

Modelagem de defeitos em circuitos integrados na perspectiva do leiaute

Larissa de Melo Soares ^[1], Cleonilson Protásio de Souza ^[2]

[1] larissa.soares@cear.ufpb.br, [2] protasio@cear.ufpb.br - ^{1,2}UFPB – Cidade Universitária, s/n - Castelo Branco, 58051-900, João Pessoa – PB.

RESUMO

O aumento da densidade de transistores em circuitos integrados (CIs), em virtude dos avanços da tecnologia de fabricação, tornaram os procedimentos de testes de CIs mais complexos. Ao passo que quanto maior a densidade, menor é o espaçamento entre os componentes e suas conexões, aumentando a probabilidade de haver falhas entre os componentes na presença de defeitos físicos. Em virtude da redução no espaçamento, os modelos de falhas clássicos, que representam defeitos físicos em circuitos integrados, já não satisfazem os requisitos atuais para testes. Por isso, atualmente tem-se estudado modelos de falhas que sejam baseados no próprio leiaute do CI e não somente em seu diagrama em nível lógico ou em nível de transistores. Tais modelos visam analisar o leiaute como um todo e verificar os pontos mais prováveis de acontecer uma falha em consequência de um possível defeito físico. Neste trabalho é feito um estudo dos modelos de falhas clássicos e os modelos baseados na perspectiva do leiaute, e são explanados os conceitos de tipos de falhas, defeitos e elementos parasitas. E, finalmente, foram obtidos resultados experimentais com um inversor e foram comparadas as respostas obtidas pela simulação analógica com esquemático, com as respostas obtidas pela simulação baseada no leiaute do inversor com os elementos parasitas.

Palavras-chave: Testes de circuitos integrados. Modelos de falhas clássicos. Modelos de falhas na perspectiva do leiaute.

ABSTRACT

The increasement of the transistors density in Integrated Circuits (ICs), due to the advance of its fabrication technologies, has made the test procedures of ICs more complex. The bigger the density, smaller is the spacing among the components and its connections, raising the probability of a short circuit among the components in the presence of a physical defect. Because of the spacing reduction, the classic faults models that represent physical defects in ICs do not suit the current requirements for defects. Therefore, current faults models are being studied to adequate the present layout. Those models aim to analyze the whole layout and verify the places where there is a bigger probability of presenting a fault as a consequence of a physical defect. In this work a study was made about the classic faults models and the models based on the perspective of the layout, and on it are explained the concepts of types of faults, defects and parasitic elements. Finally, experimental results with an inverter were obtained and compared the responses obtained by the analog simulation with schematic, with the responses obtained by simulation based on the layout with parasitic elements.

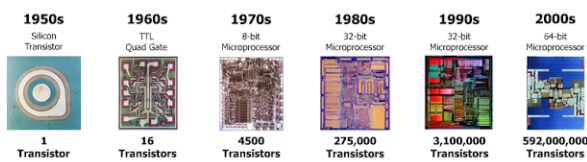
Keywords: *Testing integrated circuits. Classical fault models. Fault models on layout perspective.*

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 de aviões, de aparelhos utilizados na medicina e em carros são alguns exemplos em que é imprescindível a certeza de que estes funcionem corretamente, pois 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; JALOTE, 1974).

O aumento da confiabilidade em sistemas eletrônicos se dá com o aumento da confiabilidade dos componentes que compõem tais sistemas, pois, pelo princípio básico das técnicas de tolerância a falhas, a confiabilidade de um sistema depende da confiabilidade dos componentes que o constituem (JALOTE, 1974). Um dos principais componentes usados em sistemas eletrônicos são os circuitos integrados (CIs). Então, aumentar sua confiabilidade tem consequência na confiabilidade do sistema como um todo.

Figura 1 - Lei de Moore: A densidade de transistores dos circuitos integrados dobra a cada dois anos



Fonte: Roberto et al. (2009).

Os CIs, a cada ano que passa, estão cada vez com áreas menores, com um número maior de transistores, como pode ser visto na Figura 1. Uma das etapas de grande importância no projeto dos CIs é a de teste, que visa 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, os 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

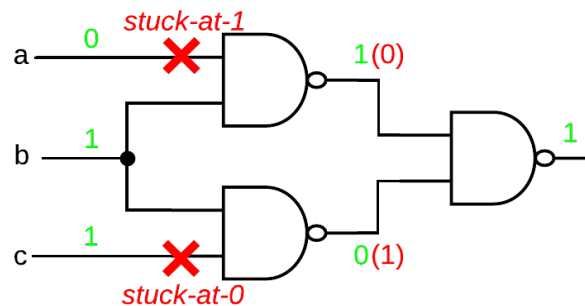
verificar se o circuito está falho ou não, comparando a resposta do teste com as do circuito considerado sem defeitos (HURST, 1998).

Para a proposição de métodos de testes em CIs ou determinação dos vetores de testes, faz-se 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 quando há uma ligação fixa entre a entrada e Vdd (tensão de alimentação), diz-se que pode ocorrer uma falha *Stuck-At*. E, como está “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, por exemplo, curto-circuito de transistores, curto entre interconexão etc.

Para melhor ilustrar o modelo de falhas *Stuck-At*, na Figura 2 observa-se que há uma falha *stuck-at-1* na entrada *a* e uma falha *stuck-at-0* na entrada *b*, significando 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. Isso significa que nem toda falha causa um erro no circuito integrado (PLUSQUELLIC, 2006b).

No modelo *stuck-at* não são modelados curtos-circuitos internos à célula, como em ligações de transistores, mas apenas aqueles que ocorrem nas interconexões entre portas lógicas. Apesar disso, esse modelo é amplamente utilizado atualmente e cobre uma alta porcentagem dos defeitos físicos passíveis de ocorrer em um circuito integrado (PLUSQUELLIC, 2006b). O modelo de falhas *stuck-at* é o mais conhecido e utilizado atualmente (HAPKE et al., 2009).

Figura 2 - Falha Stuck-At

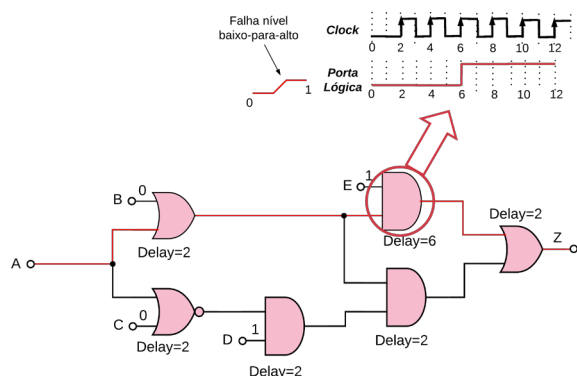


Fonte: Elaborada pela autora.

O segundo modelo mais utilizado é o *Transition Delay* (COX; RAJSKI, 1988) que modela defeitos

causando atraso na mudança de estado de 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. Existem dois tipos de falhas *transition delay*: a falha nível baixo-para-alto, quando o atraso é na transição do nível baixo para o nível alto, e a falha nível alto-para-baixo em que o atraso acontece na transição do nível alto para o nível baixo (WAICUKAUSKI et al., 1987).

Figura 3 - Falha *Transition Delay* (GARCIA, 2001)



Fonte: Elaborada pela autora.

Na Figura 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 possa ser detectado.

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 ocorria dentro da biblioteca de células padrão, que são conjuntos de

transistores e interconexões que formam as funções booleanas 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).

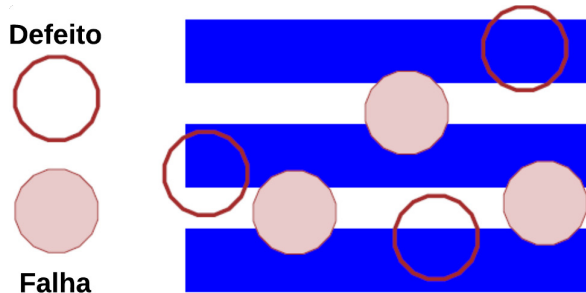
Com essas desvantagens apresentadas pelos modelos de falhas clássicos, alternativas estão sendo estudadas nos últimos anos para resolver estes problemas, sendo a mais relevante a que trata do modelamento de falhas e simulações analógicas baseadas em *netlist SPICE* (*Simulated Program with Integrated 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), diretor de engenharia da *Mentor Graphics*, definiu o que é o *Cell-Aware Test* ou CAT: modelos de falhas na perspectiva do leiaute (HAPKE et al., 2009; HAPKE et al., 2010; HAPKE et al., 2014).

2 Modelos de falhas

Nesta seção serão detalhados os modelos de falhas *Stuck-At* e *Transition Delay*, e definidos alguns conceitos essenciais usados em testes. 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, por sua vez, é a possível consequência do defeito físico. A diferença entre falha e defeito é demonstrada na Figura 4.

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 curtos-circuitos, abertos e em ponte, são os mais comuns de ocorrerem (HAPKE et al., 2014). Para isso, modelos de falhas vêm sendo estudados ao longo dos anos a fim de que a maior parte desses possíveis defeitos físicos possa ser modelada.

Figura 4 - Diferença entre falha e defeito



Fonte: Elaborada pela autora.

2.1 Modelo de Falhas *Stuck-At*

Um dos primeiros modelos de falhas e mais conhecido na área de testes (HAPKE et al., 2010; HAPKE et al., 2014), criado em 1963, é o modelo *Stuck-at* (MEI, 1974).

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 a Freescale Semicondutores, têm por objetivo entregar chips acima de 99% de cobertura de falhas *stuck-at* para os fabricantes automotivos. Isso significa dizer que os testes projetados para o circuito integrado cobrem acima de 99% das falhas *stuck-at* que possam ocorrer. Apesar de amplamente utilizado nos dias atuais, 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 defeitos físicos diretamente.

2.2 Modelo de falhas Transition Delay

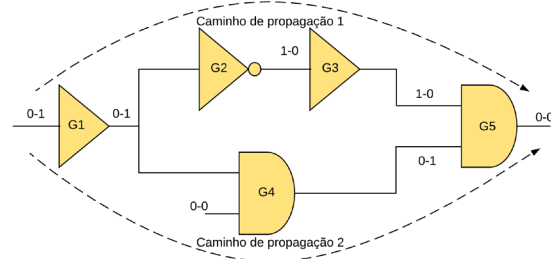
Defeitos que causam falhas, mas que não são relacionados com tempo de mudança de estado na porta lógica são chamados defeitos estáticos. Já defeitos que causam falhas e são relacionados com as mudanças de estado da porta lógica são os chamados defeitos de *delay* (CHENG et al., 2008). Para detectar defeitos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

Para inserir padrões ou vetores de testes 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 lógicos, 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 modelo de falhas *transition delay*. Na Figura 5 pode-se observar, como exemplo de modelo de falhas *transition delay*, um circuito com cinco portas lógicas. O efeito da transição (0 – 1) é aplicado 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 5 - 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 Metodologia de testes: *cell-aware test* (cat)

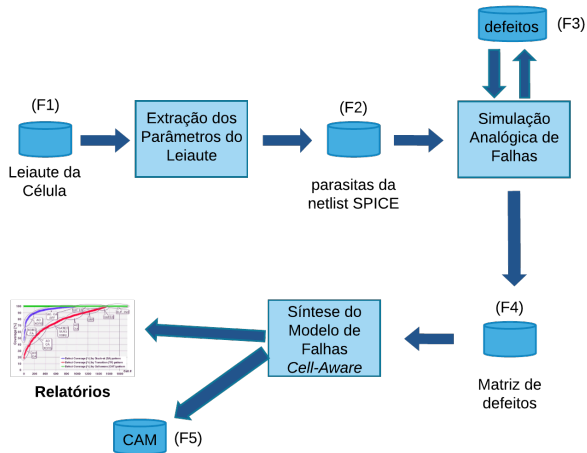
O *Cell-Aware Test* ou CAT é uma metodologia de testes que tem como principal atrativo o modelamento de falhas baseados no leiaute do circuito integrado que se deseja testar (HAPKE et al., 2009). A consequência é um teste na perspectiva do leiaute, ou seja, o leiaute da célula é analisado, bem como os transistores, resistências e capacitâncias parasitas são analisadas. A partir de todas essas informações, os modelos das falhas são descritos para um leiaute em específico. Sabendo disso, podem-se ter modelos de falhas diferentes e até mesmo um número distinto de modelos de falhas para um mesmo CI. Isso acontece porque, dependendo da forma como o leiaute é feito, 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 ao artigo Hapke et al. (2009), sua contribuição foi uma nova metodologia para atingir 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-se, assim, uma necessidade de pesquisas mais aprofundadas para a resolução deste problema. A nova metodologia *Cell-Aware* foi avaliada para tecnologias de 90nm e 150nm em 1671, células de biblioteca, testada 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).

Uma das bases da metodologia *Cell-Aware* Test é que essa é aplicada no desenvolvimento de testes para células de uma determinada biblioteca de células lógicas padrão. Assim, para cada célula (AND, OR, NAND, XOR etc.), devem ser realizados os procedimentos CAT.

Figura 6 - Etapas do Cell-Aware

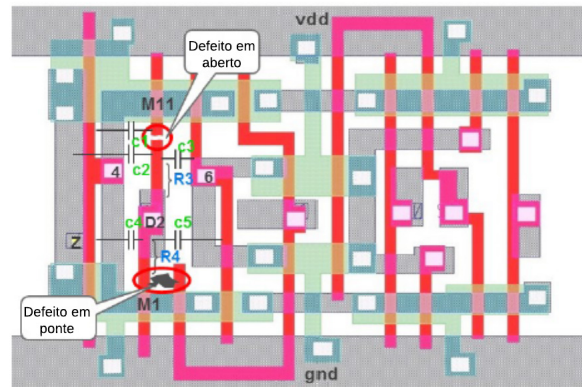


Fonte: Elaborada pela autora.

Para entender o *Cell-Aware* Test é necessário compreender o fluxograma da metodologia. Na Figura 6 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 desse leiaute, podem-se dizer 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 se baseia 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*.

Figura 6 - Defeitos que podem ocorrer no leiaute



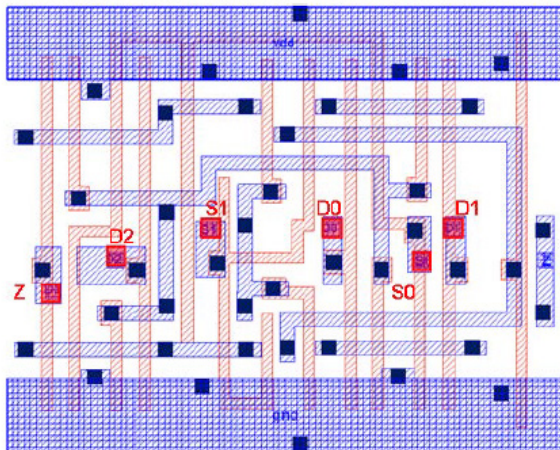
Fonte: Elaborada pela autora.

Na Figura 7 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.

Na Figura 11 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, tem-se na Figura 9 a lista de falhas possíveis em F2 para o MUX31X4. Por exemplo, sabendo que os seletores do multiplexador são *S0* e *S1*, as saídas desses seletores negados serão dadas, respectivamente, por *SON* 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 *SON*, esteja fixa em GND. Este é um exemplo, os outros 47 defeitos seguem o mesmo padrão e são descritos na Figura 9, em que *dx = E*, *P* é o nome do defeito; *E* é interconexão em que está localizado o defeito e *P* é onde o *E* está preso.

Figura 8 - Leiaute da célula do multiplexador MU-X31X4



Fonte: Hapke et al. (2009).

Figura 9 - Lista de defeitos

d1 = S0N, gnd	d17 = D0, gnd	d33 = net38, D1
d2 = S1N, gnd	d18 = vdd, gnd	d34 = net81, D0
d3 = net65, gnd	d19 = Z, net65	d35 = net38, D0
d4 = net57, gnd	d20 = S1, S0N	d36 = S1, S0
d5 = net19, gnd	d21 = S1N, S1	d37 = D2, S1
d6 = net81, gnd	d22 = net65, S1	d38 = S0, D1
d7 = net38, gnd	d23 = S0N, S0	d39 = vdd, S0N
d8 = net85, gnd	d24 = net81, S1	d40 = vdd, S1N
d9 = net35, gnd	d25 = S1, net38	d41 = vdd, net65
d10 = net31, gnd	d26 = D2, S1N	d42 = D0, S0
d11 = net69, gnd	d27 = net81, S0	d43 = net38, vdd
d12 = Z, gnd	d28 = net65, D2	d44 = vdd, Z
d13 = S1, gnd	d29 = net38, S0	d45 = vdd, S1
d14 = S0, gnd	d30 = S0N, D1	d46 = vdd, S0
d15 = D2, gnd	d31 = net81, D1	d47 = vdd, D2
d16 = D1, gnd	d32 = S0N, D0	d48 = vdd, D1

Fonte: Hapke et al. (2009).

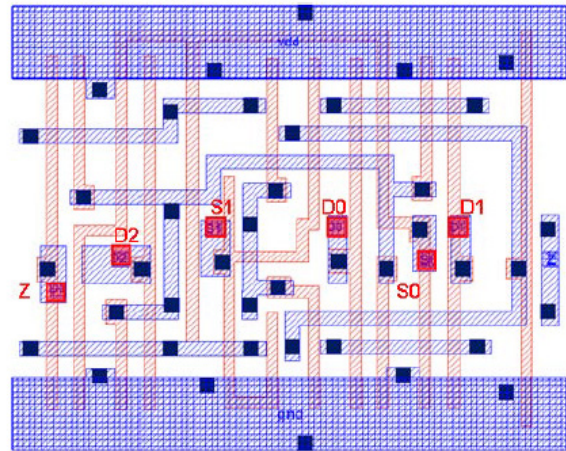
A partir dessa lista de defeitos em potencial têm-se 48 *netlists* adicionais para cada defeito, além da *netlist* sem defeitos, que é chamada de *golden netlist* (HAPKE et al., 2009).

Cada uma das 49 *netlists* será simulada, em que, como cada uma possui cinco 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.

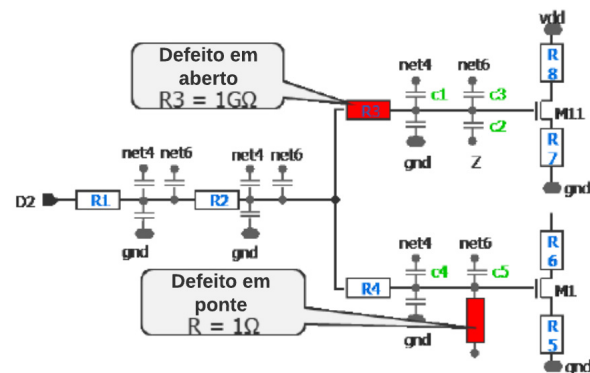
Durante a análise de todas as simulações, pode-se construir o que é denominado de Matriz de Defeitos (Figura 10).

Figura 10 - Matriz de Defeitos



A Matriz de Defeitos da célula MUX31X4 é uma tabela que contém todas as 32 entradas da célula representadas nas linhas da matrix, e os 48 defeitos são representados nas colunas da matrix. Por exemplo, na linha que tem a entrada 0111110 detecta (D), entre outros, o defeito *d41*, o qual é um curto-circuito entre a *net65* e *Vdd* (HAPKE; SCHLOEFFEL, 2012).

Figura 11 - Transistores extraídos da *netlist*



Fonte: Hapke et al. (2009).

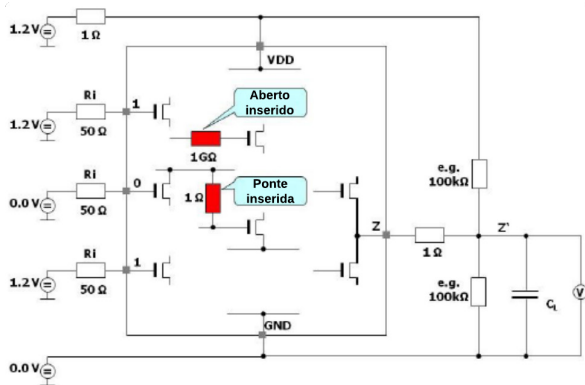
Na Figura 11 é ilustrada uma parte dos componentes (resistores, capacitâncias, entradas, transistores, Vdd e Gnd) do MUX31X4. Além dos componentes que são do leiaute, a *netlist* contém os possíveis defeitos. Cada um dos defeitos é colocado por vez, um em cada *netlist*, para que possa ser feita análise 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 12 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 12 a entrada 01010 com os defeitos em aberto e em ponte. Nas simulações analógicas, um defeito é considerado detectado se pelo menos uma das entradas resulta em uma saída que desvia em mais do que 50% da sua tensão de alimentação Vdd.

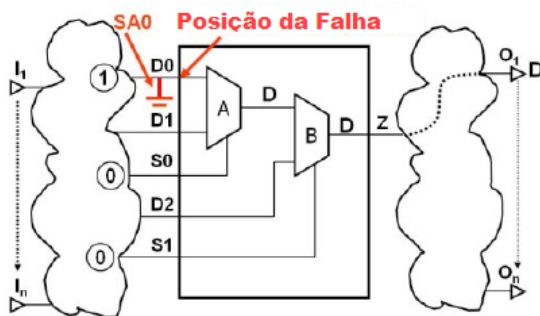
Figura 12 - Ambiente de simulação analógica



Fonte: Hapke et al. (2009).

O processo de inserção das falhas para análise das saídas é diferente para o CAT. Como pode ser visto na Figura 13, 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, só se irá precisar dessa entrada na geração dos padrões para essa falha em específico (HAPKE et al., 2009), ou seja, a condição de excitação da falha interna é colocada na entrada da célula.

Figura 13 - Processo do ATPG normal

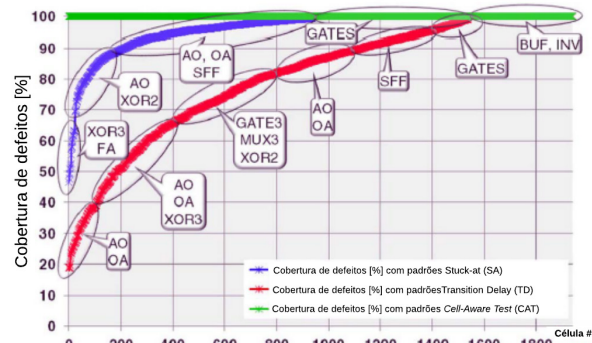


Fonte: Hapke et al. (2009).

Depois de se realizar todas as simulações analógicas e de se 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).

Figura 14 – Cobertura de defeitos do CAT em comparação com Stuck-At e Transition



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

Na Figura 14 é apresentado o gráfico com a 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 14 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%, por exemplo: AO (AND-OR) e OA (OR-AND) (HAPKE et al., 2014).

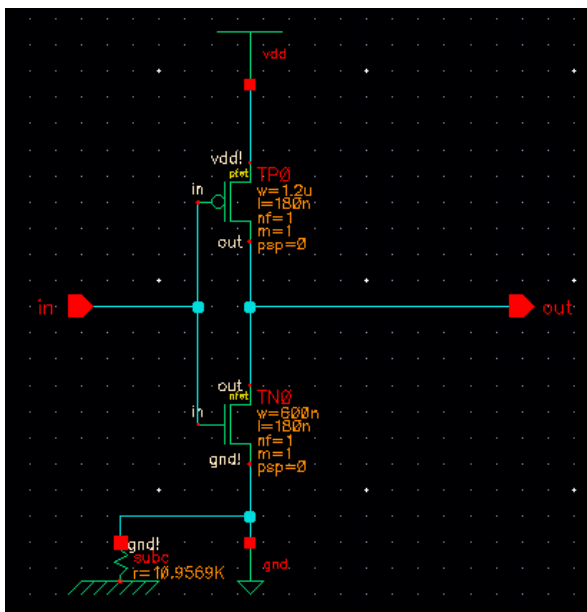
Ao analisar os conceitos do *Cell-Aware Test* surgiram dúvidas em relação aos parâmetros que seriam usados para estabelecer os valores de resistência correspondentes a cada modelo de falhas. Como os modelos de falhas são baseados no leiaute da célula, e cada leiaute da mesma célula pode ser diferente, com ligações feitas de formas e distâncias distintas, a lista de defeitos será diferente, e a forma de modelar a falha também. Os valores de resistências e capa-

citâncias para a simulação analógica serão 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 esse trabalho foram feitos estudos mais aprofundados a fim de saber como são definidos os parâmetros que levam aos modelos de falhas.

4 Exemplo de uma aplicação do cat

O *Cell-Aware Test* é uma metodologia que foi implementada pela *Mentor Graphics* em uma ferramenta específica e com bibliotecas próprias. Para realizar as primeiras etapas do fluxo idealizado pelo CAT, foram utilizadas, nesta pesquisa, ferramentas da Cadence, no sentido de poder avaliar as possibilidades de aplicação do CAT além das ferramentas *Mentor Graphics*. Dessa forma, definiu-se o esquemático de um inversor que pode ser visto na Figura 15. A partir desse esquemático serão feitas simulações DC e analógicas, leiaute, extração de parâmetros e simulações a partir do leiaute com parasitas.

Figura 15 - Esquemático do inversor

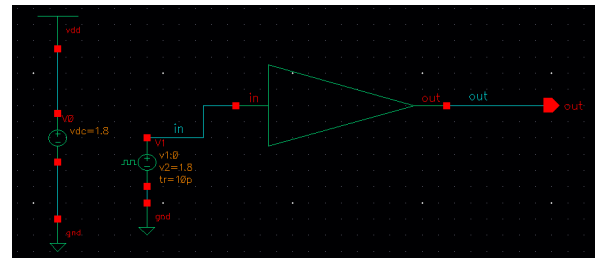


Fonte: Elaborada pela autora.

Primeiramente, será criado o símbolo do inversor que conterá o esquemático do inversor. Na Figura 16 pode-se constatar o inversor, com a fonte na entrada variando com pulsos e tensão Vdd em 1,8V.

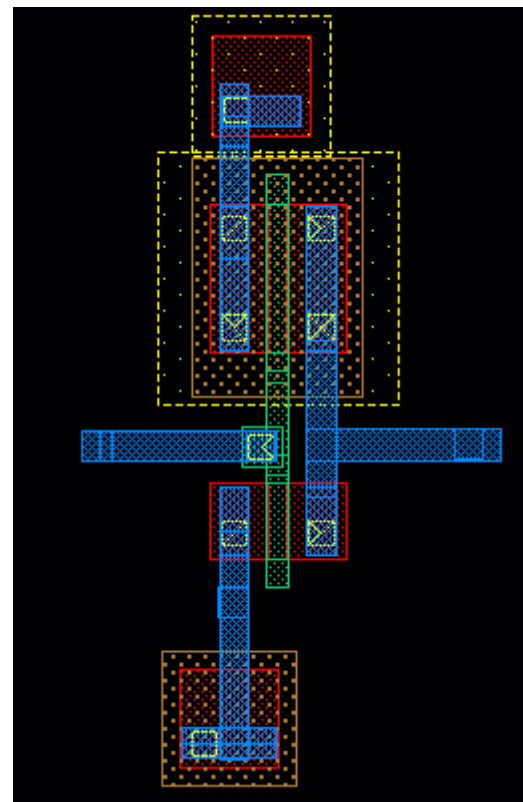
Na Figura 17 é visto o leiaute do inversor obtido a partir do esquemático. Nas próximas etapas serão feitas simulações a partir do esquemático e serão comparadas com as simulações do leiaute com suas capacitâncias para observar as respostas.

Figura 16 – Esquemático com símbolo inversor com as fontes de entrada



Fonte: Elaborada pela autora.

Figura 17 – Leiaute do inversor



Fonte: Elaborada pela autora.

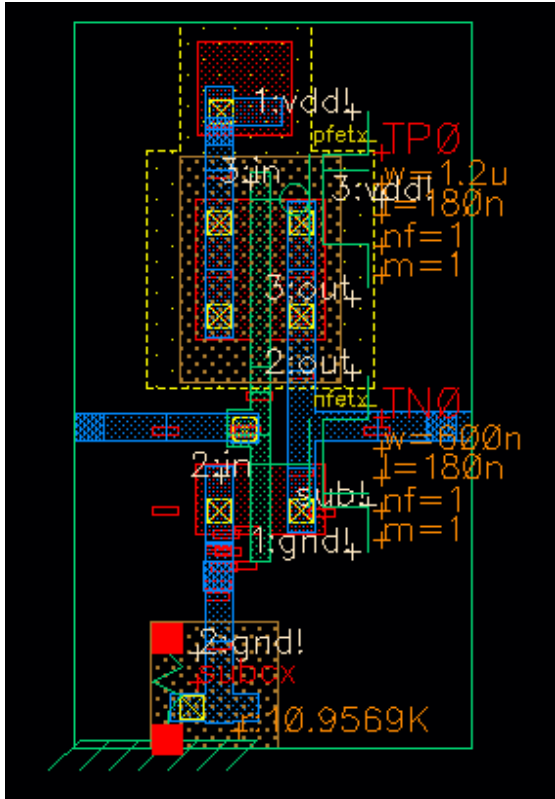
Na Figura 18 é esclarecido como os modelos de falhas que são baseados no CAT podem ser extraídos. Após o leiaute ser obtido, na ferramenta Cadence existe a opção chamada *av_extracted* em que o leiaute é mostrado juntamente com seus elementos parasitas (resistências e capacitâncias). Estes elementos não são componentes no esquemático, eles são consequência da disposição dos elementos no leiaute.

Sabe-se que a capacitância é dada pela Equação:

$$C = \epsilon_R \epsilon_0 \frac{lw}{s} \quad (1)$$

em que l e s , dados pelo comprimento e pela distância entre as interconexões, respectivamente, e w é dado pela altura das interconexões no metal.

Figura 17 – Leiaute do inversor



Fonte: Elaborada pela autora.

Sendo assim, pode-se concluir que quanto maior forem as distâncias entre as nets, menores serão as capacitâncias parasitas.

Essas capacitâncias terão influência direta no modelamento de falhas na perspectiva do leiaute.

Na Figura 19 estão representados os defeitos que podem ocorrer por material extra, algo relativamente comum no processo de manufatura. Na primeira delas, é apresentada uma capacitância e a distância entre nets C_{p1} . Se um defeito de material extra ocorrer, é fácil ver que haverá uma variação de capacitância, que é $C_{p1} \ll C_{p1}^*$ porque $s_1 \gg s_1^*$, como pode ser vista na segunda parte da Figura 19.

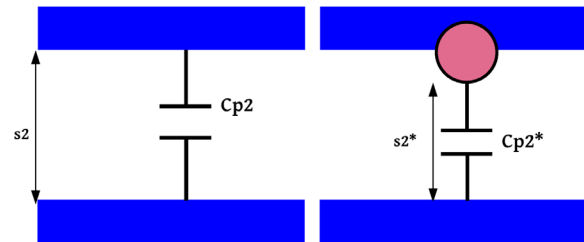
Figura 19 - Distância entre nets versus Probabilidade de mudanças nas capacitâncias (nets próximas)



Fonte: Elaborada pela autora.

Na Figura 20 estão representados os defeitos em nets mais distantes $C_{p1} > C_{p2}$, sabendo que $s_2 > s_1$. Na primeira delas, é apresentada uma capacitância C_{p2} e a distância entre nets s_2 . Neste caso, o efeito do mesmo defeito causado por material extra é menor na capacitância C_{p2} , porque, como pode ser visto na segunda parte dela, $s_2 \approx s_2$ e, como resultado, $C_{p2} \approx C_{p2}$.

Figura 20 - Distância entre nets versus probabilidade de mudanças nas capacitâncias (nets distantes)



Fonte: Elaborada pela autora.

Conclui-se então que, em um leiaute de um CI, quanto maior for a capacitância, maiores são as probabilidades de um defeito contribuir e se tornar uma falha (ALLAN; WALTON, 1997), (ALLAN, 1998), (ALLAN, 1999).

A obtenção dos elementos parasitas é primordial para o estudo do leiaute como um todo e, posteriormente, a partir destes elementos, o modelamento das falhas e o cálculo das contribuições de defeitos.

Por isso, na Figura 21 foi obtida a *netlist* com os transistores, capacitâncias e resistências. Nessa *netlist*, obtida a partir do leiaute do inversor, podem-se observar 10 capacitâncias e 13 resistências. A partir desses valores, pode ser feita uma análise geral do leiaute e prever onde terão maiores chances de que um defeito que ocorra em dado leiaute cause uma falha.

Na Figura 22 é mostrada a simulação analógica da resposta do inversor: a saída do esquemático (em azul) e a saída do leiaute com os parasitas (em vermelho).

Nesta simulação foi colocada na tensão de entrada uma forma de onda com $10ps$ para o nível lógico alto e $10ps$ para o nível lógico baixo. Ao aplicar essa tensão de entrada, foi observado quanto tempo o inversor demorava em fazer a mudança de estado lógico. Foi observado que a saída com os parasitas teve uma resposta mais lenta do que a do esquemático, e teve uma diferença de $5,6ps$.

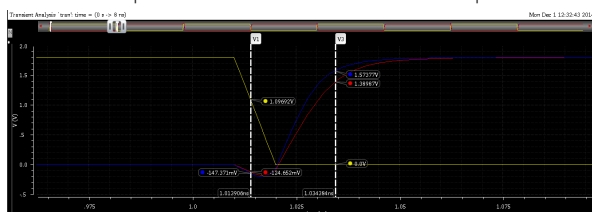
Figura 21 - Netlist dos elementos do inversor

```
// Design library name: CELL_AWARE_TEST
// Design cell name: inv
// Design view name: av_extracted
simulator lang=spectre
global 0 vdd!
parameters p5vonly=0 phires=0
include "design.scs"
include "wafer.scs"
include "allModels.scs"

// Library name: CELL_AWARE_TEST
// Cell name: inv
// View name: av_extracted
c1 (in 0) capacitor c=1.17281e-16
c2 (out 0) capacitor c=1.20111e-16
c3 (sub! 0) capacitor c=6.41216e-16
c4 (\3\:in 0) capacitor c=1.33706e-16
c5 (\2\:in 0) capacitor c=2.84264e-16
c6 (\3\:vdd! 0) capacitor c=6.40764e-17
c7 (\3\:out 0) capacitor c=1.22944e-16
c8 (\2\:out 0) capacitor c=4.74634e-17
c9 (\1\:vdd! 0) capacitor c=1.26938e-16
c10 (\1\:out 0) capacitor c=1.6682e-16
ri1 (in \1\:in) resistor r=0.4526 c=0
ri2 (\1\:in \2\:in) resistor r=55 c=0
ri3 (\1\:gnd! 0) resistor r=11.1896 c=0
ri4 (0 \2\:gnd!) resistor r=11.4312 c=0
ri5 (vdd! \2\:vdd!) resistor r=0.2663 c=0
ri6 (\2\:vdd! \1\:vdd!) resistor r=11.2732 c=0
ri7 (vdd! \3\:vdd!) resistor r=11.058 c=0
ri8 (\1\:vdd! \2\:vdd!) resistor r=11 c=0
ri9 (out \1\:out) resistor r=0.3577 c=0
ri10 (\1\:out \2\:out) resistor r=11.2418 c=0
ri11 (\1\:out \4\:out) resistor r=0.3203 c=0
ri12 (\4\:out \3\:out) resistor r=11.2732 c=0
ri13 (\3\:out \4\:out) resistor r=11 c=0
rj_1_1 (\3\:in \4\:in) resistor r=23.9167 c=0
rj_1_2 (\4\:in \2\:in) resistor r=1.8171 c=0
rj_1_3 (\1\:in \4\:in) resistor r=13.75 c=0
```

Fonte: Elaborada pela autora.

Figura 16 – Resposta do transiente do inversor: saída com capacitâncias versus saída do esquemático



Fonte: Elaborada pela autora.

5 Conclusão

Os testes de circuitos integrados são de grande importância na fabricação de CIs. Com a miniaturização desses circuitos e o aumento do número de transistores em uma menor área, o foco passou a ser as novas falhas que essa situação atual traz. O modelo de falhas proposto pelo *Cell-Aware Test* obteve ótimos resultados no que diz respeito à cobertura de falhas, faltando entender como os valores dos parâmetros dos modelos de falhas são extraídos do leiaute. Para a primeira etapa de entendimento dos modelos, são analisados os elementos parasitas do leiaute do CI,

que irão prever onde é maior a probabilidade de um defeito ocasionar uma falha. Quanto maiores as capacitâncias parasitas encontradas no leiaute, maior o número de modelos de falhas que terão no CI.

Com esse novo modelamento é fácil concluir que será mais fácil atingir uma alta cobertura de defeitos quando se tratar da perspectiva do leiaute, isso porque o modelamento agora estará direcionado ao CI específico. No experimento realizado neste artigo foi observado o quanto esses elementos parasitas podem influenciar na resposta do CI. E, se pudermos prever os problemas que podem acontecer com esses valores obtidos, por meio dos modelos de falhas, atingiremos um alto nível de desempenho nos CIs entregues à indústria.

REFERÊNCIAS

- ALLAN, G. A comparison of efficient dot throwing and shape shifting extra material critical area estimation. In: IEEE COMPUT. SOC. **Proceedings...** 1998. p. 44–52. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=732150>>.
- ALLAN, G.; WALTON, A. Efficient critical area estimation for arbitrary defect shapes. In: IEEE COMPUT. SOC. **Proceedings...** 1997. p. 20–28. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=628305>>.
- 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. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=661294>>.
- 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. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=790624>>.
- BUBEL, I. et al. AFFCCA: a tool for critical area analysis with circular defects and lithography deformed layout. In: IEEE COMPUT. SOC. PRESS. **Proceedings...** 1995, p. 10–18. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=476932>>.
- CHENG, W. T. et al. Enhancing Transition Fault Model for Delay Defect Diagnosis. In: IEEE. **Proceedings...** 2008. p. 179–184. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4711579>>.

COX, H.; RAJSKI, J. Stuck-open and transition fault testing in CMOS complex gates. IN: IEEE COMPUT. SOC. PRESS. **Proceedings...** 1988, p. 688–694. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=207853>>.

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-tes>>.

GKATZIANI, M. et al. Accurately determining bridging defects from layout. In: IEEE. **Proceedings...** 2007. p. 1-4. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4295259>>.

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. **Proceedings...**, 2009. p. 1–10.

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. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6879635>>.

HAPKE, F. et al. Defect-oriented cell-internal testing. In: TEST CONFERENCE (ITC), 2010 IEEE INTERNATIONAL. **Proceedings...** 2010. p. 1-10.

HAPKE, F.; SCHLOEFFEL, J. Introduction to the defect-oriented cell-aware test methodology for significant reduction of DPPM rates. In: IEEE. **Proceedings...** 2012, p. 1–6. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6233046>>.

HAPKE, F. et al. Gate-exhaustive and cell-aware pattern sets for industrial designs. In: IEEE. **Proceedings...** 2011, p. 1-4. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5783604>>.

HESS, C.; WEILAND, L. Wafer level defect density distribution using checkerboard test structures. In: IEEE. **Proceedings...** 1998. p. 101–106. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=688050>>.

HURST, S. L. VLSI Testing - digital and mixed analogue/digital techniques. United Kingdom: **IEE**, 1998.

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. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=229763>>.

JALOTE, P. **Fault Tolerance in Distributed System**. New Jersey: Prentice Hall, 1994.

MEI, K. Bridging and stuck-at faults. **IEEE Transactions on Computers**, v. 23, n. 7, p. 720–727, jul. 1974. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1672613>>.

PLUSQUELLIC, J. Design For Testability. 2006a. Disponível em: <http://ece-research.unm.edu/jimp/vlsi_test/slides/dft_scan1.pdf>.

PLUSQUELLIC, J. Falha stuck-at. 2006b. Disponível em: <http://ece-research.unm.edu/jimp/vlsi_test/slides/faults2.pdf>.

ROBERTO, B. et al. **Transistor e sua evolução**. 2009. Disponível em: <<http://pt.slideshare.net/brunodemetrio/transstor-e-sua-evoluo>>.

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.

STANOJEVIC, Z.; WALKER, D. FedEx - a fast bridging fault extractor. In: IEEE. **Proceedings...** 2001. p. 696–703. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=966690>>.

TZENG, C.-W.; CHENG, H.-C.; HUANG, S.-Y. 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. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4838835>>.

WAICUKAUSKI, J. et al. Transition fault simulation. **IEEE Design & Test of Computers**, v. 4, n. 2, p. 32–38, 1987. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4069962>>.

WU, C.-H.; LEE, K.-J. An Efficient Diagnosis Pattern Generation Procedure to Distinguish Stuck-at Faults and Bridging Faults. In: IEEE. **Proceedings...** 2014. p. 306–311. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6979118>>.