

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

CENTRO DE TECNOLOGIA



PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

### Adauto Luis Tadeo Bernardes da Fonseca

# Metodologia de Verificação Funcional para Circuitos Analógicos

Orientador: Prof. Dr. Fernando Rangel de Sousa

Natal, RN, Agosto de 2009

### Adauto Luis Tadeo Bernardes da Fonseca

# Metodologia de Verificação Funcional para Circuitos Analógicos

**Dissertação de Mestrado** apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da UFRN (área de concentração: Engenharia de Computação) como parte dos requisitos para obtenção do título de Mestre em Ciências.

Orientador: Prof. Dr. Fernando Rangel de Sousa

Natal, RN, Agosto de 2009

# Metodologia de Verificação Funcional para Circuitos Analógicos

Adauto Luis Tadeo Bernardes da Fonseca

Dissertação de Mestrado aprovada em 04 de setembro de 2009 pela banca examinadora composta pelos seguintes membros:

Prof. Dr. Fernando Rangel de Sousa (orientador) ..... DEE/UFRN

Prof. Dr. Elmar Uwe Kurt Melcher ..... DEE/UFCG

Prof. Dr. Vincent Patrick Marie Bourguet ...... DEE/UFCG

Prof. Dr. José Alberto Nicolau de Oliveira ..... DEE/UFRN

## Resumo

O presente trabalho tem como objetivo desenvolver uma ferramenta de verificação para circuitos analógicos. O principal objetivo desta é aumentar a automação dos processos de verificação. Além disso, proporcionar a construção de um ambiente de verificação capaz de gerar relatórios ao longo deste processo. Esta metodologia é baseada na técnica do Modelo de Ouro, no entanto, ela também propõe uma segunda técnica para verificar o modelo de referência, para se obter resultados mais confiáveis. A metodologia foi utilizada, como estudo de caso, na verificação de um amplificador operacional.

**Palavras-Chaves:** Verificação. Circuitos Analógicos. Técnicas de Verificação. Ambiente de Verificação. VHDL-AMS

## **Abstract**

This work proposes a new methodology to verify those analog circuits, providing an automated tools to help the verifiers to have a more truthful result. This work presents the development of new methodology for analog circuits verification. The main goal is to provide a more automated verification process to certify analog circuits functional behavior. The proposed methodology is based on the golden model technique. A verification environment based on this methodology was built and results of a study case based on the validation of an operational amplifier design are offered as a confirmation of its effectiveness. The results had shown that the verification process was more truthful because of the automation provided by the tool developed.

**Keywords:** Verification. Analog Circuits. Integrated Circuits. Verification Techniques. Verification Environment. VHDL-AMS

# Sumário

| Li | Lista de Ilustrações               |          |  |    |  |  |
|----|------------------------------------|----------|--|----|--|--|
| Li | ista de                            | e Quadr  | os   | XV |  |  |
| Li | ista de abreviaturas e siglas xvii |          |  |    |  |  |
| Li | ista de                            | e símbol | os   | 1  |  |  |
| 1  | Intr                               | odução   |  | 2  |  |  |
| 2  | Proc                               | cesso de | Verificação  | 8  |  |  |
|    | 2.1                                | Verific  | ação   | 8  |  |  |
|    |                                    | 2.1.1    | Verificação Funcional                              | 9  |  |  |
|    | 2.2                                | Proces   | so de Verificação                                  | 11 |  |  |
|    |                                    | 2.2.1    | Elaboração do Plano de Verificação                 | 14 |  |  |
|    |                                    |          | 2.2.1.1 Geração de Estímulos                       | 15 |  |  |
|    |                                    | 2.2.2    | Implementação dos Bancos de Teste Auto-Avaliadores | 16 |  |  |
|    |                                    | 2.2.3    | Execução dos Testes de Regressão                   | 16 |  |  |
|    |                                    | 2.2.4    | Construção do Ambiente de Verificação              | 17 |  |  |
|    | 2.3                                | Escolh   | a de uma Técnicas de Verificação                   | 18 |  |  |
|    |                                    | 2.3.1    | Técnicas baseadas em Simulações                    | 19 |  |  |
|    |                                    |          | 2.3.1.1 Técnica do Modelo de Ouro                  | 20 |  |  |
|    |                                    |          | A linguagem VHDL-AMS                               | 22 |  |  |
|    |                                    |          | 2.3.1.2 Verificação a Nível de Transação           | 25 |  |  |

|   |      |          | 2.3.1.3 Técnica do Testes Randômicos                          | 25 |
|---|------|----------|---|----|
|   |      | 2.3.2    | Técnicas não baseadas em Simulações                           | 26 |
|   |      | 2.3.3    | Técnicas Mistas   | 27 |
|   |      |          | 2.3.3.1 Técnica de Verificação em Tempo de Execução           | 28 |
|   | 2.4  | Conclu   | usão  | 29 |
| 3 | Met  | odologi  | a de Verificação de Circuitos Analógicos                      | 30 |
|   | 3.1  | Detalh   | namento da Metodologia Proposta                               | 30 |
|   | 3.2  | Constr   | rução do Ambiente de Verificação do Modelo de Referência      | 31 |
|   |      | 3.2.1    | Modelo de Referência  | 32 |
|   |      | 3.2.2    | A Implementação da técnica do Monitoramento por Autômatos     | 33 |
|   |      |          | 3.2.2.1 Componente Estímulos                                  | 34 |
|   |      |          | 3.2.2.2 Componente Monitor                                    | 35 |
|   |      | 3.2.3    | Definição das Propriedades Verificadas                        | 35 |
|   |      | 3.2.4    | Descrição do Relatório da Execução do Ambiente de Verificação | 36 |
|   | 3.3  | Impler   | mentação do Ambiente de Verificação do Circuito               | 37 |
|   | 3.4  | Conclu   | usão  | 40 |
| 4 | Estu | ıdo de ( | Caso A - Validação de um Amplificador Operacional             | 41 |
|   | 4.1  | Plano    | de Verificação  | 41 |
|   |      | 4.1.1    | Introdução  | 41 |
|   |      | 4.1.2    | Aspectos Tecnológicos   | 42 |
|   |      |          | 4.1.2.1 Objetivos   | 42 |
|   |      |          | 4.1.2.2 Especificação   | 42 |
|   |      |          | 4.1.2.3 Propriedades  | 43 |
|   |      |          | 4.1.2.4 Técnicas  | 43 |
|   |      |          | 4.1.2.5 Ferramentas   | 43 |

|   |  |   | 4.1.2.6 A   | mbiente   | 44   |
|---|--|---|---|---|--|
|   |  | 4.1.3   | Aspectos Cr   | onológicos  | 44   |
|   |  | 4.1.4   | Conclusão .   |   | 45   |
|   | 4.2  | Projeto   |   |   | 46   |
|   | 4.3  | Process   | o de Verifica   | ção   | 49   |
|   |  | 4.3.1   | Modelo de I   | Referência  | 49   |
|   |  |   | 4.3.1.1 A   | mplificador Operacional   | 49   |
|   |  |   | 4.3.1.2 V   | erificação das propriedades no Modelo Comportamental  | 52   |
|   |  |   | 4.3.1.3 R   | esultados   | 57   |
|   |  | 4.3.2   | Certificação  | de um Amplificador Operacional pela Técnica do Modelo   |  |
|   |  |   | de Ouro   |   | 62   |
|   |  |   | 4.3.2.1 In  | nplementação dos Bancos de Teste  | 62   |
|   |  |   | 4.3.2.2 D   | etalhamento dos Resultados da execução dos Bancos de Teste  | 65   |
|   |  |   |   |   |  |
|   | 4.4  | Conclu  | são   |   | 69   |
| 5 | 4.4<br>Estu                                    | Conclu<br>do de C   | são<br>aso B - Valio  | lação da implementação em C do modelo ACM   | 69<br>70   |
| 5 | 4.4<br>Estu<br>5.1                             | Conclu<br>do de C<br>Plano d  | são<br>aso B - Valio<br>e Verificação   | lação da implementação em C do modelo ACM   | 69<br><b>70</b><br>70  |
| 5 | 4.4<br>Estu<br>5.1                             | Conclu<br>do de C<br>Plano c<br>5.1.1   | são<br>aso B - Valio<br>e Verificação<br>Introdução   | lação da implementação em C do modelo ACM   | 69<br><b>70</b><br>70<br>70  |
| 5 | 4.4<br>Estu<br>5.1                             | Conclu<br>do de C<br>Plano d<br>5.1.1<br>5.1.2  | são<br>aso B - Valio<br>e Verificação<br>Introdução<br>Aspectos Te  | dação da implementação em C do modelo ACM   | <ul> <li>69</li> <li>70</li> <li>70</li> <li>71</li> </ul>   |
| 5 | 4.4<br>Estu<br>5.1                             | Conclu<br>do de C<br>Plano d<br>5.1.1<br>5.1.2<br>5.1.3                                     | são<br><b>aso B - Valio</b><br>e Verificação<br>Introdução<br>Aspectos Te<br>Objetivos .  | lação da implementação em C do modelo ACM         ) | <ul> <li>69</li> <li>70</li> <li>70</li> <li>71</li> <li>71</li> </ul>   |
| 5 | <ul><li>4.4</li><li>Estu</li><li>5.1</li></ul> | Conclu<br>do de C<br>Plano d<br>5.1.1<br>5.1.2<br>5.1.3<br>5.1.4                            | são<br>aso <b>B - Valio</b><br>e Verificação<br>Introdução<br>Aspectos Te<br>Objetivos .<br>Técnicas .  | lação da implementação em C do modelo ACM         )         )         cenológicos   | <ul> <li>69</li> <li>70</li> <li>70</li> <li>71</li> <li>71</li> <li>72</li> </ul>   |
| 5 | 4.4<br>Estu<br>5.1                             | Conclu<br>do de C<br>Plano c<br>5.1.1<br>5.1.2<br>5.1.3<br>5.1.4<br>5.1.5                   | são<br>aso B - Valio<br>e Verificação<br>Introdução<br>Aspectos Te<br>Objetivos .<br>Técnicas .<br>Propriedade  | lação da implementação em C do modelo ACM         o | <ul> <li>69</li> <li>70</li> <li>70</li> <li>71</li> <li>71</li> <li>72</li> <li>72</li> </ul>   |
| 5 | 4.4<br>Estu<br>5.1                             | Conclu<br>do de C<br>Plano d<br>5.1.1<br>5.1.2<br>5.1.3<br>5.1.4<br>5.1.5                   | são<br>aso B - Valio<br>e Verificação<br>Introdução<br>Aspectos Te<br>Objetivos .<br>Técnicas .<br>Propriedade<br>5.1.5.1 A   | Jação da implementação em C do modelo ACM         0 | <ul> <li>69</li> <li>70</li> <li>70</li> <li>71</li> <li>71</li> <li>72</li> <li>72</li> <li>73</li> </ul>                                     |
| 5 | 4.4<br>Estu<br>5.1                             | Conclu<br>do de C<br>Plano C<br>5.1.1<br>5.1.2<br>5.1.3<br>5.1.4<br>5.1.5                   | são<br>aso B - Valio<br>e Verificação<br>Introdução<br>Aspectos Te<br>Objetivos .<br>Técnicas .<br>Propriedade<br>5.1.5.1 A<br>5.1.5.2 Fe                               | Hação da implementação em C do modelo ACM         0 | <ul> <li>69</li> <li>70</li> <li>70</li> <li>71</li> <li>71</li> <li>72</li> <li>72</li> <li>73</li> <li>74</li> </ul>                         |
| 5 | 4.4<br>Estu<br>5.1                             | Conclu<br>do de C<br>Plano d<br>5.1.1<br>5.1.2<br>5.1.3<br>5.1.4<br>5.1.5<br>5.1.6          | são<br>aso B - Valio<br>e Verificação<br>Introdução<br>Aspectos Te<br>Objetivos .<br>Técnicas .<br>Propriedade<br>5.1.5.1 A<br>5.1.5.2 Fo<br>Aspectos Cr                | Hação da implementação em C do modelo ACM         0 | <ul> <li>69</li> <li>70</li> <li>70</li> <li>71</li> <li>71</li> <li>72</li> <li>72</li> <li>73</li> <li>74</li> <li>74</li> </ul>             |
| 5 | 4.4<br>Estu<br>5.1                             | Conclu<br>do de C<br>Plano d<br>5.1.1<br>5.1.2<br>5.1.3<br>5.1.4<br>5.1.5<br>5.1.6<br>5.1.7 | são<br>aso B - Valio<br>e Verificação<br>Introdução<br>Aspectos Te<br>Objetivos .<br>Técnicas .<br>Propriedade<br>5.1.5.1 A<br>5.1.5.2 Fo<br>Aspectos Cr<br>Conclusão . | dação da implementação em C do modelo ACM         0 | <ul> <li>69</li> <li>70</li> <li>70</li> <li>71</li> <li>71</li> <li>72</li> <li>72</li> <li>73</li> <li>74</li> <li>74</li> <li>75</li> </ul> |

|     | 5.2.1      | Impleme     | entação em Modelos Computacionais Simuláveis             | 79 |
|-----|------------|-------------|--|----|
| 5.3 | Proces     | sso de Veri | ficação  | 81 |
|     | 5.3.1      | Modelo      | de Ouro  | 81 |
|     | 5.3.2      | Verificaç   | ão das Propriedades                                      | 82 |
|     | 5.3.3      | Certifica   | ção pela Técnica por Modelo de Ouro                      | 86 |
|     |            | 5.3.3.1     | Implementação dos Bancos de Teste                        | 86 |
|     |            | 5.3.3.2     | Detalhamento dos Resultados obtidos na Execução dos Ban- |    |
|     |            |             | cos de Teste   | 88 |
| 5.4 | Conclu     | usão        |  | 93 |
| Con | Conclusões |             |  |    |

#### Referências

# Lista de Ilustrações

| GRÁFICO  | D 1. Representação em pizza do número de circuitos integrados projetados, des-<br>tacando a quantidade destes circuitos não apresentaram falhas, as que apresenta-<br>ram e, em separado, as que não apresentariam se estivessem sido submetidos a<br>uma metodologia de verificação | 4  |
|----------|--|----|
| FIGURA   | 1. Uma macro-representação dos fluxos de trabalhos existentes em uma casa de projetos de semicondutores. Estão apresentadas as fases para construir o circuito, projeto e fabricação, e as de sentido inverso, verificação e testes, para certificar o seu funcionamento.            | 8  |
| FIGURA 2 | 2. Representação dos universos das possibilidades de intenção do projetista, espe-<br>cificação do projeto e a sua implementação. O principal objetivo da verificação<br>funcional é garantir a conciliação destes três universos, ponto G.  | 9  |
| FIGURA   | 3. Fluxograma de um processo de verificação (VASUDEVAN, 2006). Estão apre-<br>sentadas as etapas para implementação e verificação de um circuito integrado.  | 12 |
| FIGURA   | 4. Fluxo de trabalho de um processo de verificação (CHANG; KUNDERT, 2007).<br>Estão apresentadas as etapas para a verificação de um circuito integrado analó-<br>gico.   | 13 |
| FIGURA : | 5. Fluxo de trabalho do processo de verificação Adotado. Este fluxo foi simplifi-<br>cado tendo como base os dois processos vistos.  | 13 |
| FIGURA   | <ul><li>6. Fluxo de execução do Teste de Regressão proposto por (CHANG; KUNDERT, 2007), mostrando a integração com a análise de cobertura e geração dos bancos</li></ul>   |    |

|        | de teste.  | 16 |
|--------|--|----|
| FIGURA | 7. Diagrama de blocos de uma verificação funcional, sendo o bloco <i>Estímulos</i> responsável por aplicar sinais no banco de testes, o bloco <i>Validação da Saída</i> em verificar os sinais de saída e entrada dos modelos. E todos eles mais o <i>Circuito a ser Verificado</i> , compõem um Banco de teste. | 19 |
| FIGURA | 8. Estão destacados quatro blocos que constituem o banco de teste definido pela técnica do Modelo de Ouro. O objetivo é verificar o circuito por meio de uma análise matemática certificando os sinais provenientes deste circuito e do modelo de ouro.  | 21 |
| FIGURA | 9. Componentes representando um modelo a nível de abstração baixo através do uso de transistores e o seu modelo a nível de abstração elevado, o modelo comportamental, embaixo, constituído a partir de equações matemáticas   | 21 |
| FIGURA | 10. Fluxo de trabalho desempenhado pela ferramenta e pelo projetista para a simulação de um modelo utilizando a linguagem VHDL-AMS.  | 23 |
| FIGURA | 11. Fluxo de trabalho utilizada na verificação formal, concepção das proprieda-<br>des a serem certificadas por meio da especificação, inserindo-as no verificador<br>formal juntamente com a entidade a ser certificada.  | 26 |
| FIGURA | 12. Principais componentes necessários para a execução de uma técnica de verificação Mista. Pode-se perceber a tentativade de integrar alguns conceitos das ténicas baseadas por simulação com as não baseadas   | 28 |
| FIGURA | 13. Fluxo de trabalho de uma técnica de verificação mista, a Verificação em<br>Tempo de Execução, definida por (DONG et al., 2007a)  | 29 |
| FIGURA | 14. Fluxo de Trabalho utilizado pela metodologia proposta em paralelo com o fluxo de projeto do engenheiro projetista, tendo como pré-requisito, para os   |    |

dois fluxos, a especificação do projeto. Em destaque, está o instante em que os

|        | engenheiros trocam de informações.   | 30 |
|--------|--|----|
| FIGURA | 15. Diagrama de blocos da técnica de Monitoramento por Autômatos. O bloco principal é dividido me dois componentes, Estímulos e Monitor.   | 32 |
| FIGURA | 16. Detalhamento do Bloco Monitoramento, mostrando os seus componentes internos.   | 33 |
| FIGURA | 17. Representação das máquinas de estados que compõem os componentes Estí-<br>mulos e Monitor.   | 33 |
| FIGURA | 18. Representação em bloco do componente Estímulos implementada em uma linguagem comportamental, destacando as portas de entrada e saída   | 34 |
| FIGURA | 19. Representação em bloco do componente Monitor implementada em uma linguagem comportamental, destacando as portas de entrada e saída   | 35 |
| FIGURA | 20. Identificação dos componentes utilizados pela técnica do Modelo de Ouro, apresentando como dados de entrada para a sua implementação: o conjunto de estímulos, a entidade a ser verificada e o modelo de ouro.     | 37 |
| FIGURA | 21. Banco de teste utilizado na técnica de modelo de ouro incorporando os conceitos de verificação a nível de transação. Este banco apresenta como dado de entrada somente a figura de mérito que se deseja verificar. | 38 |
| FIGURA | 22. Topologia do amplificador operacional utilizado como objeto de verificação do presente estudo de caso.   | 46 |
| FIGURA | 23. Leiaute do Amplificador Operacional, conforme a topologia apresentada na figura 22.  | 48 |

FIGURA 24. Máquinas de estados implementados nos blocos estimulo e monitor, mos-

|                     | trando como e quando devem ocorrer a aplicação dos estímulos e do seu moni-<br>toramento para a certificação da propriedade <i>pGBW</i> .  | 53 |
|---------------------|--|----|
| FIGURA              | 25. Máquinas de estados implementados nos blocos estimulo e monitor, mos-<br>trando como e quando devem ocorrer a aplicação dos estímulos e do seu moni-<br>toramento para a certificação da propriedade <i>qMF</i> .  | 54 |
| FIGURA              | 26. Máquinas de estados implementados nos blocos estimulo e monitor, mos-<br>trando como e quando devem ocorrer a aplicação dos estímulos e do seu moni-<br>toramento para a certificação da propriedade <i>sCMRR</i> .  | 55 |
| FIGURA              | 27. Máquinas de estados implementados nos blocos estimulo e monitor, mos-<br>trando como e quando devem ocorrer a aplicação dos estímulos e do seu moni-<br>toramento para a certificação da propriedade <i>tPSRR</i> .  | 55 |
| FIGURA              | 28. Máquinas de estados implementados nos blocos estimulo e monitor, mos-<br>trando como e quando devem ocorrer a aplicação dos estímulos e do seu moni-<br>toramento para a certificação da propriedade <i>qqSR</i> .   | 57 |
| GRÁFIC(             | O 2. Sinais de saída e entrada do modelo de ouro e do circuito aplicadas pelo estimulador. Assim como, as transições das máquinas de estados dos blocos estimulo e monitor. Os resultados representam os dados analisados para a validação da propriedade $pGBW$ .       | 58 |
| GRÁFIC              | O 3. Sinais de saída e entrada do modelo de ouro e do circuito aplicadas pelo estimulador. Assim como, as transições das máquinas de estados dos blocos estimulo e monitor. Os resultados representam os dados analisados para a validação da propriedade <i>sCMRR</i> . | 58 |
| GRÁFIC <sup>(</sup> | O 4. Sinais de saída e entrada do modelo de ouro e do circuito aplicadas pelo estimulador. Assim como, as transições das máquinas de estados dos blocos estimulo e monitor. Os resultados representam os dados analisados para a validação da propriedade <i>tPSRR</i> . | 59 |

| GRÁFIC( | O 5. Sinais de saída e entrada do modelo de ouro e do circuito aplicadas pelo estimulador. Assim como, as transições das máquinas de estados dos blocos estimulo e monitor. Os resultados representam os dados analisados para a validação da propriedade <i>qMF</i> . | 60 |
|---------|--|----|
| GRÁFIC( | O 6. Sinais de saída e entrada do modelo de ouro e do circuito aplicadas pelo estimulador. Assim como, as transições das máquinas de estados dos blocos estimulo e monitor. Os resultados representam os dados analisados para a validação da propriedade qqSR.        | 61 |
| FIGURA  | 29. Banco de testes utilizado para verificar o ganho diferencial do amplificador operacional.  | 63 |
| FIGURA  | 30. Banco de testes utilizado para a validação da taxa CMRR do amplificador operacional.   | 63 |
| FIGURA  | 31. Banco de testes utilizado para verificar a taxa PSRR do amplificador opera-<br>cional.   | 64 |
| FIGURA  | 32. Banco de testes utilizado para a validação da taxa <i>Slew Rate</i> do amplificador operacional.   | 64 |
| GRÁFICO | D 7. Curvas analisadas pelo bloco <i>Validador</i> para certificar o ganho diferencial do circuito. Conforme pode ser visto, o ganho diferencial do circuito implementado é superior a do modelo de ouro.  | 65 |
| GRÁFICO | O 8. Curvas analisadas pelo bloco <i>Validador</i> para certificar o CMRR do circuito.<br>Conforme pode ser visto, o CMRR do circuito implementado apresenta um com-<br>portamento melhor comparada ao do modelo de ouro.  | 66 |

GRÁFICO 9. Curvas analisadas pelo bloco Validador para certificar o PSRR do circuito.

Conforme pode ser visto, o circuito implementado apresentado uma maior rejeição da variação da tensão de alimentação no seu sinal da saída quando comparado com o do modelo de ouro, certificando o circuito nesta figura de mérito. 67

| GRÁFICO 10. Curvas analisadas pelo bloco Validador para certificar o Slew Rate do cir- |    |
|--|----|
| cuito. Conforme pode ser visto, o circuito implementado apresentado um aclive          |    |
| e declínio melhor do modelo de ouro quando aplicada um sinal da forma de               |    |
| degrau no terminal de entrada, certificando o circuito nesta figura de mérito.         | 68 |

| FIGURA 33. Representação geométrica do transistor de efeito de campo, destacano seus terminais e suas camadas.  | lo os<br>76          |
|---|----------------------|
| FIGURA 34. Circuito utilizado para validação da propriedade <i>pIdVds</i>   | 83                   |
| GRÁFICO 11. Curvas utilizadas para a validação da propriedade pIdVds, utilizando fonte variando a tensão entre os terminais de dreno e fonte do transistor par tensão de porta de 2.0 V, 2.5 V e 3.0 V.     | o um<br>a um<br>83   |
| FIGURA 35. Circuito utilizado para a validação da propriedade <i>sgmVgs</i>   | 84                   |
| GRÁFICO 12. Curvas utilizadas para a validação da propriedade <i>sgmVgs</i> , dados ob com a simulação do circuito mostrado na figura 35 para uma tensão de drei 1.31V, 1.81V e 2.5 V.                      | tidos<br>no de<br>84 |
| GRÁFICO 13. Curvas utilizadas para verificar a propriedade <i>tCgg</i> , através da simul do circuito apresentado na figura 35.   | lação<br>85          |
| GRÁFICO 14. Curvas utilizadas para validação da propriedade qCgx, apresentando va<br>iniciais iguais, para a tensão entre os terminais Dreno-Fonte igual a zero,<br>comportamento continuo até a saturação. | lores<br>e um<br>85  |

FIGURA 36. Circuito utilizado para a execução do teste de simetria dos modelos ACM e

- FIGURA 37. Circuito utilizado para se obter o erro na corrente de dreno quando utilizados espelhos de corrente com associações de transistores em serie-paralelo, quando são utilizados transistores de canal curto.
  87

- GRÁFICO 18. Validação da relação da corrente de dreno e sua transcondutância com a variação do nível de inversão para várias tensões de corpo. Com o incremento desta tensão, o valor da relação deverá tender para o valor  $1/\phi_t$  e apresentando valores não constantes na inversão fraca. 91

## Lista de Quadros

| QUADRO 1 | Especifica | ção do Am | plificador O | peracional |  | 42 |
|----------|------------|-----------|--------------|------------|--|----|
|----------|------------|-----------|--------------|------------|--|----|

QUADRO 2 Cronograma a ser seguido pela equipe de verificação. ...... 45

| QUADRO 3 Dimensionamento        | dos componentes, utilizando um capacitor de |    |
|---------------------------------|---|----|
| 0.8 pF na saída do amplificador |   | -8 |

QUADRO 4 Cronograma a ser seguido pela equipe de verificação para valida-ção da implementação do modelo ACM.75

## Lista de Códigos

| CÓDIGO 2        | Entidade referente a implementação do mod | lelo comportamen- |    |
|-----------------|---|-------------------|----|
| tal do amplific | ador operacional, escrita em VHDL-AMS     |                   | 51 |

| CODIGO 3        | Entidade VHDL-AM  | S, mostrando a | a interface | do modelo de | 2    |
|-----------------|-------------------|----------------|-------------|--------------|------|
| um transistor p | ara um modelo ACM |                |             | •••••        | . 81 |

,

# Lista de abreviaturas e siglas

| CAD     | Computer-aided design                          |
|---------|--|
| UCCM    | Unified Control Charge Model                   |
| Bsim3v3 | Berkeley Short-channel IGFET Model, versão 3.3 |
| SoC     | System OnChip                                  |
| DIBL    | Drain Induced Barrier Lowering                 |
| IP      | Intellectual property                          |
| ACM     | Advanced Compact Model                         |
| UDM     | User Definable Model                           |
| CMRR    | Common Mode Rejection Ratio                    |
| PSRR    | Power Supply Rejection Ratio                   |
| GBW     | Gain BandWidth                                 |
| ADMS    | ADvance Mixer Signal                           |
| GPL     | GNU Public License                             |

# Lista de símbolos

| $I_D$                       | Corrente de dreno   |
|-----------------------------|---|
| $V_D$                       | Tensão no terminal Dreno referenciada ao terminal Corpo           |
| $V_S$                       | Tensão no terminal Fonte referenciada ao terminal Corpo           |
| $\mu_n$                     | Mobilidade dos portadores   |
| W                           | Largura do canal  |
| L                           | Comprimento do canal  |
| $Q'_{ID(S)}$                | Densidade de cargas de inversão no dreno (fonte)                  |
| $C'_{ox}$                   | Densidade capacitiva do óxido de silício                          |
| $V_P$                       | Tensão de <i>pinch-off</i>  |
| $\mathbf{\phi}_t$           | Tensão térmica  |
| $Q'_{IP}$                   | Densidade de inversão de carga na condição de pinch-off           |
| n                           | Fator de rampa  |
| $V_{th}$                    | Tensão limiar   |
| <b>\$</b> <i>s</i> <b>0</b> | Parâmetro de ajuste do modelo ACM, definido como $2\phi_F$        |
| γ                           | Modela o efeito de corpo, considerado um parâmetro tecnológico    |
| $\phi_{sa}$                 | Tensão de superfície quando a densidade de cargas de inversão no  |
|                             | canal for igual a zero  |
| $T_0$                       | Tensão de referência  |
| т                           | Constante de ajuste, entre 1.2-1.4 para o transistor do tipo PMOS |
|                             | e 1.4-1.6 para o transistor do tipo NMOS                          |
| $\mu_{eff}$                 | Mobilidade efetiva  |
| $\alpha_{\theta}$           | Coeficiente de redução da mobilidade                              |
| $F_c$                       | Campo crítico no canal, parâmetro tecnológico                     |

## 1 Introdução

O lançamento de um produto, resultado do desenvolvimento de um circuito integrado, no mercado abrange vários aspectos técnicos e econômicos, como: um plano de viabilidade, treinamento da equipe, metodologia de concepção de circuitos, tecnologia de fabricação, ferramentas CAD, verificação, prototipagem, teste, entre outros. Dentre estas, destacam-se as questões relacionadas ao plano de viabilidade econômica, como: o custo de engenharia, o custo unitário do produto e o tempo de retorno do investimento.

O custo de engenharia para o desenvolvimento de um circuito integrado é a despesa relacionada à concepção do projeto, ou seja, as despesas geradas pela equipe de engenharia ao arquitetar uma solução prática, composta pela definição da especificação do sistema até a construção do primeiro protótipo (VAHID; GIVARGIS, 2001). Este protótipo é o primeiro projeto a ser construído fisicamente, o qual visa verificar a solução proposta pelos projetistas.

A concepção do projeto é uma das fases preliminares do ciclo de vida de um produto. Existem vários processos que a detalham, mostrando como deve ser executada, por exemplo: o *IpProcess* (LIMA et al., 2005). Neste processo são definidas quatro fases: concepção, elaboração, construção e transição, as quais abrangem desde a definição do escopo do projeto até a sua implantação, passando pela implementação, verificação e testes.

O custo unitário do componente é calculado a partir das despesas geradas na produção de um único exemplar do produto. Este valor dependerá de fatores tecnológicos, como: tamanho da área de silício utilizada, tecnologia de fabricação, topologia, consumo, mão-de-obra, empacotamento, entre outros. Estes fatores variam diretamente com as funcionalidades que o circuito integrado irá desempenhar. Como por exemplo, se uma nova versão do produto utilizasse a mesma tecnologia de fabricação para desempenhar mais funções do que a versão anterior, poderia ocasionar um aumento da área de silício, consequentemente, um aumento no seu custo unitário. Obviamente, esta previsão foi bastante simplificada já que outros fatores podem influenciar no cálculo deste custo, como: as diferentes topologias utilizadas nas duas versões do circuito ou técnicas de leiaute para redução da área de silício utilizada.

Por último, deve-se analisar o tempo de retorno de investimento, definido como o período

necessário para que as vendas do produto compensem os investimentos iniciais. O tempo calculado para este período é um dos principais indicadores da viabilidade econômica do produto a ser comercializado.

Uma forma encontrada para diminuir o tempo de retorno de investimento foi a massificação das vendas dos produtos de circuito integrado. Para tanto, foi necessária uma redução do custo unitário do produto. O aproveitamento da área de silício na fabricação foi uma das alternativas encontrada para reduzir este custo, por meio de um aumento da densidade de transistores por área de silício, alcançado com o avanço tecnológico. Desta forma, conseguiu-se construir circuitos integrados duplicando o número de transistores por área num período entre 18 a 24 meses, mantendo a veracidade da lei de Moore (MOORE, 1965).

O avanço na fabricação destes circuitos integrados só foi possível com aperfeiçoamento das quatro principais etapas químicas: oxidação, corrosão, dopagem e fotolitografia (BAKER, 2007). A fotolitografia é o principal processo químico utilizado, o significado do seu termo em latim é "escrita na pedra através da luz". Ela consiste na utilização de raios ultra-violetas para bombardear o *wafer*<sup>1</sup> dopado anteparado por máscaras de polímeros, as quais demarcam as regiões a serem removidas. Este *wafer* dopado é uma lâmina cilíndrica obtida utilizando silício puro bombardeado por partículas com o intuito de adquirir propriedades semicondutoras. Entretanto, o passo essencial foi a possibilidade de automatizar o sequenciamento destas etapas químicas durante a fabricação.

O principal custo na fabricação dos circuitos integrados é aquele gerado na produção das máscaras utilizadas na fotolitografia. O custo de fabricação de um conjunto de máscaras para a tecnologia<sup>2</sup> de 130nm é em média entre US\$ 300.000,00 e US\$ 400.000,00 (OOISHI, 2007). Este valor sobe para US\$ 1.000.000,00 quando se adota a tecnologia de 90nm e para US\$ 2.000.000,00 utilizando a tecnologia de 65nm. Atualmente, a Intel fabrica processadores utilizando a tecnologia de 45nm, como: o processador i7 (NOVINITE, 2008), para tal tecnologia o custo poderá ser maior do que US\$ 4.000.000,00 (OOISHI, 2007).

Apesar disso, houve uma diminuição no custo de engenharia no desenvolvimento de circuitos digitais, justificada pelo uso de placas de desenvolvimento, ferramentas CAD e linguagens de descrição de hardware, as quais são utilizadas para sintetizar modelos comportamentais diretamente em circuitos sem a necessidade de serem enviados, os projetos, para as fábricas de semicondutores. Com isso, foi possível implementar sistemas digitais com um alto grau de complexidade

<sup>&</sup>lt;sup>1</sup>uma fina fatia circular de material semicondutor

<sup>&</sup>lt;sup>2</sup>Medida do menor comprimento do canal de cada do transistor

À medida em que os circuitos se tornaram mais complexos, ocasionado pelo aumento do número de suas funcionalidades (CHANG; KUNDERT, 2007) percebeu-se um crescimento no número de falhas funcionais (ROBERTS; DUFORT, 1999). Uma das principais consequência disso, foi o aumento de produtos que não atingiram as especificações de projeto, decorrentes de exigências dos requisitos funcionais dos usuários.

Com o desenvolvimento de circuitos integrados integrando vários *IP Cores*<sup>3</sup>, constituindo o  $SoC^4$ , verificou-se um aumento significativo das falhas, as quais eram constatadas somente quando os dispositivos retornavam do processo de fabricação em larga escala e, algumas vezes, quando os produtos estavam sendo comercializados. Para diminuir o impacto destas, o uso de métodos que certifique o funcionamento correto destes dispositivos tornou-se extremamente importante. Para se ter uma idéia, a empresa Intel gastou 450 milhões de dólares para substituir o seu defeituoso Pentium (VASUDEVAN, 2006).

A prática de verificar está se tornando cada vez mais indispensável na concepção de circuitos integrados. Uma prova disso é que 65% dos protótipos concebidos em pastilhas de silício pela primeira vez apresentaram falhas e, desses, 70% não apresentariam falhas se estivessem sido submetidos a alguma metodologia de verificação (DUENAS, 2004). Com o intuito de se obter uma melhor visualização deste dado, foi construído o gráfico 1 no formato pizza, o qual representa o número de circuitos integrados produzidos. No gráfico, a região destacada por listras horizontais mostra a porcentagem de circuitos sem falhas, a com listras verticais representa os circuitos que falhariam mesmo se submetidos a algum tipo de metodologia de verificação e, por último, a região lisa representando a porcentagem dos circuitos que funcionariam corretamente se submetidos a alguma metodologia de verificação. Em consequência disso, surgiram várias abordagens, técnicas, linguagens, ferramentas e artefatos, com a finalidade de constituir um processo para se obter uma certificação do funcionamento dos circuitos integrados (VASU-DEVAN, 2006).

<sup>&</sup>lt;sup>3</sup>Entidade lógica protegida por uma propriedade intelectual

<sup>&</sup>lt;sup>4</sup>System-OnChip, Sistemas em Circuitos Integrados



Gráfico 1: Representação em pizza do número de circuitos integrados projetados, destacando a quantidade destes circuitos não apresentaram falhas, as que apresentaram e, em separado, as que não apresentariam se estivessem sido submetidos a uma metodologia de verificação

Com a importância da verificação adquirida ao longo dos anos, foi criada uma nova função, a do engenheiro de verificação, compondo uma equipe de verificação. Uma das principais funções desta equipe é garantir o funcionamento do projeto, concebido pelos engenheiros projetistas. Esta verificação deverá ocorrer no menor espaço de tempo possível (ANDERSON; NIGH-TINGALE, 2006). Ademais, um estudo detalhado de cada abordagem e técnicas de verificação deve ser realizado para se obter melhores resultados (PIZIALI, 2004) (BERGERON, 2003).

A partir de então, surgiram várias metodologias de verificação para circuitos, dando origem a ferramentas capazes de verificar circuitos automaticamente. Como exemplo, pode-se citar o *SystemVerilog* (BERGERON et al., 2005), que apresenta novos recursos e técnicas utilizadas no processo de verificação, como: reuso dos bancos de teste, utilização de linguagens orientadas a objetos, utilização de banco de teste a nível elevado de abstração, verificação de sistema baseada em advertência, análise de cobertura e teste randômico (ANDERSON JANICK BERGERON, 2006).

Várias metodologias de verificação atuais baseiam-se no *SystemC*, como o *VeriSC* (SILVA, 2007). Esta metodologia apresentou modificações visando a reduzir o tempo de execução do processo de verificação. Como exemplos destas modificações, têm-se: o desenvolvimento de bancos de teste antes da implementação do projeto, comparação entre os sinais provenientes dos modelos de referência e do circuito em tempo de simulação, assim como, a análise de cobertura, podendo essa alterar os bancos de teste para alcançar as metas de cobertura.

No caso dos circuitos analógicos, as despesas na concepção de um protótipo não tem diminuído com a mesma velocidade que reduziram aos dos circuitos digitais, permanecendo o seu custo de engenharia elevado. Isso é justificado pela falta de ferramentas capazes de sintetizar modelos comportamentais de forma automática em placas de desenvolvimento, como no caso dos circuitos digitais, para a realização de testes nos circuitos. Além de faltarem pessoas capacitadas na área e também modelos matemáticos compactos para simulação de circuitos, dentre outros fatores.

Os testes são efetuados em protótipos concebidos em pastilha de silício, enviados pelas fábricas de semicondutores depois de serem submetidos às etapas químicas. Este período, no caso de testes em circuitos digitais que podem ser testados em placas de desenvolvimento, apresenta uma redução no custo de execução. Com isso, o custo de engenharia na concepção de circuitos analógicos pode ser alto em relação aos circuitos digitais.

Atualmente, é possível integrar conjuntamente com blocos digitais em uma única pastilha de silício, os circuitos analógicos, constituindo os circuitos mistos. Os circuitos analógicos buscam explorar características, como: baixo ruído, baixo consumo, menor área, melhor performance, menor custo, por meio de novas arquiteturas e técnicas de projeto.

Por conta da especificidade das características vários circuitos analógicos retornaram das fábricas sem desempenhar corretamente as funcionalidades especificadas, aumentando o custo de engenharia. Isso é justificado, pelos custos das máscaras adicionais e de testes utilizados para certificar o circuito fabricado. E o custo desses alcança o valor de quase 50% do valor do produto final, e tende a aumentar para quase 100% (ROBERTS; DUFORT, 1999). A redução deste custo pode ser alcançada pela utilização de metodologias de verificação e processos que certifiquem estes circuitos antes de serem fabricados.

As metodologias de verificação de circuitos analógicos estão adotando os conceitos básicos das metodologias usadas para certificar os circuitos digitais (CHANG; KUNDERT, 2007). Um exemplo disso, são as adaptações do algoritmo de verificação de circuitos digitais, o *Model Checking* (BAIER JOOST-PIETER KATOEN, 2008), utilizado também para verificar os circuitos analógicos (HARTONG; HEDRICH; BARKE, 2002). No entanto, a sua utilização se restringe a alguns poucos circuitos analógicos, devido ao seu custo computacional, pela falta de ferramentas capazes de realizar a verificação automática.

Ao longo do estudo sobre verificação de circuitos analógicos, foi constatado que várias técnicas de verificação eram ineficientes devido ao custo computacional e tempo de lançamento do produto no mercado. Tendo em vista estes aspectos, o presente trabalho propõe a elaboração de uma metodologia de verificação para circuitos analógicos com o objetivo de proporcionar um maior nível de automatismo no processo de verificação.

Este trabalho é dividido em cinco seções além desta introdução.

A segunda seção, *Processo de Verificação*, define os conceitos básicos de um processo de verificação em circuitos integrados, fornecendo uma ênfase em circuitos analógicos. Assim como a elaboração de um plano de verificação, implementação de um ambiente de verificação,

construção de bancos de teste e geração de estímulos. Além disso, estão descritos os conceitos básicos da verificação de circuitos e o estado da arte das técnicas de verificação de circuitos analógicos.

Em seguida, a seção *Metodologia de Verificação para Circuitos Analógicos* descreve as principais desvantagens encontradas nas técnicas de verificação no estado da arte e mostra uma metodologia para verificar circuitos analógicos. O principal objetivo desta metodologia é reduzir o período de verificação mantendo o grau de confiabilidade, por meio do aumento do nível de automatização no processo de verificação.

Logo após, serão mostrados os resultados do uso desta metodologia em dois estudos de caso. Primeiramente, na validação de um amplificador operacional, detalhada na seção *Estudo de Caso A - Validação de um Amplificador Operacional*. E, na seção *Estudo de Caso B - Validação da implementação em C do modelo ACM*, apresenta os resultados do segundo estudo de caso.

Ao final tem-se a *Conclusão*, em que se encontra uma análise dos resultados dos estudos de caso e uma comparação desta metodologia com as outras existentes. Além disso, apresentam-se algumas propostas para trabalhos futuros.

## 2 Processo de Verificação

A presente seção tem como objetivo revisar os conceitos pertinentes na execução de um processo de verificação, utilizado na validação de circuitos integrados. Além disso, serão abordadas algumas técnicas de verificação de circuitos. Na próxima seção será detalhada a metodologia desenvolvida para verificar circuitos analógicos.

### 2.1 Verificação

O ato de projetar um circuito integrado consiste em traduzir a interpretação do projetista da especificação em algo a ser construído. Após o circuito ser projetado, ele será fabricado que no caso do circuito integrado, significa enviá-lo para uma fábrica, para que seja concebido em uma pastilha de silício. A investigação dos parâmetros obtidos do circuito fabricado, comprovando a equivalência entre esses e os parâmetros projetados é chamada de teste. E, a certificação do funcionamento do circuito projetado de acordo com a sua especificação é a verificação do circuito, que pode ser feita por simulação.

Como pode-se perceber na figura 1, a ação de verificar ocorre como um fluxo de trabalho de sentido contrário ao de projeto o circuito. Isso porque, o engenheiro de verificação dispõe do projeto, e o seu objetivo é encontrar, no circuito projetado, os dados definidos na especificação. Quando esta fase estiver finalizada com sucesso, o circuito é enviado para a fabricação, seguida da execução de testes certificadores da conformidade do funcionamento da entidade fabricada com o circuito projetado. Estas duas fases são responsáveis pela continuidade do projeto por confirmarem o seu correto funcionamento.



Figura 1: Uma macro-representação dos fluxos de trabalhos existentes em uma casa de projetos de semicondutores. Estão apresentadas as fases para construir o circuito, projeto e fabricação, e as de sentido inverso, verificação e testes, para certificar o seu funcionamento.

O comportamento dos circuitos analógicos apresenta uma significativa distinção quando comparado ao dos circuitos digitais. Isso deve-se principalmente ao complexo comportamento apresentado pelos transistores no domínio analógico, fortemente dependente da relação entre o comprimento e largura do canal, no caso dos transistores de efeito de campo. Consequentemente, a validação do funcionamento dos circuitos analógicos torna-se uma árdua tarefa em relação a dos circuitos digitais, podendo envolver vários tipos de análises, como: na frequência, no tempo ou na escolha do ponto de operação dos transistores. Além disso, a escolha de uma técnica de verificação dependerá do propósito do circuito, se o mesmo é linear ou não-linear, das funcionalidades desempenhadas e da sua especificação.

#### 2.1.1 Verificação Funcional

A verificação funcional é um tipo de certificação, tendo como principal objetivo comprovar o funcionamento de um circuito de acordo com a sua especificação. Em outras palavras, verificar funcionalmente é confirmar a preservação dos parâmetros descritos na especificação do circuito quando o mesmo estiver projetado.

Na construção de um circuito existem três aspectos a serem observados: a especificação do circuito; a sua implementação; e a intenção de projeto, definido como a interpretação do projetista a partir da especificação do circuito. O ato de confirmar a interseção entre estes três aspectos também é considerado como uma definição da verificação funcional.

Na figura 2, estão destacados três círculos que representam o universo de possibilidades de cada aspecto: o representado pela letra A indica o conjunto de funcionalidades do projeto, o destacado pela letra C representa o conjunto de funcionalidades da concepção do projeto e o indicado pela letra B destaca o conjunto de funcionalidades da especificação do projeto. A união (A U B U C) de todos estes conjuntos representa o potencial comportamento do circuito a ser fabricado. A verificação funcional deve promover a conciliação entre o conjunto de funcionalidades, especificação do projeto e a sua implementação, destacada pelo conjunto representado pela letra G (PIZIALI, 2004).



Figura 2: Representação dos universos das possibilidades de intenção do projetista, especificação do projeto e a sua implementação. O principal objetivo da verificação funcional é garantir a conciliação destes três universos, ponto G.

Para executar uma verificação funcional pode-se escolher um ou vários tipos de técnicas de verificação, podendo ser classificadas em: técnicas baseadas em simulação, as formais ou as mistas. Elas se distinguem pelos aspectos: escolha do tipo de análise matemática, algoritmo, métrica, modelo, linguagem, dentre outros.

Nas técnicas baseadas em simulação, o uso de simuladores numéricos como *spice*<sup>1</sup>, é necessário para encontrar o comportamento dos circuitos integrados em diversas condições de operação. Os resultados obtidos das simulações são analisados e confrontados com os dados da especificação.

No caso das técnicas não baseadas em simulação, ferramentas são utilizadas para a extração de modelos matemáticos simbólicos dos circuitos integrados. Estes modelos são submetidos a exaustivos testes por meio de algoritmos formais de verificação, certificando várias propriedades elaboradas a partir da especificação do circuito.

Com o propósito de obter melhores desempenhos e um aumento do formalismo matemático na verificação de circuitos, desenvolveram-se as técnicas de verificação mistas. Essas tem como objetivo conciliar as principais características das técnicas de verificação baseadas em simulação com as formais.

<sup>&</sup>lt;sup>1</sup>Simulated Program with Integrated Circuits Emphasis, ou Programa de Simulação com Ênfase em Circuitos Integrados)

#### A Análise de Cobertura

A Análise de Cobertura foi um dos artifícios utilizados pelo engenheiro de verificação na verificação funcional, para compreender quando o circuito integrado foi suficientemente exercitado, comprovando desta forma, a não ocorrência de falhas sob as condições de operação impostas. Para isso, foi elaborado o conceito de cobertura, ajudando a transcrever os objetivos e metas da verificação em código executável.

Uma verificação baseada na análise de cobertura consiste na implementação de bancos de teste, responsáveis em construir cenários simuláveis para a validação de cada item de cobertura. Desta forma, quando as simulações executarem todos os cenários, significa que as principais condições de funcionamento impostas foram avaliadas.

A validação dos cenários ocorre por meio da análise dos resultados provenientes das simulações, podendo utilizar uma análise por código, função ou asserção. A análise por código identifica se todo o código correspondente ao circuito (sua implementação) foi avaliado; a análise por função certifica se cada bloco do circuito está exercendo a funcionalidade conforme especificada; a análise por asserção consiste no uso de comentários imprimíveis na tela ou no relatório, acionados quando detectadas falhas de implementação (FORDE STEVE BISHOP, 1998).

A identificação dos buracos de cobertura é uma das principais dificuldades enfrentada no uso da análise por cobertura. Estes buracos são falhas de cobertura despercebidas pelos engenheiros, podendo ser provocadas pela falta de simulações ou pelo mau uso de estímulos, os quais não conseguiram estimular o circuito de maneira a identificar o comportamento falho. Em (LACHISH et al., 2002) são discutidos métodos que podem ser utilizados para identificar estes buracos de cobertura ao decorrer do processo de verificação.

### 2.2 Processo de Verificação

O processo de verificação é uma série de ações sistemáticas visando a certificação de um projeto de circuito integrado. Estas ações englobam vários aspectos tecnológicos e cronológicos. Dentre os aspectos mais importantes, pode-se citar: o fluxo de trabalho, a equipe de engenheiros, prazo de execução, metodologias e as técnicas de verificação. O principal objetivo deste processo é avaliar o projeto, indicando que o mesmo está apto a ser concebido fisicamente (WILE JOHN GOSS, 2005).

Para (VASUDEVAN, 2006) um processo de verificação inicializa com um documento de requisitos, a partir do qual define-se a especificação do produto e, juntamente, com as tecnologias adotadas consegue-se determinar a arquitetura adequada. A partir disso, inicializa-se a fase de planejamento, com o objetivo de elaborar o plano de verificação e definir os circuitos a serem implementados. Em seguida, é implementa-se o circuito e conjuntamente com a execução do plano de verificação, quando pondo as técnicas de certificação de circuitos são postas em prática. Para finalizar o processo, os resultados são medidos, utilizando métricas definidas na fase de planejamento. Este fluxo de trabalho está mostrado na figura 3.



Figura 3: Fluxograma de um processo de verificação (VASUDEVAN, 2006). Estão apresentadas as etapas para implementação e verificação de um circuito integrado.

O fluxo de trabalho do processo de verificação proposto por (CHANG; KUNDERT, 2007) está mostrado na figura 4. Em destaque estão as funções do engenheiro de projeto e do engenheiro de verificação. O pré-requisito para a inicialização do processo é a elaboração do Documento de Projeto, contendo a definição do problema a ser atacado, a especificação e a arquitetura do circuito a ser construído, além dos requisitos funcionais e não-funcionais do projeto. O próximo passo é a definição dos Riscos de Projeto, mostrando quais são os requisitos ou comportamentos, como baixo consumo ou largura de banda, mais difíceis de serem alcançados. Com este documento pronto, é possível elaborar o Plano de Verificação. Em seguida, são elaborados os Planos de Modelagem e Simulação e construídos os Bancos de Testes Auto-Avaliadores, cujas finalidades serão de mostrar os objetivos específicos na construção dos modelos utilizados e na execução da simulação e de implementar os banco de testes, respectivamente. Depois destas

etapas, é executada os Testes de Regressão, responsável em informar aos engenheiros se as metas definidas no plano de verificação foram atendidas.



Figura 4: Fluxo de trabalho de um processo de verificação (CHANG; KUNDERT, 2007). Estão apresentadas as etapas para a verificação de um circuito integrado analógico.

Com base nos dois processos de verificação revisados nos dois parágrafos precedentes, foi elaborado o fluxo de trabalho de um processo de verificação adotado neste trabalho, destacado na figura 5. Como pode-se perceber, o processo é inicializado com a especificação do circuito. Em seguida, é elaborado o plano de verificação, o qual engloba o plano de modelagem e o plano de simulação. A partir desses, são construídos os modelos comportamentais, e os ambientes de verificação, composto pelos Banco de Testes Auto-Avaliadores. Em paralelo, é projetado o circuito. E para finalizar, é executado o Teste de Regressão, certificando se o circuito foi totalmente verificado. Caso o circuito não esteja validado, os resultados destes testes podem fornecer informações aos projetistas para aperfeiçoar o circuito, aprovando-o nos próximos testes.



Figura 5: Fluxo de trabalho do processo de verificação Adotado. Este fluxo foi simplificado tendo como base os dois processos vistos.

Ao longo desta seção serão detalhadas as principais etapas deste fluxo: Elaboração do Plano de Verificação, Execução do Teste de Regressão, Implementação do Banco de Testes Auto-Avaliadores. Em seguida, serão vistas as características que compõem um ambiente de verificação.

#### 2.2.1 Elaboração do Plano de Verificação

O plano de verificação é o documento inicializador do ato de verificar. Este documento define: as metas de verificação, os métodos, membros da equipe e seus perfis, técnicas de verificação, cobertura, linguagens, ferramentas, prazo e cronograma. O primeiro item a ser definido é o escopo da verificação, para em seguida, definir qual é a cobertura a ser cumprida pelo proceso. Em seguida, deve-se escolher as técnicas e métodos, e, posteriormente, implementá-las nos ambientes de verificação com as linguagens e ferramentas adotadas (BERGERON, 2003).

Todos estes itens devem ser definidos juntamente com o documento de riscos de projeto. Este documento destaca os potenciais problemas que podem surgir durante todo o processo de concepção do projeto, incluindo a sua verificação. Com a ajuda deste levantamento de riscos, o engenheiro de verificação identificará as principais figuras de mérito a serem utilizadas para verificar o circuito.

A descoberta prematura destes riscos pode auxiliar na elaboração de estratégias para detecção de falhas. Pode-se até mesmo conciliar duas técnicas de verificação, ou alterar a abordagem de verificação, escolhendo uma abordagem baseada em simulação ou não baseada em simulação.

No plano de verificação deve constar também o nível de automatização do processo de verificação. Quanto maior for este nível melhores serão os resultados obtidos, porque quanto menor for a interação humana maior será a confiabilidade na execução do processo (BERGE-RON, 2003). A utilização de banco de testes parametrizável, possibilitando o ajuste automático de parâmetros ao longo do processo de verificação para atender a análise de cobertura, pode ser uma forma de aumentar a automação do processo, conseguindo obter resultados mais rápidos, até mesmo, para otimizar o funcionamento do circuito (ANDERSON JANICK BERGERON; NIGHTINGALE, 2006b).

As metas descritas no plano de verificação, serão listadas na forma de propriedades, elaboradas na forma de sentenças condicionais a serem verificadas no comportamento do circuito. Se todas as propriedades forem satisfeitas, a certificação do circuito será finalizada, informando que o circuito executa todas as funcionalidades especificadas.

Um plano de verificação de sucesso é peremptório para a redução do custo de projeto. Com a utilização de bons planos de verificação pode-se ter uma redução dos custos em testes, geralmente bastante elevados nos circuitos analógicos (ROBERTS; DUFORT, 1999). A comprovação da elaboração de um bom plano de verificação é confirmada quando se tem o funcionamento total de um sistema no seu primeiro ciclo de fabricação, designado pelo termo, em inglês, *first silicon sucess*.

Além dos tópicos mencionados, existe uma seção no plano de verificação, dependente da técnica de verificação adotada, que aborda a geração de estímulos, detalhada na próxima subseção

#### 2.2.1.1 Geração de Estímulos

A Geração de Estímulos é o procedimento de criar sinais em forma de tensão ou corrente a serem aplicados nos modelos matemáticos simuláveis, utilizados nas técnicas de verificação baseadas em simulação. Estas técnicas utilizam ferramentas de análise numérica, como os simuladores *spice*, para obter os resultados das condições de operação dos circuitos.

A verificação completa do circuito depende de um conjunto de estímulos. Este conjunto deve ser composta por várias formas de onda capazes de estimular o circuito para verificar as propriedades listadas no plano de verificação. O número de estímulos tem que ser é um número mínimo possível para um alcance de 100% de cobertura.

Como previsto na lista de propriedades, no conjunto de estímulos podem estar relacionadas

formas de onda mas para demonstrar o funcionamento dos circuitos em situações adversas, podendo ser: pela variação das condições de operação ou pela aplicação de novos estímulos (OZEV; ORAILOGLU, 2002). No entanto, este tipo de prática deve ser feita de forma limitada, para não aumentar o custo computacional desnecessariamente.

Na subseção *Técnicas de Verificação* serão vistas algumas técnicas para auxiliar na geração do conjunto de estímulos, como a utilização de banco de testes a nível de transação e da técnica dos Testes Randômicos.

#### 2.2.2 Implementação dos Bancos de Teste Auto-Avaliadores

Os Bancos de Testes Auto-Avaliadores são entidades geradoras de estímulos e avaliadoras de sinais provenientes dos modelos, informando aos projetistas as falhas encontradas (CHANG; KUNDERT, 2007). A execução destes bancos de testes é uma forma de aumentar o nível de automatização do processo de verificação. Como mencionado anteriormente, uma das formas de evitar erros é pela utilização de procedimentos automatizados (WILE JOHN GOSS, 2005).

Para a construção deste tipo de banco podem ser utilizadas duas técnicas: Verficação por modelo de ouro ou Verificação por conformidade dos sinais com o tempo.

#### 2.2.3 Execução dos Testes de Regressão

A execução do Teste de Regressão é a última etapa no processo de verificação, responsável pela avaliação dos resultados dos modelos. Normalmente, quando o projeto é de grande porte, são necessárias várias simulações com distintas formas de onda, podendo durar dias para serem concluídas, resultando em uma imensa quantidade de dados a analisar. Para racionalizar este tanto foram criados estes testes (MEYER, 2003) (CHANG; KUNDERT, 2007).

Estes testes são executados periodicamente durante todo o processo para informar se as metas de cobertura estão sendo satisfeitas. Desta forma, caso haja algo de errado durante as simulações, são notificados os projetistas e engenheiros de verificação.

Na figura 6, pode-se visualizar o fluxo de trabalho de um Teste de Regressão. Primeiramente, é analisada a cobertura alcançada para mostrar que o circuito foi totalmente verificado, se não, os bancos de testes serão gerados novamente com novos estímulos. E, em seguida, as simulações serão reinicializadas e é executada a análise de cobertura novamente. Se todas as propriedades forem satisfeitas, o teste de regressão finalizará o processo de verificação.



Figura 6: Fluxo de execução do Teste de Regressão proposto por (CHANG; KUNDERT, 2007), mostrando a integração com a análise de cobertura e geração dos bancos de teste.

A Análise da Cobertura, um dos passos do Teste de Regressão, é um estudo dos resultados obtidos ao longo do processo de verificação, uma forma de rastreiá-lo. No plano de verificação, a cobertura do processo pode ser definida mostrando como os estímulos e validadores devem ser particionados para verificar todas as propriedades. E, com a medição da cobertura, podese avaliar se o processo está sendo executado conforme planejado. Além disso, esta medição permite identificar o instante em que o processo de verificação deverá ser finalizado.

Com isso, foram apresentados os principais passos que compõe a execução de um processo de verificação em circuitos integrados, fornecendo uma ênfase nos circuitos analógicos. A seguir, está detalhada como deve ser a implementação de um ambiente de verificação.

### 2.2.4 Construção do Ambiente de Verificação

O ambiente de verificação é uma ferramenta na qual o projeto a ser verificado é submetido à avaliação por uma técnica de verificação. Este ambiente é composto pela implementação dos bancos de testes, geradores de estímulos, avaliadores dos sinais e da implementação da técnica de verificação adotada.

A construção de um ambiente de verificação envolve vários aspectos técnicos, como: escolha das linguagens de programação, sistema operacional, ferramenta CAD, técnica de verificação, simulador, entre outros. Além disso, devem ser elaborados, nestes ambientes, *scripts* para automatizar a integração das várias partes do processo de verificação.

Ao implementar os ambientes de verificação, uma das principais características adotada deve ser o re-uso de código para futuras utilizações em processos de verificação. A redução
do tempo de verificação de projetos futuros será o aspecto mais beneficiado pelo uso deste princípio.

O ambiente de verificação deve ser construído no instante em que o plano de verificação for finalizado. Este ambiente funciona como um programa procedural em que o engenheiro fornece a entidade a ser verificada, bem como, as propriedades a serem validadas. O principal resultado da execução deste ambiente é um relatório, informando se o projeto satisfaz a especificação. Evidentemente, a intervenção humana na execução destes ambientes deve ser minimizada.

A obtenção de resultados satisfatórios dependerá de um ambiente de verificação bastante consolidado, proporcionando bons resultados e de forma eficiente. Estes ambientes não podem ser confundidos com a construção de bancos de teste.

# 2.3 Escolha de uma Técnicas de Verificação

As técnicas de verificação são conjuntos de métodos e procedimentos essenciais à certificação de um projeto de circuito integrado. A escolha de uma técnica depende principalmente das funcionalidades desempenhadas pelo circuito, no caso dos analógicos, podem ser: para comunicação, conversão ou amplificação de sinais. Além disso, dependerá da especificação definida pelo engenheiro projetista, podendo resultar em diversos tipos de análises, como: na frequência, no tempo, no cálculo do ponto de operação, na variação dos parâmetros do processo de fabricação. A utilização de técnicas provenientes dos circuitos digitais para verificar circuitos analógicos se restringem a poucos circuitos, devido ao complexo variedade de condições de operações que precisam ser testados nos circuitos analógicos.

Normalmente, as técnicas de verificação são baseadas em dois tipos de análise matemática: análise simbólica ou análise numérica. Nos simuladores, como o *spice*, são utilizadas as análises numéricas para se obter o comportamento dos circuitos integrados. Por outro lado, para executar uma análise formal matemática é necessária a obtenção das equações matemáticas através do uso de uma análise simbólica dos circuitos, caracterizando as técnicas não baseadas em simulação. Existem técnicas de verificação que utilizam alguns conceitos da verificação por simuladores juntamente com algumas características das técnicas formais, as chamadas técnicas semi-formais.

As vantagens e desvantagens de cada uma dessas serão discutidas ao longo das próximas subseções.

### 2.3.1 Técnicas baseadas em Simulações

Nas técnicas baseadas em simulação, o banco de teste é o principal bloco a ser implementado. Estes bancos são as entidades que aplicarão os estímulos no circuito juntamente com os componentes necessários para simular a condição a ser verificada, mostrando ao término da simulação os resultados para o engenheiro de verificação analisar. Os bancos de teste são construídos utilizando uma linguagem de descrição de hardware, como: Spice, VHDL, Verilog, SystemC, VHDL-AMS, Verilog-A, SystemC-AMS, entre outras.

A validação de cada parâmetro contido na especificação do projeto ocorre com a execução de um banco de teste especifico. Desta forma, são aplicados e analisados sinais na forma de tensão ou corrente no circuito de modo a certificar as propriedades definidas no plano de verificação

A figura 7 mostra os principais componentes de uma verificação baseada em simulação: a especificação, os estímulos, o circuito a ser verificado e o analisador de sinais provenientes do circuito, sendo estes três últimos, os componentes de um banco de teste.



Figura 7: Diagrama de blocos de uma verificação funcional, sendo o bloco *Estímulos* responsável por aplicar sinais no banco de testes, o bloco *Validação da Saída* em verificar os sinais de saída e entrada dos modelos. E todos eles mais o *Circuito a ser Verificado*, compõem um Banco de teste.

O bloco *Estímulos* é responsável por aplicar sinais de tensão ou corrente para verificar uma propriedade. A utilização de algumas técnicas para a construção de um bloco de estímulo mais sofisticado pode ser requisitada, dependendo da complexidade do circuito.

O bloco validador, *Validação da Saída*, é responsável por certificar os sinais provenientes do projeto implementado conforme a especificação do circuito. Análise do resultado feita neste bloco é responsável por finalizar o processo de verificação, podendo verificar o circuito implementado.

Os dois blocos anteriores compõem um Banco de teste, responsável pela integração dos blocos com os simuladores disponíveis no ambiente de verificação, além de proporcionar dados

para a escrita de um diagnóstico ao término do processo de verificação.

Estas técnicas podem verificar circuitos de várias formas, em diferentes níveis de abstração ou em diferentes domínios. A escolha da forma como se deve verificar o circuito dependerá das suas funcionalidades, especificação e das metas de verificação.

Em relação a distintos níveis de abstração tem-se: a verificação de sistemas, responsável pela validação do sistema completo; a verificação de blocos, responsável em verificar somente partes do sistema.

No entanto, o engenheiro pode verificar o sistema ou seus blocos informando-lhes o parâmetro a ser verificado. Com isso, o ambiente de verificação se encarrega de aplicar os sinais de tensão ou corrente apropriados e de verificar os sinais de saída. Neste caso, os sinais de tensão e corrente para a certificação do parâmetro fornecido pelo engenheiro constará no banco de dados do ambiente de verificação.

Além disso, a verificação pode ser feita em níveis de abstração mais baixos como: os descritos na linguagem *spice* de transistor. Podendo incluir ainda os componentes parasitas da extração pós-leiaute, existentes devido às interconexões do circuito. Neste último caso, os resultados, da simulação do circuito, são considerados os mais próximos dos circuitos fabricados. No entanto, o custo computacional para a simulação de um circuito integrado completo a nível de leiaute é elevado, sendo utilizado somente quando todos os blocos individuais estejam minuciosamente verificados.

A deficiência desta técnica de verificação é a ausência do comprometimento com o formalismo matemático no processo de verificação. O uso do formalismo matemático proporciona uma maior confiabilidade na verificação do circuito. Além disso, ao término da verificação, o ambiente, ao qual se submete o circuito a ser verificado, não informa a existência de uma falha, e sim, a inexistência da falha para o qual o banco de teste foi construído.

### 2.3.1.1 Técnica do Modelo de Ouro

A técnica do modelo de ouro é um aperfeiçoamento da técnica vista na sub-seção anterior, para obter melhores resultados na verificação por meio de simuladores de circuitos (MARINO et al., 2005) (VASUDEVAN, 2006). Esta técnica utiliza comparadores lógicos para confrontar sinais provenientes do circuito implementado com os sinais de um modelo de referência ou de ouro. Este modelo é uma entidade desenvolvida com base nas especificações do circuito.

A figura 8 mostra o banco de teste proposto por esta técnica, destacando os seus principais blocos. Como pode ser observado, o banco proposto é dividido em quatro blocos: *Estímulo*,



Modelo de Referência, Circuito a ser Verificado e o Validador.

Figura 8: Estão destacados quatro blocos que constituem o banco de teste definido pela técnica do Modelo de Ouro. O objetivo é verificar o circuito por meio de uma análise matemática certificando os sinais provenientes deste circuito e do modelo de ouro.

O bloco *Estímulo* é utilizado para construção de um conjunto de sinais, construído pela equipe de verificação, os quais devem ser aplicados simultaneamente no modelo de referência e no circuito a ser verificado. Este engenheiro deve escolher o menor número de estímulos para a validação de um maior número de propriedades. Entretanto, o uso inadequado de estímulos pode levar a resultados de verificação falhos.

O bloco *Modelo de Referência* é uma representação, podendo ser um modelo comportamental, fiel à especificação do circuito. A construção deste bloco deve ser desprovida de erros. Uma das vantagens do uso deste modelo, além da utilização nesta técnica, seria a validação da especificação de cada bloco do circuito por meio da simulação dos seus modelos de referência. Com isso, tem-se a certeza que os blocos foram bem especificados.

O bloco *Circuito a ser Verificado* é a entidade a ser verificada. Na verificação de circuitos, esta entidade deve ser a representação mais próxima da realidade, no caso, o leiaute do circuito projetado.

E, por último, o *Validador*, bloco responsável pela certificação do circuito implementado, a entidade certificadora do funcionamento do circuito conforme o modelo de referência. Existem várias formas de comparar os dois circuitos, um exemplo destes é através de comparadores lógicos.

A modelagem comportamental de um sistema é a representação do seu comportamento por meio de um algoritmo ou por meio de equações matemáticas, considerada também como um modelo funcional. A figura 9 mostra dois modelos, o superior é um circuito implementado e o inferior é um modelo comportamental de um amplificador ideal definido pela equação Vout = G\*Vin. Ao longo deste trabalho, foi adotada a linguagem VHDL-AMS para a construção dos modelos de ouro.



Figura 9: Componentes representando um modelo a nível de abstração baixo através do uso de transistores e o seu modelo a nível de abstração elevado, o modelo comportamental, embaixo, constituído a partir de equações matemáticas.

#### A linguagem VHDL-AMS

A linguagem VHDL foi desenvolvida em 1981 para a modelagem de sistemas digitais. Na década de 90, esta linguagem foi estendida para modelar também sistemas analógicos. Com isso, foi possível modelar sistemas mistos, originando a linguagem VHDL-AMS, em inglês, AMS significa *Analog Mixed Signal*. Este nome foi fornecido informalmente para a combinação dos padrões: VHDL 1076-1993 e VHDL 1076.1-1999 (ASHENDEN G. D. PETERSON, 2002).

Um modelo descrito em VHDL-AMS é dividido em duas principais partes: a *entidade* e a *arquitetura*.

Na *entidade* devem ser definidos o nome, as portas e os parâmetros genéricos do sistema a ser modelado. Com os parâmetros genéricos é possível parametrizar os modelos no momento da sua instanciação no banco de teste. A *arquitetura* é a implementação da entidade. No caso da modelagem comportamental, esta implementação será constituída por equações matemáticas (DOBOLI; VEMURI, 2003).

A divisão do modelo entre estas duas partes possibilita a construção de várias arquiteturas para somente uma entidade. A implementação de modelos de transistores pode ser um exemplo da vantagem desta divisão, pois na modelagem destes dispositivos, há vários modelos matemáticos na literatura. Com isso, pode-se ter uma única entidade para implementar todos os modelos. Para isso, necessita-se somente mudar as equações matemáticas de cada arquitetura de acordo com o modelo que se deseja implementar.

A base matemática para encontrar o comportamento de um modelo descrito na linguagem VHDL-AMS parte da solução de um sistema de equações ordinárias diferenciais e algébricas

$$F(x, dx/dt, t) = 0 \tag{1}$$

na qual F é o vetor de expressões do modelo, x é um número de variáveis, dx/dt é o vetor das derivadas das variáveis e t é o tempo.

Ao utilizar um método de discretização de variáveis, o simulador quantifica as soluções do sistema de equações, dando origem a uma nova classe de variáveis, *quantity*.

As *quantities* são utilizadas para declaração de praticamente todas as variáveis na modelagem utilizando VHDL-AMS. Estas variáveis são úteis em simulações que necessitem de conservação de energia.

A simulação VHDL-AMS se divide em três fases: a elaboração, a inicialização e a simulação no domínio do tempo ou da frequência.

Na fase da elaboração, compilam-se os códigos fontes incluindo as entidades, as arquiteturas e os pacotes. Nesta fase, cria-se um código, de acordo com o modelo, para ser utilizado pelo simulador. Na fase seguinte, a de inicialização, o simulador por meio deste código define os valores iniciais para as *quantities* de acordo com o ponto de operação imposto pelo banco de teste, satisfazendo o sistema de equações 1, tais valores são conhecidos como os pontos de convergência. Após a inicialização, novos valores de convergência podem ser encontrados pelo simulador. Finalmente, quando o ponto de convergência sob a condição de operação esperada é encontrado, executa-se uma simulação em outro domínio para a verificar o comportamento do modelo em função da frequência ou do tempo. A figura 10 mostra o fluxo seguido pelo simulador VHDL-AMS.



Figura 10: Fluxo de trabalho desempenhado pela ferramenta e pelo projetista para a simulação de um modelo utilizando a linguagem VHDL-AMS.

Um exemplo de um código escrito na linguagem VHDL-AMS pode ser visto no código 1. Nele pode-se visualizar as duas principais partes de um código VHDL-AMS, definidas pelas palavras *entity* e *architecture*, além de alguns tipos de variáveis, como: *generic*, *port*, *terminal*, *quantity*, entre outras.

Código 1: Exemplo de um amplificador diferencial descrito na linguagem VHDL-AMS, mostrando o uso de variáveis do tipo *generic* e definição das portas do modelo na entidade.

```
entity ampdiff is
            generic (
                    G
                                     : real := 10000.0;
            );
5
            port (
                    terminal V1
                                   : electrical;
                    terminal V2
                                   : electrical;
                    terminal Vo
                                   : electrical
              );
10
   end entity ampdiff;
   architecture simple of ampdiff is
   quantity Vmais across V1 to electrical_ref;
15
   quantity Vmenos across V2 to electrical_ref;
   quantity Vout across Ivout through Vo to electrical_ref;
   quantity Vin: voltage;
20
   begin
           Vin == Vmais - Vmenos;
           Vout == G * Vin
   end architecture simple;
```

Neste exemplo, é apresentada a modelagem de um amplificador diferencial ideal. Primeiramente, pode-se visualizar a entidade definida, a *ampdiff*. Nela define-se os terminais do tipo elétrico, *electrical*: *V1*, *V2* e *Vo*. Também pode-se perceber a definição de uma variável *generic* do tipo *real*, *G*, no valor padrão de 10000.

Em seguida, tem-se a definição de uma arquitetura para a entidade *ampdiff*. Nessa, são definidas quatro *quantities*:  $V_{mais}$  e  $V_{menos}$  definidas como valores dos terminais V1 e V2 respectivamente, referenciadas ao terra pela constante *electrical\_ref*; a *quantiy* do tipo *voltage*  $V_{in}$ , definida pela equação diferença entre as duas *quantities* anteriores ;e a *quantity*  $V_{out}$ , representando a tensão de saída do modelo e definida pela equação  $V_{out} = G * Vin$ .

#### 2.3.1.2 Verificação a Nível de Transação

A verificação a nível de transação é uma das técnicas utilizadas na verificação funcional com o intuito de facilitar a construção e modificação dos banco de testes (HASSOUN et al., 2005). O conceito de transação é definido como a transferência de dados de um nível de abstração elevado em níveis mais baixos, auxiliando na parametrização dos bancos de testes (CHANG; KUNDERT, 2007) (ZHONG-HAI; YI-ZHENG, 2005). Resumidamente, uma interface é construída para auxiliar na construção do estímulo a ser aplicado por meio da tradução do parâmetro a ser verificada.

Na verificação analógica, esta técnica pode ser definida como uma interface, responsável em resgatar os parâmetros da especificação e modificar automaticamente os bancos de teste para aplicar as tensões ou correntes corretas nos modelos.

Por exemplo, se o engenheiro deseja verificar o ganho do circuito, o ambiente de verificação deve acionar uma interface para alterar o estímulo do banco de teste, modificando-o para um estímulo de pequenos sinais e executar uma simulação no domínio da frequência. Com isso, é possível adaptar os bancos de teste automaticamente para obter um maior alcance de cobertura.

#### 2.3.1.3 Técnica do Testes Randômicos

A técnica dos Testes Randômicos foi desenvolvida pelos engenheiros de verificação para a geração de estímulos aleatórios (BHASKAR et al., 2005). O objetivo desta técnica é estimular o dispositivo utilizando um maior número de sinais diferentes possíveis, possibilitando a obtenção de uma maior cobertura no processo de verificação.

Os circuitos digitais complexos apresentam um número elevado de possibilidades de estímulos dependente da quantidade de portas de entrada e saída, e de seu barramento interno. Neste caso, a verificação de todas as possibilidades para se obter resultados significativos poderiam se estender por anos. Então, resolveu-se criar um modo para exercitar os circuitos da forma mais diversificada possível.

Entretanto, o uso incorreto desta prática pode levar a exaustivas simulações sem obter resultados significativos. Apesar disso, esta técnica com a ajuda da análise de cobertura e banco de teste a nível de transação proporciona uma técnica largamente utilizada nos processos de verificação.

### 2.3.2 Técnicas não baseadas em Simulações

As técnicas de verificação não baseadas em simulação são métodos essenciais à certificação de circuitos desprovidos de estímulos (PERRY, 2005). O processo de verificação baseada em resultados de simuladores numéricos, como o *spice*, apresenta resultados rápidos. No entanto, os circuitos são estimulados por sinais de tensão e corrente. Desta forma, A complexidade do circuito ocasiona a sofisticação dos estímulos, aumentando as chances de ocorrer falhas na estimulação, sendo esses, em alguns casos, incapazes de exercitar os circuitos corretamente. Com isso, os resultados incorretos serão inevitáveis no processo de verificação (FUJITA; GHOSH; PRASAD, 2007).

Na verificação formal, como são chamadas as técnicas de verificação não baseadas em simulação, extraem-se parâmetros do circuito para a construção de um modelo matemático. Com este modelo, verificam-se as propriedades escritas, dependendo da técnica formal adotada, em forma de modelos comportamentais ou linguagens formais. Limitam-se o espaço de soluções das equações correspondente à região de operação dos dois modelos, verifica-se o circuito sem a utilização de estímulos (ZAKI; TAHAR; BOIS, 2006a). Esta limitação do espaço de soluções depende do número de variáveis relevantes a serem analisadas no comportamento do circuito projetado.

A figura 11 mostra as principais partes de uma verificação formal. Primeiramente, se faz uma interpretação da especificação em propriedades, escrita por meio de uma linguagem formal. Em seguida, estas propriedades são colocadas no verificador formal, responsável por transformar o circuito em um modelo matemático e por executar um algoritmo formal de verificação e, finalmente, informa se o modelo satisfaz as propriedades.



Figura 11: Fluxo de trabalho utilizada na verificação formal, concepção das propriedades a serem certificadas por meio da especificação, inserindo-as no verificador formal juntamente com a entidade a ser certificada.

Como exemplo de técnicas de verificação formal, pode-se citar: o *Equivalence Checking* (WAMBACQ; GERRITS, 2001) (HEDRICH; BARKE, 1998) (DRECHSLER, 2004), Model checking (STEINHORST; HEDRICH, 2008) (HARTONG; HEDRICH; BARKE, 2002) (GRABOWSKI DA-NIEL PLATTE; BARKE, 2005) e, também, tem-se a verificação por meio dos Teoremas provadores (GHOSH; VEMURI, 1999) (SUTCLIFFE, 2002).

Apesar do rigor matemático na certificação dos circuitos, o uso de um verificador formal ainda é pouco utilizado na verificação de circuitos analógicos. Seu custo computacional é elevado para a construção de um ambiente de verificação utilizando algoritmos de verificação formal adequados para circuitos analógicos.

# 2.3.3 Técnicas Mistas

As técnicas de verificação semi-formal, ou mista, são métodos de verificação de circuitos que combinam algumas características das verificações formal e não-formal (LI MITCHELL THORN-TON, 2002). Nesta abordagem, os comportamentos dos circuitos são obtidos por meio de anali-sadores numéricos, como o *spice*. Entretanto, para a construção do validador é utilizada alguns conceitos da verificação formal, como: certificação por meio de procedimentos automáticos ou algoritmos de verificação. Enquanto que na verificação não-formal, o circuito é certificado por um procedimento manual, no qual são utilizados inspeções visuais ou textuais dos resultados (DONG et al., 2007b).

A construção deste tipo de técnica é uma tentativa de construir verificadores mais confiáveis, mantendo a objetividade dos resultados de simulação (SALEM, 2002), já que que o formalismo da verificação não baseada em simulação aumenta o custo computacional do processo de verificação.

A figura 12 mostra as principais partes que compõem uma verificação semi-formal. Podem ser observados, a definição das propriedades, a geração dos estímulos e a validação dos sinais provenientes do circuito.



Figura 12: Principais componentes necessários para a execução de uma técnica de verificação Mista. Pode-se perceber a tentativade de integrar alguns conceitos das ténicas baseadas por simulação com as não baseadas

#### 2.3.3.1 Técnica de Verificação em Tempo de Execução

A verificação em Tempo de Execução é um método semi-formal, o qual descarta o uso de exploração espacial e inspeção exaustiva, como nas técnicas formais, sendo necessário somente o uso de monitores lógicos, fornecendo uma análise eficiente do comportamento dos circuitos.

Este método pode ser executado de duas formas: monitoramento *off-line* e monitoramento *on-line*. Na forma *off-line*, depois da finalização de toda a simulação, a validação dos resultados é executada. Na forma *on-line*, enquanto a simulação está em andamento, a verificação das propriedades está sendo processada. Esta última forma tem a vantagem de encontrar as violações antes da finalização de toda a simulação (ZAKI; TAHAR; BOIS, 2006b).

A implementação desta técnica ocorre com a construção de dois componentes, um estimulador e um monitor. O primeiro tem como objetivo aplicar sinais de tensão e correntes no circuito; e o segundo tem como objetivo verificar os sinais provenientes do circuito, monitorando-os para verificar as propriedades. Estes dois componentes são construídos utilizando máquinas de estados, responsáveis em informar o momento de aplicar o estímulo e de verificar o comportamento do circuito.

O fluxo de trabalho proposto por esta técnica está mostrado na figura 13. Nesta técnica, os

bancos de teste são compostos pelos componentes *Estimulos* e *Monitor*. Enquanto o circuito está sendo estimulado, a sua saída estará sendo monitorada. Se o circuito apresentar alguma violação de comportamento, o componente *Monitor* informará ao componente *Estimulos* para finalizar o estímulo, encerrando a validação. Com isso, será informado no relatório final que a propriedade não foi satisfeita.



Figura 13: Fluxo de trabalho de uma técnica de verificação mista, a Verificação em Tempo de Execução, definida por (DONG et al., 2007a)

# 2.4 Conclusão

Nesta seção foram vistos os principais aspectos envolvidos em um processo de verificação de circuitos integrados. O principal aspecto neste processo é a escolha de uma técnica de verificação. Nos circuitos digitais, já existem várias ferramentas de avançado nível de automatização, as quais implementam algumas destas técnicas vistas. Nos circuitos analógicos, a realidade é outra, as ferramentas de verificação ainda se encontram em fase de desenvolvimento. Apesar disso, foram apresentados exemplos de algumas técnicas de verificação para este tipo de circuitos. Na próxima seção, encontram-se detalhes de como estas técnicas podem ser utilizadas para a construção de uma metodologia e de ambientes de verificação para circuitos analógicos.

# 3 Metodologia de Verificação de Circuitos Analógicos

Na seção anterior detalhou-se o fluxo de execução de um processo de verificação de circuitos, com ênfase nas técnicas de verificação de circuitos analógicos. A presente seção tem como objetivo mostrar uma metodologia desenvolvida para verificar circuitos. Esta metodologia se baseia nos princípios básicos vistos quando estas técnicas foram detalhadas.

# 3.1 Detalhamento da Metodologia Proposta

A metodologia proposta mostra como algumas técnicas podem ser utilizadas na elaboração de um ambiente consistente para verificar circuitos analógicos. Esta metodologia sugere a utilização de duas técnicas de verificação, com a finalidade de certificar um modelo comportamental e, em seguida, utilizá-lo na validação do circuito implementado. Além disso, pretende-se detalhar como será a simulação e a modelagem dos circuitos, incorporando no plano de verificação, os planos de modelagem e de simulação.

Para verificar circuitos analógicos, a metodologia define três etapas a serem executadas pelos engenheiros de verificação. Consistem em: construir um modelo comportamental; implementar um ambiente de verificação e verificar o modelo construído, por meio da técnica de verificação em Tempo de Execução, tornando-o um modelo de ouro; e submeter o circuito implementado e o modelo de ouro a um teste de equivalência, por meio da técnica do Modelo de Ouro, provando o funcionamento do circuito conforme o especificado.



Figura 14: Fluxo de Trabalho utilizado pela metodologia proposta em paralelo com o fluxo de projeto do engenheiro projetista, tendo como pré-requisito, para os dois fluxos, a especificação do projeto. Em destaque, está o instante em que os engenheiros trocam de informações.

A figura 14 ilustra dois fluxos de execução. Na esquerda, um fluxo de projeto básico a ser seguido pelos engenheiros projetistas. Na direita, está mostrado o fluxo de trabalho proposto pela metodologia. Na figura está mostrado também o momento em os dois processos, de verificação e concepção de circuitos, se relacionam.

# 3.2 Construção do Ambiente de Verificação do Modelo de Referência

A técnica do Modelo de Ouro exige a utilização de um modelo de referência. Este modelo será concebido de maneira fiel à especificação do circuito. Consequentemente, o uso de uma técnica para certificar o funcionamento correto deste modelo se faz necessário, sendo a técnica do Monitoramento por Autômatos a adotada. Por meio desta técnica se constrói um ambiente de verificação para certificar modelos comportamentais.

O seu ambiente de verificação é composto por quatro principais partes: Modelo de Referência, Monitoramento, Propriedades e o Relatório. O Modelo de Referência é o implementado utilizando uma linguagem de alto nível, tendo como objetivo construir um modelo comportamental conforme a especificação do circuito. O *Monitoramento* é a implementação da técnica do Monitoramento por Autômatos, a qual engloba as entidades necessárias para aplicar os estímulos e monitorar o modelo. O *Propriedades* são dados de entrada no ambiente, mostrando as propriedades que serão verificadas no modelo. E o *Relatório* é o documento com as informações relevantes à verificação do modelo de referência, o qual mostra se o modelo construído é de ouro. Caso, o modelo não seja validado, o documento mostrará qual propriedade não foi certificada. Na figura 15 estão mostradas as partes que compõem o ambiente de verificação.



Figura 15: Diagrama de blocos da técnica de Monitoramento por Autômatos. O bloco principal é dividido me dois componentes, Estímulos e Monitor.

A seguir serão detalhadas como estas partes são construídas.

## 3.2.1 Modelo de Referência

O *Modelo de Referência* é implementado utilizando linguagens que possam descrever modelos comportamentais de circuitos analógicos e mistos, como: *VHDL-AMS*, *Verilog-AMS* ou *SystemC-AMS*. O principal objetivo da sua implementação é representar fielmente o circuito conforme a sua especificação. Na implementação do ambiente de verificação foi escolhida a linguagem *VHDL-AMS*, devido ao conhecimento prévio desta linguagem adquirida pelo presente autor em trabalhos passados. No entanto, adaptadores podem ser construídos para serem utilizados modelos de outras linguagens.

Este modelo deve ser construído a partir de equações matemáticas que modelam o comportamento do circuito a ser verificado. Com o intuito de serem reutilizados em processos de verificação futuros. Para isso, se faz necessária a utilização de parâmetros ajustáveis do início da verificação, no caso da linguagem VHDL-AMS, estes parâmetros serão do tipo generic.

### 3.2.2 A Implementação da técnica do Monitoramento por Autômatos

O *Monitoramento* é o núcleo certificador do ambiente de verificação do modelo comportamental. O resultado desta certificação depende de uma consistente implementação, visando a validação das propriedades elaboradas. A sua implementação deverá ser a tradução das propriedades em máquinas de estados, definindo, assim, os estímulos e análises.

Este núcleo certificador é composto por dois principais componentes: o *Estímulos* e o *Mo-nitor*. O primeiro é responsável por aplicar estímulos ao modelo comportamental e o segundo é responsável por analisar os sinais dos terminais de saída provindos deste modelo. A figura 16 mostra como ocorre a ligação entres estes componentes e o modelo comportamental. Além disso, pode-se perceber um outro componente chamado *Transactor* (ANDERSON JANICK BER-GERON; NIGHTINGALE, 2006a), responsável por selecionar os sinais elétricos corretos.

Além disso, pode-se perceber na figura um sinal proveniente do componente *Monitor* para o componente *Estímulos*. Este sinal indica ao componente *Estímulos* o momento em que a análise do *Monitor* foi finalizada. Desta forma, não será mais necessária a aplicação de novos estímulos, encerrando a validação da propriedade. Com isso, é possível reduzir o tempo de verificação.



Figura 16: Detalhamento do Bloco Monitoramento, mostrando os seus componentes internos.

Os componentes *Estímulos* e *Monitor* são constituídos por máquinas de estados, conforme estão detalhadas na figura 17. Cada máquina de estados é implementada com três tipos de estado: o de *inicio*, de *estímulo* ou *monitoramento*, e o de *finalização*. A quantidade de estados depende da propriedade a ser certificada e da quantidade de estímulos necessários. No entanto, normalmente são utilizados três estados para o componente *Estímulos* e quatro estados para o componente *Monitor*.



Figura 17: Representação das máquinas de estados que compõem os componentes Estímulos e Monitor.

A seguir serão vistas os detalhes da implementação de cada componente.

#### 3.2.2.1 Componente Estímulos

O componente *Estímulos* é implementado utilizando três tipos de estados: um estado do tipo *inicio*, representando um estado de repouso enquanto é esperada a inicialização do estímulo, acionada por uma porta de entrada; um estado do tipo *estímulo*, responsável por indicar o instante e o tipo de estímulo a ser aplicado no modelo; e um estado do tipo *finalização*, responsável por indicar o fim do monitoramento, encerrando a simulação, podendo ser pela existência de falhas, informado pelo componente *Monitor*. O número de estados do tipo *estímulo* dependerá do número de estímulos necessários para a validação da propriedade em questão.

A figura 18 mostra as portas de entrada e saída do componente *Estimulos*. Ao total, são três portas de entrada: *QtdEstimulo*, um valor real, no qual o engenheiro informa a quantidade de estímulos; o *inicio*, um valor binário, representando o inicio da simulação; e o *sim\_ok*, valor binário proveniente do componente *Monitor*, indicando o momento em que não será mais necessária a aplicação do estímulo no modelo ou a alteração do estimulo. Isso, se for indicada a utilização de mais estímulos pelo sinal *QtdEstimulo*. A única porta de saída, *estimular*, representa o sinal que ativará o *Transactor*.



Figura 18: Representação em bloco do componente Estímulos implementada em uma linguagem comportamental, destacando as portas de entrada e saída.

#### 3.2.2.2 Componente Monitor

O número de estados utilizado na implementação do componente *Monitor* também dependerá da propriedade a ser verificada. No entanto, serão utilizados no mínimo quatro estados para implementá-lo: um estado do tipo *inicio* para aguardar o início da aplicação dos estímulos; um estado do tipo *monitoramento*, necessário para monitorar o comportamento do modelo; e dois estados do tipo *finalização*, sendo um para sinalizar a ocorrência de falhas e um outro para mostrar que a propriedade foi satisfeita.

A figura 19 mostra as portas de entrada e saída do componente *Monitor*. Ao total são três portas: duas de saída e uma de entrada. Esta última é um terminal de entrada do sinal proveniente do modelo comportamental. As portas de saída são: *sim\_ok*, responsável por informar ao componente *Estímulos* quando é necessário alterar o estímulo ou quando não há mais a necessidade de aplicá-los no modelo; e *QtdErros*, um valor real informando a quantidade de erros, se for aplicado mais de um estímulo. Entretanto, a quantidade de portas de entrada dependerá da propriedade em questão, por exemplo, se for verificada a taxa de rejeição do ganho em modo comum de um amplificador diferencial serão necessárias duas portas de entrada, porque para o seu cálculo, necessitam-se dos ganhos diferenciais e de modo comum.



Figura 19: Representação em bloco do componente Monitor implementada em uma linguagem comportamental, destacando as portas de entrada e saída.

## 3.2.3 Definição das Propriedades Verificadas

Para a certificação do funcionamento do modelo comportamental deve-se primeiro identificar quais são as figuras de mérito ou as grandezas físicas a serem verificadas. Quando essas estiverem definidas, as propriedades são descritas conforme o comportamento esperado pelos engenheiros projetistas.

A simulação do modelo deve ser executada de forma a verificar as propriedades descritas. Um exemplo dessas, poderia ser a validação da taxa de rejeição do modo comum (CMRR) de um amplificador diferencial, cuja propriedade seria: "A taxa de rejeição do modo comum do amplificador diferencial deverá apresentar o valor mínimo de 80 dB desde a frequência *de 0 Hz até a frequência que corresponda à metade do produto ganho-banda.*". Desta forma, se o produto ganho-banda do amplificador for 3 *MHz*, serão aplicadas várias formas de onda, senóide, nos terminais de entrada do modelo com frequências variando desde os primeiros *Hz* até a frequência de 1.5 *MHz*. Com isso, o componente *Monitor* analisará os sinais de saída provenientes do modelo para identificar se até esta frequência, o modelo apresenta a taxa de CMRR de, no mínimo, 80 dB.

Não é necessária a utilização de uma linguagem de verificação formal, como a CTL ou ASL. Isso porque o uso destas linguagens exige a implementação de algoritmos específicos para a validação das propriedades do modelo. E a criação de uma nova linguagem não será necessária, sendo requisitos somente as condições de funcionamento dos circuitos descritas conforme o exemplo do paragrafo anterior.

Ao longo das simulações, cada propriedade será identificada através de uma sigla. No relatório serão impressos juntamente com o seu resultado, para sua descrição detalhada será necessária uma consulta no plano de verificação.

### 3.2.4 Descrição do Relatório da Execução do Ambiente de Verificação

A implementação do ambiente de verificação proporciona o mínimo de intervenção humana no processo de verificar circuitos analógicos. Neste caso, os engenheiros de verificação terão como resultado final, um relatório com as informações relevantes ocorridas durante as simulações.

Este relatório será criado quando todas as propriedades estiverem verificadas. Os dados especificados neste documento estão listados abaixo:

- Nome da Entidade Testada
- Nome da Entidade de Estímulos:
- Quantidade de Erros:
- Qual Propriedade apresentou comportamento falho:
- Responsáveis pela Construção dos Banco de Testes:
- Responsáveis pela Construção do Conjunto de Estímulos:
- Tempo de Simulação:

Como pode ser observado, algumas das informações contidas na listagem são fornecidas pelo engenheiro de verificação antes da inicialização do processo de verificação, para que o mesmo certifique as informações processadas ao longo das simulações.

Além disso, deve-se criar bancos de teste parametrizados a partir da utilização de arquivos de entrada, escritos pelos engenheiros de verificação, indicando quais os valores das figuras de mérito e parâmetros a serem verificados. Com isso, podem ser ajustados os estímulos por meio destes parâmetros ao decorrer do processo de verificação.

Estas funcionalidades, a criação do relatório e a integração das diversas ferramentas utilizadas compõem o ambiente de verificação implementado, construído com a ajuda das linguagens *Shell-Scripts* e *Perl*.

# 3.3 Implementação do Ambiente de Verificação do Circuito

O término da validação do modelo comportamental, tornando-o um modelo de ouro, é prérequisito para a inicialização da verificação do circuito. Nesta fase, o engenheiro projetista já deverá ter fornecido o projeto do circuito integrado ao engenheiro de verificação. Para verificálo, deve-se implementar um ambiente de verificação, utilizando a técnica do *Modelo de Ouro*.

As principais partes para a implementação da técnica do Modelo de Ouro estão mostradas na figura 20. Como pode-se perceber, para o seu funcionamento são necessários: o *Modelo de Ouro, Circuito Implementado*, o *Relatório* e o *Conjunto de Estímulos*, além da fundamentação da técnica. O *Conjunto de Estimulos* contém a coleção de formas de ondas utilizadas como estímulos de tensões ou correntes, selecionadas para serem aplicadas nos modelos. O *Modelo de Ouro* é a representação simulável fiel à especificação do projeto. O *Circuito Implementado* é a entidade a ser verificada. E, por último, o *Relatório*, sendo o artefato resultante da execução da validação do circuito. O ambiente de verificação integra todos eles, além de analisar e diagnosticar os sinais provenientes dos modelos.



Figura 20: Identificação dos componentes utilizados pela técnica do Modelo de Ouro, apresentando como dados de entrada para a sua implementação: o conjunto de estímulos, a entidade a ser verificada e o modelo de ouro.

A verificação do projeto construído pelos projetistas neste ambiente ocorre através do uso de bancos de teste, sendo os seus detalhes mostrados na figura 21. Nessa pode-se visualizar quatro principais blocos: *Estímulo*, *Circuito a ser verificado*, *Modelo de Ouro* e *Validador*. O primeiro é responsável por aplicar os estímulos nos modelos envolvidos. O segundo é o próprio projeto a ser verificado, o terceiro é o modelo comportamental, utilizado como entidade de referência. O último é o bloco certificador do circuito implementado, validando o seu funcionamento conforme o modelo de referência.



Figura 21: Banco de teste utilizado na técnica de modelo de ouro incorporando os conceitos de verificação a nível de transação. Este banco apresenta como dado de entrada somente a figura de mérito que se deseja verificar.

Todos estes blocos devem ser construídos utilizando a linguagem *Spice*, exceto o *Modelo de Ouro*, implementado com a ajuda de uma linguagem com um alto nível de abstração; e o *Validador*, construído através do uso de ferramentas que resgatem os dados da simulação, os analisem e reportem os seus resultados em um relatório final. Além disso, será necessária a utilização de simuladores que integrem modelos comportamentais, descritos na linguagem VHDL-AMS, com circuitos descritos na linguagem *Spice*.

O diferencial desta técnica, aqui proposta, é a utilização de bancos de testes a nível de transação. Com isso, não será necessária a aplicação de estímulos pelo engenheiro de verificação, necessitando apenas de informarem as figuras de mérito a serem validadas, fazendo com que estes bancos se encarreguem de escolher os estímulos adequados. Em outras palavras, a principal diferença entre os bancos de teste vistos na seção 2 e o mostrado na figura 21 é a presença do seu encapsulamento, para possibilitar a sua parametrização em tempo de simulação.

A principal vantagem disso é a possibilidade de automatizar o processo de verificação. Com esta automatização, os projetistas e os engenheiro de verificação, podem abdicar das responsabilidades de aplicar estímulos e analisar sinais, fazendo com que os engenheiros tenham a responsabilidade de implementar ambientes que possibilitem isso.

Além disso, o circuito, por meio deste ambiente, poderá ser verificado de duas formas e em duas ocasiões. No caso de diferentes formas, poderão ocorrer em diferentes níveis de abstração: a nível de transistor ou a nível de leiaute incluindo os parasitas. No caso de diferentes ocasiões, poderão ocorrer: na finalização do leiaute ou durante todo o processo de implementação do circuito, utilizando os dados das simulações para otimizar o projeto.

Estes bancos de teste são mais simples em relação àqueles apresentados pela técnica anterior. No entanto, a escolha de um simulador que integre o modelo comportamental e o circuito implementado é essencial para a execução dos bancos da técnica vista nesta subseção, em alguns casos esta escolha poderá ser custosa.

Na implementação deste banco, a parte primordial é a construção de um validador confiável. Na construção do *Validador*, o engenheiro de verificação escolhe uma análise matemática adequada para demonstrar a semelhança entre os dois modelos, a mais popular dessa são os comparadores lógicos.

O uso de simples comparadores lógicos não comprometerá a confiabilidade do projeto, já que o modelo de referência foi verificado antes da verificação do circuito. Além disso, o uso de analisadores formais ainda não se apresentam de forma confiável para a validação de circuitos analógicos com transistores em nível físico. Isso porque o seu uso requisita um alto poder de

processamento devido a complexidades dos modelos na verificação pós-leiaute.

No caso dos estímulos, devem ser selecionados sinais de tensão e corrente especificamente para verificar cada item das metas de cobertura. Para cada meta deve ser utilizado um estímulo ou um conjunto do mesmo, para verificar a propriedade no circuito sob distintas condições de operação.

Na análise dos resultados, deve ser utilizada a análise de cobertura funcional. No caso dos circuitos analógicos, cada figura de mérito ou grandeza física na especificação representa um item na lista de cobertura. Se toda a especificação do circuito for verificada, pode-se dizer que o processo de verificação cumpriu a meta em 100%.

A construção de um ambiente de verificação utilizando estes aspectos, integrando banco de teste a nível de transação e uma análise de cobertura funcional, poderá apresentar vantagens durante o processo de verificação e, alguns momentos, em futuras verificações, ocasionando uma redução no tempo de certificar circuitos.

# 3.4 Conclusão

Nesta seção foi detalhada a metodologia proposta pelo autor para ser utilizada na verificação de circuitos analógicos. Com o uso desta metodologia, os engenheiros poderão construir ambientes de verificação para certificar modelos de referência e projetos de circuitos integrados. Além disso, os integrantes do projeto poderão obter modelos de alto nível de abstração, como os modelos comportamentais e a nível de leiaute, sendo úteis para validação de sistemas e de circuitos respectivamente.

A execução de um estudo de caso, para comprovar as vantagens da metodologia de verificação, foi proposta, com o objetivo de verificar o funcionamento de um amplificador operacional. Neste estudo de caso foi utilizado um modelo comportamental escrito na linguagem *VHDL-AMS*, um circuito descrito na linguagem *Spice*, dois ambiente de verificação implementados utilizando as linguagens *Shell-Scripts* e *Perl* e um conjunto de ferramentas do pacote *ADMS* da *Mentor Graphics*. Na próxima seção estão detalhados os resultados obtidos.

# 4 Estudo de Caso A - Validação de um Amplificador Operacional

Na seção anterior foi visto como os principais conceitos de verificação de circuitos analógicos podem ser utilizados na construção de um ambiente de verificação. A metodologia proposta define uma técnica para verificar um modelo comportamental.

Um estudo de caso é detalhado nesta seção, com a finalidade de mostrar os benefícios da metodologia. Primeiramente, é mostrado o plano de verificação utilizado para verificar o circuito analógico proposto. Em seguida, é mostrada a implementação deste circuito. E, por último, os detalhes do processo de verificação, avaliando o funcionamento do modelo comportamental criado e do circuito implementado, são explanados.

# 4.1 Plano de Verificação

O plano de verificação é um documento, no qual são definidos os objetivos da verificação de um projeto. O início do processo de verificação ocorre por meio da elaboração deste documento. Nesta sub-seção será detalhado o plano de verificação utilizado na execução deste estudo de caso.

### 4.1.1 Introdução

A verificação de uma entidade analógica envolve vários aspectos desde a definição das técnicas até a composição do quadro de profissionais responsáveis. Estes aspectos podem ser divididos em dois tipos: os tecnológicos e os cronológicos. A sub-seção dos aspectos tecnológicos define as tecnologias, técnicas, propriedades, ferramentas e metas a serem cumpridas ao longo de todo o processo de verificação. A sub-seção dos aspectos cronológicos detalha o cronograma, além de definir a equipe e os perfis dos seus membros.

### 4.1.2 Aspectos Tecnológicos

Os aspectos tecnológicos são definidos com base em dois documentos: a especificação e o de riscos de projeto. A seguir, serão abordados os aspectos técnicos do processo de verificação.

### 4.1.2.1 Objetivos

O objetivo deste processo é certificar o funcionamento de um amplificador operacional, tendo como base a especificação mostrada no quadro 1. A verificação deve se basear em uma abordagem funcional, utilizando uma técnica baseada em simulação para verificar o funcionamento do circuito implementado, esta validação ocorrerá em duas fases: primeiro a verificação do circuito a nível de transistor e, posteriormente, a nível de leiaute.

| Figura de Mérito | Valor | Unidade |
|------------------|-------|---------|
| Ganho            | 80    | dB      |
| GBW              | 3     | MHz     |
| CMRR             | 80    | dB      |
| PSRR             | 80    | dB      |
| Slew Rate        | 0.3   | V/µs    |
| Margem de Fase   | 60    | 0       |

Quadro 1: Especificação do Amplificador Operacional

### 4.1.2.2 Especificação

O processo de verificação certifica os seis parâmetros destacados no quadro 1, escolhidos a partir da especificação do circuito. O amplificador operacional deverá funcionar com um ganho diferencial de 80 *dB*, com o produto ganho-banda (GBW) de 3 *MHz* e margem de fase de 60°. A taxa de rejeição do ganho de modo-comum (CMRR) é uma figura de mérito responsável em quantificar a tendência do circuito de rejeitar sinais semelhantes aplicados nos terminais de entrada do amplificador diferencial, o valor especificado para esta taxa foi de 80 *dB*. A taxa de rejeição da variação da tensão de alimentação do circuito no terminal de saída (PSRR) é uma figura de mérito que mede a interferência no sinal do terminal de saída do amplificador operacional em função da variação da tensão de alimentação, sendo o seu valor especificado em 80 *dB*. Por último, a máxima taxa de variação do sinal de saída em função do tempo, a taxa *Slew Rate*, foi especificada no valor de  $0.3V/\mu s$ .

A verificação do circuito engloba a validação de 5 propriedades, as quais estão descritas a seguir.

- Propriedade pGBW: "O produto ganho-banda do amplificador não poderá ser menor que 3 MHz."
- Propriedade sCMRR: "A taxa CMRR deverá apresentar um valor igual ou maior que 80 dB até a frequência de 3 dB."
- Propriedade tPSRR: "A taxa PSRR não poderá ser menor que 80 dB até a frequência de 3 dB."
- Propriedade qMF: "A margem de fase do amplificador não poderá exceder 60º na frequência GBW."
- Propriedade qqSR: "A taxa *Slew Rate* do amplificador não poderá ser menor que 0,3 V/μs."

### 4.1.2.4 Técnicas

A abordagem adotada foi a verificação funcional, restringindo ao uso de técnicas de verificação baseadas em simulação. Com isso, não serão necessárias a utilização de técnicas formais. No caso, foi escolhida a técnica do Modelo de Ouro para verificar o circuito, e uma técnica de verificação em Tempo de Execução, a técnica do Monitoramento por Autômatos, para certificar o modelo comportamental, conforme definida na metodologia proposta.

### 4.1.2.5 Ferramentas

As ferramentas apropriadas para construir os modelos envolvidos, compilá-los e simulálos são fornecidos para o sistema operacional *Linux*, no pacote ADMS provido pela empresa *Mentor Graphics*.

Este pacote, o *ADvanced Mixed Signal* (ADMS), é utilizado para a construção, simulação e validação do modelo comportamental além de simular arquivos *spices*. O *ADMS* disponibiliza as ferramentas: *vasim* para a simulação; *vacom* para a compilação dos arquivos fontes de VHDL-AMS, Verilog-AMS; e *ezwave* para a visualização dos resultados das simulações. Para a

simulação dos circuitos descritos na linguagem *Spice* será utilizada a ferramenta *eldo*. Uma ferramenta de matemática simbólica, *octave*, proveniente do pacote básico do sistema operacional adotado será utilizada para a construção do bloco *Validador* dos bancos de teste especificados pela técnica do Modelo de Ouro. Além disso, serão necessários os interpretadores de comando para as linguagens *Bash* e *Perl*, utilizadas para construção dos ambientes de verificação

#### 4.1.2.6 Ambiente

O ambiente de verificação utilizado para executar as técnicas deve ser implementado no sistema operacional *Linux*, incluindo o desenvolvimento de *scripts* por meio das linguagens *Shell-Scripts* e *Perl*. Na validação do amplificador operacional, será necessária a construção de dois ambientes de verificação. O primeiro a ser construído auxilia na validação do modelo comportamental com base nos conceitos da técnica do Monitoramento por Autômatos, e o segundo auxilia na validação do circuito implementado.

O núcleo dos ambientes de verificação, nestas técnicas, serão os bancos de teste. Na execução do núcleo definido pela técnica do Modelo de Ouro, os dados de entrada são: o modelo de ouro, o conjunto de estímulos e o circuito implementado. Este último é fornecido pelos engenheiros projetistas, e os outros são de responsabilidade da equipe de verificação. No caso da técnica Monitoramento por Autômatos, os dados de entrada do ambiente de verificação são: as propriedades a serem verificadas e o modelo comportamental, sendo de responsabilidade do engenheiro de verificação a construção do modelo e dos bancos de teste.

### 4.1.3 Aspectos Cronológicos

Nesta sub-seção do plano de verificação detalham-se os aspectos cronológicos, mais especificamente sobre o cronograma e a equipe.

O principal objetivo é mostrar como funcionam os ambientes de verificação construídos a partir da metodologia proposta. Com isso, serão verificadas somente cinco propriedades no amplificador operacional, desta forma, a equipe será composta somente por um membro, o presente autor.

Este engenheiro de verificação terá como meta cobrir o cronograma mostrado no quadro 2. Nele pode-se visualizar todas as etapas a serem executadas, divididas pelo número de semanas.

Nas primeiras duas semanas realiza-se um estudo na especificação do projeto, definindo o modelo comportamental a ser construído, mostrando as principais equações a serem implementadas, linguagem utilizada, as portas de saída e entrada e os parâmetros tecnológicos. Na segunda etapa, o modelo comportamental será implementado ao longo de três semanas, utilizando o modelo proposto na etapa anterior. Na quarta semana será iniciada a construção do ambiente de verificação com duração de quatro semanas, inicializando, em seguida, a validação do modelo implementado. Esta etapa terá duração de duas semanas e poderá retornar a etapa anterior para otimizar o modelo. Com a finalização da validação deste modelo, nas duas últimas semanas será executada a etapa de verificação do circuito implementado, certificando-o em dois níveis de abstração: esquemático e leiaute.

Quadro 2: Cronograma a ser seguido pela equipe de verificação.

| Atividade/Semana               | Sem.1 | Sem.2 | Sem.3 | Sem.4 | Sem.5 | Sem.6 | Sem.7 | Sem.8 |
|--------------------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| Estudo da Especificação        | X     | X     |       |       |       |       |       |       |
| Construção do Modelo Comp.     |       | X     | X     | X     |       |       |       |       |
| Validação do Modelo Comp.      |       |       |       | X     | Х     |       |       |       |
| Const. Ambiente de Verificação |       |       |       | X     | Х     | X     | Х     |       |
| Validação do Ckt Impl./Leiaute |       |       |       |       |       |       | Х     | X     |

# 4.1.4 Conclusão

O Plano de Verificação é o ponto de partida na execução do processo de verificação. Com estes aspectos bem definidos e de forma clara, os engenheiros de verificação obterão uma maior capacidade de executar este processo, alcançando objetivos mais satisfatórios.

# 4.2 Projeto

Nesta sub-seção serão vistos os principais pontos a serem abordados no projeto do amplificador operacional, o qual foi implementado por uma equipe de projetistas. As principais características deste circuito são: elevado ganho diferencial, alta impedância de entrada e baixa impedância de saída.

A especificação do amplificador operacional está mostrada no quadro 1. A topologia a ser projetada está detalhada na figura 22. A análise do circuito pode ser encontrada em (GRAY PAUL HURST; MEYER, 2001), de onde podem-ser extraídas as equações (2)-(4).





A taxa *Slew Rate* (SR) deste amplificador é limitada pelo capacitor de compensação ( $C_c$ ) e diretamente proporcional à corrente ( $I_s$ ) do transistor M5, conforme indica a equação (2). Esta corrente é responsável por polarizar o par diferencial do primeiro estágio.

$$SR = \frac{I_5}{C_c} \tag{2}$$

Por meio da equação 3 é possível calcular o ganho diferencial do amplificador, destacado por estágio. Como pode-se perceber, o ganho diferencial do amplificador é proporcional às transcondutâncias dos transistores M1 e M7 e limitado pelas correntes polarizadoras dos tran-

sistores do par diferencial e do transistor M6, e pelos parâmetros tecnológicos dos transistores M1, M3, M6 e M7, implícitas no cálculo de suas resistências de saída. Este parâmetro varia dependendo do nível de inversão, comprimento do canal e das tensões aplicadas entre os terminais Dreno e Fonte.

$$A_{V} = \underbrace{-g_{m2}(R_{ds_{4a}} / / g_{m_{4b}}R_{ds_{4b}}R_{ds_{2}})}_{1^{\circ} \text{ Estágio}} \cdot \underbrace{-g_{m7}(R_{ds_{7}} + R_{ds_{6}})}_{2^{\circ} \text{ Estágio}}$$
(3)

 $R_{ds_{3a,3b,1,7,6}}$  são as resistências de saída dos transistores *M3a*, *M3b*, *M1*, *M7* e *M6*;  $g_{m1,m7,m3b,m6}$  são as transcondutâncias dos transistores *M1,M7*, *M3b* e *M6*.

O produto ganho-banda do amplificador é calculado a partir da equação (4). Como pode-se perceber, esta grandeza é diretamente proporcional à transcondutância dos transistores do par diferencial e inversamente proporcional à capacitância de compensação.

$$GBW = \frac{g_{m1}}{C_c} \tag{4}$$

 $g_{m1}$  é a transcondutância do transistor M1 e  $C_c$  é o capacitor de compensação.

Um detalhe interessante nesta arquitetura é a utilização de um resistor de compensação para anular o pólo não-dominante de maior valor. Isso é necessário para que seja obtido um alto produto ganho-banda com baixa transcondutância  $g_{m7}$ , sem comprometer o ganho diferencial. Com tal finalidade, é utilizado um transistor com a função de resistor para o cancelamento deste pólo. O dimensionamento de  $M_{r7}$  se dá a partir da equação 5.

$$\frac{W_r}{L_r} = \frac{C_c}{C_L + C_c} \cdot \sqrt{\frac{S_{r2}S_7I_7}{I_{r2}}}$$
(5)

 $S_{r2,7}$  são as relações das larguras e comprimentos dos transistores Mr2 e M7,  $C_c$  é o capacitor de compensação,  $C_L$  é a carga capacitiva na saída e  $I_{r2,7}$  são as correntes de dreno dos transistores Mr2 e M7 (ALLEN DOUGLAS R. HOLBERG, 2008),  $\frac{W_r}{L_r}$  é a relação do comprimento e largura de canal do transistor  $M_r$ .

$$\frac{W_{r1}}{L_{r1}} = \frac{I_{r2}}{I_7} \cdot \frac{W_7}{L_7}$$
(6)

O circuito foi projetado utilizando a tecnologia de  $0.5 \,\mu$ m fornecida pela AMIS. A dimensão dos transistores e do capacitor de compensação estão mostrados no quadro 3. Na figura 23 está mostrado o leiaute do amplificador projetado. Na construção do leiaute foram utilizadas técnicas de casamento em cada par de transistores, assim como, colocados componentes "burros" <sup>1</sup> nas extremidades do circuito para garantir que no processo de fabricação os transistores sejam igualmente afetados pelas etapas químicas. O circuito ocupou uma área de  $0.043 mm^2$ 

| Componente | Valor (W/L) | Unidade |
|------------|-------------|---------|
| Componente |             | Omuaue  |
| M1/M2      | 50µ / 4µ    | m / m   |
| M3a/M4a    | 50µ / 4µ    | m / m   |
| M3b/M4b    | 75µ / 2µ    | m / m   |
| M5/M6      | 250µ / 150µ | m / m   |
| M7         | 84µ / 1.2µ  | m / m   |
| Mr1/Mr2    | 60µ / 2µ    | m / m   |
| Mr3        | 250µ / 150µ | m / m   |
| Mr         | 50µ / 2µ    | m / m   |
| Cc         | 3 p         | F       |

Quadro 3: Dimensionamento dos componentes, utilizando um capacitor de 0.8 pF na saída do amplificador



Figura 23: Leiaute do Amplificador Operacional, conforme a topologia apresentada na figura 22.

<sup>&</sup>lt;sup>1</sup>componentes que não exercem nenhuma funcionalidade no circuito

# 4.3 Processo de Verificação

O objetivo deste processo de verificação é certificar a implementação do amplificador operacional. Ao término da execução deste processo, serão apresentados dois artefatos para comprovar os resultados do emprego da metodologia proposta. Os dois artefatos são documentos, comprovando a construção e validação do modelo de ouro, além da certificação do circuito implementado. Ao londo desta sub-seção, serão detalhados os resultados contidos nestes artefatos.

## 4.3.1 Modelo de Referência

O modelo de referência foi construído utilizando a linguagem VHDL-AMS, a escolha desta linguagem foi motivada pela familiarização da equipe de verificação por meio da codificação de outros modelos comportamentais anteriores a este trabalho.

Este modelo foi concebido com o objetivo de ser re-utilizável em verificações de outros amplificadores operacionais. Desta forma, os parâmetros da especificação foram definidos como variáveis do tipo *generic*. Com isso, é possível alterar seus valores no momento da instanciação do modelo comportamental no arquivo principal de simulação.

A seguir, estão detalhadas as etapas de concepção e validação do modelo de referência.

#### 4.3.1.1 Amplificador Operacional

O comportamento ideal de um amplificador operacional pode ser representado pelo circuito mostrado na figura 1. Este amplificador, idealmente, apresenta como tensão de saída as respostas computadas por:



Figura 1: Modelo Comportamental Ideal de um amplificador operacional.  $V_{id}$  representa a entrada diferencial entre os dois terminais de entrada do amplificador,  $av_{id}$  representa o sinal do terminal de saída do amplificador,  $a \notin o$  ganho diferencial do amplificador e  $V_o \notin o$  terminal de saída.



Figura 2: Modelo Comportamental, incorporando algumas imperfeições para garantir o seu funcionamento conforme a prática.

$$Vout_{eanho} = G \cdot V_{id} \tag{7}$$

em que G é o ganho diferencial do amplificador, Vid é a entrada diferencial.

Entretanto, para obter um comportamento mais próximo do circuito implementado, foram adicionados os efeitos de segunda ordem mais relevantes de um amplificador operacional, reformulando a equação 7 para acrescentar os parâmetros mostrados na especificação do circuito, quadro 1. Na figura 2, está mostrado o esquemático de um amplificador operacional com algumas imperfeições mais comuns. Com este modelo é possível obter resultados mais próximos dos circuitos na prática.

Com o intuito de adicionar o efeito definido pelo decaimento do ganho em função do aumento da frequência, quantificado pelo parâmetro produto ganho-banda, altera-se (7), obtendo (8).

$$Vout_{gbw} = \frac{Vout_{ganho}}{\left(\frac{s \cdot G}{2\pi \cdot GBW} + 1\right)}$$
(8)

GBW é o produto ganho-banda, G é o ganho em malha aberta do amplificador.

O próximo parâmetro adicionado foi o *CMRR*, a taxa de rejeição do sinal de saída no modo comum. Idealmente, o amplificador operacional apresenta somente ganho diferencial, sendo o ganho em modo comum zero, tornando a taxa *CMRR* infinita. Para adicionar este efeito no sinal de saída do amplificador é acrescentado um termo em 8, cujo valor é a soma das tensões aplicadas nos dois terminais de entrada do amplificador dividido pela taxa *CMRR* e multiplicado pelo ganho diferencial, conforme (9).

$$Vout_{gbw,cmrr} = Vout_{gbw} + \frac{G}{10^{\frac{CMRR}{20.0}}} \cdot V_{ic}$$
(9)

 $Vout_{gbw}$  é o ganho diferencial definido por (8), CMRR é o valor da taxa CMRR em dB e  $V_{ic}$  é a soma das tensões aplicadas nos terminais de entrada.

O outro parâmetro acrescentado foi a taxa de variação do sinal de saída do amplificador em função da variação da tensão de alimentação, o *PSRR*. Para isso, altera-se a equação de saída do modelo para a formar apresentada pela equação 10, a qual foi encontrada por meio da soma da equação 9 com a derivada da tensão de alimentação em relação ao tempo.

$$Vout_{gbw,cmrr,psrr} = Vout_{gbw,cmrr} + G \cdot \frac{d(V_{dd})}{dt} \cdot \frac{1}{10^{\frac{PSRR}{20.0}}}$$
(10)

 $Vout_{gbw,cmrr}$  é a tensão de saída com os efeitos *CMRR* e *GBW*,  $V_{dd}$  é a tensão de alimentação, *PSRR* é o valor da taxa *PSRR* em *dB*.

O efeito do *Slew Rate* foi adicionado com a ajuda de uma função da linguagem VHDL-AMS, a *Q'slew(tsub,tdesc)*. Os argumentos *tsub* e *tdesc* são os atrasos da tensão na subida e na descida, respectivamente, em função do tempo.

No modelo comportamental do amplificador operacional foram adicionados vários outros efeitos, como variação de temperatura, *Offset DC*, impedância de saída e entrada, corrente máxima e mínima de polarização, entre outros. Com o código 2, pode-se visualizar na linguagem VHDL-AMS a entidade implementada do modelo comportamental.

Código 2: Entidade referente a implementação do modelo comportamental do amplificador operacional, escrita em VHDL-AMS

| entity | ampop is  |                         |   |
|--------|-----------|-------------------------|---|
|        | generic ( |                         |   |
|        | G         | : real := 10000.0;      | Open-Loop Gain [V/V]                      |
| 5      | CMRR      | : $real := 80.0;$       | Common Mode Rejection Ratio [dB]          |
|        | SRp       | : real := 0.3;          | Slew Rate (Rise) [V/uS]                   |
|        | SRn       | : real := 0.3;          | Slew Rate (Fall) [V/uS]                   |
|        | Vomin     | : real := 0.9;          | Vout Min [% of Vo]                        |
|        | Vomax     | : real := 0.9;          | Vout Max [% of Vo]                        |
| 10     | GBW       | : real := 1.5;          | Gain-Bandwidth [MHz]                      |
|        | Voffset   | : $real := 0.0e-6;$     | — Input Voltage Offset [V]                |
|        | Ro        | : real := 80.0;         | — Output Resistence [Ohms]                |
|        | PSRRp     | : real := 80.0;         | Power Supply Rejection Ratio + (VDD) [dB] |
|        | PSRRn     | : real := 80.0;         | Power Supply Rejection Ratio - (VSS) [dB] |
| 15     | Icc       | : real := $3.0e-3;$     | Power Supply Current [A]                  |
|        | ib        | : real := $500.0e - 9;$ | — Input Bias Current [A]                  |
|        | ios       | : $real := 200.0e-9;$   | — Input Offset Current [A]                |
|        | Rid       | : real := 1.0e6;        | — Input Differential Resistence [Ohms]    |

|     |                | Ricm         | : real := 100.0e3;  | — Input Common Mode Resistence [Ohms]      |
|-----|----------------|--------------|---------------------|--|
| 20  |                | Vdrift       | : real := 0.0;      | — Input Voltage Offset Drift [uV/°C]       |
|     |                | Idrift       | : real := 0.5;      | — Input Current Offset Drift [nA/°C]       |
|     |                | temp         | : real := 25.0;     | — Default Temperature [°C]                 |
|     |                | tmin         | : real := $-55.0$ ; | — Minimum Operating Temperature [°C]       |
|     |                | tmax         | : real := 125.0;    | — Maximum Operating Temperature [°C]       |
| 25  |                | iosc         | : real := 60.0e-3;  | — Output Short Circuit Current [A]         |
|     |                | Vimax        | : real := 0.8;      | — Maximum Input Voltage Allowed [% of VDD] |
|     |                | Vimin        | : real := 0.8;      | — Minimum Input Voltage Allowed [% of VSS] |
|     |                | temperature  | : real := 277.0     | — Temperature                              |
| 30  | ):             |              |                     |  |
|     | port(          |              |                     |  |
|     | <b>F</b> ····( | terminal vdd | : electrical;       |  |
|     |                | terminal vss | : electrical:       |  |
|     |                | terminal V1  | : electrical:       |  |
| 35  |                | terminal V2  | : electrical:       |  |
|     |                | terminal Vo  | : electrical        |  |
|     | ):             |              |                     |  |
| on  | d antity amp   | <b></b>      |                     |  |
| CII | u chilly ampo  | op,          |                     |  |

### 4.3.1.2 Verificação das propriedades no Modelo Comportamental

A validação das propriedades do modelo de referência foi executada com base na técnica do Monitoramento por Autômatos. Nessa, são analisadas as conformidades das tensões ou correntes provenientes do modelo comportamental com as propriedades definidas no plano de verificação.

A primeira propriedade verificada, *pGBW*, pôde ser validada por meio do cálculo do produto ganho-banda do modelo. As máquinas de estados utilizadas para aplicar o estímulo e efetuar este cálculo, de responsabilidades dos componentes *Estímulos* e *Monitor*, respectivamente, estão mostradas na figura 24.

Na figura, pode-se observar que o gerador de estímulos, componente *Estímulo*, é implementado com três estados: um do tipo *inicio* (*SO*), um do tipo *estimulo* (*S1*) e um do tipo *finalização* (*S2*). E, o outro componente, *Monitor*, é construído com cinco estados: um do tipo *inicio* (*M0*), dois do tipo de *monitoramento* (*M1,M2*) e dois do tipo *finalização*, *M3* para indicar o comportamento falho e *M4* para indicar o comportamento correto.

Inicialmente, as duas máquinas se encontram nos estados *S0* e *M0*. Em um determinado instante, o sinal *comeco* é acionado, fazendo com que o componente *Estimulo* aplique um estímulo no modelo de referência e o componente *Monitor* comece a monitorar o sinal de saída do mesmo, estados *S1* e *M1*. Periodicamente, o componente *Monitor* calcula o ganho diferencial do amplificador, estado *M2*, e compara este valor com 0 *dB* até encontrar a frequência em que o ganho diferencial seja igual a 1  $\frac{V}{V}$ , representando a largura de banda do modelo, e, com isso, o componente *Estimulo* finaliza a aplicação do sinal de estímulo, estado *S2*. Se a frequência encontrada for maior ou igual ao especificado, o estado seguinte, do componente *Monitor*, será o M4, satisfeito, mostrando que a propriedade foi satisfeita; caso contrário, o estado seguinte será o M3, estado de erro.



Figura 24: Máquinas de estados implementados nos blocos estimulo e monitor, mostrando como e quando devem ocorrer a aplicação dos estímulos e do seu monitoramento para a certificação da propriedade *pGBW*.

Esta figura de mérito necessita de um monitoramento na frequência, e para que as máquinas de estados funcionem, as simulações precisam ocorrer no domínio do tempo. Um artifício utilizado para contornar isso foi a execução de várias simulações no tempo com senóides de várias frequências até alcançar a frequência do ganho unitário, fazendo a comparação dessa com a esperada.

Para implementar este artifício foi necessária a utilização de *scripts* de automatização, com a ajuda da linguagem *Shell-Scripts* e VHDL-AMS, para incrementar o valor da frequência da senóide aplicado pela fonte de entrada em cada ciclo de simulação. Este passo é repetido até o instante em que é encontrada uma violação ou até o término da simulação. Uma vantagem do uso deste artifício, é a possibilidade de verificar o produto ganho-banda introduzindo os efeitos de segunda ordem, os quais podem ser mascarados na simulação no domínio da frequência.

A segunda propriedade, qMF, foi verificada calculando a margem de fase do modelo comportamental. As máquinas de estados utilizadas para monitorar e estimular o modelo estão mostradas na figura 25.

A verificação desta propriedade segue o mesmo princípio da validação da propriedade anterior, apresentando os mesmos estados no componente *Estímulo* e *Monitor*. As únicas diferenças estão nos valores utilizados para a comparação e cálculo da margem de fase (BAKER, 2007).

Como pode ser visualizado na figura: no instante inicial, as máquinas de estados dos com-
ponentes *Estimulo* e *Monitor* estão nos estados S0 e M0, respectivamente, quando o sinal *co-meco* é ativado, as máquinas de estados seguirão para os estados S1 e M1, aplicando, assim, o sinal de estímulo no modelo e acionando o monitoramento do sinal de saída do modelo. Ao término do primeiro período do estímulo aplicado, a máquina do componente *Monitor* seguirá para o estado M2 e a do componente *Estimulo* para o estado S2. Nos próximos estímulos não serão utilizadas as frequências utilizadas até o momento. No estado M2, o componente *Monitor* calcula a margem de fase, se o seu valor for maior ou igual ao especificado, o estado seguinte será o M3, se for menor, o estado seguinte será o M4.



Figura 25: Máquinas de estados implementados nos blocos estimulo e monitor, mostrando como e quando devem ocorrer a aplicação dos estímulos e do seu monitoramento para a certificação da propriedade qMF.

A terceira propriedade verificada foi a *sCMRR*, sendo possível a sua validação por meio do cálculo da taxa de rejeição do modo comum decorrente dos resultados do comportamento proveniente do modelo. As máquinas de estados utilizadas para estimular e monitorar estão mostradas na figura 26. Estas máquinas diferem das demais somente nas condições do valor desta taxa e no momento em que é realizada a comparação dos valores.

Neste caso, serão instanciados dois modelos comportamentais. O componente *Estímulo* aplica os mesmos valores de tensão nos dois terminais de entrada de um dos modelos e o componente *Monitor* calcula o seu ganho no modo comum. No outro modelo, o componente *Esti-mulo* aplica estímulos distintos nos dois terminais de entrada para calcular o ganho diferencial, para que, posteriormente a taxa *CMRR* do modelo seja calculado.

A figura 26 mostra as máquinas de estados construídas para verificar esta propriedade. Enquanto, o sinal *comeco* não for acionado as máquinas de estados dos componentes *Estimulo* e *Monitor* estarão nos estados S0 e M0, respectivamente, quando este sinal for ativado, as máquinas irão para os estado S1 e M1. Quando o *CMRR* for calculado, o estado do componente *Estimulo* será *S2*, ou seja, não será mais aplicado o estímulo, e o componente *Monitor* estará no estado *M2*, ou seja, o valor do *CMRR* encontrado será comparado com o resultado especificado. Caso seja satisfeito, o estado seguinte do componente *Monitor* será *M3*, caso contrário, será o *M4*.



Figura 26: Máquinas de estados implementados nos blocos estimulo e monitor, mostrando como e quando devem ocorrer a aplicação dos estímulos e do seu monitoramento para a certificação da propriedade *sCMRR*.

A quarta propriedade verificada, *tPSRR*, será validada por meio do cálculo da taxa de rejeição da variação do sinal de saída do amplificador em função da variação da sua tensão de alimentação. A figura 27 mostra as máquinas de estados para a validação desta propriedade.

Neste caso, o estímulo gerado será aplicado no terminal  $V_{dd}$  do modelo. O funcionamento destas máquinas de estado ocorre da seguinte forma: no instante em que o sinal *comeco* for acionado, as máquinas de estados dos componentes *Estimulo* e *Monitor* seguirão dos estados S0 e M0 para os estados S1 e M1, respectivamente; nestes estados serão aplicados os estímulos e será inicializado o monitoramento; no momento em que a taxa PSRR for calculado, o estímulo se encerrará (estado S2 do componente *Estimulo*), e o valor da taxa *PSRR* é comparado com o valor especificado (o estado M2 do componente *Monitor*); com isso, o estado seguinte será M3, caso satisfeita a propriedade ou M4, caso a propriedade não esteja satisfeita.



Figura 27: Máquinas de estados implementados nos blocos estimulo e monitor, mostrando como e quando devem ocorrer a aplicação dos estímulos e do seu monitoramento para a certificação da propriedade *tPSRR*.

A quinta, e última, propriedade verificada foi a *qqSR*. Neste caso, não foram necessárias, por ser uma figura de mérito avaliada no domínio do tempo, executar várias simulações com senóides de distintas frequências, ou seja, será aplicada somente uma forma de onda, uma função degrau. Para este estímulo, o componente *Monitor* calcula a derivada do sinal do terminal de saída do modelo em um determinado intervalo de tempo e compara-o com o valor da taxa *Slew Rate* especificado.

As máquinas de estados, dos componentes *Estímulo* e *Monitor*, utilizadas na verificação da propriedade *qqSR* encontram-se na figura 28. Assim como, nas outras certificações, o estado inicial das duas máquinas dos componentes *Estimulo* e *Monitor* são representados pelos estados

*S0* e *M0*, respectivamente. No instante em que o sinal *comeco* é acionado, estas máquinas seguirão para os estados *S1* e *M1*, representando a inicialização do estímulo e do monitoramento. Neste momento, aplica-se uma função degrau no terminal de entrada do modelo, é calculada a derivada da curva do seu sinal de saída, no estado *M2* pelo componente *Monitor*, se este valor for inferior ao valor especificado, a máquina seguirá para o estado *M4*, caso a propriedade seja satisfeita, o estado seguinte será *M3*. No instante do cálculo da taxa *Slew Rate*, a máquina de estados do componente *Estimulo* seguirá para o estado *S2*, encerrando a aplicação do sinal de estímulo no modelo.



Figura 28: Máquinas de estados implementados nos blocos estimulo e monitor, mostrando como e quando devem ocorrer a aplicação dos estímulos e do seu monitoramento para a certificação da propriedade *qqSR*.

#### 4.3.1.3 Resultados

Neste momento serão detalhados os resultados obtidos com a certificação do modelo comportamental, tornando-o um modelo de ouro. O gráfico 2 mostra as curvas analisadas pelo ambiente de verificação para certificar a propriedade *pGBW*.

Nos primeiros instantes, pode-se perceber que as máquinas de estado dos componentes *Estimulo* e *Monitor*, se encontram nos estados *M0* e *S0*, respectivamente. Depois de um certo intervalo de tempo são inicializados os dois componentes por meio da ativação do sinal *comeco*, alterando os estados das máquinas destes componentes para os estados *M1* e *S1* respectivamente, significando que será aplicado um sinal no terminal de entrada do modelo e monitorada a sua saída. Quando a máquina de estado do componente *Monitor* alcançou o estado *M4*, indicando que a propriedade foi satisfeita sob aquelas condições de operação, a simulação é abortada. No gráfico 2, os traços em negrito representam o sinal do estimulador, uma senóide com frequência de 3 MHz, e os traços claros representam o sinal de saída do modelo.



Gráfico 2: Sinais de saída e entrada do modelo de ouro e do circuito aplicadas pelo estimulador. Assim como, as transições das máquinas de estados dos blocos estimulo e monitor. Os resultados representam os dados analisados para a validação da propriedade pGBW.

O gráfico 3 mostra os sinais analisados para a verificação da propriedade *sCMRR*. No gráfico pode-se visualizar três curvas: a intermediária representa o sinal no terminal de saída do modelo quando são aplicados sinais iguais nos seus dois terminais de entrada, bem como, o sinal de estímulo aplicado; na inferior, é mostrado o sinal do terminal de saída do modelo quando é aplicado o mesmo estímulo no terminal de entrada positivo  $V_+$  e nenhum estímulo no terminal de entrada negativo,  $V_-$ ; e o gráfico superior destaca a transição entre os estados das máquinas dos componentes *Monitor* e *Estimulo* ao longo da simulação. Inicialmente, os estados das máquinas de estados dos dois componentes são *S0* e *M0*. Em seguida, o estimulador é acionado por meio do sinal *comeco*, então as suas máquinas de estados seguiram para os estado *S1* e *M1*. Logo após, no estado M2, o componente *Monitor* calcula o *CMRR* do modelo e, como o resultado foi satisfatório, o seu estado seguinte foi *M3*, finalizando a simulação, ou seja, a máquina de estados do componente *Estimulo* foi para o estado *S2*.



Gráfico 3: Sinais de saída e entrada do modelo de ouro e do circuito aplicadas pelo estimulador. Assim como, as transições das máquinas de estados dos blocos estimulo e monitor. Os resultados representam os dados analisados para a validação da propriedade *sCMRR*.

No gráfico 4 estão mostrados os resultados obtidos com a verificação da propriedade *tPSRR*, no qual se destacam três curvas. A curva intermediária destaca o sinal de entrada aplicado no terminal de alimentação, o estímulo; a curva inferior representa o sinal de saída; e a superior mostra a transição das máquinas de estados dos componentes *Monitor* e *Estimulo*. Como podese visualizar, as máquinas de estados dos dois componentes se encontram nos estados *M0* e *S0* até a inicialização da verificação, seguindo para os estados *M1* e *S1*, aplicando, assim, uma pertubação no terminal de alimentação do modelo acionada pelo componente *Estimulo* e iniciando a análise do sinal de saída do modelo monitorada pelo componente *Monitor*. No estado seguinte, *M2*, o componente *Monitor* calcula o *PSRR* do modelo, e verifica se o parâmetro calculado é igual ao especificado, como é o caso, a máquina vai para o estado *M3*, satisfazendo a propriedade. Ao alcançar o estado *M3*, a máquina de estados do componente *Estimulo* seguiu para o estado *S2*, indicando o fim deste ciclo de verificação.



Gráfico 4: Sinais de saída e entrada do modelo de ouro e do circuito aplicadas pelo estimulador. Assim como, as transições das máquinas de estados dos blocos estimulo e monitor. Os resultados representam os dados analisados para a validação da propriedade *tPSRR*.

O gráfico 5 mostra os resultados da simulação para verificar a propriedade qMF. Neste gráfico, tem-se duas curvas, a superior representa a transição das máquinas de estados do componente *Estimulo* e *Monitor*, enquanto a inferior mostra os sinais de entrada e saída do modelo. A curva com os traços em negrito representa o sinal de estímulo e a outra destaca o sinal de saída do modelo. No componente *Monitor*, quando a máquina de estados alcançou o estado M2, imperceptível no gráfico, é calculada a margem de fase e a máquina de estado transita para o estado M4, mostrando que a propriedade não foi satisfeita.



Gráfico 5: Sinais de saída e entrada do modelo de ouro e do circuito aplicadas pelo estimulador. Assim como, as transições das máquinas de estados dos blocos estimulo e monitor. Os resultados representam os dados analisados para a validação da propriedade qMF.

A última propriedade verificada foi a qqSR, os resultados estão mostrados no gráfico 6. Nesse, são destacadas duas curvas, a superior representa a transição das máquinas de estados, enquanto a inferior mostra as curvas analisadas pelo componente *Monitor*. Nesta última curva, a linha com os traços em negrito representa o sinal aplicado no terminal de entrada do modelo, e a linha tracejada clara é o sinal gerado no terminal de saída do modelo. Nos estados S1 e M1, o componente *Estimulo* aplica o estímulo no modelo e o componente *Monitor* calcula a derivada da curva durante um intervalo de tempo. Posteriormente, este valor é comparado com o valor especificado. Como a propriedade foi satisfeita, a máquina de estados do componente *Estimulo* seguiu para o estado M3. Enquanto isso, a máquina de estados do componente *Estimulo* seguiu para o estado S2, significando a finalização do estímulo aplicado. Note, neste caso, a simulação não será finalizada por que também poderá ser calculada a derivada da curva para o sinal de declive da função degrau. No entanto, a propriedade certificada abrange somente a taxa *Slew Rate* de subida.



Gráfico 6: Sinais de saída e entrada do modelo de ouro e do circuito aplicadas pelo estimulador. Assim como, as transições das máquinas de estados dos blocos estimulo e monitor. Os resultados representam os dados analisados para a validação da propriedade qqSR.

Os resultados obtidos destas análises foram impressos em relatórios informando se o modelo comportamental é um modelo de ouro. Caso o modelo não esteja certificado, o relatório mostrará qual propriedade não foi satisfeita, enfatizando sob quais condições o circuito foi analisado. Este documento é escrito durante o processo de verificação, a conclusão dos seus resultados é pré-requisito para dar continuidade ao processo, seguindo para a fase de validação do circuito implementado pela técnica do Modelo de Ouro.

# 4.3.2 Certificação de um Amplificador Operacional pela Técnica do Modelo de Ouro

Nesta sub-seção serão detalhados os resultados da validação do circuito implementado pela técnica do Modelo de Ouro. No entanto, antes de mostrar os resultados, será mostrada a construção dos bancos de teste utilizados na execução desta etapa.

#### 4.3.2.1 Implementação dos Bancos de Teste

Nesta etapa de verificação, o principal objetivo é comparar os sinais provenientes do circuito implementado e do modelo de ouro. Caso, os sinais apresentados pelo circuito implementado tenha um comportamento melhor do que os sinais do modelo de ouro, o processo de verifi-

cação certificará o seu funcionamento. Nesta técnica, resultados satisfatórios são alcançados dependendo da construção de bons bancos de teste.

Na figura 29 está mostrado o banco de teste utilizado para verificar o ganho diferencial do amplificador operacional. No caso, foi utilizado um estímulo do tipo *AC* no terminal positivo e o mesmo estímulo de polaridade invertida no terminal negativo. Os mesmos estímulos, e da mesma forma, serão aplicados no modelo de ouro e no circuito implementado. O bloco *Validador* se encarregará em comparar os sinais provenientes dos modelos. Neste caso, quando o amplificador implementado tiver um ganho superior ao apresentado pelo modelo de referência, este bloco não acusará falha de implementação, caso contrário, uma falha será reportada no relatório final.



Figura 29: Banco de testes utilizado para verificar o ganho diferencial do amplificador operacional.

Na validação da taxa de rejeição do ganho em modo comum do amplificador foi utilizado o banco de teste mostrado na figura 30. Neste caso foram utilizados quatro amplificadores operacionais: dois do modelo de ouro e dois do circuito implementado. Desta forma, o bloco *Validador* poderá medir os ganhos diferenciais e do modo comum, parâmetros necessários para calcular o *CMRR* dos dois modelos e, assim, efetua-se a comparação entre os valores.



Figura 30: Banco de testes utilizado para a validação da taxa CMRR do amplificador operacional.

A validação da taxa de rejeição da variação da tensão de alimentação no sinal de saída do circuito foi executada utilizando o banco de teste mostrado na figura 31. Neste caso, o estímulo AC é aplicado no terminal  $V_{dd}$  do modelo de ouro e do circuito implementado. O terminal positivo dos modelos é conectado ao terra e o negativo é conectado ao terminal de saída. O bloco *Validador* calcula o *PSRR* dos dois modelos e depois os compara.



Figura 31: Banco de testes utilizado para verificar a taxa PSRR do amplificador operacional.

O último banco de teste construído foi para a validação da taxa *Slew Rate*. A figura 32 mostra o banco utilizado para certificar esta figura de mérito. Neste caso, serão aplicados sinais nos terminais positivos na forma de uma função degrau. Durante um determinado intervalo de tempo será calculada a derivada do sinal de saída em relação ao tempo quando o sinal de entrada estiver incrementando e, posteriormente, calcula-se a mesma derivada no instante em que o sinal de entrada estiver decrementando. Deve-se efetuar estes cálculos para os dois modelos e, em seguida, quando a simulação estiver finalizada, comparam as quatros derivadas calculadas.



Figura 32: Banco de testes utilizado para a validação da taxa *Slew Rate* do amplificador operacional.

Com a implementação destes bancos de teste visto anteriormente foi possível construir um ambiente de verificação consistente para a validação automática de um circuito implementado com o auxílio das linguagens *Spice*, *Octave* e *Shell-Scripts*. A seguir, serão vistos os resultados das simulações com o modelo de ouro e com o circuito implementado.

#### 4.3.2.2 Detalhamento dos Resultados da execução dos Bancos de Teste

Os resultados do uso da técnica do Modelo de Ouro são obtidos por meio de simuladores que possibilitem a integração de modelos comportamentais e modelos descritos na linguagem *Spice*. O gerado no uso desta técnica também é um relatório destacando as figuras de mérito satisfeitas e as não satisfeitas. Entretanto, a seguir, estão mostradas as curvas utilizadas pelo bloco *Validador* provenientes da execução dos bancos de teste vistos.

No gráfico 7 estão apresentadas as curvas do ganho diferencial do amplificador operacional implementado e do modelo de ouro. Pode-se perceber que o ganho do amplificador implementado é superior ao do modelo de ouro, ou seja, o bloco *Validador* certificou esta propriedade.



Gráfico 7: Curvas analisadas pelo bloco *Validador* para certificar o ganho diferencial do circuito. Conforme pode ser visto, o ganho diferencial do circuito implementado é superior a do modelo de ouro.

O gráfico 8 mostra as curvas, do circuito implementado e do modelo de ouro, geradas pela execução do banco de teste mostrado na figura 30. No caso, está sendo avaliada a taxa *CMRR*, pode-se perceber que esta taxa no amplificador implementado apresenta um melhor comportamento quando comparado com o apresentado pelo modelo de ouro, satisfazendo a propriedade em questão.



Gráfico 8: Curvas analisadas pelo bloco *Validador* para certificar o CMRR do circuito. Conforme pode ser visto, o CMRR do circuito implementado apresenta um comportamento melhor comparada ao do modelo de ouro.

O gráfico 9 mostra os resultados da simulação do banco de teste mostrado na figura 31. No gráfico, pode-se perceber as curvas geradas após o cálculo efetuado pelo bloco *Validador* para encontrar as grandezas *PSRR* dos dois modelos. Como pode ser comprovado, neste caso, o resultado obtido pelo circuito também foi melhor do que esperado, proveniente do modelo de ouro.



Gráfico 9: Curvas analisadas pelo bloco *Validador* para certificar o *PSRR* do circuito. Conforme pode ser visto, o circuito implementado apresentado uma maior rejeição da variação da tensão de alimentação no seu sinal da saída quando comparado com o do modelo de ouro, certificando o circuito nesta figura de mérito.

No caso da verificação da taxa *Slew Rate* foi executada uma simulação em função do tempo utilizando o banco de teste mostrado na figura 32. O bloco *Estimulo* aplicará formas de ondas da função degrau na entrada dos modelos, conforme mostrado no gráfico 10. Esse também exibe os resultados obtidos, como se pode perceber, no circuito implementado, um aclive e declínio melhor quando comparados com o modelo de ouro.



Gráfico 10: Curvas analisadas pelo bloco *Validador* para certificar o *Slew Rate* do circuito. Conforme pode ser visto, o circuito implementado apresentado um aclive e declínio melhor do modelo de ouro quando aplicada um sinal da forma de degrau no terminal de entrada, certificando o circuito nesta figura de mérito.

Os resultados obtidos verificam o circuito implementado, certificando-o para a construção do seu leiaute. Uma vez que esse esteja projetado, uma nova rodada na execução destes bancos de teste poderá ocorrer para a certificação deste leiaute, podendo ser utilizado o circuito, invés do modelo de ouro, como referência. A técnica do Modelo de Ouro utilizando *scripts* de automatização e com entrada de dados a nível de transação pode ser uma nova abordagem a ser adotada pelos engenheiros de verificação de circuitos analógicos, o qual foi comprovada uma significativa utilidade neste trabalho. Com isso, pode-se fazer análises mais profundas nos circuitos, possibilitando a construção de circuitos mais requintados, obtendo um menor tempo de verificação, com maiores garantias de funcionamento do circuito fabricado.

# 4.4 Conclusão

Nesta seção foi visto como a metodologia proposta pelo autor pode ser utilizada na verificação de um dos blocos mais fundamentais dos circuitos analógicos, o amplificador operacional. Desta forma, foi possível comprovar a utilização de um processo automatizado com elevado nível de abstração na verificação de circuitos analógicos. Na seção seguinte, tem-se os detalhes do uso desta metodologia na validação da implementação em um simulador *Spice*, na linguagem C, do modelo ACM.

# 5 Estudo de Caso B - Validação da implementação em C do modelo ACM

Na presente seção, os resultados da validação da implementação do modelo ACM na linguagem C serão detalhados. A metodologia vista na seção 3 foi utilizada para criar o ambiente de verificação necessário. Inicialmente, será mostrado o plano de verificação, em seguida, a implementação do modelo de ouro e, por último, os resultados da validação do modelo implementado na linguagem C.

# 5.1 Plano de Verificação

O plano de verificação define os principais aspectos no âmbito tecnológico e cronológico para serem seguidos ao decorre do processo de verificação. No caso, serão definidos estes aspectos com o intuito de verificar a implementação do modelo ACM em um simulador computacional de circuitos analógicos, simulador *eldo*.

# 5.1.1 Introdução

Atualmente, a evolução do processo de fabricação provocou uma sofisticação no comportamento dos transistores de efeito de campo quando foram alcançados comprimentos na escala de dezenas de nanômetros. A este comportamento foi designado o termo de efeitos de segunda ordem e de canal curto, e sendo a sua origem explicada pelos conceitos da física quântica.

Com a finalidade de descrever o comportamento dos transistores, incluindo também os efeitos de canal curto, vários grupos desenvolveram modelos matemáticos para conjeturar o comportamento real dos transistores de canal curto, com o intuito de desenvolver simuladores computacionais para circuitos analógicos. Tais modelos derivam-se do modelo de (PAO; SAH, 1966), sendo obtidos por meio de aproximações matemáticas assumindo a invariância do potencial de superfície ou da densidade de cargas de inversão nos terminais do transistor, simplificando-o em modelos consistentes e simples de serem implementados. A implementação destes modelos físico-matemáticos em simuladores computacionais se faz necessária para a validação de projetos de circuitos integrados antes de fabricá-los em pastilha de silício. Entretanto, para as suas implementações em simuladores computacionais são necessárias aproximadamente mil linhas de código escritas na linguagem C. Para isso, com a ajuda de uma equipe de profissionais especializados se traduz as equações matemáticas originárias do modelo em um código incorporado no simulador computacional.

Todavia, o processo de interpretar as equações, codificando-as, poderá ocasionar falhas de implementação nestes simuladores. Além disso, devido a complexidade dos efeitos de segunda ordem e canal curto, tornou a modelagem destes dispositivos uma tarefa complexa, ocasionando na modelagem de modelos físico-matemáticos inválidos, causando até o não funcionamento de vários circuitos integrados ao retornarem das fábricas. Consequentemente, algumas falhas encontradas nos simuladores podem ser também devido a má codificação, gerando algumas vezes falhas de convergência na simulação de circuitos. A utilização de modelos comportamentais para a sua certificação estão se tornado uma ferramenta requisitada.

Notoriamente, necessita-se a execução de um processo de verificação para a validação da implementação do modelo ACM em um simulador computacional. Os aspectos tecnológicos e cronológicos deste processo serão descritos a seguir, assim como, as metas a serem alcançadas.

# 5.1.2 Aspectos Tecnológicos

Os aspectos tecnológicos tem como o escopo definir: os objetivos da verificação, as propriedades a serem validadas, as técnicas utilizadas no processo de verificação, as ferramentas e como será a construção do ambiente certificador.

## 5.1.3 Objetivos

O objetivo do processo de verificação é certificar o funcionamento da implementação do modelo ACM em um simulador computacional de acordo com os comportamentos propostos por (TSIVIDIS, 2003). A verificação proposta é baseada na verificação funcional, utilizando a técnica baseada em simulação. Neste caso, a validação do modelo de ouro será por meio da fundamentação física-matemática com o auxílio das propriedades fundamentais do transistor de efeito de campo.

# 5.1.4 Técnicas

A utilização de uma técnica de verificação para verificar o modelo de ouro é desnecessária tendo em vista a fundamentação física-matemática apresentada pelos autores do modelo ACM. No entanto, serão utilizadas, propriedades fundamentais do comportamento deste transistor para comprovar o seu correto funcionamento sob várias condições de operação.

Ao término da construção do modelo de ouro, inicia-se a validação do modelo implementado no simulador. Neste caso, serão utilizadas instâncias do modelo de ouro e do modelo implementado, aplicando-lhes várias formas de ondas de tensão e corrente, conforme a técnica do Modelo de Ouro. Desta forma, serão verificadas a semelhança dos sinais provenientes dos dois modelos.

## 5.1.5 **Propriedades**

A comprovação do comportamento do modelo de ouro e do modelo implementado será efetuada por meio da validação de uma lista de propriedades. No caso da validação do modelo de ouro foram escolhidos, como propriedades a serem verificadas, valores de tensão, corrente de dreno e transcondutância de porta para serem respeitadas conforme a lei quadrática dos transistores de efeito de campo. Além disso, foram verificados o comportamento de diversas capacitâncias existentes entre os terminais do transistor. No caso da validação do modelo implementado, foram escolhidas as propriedades propostas por (TSIVIDIS, 2003).

A validação das seguintes propriedades se faz necessária para a comprovação do funcionamento do modelo de ouro:

- Propriedade **pIdVds**: "A corrente do terminal Dreno deve ser de 110  $\mu$ A, 211  $\mu$ A e 344  $\mu$ A na saturação para uma tensão de porta de 2V, 2.5V e 3.0V, respectivamente, para um transistor de dimensões W/L de 2, largura do óxido de silício de  $150x10^{-10}m$  e tensão limiar de 0.69V com um erro de no máximo de 10%".
- Propriedade sgmVgs: "A transcondutância de Porta do transistor deve ser de 168μS, 233μS e 297μS para uma tensão de porta de 2V, 2.5V e 3.0V, com a tensão entre os terminais de dreno-fonte de 1.31V, 1.81V e 2.31V respectivamente, para um transistor de dimensões W/L de 20, largura do óxido de silício de 150x10<sup>-10</sup>m e tensão limiar de 0.69V com um erro de no máximo de 10%".
- Propriedade **tCgg**: "O comportamento da capacitância de Porta-Porta deve ser continuo na transição entre as fases do transistores de acumulação, inversão fraca, moderada e

forte, obtendo as mesmas quantidade de cargas normalizadas na acumulação e na inversão forte, conforme mostrada em (GALUP-MONTORO, 2007)".

 Propriedade qCgx: "As capacitâncias de Porta-Dreno e Porta-Fonte em função da tensão entre o dreno e fonte deverão apresentar um comportamento continuo até a saturação das cargas nos terminais. Além disso, na tensão, entre os terminais dreno e fonte, igual a zero, as duas capacitâncias devem apresentar valores iguais".

As propriedades utilizadas para comprovar o funcionamento da implementação pela técnica do Modelo de Ouro estão listadas abaixo:

- Propriedade **pInversaoFracaForte**: "O comportamento da transcondutância de Porta deve apresentar uma curva suave na transição entre os níveis de inversão fraco e forte".
- Propriedade **sTransconduntânciaSaida**: "O comportamento da transcondutância de saída, entre os terminais Dreno e Fonte, deve apresentar uma transição suave entre as regiões de tríodo e saturação".
- Propriedade tSimetria: "A corrente de dreno, próximo a 0 V entre os terminais de dreno e fonte, não deve apresentar nenhuma descontinuidade e nem para a suas derivadas, testando, assim, a inversão da corrente de dreno quando se inverte a tensão entre os terminais Dreno e Fonte".
- Propriedade qAssintotico: "O comportamento da transcondutância de Porta deve ser assintótico para o valor 1/φ<sub>t</sub> quando houver um incremento no valor da tensão entre os terminais Fonte e Corpo. Além disso, não deverá apresentar nenhuma descontinuidade na transição da inversão fraca até a forte".

#### 5.1.5.1 Ambiente

O núcleo do ambiente de verificação, conforme a técnica do Modelo de Ouro, será composto pelos bancos de teste, concebidos com base nas propriedades vistas anteriormente, responsáveis em aplicar estímulos e verificar se os sinais de saídas estão satisfazendo as propriedades, certificando, deste modo, o funcionamento do modelo. Nesta técnica de verificação, os blocos de entrada serão: o modelo de ouro, o conjunto de estímulos e o modelo implementado. Este último é fornecido pelo engenheiro responsável pela implementação do modelo na linguagem C, e os outros são de responsabilidade do engenheiro de verificação.

Este ambiente de verificação será construído com a ajuda do desenvolvimento de *scripts* de automatização através do uso das linguagens *Shell-Scripts* e Perl.

#### 5.1.5.2 Ferramentas

O pacote *ADvanced MS* (ADMS) da *Mentor Graphics* será utilizado para a construção, simulação e validação do modelo comportamental. Tal pacote disponibiliza as seguintes ferramentas: *vasim* para a simulação, *vacom* para a compilação dos arquivos fontes da linguagem VHDL-AMS, *e zwave* para a visualização dos resultados. Para a simulação dos circuitos descritos na linguagem *Spice* será utilizada a ferramenta *eldo* também disponibilizada por este pacote. Além disso, a ferramenta de matemática simbólica, *Octave*, será requisitada para a realização do bloco *Validador* do banco de testes.

Algumas das ferramentas adotadas além das linguagens de automatização, necessárias para a construção do ambiente de verificação proposto, são disponibilizados no sistema operacional Linux sob a licença GPL<sup>1</sup>, como por exemplo: o *Octave*. Desta forma, a utilização deste sistema se faz necessária ao longo do processo de verificação. Assim, foi escolhido o Linux CentOS <sup>2</sup> devido a sua licença livre e sua compatibilidade com o Linux *Red Hat*, atual versão suportada pelo pacote ADMS.

## 5.1.6 Aspectos Cronológicos

Nesta seção do plano detalha-se os aspectos relacionados com o cronograma e a equipe de verificação.

A verificação do modelo de ouro abrange somente quatro propriedades e mais cinco propriedades na execução da técnica do Modelo de Ouro, e como o principal objetivo desta verificação é mostrar como este ambiente de verificação deve ser construído, a equipe de execução será composta somente por um membro, o presente autor.

Este engenheiro de verificação tem como meta cobrir o cronograma mostrado no Quadro 4. Nele pode-se visualizar todas as etapas a serem executadas, dividida pelo número de semanas.

Nas três primeiras semanas será realizado um estudo da fundamentação físico-matemático do funcionamento do transistor proposta pelos autores do modelo ACM, além de definir como o modelo comportamental será construído, mostrando as principais equações a serem implementadas, linguagem utilizada, as portas de saída e entrada e os parâmetros tecnológicos. Na segunda etapa, o modelo comportamental será implementado ao longo de duas semanas. Na quarta semana, será inicializada a comprovação das propriedades do modelo comportamental,

<sup>&</sup>lt;sup>1</sup>Gnu Public License - http://www.gnu.org/copyleft/gpl.html

<sup>&</sup>lt;sup>2</sup>The Community ENTrise Operating System - www.centos.org/

esta validação terá duração de uma semana e poderá retornar a etapa anterior para a otimização do modelo. Com a finalização da validação deste modelo, nas três semanas seguintes será construído o ambiente de verificação para a verificação do modelo implementado no simulador computacional, e nas duas últimas semanas será executada a sua verificação.

Quadro 4: Cronograma a ser seguido pela equipe de verificação para validação da implementação do modelo ACM.

| Atividade/Semana            | Sem.1 | Sem.2 | Sem.3 | Sem.4 | Sem.5 | Sem.6 | Sem.7 | Sem.8 |
|-----------------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| Fund. do Modelo ACM         | X     | X     | X     |       |       |       |       |       |
| Constr. do Modelo Ouro      |       |       | X     | X     |       |       |       |       |
| Comprov. do Modelo Ouro     |       |       |       |       | X     |       |       |       |
| Const. Ambiente de Verif.   |       |       |       |       | Х     | X     | Х     |       |
| Valid. do Mod. no Simulador |       |       |       |       |       |       | Х     | X     |

# 5.1.7 Conclusão

O plano de verificação é o ponto de partida na execução do processo de verificação. Com estes aspectos bem definidos e de forma clara, os engenheiros de verificação obterão uma maior capacidade de executá-lo, alcançando os objetivos com maior eficiência.

# 5.2 Modelo ACM

Os transistores de efeito de campo tem quatro terminais: Dreno, Porta, Fonte e Corpo. O transistor tipo N, mostrado na figura 33, é constituído na sua parte inferior, um substrato do tipo P, com duas regiões do tipo N na parte intermediaria, caracterizando os terminais de Dreno e de Fonte; e na parte superior entre estes dois terminais tem-se o terminal Porta construído com um material de poli-silício, e separando-o do substrato tipo P se localiza o óxido de silício.



Figura 33: Representação geométrica do transistor de efeito de campo, destacando os seus terminais e suas camadas.

A caracterização dos modelos físicos-matemáticos que descrevem o funcionamento dos transistores de efeito de campo se baseia no modelo de (PAO; SAH, 1966). Como este modelo é complexo para a implementação em simuladores computacionais, são construídos modelos compactos baseados em dois tipos de análise: a aproximação da densidade de cargas de inversão nos terminais ou a aproximação da tensão de superfície.

Os modelos de transistores baseados na densidade de cargas de inversão são elaborados conforme o modelo unificado de controle de cargas, do inglês, *Unified Charge Control Model* (UCCM). Este modelo define uma equação base para o cálculo das variações da densidade de carga de inversão em função da variação da tensão de canal do transistor, conforme mostra a equação 11.

$$dQ'_I \left(\frac{1}{nC'_{ox}} - \frac{\phi_I}{Q'_I}\right) = dV_C \tag{11}$$

 $V_C$  é a tensão no canal,  $Q'_I$  é a densidade de cargas de inversão, n é o fator de rampa,  $C'_{ox}$  é a densidade capacitiva do óxido de sicílio,  $\phi_t$  é a tensão térmica.

O modelo *Advanced Compact MOSFET* (ACM) é um conjunto de equações matemáticas baseadas no modelo UCCM e desenvolvido para ser um modelo simples, continuo e consistente em todos os níveis de inversão (CUNHA; SCHNEIDER; GALUP-MONTORO, 1998), definindo uma única equação de corrente de dreno cobrindo todas as regiões de inversão, desde da inversão fraca até a inversão forte (GALUP-MONTORO, 2007), sem a necessidade de interpolações matemáticas como nos modelos mais usados pela indústria. A simplicidade deste modelo também está na quantidade de parâmetros tecnológicos, sendo em torno de 15 parâmetros, enquanto que o modelo popular, o Bsim3v3, o número de parâmetro excede 130.

A seguir estão relacionadas as equações utilizadas para a implementação do modelo ACM. Para maiores detalhes pode ser consultado o livro (GALUP-MONTORO, 2007).

$$I_D = \frac{\mu_n W}{L} \left[ \frac{Q_{IS}'^2 - Q_{ID}'^2}{2nC_{ox}'} - \phi_t (Q_{IS}' - Q_{ID}') \right]$$
(12)

$$\frac{V_P - V_{D(S)}}{\phi_t} = \frac{Q'_{IP} - Q'_{ID(S)}}{nC'_{ox}\phi_t} + \ln\left(\frac{Q'_{ID(S)}}{Q'_{IP}}\right)$$
(13)

$$Q'_{IP} = -nC'_{ox}\phi_t \tag{14}$$

$$V_{FB} = V_{th} - \phi_{s0} - \gamma \sqrt{\phi_{s0}} \tag{15}$$

$$V_G - V_{FB} = \phi_{sa} + \gamma \sqrt{\phi_{sa} + \phi_t (e^{\frac{-\phi_{sa}}{\phi_t}} - 1)}$$
(16)

$$n = 1.0 + \frac{\gamma(1 - e^{\frac{-\phi_{sa}}{\phi_t}})}{2\sqrt{\phi_{sa} + \phi_t(e^{\frac{-\phi_{sa}}{\phi_t}} - 1))}}$$
(17)

$$V_P = \phi_{sa} - \phi_{s0} + \phi_t \left[ 1 + ln \left( \frac{n}{n-1} \right) \right]$$
(18)

$$\left(V_G - V_{t0} + \phi_{so} + \gamma \sqrt{\phi_{s0}} - \phi_{sa}\right)^2 = \gamma^2 \left[\phi_t \left(e^{\frac{-\phi_{sa}}{\phi_t}} - 1\right) + \phi_{sa}\right]$$
(19)

 $I_D$  é a corrente de Dreno,  $V_D$  e  $V_S$  são as tensões dos terminais Dreno e Fonte referenciadas ao terminal Corpo,  $\mu_n$  é a mobilidade dos portadores, W é a largura do canal, L é o comprimento do canal,  $Q'_{ID(S)}$  é a densidade de cargas de inversão no terminal Dreno (Fonte),  $C'_{ox}$  é a densidade capacitiva do óxido de silício,  $V_P$  é a tensão na condição de *pinch-off*,  $\phi_t$  é a tensão térmica,  $Q'_{IP}$ é a densidade de inversão de carga na condição de *pinch-off*, *n* é o fator de rampa,  $V_{th}$  é a tensão limiar,  $\phi_{s0}$  é um parâmetro de ajuste do modelo ACM, definido como  $2\phi_F$ ,  $\gamma$  modela o efeito de corpo, considerado um parâmetro tecnológico,  $\phi_{sa}$  é a tensão de superfície quando a densidade de cargas de inversão no canal for igual a zero.

A seguir, estão listadas as equações para a modelagem dos efeitos de segunda ordem e de canal curto.

$$\mu_0(T) = \mu_0(T_0) \left(\frac{T}{T_0}\right)^{-m}$$
(20)

$$\mu_{eff} = \frac{\mu_0(T)}{1 + \alpha_0 \gamma \sqrt{V_{P0} + \phi_{S0}}} \tag{21}$$

$$v_{sat} = \frac{F_c}{\mu_{eff}} \tag{22}$$

$$q'_{IDsat} = \frac{Q'_{IDsat}}{Q'_{ip}}, \quad q'_{IS} = \frac{Q'_{IS}}{Q'_{ip}}$$
 (23)

$$\frac{V_{DSsat}}{\phi_t} = q'_{IS} - q'_{IDsat} + \ln\left(\frac{q'_{IS}}{q'_{IDsat}}\right)$$
(24)

$$Q'_{IDsat} = Q'_{IS} - nC'_{ox}\phi_t - Q'_A\Upsilon$$
<sup>(25)</sup>

$$\Upsilon = \left[1 - \sqrt{1 - \frac{2(Q_{IS}' - nC_{ox}' \phi_t)}{Q_A'} + \left(\frac{nC_{ox}' \phi_t}{Q_A'}\right)^2}\right]$$
(26)

 $Q'_A = nC'_{ox}L_{eq}F_c \tag{27}$ 

$$L_{eq} = L - \Delta L \tag{28}$$

$$\Delta L = \lambda L_C \ln \left( 1 + \frac{V_D - V_{DSsat}}{L_C \frac{v_{sat}}{\mu_{eff}}} \right)$$
(29)

$$L_C = \sqrt{\frac{\varepsilon_{si} X_J}{C'_{ox}}} \tag{30}$$

$$V_P = V_{P0} + \frac{\sigma}{L^2} (V_D + V_S)$$
(31)

$$I_D = I_{D_{CL}} \frac{1}{1 + \sqrt{\frac{(Q'_{IS} - Q'_{ID})^2}{Q'_A} + \left(\frac{1}{10}\right)^2}}$$
(32)

 $T_0$  é uma tensão de referência, *m* é uma constante de ajuste, entre 1.2-1.4 para o transistor do tipo PMOS e 1.4-1.6 para o transistor do tipo NMOS.  $\mu_{eff}$  é a mobilidade efetiva,  $\alpha_{\theta}$  é o coeficiente de redução da mobilidade,  $F_c$  é o campo crítico no canal, parâmetro tecnológico,  $V_{DSsat}$  é a tensão entre os terminas de Dreno e Fonte quando o canal estiver saturado, ou seja, quando a quantidade de cargas de inversão no canal estiver com o valor de  $Q'_{IDsat}$ ,  $L_{eq}$  é o comprimento equivalente transistor, incluindo o efeito de modulação de canal, equação 28,  $Q'_A$  é a densidade de carga de *Early*, definida pela equação 27,  $\Upsilon$  é um parâmetro computado pela equação 26,  $\lambda$  é o parâmetro da modulação de canal,  $X_J$  é a profundidade da junção,  $\varepsilon_{si}$  é a permissividade elétrica do silício,  $L_c$  é um parâmetro tecnológico da modulação de canal,  $V_{sat}$  é a velocidade dos portadores no canal na condição de saturação,  $\Delta L$  é o ajuste devido a modulação de canal,  $I_{D_{CL}}$  é a corrente de dreno para o canal longo,  $\sigma$  é a magnitude do fator DIBL e  $V_{P0}$  é a tensão de *pinch-off* para transistores de canal longo, dependente somente da tensão aplicada ao terminal Porta.

#### 5.2.1 Implementação em Modelos Computacionais Simuláveis

O modelo ACM foi implementado para ser utilizado nas ferramentas *eldo* da *Mentor Graphics* (SIEBEL, 2007) e *Smash* da *Dolphin* (FILHO, 1999). Este estudo de caso tem como objetivo verificar a implementação no *eldo*, fornecida por (SIEBEL, 2007).

O modelo foi codificado através do uso da linguagem de programação C, utilizando os recursos da ferramenta *link\_eldo*, proveniente do pacote UDM, do inglês, *User Definable Model*. Com esta ferramenta, foi possível compilar o código, gerando uma biblioteca dinâmica. Essa, por sua vez, pode ser acionada na execução do *eldo*.

# 5.3 Processo de Verificação

Nesta seção, estão os detalhes do processo de verificação utilizado para verificar a implementação do modelo ACM.

# 5.3.1 Modelo de Ouro

Embora a implementação do modelo ACM se encontre em simuladores comerciais (FILHO, 1999), o seu modelo comportamental ainda não foi construído. Um modelo deste tipo ajuda na validação da implementação em ferramentas comerciais, ou até mesmo, pode ser implementado tradutores com a finalidade de transformar o modelo comportamental em modelos utilizados em simuladores *Spice* (V.R.; CARTER, 2000) (HU et al., 2005).

A linguagem adotada para construir o modelo de ouro foi a VHDL-AMS. Com esta linguagem é possível desenvolver poderosas ferramentas, suportando a criação de modelos de vários níveis de abstração, incluindo componentes elétricos e não-elétricos (CHRISTEN; BAKA-LAR, 1999). Além disso, VHDL-AMS fornece soluções para equações diferenciais ordinárias e algébricas com variáveis implícita ou explicita (ASHENDEN G. D. PETERSON, 2002).

A entidade do modelo de ouro está mostrada no código 3, cuja finalidade é definir a interface do transistor para posteriormente escrever a arquitetura com as equações do modelo ACM.

| <u> </u>   | IUUEIU ACIVI |                          |  |
|------------|--------------|--------------------------|--|
| E          | NTITY ACM IS |                          |  |
| 2 <b>g</b> | eneric (     |                          |  |
|            | W            | : $real := 1.0e-6;$      |  |
| 4          | L            | : $real := 1.0e-6;$      |  |
|            | TOX          | : $real := 241.0e - 10;$ |  |
| 6          | U0           | : real := $0.079;$       |  |
|            | VT0          | : real := 0.90;          |  |
| 8          | GAMMA        | : real := $0.77$ ;       |  |
|            | TEMP         | : real := 300.0;         |  |
| 10         | PHIS0        | : real := 0.61;          |  |
|            | SIGMA        | : $real := 3.0e - 15;$   |  |
| 12         | XJ           | : $real := 0.25e-6;$     |  |
|            | LAMBDA       | : $real := 0.25;$        |  |
| 14         | THETA        | : real := 0.008;         |  |
|            | VSAT         | : real := 177269.0;      |  |
| 16         | NA           | : $real := 2.498 E17;$   |  |
|            |              |                          |  |

Código 3: Entidade VHDL-AMS, mostrando a interface do modelo de um transistor para um modelo ACM

```
M : real := 1.2 );

18 port (

terminal DRAIN, GATE, SOURCE, BULK: electrical);

20 end entity ACM;
```

Um modelo implementado pela linguagem VHDL-AMS pode ser executado por processos concorrentes e processos simultâneos, mecanismos fornecidos pela linguagem para computar resultados das equações de forma paralela ou serial.

Os processos simultâneos são poderosos artifícios utilizados para implementar modelos com equações na forma analítica conforme descritas na literatura, preservando as relações implícitas. Um exemplo disso, é a implementação da relação da densidade de inversão de carga nos terminais do transistor em função da tensão nos seus terminais, equação 13, sendo necessária somente duas linhas de código para a sua implementação, conforme mostrado no código 4. Se implementassem na linguagem de programação C, seria necessária algumas dezenas de linhas de código para calcular os valores destas densidades, além das manipulações algébricas para transformá-la em uma equação explicita.

Código 4: Parte da codificação da arquitetura do modelo ACM descrita em VHDL-AMS destacando a implementação dos cálculos das densidade de cargas nos terminais do transistor. O simulador/compilador VHDL-AMS calcula estes valores mesmo sendo em equações implícitas.

```
2 architecture behavioral of ACM
                                   is
                          DRAIN
                                  to BULK;
  quantity VD across
                                  to BULK;
4 quantity VG across
                          GATE
  quantity VS across
                          SOURCE to BULK:
6 quantity Ids through
                          DRAIN
                                  to SOURCE;
  . . .
8 begin
  (VP-VS)/PHIT == (QIP-QIS)/(n*COX*PHIT)+log(QIS/QIP);
10 (VP-VD)/PHIT == (QIP-QID)/(n*COX*PHIT)+log(QID/QIP);
12 end architecture behavioral;
```

## 5.3.2 Verificação das Propriedades

Nesta sub-seção serão detalhados os resultados da validação do modelo de ouro.

A primeira propriedade, *pIdVds*, foi validada construindo um banco de teste conforme mostrado na figura 34. O banco de teste tem somente um transistor com uma fonte variável conectada ao terminal Dreno e uma fonte não-variável conectada no terminal Porta. O transistor é polarizado em inversão forte, por meio de três tensões aplicadas, em simulações distintas, no terminal de Porta: 2 V, 2.5 V e 3.0 V. Em seguida, a tensão no terminal de Dreno varia desde 0 V até a tensão de 5 V, ao término da simulação é verificada se os valores das correntes de dreno na saturação satisfazem os valores previstos na propriedade *pIdVds*.



Figura 34: Circuito utilizado para validação da propriedade *pIdVds*.

O resultado da propriedade *pIdVds* pode ser conferido com a ajuda do gráfico 11, pode-se perceber que a corrente de Dreno na saturação para uma tensão no terminal Porta de 2.0V foi de aproximadamente  $110\mu$ A, para a tensão de Porta de 2.5V foi de aproximadamente  $220\mu$ A e para tensão de 3.0 V de  $350\mu$ A. Com isso, a propriedade foi satisfeita.



Gráfico 11: Curvas utilizadas para a validação da propriedade pIdVds, utilizando um fonte variando a tensão entre os terminais de dreno e fonte do transistor para um tensão de porta de 2.0 V, 2.5 V e 3.0 V.

A propriedade *sgmVgs* verifica a transcondutância de porta para alguns valores de tensão entre os terminais Dreno e Fonte. O circuito utilizado para verificar esta propriedade está mos-

trado na figura 35, podendo observar a instância do modelo comportamental, o transistor, com uma fonte variável conectada ao terminal Porta e uma fonte não-variável conectada ao terminal Dreno. Para os valores de tensão no terminal Dreno: 1.31 V, 1.81 V e 2.5 V, são executadas três simulações. Ao final de todas, calcula-se a derivada da corrente de dreno de cada uma para obter as suas transcondutâncias de Porta.



Figura 35: Circuito utilizado para a validação da propriedade sgmVgs.

O resultado da propriedade sgmVgs pode ser conferido no gráfico 12, pode-se perceber que a transcondutância de Porta para a tensão no terminal Dreno de 1.31 V foi de aproximadamente  $260\mu$ S, para a tensão de Dreno de 1.81V foi de aproximadamente  $380\mu$ S e para a tensão de 2.5 V, o valor foi de  $560\mu$ S. Como o resultado apresenta um erro menor que o esperado, é satisfeita a propriedade.



Gráfico 12: Curvas utilizadas para a validação da propriedade *sgmVgs*, dados obtidos com a simulação do circuito mostrado na figura 35 para uma tensão de dreno de 1.31V, 1.81V e 2.5 V.

O circuito mostrado na figura 35 também é utilizado para a validação da propriedade tCgg. No entanto, no caso, o terminal Dreno e Fonte são aterrados, e o terminal Corpo será conectado a uma fonte não-variável. A simulação é executada utilizando duas tensões distintas para o terminal Corpo: 0V e 1.0V. Os resultados estão mostrados no gráfico 13. Nele, pode-se perceber o comportamento contínuo apresentado pela capacitância Porta-Porta desde a acumulação até a inversão forte, normalizada pela capacitância do óxido de silício.



Gráfico 13: Curvas utilizadas para verificar a propriedade tCgg, através da simulação do circuito apresentado na figura 35.

O mesmo circuito utilizado na simulação anterior serve para verificar a propriedade qCgx. No entanto, no caso, os terminais Dreno, Fonte e Corpo serão aterrados. A simulação ocorre com a variação da fonte conectada no terminal Porta do valor de 0.0 V até o valor de 3.0 V. O gráfico 14 mostra as curvas das capacitâncias computadas na simulação do circuito. Nelas, pode-se perceber o comportamento continuo das capacitâncias desde a região de tríodo até a saturação, com valores iniciais iguais, validando assim a propriedade. Alguns modelos de transistores podem apresentar um comportamento falho quando não apresentam valores iniciais iguais para estas capacitâncias (TSIVIDIS, 2003).



Gráfico 14: Curvas utilizadas para validação da propriedade qCgx, apresentando valores iniciais iguais, para a tensão entre os terminais Dreno-Fonte igual a zero, e um comportamento continuo até a saturação.

# 5.3.3 Certificação pela Técnica por Modelo de Ouro

Ao término da validação das propriedades do modelo comportamental, transformando-o em um modelo de ouro, pode-se inicializar a construção do ambiente de verificação para executar a técnica do Modelo de Ouro. Nesta sub-seção estão apresentados os resultados da execução deste ambiente para verificar o modelo ACM implementado no simulador *eldo*. Além da comparação entre os dois modelos, examinou-se a similaridade dos resultados com as curvas apresentadas por um modelo de transistores largamente utilizado pela indústria de semicondutores, o Bsim3v3.

#### 5.3.3.1 Implementação dos Bancos de Teste

O banco de teste é um bloco responsável em aplicar os estímulos e verificar os sinais de saída dos modelos. Para a validação das propriedades do modelo implementado no simulador foi utilizado como base o banco de testes proposto na seção 3. Neste banco instancia-se dois modelos: o modelo de ouro e o modelo a ser verificado.

Os bancos de teste utilizados para certificar as propriedades *pInversaoFracaForte* e *qAssintotico* se baseiam no circuito mostrado na figura 35. Na validação da propriedade *sTransconduntânciaSaida*, o banco de teste é construído com base no circuito mostrado na figura 34. Na validação da propriedade *tSimetria*, o circuito utilizado está mostrado na figura 36. Nela pode-se perceber a instanciação de três fontes independentes e uma dependente no terminal Fonte. O uso da fonte dependente é para visualizar o comportamento do transistor quando ocorre a inversão da tensão Dreno-Fonte para valores de transição entre valores negativos e positivo da fonte independente  $V_x$  ocasionando uma inversão dos terminais Dreno e Fonte. Isso ocorre, porque o transistor de efeito de campo é um dispositivo simétrico, sendo determinado os seus terminais pelo modo como é polarizado.



Figura 36: Circuito utilizado para a execução do teste de simetria dos modelos ACM e Bsim3v3.

O último teste submetido aos modelos ACM e Bsim3v3 foi a verificação do erro da corrente de Dreno quando se é implementado um circuito com uma associação de transistores, incluindo um espelho de corrente. O circuito utilizado para este teste está mostrado na figura 37. Este circuito tem uma fonte de corrente de  $20 \,\mu$ A, denominada pela sigla  $I_{ref}$ . A corrente desta fonte é espelhada por um transistor com os terminais de Porta e Dreno conectados, caracterizando que o transistor se encontra sempre na região de saturação. Além disso, tem-se mais quatro transistores ligados em serie-paralelo, e deve ser verificada em cada ramo desses a corrente de  $10 \,\mu$ A, denominada por  $I_o$ .



Figura 37: Circuito utilizado para se obter o erro na corrente de dreno quando utilizados espelhos de corrente com associações de transistores em serie-paralelo, quando são utilizados transistores de canal curto.

No entanto, normalmente, os modelos apresentam um erro quando é implementado circuitos com associação de transistores de canal curto. O cálculo deste erro é efetuado por meio da equação 33. A simulação do circuito deve variar o comprimento de canal do transistor desde o comprimento mínimo até o comprimento de 10 vezes deste mínimo comprimento.

$$Erro(\%) = 1 - \left| \frac{I_{ref} - I_o}{I_{ref}/2} \right|$$
(33)

#### 5.3.3.2 Detalhamento dos Resultados obtidos na Execução dos Bancos de Teste

Nesta sub-seção detalham-se os resultados das simulações dos bancos de teste implementados para verificar as propriedades *pInversaoFracaForte*, *qAssintotico*, *sTranscondutânciaSaida* e *tSimetria*.

A propriedade *pInversaoFracaForte* tem como objetivo verificar o comportamento da corrente de Dreno na escala logarítmica desde a inversão fraca até a inversão forte. Se alguma descontinuidade for identificada na transição entre estes níveis de inversão significa que o modelo é implementado com interpolações matemáticas, ou seja, construído por meio de várias equações, uma para cada nível de inversão, caracterizando a invalidade do modelo. O gráfico 15 mostra as curvas dos modelos ACM em VHDL-AMS e em C e do modelo Bsim3v3 na transição do nível de inversão fraca para o nível de inversão forte. Como pode-se perceber as curvas apresentam comportamentos válidos, no nível de inversão moderada. Entretanto, para um fraquíssimo nível de inversão o modelo Bsim3v3 não apresentou um comportamento esperado, como é visto nos modelos ACM, mostrando que, neste nível de inversão, o modelo Bsim3v3 não está mostrando um comportamento válido comprovando que este modelo não está bem modelado em níveis fraquíssimos de inversão.



Gráfico 15: Validação da corrente de dreno na transição da inversão fraca para a forte em escala logaritmo com um  $V_{DS} = 1.0V$ . As curvas são dos modelos ACM em VHDL-AMS e em C, e com o modelo largamente utilizado pela indústria, BSIM3v3, o qual não possui um comportamento adequado na inversão fraquíssima.

A propriedade *tTranscondunciaSaida* verifica alguma descontinuidade na transcondutância de Dreno na transição entre as regiões de operação de tríodo e saturação. Isso, permite identificar alguma falha ocasionada pelo uso de uma cláusula *IF* para distinguir o uso de diferentes equações para cada região de operação do transistor.

No gráfico 16, tem-se os resultados da validação desta propriedades para os modelos. Nas curvas, pode-se perceber que os modelos apresentam uma suave transição entre as regiões de operação, indicando a ausência de alguma cláusula de condição no modelo, satisfazendo a propriedade.


Gráfico 16: Verificação do comportamento da transcondutância de dreno na transição da região de tríodo e saturação para os modelos ACM e BSIM3v3, os três modelos apresentaram boa transição entre as regiões de tríodo e saturação.

A propriedade *tSimetria* é utilizado para visualizar o comportamento da corrente de Dreno sob uma tensão entre os terminais de Dreno e Fonte próxima ao valor de 0.0 V,  $V_{DS} \cong 0V$ . Para ser visualizado o comportamento válido, deve-se identificar uma transição suave ao passar neste ponto de operação, podendo perceber a inversão dos terminais no transistor.

O gráfico 17 mostra as curvas para os três modelos e suas derivadas. Estas últimas, verificam o comportamento dos três modelos, mostrando que os modelos implementados são simétricos.



Gráfico 17: Curvas utilizada para validação dos modelos ACM e BSIM3v3 para o teste de simetria, provando que o modelo é simétrico para tensões de dreno-fonte próximo a zero V, os três modelos apresentaram ser simétricos.

A propriedade *tAssintotico* verifica o comportamento da relação gm/Id, transcondutância de Porta sobre a corrente de Dreno, em função do nível de inversão para várias tensões aplicadas no terminal Corpo. A continuidade deve ser preservada durante a transição no nível de inversão fraca para o nível de inversão forte. Além disso, o valor dessa relação deve apresentar uma tendência assintótica para o valor  $1/\phi_t$  com o incremento da tensão aplicada no terminal Corpo. Vários modelos apresentam valores constantes no nível de inversão fraca para cada incremento, o que os tornam modelos inválidos (TSIVIDIS, 2003).

O gráfico 18 mostra as curvas para a validação da propriedade *tAssintotico* para os modelos ACM e Bsim3v3. A curva apresentada pela implementação do modelo ACM em C não mostrou uma curva satisfatória quando comparado ao modelo de ouro. Isso porque, o comportamento esperado era a não obtenção de valores constantes no nível de inversão fraca, conforme pode ser comprovado com a curva desempenhada pelo modelo de ouro, a qual mostra um incremento da relação gm/Id em função do nível de inversão no canal. O modelo Bsim3v3 apresentou o comportamento esperado.



Gráfico 18: Validação da relação da corrente de dreno e sua transcondutância com a variação do nível de inversão para várias tensões de corpo. Com o incremento desta tensão, o valor da relação deverá tender para o valor  $1/\phi_t$  e apresentando valores não constantes na inversão fraca.

No último teste proposto, verificando o comportamento dos modelos com a simulação do circuito mostrado da figura 37, o modelo do ACM implementado no simulador não convergiu. O erro foi descrito no relatório e enviado para o autor do modelo. No entanto, os resultados da simulação incluindo somente o modelo Bsim e o modelo de ouro estão mostrados no gráfico 19.



Gráfico 19: Erros no espelhos de corrente na associação de transistores do modelo de ouro do ACM e do modelo Bsim3v3. O modelo ACM implementado no simulador não convergiu para variação de parâmetros. Os resultados mostram que mesmo para canais curtos e longos o modelo ACM mostra menores erros em relação as correntes de dreno do modelo Bsim3v3.

## 5.4 Conclusão

Esta seção mostrou a validação da implementação do modelo ACM na linguagem C através do uso da técnica do Modelo de Ouro, utilizando um modelo escrito na linguagem VHDL-AMS. O uso desta linguagem proporcionou a descrição do modelo ACM na forma encontrada na literatura, utilizando-se as equações com variáveis explicitas ou até mesmo implícitas. Com isso, o modelo de ouro foi construído utilizando as exatas equações do modelo ACM, com base na literatura (GALUP-MONTORO, 2007), sem a necessidade de artifícios matemáticos ou manipulações algébricas.

Para certificar o modelo comportamental, foram utilizadas algumas propriedades básicas, como: verificação das correntes de Dreno e transcondutâncias de Porta por meio da lei quadrática; e também para verificar o comportamento das capacitâncias de Porta-Porta, Dreno e Fonte.

Ao final da seção, são mostrados resultados das execuções dos bancos de teste proposto por TSIVIDIS para validação de modelos de transistores. Com isso, comparou-se os dois modelos,

além disso, fez-se uma comparação com o modelo largamente utilizado Bsim3v3 com o intuito de mostrar as vantagens do ACM.

Com os resultados, foi possível mostrar que o modelo ACM apresentou um melhor comportamento quando comparado ao modelo Bsim3v3. Entretanto, o modelo no simulador não foi verificado por não convergir em um dos testes submetidos e por apresentar um comportamento inválido para a relação gm/Id.

Na próxima seção, tem-se as conclusões do autor ao desenvolver a metodologia para circuitos analógicos e implementar os ambientes de verificação. Além disso, tem-se algumas recomendações para trabalhos futuros que podem vir a surgir a partir deste documento.

## 6 Conclusões

Na concepção de um produto, um dos principais objetivos é a redução de custos de produção, além do custo de engenharia. Somente o custo dos testes em circuitos podem alcançar valores próximos a 50% do valor total do produto.

Além disso, a cada período de tempo as funcionalidades destes circuitos integrados aumentam. Com isso, há um significativo aumento das falhas encontradas ao longo do processo de concepção. Estas falhas quando constatadas no período de testes de protótipos básico, eleva o custo de engenharia, incluindo o custo de reprojetar o circuito, e quando constatadas falhas com produtos no mercado podem ter resultados economicamente desastrosos.

Isso porque o lançamento de um produto defeituoso no mercado pode ocasionar custos altíssimos as empresas podendo levá-las a falência. Desta forma, o uso de processos certificadores do funcionamento do circuito antes desses serem enviados às fábricas é uma forma de reduzir o seu custo de engenharia.

Uma das formas encontradas para reduzir este custo é a execução de um processo de verificação de circuitos integrados. O uso deste processo tem como principal objetivo reduzir o custo de testes, garantido que os circuitos irão funcionar quando fabricados em pastilhas de silício.

A verificação em circuitos digitais está avançada em relação a dos circuitos analógicos, sendo justificado pelo comportamento desempenhado pelos circuitos digitais em relação aos dos circuitos analógicos. Além disso, esses herdaram uma grande parte das funcionalidades dos sistemas, fazendo com que os circuitos analógicos desempenhem, quase sempre, funcio-nalidades como conversão de grandeza ou amplificação de sinais. A construção de circuitos digitais alcançaram elevados níveis de otimização mostrando que os circuitos integrados do domínio analógico são os responsáveis pelo maior do número de falhas dos sistemas.

Uma das principais vantagens na verificação do circuitos digitais é a sua independência da ação humana no processo de verificação, que é quase totalmente automatizado, artificio ainda não explorado pelos engenheiros de verificação analógica. Há várias tentativas de utilizar algoritmos de verificação digital na identificação de falhas no projeto de circuitos analógicos. Como a natureza dos circuitos digitais é distinta dos analógicos, os engenheiros de verificação

não obtiveram muito êxito.

Entretanto, algumas alterações na adoção dos conceitos digitais para verificar os circuitos analógicos foram sendo feitas. Com isso, foi possível verificar o funcionamento de circuitos de unidades básicas.

Em (CHANG; KUNDERT, 2007) foi proposto um processo de verificação para circuitos analógicos. Neste processo, são utilizados modelos comportamentais para verificar circuitos. No entanto, não é proposta uma forma para certificar o modelo comportamental. A principal diferença do trabalho de CHANG; KUNDERT e o que está sendo proposto, é a utilização de uma técnica de verificação para automatizar a verificação do funcionamento correto do modelo construído, uma técnica para verificar automaticamente modelos comportamentais.

Esta dissertação apresentou uma metodologia prática para ser utilizada na verificação de circuitos analógicos. Para isso, foram sugeridas a inclusão de duas técnicas de verificação em um processo de verificação já conhecido na literatura. Estas técnicas têm como ponto primordial aumentar o nível de automatização do processo, excluindo ao máximo a interferência humana. Além disso, foi possível obter um maior formalismo tornando o processo de verificação numa abordagem semi-formal. Consequentemente, no uso desta metodologia conseguiu-se uma maior confiabilidade nos resultados obtidos.

Esta metodologia foi aplicada na verificação do funcionamento de um amplificado operacional. Com os resultados obtidos, mostrou-se a utilidade de se construir um ambiente de verificação com bancos de teste a nível de abstração mais elevado, a nível de transação. Além disso, foi utilizada a técnica do Modelo de Ouro para verificar uma implementação de um modelo de transistores, o ACM, em um simulador de circuitos. Com esta verificação, pode-se ter uma verificação de algumas centenas linhas de código em C com o uso de modelos comportamentais, contendo as exatas equações definidas na literatura do modelo ACM.

O tempo para verificar o amplificador operacional utilizando as técnicas propostas pela metodologia foi de aproximadamente 14 segundos, sem a iteração humana. No entanto, o tempo gasto para verificar o modelo de ouro foi de aproximadamente 2 minutos. Apesar do tempo gasto para desenvolver estes ambientes de verificação, pode-se obter uma redução no tempo de verificação do circuito implementado. Desta forma, pode-se concluir que a ferramenta proporcionou um significativo avanço na verificação dos parâmetros da especificação do circuito.

Como trabalhos futuros, pode-se sugerir a construção de um ambiente de verificação no qual possa elaborar documentos contendo sugestões de otimização caso ocorra alguma falha no funcionamento do circuito. Além disso, pode ser construído uma malha de realimentação para o circuito, com o intuito de o processo de verificação alterar algumas partes da implementação do projeto para proporcionar ajustes finos no projeto. Também podem ser utilizados os conceitos de engenharia ontológica para construir uma ferramenta capaz de desenvolver bancos de teste diretamente das propriedades definidas no plano de verificação. Desta forma, as sentenças podem ser utilizadas como dados de entrada nos ambientes de verificação, assim, estes deverão construir o banco de teste específico para aquela propriedade.

## Referências

ALLEN DOUGLAS R. HOLBERG, A. P. E. *CMOS Analog Circuit Design*. [S.l.]: Oxford University Press, 2008. ISBN 0195116445.

ANDERSON, J. B. E. C. A. H. T.; NIGHTINGALE, A. Systemverilog reference verification methodlogy: Vmm. In: . [S.l.: s.n.], 2006.

ANDERSON JANICK BERGERON, A. N. T. Systemverilog reference verification methodlogy: introduction. In: . [S.l.: s.n.], 2006.

ANDERSON JANICK BERGERON, E. C. A. H. T.; NIGHTINGALE, A. Systemverilog reference verification methodlogy: Esl. In: . [S.l.: s.n.], 2006.

ANDERSON JANICK BERGERON, E. C. A. H. T.; NIGHTINGALE, A. Systemverilog reference verification methodlogy: Rtl. In: . [S.l.: s.n.], 2006.

ASHENDEN G. D. PETERSON, D. A. T. P. J. *The System Designer's Guide to VHDL-AMS*. [S.1.]: Morgan Kaufmann, 2002.

BAIER JOOST-PIETER KATOEN, K. G. L. C. *Principles of Model Checking*. [S.1.]: The MIT Press, 2008. ISBN 026202649X.

BAKER, R. J. *CMOS Circuit Design, Layout, and Simulation*. [S.l.]: Wiley-IEEE Pres, 2007. ISBN 0470229411.

BERGERON, J. Writing Testbenches: Functional Verification of HDL Models, Second Edition. [S.l.]: Springer, 2003. Hardcover. ISBN 1402074018.

BERGERON, J. et al. *Verification Methodology Manual for SystemVerilog*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005. ISBN 0387255389.

BHASKAR, K. U. et al. A universal random test generator for functional verification of microprocessors and system-on-chip. In: *VLSID '05: Proceedings of the 18th International Conference on VLSI Design held jointly with 4th International Conference on Embedded Systems Design*. Washington, DC, USA: IEEE Computer Society, 2005. p. 207–212. ISBN 0-7695-2264-5.

CHANG, H.; KUNDERT, K. Verification of complex analog and rf ic designs. *Proceedings of the IEEE*, v. 95, n. 3, p. 622–639, March 2007. ISSN 0018-9219.

CHRISTEN, E.; BAKALAR, K. Vhdl-ams-a hardware description language for analog and mixed-signal applications. In: . [S.l.: s.n.], 1999. v. 46, n. 10, p. 1263–1272.

CUNHA, A.; SCHNEIDER, M.; GALUP-MONTORO, C. An mos transistor model for analog circuit design. *Solid-State Circuits, IEEE Journal of*, v. 33, n. 10, p. 1510–1519, Oct 1998. ISSN 0018-9200.

DOBOLI, A.; VEMURI, R. Behavioral modeling for high-level synthesis of analog and mixed-signal systems from vhdl-ams. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, v. 22, n. 11, p. 1504–1520, Nov. 2003. ISSN 0278-0070.

DONG, Z. J. et al. Checking properties of pll designs using run-time verification. *Microelectronics*, 2007. *ICM* 2007. *Internatonal Conference on*, p. 125–128, Dec. 2007.

DONG, Z. J. et al. Run-time verification using the vhdl-ams simulation environment. *Circuits and Systems*, 2007. *NEWCAS 2007. IEEE Northeast Workshop on*, p. 1513–1516, Aug. 2007.

DRECHSLER, R. Advanced Formal Verification. Norwell, MA, USA: Kluwer Academic Publishers, 2004. ISBN 1402077211.

DUENAS, C. Verification and test challenges in soc designs. *Integrated Circuits and Systems Design, 2004. SBCCI 2004. 17th Symposium on*, p. 9–, Sept. 2004.

FILHO, O. D. C. G. *UM MODELO COMPACTO DO TRANSISTOR MOS PARA SIMULAÇÃO DE CIRCUITOS*. Tese (Doutorado) — Federal University of Santa Catarina, Brazil, Setembro 1999.

FORDE STEVE BISHOP, R. S. V. S. Streamlining hdl code coverage analysis. *EETimes Magazine*, Dec. 1998.

FUJITA, M.; GHOSH, I.; PRASAD, M. Verification Techniques for System-Level Design. [S.l.]: Morgan Kaufmann, 2007. ISBN 0123706165.

GALUP-MONTORO, M. S. C. *Mosfet Modeling for Circuit Analysis And Design*. [S.l.]: World Scientific Publishing Company, 2007.

GHOSH, A.; VEMURI, R. Formal verification of synthesized analog designs. In: . [S.l.: s.n.], 1999. p. 40–45.

GILDENBLAT G. D. J. SMIT, D. B. M. K. G. Introduction to psp mosfet model. (*Invited paper*) Workshop on Compact Modeling, p. 19–24, 2005.

GRABOWSKI DANIEL PLATTE, L. H. D.; BARKE, E. Time constrained verification of analog circuits using model-checking algorithms. In: *Proceedings of the First Workshop on Formal Verification of Analog Circuits (FAC 2005)*. [S.1.]: Sciense Direct, 2005. p. 37–52.

GRAY PAUL HURST, S. L. P.; MEYER, R. G. Analysis and Design of Analog Integrated *Circuits*. [S.l.]: Wiley, 2001. ISBN 0471321680.

HARTONG, W.; HEDRICH, L.; BARKE, E. Model checking algorithms for analog verification. In: . [S.l.: s.n.], 2002. p. 542–547. ISSN 0738-100X.

HASSOUN, S. et al. A transaction-based unified architecture for simulation and emulation. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, v. 13, n. 2, p. 278–287, Feb. 2005. ISSN 1063-8210.

HEDRICH, L.; BARKE, E. A formal approach to verification of linear analog circuits with parameter tolerances. In: [S.l.: s.n.], 1998. p. 649–654.

HU, B. et al. Developing device models. *IEEE CIRCUITS and DEVICES MAGAZINE*, p. 6–11, 2005.

LACHISH, O. et al. Hole analysis for functional coverage data. *Design Automation Conference*, 2002. *Proceedings*. 39th, p. 807–812, 2002. ISSN 0738-100X.

LI MITCHELL THORNTON, S. S. L. Integrated design validation: Combining simulation and formal verification for digital integrated circuits. *Journal of Systemics, Cybernetics and Informatics*, v. 4, n. 2, p. 22–30, 2002.

LIMA, M. et al. ipprocess: Using a process to teach ip-core development. In: *MSE '05: Proceedings of the 2005 IEEE International Conference on Microelectronic Systems Education.* Washington, DC, USA: IEEE Computer Society, 2005. p. 27–28. ISBN 0-7695-2374-9.

MARINO, C. et al. Mixed signal behavioral verification using vhdl-ams. In: . [S.l.: s.n.], 2005. v. 2, p. 115–118.

MEYER, A. Principles of Functional Verification. [S.I.]: Newnes, 2003. ISBN 0750676175.

MOORE, G. E. *Cramming more components onto integrated circuits*. Electronics, 1965. Disponível em: <a href="http://download.intel.com/research/silicon/moorespaper.pdf">http://download.intel.com/research/silicon/moorespaper.pdf</a>>.

NOVINITE. Intel launches core i7, claims fastest processor on planet. *Sofia News Agency*, November 2008. Disponível em: <a href="http://www.novinite.com/view\_news.php?id=98999">http://www.novinite.com/view\_news.php?id=98999</a>>.

OOISHI, M. Tsmc takes lead in 45nm ic mass production. *Nikkei Electronics Asia*, July 2007. Disponível em: <a href="http://techon.nikkeibp.co.jp/article/HONSHI/20070626/134824/>">http://techon.nikkeibp.co.jp/article/HONSHI/20070626/134824/></a>.

OZEV, S.; ORAILOGLU, A. An integrated tool for analog test generation and fault simulation. In: . [S.1.: s.n.], 2002. p. 267–272.

PAO, H.; SAH, C. Effects of diffusion current on characteristics of metal-oxide (insulator)-semiconductor transistors. *Solid-State Electronics*, v. 9, n. 10, p. 927–937, October 1966. ISSN 00381101.

PERRY, H. F. D. *Applied Formal Verification: For Digital Circuit Design*. [S.l.]: McGraw-Hill Professional, 2005. ISBN 007144372X.

PIZIALI, A. Functional Verification Coverage Measurement and Analysis. [S.I.]: Springer, 2004. Hardcover. ISBN 1402080255.

ROBERTS, G.; DUFORT, B. Making complex mixed-signal telecommunication integrated circuits testable. *Communications Magazine, IEEE*, v. 37, n. 6, p. 90–96, Jun 1999. ISSN 0163-6804.

SALEM, A. Semi-formal verification of vhdl-ams descriptions. In: *Circuits and Systems*, 2002. *ISCAS 2002. IEEE International Symposium on*. [S.l.: s.n.], 2002. v. 5, p. V–333–V–336 vol.5.

SIEBEL, O. F. *Um modelo eficiente do transistor MOS para projeto de circuito VLSI*. Dissertação (Mestrado) — Federal University of Santa Catarina, Brazil, March 2007.

SILVA, K. R. G. da. *Uma Metodologia de Verificação Funcional Visando o Acompanhamento do Processo de Desenvolvimento de Circuitos Digitais Integráveis*. 160 f. Tese (Doutorado em Engenharia Elétrica) — Universidade Federal de Campina Grande, Campina Grande, 2007.

STEINHORST, S.; HEDRICH, L. Model checking of analog systems using an analog specification language. In: *DATE '08: Proceedings of the conference on Design, automation and test in Europe*. New York, NY, USA: ACM, 2008. p. 324–329. ISBN 978-3-9810801-3-1.

SUTCLIFFE, G. Automated theorem proving: A review. AI Magazine, v. 23, n. 1, p. 121, 2002.

TSIVIDIS, Y. *Operation and Modeling of the MOS Transistor*. [S.l.]: Oxford University Press, 2003.

VAHID, F.; GIVARGIS, T. Embedded System Design: A Unified Hardware/Software Introduction. New York, NY, USA: John Wiley & Sons, Inc., 2001. ISBN 0471386782.

VASUDEVAN, S. *Effective Functional Verification: Principles and Processes*. [S.I.]: Springer, 2006. Hardcover. ISBN 0387286012.

V.R., K.; CARTER, H. Modeling and simulating semiconductor devices using vhdl-ams. In: *IEEE/ACM International Workshop on Behavioral Modeling and Simulation*. [S.l.: s.n.], 2000. p. 22–27.

WAMBACQ, G. G. P.; GERRITS, J. Low-Power Design Techniques and CAD Tools for Analog and RF Integrated Circuits. [S.I.]: Springer, 2001. ISBN 0792374320.

WILE JOHN GOSS, W. R. B. Comprehensive Functional Verification: The Complete Industry Cycle (Systems on Silicon). [S.I.]: Morgan Kaufmann, 2005. ISBN 0127518037.

ZAKI, M. H.; TAHAR, S.; BOIS, G. Formal verification of analog and mixed signal designs: Survey and comparison. *Circuits and Systems, 2006 IEEE North-East Workshop on*, p. 281–284, June 2006.

ZAKI, M. H.; TAHAR, S.; BOIS, G. A practical approach for monitoring analog circuits. In: *GLSVLSI '06: Proceedings of the 16th ACM Great Lakes symposium on VLSI*. New York, NY, USA: ACM, 2006. p. 330–335. ISBN 1-59593-347-6.

ZHONG-HAI, W.; YI-ZHENG, Y. The improvement for transaction level verification functional coverage. *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*, p. 5850–5853 Vol. 6, May 2005.