

UNIVERSIDADE FEDERAL DA PARAÍBA
Larissa de Melo Soares

**Análise de Defeitos na Perspectiva do Leiaute
por meio de Extração de Elementos Parasitas**

João Pessoa

2017

UNIVERSIDADE FEDERAL DA PARAÍBA
Larissa de Melo Soares

Análise de Defeitos na Perspectiva do Leiaute por meio de Extração de Elementos Parasitas

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica - PPGEE, da Universidade Federal da Paraíba - UFPB, como requisito para obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Cleonilson Protásio de Souza

JOÃO PESSOA

2017

S323a Soares, Larissa de Melo

Análise de Defeitos na Perspectiva do Leiaute por meio de Extração de Elementos Parasitas / Larissa de Melo Soares. - JOÃO PESSOA, 2017. 100f. : il. (algumas color.)

Orientador: Prof. Dr. Cleonilson Protásio de Souza
Dissertação (Mestrado) - UFPB/CEAR

1. Engenharia Elétrica. 2. Engenharia de Testes. 3. Circuitos Integrados.
4. Defeito. 5. Leiaute. 6. Testes. I. Título II. Cleonilson Protásio de Souza.

UFPB/BC

CDU: 621.38

UNIVERSIDADE FEDERAL DA PARAÍBA
Larissa de Melo Soares

Análise de Defeitos na Perspectiva do Leiaute por meio de Extração de Elementos Parasitas

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica - PPGEE, da Universidade Federal da Paraíba - UFPB, como requisito para obtenção do título de Mestre em Engenharia Elétrica.

A Comissão Examinadora abaixo assinada, aprova a Dissertação de Mestrado

Prof. Dr. Cleonilson Protásio de Souza
Orientador - Presidente da Banca

Prof. Dr. Antônio Carlos Cavalcanti
Membro Externo

**Prof. Dr. Diomadson Rodrigues
Belfort**
Membro Externo

JOÃO PESSOA, 24 de fevereiro de 2017

Dedicatória

Este trabalho é dedicado aos meus pais, Mário Francisco Ferreira Soares e Soraya Alves de Melo Soares.

Agradecimentos

Agradeço primeiramente a Deus, à minha família que sempre me apoiou em todas as minhas decisões na vida acadêmica. Essas pessoas são minha mãe Soraya Alves de Melo Soares, que me deu todo suporte ao longo dessa jornada, ao meu pai Mário Francisco Ferreira Soares, que nunca me deixou desistir dos meus objetivos e ao meu sobrinho Arthur de Melo Paiva, que mesmo com pouca idade e sem perceber me deu o apoio mais sincero e gratuito que poderia receber. Aos membros da minha família que mesmo distantes me apoiam sempre como Paulo Lavedoque, Rita Ferreira, Kelsilene Melo (por ser minha segunda mãe), Alexandre Brasil, Talita Melo, Anderson Melo (pelo apoio incondicional à minha vida acadêmica), Rosilene Ribas, Júlia Melo, Paulo Alexsander Melo (por ser um dos responsáveis da escolha em Engenharia Elétrica), Erika Barros e Heitor Melo.

Agradeço ao Professor Cleonilson Protásio de Souza pelos ensinamentos acadêmicos e que tornou possível a realização deste mestrado.

Agradeço aos amigos Ítalo, Rafael Ângelo, Rafael Duarte, Pamela (pela amizade e ajuda excepcional neste trabalho), Leonardo (por seus conselhos na pesquisa), Verônica, Victor, Ranieri, Lucas, Luan, Nathália, Larissa Aguiar, Andréa, Tathiana, Dallianny, Hugo, Bárbara e a todos os amigos que não me vem na memória mas que guardo na memória e agradeço sempre.

Agradeço à CAPES pelo fomento às minhas atividades de pesquisa.

E, por fim, agradeço à todos que direta ou indiretamente contribuíram para este trabalho.

Resumo

Modelos de falhas clássicos, que representam defeitos físicos em circuitos integrados, já não satisfazem completamente os requisitos atuais para testes. Por isso, atualmente tem-se estudado modelos de falhas que sejam baseados no próprio leiaute do circuito integrado a ser testado, ao invés de se basear no diagrama lógico do circuito como é feito nos modelos clássicos. Modelos baseados em leiaute possibilitam verificar as áreas mais prováveis de acontecer uma falha em consequência de um defeito físico. Nesta dissertação é realizado um estudo dos principais modelos clássicos de falhas e dos modelos baseados na perspectiva do leiaute em que são explanados conceitos como: defeitos físicos, falhas, elementos parasitas, área crítica, nível de defeito e suas correlações. Posteriormente, é proposta uma modelagem de defeitos na perspectiva do leiaute e apresentado um método de extração de elementos parasitas do leiaute utilizando o CADENCE juntamente com as análises simulacionais e resultados obtidos. Ao final, será explicitado como elementos parasitas de leiaute são relacionados com as probabilidades de ocorrência de falhas devido ao projeto do leiaute. Também foram obtidos resultados satisfatórios com relação à mudanças do leiaute para minimizar os efeitos dos elementos parasitas no leiaute. Estes resultados serão mostrados através de um *heat map* que demonstrará as diferenças entre o leiaute modificado e o original.

Palavras-chave: testes de circuitos integrados, modelos de falhas clássicos, modelos de falhas na perspectiva do leiaute, *heat map*.

Abstract

Classic fault models which represent physical faults on Integrated Circuits (IC) do not fulfill completely current requirements for tests. Therefore, studies are being made about fault models that are based on the IC's layout under test, instead of basing on the IC's logical diagram, which is the technique used on classic models. Models based on layouts make possible to verify the most probable areas to happen faults in consequence of a physical defect. In this dissertation, a study is made about the main classic fault models and the models based on the perspective of the layout where some concepts are defined, like: physical faults, faults, parasite elements, critical area, fault level and its correlations. After that, a fault modeling is proposed in the layout perspective and a method of extracting parasite elements of the layout is presented using CADENCE along simulational analysis and obtained results. By the end, it will be showed how parasite elements of the layout are related to the probability of fault occurrences due to the layout's project. Satisfactory results were also obtained with respect to the layout changes to minimize the effects of the parasitic elements in the layout. These results show through a heat map that will demonstrate how the differences between the modified and original layout.

Keywords: testing integrated circuits, classical fault models, fault models on layout perspective, heat map.

Lista de ilustrações

Figura 1.1 – Lei de Moore: A densidade de transistores dos circuitos integrados dobra a cada dois anos.	14
Figura 1.2 – Falha <i>Stuck-At</i>	15
Figura 1.3 – Falha <i>Transition Delay</i> (GARCIA, 2001).	16
Figura 1.4 – Defeitos extraídos de um leiaute.	18
Figura 1.5 – Área Crítica.	18
Figura 2.1 – Diferença entre falha e defeito	22
Figura 2.2 – Circuito com falhas, C1 <i>stuck-at-1</i> e C2 <i>stuck-at-0</i> (WU; LEE, 2014). .	23
Figura 2.3 – Se Caminho de propagação 1 > Caminho de propagação 2 , o efeito da falha na NAND da saída seria 0- 1 -0.	24
Figura 3.1 – Etapas do <i>Cell-Aware</i>	25
Figura 3.2 – Leiaute da célula de um multiplexador.	26
Figura 3.3 – Leiaute da célula do multiplexador MUX31X4.	27
Figura 3.4 – Transistores extraídos da netlist.	29
Figura 3.5 – Ambiente de simulação analógica.	29
Figura 3.6 – Processo do ATPG normal.	30
Figura 3.7 – TPG para uma falha em ponte interna à célula.	30
Figura 3.8 – Cobertura de defeitos do CAT em comparação com <i>Stuck-At</i> e <i>Transition Delay</i>	32
Figura 4.1 – Função da Área Crítica (BUBEL et al., 1995a).	33
Figura 4.2 – Método de expansão de polígonos para defeitos convexos arbitrários (ALLAN; WALTON, 1997a).	34
Figura 4.3 – Dinâmica do algoritmo que computa a Área Crítica (BUBEL et al., 1995a).	35
Figura 4.4 – Material Extra (ALLAN; WALTON, 1998).	35
Figura 4.5 – Material Perdido (ALLAN; WALTON, 1998).	36
Figura 5.1 – Distribuição de probabilidade para o tamanho do defeito	38
Figura 5.2 – <i>Wafer</i> contendo 648 chips.	39
Figura 5.3 – Área Crítica para um defeito de diâmetro x	40
Figura 6.1 – Fabricação CMOS: Primeiras etapas.	45
Figura 6.2 – Fabricação CMOS: Criação do <i>n-well</i> e Crescimento de óxido de campo. .	46
Figura 6.3 – Fabricação CMOS: Máscaras.	46
Figura 6.4 – Camadas de metal e vias (ou contatos).	47
Figura 6.5 – Camadas de metal e vias (ou contatos).	48
Figura 6.6 – Visão lateral das camadas de metais.	48
Figura 6.7 – Visão lateral das camadas de metais.	49

Figura 6.8 – Capacitâncias Parasitas (visão lateral).	50
Figura 6.9 – Modos de Extração de Capacitâncias no Assura RCX.	51
Figura 6.10–Diferença entre falha e defeito em nível de leiaute.	51
Figura 6.11–Distâncias entre nets <i>versus</i> Probabilidade de defeitos devido às capaci- tâncias parasitas.	52
Figura 7.1 – Fluxograma da Extração dos Elementos Parasitas.	54
Figura 7.2 – Porta lógica da célula <i>NOR4_A_5V</i>	54
Figura 7.3 – Leiaute da célula <i>NOR4_A_5V</i>	55
Figura 7.4 – Análise DRC.	55
Figura 7.5 – Análise LVS (CDL).	56
Figura 7.6 – QRC Assura <i>Setup</i>	57
Figura 7.7 – QRC Assura <i>Extraction</i>	57
Figura 7.8 – Extração de elementos parasitas da célula <i>NOR4_A_5V</i>	58
Figura 7.9 – Elementos parasitas: capacitâncias	58
Figura 7.10–Zoom nas capacitâncias entre as nets A e B.	59
Figura 7.11–Visão do topo das nets selecionadas.	59
Figura 7.12–Visão frontal das nets selecionadas.	60
Figura 7.13–Visão lateral das nets selecionadas.	61
Figura 8.1 – Programa no MATLAB para <i>Heat Map</i> de um leiaute.	63
Figura 8.2 – Leiaute com elementos parasitas e localizações.	64
Figura 8.3 – Leiaute da Célula <i>NOR4_A_5V</i> Original.	64
Figura 8.4 – Leiaute da Célula <i>NOR4_A_5V</i> Modificado.	65
Figura 8.5 – <i>Heat Map</i> com Célula <i>NOR4_A_5V</i> Original.	66
Figura 8.6 – <i>Heat Map</i> com Leiaute da Célula <i>NOR4_A_5V</i> Modificado.	66
Figura 9.1 – Multiplexador usado como exemplo dado pelo Hapke e implementado no CADENCE com defeito d1	68
Figura 9.2 – 32 possíveis entradas em $160\mu s$	68
Figura 9.3 – Resultado da simulação analógica com defeito d1 (S0N, gnd).	69
Figura A.1 – Resultado da simulação analógica com o defeito d2	76
Figura A.2 – Resultado da simulação analógica com o defeito d3	76
Figura A.3 – Resultado da simulação analógica com o defeito d4	77
Figura A.4 – Resultado da simulação analógica com o defeito d5	77
Figura A.5 – Resultado da simulação analógica com o defeito d6	78
Figura A.6 – Resultado da simulação analógica com o defeito d7	78
Figura A.7 – Resultado da simulação analógica com o defeito d8	79
Figura A.8 – Resultado da simulação analógica com o defeito d9	79
Figura A.9 – Resultado da simulação analógica com o defeito d10	80
Figura A.10–Resultado da simulação analógica com o defeito d11	80
Figura A.11–Resultado da simulação analógica com o defeito d12	81

Figura A.12–Resultado da simulação analógica com o defeito d13	81
Figura A.13–Resultado da simulação analógica com o defeito d14	82
Figura A.14–Resultado da simulação analógica com o defeito d15	82
Figura A.15–Resultado da simulação analógica com o defeito d16	83
Figura A.16–Resultado da simulação analógica com o defeito d17	83
Figura A.17–Resultado da simulação analógica com o defeito d18	84
Figura A.18–Resultado da simulação analógica com o defeito d19	85
Figura A.19–Resultado da simulação analógica com o defeito d20	85
Figura A.20–Resultado da simulação analógica com o defeito d21	86
Figura A.21–Resultado da simulação analógica com o defeito d22	87
Figura A.22–Resultado da simulação analógica com o defeito d23	87
Figura A.23–Resultado da simulação analógica com o defeito d24	88
Figura A.24–Resultado da simulação analógica com o defeito d25	88
Figura A.25–Resultado da simulação analógica com o defeito d26	89
Figura A.26–Resultado da simulação analógica com o defeito d27	89
Figura A.27–Resultado da simulação analógica com o defeito d28	90
Figura A.28–Resultado da simulação analógica com o defeito d29	90
Figura A.29–Resultado da simulação analógica com o defeito d30	91
Figura A.30–Resultado da simulação analógica com o defeito d31	91
Figura A.31–Resultado da simulação analógica com o defeito d32	92
Figura A.32–Resultado da simulação analógica com o defeito d33	92
Figura A.33–Resultado da simulação analógica com o defeito d34	93
Figura A.34–Resultado da simulação analógica com o defeito d35	93
Figura A.35–Resultado da simulação analógica com o defeito d36	94
Figura A.36–Resultado da simulação analógica com o defeito d37	94
Figura A.37–Resultado da simulação analógica com o defeito d38	95
Figura A.38–Resultado da simulação analógica com o defeito d39	95
Figura A.39–Resultado da simulação analógica com o defeito d40	96
Figura A.40–Resultado da simulação analógica com o defeito d41	96
Figura A.41–Resultado da simulação analógica com o defeito d42	97
Figura A.42–Resultado da simulação analógica com o defeito d43	97
Figura A.43–Resultado da simulação analógica com o defeito d44	98
Figura A.44–Resultado da simulação analógica com o defeito d45	98
Figura A.45–Resultado da simulação analógica com o defeito d46	99
Figura A.46–Resultado da simulação analógica com o defeito d47	99
Figura A.47–Resultado da simulação analógica com o defeito d48	100

Lista de tabelas

Tabela 3.1 – Lista de defeitos	27
Tabela 3.2 – Matriz de Defeitos	28
Tabela 9.1 – Matriz de Defeitos obtidas por Simulação no CADENCE	70

Lista de abreviaturas e siglas

ATE	<i>Automatic Test Equipment</i>
ATPG	<i>Automatic Test Pattern Generation</i>
BIST	<i>Built-In Self Test</i>
CAM	<i>Cell-Aware Model</i>
CAT	<i>Cell-Aware Test</i>
CDL	<i>Compiler Description Language</i>
CI	Circuito Integrado
DFT	<i>Design-for-Testability</i>
DRC	<i>Design Rule Check</i>
ERC	Electrical Rule Check
IBM	<i>International Business Machine</i>
LVS	<i>Layout Versus Schematic</i>
SPICE	<i>Simulated Program with Integrated Circuits Emphasis</i>
TPG	<i>Test Pattern Generation</i>

Sumário

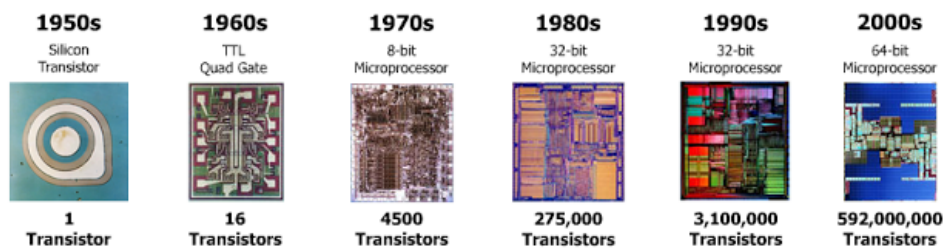
1	Introdução	14
1.1	Objetivos Gerais e Específicos	19
1.2	Materiais e Métodos	20
I	Referenciais teóricos	21
2	Modelos de Falhas	22
2.1	Modelo de falhas <i>Stuck-at</i>	22
2.2	Modelo de falhas <i>Transition Delay</i>	23
3	<i>Cell-Aware Test</i>	25
4	Área Crítica	33
5	Cálculo de Área Crítica baseado no Leiaute da Célula	37
5.1	Função Densidade de Probabilidade do Defeito	38
5.2	Área Crítica	39
5.3	Contribuição de defeitos em interconexões paralelas	40
6	Processo de Fabricação CMOS	44
6.1	Processo de Fotolitografia, Criação de <i>n-well</i> e Crescimento do FOX	44
6.2	Processo de Metalização	46
6.3	Tipos de Capacitâncias	49
6.4	Elementos Parasitas como Modelos de Falhas	51
II	Proposta	53
7	Extração de Elementos Parasitas no CADENCE	54
8	Aplicação da Proposta do <i>Heat Map</i> em Células-Padrão da Biblioteca	62
9	Implicações da Área Crítica em <i>Cell-Aware Test</i>	67
	Conclusão	71
	Referências	72
	Apêndices	75
	APÊNDICE A – Resultados de Simulação	76

1 Introdução

Nos dias atuais, os sistemas eletrônicos estão cada vez mais presentes no nosso cotidiano. Nesse contexto, há um princípio fundamental: quanto maior é o benefício que esses sistemas oferecem ao nosso bem-estar, maior é o potencial de causarem algum dano quando falham ou não funcionam da maneira correta (SOUZA, 2005). Sistemas eletrônicos que são componentes de aviões, de aparelhos utilizados na medicina e de dispositivos eletrônicos de carros, são alguns exemplos em que a certeza de que esses sistemas funcionem corretamente é imprescindível. Quando esses sistemas não funcionam da forma esperada, danos graves podem ser causados e vidas podem ser perdidas (SOUZA, 2005). Então, para tais aplicações, é necessário o desenvolvimento de sistemas que tenham alta confiabilidade (SOUZA, 2005; CLIFFS, 1994).

A confiabilidade dos sistemas eletrônicos se dá com o aumento da confiabilidade dos componentes desses sistemas, que no exemplo citado são os **circuitos integrados** (CIs). Tal afirmativa sobre a confiabilidade baseia-se no princípio básico das técnicas de tolerância à falhas que explicam que o sistema como um todo depende da confiabilidade dos componentes que o constituem (CLIFFS, 1994).

Figura 1.1 – Lei de Moore: A densidade de transistores dos circuitos integrados dobra a cada dois anos.



Fonte: Roberto et al. (2009).

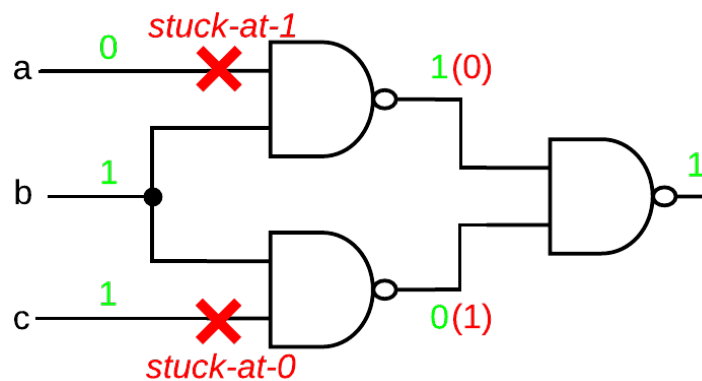
Os circuitos integrados, a cada ano que passa, estão cada vez com áreas menores, com um número maior de transistores e mais integrados, como pode ser visto na Fig. 1.1. Uma das etapas de grande importância na fabricação de CIs são os testes que visam assegurar que o número de circuitos defeituosos que passem para os próximos níveis de manufatura do produto seja mínimo (SOUZA, 2005). Esses testes são necessários para verificar se o circuito integrado não está falho. Normalmente, testes de CIs são realizados utilizando **Equipamentos de Teste Automático** (ATE). Esses equipamentos são usados na aplicação de vetores de testes no circuito e no recebimento de suas respostas, e o objetivo final é verificar se o circuito está falho ou não, comparando a resposta do teste com a

esperada (SOUZA, 2005).

Para realizar os testes é necessário incluir, no projeto do CI, componentes tais como *flip-flops* e chaves que mudam o estado do CI para o estado de teste (HURST, 1998). A essa aplicação de componentes e técnicas de testes deu-se o nome de **Design For Testability** (DFT). A DFT é o desenvolvimento, projeto, aplicação de componentes e técnicas que serão aplicadas no CI (PLUSQUELLIC, 2006a).

Para a proposição de métodos de testes em CIs é necessário utilizar modelos de falhas, que podem ser definidos como representações dos possíveis defeitos físicos presentes no CI. Por exemplo, para modelar um defeito em que a entrada da porta lógica está presa em nível lógico 1, como por exemplo quando há uma ligação fixa entre a entrada e VDD (tensão de alimentação), dizemos que pode ocorrer uma falha **Stuck-At**. E, como é presa em nível lógico 1, dá-se o nome de **stuck-at-1** para essa falha. O mesmo modelo de falhas pode modelar vários tipos de defeitos físicos. Na Figura 1.2 observa-se que há uma falha *stuck-at-1* na entrada *a* e uma falha *stuck-at-0* na entrada *b*, o que significa que, por algum defeito físico, o valor lógico dessas entradas se manteve constante. Neste exemplo pode-se ver que as falhas nas entradas não ocasionaram um **erro** na saída. Então, isso significa que nem toda falha causa um erro no circuito integrado (PLUSQUELLIC, 2006b). No modelo *stuck-at*, curto-circuitos internos à célula como em transistores em curto ou aberto, são modelados como falhas nas entradas e saídas das portas lógicas. O modelo *stuck-at* é amplamente utilizado atualmente e cobre uma alta porcentagem dos defeitos físicos passíveis de ocorrer em um circuito integrado (PLUSQUELLIC, 2006b).

Figura 1.2 – Falha *Stuck-At*.

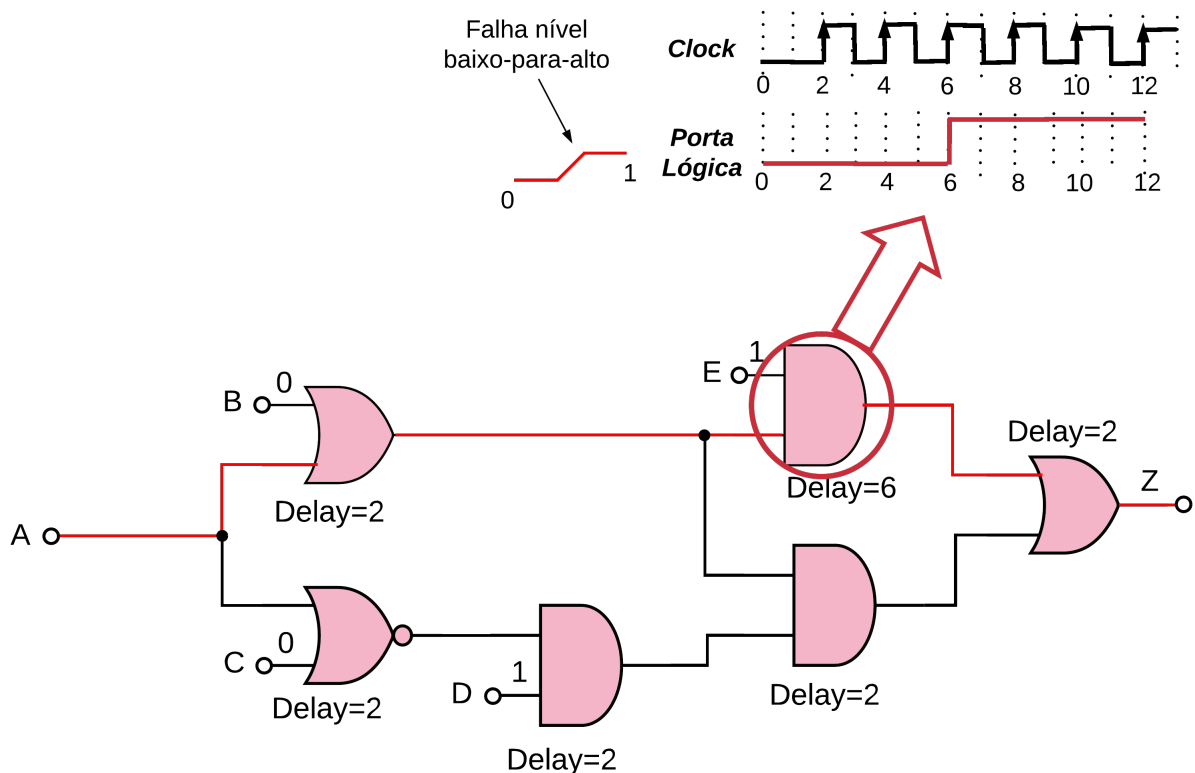


Fonte: Elaborada pela autora.

O modelo de falhas *stuck-at* é o mais conhecido e utilizado atualmente (PLUSQUELLIC, 2006b). O segundo modelo mais utilizado é o **Transition Delay** (COX; RAJSKI, 1988) que modela os defeitos que causam atraso na mudança de estado em portas lógicas. CIs que operam em alta frequência são mais afetados por falhas *transition delay*. Isso acontece devido à rapidez do chaveamento dos estados lógicos em cada porta lógica.

À medida que essas transições ocorrem em um curto espaço de tempo, há a necessidade de saber se estão ocorrendo de forma correta. Assim como nas falhas *stuck-at*, as falhas são modeladas nas entradas e saídas das portas lógicas. Existem dois tipos de falhas *transition delay*: a **falha nível baixo-para-alto**, que é quando o atraso é na transição de nível baixo para o nível alto e a **falha nível alto-para-baixo** em que atraso acontece na transição de nível alto para o nível baixo (WAICUKAUSKI et al., 1987).

Figura 1.3 – Falha *Transition Delay* (GARCIA, 2001).



Fonte: Elaborada pela autora.

Na Figura 1.3 pode ser visto um exemplo de falha *transition delay*, em que a porta lógica em destaque está com um atraso maior do que o *clock* do circuito. Em consequência disso, a porta lógica que tem a falha demorará três pulsos de *clock* para responder à mudança de nível lógico. Esse atraso é dado na mudança de nível lógico de 0 para 1, ou seja, falha nível baixo-para-alto.

Tanto o modelo de falhas *Stuck-At* quanto o modelo *Transition Delay* representam defeitos físicos que podem ocorrer nos circuitos integrados. Os fabricantes precisam de modelos altamente eficientes para que um maior número de defeitos possam ser detectados. Com o tempo, os clientes (fabricantes de produtos que tem como componentes os circuitos integrados) começaram a reportar que recebiam CIs defeituosos de seus fabricantes, ou

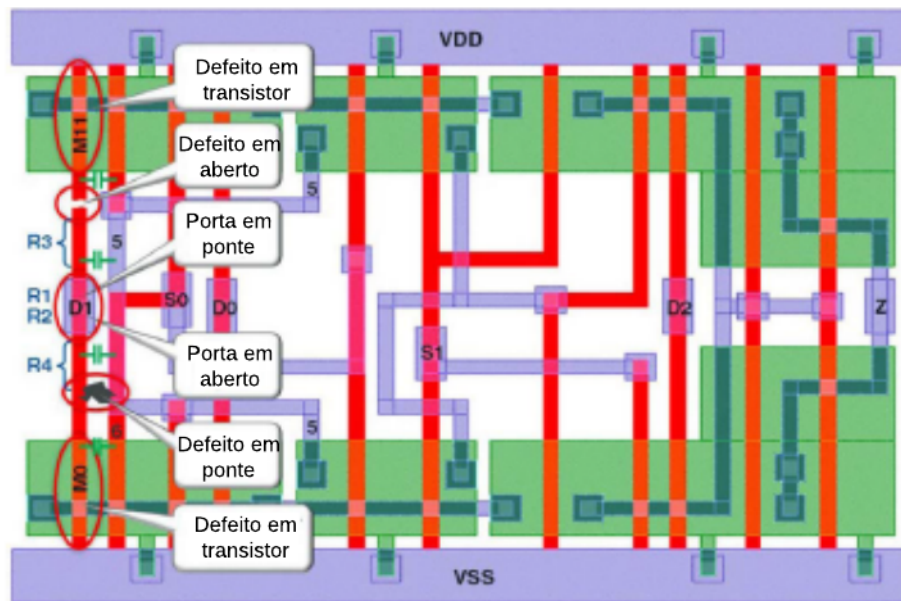
seja, esses componentes passavam nos testes, mas na verdade estavam com problema. Ao investigar o problema, descobriu-se que a maioria dos defeitos que passavam despercebidos aos testes ocorriam dentro da biblioteca de células padrão, que são conjuntos de transistores e interconexões que formam funções booleanas tais como AND, OR, NAND, XOR (HAPKE et al., 2014). Muitos desses defeitos que, são internos à célula, permanecem não-detectados quando se usam ferramentas tradicionais de **Automatic Test Pattern Generator** (ATPG).

Além do problema de componentes defeituosos que passam nos testes, vale salientar que os modelos de falhas clássicos abordam o problema de defeitos apenas parcialmente (HAPKE et al., 2014). Com essas desvantagens apresentadas pelos modelos de falhas clássicos, alternativas estão sendo estudadas nos últimos anos para resolver estes problemas. Sendo a mais relevante a que trata do modelamento de falhas e simulações analógicas baseadas em *netlist* SPICE (*Simulated Program with Integrated Circuits Emphasis*) contendo apenas transistores e não os objetos parasitas (capacitores, resistores), mas ainda assim considerando o leiaute físico (HAPKE et al., 2014). Com isso, Hapke et al. (2014) juntamente com a Mentor Graphics[©] definiram o que é o **Cell-Aware Test** ou **CAT**, que são modelos de falhas na perspectiva do leiaute (HAPKE et al., 2009; HAPKE et al., 2010; HAPKE et al., 2014).

Nas pesquisas em torno do CAT é mostrada a necessidade atual de que os modelos de falhas sejam baseados no leiaute do circuito, e não apenas em modelos generalizados. Foram criados modelos de falhas específicos, baseado nos defeitos físicos que possam ocorrer, como pode ser visto na Figura 1.4. Os modelos de falhas do CAT se baseiam nas resistências e capacitâncias parasitas (que não são as componentes do circuito, e sim as que surgem pelas distâncias das interconexões), além das entradas e saídas dos circuitos integrados para criar os modelos (HAPKE et al., 2009; HAPKE et al., 2010; HAPKE et al., 2011; HAPKE et al., 2014).

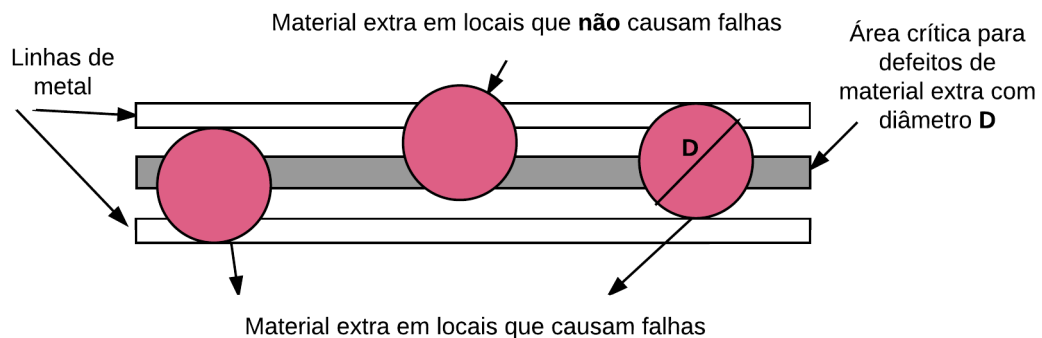
Estudos baseados no leiaute já haviam sido feitos em anos anteriores (JACOMET; GUGGENBUHL, 1993; BUBEL et al., 1995a; ALLAN; WALTON, 1997a; ALLAN; WALTON, 1998; ALLAN; WALTON, 1999a; Chao-Wen Tzeng; Han-Chia Cheng; Shi-Yu Huang, 2009). Esses estudos eram baseados no que se chamou de **Área Crítica**. Na Figura 1.5 pode-se observar que área crítica é a área entre as interconexões em que é mais provável de acontecer uma falha dado um defeito físico. Se tal defeito não conectar as interconexões, ele não é considerado uma falha. Esse conceito foi motivado pela necessidade de criar modelos que fossem baseados no leiaute dos circuitos integrados (ALLAN; WALTON, 1998). Dessa forma criou-se uma maneira de calcular a área crítica de acordo com o tamanho dos defeitos físicos que possam ocorrer. Dados os conceitos definidos pelo *Cell-Aware Test* (HAPKE et al., 2014) e de Área Crítica (JACOMET; GUGGENBUHL, 1993), ambos

Figura 1.4 – Defeitos extraídos de um leiaute.



Fonte: Adaptada de (HAPKE et al., 2014).

Figura 1.5 – Área Crítica.



Fonte: Elaborada pela autora.

baseados em leiaute, buscou-se com este trabalho relacionar os dois conceitos para que se possa comprovar que eles se complementam.

Ao relacionar os conceitos da área crítica com os cálculos necessários para criar os modelos de falhas CAT (HAPKE et al., 2014), viu-se que é possível uma análise visual das regiões com maior suscetibilidade à falhas através do *heat map*. Dada essa análise, a proposta do trabalho é de uma nova forma de avaliação na perspectiva do leiaute que utiliza *heat map* como forma de observação e estudos de regiões propícias à falhas. Nesse tipo de avaliação do leiaute, elementos parasitas são usados como parâmetros para identificar em que regiões defeitos físicos tem maior probabilidade de se tornarem falhas. O *heat map* é a representação gráfica de uma matriz de valores, sendo assim, uma matriz de valores de elementos parasitas foi usada como dados de um *heat map* para obtenção de regiões com

maior suscetibilidade à falhas. Essa proposta foi realizada usando ferramentas CADENCE, para obtenção do leiaute de um determinado circuito lógico e MATLAB, para gerar o *heat map*.

Neste trabalho será dissertado o conceito de modelo de falhas no Capítulo 1. No Capítulo 2 serão analisados os conceitos do *Cell-Aware Test* (CAT), em seguida, no Capítulo 3 será analisada Área Crítica. No Capítulo 4 serão relacionados os dois conceitos e serão vistas as implicações da Área Crítica em *Cell-Aware Test*. No Capítulo 5 será analisado o Processo de Fabricação CMOS, Metalização e Tipos de Capacitâncias Parasitas. Na sequência, no Capítulo 6 serão analisados do Elementos Parasitas como Modelos de Falhas, e no Capítulo 7 como é feita essa Extração de Elementos Parasitas no CADENCE. Para em seguida, no Capítulo 8 ser mostrada a Aplicação da Proposta *Heat Map* em Células-Padrão da Biblioteca. Para em seguida, ser retomada as Implicações de Área Crítica em *Cell-Aware Test*. Na Conclusão será dissertado sobre os trabalhos feitos com o CADENCE, tanto em simulações analógicas quanto em extração de elementos parasitas, que irão comprovar os conceitos apresentados neste trabalho.

1.1 Objetivos Gerais e Específicos

O objetivo geral do trabalho é fazer uma análise de defeitos na perspectiva do leiaute em células de bibliotecas padrão. Ao analisar trabalhos que vem sendo feitos no estado-da-arte observou-se uma crescente aplicação e estudos de modelos de falhas baseados no leiaute. As pesquisas atuais utilizam ao invés de portas lógicas, transistores e elementos parasitas para se basear nos modelos falhas (HAPKE et al., 2009; HAPKE; SCHLOEFFEL, 2012; HAPKE et al., 2014). Nesse contexto, neste trabalho foram feitas simulações analógicas com a inclusão de falhas extraídas de um multiplexador exemplo do artigo do Hapke para observar como as saídas seriam afetadas pelas falhas, para em seguida documentar os resultados na chamada Matriz de Defeitos, que a posteriore é utilizada para obter padrões de testes. Após essas análises, foram feitos estudos dos conceitos de área crítica, densidade de defeitos, função densidade de probabilidade de defeitos e cálculo dos defeitos, para compreender como essas falhas são modeladas. Esses conceitos são de grande importância, pois com eles conseguiu-se desenvolver neste trabalho um fluxograma de análise de defeitos próprio. O fluxograma proposto começa com a extração dos elementos parasitas do leiaute de uma célula-padrão, para em seguida obter o arquivo de capacitâncias parasitas que será utilizado para fazer um *heat map* das regiões em que serão mais prováveis de que um defeito se torne uma falha. A proposta é de que a análise na perspectiva do leiaute seja baseada em *heat maps* com seus valores dados por capacitâncias parasitas e não mais pelos modelos de falhas matemáticos.

Os objetivos específicos do trabalho são:

- Realizar simulações analógicas com modelos de falhas na perspectiva do leiaute;
- Analisar como extrair elementos parasitas do leiaute;
- Associar os valores de capacitâncias à níveis de defeitos;
- Proposta de análise do leiaute a partir do *heat map*; e
- Proposta do *heat map* realizado no CADENCE e MATLAB.

1.2 Materiais e Métodos

A metodologia utilizada para simulação de modelos de falhas na perspectiva do leiaute, foi baseada em realizações de simulações analógicas na ferramenta spectre do CADENCE para observar as respostas de um multiplexador com falhas em ponte pré-definidas (HAPKE et al., 2009). A metodologia para o desenvolvimento do fluxograma de extração de *heat map*, proposto neste trabalho para análise na perspectiva do leiaute, foi a extração de elementos parasitas para a avaliação das regiões de maior probabilidade de falhas no leiaute. Essa extração foi feita com o Assura RCX, que extrai elementos parasitas como resistores, capacitores e indutâncias. Após a extração foi utilizado o MATLAB com os valores de capacitâncias e localizações (x,y) para a obtenção do *heat map* do dado leiaute.

A obtenção do *heat map* se inicia ao exportar a imagem do leiaute do CADENCE para o MATLAB. Com a imagem do leiaute no MATLAB é possível avaliar em que localizações estão contidos os elementos parasitas (capacitâncias). Para obter as regiões de suscetibilidade à falhas, os cálculos são feitos a partir da matriz bidimensional dos elementos parasitas, que dará como resultado, com as dadas funções do MATLAB, o *heat map*.

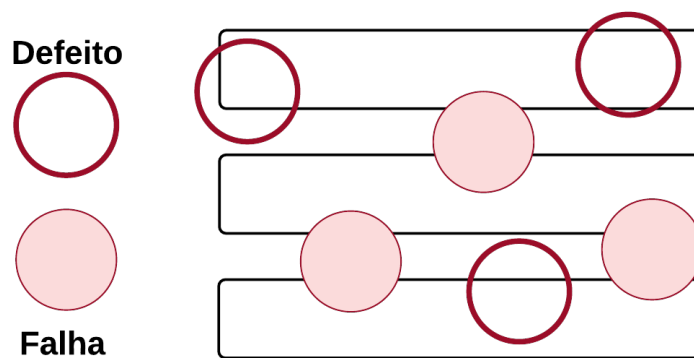
Parte I

Referenciais teóricos

2 Modelos de Falhas

Antes de descrever sobre modelos de falhas, é necessário definir alguns conceitos essenciais para se tratar de testes e que serão utilizados ao longo de todo o trabalho. Um desses conceitos são os defeitos físicos, que podem ser causados por material extra no processo de fabricação, ou seja, é o que pode ocorrer fisicamente no CI (ALLAN, 1998). Falha é a possível consequência do defeito físico, se houver material extra que cause um curto entre as interconexões, por exemplo, haverá uma falha. Esse conceito, da diferença entre falha e defeito é demonstrado na Figura 2.1.

Figura 2.1 – Diferença entre **falha** e **defeito**.

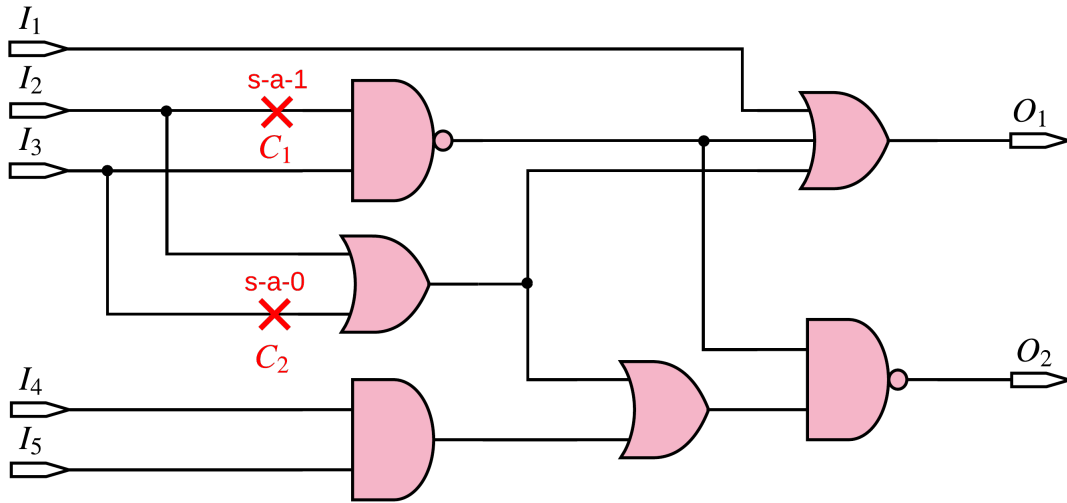


Fonte: Elaborada pela autora.

Uma etapa primordial no processo de testes é a etapa da aplicação dos modelos de falhas. Modelos de falhas são representações de defeitos físicos que podem ocorrer em circuitos integrados (ALLAN, 1998). Defeitos como curto-circuitos, abertos e em ponte são os mais comuns de ocorrerem (HAPKE et al., 2014). Para isso, modelos de falhas vem sendo estudados ao longo dos anos para que a maior parte desses possíveis defeitos físicos possa ser modelada.

2.1 Modelo de falhas *Stuck-at*

Um dos primeiros modelos de falhas e mais conhecido na área de testes foi criado em 1963 é o modelo *Stuck-at* (MEI, 1974). Esse modelo é amplamente utilizado e modelam diferentes tipos de defeitos, como entradas/saídas fixas em nível lógico (*stuck-at-1*) e entradas/saídas fixas em nível lógico (*stuck-at-0*). Pode-se observar pela Figura 2.2 que esses nomes podem ser abreviados como *s-a-1* e *s-a-0*.

Figura 2.2 – Circuito com falhas, C1 *stuck-at-1* e C2 *stuck-at-0* (WU; LEE, 2014).

Fonte: Elaborada pela autora.

O modelo de falhas *stuck-at* é muito utilizado na indústria (HAPKE et al., 2010). As empresas responsáveis pela fabricação dos circuitos integrados da indústria automotiva, como exemplo Freescale Semicondutores, tem por objetivo entregar chips com 99% de cobertura de falhas *stuck-at* para os fabricantes automotivos. Isso significa dizer que os testes projetados para o circuito integrado cobrem 99% das falhas *stuck-at* que possam ocorrer. Apesar de amplamente utilizado atualmente, esse modelo de falhas vem apresentando uma dificuldade em atingir a cobertura que é requisitada. Fato esse que é dado por um considerável número de defeitos físicos que não são modelados pelas falhas *stuck-at*. Para isso, são usados outros modelos de falhas que considerem esses defeitos físicos.

2.2 Modelo de falhas *Transition Delay*

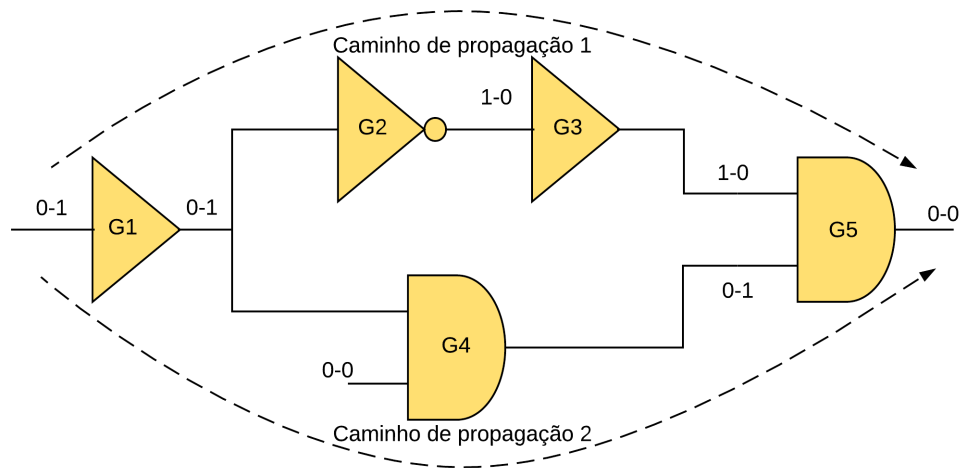
Defeitos que causam falhas mas que não são relacionados com tempo de mudança de estado na porta lógica são chamados **defeitos estáticos**. Já defeitos que causam falhas e são relacionados com as mudança de estado da porta lógica, são os chamados **defeitos de delay** (CHENG et al., 2008). Para detectar defeitos de *delay*, padrões específicos são utilizados.

Para inserir os padrões específicos para falhas *transition delay* é necessário fazer transições de níveis lógicos a cada pulso de *clock* para verificar se há falhas (CHENG et al., 2008). Como são observadas as transições de níveis lógico, o modelo de falha *transition delay* requer pelo menos dois pulsos de *clock* para serem detectadas. Para testar a falha é necessário colocar valor de transição nesses dois ciclos, mudar de 0 – 1 (borda de subida)

ou 1 – 0 (borda de descida) e verificar se há mudança de estados de um ciclo para o outro. Essa é a definição para os testes do modelos de falhas *transition delay*.

Na Figura 2.3 pode-se observar como exemplo de modelo de falhas *transition delay* um circuito com cinco portas lógicas. O efeito da transição (0 – 1) é colocado na entrada da primeira porta lógica podendo se propagar através de caminhos múltiplos e, finalmente convergir para algum ponto (CHENG et al., 2008). No caso do exemplo, o efeito da transição na entrada da porta G1 pode propagar-se para G5 em dois diferentes caminhos. Como o caminho de propagação 1 é maior do que o caminho de propagação 2, causará um efeito de falha na saída 0-1-0.

Figura 2.3 – Se **Caminho de propagação 1** > **Caminho de propagação 2**, o efeito da falha na NAND da saída seria 0-**1**-0.



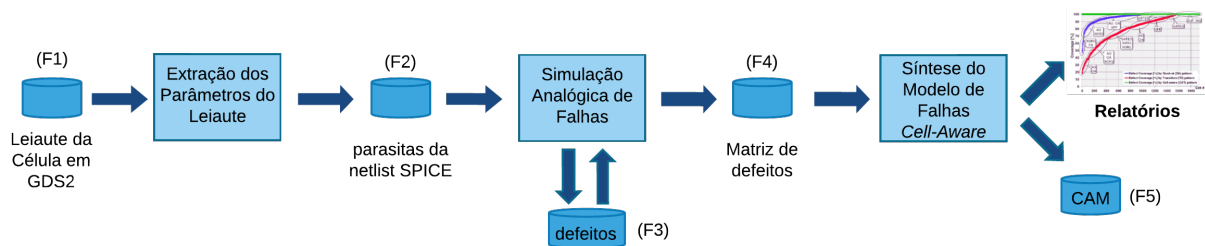
Fonte: Elaborada pela autora.

3 *Cell-Aware Test*

O *Cell-Aware Test* ou CAT é uma metodologia de testes que tem como principal metodologia modelos de falhas baseados no leiaute (HAPKE et al., 2009). O resultado é um teste na perspectiva do leiaute, ou seja, o leiaute da célula é analisado, bem como os transistores, resistências e capacitâncias parasitas são analisadas. E, a partir de todas essas informações, os modelos das falhas são descritos para esse leiaute em específico. Sabendo-se disso, pode-se ter modelos de falhas diferentes, e até mesmo um número distinto de modelos de falhas para a mesma célula. Isso acontece porque, dependendo da forma como o leiaute é projetado, os valores das capacitâncias e resistências parasitas mudam, e as falhas podem ser determinadas de acordo com esses valores (HAPKE et al., 2009).

A primeira citação do CAT foi em Hapke et al. (2009), sua contribuição foi uma nova metodologia para detectado defeitos internos à célula. Isso porque estudos feitos para analisar os defeitos, que não estavam sendo cobertos pelos modelos de falhas atuais, mostraram que os defeitos internos às células eram os que estavam passando pelos testes sem serem detectados, gerando assim, uma necessidade de pesquisas mais aprofundadas para a resolução desse problema. A nova metodologia *Cell-Aware* foi avaliada para tecnologias de 90nm e 150nm em 1671 células de biblioteca, testado em 10 *designs* industriais reais com até 50 milhões de falhas. Os resultados experimentais mostraram, em média, um crescimento de 1,2% na cobertura de defeitos (HAPKE et al., 2009).

Figura 3.1 – Etapas do *Cell-Aware*.



Fonte: Elaborada pela autora.

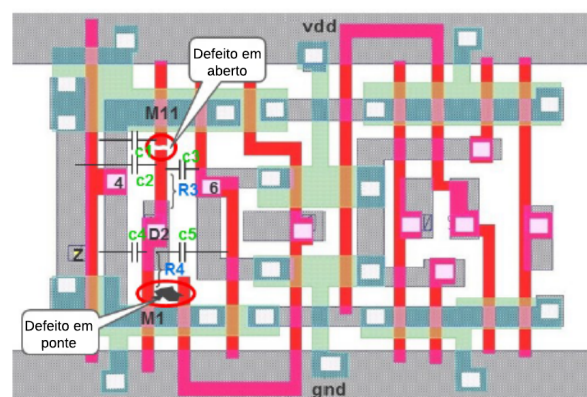
Para entender o *Cell-Aware Test* é necessário entender o fluxograma dessa metodologia. Na Fig. 3.1 é descrito o fluxograma do CAT, que começa com os dados do leiaute da célula que serão armazenados em um arquivo que contém os leiautes das bibliotecas. A esse arquivo dá-se o nome de *F1*. Em seguida, é feita a extração dos parâmetros desse leiaute, como resistências e capacitâncias parasitas. De acordo com os valores extraídos pode-se inferir quais locais são mais prováveis de haver falhas, conforme as regras do

Cell-Aware. Essas informações do leiaute são armazenadas em *F2*, os prováveis defeitos físicos são armazenados em *F3*. Com as informações das falhas, pode ser feita a ‘matriz de defeitos que ficará em *F4*. Depois da síntese do modelo de falhas, têm-se a análise de cobertura, e finalmente o CAM (*Cell-Aware Model*) em *F5* (HAPKE et al., 2014).

Para que o teste tenha uma eficiência alta é preciso que o modelo de falhas seja baseado no leiaute, isso porque quando baseia-se no leiaute os defeitos físicos são modelados com maior precisão (HAPKE et al., 2014). Para isso, a primeira etapa proposta pelo CAT é obter o leiaute da célula em que serão criados os modelos e armazená-los em um arquivo para a próxima etapa do fluxograma *Cell-Aware*.

Na Figura 3.2 são mostrados exemplos de defeitos físicos modelados em falhas. Alguns dos defeitos são: **defeito em aberto** (*Open defect*), que na simulação é representado por altas resistências; e **defeito em ponte** (*Bridge defect*), que é representado por capacitâncias. No trabalho descrito em Hapke et al. (2014) não são apresentados os valores das resistências e capacitâncias que correspondem às falhas. Esses valores foram investigados baseado em outros artigos, e serão analisados no Capítulo 4.

Figura 3.2 – Leiaute da célula de um multiplexador.



Fonte: Adaptada de Hapke e Schloeffel (2012).

Na Figura 3.3 é representado um exemplo do arquivo em $F1$, que é o leiaute de um multiplexador de três entradas ($D0, D1, D2$), dois seletores ($S0, S1$) e a saída Z . Esse será o primeiro arquivo a ser usado para o fluxo do CAT.

Depois de examinar o leiaute ($F1$) e analisar os possíveis defeitos que a célula possui, têm-se na Tabela 3.1 a lista de falhas consideradas em $F2$ para o MUX31X4.

Por exemplo, sabendo que os seletores do multiplexador são $S0$ e $S1$, as saídas desses seletores negadas serão dadas, respectivamente, por $S0N$ e $S1N$. Sabendo disso, pode-se analisar a lista de falhas possíveis: analisando **d0** pode-se dizer que, é possível ocorrer uma falha em que a saída negada do seletor $S0$, dada por $S0N$ esteja fixa em

Tabela 3.2 – Matriz de Defeitos

Input	d1	d2	d3	d4	...	d41	d42	d43	d44	d45	d46	d47	d48
00000	-	-	-	-		-	-	-	D	-	-	-	-
00001	D	D	-	-		D	D	D	-	D	D	-	-
00010	-	-	-	-		-	-	-	D	-	D	-	-
00011	-	D	-	-		D	-	D	-	D	-	-	-
00100	-	-	-	-		-	-	-	D	D	-	-	-
00101	D	-	-	-		D	D	D	-	-	D	-	-
00110	-	-	-	-		-	-	-	D	D	D	-	-
00111	-	-	-	-		D	-	D	-	-	-	-	-
01000	-	-	-	-		-	-	-	D	-	-	-	D
...													
11111	-	-	-	-		D	-	-	-	-	-	-	-

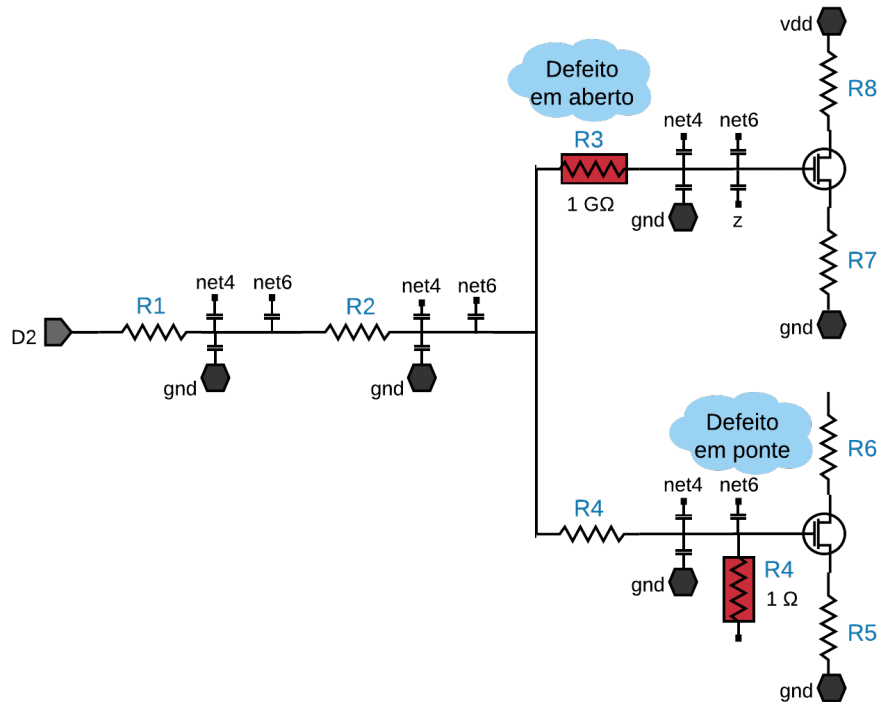
uma das 49 *netlists* são simuladas, no entanto, como cada uma possui 5 bits de entrada, serão necessárias 32 simulações para cada *netlist*, totalizando 1568 simulações. As saídas das simulações são analisadas e comparadas com a saída *golden*. Se a saída para uma dada entrada for igual à resposta *golden* significa que a entrada não-detectou o defeito. Se a resposta for diferente à *golden* significa que a entrada expôs a falha, ou seja, que ela detectou a falha.

Durante a análise de todas as simulações, pode-se construir o que é denominado de **Matriz de Defeitos** (Tabela 3.2). A Matriz de Defeitos da célula é uma tabela que contém todas as 32 entradas possíveis da célula representadas nas **linhas** da matriz e, os 48 defeitos que são representados nas **colunas** na matriz. Por exemplo, na linha que tem a entrada '11111', essa entrada detecta (**D**), dentre outros, o defeito *d41* o qual é um curto-circuito entre a *net65* e *vdd* (HAPKE; SCHLOEFFEL, 2012).

Na Figura 3.4 é ilustrada **uma parte** dos componentes (resistores, capacitâncias, entradas, transistores, VDD e GND) do MUX31X4. Além dos componentes que são do leiaute, a *netlist* contém os possíveis defeitos. Cada um dos defeitos é colocado por vez, um em cada *netlist* para que possa ser feita análise das saídas com cada defeito por vez. Os resistores que estão em vermelho são os defeitos que não são do leiaute, que foram colocados para simular defeitos físicos. Os outros elementos são extraídos do leiaute da célula que está sendo analisada.

Na Figura 3.5 pode-se ver um exemplo de como ficam os componentes e a alimentação das entradas em uma das simulações analógicas. Os elementos extraídos da *netlist*, incluindo objetos parasitas, são usados como entrada para a simulação analógica, em que cada defeito é simulado (HAPKE; SCHLOEFFEL, 2012). É possível observar na Figura 3.5 a entrada '101' com os defeitos em aberto e em ponte. Nas simulações analógicas, um defeito é considerado detectado se pelo menos uma das entradas resulte em uma saída

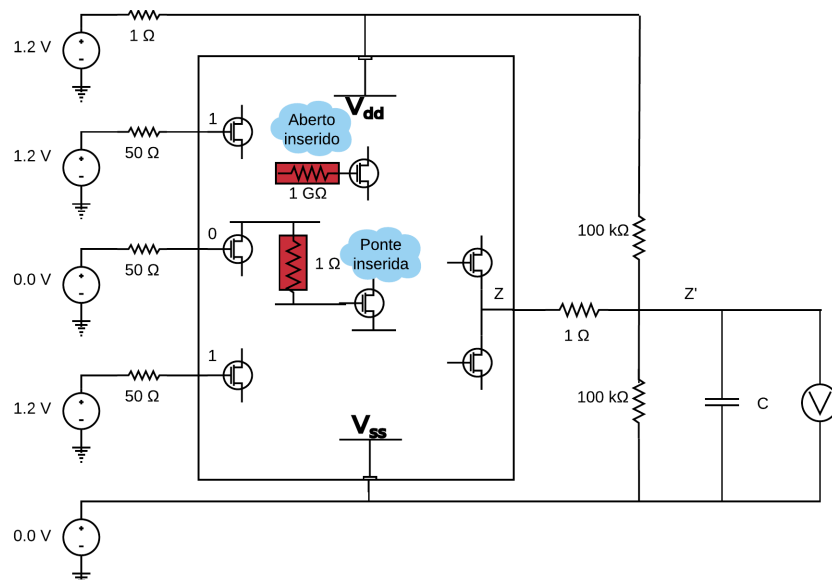
Figura 3.4 – Transistores extraídos da netlist.



Fonte: Adaptada de Hapke e Schloeffel (2012).

que desvia em mais do que 50% da sua tensão de alimentação VDD.

Figura 3.5 – Ambiente de simulação analógica.

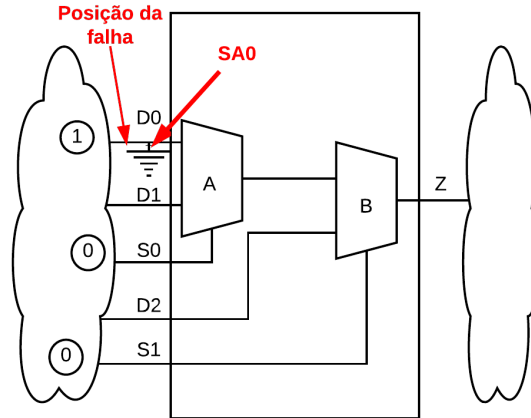


Fonte: Adaptada de Hapke e Schloeffel (2012).

O processo de inserção das falhas com o SA-ATPG para análise das saídas é diferente para o CAT-ATPG. Como pode ser visto na Figura 3.6 o modelo de falhas *stuck-at-0* é modelado na entrada *D0* da célula MUX31X4 e a condição para que essa falha

seja observada na saída é $D0 = 1$, $S0 = 1$ e $S1 = 1$. Assim, o ATPG *stuck-at* tradicional que irá gerar os padrões só irá precisar dessa entrada para essa falha em específico (HAPKE et al., 2009). Ou seja, a condição de excitação da falha interna é colocada na entrada da célula. E o ATPG não precisará inserir deterministicamente outras entradas.

Figura 3.6 – Processo do ATPG normal.

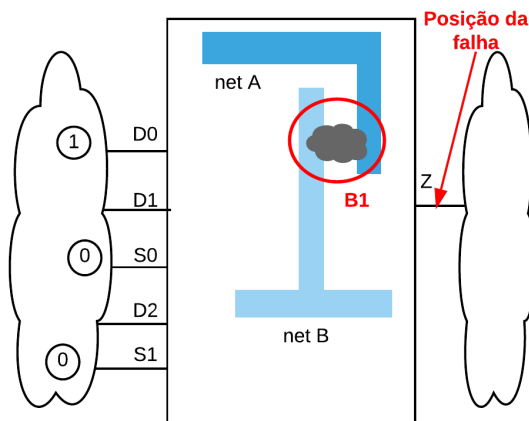


Fonte: Adaptada de Hapke et al. (2009).

O processo de gerar padrões do CAT-ATPG para o mesmo multiplexador pode ser visto na Fig. 3.7. Neste caso, assume-se uma falha em ponte entre as nets A e B como está indicado no leiaute da figura (HAPKE et al., 2009).

A posição inicial para uma falha baseada no CAT é sempre a saída da porta. A condição para a excitação da falha e a sua propagação para a saída não tem relação com as entradas pré-definidas do ATPG (HAPKE et al., 2009). O ATPG aplica estritamente as condições necessárias nas entradas da porta da célula, como definido pelo modelo CAT correspondente (HAPKE et al., 2009).

Figura 3.7 – TPG para uma falha em ponte interna à célula.



Fonte: Adaptada de Hapke et al. (2009).

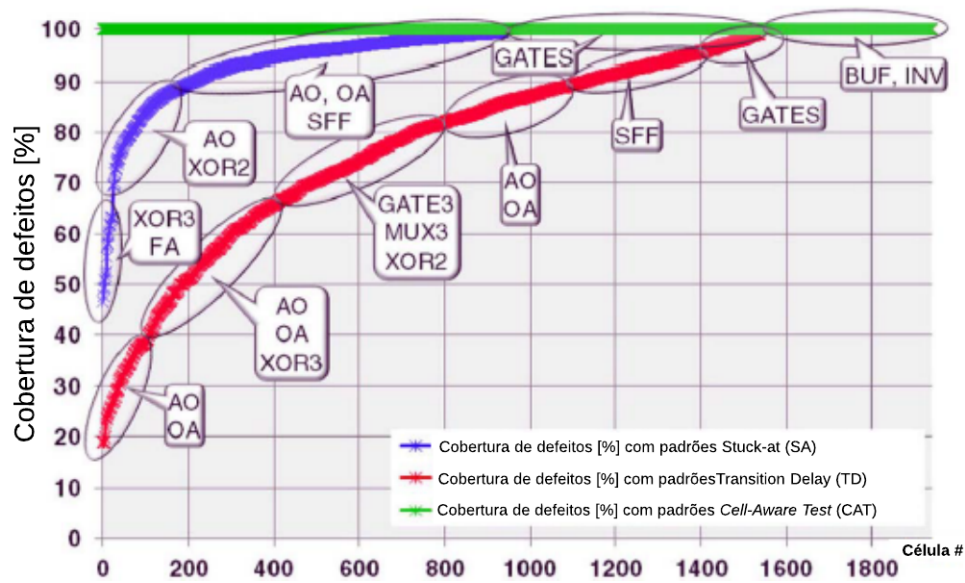
Assim, ao considerar uma falha em ponte $B1$, é necessário que as entradas sejam $D0 = 1$, $D2 = 1$, $S0 = 1$ e $S1 = 0$. Isso significa que o CAT-ATPG forçou o valor de uma entrada a mais, no caso $D2$, unicamente com o objetivo de detectar uma possível falha em ponte $B1$ (HAPKE et al., 2009). Ou seja, o que diferencia o CAT-ATPG do ATPG tradicional é a aplicação do padrão de forma determinística com a finalidade de detectar falhas internas à célula.

Depois de finalizada todas as simulações analógicas e de obter a matriz de defeitos, é possível fazer uma comparação dos resultados obtidos pelo CAT com os modelos de falhas *Stuck-At* e *Transition Delay* (HAPKE et al., 2014).

Na Figura 3.8 é apresentado o gráfico com relação entre as células e a cobertura de defeitos. O eixo horizontal representa as células da biblioteca, sendo numeradas de 1 a 1940. O eixo vertical indica a porcentagem de cobertura de defeitos da seguinte forma: a curva em azul é a cobertura de defeitos para as pontes, abertos e defeitos em transistores que são detectáveis, e que é conseguido usando apenas os padrões dos modelos de falhas *Stuck-At*; a curva em vermelho é a cobertura para os mesmos defeitos que são detectáveis, agora usando apenas os padrões dos modelos de falhas *Transition Delay*; a curva em verde é a cobertura de defeitos em porcentagem quando são utilizados padrões baseados no CAT.

No gráfico da Figura 3.8 é mostrado que, para os padrões baseados em modelos *Stuck-At*, a cobertura de defeitos é menor do que 100% para cerca de 50% das células, e em algumas células a cobertura gira em torno de 46%. Mas a cobertura é menor quando se trata de padrões *Transition Delay*, em que aproximadamente 80% das células não alcançam os 100% de cobertura de defeitos e em torno de 200 células têm a cobertura de defeitos com menos de 50%, algumas células com apenas 20%, como por exemplo AO (AND-OR) e OA (OR-AND) (HAPKE et al., 2014). Ao fazer uma análise do gráfico observou-se que os modelos de falhas utilizados foram os do CAT, enquanto os padrões de testes aplicados foram os do *stuck-at*, *transition delay* e CAT, ou seja, se o CAT modela 30 modelos de falhas para um leiaute em específico, os padrões gerados a partir do modelo *stuck-at* deverão ser capazes de cobrir essas 30 falhas. Sendo que, os padrões baseados em *stuck-at* modelaram apenas 12 falhas no mesmo circuito. Assim, os padrões que foram gerados a partir de 12 falhas modeladas não seriam capazes de cobrir as 30 falhas do CAT. O mesmo acontece com o modelo de falhas *transition delay*, se ele modelar 16 falhas, os padrões irão cobrir essas 16 falhas. No gráfico os padrões CAT que modelaram 30 falhas geram padrões para cobrir essas 30 falhas; os padrões *stuck-at* que geram padrões pra 12 falhas devem cobrir 30 falhas; os padrões *transition delay* que geram padrões pra 16 falhas devem cobrir 30 falhas. O resultado parece injusto quando se fala da cobertura das falhas específicas do CAT.

Figura 3.8 – Cobertura de defeitos do CAT em comparação com *Stuck-At* e *Transition Delay*.



Fonte: Adaptada de Hapke et al. (2014).

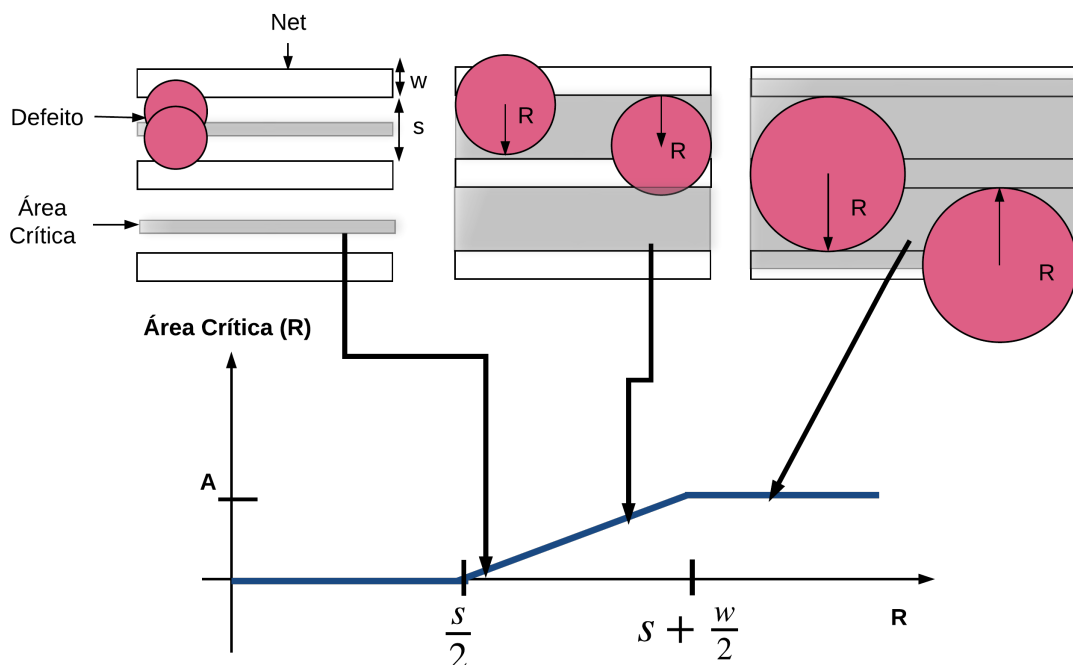
Além disso, no desenvolvimento desta pesquisa sobre os conceitos do *Cell-Aware Test*, surgiram dúvidas em relação aos parâmetros que seriam usados para estabelecer os valores de resistência e capacitâncias correspondentes a cada modelagem de falhas. Como os modelos de falhas são baseado no leiaute da célula, e cada leiaute da mesma célula pode ser diferente, com ligações feitas de formas e distâncias distintas, a lista de defeitos será diferente, e a forma de modelar falhas também. Os valores de resistências e capacitâncias para a simulação analógica serão então distintos. Não fica claro, nos exemplos expostos nos artigos sobre o CAT publicados nos últimos anos, que valores de parâmetros são usados. Por isso, para este trabalho foram feitos estudos mais aprofundados para saber como podem ser definidos os parâmetros. Na próxima seção, ir-se-á descrever alguns conceitos que serão base para a determinação de quais locais do leiaute são mais possíveis probabilisticamente de ocorrerem falhas. Tal estudo é de grande importância, pois pelo modelo CAT é possível o estabelecimento de inúmeros pontos de falhas tornando a tarefa do ATPG impraticável.

4 Área Crítica

Jacomet e Guggenbuhl (1993) relataram que modelos clássicos de falhas, como o modelo *Stuck-At*, não consideram o leiaute da célula. Além disso, não tem relação com a geometria do leiaute do circuito e, em alguns casos, não são baseados na tecnologia de fabricação. Outro fator importante é que, com a diminuição dos circuitos integrados e o consequente aumento na densidade de transistores, as falhas em ponte estão se tornando cada vez mais relevantes (JACOMET; GUGGENBUHL, 1993). Esses problemas foram analisados e soluções estudadas para que os modelos de falhas cobrissem o maior número de defeitos possíveis. Com isso, chegou-se à modelos de falhas que fossem baseados no estudo da área crítica do leiaute.

Nos anos seguintes, Bubel et al. (1995a) definiram o conceito de Área Crítica como sendo a probabilidade de uma dada área do leiaute conter defeitos, ou seja, quanto maior o defeito e menor o espaçamento entre as nets, maior a probabilidade dele causar uma falha (BUBEL et al., 1995a).

Figura 4.1 – Função da Área Crítica (BUBEL et al., 1995a).



Fonte: Elaborada pela autora.

Na Figura 4.1 é mostrado o gráfico da função Área Crítica (R), em que R é o valores do raio do defeito. Quando se tem duas interconexões com distância s e largura w , analisa-se qual a probabilidade de um defeito de raio R entre as nets causar uma

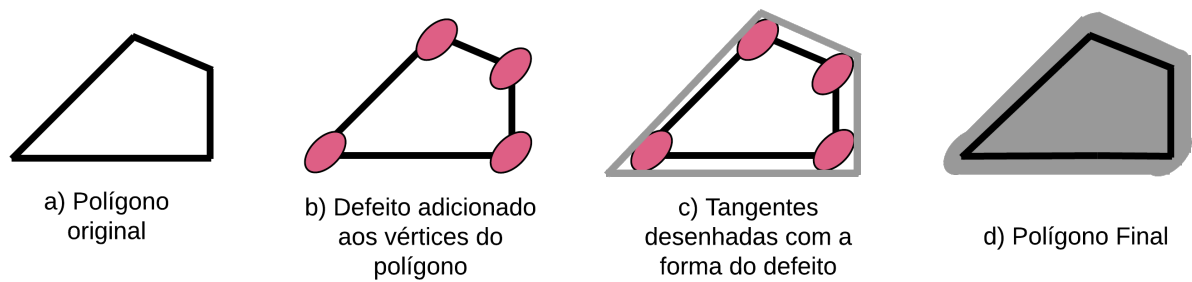
falha (BUBEL et al., 1995a). Na primeira parte da figura nota-se que os defeitos tem o menor raio, mas que ainda criam uma área crítica. Os demais defeitos tem raio maior, e quanto maior o raio do defeito, maior a área crítica. Quando os defeitos circulares tem um diâmetro ($x = 2R$) menor do que espaçamento (s) entre as interconexões eles não conseguem conectá-los eletricamente e a probabilidade de causar uma falha é nula. Quando o diâmetro é maior do que o espaçamento, a probabilidade aumenta de acordo com essa variação de diâmetro.

Formalmente, a probabilidade de área crítica, $h(x)$, pode ser definida como uma função do diâmetro do defeito e da distância entre as interconexões. O cálculo que é feito conforme Figura 4.1 é dado pela equação Eq. 4.1.

$$h(x) = \begin{cases} 0, & \text{para } 0 < x \leq s \\ \frac{x-s}{s+w}, & \text{para } s \leq x \leq 2s+w \\ 1, & \text{para } (2s+w) \leq x \end{cases} \quad (4.1)$$

em que w é a largura da interconexão, s é o espaçamento entre duas interconexões, x é o diâmetro do defeito e a função $h(x)$ é a sensibilidade à defeito, ou seja, a probabilidade do defeito ocasionar uma falha (ALLAN; WALTON, 1998). As funções de sensibilidade são representações da área crítica do leiaute ou a região a qual o defeito de um diâmetro x provavelmente causará uma falha.

Figura 4.2 – Método de expansão de polígonos para defeitos convexos arbitrários (ALLAN; WALTON, 1997a).

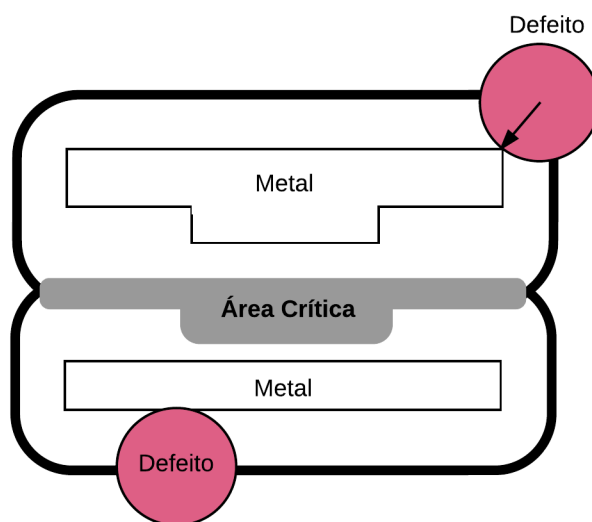


Fonte: Elaborada pela autora.

Em Allan e Walton (1997a) completaram o conceito de área crítica ao conceituar que ela pode ser encontrada pela interseção de regiões expandidas da forma geométrica. Na Figura 4.2 é mostrado um polígono de geometria arbitrária que é expandido através dos defeitos inseridos em suas vértices. Nela pode-se observar que o primeiro polígono é o original, o segundo está com defeitos elípticos em seus vértices, para que, na próxima etapa possam haver a expansão do polígono e, na quarta e última etapa pode ser vista a expansão desse polígono em área crítica. Esse método é importante pois nem todas as

interconexões ficam paralelas, então é necessário expandir essas áreas e verificar quando elas se cruzam. Quando isso acontecer, essa será considerada área crítica (Figura 4.3).

Figura 4.3 – Dinâmica do algoritmo que computa a Área Crítica (BUBEL et al., 1995a).

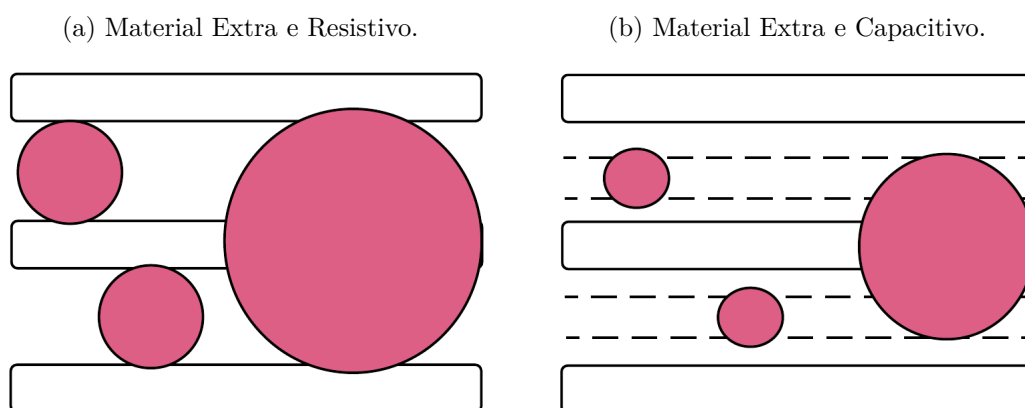


Fonte: Elaborada pela autora.

Em Allan e Walton (1998) foram definidos os conceitos de **Material Extra ou Perdido** e **Falha Resistiva ou Capacitiva**, os quais serão conceituados a seguir.

Falhas causadas por Material Extra: é quando um defeito físico é uma partícula a mais no circuito integrado podendo causar um curto-circuito entre interconexões (Figuras 4.4a e 4.4b) (BUBEL et al., 1995a).

Figura 4.4 – Material Extra (ALLAN; WALTON, 1998).

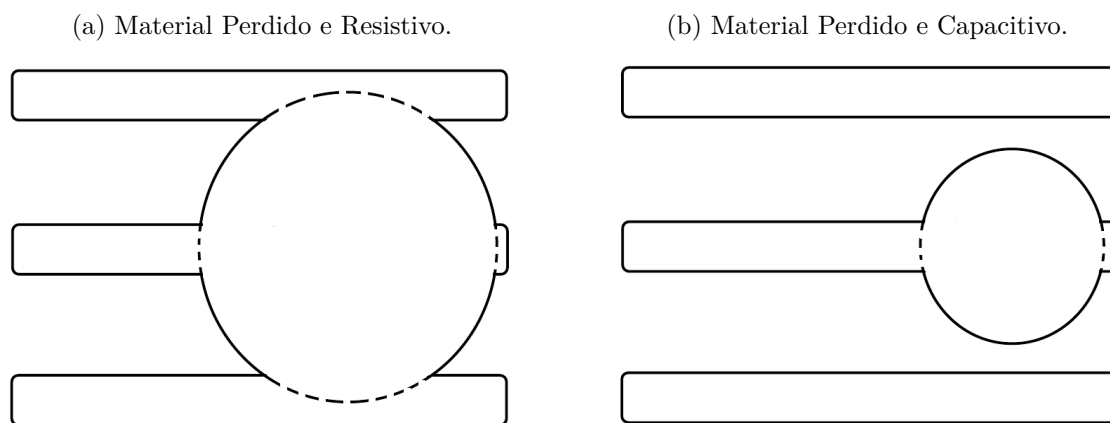


Fonte: Elaborada pela autora.

Falhas causadas por Material Perdido: é quando o defeito físico causa perda de material do circuito integrado. Esse fato ocorre com menos frequência mas também é

objeto de estudo. Esse tipo de defeito físico acaba fazendo com que o material do CI seja danificado com a perda de material, e conseqüentemente, pode causar o mal funcionamento do circuito (Figuras 4.5a e 4.5b) (BUBEL et al., 1995a).

Figura 4.5 – Material Perdido (ALLAN; WALTON, 1998).



Fonte: Elaborada pela autora.

Falhas Resistivas: são os defeitos que causam um curto-circuito entre duas interconexões distintas, como pode ser visto na Figuras 4.4a e 4.5a. Essas falhas podem ser modeladas eletricamente por resistores de baixo valor, para simular o curto, conectados entre as interconexões (ALLAN; WALTON, 1998).

Falhas Capacitivas: são os defeitos que não chegam a conectar as interconexões eletricamente, mas encurtam a distância entre elas. Essa representação pode ser vista na Figuras 4.4b e 4.5b. Essa distância reduzida pode ser representada por um capacitor. As falhas capacitivas são difíceis de se detectar e podem causar falhas de *transition delay*. O ponto principal deste trabalho é calcular a área crítica de acordo com os cálculos propostos no próximo capítulo. Esses cálculos são necessário para analisar o quanto os elementos parasitas poderão interferir no leiaute da célula.

5 Cálculo de Área Crítica baseado no Leiaute da Célula

Gkatziani et al. (2007) abordaram o problema dos modelos de falhas que usaram somente o conceito de área crítica para criar seus modelos baseados em leiaute. Eles relataram que os métodos tradicionais de cálculo de área crítica relacionavam a proximidade e o tamanho das interconexões no cálculo do tamanho do defeito, mas não indicavam a contribuição de pontes induzidas por defeitos aleatórios. Nesse artigo foi mostrado como capacitâncias de acoplamento podem ser inseridas para o cálculo de área crítica, até mesmo quando estão em níveis de metal diferentes (GKATZIANI et al., 2007).

A análise da área crítica é o ponto de partida para identificar pontos do circuito integrado em que os defeitos físicos tem maior probabilidade de causar uma falha. Para isso, calcula-se a probabilidade de ocorrência um defeito físico em um circuito integrado. Sabe-se que o processo de fabricação é minuciosamente feito em uma sala limpa, então, a possibilidade de uma partícula interferir nesse processo é pequena, mas não inexistente. O gráfico da Figura 5.1 ilustra essa afirmativa. Nela, é mostrada a distribuição de probabilidade para o tamanho do defeito inicial dado por $X_0 = 1nm$ e, observa-se que quanto maior o defeito, menor a probabilidade dele ocorrer em um circuito integrado. Enquanto que, quanto menor a partícula, maior a probabilidade dela interferir no processo de fabricação do CI.

A fundamentação do gráfico da Figura 5.1 é dada pela Eq. 5.1 (GKATZIANI et al., 2007).

$$f(x) = \frac{(p-1)x_0^{(p-1)}}{x^p}, \quad \text{para } x_0 \leq x \leq \infty \quad (5.1)$$

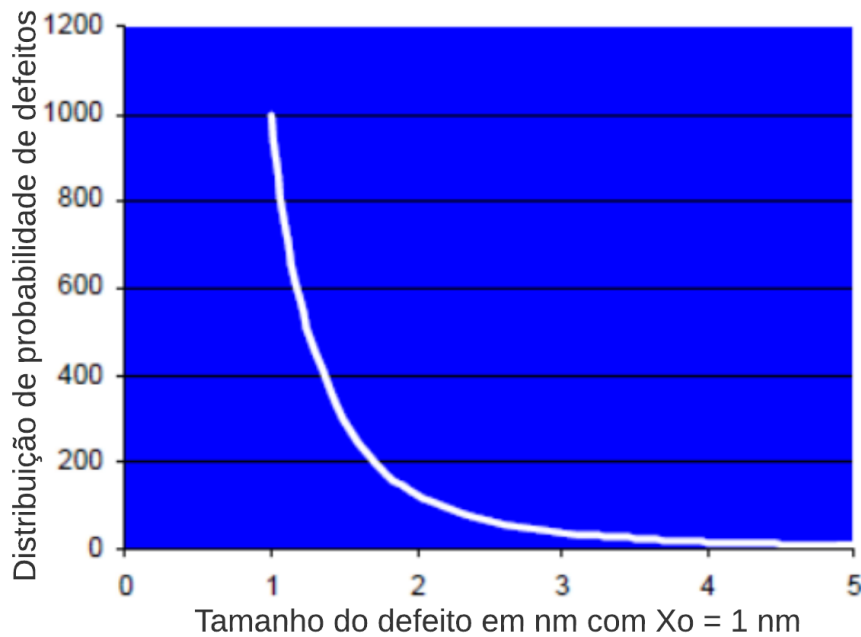
em que x é o diâmetro do defeito, sendo p dado pelo ITRS Roadmap e $f(x)$ é a **função densidade de probabilidade de defeito** para um dado tamanho de defeito.

Para analisar os tamanhos dos defeitos, a integral de $f(x)$ é calculada, como pode ser visto na Eq. 5.2. A equação a seguir será satisfeita se $p \neq 1$ ou 2.

$$\int_{x_0}^{\infty} f(x) dx = \int_{x_0}^{\infty} \frac{(p-1)x_0^{(p-1)}}{x^p} dx = \frac{(p-1)x_0^{(p-1)}}{-(p-1)} x^{-(p-1)} \Big|_{x_0}^{\infty} = 1 \quad (5.2)$$

Para simplificar o entendimento dos cálculo da área crítica associado às capacitâncias de acoplamento, iremos dividir em três partes: **Função Densidade de Probabi-**

Figura 5.1 – Distribuição de probabilidade para o tamanho do defeito



Fonte: Adaptada de Gkatziani et al. (2007).

lidade do Defeito, Área Crítica e Contribuição dos defeitos em interconexões paralelas.

5.1 Função Densidade de Probabilidade do Defeito

O primeiro dado que deve ser levado em consideração é a densidade de defeitos dada pela tecnologia (GKATZIANI et al., 2007). Para um *wafer* específico de uma dada tecnologia, o rendimento Y é calculado usando o mapa do *wafer*, que é ilustrado na Figura 5.2 (HESS; WEILAND, 1998).

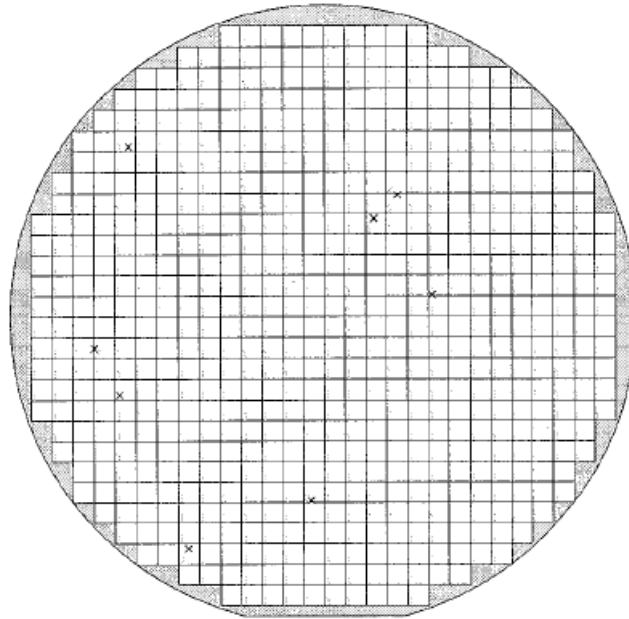
O cálculo do rendimento é dado pelo número de chips que passam pelo processo de testes, ou seja, que não estão defeituosos, e pelo número total de chips (HESS; WEILAND, 1998).

$$Y = \frac{\text{nº de chips não defeituosos}}{\text{nº total de chips}} \quad (5.3)$$

Baseado na área do chip A , o valor da densidade de defeitos D pode ser calculado usando a Eq. 5.4.

$$D = \frac{1 - Y}{A} \quad (5.4)$$

Figura 5.2 – Wafer contendo 648 chips.



Fonte: Hess e Weiland (1998).

Considera-se que a Distribuição de Defeitos é dada pela multiplicação da densidade de defeitos de uma dada tecnologia, D_0 , e a função densidade de probabilidade de defeitos $f(x)$, resultando assim, na Eq. 5.5.

$$F(x) = D_0 f(x) \quad (5.5)$$

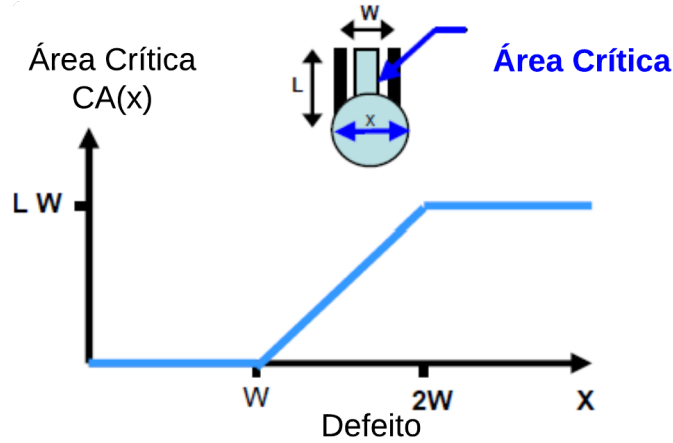
Concluída a primeira parte dos equacionamentos, o próximo conceito é o da **Área Crítica**.

5.2 Área Crítica

Para determinar a área crítica, analisam-se duas interconexões adjacentes com tamanho L e espaçamento entre elas W , obtendo assim as funções da Área Crítica, dada pela Eq. 5.6 e é ilustrada na Figura 5.3.

Na Fig. 5.3 é representado um gráfico da função da área crítica, $CA(x)$. Nessa função a área crítica é nula quando o diâmetro x é menor do que a distância W entre as interconexões. Já quando a diâmetro é entre W e $2W$ há a possibilidade de conexão, então a área crítica é dada por $Lx - LW$. E, quando o diâmetro do defeito é igual ou maior do que $2W$ a área crítica é LW .

$$CA(x) = \begin{cases} 0, & \text{para } 0 \leq x \leq W \\ Lx - LW, & \text{para } W < x < 2W \\ LW, & \text{para } x \geq 2W \end{cases} \quad (5.6)$$

Figura 5.3 – Área Crítica para um defeito de diâmetro x .

Fonte: Adaptada de Gkatziani et al. (2007)

A partir da Eq. 5.6 pode-se calcular o número de defeitos que se pode ter no leiaute, incluindo no cálculo a densidade de defeito, D_0 , e a função densidade de probabilidade de um tamanho de defeito, $f(x)$. Para essa análise serão consideradas duas interconexões paralelas, então, $F(x)$, que é igual a $D_0 f(x)$, é a **distribuição do tamanho do defeito**, e a unidade de $F(x)$ é dada em defeitos por unidade de área. Se forem multiplicados o valor de $F(x)$ pela área crítica e integrado o defeito de tamanho x , tem como resultado a **Contribuição de defeitos para o par de interconexões** (GKATZIANI et al., 2007).

5.3 Contribuição de defeitos em interconexões paralelas

Considerando D_{LW} como sendo a contribuição de defeitos para duas interconexões paralelas, pode-se introduzir os conceitos anteriores de área crítica, densidade de defeitos e função densidade de probabilidade de defeitos em uma integral, obtendo-se assim D_{LW} como pode ser visto na Eq. 5.7:

$$D_{LW} = \int_0^{\infty} CA(x) D_0 f(x) dx \quad (5.7)$$

Desenvolvendo os termos da Eq. 5.7 em soma de integrais para as três diferentes possibilidades de áreas críticas:

$$\begin{aligned} D_{LW} &= \int_0^W 0 D_0 \frac{(p-1) x_0^{(p-1)}}{x^p} dx \\ &+ \int_W^{2W} (Lx - LW) D_0 \frac{(p-1) x_0^{(p-1)}}{x^p} dx \\ &+ \int_{2W}^{\infty} LW D_0 \frac{(p-1) x_0^{(p-1)}}{x^p} dx \end{aligned} \quad (5.8)$$

Resolvendo o primeiro termo da Eq. 5.8 tem-se o resultado da integral:

$$\int_0^W 0D_0 \frac{(p-1)x_0^{(p-1)}}{x^p} dx = 0$$

Ao resolver o segundo termo da Eq. 5.8 tem-se:

$$\begin{aligned} & \int_W^{2W} (Lx - LW) D_0 \frac{(p-1)x_0^{(p-1)}}{x^p} dx \\ &= \int_W^{2W} (Lx) D_0 \frac{(p-1)x_0^{(p-1)}}{x^p} dx - \int_W^{2W} (LW) D_0 \frac{(p-1)x_0^{(p-1)}}{x^p} dx \end{aligned} \quad (5.9)$$

Para o primeiro termo da Eq. 5.9, o resultado é:

$$\int_W^{2W} (Lx) D_0 \frac{(p-1)x_0^{(p-1)}}{x^p} dx \quad (5.10)$$

Desenvolvendo a Eq. 5.10:

$$\begin{aligned} &= LD_0 \int_W^{2W} \frac{(p-1)x_0^{(p-1)}}{x^{(p-1)}} dx \\ &= -LD_0 (p-1)x_0^{(p-1)} \left(\frac{1}{(p-2)x^{(p-2)}} \right) \Bigg|_W^{2W} \\ &= -LD_0 (p-1)x_0^{(p-1)} \left(\frac{1}{(p-2)(2W)^{(p-2)}} \right) + LD_0 (p-1)x_0^{(p-1)} \left(\frac{1}{(p-2)W^{(p-2)}} \right) \\ &= LD_0 (p-1)x_0^{(p-1)} \left(\frac{1}{(p-2)} \right) \left(\frac{1}{W^{(p-2)}} \right) \left(1 - \frac{1}{2^{(p-2)}} \right) \end{aligned}$$

Para o segundo termo da Eq. 5.9 temos:

$$- \int_W^{2W} (LW) D_0 \frac{(p-1)x_0^{(p-1)}}{x^p} dx \quad (5.11)$$

Desenvolvendo a Eq. 5.11:

$$\begin{aligned} &= -LW D_0 (p-1)x_0^{(p-1)} \left(\frac{1}{-(p-1)x^{(p-1)}} \right) \Bigg|_W^{2W} \\ &= LW D_0 x_0^{(p-1)} \left(\frac{1}{x^{(p-1)}} \right) \Bigg|_W^{2W} \end{aligned}$$

$$\begin{aligned}
&= LW D_0 x_0^{(p-1)} \left(\frac{1}{(2W)^{(p-1)}} - \frac{1}{W^{(p-1)}} \right) \\
&= -LW D_0 x_0^{(p-1)} \left(\frac{1}{W^{(p-1)}} \right) \left(1 - \frac{1}{2^{(p-1)}} \right)
\end{aligned}$$

Para o terceiro termo da Eq. 5.8 tem o resultado:

$$\int_{2W}^{\infty} LW D_0 \frac{(p-1) x_0^{(p-1)}}{x^p} dx \quad (5.12)$$

Desenvolvendo a Eq. 5.12

$$\begin{aligned}
&= LW D_0 (p-1) x_0^{(p-1)} \left(\frac{-1}{(p-1) x^{(p-1)}} \right) \Big|_{2W}^{\infty} \\
&= LW D_0 (p-1) x_0^{(p-1)} \left(\frac{1}{(p-1) (2W)^{(p-1)}} \right)
\end{aligned}$$

Considerando os resultados anteriores, então:

$$D_{LW} = 0 + \int_W^{2W} (Lx - LW) D_0 \frac{(p-1) x_0^{(p-1)}}{x^p} dx + \int_{2W}^{\infty} LW D_0 \frac{(p-1) x_0^{(p-1)}}{x^p} dx \quad (5.13)$$

$$\begin{aligned}
&= LD_0 (p-1) x_0^{(p-1)} \left(\frac{1}{(p-2)} \right) \left(\frac{1}{W^{(p-2)}} \right) \left(1 - \frac{1}{2^{(p-2)}} \right) \\
&- LW D_0 x_0^{(p-1)} \left(\frac{1}{W^{(p-1)}} \right) \left(1 - \frac{1}{2^{(p-1)}} \right) \\
&+ LW D_0 x_0^{(p-1)} \left(\frac{1}{(2W)^{(p-1)}} \right)
\end{aligned}$$

$$D_{LW} = LD_0 x_0^{(p-1)} \left(\frac{1}{W^{(p-2)}} \right) \left[(p-1) \left(\frac{1}{(p-2)} \right) \left(1 - \frac{1}{2^{(p-2)}} \right) - \left(1 - \frac{1}{2^{(p-1)}} \right) + \left(\frac{1}{2^{(p-1)}} \right) \right] \quad (5.14)$$

De acordo com o ITRS (*International Technology Roadmap for Semiconductors*), o valor de p mais comum de ser usado para todas as camadas é de $p = 3$, então, substituindo os valores de p na Eq. 5.14 tem-se:

$$D_{LW} = LD_0 x_0^2 \left(\frac{1}{W} \right) \left[\left(2 \cdot \frac{1}{1} \cdot \frac{1}{2} \right) - \left(\frac{3}{4} \right) + \left(\frac{1}{4} \right) \right] = LD_0 x_0^2 \left(\frac{1}{W} \right) \left[1 - \left(\frac{3}{4} \right) + \left(\frac{1}{4} \right) \right]$$

$$D_{LW} = LD_0x_0^2 \left(\frac{1}{W} \right) \left[\frac{1}{4} + \left(\frac{1}{4} \right) \right] = LD_0x_0^2 \left(\frac{1}{W} \right) \left[\frac{1}{2} \right] = \frac{LD_0x_0^2}{2W}$$

Então, tem-se o seguinte resultado final para D_{LW} :

$$D_{LW} = \frac{LD_0x_0^2}{2W} \quad (5.15)$$

Ao obter D_{LW} resolve-se o cálculo da contribuição dos defeitos em interconexões paralelas. Para obter os parâmetros de resistências e capacitâncias para as interconexões de cada nível de metal, é necessário saber as propriedades de *timing*. As capacitâncias de acoplamento entre as interconexões são os parâmetros que precisam ser extraídos (GKATZIANI et al., 2007). A capacitância de acoplamento é dada pela Eq. 5.16:

$$C = \frac{\varepsilon_R \varepsilon_0 L h}{W} \quad (5.16)$$

em que L e W , que foram vistos na Figura 5.3, dados pelo comprimento e pela distância entre as interconexões, respectivamente, e h é dado pela altura das interconexões no metal. Nessa equação ε_0 é o valor do dielétrico no vácuo e ε_R é a constante dielétrica do material entre as duas interconexões. Para o dióxido de silício o ε_0 é aproximadamente 3,9 (GKATZIANI et al., 2007). Substituindo a Eq. 5.16 em 5.15 tem-se o resultado de D_{LW} na Eq. 5.17:

$$D_{LW} = \frac{LD_0x_0^2}{2W} = C \times \left(\frac{D_0x_0^2}{2\varepsilon_R \varepsilon_0 h} \right) \quad (5.17)$$

Com o cálculo da Contribuição de Defeitos D_{LW} associadas às capacitâncias de acoplamento, pode-se precisar em quais locais se tem a maior probabilidade de defeitos causarem falhas, ou seja, ao se fazer a análise do leiaute e verificar todos os pontos de capacitâncias, tem-se modelos de falhas específicos para a célula em estudo.

6 Processo de Fabricação CMOS

Como o processo de fabricação de um CI geralmente se utiliza diferentes camadas, é necessário saber como ocorre o processo de fabricação do CMOS e como as capacitâncias parasitas surgem durante esse processo. Esse entendimento faz-se necessário pois, através dele, obter-se-á os conceitos sobre como as capacitâncias são formadas entre os metais e o isolante.

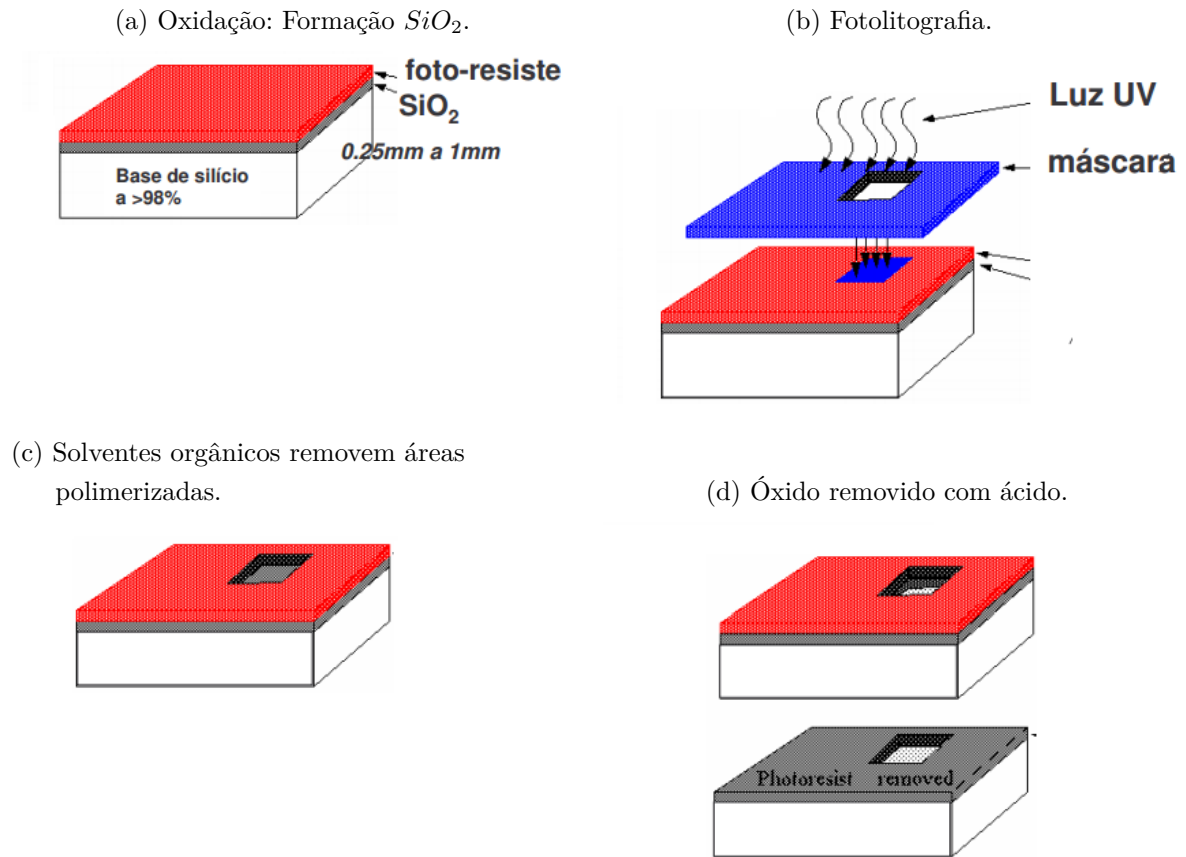
6.1 Processo de Fotolitografia, Criação de *n-well* e Crescimento do FOX

Os circuitos CMOS são fabricados em uma *wafer* de silício. Esses *wafers* são fatias circulares de silício de cristal único, com pureza maior do que 98%, como pode ser visto na Fig. 6.1a, e são levemente dopados (SILVA, 2007). O substrato é comum a todos os dispositivos. Considerando um substrato dopado com impurezas do tipo p, para produzir transistor de canal P, que requerem substrato do tipo n, faz-se necessário dopar uma região com impurezas do tipo n.

Em seguida, é feito o processo de fotolitografia, que é o processo pelo qual áreas predefinidas são mascaradas (protegidas). Na fotolitografia, processo que pode ser visto na Fig. 6.1b, o *wafer* é coberto por um polímero denominado *photoresist*. Este material é posteriormente sensibilizado pela incidência de luz, ou seja, se torna sensível (positivo) ou resistente (negativo) à remoção por um solvente orgânico (SILVA, 2007).

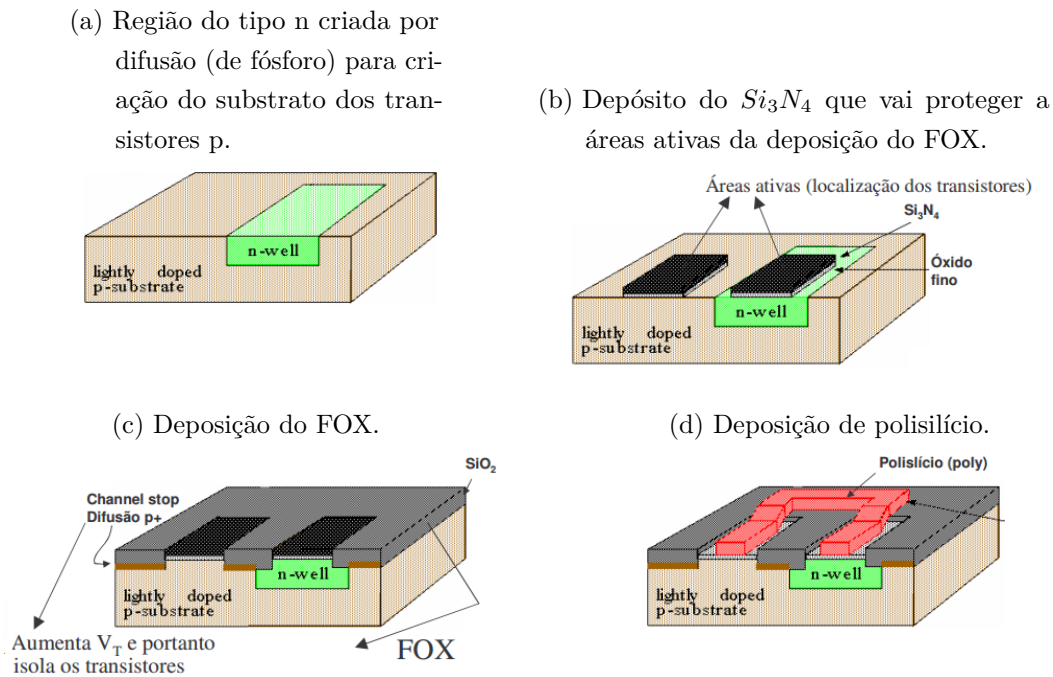
Após a ação dos solventes ter-se-á regiões com material exposto (Fig. 6.1c), que pode ser removido por ácidos (Fig. 6.1d). Nesta etapa o fotoresiste é removido e o substrato fica exposto, pronto para se criar o *n-well* (poço *n*). O fotoresiste é removido pois poderia derreter no processo de implante (SILVA, 2007).

Figura 6.1 – Fabricação CMOS: Primeiras etapas.



Fonte: (SILVA, 2007)

Depois da remoção do óxido ter sido feita pelo ácido, a próxima etapa é a criação da *n-well* (Fig. 6.2a). Nessa etapa a região do tipo n é criada por difusão (de fósforo) para criação do substrato dos transistores p. Em que primeiro é depositado Si_3N_4 que vai proteger as áreas ativas da deposição do FOX (*Field Oxide: SiO_2*), como pode ser visto na Fig. 6.2b. Para, na sequência, acontecer o crescimento de FOX (Fig. 6.2c) (SILVA, 2007).

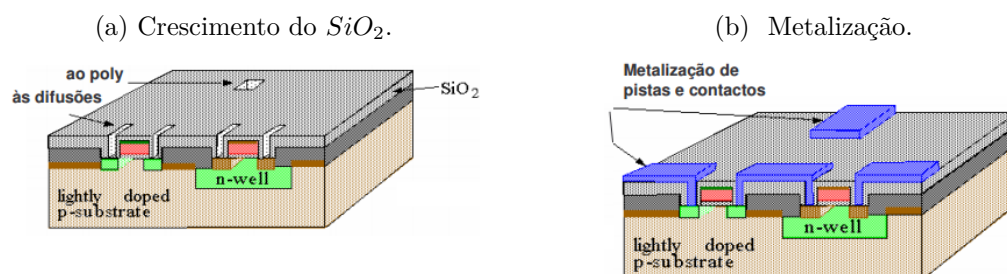
Figura 6.2 – Fabricação CMOS: Criação do *n-well* e Crescimento de óxido de campo.

Fonte: (SILVA, 2007)

6.2 Processo de Metalização

Após as primeiras etapas, todo o *wafer* é coberto com uma camada de SiO_2 (isolante), como pode ser visto na Fig. 6.3a. Nessa etapa serão deixados espaços para a inserção de contatos. Na Fig. 6.3b pode ser vista a metalização, ou seja, a camada de metal é disposta nos espaços vazios deixados pelo crescimento do isolante (SILVA, 2007).

Figura 6.3 – Fabricação CMOS: Máscaras.

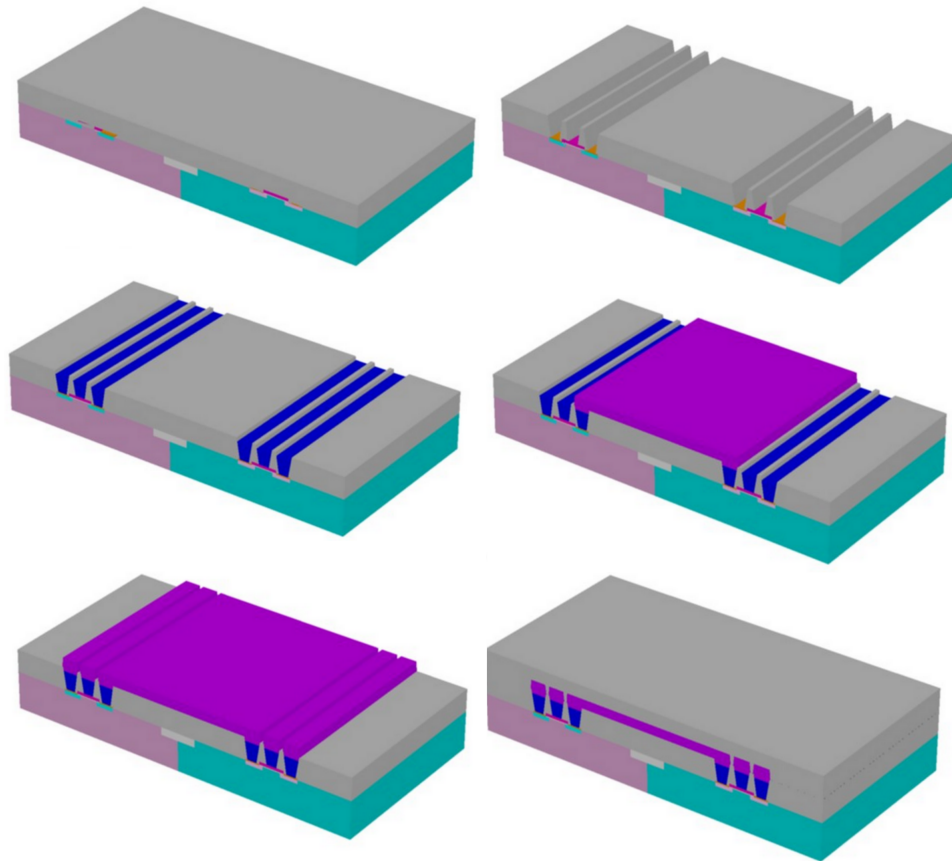


Fonte: (SILVA, 2007)

Na Fig. 6.4 pode ser visto como é realizado o processo de metalização. Durante toda a evolução desse processo, o isolante é colocado (SiO_2 em cinza), em seguida é tirado

parte dele para a deposição das vias ou contatos (azul), para somente depois, ser colocada a camada de metal (roxo). É nessa etapa que as capacitâncias parasitas se formam, da junção de metal-isolante-metal que ocorre durante esse processo de metalização.

Figura 6.4 – Camadas de metal e vias (ou contatos).



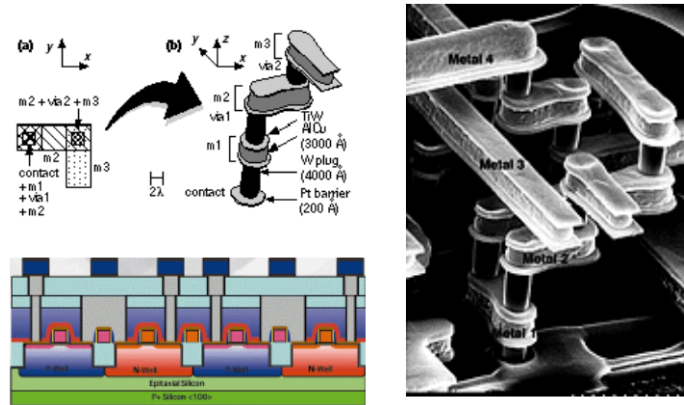
Fonte: (MOSHTAQ KEYVAN ALIREZA ZAD, 2016)

Na primeira camada de metal são usados contatos, e da segunda camada de metal em diante são usadas vias, como pode ser visto na Fig. 6.5.

Neste trabalho, feito no CADENCE usando uma tecnologia da IBM (*International Business Machine*), foram utilizados os seus parâmetros e medidas nos cálculos e análises. Na Fig. 6.6 podem ser vistas as várias espessuras utilizadas para metais em diferentes camadas. É possível ver também que é utilizado o *Stud Contact* (CA) entre o polisilício e o metal na primeira camada (M1). Entre metais em níveis superiores são usados vias (V1, V2, V3). Pelas especificações da IBM, o material isolante entre metais M1 e abaixo dele é o óxido, e o material isolante da metal M1 acima é o nitrato. Dados esses que serão importantes para o cálculo das capacitâncias, visto que é necessário saber a permissividade do material isolante.

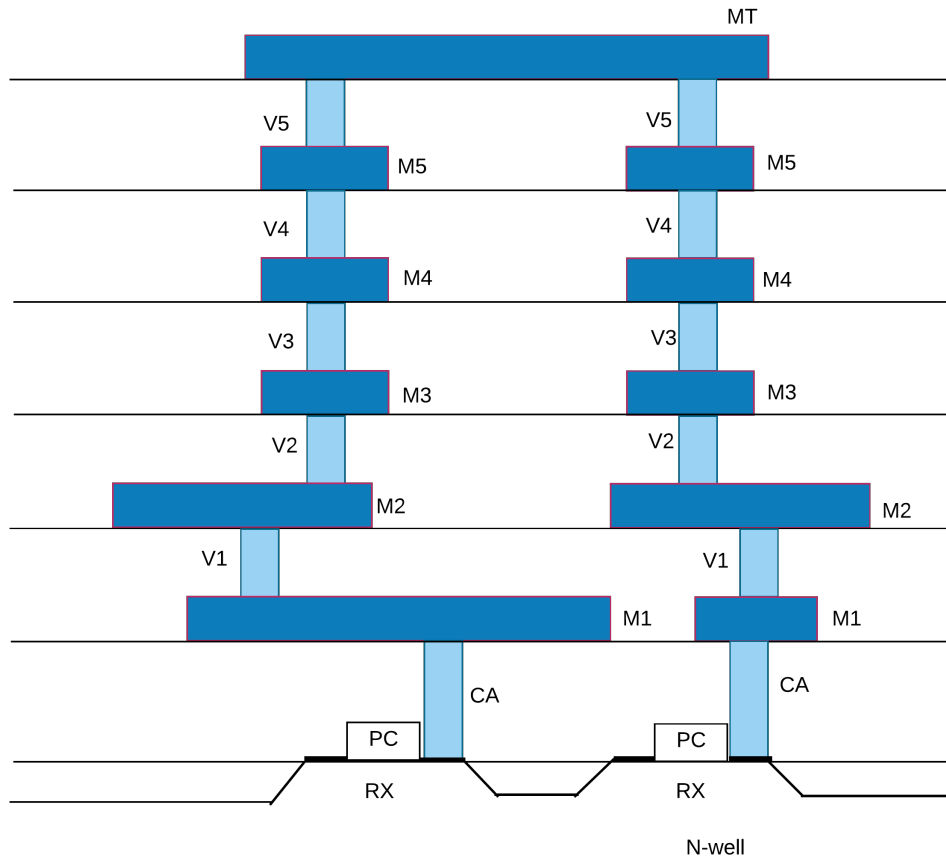
Além das especificações do material isolante a ser usado para os cálculos de

Figura 6.5 – Camadas de metal e vias (ou contatos).



Fonte: (SILVA, 2007).

Figura 6.6 – Visão lateral das camadas de metais.



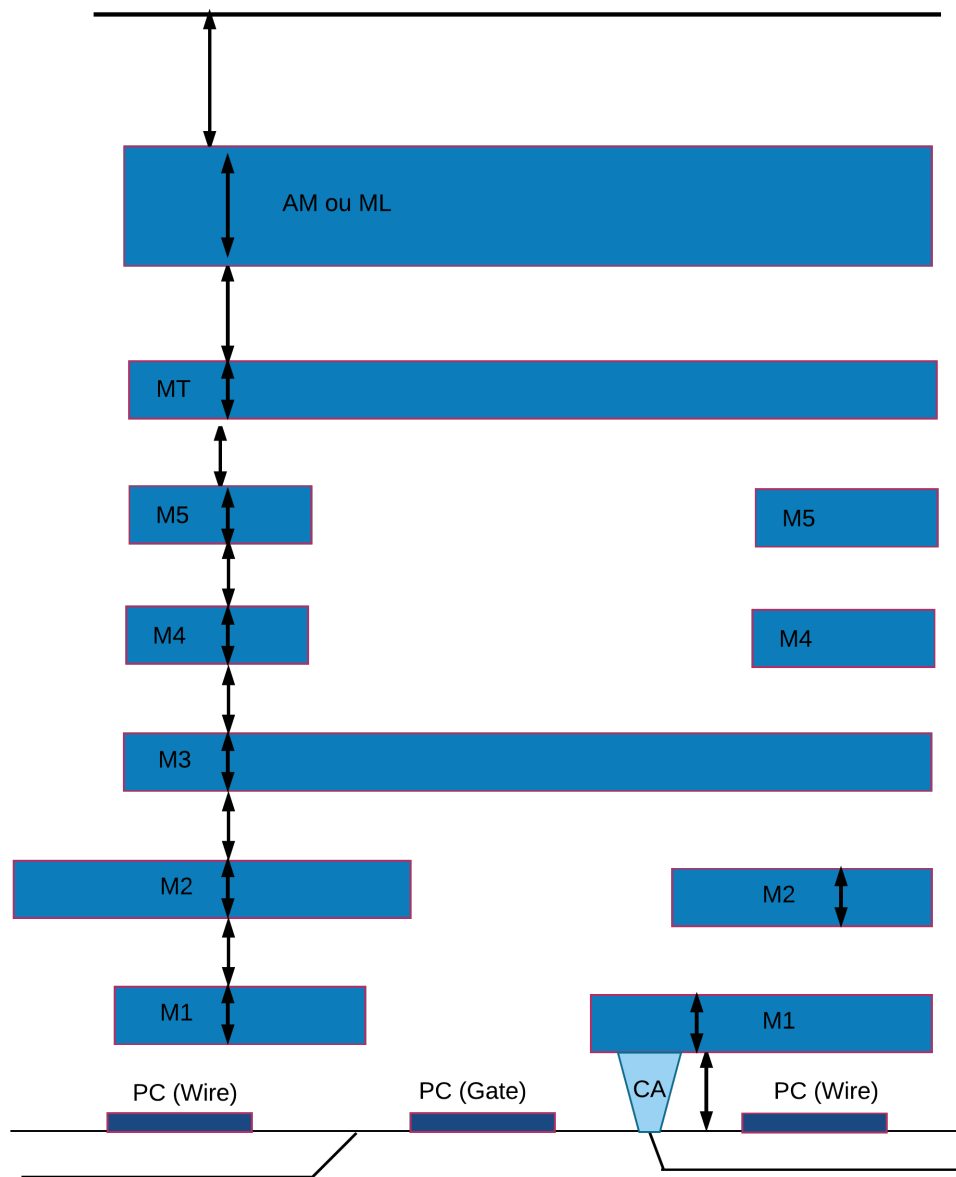
Fonte: Elaborada pela autora.

permissividade, é necessário saber as alturas das diferentes camadas de metais e polisilício.

Na Fig. 6.7 podem ser observados os valores de altura (h) dos metais em diferentes níveis e também o espaçamento entre eles. Esses valores também serão importantes no cálculo das capacitâncias parasitas em diferentes camadas de metal e as que existirem na

mesma camada.

Figura 6.7 – Visão lateral das camadas de metais.

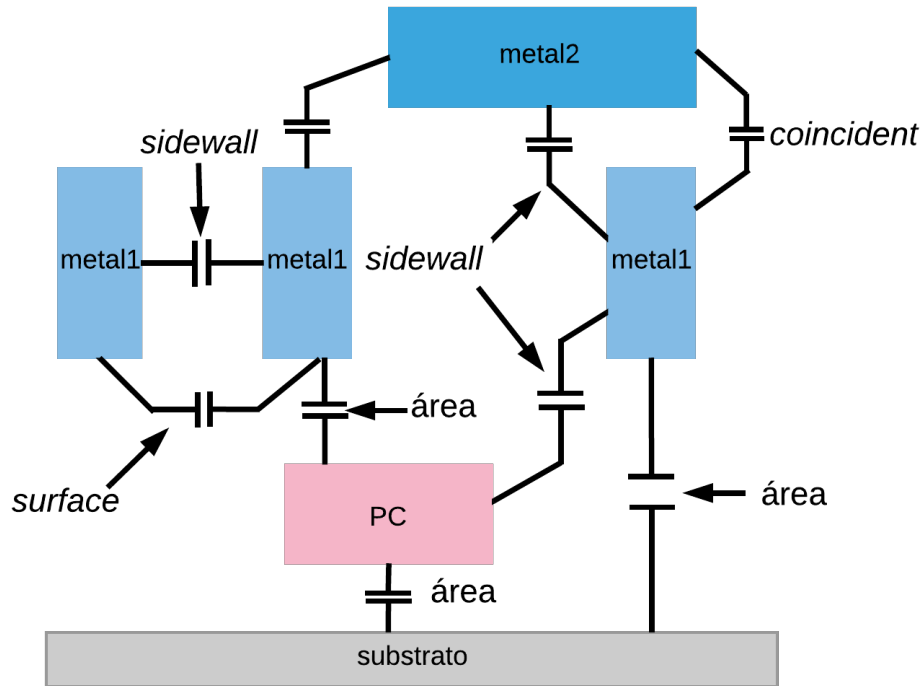


Fonte: Elaborada pela autora

6.3 Tipos de Capacitâncias

As capacitâncias parasitas são elementos não presentes no projeto do circuito que derivam das suas interconexões e dimensões (altura h , largura w e comprimento l). Essas capacitâncias são de dois tipos: *intralayer* (mesma camada) e *interlayer* (entre camadas) (IBM, 2014b). A ferramenta de extração Assura RCX compara as formas e medidas em nets selecionadas para obter os valores de capacitâncias, que podem ser: *sidewall*, *area*, *surface* e *coincident*. Como pode ser visto na Fig. 6.8, as capacitâncias *surface* são intralayer; as *coincident* e *area* são interlayer; e as *sidewall* podem ser interlayer ou intralayer.

Figura 6.8 – Capacitâncias Parasitas (visão lateral).



Fonte: Elaborada pela autora.

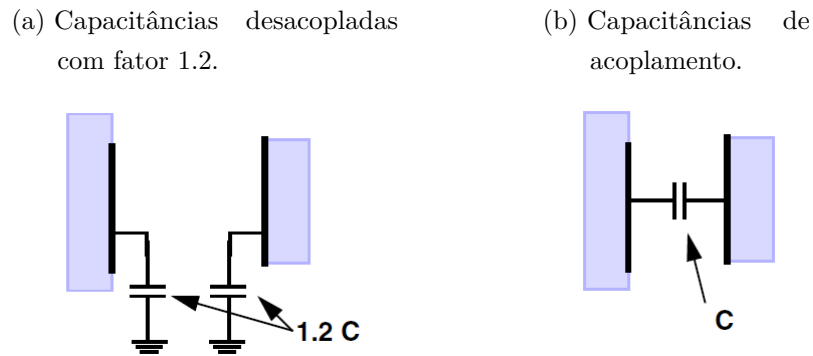
Na Fig. 6.8 tem-se uma visão lateral das camadas. Nesta visão lateral pode-se ver a altura (h) e a largura (w), dados necessários para os cálculos das capacitâncias.

O cálculo da capacitância total C_T entre os metais M1, por exemplo, será dada pela Eq. 6.1:

$$C_T = C_{sidewall} + C_{surface} + C_{area} \quad (6.1)$$

A ferramenta Assura RCX irá reportar as capacitâncias parasitas através de um arquivo em netlist em spice (netlist.sp) em que constará todas os elementos parasitas (resistências e capacitâncias). Ao extrair os elementos parasitas pode-se escolher, de acordo com a ferramenta, duas opções de acoplamento: acopladas ou desacopladas. Na Fig. 6.9 pode-se ver a diferença entre as duas formas de extração. Em uma delas, a Fig. 6.9b, as capacitâncias estão desacopladas e ligadas diretamente ao *ground* por um fator de 1.2. Na Fig. 6.9a é mostrada a capacitância de acoplamento, que são as capacitâncias entre camadas. Para efeitos de estudos das capacitâncias parasitas foram analisadas as capacitâncias extraídas no modo acopladas. Isso porque nesse tipo de extração serão consideradas as capacitâncias *interlayer* e *intralayer*.

Figura 6.9 – Modos de Extração de Capacitâncias no Assura RCX.

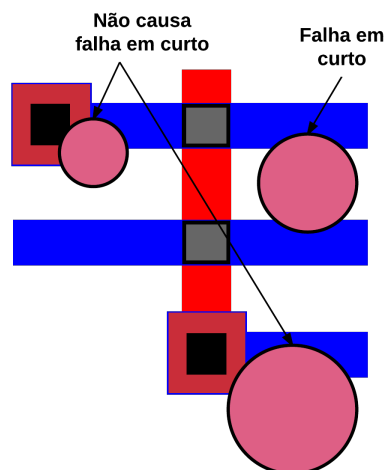


Fonte: Elaborada pela autora.

6.4 Elementos Parasitas como Modelos de Falhas

Como pode ser visto na Fig. 6.10, defeitos podem ser causados por contaminação ou partículas de vários tipos, que são representados pelos círculos rosa na Fig. 6.10, os quais podem ser modelados como defeitos de **material extra**, os quais causam **curtos** ou diminui a resistência entre nets (BUBEL et al., 1995b). Contudo, ainda pode existir defeitos causados por **falta de material** os quais causam **quebras** levando a circuitos abertos ou aumento de resistência.

Figura 6.10 – Diferença entre falha e defeito em nível de leiaute.



Fonte: Elaborada pela autora.

Além dos defeitos de material extra causarem curtos ou diminuição de resistência, eles também podem causar uma variação na capacitância entre as nets (Fig. 6.11). Por exemplo, na Fig. 6.11a é mostrada a capacitância C_{p1} e a distância s_1 . Se um defeito de

material extra ocorre, é razoável considerar que uma variação de capacitância irá ocorrer, que é dada por $Cp_1 \ll Cp_1^*$ causado pela distância $s_1 \gg s_1^*$, como pode ser visto na Fig. 6.11b. Contudo, se a distância entre nets for maior, como mostrado na Fig. 6.11c, então $Cp_1 > Cp_2$ já que $s_1 < s_2$. Dessa forma, o efeito do defeito de material extra é menor do que Cp_2 porque, como mostrado na Fig. 6.11d, $s_2 \approx s_2^*$, e como consequência, $Cp_2 \approx Cp_2^*$.

Conclui-se que, em um leiaute de CI, quanto menor for a distância, maior é a capacitância parasita, e maior é a probabilidade que o defeito contribua para uma falha (ALLAN; WALTON, 1999b; ALLAN; WALTON, 1997b; ALLAN; WALTON, 1998).

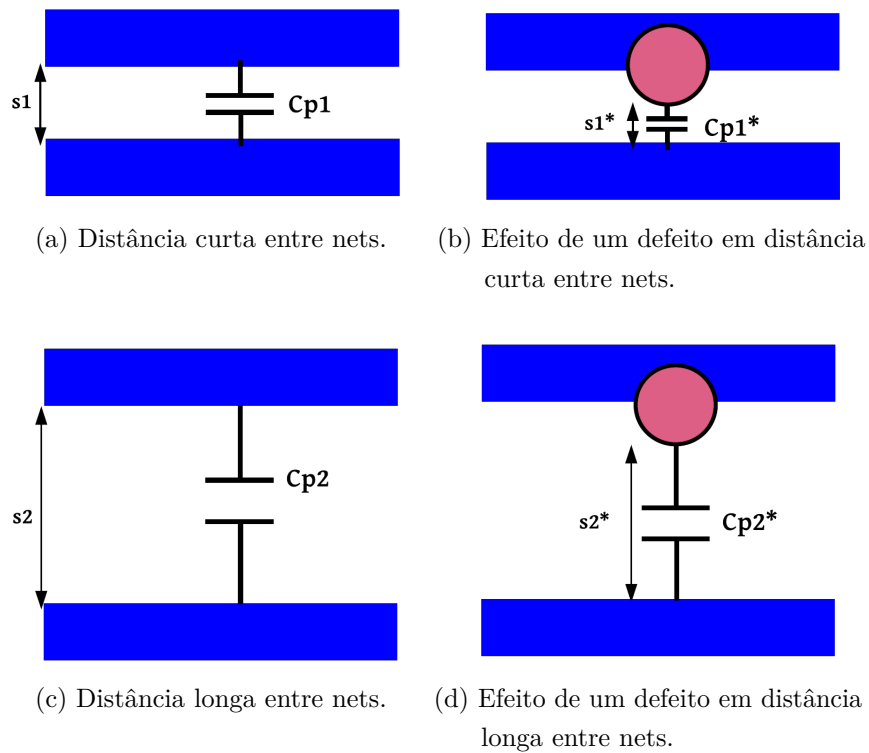


Figura 6.11 – Distâncias entre nets *versus* Probabilidade de defeitos devido às capacitâncias parasitas.

Fonte: Elaborada pela autora

Esses conceitos de capacitâncias parasitas serão utilizados para modelar as regiões do leiaute que tem maior probabilidade de que um defeito leve a uma falha. Serão usados também para mapear o leiaute e sugerir mudanças para melhorar o *heat map* como um todo.

Parte II

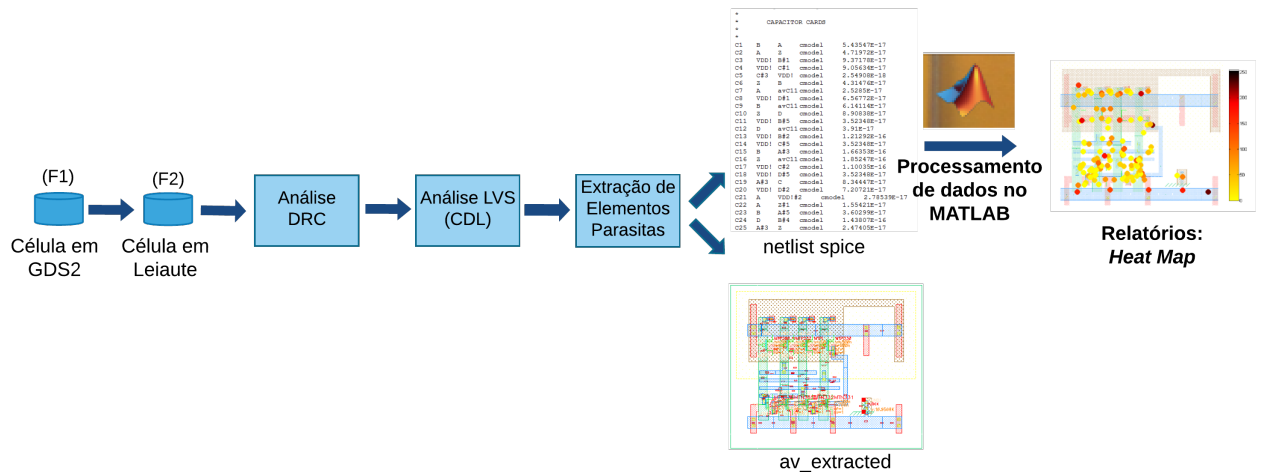
Proposta

7 Extração de Elementos Parasitas no CA- DENCE

Para a extração dos elementos parasitas no CADENCE e análise desses parâmetros foi feito um fluxograma (Fig. 7.1) para ao final obter os dados necessários para análise da célula do leiaute.

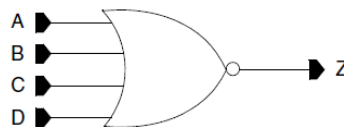
Para exemplificar os valores de capacitância em leiautes de uma dada tecnologia, foram utilizadas as células-padrão da biblioteca da IBM **CMRF7SF (CMOS7RF) 5V 18-Track RVT** com 381 células (FOUNDRY, 2014). O banco de dados da IBM CMOS 7RF (CMRF7SF) descreve células-padrão que são implementadas com um processo de litografia de $0,18\mu m$. O comprimento de todas as células é de $10,08\mu m$ (FOUNDRY, 2014). Essas células descrevem funções Booleanas primitivas, complexas e *latches*. A título de exemplo e estudos utilizou-se a célula *NOR4_A_5V*, que corresponde à uma porta lógica NOR com 4 entradas, como pode ser visto na Fig. 7.2.

Figura 7.1 – Fluxograma da Extração dos Elementos Parasitas.



Fonte: Elaborada pela autora.

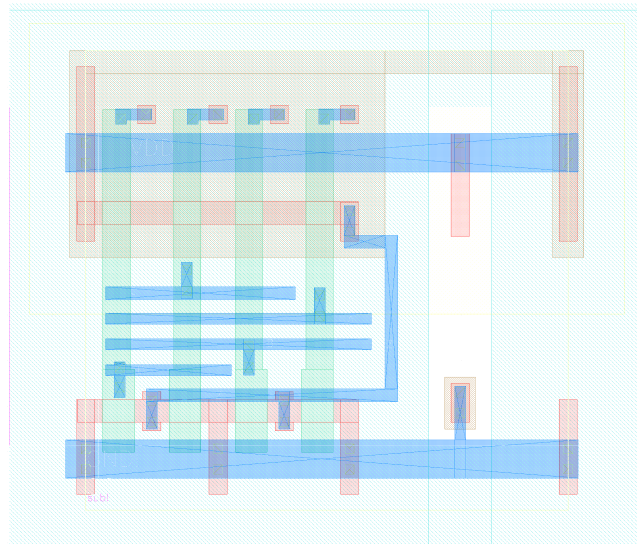
Figura 7.2 – Porta lógica da célula *NOR4_A_5V*.



Fonte: (FOUNDRY, 2014)

As células-padrão foram obtidas arquivo texto no formato GDS2 (F1) para posteriormente, serem analisadas em leiaute (F2). Essa etapa foi necessária para a análise DRC (*Design Rule Check*) da célula. A análise DRC determina se o leiaute físico atende uma série de regras de *design*. Na Fig. 7.3, pode-se ver o leiaute da porta lógica *NOR4_A_5V*.

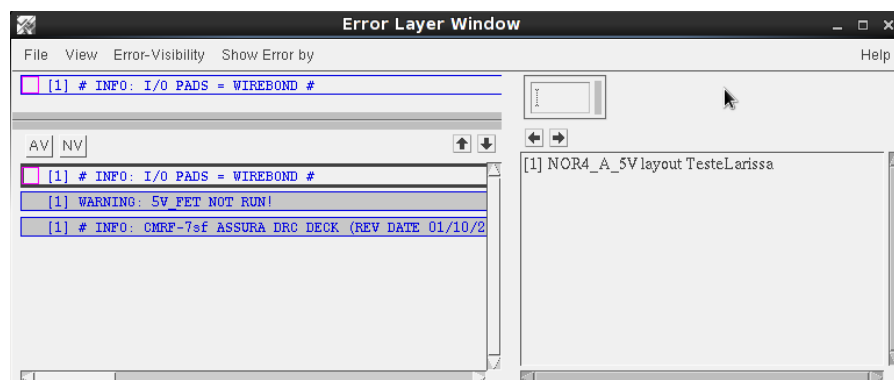
Figura 7.3 – Leiaute da célula *NOR4_A_5V*.



Fonte: Elaborada pela autora

Na Fig. 7.4 pode ser vista a análise DRC feita em uma das células-padrão da IBM. Nesta figura pode-se ver que a célula *NOR4_A_5V* atendeu à todos os requisitos e parâmetros recomendados. Contendo, apenas, um *warning* que não afetaria as próximas etapas.

Figura 7.4 – Análise DRC.



Fonte: Elaborada pela autora

Depois da análise DRC, é feita a análise em LVS (*Layout versus Schematic*), como as células padrão estavam em GDS2 originalmente, não se teve acesso aos esquemáticos. Portanto, para comparação LVS utilizou-se o CDL (*Compiler Description Language*) que

é uma linguagem de descrição para as estruturas de dados como *packages*, classes, etc. Na Fig. 7.5 é possível ver as modificações feitas no LVS para ser possível utilizar os arquivos em CDL ao invés do esquemático.

Figura 7.5 – Análise LVS (CDL).

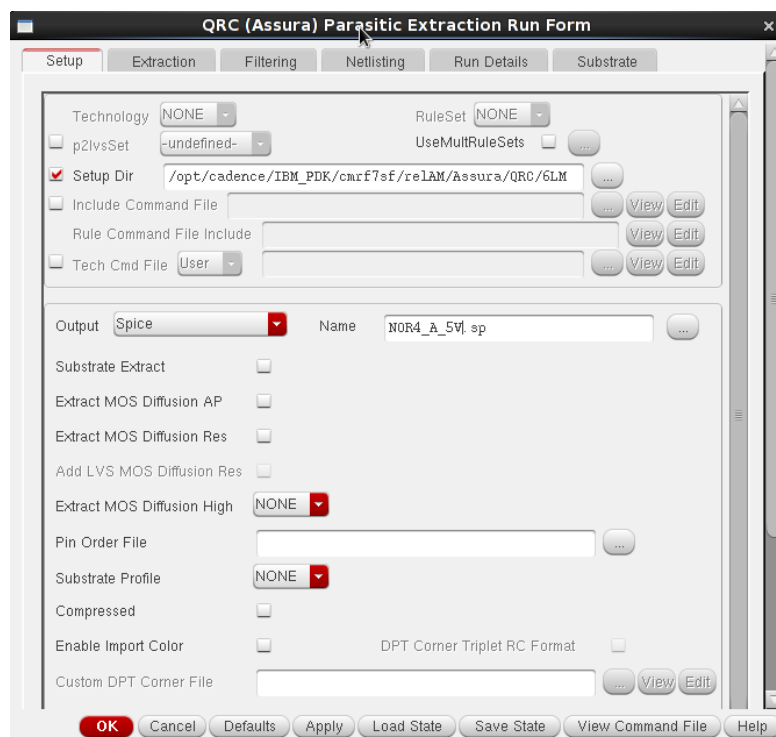


Fonte: Elaborada pela autora

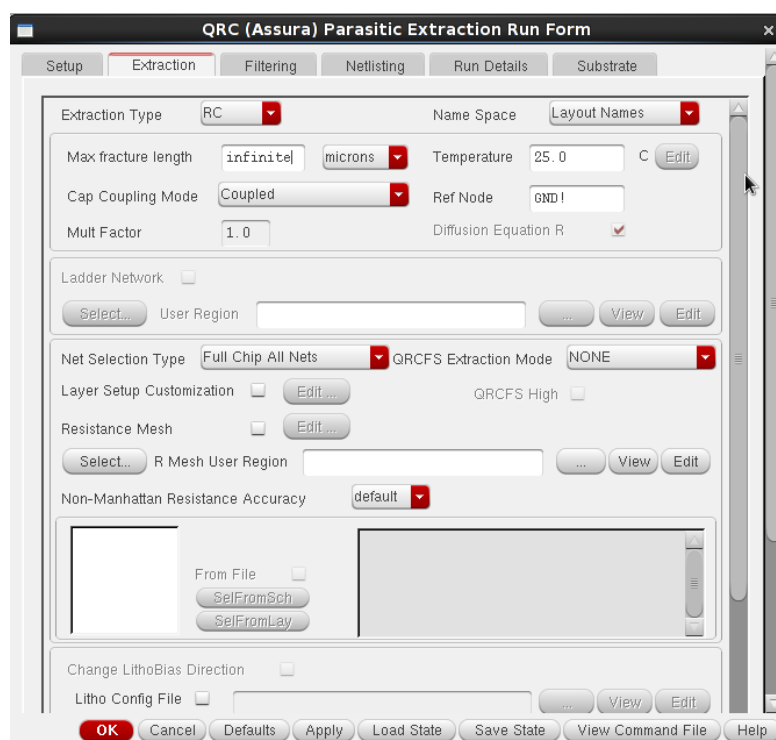
A partir do leiaute, e após o DRC e LVS, é possível obter os elementos parasitas através da ferramenta Assura QRC (IBM, 2014b). Na Fig. 7.6 é possível ver o *setup* do QRC em que pode-se escolher os tipos de saídas, neste trabalho foram utilizadas as saídas spice e *av_extracted*. Na Fig. 7.7 é possível ver a escolha do tipo de extração, que para esse trabalho foi escolhido modo de acoplamento **Cap Coupling Mode: Coupled**. Ao escolher a saída *av_extracted* tem-se como resultado a Fig. 7.8 que tem o leiaute da célula com as seus elementos parasitas. Esses elementos parasitas estarão localizados nas proximidades das nets a que correspondem.

Na Fig. 7.9 pode-se ver um exemplo de saída spice. Entre outras informações, ela é composta pelas capacitâncias parasitas. Esses valores em lista são importantes para posteriormente ser feito o *Heat Map* ou mapa de calor.

Ao extrair os elementos parasitas e obter os valores em arquivos spice (.sp), pode-se analisar como essas capacitâncias são calculadas pela ferramenta Assura QRC (IBM, 2014a). Essa etapa é de grande importância, pois nela é possível fazer um apanhado geral

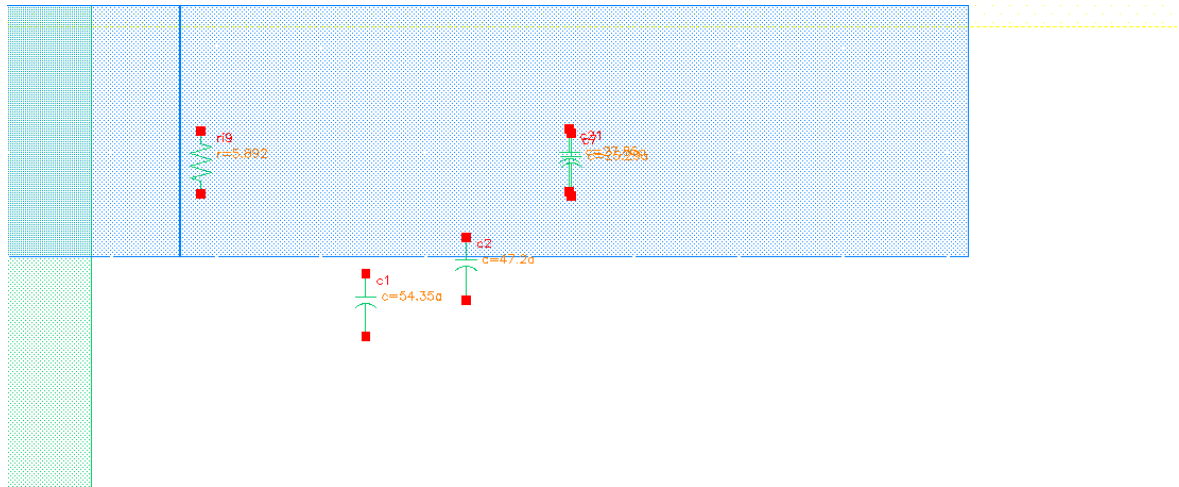
Figura 7.6 – QRC Assura *Setup*.

Fonte: Elaborada pela autora

Figura 7.7 – QRC Assura *Extraction*.

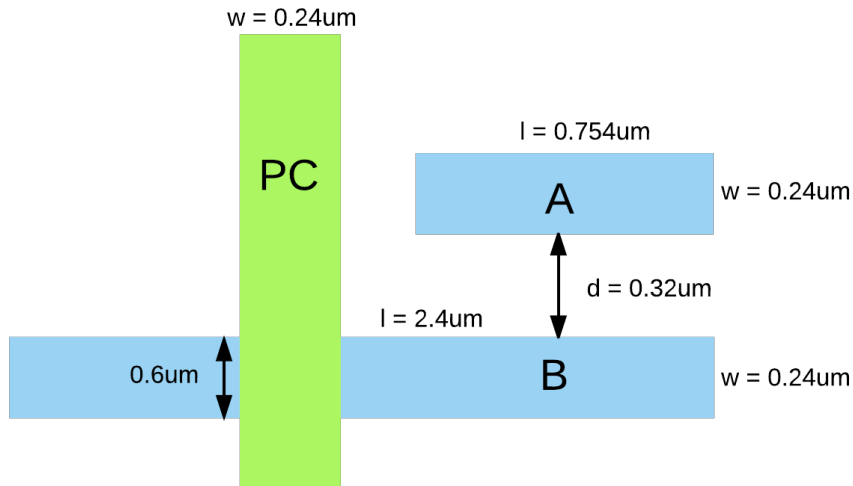
Fonte: Elaborada pela autora

Figura 7.10 – Zoom nas capacitâncias entre as nets A e B.



Fonte: Elaborada pela autora

Figura 7.11 – Visão do topo das nets selecionadas.



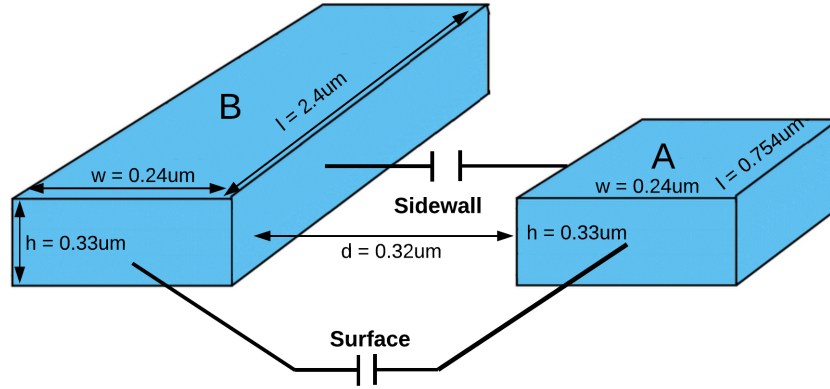
Fonte: Elaborada pela autora

Essas distâncias são dadas em μm (micrometros), e na visão do topo podem ser vistos o comprimento (l) e a largura (w).

Os valores de comprimento (A: $l = 0,754\mu m$ e B: $l = 2,4\mu m$) e largura (A: $w = 0,24\mu m$ e B: $w = 0,24\mu m$) foram obtidos no próprio leiaute. E a altura (A: $h = 0,33\mu m$ e B: $h = 0,33\mu m$) foi dada pelo fabricante para os metais e polisilício. Esses valores especificados podem ser vistos na visão frontal das nets da Fig. 7.12. Em que se pode observar que existem duas capacitâncias entre metais M1. A permissividade do material isolante (SiO_2) é $\varepsilon_R = 4,1$ e a permissividade $\varepsilon_0 = 8,854e^{-12} \frac{F}{m}$.

Os valores de capacitâncias *sidewall* e *surface* são dados pelas Eqs. 7.1 e 7.2,

Figura 7.12 – Visão frontal das nets selecionadas.



Fonte: Elaborada pela autora

respectivamente.

$$C_{sidewall} = \frac{\varepsilon_R \varepsilon_0 A}{d} = \frac{4,1 \times 8,854e^{-12} \frac{F}{m} \times 0,754\mu m \times 0,33\mu m}{0,32\mu m} = 0,89845965e^{-17} F \quad (7.1)$$

$$C_{surface} = \frac{\varepsilon_R \varepsilon_0 A}{d} = \frac{4,1 \times 8,854e^{-12} \frac{F}{m} \times 0,24\mu m \times 0,33\mu m}{0,32\mu m} = 2,822660733e^{-17} F \quad (7.2)$$

Na Fig. 7.13 pode ser vista a capacitância *area*, que é dada em diferentes níveis e com diferentes materiais (M1 e polisilício). Essa capacitância é dada pela área sombreada no polisilício, com valores de comprimento $l = 0,6\mu m$ e largura $w = 0,24\mu m$. E a distância entre os diferentes níveis é dada pelo fabricante e tem valor $h = 0,31\mu m$.

O resultado do efeito da distância entre o polisilício e o metal M1 é dado pela capacitância *area* dado pela Eq. 7.3.

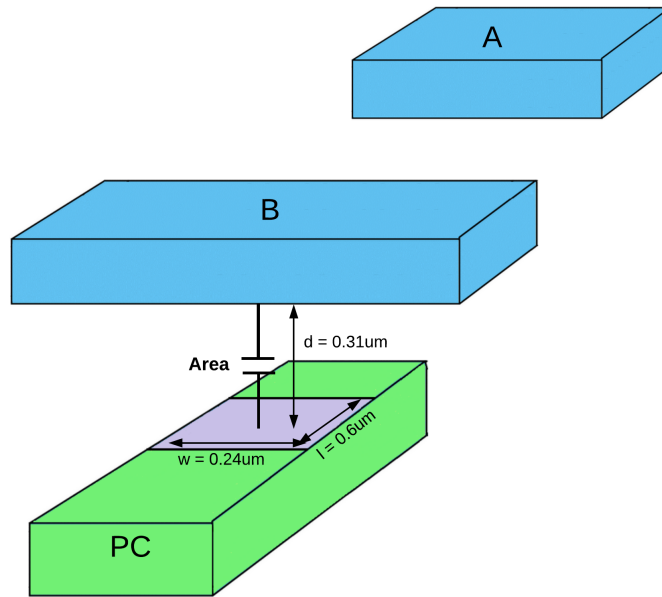
$$C_{area} = \frac{\varepsilon_R \varepsilon_0 A}{d} = \frac{4,1 \times 8,854e^{-12} \frac{F}{m} \times 0,6\mu m \times 0,24\mu m}{0,31\mu m} = 1,68625858065e^{-17} F \quad (7.3)$$

Sendo assim, a capacitância total é dada pela soma das três capacitâncias (*surface*, *sidewall* e *area*):

$$C_{Total} = C_{lateral} + C_{surface} + C_{area} = 5,403778964e^{-17} F \quad (7.4)$$

Sendo o erro dado pela Eq. 7.5:

Figura 7.13 – Visão lateral das nets selecionadas.



Fonte: Elaborada pela autora

$$\begin{aligned}
 erro &= \frac{C_{Total_{spice}} - C_{Total}}{C_{Total_{spice}}} * 100\% \\
 &= \frac{5,43547e^{-17} - 5,403778964e^{-17}}{5,43547e^{-17}} * 100\% \\
 &= 0,583041\%
 \end{aligned} \tag{7.5}$$

8 Aplicação da Proposta do *Heat Map* em Células-Padrão da Biblioteca

A análise do leiaute de acordo com nossa proposta é baseada na construção de um *Heat Map*, que é um mapa de calor que indica em quais regiões as capacitâncias são mais altas. Esse mapa de calor foi obtido através do MATLAB. Na Fig. 8.1 é possível ver o fluxograma da programação em MATLAB.

- **Ler arquivo imagem .png**

O fluxograma se inicia com a obtenção do arquivo imagem .png do leiaute obtido pelo CADENCE. Esse arquivo imagem é obtido do leiaute de uma célula padrão. Esse arquivo é exportado do CADENCE para o MATLAB. Com o arquivo exportado é possível a leitura do arquivo .png no MATLAB. Com ele, é possível obter as imagens das capacitâncias e suas localizações. Essas localizações são armazenadas posteriormente em arquivo Excel (.xlsx)

- **Ler arquivo Excel com localizações (x,y) das capacitâncias**

Com o arquivo Excel (.xlsx) é possível obter a matriz bidimensional de valores de capacitâncias para o MATLAB. Esses valores são inseridos em um *loop* finito que irá aumentar as regiões das capacitâncias em 40 vezes para ficar mais clara as regiões de possíveis falhas.

- **Normalizar os valores de capacitância (0 a 255)**

Os valores de capacitâncias então são normalizados para valores de 0 a 255. As cores do *heat map* variam nessa escala de valores, sendo 0 a cor mais clara e 255 a cor mais escura. Para normalizar os valores de capacitâncias foi feito o seguinte cálculo:

$$\text{delta} = \frac{255}{\text{maxima capacitancia} - \text{minima capacitancia}} \quad (8.1)$$

Os valores de capacitâncias são multiplicados por delta, em consequência disso, os valores a partir de então terão variação de 0 a 255.

- **Associar os valores de capacitância normalizados com cores (*heat map*)**

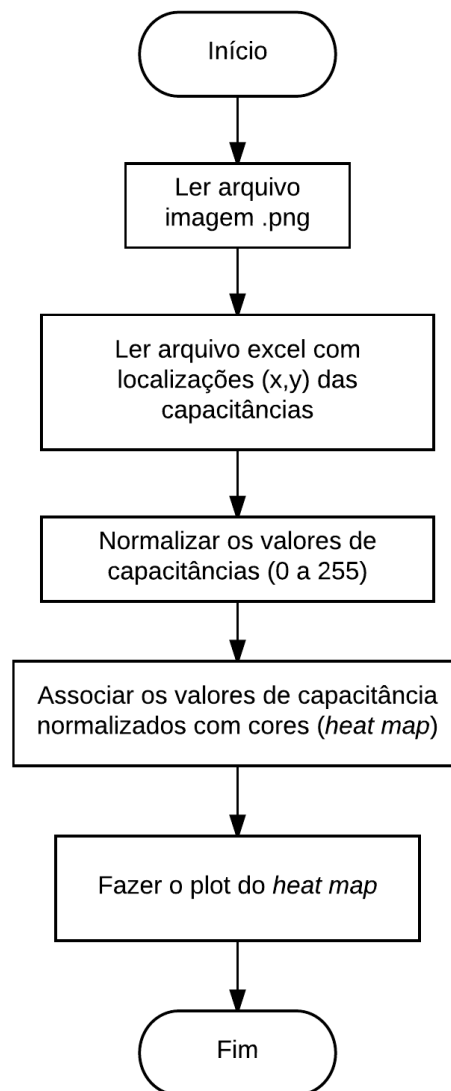
Os valores de capacitâncias são associados à cores do *heat map*. Isso ocorre quando insere-se a matriz bidimensional de valores nas funções **imagesc**, **meshgrid** e **colormap(hot)**, retornando assim o *heat map*.

- **Fazer o plot do *heat map***

Após realizadas as funções que retornam o *heat map* é possível associá-lo à imagem

do leiaute. É feito o plot do *heat map* juntamente com o arquivo imagem .png do leiaute. Assim, é possível observar as regiões de mais escuras e, conseqüentemente, de maior suscetibilidade à falhas.

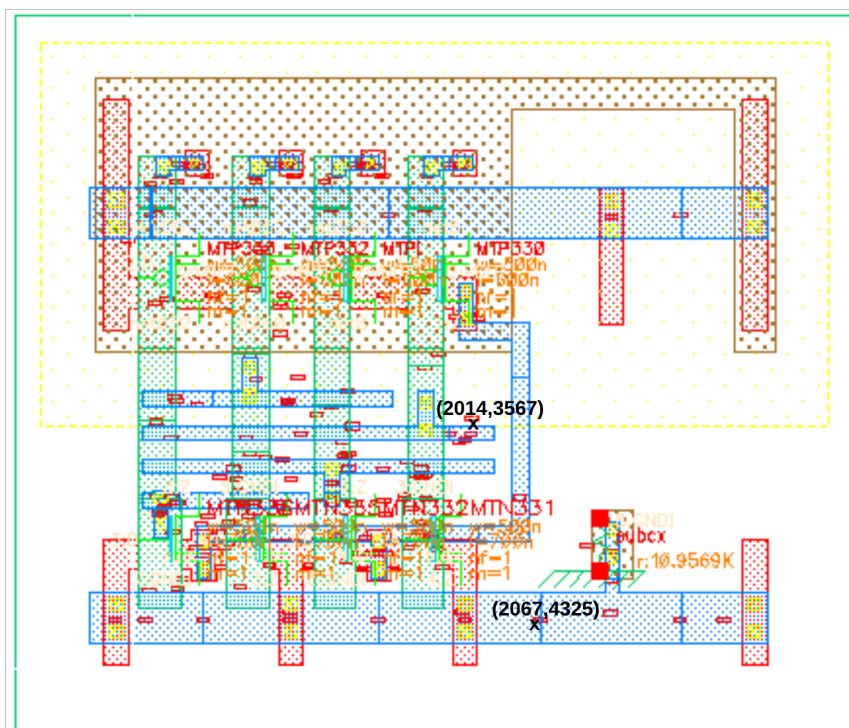
Figura 8.1 – Programa no MATLAB para *Heat Map* de um leiaute.



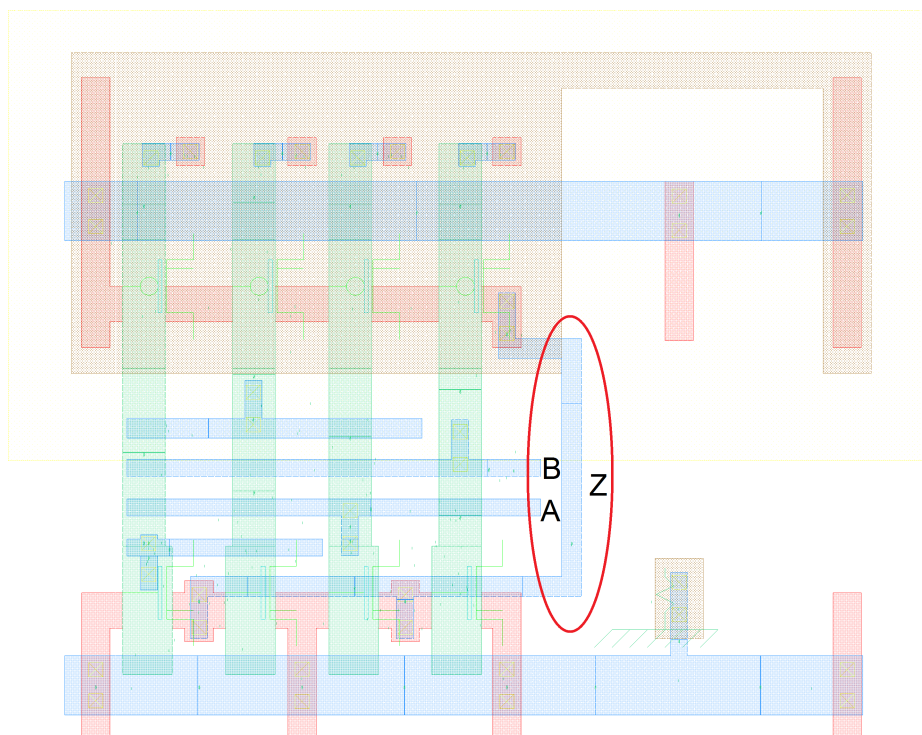
Fonte: Elaborada pela autora

Sabendo-se analisar o *heat map* com as regiões mais sujeitas à falhas, é possível sugerir mudanças ao projetista para que as capacitâncias diminuam e assim, diminuir o potencial de falha de uma determinada região. Na Fig. 8.3 é possível ver o leiaute da célula *NOR4_A_5V*. Nesse leiaute está sinalizada a net da saída *Z*. Quando há uma mudança nessa net, as entradas *A* e *B* são afetadas. Ao aumentar a distância entre *Z* e *A*, *B*, diminui-se a capacitância parasita nessas nets. Na Fig. 8.4 é possível ver como foi feita essa alteração no leiaute.

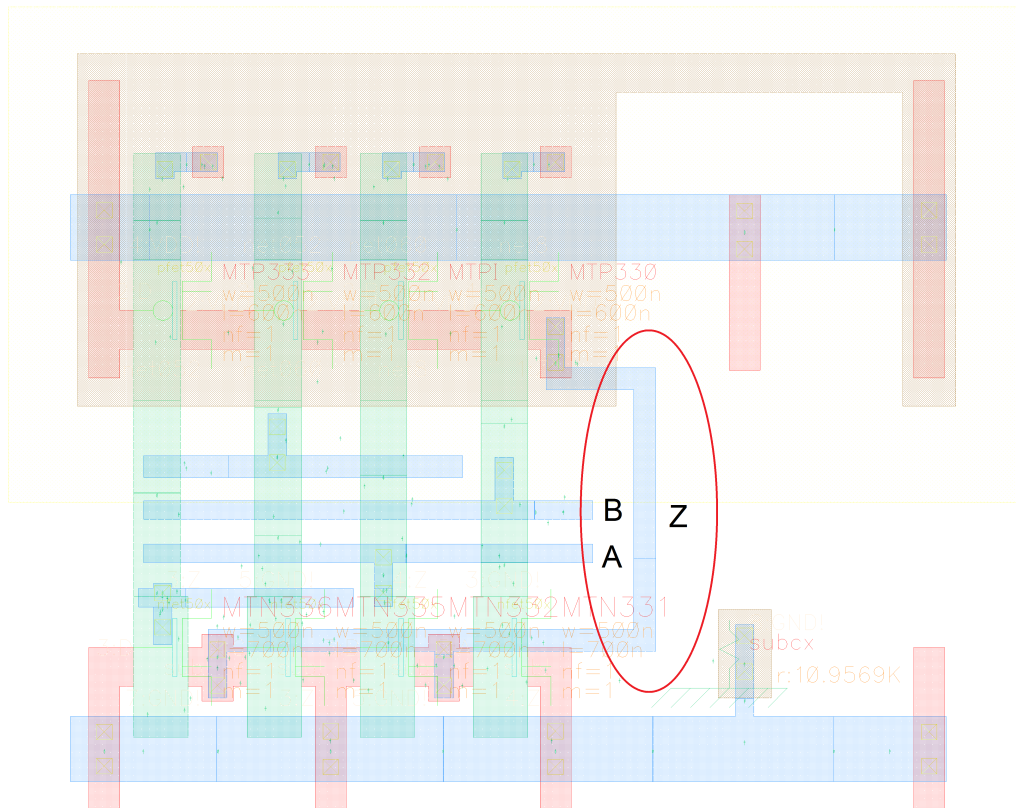
Figura 8.2 – Leiaute com elementos parasitas e localizações.



Fonte: Elaborada pela autora

Figura 8.3 – Leiaute da Célula *NOR4_A_5V* Original.

Fonte: Elaborada pela autora

Figura 8.4 – Leiaute da Célula *NOR4_A_5V* Modificado.

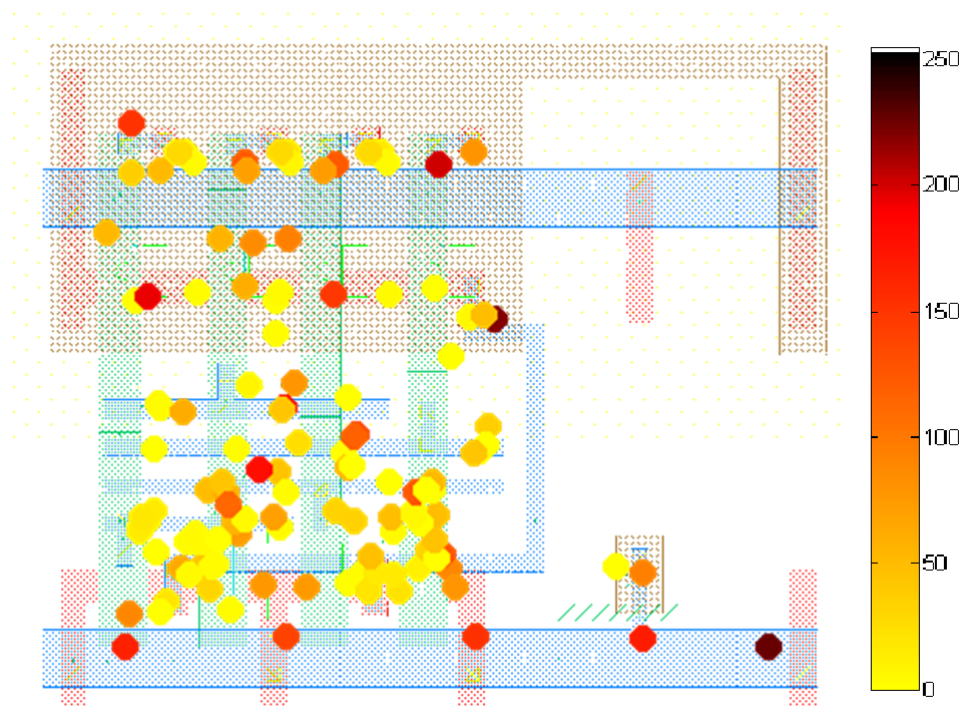
Fonte: Elaborada pela autora

Na Fig. 8.5 é mostrado o *Heat Map* do leiaute original da célula *NOR4_A_5V* com as regiões mais quentes (laranja, vermelho, marrom) como sendo as que tem maiores probabilidades de falhas. Quando a distância aumenta, as ilhas de calor localizadas nas proximidades das nets A, B e Z, as cores (amarelo e laranja) predominam, já que as capacitâncias diminuíram em relação ao leiaute original. Na Fig. 8.6 é possível notar duas regiões que claramente modificaram apenas pela mudança de distância. As capacitâncias parasitas ficaram menores do que no leiaute original.

Essa foi uma pequena mudança que poderia ser sugerida ao projetista para modificar as capacitâncias e tornar essas regiões menos propícias à falhas. Para que essas mudanças possam ocorrer é preciso analisar também os requisitos de DRC, que foram levados em conta para essa mudança.

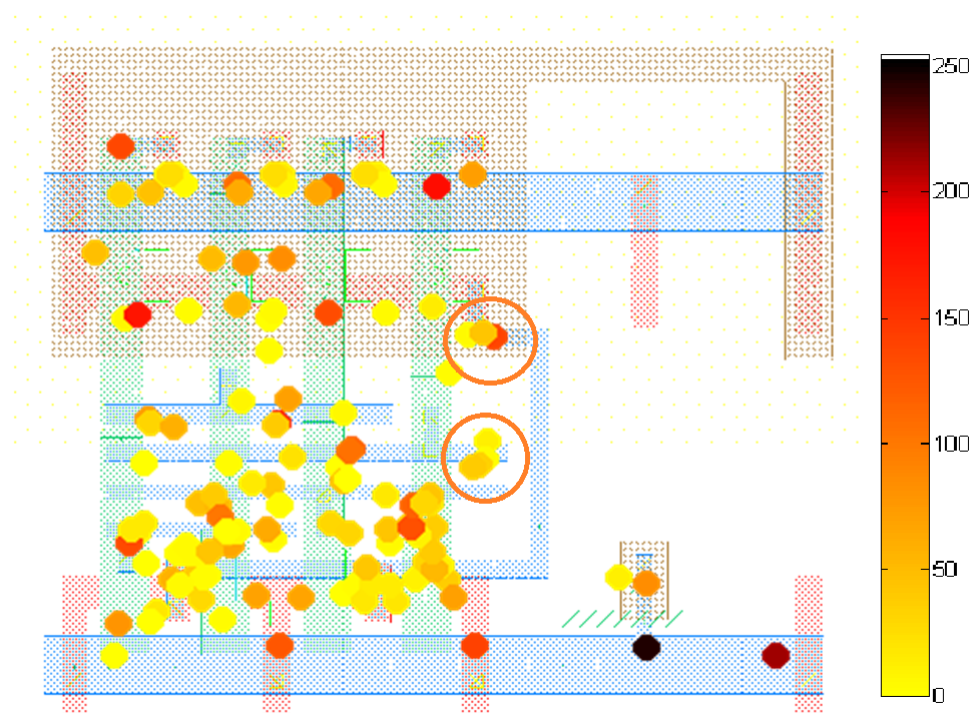
A proposta é usar *heat maps* para fazer análise de leiaute e verificar regiões com maior probabilidade de que um defeito evolua para uma falha. Nessa proposta, ao invés de usar modelos de falhas seria possível o projetista analisar as regiões mais passíveis de falhas e sugerir mudanças.

Figura 8.5 – Heat Map com Célula NOR4_A_5V Original.



Fonte: Elaborada pela autora

Figura 8.6 – Heat Map com Leiaute da Célula NOR4_A_5V Modificado.



Fonte: Elaborada pela autora

9 Implicações da Área Crítica em *Cell-Aware Test*

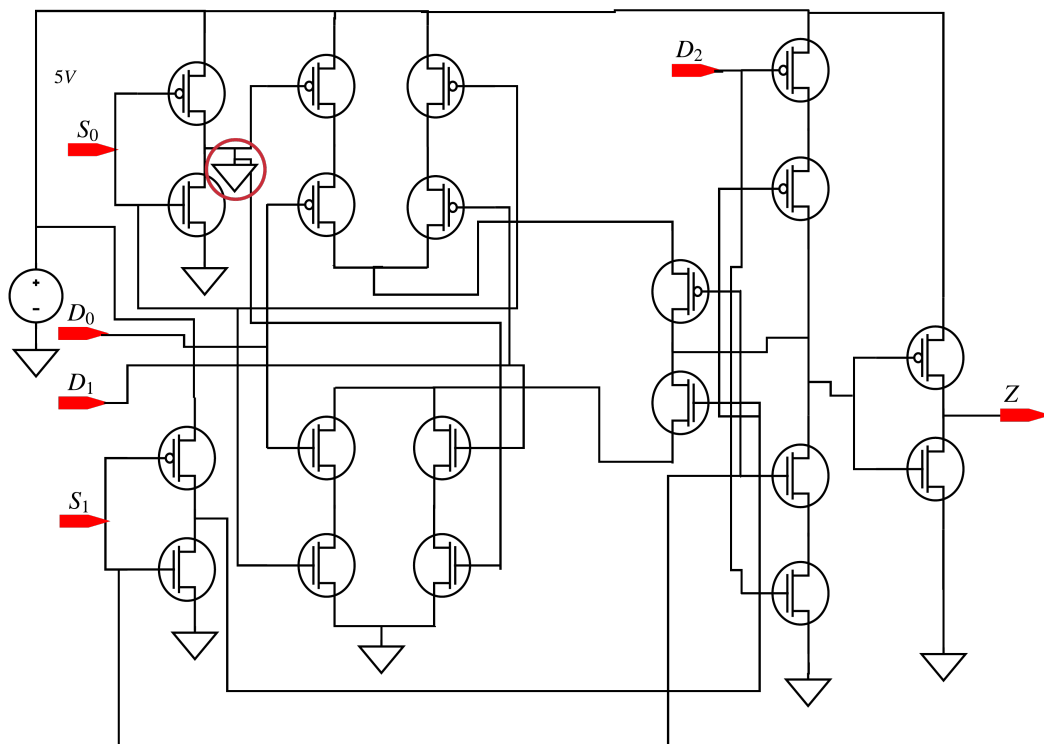
Com os estudos feitos no decorrer deste trabalho, foram vistos conceitos em que correspondências entre os conceitos de área crítica e *Cell-Aware Test* podem ser feitas. A área crítica utiliza os elementos do leiaute como base para os modelos de falhas, já o *Cell-Aware Test* não especifica exatamente como extrai os modelos de falhas. Em Hapke et al. (2014), em que são mostrados os modelos de falhas em *Cell-Aware Test*, mas não são especificados como analisam os pontos do leiaute que serão considerados das falhas. Apenas as falhas correspondentes às *stuck-at*, nas entradas e saídas das portas lógicas, são passíveis de entendimento no que diz respeito aos modelos de falhas baseado no leiaute. Por esse motivo, a área crítica foi estudada e analisada para entender melhor o conceito de modelos de falhas baseados em leiaute. A esse estudo de comparação do modelo de falhas com a área crítica foi dado nesse trabalho o nome de **Análise de Defeitos na Perspectiva do Leiaute por meio de Extração de Elementos Parasitas**. A partir dessa associação de conceitos, tem-se a possibilidade de determinar os pontos específicos do leiaute em que há maior probabilidade de um defeito causar uma falha.

O objetivo das simulações no CADENCE é desenhar o leiaute, obter as resistências e capacitâncias para assim, obter os modelos de falhas e, em seguida, comparar com os modelos do *Cell-Aware Test*. Com isso tem-se a correspondência da **Modelagem de Defeitos na Perspectiva do Leiaute** com o *Cell-Aware Test* (HAPKE et al., 2009), (HAPKE et al., 2011), (HAPKE; SCHLOEFFEL, 2012), (HAPKE et al., 2014).

Para constatar que será possível fazer as simulações do *Cell-Aware Test* utilizou um multiplexador feito pela autora do texto para verificar a possível semelhança entre os resultados de saída. Primeiramente, foi feita a simulação analógica do multiplexador (Fig. 9.1) com 3 entradas, 2 seletores e 1 saída. Sendo assim, foram inseridas todas as 32 possíveis entradas (Fig. 9.2), e as saídas foram analisadas. Após obter as saídas do MUX sem nenhum defeito inserido, pode-se dizer que obteve-se a saída *gold* do MUX. Em seguida, individualmente, os defeitos foram inseridos, e as saídas comparadas com a resposta *gold*.

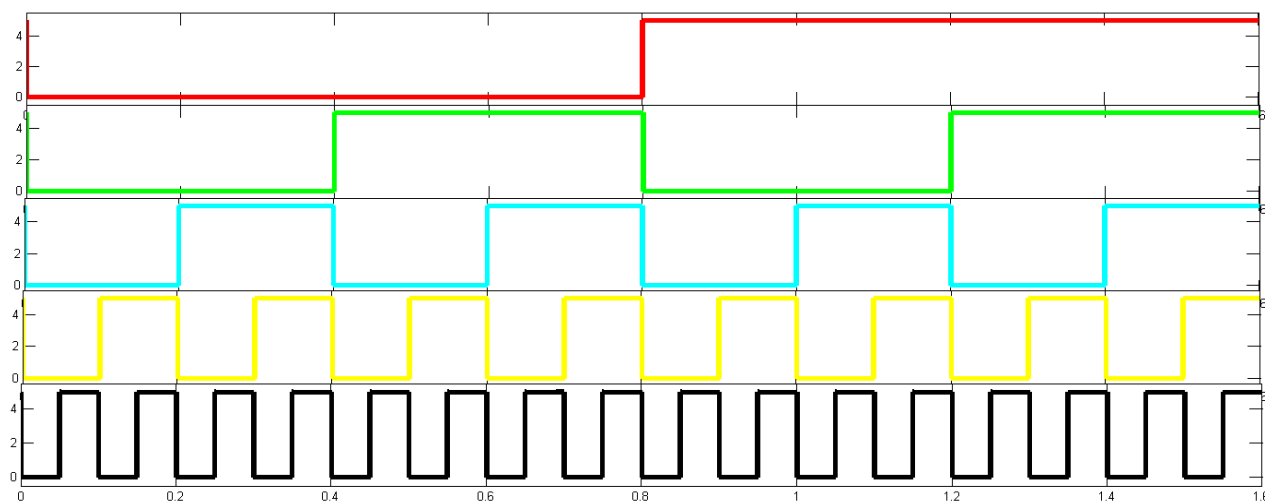
Como exemplo, na Fig. 9.1 pode-se ver que o multiplexador está com um defeito inserido. Esse defeito se dá na saída negada do *S0*, que é a saída *S0N*. Dado esse defeito, a resposta ao estímulo DC em que todas as entradas possíveis (00000 – 11111) são inseridas, é dado o resultado visto na Fig. 9.3. Os resultados em azul são correspondentes à saída

Figura 9.1 – Multiplexador usado como exemplo dado pelo Hapke e implementado no CADENCE com defeito **d1**



Fonte: Elaborada pela autora.

Figura 9.2 – 32 possíveis entradas em $160\mu s$

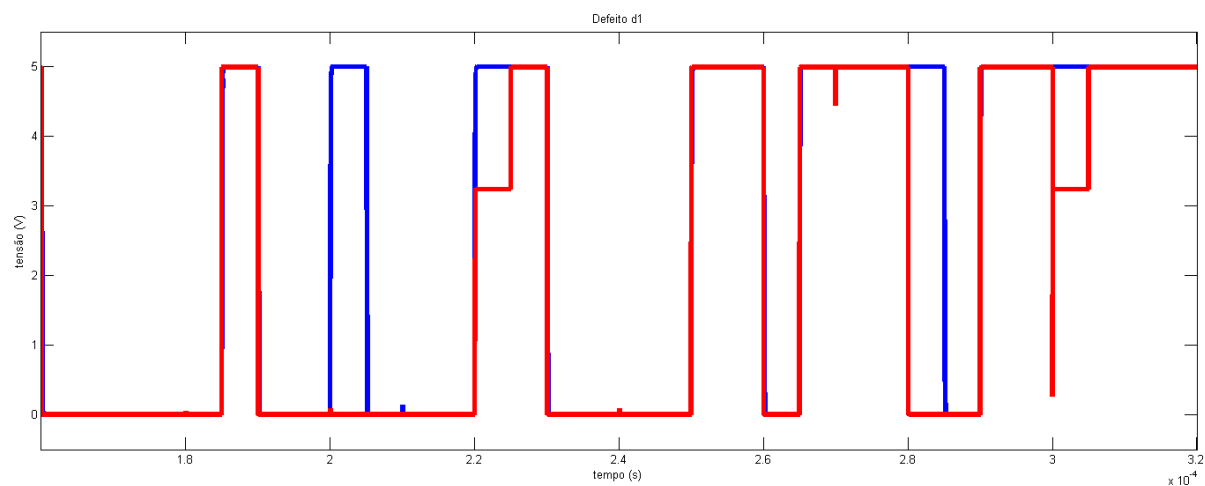


Fonte: Elaborada pela autora.

gold e a vermelho correspondente à saída com o defeito **d1**. Para avaliar se a saída é de nível lógico 0 ou 1, foi utilizado um limiar de 50%, ou seja, se a tensão esperada na saída desviar mais do que 50% do valor esperado, essa saída será considerada falha. No Apêndice

A serão mostrados os resultados de simulação analógicas para os defeitos **d2** - **d48**.

Figura 9.3 – Resultado da simulação analógica com defeito **d1** (S0N, gnd).



Fonte: Elaborada pela autora.

Na Tabela 9.1 são mostrados os resultados obtidos ao inserir todos os defeitos, um a um, na simulação analógica. Tendo como resultado uma equivalência com os resultados obtidos em Hapke et al. (2009). Dado que somente uma parte do resultado foi mostrado em Hapke et al. (2009), todos os dados passíveis de comparação foram iguais aos obtidos neste trabalho.

Tabela 9.1 – Matriz de Defeitos obtidas por Simulação no CADENCE

	Entrada	S/ falhas	d37	d38	d39	d40	d41	d42	d43	d44	d45	d46	d47	d48
1	00000	0	0	0	0	0	0	0	0	1	0	0	0	0
2	00001	0	0	0	0	0	0	1	0	1	0	0	0	0
3	00010	0	1	0	0	0	0	0	0	1	0	0	0	0
4	00011	0	1	0	0	0	0	0	0	1	0	0	0	0
5	00100	0	0	0	0	0	0	0	0	1	0	0	0	0
6	00101	1	1	0	0	0	0	1	0	1	0	1	0	0
7	00110	0	1	0	0	0	0	0	0	1	0	0	0	0
8	00111	0	1	0	0	0	0	0	0	1	0	0	0	0
9	01000	1	1	0	0	0	0	1	0	1	0	0	0	1
10	01001	0	0	0	0	0	0	1	0	1	0	0	0	0
11	01010	0	1	0	0	0	0	0	0	1	0	0	0	0
12	01011	0	1	0	0	0	0	0	0	1	0	0	0	0
13	01100	1	1	1	1	1	1	1	0	1	0	0	0	1
14	01101	1	1	1	1	0	0	1	0	1	0	1	0	1
15	01110	0	1	0	0	0	0	0	0	1	0	0	0	0
16	01111	0	1	0	0	0	0	0	0	1	0	0	0	0
17	10000	0	0	0	0	0	0	0	0	1	0	0	0	0
18	10001	0	0	0	0	0	0	1	0	1	0	0	0	0
19	10010	1	1	1	0	0	0	1	1	1	1	0	1	0
20	10011	1	1	1	0	0	Indef	1	1	1	1	1	1	0
21	10100	0	0	0	0	0	0	0	0	1	0	0	0	0
22	10101	1	1	0	0	Indef	Indef	1	0	1	0	1	1	0
23	10110	1	1	1	1	0	0	1	1	1	1	0	1	0
24	10111	1	1	1	0	0	0	1	1	1	1	1	1	0
25	11000	1	1	0	0	0	0	1	0	1	0	0	1	1
26	11001	0	0	0	0	0	Indef	1	1	1	0	0	0	0
27	11010	1	1	1	0	0	1	1	1	1	1	0	1	1
28	11011	1	1	1	0	0	Indef	1	1	1	1	1	1	1
29	11100	1	1	1	1	1	1	1	0	1	0	0	1	1
30	11101	1	1	1	0	Indef	Indef	1	0	1	0	1	1	1
31	11110	1	1	1	1	0	1	1	1	1	1	0	1	1
32	11111	1	1	1	0	0	Indef	1	1	1	1	1	1	1

Conclusão

O processo de testes é de grande importância na fabricação de circuitos integrados. Com a miniaturização e aumento do número de transistores em uma menor área, o foco na área de testes passou a ser as novas falhas que essa situação atual traz. Uma recente metodologia de testes, proposta pela Mentor, é o *Cell-Aware Test* que obteve resultados interessantes no que diz respeito à cobertura de falhas, porém não é claro como os valores dos parâmetros dos modelos de falhas são extraídos do leiaute. O conhecimento de como esses modelos de falhas são analisados é importante para a aplicação dessa metodologia em outros cenários, por exemplo, com ferramentas CADENCE.

Dessa forma, foi realizado neste trabalho um estudo aprofundado de metodologias de testes que exploram o leiaute do CI e foram aplicados diferentes conceitos, tais como: área crítica, densidade de defeito, densidade de probabilidade de defeito, a fim de ser possível aplicar conhecimentos do leiaute de um CI específico, extrair elementos parasitas que dependendo dos valores apresentam regiões do leiaute com alta probabilidade de falhas e assim modelar falhas somente nessas regiões usando a metodologia *Cell-Aware Test*. Foram utilizadas ferramentas da CADENCE para a extração de elementos parasitas do leiaute. Desse processo, foi possível analisar como as capacitâncias influenciam nas regiões e nas suas probabilidades de ocorrerem falhas.

Em seguida, ao fazer uma análise mais aprofundada do leiaute pela extração de elementos parasitas foi possível observar que, ao fazer pequenas alterações, era possível melhorar os valores de capacitâncias do leiaute e modificar a probabilidade de falha. Nesse estudo foi proposto um mapeamento do tipo *heat map* com os valores de capacitâncias e obter as regiões de maiores probabilidades de falhas. Essa contribuição é de grande importância para que o projetista perceba como o leiaute se comportará na presença de um defeito físico que, se necessário, o profissional poderá sugerir mudanças ao projetista para melhorar o leiaute como um todo e aperfeiçoar o processo de teste.

Referências

ALLAN, G. A comparison of efficient dot throwing and shape shifting extra material critical area estimation. In: . IEEE Comput. Soc, 1998. p. 44–52. ISBN 978-0-8186-8832-4. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=732150>>. Citado na página 22.

ALLAN, G.; WALTON, A. Efficient critical area estimation for arbitrary defect shapes. In: . IEEE Comput. Soc, 1997. p. 20–28. ISBN 978-0-8186-8168-4. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=628305>>. Citado 3 vezes nas páginas 8, 17 e 34.

ALLAN, G.; WALTON, A. Efficient critical area estimation for arbitrary defect shapes. In: . **IEEE Comput. Soc**, 1997. p. 20–28. ISBN 978-0-8186-8168-4. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=628305>>. Citado na página 52.

ALLAN, G.; WALTON, A. Critical area extraction for soft fault estimation. *IEEE Transactions on Semiconductor Manufacturing*, v. 11, n. 1, p. 146–154, fev. 1998. ISSN 08946507. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=661294>>. Citado 6 vezes nas páginas 8, 17, 34, 35, 36 e 52.

ALLAN, G.; WALTON, A. Efficient extra material critical area algorithms. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, v. 18, n. 10, p. 1480–1486, out. 1999. ISSN 02780070. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=790624>>. Citado na página 17.

ALLAN, G.; WALTON, A. Efficient extra material critical area algorithms. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, v. 18, n. 10, p. 1480–1486, out. 1999. ISSN 02780070. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=790624>>. Citado na página 52.

BUBEL, I. et al. AFFCCA: a tool for critical area analysis with circular defects and lithography deformed layout. In: . IEEE Comput. Soc. Press, 1995. p. 10–18. ISBN 978-0-8186-7107-4. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=476932>>. Citado 6 vezes nas páginas 8, 17, 33, 34, 35 e 36.

BUBEL, I. et al. AFFCCA: a tool for critical area analysis with circular defects and lithography deformed layout. In: . **IEEE Comput. Soc. Press**, 1995. p. 10–18. ISBN 978-0-8186-7107-4. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=476932>>. Citado na página 51.

Chao-Wen Tzeng; Han-Chia Cheng; Shi-Yu Huang. Layout-Based Defect-Driven Diagnosis for Intracell Bridging Defects. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, v. 28, n. 5, p. 764–769, maio 2009. ISSN 0278-0070, 1937-4151. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4838835>>. Citado na página 17.

- CHENG, W.-T. et al. Enhancing Transition Fault Model for Delay Defect Diagnosis. In: . IEEE, 2008. p. 179–184. ISBN 978-0-7695-3396-4. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4711579>>. Citado 2 vezes nas páginas 23 e 24.
- CLIFFS, E. *Fault Tolerance in Distributed System*. New Jersey: Prentice Hall, 1994. Citado na página 14.
- COX, H.; RAJSKI, J. Stuck-open and transition fault testing in CMOS complex gates. In: . IEEE Comput. Soc. Press, 1988. p. 688–694. ISBN 978-0-8186-0870-4. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=207853>>. Citado na página 15.
- FALHA stuck-at. Disponível em: <http://ece-research.unm.edu/jimp/vlsi_test/slides/html/faults2.html>. Nenhuma citação no texto.
- FOUNDRY, I. *CMOS 7RF (CMRF7SF) 5.0 V Standard Cell Databook*. 2014. Citado na página 54.
- GARCIA, R. *Rethink fault models for submicron-IC test*. 2001. Disponível em: <<http://www.edn.com/design/test-and-measurement/4382876/Rethink-fault-models-for-submicron-IC-test>>. Citado 2 vezes nas páginas 8 e 16.
- GKATZIANI, M. et al. Accurately Determining Bridging Defects from Layout. In: . IEEE, 2007. p. 1–4. ISBN 978-1-4244-1161-0. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4295259>>. Citado 4 vezes nas páginas 37, 38, 40 e 43.
- HAPKE, F. et al. Defect-oriented cell-aware ATPG and fault simulation for industrial cell libraries and designs. In: *Test Conference, 2009. ITC 2009. International*. [S.l.: s.n.], 2009. p. 1–10. Citado 9 vezes nas páginas 17, 19, 20, 25, 27, 30, 31, 67 e 69.
- HAPKE, F. et al. Cell-Aware Test. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, v. 33, n. 9, p. 1396–1409, set. 2014. ISSN 0278-0070, 1937-4151. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6879635>>. Citado 8 vezes nas páginas 17, 18, 19, 22, 26, 31, 32 e 67.
- HAPKE, F. et al. Defect-oriented cell-internal testing. In: *Test Conference (ITC), 2010 IEEE International*. [S.l.: s.n.], 2010. p. 1–10. Citado 2 vezes nas páginas 17 e 23.
- HAPKE, F.; SCHLOEFFEL, J. Introduction to the defect-oriented cell-aware test methodology for significant reduction of DPPM rates. In: . IEEE, 2012. p. 1–6. ISBN 978-1-4673-0697-3 978-1-4673-0696-6 978-1-4673-0695-9. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6233046>>. Citado 5 vezes nas páginas 19, 26, 28, 29 e 67.
- HAPKE, F. et al. Gate-Exhaustive and Cell-Aware pattern sets for industrial designs. In: . IEEE, 2011. p. 1–4. ISBN 978-1-4244-8500-0. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5783604>>. Citado 2 vezes nas páginas 17 e 67.
- HESS, C.; WEILAND, L. Wafer level defect density distribution using checkerboard test structures. In: . IEEE, 1998. p. 101–106. ISBN 978-0-7803-4348-1. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=688050>>. Citado 2 vezes nas páginas 38 e 39.

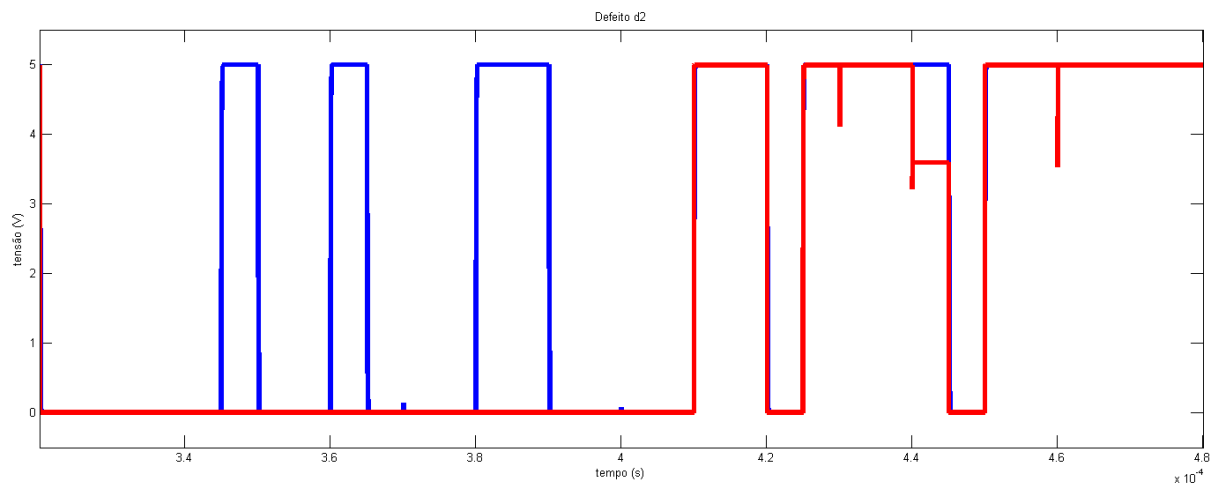
- HURST, S. L. *VLSI Testing - digital and mixed analogue/digital techniques*. United Kingdom: **IEE**, 1998. Citado na página 15.
- IBM, M. S. T. D. *Assura Physical Verification Command Reference*. 2014. Citado na página 56.
- IBM, M. S. T. D. *Assura Physical Verification User Guide*. 2014. Citado 2 vezes nas páginas 49 e 56.
- JACOMET, M.; GUGGENBUHL, W. Layout-dependent fault analysis and test synthesis for CMOS circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, v. 12, n. 6, p. 888–899, jun. 1993. ISSN 02780070. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=229763>>. Citado 2 vezes nas páginas 17 e 33.
- MEI, K. Bridging and Stuck-At Faults. *IEEE Transactions on Computers*, C-23, n. 7, p. 720–727, jul. 1974. ISSN 0018-9340. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1672613>>. Citado na página 22.
- MOSHQAQ KEYVAN ALIREZA ZAD, E. M. *Metallization*. 2016. Citado na página 47.
- PLUSQUELLIC, J. *Design For Testability*. 2006. Disponível em: <http://ece-research.unm.edu/jimp/vlsi_test/slides/dft_scan1.pdf>. Citado na página 15.
- PLUSQUELLIC, J. *Falha stuck-at*. 2006. Disponível em: <http://ece-research.unm.edu/jimp/vlsi_test/slides/faults2.pdf>. Citado na página 15.
- ROBERTO, B. et al. *Transistor e sua evolução*. 2009. Disponível em: <<http://pt.slideshare.net/brunodemetrio/transstor-e-sua-evolucao>>. Citado na página 14.
- SILVA, J. M. da. *Fabricação CMOS: Passos Básicos*. 2007. Disponível em: <<http://paginas.fe.up.pt/~vgt/Disciplinas/E3/tecnologia.pdf>>. Citado 4 vezes nas páginas 44, 45, 46 e 48.
- SOUZA, C. *Uma Arquitetura Autotestável para Circuitos Digitais Baseada no Algoritmo de Berlekamp-Massey e em Sistemas Imunológicos Artificiais*. Tese (Doutorado) — Universidade Federal de Campina Grande, Campina Grande, 2005. Citado 2 vezes nas páginas 14 e 15.
- STANOJEVIC, Z.; WALKER, D. FedEx - a fast bridging fault extractor. In: . IEEE, 2001. p. 696–703. ISBN 978-0-7803-7169-9. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=966690>>. Nenhuma citação no texto.
- WAICUKAUSKI, J. et al. Transition Fault Simulation. *IEEE Design & Test of Computers*, v. 4, n. 2, p. 32–38, 1987. ISSN 0740-7475. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4069962>>. Citado na página 16.
- WU, C.-H.; LEE, K.-J. An Efficient Diagnosis Pattern Generation Procedure to Distinguish Stuck-at Faults and Bridging Faults. In: . IEEE, 2014. p. 306–311. ISBN 978-1-4799-6030-9. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6979118>>. Citado 2 vezes nas páginas 8 e 23.

Apêndices

APÊNDICE A – Resultados de Simulação

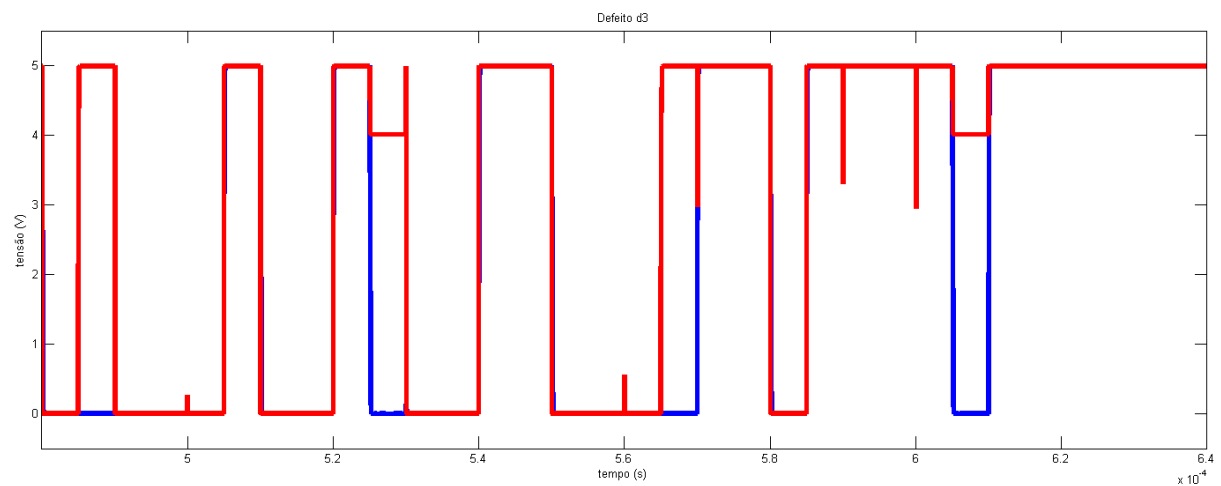
Nas Figs. A.1 - A.47 são mostrados os resultados de simulação analógica para os defeitos **d2** - **d48**. Em que a saída em azul ou amarelo corresponde à saída sem defeito e a saída em vermelho a resposta ao defeito.

Figura A.1 – Resultado da simulação analógica com o defeito **d2**.

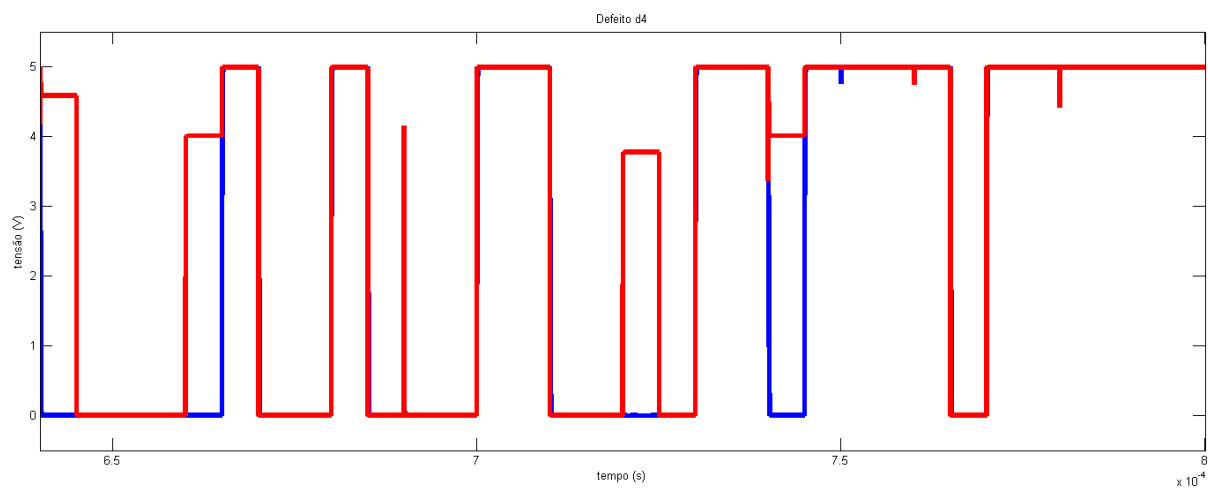


Fonte: Elaborada pela autora.

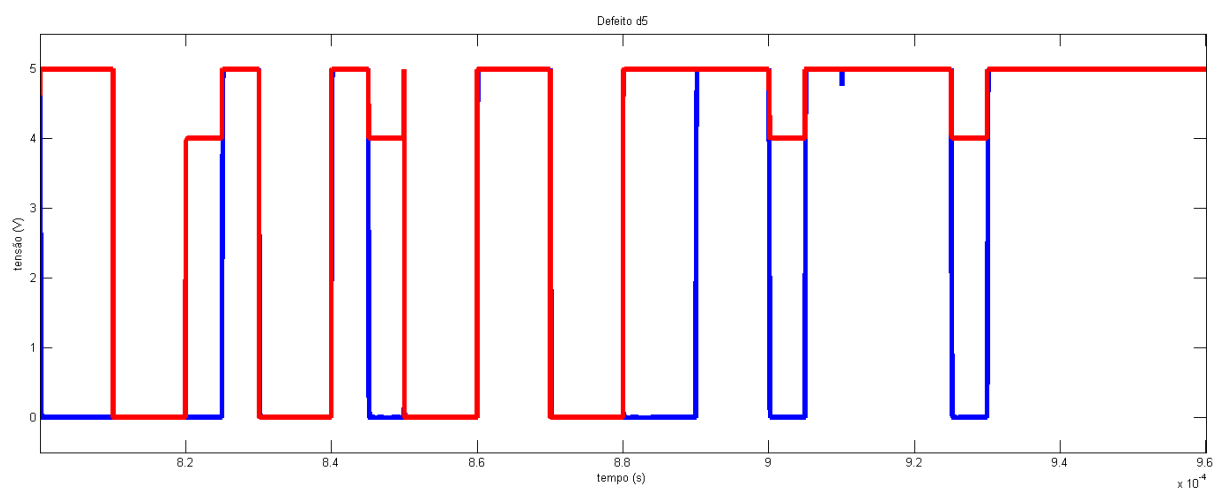
Figura A.2 – Resultado da simulação analógica com o defeito **d3**.



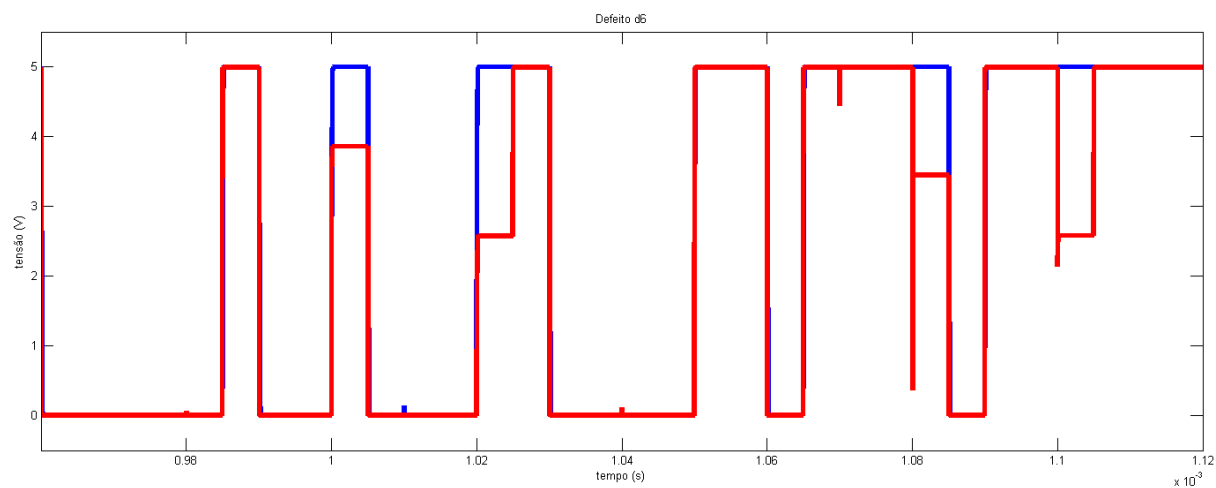
Fonte: Elaborada pela autora.

Figura A.3 – Resultado da simulação analógica com o defeito **d4**.

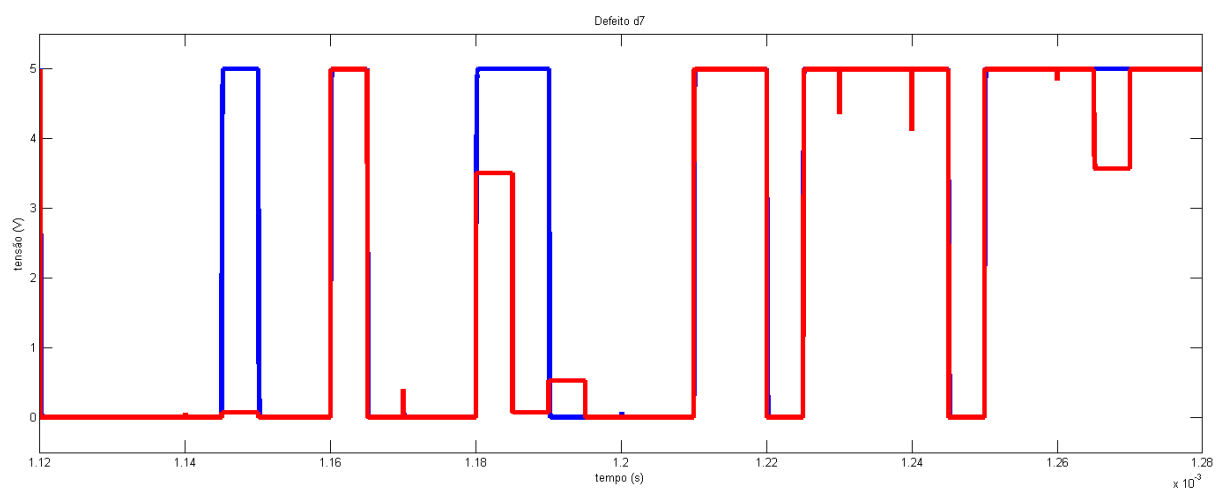
Fonte: Elaborada pela autora.

Figura A.4 – Resultado da simulação analógica com o defeito **d5**.

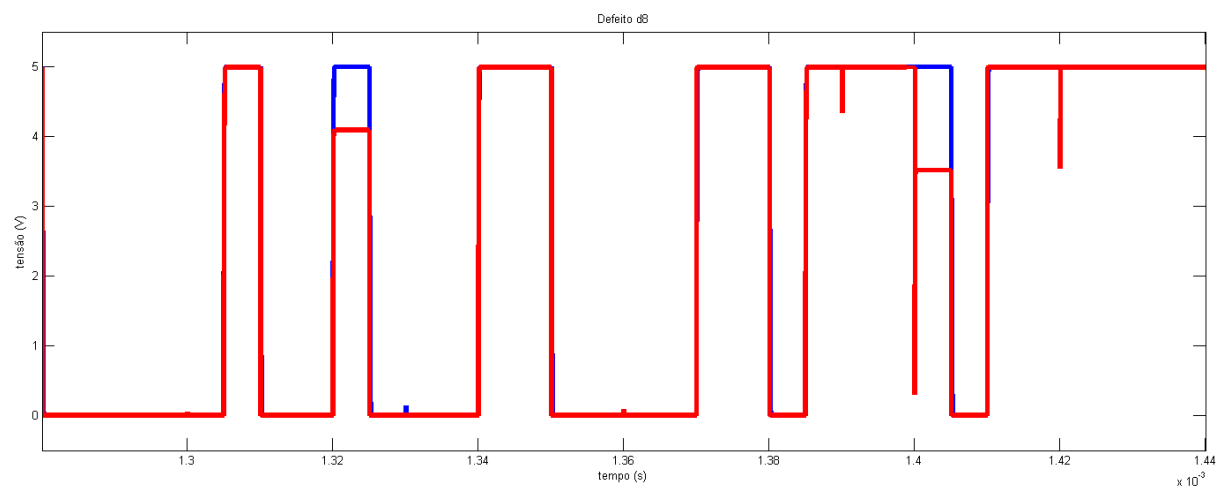
Fonte: Elaborada pela autora.

Figura A.5 – Resultado da simulação analógica com o defeito **d6**.

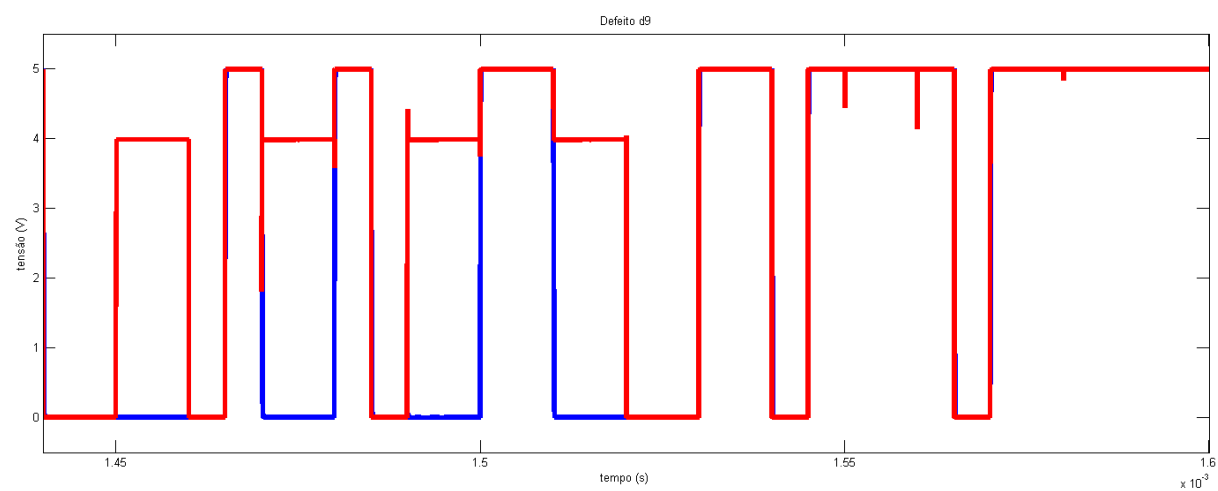
Fonte: Elaborada pela autora.

Figura A.6 – Resultado da simulação analógica com o defeito **d7**.

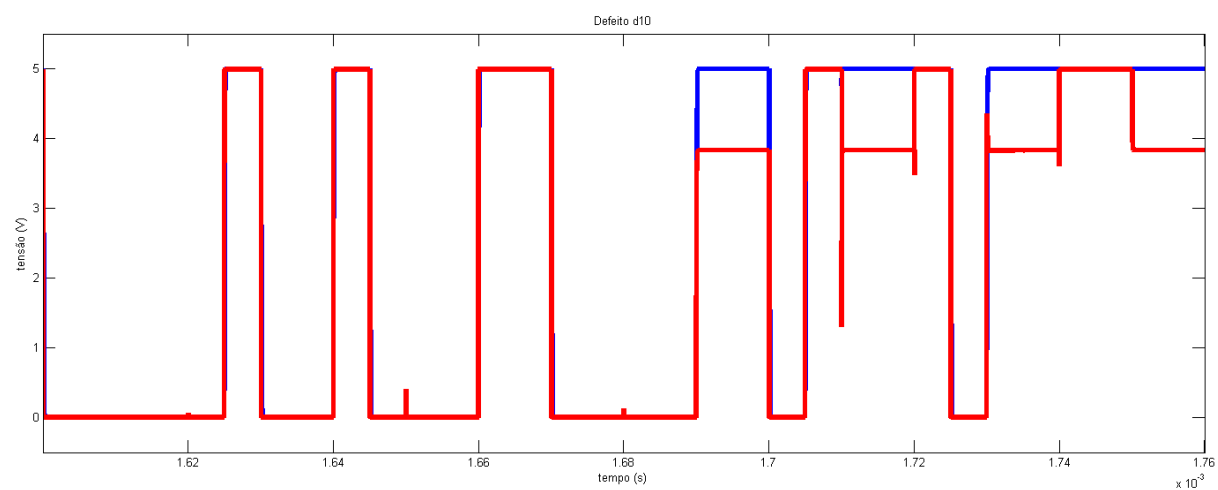
Fonte: Elaborada pela autora.

Figura A.7 – Resultado da simulação analógica com o defeito **d8**.

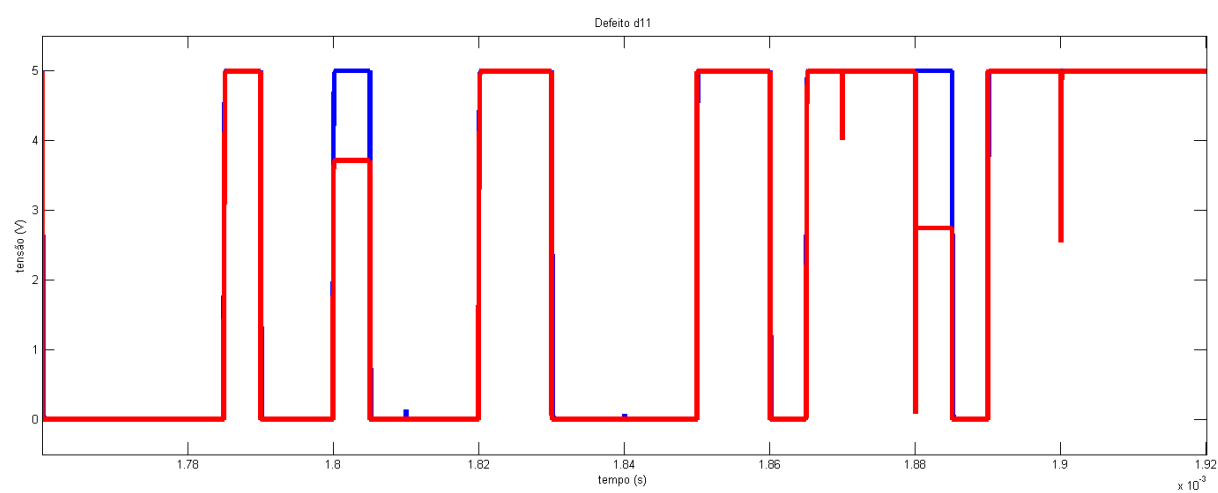
Fonte: Elaborada pela autora.

Figura A.8 – Resultado da simulação analógica com o defeito **d9**.

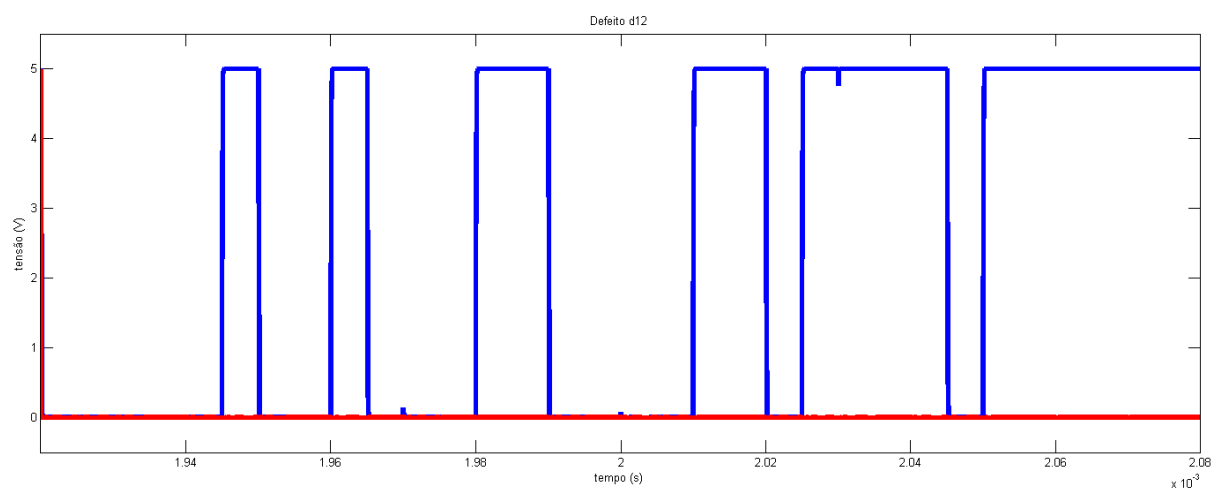
Fonte: Elaborada pela autora.

Figura A.9 – Resultado da simulação analógica com o defeito **d10**.

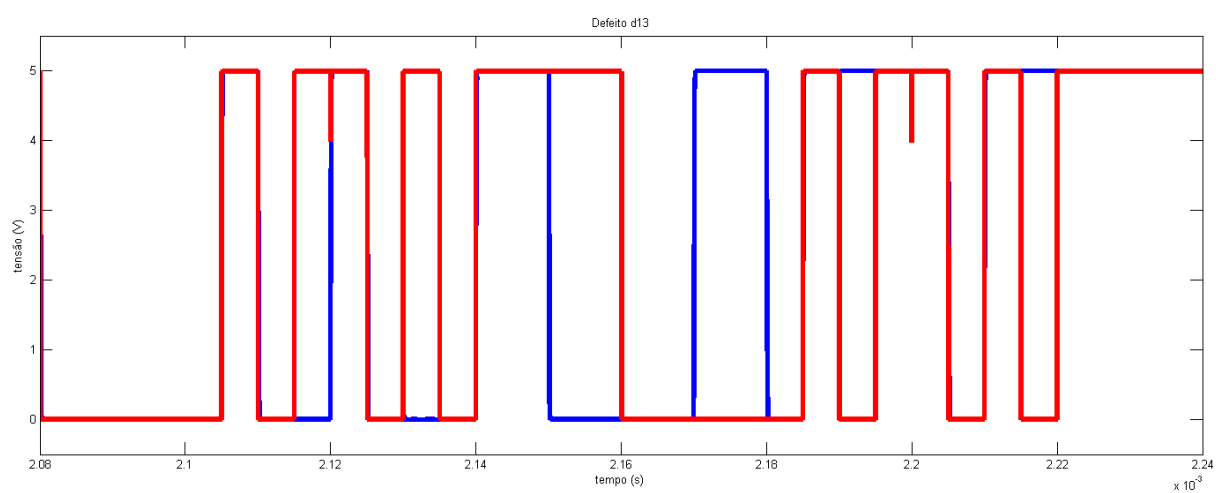
Fonte: Elaborada pela autora.

Figura A.10 – Resultado da simulação analógica com o defeito **d11**.

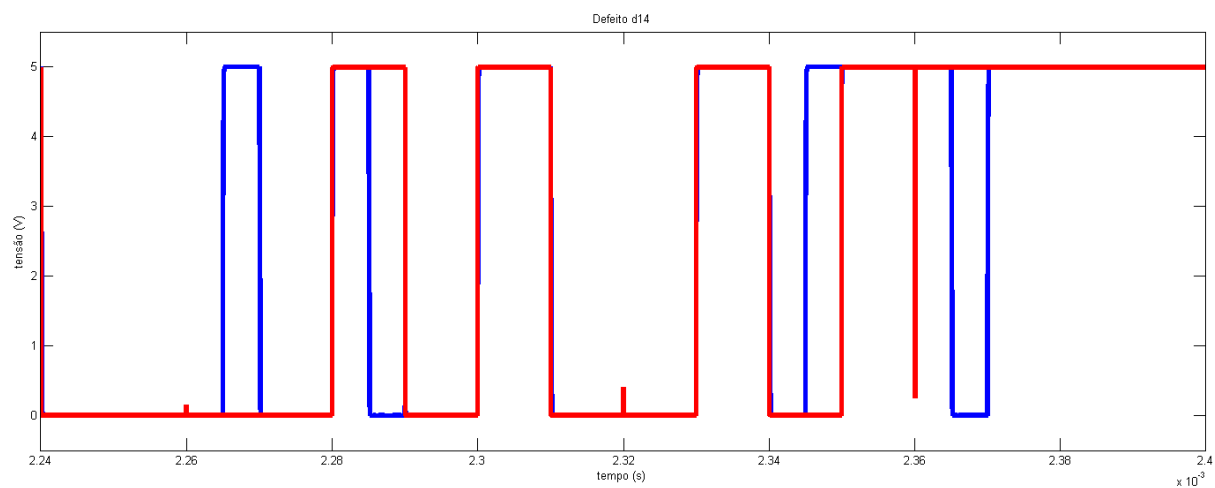
Fonte: Elaborada pela autora.

Figura A.11 – Resultado da simulação analógica com o defeito **d12**.

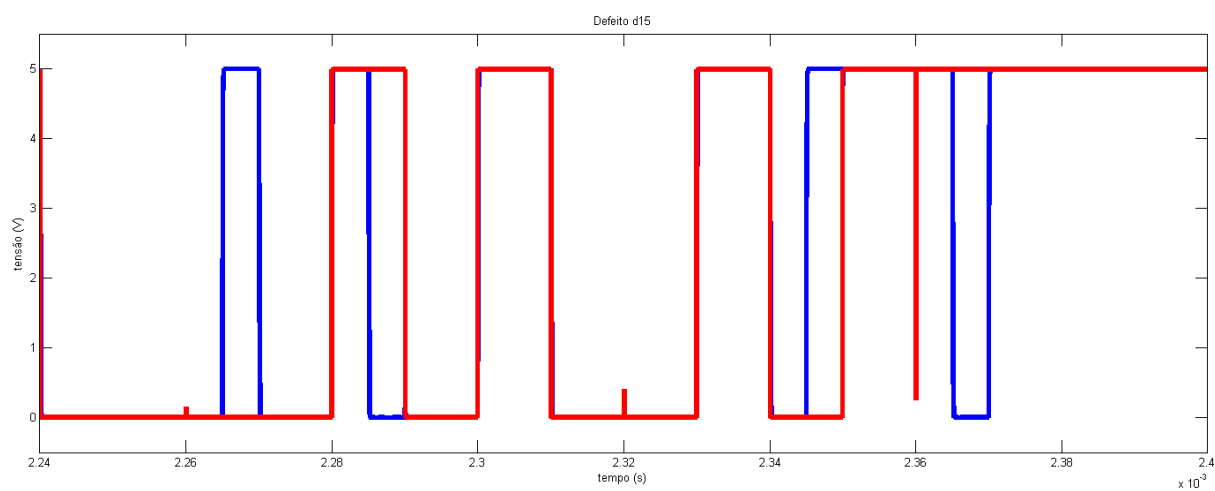
Fonte: Elaborada pela autora.

Figura A.12 – Resultado da simulação analógica com o defeito **d13**.

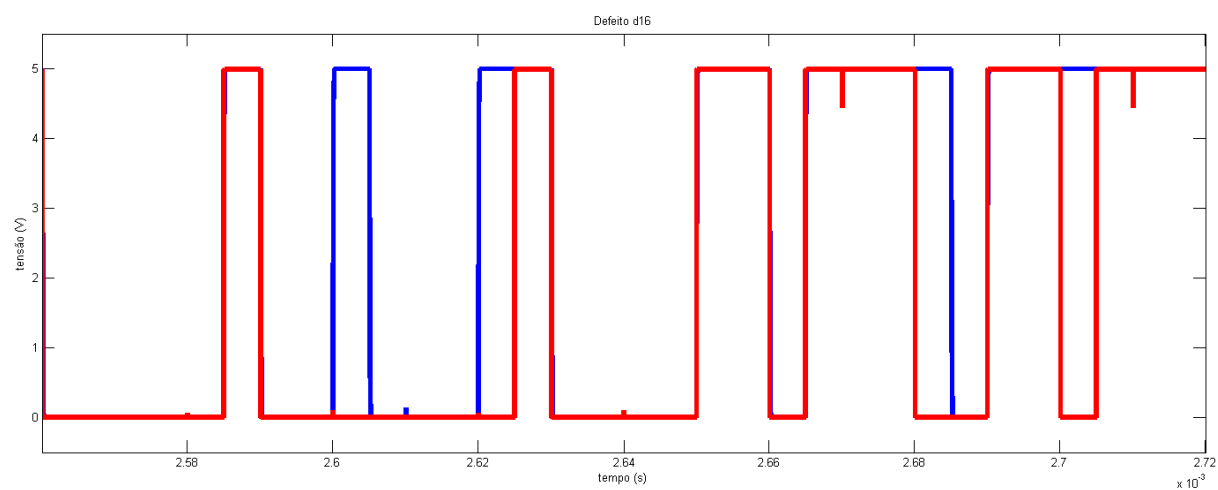
Fonte: Elaborada pela autora.

Figura A.13 – Resultado da simulação analógica com o defeito **d14**.

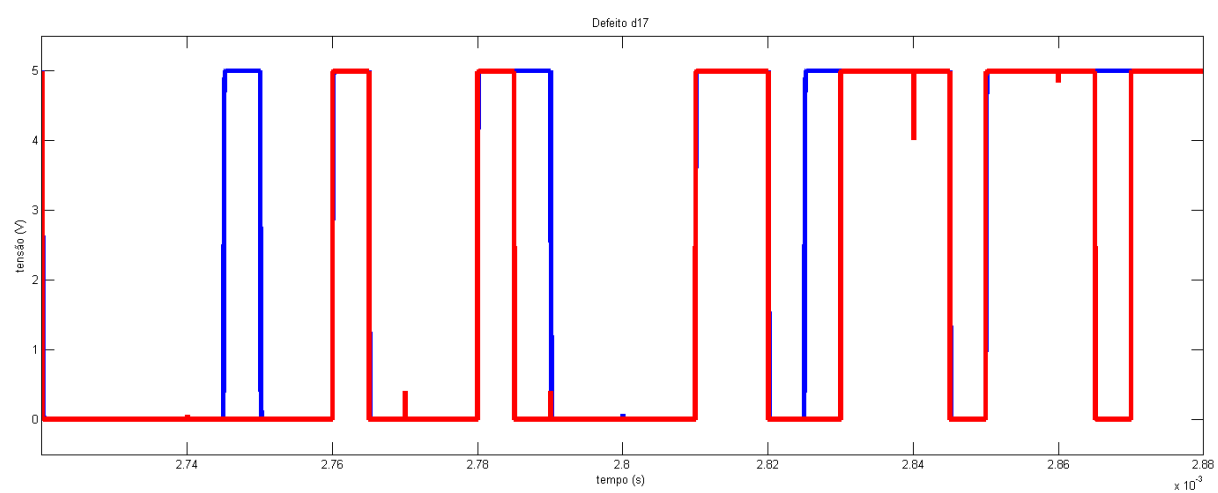
Fonte: Elaborada pela autora.

Figura A.14 – Resultado da simulação analógica com o defeito **d15**.

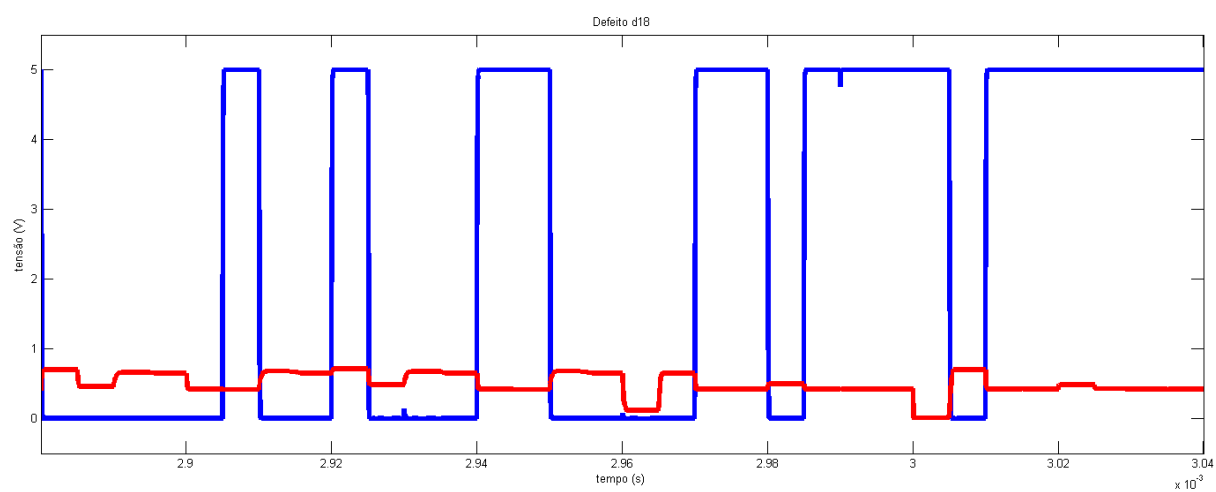
Fonte: Elaborada pela autora.

Figura A.15 – Resultado da simulação analógica com o defeito **d16**.

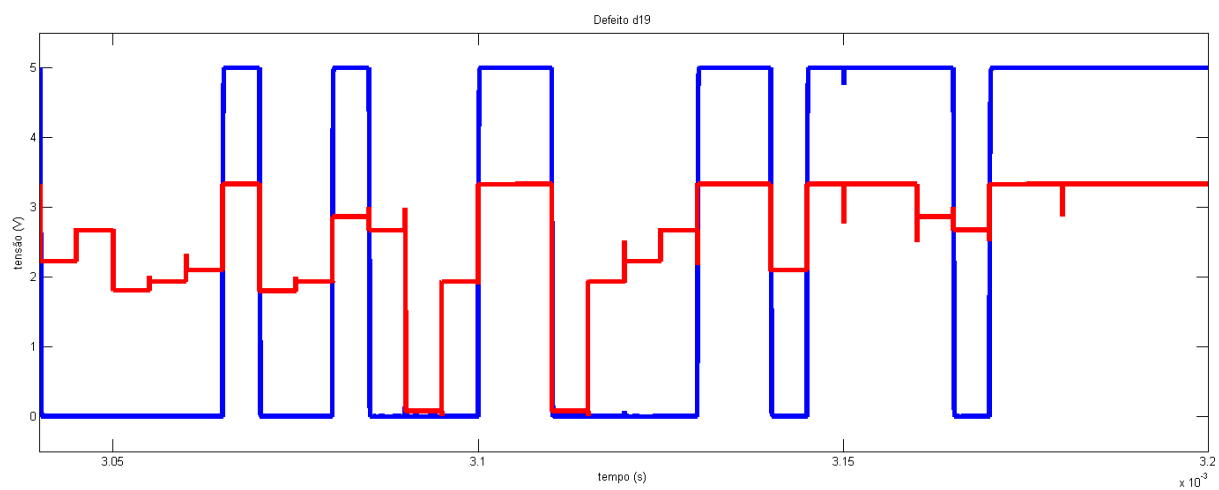
Fonte: Elaborada pela autora.

Figura A.16 – Resultado da simulação analógica com o defeito **d17**.

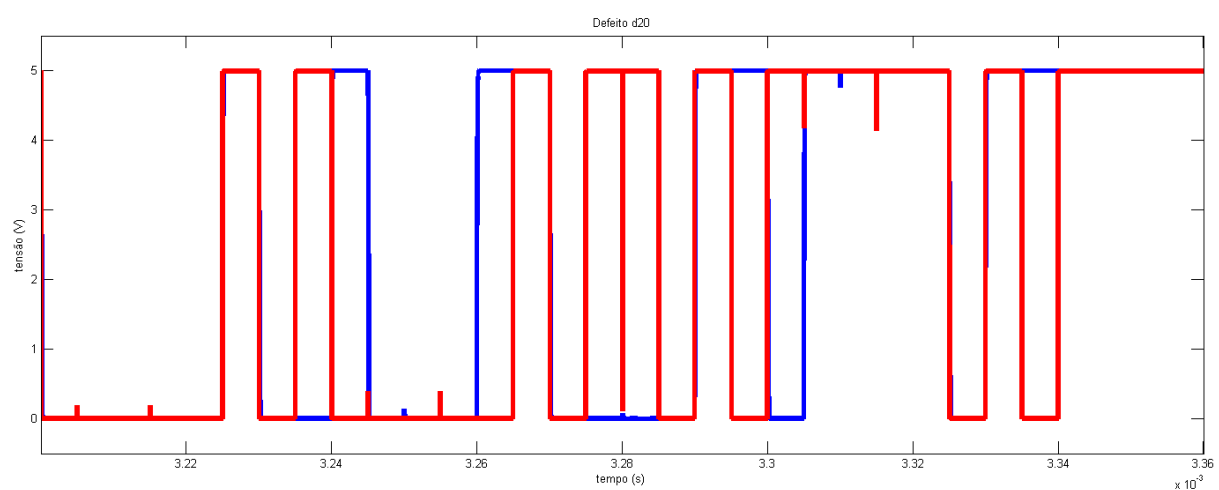
Fonte: Elaborada pela autora.

Figura A.17 – Resultado da simulação analógica com o defeito **d18**.

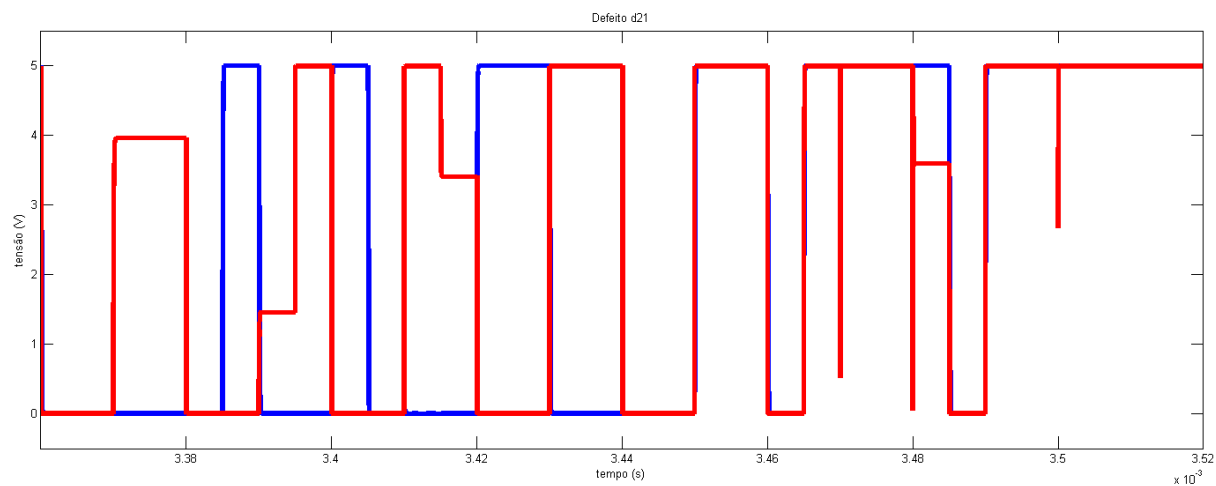
Fonte: Elaborada pela autora.

Figura A.18 – Resultado da simulação analógica com o defeito **d19**.

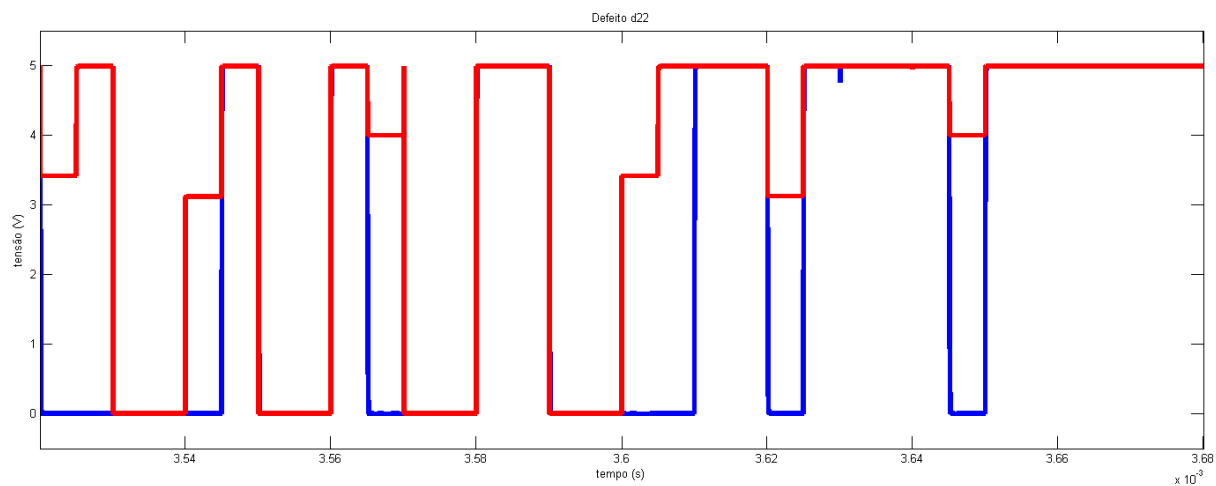
Fonte: Elaborada pela autora.

Figura A.19 – Resultado da simulação analógica com o defeito **d20**.

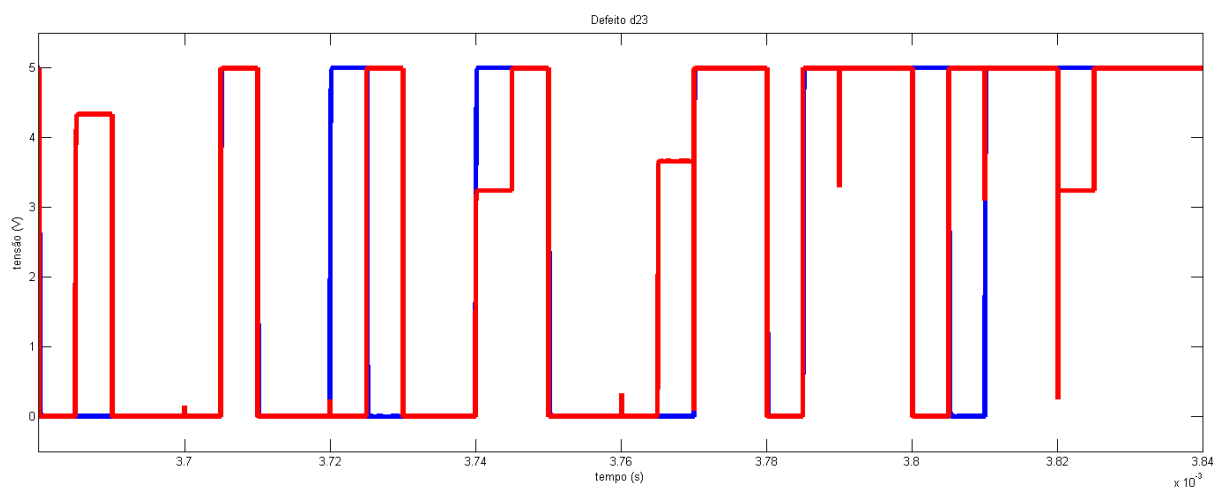
Fonte: Elaborada pela autora.

Figura A.20 – Resultado da simulação analógica com o defeito **d21**.

Fonte: Elaborada pela autora.

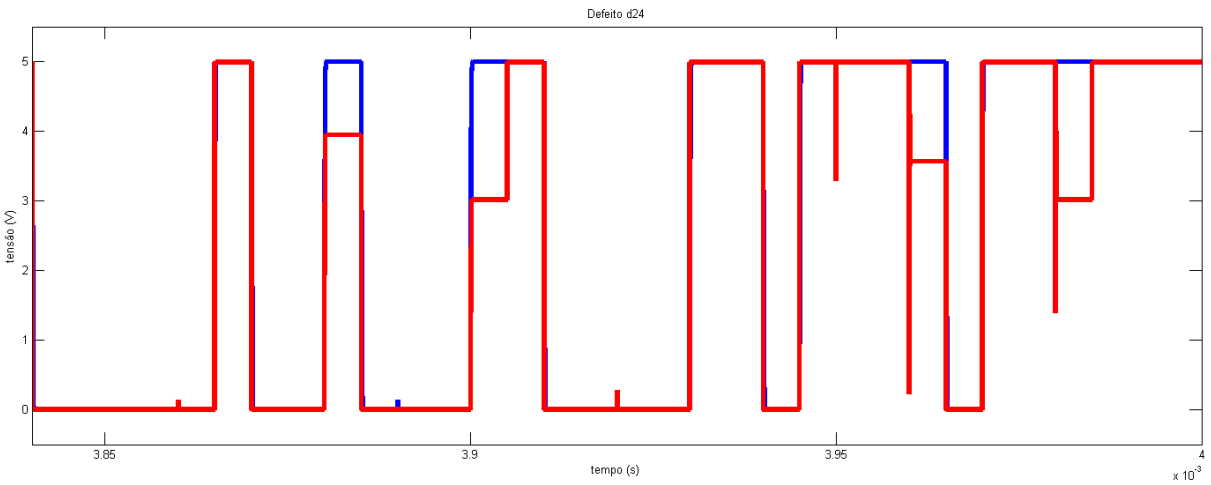
Figura A.21 – Resultado da simulação analógica com o defeito **d22**.

Fonte: Elaborada pela autora.

Figura A.22 – Resultado da simulação analógica com o defeito **d23**.

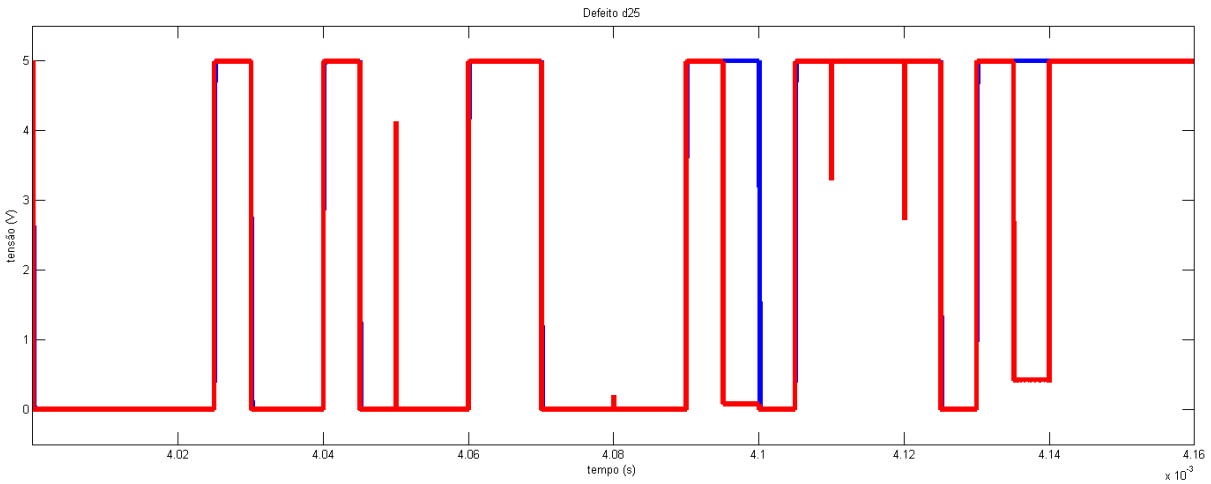
Fonte: Elaborada pela autora.

Figura A.23 – Resultado da simulação analógica com o defeito **d24**.



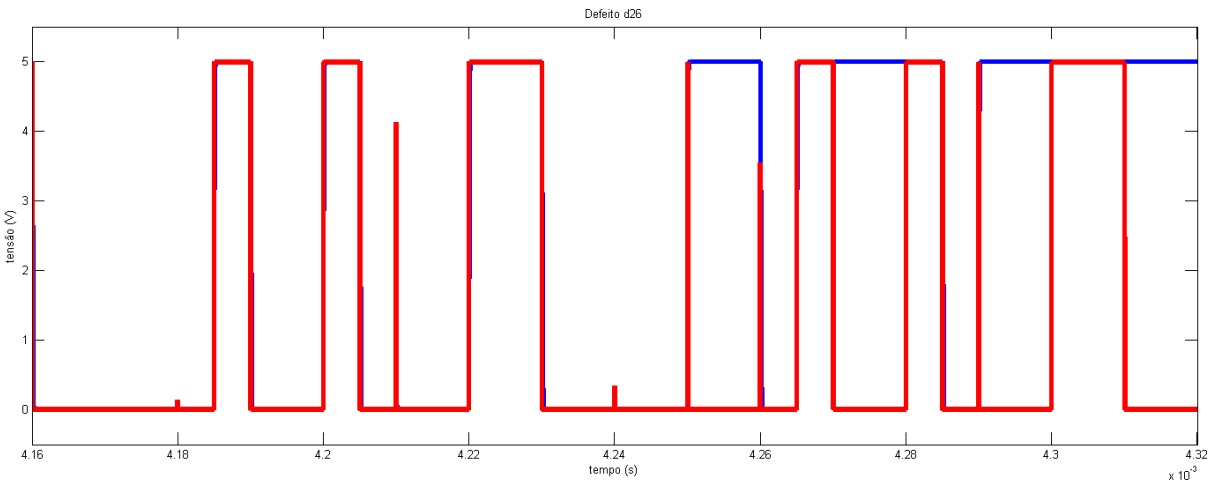
Fonte: Elaborada pela autora.

Figura A.24 – Resultado da simulação analógica com o defeito **d25**.



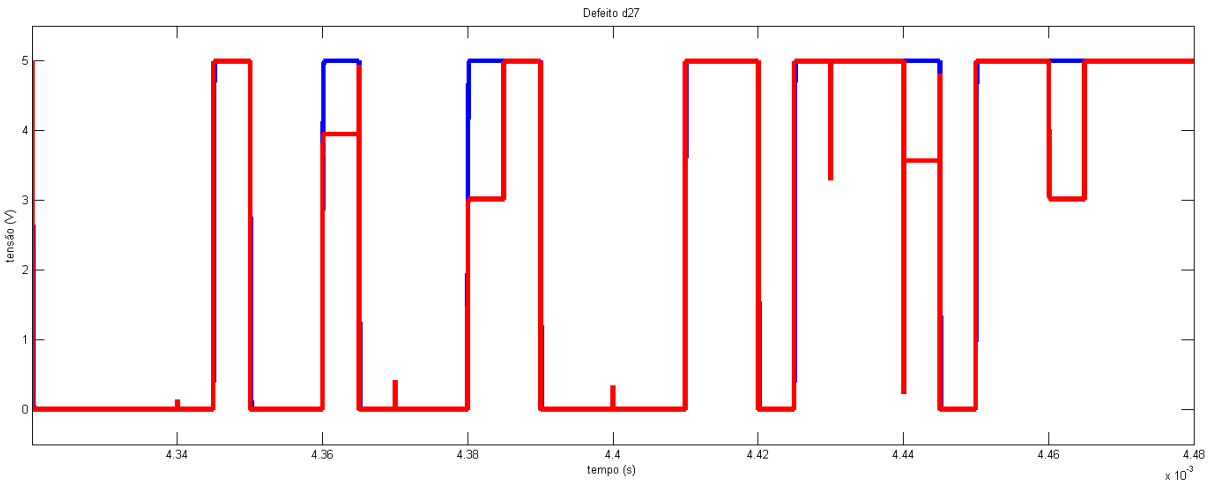
Fonte: Elaborada pela autora.

Figura A.25 – Resultado da simulação analógica com o defeito **d26**.



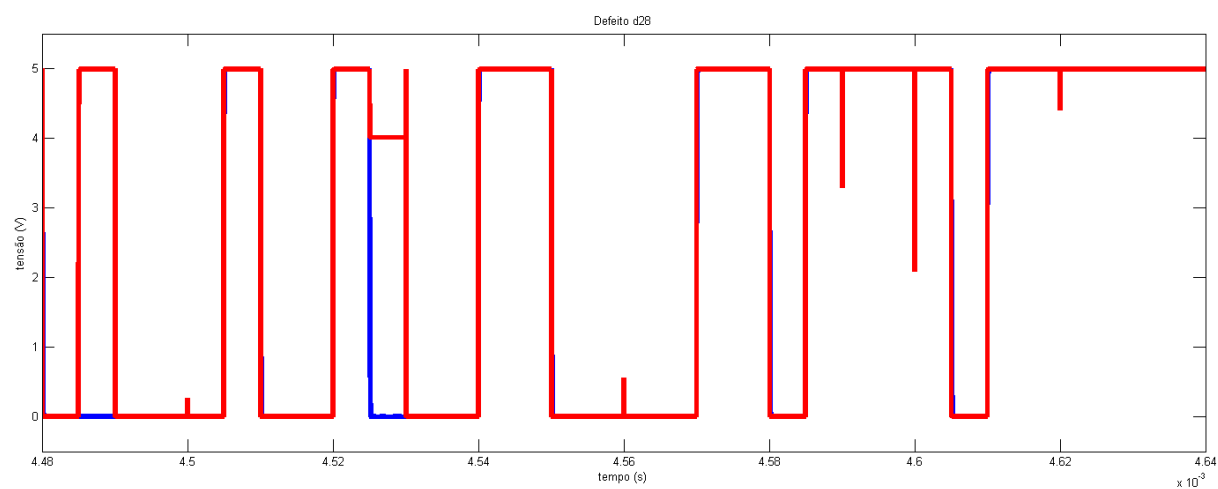
Fonte: Elaborada pela autora.

Figura A.26 – Resultado da simulação analógica com o defeito **d27**.



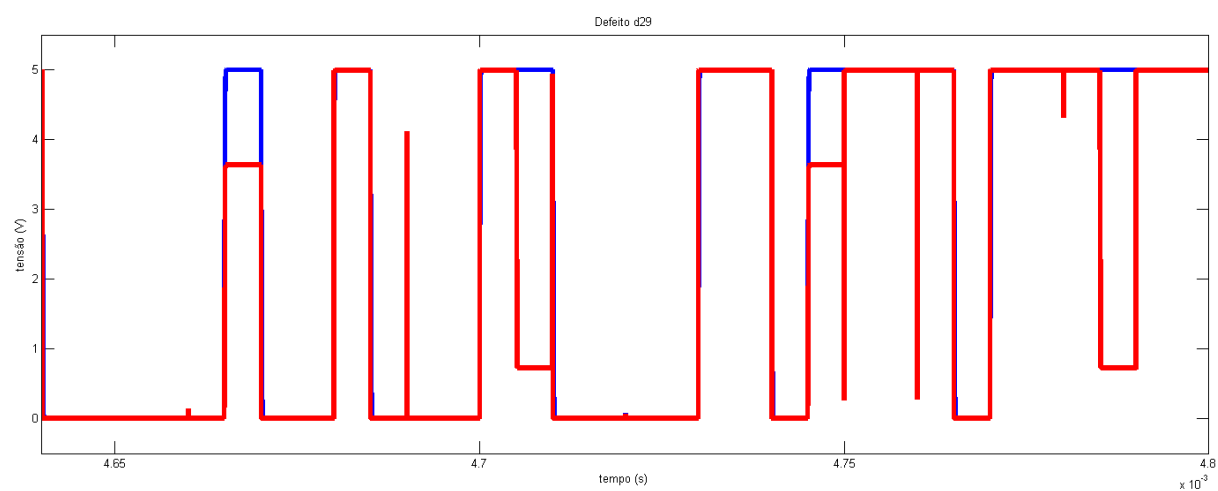
Fonte: Elaborada pela autora.

Figura A.27 – Resultado da simulação analógica com o defeito **d28**.

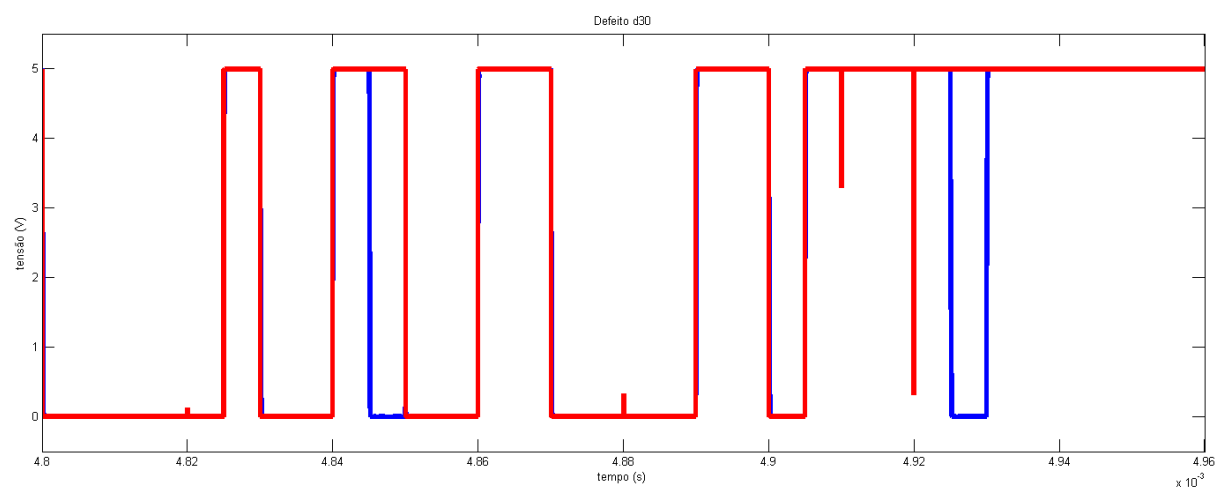


Fonte: Elaborada pela autora.

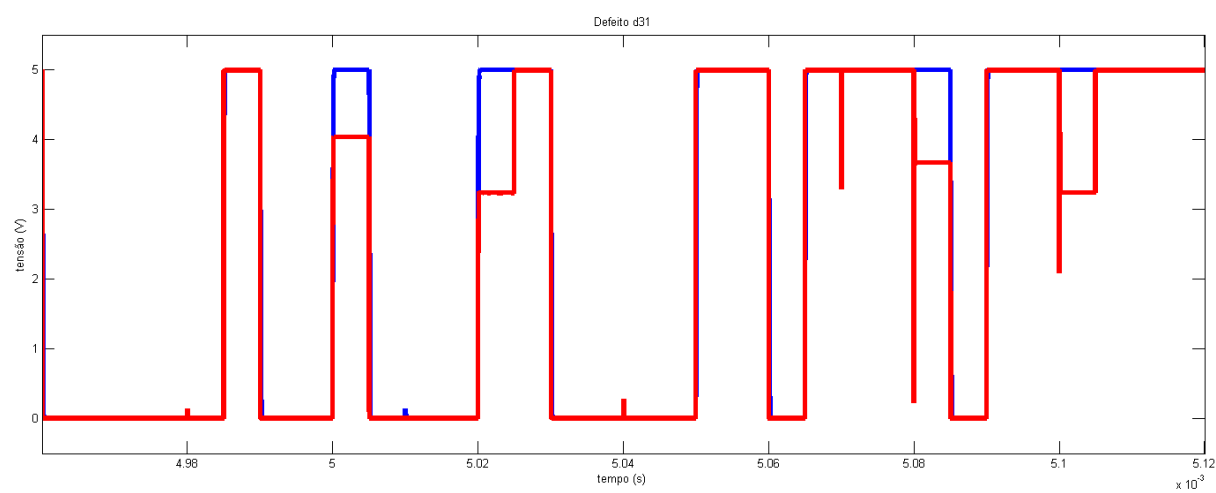
Figura A.28 – Resultado da simulação analógica com o defeito **d29**.



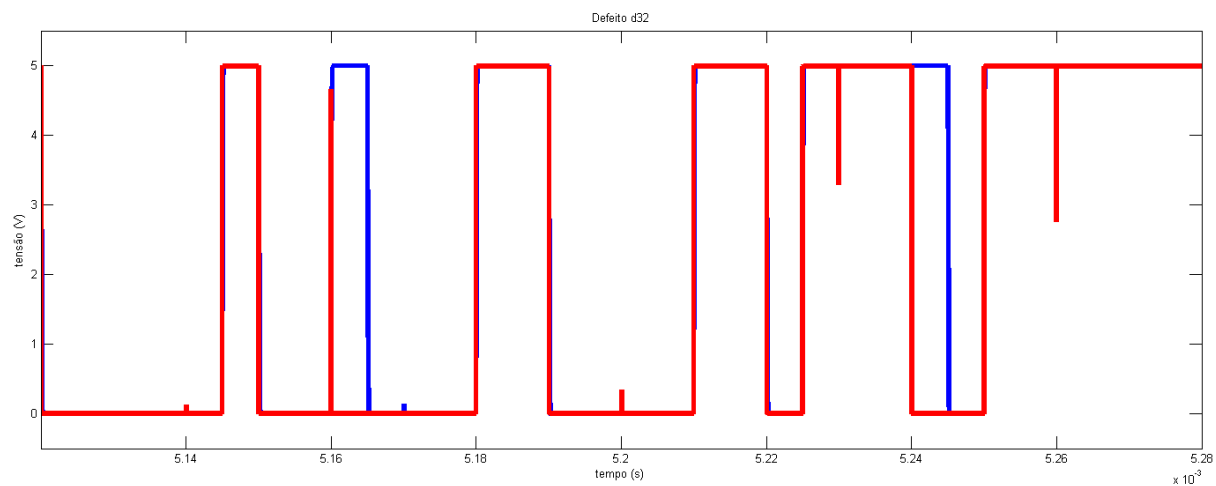
Fonte: Elaborada pela autora.

Figura A.29 – Resultado da simulação analógica com o defeito **d30**.

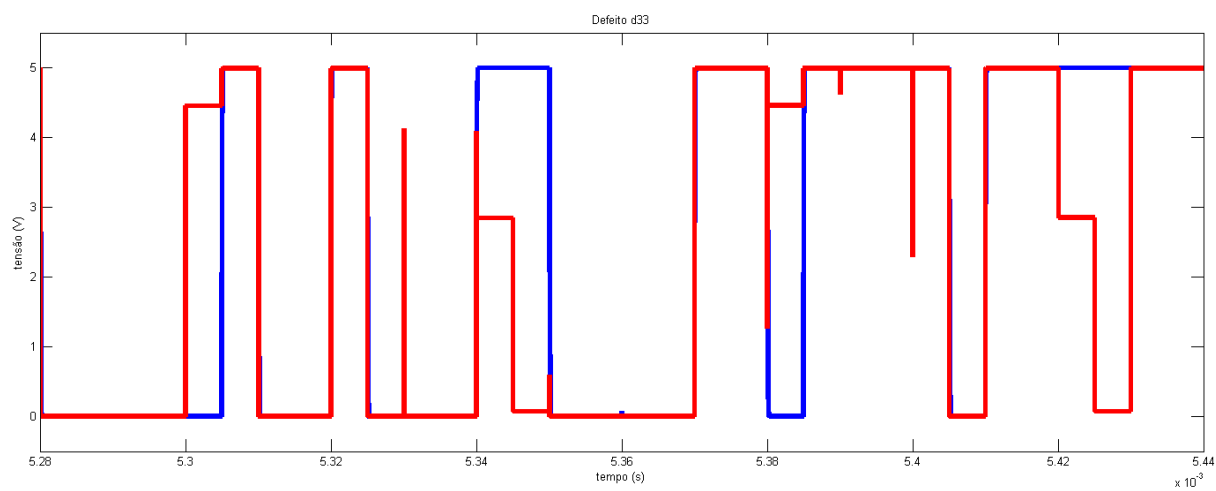
Fonte: Elaborada pela autora.

Figura A.30 – Resultado da simulação analógica com o defeito **d31**.

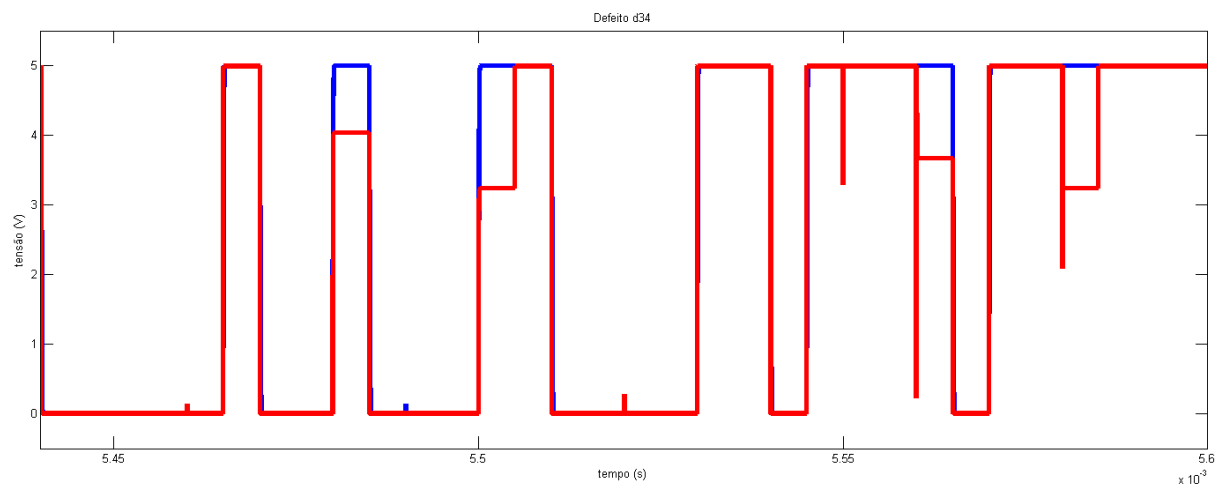
Fonte: Elaborada pela autora.

Figura A.31 – Resultado da simulação analógica com o defeito **d32**.

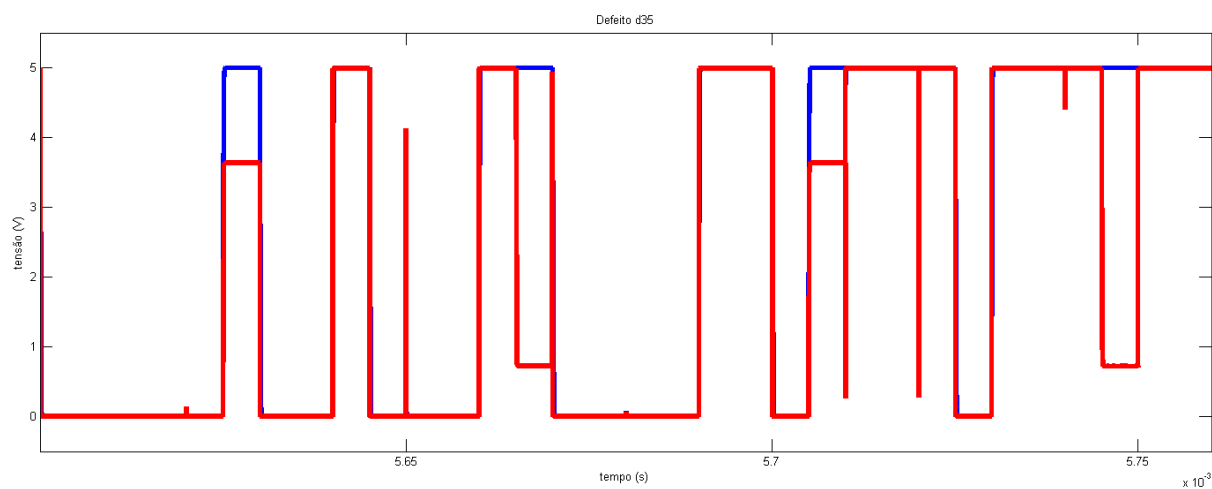
Fonte: Elaborada pela autora.

Figura A.32 – Resultado da simulação analógica com o defeito **d33**.

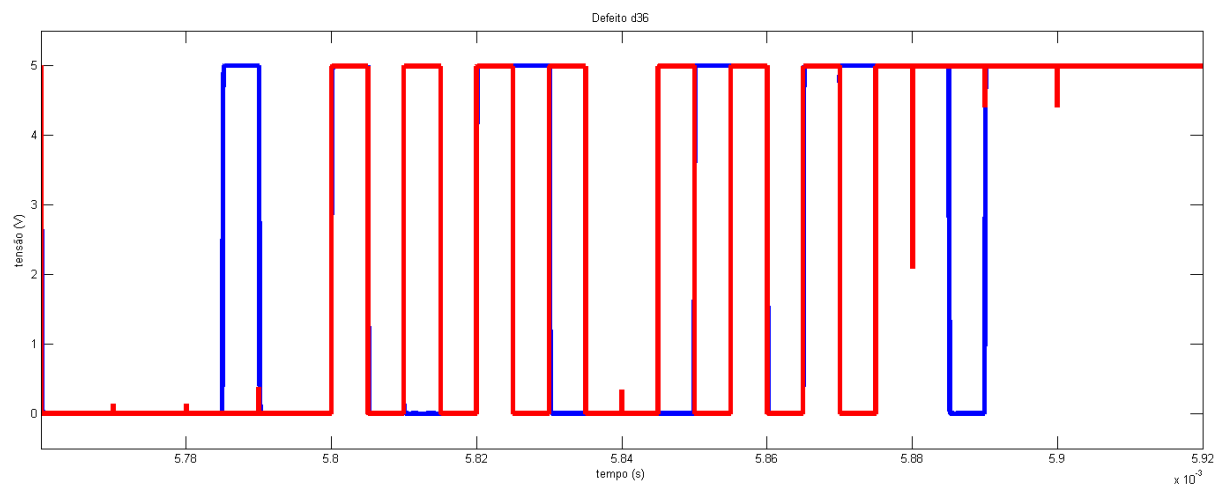
Fonte: Elaborada pela autora.

Figura A.33 – Resultado da simulação analógica com o defeito **d34**.

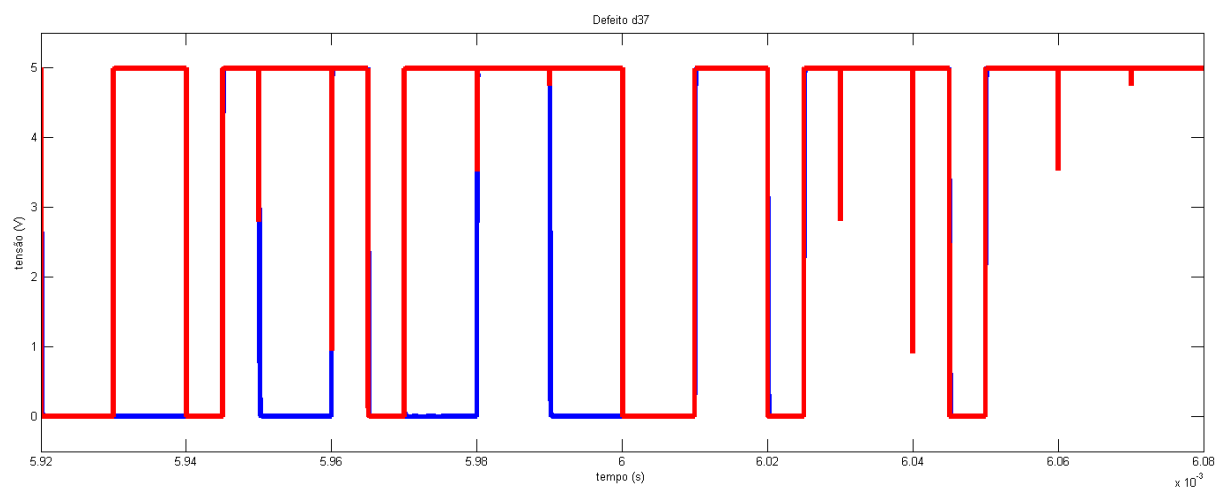
Fonte: Elaborada pela autora.

Figura A.34 – Resultado da simulação analógica com o defeito **d35**.

Fonte: Elaborada pela autora.

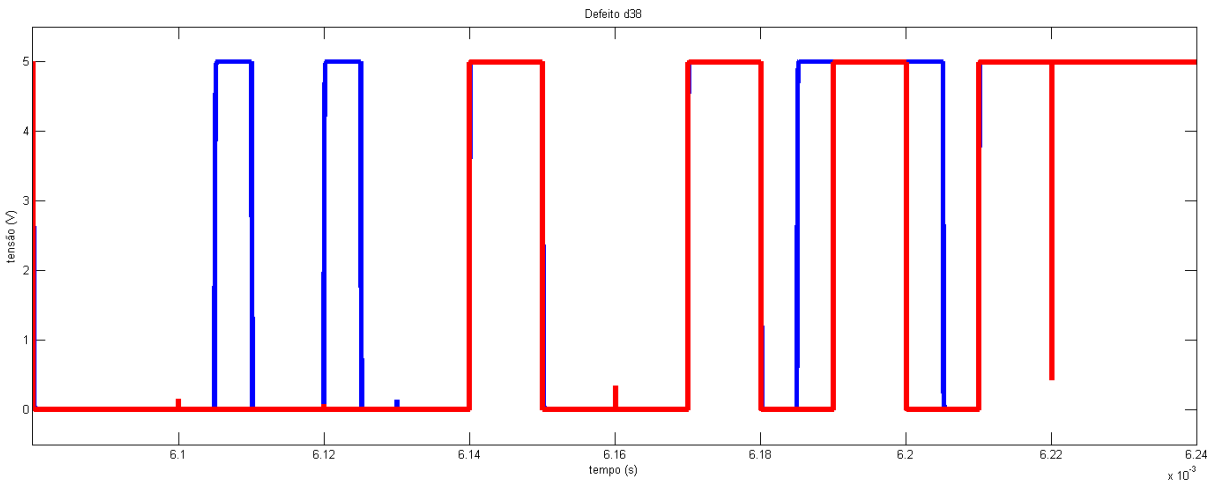
Figura A.35 – Resultado da simulação analógica com o defeito **d36**.

Fonte: Elaborada pela autora.

Figura A.36 – Resultado da simulação analógica com o defeito **d37**.

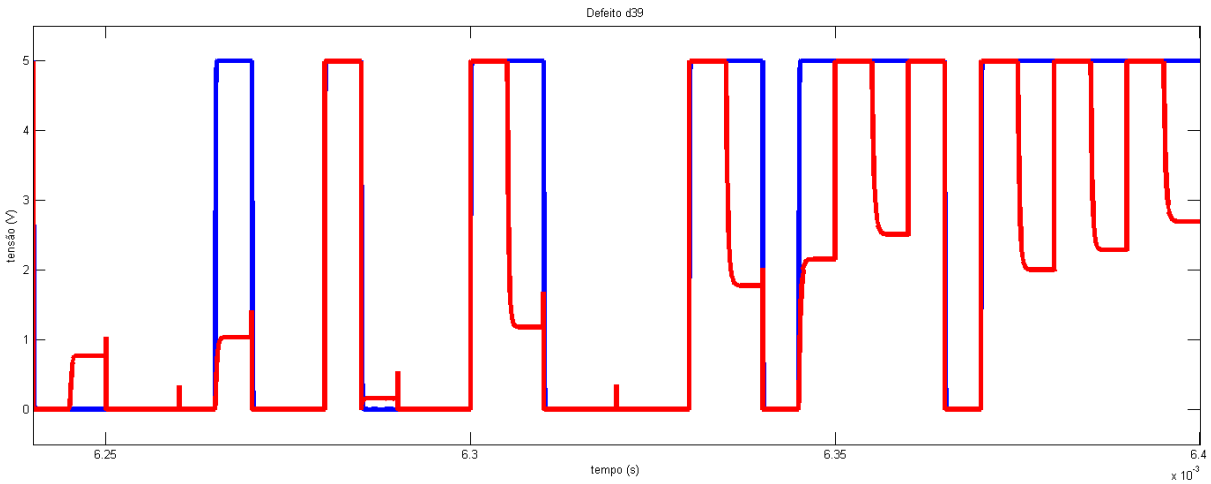
Fonte: Elaborada pela autora.

Figura A.37 – Resultado da simulação analógica com o defeito **d38**.

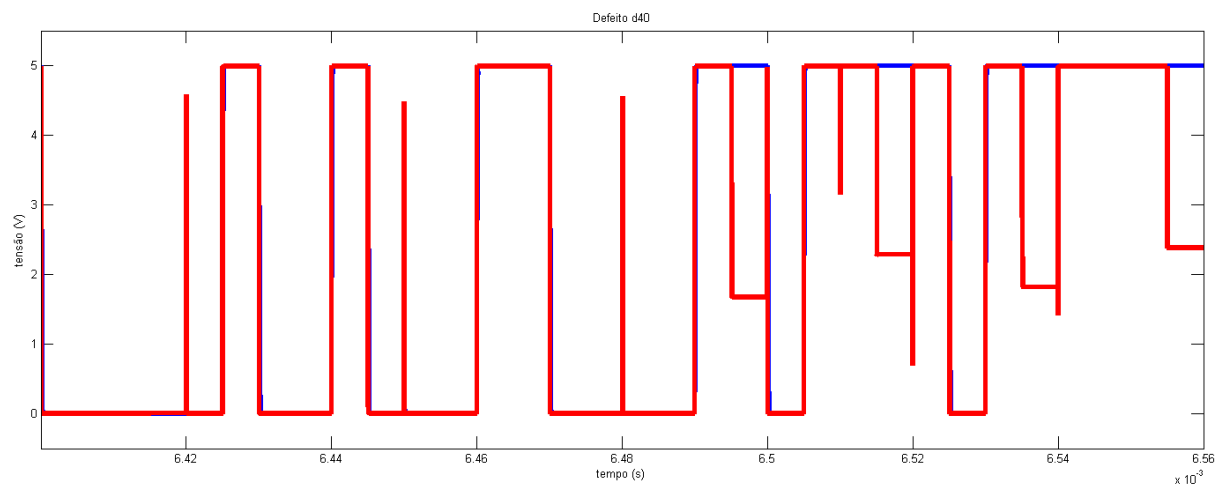


Fonte: Elaborada pela autora.

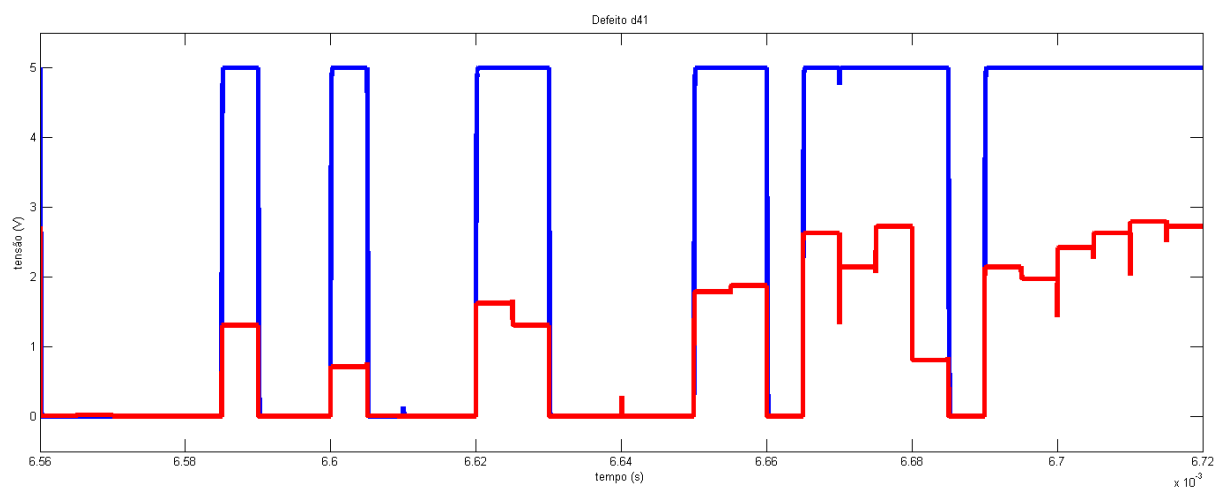
Figura A.38 – Resultado da simulação analógica com o defeito **d39**.



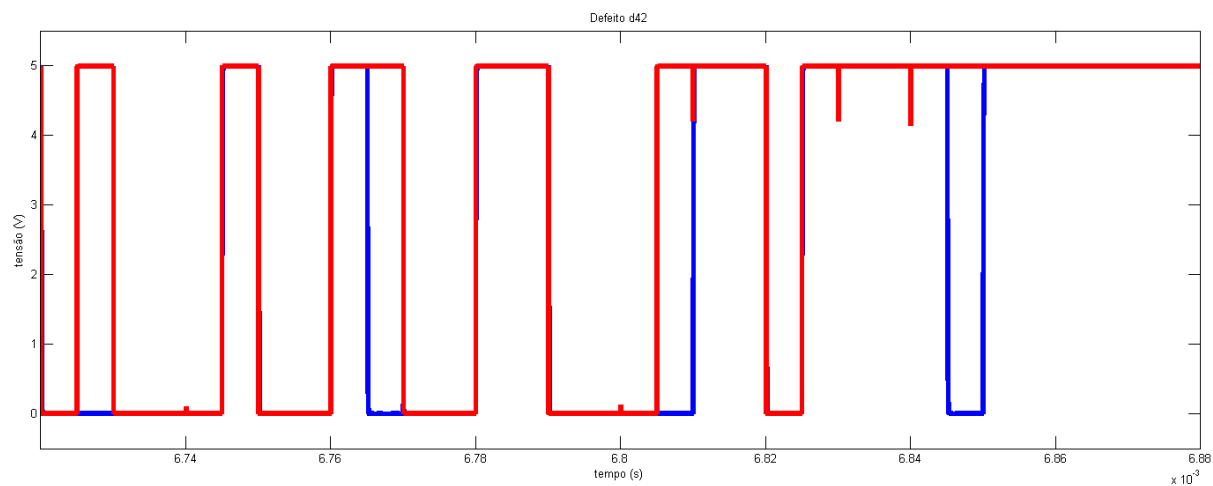
Fonte: Elaborada pela autora.

Figura A.39 – Resultado da simulação analógica com o defeito **d40**.

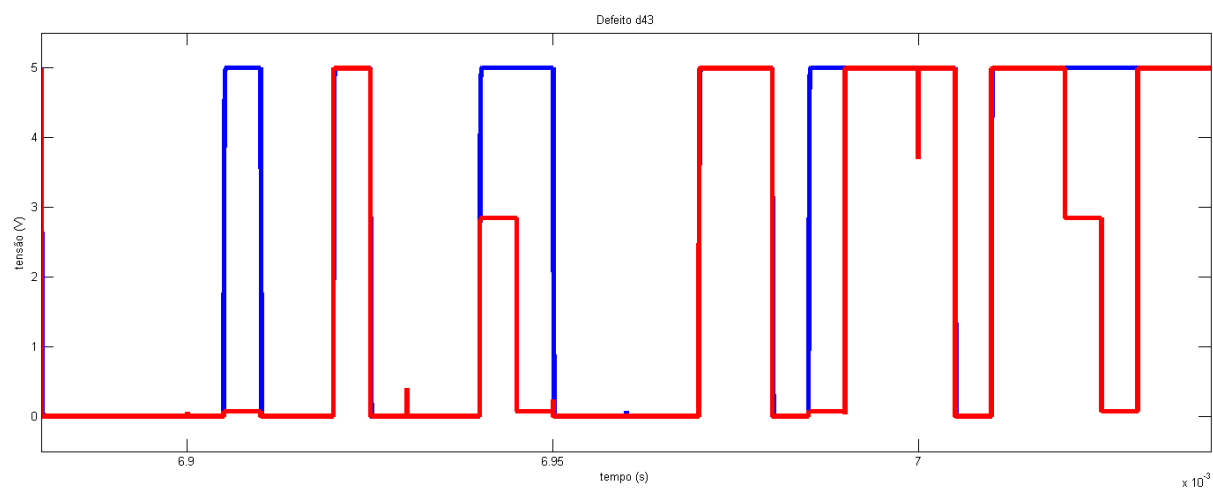
Fonte: Elaborada pela autora.

Figura A.40 – Resultado da simulação analógica com o defeito **d41**.

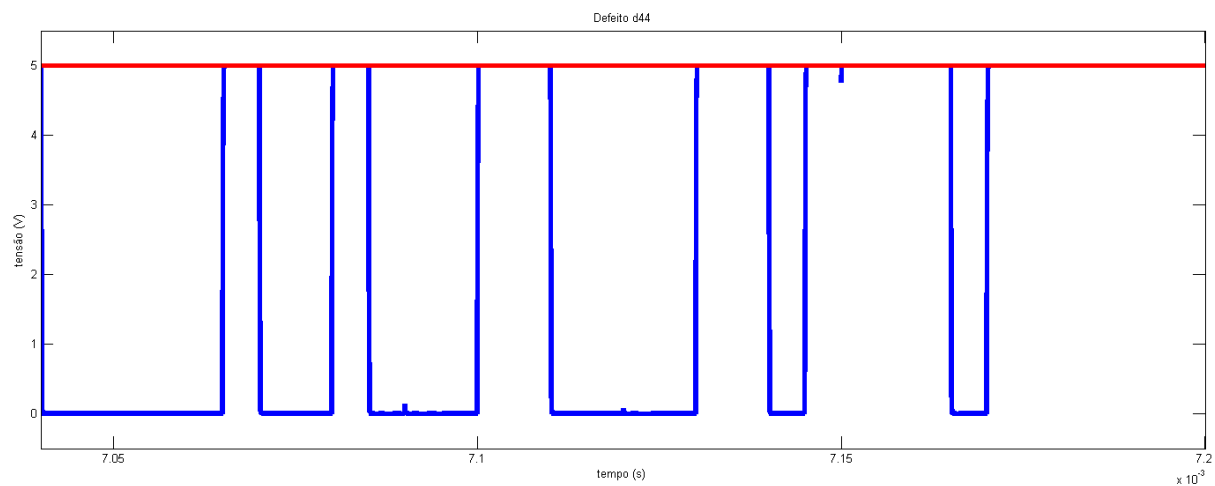
Fonte: Elaborada pela autora.

Figura A.41 – Resultado da simulação analógica com o defeito **d42**.

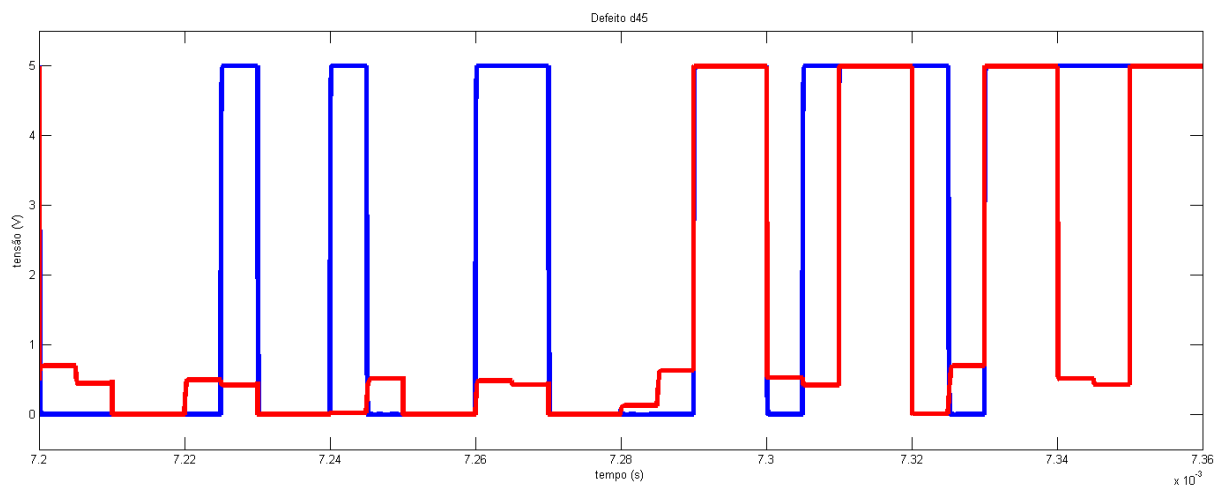
Fonte: Elaborada pela autora.

Figura A.42 – Resultado da simulação analógica com o defeito **d43**.

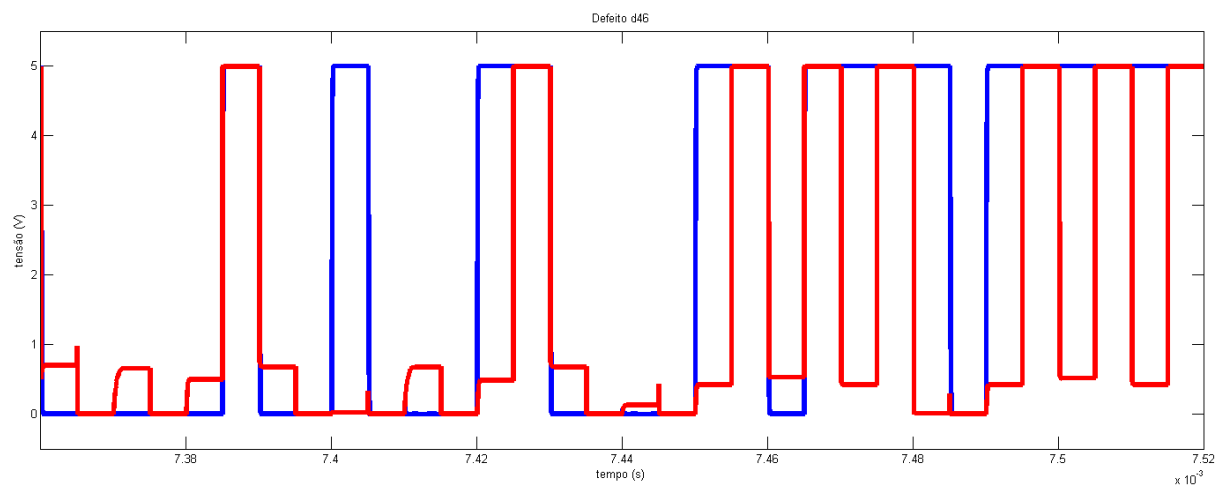
Fonte: Elaborada pela autora.

Figura A.43 – Resultado da simulação analógica com o defeito **d44**.

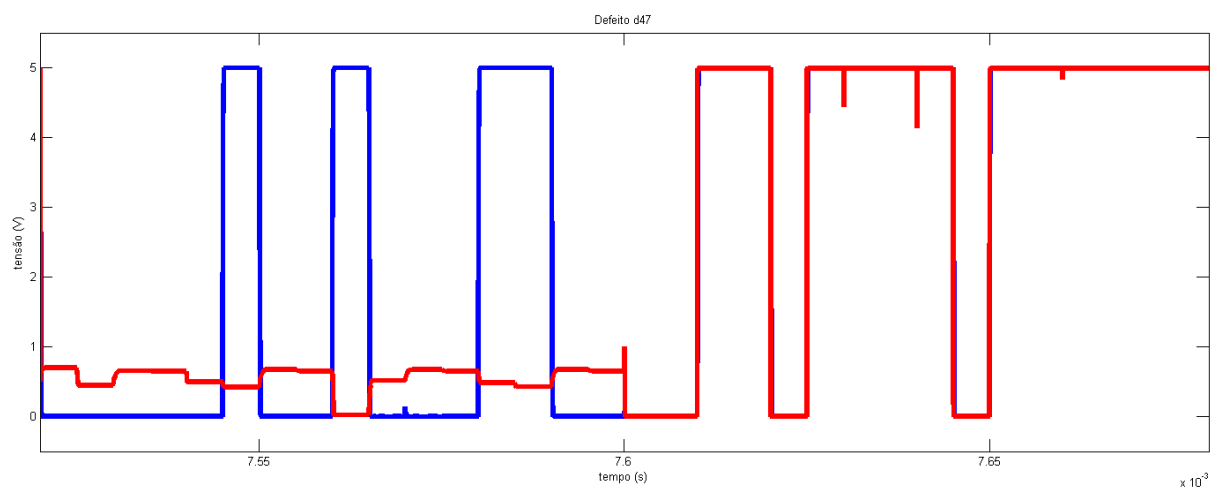
Fonte: Elaborada pela autora.

Figura A.44 – Resultado da simulação analógica com o defeito **d45**.

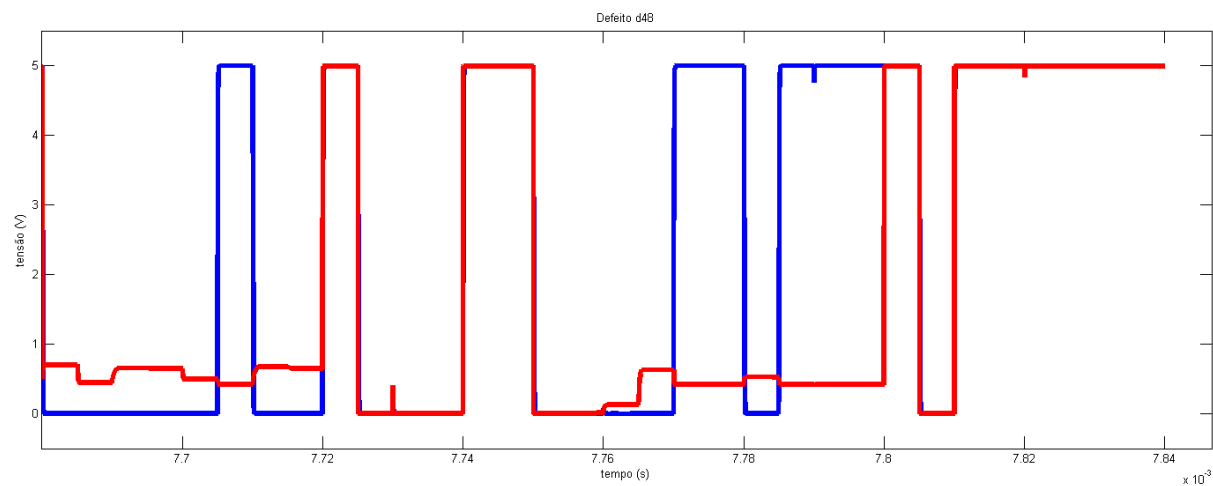
Fonte: Elaborada pela autora.

Figura A.45 – Resultado da simulação analógica com o defeito **d46**.

Fonte: Elaborada pela autora.

Figura A.46 – Resultado da simulação analógica com o defeito **d47**.

Fonte: Elaborada pela autora.

Figura A.47 – Resultado da simulação analógica com o defeito **d48**.

Fonte: Elaborada pela autora.

