tensão mínima (*Minimum Voltage*), tensão máxima (*Maximum Voltage*) e o incremento de tensão entre cada ponto da varredura (*Step Voltage*), além do nome da fonte sobre a qual essa varredura é aplicada.

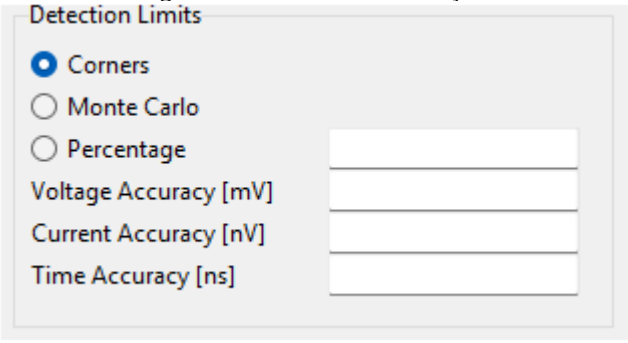
Para as simulações transientes (TRAN), devem ser especificados os parâmetros referentes ao degrau de tensão aplicado à entrada, incluindo a tensão inicial (*Initial Voltage*), tensão final (*Final Voltage*), atraso (*Delay*), tempo de subida (*Rise Time*), tempo de descida (*Fall Time*), largura do pulso (*Pulse Width*) e período (*Period*). Também deve ser indicado o nome da fonte que apresentará o comportamento de degrau descrito, juntamente com os

parâmetros temporais da simulação, como tempo final (*Final Time*), passo temporal (*Time Step*) e, opcionalmente, o tempo inicial (*Initial Time*). Caso este último não seja definido, será considerado o valor padrão igual a zero. No caso das simulações transientes, recomenda-se que os parâmetros sejam configurados de modo a permitir a simulação completa de dois degraus de tensão: um correspondente à variação do valor inicial para o valor final e outro ao retorno do valor final para o valor inicial. Essa configuração possibilita que as análises de configurações de testes realizados posteriormente considerem ambas as transições do pulso, garantindo uma avaliação mais abrangente do comportamento do circuito.

Outro parâmetro a ser escolhido corresponde à metodologia para determinação dos limites de aceitação. Para a ferramenta, esses limites de aceitação correspondem diretamente a tensões nodais para o caso de teste DC; amplitudes ou fase das 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, a ferramenta disponibiliza como principais metodologias para a determinação dos limites de aceitação as análises de *corners*, Monte Carlo ou a definição de limites percentuais em relação à resposta *fault-free*. A Figura 11 ilustra essa escolha.

Figura 11 – Limites de Detecção.



Fonte: elaborado pelo autor.

Na metodologia baseada em *corners*, o algoritmo realiza automaticamente a simulação do circuito sem falhas considerando os quatro *corners* da tecnologia. A partir dessas

simulações, os limites de aceitação das tensões e correntes nodais são determinados de forma automática. Essa determinação é realizada com base na identificação dos valores extremos obtidos entre as quatro respostas de *corners*. Assim, em um teste DC, por exemplo, os limites de aceitação de uma determinada tensão nodal são definidos pelos valores máximo e mínimo de tensão observados nas simulações de *corners* para esse nó.

Por outro lado, quando a metodologia de Monte Carlo é selecionada, o algoritmo executa automaticamente simulações de Monte Carlo do circuito *fault-free*, em uma quantidade definida pelo usuário da ferramenta, com o objetivo de caracterizar estatisticamente as tensões e correntes nodais. A partir dessas simulações, são estimados a média e o desvio padrão de cada grandeza nodal. Dessa forma, os limites de aceitação passam a ser definidos como uma faixa em torno da média, correspondente a mais ou menos um determinado número de desvios padrão, sendo esse número especificado 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.

Por fim, caso se deseje realizar uma análise mais simplificada, sem considerar explicitamente os efeitos de variabilidade do circuito na determinação dos limites de aceitação, a ferramenta também permite a definição desses limites por meio de uma faixa percentual em torno da resposta nominal do circuito *fault-free*. Nesse contexto, as tensões e correntes nodais são consideradas aceitáveis desde que não ultrapassem os limites percentuais de variação previamente estabelecidos em relação à resposta nominal esperada.

Na determinação do dicionário de falhas, alguns detalhes adicionais são fundamentais para garantir que a simulação represente adequadamente o comportamento real dos circuitos durante o teste. Entre esses parâmetros destacam-se as precisões de tensão, corrente e tempo do testador — denominadas, respectivamente, *Voltage Accuracy*, *Current Accuracy* e *Time Accuracy* — conforme ilustrado na Figura 11.

Essas precisões estão diretamente relacionadas ao fato de que, em determinadas condições de simulação, algumas grandezas elétricas podem assumir valores muito pequenos, tanto nas respostas dos circuitos com falhas quanto nos limites de aceitação utilizados nos

testes. Nessas situações, pequenas diferenças numéricas podem provocar falsas detecções de falhas ou, inversamente, falhas não detectadas, devido à resolução limitada dos equipamentos de teste. Assim, ainda que a resposta simulada para um circuito com falha se encontre fora do limite de aceitação teórico, essa diferença pode ser inferior à capacidade mínima de resolução do testador, impossibilitando sua detecção prática. Esse fenômeno pode resultar tanto em uma redução da cobertura de falhas quanto em *yield loss*, causada pela classificação incorreta de circuitos sem falha como defeituosos.

Para mitigar esse problema, a ferramenta desenvolvida foi projetada para considerar explicitamente a precisão do testador, permitindo que esses parâmetros sejam configurados pelo usuário durante a execução. As precisões de tensão e corrente são aplicadas diretamente nos testes DC, nos testes AC de módulo e nos testes transitórios que envolvem a medição do parâmetro de *overshoot*. Nesses casos, a precisão atua como uma margem adicional sobre o limite de aceitação: para que uma falha seja considerada detectável, não basta que a resposta do circuito com falha ultrapasse o limite de aceitação, é necessário que a diferença entre a resposta e o limite seja maior do que a precisão do testador. Dessa forma, assegura-se que o desvio observado seja efetivamente distinguível pelo equipamento de teste. A escolha entre a precisão de tensão ou de corrente é feita conforme a natureza da grandeza medida no respectivo teste.

De forma análoga, a precisão de tempo é utilizada em testes transitórios que envolvem medições de parâmetros temporais, como o tempo de acomodação (*settling time*). Nesses casos, a precisão de tempo também configura uma faixa adicional para além dos limites de aceitação que a resposta de uma falha precisa transpor para ser caracterizada como detectável.

Um caso particular ocorre nos testes AC de fase, nos quais a ferramenta deve considerar simultaneamente duas precisões: a de tempo e a de tensão (ou corrente), dependendo da grandeza medida no nó de observação. A precisão de tempo é convertida para uma precisão de fase, expressa em radianos. Isso ocorre porque um testador com frequência de amostragem limitada também possui uma resolução temporal finita, o que implica em uma diferença mínima de fase necessária para distinguir dois sinais. Assim, a ferramenta considera essa diferença mínima, dependente da frequência do sinal, como o limiar a partir do qual uma falha de fase é considerada detectável.

A segunda precisão considerada nos testes de fase é associada à grandeza elétrica observada (tensão ou corrente). Nesse caso, essa precisão é aplicada ao módulo do sinal, pois influencia indiretamente a detecção de fase. Isso se deve ao fato de que o testador só consegue avaliar a fase de um sinal se este possuir amplitude mínima suficiente para ser corretamente amostrado. Nesse contexto, a ferramenta adota o seguinte critério: uma falha só pode ser

avaliada quanto à fase se o módulo da grandeza elétrica no nó de observação for maior que a precisão configurada ou se o limite mínimo de aceitação no teste AC de módulo dessa mesma grandeza for maior que essa precisão.

Em outras palavras, se o limite mínimo de aceitação do módulo de tensão (ou corrente) for superior à precisão do testador, então um circuito sem falhas apresentará amplitude suficiente para permitir a avaliação da fase. Se esse critério for atendido e o sinal medido na hora do teste não atingir a amplitude mínima exigida, o circuito é automaticamente classificado como contendo falha, independentemente da análise de fase. Se por outro lado forem satisfeitas as condições de amplitude mínima, a ferramenta procede à avaliação de fase, classificando as falhas como detectáveis sempre que a resposta do circuito com falha exceder o limite de aceitação para a fase por uma diferença maior que a precisão temporal convertida em radianos.

Ao término dessa etapa de configuração dos parâmetros de simulação, com todos os valores devidamente configurados, deve ser acionado o botão **"***Run Simulation***"**. Nesse momento, o programa gera um novo arquivo SPICE correspondente ao circuito sem falhas. Esse arquivo apresenta estrutura muito semelhante à do arquivo de entrada utilizado na etapa 1 (Seção 3.1.1), diferenciando-se apenas pela inclusão dos parâmetros de simulação definidos e pela adição de um bloco de controle contendo todos os tipos de simulação configurados (AC, DC e/ou TRAN), juntamente com os comandos necessários para o armazenamento dos respectivos resultados.

Posteriormente, o programa roda um algoritmo em Python 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, simulações Monte Carlo desse mesmo circuito, ou um simples limite percentual em torno da resposta nominal do circuito sem falhas, dependendo do que foi escolhido. 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.

Após a conclusão das simulações, todos os arquivos resultantes gerados pelo SPICE são lidos e seus dados são importados para o ambiente Python, onde é realizada a etapa de pós-processamento. Nesse ambiente, são determinados os limites de aceitação para cada uma das tensões ou correntes nodais, com base nos resultados provenientes das simulações de *corners*, Monte Carlo ou limite percentual.

Com esses limites definidos, o algoritmo procede à análise sequencial das respostas obtidas para os circuitos com falha. Para cada sinal de entrada simulado, são verificadas as tensões ou correntes nodais que permanecem dentro ou fora dos limites de aceitação. A partir dessas informações, o programa constrói uma matriz de falhas (MF) para cada tipo de análise — AC, DC e TRAN —, contendo os dados que indicam quais configurações de teste[1] são capazes de detectar cada uma das falhas modeladas. Em outras palavras, cada matriz de falhas representa, para um domínio de análise específico, as condições em que as falhas provocam respostas fora dos limites, considerando também as precisões de Tensão, Corrente e Tempo.

Essas matrizes são fundamentais para os cálculos realizados posteriormente pela janela "Results" (Seção 3.2), sendo, portanto, importante descrever sua estrutura de forma detalhada. Cada matriz de falhas possui três dimensões:

- O primeiro índice corresponde aos sinais de entrada simulados;
- O segundo índice corresponde às variáveis avaliadas, ou seja, tensões ou correntes nodais, abrangendo tanto os nós de saída quanto os nós internos;
- O terceiro, às falhas consideradas no modelo.

Dessa forma, cada elemento da matriz é preenchido com um valor binário que indica se determinada falha é detectada (valor 1) ou não (valor 0) por uma configuração de teste específica.

Sejam definidos os vetores:

- $I = [I_1, I_2, \ldots, I_s]$, contendo os sinais de entrada simulados;
- $C = [C_1, C_2, \ldots, C_n]$, representando as variáveis analisadas do circuito;
- $F = [F_1, F_2, \ldots, F_m]$, representando as falhas modeladas.

Assim, "s" representa o número total de pontos simulados em SPICE (dependente do passo adotado nas simulações), "n" o número total de variáveis (tensões e correntes nodais) do circuito e "m" o número total de falhas do modelo. Cada tipo de simulação (AC, DC e TRAN) possui o seu próprio vetor de entradas I, uma vez que as grandezas analisadas diferem entre os domínios: tensões contínuas para DC, frequências para AC e parâmetros transientes para TRAN.

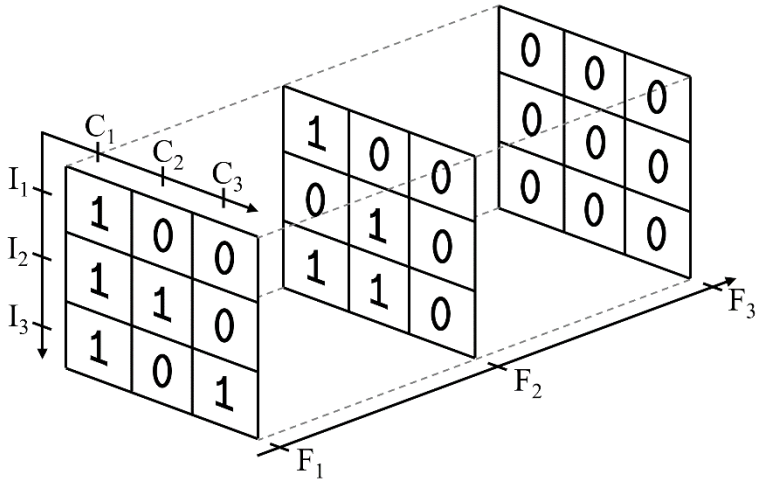
Designando-se a matriz de falhas por MF, define-se que o elemento MF(x,y,z) assume o valor 1 quando a falha F_x é detectada por meio da medição realizada da váriável C_y, sob a

[1] Configuração de teste corresponde à combinação de variável (tensão ou corrente do nó que será medido na execução do teste) e sinal de entrada utilizados em um teste.

condição de que o sinal de entrada aplicado seja I_z. Caso contrário, ou seja, quando a falha não é detectada nessa configuração de medição e excitação, o elemento correspondente da matriz assume o valor 0.

A Figura 12 ilustra um exemplo simplificado de matriz de falhas, resultante de uma simulação hipotética de um circuito contendo três variáveis, três sinais de entrada e três falhas modeladas. A análise da distribuição dos valores “1” ao longo da matriz permite identificar as combinações de sinais e medições que resultam na detecção de cada falha. Observa-se que a falha F_1 é detectada por meio da medição da variável C_1 para todos os sinais de entrada considerados. Entretanto, a detecção dessa mesma falha a partir de medições realizadas em outras variáveis depende da aplicação de sinais de entrada específicos. No caso da falha F_2, verifica-se que ela não é identificada através da medição da variável C_3; contudo, pode ser detectada em determinadas configurações de teste associadas às demais variáveis do circuito. Por fim, constata-se que a falha F_3 não é detectada em nenhuma das configurações de teste analisadas.

Figura 12 - Exemplo de matriz de falhas.



Fonte: elaborado pelo autor.

Após a geração das matrizes de falhas, o algoritmo salva um arquivo no formato “.npz” na mesma pasta onde está o arquivo de descrição SPICE do circuito (Seção 3.1.1) juntamente com todos os arquivos e resultados de simulações SPICE das falhas. Esse formato de arquivo ‘.npz’ é nativo do ambiente Python e utilizado para armazenar estruturas de dados de forma compacta e eficiente. Nesse arquivo, são salvas todas as informações necessárias para as etapas posteriores de análise, incluindo:

- As tensões nodais resultantes das simulações SPICE;

- Os limites de aceitação determinados;
- As matrizes de falhas correspondentes a cada tipo de simulação (AC, DC e TRAN);
- Demais variáveis relacionadas aos parâmetros das simulações.

Inicialmente, todas as tensões e correntes simuladas do circuito são consideradas como possíveis variáveis a serem medidas nos testes, de modo a permitir a análise da cobertura máxima de falhas e a avaliação da relevância de se disponibilizar acesso à medição de nós internos do circuito. No entanto, na janela “Results” é possível selecionar quais desses nós devem efetivamente compor o conjunto de pontos de teste. Dessa forma, a ferramenta passa a considerar apenas os nós selecionados na determinação do melhor conjunto de testes. Essa abordagem possibilita comparar diferentes estratégias de acesso e medir o impacto da restrição de pontos de teste sobre a capacidade de detecção de falhas, sem a necessidade de reexecutar as simulações SPICE, o que representa uma economia significativa de tempo de processamento.

Por fim, a janela “AnalogCircuitsTest” termina sua execução e se fecha automaticamente. Desde momento em diante, o arquivo ‘.npz’ de resultados pode ser acessado pela execução da janela “Results”, que será descrita na Seção 3.2, sempre que se quiser consultar o comportamento de algum nó do circuito na presença ou não de alguma falha específica, ou verificar o conjunto dos melhores testes juntamente com sua cobertura de falhas.

3.2 JANELA “RESULTS”

De maneira geral, a janela “Results” é responsável pela leitura e processamento do arquivo .npz, que contém todos os resultados das simulações SPICE gerados previamente pela janela “AnalogCircuitsTest”, descrita na Seção 3.1. A partir dessas informações, a ferramenta realiza a análise completa dos dados e determina quais são as melhores configurações de teste possíveis dentro das condições estabelecidas pelo usuário. Além disso, a janela “Results” também funciona como uma interface visual para consulta gráfica de todos os resultados das simulações SPICE.

Durante o seu funcionamento, a janela exibe uma série de informações que auxiliam na interpretação dos resultados e na tomada de decisão quanto à eficiência das configurações de teste. Entre os dados apresentados, estão os parâmetros utilizados nas simulações SPICE, a máxima cobertura de falhas que pode ser alcançada considerando todas as condições simuladas, e o conjunto de testes considerado ótimo, ou seja, aquele que proporciona a maior capacidade de detecção de falhas com o menor número de medições necessário.

Ao ser executada, a janela “Results” demanda que o usuário selecione manualmente o arquivo de resultados que deseja analisar. Essa metodologia permite que análises distintas sejam

realizadas posteriormente, sem a necessidade de repetir o processo de simulação SPICE, economizando assim tempo e recursos computacionais.

Para uma explicação mais detalhada de seu funcionamento, a presente Seção está organizada da seguinte forma:

- Seção 3.2.1: descreve o procedimento adotado para a determinação da máxima cobertura de falhas possível, considerando todas as combinações de sinais de entrada, nós e falhas modeladas;
- Seção 3.2.2: apresenta o método de escolha do melhor conjunto de testes, detalhando os critérios de otimização e os parâmetros empregados na seleção das configurações mais eficientes;
- Seção 3.2.3: oferece uma visão geral da interface gráfica da janela “Results”, destacando as principais áreas e elementos informativos apresentados ao usuário;
- Seção 3.2.4: explica quais parâmetros são interativos, ou seja, aqueles que podem ser modificados pelo usuário diretamente na interface, provocando o recálculo da cobertura de falhas e a consequente redefinição do conjunto ótimo de testes;
- Seção 3.2.5: apresenta funcionalidades adicionais da ferramenta, voltadas à análise de combinação de testes no conjunto final, visando possíveis incrementos na cobertura de falhas sem alterar significativamente o tempo de teste ou alcançar a mesma cobertura de falha com um conjunto de teste que demande menos tempo de execução e seja, portanto, mais barato.

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 a partir da matriz de falhas, conforme apresentado na Equação (1):

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

onde “S” corresponde ao número total de pontos simulados no SPICE, o que depende do passo de simulação utilizado e dos tipos de simulação adotados (AC, DC e TRAN); “N” corresponde ao número total de variáveis do circuito; e MF é o valor binário da posição correspondente da matriz de falhas, como já mencionado.

Basicamente, o índice da primeira dimensão da matriz de falhas é fixado no índice da falha em questão (z), 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. Esse procedimento precisa ser feito separadamente para cada um dos tipos de matrizes de falhas, AC, DC e TRAN para que no final os resultados sejam somados.

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), assim como é mostrado na Equação (2).

$$CF = \frac{N\acute{u}mero\ de\ falhas\ detectaveis}{N\acute{u}mero\ total\ de\ falhas\ do\ modelo} . 100\% \qquad (2)$$

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 realizada por meio de um algoritmo sequencial de otimização, implementado em Python, cujo objetivo é selecionar, entre todas as condições de teste simuladas, aquelas que proporcionem a maior cobertura de falhas possível

com o menor número de testes necessários. Assim, o resultado final do processo é um conjunto mínimo de testes capaz de detectar todas as falhas classificadas como detectáveis pelo modelo.

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 impõe 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 últimos índices da matriz de falhas (MF) correspondam aos nós do circuito e aos sinais de entrada simulados, basta fixá-los nos valores específicos de cada configuração de teste e somar os valores desta 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 da grandeza C_y é obtido de acordo com a Equação (3):

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

Como o valor NFD contabiliza apenas as falhas detectadas por cada configuração de teste de forma individual, sem considerar a combinação de testes no conjunto final, ele é utilizado como terceiro critério de escolha. O segundo critério, por sua vez, deriva de uma variação desse parâmetro, denominada NFD'. Essa grandeza contabiliza somente o número de novas falhas detectadas por cada teste, desconsiderando aquelas que já são detectadas pelos testes previamente selecionados. Para isso, NFD' é inicialmente igual a NFD; entretanto, o algoritmo vai subtraindo os valores de NFD' de um em um para cada falha que é já é detectável por um teste previamente escolhido para o conjunto, sendo assim, NFD' é calculado através da Equação (4):

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

A 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 da faixa de possíveis sinais de entrada a 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 contabiliza o número de *steps* de simulação do sinal de entrada que resultam na detecção do mesmo conjunto de falhas. Como todos esses estímulos de entrada conduzem a respostas equivalentes com relação à detecção de falhas, eles são considerados redundantes e passam a representar um único candidato a teste. Se um teste contabilizar uma faixa maior de estímulos de entrada que levem a mesma detecção de falhas, essa terá a preferência, sendo o valor central da faixa de possíveis sinais de entrada escolhido de fato para ser utilizado no teste.

A Figura 13 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 13 - Representação do algoritmo para determinação do melhor conjunto de testes.

```
1  Conjunto de Testes = ø;
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 = ø;
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 =
                         (Is,Nn);
15                     Melhor CF total = NFD(Is,Nn);
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 = (Is,Nn);
21                         Melhor CF total = CF total;
22                 else if CF total == Melhor CF
                     total then
23                     Contabiliza os ranges de
                         possíves sinais de entrada;
24                     if Range da configuração de
                         teste (Is,Nn) > Range da
                         melhor configuração de teste
                         then
25                         Melhor CF = CF;
26                         Melhor configuração de
                             teste = (Is,Nn);
27                         Melhor CF total = CF
                             total;
28     Conjunto de Testes = Conjunto de Testes ∪
         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 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 representa 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'.

Além disso, foram analisadas estratégias complementares de combinação de testes, com o objetivo de reduzir o tempo total de execução do conjunto de ensaios e, consequentemente, diminuir o custo global do processo de teste. Essas estratégias exploram a possibilidade de executar, de forma conjunta, testes cujas condições de estímulo e medição possam ser compartilhadas, evitando redundâncias e aproveitando transições já presentes na sequência de excitação. As abordagens propostas são descritas em maior detalhe na Seção 3.2.3.

3.2.3 Análises de Possíveis Combinações de Teste no Conjunto Final

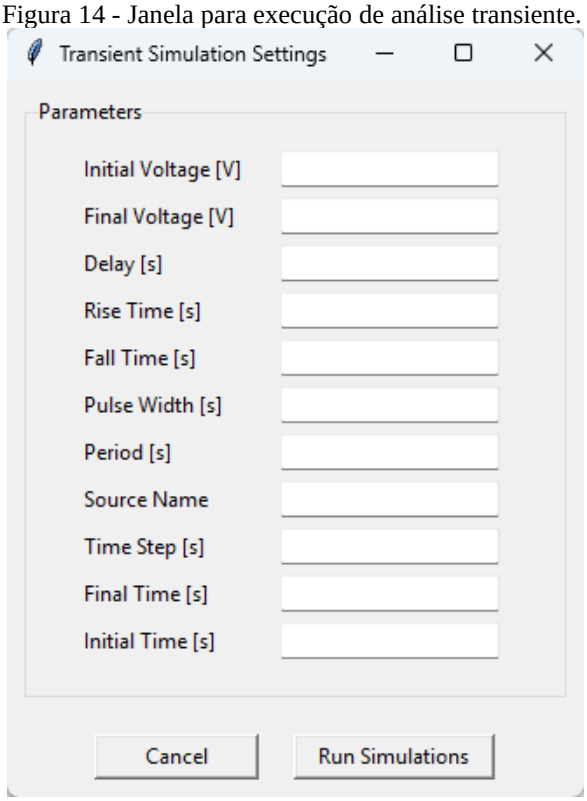
Dependendo das condições simuladas, o conjunto final de testes determinado pela ferramenta pode incluir testes de diferentes naturezas, tais como testes DC, AC ou transientes. No caso dos testes transientes, são avaliados parâmetros como o tempo de acomodação do sinal e o *overshoot*, os quais requerem a aplicação de um degrau de tensão na entrada do circuito. Esse degrau, por sua vez, é formado por duas tensões DC, uma tensão inicial e uma tensão final.

Essa característica permite a combinação natural entre testes DC e testes transientes. Por exemplo, ao aplicar uma tensão DC na entrada e aguardar o regime permanente na saída, realiza-se um teste DC. Em seguida, ao aplicar um degrau para uma segunda tensão DC, executa-se um teste transiente. Após o estabelecimento do novo regime permanente, pode-se realizar um segundo teste DC. Assim, a execução de um único teste transiente pode ser associada a dois testes DC adicionais praticamente sem acréscimo no tempo total de simulação. De maneira análoga, a transição de tensão necessária entre dois testes DC pode ser aproveitada para a realização de um teste transiente.

Com base nesse princípio, a primeira estratégia de otimização consiste justamente na combinação de dois testes DC com um teste transiente. Como a transição entre dois níveis de tensão DC já ocorre naturalmente durante a aplicação da sequência de testes, a análise transiente pode ser incluída sem aumento significativo no tempo de execução. Dessa forma, é possível elevar a cobertura de falhas ou reduzir o número total de testes independentes, preservando a eficiência temporal do conjunto.

A ferramenta desenvolvida incorpora essa estratégia por meio de uma funcionalidade específica acessível na janela Results, denominada "*Transient Analysis*". Esse recurso é habilitado somente quando o conjunto de testes selecionado pelo algoritmo apresentado na Seção 3.2.2 contém pelo menos dois testes DC, condição necessária para a formação do degrau de tensão. Quando ativado, o programa abre a janela ilustrada na Figura 14, na qual já fica pré-selecionado quais tensões DC serão utilizadas como nível inicial e final do degrau (tensões de testes DC já presentes no conjunto de testes). Além disso, deve-se definir parâmetros adicionais relevantes para a execução das simulações SPICE no modo transiente.

Figura 14 - Janela para execução de análise transiente.



Fonte: elaborado pelo autor.

Ao pressionar o botão "*Run Simulation*" na janela da Figura 14, 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.

Essa funcionalidade permite, portanto, aperfeiçoar o conjunto de testes, aproveitando transições já existentes entre tensões DC para incorporar análises transientes adicionais de forma eficiente e sem penalidades significativas no tempo de simulação.

A segunda estratégia de otimização considera a combinação dos testes AC de módulo e fase. A medição da resposta do sinal AC de um circuito fornece simultaneamente informações sobre a amplitude e a fase, permitindo que ambos os parâmetros sejam avaliados no mesmo teste. Em consequência, a execução combinada desses testes pode reduzir o tempo total de execução do conjunto, sem comprometer a cobertura de falhas alcançada.

Diferentemente da estratégia de combinação aplicada aos testes DC e transientes, a combinação entre testes AC de módulo e fase não exige a geração ou simulação de novos arquivos SPICE. O dicionário de falhas AC já armazena, para cada frequência simulada, os resultados referentes ao módulo e à fase. Assim, quando a ferramenta é configurada para permitir a utilização de ambos os tipos de teste AC, a análise das combinações é realizada automaticamente. Caso o usuário restrinja a busca, desabilitando a utilização de módulo, de fase, ou de ambos, as combinações deixam de ser consideradas.

Uma vez habilitada a possibilidade de executar ambos os tipos de teste AC, a ferramenta procede da seguinte forma. A seleção do conjunto de testes é realizada sequencialmente, obedecendo aos critérios e à lógica apresentados na Seção 3.2.2. Quando um teste AC é escolhido e incorporado ao conjunto, o algoritmo verifica se a inclusão do teste complementar (módulo ou fase, correspondente à mesma frequência) oferece a detecção de pelo menos uma nova falha além daquelas já detectadas pelos testes selecionados até o momento. Se houver benefício adicional, o teste complementar é imediatamente adicionado ao conjunto. Caso contrário, ele é descartado.

Com essa estratégia, o conjunto de testes final já é construído levando em conta apenas as combinações AC que efetivamente impactam na detecção de falhas do conjunto. Além disso,

a inclusão dos testes complementares influencia diretamente a escolha dos próximos testes, uma vez que as falhas detectadas por eles deixam de ser consideradas como pendentes nas etapas subsequentes.

Entretanto, um teste AC complementar que tenha sido útil no momento de sua inserção pode tornar-se redundante ao final do processo de seleção. Isso ocorre, por exemplo, quando o teste complementar detecta algumas falhas adicionais em relação aos testes anteriores, mas não detecta falhas mais críticas que orientam a escolha dos testes subsequentes. Nessa situação, os testes seguintes podem vir a detectar todas as falhas que motivaram inicialmente a inclusão do teste complementar, tornando-o desnecessário no conjunto final.

Para evitar que tais redundâncias permaneçam no conjunto de testes, a ferramenta realiza uma segunda execução completa do algoritmo de seleção do conjunto de testes, desta vez sem considerar a possibilidade de combinações entre testes AC. Ao término, são obtidos dois conjuntos otimizados para as mesmas condições de entrada:

- Conjunto 1: considerando combinações AC;
- Conjunto 2: sem considerar combinações AC.

Em seguida, o algoritmo compara ambos os conjuntos. Primeiramente, computa-se o número de testes do Conjunto 1, descontando os testes AC complementares (isto é, aqueles adicionados exclusivamente em função da combinação). Esse valor é então comparado ao número de testes do Conjunto 2. Se os valores forem iguais, a ferramenta seleciona o Conjunto 2, pois envolve menor número de medições físicas (uma vez que não utiliza pares módulo–fase). Entretanto, se o Conjunto 2 apresentar maior número de testes, então o Conjunto 1 é selecionado, pois a inclusão dos testes complementares AC reduziu o número total de testes necessários ou, no mínimo, não trouxe prejuízo à eficiência global.

Dessa forma, a ferramenta assegura que as combinações de testes AC sejam utilizadas apenas quando efetivamente contribuírem para a redução do conjunto final de testes, evitando redundâncias e garantindo a seleção do conjunto mais eficiente em termos de cobertura de falhas e custo de execução.

Uma terceira e última estratégia de combinação considera a utilização conjunta dos testes transitórios de *overshoot* e tempo de acomodação. Essa abordagem explora o fato de que um mesmo degrau de tensão aplicado à entrada do circuito permite a extração sequencial dessas duas grandezas — *overshoot* e tempo de acomodação — sem a necessidade de estímulos adicionais. O mecanismo de seleção segue a mesma lógica adotada para a combinação de testes AC de módulo e fase. Assim, sempre que um teste transiente é incorporado ao conjunto, avalia-se se o seu teste complementar, correspondente à medição da outra grandeza (*overshoot* ou

tempo de acomodação) associada ao mesmo degrau de tensão, contribui para a detecção incremental de falhas. Caso apresente ganho de cobertura, ele é incluído; caso contrário, é descartado.

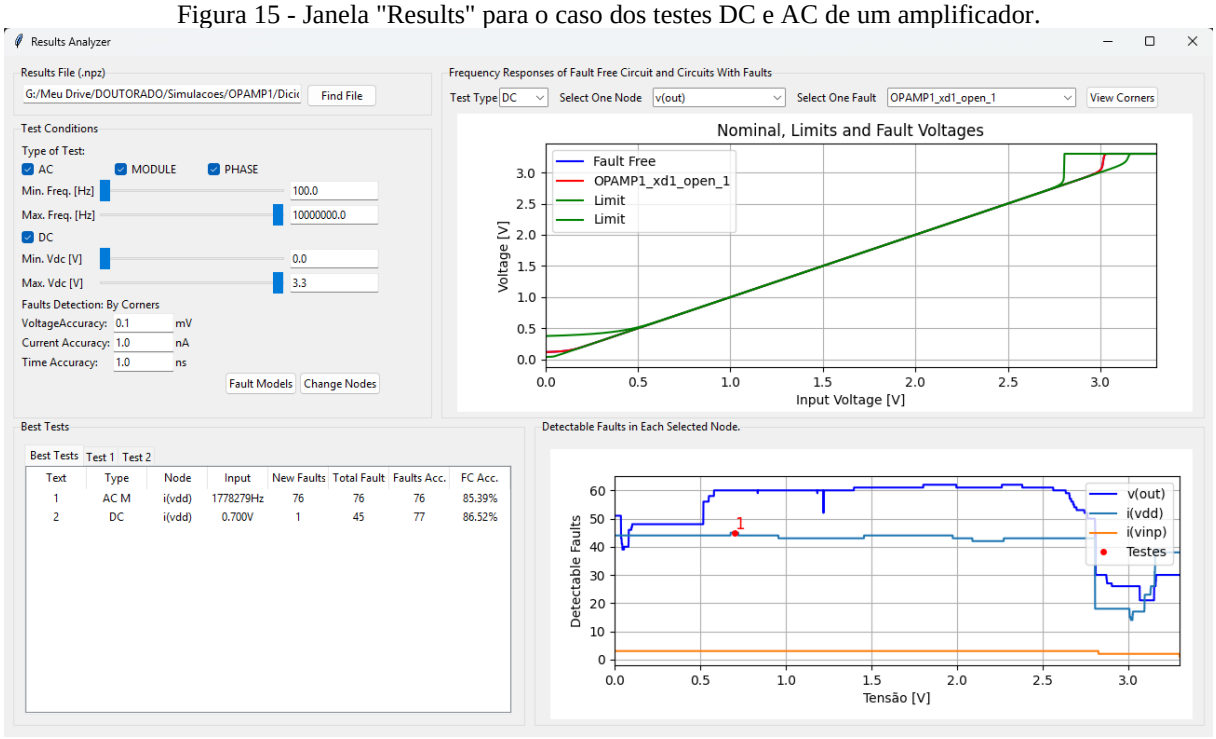
Ao final do processo de seleção do conjunto de testes, a ferramenta também compara o conjunto resultante com aquele formado sem a aplicação dessa estratégia de combinação, de modo a verificar se a inclusão dos testes complementares efetivamente trouxe benefícios, através da redução do tempo total de execução do conjunto. Somente então é definido qual dos dois conjuntos (considerando ou não a combinação de testes transitórios) será selecionado como o conjunto final otimizado.

3.2.4 Exibição dos Resultados

Durante a descrição do funcionamento da *toolbox*, serão mostradas imagens das suas janelas em execução. Para esses exemplos, meramente ilustrativos, foi utilizado o amplificador operacional “OPAMP1”, circuito que faz parte do *Benchmark* do grupo IEEE P2427 e que será descrito em maiores detalhes na Seção 4.5.1. Sendo assim, para essas figuras, contidas na presente seção, o intuito principal não é demonstrar os resultados contidos nelas e, sim, ilustrar as funções que a janela oferece e a maneira como os resultados são expostos.

A Figura 15 apresenta o formato gráfico da janela *Results*. Para fins de ilustração do visual da ferramenta, foi selecionado um exemplo que contempla testes dos tipos DC e AC, razão pela qual o conjunto ótimo de testes exibido inclui ambos os tipos de sinais de entrada. No entanto, os gráficos mostrados nesta figura referem-se exclusivamente a resultados de sinais DC, utilizados como exemplo representativo da funcionalidade da interface. A janela é organizada em quatro quadrantes principais, cada um dedicado à exibição de um conjunto distinto de informações.

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



Fonte: elaborado pelo autor.

O primeiro quadrante, localizado no canto superior esquerdo, apresenta as condições de simulação correspondentes à etapa anterior do processo, descrita na janela *AnalogCircuitTest* (Seção 3.1). Nesse espaço, o usuário pode visualizar o nome e o caminho do arquivo de resultados atualmente em análise, bem como os parâmetros utilizados nas simulações SPICE. Esses parâmetros podem ser editados manualmente, permitindo restringir o escopo das análises e desconsiderar determinadas configurações de teste que poderiam ser selecionadas automaticamente pelo algoritmo durante a composição do conjunto final de testes.

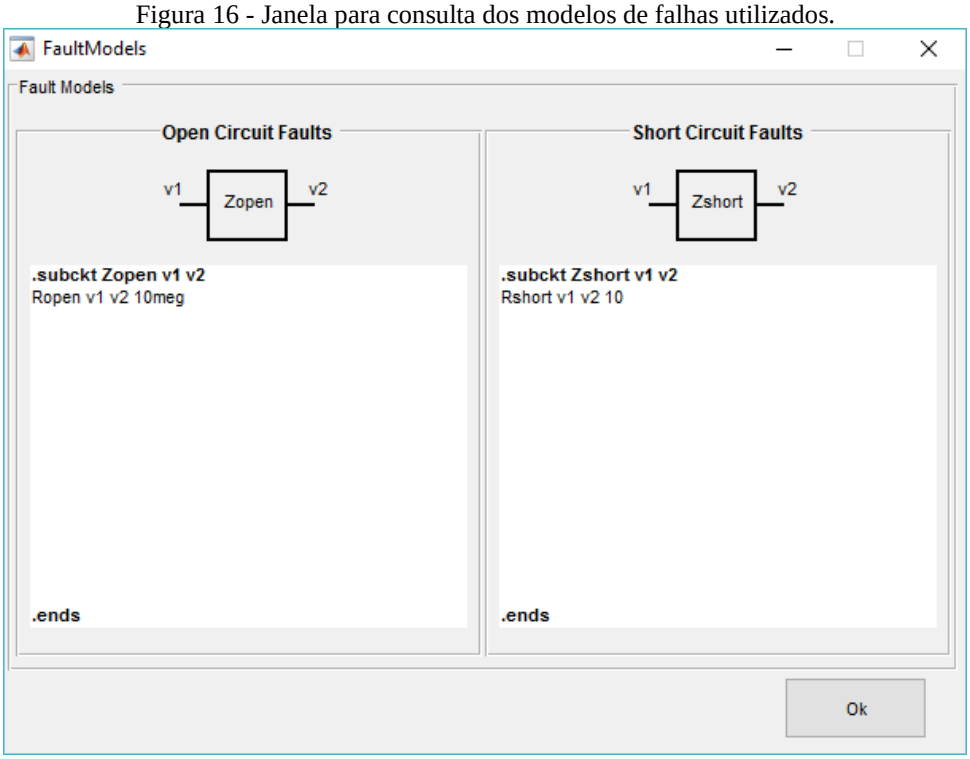
O segundo quadrante (Figura 15), localizado no canto superior direito, exibe as respostas nodais obtidas nas simulações SPICE. Por meio do menu suspenso identificado como *Test Type*, o usuário seleciona o tipo de teste desejado — no exemplo, um teste DC. Em seguida, utilizando o menu *Select One Node*, é possível escolher o nó do circuito cuja resposta se deseja visualizar. O gráfico intitulado *Nominal, Limits and Fault Voltages* mostra a tensão ou corrente simulada no circuito nominal (sem falhas) em função do sinal de entrada, juntamente com os limites de aceitação empregados nas análises de detecção de falhas. Nesse mesmo gráfico, também é exibida a resposta do circuito com falha, que pode ser selecionada por meio do menu *Select One Fault To Plot*.

Os dois quadrantes inferiores concentram as informações relacionadas à detecção de falhas e à composição do conjunto de testes. No quadrante inferior direito, é apresentado o

gráfico denominado *Detectable Faults in Each Selected Node*, que mostra as curvas do número de falhas detectadas em função dos sinais de entrada para cada nó selecionado no conjunto final de testes e para o nó selecionado no segundo quadrante (superior direito). Os pontos marcados com círculos indicam os testes selecionados para compor o conjunto de teste. Finalmente, no quadrante inferior esquerdo, é exibida uma tabela contendo o conjunto final dos melhores testes, sintetizando os resultados obtidos pelo algoritmo de seleção.

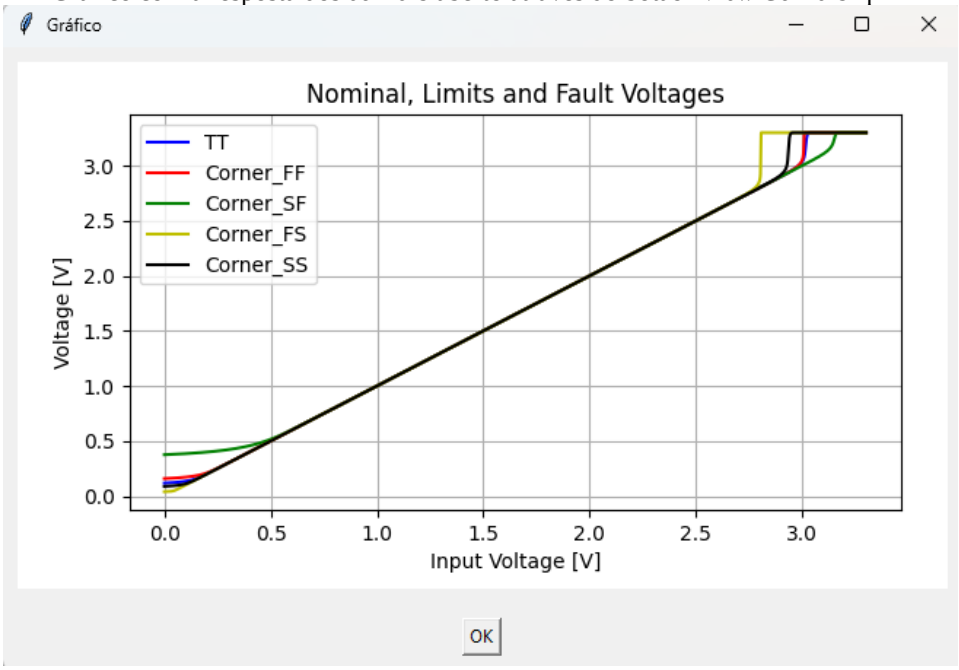
Assim como pode ser visto na Figura 15, existem alguns botões possíveis de serem pressionados. Alguns deles se referem apenas a consultas a informações, como por exemplo, o botão chamado de "*Fault Models*", que, ao ser pressionado, abre uma janela semelhante à da Figura 16, onde pode-se visualizar os 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 "Vout" para sinais de entrada AC. O nó em questão deve estar selecionado no *pop-up menu* "*Select One Node*" 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.

Figura 16 - Janela para consulta dos modelos de falhas utilizados.



Fonte: elaborado pelo autor.

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

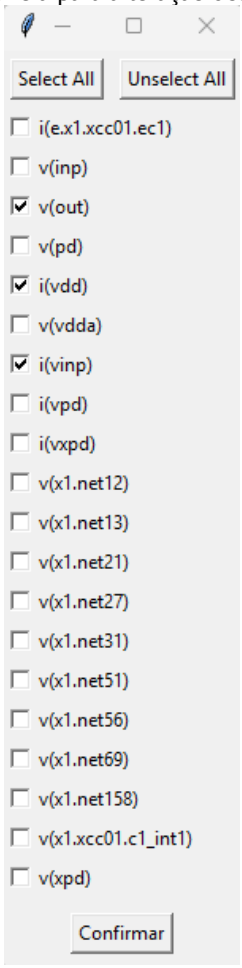


Fonte: elaborado pelo autor.

3.2.5 Possibilidades de Alterações das Análises

Um dos recursos disponíveis na interface da ferramenta, que permite modificar as condições de teste e, consequentemente, os resultados obtidos, é o botão denominado “*Change Nodes*”, responsável pela alteração dos nós de teste. Ao ser acionado, este botão abre uma janela semelhante à mostrada na Figura 18, na qual o usuário pode selecionar novos nós para análise. Após a seleção, basta confirmar a escolha por meio do botão “OK”, e o programa executa automaticamente o reprocessamento dos resultados, considerando os novos nós especificados, determinando um novo conjunto de testes.

Figura 18 - Janela para alteração dos nós de teste.



Fonte: elaborado pelo autor.

Além da seleção dos nós, o usuário também pode ajustar as precisões de tensão, corrente e tempo do testador. Essa modificação implica na redefinição completa do dicionário de falhas, uma vez que as tolerâncias afetam diretamente o critério de detecção, resultando, portanto, em uma nova determinação do conjunto de testes. Consequentemente, um novo conjunto de testes otimizado também é determinado.

Outro parâmetro ajustável refere-se às faixas dos sinais de entrada que compõem o intervalo de análise. No caso de testes AC, é possível alterar as frequências mínima e máxima consideradas; já nos testes DC, podem ser modificados os valores de tensão mínima e máxima. Essas alterações provocam a reavaliação do conjunto de testes, sem, contudo, modificar o dicionário de falhas previamente estabelecido — apenas restringem os valores dentro deste dicionário que o algoritmo pode selecionar para compor o novo conjunto ótimo de testes.

Também é possível, por meio dos botões de seleção disponíveis na interface, restringir as configurações de teste consideradas na determinação do melhor conjunto de testes. Essa funcionalidade permite excluir determinados tipos de ensaios, conforme o interesse da análise.

Por exemplo, pode-se desconsiderar configurações de teste DC ou AC de forma global, ou ainda restringir apenas subtipos específicos dentro das análises AC, como os testes de módulo ou, de maneira análoga, os testes de fase. Essa flexibilidade possibilita ao usuário avaliar diferentes cenários de composição do conjunto de testes, observando o impacto da exclusão de determinados tipos de estímulos sobre a cobertura de falhas e o tempo total de execução.

Em qualquer uma dessas modificações, o programa realiza automaticamente o recálculo dos resultados de acordo com as novas condições definidas. No entanto, as frequências e tensões de entrada especificadas não podem ultrapassar o intervalo originalmente utilizado nas simulações SPICE. Caso seja necessário analisar faixas de valores fora desse limite, é indispensável recorrer à janela “AnalogCircuitsTest”, descrita na Seção 3.1, a partir da qual podem ser executadas novas simulações SPICE com intervalos de análise atualizados

3.3 SIMULAÇÕES DE TESTES FUNCIONAIS

Na Seção 2.1.2, foram comparados os testes funcionais e estruturais quanto às suas características de qualidade e custo de execução. Nesse contexto, destaca-se que, embora os testes funcionais geralmente apresentem custos superiores aos testes estruturais, eles podem proporcionar uma cobertura de falhas maior. A cobertura de falhas nos testes estruturais, contudo, pode ser significativamente aprimorada mediante a utilização de ferramentas automatizadas para a geração de vetores de teste, como a ferramenta desenvolvida e descrita neste Capítulo 3.

Em função disso, um novo algoritmo foi desenvolvido para analisar simulações de testes funcionais, permitindo, assim, a comparação direta entre os resultados destes e os testes estruturais tradicionalmente produzidos pela ferramenta.

Cabe ressaltar ainda que um dos principais diferenciais da metodologia de testes estruturais proposta é a independência em relação ao conhecimento específico do circuito. Tal independência decorre do fato de que os limites de detecção são automaticamente determinados por meio de simulações de variações de processo, seja através de análise de *corners* ou de simulações do tipo Monte Carlo. Por outro lado, os testes funcionais usualmente demandam conhecimento especializado sobre o circuito, uma vez que cada parâmetro funcional possui valores típicos e limites de aceitação estabelecidos em seu respectivo *datasheet*.

Entretanto, os estudos de caso empregados referem-se a componentes projetados em ambiente acadêmico, sem fins comerciais, e, consequentemente, sem *datasheets* disponíveis. Dessa forma, tornou-se necessária a determinação dos limites de aceitação via simulação. Assim, os algoritmos de injeção de falhas descritos na Seção 3.1 permaneceram sendo

utilizados para a avaliação dos testes funcionais; porém, durante a análise dos resultados, o algoritmo passou a avaliar os parâmetros funcionais, ao invés das tensões nodais.

A ferramenta foi configurada para a análise de circuitos amplificadores, sendo selecionados os seguintes parâmetros funcionais para a avaliação: ganho DC, *Slew Rate* (SR) e frequência de corte (-3 dB). Para essa finalidade, a ferramenta injeta falhas no circuito e executa automaticamente dois tipos de simulações: Análise AC e Análise Transiente. A Análise AC é empregada tanto para a determinação da frequência de corte quanto para a estimativa do ganho DC, através da análise do comportamento do circuito em baixas frequências. Já a Análise Transiente é realizada mediante a aplicação de um degrau de tensão na entrada, possibilitando a determinação do *Slew Rate*.

Dessa forma, os três parâmetros funcionais são determinados tanto para o circuito *fault free* quanto para todos os circuitos contendo, individualmente, cada uma das falhas definidas no modelo. Além disso, a ferramenta executa simulações de variação de processo, seja via análise de *corners* ou Monte Carlo, conforme selecionado pelo usuário, a fim de estabelecer os limites de aceitação para cada parâmetro funcional.

Especificamente, caso a análise de *corners* seja escolhida, o algoritmo executa quatro simulações AC e Transiente correspondentes aos extremos tecnológicos para o circuito livre de falhas, considerando os piores casos como limites de aceitação nos testes. Alternativamente, na escolha pela simulação de Monte Carlo, o algoritmo realiza o número de simulações estipulado pelo usuário, determinando os limites de aceitação com base nos conceitos estatísticos de média e desvio padrão.

Por fim, o algoritmo calcula a cobertura de falhas de cada teste funcional individualmente, bem como do conjunto de testes que inclui a medição dos três parâmetros funcionais. Essa abordagem viabiliza a comparação entre os resultados obtidos pelos testes funcionais e os testes estruturais tradicionalmente determinados pelo algoritmo de automatização, ampliando a avaliação da eficácia de cada metodologia.

3.4 EXPORTAÇÃO DA FERRAMENTA DE MATLAB PARA PYTHON

Um dos principais objetivos deste trabalho consiste na ampliação do acesso e da usabilidade da ferramenta de análise de configurações de teste, visando sua utilização por um público mais amplo e independente da necessidade de licenciamento de softwares proprietários. Nesse contexto, a dependência da plataforma MATLAB, requerida para a execução dos códigos originais, representa uma limitação significativa, especialmente considerando que a integração

com o simulador HSPICE™ (Synopsys) ocorre por meio da *toolbox* “HSPICE Toolbox for MATLAB” (PERROTT, 2011; ZILCH, 2021).

Dessa forma, optou-se pela exportação da ferramenta para uma linguagem de programação de uso livre e ampla adoção: Python. O processo de migração iniciou-se com a conversão dos códigos responsáveis pela automação da injeção de falhas, garantindo a preservação de todas as funcionalidades da ferramenta anterior feita em MATLAB. Nesta etapa, as principais adequações concentraram-se em aspectos sintáticos e estruturais das linguagens, uma vez que as funcionalidades lógicas permaneceram inalteradas. Assim, tanto a versão original em MATLAB quanto a nova versão em Python permitem a injeção automatizada de falhas e a execução de simulações via qualquer simulador compatível com a linguagem SPICE, desde que os comandos de configuração e exportação dos resultados sejam corretamente definidos nos arquivos de entrada (descrição SPICE do circuito *fault free*).

Deve-se destacar que diferentes simuladores SPICE, como HSPICE, LTSPICE e NGSPICE, apresentam variações na sintaxe dos comandos necessários para a execução de simulações e, principalmente, na geração e estruturação dos arquivos de saída. Essas diferenças impactam diretamente a leitura e o processamento automatizado dos resultados, razão pela qual a versão original da ferramenta em MATLAB estava restrita ao uso do HSPICE, enquanto a versão em Python foi projetada para operar com simuladores *open source*, como LTSPICE e NGSPICE.

Assim, a adaptação das funcionalidades descritas na Seção 3.1.3, responsáveis pela leitura dos resultados e determinação do dicionário de falhas, demandou um redesenvolvimento substancial. Este redesenvolvimento objetivou a compatibilização da ferramenta com os formatos de arquivos gerados pelos simuladores NGSPICE e LTSPICE. Apesar das diferenças na configuração dos comandos de exportação — especialmente no que se refere à declaração dos parâmetros de saída para simulações do tipo AC, DC e Transiente —, há elevada similaridade entre os arquivos gerados contendo os resultados das simulações, geralmente no formato “.raw”. Consequentemente, o algoritmo foi estruturado de modo a realizar pequenas adequações no arquivo de entrada, conforme o simulador escolhido, mas mantendo uma rotina unificada para a leitura e interpretação dos arquivos de saída.

Na versão MATLAB, a leitura dos arquivos de resultados é realizada mediante as funções disponibilizadas por uma *toolbox* (PERROTT, 2011). Em contraste, na versão Python, todas as rotinas de leitura foram desenvolvidas integralmente do zero, com o objetivo de assegurar compatibilidade e eficiência no processamento.

Em Python, foram implementadas funções específicas de leitura para cada tipo de simulação: AC, DC e Transiente. Nos casos das simulações DC, as tensões nodais extraídas dos arquivos de saída são armazenadas em uma matriz tridimensional, cuja estrutura é definida da seguinte forma: A primeira dimensão corresponde ao índice do circuito analisado (i.e., circuito *fault-free* ou com determinada falha injetada); A segunda dimensão corresponde ao índice das variáveis no circuito (tensões ou correntes nodais); A terceira dimensão diferencia os sinais de entrada aplicados. Para simulações do tipo AC, há ainda uma quarta dimensão, a qual discrimina se o valor armazenado corresponde ao módulo ou à fase da grandeza a ser medida, resultando, portanto, em uma matriz de quatro dimensões para este tipo de análise.

Por outro lado, a análise Transiente apresenta características singulares que inviabilizam a adoção da mesma estrutura matricial utilizada nas análises AC e DC. A principal razão para essa diferenciação reside na natureza do algoritmo de integração numérica implementado nos simuladores SPICE, o qual realiza a variação automática do passo de simulação ao longo do tempo. Este mecanismo, conhecido como "*time-step control*", ajusta dinamicamente o tamanho do incremento temporal: reduzindo-o em regiões onde há rápidas variações no comportamento do circuito (e.g., transições abruptas), e ampliando-o quando o circuito se encontra em um regime estacionário.

Como consequência direta dessa variação adaptativa do passo de simulação, o número de pontos de amostragem gerados em cada simulação transiente é variável e dependente das características dinâmicas do circuito e dos sinais de entrada. Assim, mesmo que o conjunto de circuitos simulados seja topologicamente muito parecidos, o fato de serem submetidos a diferentes condições de falha faz divergir significativamente o número e a distribuição temporal dos pontos simulados. Essa variabilidade compromete a compatibilidade dos índices na estrutura matricial de armazenamento, impedindo que as tensões nodais sejam organizadas em matrizes com dimensões fixas e uniformes, como ocorre nas análises AC e DC.

Em virtude dessa limitação, optou-se por salvar as tensões e correntes nodais dos circuitos nos testes transientes como uma estrutura de dados, para então usar essa estrutura para extrair os parâmetros derivados dos resultados das análises transientes, que são relevantes para os objetivos da ferramenta de análise de falhas: *Overshoot* e o Tempo de acomodação. Esses parâmetros, ao representarem informações resumidas da dinâmica do circuito, podem ser armazenados de forma padronizada e comparável entre diferentes circuitos e condições de falha.

Por fim, uma vez processados e armazenados os resultados das simulações no ambiente Python, foi desenvolvido um algoritmo para determinação da matriz de falhas, assegurando

compatibilidade total com o formato originalmente utilizado na ferramenta em MATLAB, conforme já ilustrado na Figura 12. A partir desta matriz, são realizadas as análises de cobertura de falhas, identificação das falhas mais difíceis de serem detectadas e seleção do melhor conjunto de testes, seguindo os mesmos critérios e procedimentos descritos nas Seções 3.2.1, 3.2.2 e 3.2.3.

Vale destacar que a versão da ferramenta implementada em Python incorporou diversas funcionalidades adicionais em relação à versão original desenvolvida em MATLAB. Entre essas melhorias, inclui-se a possibilidade de considerar medições de corrente elétrica como grandezas observáveis durante a execução dos testes, o que amplia substancialmente o escopo de análise em comparação à versão anterior, que restringia as medições a variáveis de tensão. Além disso, a nova versão passou a permitir a utilização da fase de sinais AC como parâmetro de teste, ao passo que, na implementação em MATLAB, apenas a amplitude desses sinais era contemplada.

Também foram introduzidas na versão Python, rotinas específicas para avaliar combinações complementares de testes AC, envolvendo simultaneamente módulo e fase, e para explorar possíveis combinações de testes transitórios, analisando de maneira integrada as métricas de *overshoot* e de tempo de acomodação. Essas funcionalidades ampliam a capacidade da ferramenta de identificar cenários de teste mais eficientes, permitindo avaliar, de forma sistemática, se a inclusão de testes complementares resulta em benefícios quanto ao tempo total de execução do conjunto de testes.

4 METODOLOGIA EXPERIMENTAL – ESTUDOS DE CASO

Neste capítulo, serão detalhadas as metodologias utilizadas para realização dos estudos de configurações de teste feitos através da ferramenta desenvolvida. Em um primeiro momento, ainda utilizando uma versão anterior da ferramenta rodando em MATLAB, foram analisadas as configurações de teste de um filtro *Butterworth* de um estágio. Nesse estudo de caso, foram testadas configurações de teste envolvendo sinais de entrada AC e DC. Posteriormente, com a utilização da análise transiente descrita na Seção 3.2.3, investigou-se a possibilidade de incremento na cobertura de falhas por meio da execução de um teste transiente, utilizando um sinal de degrau cujos valores inicial e final correspondem a dois pontos de polarização DC já presentes no conjunto dos melhores testes obtidos anteriormente, visando à detecção de falhas não observáveis sob condições estáticas.

No segundo estudo de caso, ainda com o algoritmo rodando em MATLAB, o mesmo filtro *Butterworth* de um estágio foi simulado no contexto de testes funcionais. Desse modo, buscou-se a possibilidade de comparação entre os resultados de detecção de falhas obtidos através dos testes estruturais já determinados pela ferramenta e os resultados de testes funcionais que normalmente são utilizados para esse tipo de circuito.

No terceiro estudo de caso, a ferramenta foi aplicada para a determinação da máxima cobertura de falhas e para a identificação do conjunto ótimo de estímulos de teste no contexto de um filtro passa-baixas de dois estágios, composto por dois blocos idênticos em cascata, cada um implementado por um amplificador operacional totalmente diferencial de um estágio. Inicialmente, as configurações de teste foram avaliadas considerando o circuito completo, foi considerado que apenas os nós de saída são possíveis nós de teste acessíveis às ponteiras do testador. Além disso, analisou-se a cobertura de falhas associada à aplicação dos testes diretamente sobre cada um dos amplificadores constituintes do circuito, configurados para operar em malha aberta, com o objetivo de quantificar os potenciais ganhos em termos de qualidade do teste decorrentes da adoção de técnicas de *Design-for-Testability* que viabilizem a realização de tais testes diretamente no *chip*. Por fim, ainda no contexto de DfT, a ferramenta foi utilizada para determinação de qual nó interno do circuito resultaria no maior incremento de cobertura de falhas, o qual, pode ser interessante de ser acessado caso melhore significativamente a qualidade do teste.

O quarto estudo de caso, apresentado na Seção 4.4, foi conduzido utilizando o novo algoritmo desenvolvido em Python e descrito detalhadamente no Capítulo 3. Esse estudo teve como objetivo validar a ferramenta proposta por meio da comparação entre os resultados de detecção de falhas obtidos via simulação e aqueles provenientes da emulação física de falhas

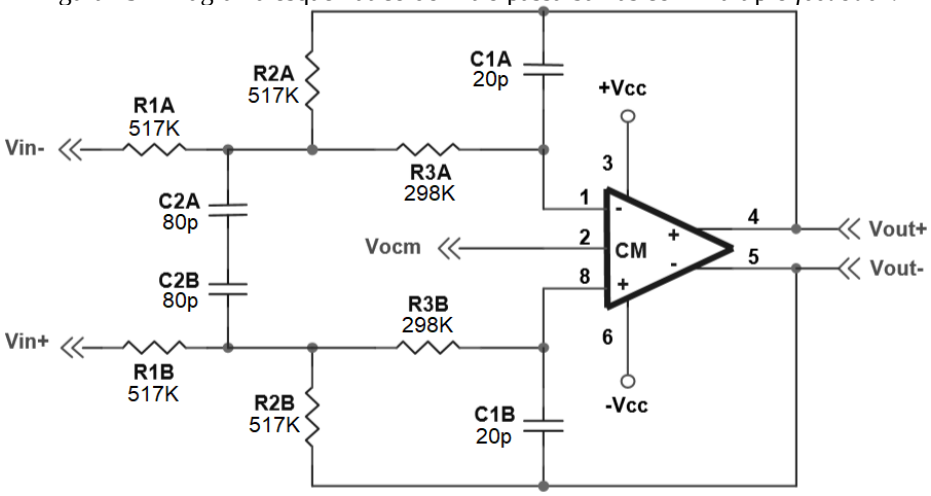
em hardware. Para isso, foram implementadas falhas reais em um circuito físico, permitindo avaliar experimentalmente a capacidade do algoritmo em reproduzir, de forma confiável, o comportamento observado no dispositivo real.

Na Seção 4.5 são apresentados os três últimos estudos de caso, cuja finalidade foi avaliar de forma abrangente todas as novas funcionalidades incorporadas à ferramenta desenvolvida em Python, bem como ampliar o escopo das análises para circuitos amplamente utilizados como referência na literatura. Para esse propósito, foram selecionados circuitos pertencentes ao grupo de *benchmarks* P2427 da IEEE, reconhecido por sua relevância em estudos de diagnóstico e detecção de falhas em sistemas analógicos. Os circuitos avaliados incluem um amplificador operacional (OPAMP1), um regulador de tensão (VREG1) e um comparador de tensão (COMP1), este último representando um exemplo de circuito com comportamento não linear. Os resultados obtidos para este e os demais estudos de caso encontram-se apresentados no Capítulo 5.

4.1 FILTRO BUTTERWORTH DE UM ESTÁGIO

O primeiro estudo de caso considerou a análise de uma topologia de filtro passa-baixas com múltipla realimentação, cuja implementação está ilustrada na Figura 19. Os valores dos componentes passivos (resistores e capacitores) foram devidamente dimensionados para a síntese de um filtro *Butterworth* passa-baixas, com frequência de corte de 10 kHz. A definição dos limites de detecção para os testes foi realizada com base na análise de variações de *corners*, garantindo a robustez das análises frente às dispersões tecnológicas do processo de fabricação.

Figura 19 - Diagrama esquemático do filtro passa baixas com múltiplo *feedback*.

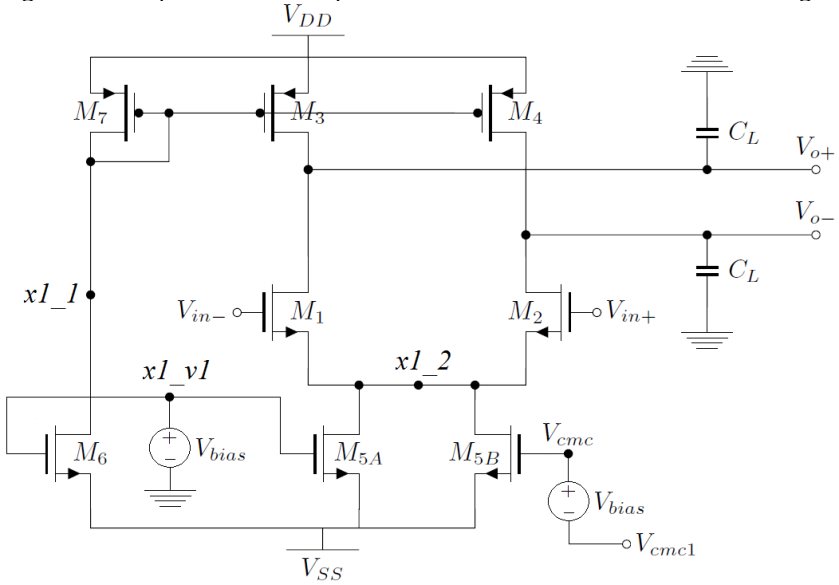


Fonte: elaborado pelo autor.

Na implementação do circuito, foi empregado um amplificador totalmente diferencial (FDA), modelado na tecnologia CMOS XFAB de 180 nm, cujo esquema elétrico é apresentado na Figura 20 (OLIVEIRA; SEVERO; GIRARDI, 2014). O núcleo amplificador é constituído pelo par diferencial de entrada formado pelos transistores M1 e M2, responsáveis pela amplificação diferencial. A carga ativa deste par é implementada pelos transistores M3 e M4, configurados em espelho de corrente com o transistor M7, de modo a garantir uma corrente de carga balanceada e estabilizada. O *biasing* do espelho de corrente e do par diferencial é realizado por meio das correntes de polarização definidas pelos transistores M6 e M5, respectivamente.

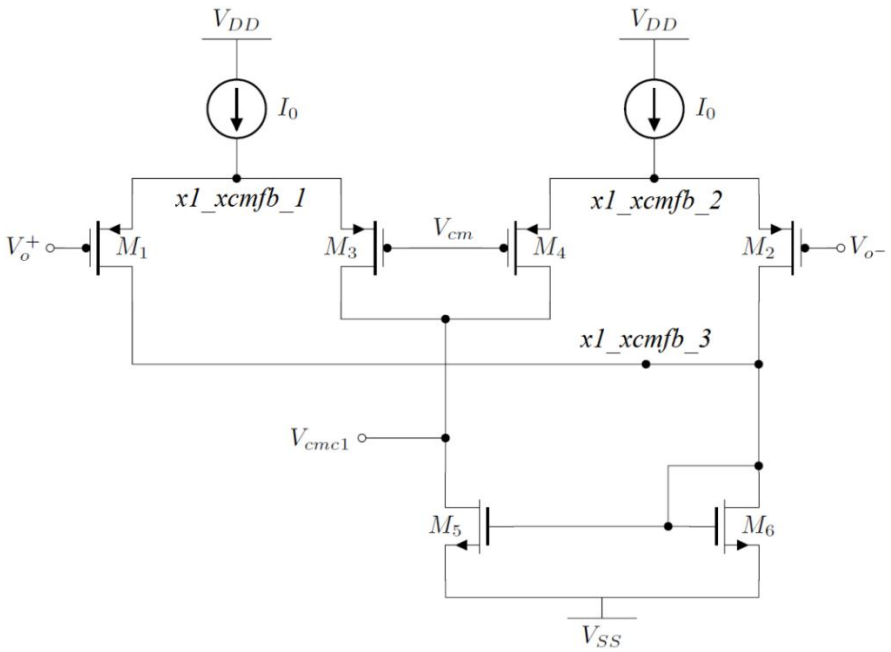
O circuito de realimentação de modo comum encontra-se descrito na Figura 21, onde os transistores M1 e M2 desempenham a função de entrada para as saídas positiva e negativa do amplificador. A detecção e correção do nível de tensão de modo comum são realizadas pelos transistores M3 e M4, que efetuam a comparação entre o nível de modo comum das saídas e a tensão de referência desejada (Vcm). A estabilização da corrente no circuito CMFB é assegurada pela atuação dos transistores M5 e M6, conectados em configuração de espelho de corrente, juntamente com as fontes de corrente que definem a polarização adequada do sistema (BENDER, 2015).

Figura 20 - Esquemático do amplificador totalmente diferencial de um estágio.



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

Figura 21 - Esquemático elétrico do circuito CMFB.



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

Na descrição em SPICE, tanto o amplificador totalmente diferencial (FDA) representado na Figura 20 quanto o circuito de realimentação de modo comum (CMFB), ilustrado na Figura 21, foram modelados como subcircuitos. A estrutura hierárquica da descrição foi organizada de forma que o bloco CMFB fosse instanciado diretamente no interior do subcircuito correspondente ao FDA, sob a designação "xcmfb". Assim, para a descrição completa do circuito, bastou instanciar o subcircuito do FDA, denominado "x1", interligando as tensões de alimentação, a referência de modo comum desejada (Vcm) e os terminais da entrada diferencial.

Na ferramenta utilizada (*toolbox*), os nós internos de um subcircuito são renomeados automaticamente com um prefixo que corresponde ao identificador do subcircuito na instância, seguido por um caractere de sublinhado "_". Assim, conforme ilustrado na Figura 20, um nó denominado “1” na descrição original SPICE do subcircuito "x1" passa a ser referenciado na toolbox como “x1_1”. No caso de subcircuitos hierárquicos, ou seja, quando há um subcircuito encapsulado dentro de outro, a mesma convenção é aplicada de forma recursiva. Por exemplo, conforme ilustrado na Figura 21, o nó “1” do subcircuito “xcmfb”, que está instanciado no interior do subcircuito “x1”, será referenciado pela *toolbox* como “x1_xcmfb_1”.

Na Tabela 1 são apresentados os parâmetros de dimensionamento dos componentes do circuito do FDA. Ressalta-se que, para fins de simplificação e foco na análise de falhas, a tensão

de polarização (Vbias) e a corrente de polarização (I0) foram consideradas ideais, assumindo valores constantes e sem variação no contexto das simulações realizadas.

Tabela 1 - Especificações do projeto do circuito FDA.

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

Fonte: (CHINAZZO, 2016) adaptado de Oliveira, Severo e Girardi (2014).

O modelo de falhas adotado neste estudo de caso abrangeu seis falhas catastróficas por transistor, sendo estas: circuito aberto nos terminais de gate, dreno e fonte, além de curtos-circuitos entre as combinações *gate-drain*, *gate-source* e *source-drain*. A simulação de falhas de circuito aberto foi realizada mediante a inserção de um resistor de alta impedância (10 MΩ) em série com o terminal correspondente, enquanto que, para as falhas de curto-circuito, foi adicionado um resistor de baixa impedância (10 Ω) conectado em paralelo entre os terminais afetados (OLIVEIRA; SEVERO; GIRARDI, 2014). No bloco de ganho do amplificador, foram excluídas da análise as falhas de curto-circuito do tipo *gate-source* nos transistores M5A e M6, bem como *gate-drain* no transistor M7, visto que tais falhas implicariam em curtos diretos sobre fontes de tensão ideais ou conexões já existentes na estrutura do circuito, inviabilizando a simulação adequada. O mesmo critério foi aplicado ao transistor M6 no circuito de realimentação de modo comum (CMFB). Assim, o modelo final contemplou um total de 45 falhas catastróficas no bloco de ganho e 35 falhas catastróficas no bloco CMFB.

No que tange à modelagem de falhas paramétricas, foram consideradas variações no comprimento do canal (L) e na tensão de *threshold* (Vth) de cada transistor. As perturbações no comprimento de canal adotaram uma variação de ±25% (BROSA; FIGUEIRAS, 2000). As variações de tensão de *threshold* foram fixadas em ±50 mV (BORKAR et al. (2003).

O desvio padrão da tensão de *threshold* para a tecnologia CMOS de 180 nm é da ordem de 10 mV, o que implica uma dispersão de aproximadamente ±30 mV em três desvios padrões. Complementarmente, as falhas paramétricas devem ser modeladas dentro do intervalo de três a seis desvios padrões, ou seja, entre ±30 mV e ±60 mV (DENG; SHI; ZHANG, 2012) o que valida a escolha adotada neste estudo.

As variações no comprimento do canal foram de ±25% (BENDER, 2015), com exceções aplicadas aos transistores M1 e M2 do bloco de ganho, para os quais a variação de -25% no comprimento do canal foi suprimida, a fim de evitar a violação do limite inferior estabelecido pela tecnologia utilizada. Cabe salientar que essas alterações paramétricas não visam representar diretamente as variações típicas de processo, mas sim modelar defeitos localizados que impactam características elétricas críticas, como a transcondutância (gm) e a resistência do canal, permitindo uma avaliação mais abrangente do comportamento do circuito sob condições de degradação ou falhas específicas.

Adicionalmente, o modelo de falhas incorporou falhas nos componentes passivos do circuito. Para além das falhas catastróficas, foram introduzidas variações paramétricas de ±25% nos valores nominais das resistências e capacitâncias, possibilitando a análise do impacto de desvios extremos nestes elementos. Como resultado, o modelo global de falhas compreendeu um total de 80 falhas catastróficas e 40 falhas paramétricas associadas ao FDA, bem como 20 falhas catastróficas e 20 falhas paramétricas nos componentes passivos. Assim, o conjunto completo de falhas consideradas neste estudo perfaz um total de 160 falhas, englobando tanto falhas destrutivas quanto variações paramétricas críticas para a avaliação da robustez do circuito.

4.1.1 Testes AC e DC

O filtro passa-baixas com topologia de múltipla realimentação foi submetido a simulações em regime permanente, visando à avaliação de sua resposta frente a estímulos de natureza DC e AC. As análises DC foram conduzidas considerando a faixa de tensões compatível com a especificação de alimentação do circuito, variando linearmente de -0,9 V a +0,9 V, com incrementos de 1 mV, de forma a garantir uma adequada discretização do espaço de entrada. As análises AC foram realizadas com excitação senoidal de amplitude suficientemente pequena para preservar a linearidade, cobrindo a faixa de frequência de 10 Hz a 2 MHz. Tal faixa foi definida de modo a garantir a caracterização completa do comportamento em frequência do circuito, abrangendo mais de duas décadas acima da frequência de corte do filtro, especificada em 10 kHz. Além disso, de forma a garantir o realismo nas condições de

detecção, foi utilizado um limite mínimo de diferença de tensão detectável pelo testador em 0,5 mV.

4.1.2 Teste Transiente Adicional

Considerando que o conjunto final de testes pode envolver múltiplas configurações DC, propôs-se a execução de uma análise transiente adicional, explorando a resposta do circuito a uma excitação do tipo degrau, entre dois valores DC previamente selecionados dentre aqueles pertencentes ao conjunto ótimo de testes definido na Seção 4.1.1. Tal abordagem, fundamentada na metodologia descrita na Seção 3.2.3, visa a otimização do conjunto de testes por meio da inserção de estímulos dinâmicos que não comprometam significativamente o tempo global de execução nem os custos associados ao teste. A análise busca identificar incrementos na cobertura de falhas a partir da exploração das respostas transitórias associadas às configurações DC já previstas, evitando assim a introdução de estímulos redundantes.

O modelo de falhas utilizado nesta etapa permaneceu inalterado em relação à Seção 4.1, e os limites de aceitação continuaram baseados nas respostas obtidas nos quatro *corners* da tecnologia, garantindo assim a consistência na avaliação dos desvios de comportamento. Os valores de tensão inicial e final do degrau aplicado na análise transiente foram definidos a posteriori, a partir das configurações DC que compõem o conjunto de testes mais eficiente, conforme determinado pelas análises realizadas na seção anterior.

4.2 DETECÇÃO DE FALHAS ATRAVÉS DE TESTES FUNCIONAIS

Na Seção 2.1.2, os testes funcionais são comparados com os testes estruturais quanto a qualidade e preço. Sendo assim, complementarmente às análises estruturais feitas na Seção 4.1, foram realizados testes funcionais com o objetivo de detectar falhas através da verificação de parâmetros críticos de desempenho do filtro: ganho DC, *Slew Rate* e frequência de corte (-3 dB). Para garantir a comparabilidade entre os testes funcionais e estruturais, os limites de aceitação para cada parâmetro funcional foram determinados com base nas respostas obtidas nos quatro *corners*, replicando a abordagem metodológica previamente adotada.

Assim, uma falha funcional foi considerada detectada sempre que um parâmetro avaliado apresentasse desvio além do limite máximo ou mínimo estabelecido para o respectivo *corner*, caracterizando degradação ou comprometimento da funcionalidade do circuito. Para viabilizar a execução dos testes funcionais, a ferramenta utilizada foi minimamente adaptada, de acordo com a Seção 3.3, assegurando a manutenção da integridade do núcleo computacional

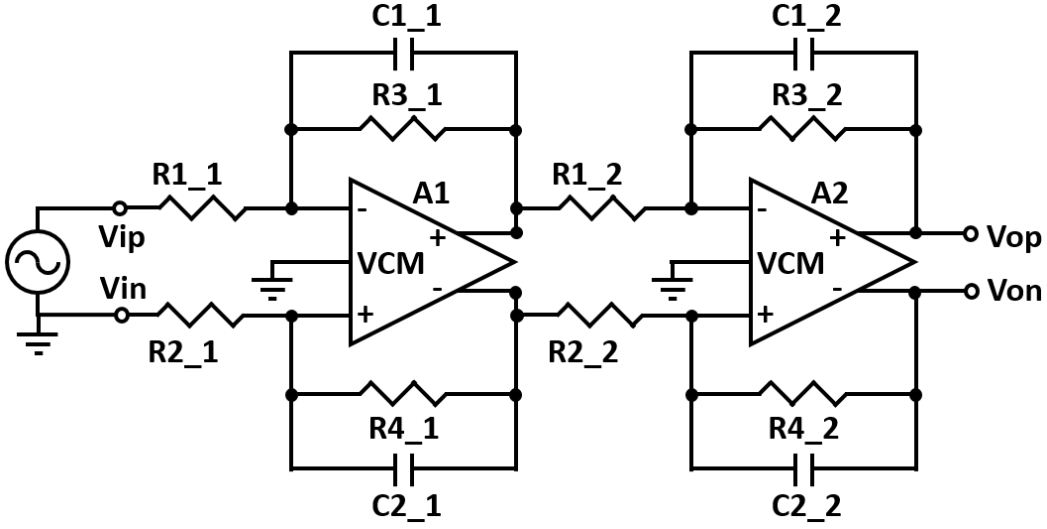
e dos algoritmos originais, enquanto se implementaram ajustes específicos para a extração automatizada dos parâmetros funcionais requeridos nesta etapa.

4.3 FILTRO PASSA-BAIXAS DE DOIS ESTÁGIOS

O segundo estudo de caso foi definido com o objetivo de avaliar a capacidade da ferramenta na análise de estratégias baseadas em técnicas de *Design for Testability*, visando à maximização da cobertura de falhas do conjunto final de testes. Para tal, foi selecionado um filtro passa-baixas composto por dois estágios idênticos conectados em cascata, cada qual implementado com ganho unitário na banda de passagem e frequência de corte fixada em 10 kHz, garantindo linearidade e estabilidade no espectro de interesse. O diagrama esquemático do referido filtro encontra-se representado na Figura 22.

A topologia do circuito adota elementos passivos com valores padronizados: todos os resistores possuem resistência nominal de 570 kΩ, enquanto todos os capacitores apresentam capacitância de 28 pF, assegurando a simetria e a uniformidade entre os dois estágios. Cada um dos amplificadores diferenciais utilizados na implementação do filtro corresponde ao modelo desenvolvido por Oliveira, Severo e Girardi (2014), projetado em tecnologia CMOS XFAB 180 nm, cujo detalhamento esquemático e estrutural foi previamente apresentado na Seção 4.1.

Figura 22 - Diagrama esquemático do filtro passa-baixas de dois estágios.



Fonte: elaborado pelo autor.

A modelagem hierárquica típica na linguagem SPICE permite que um subcircuito seja declarado uma única vez e posteriormente instanciado múltiplas vezes no circuito, promovendo modularidade e reutilização. No caso específico do filtro ilustrado na Figura 22, composto por dois FDAs idênticos, esta abordagem possibilita a declaração única do subcircuito

correspondente ao FDA, que pode então ser instanciado duas vezes, cada uma com identificadores distintos, correspondentes ao primeiro e segundo estágios. No entanto, essa prática introduz uma limitação para a inserção seletiva de falhas: caso ambos os FDAs sejam instâncias de um mesmo subcircuito, qualquer modificação estrutural ou inserção de falha no subcircuito impactará simultaneamente todas as instâncias, impedindo a individualização das falhas.

A solução para esse impasse consiste na duplicação do subcircuito do FDA na descrição SPICE, criando duas versões idênticas em termos estruturais, mas com identificadores distintos, de modo que cada FDA seja associado a uma instância exclusiva do subcircuito. Internamente, os nomes dos componentes podem ser mantidos idênticos entre as cópias, pois a diferenciação necessária será garantida exclusivamente pelos nomes distintos atribuídos aos subcircuitos nas declarações de instância

4.3.1 Análises AC e DC do Filtro Passa-Baixas Completo

Em um primeiro momento, o filtro da Figura 22 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 ponteiras do testador.

De maneira análoga ao estudo de caso do Capítulo 4, o modelo de falhas adotado considerou as seis falhas catastróficas por transistor previamente definidas, com as mesmas metodologias de implementação via resistores. No entanto, assim como na análise anterior, foram excluídas do modelo as falhas de curto-circuito entre gate e dreno do transistor M6 no CMFB e do transistor M7 no bloco principal, visto que tais conexões já existem no projeto do circuito. Da mesma forma, as falhas de curto entre *gate* e *source* dos transistores M5A e M6 também não foram consideradas, em virtude da idealização das tensões de polarização "Vbias".

No que se refere às falhas paramétricas, a modelagem manteve as mesmas premissas estabelecidas no Capítulo 4, considerando variações de ±25% no comprimento de canal (L) e ±50 mV na tensão de *threshold* (Vth) para cada transistor.

Dada a existência de dois estágios e dois FDAs neste circuito, o modelo de falhas totalizou 240 falhas nos amplificadores (120 falhas em cada um). Além disso, o modelo de falhas também contempla os componentes passivos do circuito, com a inclusão de falhas catastróficas e falhas paramétricas (variações de ±25%) em cada resistor e capacitor presentes no filtro de dois estágios. Assim, o modelo completo de falhas para este estudo totalizou 288 falhas.

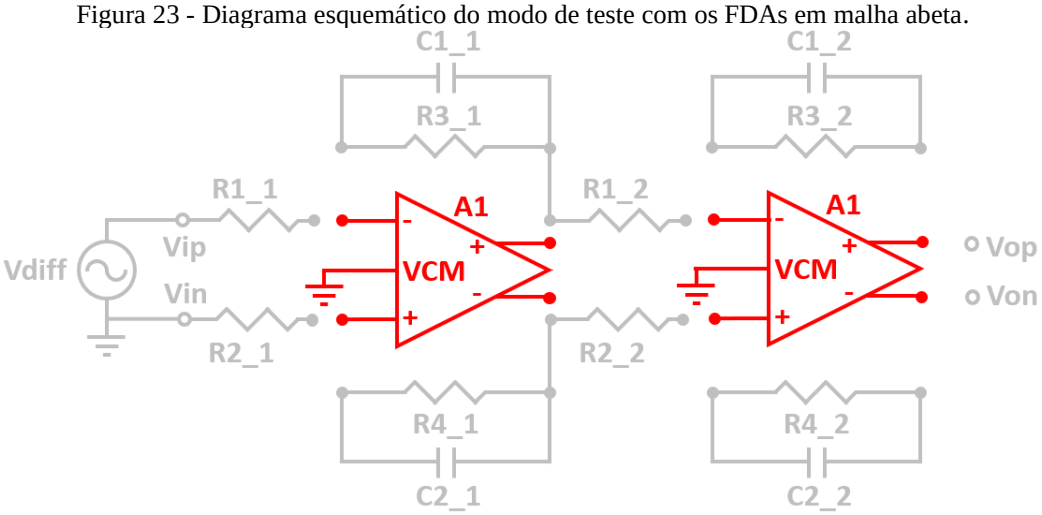
Os limites de tensão para a discriminação entre falhas detectáveis e não-detectáveis foram determinados com base, novamente, na análise dos *corners* da tecnologia. As estratégias de teste aplicadas a este filtro também se mantiveram alinhadas às estabelecidas anteriormente, contemplando as análises nos domínios DC (varreduras de tensão na faixa de ±0,9 V, com incrementos de 1 mV) e AC (sinais variando de 10 Hz a 1 MHz, com 100 pontos por década).

De forma a garantir o realismo nas condições de detecção, foi mantido o limite mínimo de diferença de tensão detectável pelo testador em 0,5 mV, conforme critério previamente estabelecido na Seção 4.1. Os resultados são mostrados na Seção 5.3.1.

4.3.2 Amplificador Totalmente Diferencial em Malha Aberta

Como alternativa de abordagem, foram analisadas as configurações de teste aplicadas ao amplificador totalmente diferencial (FDA) (OLIVEIRA; SEVERO; GIRARDI, 2014), que compõe o filtro passa-baixas de dois estágios descrito na Seção 4.1. Entretanto, nesta etapa, a análise foi realizada com o circuito em malha aberta, conforme ilustrado no esquemático da Figura 23. O objetivo dessa análise é avaliar a viabilidade da implementação de técnicas de DfT, visando à inserção de pontos de acesso que permitam a aplicação de estímulos e a observação de respostas diretamente nos nós internos do circuito, possibilitando o teste individualizado de cada um dos amplificadores que compõem o filtro.

Figura 23 - Diagrama esquemático do modo de teste com os FDAs em malha abeta.



Fonte: elaborado pelo autor.

Deve-se salientar que, embora o ganho do FDA em malha aberta seja finito, ele apresenta valor elevado. Assim, na realização do teste AC, torna-se imprescindível a utilização

de sinais de entrada com amplitudes reduzidas, a fim de evitar a saturação do sinal de saída e garantir a linearidade das respostas observadas.

Através da análise do circuito em malha aberta, objetiva-se quantificar o incremento potencial na cobertura de falhas decorrente da possibilidade de testar os amplificadores isoladamente, além da abordagem convencional de teste do filtro como um sistema integrado. Considerando-se que, no modelo de falhas adotado — o mesmo da Seção 4.3.1 —, mais de 80% das 288 falhas simuladas estão associadas aos componentes internos dos amplificadores, esta análise reforça a importância crítica desses blocos na definição de estratégias eficientes de teste.

Para garantir a consistência metodológica e permitir a comparação direta entre os resultados obtidos, o modelo de falhas aplicado a cada amplificador em malha aberta manteve-se idêntico ao descrito na Seção 4.3.1, incluindo as mesmas variações paramétricas e falhas catastróficas. De forma análoga, os sinais de estímulo considerados para os testes seguiram as mesmas especificações técnicas previamente definidas para o FDA

4.3.3 Utilização Adicional de um Nó Interno Do Amplificador Como Nó de Teste

Ao término das análises apresentadas na Seção 4.3.2, foi realizada uma nova avaliação, viabilizada pelas funcionalidades de reanálise da ferramenta. Nesta etapa, considerou-se a possibilidade de inserção de uma estrutura de DfT com o objetivo de monitorar um único nó interno do circuito, visando ao incremento da cobertura de falhas total.

Para a seleção do nó a ser monitorado, foram avaliadas exaustivamente todas as possíveis alternativas de pontos de observação interna. O critério adotado para escolha foi a capacidade do nó em questão de viabilizar a detecção do maior número possível de falhas não detectadas pelas estratégias de teste previamente aplicadas. Dessa forma, busca-se maximizar a cobertura de falhas sob a restrição de se monitorar apenas um único nó interno do circuito, otimizando a eficácia do método frente às limitações práticas de acesso e instrumentação

4.4 VALIDAÇÃO DA FERRAMENTA ATRAVÉS DE EMULAÇÃO DE FALHAS POR HARDWARE

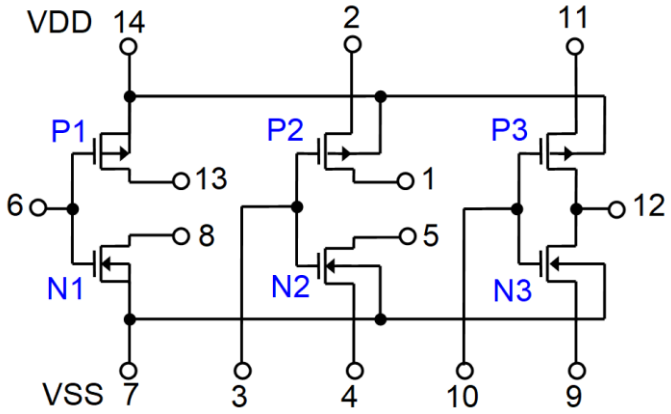
Em uma metodologia diferente, esse estudo de caso visou a validação da ferramenta através de emulação de falhas em um circuito real. Sendo assim, diferentemente dos estudos anteriores, baseados exclusivamente em dados provenientes de simulações SPICE, este estudo de caso integra duas frentes complementares: (i) simulações estruturais para determinar o conjunto ótimo de testes (Seção 4.4.1) e (ii) emulação de falhas catastróficas em bancada (Seção

4.4.2), permitindo avaliar, na prática, a correspondência entre os resultados previstos e os observados experimentalmente.

4.4.1 Simulação de Configurações de Teste

O circuito adotado para a validação experimental foi o CD4007, um circuito integrado CMOS composto por três transistores NMOSFET e três PMOSFET de enriquecimento, dispostos em dois pares complementares e um inversor, assim como pode ser visto na Figura 24. Sua principal vantagem, para fins de experimentação, é a acessibilidade individual aos terminais dos transistores, o que permite a construção de topologias analógicas diversas e viabiliza a injeção direta de falhas.

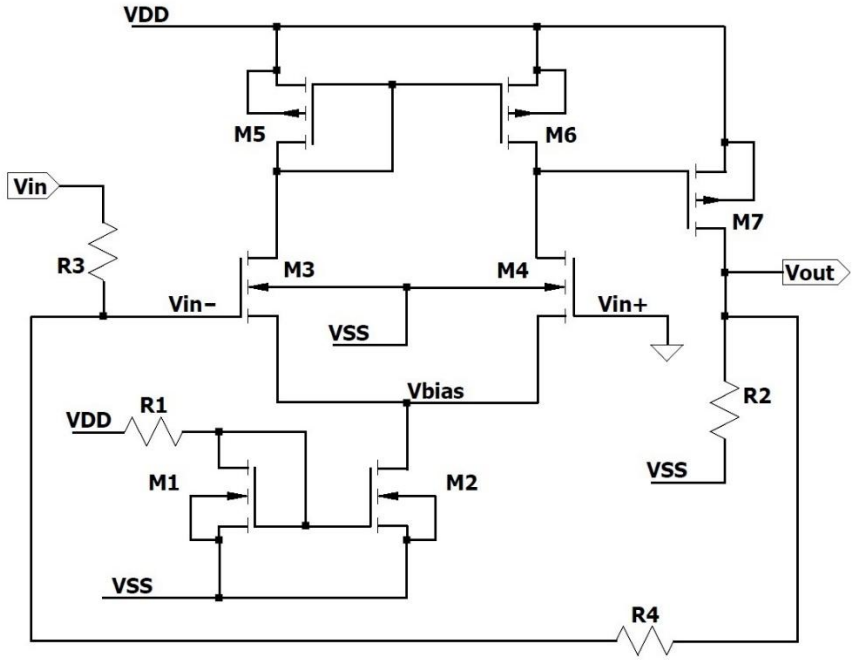
Figura 24 – Circuito interno do CD4007.



Fonte: Arabi (1997).

A partir de três unidades do CD4007 previamente selecionadas, implementou-se em *protoboard* uma versão simplificada de um amplificador operacional de dois estágios que pode ser visto na Figura 25. O circuito foi configurado como inversor realimentado externamente pelos resistores R3 e R4, resultando em ganho de –5 V/V. Os resistores R2 e R1 funcionam, respectivamente, como carga e gerador de corrente de referência para o espelho de corrente (composto pelos transistores M1 e M2), responsável pela polarização do amplificador operacional. Os valores dos resistores empregados são: R1 = 22 kΩ, R2 = 22 kΩ, R3 = 10 kΩ e R4 = 50 kΩ. Os parâmetros de dimensionamento dos transistores, utilizados nas simulações SPICE, estão apresentados na Tabela 2 (FULLER, 2020).

Figura 25 – Esquemático do amplificador operacional implementado.



Fonte: Trabalho do aluno de mestrado Luis Duarte que está sendo encaminhado para publicação.

Tabela 2 - Parâmetros de dimensionamento dos transistores.

Parâmetro SPICE	NMOSFET	PMOSFET
Vto	1,4V	-1,4V
W	170µm	360µm
L	12µm	12µm
tox	68nm	68nm
Cgso	340pF	450pF
Cgdo	340pF	450pF
Cgbo	575pF	575pF
NRD	0,1	0,1
NRS	0,1	0,1

Fonte: Adaptado de Fuller (2020).

As simulações estruturais foram executadas com a versão Python da ferramenta (Seções 3.1 e 3.2), utilizando o NGSPICE como simulador. O modelo de falhas considerou apenas falhas catastróficas — curtos-circuitos e circuitos abertos — totalizando 48 falhas após exclusões topológicas. Os circuitos abertos foram modelados com resistores de 10 MΩ em série, e os curtos-circuitos, com resistores de 10 Ω em paralelo.

Como o modelo SPICE adotado não contemplava variabilidade de processo, adotou-se um limite de detecção fixo de ±10% em torno da tensão nominal simulada no nó de teste. Tal escolha é conservadora, já que para processos comerciais reais típicos os limites ficam em ±5%, mas garante robustez e reduz a probabilidade de falsos alarmes na comparação entre simulação e bancada.

.

4.4.2 Medição Experimental

Uma vez a ferramenta tendo rodado todas as simulações e determinado o melhor conjunto de testes, a ideia foi testar o protótipo do circuito nas mesmas condições de teste especificadas pela ferramenta de modo a comparar a detecção de falhas esperada via simulação com a medição prática dos testes. Sendo assim, nos testes do protótipo, foram utilizados os mesmos estímulos de entrada e pontos de teste definidos pela ferramenta, foram injetadas fisicamente as mesmas falhas do modelo simulado, realizando a medição das tensões ou sinais obtidos, determinação se a falha é detectada ou não e comparação com as simulações.

Após cada injeção, o nó de teste é medido e o resultado correspondente é comparado com o resultado simulado pela ferramenta. Como essas etapas foram repetidas 48 vezes (uma para cada falha injetada), antes e após cada teste o circuito de *hardware* foi restaurado para o seu design original (*fault free*) e verificado quanto ao seu correto funcionamento. Essa verificação foi realizada conferindo-se o ganho em malha fechada do CUT, com Vin = 0.8 Vpp @10 kHz, conforme exemplificado no procedimento experimental.

O protótipo completo da bancada de testes foi montado com as conexões entre os transistores internos realizadas por meio de fios *jumper*. A alimentação foi fornecida por uma fonte de tensão dual com +6 VDC e -6 VDC. Para verificar a operação adequada do circuito, utilizou-se um osciloscópio digital. Além disso, um gerador de referência DC de precisão, desenvolvido internamente, foi utilizado para fornecer a tensão de entrada de teste ao circuito amplificador operacional.

Os instrumentos utilizados na bancada de testes incluem: gerador de funções Tektronix AFG 2021, osciloscópio digital de armazenamento de dois canais Tektronix TDS201B, fonte de alimentação dual Tektronix PS 503A, gerador de sinal DC de referência (alimentado por bateria), multímetro digital Fluke 16B e protoboard contendo o protótipo de hardware do amplificador operacional.

Para verificar a funcionalidade do amplificador operacional, foram medidas algumas especificações típicas. A tensão de *offset* medida foi de 35 mV, enquanto o ganho em malha aberta foi de 50 dB, estando em conformidade com as simulações realizadas.

4.5 GRUPO P2427 E BENCHMARK DE CIRCUITOS ANALÓGICOS

O grupo de trabalho IEEE P2427 da IEEE Standards Association — oficialmente intitulado "*Working Group on Analogue and Mixed-Signal Benchmark Circuits"* — foi criado com o objetivo de disponibilizar publicamente circuitos analógicos, mistos (*analogue/mixed-signal*) e de sinal digital, a fim de constituir um conjunto padronizado de *benchmarks* para circuitos que permitam comparações de metodologias, ferramentas e fluxos de verificação e teste (IEEE STANDARDS ASSOCIATION, 2023a).

Dentro dessa iniciativa, são oferecidos circuitos-modelo representativos de blocos analógicos e de sinal misto amplamente utilizados em sistemas integrados, como amplificadores operacionais, reguladores de tensão, referências de banda-gap e *phase-locked loops* (PLLs). Esses circuitos foram originalmente desenvolvidos por entidades da indústria, como a AMS AG e a Mentor Graphics (Siemens EDA), e posteriormente tornaram-se públicos sob a coordenação do grupo (IEEE STANDARDS ASSOCIATION, 2023a).

A página específica do projeto intitulada "*Analogue Benchmark Circuits*" disponibiliza os *netlists* SPICE e *testbenches* correspondentes a diversos blocos, incluindo OPAMP1, VREG1, BANDGAP1 e PLL1 (IEEE STANDARDS ASSOCIATION, 2023b). O objetivo é fornecer uma base comum para avaliação de metodologias de teste, análise de falhas e automação de simulações em circuitos analógicos e mistos. A padronização desses benchmarks facilita a reprodutibilidade dos resultados e promove comparações consistentes entre diferentes abordagens de pesquisa. O padrão também descreve listas formais de falhas potenciais, a fim de permitir a comparação entre diferentes metodologias de simulação de falhas, técnicas de projeto para testabilidade (*Design for Testability*) e estratégias de geração de testes para circuitos analógicos e mistos (SUNTER; SARSON, 2017).

Junto dos arquivos disponibilizados pelo grupo P2427, estão todos os parâmetros necessários para simulação dos componentes no modelo BSIM3v3 em condições típicas e em cada um dos 4 *corners*. Entretanto, os modelos de transistores disponibilizados originalmente pela IBM no conjunto de *benchmarks* utilizam a sintaxe adotada por simuladores proprietários como o HSPICE e o Spectre, onde o modelo físico BSIM3v3 é declarado com o parâmetro LEVEL=49. No entanto, no ambiente NGSPICE (simulador utilizado nesse trabalho) o modelo equivalente ao BSIM3v3 é identificado como LEVEL=8. Dessa forma, a principal modificação necessária para compatibilizar os arquivos foi a substituição de todas as declarações LEVEL=49 por LEVEL=8, garantindo que o simulador reconheça corretamente o tipo de modelo físico empregado. Além dessa mudança, alguns parâmetros específicos do formato HSPICE/Spectre não são reconhecidos pelo NGSPICE. Esses parâmetros — tais como

VERSION, NOIMOD, ACM, PRT, PVAG e outros campos de controle de saída — foram removidos, uma vez que não influenciam diretamente no comportamento elétrico do modelo, mas apenas em aspectos de compatibilidade e depuração nos simuladores originais.

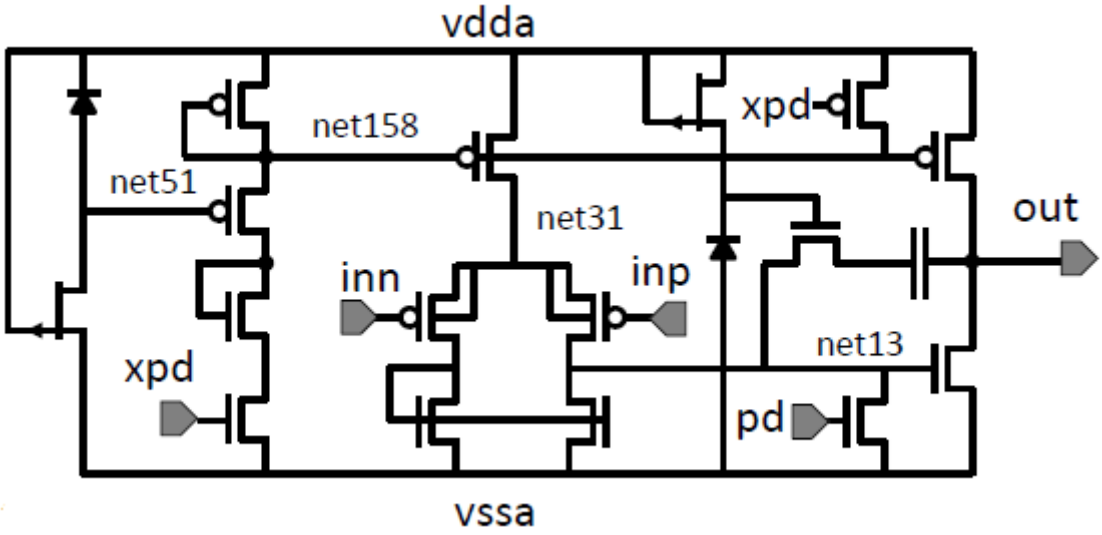
Os modelos de dispositivos passivos (resistores, capacitores e diodos) não necessitam de alterações estruturais significativas, sendo apenas ajustados para garantir compatibilidade sintática, mantendo-se o mesmo conjunto de parâmetros físicos originais.

Neste trabalho de doutorado, cujo foco é a determinação automatizada de configurações de teste otimizadas para circuitos integrados analógicos via simulações SPICE, foram escolhidos três circuitos *benchmark* do grupo IEEE P2427s para aplicar na metodologia proposta: o OPAMP1, o VREG1 e o COMP1 (IEEE STANDARDS ASSOCIATION, 2023b). As suas descrições em maiores detalhes seguem nas respectivas Seções 4.5.1, 4.5.2 e 4.5.3.

4.5.1 OPAMP1

O circuito OPAMP1 disponibilizado pelo conjunto de *benchmarks* do P2427 é mostrado na Figura 26, o arquivo disponibilizado vem acompanhado de *netlist* SPICE e *testbench* revisado (IEEE STANDARDS ASSOCIATION, 2023b). O OPAMP1 corresponde a um amplificador operacional CMOS de uso geral, projetado para servir como circuito de referência em experimentos de diagnóstico e detecção de falhas. Sua topologia segue o padrão clássico de amplificadores operacionais de precisão, composta por três estágios principais: (i) um par diferencial de entrada que realiza a amplificação inicial da diferença de tensão entre os terminais de entrada; (ii) um estágio de ganho intermediário, que amplia o sinal de erro e define o ganho em malha aberta; e (iii) um estágio de saída do tipo *push-pull*, responsável por fornecer corrente de carga ao nó de saída, com baixa impedância. O circuito opera entre as tensões de alimentação VDDA (positiva) e VSSA (referência), podendo incluir pinos auxiliares de controle de polarização e desligamento (*power-down*).

Figura 26 – OPAMP1 amplificador operacional básico disponibilizado pelo conjunto de benchmarks do P2427.



Fonte: IEEE Standards Association P2427.

O modelo de falhas adotado para o OPAMP1 segue a metodologia definida pelo grupo IEEE P2427, a qual associa a cada elemento do circuito uma lista de defeitos potenciais (*potential defect list*). Cada componente — resistor, capacitor, diodo ou transistor — é descrito por duas possíveis condições de falha, denominadas preLRL (*Lower Relative Likelihood*) e preHRL (*Higher Relative Likelihood*).

A primeira, preLRL, representa falhas do tipo aberto ou *stuck-off*, nas quais ocorre perda de condução elétrica, podendo ser causadas por descontinuidades metálicas, oxidação de contatos ou degradação de interfaces semicondutoras.

A segunda, preHRL, corresponde a falhas do tipo curto ou *stuck-on*, que simulam condições de condução excessiva, tais como curtos entre junções, perfuração de óxido de *gate* ou fugas dielétricas entre nós adjacentes.

Os valores associados a preLRL e preHRL não devem ser interpretados como parâmetros elétricos (por exemplo, resistências ou capacitâncias), mas sim como índices de probabilidade relativa de ocorrência do defeito, geralmente proporcionais à área física do componente no *layout* (SUNTER; SARSON, 2017). Dessa forma, o modelo permite que diferentes ferramentas e metodologias de injeção de falhas adotem parametrizações próprias (por exemplo, resistência de 10 Ω para curto e 10 MΩ para circuito aberto), sem perder compatibilidade com a definição estrutural do *benchmark*.

No contexto experimental desta tese, o circuito OPAMP1 foi utilizado como caso de estudo para avaliação da metodologia desenvolvida de seleção automática de sinais de teste e identificação de nós observáveis. Esse circuito foi escolhido por apresentar características típicas de blocos analógicos reais, como comportamento não linear, presença de capacitores de

compensação e transistores operando em diferentes regiões de condução, o que o torna adequado para a análise da cobertura de falhas e da eficiência dos estímulos de teste.

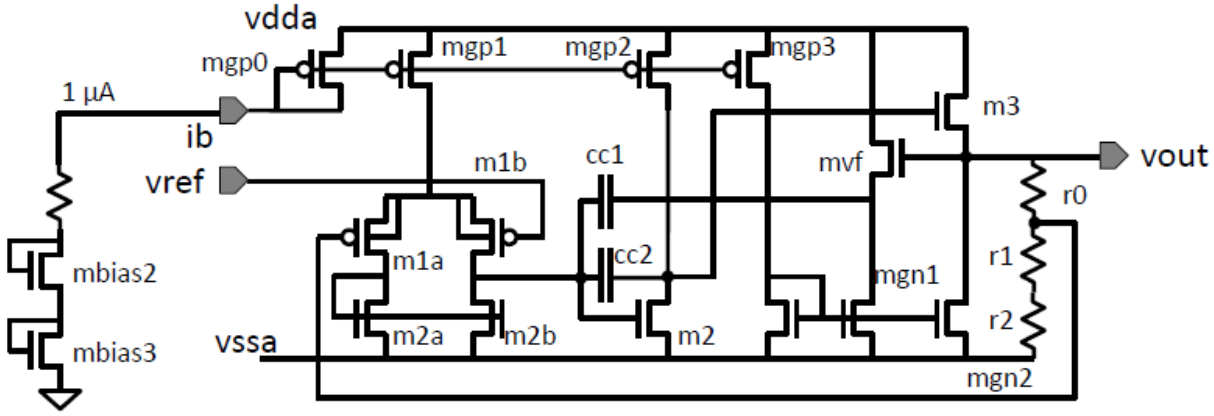
Na lista de defeitos potenciais disponibilizada pelo grupo P2427, são descritos dois defeitos por componente, com exceção dos dois resistores de difusão que não tiveram defeitos mencionados. Sendo assim, a lista é composta por 36 defeitos. Rodando o circuito na janela "AnalogCircuitTest", descrita na Seção 3.1, o algoritmo de determinação automática gerou um modelo de falhas contendo 89 falhas catastróficas, duas (uma *open* e outra *short*) em cada componente de dois terminais (resistores, capacitores e diodos) e 6 falhas em cada transistor (3 opens e 3 shorts) com exceção de 3 transistores que já possuíam ligações diretas entre dois de seus terminais na própria topologia do circuito, o que implica na não adição dessa falha de curto-circuito no modelo. Esse modelo com 89 falhas catastróficas foi utilizado na simulação de falhas, não foram consideradas falhas paramétricas, uma vez que a lista de defeitos disponibilizada pelo grupo P2427 trata de falhas catastróficas.

Para simulação, o amplificador foi colocado em malha fechada, com a ligação direta entre a sua saída (out) e a entrada negativa (inn). A tensão de alimentação utilizada foi de 3,3V para vdda e 0v para vssa (Figura 26). Os pinos xpd e pd foram conectados a tensões de 3,3V e 0V respectivamente para ativar o funcionamento do amplificador. A entrada positiva (inp) foi simulada para tensões DC de 0V até 3,3V com passos de 1mV e também foi simulada para sinais AC com 1V de amplitude e frequências de 100Hz até 10MHz, uma vez que o GBW (*Gain-Bandwidth product*) do amplificador é de 2MHZ (SUNTER; SARSON, 2017). Além disso, foi considerado um capacitor de carga ligado ao pino de saída (out) de 50fF, valor já presente no arquivo disponibilizado pelo grupo P2427. Os limites de detecção foram determinados por *Corners* e as precisões de tensão, corrente e tempo foram consideradas como sendo respectivamente: 100uV, 1nA e 0,1ns.

4.5.2 VREG1

O circuito VREG1, também disponibilizado pelo conjunto de benchmarks do grupo IEEE P2427, é apresentado na Figura 27. O arquivo fornecido pelo grupo inclui o *netlist* SPICE completo, acompanhado de *testbench* revisado, com parâmetros de simulação e condições de operação definidas (IEEE STANDARDS ASSOCIATION, 2023b). Esse circuito corresponde a um regulador de tensão CMOS de uso geral, projetado para atuar como referência padronizada em experimentos de simulação de falhas, geração automática de testes e avaliação de técnicas de diagnóstico em circuitos analógicos.

Figura 27 – VREG1 regulador de tensão disponibilizado pelo conjunto de benchmarks do P2427.



Fonte: IEEE Standards Association P2427.

O VREG1 representa uma implementação típica de regulador de tensão linear com realimentação, amplamente utilizado em circuitos integrados analógicos e mistos para o controle e estabilização de níveis de tensão internos. O circuito é composto por um amplificador de erro, uma referência de tensão, uma rede de realimentação resistiva e um transistor de passagem que atua como elemento de controle de corrente. A arquitetura segue o princípio clássico dos reguladores lineares CMOS: o amplificador de erro compara a tensão de saída (VOUT) com uma tensão de referência (VREF) e ajusta o transistor de passagem para manter a tensão de saída constante, mesmo sob variações de carga ou de tensão de alimentação (VDDA). O nó ib corresponde ao ponto de injeção de corrente de polarização, enquanto VSSA representa o terminal de referência (terra analógico).

O regulador de tensão apresenta características adicionais de realimentação e polarização contínua, o que o torna particularmente útil para a análise de falhas em circuitos com comportamento estático, além de permitir avaliar a sensibilidade de diferentes nós às variações provocadas por defeitos abertos e curtos. Dessa forma, o VREG1 contribui para a validação experimental do método desenvolvido, fornecendo um cenário de teste representativo de blocos analógicos presentes em sistemas integrados reais (IEEE Standards Association, 2023b).

Assim como no caso do amplificador operacional OPAMP1, o modelo de falhas adotado para o VREG1 segue a metodologia padronizada pelo grupo IEEE P2427, na qual cada componente do circuito é associado a uma lista de defeitos potenciais (*potential defect list*). Cada elemento — resistor, capacitor, diodo ou transistor MOS — é descrito por duas condições de falha possíveis: preLRL e preHRL, sendo assim essa lista considera 44 defeitos. Rodando o circuito na janela “AnalogCircuitTest”, descrita na Seção 3.1, o algoritmo de determinação

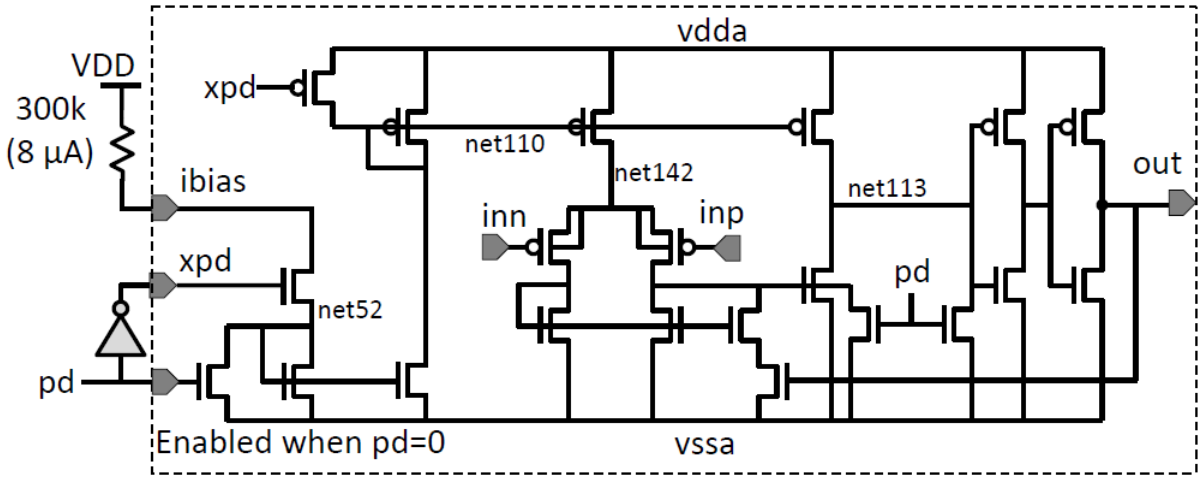
automática gerou um modelo de falhas contendo 103 falhas catastróficas, duas (uma *open* e outra *short)* em cada componente de dois terminais (resistores, capacitores) e 6 falhas em cada um dos 16 transistor (3 *opens* e 3 *shorts*) excluindo 5 falhas de curto-circuito entre terminais de transistores que já possuíam essa ligações na própria topologia do circuito. Esse modelo com 103 falhas catastróficas foi utilizado na simulação de falhas, não foram consideradas falhas paramétricas, uma vez que a lista de defeitos disponibilizada pelo grupo P2427 trata de falhas catastróficas.

Analogamente ao caso do OPAMP1, nesse caso do regulador de tensão, o pino de tensão de referência é que foi considerado como nó de entrada, sendo simulados sinais de tensões contínua de 0V até 3,3V com passos de 1mv e sinais de tensões AC de 100Hz até 10GHz simulando 100 pontos por década. Além disso, foi considerado um capacitor de carga ligado ao pino de saída (out) de 1nF em paralelo com um resistor de carga de 600Ohms, valores já presentes no arquivo disponibilizado pelo grupo P2427. Os limites de detecção foram determinados por *Corners* e novamente as precisões de tensão, corrente e tempo foram consideradas como sendo respectivamente: 100uV, 1nA e 0,1ns.

4.5.3 COMP1

O circuito COMP1, disponibilizado no conjunto de *A/MS Benchmark Circuits* associado ao IEEE P2427, representa um comparador CMOS de alta velocidade desenvolvido especificamente para servir como célula padrão em estudos de simulação de falhas, DfT e estratégias de teste orientadas a defeitos em circuitos analógicos e mistos (SUNTER; SARSON, 2017). O esquemático deste circuito é mostrado pela Figura 28. Assim como os demais blocos incluídos nesse *benchmark*, o COMP1 é fornecido com *netlist* SPICE completo, modelos de processo realistas, condições de teste detalhadas e uma lista estruturada de possíveis defeitos físicos, o que possibilita a execução de experimentos reprodutíveis e comparáveis entre diferentes pesquisadores. Diferentemente dos circuitos totalmente analógicos já analisados nesta tese, como o OPAMP1 e o VREG1, o comparador COMP1 introduz um comportamento fortemente não linear e dependente do regime transitório, fornecendo uma nova perspectiva para a metodologia adotada.

Figura 28 – COMP1 Comparador de alta velocidade com pino de habilitação (pd) disponibilizado pelo conjunto de benchmarks do P2427.



Fonte: IEEE Standards Association P2427.

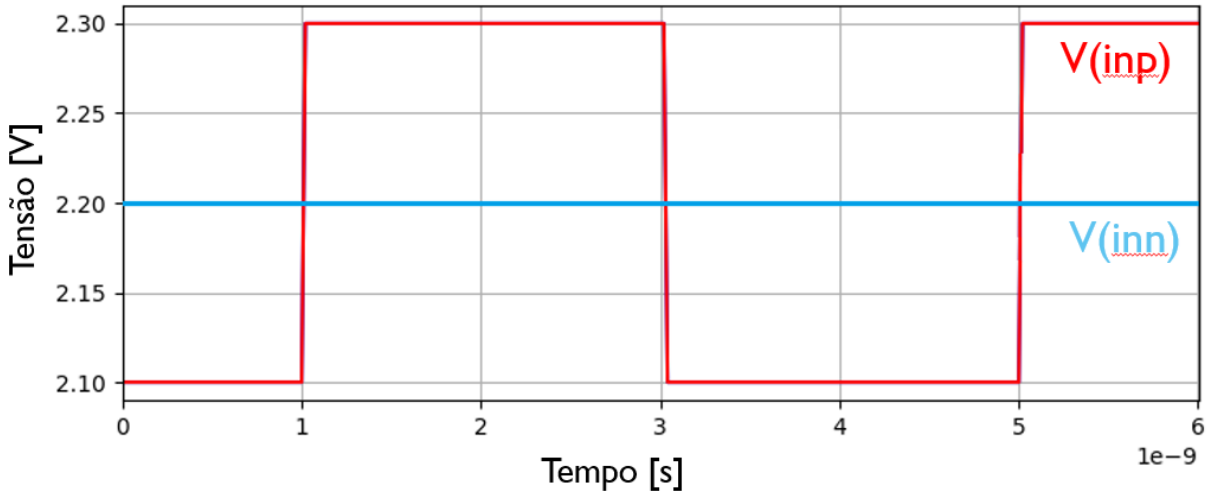
O COMP1 opera como um bloco de decisão, convertendo diferenças de tensão analógica em níveis digitais através de um conjunto de estágios de ganho e regeneração. Sua arquitetura inclui um par diferencial de entrada, cargas ativas, amplificação em cascata e elementos responsáveis por estabelecer histerese, além de um mecanismo de desligamento (*power-down*) que permite observar o comportamento de corrente estática em condições controladas. Essa estrutura, somada à alta sensibilidade a pequenos desbalanços e ao forte acoplamento entre características paramétricas e resposta temporal, faz do comparador um circuito particularmente adequado para estudos de detecção de defeitos em regime transitório. No contexto deste trabalho, essa característica torna o COMP1 complementar aos circuitos OPAMP1 e VREG1, uma vez que amplia o espectro de arquiteturas avaliadas, incluindo agora um circuito intrinsecamente não linear cujo funcionamento está intrinsecamente relacionado ao comportamento dinâmico, *overshoot* e tempo de acomodação.

O modelo de falhas adotado para o COMP1 segue a metodologia padronizada pelo grupo IEEE P2427, na qual cada componente do circuito é associado a uma lista de defeitos potenciais. Para esse circuito, a lista proposta pelo grupo P2427 compreende 42 falhas sendo duas falhas (uma *open* e outra *short*) em cada um dos 21 transistores que compõem o modelo. Rodando o circuito na janela “AnalogCircuitTest”, descrita na Seção 3.1, o algoritmo de determinação automática gerou um modelo de falhas contendo 123 falhas catastróficas, sendo seis falhas em cada um dos 21 transistor (3 *opens* e 3 *shorts*) excluindo 3 falhas de curto circuito entre terminais de transistores que já possuíam essas ligações na própria topologia do circuito. Assim como nos outros circuitos desse mesmo *benchmark*, esse modelo com 123 falhas

catastróficas foi utilizado na simulação de falhas, não foram consideradas falhas paramétricas, uma vez que a lista de defeitos disponibilizada pelo grupo P2427 trata de falhas catastróficas.

Analogamente aos demais circuitos simulados deste mesmo *benchmark* (Seções 4.5.1 e 4.5.2), os limites de detecção do COMP1 foram determinados a partir da análise dos diferentes *corners* de processo. As simulações foram conduzidas exclusivamente no regime transitório, uma vez que o comparador apresenta comportamento fortemente dependente da dinâmica de comutação. Para esse conjunto de experimentos, a tensão de referência conectada à entrada negativa (inn) foi fixada em 2,2 V, conforme definido no próprio *netlist* disponibilizado pelo grupo IEEE P2427. A entrada positiva (inp), mostrada na Figura 28, foi excitada por um degrau de tensão variando de 2,1 V para 2,3 V, representando a condição de transição necessária para provocar a comutação interna do comparador, assim como mostra a Figura 29.

Figura 29 – Degrau de tensão aplicado às entradas do comparador COMP1 para testes transientes.

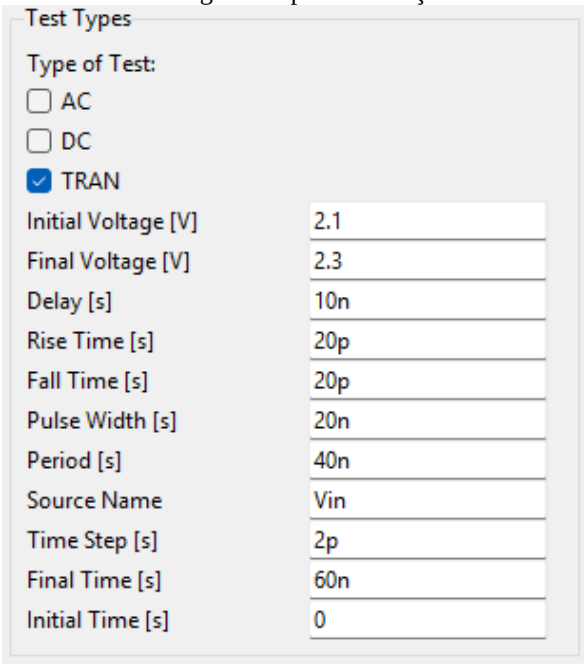


Fonte: elaborado pelo autor.

Os demais parâmetros associados à simulação transitória encontram-se apresentados na Figura 30. Ressalta-se que os degraus de entrada foram configurados de forma a garantir a completa caracterização do comportamento dinâmico do circuito em ambas as direções de transição. Assim, além do estímulo de 2,1 V para 2,3 V, também foi aplicada a transição inversa, de 2,3 V para 2,1 V. Essa abordagem permite avaliar a sensibilidade do comparador à detecção de falhas diante das duas possibilidades de excitação transitória, assegurando que tanto o processo de subida quanto o de descida da entrada sejam analisados. A inclusão dos dois degraus é essencial, pois o mecanismo de regeneração interna do comparador pode apresentar assimetria em função das características dos transistores e do impacto de defeitos inseridos no

circuito. Dessa forma, a metodologia adotada possibilita uma avaliação mais completa e robusta das condições de detecção de falhas no COMP1.

Figura 30 – Parâmetros configurados para simulação transiente do COMP1.



Fonte: elaborado pelo autor.

O uso do COMP1 neste trabalho cumpre, portanto, três funções principais. Primeiro, ele proporciona a avaliação da metodologia de teste em um cenário onde a resposta é fortemente não linear, permitindo verificar a capacidade do método de distinguir variações abruptas e fenômenos altamente dependentes do comportamento transitório. Por fim, incluir um comparador de alta velocidade como parte do conjunto de circuitos analisados confere maior generalidade à tese, demonstrando que a metodologia não se limita a blocos amplificadores ou reguladores, mas também se aplica a circuitos de decisão essencialmente mistos, que combinam características analógicas e digitais.

Por fim, realizou-se uma análise adicional aplicada ao circuito comparador COMP1, com o objetivo de confrontar a heurística utilizada na determinação do melhor conjunto de testes por parte da ferramenta. Para isso, procedeu-se a uma avaliação exaustiva de todas as combinações possíveis de testes, investigando se alguma delas seria capaz de superar a cobertura de falhas obtida pelo algoritmo de otimização ou, alternativamente, se seria possível alcançar o mesmo nível de cobertura utilizando um número menor de configurações de teste.

Essa análise foi conduzida especificamente para o comparador devido ao fato de que, nesse caso, foi empregada uma análise transiente na qual existem apenas quatro testes possíveis

para cada variável nodal do circuito. Esses testes correspondem às métricas de *overshoot* e tempo de acomodação associadas a degraus de tensão de subida e de descida aplicados na entrada do circuito.

Dessa forma, o número total de configurações de teste possíveis torna-se significativamente menor do que nos casos em que foram realizadas análises DC e AC. Nessas abordagens, devido às múltiplas variações incrementais dos sinais de entrada nas simulações, o número de possíveis testes associados a cada variável nodal chega até mesmo a ordem de milhares.

5 RESULTADOS E DISCUSSÕES

Neste 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 neste 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 computador com 64bits de barramento de dados, processador Intel® Core™ i-7 de 3,6 GHz e memória RAM de 16GB.

Tabela 3 - Tempos de execução do algoritmo.

Seção	Circuito	Análises	Tempo 1	Tempo 2	Nº de Falhas Analisadas
4.1.1	Filtro *Butterworth*	AC e DC	16min	28s	160
4.1.2	Filtro *Butterworth*	Transiente	11min	16s	160
4.2	Filtro *Butterworth*	Parâmetros Funcionais	-	12s	160
4.3.1	Filtro Dois Estágios	AC e DC	28min	55s	288
4.3.2	Amplificador Malha Aberta	AC e DC	11min	26s	120
4.3.3	Amplificador Malha Aberta	AC e DC nó adicional	-	2s	120
4.4.1	Amplificador com CD4007	DC	23s	6s	48
4.5.1	OPAMP1	AC e DC	30s	7s	89
4.5.2	VREG1	AC e DC	41s	8s	103
4.5.3	COMP1	TRAN	3min e 22s	1,5s	123

Fonte: Elaborado pelo autor.

Vale destacar que os estudos de caso apresentados nas Seções 4.1, 4.2 e 4.3 foram executados utilizando a primeira versão da ferramenta desenvolvida em MATLAB, enquanto os estudos das Seções 4.4 e 4.5 foram conduzidos com a nova implementação em Python. Na versão em MATLAB, as chamadas ao simulador SPICE eram realizadas por meio de comandos DOS, executados diretamente no sistema operacional. Esses comandos nada mais são do que instruções enviadas ao terminal do Windows para iniciar processos externos — no caso, instâncias independentes do simulador NGSPICE. Embora essa abordagem permita que múltiplas simulações sejam iniciadas em paralelo, ela não oferece mecanismos adequados de controle, sincronização ou monitoramento sobre o término desses processos.

Para mitigar parcialmente essa limitação, um atraso manual era introduzido entre chamadas consecutivas ao simulador, com o objetivo de evitar que todas as simulações fossem iniciadas simultaneamente. Contudo, esse procedimento não garantia sincronismo real. Em diversos casos, o atraso configurado era inferior ao tempo necessário para a conclusão da simulação anterior, resultando na execução paralela indesejada de múltiplas instâncias. Em outros casos, o atraso era excessivo, criando intervalos ociosos entre o término de uma simulação e o início da próxima, reduzindo a eficiência total do processo. Assim, o controle temporal baseado em atrasos fixos mostrou-se impreciso e inadequado para garantir uma execução sequencial confiável.

Na nova versão da ferramenta desenvolvida em Python, essas limitações foram eliminadas. A utilização da biblioteca *subprocess* — um módulo nativo da linguagem projetado especificamente para criar, gerenciar e sincronizar processos externos — permite que o programa invoque o simulador SPICE de forma totalmente controlada. Com o uso de funções como subprocess.run() ou subprocess.Popen(), o algoritmo pode iniciar cada simulação individualmente e, de maneira automática, aguardar a sua conclusão antes de iniciar a próxima. Esse comportamento garante sincronização explícita entre as execuções, sem a necessidade de atrasos artificiais ou estimativas temporais. Além disso, o *subprocess* possibilita capturar saídas, receber códigos de retorno e tratar erros de forma estruturada, garantindo maior robustez ao fluxo de execução.

Dessa forma, a migração da ferramenta para Python não apenas removeu as ineficiências decorrentes da tentativa de sincronização manual presente na versão em MATLAB, mas também resultou em um processo de simulação mais determinístico, otimizado e significativamente mais rápido. O controle integral sobre cada chamada ao simulador, proporcionado pela biblioteca *subprocess*, contribuiu diretamente para a confiabilidade e reprodutibilidade dos resultados apresentados nesta tese.

5.1 FILTRO BUTTERWORTH DE UM ESTÁGIO

Nessa seção, serão detalhados os resultados do estudo de caso do filtro Butterworth de um estado: em um primeiro momento, resultados de testes estruturais nas Seções 5.1.1 e 5.1.2; posteriormente, resultados de simulação de testes funcionais na Seção 5.2.

5.1.1 Testes AC e DC

Para as condições descritas na Seção 4.1.1, o conjunto final de testes é composto por quatro testes: dois em regime AC e dois em regime DC. A Tabela 4 apresenta esse conjunto de

testes obtido por meio da ferramenta utilizada, juntamente com os respectivos nós de observação. Além disso, a tabela também exibe a cobertura acumulada de falhas à medida que os testes são aplicados sequencialmente, de acordo com o índice de teste. Considerando esse conjunto de testes, a cobertura total de falhas alcançada foi de 96,25%.

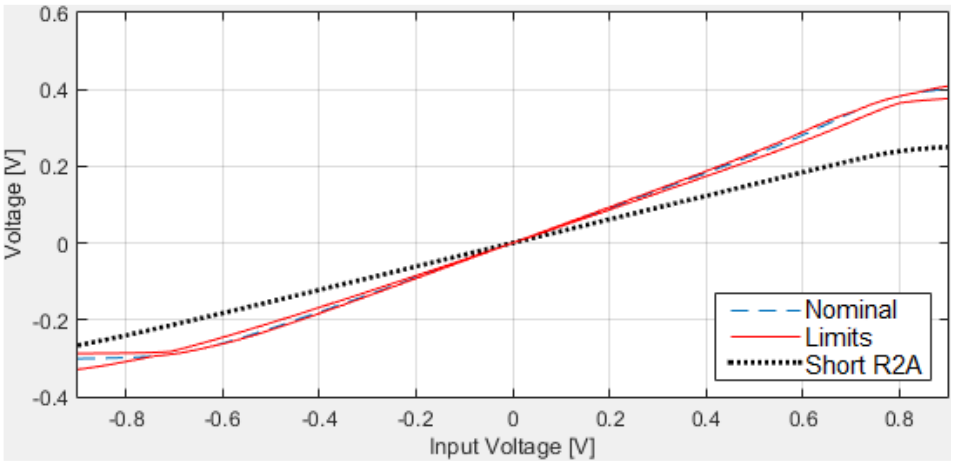
Tabela 4 – Melhor conjunto de testes.

Índice do Teste	Tipo do Teste	Sinal de Entrada	Nó	Cobertura de Falhas Cumulativa
1	AC	1,585MHz	Vo-	56,25%
2	AC	58,88KHz	Vo-	71,88%
3	DC	801mV	Vo+	91,88%
4	DC	639mV	Vo-	96,25%

Fonte: Elaborado pelo autor.

A Figura 31 apresenta a resposta da análise de varredura DC realizada no nó Vo+, juntamente com os limites de teste definidos pela ferramenta e um exemplo de resposta DC do nó quando um curto é injetado em R2A. Nesse estudo de caso, os limites de teste foram determinados por meio da análise de *corners.*

Figura 31 – Resposta aos diferentes sinais DC de entrada para circuito sem falhas e para o caso de curto-circuito no resistor R2A.

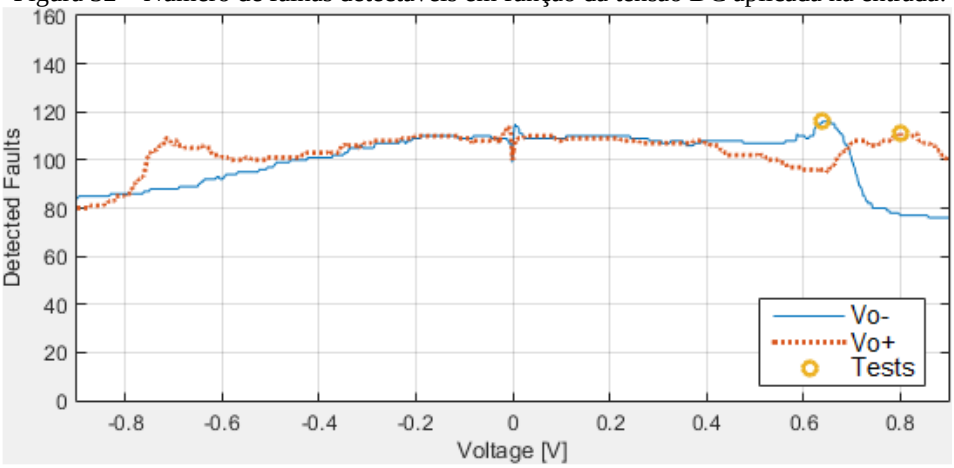


Fonte: elaborado pelo autor.

Já a Figura 32 ilustra uma das saídas gráficas da ferramenta, mostrando a quantidade de falhas detectáveis em cada possível nó de teste selecionado, em função das tensões DC de entrada simuladas. Sendo assim, nesse gráfico, cada curva representa o resultado associado a um possível nó de teste, enquanto que cada ponto ao longo dessas curvas indica o número de falhas detectadas (eixo das ordenadas) quando a tensão desse nó é medida sob a aplicação do

sinal de entrada correspondente ao eixo das abscissas. Observa-se que o algoritmo escolheu os dois testes associados aos valores máximos dessas funções.

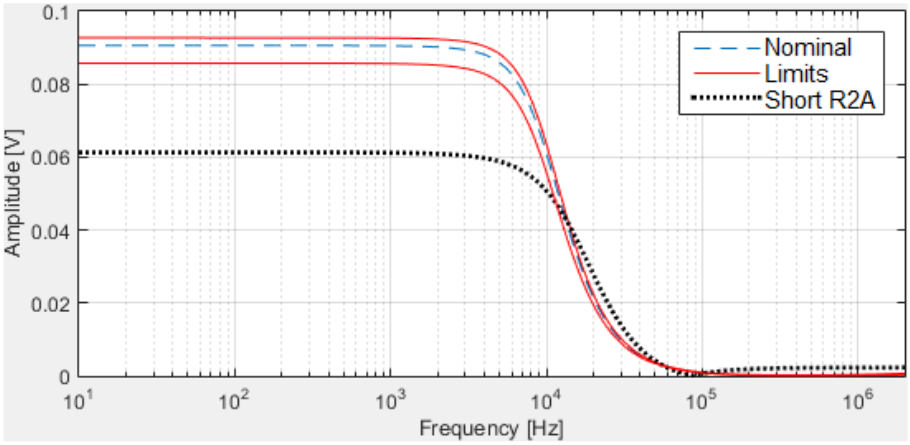
Figura 32 – Número de falhas detectáveis em função da tensão DC aplicada na entrada.



Fonte: elaborado pelo autor.

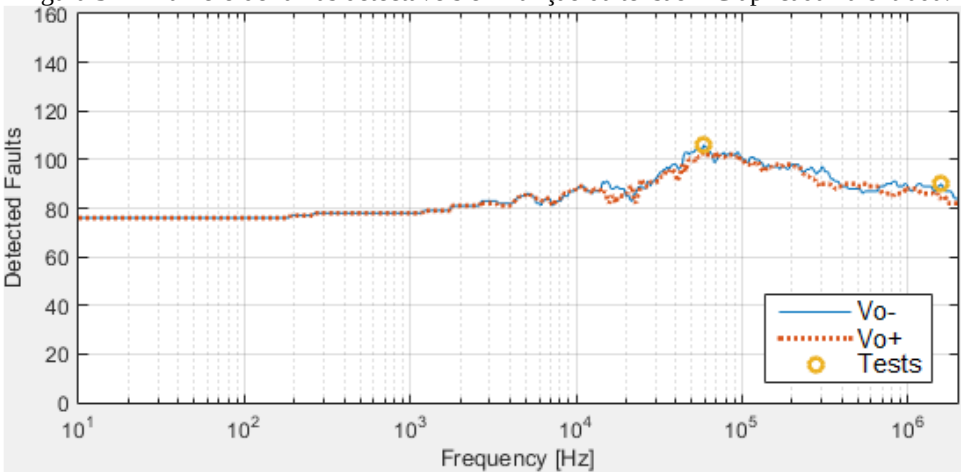
Na Figura 33, é apresentada a resposta em frequência do nó Vo+, incluindo os limites de teste e um exemplo de resposta defeituosa. A Figura 34 exibe a quantidade de falhas detectáveis em cada nó de saída selecionado, em função da frequência do sinal de entrada. Nesse gráfico, cada curva representa o resultado associado a um possível nó de teste, enquanto que cada ponto ao longo dessas curvas indica o número de falhas detectadas (eixo das ordenadas) quando a amplitude de tensão desse nó é medida sob a aplicação do sinal de entrada com frequência correspondente ao eixo das abscissas. Neste caso, a marcação nos testes escolhidos (Figura 34) revela que o primeiro teste AC da Tabela 4 (índice 1) não corresponde a um valor máximo da função. Esse fato destaca a importância do primeiro critério de decisão adotado pelo algoritmo: priorizar a detecção da falha mais difícil de identificar e garantir sua detecção.

Figura 33 - Resposta aos diferentes sinais AC de entrada para circuito sem falhas e para o caso de curto-circuito no resistor R2A.



Fonte: elaborado pelo autor.

Figura 34 – Número de falhas detectáveis em função da tensão AC aplicada na entrada.



Fonte: elaborado pelo autor.

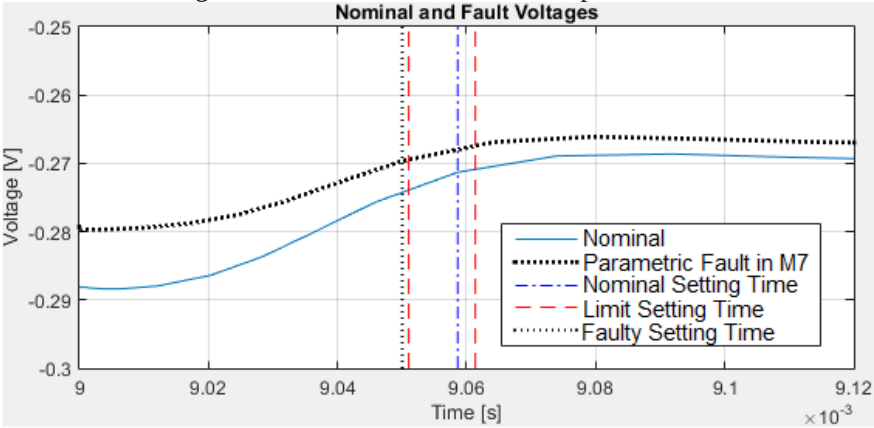
5.1.2 Teste Transiente Adicional

Como mostrado na Tabela 4, o conjunto final de testes inclui dois testes DC. Uma vez que a execução desses testes exige a estabilização do circuito em duas tensões DC distintas, analisou-se a possibilidade de utilizar o tempo de transição entre esses estados para realizar um teste transiente adicional. Esse teste extra, viabilizado pela ferramenta, permite aumentar a cobertura de falhas sem impactar o tempo total de execução do conjunto de testes.

Os resultados demonstraram que um teste transiente utilizando um degrau de 801 mV para 639 mV pode detectar uma falha adicional ao medir o tempo de acomodação do nó de saída negativo (Vo−). Em outras palavras, se esse teste fosse incorporado ao conjunto da Tabela 4, a cobertura de falhas aumentaria para 96,87%, sem acréscimo no tempo de execução e, consequentemente, sem impacto no custo do teste. A Figura 35 apresenta a falha adicional

detectada pelo teste transiente. Essa falha provoca uma variação de 50 mV na tensão de limiar do transistor M7 no bloco de ganho do FDA (Figura 20).

Figura 35 – Falha adicional detectada pelo teste transiente.



Fonte: elaborado pelo autor.

5.2 DETECÇÃO DE FALHAS ATRAVÉS DE TESTES FUNCIONAIS

Na avaliação do teste funcional, 125 falhas foram detectadas, considerando os parâmetros funcionais utilizados como resposta nos testes simulados. Isso resulta em uma cobertura de falhas de 78,13% quando todos os parâmetros são analisados. A Tabela 5 apresenta a cobertura de falhas alcançada ao avaliar cada parâmetro funcional em ambos os nós de saída.

Tabela 5 – Resultados testes funcionais.

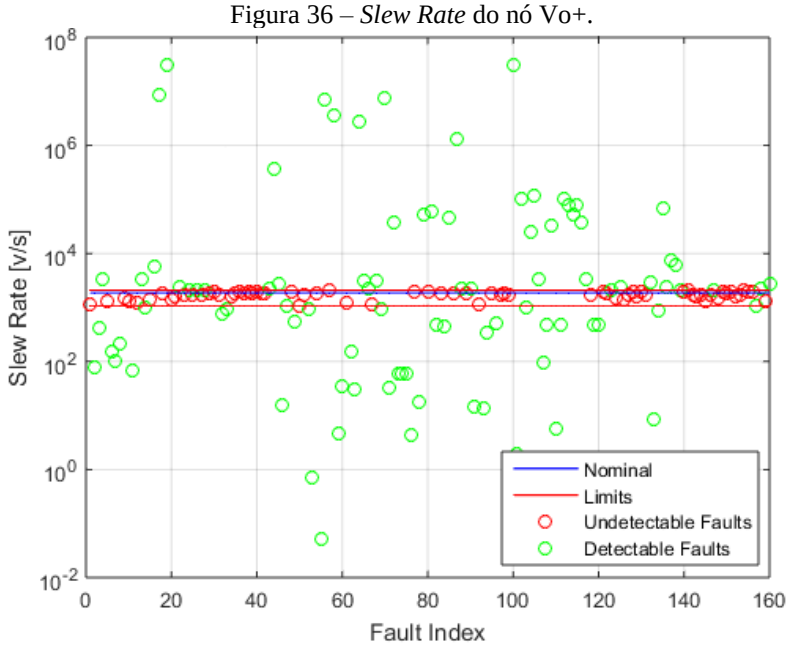
Parâmetro Funcional	Nó	Falhas Detectáveis	Cobertura de Falhas
Ganho de Tensão	Vo+	76	47,50%
Ganho de Tensão	Vo-	76	47,50%
Slew Rate	Vo+	92	57,50%
Slew Rate	Vo-	104	65,00%
Frequência de -3dB	Vo+	97	60,63%
Frequência de -3dB	Vo-	97	60,63%
TOTAL	**Vo+/Vo-**	**125**	**78,13%**

Fonte: Elaborado pelo autor.

Na Figura 36 e na Figura 37 estão os resultados do parâmetro *Slew Rate* para todas as falhas simuladas, para os respectivos nós Vo+ e Vo−. Nessas figuras, cada círculo representa uma falha individual, enquanto o eixo das abscissas corresponde a um índice sequencial utilizado para a identificação de cada falha. As linhas horizontais em vermelho indicam os limites de tolerância, os quais correspondem aos piores casos obtidos a partir da análise de

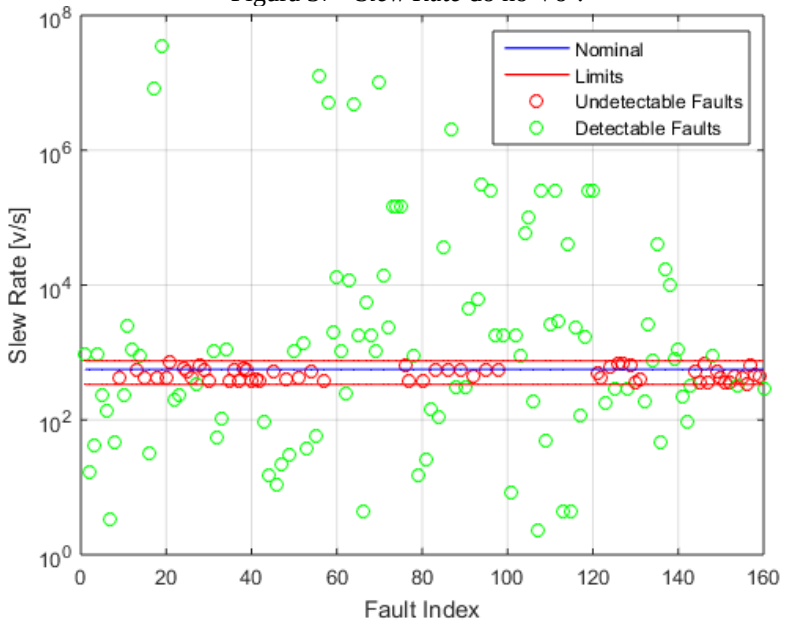
corners. Dessa forma, os circuitos com falhas que apresentam taxas de variação (*slew rate*) fora dos limites de aceitação são representados por círculos verdes, indicando que essas falhas podem ser detectadas por esse teste específico. Da mesma forma, valores de SR dentro dos limites são representados por círculos vermelhos, sinalizando que essas falhas não foram detectadas por esse teste.

Figura 36 – *Slew Rate* do nó Vo+.



Fonte: elaborado pelo autor.

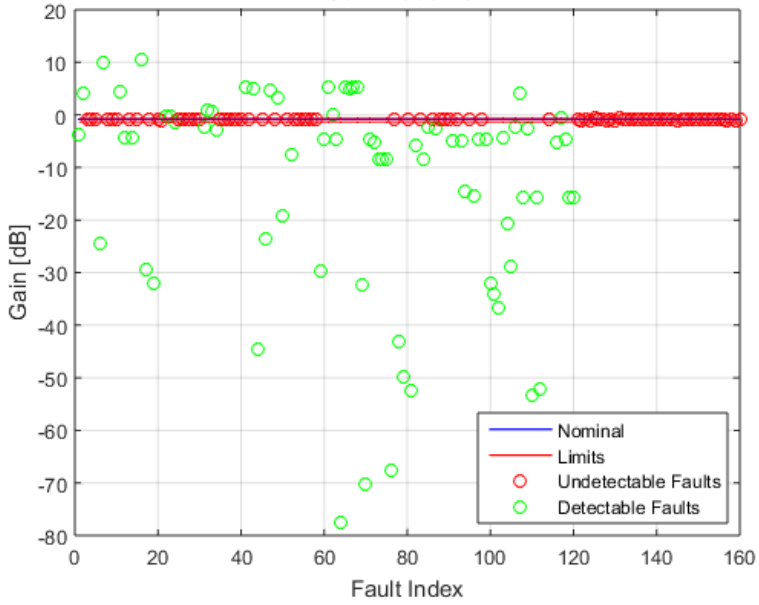
Figura 37 - *Slew Rate* do nó Vo-.



Fonte: elaborado pelo autor.

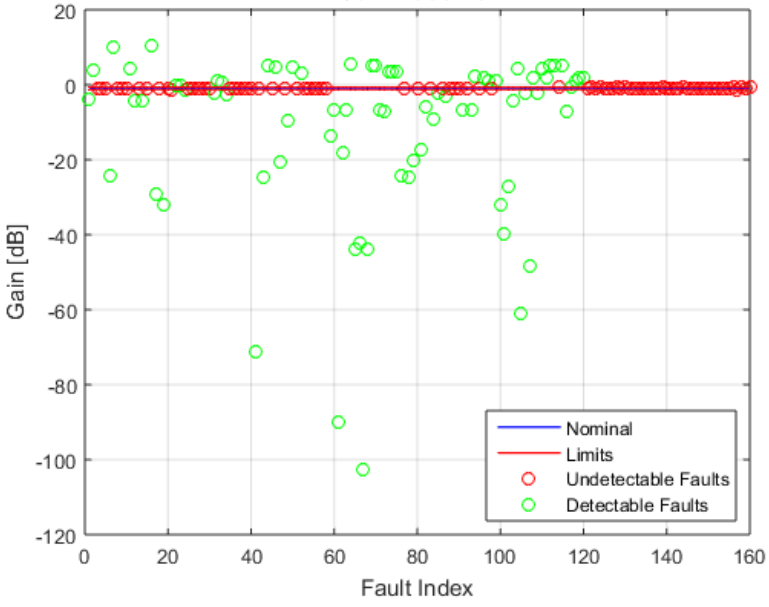
De maneira similar, as Figuras Figura 38, Figura 39, Figura 40 e Figura 41 mostram os resultados considerando os parâmetros de ganho DC e frequência de corte do filtro, quando avaliados no nós Vo+ e Vo- como ponto de observação. É interessante notar que os diferentes parâmetros apresentam uma dispersão distinta em relação às falhas injetadas, como pode ser observado na comparação entre os gráficos de SR e ganho, por exemplo. Essa variação também pode ser utilizada para auxiliar os desenvolvedores de teste a avaliarem qualitativamente a sensibilidade de cada parâmetro em relação às falhas injetadas.

Figura 38 – Ganho de tensão para nó Vo+.



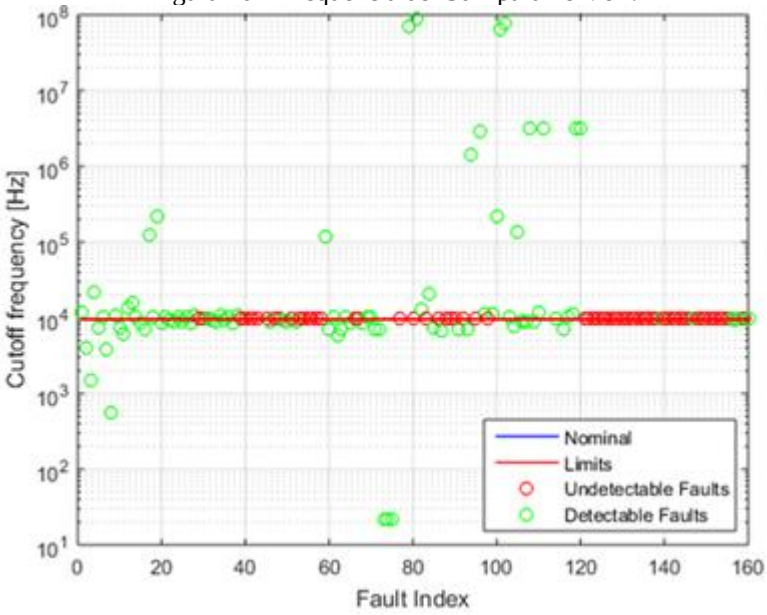
Fonte: elaborado pelo autor.

Figura 39 - Ganho de tensão para nó Vo-.



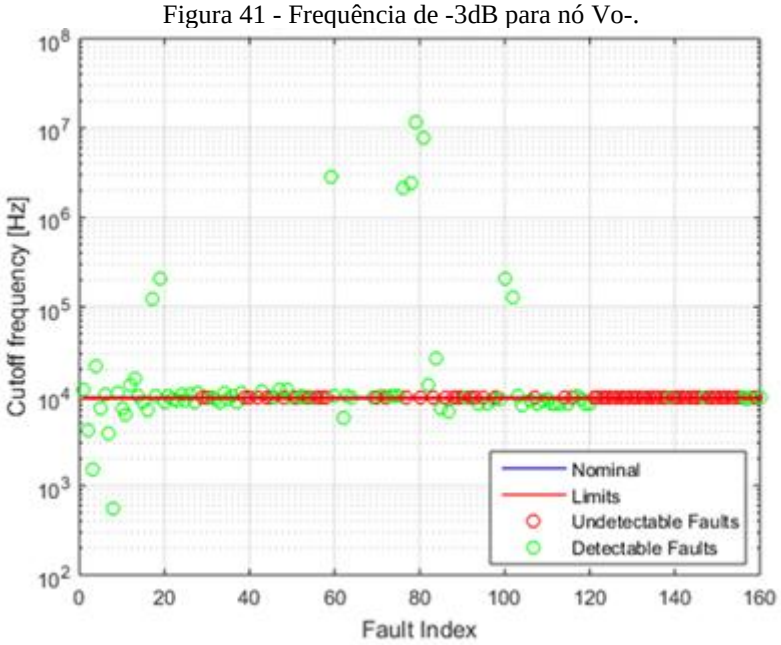
Fonte: elaborado pelo autor.

Figura 40 – Frequência de -3dB para nó Vo+.



Fonte: elaborado pelo autor.

Figura 41 - Frequência de -3dB para nó Vo-.



Fonte: elaborado pelo autor.

Como pode ser observado na Seção 5.1.1, a ferramenta automática de geração de conjuntos de teste selecionou um conjunto com apenas quatro testes, alcançando uma cobertura de falhas de 96,25%. Além disso, esses quatro testes estão relacionados a medições simples de tensões DC e amplitudes de tensões AC. Enquanto isso, os testes funcionais estão associados a parâmetros cuja medição é mais complexa. Ainda assim, neste estudo de caso, três testes funcionais aplicados em ambos os nós de saída atingiram uma cobertura de falhas de apenas 78,12%.

5.3 FILTRO PASSA-BAIXAS DE DOIS ESTÁGIOS

Nessa seção serão expostos os resultados de simulação do filtro passa-baixas de dois estágios. Além disso, serão expostos os resultados de incremento de cobertura de falhas possibilitado pelo uso de técnicas de DfT.

5.3.1 Análises AC e DC do Filtro Passa-Baixas Completo

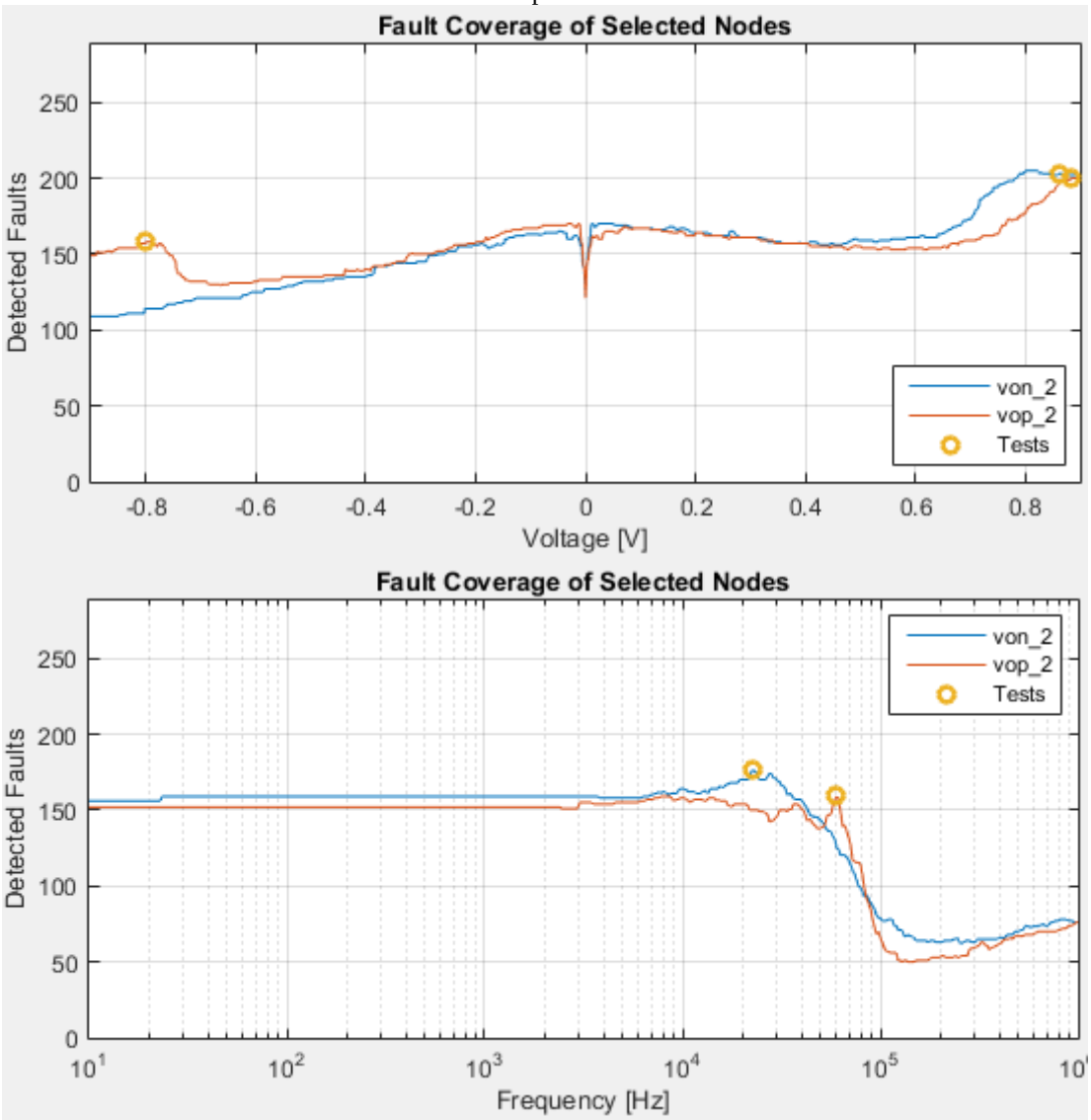
As simulações realizadas com o circuito completo do filtro passa-baixas de dois estágios demonstraram que é possível alcançar uma cobertura de falhas de 87,15% com o conjunto de testes apresentado na Tabela 6. Nesta configuração, apenas os dois nós de saída primários do filtro foram considerados como pontos de observação. Além disso, na Figura 42, pode-se observar o número de falhas detectadas em cada configuração de teste, bem como a marcação dos testes escolhidos pela ferramenta.

Tabela 6 – Melhor conjunto de testes filtro completo.

Índice do Teste	Tipo do Teste	Sinal de Entrada	Nó	Cobertura de Falhas Cumulativa
1	AC	22387Hz	Vo-	61,11%
2	DC	885mV	Vo+	82,99%
3	DC	861mV	Vo-	84,72%
4	AC	60255Hz	Vo+	86,11%
5	DC	-800mV	Vo+	87,15%

Fonte: Elaborado pelo autor.

Figura 42 – Números de falhas detectáveis nos nós de saída do filtro passa-baixas de dois estágios em função de sinais DC e AC aplicados na entrada.



Fonte: elaborado pelo autor.

Considerando que o modelo de falhas completo contempla um total de 288 falhas, uma cobertura de 87,15% implica que apenas 37 falhas permanecem não detectadas pelos cinco testes aplicados. Para ampliar a detecção de falhas, torna-se fundamental realizar testes com os amplificadores isolados do restante do circuito, uma vez que 35 das 37 falhas remanescentes estão associadas a componentes internos dos amplificadores. Esses resultados são mostrados na Seção 5.3.2.

5.3.2 Amplificador Totalmente Diferencial em Malha Aberta

Conforme ilustrado na Figura 23, os amplificadores totalmente diferenciais foram simulados em configuração de malha aberta e os resultados de detecção de falhas são mostrados

na Figura 43. Os resultados obtidos indicam que é possível detectar 112 falhas em cada amplificador mediante a aplicação de apenas 3 testes. A Tabela 7 apresenta esses 3 testes adicionais.

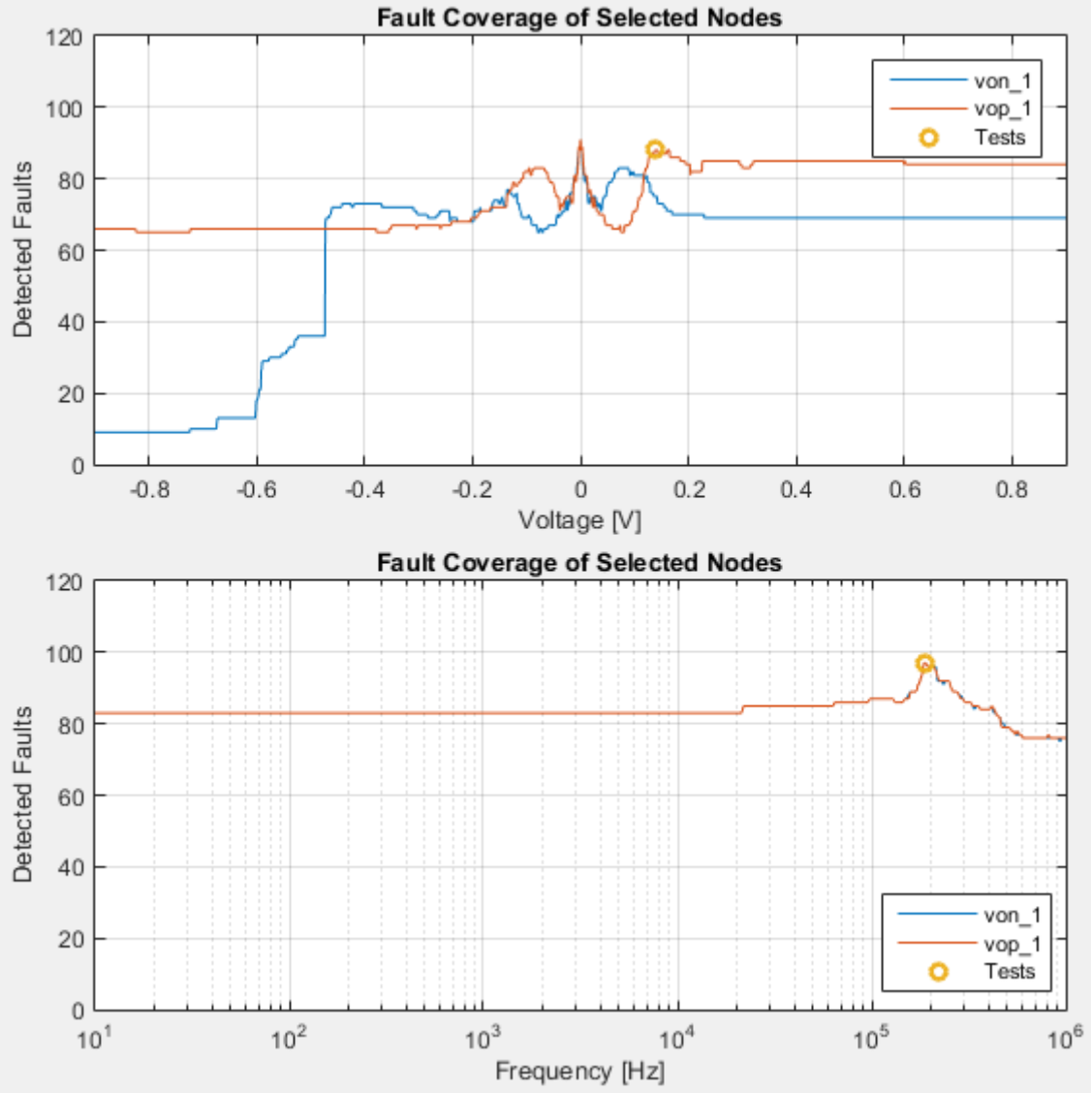
No caso do amplificador do primeiro estágio, verificou-se que 14 falhas, que não puderam ser detectadas pelo conjunto de testes da Tabela 6, podem agora ser identificadas com o conjunto de testes da Tabela 7. Para o amplificador do segundo estágio, foi possível detectar 7 falhas adicionais utilizando o conjunto de testes da Tabela 7.

Tabela 7 – Melhor conjunto de testes amplificador em malha aberta.

Índice do Teste	Tipo do Teste	Sinal de Entrada	Nó	Cobertura de Falhas Cumulativa
1	AC	186209Hz	Vo-	80,83%
2	AC	186209Hz	Vo+	82,50%
3	DC	138mV	Vo+	93,33%

Fonte: Elaborado pelo autor.

Figura 43 – Número de falhas detectáveis nos nós de saída para amplificador totalmente diferencial em malha aberta em função de sinais DC e AC aplicados na entrada.



Fonte: elaborado pelo autor.

Portanto, a adição de estruturas de DfT ao circuito, que viabilizem a execução de testes em malha aberta dos amplificadores, possibilita a detecção de um total de 21 falhas adicionais. Esse incremento resulta em uma nova cobertura de falhas de 94,44%. Nessa nova configuração, 16 falhas permanecem indetectáveis: 14 associadas a componentes internos dos amplificadores e 2 relacionadas a componentes passivos do filtro, sendo elas, os desvios de +25% no valor da resistência R2_1 e da capacitância C1_1 (Figura 22).

Em uma análise preliminar, pode-se supor que todas as 8 falhas não detectáveis nos testes realizados com o amplificador em configuração de malha aberta também permaneceriam indetectáveis quando o amplificador fosse submetido aos testes com o filtro completo. Tal hipótese foi confirmada para o amplificador do primeiro estágio (A1). Entretanto, verificou-se

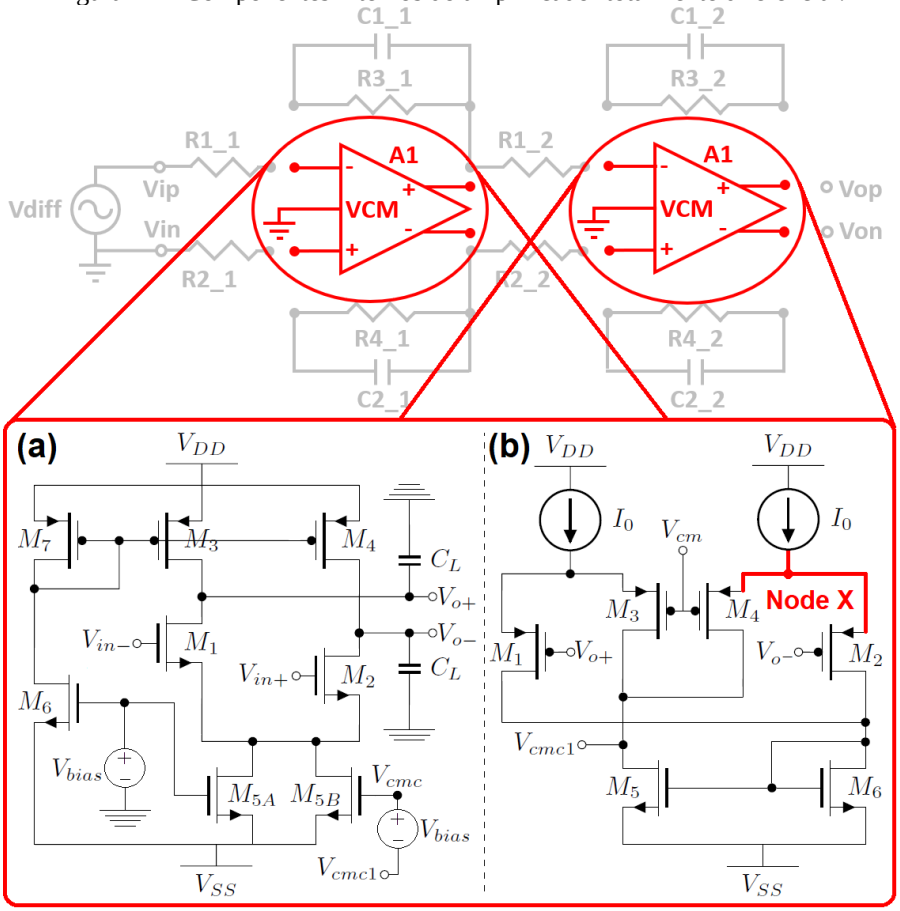
uma exceção no segundo estágio: uma falha específica, correspondente a um desvio de -25% no comprimento de canal do transistor X5b, pertencente ao bloco amplificador do FDA do segundo estágio do filtro, foi detectada durante os testes do filtro completo apresentados na Seção 5.3.1, apesar de não ter sido detectada nos testes realizados com o amplificador isolado em malha aberta, conforme discutido nesta seção.

Visando aumentar ainda mais a cobertura de falhas, especialmente sobre as 14 falhas ainda não detectadas nos componentes internos dos amplificadores, foi considerado o monitoramento de um nó interno adicional como ponto de observação no modo de teste em malha aberta. Todos os nós internos do circuito foram facilmente analisados utilizando a ferramenta e o resultado é mostrado na Seção 5.3.3.

5.3.3 Utilização Adicional de um Nó Interno do Amplificar Como Nó de Teste

Buscando cobrir pelo menos parte das falhas não detectadas, um nó interno do amplificador foi considerado como ponto adicional de observação no modo de teste em malha aberta. Foi identificado que a observação do nó mostrado na Figura 21, que conecta as fontes dos transistores M2 e M4, permite detectar duas falhas adicionais. Esse nó, denominado "nó X", está ilustrado na Figura 44. Sendo assim, caso esse nó X seja acessível por meio de estruturas internas de DfT durante a execução dos testes, a cobertura total de falhas poderá ser elevada para 95,83%.

Figura 44 – Componentes internos do amplificador totalmente diferencial.



Fonte: elaborado pelo autor.

Naturalmente, a adição de estruturas de DfT implica em uma redução de rendimento (*overhead* de área) e, em alguns casos, pode acarretar penalidades no desempenho do circuito. Existe, portanto, um *trade-off* entre esses impactos adversos e o incremento na cobertura de falhas, o qual deve ser criteriosamente analisado pelo projetista ou pelo engenheiro de DfT. Caso a implementação de DfT seja realizada com o objetivo de garantir uma qualidade mínima de teste, possibilitando a aplicação de métodos de teste estrutural, a redução no rendimento pode ser facilmente compensada pela diminuição do tempo de teste, especialmente quando comparado a metodologias baseadas em especificações funcionais. Essa redução no tempo de teste contribui, por sua vez, para a diminuição do custo total do circuito integrado, uma vez que o processo de teste pode representar uma parcela significativa do custo final do dispositivo.

5.4 VALIDAÇÃO DA FERRAMENTA ATRAVÉS DE EMULAÇÃO DE FALHAS POR HARDWARE

Nessa seção, serão expostos os resultados obtidos para o circuito amplificador construído com CD4007. Em um primeiro momento, na Seção 5.4.1 serão expostos os

resultados de simulação. Em um segundo momento, na Seção 5.4.2 serão expostos os resultados das medições experimentais.

5.4.1 Simulação de Configurações de Teste

Com a execução do algoritmo em Python da ferramenta de geração de testes, foi sugerida, neste caso, uma sequência de teste composta por um único teste em corrente contínua (DC), com tensão de 72mV na entrada do circuito. Nesse teste, o nó de saída é medido de modo que tensões dentro da faixa de -396mV até -324mV são consideradas dentro dos limites de aceitação e tensões fora dessa faixa caracterizam o circuito como falho. Na Tabela 8, pode-se observar a cobertura de falhas obtida com esse teste. Sendo assim, apenas 5 das 48 falhas do modelo não puderam ser detectadas através desse teste.

Tabela 8 – Melhor conjunto de testes amplificador com CD4007.

Índice do Teste	Tipo do Teste	Sinal de Entrada	Medição	Falhas Detectadas	Falhas Acumuladas	Cobertura de Falhas Cumulativa
1	DC	72mv	v(out)	43	43	89,58%

Fonte: Elaborado pelo autor.

5.4.2 Medição Experimental

A injeção física das mesmas 48 falhas permitiu realizar uma comparação direta entre o comportamento real do circuito e os resultados previstos pela ferramenta. Os resultados experimentais de detecção são apresentados na Tabela 9. Observa-se uma similaridade entre os dois conjuntos de dados: a ferramenta estimou, via simulação SPICE, que 43 das 48 falhas seriam detectáveis pela execução de um único teste DC, enquanto a emulação física demonstrou que 41 falhas foram efetivamente identificadas quando o mesmo teste foi aplicado ao protótipo em *hardware*. Essa proximidade confirma a coerência entre a modelagem de falhas, a metodologia de seleção de testes e a resposta prática do circuito.

Tabela 9 – Conjunto de testes II OPAMP1 considerando a possibilidade de medição da tensão no nó de saída, corrente de alimentação, corrente de entrada e corrente nos pinos pd e xpd.

Tipo de Falha	Terminal	Quantidade	Nº de falhas detectáveis simulação	Nº de falhas detectáveis emulação
Aberto	Dreno	7	6	7
Aberto	Gate	7	5	2
Aberto	Fonte	7	7	7
Aberto	Resistor	4	3	3
Curto	Dreno-Gate	5	4	4
Curto	Dreno-Fonte	7	7	7
Curto	Gate-Fonte	7	7	7
Curto	Resistor	4	4	4

Fonte: Elaborado pelo autor.

A principal discrepância entre simulação e experimento refere-se às falhas de circuito aberto no terminal de *gate* dos transistores, conforme evidenciado na Tabela 9. Essa divergência é tecnicamente justificável, considerando que o teste utilizado é estritamente DC. Em regime permanente, a corrente de *gate* de um MOSFET ideal tende a zero; portanto, a inserção de uma resistência de alto valor (10 MΩ) em série com o terminal de *gate* exerce impacto limitado sobre os valores estáticos de tensão do circuito. Nesses casos, a diferença entre o circuito íntegro e o circuito com falha manifesta-se predominantemente nas características dinâmicas — como tempos de carregamento de capacitâncias parasitas e respostas transitórias — que não são exploradas em testes DC. Assim, falhas de *gate* aberto são classicamente mais detectáveis por testes AC ou transitórios, nos quais a impedância efetiva do *gate* altera significativamente o comportamento temporal do sinal.

Os resultados experimentais confirmam essa tendência: apenas duas falhas de *gate* aberto foram detectadas no *hardware*, em contraste com o número maior previsto pela simulação. Tal diferença decorre de limitações inerentes ao modelo de simulação utilizado. O NGSPICE emprega modelos MOS relativamente simplificados e com parametrizações genéricas, que podem superestimar o efeito de determinadas falhas em análises puramente estáticas. Além disso, a ausência de variabilidade de processo no modelo SPICE reduz o realismo da simulação, particularmente no que diz respeito ao comportamento de alta impedância associado aos terminais de *gate*.

Apesar dessas limitações, a concordância global entre simulação e experimento — 90% de cobertura estimada contra 85% obtida — demonstra que a ferramenta fornece previsões

adequadas para o propósito de seleção automatizada de testes em circuitos analógicos sujeitos a falhas catastróficas.

5.5 GRUPO P2427 E BENCHMARK DE CIRCUITOS ANALÓGICOS

Neste trabalho, a ferramenta proposta foi aplicada a três circuitos pertencentes ao conjunto P2427: o amplificador operacional OPAMP1, o regulador de tensão VREG1 e o comparador COMP1. Cada um desses circuitos apresenta características estruturais e funcionais distintas que permitem explorar diferentes aspectos da metodologia desenvolvida. O OPAMP1, por sua natureza amplificadora e comportamento rico em regime AC, possibilita avaliar a eficácia de testes baseados em frequência e a exploração de medições de corrente como observáveis adicionais. O VREG1, por sua vez, representa um circuito de regulação cuja resposta é altamente dependente de pequenas variações em parâmetros internos, constituindo um cenário ideal para a investigação da complementaridade entre testes de módulo e fase. Por fim, o COMP1 oferece um ambiente adequado para a análise de testes transientes, permitindo avaliar o desempenho do algoritmo de otimização em cenários onde grandezas dinâmicas, como *overshoot* e tempo de acomodação, desempenham papel determinante na detecção de falhas.

A análise realizada para cada um desses circuitos envolve a execução sistemática de simulações considerando diferentes estímulos (DC, AC e transitórios), múltiplos tipos de medições (tensão e corrente) e um modelo de falhas abrangente. As subseções a seguir detalham os resultados alcançados em cada um dos três circuitos, destacando os conjuntos de testes selecionados automaticamente, sua cobertura de falhas, a contribuição incremental de cada teste e as análises complementares realizadas para validação da solução obtida.

5.5.1 OPAMP1

As simulações iniciais demonstraram que a detecção de 69 das 89 falhas consideradas é possível apenas pela observação da amplitude da tensão no nó de saída (cobertura de falhas igual a 77.53%). Destaca-se que 68 dessas falhas podem ser detectadas por um único teste AC realizado na frequência de 5,248 MHz. Uma falha adicional é detectável por meio de um segundo teste AC na frequência de 46,77 kHz, formando o Conjunto de Testes I (Tabela 10).

Tabela 10 – Conjunto de testes I do OPAMP1 considerando apenas medição da tensão no nó de saída.

Índice do Teste	Tipo do Teste	Sinal de Entrada	Medição	Falhas Detectadas	Falhas Acumuladas	Cobertura de Falhas Cumulativa
1	AC Módulo	5,248MHz	v(out)	68	68	76,40%
2	AC Módulo	46,77KHz	v(out)	68	69	77,53%

Fonte: Elaborado pelo autor.

Sem precisar utilizar técnicas de *Design for Test* para acessar nós internos, é possível adicionar a medição das correntes de alimentação, de entrada, do pino pd e do pino xpd ao escopo dos testes. Com isso, a cobertura total de falhas ascende para 91,01% (81/89), aumentando o número de testes para 4. Neste cenário, a ferramenta de otimização selecionou um novo conjunto de testes (Tabela 11), o Conjunto de Testes II, que utiliza apenas medições de corrente, dispensando a medição da tensão de saída.

Tabela 11 – Conjunto de testes II OPAMP1 considerando a possibilidade de medição da tensão no nó de saída, corrente de alimentação, corrente de entrada e corrente nos pinos pd e xpd.

Índice do Teste	Tipo do Teste	Sinal de Entrada	Medição	Falhas Detectadas	Falhas Acumuladas	Cobertura de Falhas Cumulativa
1	AC Módulo	1,778MHz	i(vdd)	76	76	85,39%
2	AC Fase	2,188MHz	i(pd)	68	79	88,76%
3	DC	0,700V	i(vdd)	45	80	89,89%
4	DC	0,000V	i(xpd)	4	81	91,01%

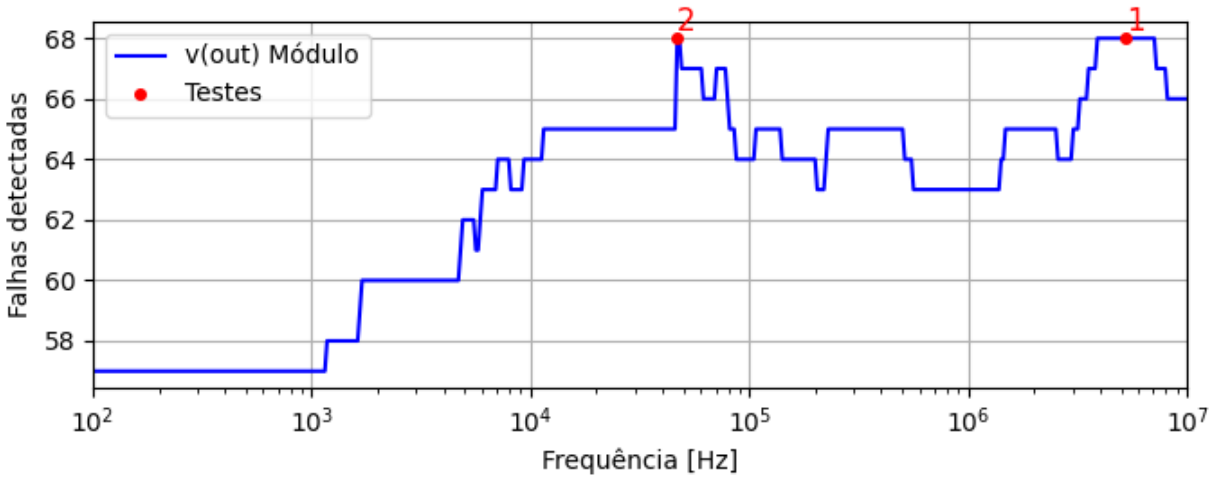
Fonte: Elaborado pelo autor.

O processo de seleção de testes da ferramenta é ilustrado nas Figuras Figura 45, Figura 46 e Figura 47, fornecendo o contexto para a escolha dos pontos de teste. A Figura 45 mostra o número de falhas detectadas em função da frequência, considerando a amplitude de tensão no nó de saída. Os pontos de teste do Conjunto I (Tabela 10) estão marcados, indicando que a ferramenta selecionou ambos os testes em picos de máxima detecção para a curva de tensão de saída. Analogamente, a Figura 46 apresenta as curvas de detecção de falhas para as medições de corrente de alimentação e corrente no pino de entrada. Observa-se que a medição da amplitude da corrente de alimentação apresenta um pico de detecção de falhas próximo a 2 MHz. O algoritmo selecionou o primeiro teste do Conjunto II neste pico.

As escolhas do terceiro e quarto teste do Conjunto II são detalhadas na Figura 47, que exibe as detecções de falha para testes do tipo DC. Percebe-se que, para o teste DC, a medição de corrente do pino xpd apresentou uma capacidade de detecção de falhas individualmente mais

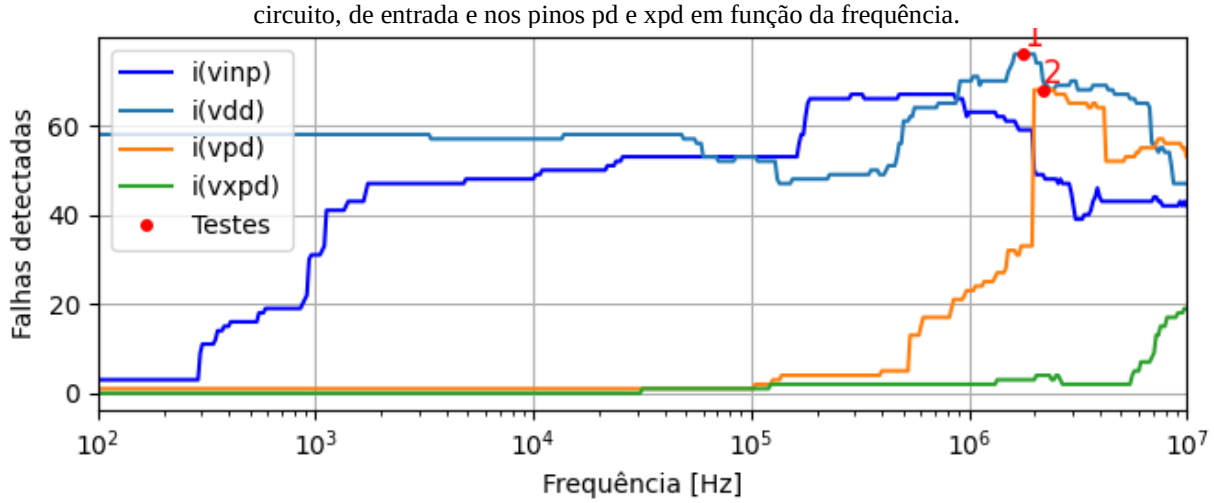
baixa do que a medição da corrente de alimentação. No entanto, a escolha do quarto teste de corrente se baseia na sua capacidade complementar de detectar uma falha adicional que não havia sido coberta pelos outros testes deste conjunto (Tabela 11).

Figura 45 – Número de falhas detectáveis através da medição da amplitude de tensão no nó de saída em função da frequência.



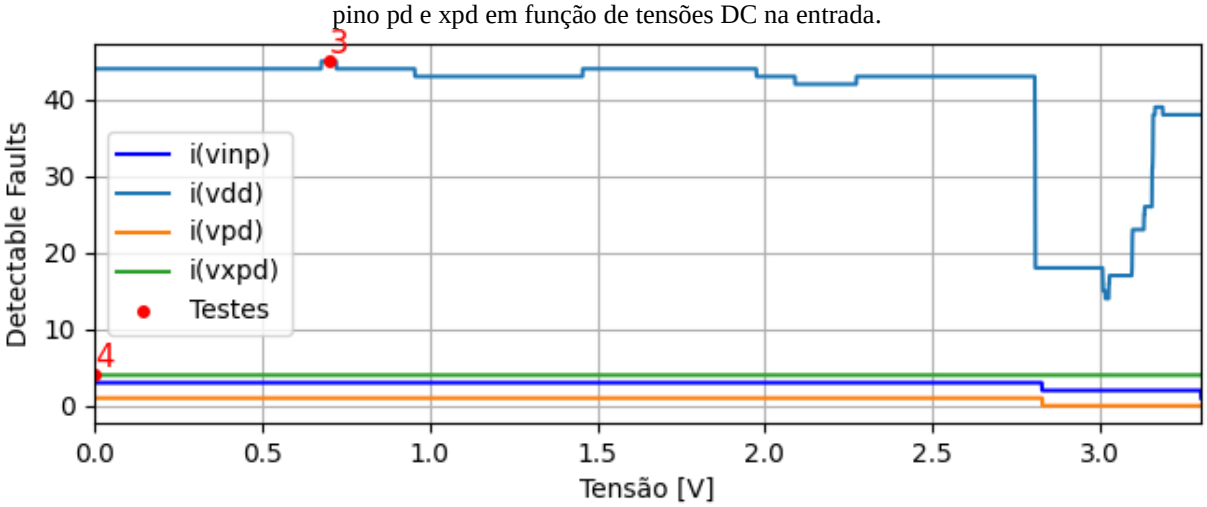
Fonte: elaborado pelo autor.

Figura 46 – Número de falhas detectáveis através da medição das amplitudes das correntes de alimentação do circuito, de entrada e nos pinos pd e xpd em função da frequência.



Fonte: elaborado pelo autor.

Figura 47 – Número de falhas detectáveis através da medição das correntes de alimentação do circuito, entrada, pino pd e xpd em função de tensões DC na entrada.

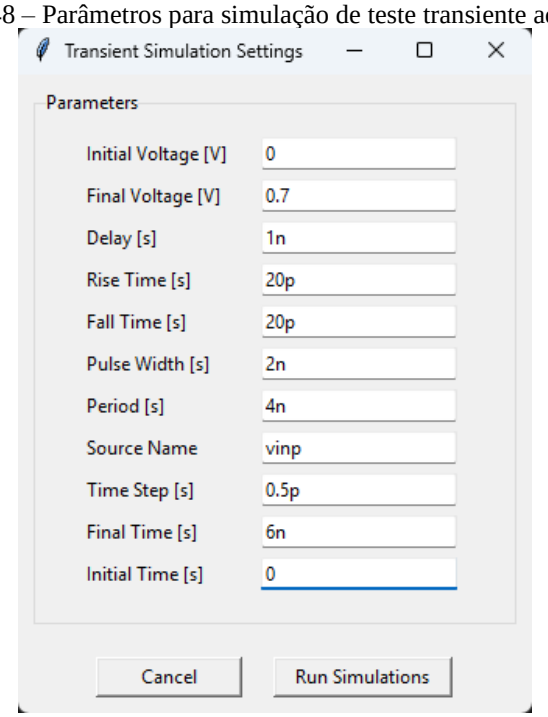


Fonte: elaborado pelo autor.

Vale ressaltar que, tanto no Conjunto de Testes I (Tabela 10) quanto no Conjunto de Testes II (Tabela 11), a ferramenta não selecionou testes AC complementares (medição da outra grandeza: módulo ou fase na mesma frequência), conforme discutido na Seção 3.2.3. Embora o algoritmo tenha considerado essa possibilidade durante o processo de determinação dos conjuntos de testes, a inclusão desses testes complementares não demonstrou qualquer vantagem prática em termos de aumento da cobertura de falhas ou substituição de um dos testes do conjunto final.

Na Tabela 11, observa-se que o algoritmo selecionou dois testes DC no conjunto. Essa característica permite analisar a possibilidade de explorar a transição entre esses dois pontos de operação DC como estímulo adicional para a execução de um teste transiente, conforme discutido previamente na Seção 3.2.3. Com base nisso, foram realizadas simulações transitórias considerando o degrau de tensão aplicado à entrada, variando de 0 V para 0,7 V, bem como a transição inversa. Os demais parâmetros empregados nessas simulações encontram-se apresentados na Figura 48, que ilustra inclusive a interface da ferramenta na funcionalidade destinada à avaliação de novos testes transitórios gerados a partir de dois testes DC previamente selecionados no conjunto otimizado.

Figura 48 – Parâmetros para simulação de teste transiente adicional.



Fonte: elaborado pelo autor.

Os resultados obtidos demonstraram que a medição do *overshoot* da corrente no nó xpd durante a aplicação do degrau direto permite detectar três falhas adicionais, elevando a cobertura total para 94,38%, sem aumento no tempo final de execução do conjunto de testes. Isso ocorre porque, para a execução dos dois testes DC originais, o degrau de tensão entre 0 V e 0,7 V já precisaria ser aplicado de qualquer maneira, de modo que a análise transitória associada não introduz sobrecarga temporal ao processo de teste.

Além disso, ao aplicar o degrau inverso (0,7 V para 0 V) e monitorar a tensão no nó pd, verificou-se a detecção de outras duas falhas adicionais. Assim, a cobertura de falhas alcança 96,63%. O conjunto de testes final considerando esses testes transientes adicionais é mostrado na Tabela 12.

Tabela 12 – Conjunto de testes II OPAMP1 considerando a execução de testes transientes adicionais.

Índice do Teste	Tipo do Teste	Sinal de Entrada	Medição	Falhas Detectadas	Falhas Acumuladas	Cobertura de Falhas Cumulativa
1	AC Módulo	1,778MHz	i(vdd)	76	76	85,39%
2	AC Fase	2,188MHz	i(pd)	68	79	88,76%
3	DC	0,700V	i(vdd)	45	80	89,89%
4	DC	0,000V	i(xpd)	4	81	91,01%
5	TRAN	Overshoot 1	i(xpd)	52	84	94,38%
6	TRAN	Overshoot 2	i(pd)	42	86	96,63%

Fonte: Elaborado pelo autor.

5.5.2 VREG1

Para o caso do regulador de tensão, as simulações demostraram que 100 das 103 falhas consideradas podem ser detectadas observando apenas a tensão no nó de saída do circuito, o que representa uma cobertura de falhas de 97,09%. Entretanto, tal cobertura de falhas é alcançada através da execução de 5 testes, todos AC. Esse conjunto de teste é mostrado na Tabela 13 onde pode-se observar que os testes com índice 1 e 2 são testes complementares, pares Módulo e Fase que requerem a medição do mesmo nó com o circuito submetido ao mesmo estímulo de entrada, porém medindo o módulo do sinal no teste com índice 1 e a fase no teste com índice 2, o que poderia ser feito simultaneamente, tornando o tempo de execução desses testes praticamente o mesmo tempo de execução de um único teste. O mesmo acontece com os testes com índice 3 e 4.

Tabela 13 – Melhor conjunto de testes OPAMP1 considerando apenas medição da tensão no nó de saída.

Índice do Teste	Tipo do Teste	Sinal de Entrada	Medição	Falhas Detectadas	Falhas Acumuladas	Cobertura de Falhas Cumulativa
1	AC Fase	1,288MHz	Vout	80	80	77,67%
2	AC Módulo	1,288MHz	Vout	77	91	88,35%
3	AC Módulo	4,786MHz	Vout	82	98	95,15%
4	AC Fase	4,786MHz	Vout	55	99	96,12%
5	AC Módulo	3,631MHz	Vout	78	100	97,09%

Fonte: Elaborado pelo autor.

Novamente, analisando possibilidades de incrementar a cobertura de falhas, ou diminuir o número de testes do conjunto, foram consideradas as medições de grandezas que estão acessíveis à medição durante a execução dos testes sem a necessidade de estruturas de DfT para acessar nós internos. Nesse contexto, foram adicionados a possibilidade de medir a corrente de

alimentação e a corrente no nó Vref que representa o pino onde o sinal de entrada é inserido. Com isso, a cobertura de falhas pode ser incrementada para 100%, representando a detecção de todas as 103 falhas com apenas 3 testes, sendo eles descritos na Tabela 14. Vale a pena ressaltar, que nesse caso, a inclusão de testes AC complementares não foi benéfica para a redução do tempo de teste, sendo assim, a ferramenta escolheu um conjunto de testes que necessita de três frequências diferentes como sinais de entrada.

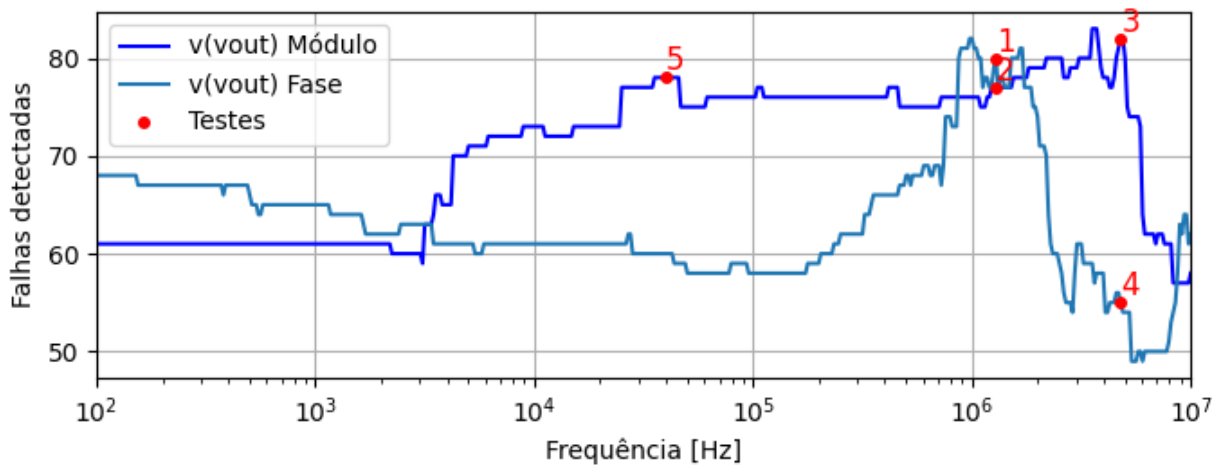
Tabela 14 – Melhor conjunto de testes OPAMP1 considerando apenas medição da tensão no nó de saída e correntes de alimentação e de entrada.

Índice do Teste	Tipo do Teste	Sinal de Entrada	Medição	Falhas Detectadas	Falhas Acumuladas	Cobertura de Falhas Cumulativa
1	AC Módulo	3,388MHz	i(Vref)	85	85	82,52%
2	AC Módulo	4,786MHz	i(Vdd)	82	98	95,15%
3	AC Módulo	812,8KHz	i(Vref)	91	103	100,00%

Fonte: Elaborado pelo autor.

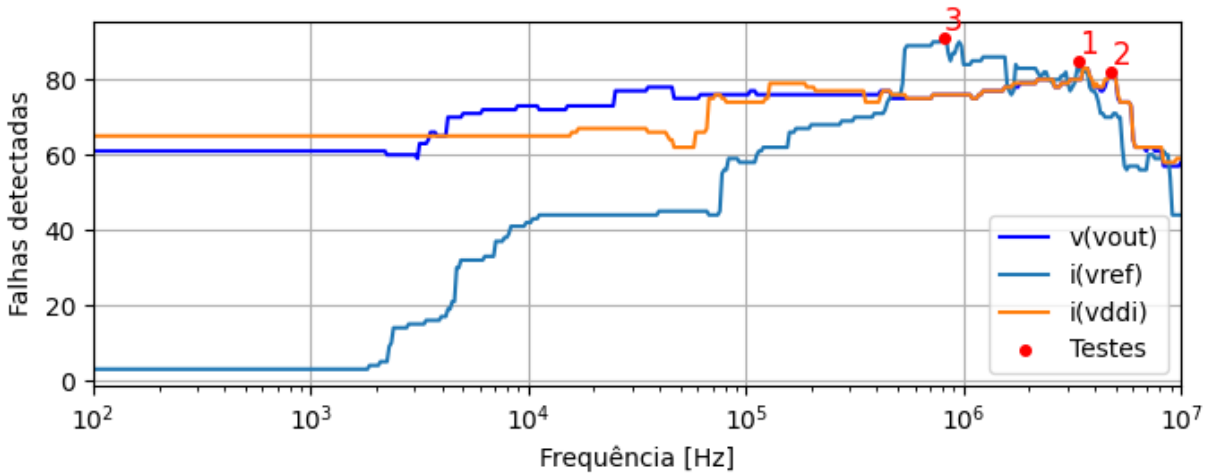
Na Figura 49, é mostrado o número de falhas detectadas em cada frequência considerando a medição da amplitude e fase da tensão no nó de saída, juntamente com a marcação dos testes presentes na Tabela 13. Enquanto que na Figura 50 é mostrado um gráfico análogo, porém mostrando os testes da Tabela 14 juntamente com o número de falhas detectadas com a medição da amplitude dos sinais: tensão de saída do circuito, corrente elétrica de alimentação do circuito e corrente elétrica do pino de tensão de referência. É possível perceber que a possibilidade de medir corrente elétrica nos testes incrementa significativamente os resultados dos testes, resultando em uma cobertura de falhas maior e com menos medições se comparado aos resultados considerando apenas a medição da tensão de saída.

Figura 49 – Número de falhas detectáveis através da medição da amplitude de tensão no nó de saída em função da frequência.



Fonte: elaborado pelo autor.

Figura 50 – Número de falhas detectáveis através da medição da amplitude da corrente de alimentação do circuito em função da frequência.



Fonte: elaborado pelo autor.

5.5.3 COMP1

No caso do comparador, verificou-se que é possível obter uma cobertura de falhas de 72,36% (89/123) monitorando-se exclusivamente a tensão de saída do circuito. Nessa configuração, o conjunto ótimo de testes inclui três simulações transitórias, conforme apresentado na Tabela 15. Considerando apenas a medição de um único nó — o nó de saída Vout — há, em princípio, quatro testes transitórios distintos que poderiam ser realizados: Overshoot 1 e *Settling Time* 1, correspondentes à aplicação do primeiro degrau de entrada (variação de 2,1 V para 2,3 V), e Overshoot 2 e *Settling Time* 2, associados ao degrau inverso (variação de 2,3 V para 2,1 V).

Entretanto, a análise dos resultados revelou que um desses testes não produziu detecções adicionais de falhas quando comparado aos demais, caracterizando-se, portanto, como redundante. Por essa razão, ele não foi adicionado ao conjunto final selecionado pelo algoritmo. Assim, o processo de otimização confirmou que três testes são suficientes para alcançar a cobertura mencionada, evitando a inclusão de medições que não contribuiriam para o aumento da detecção de defeitos e que apenas ampliariam o tempo e o custo total de execução do conjunto de testes.

Tabela 15 – Melhor conjunto de testes COMP1 considerando apenas medição da tensão no nó de saída.

Índice do Teste	Tipo do Teste	Sinal de Entrada	Medição	Falhas Detectadas	Falhas Acumuladas	Cobertura de Falhas Cumulativa
1	TRAN	Set. Time 1	Vout	82	82	66,67%
2	TRAN	Overshoot 1	Vout	21	86	69,92%
3	TRAN	Set. Time 2	Vout	81	89	72,36%

Fonte: Elaborado pelo autor.

De maneira análoga ao procedimento adotado para os demais circuitos pertencentes ao conjunto P2427, foi analisada a viabilidade de utilizar medições de outras grandezas internas do comparador que não requerem a implementação de técnicas adicionais de DfT. Em outras palavras, consideraram-se apenas sinais que já são acessíveis durante a execução dos testes convencionais, sem a necessidade de estruturas específicas de acesso. Nesse contexto, foram incluídas como potenciais observáveis as correntes de alimentação do circuito, as correntes nos nós de entrada positiva e negativa, bem como as correntes associadas aos pinos pd e xpd.

A incorporação dessas grandezas ampliou significativamente o conjunto de possibilidades avaliadas pelo algoritmo de seleção de testes. Como resultado, a ferramenta foi capaz de identificar um conjunto, composto por apenas três testes, capaz de alcançar uma cobertura de falhas de 94,31%, conforme apresentado na Tabela 16. Esse desempenho evidencia que a exploração de medições de corrente pode contribuir de maneira substancial para o aumento da capacidade de detecção de falhas, mantendo ao mesmo tempo a eficiência temporal do conjunto de testes.

Tabela 16 – Melhor conjunto de testes COMP1 considerando medição da tensão no nó de saída e corrente nos pinos de entrada, pd, xpd, vref e vdd.

Índice do Teste	Tipo do Teste	Sinal de Entrada	Medição	Falhas Detectadas	Falhas Acumuladas	Cobertura de Falhas Cumulativa
1	TRAN	Overshoot 2	i(pd)	88	88	71,54%
2	TRAN	Overshoot 1	i(inp)	53	112	91,06%
3	TRAN	Overshoot 2	i(vdd)	91	116	94,31%

Fonte: Elaborado pelo autor.

No caso do comparador, cujo estudo foi conduzido exclusivamente com testes transientes, realizou-se uma nova análise dos resultados com o objetivo de avaliar a eficácia do algoritmo de seleção de testes implementado na ferramenta. Considerando que o conjunto ótimo determinado automaticamente — apresentado na Tabela 16 — é composto por três testes e alcança uma cobertura de falhas de 94,31%, foi desenvolvido um algoritmo auxiliar, independente da ferramenta principal, para realizar uma verificação exaustiva da solução encontrada.

Esse algoritmo adicional foi projetado para ler a matriz de falhas, utilizar exatamente os mesmos nós de observação empregados pela ferramenta e testar sistematicamente todas as combinações possíveis de testes. Inicialmente, foram avaliadas todas as combinações de dois testes, com o objetivo de verificar se a cobertura de 94,31% poderia ser atingida com um conjunto ainda menor. Os resultados mostraram que nenhuma combinação de dois testes foi capaz de alcançar a mesma cobertura obtida pela ferramenta.

Em seguida, procedeu-se à avaliação exaustiva de todas as combinações possíveis de três testes, de modo a investigar se alguma delas seria capaz de superar a cobertura encontrada pelo algoritmo de otimização. Também nesse caso não foi identificada nenhuma combinação que resultasse em um número maior de falhas detectadas. Por fim, foram examinadas todas as combinações de quatro testes, com o intuito de verificar se a adição de um teste extra poderia resultar na detecção de mais de 116 falhas. Nenhuma das combinações avaliadas atingiu desempenho superior ao conjunto selecionado automaticamente pela ferramenta, nem em termos de cobertura total nem em termos do número mínimo de testes necessários.

Além de confirmar a eficiência da solução obtida pelo algoritmo de otimização, essa análise exaustiva demonstrou um aspecto adicional: conforme ilustrado pelos dados da Tabela 17, o tempo de simulação necessário para avaliar todas as combinações possíveis foi significativamente superior ao tempo demandado pela ferramenta para identificar o conjunto ótimo (em torno de 1,5s segundo Tabela 3). Isso evidencia não apenas a qualidade da solução

encontrada, mas também a eficiência computacional do método de seleção, que evita um processo combinatório extremamente custoso sem comprometer a qualidade do conjunto final de testes.

Tabela 17 – Resultados das avaliações exaustivas de todas as possíveis combinações de 2, 3 e 4 testes.

Número de Testes	Maior Detecção de Falhas	Tempo de Simulação
2	112	0,5s
3	116	14,2s
4	116	5min e 58s

Fonte: Elaborado pelo autor.

6 CONCLUSÕES

Esta tese apresentou o desenvolvimento, aprimoramento e validação de uma ferramenta computacional destinada à geração automática e otimização de conjuntos de testes estruturais para detecção de falhas em circuitos analógicos. O trabalho envolveu contribuições teóricas, metodológicas e experimentais distribuídas ao longo de quatro eixos principais: (i) reformulação completa da ferramenta, anteriormente implementada em MATLAB, para uma nova versão em Python; (ii) ampliação das capacidades analíticas por meio da inclusão de novas grandezas observáveis, novos tipos de testes e mecanismos avançados de combinação; (iii) validação do algoritmo por meio de estudos de caso com simulações SPICE, abordagens funcionais, estratégias de DfT e experimentos físicos de injeção de falhas; e (iv) aplicação da metodologia aos circuitos analógicos e mistos do conjunto de *benchmarks* IEEE P2427.

Como a descrição SPICE do circuito sob teste é a principal entrada da ferramenta, o fluxo geral de operação torna-se intuitivo para projetistas de circuitos analógicos e de sinais mistos. Simultaneamente, a ferramenta apresenta um nível de generalidade suficiente para ser aplicada a uma ampla variedade de blocos analógicos, sem exigir conhecimento prévio aprofundado sobre a funcionalidade e as especificações do CI, configurando-se também como uma solução amigável para engenheiros de teste. As capacidades analíticas da ferramenta a tornam uma opção versátil para o desenvolvimento de testes estruturais. Além disso, ela pode auxiliar nas decisões de projeto, ao identificar falhas de difícil detecção e fornecer subsídios para a implementação de técnicas de *Design-for-Testability*, visando o aumento da cobertura de falhas.

O algoritmo proposto adota critérios distintos para determinar, como resultado final, uma lista de estímulos a serem aplicados ao circuito, bem como os nós de observação que, em conjunto, proporcionam uma cobertura de falhas otimizada com um bom compromisso em relação ao tempo de teste.

Inicialmente, demonstrou-se que a reestruturação da ferramenta em Python permitiu ganhos significativos em modularidade, desempenho e escalabilidade. Novos recursos foram incorporados, como a medição de correntes em testes AC, DC e transientes, o suporte simultâneo às grandezas módulo e fase, a criação automática de testes transientes derivados de pares de testes DC e a análise sistemática de combinações de testes complementares (AC módulo/fase e transientes *overshoot*/tempo de acomodação). O algoritmo de seleção de testes também foi reformulado, reduzindo substancialmente o tempo de execução sem perda de exatidão.

Os estudos de caso apresentados nos capítulos subsequentes demonstraram, de forma quantitativa, a eficiência do método. No filtro Butterworth de segunda ordem, foi alcançada cobertura de falhas de 96,87% com apenas cinco testes, demonstrando a eficiência da abordagem estrutural frente a testes funcionais, que atingiram apenas 78,13%. No filtro de dois estágios, verificou-se que a inserção de estruturas de DfT permitiu elevar a cobertura de 87,15% para 95,83%, evidenciando o papel crítico de observabilidade e acessibilidade de nós internos.

A validação experimental realizada com o circuito baseado no CD4007 representou um marco importante na comprovação da robustez da ferramenta. As medições físicas de 48 falhas em hardware resultaram na detecção de 41 falhas, número bastante próximo das 43 previstas via simulação, demonstrando forte equivalência entre o comportamento real e simulado e reforçando a confiabilidade do modelo de falhas e dos critérios de detecção.

A aplicação da ferramenta aos circuitos do grupo IEEE P2427 permitiu avaliar sua eficácia em topologias padronizadas amplamente utilizadas pela comunidade científica. Os resultados mostraram aumentos expressivos de cobertura de falhas:

- OPAMP1: de 77,53% (somente tensão) para 96,63% com uso de correntes e testes transientes;
- VREG1: de 97,09% (tensão) para 100% com apenas três testes incluindo correntes;
- COMP1: de 72,36% (tensão) para 94,31% utilizando apenas testes transientes baseados em corrente.

Além disso, verificações combinatórias exaustivas confirmaram matematicamente que os conjuntos selecionados automaticamente pelo algoritmo são ótimos, já que nenhuma combinação de até quatro testes superou os resultados alcançados automaticamente.

De forma geral, os resultados obtidos comprovam a evolução substancial da nova versão da ferramenta, que se mostrou mais flexível, mais precisa e consideravelmente mais eficiente que a versão original. A inclusão de novas grandezas observáveis, novos tipos de testes e critérios de combinação ampliou o espaço de busca e possibilitou melhorar significativamente a cobertura de falhas com um número reduzido de testes. A validação experimental demonstrou que a metodologia não apenas funciona em ambiente de simulação, mas também se traduz em resultados confiáveis quando aplicada a circuitos físicos.

Por fim, esta tese contribui para o avanço das técnicas de teste estrutural de circuitos analógicos, oferecendo uma ferramenta automatizada, escalável e compatível com cenários reais de teste. Espera-se que o método desenvolvido sirva de base para futuras pesquisas em geração automática de testes, integração com estruturas DfT, modelos de falhas mais refinados,

aplicações em sistemas mistos de maior complexidade e até mesmo em metodologias híbridas envolvendo aprendizado de máquina e otimização inteligente.

7 SUGESTÕES PARA TRABALHOS FUTUROS

Ao término deste trabalho, foram identificadas oportunidades de continuidade e expansão da ferramenta proposta. Nesse contexto, são listadas algumas dessas possibilidades de aprimoramentos e expansões que podem ser utilizadas em trabalhos futuros:

- 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;
- Analisar a variabilidade dos circuitos com falha, para tornar a detecção mais realista.
- 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

AGRAWAL, V. D.; SETH, S. C.; AGRAWAL, P. Fault Coverage Requirement In Production Testing Of LSI Circuits. **IEEE Journal of Solid-State Circuits**, New York, v. 17, n. 1, p. 57–63, 1982. DOI: 10.1109/JSSC.1982.1051776.

ALZAHER, 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.

ARABI, K.; KAMINSKA, B. Oscillation built-in self test (OBIST) scheme for functional and structural testing of analog and mixed-signal integrated circuits. In: **INTERNATIONAL TEST CONFERENCE**, 1997, Washington, DC. Proceedings [...]. Washington, DC: IEEE, 1997. p. 786–795. Disponível em: https://doi.org/10.1109/TEST.1997.639692. Acesso em: 4 jun. 2025.

ARAMOON, O.; QU, G.; CUI, A. Building Hardware Security Primitives Using Scan-Based Design-For-Testability. In: **IEEE International Midwest Symposium On Circuits And Systems (MWSCAS),** 65., 2022, Fukuoka, Japão. Proceedings… New York: IEEE, 2022. p. 1–6. DOI: 10.1109/MWSCAS54063.2022.9859460.

ARSLAN, Baris; ORAILOGLU, Alex. **Tracing the Best Test Mix through Multi-Variate Quality Tracking.** Berkeley: Ieee 31st Vlsi Test Symposium (vts), 2013. 6 p.

AZAM, S.; DALLO'ORA, N.; FRACCAROLI, E.; GILLON R.; FUMMI, F.. Analog Defect Injection And Fault Simulation Techniques: A Systematic Literature Review. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, New York, v. 43, n. 5, p. 1345–1362, 2024. DOI: 10.1109/TCAD.2024.3361234.

BAKER, R. Jacob. CMOS: Circuit Design, Layout, and Simulation. 4. ed. Hoboken: Wiley-IEEE Press, 2019.

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 Microarchtecture. **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 Oorder 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.

CHAUDHURI, J.; BHATTACHARYA, M.; CHAKRABARTY, K. DAWN: Efficient Trojan Detection in Analog Circuits Using Circuit Watermarking and Neural Twins. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, New York, v. 43, n. 10, p. 2930–2943, out. 2024. DOI: 10.1109/TCAD.2024.3384948.

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, A. L.. **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.

CORTES, F. P.; GIRARDI, A.; RÓDIO, F.; FABRIS, E.; BAMPI, S.. Monte Carlo and Corner Simulations for Analog VLSI Design. Instituto de Informática – Universidade Federal do Rio Grande do Sul. Porto Alegre, RS, Brasil, 2003.

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.

DEB, S.; KRISHNA, S.; PATTIPATI, K. R.; RAGHAVAN, V.; SHAKERI, M. SHRESTHA, R. Multi-Signal Flow Graphs: a Novel Approach for System Testability Analysis and Fault Diagnosis. **IEEE Aerospace and Electronic Systems Magazine**, New York, v. 10, n. 5, p. 14–25, maio 1995.

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.

DRI, E.; PERETTI, G.; ROMERO, E. A Built-In Self-Test for Analog Reconfigurable Filters Implemented in a Mixed-Signal Configurable Processor. **Analog Integrated Circuits and Signal Processing**, Cham, v. 112, n. 2, p. 355–365, 2022. DOI: 10.1007/s10470-022-02077-7.

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

FULLER, L. Reverse engineering to obtain CD4007 parameters. Dr. Lynn Fuller webpage, Microelectronic Engineering at Rochester Institute of Technology (RIT), 2020.

GANJI, M.; SAIKIRAN, M.; BHATHEJA, K.; CHEN, D. Digital-Like Built-In Defect-Oriented Test For Analog-Mixed Signal Circuits. **Design+**, v. 1, n. 1, p. 1–14, 2024. DOI: 10.36922/dp.4351.

HARRIS, C. R.; MILLMAN, K. J.; VAN DER WALT, S. J.; et al. Array programming with NumPy. *Nature*, v. 585, p. 357-362, 2020. DOI: 10.1038/s41586-020-2649-2.

HOROWITZ, Paul; HILL, Winfield. The Art of Electronics. 3. ed. Cambridge: Cambridge University Press, 2015.

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.

IEEE. Chapter 17 – Test Technology. In: **IEEE Heterogeneous Integration Roadmap (HIR)**. New York: IEEE Press, 2023. Disponível em: https://eps.ieee.org/images/files/HIR_2023/ch17_test.pdf . Acesso em: 2 fev. 2026.

IEEE STANDARDS ASSOCIATION. IEEE P2427 – Project Authorization Request: Standard for Analog Defect Modeling and Coverage for Analog and Mixed Signal Circuits. New York: IEEE Standards Association, 2018. Disponível em: https://development.standards.ieee.org/myproject-web/public/view.html#pardetail/6359. Acesso em: 02 fev. 2026.

IEEE STANDARDS ASSOCIATION. IEEE P2427 Working Group – Analogue and Mixed-Signal Benchmark Circuits. Disponível em: https://sagroups.ieee.org/2427/. Acesso em: 12 nov. 2025.

IEEE STANDARDS ASSOCIATION. Analogue Benchmark Circuits. Disponível em: https://sagroups.ieee.org/2427/analogue-benchmark-circuits/. Acesso em: 12 nov. 2025.

INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. **IEEE Std 1149.1-2013**: IEEE standard for test access port and boundary-scan architecture. New York: IEEE, 2013.

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.

KARTHI, S. P.; KAVITHA, R. Detecting and classifying parametric faults in analog circuits. **Analog Integrated Circuits and Signal Processing**, Dordrecht, v. 120, n. 3, p. 5401–5415, 2024. DOI: 10.1007/s00034-024-02722-1.

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.

MERRETT, M.; ASENOV, P.; WANG, Y.; ZWOLINSKI, M.; REID, D.; MILLAR, C.; ROY, S.; LIU, Z.; FURBER, S.; ASENOV, A. Modelling Circuit Performance Variations Due To Statistical Variability: Monte Carlo Static Timing Analysis. In: **Design, Automation & Test In Europe Conference (DATE)**, 2011.

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

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.

NAGARAJ, V. Design-for-Test (DFT) strategies for high-performance computing and graphics chips. **International Journal Of Signal Processing, Embedded Systems And Vlsi Design** Volume 05, Issue 01, 2025, pages 10-34. 2025.

OLIPHANT, T. E. "Python for Scientific Computing," in **Computing in Science & Engineering**, vol. 9, no. 3, pp. 10-20, May-June 2007, doi: 10.1109/MCSE.2007.58.

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.cppsim.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.

SEDRA, Adel S.; SMITH, Kenneth C. Microelectronic Circuits. 6ª ed. Oxford University Press, New York, Oxford, 2010.

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

SHASHIDHARA, H. B.; YELLAMPALII, S.; GOUDANAVAR, V. "Board level JTAG/boundary scan test solution," *International Conference on Circuits, Communication, Control and Computing*, Bangalore, India, 2014, pp. 73-76, doi: 10.1109/CIMCA.2014.7057760.

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.

SUNTER, S.; SARSON, P. A/MS benchmark circuits for comparing fault simulation, DFT, and test generation methods. In: **IEEE INTERNATIONAL TEST CONFERENCE (ITC)**, Fort Worth, TX. Proceedings [...]. Piscataway: IEEE, 2017.

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.

TOUBA, N. A. Survey of Test Vector Compression Techniques. **IEEE Design & Test of Computers**, New York, v. 23, n. 4, p. 294–303, July–Aug. 2006.

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.

VARIYAM, P. N.; CHERUBAL, S.; CHATTERJEE, A. Prediction of Analog Performance Parameters Using Fast Transient Testing. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, v. 21, n. 3, p. 349–361, mar. 2002

WANG, L. T; TIAN, H.; ZHANG, H. Soft fault diagnosis of analog circuits based on semi-supervised support vector machine. **Analog Integr Circ Sig Process** 108, 305–315 (2021). https://doi.org/10.1007/s10470-021-01851-w

WANG, L. T.; WU, C. W.; WEN, X**.** *VLSI Test Principles and Architectures: Design for Testability*. 1. ed. Amsterdam; Boston: Elsevier / Morgan Kaufmann, 2006.

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.

YANG, C.; TIAN, S.; LONG, B. Test Points Selection for Analog Fault Dictionary Techniques. **Journal of Electronic Testing: Theory and Applications**, New York, v. 25, n. 1, p. 3–13, fev. 2009. DOI: 10.1007/s10836-008-5088-9.

YURISH, S. Y. Foreword. In: YURISH, Sergey Y. (ed.). Microelectronic Devices and Technologies: **proceedings of the 7th International Conference on Microelectronic Devices and Technologies** (MicDAT 2025). Ponta Delgada, Portugal: IFSA Publishing, 2025. p. 4–5. ISBN 978-84-09-76917-9.

ZAKIZADEH, J.; DAS, S. R.; ASSAF, M. H.; PETRIU, E. M.; SAHINOGLU, M.; JONE, W. Testing Analog and Mixed-Signal Circuits With Built-In Hardware – A New Approach. **IEEE Transactions on Instrumentation and Measurement**, New York, v. 56, n. 3, p. 166–171, 2005.

ZHOU, L; YANG, D; CHEN, L; ZHUANG, W; ZHANG, S; XIONG, Y. A review of test stimulus compression methods for ultra-large-scale integrated circuits. Applied Sciences, Basel, v. 14, n. 23, p. 1–19, 2024. DOI: 10.3390/app142310769.

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.

ZILCH, Lucas Bernardo. **Geração Automática de Padrões para Teste Estrutural de Circuitos Analógicos.** 85 f. Dissertação (Mestrado em Microeletrônica) – Programa de Pós Graduação em Microeletrônica, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2021.