

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

ANDRÉ LUCAS CHINAZZO

**Desenvolvimento de Teste de Amplificadores
Diferenciais utilizando Simulações SPICE
Automatizadas**

Porto Alegre

2016

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

ANDRÉ LUCAS CHINAZZO

**Desenvolvimento de Teste de Amplificadores Diferenciais
utilizando Simulações SPICE Automatizadas**

Projeto de Diplomação apresentado ao Departamento de Engenharia Elétrica da Escola de Engenharia da Universidade Federal do Rio Grande do Sul, como requisito parcial para Graduação em Engenharia Elétrica

Orientador: Prof. Dr. Tiago Roberto Balen

Porto Alegre

2016

CIP - Catalogação na Publicação

Chinazzo, Andre Lucas
Desenvolvimento de Teste de Amplificadores
Diferenciais utilizando Simulações SPICE
Automatizadas / Andre Lucas Chinazzo. -- 2016.
96 f.

Orientador: Prof. Dr. Tiago Roberto Balen.

Trabalho de conclusão de curso (Graduação) --
Universidade Federal do Rio Grande do Sul, Escola de
Engenharia, Curso de Engenharia Elétrica, Porto
Alegre, BR-RS, 2016.

1. teste analógico. 2. circuitos integrados. 3.
dicionário de falhas. 4. vetores de teste. I. Balen,
Prof. Dr. Tiago Roberto, orient. II. Título.

ANDRÉ LUCAS CHINAZZO

Desenvolvimento de Teste de Amplificadores Diferenciais utilizando Simulações SPICE Automatizadas

Projeto de Diplomação apresentado ao Departamento de Engenharia Elétrica da Escola de Engenharia da Universidade Federal do Rio Grande do Sul, como requisito parcial para Graduação em Engenharia Elétrica

Prof. Dr. Tiago Roberto Balen
Orientador - UFRGS

Prof. Dr. Ály Ferreira Flores Filho
Chefe do Departamento de Engenharia
Elétrica (DELET) - UFRGS

Aprovado em ___ de _____ de 2016.

BANCA EXAMINADORA

Prof. Dr. Tiago Roberto Balen
UFRGS

Prof. Dr. Hamilton Duarte Klimach
UFRGS

**Prof. Msc. Paulo César Comassetto de
Aguirre**
UNIPAMPA

*Aos meus pais, Osmar Francisco Chinazzo e Adelise Ioris Chinazzo;
e a todos aqueles que tomarem interesse neste estudo.*

Agradecimentos

Agradeço aos meus pais o constante incentivo, a confiança, e principalmente o exemplo que sempre foram para mim.

Agradeço à minha namorada, Laís Muntini, a compreensão da minha ausência e o carinho de sempre. Obrigado por me cativar.

Agradeço ao Prof. Tiago Roberto Balen a orientação dada durante a realização deste trabalho, mas principalmente por ser um ótimo professor. Agradeço também ao Prof. Paulo César Comassetto de Aguirre a disposição em sanar minhas dúvidas sobre o trabalho.

Agradeço aos meus amigos que participaram comigo, nesses últimos anos, de histórias que vou lembrar, provavelmente, a minha vida inteira.

Agradeço, enfim, a todos que tornaram a realização deste trabalho possível. Obrigado.

The one who does not remember history is bound to live through it again.

George Santayana

Resumo

Neste trabalho o teste de circuitos integrados analógicos é abordado. É desenvolvida uma ferramenta de automatização de injeção e simulação de falhas, assim como posterior análise dos resultados das simulações. Essa ferramenta é utilizada para criação de dicionários de falhas analógicas. A validação da ferramenta foi realizada pela reprodução idêntica de resultados passados obtidos de forma manual. O tempo de criação do dicionário de falhas completo utilizando a ferramenta é dominado pelo tempo de simulação de todos os circuitos. Propôs-se, então, uma metodologia de seleção otimizada de conjuntos de vetores de teste baseada na resposta em frequência de diversos nós do circuito, assumindo-se que este opera em sua faixa linear. A metodologia consiste em obter a resposta em frequência dos circuitos com falha injetada e compará-las com a resposta dos circuitos nominal e nos *corners* da tecnologia. Três circuitos diferentes foram utilizados como prova de conceito da funcionalidade da metodologia: um FDA de um estágio (*Fully-Differential Amplifier*), projetado por Oliveira, Severo e Girardi (2014); um segundo FDA, projetado por Aguirre (2014), de dois estágios principais e um estágio de compensação; e por último a aplicação do último FDA, um filtro analógico de terceira ordem, projetado também por Aguirre (2014), que faz parte de um modulador sigma-delta. São feitas análises sobre a gama de possibilidades de escolha dos vetores de teste que podem proporcionar inclusive a cobertura completa das falhas.

Palavras-chave: teste analógico, circuitos integrados, dicionário de falhas, vetores de teste.

Abstract

This work addresses the test of analog integrated circuits. An automatization tool for fault injection, circuit simulation and post simulation analysis is developed. The tool is then used in the creation of analog fault dictionaries. The tool was validated by the identical reproduction of past results, which were obtained manually. Using the tool, the creation time of a fault dictionary is dominated by the time taken by the simulation of all circuits. An optimized set of test vectors selection methodology based upon the frequency response of various nodes was proposed, assuming the circuit to work at its linear range. The methodology consists in obtaining the frequency response of faulty circuits and comparing it to the frequency response of the nominal and the technology corners circuits. Three circuits were then used as proof of concept: a one-stage FDA (Fully-Differential Amplifier) designed by Oliveira, Severo e Girardi (2014); a second FDA, designed by Aguirre (2014), using two main stages and one compensation stage; and finally an application of the last FDA, an analog third-order filter, also designed by Aguirre (2014), which is part of a sigma-delta modulator. The possible ways of choosing the test vectors are discussed, and some examples of even full fault coverage are presented.

Keywords: analog test, integrated circuits, fault dictionary, test vectors.

Lista de Figuras

Figura 1 – Estimativa de custo relativo por falha em cada etapa da produção de um sistema eletrônico.	21
Figura 2 – <i>Corners</i> da tecnologia	24
Figura 3 – (a) Amplificador Totalmente Diferencial e (b) Amplificador Operacional Convencional.	25
Figura 4 – Exemplo da Análise de <i>Corners</i>	27
Figura 5 – Exemplo de utilização do algoritmo de injeção de falhas.	29
Figura 6 – Definição dos limites toleráveis pela análise de <i>corners</i>	31
Figura 7 – Exemplo de resposta em frequência de circuito defeituoso fora dos limites toleráveis.	32
Figura 8 – Diagrama esquemático das topologias (A) PMOS-DA e (B) NMOS-DA.	32
Figura 9 – Diagrama esquemático do bloco de amplificação diferencial projetado em tecnologia XFAB 0,18 μm	33
Figura 10 – Diagrama esquemático do bloco de CMFB projetado em tecnologia XFAB 0,18 μm	34
Figura 11 – Diagrama esquemático dos estágios principais e de compensação em avanço do FDA projetados em tecnologia IBM 0,13 μm	37
Figura 12 – Diagrama esquemático do bloco de CMFB projetado em tecnologia IBM 0,13 μm	38
Figura 13 – Diagrama esquemático do bloco de polarização do FDA projetado em tecnologia IBM 0,13 μm	39
Figura 14 – Diagrama de blocos do SDM projetado em tecnologia IBM 0,13 μm	41
Figura 15 – Diagrama esquemático do filtro analógico projetado em tecnologia IBM 0,13 μm	42
Figura 16 – Diagrama esquemático do filtro analógico em malha fechada projetado em tecnologia IBM 0,13 μm	43
Figura 17 – Cobertura de falhas para diferentes nós do FDA projetado em tecnologia XFAB 0,18 μm	47
Figura 18 – Cobertura de falhas para diferentes nós do FDA projetado em tecnologia IBM 0,13 μm	48
Figura 19 – Resposta ao degrau dos nós de interesse do FDA projetado em tecnologia IBM 0,13 μm	49
Figura 20 – Resultado da aplicação do conjunto de testes proposto para o FDA projetado em tecnologia IBM 0,13 μm	51
Figura 21 – Cobertura de falhas para diferentes nós do filtro em malha aberta do SDM projetado em tecnologia IBM 0,13 μm	52

Figura 22 – Resposta ao degrau da tensão de saída do filtro em malha aberta do SDM projetado em tecnologia IBM 0,13 μm	53
Figura 23 – Resposta ao degrau da tensão de saída do filtro em malha fechada do SDM projetado em tecnologia IBM 0,13 μm	53
Figura 24 – Cobertura de falhas para diferentes nós do filtro em malha fechada do SDM projetado em tecnologia IBM 0,13 μm	54
Figura 25 – Resposta em frequência da tensão de saída do filtro em malha fechada do SDM projetado em tecnologia IBM 0,13 μm	94
Figura 26 – Resposta ao degrau dos nós de interesse dos FDAs 1 e 2 do filtro em malha fechada do SDM projetado em tecnologia IBM 0,13 μm	95
Figura 27 – Resposta ao degrau dos nós de interesse dos FDAs 3 e 4 do filtro em malha fechada do SDM projetado em tecnologia IBM 0,13 μm	96

Lista de Tabelas

Tabela 1 – Dimensões e valores de projeto das topologias PMOS-DA e NMOS-DA.	33
Tabela 2 – Dimensões e valores de projeto do FDA projetado em tecnologia XFAB 0,18 μm .	34
Tabela 3 – Configurações de testes aplicadas ao FDA projetado em tecnologia XFAB 0,18 μm .	35
Tabela 4 – Dimensões dos transistores do FDA projetado em tecnologia IBM 0,13 μm .	40
Tabela 5 – Valores dos componentes passivos do filtro analógico projetado em tecnologia IBM 0,13 μm .	41
Tabela 6 – Tensões obtidas pela ferramenta de automatização para falhas em M_1 : análise das topologias PMOS-DA e NMOS-DA.	45
Tabela 7 – Cobertura de falhas dos nós principais do FDA projetado em tecnologia XFAB 0,18 μm .	46
Tabela 8 – Conjunto de testes AC para o FDA projetado em tecnologia XFAB 0,18 μm .	47
Tabela 9 – Conjunto de testes AC para o FDA projetado em tecnologia IBM 0,13 μm .	49
Tabela 10 – Conjunto de testes AC completo para o filtro do SDM projetado em tecnologia IBM 0,13 μm .	55
Tabela 11 – Conjunto de testes AC via tensões de saída para o filtro do SDM projetado em tecnologia IBM 0,13 μm .	56
Tabela 12 – Conjunto de testes AC via tensão de saída inversora do filtro do SDM projetado em tecnologia IBM 0,13 μm .	57
Tabela 13 – Conjunto de teste pela análise dos gráficos de CF x Frequência x Nó do filtro do SDM projetado em tecnologia IBM 0,13 μm .	58

Lista de Abreviaturas e Siglas

AC	<i>Alternating Current</i>
ADC	<i>Analog-to-Digital Converter</i>
CF	Cobertura de Falhas
CI	Circuito Integrado
CIFF	<i>Chain of Integrators with Feedforward</i>
CMFB	<i>Common Mode Feedback</i>
CUT	<i>Circuit Under Test</i>
DAC	<i>Digital-to-Analog Converter</i>
DC	<i>Direct Current</i>
FDA	<i>Fully-Differential Amplifier</i>
FII	<i>Fault Injection Instruction</i>
GBW	<i>Gain-Bandwidth Product</i>
MIM	<i>Metal-Insulator-Metal</i>
NMOS	<i>N-channel Metal Oxide Semiconductor</i>
NMOS-DA	NMOS - <i>Diferential Amplifier</i>
OD	<i>Open Drain</i>
OG	<i>Open Gate</i>
OS	<i>Open Source</i>
OPAMP	<i>Operational Amplifier</i>
PMOS	<i>P-channel Metal Oxide Semiconductor</i>
PMOS-DA	PMOS - <i>Diferential Amplifier</i>
SDG	<i>Short Drain-Gate</i>
SDS	<i>Short Drain-Source</i>

SDM	<i>Sigma-Delta Modulator</i>
SGS	<i>Short Gate-Source</i>
SNR	<i>Signal-to-Noise Ratio</i>
SPICE	<i>Simulation Program with Integrated Circuits Emphasis</i>

Sumário

1	INTRODUÇÃO	17
2	REVISÃO BIBLIOGRÁFICA	20
2.1	Teste de Circuitos Analógicos	20
2.1.1	Teste em Produção	20
2.1.2	Teste Funcional e Teste Estrutural	21
2.1.3	Modelos de Falhas Comuns	22
2.1.4	Dicionário de Falhas	23
2.2	Análise de <i>corners</i>	23
2.3	Amplificador Totalmente Diferencial	24
3	METODOLOGIA EXPERIMENTAL	26
3.1	Automatização das Simulações	26
3.1.1	Alteração de <i>Corners</i> e Injeção de Falhas	26
3.1.2	Execução e Verificação das Simulações	28
3.1.3	Construção do Dicionário de Falhas	29
3.2	Validação da Ferramenta de Automatização Desenvolvida	30
3.2.1	Amplificadores Diferenciais de dois Estágios com Saída Simples	30
3.2.2	Amplificador Totalmente Diferencial de um Estágio	31
3.3	Estudo de Caso: Modulador Sigma-Delta	36
3.3.1	Amplificador Totalmente Diferencial de dois Estágios	36
3.3.2	Filtro Analógico do Modulador Sigma-Delta	41
4	RESULTADOS E DISCUSSÕES	44
4.1	Validação da Ferramenta de Automatização Desenvolvida	44
4.1.1	Amplificadores Diferenciais de dois Estágios com Saída Simples	44
4.1.2	Amplificador Totalmente Diferencial de um Estágio	45
4.2	Estudo de Caso: Modulador Sigma-Delta	47
4.2.1	Amplificador Totalmente Diferencial de dois Estágios	48
4.2.2	Filtro Analógico do Modulador Sigma-Delta	50
4.2.2.1	Conjunto de Testes Completo	55
4.2.2.2	Conjunto de Testes via Tensões de Saída na faixa de 1 MHz a 1 GHz	55
4.2.2.3	Conjunto de Testes via Tensão de Saída Inversora na faixa de 1 MHz a 1 GHz	56
4.2.2.4	Conjunto de Testes pela Análise dos Gráficos de CF x Frequência x Nó na faixa de 1 MHz a 1 GHz	57

5	CONCLUSÕES	59
6	PROPOSTAS DE TRABALHOS FUTUROS	61
	REFERÊNCIAS BIBLIOGRÁFICAS	62
	APÊNDICES	65
	APÊNDICE A – ALGORITMO EM MATLAB DE AUTOMATIZAÇÃO DO TESTE DC	66
	APÊNDICE B – ALGORITMO EM MATLAB DE AUTOMATIZAÇÃO DO TESTE AC	68
	APÊNDICE C – FUNÇÃO EM MATLAB PARA ALTERAÇÃO DE <i>CORNER</i>	70
	APÊNDICE D – FUNÇÃO EM MATLAB PARA ENCAMINHAMENTO DE INJEÇÃO DE FALHA	71
	APÊNDICE E – FUNÇÃO EM MATLAB PARA INJEÇÃO DE FALHAS DE CIRCUITO ABERTO	73
	APÊNDICE F – FUNÇÃO EM MATLAB PARA INJEÇÃO DE FALHAS DE CURTO CIRCUITO	76
	APÊNDICE G – FUNÇÃO EM MATLAB PARA INJEÇÃO DE FALHAS PARAMÉTRICAS	79
	APÊNDICE H – FUNÇÃO EM MATLAB PARA CONVERSÃO DE FATORES DE MULTIPLICAÇÃO	82
	APÊNDICE I – FUNÇÃO EM MATLAB PARA EXECUÇÃO E VERIFICAÇÃO DAS SIMULAÇÕES	83
	APÊNDICE J – FUNÇÃO EM MATLAB PARA CRIAÇÃO DO DICIONÁRIO DE FALHAS DC	86
	APÊNDICE K – FUNÇÃO EM MATLAB PARA CRIAÇÃO DO DICIONÁRIO DE FALHAS AC	88
	APÊNDICE L – FUNÇÃO EM MATLAB PARA ESCOLHA OTIMIZADA DE CONJUNTO DE TESTES	90

APÊNDICE M – FUNÇÃO EM MATLAB PARA ESCOLHA OTIMIZADA DE CONFIGURAÇÃO DE TESTE . . .	91
APÊNDICE N – DESCRIÇÃO EM SPICE DO OPAMP PMOS-DA	92
APÊNDICE O – EXEMPLO DE ARQUIVO DE INSTRUÇÕES PARA INJEÇÃO DE FALHAS (FII)	93
APÊNDICE P – GRÁFICOS DA RESPOSTA EM FREQUÊNCIA E RESPOSTA AO DEGRAU DO SDM	94

1 Introdução

O avanço da tecnologia de semicondutores é uma realidade que, muitas vezes, passa despercebida por seus próprios beneficiários. A miniaturização, embora seja uma consequência palpável deste avanço, a exemplo dos computadores que na década de 50 tinham o tamanho de uma sala, e hoje são suficientemente pequenos para serem transportados numa mochila, é um efeito cujos possibilitadores são desconhecidos dos consumidores.

A indústria de semicondutores, assim como qualquer outra, busca a minimização de custos na fabricação de seus produtos. Tal redução de custo é dada, principalmente, pelo aumento da densidade de componentes, importando em aumento da produtividade (*yield*). O custo por transistor de um circuito integrado (CI) digital é reduzido pela metade a cada dois anos, ao passo que os próprios componentes se tornam mais eficazes e velozes, resultando numa duplicação efetiva da performance a cada dezoito meses (MOORE, 1965; MOORE, 1975).

Em contrapartida à redução de custo por componente, o aumento da capacidade de integração torna significativos efeitos antes desprezados, como os efeitos de canal curto e de corrente de fuga (TAUR et al., 1997). Além de modelos mais complexos para descrição do comportamento dos CIs, estes se tornam mais susceptíveis a falhas, tanto relacionadas ao processo de fabricação, quanto à impureza do material base, o silício.

O funcionamento adequado de um CI depende da boa condução de todas as etapas de sua fabricação, desde o projeto até o encapsulamento. Falhas de projeto podem ser oriundas, por exemplo, de simplificações exageradas de modelos. Existe também, principalmente com a redução das dimensões físicas dos transistores, a possibilidade de impurezas no *wafers* de silício causarem alterações no comportamento final do circuito. A própria precisão limitada das técnicas utilizadas na fabricação dos CIs pode inserir, entre outras falhas, curto circuitos, falhas de contatos e componentes parasitas.

Uma vez que o CI fabricado assume um padrão estocástico quanto às falhas, a verificação do seu funcionamento é indispensável. Testes sobre os circuitos são aplicados durante diversas etapas da sua fabricação, pois o custo de reparo é tão menor quanto antes se identificam as falhas (BALEN, 2006). A complexidade dos testes aplicados depende fortemente da natureza e das especificações do circuito. Circuitos digitais exigem testes menos precisos, já que esses são baseados em sinais lógicos, no entanto, por vezes, ainda muito complexos devido ao número de componentes empregado. Circuitos analógicos, que se baseiam em sinais de corrente e tensão elétricas (e suas magnitudes), requerem testes mais específicos e, portanto, mais caros.

O custo de um teste é definido, primeiramente, pelo tempo de execução por circuito (*throughput*) (SOUDERS; STENBAKKEN, 1990), enquanto a cobertura de falhas (CF) pode ser relacionada à confiabilidade do teste. Geralmente, há uma relação direta do custo do teste com sua confiabilidade, sendo testes mais exaustivos, portanto mais caros, também mais confiáveis. O ponto ótimo da relação pode ser extremamente complexo de ser encontrado, e, inclusive, variar ao longo do período de produção (ARSLAN; ORAILOGLU, 2013).

A produção de circuitos analógicos é mais custosa que a de seus análogos digitais (de mesma funcionalidade), tanto pela complexidade do próprio projeto quanto pela execução de testes mais específicos. Por este motivo a implementação digital de um sistema eletrônico é, em geral, preferida. Entretanto, muitas aplicações necessitam (i.e., não existe análogo digital) sistemas analógicos, como é o caso do interfaceamento dos sistemas digitais com o mundo físico. Um dos circuitos amplamente utilizados, tanto na interface digital-analógica quanto em outros blocos, são os amplificadores operacionais (OPAMP - *Operational Amplifier*).

Os amplificadores operacionais se dividem em basicamente duas categorias: os mais convencionais sendo os com saída de tensão simples e os amplificadores totalmente diferenciais. FDAs são utilizados em aplicações de alto desempenho, quando as especificações do sistema em questão não são satisfeitas com uso de OPAMPs convencionais, principalmente quando se deseja aumentar a relação sinal-ruído (SNR - *Signal-to-Noise Ratio*). O funcionamento de um FDA é baseado na realimentação interna feita pelo circuito de Realimentação de Modo Comum (CMFB - *Common Mode Feedback*). Esta realimentação permite desacoplamento da tensão de modo comum do sinal de entrada e do sinal de saída.

Este trabalho foi motivado inicialmente na ideia de que, por ser o bloco de realimentação interna do FDA, o circuito de CMFB pode ser utilizado como bloco verificador do funcionamento adequado do FDA, anteriormente proposta por Bender (2015). Em um primeiro momento, neste trabalho é feita a análise do desempenho do circuito de CMFB como detector de falhas. É obtida, portanto, a cobertura de falhas (CF) de diversos nós do FDA a fim de concluir viável e vantajosa a utilização do circuito de CMFB como verificador.

A proposta consiste em realizar simulações SPICE¹ para identificar as respostas características dos circuitos com e sem falha, de modo a poder discriminar o segundo dos primeiros. A resposta do circuito depende, por óbvio, do sinal de teste aplicado. Como forma de aumentar a cobertura de falhas, são realizados diferentes testes, consistindo de diferentes configurações do FDA e diferentes sinais de teste. Este processo é conhecido

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

como construção do dicionário de falhas. Fica claro que são necessárias muitas simulações, tornando este passo cansativo e muito vulnerável a erros humanos.

Criou-se, então, um algoritmo em MATLAB para automatizar a injeção de falhas, as simulações e a análise dos resultados. Este algoritmo, referido aqui como ferramenta de automatização, possibilita a injeção de falhas catastróficas e paramétricas genéricas e em qualquer componente no circuito. Isto permite ao projetista do circuito dedicar mais tempo para o aperfeiçoamento deste e torna a análise dos resultados, i.e. criação do dicionário de falhas, mais veloz.

Em um segundo momento, propõe-se a metodologia de criação de um dicionário de falhas para testes AC (*Alternating Current*). Este teste é limitado para circuitos analógicos operando como um sistema linear, já que é baseado na resposta em frequência do circuito. Essa metodologia foi motivada pela inferência de que as diferentes configurações de testes utilizadas eram definidas de forma arbitrária, sem se ter o conhecimento das falhas que estas poderiam detectar. Neste trabalho é avaliada a metodologia de testes AC proposta para padronizar a etapa de testes de circuitos analógicos lineares, fazendo-se uso da ferramenta de automatização previamente desenvolvida.

Como etapas de avaliação da metodologia de testes AC, são feitas primeiramente comparações entre este tipo de teste e os testes DC (*Direct Current*) e transiente, aplicados no FDA projetado em Cardoso (2012) e já avaliado em Bender (2015). Em um segundo momento, a metodologia é aplicada no Modulador Sigma-Delta (SDM - *Sigma-Delta Modulator*) projetado em Aguirre (2014). Primeiramente, é analisado o OPAMP, projetado pelo mesmo autor, de forma isolada, obtendo-se a cobertura de falhas de diferentes nós do circuito para diferentes configurações de teste. Discutem-se, então, as modificações necessárias no circuito que implementa o filtro do SDM de forma a viabilizar o teste AC.

Este trabalho é apresentado em seis Capítulos. O Capítulo 2 apresenta uma revisão bibliográfica sobre testes de circuitos analógicos, análise de variabilidade na produção de circuitos integrados e revisão do circuito de um amplificador totalmente diferencial. O desenvolvimento da ferramenta de automatização, assim como as metodologias de teste empregadas neste trabalho são detalhadas no Capítulo 3. Os resultados dos experimentos propostos no Capítulo anterior são apresentados no Capítulo 4, em paralelo com discussões a respeito destes. Finalmente, conclui-se sobre os resultados obtidos no decorrer do trabalho no Capítulo 5. O Capítulo 6 sugere estudos complementares para a continuação deste.

2 Revisão Bibliográfica

2.1 Teste de Circuitos Analógicos

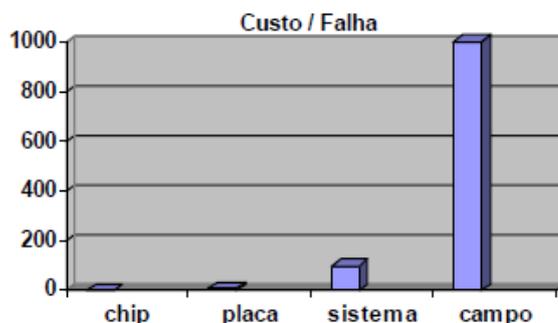
Em comparação com a técnicas de teste de circuitos digitais, o teste analógico teve seu desenvolvimento dado em passos muito menores. Enquanto já há aproximadamente três décadas existem ferramentas automáticas de geração de testes para circuitos digitais, essa tarefa ainda é realizada manualmente para sistemas analógicos e é, normalmente, baseada na experiência do projetista (KABISATPATHY; BARUA; SINHA, 2005; MILOR, 1998). Isso se deve, principalmente, ao fato de que os circuitos digitais são compostos por blocos extremamente simples, que combinados realizam funções complexas, enquanto circuitos analógicos geralmente não podem ser divididos em blocos tão simples. A falta de padronização dos testes analógicos também é devida a natureza ampla de possibilidades de sinais analógicos, mais precisamente, infinita. Ao contrário dos sinais digitais que assumem valores discretos, portanto finitos e fáceis de serem medidos, os sinais analógicos podem assumir infinitos valores, e portanto devem ser medidos com equipamentos precisos.

Sabendo-se que as metodologias de teste de circuitos analógicos diferem drasticamente entre si, definem-se métricas para tornar comparações possíveis. Normalmente, o tempo de teste e a cobertura de falhas do teste são os parâmetros que definem a eficiência do teste. Quanto mais breve for e mais falhas for capaz de detectar, melhor é o teste. No sentido de melhorar a eficiência dos testes de circuitos analógicos, diversas metodologias de teste vêm sendo propostas (MILOR; VISVANATHAN, 1987; MILOR; VISVANATHAN, 1989; SOUDERS; STENBAKKEN, 1991; AUGUSTO; ALMEIDA, 1995; BROSA; FIGUEIRAS, 1999; HALDER; CHATTERJEE, 2004; DENG; SHI; ZHANG, 2012; PETRASHIN et al., 2013).

2.1.1 Teste em Produção

Os testes realizados ainda na etapa de produção do CI visam detectar o quanto antes falhas que possam causar um comportamento inesperado do circuito. Uma das motivações para aplicação de testes antes mesmo do encapsulamento do CI é a teoria de que o custo de uma falha cresce dez vezes em cada etapa de produção pela qual passa despercebida. Conhecida como *rule of ten* (BUSHNELL; AGRAWAL, 2002) apud (DAVIS, 1982), a teoria é bem aceita pela indústria eletrônica, e inclusive propõe-se que passe a ser chamada de *rule of twenty*, já que os sistemas eletrônicos atuais são muito mais complexos do que quando a teoria foi proposta, em 1982. A Figura 1 ilustra esta teoria.

Figura 1 – Estimativa de custo relativo por falha em cada etapa da produção de um sistema eletrônico.



Fonte: (BALEN, 2006, p. 19).

Os equipamentos que realizam os testes na etapa de produção são conhecidos como ATEs (*Automatic Test Equipment*), que variam de preço conforme a precisão e faixa de frequência que se pretende medir, mas podem facilmente chegar a casa dos milhões de dólares. Dessa forma, o custo do teste pode representar uma parcela significativa do custo final do CI. Por isso, o tempo de execução do teste, que é diretamente proporcional ao custo do teste, deve ser minimizado (SOUDERS; STENBAKKEN, 1990).

2.1.2 Teste Funcional e Teste Estrutural

Os testes de circuitos analógicos podem ser classificados de acordo com seus objetivos. Denomina-se teste funcional aquele que avalia o funcionamento do circuito, e conclui se está ou não dentro das especificações. Por exemplo, um amplificador operacional falha no teste funcional caso seu ganho em malha aberta seja menor do que um limite predefinido (especificação). Da mesma forma, uma porta AND falha no teste funcional caso se comporte como uma porta OR, ou qualquer outra que não uma AND. Nota-se que conforme a complexidade dos sistemas integrados aumenta, a verificação de todas suas especificações pode tornar-se inviável.

Criou-se, então, a metodologia de teste estrutural, que verifica, basicamente, se o circuito não possui defeitos de fabricação, ou apresenta uma variação exagerada de algum parâmetro. Por exemplo, caso se quisesse fabricar um resistor de $1\text{ k}\Omega$, e ao testar este componente nota-se que sua resistência é de $2\text{ k}\Omega$, provavelmente ele seria considerado defeituoso.

O teste estrutural baseia-se na modelagem de diversas falhas, que são causadoras dos defeitos posteriormente identificados, que podem ocorrer num circuito. O conjunto de falhas leva o nome de modelo de falhas, e é recurso para a construção do dicionário de

falhas. Para que o teste estrutural seja viável, é necessário um modelo de falhas preciso (ARSLAN; ORAILOGLU, 2013).

2.1.3 Modelos de Falhas Comuns

Circuitos digitais e analógicos apresentam grandes diferenças de comportamento entre si quando nestes estão presentes falhas. Circuitos digitais podem assumir uma gama finita de estados, de forma que cada estado é diferente de qualquer outro. Portanto, se um circuito digital assume um estado diferente do estado esperado, ele é defeituoso. A modelagem de falhas destes, então, é facilitada, a exemplo do modelo *Stuck-at*, que pode ser aplicado à maior parte dos circuitos digitais (GALIAY; CROUZET; VERGNIAULT, 1980). O modelo *Stuck-at* assume que apenas um nó do circuito é falho, sendo este sempre "0" ou "1" independente de qualquer outro sinal. É fácil concluir que circuitos digitais têm um número de possíveis falhas que cresce linearmente com o número de nós do circuito (HUGHES, 1988). As falhas dos circuitos analógicos, por sua vez, não podem ser modeladas da mesma maneira, já que sinais no domínio analógico podem assumir infinitos valores de tensão.

A verificação de circuitos analógicos pode ser realizada por dois tipos de testes: teste orientado a especificações ou teste orientado a defeitos. O primeiro pode se tornar impraticável, pelo custo de equipamentos e tempo de teste necessários, à medida que o número de especificações aumenta (SOUDERS; STENBAKKEN, 1991). Logo, o teste orientado a defeitos tem recebido atenção especial tanto da indústria quanto da comunidade científica, visto o grande número de publicações sobre o assunto.

As falhas, que são um modelo abstrato de um defeito físico, assumida para o teste podem ser divididas em dois grupos: falhas paramétricas e falhas catastróficas. As falhas paramétricas decorrem do desvio do valor nominal dos parâmetros (daí o nome) do circuito, e são dadas quando tais parâmetros estão fora dos limites de tolerância (variações acima de 3σ ou 6σ) (KONDAGUNTURI et al., 1999; DENG; SHI; ZHANG, 2012). Segundo Kondagunturi et al. (1999), as falhas catastróficas se apresentam como terminais de componentes sem contatos ou com curto circuitos (não projetados) entre si, e análogas às falhas *Stuck-at* de circuitos digitais.

No âmbito das simulações, as falhas devem ser injetadas como componentes parasitas ou variações de parâmetros. Para as falhas catastróficas, as falhas de circuito aberto podem ser modeladas com um resistor de valor elevado, R_s , entre o terminal defeituoso e sua conexão projetada, enquanto para as falhas de curto circuito são introduzidos resistores de baixo valor, R_p , entre os terminais defeituosos. O valor de R_s e R_p utilizado não é único, enquanto para Kondagunturi et al. (1999) $R_s = 100\text{ M}\Omega$ e $R_p = 1\text{ }\Omega$, Petrashin et al. (2013) utiliza $R_s = 10\text{ M}\Omega$ e $R_p = 10\text{ }\Omega$. Já as falhas paramétricas são modeladas pela

variação relativa de parâmetros utilizados na simulação, como por exemplo a variação do comprimento de canal de um transistor em $\pm 25\%$ (BROSA; FIGUERAS, 1999).

2.1.4 Dicionário de Falhas

O primeiro passo da construção do dicionário de falhas é a definição do modelo de falhas a ser empregado. O modelo de falhas pode ser definido pela experiência do engenheiro de teste, pelos históricos de modos de falha na fabricação do circuito ou pela informação do *layout* do circuito.

Define-se, então, os nós cujas tensões poderiam ser medidas. Inicialmente, é comum escolherem-se todos os nós acessíveis do circuito e, após a análise dos resultados, definir quais são os nós relevantes.

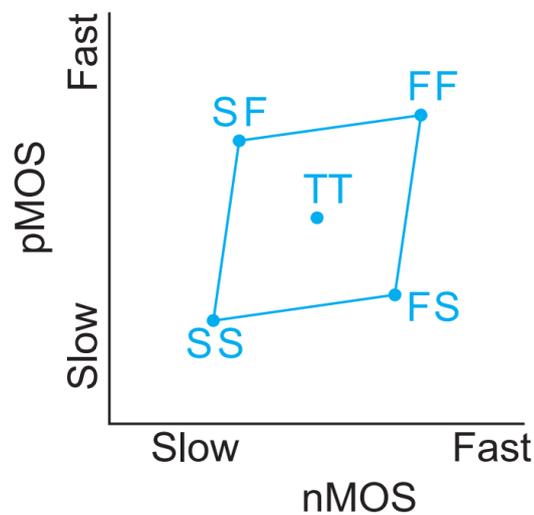
A última definição a ser feita são os sinais de teste, ou vetores de teste, a serem simulados, e posteriormente aplicados ao circuito. Segundo Kabisatpathy, Barua e Sinha (2005), até então não existe um algoritmo para a seleção de vetores de teste a serem aplicados em circuitos analógicos. A metodologia de tentativa e erro, abreviada pelo conhecimento do circuito em particular, é normalmente empregada, e.g. Bender (2015).

Finalmente, simula-se a aplicação dos vetores de teste no circuito nominal (*fault free*) e nos circuitos contendo as falhas modeladas. Os resultados das simulações, por exemplo as tensões nodais, referentes a cada vetor de teste e falha modelada constituem, portanto, o dicionário de falhas.

Neste trabalho, definiram-se, também, faixas de tolerância para as tensões nos nós de forma que, caso esta esteja fora dos limites, o circuito é considerado falho. Esta é uma proposta derivada do conceito de teste *go/no-go* (MILOR; VISVANATHAN, 1987), que avalia simplesmente se o circuito apresenta ou não alguma falha. Dessa forma, o dicionário de falhas, que pode ser uma estrutura binária, consiste da informação sobre a observabilidade de falhas (assinala-se '1' para falha observada ou '0' para falha não observada) de um dado nó, aplicando-se um determinado vetor de teste.

2.2 Análise de *corners*

As variações no processo de fabricação de circuitos integrados são inerentes à tecnologia de fabricação. Para o projetista do sistema, estas variações podem ser resumidas pelos efeitos que causam no comportamento dos componentes, em especial nos transistores. Os extremos das variações são conhecidos como *corners* da tecnologia, normalmente designados como *typical*, *fast* e *slow*. A Figura 2 ilustra o intervalo de tolerância entre *corners*.

Figura 2 – *Corners* da tecnologia

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

A análise de *corners* é muito utilizada durante a etapa de projeto de CI's, pois definem os limites do comportamento do circuito fabricado. De forma geral, se todas as especificações do projeto são atingidas em cada um dos *corners*, a probabilidade de um circuito fabricado não estar dentro das especificações é muito baixa. Ainda, na Subseção 3.1.3 ficará claro que este circuito poderia ser considerado falho.

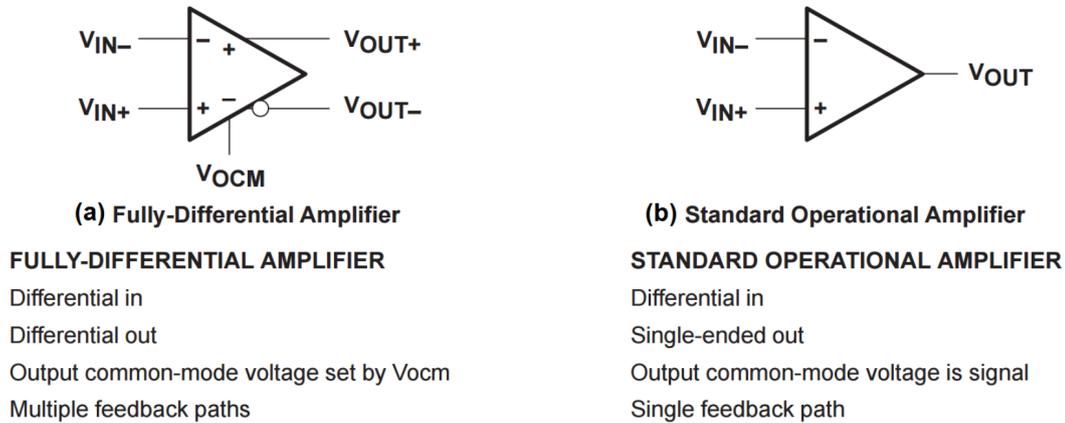
É importante ressaltar que a análise de *corners* é uma alternativa, muito mais barata em tempo de simulação, para a análise de Monte Carlo. Ainda que aquela seja menos completa e representativa das variações reais do processo de fabricação, ela ainda é capaz de verificar as condições de operação de circuitos analógicos (SYLVESTER, 2013).

2.3 Amplificador Totalmente Diferencial

Como o nome sugere, os FDAs têm não apenas suas entradas na forma diferencial, mas também suas tensões de saída. A Figura 3 ilustra algumas diferenças entre FDAs e OPAMPs convencionais. FDAs vêm sendo utilizados há décadas em aplicações de áudio, telefonia e transmissão de dados, mas no presente sua principal aplicação é na conversão de sinais analógicos para sinais digitais (ADC - *Analog-to-Digital Converter*) em alta velocidade (KARKI, 2002).

Uma das principais vantagens dadas pelo uso de um FDA é o aumento da relação sinal-ruído. A diminuição da potência do ruído na saída do amplificador é decorrência da maior rejeição do 'ruído acoplado', característica comum de sinais diferenciais. Por motivos similares, a distorção harmônica é reduzida em frequências pares. Dessa forma, FDAs são

Figura 3 – (a) Amplificador Totalmente Diferencial e (b) Amplificador Operacional Convencional.



Fonte: adaptado de Karki (2002)

escolhidos como interface entre sinais analógicos, providos, por exemplo, por sensores, e ADCs de alta velocidade (portanto baixa impedância de entrada) em sistemas de alta precisão. Eventualmente, o estágio de condicionamento do sinal analógico, implementado com OPAMPs, é considerado parte do conversor.

A principal diferença entre um OPAMP convencional, de saída simples, e um FDA é a existência da realimentação interna feita pelo circuito de realimentação de modo comum neste. Esta realimentação permite desacoplamento da tensão de modo comum do sinal de entrada e do sinal de saída, aumentando seu CMRR (*Common-Mode Rejection Rate*). O sinal de realimentação é, geralmente, responsável por parte da corrente de polarização dos estágios de amplificação principais, e variam de forma a estabilizarem a tensão de modo comum da saída em um valor predeterminado, normalmente o ponto médio entre as tensões de alimentação. Nota-se que o sinal de realimentação de modo comum, portanto, é função da soma das tensões de saída do FDA. Este fato é explorado, neste trabalho, a fim de verificar se a sensibilidade a falhas daquele nó é maior que a sensibilidade destes.

3 Metodologia Experimental

Este Capítulo pretende descrever de forma intuitiva e reproduzível o desenvolvimento do trabalho. Em um primeiro momento, é descrito em detalhes o funcionamento da ferramenta de automatização das simulações desenvolvida, assim como seus objetivos. A segunda Seção do Capítulo descreve os circuitos, modelos de falhas e configurações de teste utilizados para verificar a confiabilidade da ferramenta recém desenvolvida, que consiste na reprodução dos resultados apresentados em Bender (2015). Finalmente, na terceira Seção, descreve-se o modulador sigma-delta, projetado por Aguirre (2014), que foi objeto do estudo de caso do presente trabalho. São detalhados, também na terceira Seção, o modelo de falha empregado e a análise AC realizada nos blocos de interesse.

3.1 Automatização das Simulações

Tendo em vista quão tedioso e demorado pode se tornar o processo de simulação de inúmeros circuitos quando feito manualmente, foi desenvolvido em MATLAB R2015b um algoritmo de automatização. Este algoritmo toma como base a descrição do circuito sem falhas e um arquivo contendo uma lista de falhas a serem inseridas. O algoritmo cria, então, a descrição de cada circuito contendo uma das falhas, simula cada um dos circuitos, e analisa seus resultados.

A automatização divide-se em três partes. Nessa Seção, cada uma das partes é detalhada, a começar pelo método de alteração de *corners* e injeção de falhas; passando, então, pela execução e verificação de sucesso das simulações; e, finalmente, descrevendo construção do dicionário de falhas dos nós de interesse e frequências de teste.

Os *scripts* que executam o algoritmo de automatização, assim como todas as funções¹ utilizadas, estão em formato de texto nos Apêndices A até M. Toda implementação foi realizada em MATLAB, e verificada na versão R2015b.

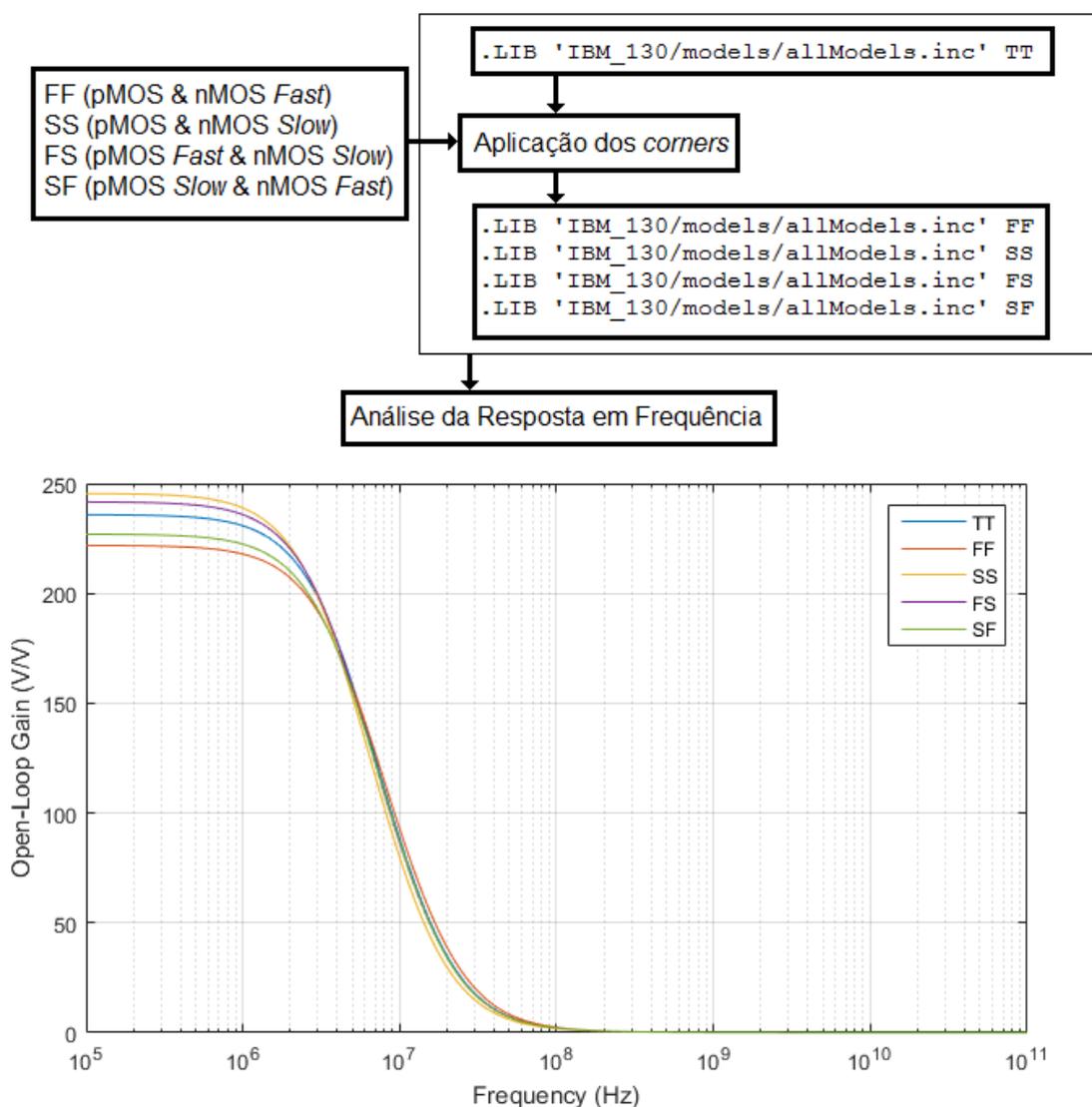
3.1.1 Alteração de *Corners* e Injeção de Falhas

A análise de *corners* é realizada durante a simulação em SPICE pela simples alteração da biblioteca de modelos da tecnologia de fabricação. Em geral, a *foundry* disponibiliza os limites normais de variação em uma lista de parâmetros que pode ser acessada pela sigla do *corner* a que se relaciona. A alteração do *corner* a ser simulado é feita, portanto, alterando-se a sigla endereçada no arquivo de simulação.

¹ Funções não contidas no pacote original do MATLAB R2015b, nem no pacote HSPICE Toolbox for MATLAB.

A Figura 4 exemplifica a análise de *corners* para o OPAMP projetado na tecnologia IBM 8RF-DM de $0,13\ \mu\text{m}$, que é o assunto do estudo de caso da Seção 3.3. Fica claro que o projetista precisa informar para o *script* de automatização o caminho (em inglês *path*) para a biblioteca de modelos da tecnologia em questão, assim como as siglas de cada *corner*. Com estas informações, são criados os arquivos de simulação para cada *corner*, estes são simulados, e finalmente seus resultados utilizados para criação do dicionário de falhas.

Figura 4 – Exemplo da Análise de *Corners*.



Fonte: elaborado pelo autor.

A injeção de falhas para simulação SPICE também consiste na edição da descrição do circuito projetado. Seja pela alteração de valores nominais (falhas paramétricas),

ou inclusão de componentes não desejados (resistores em série ou paralelo para falhas catastróficas) e alteração das conexões.

Definido o modelo de falhas, contendo n possíveis falhas ocorrentes em um circuito, passa-se então a criar n arquivos de descrição dos circuitos, cada um com uma falha injetada. É importante ressaltar que a estratégia utilizada é baseada na suposição de que a probabilidade de um defeito mascarar outro é muito baixa. Isto torna possível utilizar um modelo de falhas únicas. Caso fosse empregado um modelo de falhas múltiplas, o número de simulações necessárias aumentaria exponencialmente com o número de componentes do circuito, tornando a execução do teste inviável mesmo que de forma automatizada.

O método de injeção automatizada de falhas desenvolvido neste trabalho necessita apenas do arquivo de descrição em SPICE do circuito sem falhas (também chamado de *netlist*, geralmente com extensão '.sp') e de um arquivo exclusivo (de extensão '.fii'²) contendo todas as falhas que se desejam injetar. A sintaxe de descrição de cada falha contida no arquivo exclusivo foi escolhida de forma a manter o arquivo compreensível. Um exemplo de funcionamento é ilustrado na Figura 5, onde a *netlist* descreve um filtro RC passivo e nele são adicionados três defeitos: defeito de circuito aberto no primeiro nó de C1; defeito de curto circuito entre o primeiro e o segundo nó de R1; e defeito de desvio de 10% no valor da capacitância de C1. Um segundo exemplo de um arquivo de instrução para injeção de falhas (FII) pode ser visto no Apêndice O. Este arquivo FII foi utilizado para descrever as falhas do amplificador operacional PMOS-DA (Apêndice N), que é estudado na Subseção 3.2.1.

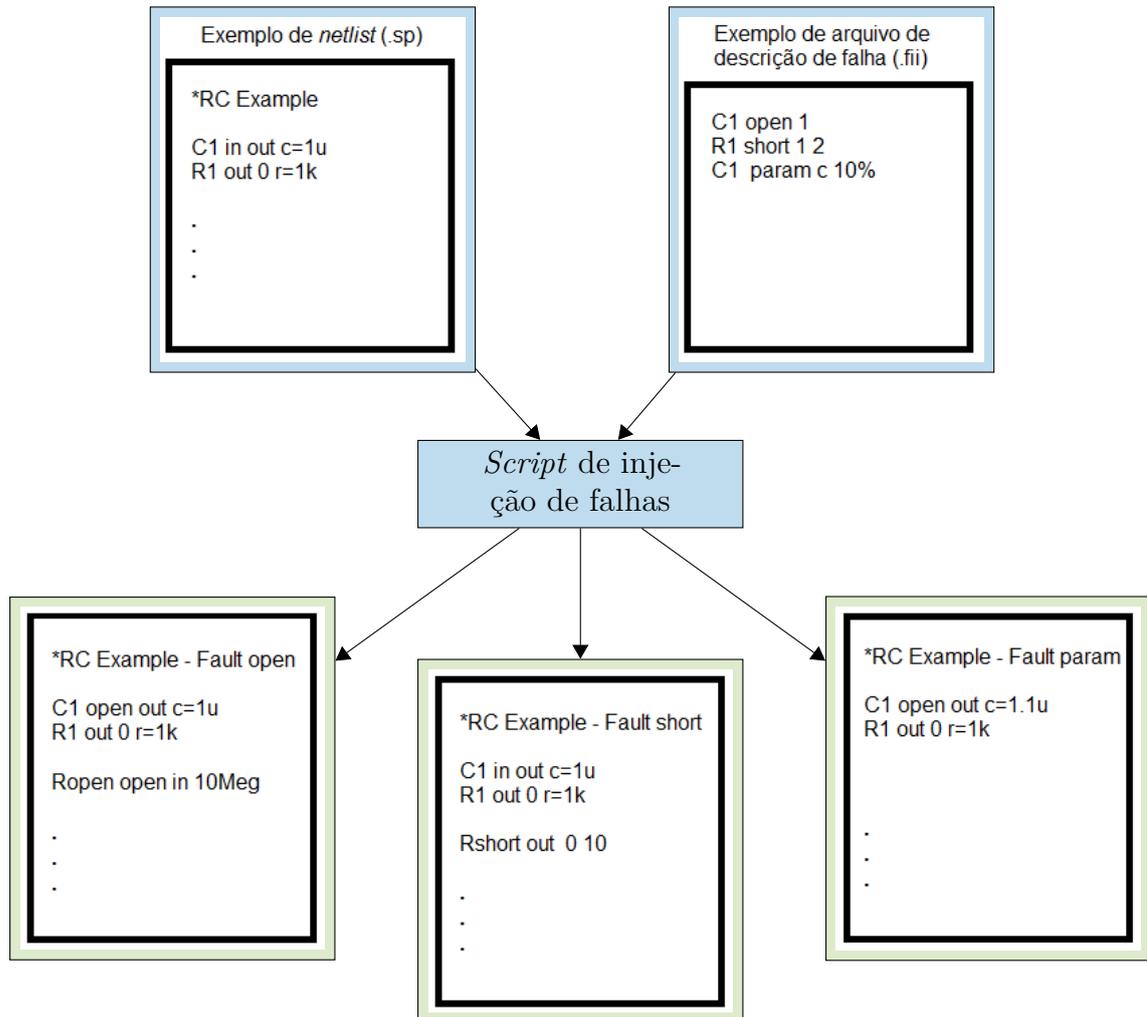
3.1.2 Execução e Verificação das Simulações

O simulador de circuitos elétricos utilizado neste trabalho foi o HSPICE. Este programa possibilita ao usuário a simulação via linha de comando, gerando diversos arquivos contendo os resultados ao final da simulação. Tendo todas as descrições de circuitos necessárias, faz-se uma chamada ao simulador para cada arquivo, com o objetivo de obter-se os resultados das simulações.

Como a chamada ao simulador é externa ao ambiente do MATLAB, via sistema operacional, é possível que algumas das simulações falhem. Portanto, após todas as chamadas ao simulador, é feita uma verificação dos arquivos criados. Caso note-se alguma falha, o usuário é comunicado sobre o erro e instruído sobre como proceder. Chama-se novamente o simulador até o sucesso de todas as simulações. Por fim, é criado um diretório auxiliar, onde ficam armazenadas todas as descrições e resultados da análise de *corners* e dos circuitos com falha.

² A sigla FII corresponde a *Fault Injection Instruction*.

Figura 5 – Exemplo de utilização do algoritmo de injeção de falhas.



3.1.3 Construção do Dicionário de Falhas

Os resultados das simulações são armazenados em formato binário, em arquivos padronizados pelo simulador. Foi utilizado neste trabalho o HSPICE Toolbox for MATLAB ((PERROTT, 2011)) para realizar a interpretação dos arquivos gerados pelo simulador e converter as informações relevantes para vetores do MATLAB. Em um primeiro momento, no entanto, foi desenvolvido um programa para este propósito, que, apesar do bom funcionamento, é consideravelmente mais lento do que as funções desenvolvidas por Perrott (2011).

De posse dos resultados das simulações em vetores do MATLAB, analisa-se em que configuração de teste cada falha pode ser detectada. Se em uma dada configuração de teste detecta-se uma certa falha, marca-se no dicionário de falhas tal informação. Tendo o dicionário de falhas completo, o projetista do teste a ser empregado no circuito pode escolher as configurações de testes que melhor satisfaçam suas necessidades.

A decisão de assinalar uma falha como sendo detectável para uma configuração de teste é tomada baseada em limites toleráveis da variável medida (tensão/corrente DC, amplitude do sinal AC, *overshoot*, tempo de acomodação etc). Os limites, neste trabalho, são dados pelo valor máximo e mínimo entre todos os *corners*.

Neste trabalho foi utilizada principalmente a resposta em frequência do circuito. Neste caso, a configuração de teste é dada por duas variáveis controláveis: a frequência de teste e o nó cuja tensão é medida. Portanto, o dicionário de falhas tem informação sobre quais falhas podem ser detectadas em quais nós de tensão, em cada frequência de teste. Pode-se chamar também de biblioteca de dicionários de falhas, sendo cada dicionário referente a um único nó, contendo as falhas detectadas em cada frequência de teste. Claramente a decisão sobre a hierarquia da biblioteca é arbitrária, já que cada variável de controle é independente das outras.

A Figura 6 ilustra a faixa de tolerância (em vermelho) da amplitude da tensão, definida pela resposta dos *corners* da tecnologia. O circuito simulado é o mesmo apresentado na Subseção 3.3.1, e o nó analisado é o nó de saída do primeiro bloco de CMFB. Nota-se, neste exemplo, que o limite superior é definido pela amplitude da tensão da resposta do *corner* SS em baixas frequências, mas em altas frequências passa a ser definido pela resposta do *corner* FS.

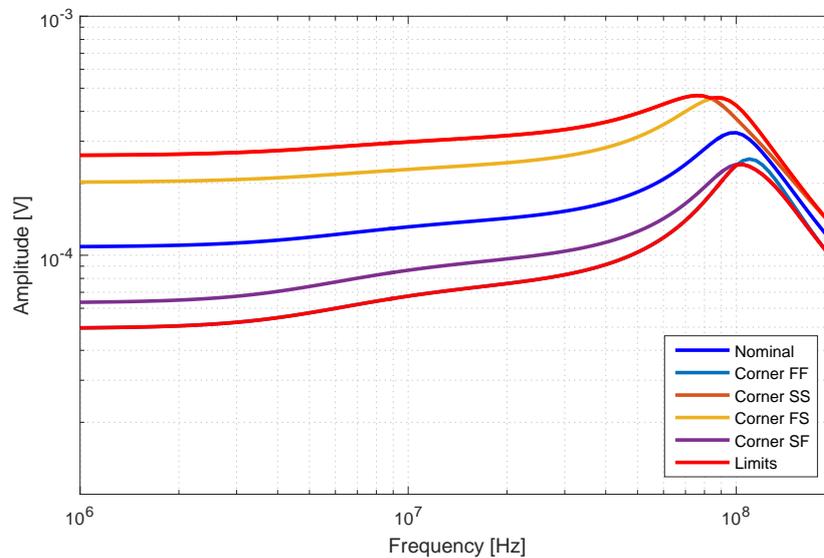
A Figura 7 exemplifica a faixa de frequências em que uma falha pode ser detectada avaliando-se a tensão em um certo nó, neste caso o nó V_{ctrl1} (Subseção 3.3.1). A resposta do circuito com tal falha é representada em preto, enquanto os limites aceitáveis, em vermelho, foram previamente obtidos (Figura 6). Nota-se que a partir da frequência de aproximadamente 80 MHz a falha pode ser detectada neste nó de tensão.

3.2 Validação da Ferramenta de Automatização Desenvolvida

Durante o desenvolvimento da ferramenta de automatização das simulações, o seu bom funcionamento foi verificado baseado em resultados passados apresentados em Bender (2015). No trabalho em questão são apresentados três estudos de caso: "Amplificadores Diferenciais de dois Estágios com Saída Simples"; "Amplificador Totalmente Diferencial de um Estágio"; e "Amplificador Operacional de dois Estágios com Saídas Diferenciais". Como as simulações realizadas não contêm o modelo completo de falhas para o último caso, decidiu-se, neste trabalho, não replicar os experimentos deste. Foram replicadas, portanto, todas as simulações realizadas no primeiro e segundo estudos de caso.

3.2.1 Amplificadores Diferenciais de dois Estágios com Saída Simples

Foram utilizados dois amplificadores diferenciais de dois estágios de topologias complementares com compensação Miller. O primeiro utiliza transistores do tipo PMOS

Figura 6 – Definição dos limites toleráveis pela análise de *corners*.

Fonte: elaborado pelo autor.

(*P-channel Metal Oxide Semiconductor*) para realização do par diferencial, e portanto é denominado PMOS-DA (PMOS - *Diferencial Amplifier*). O seu complementar, denominado NMOS-DA, tem seu par diferencial projetado com transistores NMOS (CARDOSO, 2012).

A Figura 8 apresenta o esquemático dos dois OPAMPs complementares, com destaque para a identificação de cada nó do circuito. As dimensões dos transistores e valores nominais do projeto são apresentados na Tabela 1.

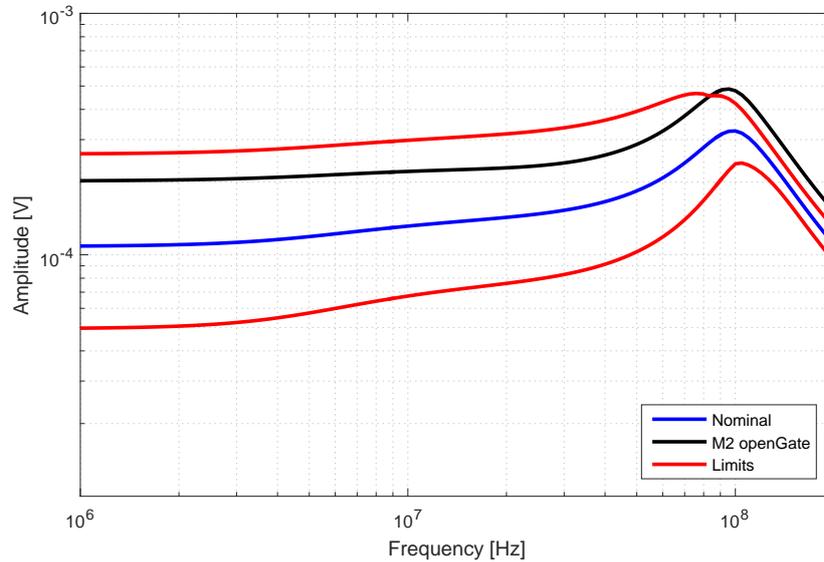
A exemplo da análise realizada em Bender (2015), foram consideradas apenas falhas catastróficas nesses circuitos. Também é importante salientar que não foi feita a análise de *corners* desses, de forma que as margens de tolerância foram definidas arbitrariamente pela autora por um erro relativo de 5% da variável medida, a tensão em cada nó.

Foi realizado um único teste DC, com as entradas de sinal, V_{in+} e V_{in-} , aterradas, e com as tensões de alimentação nominais. O modelo de falha contém seis falhas por transistor, sendo elas denominadas: OD (*open drain*); OG (*open gate*); OS (*open source*); SDG (*short drain-gate*); SDS (*short drain-source*); e SGS (*short gate-source*). Cada um dos amplificadores faz uso de oito transistores, somando-se 48 falhas por circuito. Portanto, são necessárias 98 (uma por falha e uma dos circuitos *fault free*) simulações para essa análise.

3.2.2 Amplificador Totalmente Diferencial de um Estágio

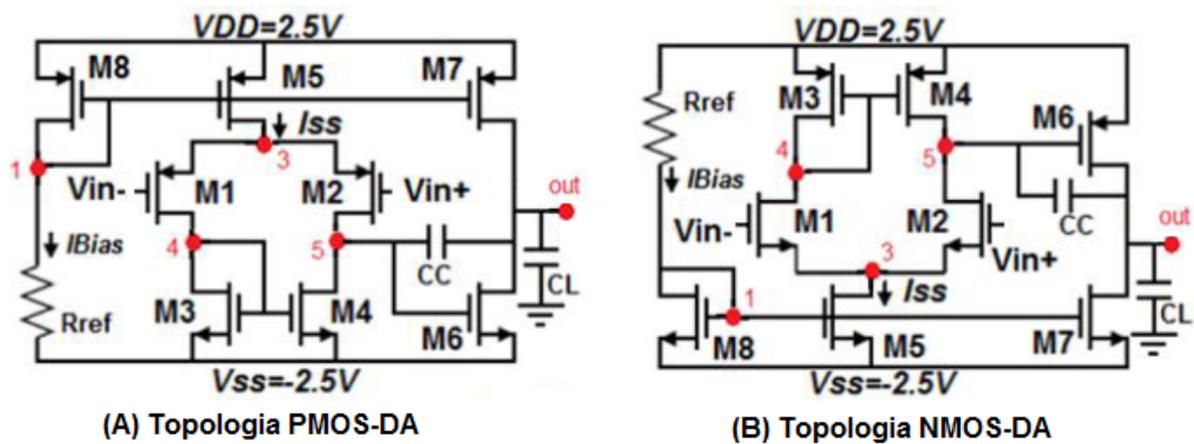
Como segunda etapa de validação da ferramenta de automatização, foram utilizados os resultados obtidos no segundo estudo de caso de Bender (2015). Nesse Capítulo, a

Figura 7 – Exemplo de resposta em frequência de circuito defeituoso fora dos limites toleráveis.



Fonte: elaborado pelo autor.

Figura 8 – Diagrama esquemático das topologias (A) PMOS-DA e (B) NMOS-DA.



Fonte: adaptado de Cardoso (2012)

autora passa a utilizar um modelo de falhas mais completo, contendo também as falhas paramétricas. Além disso, são empregadas diversas configurações de testes, incluindo testes DC e testes transientes.

O amplificador totalmente diferencial objeto do estudo de caso em questão foi projetado em tecnologia XFAB 0,18 μm por Oliveira, Severo e Girardi (2014). Como visto

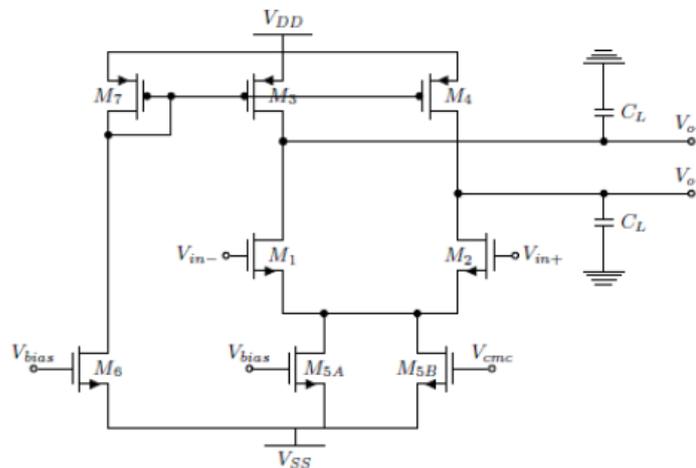
Tabela 1 – Dimensões e valores de projeto das topologias PMOS-DA e NMOS-DA.

Parâmetro	Topologia		Unidade
	PMOS-DA	NMOS-DA	
W_1/L_1	4/1	1,5/1	$\mu\text{m}/\mu\text{m}$
W_2/L_2	4/1	1,5/1	$\mu\text{m}/\mu\text{m}$
W_3/L_3	1/1	5/1	$\mu\text{m}/\mu\text{m}$
W_4/L_4	1/1	5/1	$\mu\text{m}/\mu\text{m}$
W_5/L_5	2,5/1	1,5/1	$\mu\text{m}/\mu\text{m}$
W_6/L_6	24/1	60/1	$\mu\text{m}/\mu\text{m}$
W_7/L_7	30/1	9/1	$\mu\text{m}/\mu\text{m}$
W_8/L_8	2,5/1	1,5/1	$\mu\text{m}/\mu\text{m}$
Resistor R_{ref}	335	352	$\text{k}\Omega$
Capacitor CC	4,4	4,4	pF
P_{diss}	650	350	μW

Fonte: adaptado de Cardoso (2012)

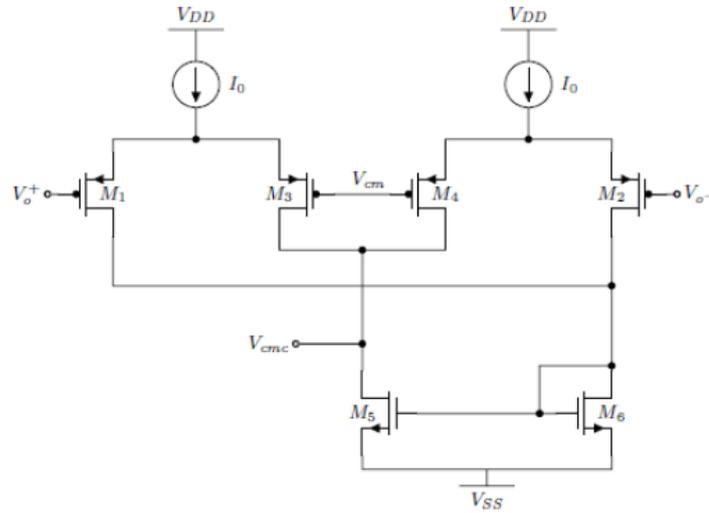
na Seção 2.3, o FDA necessita de um circuito de CMFB para estabilizar a tensão de modo comum de saída. Por fins de melhor compreensão, a Figura 9 apresenta apenas o esquemático do bloco de amplificação diferencial, enquanto a Figura 10 mostra o esquemático do bloco de CMFB. As tensões e correntes de polarização são consideradas fixas e ideais. A Tabela 2 descreve os valores de projeto utilizados.

Figura 9 – Diagrama esquemático do bloco de amplificação diferencial projetado em tecnologia XFAB 0,18 μm .



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

Figura 10 – Diagrama esquemático do bloco de CMFB projetado em tecnologia XFAB 0,18 μm .



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

Tabela 2 – Dimensões e valores de projeto do FDA projetado em tecnologia XFAB 0,18 μm .

Parâmetro	Bloco		Unidade
	Amp. Diferencial	CMFB	
W_1/L_1	36,29/0,197	35,91/0,8918	$\mu\text{m}/\mu\text{m}$
W_2/L_2	36,29/0,197	35,91/0,8918	$\mu\text{m}/\mu\text{m}$
W_3/L_3	27,41/6,83	35,91/0,8918	$\mu\text{m}/\mu\text{m}$
W_4/L_4	27,41/6,83	35,91/0,8918	$\mu\text{m}/\mu\text{m}$
W_5/L_5	15,3/9,75	7,19/0,4628	$\mu\text{m}/\mu\text{m}$
W_6/L_6	15,3/9,75	7,19/0,4628	$\mu\text{m}/\mu\text{m}$
W_7/L_7	27,41/6,83	–	$\mu\text{m}/\mu\text{m}$
V_{bias}	–167,0.45	–	mV
I_o	–	15,19	μA

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

O modelo de falhas utilizado assume seis falhas catastróficas por transistor (as mesmas citadas na Subseção 3.2.1), além das variações extremas de $\pm 25\%$ do comprimento do canal e de 50 mV da tensão de *threshold* dos transistores. Todas essas somam, então, nove falhas por transistor. A autora do estudo justifica a escolha de valores para as variações paramétricas baseada em diversas publicações (BROSA; FIGUERAS, 1999; ALLANI, 2010; BORKAR et al., 2003; DENG; SHI; ZHANG, 2012). São excluídas do modelo seis falhas: SGD em M_7 do bloco principal, pois é um curto-circuito que faz parte do projeto;

SGD em M_6 do bloco de CMFB, pelo mesmo motivo; SDS em M_{5A} e M_6 do bloco principal, pois assume-se a tensão de polarização V_{bias} como ideal; e falha paramétrica de variação negativa do comprimento de canal dos transistores M_1 e M_2 do bloco principal, porque o comprimento final seria menor do que o limite da tecnologia, de $0,18 \mu\text{m}$.

Por assumir um modelo mais elaborado de falhas, foram necessárias diversas configurações de teste para se obter uma cobertura de falhas elevada. As configurações de conexão do CUT (*Circuit Under Test*) para os diversos testes simulados são apresentadas na Tabela 3.

Tabela 3 – Configurações de testes aplicadas ao FDA projetado em tecnologia XFAB $0,18 \mu\text{m}$.

Nome	Sinal de Teste	Descrição da Configuração
DC1	Vip=0; Vin=0	Malha Aberta
DC2	Vip=Vdd; Vin=Vee	Malha Aberta
DC3	Vip=Vee; Vin=Vdd	Malha Aberta
TR1	Onda quadrada de $0,8V_{pp}$ em Vin	Malha Fechada: Vip = Von
TR2	Onda quadrada de $0,8V_{pp}$ em Vip	Malha Fechada: Vin = Vop

A verificação da ferramenta de automatização levou em consideração os resultados que fazem uso da análise de *corners* (BENDER, 2015, p. 47-52). Nessa análise é apresentada a ideia de definir os limites das variáveis de medida de teste pela análise de *corners*. A autora define um novo limite de erro relativo (previamente de 5%) baseado no máximo erro relativo entre as tensões da análise DC nos *corners*, ampliando a faixa de tolerância para 35% em relação à tensão nominal.

Um conceito similar é empregado para os testes transientes, mas difere no sentido de utilizar os valores absolutos entre *corners*, e não mais definir uma faixa de erro relativo. O valor de tensão medido nos testes permitiu, portanto, à autora quantificar os limites de *overshoot* e *settling time* dos circuitos sem falhas. No entanto, nem com as informações presentes no texto, nem com o acesso aos arquivos de simulação criados pela autora, foi possível identificar os valores limites de *settling time* definidos para julgar defeituoso ou não um circuito. Dessa forma, a validação da ferramenta de automatização foi feita apenas com base nos testes DC.

Validado o funcionamento da ferramenta de automatização, propõe-se construir o dicionário de falhas baseado na resposta em frequência do circuito, ou também dicionário de falhas AC. A motivação para tal proposta foi ter percebido a arbitrariedade com que normalmente se escolhe a configuração de teste de circuitos analógicos. Com um dicionário contendo um grande número de possíveis configurações de teste, que não são arbitrariamente definidas, mas sim pela varredura da resposta em frequência, imagina-se que seja possível otimizar o tempo de teste necessário, e também aumentar-se a cobertura

de falhas. Também é válido ressaltar que, para cada um dos testes DC ou transientes, é necessária uma descrição em SPICE diferente, enquanto que para o teste AC, o simulador SPICE permite a análise da resposta em frequência completa do circuito com apenas um *netlist*, resultando em um tempo de simulação muito menor.

O teste AC simulado no circuito em questão consiste de aplicar um sinal de amplitude reduzida nas entradas do FDA e verificar a resposta em frequência dos nós de interesse. O sinal de entrada deve ter amplitude reduzida para garantir a linearidade da resposta do circuito. Neste caso, foi escolhida uma amplitude de 1 mV.

3.3 Estudo de Caso: Modulador Sigma-Delta

A fim de fazer bom uso da ferramenta de automatização de simulações, escolheu-se um circuito mais complexo que os apresentados até então. Buscou-se um circuito que se mostrasse robusto e que já tivesse sido fabricado, para que os resultados deste trabalho pudessem ser implementados ao teste do CI. O circuito escolhido para ser estudado foi o modulador sigma-delta projetado em Aguirre (2014).

O modulador sigma-delta foi projetado na topologia de laço único, *single-bit*, de terceira ordem do tipo Cascata de Integradores com Alimentação em Avanço e realimentação local (SCHREIER; TEMES, 2004). Dessa forma, faz-se o uso de quatro FDAs (idênticos por escolha do projetista) e diversos componentes passivos no laço principal do modulador, além de um quantizador de um *bit*, chaves digitais para implementar o DAC (*Digital-to-Analog Converter*) de retorno.

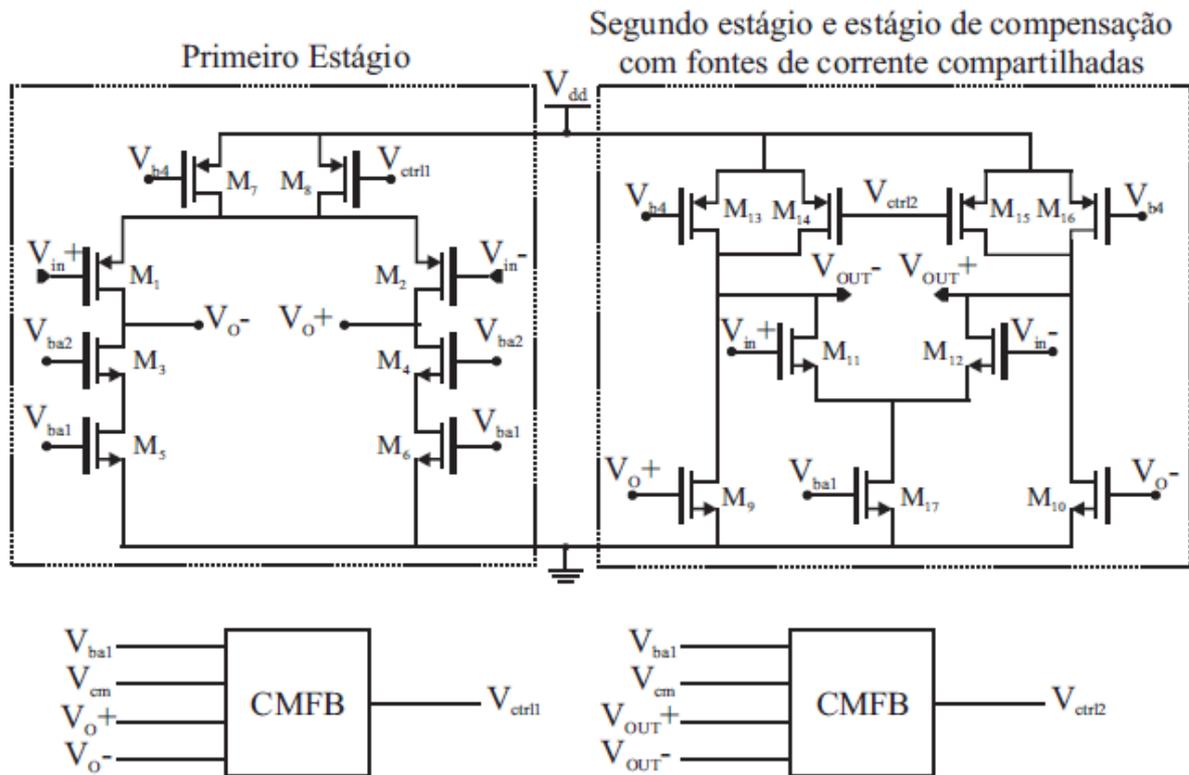
O presente trabalho apresenta a análise da cobertura de falhas nos FDAs utilizados no laço principal do modulador. Primeiramente, é realizada a análise de um FDA isolado para verificar a viabilidade dos testes no circuito completo. Uma vez verificada a viabilidade dos testes, dadas as observações necessárias, propõe-se um método de teste minimamente invasivo para o modulador sigma-delta completo.

3.3.1 Amplificador Totalmente Diferencial de dois Estágios

O FDA projetado na tecnologia IBM 8RF-DM de $0,13 \mu\text{m}$ consiste de dois estágios de ganho diferencial e um estágio de compensação em avanço. A técnica de compensação de fase por realimentação em avanço se mostrou vantajosa em relação à compensação de Miller por aumentar o GBW (*Gain-Bandwidth product*) do FDA, ao custo de um amplificador a mais. O FDA necessita de dois blocos de CMFB, o primeiro dedicado ao primeiro estágio amplificador e o segundo compartilhado entre o segundo estágio amplificador e o estágio de compensação. Essa separação é necessária pelo fato dos estágios amplificadores terem correntes de polarização diferentes.

A Figura 11 apresenta o esquemático completo dos três estágios do FDA e ilustra a conexão realizada com os blocos de CMFB. O primeiro estágio utiliza um par diferencial PMOS e fontes de corrente *cascode*. O segundo estágio é implementado por dois amplificadores fonte comum, enquanto a compensação é feita por um amplificador diferencial clássico.

Figura 11 – Diagrama esquemático dos estágios principais e de compensação em avanço do FDA projetados em tecnologia IBM 0,13 μm .

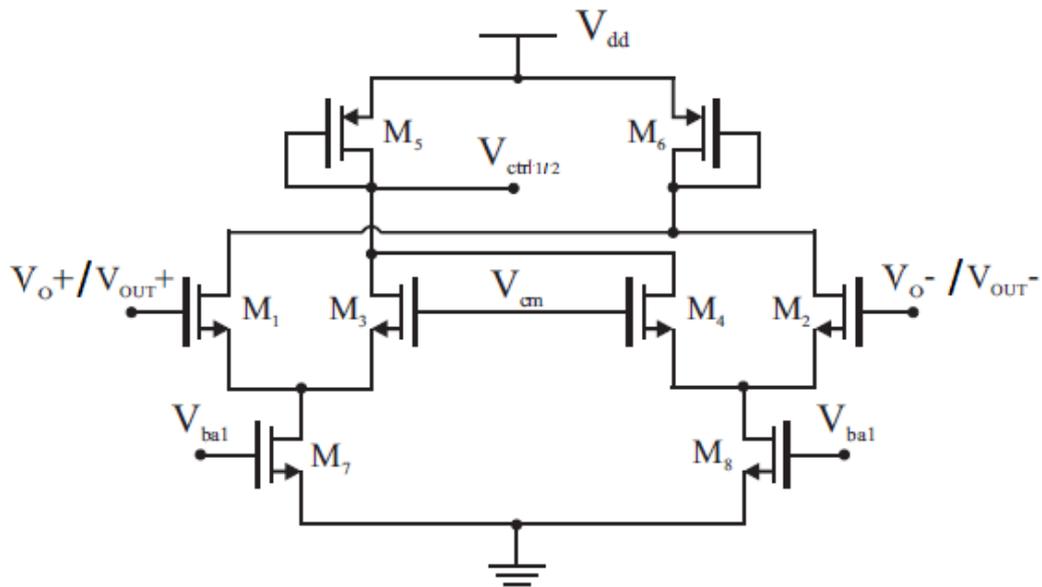


Fonte: adaptado de Aguirre (2014)

O esquemático interno dos blocos de CMFB é apresentado na Figura 12. O funcionamento do bloco é baseado em dois pares diferenciais que comparam as tensões de saída do amplificador com a tensão V_{cm} , igual 0,6 V, metade da tensão de alimentação. Quando a tensão de modo comum da saída do amplificador for igual à 0,6 V, a tensão de controle V_{ctrl} é responsável por 50% da corrente de polarização do amplificador.

A polarização do FDA foi projetada de forma a manter todos os transistores saturados e a topologia utilizada é apresentada por Baker (2010). A Figura 13 mostra o esquemático do circuito de polarização. Nota-se que os transistores M_4 , M_5 e M_6 formam um espelho de corrente, cuja corrente é definida pela fonte I de $20 \mu\text{A}$, implementada externamente ao CI por um resistor de $40 \text{ k}\Omega$. A Tabela 4 contém as dimensões dos transistores de todos os blocos do FDA. A tensão de alimentação é de 1,2 V.

Figura 12 – Diagrama esquemático do bloco de CMFB projetado em tecnologia IBM 0,13 μm .



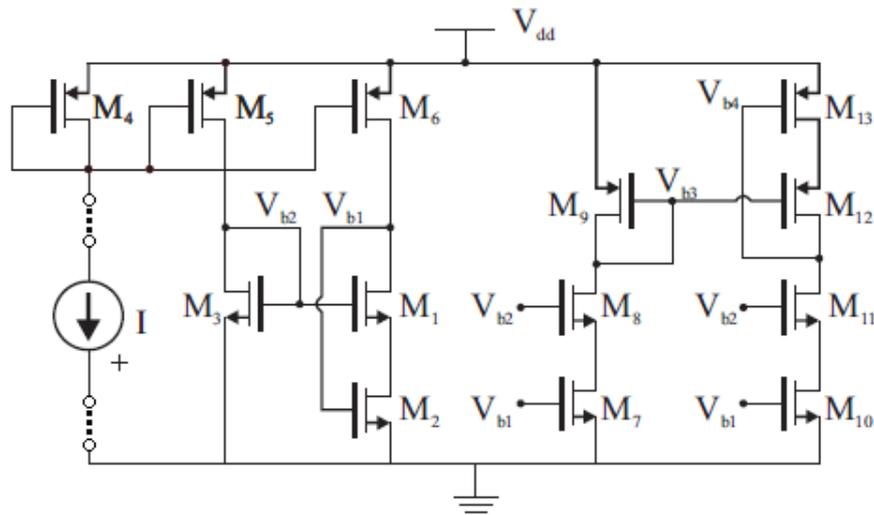
Fonte: adaptado de Aguirre (2014)

Tendo em vista a facilidade de se injetar falhas com a ferramenta desenvolvida, decidiu-se utilizar um modelo de falhas mais completo para os próximos circuitos. O modelo agora conta com seis falhas catastróficas, como na Subseção 3.2.1, e seis falhas paramétricas por transistor. Em relação ao modelo de falhas utilizado na Subseção 3.2.2, acrescentaram-se as falhas de desvio de negativo da tensão de *threshold* e desvios de $\pm 25\%$ de largura do canal (SEGURA; HAWKINS, 2004). Também foi alterado o valor do desvio de V_{th} , por tratarem-se de tecnologias de fabricação diferentes. O nó tecnológico de 130 nm tem desvio padrão, σ , da tensão de *threshold* igual a 15 mV (GREER; KORKIN; LABANOWSKI, 2003), i.e. o valor de V_{th} real dos transistores fabricados em 0,13 μm está dentro da faixa de $\pm 15\text{ mV}$ (1σ) do valor nominal para 68% dos casos. Portanto, para ser considerada falha utilizou-se uma diferença de $\pm 90\text{ mV}$ (diferença de 6σ) (DENG; SHI; ZHANG, 2012).

O circuito completo do FDA projetado contém 46 transistores, totalizando 552 falhas. No entanto, são ignoradas as falhas de curto-circuito entre os terminais de dreno e porta quando este curto-circuito é parte do projeto. Dessa forma, excluem-se sete falhas do modelo: SDG em M_5 e M_6 dos blocos de CMFB (duplicadas por serem dois blocos) e SDG em M_3 , M_4 e M_9 do bloco de polarização. O modelo de falhas final, portanto, compreende 545 falhas analisadas.

Levando em consideração os resultados e discussões que serão vistos no Capítulo 4, Subseção 4.1.2, optou-se por realizar a simulação de um teste AC. O teste AC consiste de

Figura 13 – Diagrama esquemático do bloco de polarização do FDA projetado em tecnologia IBM 0,13 μm .



Fonte: adaptado de Aguirre (2014)

excitar o circuito a ser testado com estímulos AC e verificar a amplitude da tensão nos nós de interesse. Portanto, na etapa de planejamento do teste AC, analisa-se a resposta em frequência da amplitude da tensão nos nós de interesse. A faixa de frequência foi definida de forma a incluir as singularidades do circuito, i.e. os polos e o zero implementados pelo FDA. Os nós de saída não inversora e inversora, assim como os dois nós de realimentação de modo comum, foram escolhidos como nós de interesse, a exemplo de Bender (2015). É válida a ressalva de que quando estes testes foram iniciados, ainda esperava-se concluir sobre viabilidade de utilizar os nós de realimentação de modo comum como um *checker* do FDA.

Foi empregada a configuração de teste em malha aberta, com o cuidado de estimular a entrada do FDA com um sinal de amplitude suficientemente pequena para que este opere na sua faixa linear. De forma a manter uma boa margem de segurança, escolheu-se o sinal AC com amplitude de 1 mV.

Tabela 4 – Dimensões dos transistores do FDA projetado em tecnologia IBM 0,13 μm .

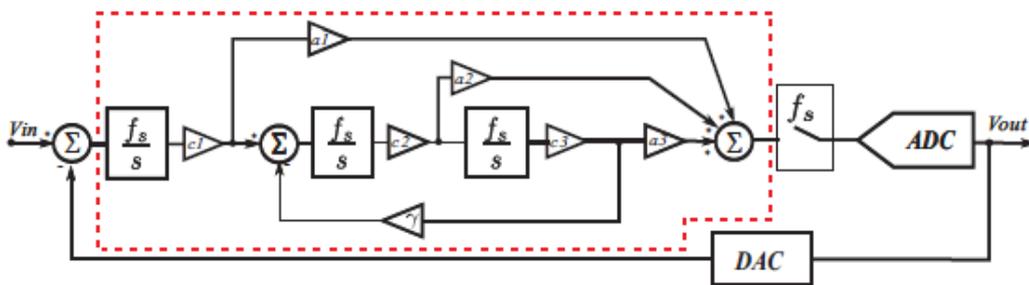
Parâmetro	Bloco			Unidade
	OPAMP	CMFB	Polarização	
W_1/L_1	6, 0/0, 6	96, 0/0, 6	10, 0/0, 6	$\mu\text{m}/\mu\text{m}$
W_2/L_2	6, 0/0, 6	96, 0/0, 6	10, 0/1, 2	$\mu\text{m}/\mu\text{m}$
W_3/L_3	10, 0/0, 6	96, 0/0, 6	2, 0/3, 6	$\mu\text{m}/\mu\text{m}$
W_4/L_4	10, 0/0, 6	96, 0/0, 6	10, 0/1, 2	$\mu\text{m}/\mu\text{m}$
W_5/L_5	10, 0/1, 2	80, 0/1, 2	10, 0/1, 2	$\mu\text{m}/\mu\text{m}$
W_6/L_6	10, 0/1, 2	80, 0/1, 2	10, 0/1, 2	$\mu\text{m}/\mu\text{m}$
W_7/L_7	10, 0/1, 2	80, 0/1, 2	10, 0/1, 2	$\mu\text{m}/\mu\text{m}$
W_8/L_8	10, 0/1, 2	80, 0/1, 2	10, 0/0, 6	$\mu\text{m}/\mu\text{m}$
W_9/L_9	5, 0/2, 4	–	2, 4/1, 2	$\mu\text{m}/\mu\text{m}$
W_{10}/L_{10}	5, 0/2, 4	–	10, 0/1, 2	$\mu\text{m}/\mu\text{m}$
W_{11}/L_{11}	12, 0/0, 6	–	10, 0/0, 6	$\mu\text{m}/\mu\text{m}$
W_{12}/L_{12}	12, 0/0, 6	–	8, 0/0, 6	$\mu\text{m}/\mu\text{m}$
W_{13}/L_{13}	10, 0/1, 2	–	10, 0/1, 2	$\mu\text{m}/\mu\text{m}$
W_{14}/L_{14}	10, 0/1, 2	–	–	$\mu\text{m}/\mu\text{m}$
W_{15}/L_{15}	10, 0/1, 2	–	–	$\mu\text{m}/\mu\text{m}$
W_{16}/L_{16}	10, 0/1, 2	–	–	$\mu\text{m}/\mu\text{m}$
W_{17}/L_{17}	20, 0/1, 2	–	–	$\mu\text{m}/\mu\text{m}$

Fonte: adaptado de Aguirre (2014)

3.3.2 Filtro Analógico do Modulador Sigma-Delta

O modulador sigma-delta projetado em Aguirre (2014) conta com um laço único, um filtro de terceira ordem, um quantizador e um DAC, ambos com resolução de um *bit*. A topologia do filtro utilizada é conhecida como Cascata de Integradores com Alimentação em Avanço (CIFF - *Chain Integrator with Feedforward*), incrementada de uma realimentação local a fim de formar um ressonador. A Figura 14 exhibe o diagrama de blocos do SDM em tempo contínuo projetado, onde os coeficientes, adaptados para a implementação analógica do filtro (destacado em vermelho), são dados por: $[c_1, c_2, c_3, a_1, a_2, a_3, \gamma] = [0,1, 1, 0,1, 1,7, 0,61, 1,4, 0,015]$ e a frequência de amostragem, f_s , igual a 128 MHz.

Figura 14 – Diagrama de blocos do SDM projetado em tecnologia IBM 0,13 μm .



Fonte: adaptado de Aguirre (2014)

A realização analógica do filtro, ilustrada na Figura 15, emprega quatro FDAs (ver Subseção 3.3.1), três deles implementando os três integradores e um último dedicado à soma dos sinais intermediários. Os componentes passivos foram implementados por resistores de precisão de polisilício e capacitores tipo *dual* MIM (*Metal-Insulator-Metal*). Os valores destes componentes são detalhados na Tabela 5, de forma a implementarem os coeficientes do filtro citados anteriormente.

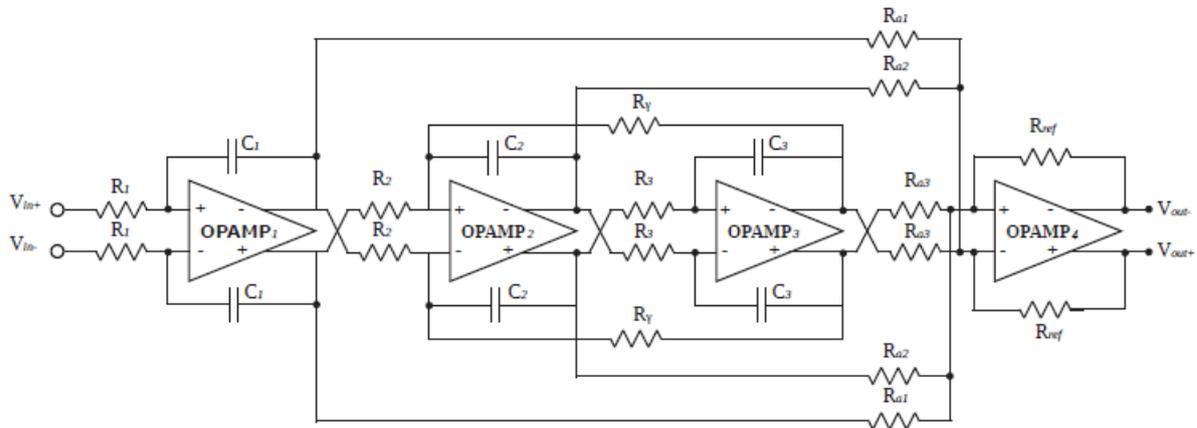
Tabela 5 – Valores dos componentes passivos do filtro analógico projetado em tecnologia IBM 0,13 μm .

Componente	Índice k								Unidade
	X_1	X_2	X_3	X_{a1}	X_{a2}	X_{a3}	X_{ref}	X_γ	
Resistor R_k	25,00	7,81	78,10	14,70	41,00	17,80	25,00	520,00	$\text{k}\Omega$
Capacitor C_k	3,125	1,00	1,00	–	–	–	–	–	pF

Fonte: adaptado de Aguirre (2014)

O modelo de falhas utilizado emprega os mesmos tipos de falhas que o modelo proposto na Subseção 3.3.1, contendo, a priori, 12 falhas por transistor. São excluídas

Figura 15 – Diagrama esquemático do filtro analógico projetado em tecnologia IBM $0,13 \mu\text{m}$.



Fonte: adaptado de Aguirre (2014)

do modelo as falhas de curto-circuito entre os terminais de porta e dreno, quando este curto-circuito faz parte do projeto do circuito. Não foram incluídas falhas nos componentes passivos do filtro. É importante notar que apesar de serem necessários quatro FDAs, pode-se utilizar o mesmo circuito de polarização para todos. Em nível de blocos, então, injetam-se falhas em quatro blocos de amplificadores operacionais (204 falhas por bloco), oito blocos de realimentação de modo comum (94 falhas por bloco), e um bloco de polarização (153 falhas no bloco). O modelo de falhas contempla um total de 1721 falhas.

Simulou-se a aplicação do teste AC e construiu-se o dicionário de falhas para o filtro em malha aberta. Apesar das simulações AC apresentarem bons resultados, posteriores simulações transientes revelaram que o circuito estava operando fora da faixa linear. Este resultado é decorrência direta da existência de polos sobre o eixo imaginário do plano complexo, causando uma amplificação exagerada de tensões de *offset* e, portanto, saturação das tensões internas dos FDAs.

Foi proposta, então, uma realimentação negativa do filtro a fim de remover os polos existentes no eixo imaginário. A realimentação proposta, ilustrada em destaque na Figura 16, utiliza dois resistores iguais aos resistores de entrada de sinal, R_1 , portanto, tem-se uma realimentação unitária.

Simulou-se novamente a aplicação do teste AC, com faixa de frequência entre 10 Hz e 100 GHz e amplitude de 0,1 V, garantindo-se a operação linear do circuito. Criou-se o dicionário de falhas completo para o teste AC, que foi recurso para comparação de diferentes conjuntos de testes propostos.

4 Resultados e Discussões

Este Capítulo apresenta os resultados das análises descritas no Capítulo 3, assim como inicia discussões sobre tais resultados, com justificativas para as decisões tomadas. Os primeiros resultados do trabalho consistem do bom funcionamento da ferramenta de automatização das simulações SPICE desenvolvida. O Capítulo segue com a apresentação dos resultados da metodologia de construção do dicionário de falhas AC, automatizado pela ferramenta recém validada, como proposta de padronização dos testes em circuitos integrados analógicos, mais especificamente circuitos analógicos lineares. São feitas considerações sobre quando a metodologia é aplicável.

4.1 Validação da Ferramenta de Automatização Desenvolvida

A validação da ferramenta de automatização de injeção de falhas, execução de simulações SPICE e construção do dicionário de falhas dá-se pela reprodução dos resultados apresentados em Bender (2015). Os resultados utilizados para validação consistem de partes do dicionário completo de falhas e da cobertura de falhas total de cada teste realizado pela autora em dois estudos de caso distintos.

4.1.1 Amplificadores Diferenciais de dois Estágios com Saída Simples

Os primeiros dois circuitos¹ utilizados para validação da ferramenta de automatização são relativamente simples, e contêm poucos componentes, possibilitando a descrição de forma detalhada do processo de construção do dicionário de falhas pela autora. Isso possibilitou uma fase de *debugging* da ferramenta de injeção de falhas muito rápida.

A autora apresenta as tabelas² contendo as tensões nodais dos pontos de operação DC do circuito sem falhas e de alguns circuitos com falha. Assim, a identificação de erros nas descrições de circuitos com falhas criados pela ferramenta de automatização foi facilitada, pois sabe-se diretamente qual tipo de falha e em qual componente estava sendo descrita de forma errada. Na próxima Subseção fica claro que quando só se tem a informação da cobertura total de falhas, a etapa de *debugging* da ferramenta é mais demorada.

A Tabela 6 apresenta os valores obtidos pela ferramenta de automatização, após todos os erros serem corrigidos. Nota-se que os valores da Tabela 6 são idênticos aos valores

¹ BENDER, I. D. *Teste de Amplificadores Diferenciais através de Medida DC e Transiente de Tensões Internas de Polarização*. 2015. Dissertação, Mestrado em Microeletrônica, Programa de Pós-Graduação em Microeletrônica. Universidade Federal do Rio Grande do Sul.

² Ibidem, p. 34-35.

apresentados pela autora do primeiro estudo³, confirmando o funcionamento adequado da ferramenta de automatização.

Tabela 6 – Tensões obtidas pela ferramenta de automatização para falhas em M_1 : análise das topologias PMOS-DA e NMOS-DA.

Nó	Topologia e Falha							Unidade
	<i>Fault Free</i>	OG	PMOS-DA				SDS	
			SGS	OD	SDG	OS		
1	1,0187	1,0187	1,0187	1,0187	1,0187	1,0187	1,0187	V
3	1,4265	1,4265	0,0001	2,4763	1,4224	2,4915	-0,6641	V
4	-0,9485	-0,9485	-1,8574	-1,3029	-0,0003	-1,3754	-0,6641	V
5	-0,9485	-0,9485	-1,8574	2,4731	-2,4132	2,4903	-2,5000	V
<i>out</i>	-2,364	-2,364	2,4946	-2,4510	2,5000	-2,451	2,5000	V
	<i>Fault Free</i>	OG	NMOS-DA				SDS	
			SGS	OD	SDG	OS		
1	-1,0673	-1,0673	-1,0673	-1,0673	-1,0673	-1,0673	-1,0673	V
3	-1,2716	-1,2716	-0,0001	-2,4699	-1,2726	-2,4903	1,2232	V
4	1,3438	1,3438	2,0674	1,5412	0,0016	1,6049	1,2233	V
5	1,3438	1,3438	2,0674	-2,4620	2,4756	-2,4880	2,5000	V
<i>out</i>	-2,211	-2,211	-2,5000	2,4577	-2,5000	2,4578	-2,5000	V

Fonte: elaborado pelo autor.

Cabe ainda notar que o modelo de falhas utilizado continha apenas falhas catastróficas, portanto nessa etapa ainda não foi validada a injeção de falhas paramétricas. Isso porque as falhas catastróficas são modeladas pela inserção de resistores na descrição em SPICE do circuito, enquanto as falhas paramétricas são simuladas pela alteração do valor atribuído ao parâmetro em questão. Essa diferença implica em métodos diferentes de leitura e modificação da descrição do circuito *fault free*.

4.1.2 Amplificador Totalmente Diferencial de um Estágio

No Capítulo 4 da dissertação⁴ a autora apresenta quatro análises de testes diferentes para o FDA projetado em Oliveira, Severo e Girardi (2014). Julgou-se, neste trabalho, suficiente reproduzir os resultados obtidos na Subseção 4.4.2⁵, a fim de validar o bom funcionamento da ferramenta de injeção de falhas. Na referida Subseção é empregado um modelo de falhas catastróficas e paramétricas, assim como faz-se o uso de subcircuitos na descrição em SPICE do circuito. Dessa forma verifica-se a generalidade da ferramenta de injeção de falhas automática quanto aos tipos de falhas e formas de descrição do circuito.

³ Ibidem.

⁴ Ibidem, p. 40-58

⁵ Ibidem, p. 48-52

As primeiras simulações realizadas pela ferramenta de automatização apresentaram resultados diferentes dos obtidos pela autora. Como não há na dissertação uma descrição mais detalhada dos resultados, fez-se necessária a comparação de todos os arquivos de simulações criados pela autora e pela ferramenta automática. Este passo foi repetido diversas vezes até certificar-se do correto funcionamento da ferramenta. Finalmente, as coberturas de falhas obtidas pela ferramenta, para os considerados três principais nós do amplificador, são apresentadas na Tabela 7. Nota-se que cobertura de falhas do nó V_{cmc} é idêntica à obtida pela autora.

Tabela 7 – Cobertura de falhas dos nós principais do FDA projetado em tecnologia XFAB $0,18\ \mu\text{m}$.

Teste	Nó			Unidade
	V_{o+}	V_{o-}	V_{cmc}	
DC1	80,00	80,00	58,33	%
DC1 \cup DC2	80,00	83,33	73,33	%
DC1 \cup DC2 \cup DC3	80,00	83,33	78,33	%

Fonte: elaborado pelo autor.

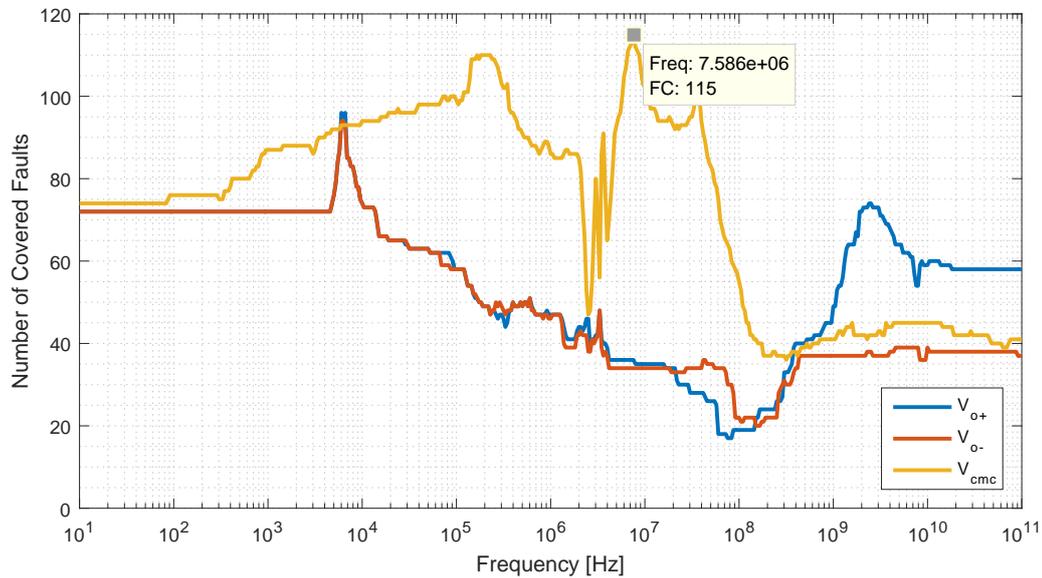
Uma vez considerada funcional a ferramenta de automatização de injeção de falhas, passou-se a utilizá-la para fazer a avaliação do teste AC. A Figura 17 mostra a cobertura de falhas obtida em cada configuração de teste. O teste AC simulado resume-se a aplicar um sinal AC com 1 mV de amplitude nas entradas do FDA, estando este em malha aberta, assim como nos teste DC anteriores.

Nota-se que o dicionário de falhas completo do teste AC permite ao engenheiro de teste uma avaliação rápida de qual configuração de teste deve ser empregada. Por exemplo, pela Figura 17 percebe-se que monitorando a amplitude da tensão no nó de saída do bloco de CMFB do FDA, e aplicando-se um sinal de entrada com frequência de 75,86 kHz, alcança-se uma cobertura de falhas de 95,83%, valor consideravelmente mais elevado que os 78,33% obtidos se aplicados os três testes DC anteriores.

Adicionalmente, verifica-se que para esse circuito, dado o modelo de falhas utilizado, duas frequências de teste são suficientes para obter-se a cobertura completa de falhas, monitorando apenas o nó V_{cmc} . Chama-se de conjunto de testes, portanto, a combinação de configurações de testes proposta, que é apresentada na Tabela 8.

É importante salientar que apesar do teste em malha aberta não ser o mais robusto em termos práticos (ruído e *offset* de entrada poderiam inviabilizar as medidas), ainda é válido em ambiente de simulação como prova de conceito de que a criação de um dicionário de falhas amplo é útil para a escolha de um conjunto de testes.

Figura 17 – Cobertura de falhas para diferentes nós do FDA projetado em tecnologia XFAB 0,18 μm .



Fonte: elaborado pelo autor.

Tabela 8 – Conjunto de testes AC para o FDA projetado em tecnologia XFAB 0,18 μm .

Frequência de teste	Nó de teste
7,5858 MHz	V_{cmc}
3,6308 MHz	V_{cmc}

Fonte: elaborado pelo autor.

4.2 Estudo de Caso: Modulador Sigma-Delta

Ainda na Seção anterior, verificam-se as vantagens da criação do dicionário de falhas do teste AC, no entanto para um circuito pequeno e com um modelo de falhas reduzido. A fim de avaliar esta metodologia de teste, propõe-se aplicá-lo no filtro analógico de terceira ordem contido no laço principal do modulador sigma-delta projetado em Aguirre (2014), incrementando-se também o modelo de falhas.

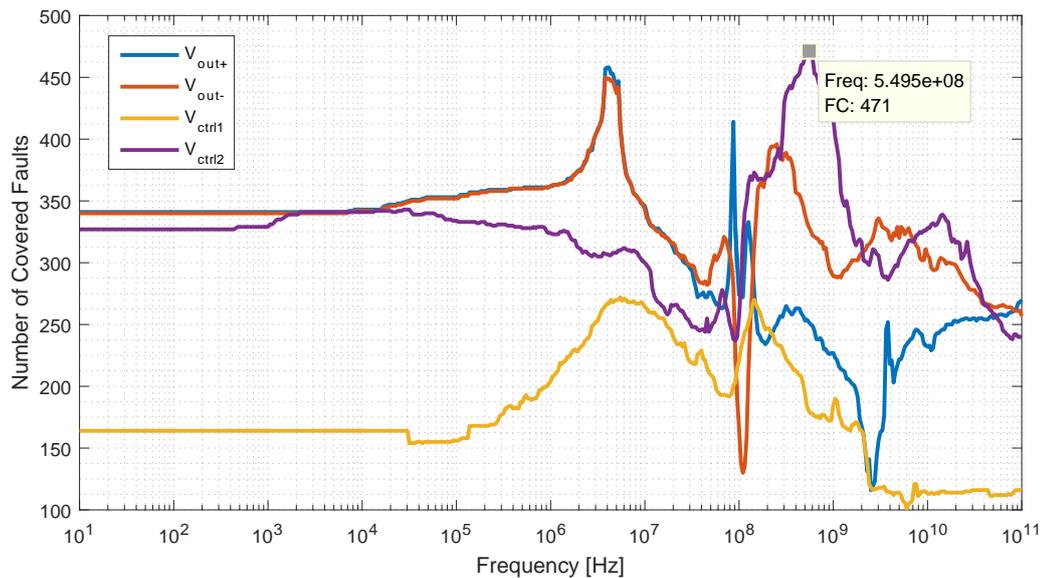
Em um primeiro momento, estuda-se o FDA utilizado para realização do filtro de forma isolada, e verifica-se a aplicabilidade do teste neste. Estuda-se, então, a possibilidade de aplicação do teste no filtro completo. Avalia-se inviável a aplicação do teste sem realizar alterações na topologia do laço, por este conter integradores sem perdas (polos sobre o eixo

imaginário). Então, propõem-se uma realimentação minimamente invasiva, que necessita de apenas dois resistores iguais aos resistores já presentes na topologia. Os resultados do dicionário de falhas do teste AC aplicado ao laço realimentado são apresentados e discutidos no final deste Capítulo.

4.2.1 Amplificador Totalmente Diferencial de dois Estágios

Foi criado o dicionário de falhas do teste AC para os quatro principais nós do FDA, V_{out+} , V_{out-} , V_{ctrl1} , V_{ctrl2} . A figura 18 mostra a cobertura de falhas de cada nó para a faixa de frequência testada. Escolhendo-se a melhor configuração única de teste, é possível detectar 471 das 545 falhas modeladas (86,42%). Para tanto, deve-se aplicar um sinal de teste com frequência de 549,5 MHz, e monitorar a amplitude da tensão do nó de saída do bloco de CMFB final, V_{ctrl2} .

Figura 18 – Cobertura de falhas para diferentes nós do FDA projetado em tecnologia IBM 0,13 μm .



Fonte: elaborado pelo autor.

A Tabela 9 apresenta o conjunto de testes cuja cobertura de falhas é máxima, aplicando-se frequências de teste entre 1 MHz e 1 GHz. No caso desse circuito, a cobertura de falhas máxima é de 99,63%, não podendo serem observadas as falhas de desvio positivo do comprimento do canal de M4 e negativo da largura do canal de M6, ambos do segunda bloco de CMFB.

Como o teste AC se baseia na medida da amplitude das tensões nodais em regime permanente, o tempo de teste, obviamente, depende do tempo de estabilização de cada

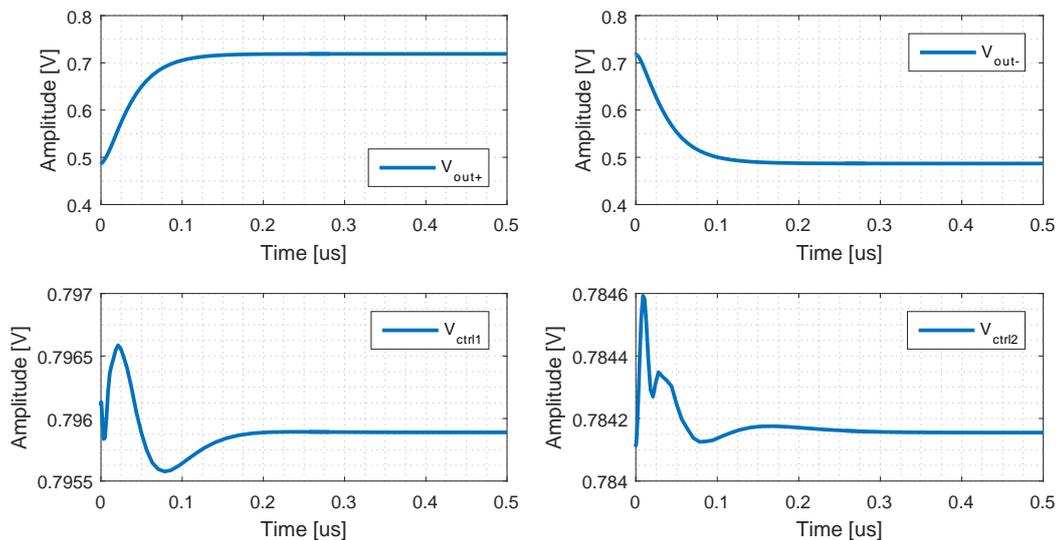
Tabela 9 – Conjunto de testes AC para o FDA projetado em tecnologia IBM 0,13 μm .

Frequência de teste	Nó de teste
3,9811 MHz	V_{out+}
13,1826 MHz	V_{ctrl1}
87,0964 MHz	V_{out+}
125,8925 MHz	V_{out+}
630,9574 MHz	V_{out-}
912,0108 MHz	V_{ctrl2}

Fonte: elaborado pelo autor.

nó. A Figura 19 mostra a resposta temporal das tensões dos nós de interesse à aplicação de uma onda quadrada, com amplitude de 1 mV, na entrada do FDA. Fica claro que, independente do nó de medida, todas as tensões estão em regime permanente a partir de 0,5 μs .

Figura 19 – Resposta ao degrau dos nós de interesse do FDA projetado em tecnologia IBM 0,13 μm .



Fonte: elaborado pelo autor.

Sabendo que 0,5 μs é um intervalo de tempo suficientemente grande para todas as tensões de interesse entrarem em regime permanente, somou-se o tempo de cinco ciclos completos da frequência de teste mais lenta, 3,9811 MHz, para se obter o tempo de teste

necessário para a aplicação de cada vetor de teste, T_{tpv} , resultando em aproximadamente $2 \mu s$, pela equação 4.1. Portanto, o tempo de teste total, considerando uma margem de segurança grande, é de apenas $12 \mu s$, pois seriam aplicados seis vetores de teste consecutivos. A Figura 20 mostra o resultado simulado das tensões dos nós de teste para suas respectivas frequências de teste. Nota-se claramente que todos as tensões completam vários ciclos, possibilitando uma medida precisa pelo equipamento de teste.

$$T_{tpv} = \max_{1 \leq i \leq n} T_s(i) + \frac{5}{\min_{1 \leq j \leq m} F_t(j)} \quad (4.1)$$

onde $T_s(i)$ é o tempo de acomodação da resposta ao degrau da tensão do i -ésimo nó de interesse, n é o número de nós de interesse e $F_t(j)$ é a frequência de teste do j -ésimo vetor de teste do conjunto de m vetores de teste.

Ainda a tempo, nota-se que curiosamente, caso se queira a maior cobertura de falhas possível com o menor número de configurações de teste, o conjunto de testes não inclui a frequência de 549,5 MHz, como pode ser visto na Tabela 9. Isso porque aquele teste não detecta as falhas mais sutis, apesar de detectar grande parte das outras falhas. Dessa forma, aplicando-se diferentes testes que detectam falhas mais difíceis de serem observadas, acaba-se dispensando testes que detectam falhas mais grosseiras. Conclui-se, portanto, que para minimizar o tempo de teste, i.e. minimizar o número de testes a serem aplicados, deve-se minimizar também a intersecção entre os conjuntos de falhas detectáveis de cada teste.

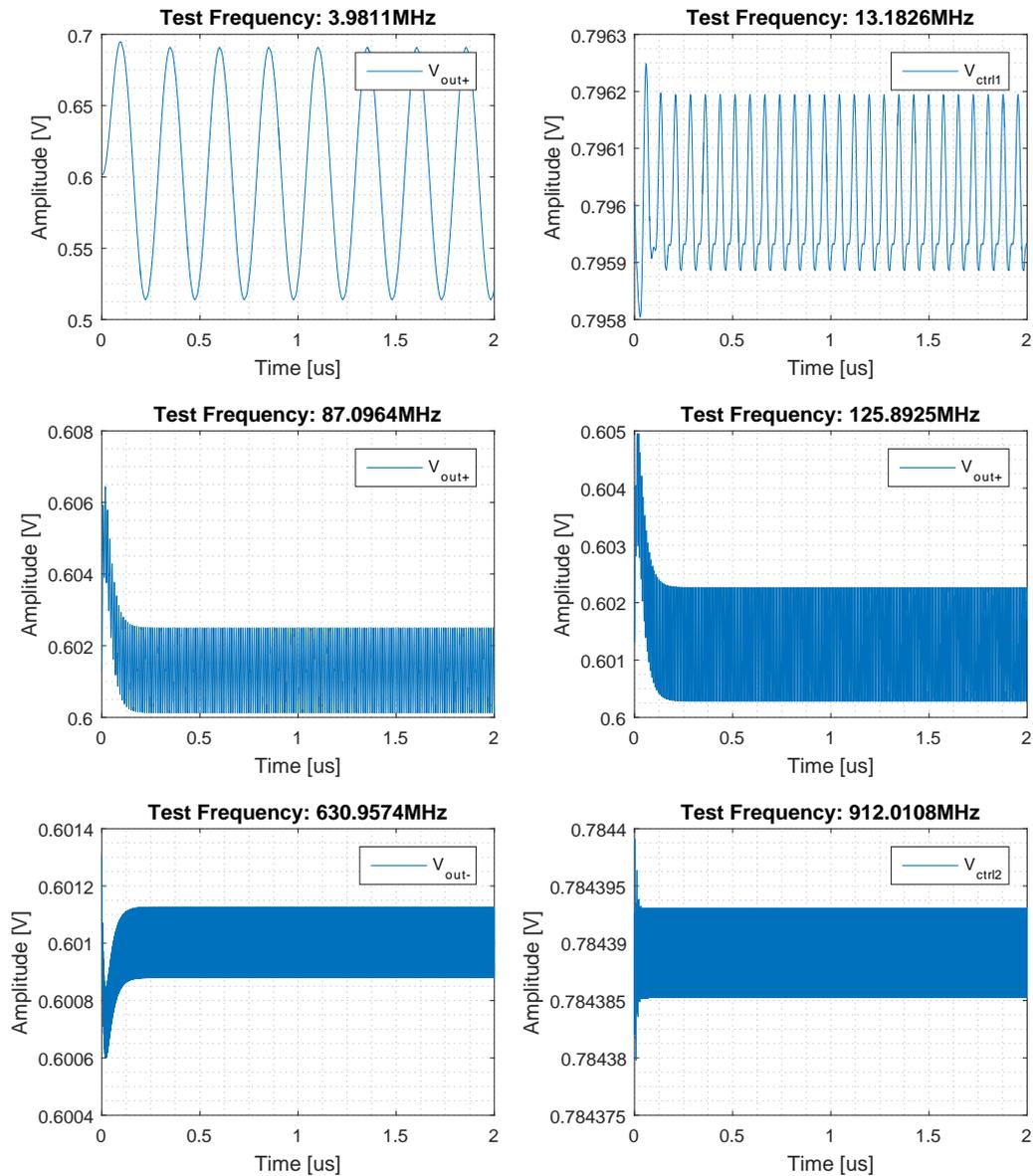
4.2.2 Filtro Analógico do Modulador Sigma-Delta

De forma análoga à análise de cada um dos FDAs isoladamente, construiu-se também um dicionário de falhas do teste AC aplicado no laço principal do modulador sigma-delta. Os resultados da cobertura de falhas para os quatro principais nós do último FDA do laço são apresentados da Figura 21.

Apesar da análise AC aparentar ter bons resultados, podendo-se observar até 65,77% das falhas, de um total de 1721, as simulações da resposta transiente mostram que o circuito, em laço aberto, não opera em sua região linear, como mostra a Figura 22. Portanto, o teste AC não é apropriado para o teste do laço principal em malha aberta. O comportamento instável do sistema se dá, provavelmente, pela existência de polos sobre o eixo imaginário, fazendo com que tensões DC (*offset* nos sinais intermediários) sejam amplificadas, saturando os amplificadores e causando um comportamento não linear.

Como forma de estabilizar o sistema, propôs-se realimentá-lo com resistores de valor igual aos resistores de entrada do sinal. O resultado da simulação transiente para a tensão de saída do filtro com laço de realimentação, Figura 23, é estável e condiz com o ganho DC, de -10.1 dB (ver Figura 25, Apêndice P.), previsto pela resposta em frequência.

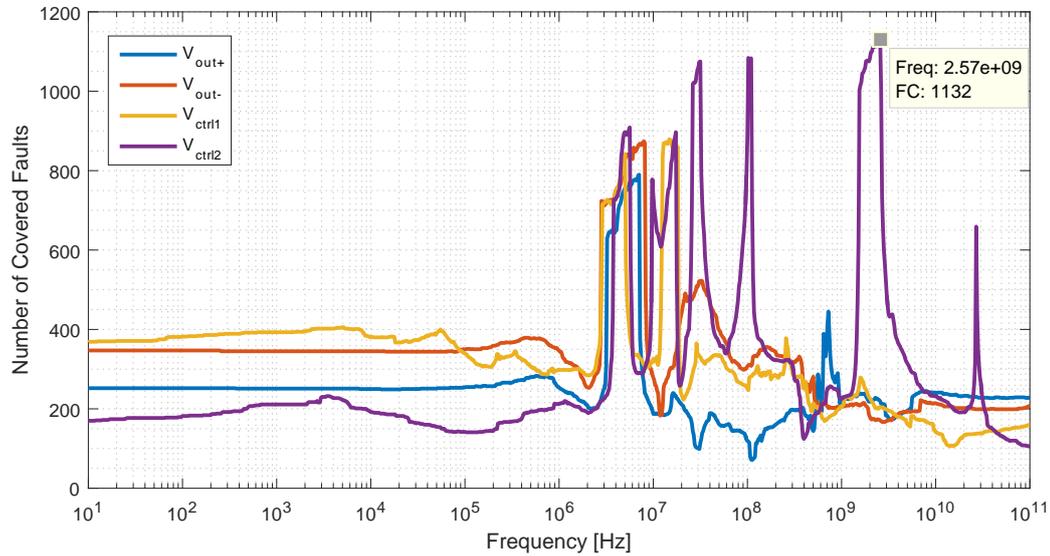
Figura 20 – Resultado da aplicação do conjunto de testes proposto para o FDA projetado em tecnologia IBM 0,13 μm .



Fonte: elaborado pelo autor.

Verificada a resposta linear do sistema, construiu-se um novo dicionário de falhas do teste AC. A Figura 24 mostra a cobertura de falhas dos nós principais de cada um dos quatro amplificadores utilizados para implementação do filtro. O nó que apresenta a maior CF para um teste único é o nó de saída inversora do segundo FDA, com um sinal

Figura 21 – Cobertura de falhas para diferentes nós do filtro em malha aberta do SDM projetado em tecnologia IBM 0,13 μm .

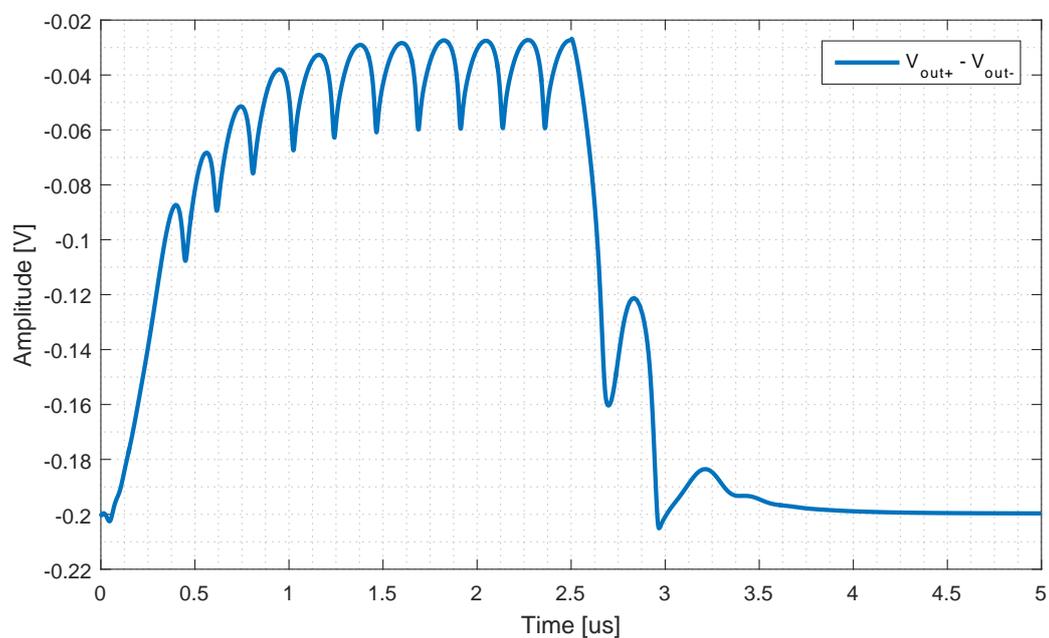


Fonte: elaborado pelo autor.

de entrada com frequência de 123 MHz. Esse teste observa 1192 das 1721 falhas modeladas (69.26%).

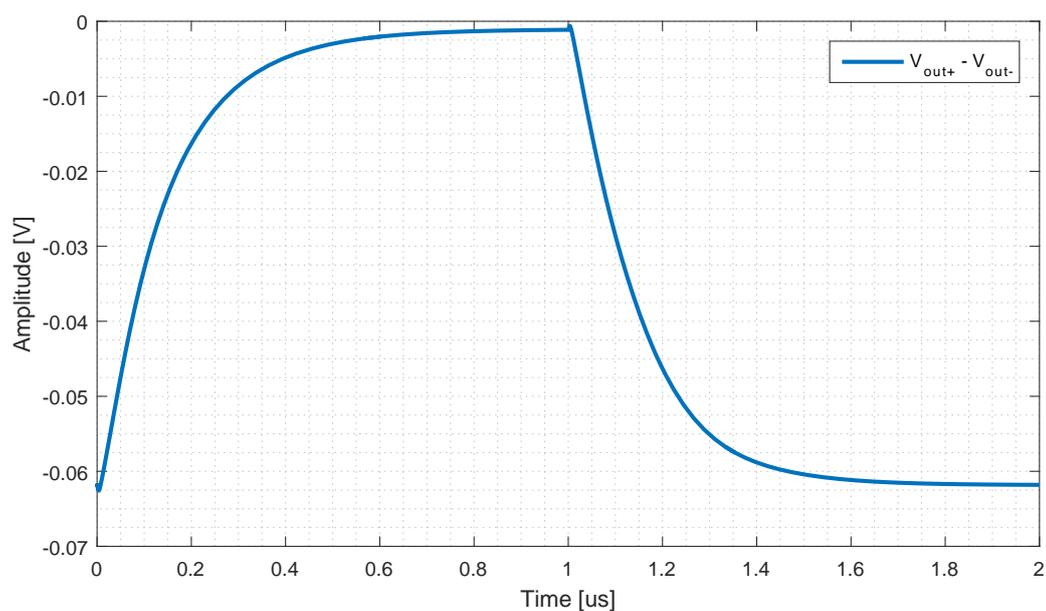
De posse do dicionário de falhas, contendo 16 diferentes nós de teste e 1000 frequências de teste, propõe-se seis conjuntos de testes diferentes. Para a definição de cada conjunto de testes se estipulou quais tensões podem ser medidas, assim como uma faixa de frequência de testes. O conjunto resultante é aquele que proporciona a maior cobertura de falhas com o menor número de configurações de teste. Para o cálculo do tempo de teste, assumiu-se uma espera de $10 \mu\text{s}$ (ver Figuras 26 e 27, Apêndice P) mais cinco períodos da menor frequência de teste contida no respectivo conjunto por configuração de teste.

Figura 22 – Resposta ao degrau da tensão de saída do filtro em malha aberta do SDM projetado em tecnologia IBM 0,13 μm .



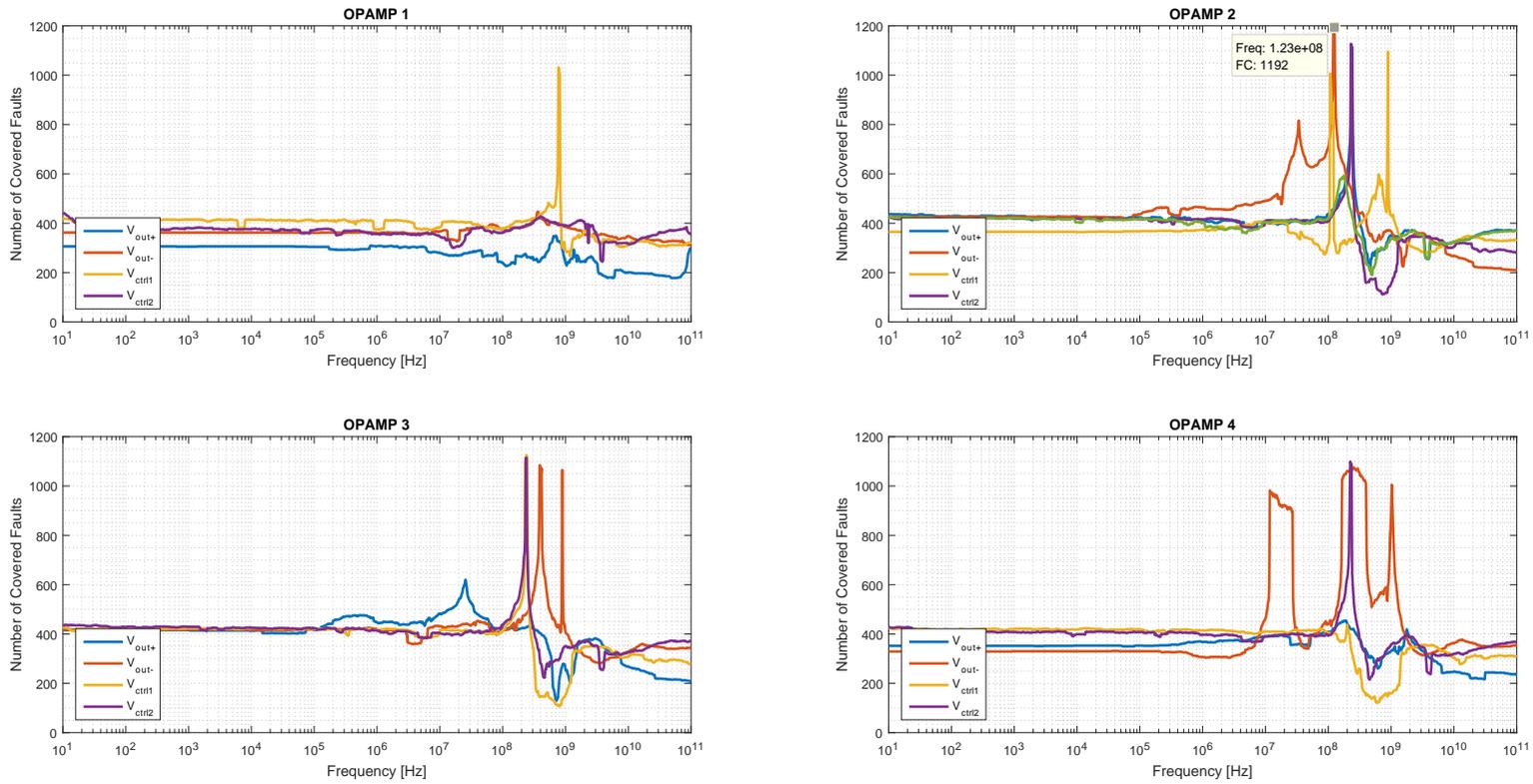
Fonte: elaborado pelo autor.

Figura 23 – Resposta ao degrau da tensão de saída do filtro em malha fechada do SDM projetado em tecnologia IBM 0,13 μm .



Fonte: elaborado pelo autor.

Figura 24 – Cobertura de falhas para diferentes nós do filtro em malha fechada do SDM projetado em tecnologia IBM 0,13 μm .



Fonte: elaborado pelo autor.

4.2.2.1 Conjunto de Testes Completo

Denominou-se conjunto de testes completo este que assume a possibilidade de medida de todos os 16 nós principais do filtro, na faixa de frequência de 10 Hz a 100 GHz. As configurações de teste são apresentadas na Tabela 10, assim como a cobertura de falhas, o tempo de teste e o número de pontos de teste necessários.

Tabela 10 – Conjunto de testes AC completo para o filtro do SDM projetado em tecnologia IBM 0,13 μm .

Frequência de teste	Nó de teste
107,1519 MHz	V_{out-} no FDA 2
120,2264 MHz	V_{out-} no FDA 2
120,2264 MHz	V_{out+} no FDA 2
257,0396 MHz	V_{out-} no FDA 4
812,8305 MHz	V_{ctrl1} no FDA 1
891,2509 MHz	V_{out-} no FDA 3
Resumo	
Cobertura de Falhas	100%
Tempo total de teste	60,28 μs
Pontos de teste	5

Fonte: elaborado pelo autor.

Nota-se que, apesar da frequência de 120,2264 MHz ser utilizada duas vezes, o tempo de teste assumido conservadoramente é de 60,28 μs . Isso porque se assume que apenas uma tensão possa ser medida por vez, no caso de um equipamento de teste que disponha de apenas uma ponteira de prova.

4.2.2.2 Conjunto de Testes via Tensões de Saída na faixa de 1 MHz a 1 GHz

O segundo conjunto de testes proposto assume a possibilidade de medida de apenas as duas tensões de saída do último FDA. A faixa de frequência do sinal de teste também foi reduzida para 1 MHz a 1 GHz. As configurações de teste são apresentadas na Tabela 11, assim como o resumo do teste.

Utilizando apenas os nós de saída do filtro a cobertura de falhas cai em aproximadamente 5%, mesmo que o tempo de teste aumente 38%. Percebe-se, portanto, que existe um *trade-off* entre as três variáveis, cobertura de falhas, tempo de teste e adaptação do circuito às medidas.

Tabela 11 – Conjunto de testes AC via tensões de saída para o filtro do SDM projetado em tecnologia IBM 0,13 μm .

Frequência de teste	Nó de teste
11,7490 MHz	V_{out-} no FDA 4
17,7828 MHz	V_{out-} no FDA 4
26,9153 MHz	V_{out-} no FDA 4
165,9587 MHz	V_{out-} no FDA 4
208,9296 MHz	V_{out+} no FDA 4
257,0396 MHz	V_{out-} no FDA 4
398,1072 MHz	V_{out-} no FDA 4
1,0000 GHz	V_{out-} no FDA 4
Resumo	
Cobertura de Falhas	95,24%
Tempo total de teste	83.40 μs
Pontos de teste	2

Fonte: elaborado pelo autor.

Nota-se também que o nó de saída não inversor é utilizado apenas uma vez dentro de um conjunto de oito medidas. Dessa forma, assume-se que ele é muito menos sensível a falhas do que seu complementar. O próximo conjunto de testes é baseado nesta suposição.

4.2.2.3 Conjunto de Testes via Tensão de Saída Inversora na faixa de 1 MHz a 1 GHz

O terceiro conjunto de testes foi proposto com base nos resultados obtidos pelo conjunto de testes que utiliza as tensões de saída do filtro analógico do SDM em questão. Assume-se que a amplitude do nó inversor seja muito mais sensível às falhas do que a amplitude do nó não inversor pela análise da Tabela 11, onde o nó não inversor é monitorado em apenas uma das oito configurações de teste. Dessa forma, propõe-se medir apenas a tensão de saída inversora do filtro, na mesma faixa de frequência de 1 MHz a 1 GHz. As configurações de teste são apresentadas na Tabela 12, assim como o resumo do teste.

Ao ser reduzido pela metade o número de pontos de teste, diminui-se também o tempo de teste em 12,5%, ao custo de 103 falhas não serem detectadas com relação ao conjunto de testes anterior. Nota-se também que todas as configurações de teste contidas neste conjunto são configurações do teste anterior também. Excluiu-se apenas a configuração de teste que utilizava a tensão de saída não inversora. Pode-se concluir,

Tabela 12 – Conjunto de testes AC via tensão de saída inversora do filtro do SDM projetado em tecnologia IBM 0,13 μm .

Frequência de teste	Nó de teste
11,7490 MHz	V_{out-} no FDA 4
17,7828 MHz	V_{out-} no FDA 4
26,9153 MHz	V_{out-} no FDA 4
165,9587 MHz	V_{out-} no FDA 4
257,0396 MHz	V_{out-} no FDA 4
398,1072 MHz	V_{out-} no FDA 4
1,0000 GHz	V_{out-} no FDA 4
Resumo	
Cobertura de Falhas	95,18%
Tempo total de teste	72,98 μs
Pontos de teste	1

Fonte: elaborado pelo autor.

portanto, que as 103 falhas detectadas pelo nós de saída não inversora não podem ser detectadas pela tensão do nó de saída inversora nesta faixa de frequência.

4.2.2.4 Conjunto de Testes pela Análise dos Gráficos de CF x Frequência x Nó na faixa de 1 MHz a 1 GHz

O quarto conjunto de testes propõe o monitoramento das amplitudes dos sinais de saída inversora dos OPAMPs 2 e 4, assim como do sinal de saída do segundo bloco de CMFB do OPAMP 4. As escolhas dos nós foram feitas pela análise da Figura 24, pelo critério de 'área embaixo da curva' verificado 'a olho'. A motivação para a proposta é verificar se apenas com uma análise rápida do gráfico de CF x Frequência x Nó é possível escolher um conjunto de testes com um bom equilíbrio entre CF, tempo de teste e número de pontos de teste. A faixa de frequência do sinal de teste permanece igual àquela do último conjunto de testes, de 1 MHz a 1 GHz. As configurações de teste são apresentadas na Tabela 13, assim como o resumo do teste.

Atingindo a cobertura completa das 1721 falhas modeladas, nota-se que este conjunto de testes é uma comparação direta com o primeiro conjunto proposto. Enquanto aquele utiliza menos (-40%) pontos de teste, este consome menos (-35.6%) tempo, ambos obtendo a mesma cobertura de falhas. Cabe portanto ao engenheiro de teste avaliar os

Tabela 13 – Conjunto de teste pela análise dos gráficos de CF x Frequência x Nó do filtro do SDM projetado em tecnologia IBM 0,13 μm .

Frequência de teste	Nó de teste
12,5893 MHz	V_{out-} no FDA 4
107,1519 MHz	V_{out-} no FDA 2
120,2264 MHz	V_{out-} no FDA 2
208,9296 MHz	V_{out-} no FDA 4
234,4229 MHz	V_{ctrl2} no FDA 4
257,0396 MHz	V_{out-} no FDA 4
645,6542 MHz	V_{out-} no FDA 2
891,2509 MHz	V_{out-} no FDA 2
1,0000 GHz	V_{out-} no FDA 4

Resumo	
Cobertura de Falhas	100%
Tempo total de teste	93,57 μs
Pontos de teste	3

Fonte: elaborado pelo autor.

custos de se adaptar o circuito a fim de poderem ser medidas mais tensões e comparar com o ganho obtido pelo aumento do *throughput* da etapa de teste.

5 Conclusões

Este trabalho apresentou o desenvolvimento de uma ferramenta de automatização de injeção e simulação de falhas, assim como posterior análise dos resultados das simulações. Esta ferramenta se mostrou de muita valia por simplificar e agilizar o processo de injeção de falhas e simulação de circuitos eletrônicos, processo esse que se faz necessário para criação de um dicionário de falhas. A validação da ferramenta foi realizada pela reprodução idêntica de resultados passados, obtidos por Bender (2015), no seu trabalho final de mestrado. Não se possui informação precisa sobre o tempo gasto pela autora para a criação de todos os arquivos de simulação, nem para a análise manual dos resultados, impossibilitando uma avaliação quantitativa da aceleração obtida pela ferramenta. Pode-se concluir, no entanto, que o tempo de criação do dicionário de falhas completo é dominado pelo tempo de simulação de todos os circuitos, sendo esse tempo, a priori, irreduzível.

Feita a validação dos resultados obtidos pela ferramenta, propôs-se uma metodologia de seleção de conjuntos de vetores de teste baseada na resposta em frequência de diversos nós do CUT, levando em consideração que este deve operar de forma linear. A metodologia foi aplicada com sucesso em três circuitos diferentes. O primeiro circuito é um FDA, projetado em Oliveira, Severo e Girardi (2014), contendo 14 transistores. Foi definido um modelo de falhas contendo seis falhas catastróficas por transistor, além de três falhas paramétricas. Excluiu-se do modelo seis falhas que se mostraram indetectáveis por fazerem parte do circuito sem falhas ou pelo circuito simulado utilizar tensões de polarização ideais ou por violarem as regras de projeto. O modelo de falhas reduzido abrange 120 falhas. A metodologia de análise da resposta em frequência mostrou a possibilidade de aplicação de um único vetor de teste capaz de observar 95,83% das falhas, medindo-se a tensão de apenas um nó. Adicionando-se um segundo vetor de teste, obteve-se a cobertura completa das falhas.

O segundo circuito no qual foi empregada a metodologia baseada na resposta em frequência foi o FDA projetado em Aguirre (2014). Este FDA foi implementado com 46 transistores. O modelo de falhas por transistor foi estendido para seis falhas catastróficas e seis falhas paramétricas, e excluíram-se do modelo as falhas de curto-circuito presentes na topologia sem falhas. O modelo de falha final contém 545 falhas, das quais 471 (86,42%) podem ser detectadas pela aplicação de um único vetor de teste, monitorando-se apenas a tensão de um nó. A cobertura de falhas máxima para esse circuito, considerando testes na faixa de frequência entre 1 MHz e 1 GHz e acesso a quatro dos nós internos, é de 99,63%, aplicando-se um conjunto de seis vetores de teste. O tempo total do teste foi estimado em 12 μ s.

O terceiro circuito, e principal, é o filtro analógico utilizado também em Aguirre (2014) para o projeto de um modulador sigma-delta. O filtro é composto por uma cascata de integradores, com realimentação interna, e caminhos de *feedforward*. Enfatizou-se em uma primeira análise que o método proposto é limitado a circuitos lineares e assintoticamente estáveis, quando por meio de análise transiente notou-se que o circuito saía de sua faixa de operação linear pela existência de polos no eixo imaginário do plano complexo. Propôs-se a inclusão de dois resistores a fim de realimentar negativamente o circuito, estabilizando-o. O modelo de falhas por transistor é igual ao utilizado no segundo circuito, totalizando 1721 falhas simuladas para o terceiro. O vetor de teste único com maior cobertura de falhas é capaz de observar 1192 (69,26%). Foram discutidos diversos conjuntos de testes, escolhidos por diferentes restrições, que podem ser aplicados no teste do filtro. Foi possível obter-se a cobertura completa das 1721 falhas modeladas com um conjunto de testes que utiliza cinco pontos de teste e leva 60,28 μs para ser executado. Diminuindo-se o número de pontos de teste para três, escolhe-se um conjunto de testes diferente, que leva 93,57 μs para ser executado, e também proporciona a cobertura de falhas completa. Diminuindo-se para um único ponto de teste, nota-se que a cobertura de falhas cai para 95,18%, e o tempo de execução do conjunto de testes é de 72,98 μs .

Fica claro que a escolha do teste mais apropriado varia de acordo com as restrições particulares do circuito, como pontos de acesso à nós internos, e constitui um triângulo de *trade-off*, onde paga-se uma maior cobertura de falhas com maior tempo de teste ou maior número de pontos de teste.

Finalmente, resume-se como contribuições principais deste trabalho: 1) o desenvolvimento de uma ferramenta completamente automática de criação do dicionário de falhas baseado em um modelo de falhas para circuitos analógicos; e 2) proposta de metodologia de seleção otimizada de conjuntos de vetores de teste analógicos baseada na resposta em frequência do CUT.

6 Propostas de Trabalhos Futuros

Fica como sugestão de estudo a avaliação da viabilidade de roteamento, ou inserção de *micropads*, dos/nos nós internos, cujas tensões quer-se medir.

Propõe-se incrementar o trabalho aqui descrito pela redefinição das faixas de tolerância do teste *go/no-go* pela análise de Monte Carlo, que prova-se mais robusta, porém muito mais custosa computacionalmente, do que a análise de *corners* realizada.

A utilização do dicionário de falhas não binário, i.e contendo as tensões dos nós para cada configuração de teste e falha injetada, para um possível isolamento e diagnóstico de falhas é uma última sugestão de estudo complementar.

Referências Bibliográficas

- AGUIRRE, P. C. C. de. *Projeto e Análise de Moduladores Sigma-Delta em Tempo Contínuo Aplicados à Conversão AD*. 2014. Dissertação, Mestrado em Engenharia de Elétrica, Programa de Pós-Graduação em Engenharia Elétrica. Universidade Federal do Rio Grande do Sul. Citado 15 vezes nas páginas 7, 8, 19, 26, 36, 37, 38, 39, 40, 41, 42, 43, 47, 59 e 60.
- ALLANI, M. *Study of Process Variability on Performance and Power*. 2010. Report. Citado na página 34.
- ARSLAN, B.; ORAILOGLU, A. Tracing the best test mix through multi-variate quality tracking. In: *VLSI Test Symposium (VTS), 2013 IEEE 31st*. [S.l.: s.n.], 2013. p. 1–6. ISSN 1093-0167. Citado 2 vezes nas páginas 18 e 22.
- AUGUSTO, J. S.; ALMEIDA, C. F. B. Fully automatic dc fault dictionary construction and test nodes selection for analogue fault diagnosis. In: *European Design and Test Conference, 1995. ED TC 1995, Proceedings*. [S.l.: s.n.], 1995. p. 605. Citado na página 20.
- BAKER, R. J. *CMOS Circuit Design, Layout, and Simulation*. 3. ed. [S.l.]: Wiley-IEEE Press, 2010. ISBN 978-0-470-88132-3. Citado na página 37.
- BALEN, T. R. *Teste de Dispositivos Analógicos Programáveis (FPAAS)*. 2006. Dissertação, Mestrado em Engenharia de Elétrica, Programa de Pós-Graduação em Engenharia Elétrica. Universidade Federal do Rio Grande do Sul. Citado 2 vezes nas páginas 17 e 21.
- BENDER, I. D. *Teste de Amplificadores Diferenciais através de Medida DC e Transiente de Tensões Internas de Polarização*. 2015. Dissertação, Mestrado em Microeletrônica, Programa de Pós-Graduação em Microeletrônica. Universidade Federal do Rio Grande do Sul. Citado 10 vezes nas páginas 18, 19, 23, 26, 30, 31, 35, 39, 44 e 59.
- BORKAR, S. et al. Parameter variations and impact on circuits and microarchitecture. In: *Design Automation Conference, 2003. Proceedings*. [S.l.: s.n.], 2003. p. 338–342. Citado na página 34.
- BROSA, A. M.; FIGUERAS, J. On maximizing the coverage of catastrophic and parametric faults. In: *European Test Workshop 1999. Proceedings*. [S.l.: s.n.], 1999. p. 123–128. Citado 3 vezes nas páginas 20, 23 e 34.
- BUSHNELL, M. L.; AGRAWAL, V. D. *CMOS ELECTRONICS - HOW IT WORKS, HOW IT FAILS*. [S.l.]: Kluwer Academic Publishers, 2002. Citado na página 20.
- CARDOSO, G. S. *Impacto dos Desvios de Tensão de Limiar Induzidos por Radiação Ionizante no Desempenho dos Blocos Básicos de Dois Amplificadores Operacionais Complementares*. 2012. Dissertação, Mestrado em Microeletrônica, Programa de Pós-Graduação em Microeletrônica. Universidade Federal do Rio Grande do Sul. Citado 4 vezes nas páginas 19, 31, 32 e 33.

- DAVIS, B. *The Economics of Automatic Testing*. [S.l.]: McGraw Hill, 1982. Citado na página 20.
- 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. ISSN 0018-9456. Citado 4 vezes nas páginas 20, 22, 34 e 38.
- GALIAY, J.; CROUZET, Y.; VERGNIAULT, M. Physical versus logical fault models mos lsi circuits: Impact on their testability. *IEEE Transactions on Computers*, C-29, n. 6, p. 527–531, June 1980. ISSN 0018-9340. Citado na página 22.
- GREER, J.; KORKIN, A.; LABANOWSKI, J. *Nano and Giga Challenges in Microelectronics*. 1. ed. [S.l.]: Elsevier B.V., 2003. ISBN 0-444-51494-5. Citado na página 38.
- HALDER, A.; CHATTERJEE, A. Automated test generation and test point selection for specification test of analog circuits. In: *Quality Electronic Design, 2004. Proceedings. 5th International Symposium on*. [S.l.: s.n.], 2004. p. 401–406. Citado na página 20.
- HUGHES, J. L. A. Multiple fault detection using single fault test sets. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, v. 7, n. 1, p. 100–108, Jan 1988. ISSN 0278-0070. Citado na página 22.
- KABISATPATHY, P.; BARUA, A.; SINHA, S. *Fault Diagnosis of Analog Integrated Circuits*. 1. ed. [S.l.]: Springer, 2005. (Frontiers in Electronic Testing). ISBN 9780387257426,038725742X. Citado 2 vezes nas páginas 20 e 23.
- KARKI, J. *Fully-Differential Amplifiers*. 2002. Application Report SLOA054D. Citado 2 vezes nas páginas 24 e 25.
- 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. Citado na página 22.
- MILOR, L.; VISVANATHAN, V. Detection of catastrophic faults in analog integrated circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, v. 8, n. 2, p. 114–130, Feb 1989. ISSN 0278-0070. Citado na página 20.
- MILOR, L. S. A tutorial introduction to research on analog and mixed-signal circuit testing. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, v. 45, n. 10, p. 1389–1407, Oct 1998. ISSN 1057-7130. Citado na página 20.
- MILOR, L. S.; VISVANATHAN, V. Efficient go/no-go testing of analog circuits. In: *International Symposium on Circuits and Systems*. [S.l.: s.n.], 1987. Citado 2 vezes nas páginas 20 e 23.
- MOORE, G. E. Cramming more components onto integrated circuits. *Electronics*, p. 114–117, 1965. Citado na página 17.
- MOORE, G. E. Progress in digital integrated electronics. *Electron Devices Meeting*, v. 21, p. 11–13, 1975. Citado na página 17.
- NAGEL, L. W.; PEDERSON, D. *SPICE (Simulation Program with Integrated Circuit Emphasis)*. [S.l.], 1973. Citado na página 18.

- OLIVEIRA, A. C. de; SEVERO, L. C.; GIRARDI, A. *A Two-Step Methodology for Automatic Design of Fully Differential Amplifiers with Output Balance*. 2014. Citado 7 vezes nas páginas 7, 8, 32, 33, 34, 45 e 59.
- 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: 13/06/2016. Citado na página 29.
- PETRASHIN, P. et al. Low-cost dc bist for analog circuits: A case study. In: *2013 14th Latin American Test Workshop - LATW*. [S.l.: s.n.], 2013. p. 1–4. ISSN 2373-0862. Citado 2 vezes nas páginas 20 e 22.
- SCHREIER, R.; TEMES, G. C. *Understanding Delta-Sigma Data Converters*. 1. ed. [S.l.]: Wiley-IEEE Press, 2004. ISBN 978-0-471-46585-0. Citado na página 36.
- SEGURA, J.; HAWKINS, C. F. *CMOS ELECTRONICS - HOW IT WORKS, HOW IT FAILS*. 1. ed. [S.l.]: Wiley-IEEE Press, 2004. ISBN 0-471-47669-2. Citado na página 38.
- SOUDERS, T. M.; STENBAKKEN, G. N. A comprehensive approach for modeling and testing analog and mixed-signal devices. In: *Test Conference, 1990. Proceedings., International*. [S.l.: s.n.], 1990. p. 169–176. Citado 2 vezes nas páginas 18 e 21.
- SOUDERS, T. M.; STENBAKKEN, G. N. Cutting the high cost of testing. *IEEE Spectrum*, v. 28, n. 3, p. 48–51, March 1991. ISSN 0018-9235. Citado 2 vezes nas páginas 20 e 22.
- SYLVESTER, M. *Circuit Sizing w/ Corner Models Challenges and Applications*. 2013. Improve Design Performance and Yield MunEDA. Citado na página 24.
- TAUR, Y. et al. Cmos scaling into the nanometer regime. *Proceedings of the IEEE*, v. 85, n. 4, p. 486–504, Apr 1997. ISSN 0018-9219. Citado na página 17.
- 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. Citado na página 24.

Apêndices

APÊNDICE A – Algoritmo em MATLAB de Automatização do Teste DC

```

%% Add path to third party functions
addpath <path>/<to>/fi_tool
addpath <path>/<to>/HspiceToolbox

%% Edit the constants in this block to your needs !!

bsFileFullName = '<netlist>.sp';

[~, baseCircuitFileName, baseCircuitFileExt] = ...
fileparts(bsFileFullName);

% .fii file must be in the same directory and have the same name
% as the netlist file
FIIFilename = [baseCircuitFileName '.fii'];

%Test Definitions
%testFreqInterval must be within the simulated interval!
testFreqInterval = [<minTestFreq> <maxTestFreq>];
%Nodes which amplitudes will be stored in MATLAB environment
mainNodes = [
    cellstr('v_<node1>')
    cellstr('v_<node2>')
    cellstr('v_<node3>')
];
%Nodes selected to be in the final test vector. Refers to mainNodes indexes
testNodeID = [1 3];

% Mind the double quotes in the library path
libstr = '.lib '<path>\<to>\<lib>.lib''';
corner = [ cellstr('<corner1>')
           cellstr('<corner2>')
           cellstr('<corner3>')
           cellstr('<corner4>')
];

hspicePath = '<path>\<to>\<HSPICE>\BIN\hspice.bat';
delay = 0; %Time delay between simulations.
%Folder to where simulation files will be transferred
auxDir = 'faulty_and_corner_circuits';

```

```
%Although the test is DC, it should be applied a pulse in the power supply
analysisType = 'tran';

save('setting.mat', 'bsFileFullName', 'FIIFilename', 'testFreqInterval', ...
    'testNodeID', 'mainNodes', 'libstr', 'corner', 'hspicePath', ...
    'delay', 'auxDir', 'analysisType');

%% Reads base circuit file and apply corners and faults

cornerCir = applyCorner(bsFileFullName, corner, libstr);
faultyCir = injFaults(bsFileFullName, FIIFilename);

save('circuitNames.mat', 'cornerCir', 'faultyCir');

%% Runs simulations in HSpice

runSim( hspicePath, bsFileFullName, auxDir, ...
    cornerCir, faultyCir, delay, analysisType );

%% Stores the Frequency Response Results into MATLAB variables

[vNomAndCorner, vMax, vMin, vFault, faultMatrix] = ...
analyseDCResults(bsFileFullName, cornerCir, faultyCir, auxDir, mainNodes);

save('frResults.mat', 'vNomAndCorner', 'vMax', 'vMin', ...
    'vFault', 'faultMatrix');

%%

fprintf('\tFault Coverage\nNode name\tFC (%%)\n');
for n=1:length(mainNodes)
    fprintf('%s\t\t%.2f\n', char(mainNodes(n)), ...
        (sum(faultMatrix(n,:))/length(faultMatrix(n,:)))*100);
end

%%
rmpath <path>/<to>/fi_tool
rmpath <path>/<to>/HspiceToolbox
```

APÊNDICE B – Algoritmo em MATLAB de Automatização do Teste AC

```

%% Add path to third party functions
addpath <path>/<to>/fi_tool
addpath <path>/<to>/HspiceToolbox

%% Edit the constants in this block to your needs !!

bsFileFullName = '<netlist>.sp';

[~, baseCircuitFileName, baseCircuitFileExt] = ...
fileparts(bsFileFullName);

% .fii file must be in the same directory and have the same name
% as the netlist file
FIIFilename = [baseCircuitFileName '.fii'];

%Test Definitions
%testFreqInterval must be within the simulated interval!
testFreqInterval = [<minTestFreq> <maxTestFreq>];
%Nodes which amplitudes will be stored in MATLAB environment
mainNodes = [
    cellstr('v_<node1>')
    cellstr('v_<node2>')
    cellstr('v_<node3>')
];
%Nodes selected to be in the final test vector. Refers to mainNodes indexes
testNodeID = [1 3];

% Mind the double quotes in the library path
libstr = '.lib '<path>\<to>\<lib>.lib''';
corner = [ cellstr('<corner1>')
           cellstr('<corner2>')
           cellstr('<corner3>')
           cellstr('<corner4>')
];

hspicePath = '<path>\<to>\<HSPICE>\BIN\hspice.bat';
delay = 0; %Time delay between simulations.
%Folder to where simulation files will be transferred
auxDir = 'faulty_and_corner_circuits';

```

```
analysisType = 'ac';

save('setting.mat', 'bsFileFullName', 'FIIFilename', 'testFreqInterval', ...
    'testNodeID', 'mainNodes', 'libstr', 'corner', 'hspicePath', ...
    'delay', 'auxDir', 'analysisType');

%% Reads base circuit file and apply corners and faults

cornerCir = applyCorner(bsFileFullName, corner, libstr);
faultyCir = injFaults(bsFileFullName, FIIFilename);

save('circuitNames.mat', 'cornerCir', 'faultyCir');

%% Runs simulations in HSpice

runSim( hspicePath, bsFileFullName, auxDir, ...
        cornerCir, faultyCir, delay, analysisType );

%% Stores the Frequency Response Results into MATLAB variables

[f, vNomAndCorner, vMax, vMin, vFault, faultMatrix] = ...
analyseACResults(bsFileFullName, cornerCir, faultyCir, auxDir, mainNodes);

save('frResults.mat', 'f', 'vNomAndCorner', 'vMax', 'vMin', ...
    'vFault', 'faultMatrix');

%%

senseFreqNodeID = ...
    findMaxFCTestConfig( faultMatrix, f, testFreqInterval, testNodeID );
senseFreq = f(senseFreqNodeID(:,1));
senseNode = mainNodes(senseFreqNodeID(:,2));

save('senseFreqAndNode.mat', 'senseFreqNodeID', 'senseFreq', 'senseNode');

%%
rmpath C:/Users/Andre/Dropbox/TCC_20161/matlab_script
rmpath C:/HspiceToolbox
```

APÊNDICE C – Função em MATLAB para Alteração de *Corner*

```

function [ cornerCir, err ] = ...
    applyCorner(baseCircuitFileName, corner, libstr)
%APPLYCORNER Creates a netlist for each corner given

[~,bcFileName,bcFileExt] = fileparts(baseCircuitFileName);
[bcFID, err] = fopen(baseCircuitFileName, 'r');
if bcFID < 0
    display([baseCircuitFileName ': ' err]);
    cornerCir = [];
    return
end

for i=1:length(corner)
    cornerCir(i) = cellstr([bcFileName '_corner_' char(corner(i))]);
    cFID(i) = fopen([char(cornerCir(i)) bcFileExt], 'w');

    line = fgets(bcFID);
    fprintf(cFID(i), [strtok(line, ...
        sprintf('\r\n')) ' - at corner ' char(corner(i))]);

    headerMessage = ...
        sprintf('\n*This is an automatically generated circuit file.\n');
    fprintf(cFID(i), headerMessage);

    while ~feof(bcFID)
        line = fgets(bcFID);
        if ~isempty(strfind(line, libstr))
            fprintf(cFID(i), [libstr ' ' char(corner(i)) sprintf('\n')]);
        else
            fprintf(cFID(i), line);
        end
    end
    fprintf(cFID(i), ['\n*Created at ' datestr(now) '.\n']);
    fclose(cFID(i));
    frewind(bcFID);
end

fclose(bcFID);

```

APÊNDICE D – Função em MATLAB para Encaminhamento de Injeção de Falha

```

function [ faultyCir, err ] = injFaults( baseCircuitFileName, FIIFileName )
% Reads base circuit file and apply all faults

[~, baseCircuitFileName, baseCircuitFileExt] = ...
    fileparts(baseCircuitFileName);

[fiiFID, err] = fopen(FIIFileName, 'r');
if fiiFID < 0
    display([FIIFileName ': ' err]);
    faultyCir = [];
    return
end

i = 1;
while ~feof(fiiFID)
    line = fgets(fiiFID);
    [sbc, remainder] = strtok(line, '.');
    if isempty(remainder)
        [comp, remainder] = strtok(line, ' ');
        sbc = [];
    else
        [comp, remainder] = strtok(remainder(2:end), ' ');
    end
    [fault, remainder] = strtok(remainder, ' ');
    if strcmp(fault, 'open')
        while remainder
            [node1, remainder] = strtok(remainder, ' ');
            [r, err] = ...
                injFaultOpen([baseCircuitFileName baseCircuitFileExt], sbc, ...
                    comp, str2num(node1));
            if isempty(r)
                display(err);
                fclose(fiiFID);
                faultyCir = [];
                return
            end
            faultyCir(i) = cellstr(r);
            i = i + 1;
        end
    end
end

```

```
elseif strcmp(fault,'short')
    while remainder
        [node1, remainder] = strtok(remainder, ' ');
        [node2, remainder] = strtok(remainder, ' ');
        [r, err] = ...
injFaultShort([baseCircuitFileName baseCircuitFileExt], sbc, ...
               comp, str2num(node1), str2num(node2));
        if isempty(r)
            display(err);
            fclose(fiiFID);
            faultyCir = [];
            return
        end
        faultyCir(i) = cellstr(r);
        i = i + 1;
    end
elseif strcmp(fault,'param')
    while remainder
        [parameter, remainder] = strtok(remainder, ' ');
        [deviation, remainder] = strtok(remainder, ' ');
        [r, err] = ...
injFaultParam([baseCircuitFileName baseCircuitFileExt], sbc, ...
               comp, parameter, strtrim(deviation));
        if isempty(r)
            display(err);
            fclose(fiiFID);
            faultyCir = [];
            return
        end
        faultyCir(i) = cellstr(r);
        i = i + 1;
    end
end
end
fclose(fiiFID);
end
```

APÊNDICE E – Função em MATLAB para Injeção de Falhas de Circuito Aberto

```

function [r, err] = injFaultOpen( bcName, sbc, comp, node )
% Creates a new circuit file with the specified fault injected

[~,bcFileName,bcFileExt] = fileparts(bcName);
[bcFID, err] = fopen(bcName, 'r');
if bcFID < 0
    err = [bcName ': ' err];
    r = [];
    return
end

if isempty(sbc)
    faultyCirName = [ bcFileName '_open_' ...
                    comp '_on_' num2str(node)];
else
    faultyCirName = [ bcFileName '_open_' ...
                    sbc '_' comp '_on_' num2str(node)];
end

[ficFID, err] = fopen([faultyCirName bcFileExt], 'w');
line = fgets(bcFID);

if isempty(sbc)
    fprintf(ficFID, [strtok(line, ...
sprintf('\r\n')) ' - Open circuit fault injected at ' comp]);
else
    fprintf(ficFID, [strtok(line, ...
sprintf('\r\n')) ' - Open circuit fault injected at ' sbc '.' comp]);
end

headerMessage = ...
[ sprintf('\n*This is an automatically generated circuit file.\n') ...
sprintf('*The circuit described here contains a 10 MegOhms') ...
sprintf('resistor representing an open circuit fault.\n\n')];
fprintf(ficFID, headerMessage);

if ~isempty(sbc)
    line = fgets(bcFID);
    while ~feof(bcFID) && isempty(strfind(line, ['.subckt ' sbc ' ']))

```

```

        fprintf(ficFID, line);
        line = fgets(bcFID);
    end
    fprintf(ficFID, line);
end

findComp = 0;
findFirstControl = 0;
while ~feof(bcFID)
    line = fgets(bcFID);
    if ~findFirstControl
        [firststr, remainder] = strtok(line, ' ');
        if strcmpi(firststr, '.subckt')
            fprintf(ficFID, line);
            line = fgets(bcFID);
            [firststr, ~] = strtok(line, sprintf(' \r\n'));
            while ~strcmpi(firststr, '.ends')
                fprintf(ficFID, line);
                line = fgets(bcFID);
                [firststr, ~] = strtok(line, sprintf(' \r\n'));
            end
        elseif strcmpi(firststr, comp)
            findComp = 1;
            fprintf(ficFID, firststr);
            for n=1:node
                [nodestr, remainder] = strtok(remainder, ' ');
                if n < node
                    fprintf(ficFID, [' ' nodestr]);
                end
            end
            fprintf(ficFID, [' openfault' remainder]);
        elseif line(1) == '.' && ...
            ~strcmpi(firststr, '.include') && ...
            ~strcmpi(firststr, '.lib') && ...
            findFirstControl == 0
            fprintf(ficFID, ...
                ['\nropen' comp num2str(node) ' openfault ' nodestr ' 10meg\n\n']);
            fprintf(ficFID, line);
            findFirstControl = 1;
        else
            fprintf(ficFID, line);
        end
    else
        fprintf(ficFID, line);
    end
end
end

```

```
fprintf(ficFID, ['\n*Created at ' datestr(now) '.\n']);
fclose(bcFID);
fclose(ficFID);

if findComp == 0
    err = [bcName ': Component ' comp ' or subcircuit ' sbc ' not found.'];
    r = [];
    return
end

r = faultyCirName;
end
```

APÊNDICE F – Função em MATLAB para Injeção de Falhas de Curto Circuito

```

function [r, err] = injFaultShort( bcName, sbc, comp, node1, node2 )
% Write a new circuit file with the specified fault injected

[~,bcFileName,bcFileExt] = fileparts(bcName);
[bcFID, err] = fopen(bcName, 'r');
if bcFID < 0
    err = [bcName ': ' err];
    r = [];
    return
end

if isempty(sbc)
    faultyCirName = [ bcFileName '_short_' comp ...
        '_between_' num2str(node1) '_and_' num2str(node2)];
else
    faultyCirName = [ bcFileName '_short_' sbc '_' comp ...
        '_between_' num2str(node1) '_and_' num2str(node2)];
end

[ficFID, err] = fopen([faultyCirName bcFileExt], 'w');
line = fgets(bcFID);

if isempty(sbc)
    fprintf(ficFID, [strtok(line, ...
sprintf('\r\n')) ' - Short circuit fault injected at ' comp]);
else
    fprintf(ficFID, [strtok(line, ...
sprintf('\r\n')) ' - Short circuit fault injected at ' sbc '.' comp]);
end

headerMessage = ...
[ sprintf('\n*This is an automatically generated circuit file.\n') ...
sprintf('*The circuit described here contains a 10 Ohms') ...
sprintf('resistor representing an short circuit fault.\n\n')];
fprintf(ficFID, headerMessage);

if ~isempty(sbc)
    line = fgets(bcFID);
    while ~feof(bcFID) && isempty(strfind(line, ['.subckt ' sbc ' ']))

```

```
fprintf(ficFID, line);
line = fgets(bcFID);
end
fprintf(ficFID, line);
end

findComp = 0;
findFirstControl = 0;
while ~feof(bcFID)
    line = fgets(bcFID);
    if ~findFirstControl
        [firststr, remainder] = strtok(line, ' ');
        if strcmpi(firststr, '.subckt')
            fprintf(ficFID, line);
            line = fgets(bcFID);
            [firststr, ~] = strtok(line, sprintf(' \r\n'));
            while ~strcmpi(firststr, '.ends')
                fprintf(ficFID, line);
                line = fgets(bcFID);
                [firststr, ~] = strtok(line, sprintf(' \r\n'));
            end
        elseif strcmpi(firststr, comp)
            findComp = 1;
            for n=1:node1
                [node1str, remainder] = strtok(remainder, ' ');
            end
            [~, remainder] = strtok(line, ' ');
            for n=1:node2
                [node2str, remainder] = strtok(remainder, ' ');
            end
            fprintf(ficFID, line);
            elseif line(1) == '.' && ...
                ~strcmpi(firststr, '.include') && ...
                ~strcmpi(firststr, '.lib') && ...
                findFirstControl == 0
            fprintf(ficFID, [sprintf('\n') 'rshort' comp num2str(node1) ...
                num2str(node2) ' ' node1str ' ' node2str ' 10' sprintf('\n\n')]);
            fprintf(ficFID, line);
            findFirstControl = 1;
        else
            fprintf(ficFID, line);
        end
    else
        fprintf(ficFID, line);
    end
end
end
```

```
fprintf(ficFID, ['\n*Created at ' datestr(now) '\n']);
fclose(bcFID);
fclose(ficFID);

if findComp == 0
    err = [bcName ': Component ' comp ' or subcircuit ' sbc ' not found.'];
    r = [];
    return
end

r = faultyCirName;
end
```

APÊNDICE G – Função em MATLAB para Injeção de Falhas Paramétricas

```

function [r1, err] = injFaultParam( bcName, sbc, comp, param, dev)
% Write a new circuit file with the specified fault injected

[~, bcFileName, bcFileExt] = fileparts(bcName);
[bcFID, err] = fopen(bcName, 'r');
if bcFID < 0
    err = [bcName ': ' err];
    r1 = [];
    return
end

if isempty(sbc)
    faultyCirNameep = [bcFileName '_deviation_' comp '_in_' ...
        param '_by_' dev];
else
    faultyCirNameep = [bcFileName '_deviation_' sbc '_' comp '_in_' ...
        param '_by_' dev];
end

[ficpFID, ~] = fopen([faultyCirNameep bcFileExt], 'w');
line = fgets(bcFID);

if isempty(sbc)
    fprintf(ficpFID, [strtok(line, ...
sprintf('\r\n')) ' - Parameter fault injected at ' comp]);
else
    fprintf(ficpFID, [strtok(line, ...
sprintf('\r\n')) ' - Parameter fault injected at ' sbc '.' comp]);
end

headerMessagep = ...
[ sprintf('\n*This is an automatically generated circuit file.\n') ...
sprintf('*The circuit described here has the value of %s', param)];
fprintf(ficpFID, headerMessagep);
fprintf(ficpFID, ' of %s deviated by %s.\n\n', comp, dev);

if ~isempty(sbc)
    line = fgets(bcFID);
    while ~feof(bcFID) && isempty(strfind(line, ['.subckt ' sbc ' ']))

```

```

        fprintf(ficpFID, line);
        line = fgets(bcFID);
    end
    fprintf(ficpFID, line);
end

inSbc = 0;
findComp = 0;
findFirstControl = 0;
while ~feof(bcFID)
    line = fgets(bcFID);
    if ~findFirstControl
        [firststr, ~] = strtok(line, ' ');
        if strcmpi(firststr, '.subckt')
            fprintf(ficpFID, line);
            line = fgets(bcFID);
            [firststr, ~] = strtok(line, sprintf(' \r\n'));
            while ~strcmpi(firststr, '.ends')
                fprintf(ficpFID, line);
                line = fgets(bcFID);
                [firststr, ~] = strtok(line, sprintf(' \r\n'));
            end
        elseif strcmpi(firststr, comp)
            findComp = 1;
            i = strfind(line, [param '=']) + 1 + length(param);
            if isempty(i)
                fprintf(ficpFID, line);
                line = fgets(bcFID);
                while line(1) == '+' && isempty(i)
                    i = strfind(line, [param '=']) + 1 + length(param);
                    if isempty(i)
                        fprintf(ficpFID, line);
                        line = fgets(bcFID);
                    end
                end
            end
        end
        if isempty(i)
            fclose(bcFID);
            fclose(ficpFID);
            err = [bcName ': Parameter ' param ' not found.'];
            r1 = [];
            r2 = [];
            return
        end
        [valueStr, remainder] = ...
            strtok(line(i:end), sprintf(' \r\n'));
        value = paramStr2Num(valueStr);
    end
end

```

```
    if dev(end) == '%'
        fprintf( ficpFID, [line(1:i-1) ...
            num2str( (1+str2num(dev(1:end-1))/100) * value ) remainder] );
    else
        fprintf( ficpFID, [line(1:i-1) ...
            num2str(value + str2num(dev)) remainder] );
    end
elseif line(1) == '.' && ...
    ~strcmpi(firststr, '.include') && ...
    ~strcmpi(firststr, '.lib') && ...
    findFirstControl == 0
    fprintf(ficpFID, line);
    findFirstControl = 1;
else
    fprintf(ficpFID, line);
end
else
    fprintf(ficpFID, line);
end
end

fprintf(ficpFID, ['\n*Created at ' datestr(now) '\n']);
fclose(bcFID);
fclose(ficpFID);

if findComp == 0
    err = [bcName ': Component ' comp ' or subcircuit ' sbc ' not found.'];
    r1 = [];
    return
end

r1 = faultyCirNamep;

end
```

APÊNDICE H – Função em MATLAB para Conversão de Fatores de Multiplicação

```

function [ paramn ] = paramStr2Num( param )
% paramStr2Num Converts a string containing the scale
% factor (SPICE like) to numeric
param = strtrim(param);
switch param(length(param))
    case {'f', 'F'}
        paramn = str2num(param(1:length(param)-1))*1e-15;
    case {'p', 'P'}
        paramn = str2num(param(1:length(param)-1))*1e-12;
    case {'n', 'N'}
        paramn = str2num(param(1:length(param)-1))*1e-9;
    case {'u', 'U'}
        paramn = str2num(param(1:length(param)-1))*1e-6;
    case {'m', 'm'}
        paramn = str2num(param(1:length(param)-1))*1e-3;
    case {'k', 'K'}
        paramn = str2num(param(1:length(param)-1))*1e3;
    case {'g', 'G'}
        if param(length(param)-1) == 'e' || param(length(param)-1) == 'E'
            paramn = str2num(param(1:length(param)-3))*1e6;
        else
            paramn = str2num(param(1:length(param)-1))*1e9;
        end
    case {'t', 'T'}
        paramn = str2num(param(1:length(param)-1))*1e12;
    otherwise
        paramn = str2num(param);
end

```

APÊNDICE I – Função em MATLAB para Execução e Verificação das Simulações

```

function [] = ...
runSim( hspice, bsFileFullName, auxDir, cornerCir, faultyCir, delay, type )
%RUNSIM Calls HSPICE for each netlist. Checks if AC or TRAN results were
% created.

[~, baseCircuitFileName, baseCircuitFileExt] = ...
fileparts (bsFileFullName);

switch type
    case 'ac'
        ext = '.ac0';
    case 'tran'
        ext = '.tr0';
    otherwise
        error('Unknown analysis type %s.', type)
end

dos([hspice ' ' baseCircuitFileName baseCircuitFileExt ' -o']);
for i=1:length(cornerCir)
    dos([hspice ' ' char(cornerCir(i)) baseCircuitFileExt ' -o']);
    pause(delay);
end

for i=1:length(faultyCir)
    dos([hspice ' ' char(faultyCir(i)) baseCircuitFileExt ' -o']);
    pause(delay);
end

% Verify the completeness of the simulation. Might need user's action!!!
[~, fileStr] = dos('dir /B');
simulationOKflag = 1;
if isempty(strfind(fileStr, [baseCircuitFileName ext]))
    simulationOKflag = 0;
end
if simulationOKflag
for i=1:length(cornerCir)
    if isempty(strfind(fileStr, [char(cornerCir(i)) ext]))
        simulationOKflag = 0;
        break;
    end
end
end

```

```
        end
    end
end
end
if simulationOKflag
for i=1:length(faultyCir)
    if isempty(strfind(fileStr, [char(faultyCir(i)) ext]))
        simulationOKflag = 0;
        break;
    end
end
end

while ~simulationOKflag
errorMsg = ['We might have identified errors in HSpice calls. For we ' ...
           'to try to correct the error(s), it is needed that the ' ...
           'user closes all Windows error pop-ups. ' sprintf('\n\t') ...
           'HAVE YOU ALREADY CLOSED ALL WINDOWS ERROR POP-UPS?'];
while ~strcmpi(questdlg(errorMsg), 'Yes')
end

[~, fileStr] = dos('dir /B');
simulationOKflag = 1;
if isempty(strfind(fileStr, [baseCircuitFileName ext]))
    dos([hspice ' ' baseCircuitFileName baseCircuitFileExt ' -o']);
    simulationOKflag = 0;
end
for i=1:length(cornerCir)
    if isempty(strfind(fileStr, [char(cornerCir(i)) ext]))
        dos([hspice ' ' char(cornerCir(i)) baseCircuitFileExt ' -o']);
        pause(delay);
        simulationOKflag = 0;
    end
end
for i=1:length(faultyCir)
    if isempty(strfind(fileStr, [char(faultyCir(i)) ext]))
        dos([hspice ' ' char(faultyCir(i)) baseCircuitFileExt ' -o']);
        pause(delay);
        simulationOKflag = 0;
    end
end

pause(3*delay+30);
dos('del /Q *.ic0');
dos('del /Q *.pa0');
dos('del /Q *.st0');
dos(['mkdir ' auxDir]);
```

```
pause(delay);  
dos(['move /-Y ' baseCircuitFileName '_*.* ' auxDir]);  
pause(delay);
```

```
end
```

APÊNDICE J – Função em MATLAB para Criação do Dicionário de Falhas DC

```

function [vNomAndCorner, vMax, vMin, vFault, faultMatrix] = ...
    analyseDCResults(bsFileFullName, cornerCir, faultyCir, auxFolder, node)
%Analyse the Operation Point results given the corner and faulty voltages
%of every node of the circuit.

[~, baseCircuitFileName, ~] = ...
fileparts (bsFileFullName);

fullTRAN = loadsig([baseCircuitFileName '.tr0']);

vNomAndCorner = zeros (length (node), length (cornerCir)+1);

t = evalsig (fullTRAN, 'TIME');

for nodeID=1:length (node)
    aux = evalsig (fullTRAN, char (node (nodeID)));
    vNomAndCorner (nodeID,1) = aux (end);
end

for cornerID = 1:length (cornerCir)
    fullTRAN = loadsig ([auxFolder '/' char (cornerCir (cornerID)) '.tr0']);
    for nodeID=1:length (node)
        aux = evalsig (fullTRAN, char (node (nodeID)));
        vNomAndCorner (nodeID, cornerID+1) = aux (end);
    end
end

vMax = (1 + 0.35)*vNomAndCorner (:,1);%max (vNomAndCorner, [],2);
vMin = (1 - 0.35)*vNomAndCorner (:,1);%min (vNomAndCorner, [],2);

vFault = zeros (length (node), length (faultyCir));
for faultID = 1:length (faultyCir)
    fullTRAN = loadsig ([auxFolder '/' char (faultyCir (faultID)) '.tr0']);
    for nodeID=1:length (node)
        aux = evalsig (fullTRAN, char (node (nodeID)));
        vFault (nodeID, faultID) = aux (end);
    end
    disp (['Reading files: ' num2str (faultID/length (faultyCir)*100) ...
        '% completed.']);
end

```

```
faultMatrix = int8(zeros(length(node),length(faultyCir)));
for faultID = 1:length(faultyCir)
    for nodeID=1:length(node)
        if vMax(nodeID) < vMin(nodeID)
            aux = vMax(nodeID);
            vMax(nodeID) = vMin(nodeID);
            vMin(nodeID) = aux;
        end
        if vFault(nodeID, faultID) > vMax(nodeID) ...
            || vFault(nodeID, faultID) < vMin(nodeID)
            faultMatrix(nodeID, faultID) = 1;
        end
    end
end
end
```

APÊNDICE K – Função em MATLAB para Criação do Dicionário de Falhas AC

```

function [f, vNomAndCorner, vMax, vMin, vFault, faultMatrix] = ...
    analyseACResults(bsFileFullName, cornerCir, faultyCir, auxFolder, node)
%Analyse the AC Analysis results given the corner and faulty voltages
%of every node of the circuit.

[~, baseCircuitFileName, ~] = ...
fileparts (bsFileFullName);

fullFR = loadsig([baseCircuitFileName '.ac0']);

vNomAndCorner = ...
complex( zeros (length (fullFR (1).data), length (node), length (cornerCir)+1) );

f = evalsig (fullFR, 'HERTZ');

for nodeID=1:length (node)
    vNomAndCorner (:, nodeID, 1) = abs (evalsig (fullFR, char (node (nodeID))));
end

for cornerID = 1:length (cornerCir)
    fullFR = loadsig ([auxFolder '/' char (cornerCir (cornerID)) '.ac0']);
    for nodeID=1:length (node)
        vNomAndCorner (:, nodeID, cornerID+1) = ...
            abs (evalsig (fullFR, char (node (nodeID))));
    end
end

vMax = max (vNomAndCorner, [], 3);
vMin = min (vNomAndCorner, [], 3);

vFault = ...
complex( zeros (length (fullFR (1).data), length (node), length (faultyCir)) );
for faultID = 1:length (faultyCir)
    disp (['Reading files: ' num2str (faultID/length (faultyCir)*100) ...
        '% completed.']);
    fullFR = loadsig ([auxFolder '/' char (faultyCir (faultID)) '.ac0']);
    for nodeID=1:length (node)
        vFault (:, nodeID, faultID) = ...
            abs (evalsig (fullFR, char (node (nodeID))));
    end
end

```

```
end

faultMatrix = int8(zeros(length(f), length(node), length(faultyCir)));
for faultID = 1:length(faultyCir)
    for nodeID=1:length(node)
        for fidx=1:length(f)
            if vFault(fidx,nodeID,faultID) > vMax(fidx,nodeID) ...
                || vFault(fidx,nodeID,faultID) < vMin(fidx,nodeID)
                faultMatrix(fidx,nodeID,faultID) = 1;
            end
        end
    end
end
end
end
```

APÊNDICE L – Função em MATLAB para Escolha Otimizada de Conjunto de Testes

```

function [ maxFCConfig ] = ...
    findMaxFCtestConfig( faultMatrix, f, fInterval, tNodeID )
% Returns the vector with the pairs of freq. and node IDs which gives the
% maximum fault coverage possible.

observability = 0;
numNoObsFaults = 0;
fMinID = find(f>fInterval(1),1);
fMaxID = find(f>fInterval(2),1)-1;
for faultID=1:length(faultMatrix(1,1,:))
    numConfig = sum(sum(faultMatrix(fMinID:fMaxID,tNodeID,faultID)));
    if numConfig == observability
        numNoObsFaults = numNoObsFaults + 1;
    end
end
observability = observability + 1;
display(['Maximum FC is ' ...
    num2str((faultID-numNoObsFaults)/faultID*100) '%.']);

configFCvec = int8(zeros(length(faultMatrix(1,1,:)),1));
maxFCConfig = [];
i=1;
while sum(configFCvec) < (length(faultMatrix(1,1,:)) - numNoObsFaults)
    for faultID=1:length(faultMatrix(1,1,:))
        numConfig = sum(sum(faultMatrix(fMinID:fMaxID,tNodeID,faultID)));
        if numConfig == observability
            if ~configFCvec(faultID)
                [ bestConfig, configFCvec ] = ...
findBestConfig( faultMatrix, faultID, configFCvec, f, fInterval, tNodeID );
                maxFCConfig(i,:) = bestConfig;
                i = i + 1;
            end
        end
    end
    observability = observability + 1;
end
end
end

```

APÊNDICE M – Função em MATLAB para Escolha Otimizada de Configuração de Teste

```

function [ bestConfig, configFCvecOUT ] = ...
findBestConfig( faultMatrix, faultID, configFCvecIN, f, fInterval, tNodeID)
% Returns the pair of freq. and node IDs which refers to the test freq. and
% node with the maximum number of new faults covered, including the once
% passed as argument.

bestConfigFC = 0;
for freqID=length(faultMatrix(:,1,1))-1:1
    if f(freqID) >= fInterval(1) && f(freqID) <= fInterval(2)
        for i2 = tNodeID
            if faultMatrix(freqID,i2,faultID)
                fcVec = bitor(configFCvecIN, squeeze(faultMatrix(freqID,i2,:)));
                if bestConfigFC < sum(fcVec)
                    bestConfigFC = sum(fcVec);
                    configFCvecOUT = fcVec;
                    bestConfig = [freqID i2];
                end
            end
        end
    end
end
end
end
end
end

```

APÊNDICE N – Descrição em SPICE do OPAMP PMOS-DA

PMOS 500nm

```
.lib 'ami_06.lib' MOSFET

m1 4 in- 3 vdd pmos l=1u w=4u
m2 5 in+ 3 vdd pmos l=1u w=4u
m3 4 4 vss vss nmos l=1u w=1u
m4 5 4 vss vss nmos l=1u w=1u
m5 3 1 vdd vdd pmos l=1u w=2.5u
m6 out 5 vss vss nmos l=1u w=24u
m7 out 1 vdd vdd pmos l=1u w=30u
m8 1 1 vdd vdd pmos l=1u w=2.5u
c1 5 out 4.4pf
rref 1 vss 335k
c11 out 0 20p

vin+ in+ 0 0
vin- in- 0 0
v4 vdd 0 2.5
v5 vss 0 -2.5

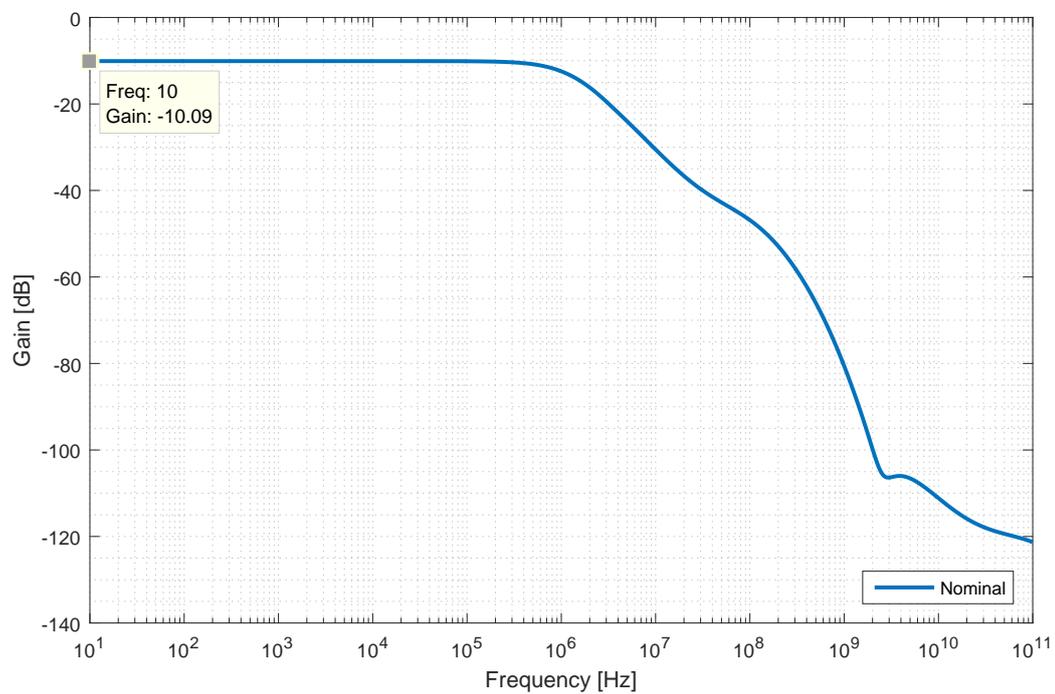
.OPTIONS INGOLD=2 ABSTOL=1E-12
.tran 0.005u 30u start = 25u
.option post
.print tran I(m5) v(5) v(in-) v(out)
.end
```

APÊNDICE O – Exemplo de arquivo de Instruções para Injeção de Falhas (FII)

```
m1 open 1 2 3
m2 open 1 2 3
m3 open 1 2 3
m4 open 1 2 3
m5 open 1 2 3
m6 open 1 2 3
m7 open 1 2 3
m8 open 1 2 3
m1 short 1 2 1 3 2 3
m2 short 1 2 1 3 2 3
m3 short 1 2 1 3 2 3
m4 short 1 2 1 3 2 3
m5 short 1 2 1 3 2 3
m6 short 1 2 1 3 2 3
m7 short 1 2 1 3 2 3
m8 short 1 2 1 3 2 3
```

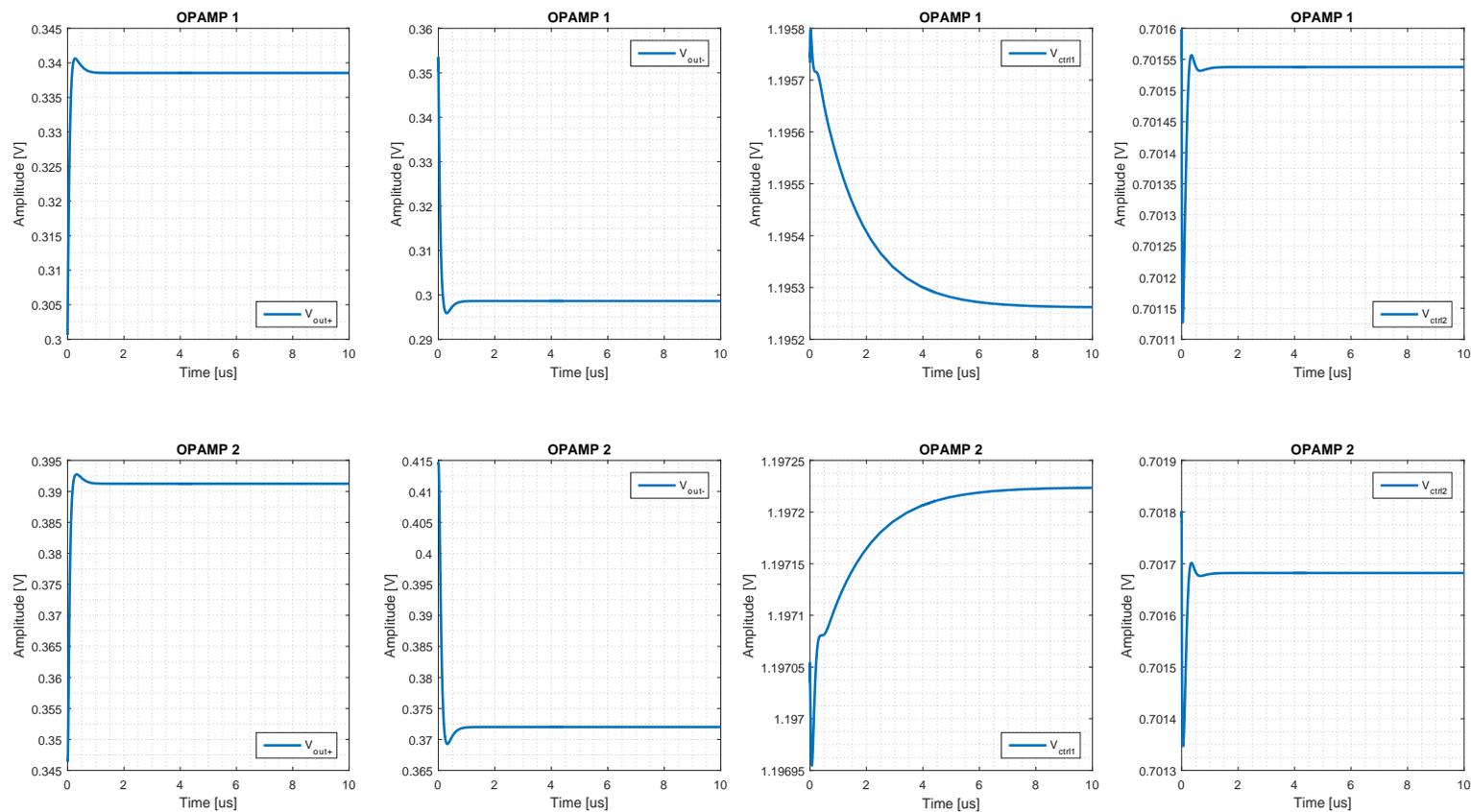
APÊNDICE P – Gráficos da Resposta em Frequência e Resposta ao Degrau do SDM

Figura 25 – Resposta em frequência da tensão de saída do filtro em malha fechada do SDM projetado em tecnologia IBM 0,13 μm .



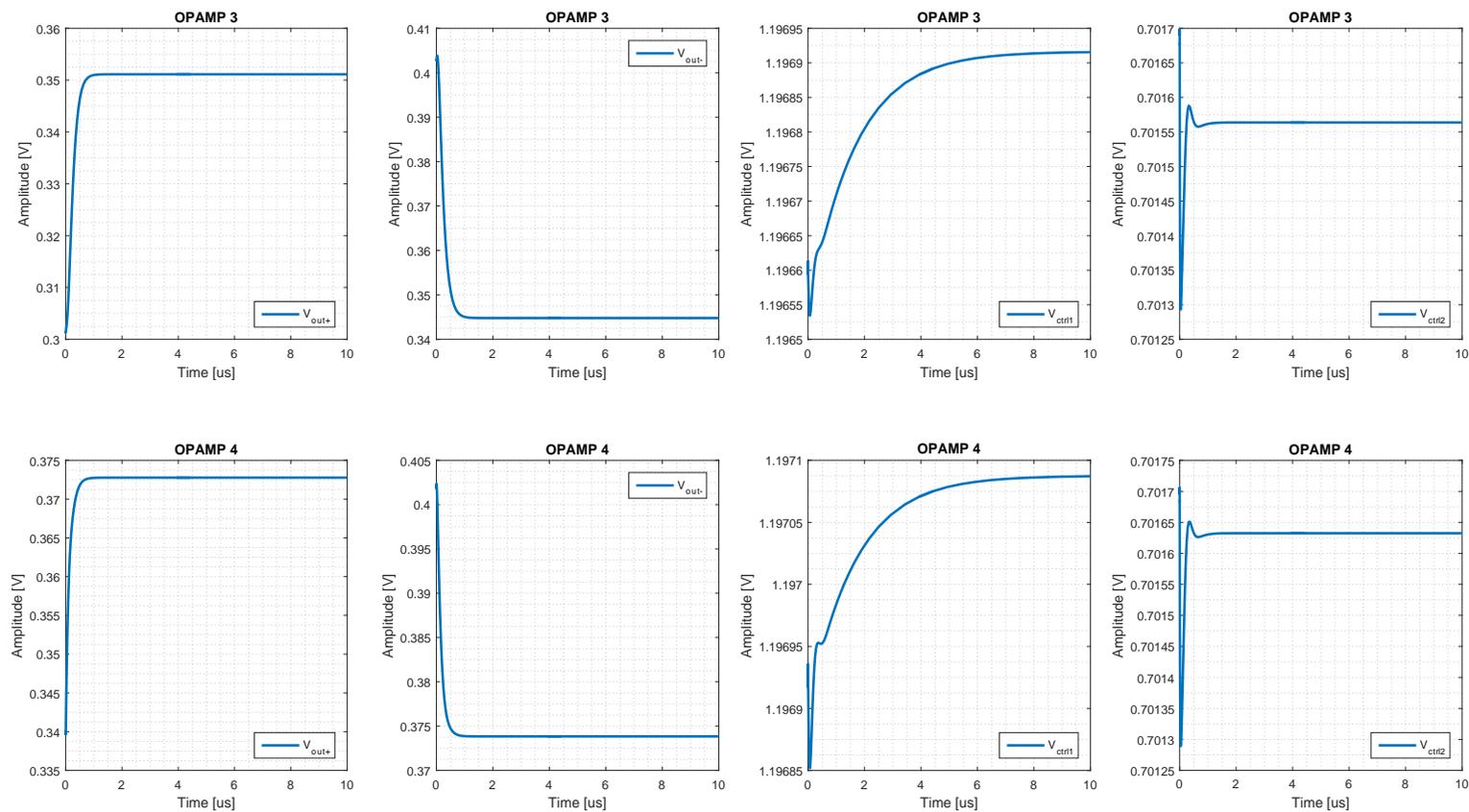
Fonte: elaborado pelo autor.

Figura 26 – Resposta ao degrau dos nós de interesse dos FDAs 1 e 2 do filtro em malha fechada do SDM projetado em tecnologia IBM $0,13\ \mu\text{m}$.



Fonte: elaborado pelo autor.

Figura 27 – Resposta ao degrau dos nós de interesse dos FDAs 3 e 4 do filtro em malha fechada do SDM projetado em tecnologia IBM $0,13\ \mu\text{m}$.



Fonte: elaborado pelo autor.