

# ADDEs – Uma ferramenta para estimar o efeito BTI em portas lógicas CMOS

Schivitz, Rafael B., Meinhardt, Cristina, Butzen, Paulo F.

**Resumo** — Com a redução nas dimensões dos transistores, diversos efeitos indesejáveis, que antes poderiam ser desconsiderados no projeto de um circuito integrado, começaram a tomar proporções cada vez maiores. Um destes efeitos indesejados é chamado de efeito de envelhecimento. Dentre os diversos mecanismos que causam o envelhecimento de um circuito, o BTI (*Bias Temperature Instability*) é considerado o principal. Ele causa um aumento da tensão de limiar dos transistores, provocando uma redução na velocidade de operação do sistema, podendo gerar uma falha na computação. Esse trabalho apresenta o desenvolvimento de uma ferramenta capaz de realizar a estimativa da degradação do atraso causado pelo efeito BTI em portas lógicas CMOS. A informação fornecida pela ferramenta possui grande relevância para o projeto de circuitos mais robustos em relação aos efeitos de envelhecimento.

**Palavras chave:** BTI, envelhecimento, ferramentas de EDA, microeletrônica.

## I. INTRODUÇÃO

A miniaturização dos circuitos integrados (CIs) CMOS, permitiu o desenvolvimento de sistemas mais rápidos e que consomem cada vez menos energia. Contudo, com essa redução nas dimensões dos transistores, novos fatores que antes eram desconsiderados no projeto desses circuitos tiveram que ser levados em conta. Dentre os mais importantes, podemos citar a potência estática, a variabilidade do processo de fabricação e os efeitos de envelhecimento [1]-[3].

Os circuitos integrados apresentam, durante a sua vida útil, um processo de degradação das suas características iniciais. Conforme a sua utilização, o circuito começa a perder as suas características ou modificar o seu comportamento. Esse processo de degradação é chamado de envelhecimento. Ele pode ser definido como um processo lento e cumulativo, onde ocorre um desgaste nas estruturas dos componentes dos circuitos provocados pelo tempo de operação e também pelas condições de funcionamento (temperatura, tensão de alimentação, frequência de operação, dentre outras).

Rafael B. Schivitz é bacharel em Engenharia de Computação e atualmente é aluno de pós-graduação em Engenharia de Computação na Universidade Federal do Rio Grande – FURG (rafaelschivitz@furg.br).

Paulo F. Butzen é doutor em Microeletrônica pela Universidade Federal do Rio Grande do Sul – UFRGS e atualmente atua como professor do Centro de Ciências Computacionais da Universidade Federal do Rio Grande – FURG (paulobutzen@furg.br).

Cristina Meinhardt é doutora em Computação pela Universidade Federal do Rio Grande do Sul – UFRGS e atualmente atua como professora no Centro de Ciências Computacionais da Universidade Federal do Rio Grande - FURG (cristinameinhardt@furg.br)

Diversos fenomenos contribuem para degradar em longo prazo a estrutura dos transistores [4]. Dentre eles, podemos destacar:

- *Bias Temperature Instability* (BTI);
- *Hot Carrier Injection* (HCI);
- *Time Dependent Dielectric Breakdown* (TDDB);

O efeito considerado dominante no processo de envelhecimento é o BTI. Ele é caracterizado por aumentar a tensão de limiar dos transistores,  $V_{th}$ , o que acarreta a degradação da capacidade de condução dos mesmos. Estas alterações em  $V_{th}$  irão modificar os tempos de propagação dos sinais, podendo causar uma falha na computação do sistema. Ao considerar-se aplicações onde a segurança é crítica (por exemplo, para aplicações utilizadas na indústria aeronáutica), uma falha na computação seria de extremo perigo.

Esse efeito é normalmente caracterizado de maneiras diferentes de acordo com cada transistor e sua condição de degradação. Nos transistores NMOS ocorre o efeito PBTI (*Positive Bias Temperature Instability*), enquanto que nos transistores PMOS ocorre o efeito NBTI (*Negative Bias Temperature Instability*). Esse trabalho foca no efeito BTI, cobrindo tanto o NBTI quanto o PBTI.

Os circuitos CMOS, alvos deste trabalho, são projetados usando a metodologia *standard cell*. Esta metodologia explora diversos algoritmos de síntese que a partir da descrição do circuito e de uma biblioteca de portas lógicas, fornecem ao final do processo de síntese o conjunto de portas lógicas já posicionadas e conectadas que irão ser utilizadas para fabricar o circuito desejado. Contudo, os efeitos de envelhecimento não estão diretamente presentes no fluxo de projeto tradicional, o que torna incerto o comportamento dos circuitos ao longo da sua vida útil.

Desta forma, é de grande importância a realização de uma estimativa do efeito de envelhecimento nos circuitos. Como a metodologia *standard cell* utiliza portas lógicas como blocos básicos, a estimativa da degradação nestes componentes torna possível sua análise nas etapas iniciais do projeto. Ao ter a informação sobre um respectivo efeito nas etapas iniciais do projeto abre-se a possibilidade da realização de um projeto de um circuito menos susceptível aos efeitos de envelhecimento. Para a realização dessa estimativa, o desenvolvimento de uma ferramenta que forneça essa informação de forma automática é de grande valor. Essa informação sobre a estimativa de degradação poderá mudar o modo como os circuitos são projetados e torná-los cada vez menos vulneráveis aos efeitos de envelhecimento.

Ao considerar a importância dos efeitos de envelhecimento, obtém-se uma informação de extremo valor ao projetista do circuito. Por isso, a proposta desse trabalho é o desenvolvimento de uma ferramenta que seja capaz de estimar a degradação do efeito BTI no atraso de propagação em portas lógicas CMOS, de forma automatizada e rápida.

Na Seção II são abordados os conceitos de atrasos das portas lógicas seguido por uma breve discussão do fluxo de projeto de um circuito integrado. Por fim, serão apresentados detalhes relacionados com o efeito de envelhecimento BTI. A seguir na Seção III teremos uma explicação sobre como é realizada a estimativa de degradação causada pelo efeito BTI com duas diferentes estratégias. O desenvolvimento da ferramenta para realização dessa estimativa de forma automatizada é mostrado na Seção IV. A Seção V descreve a validação da ferramenta, com um conjunto de portas lógicas e resultados das simulações. Nesta seção também são discutidas possíveis fontes de algumas imprecisões do modelo implementado. Finalmente, a Seção VI apresenta as conclusões obtidas com o desenvolvimento do trabalho proposto.

## II. FUNDAMENTAÇÃO TEÓRICA

Esta seção tem como objetivo apresentar os conhecimentos básicos relacionados com o projeto de circuitos integrados que serão necessários para a melhor compreensão dos métodos de estimativa implementados. A discussão teórica sobre o transistor MOS que pode ser encontrada em [5]. A seguir serão apresentados os conceitos relacionados com os atrasos das portas lógicas, visto que estes serão o alvo da estimativa da degradação. Na sequência será discutido o fluxo de projeto baseado em biblioteca de células, principal ponto de aplicação dos dados gerados. Por fim, serão introduzidos detalhes relacionados com o efeito de envelhecimento investigado no trabalho, o BTI.

### A. Atrasos

Os atrasos das portas lógicas são as informações básicas para a análise do desempenho dos sistemas. É a partir dele que é definida a máxima frequência de operação do sistema. Usualmente, esta frequência é definida como o inverso da soma dos tempos de propagação de cada porta lógica presente no(s) caminho(s) crítico(s). Além dos tempos de propagação, existem os tempos de transição que caracterizam o tempo de transição entre os níveis lógicos. Também tem relação direta com o desempenho do sistema além de serem importantes na minimização do consumo de potência. Estes tempos relacionados com o atraso da porta lógica são apresentados abaixo e estão ilustrados na Fig. 1:

- *Rise time* ( $t_r$ ): Reflete o tempo que o sinal leva para mudar do estado lógico baixo para o estado lógico alto. Normalmente é medido como o intervalo de tempo para que um sinal vá de 10% até 90% do seu valor em tensão.
- *Fall time* ( $t_f$ ): Reflete o tempo que o sinal leva para mudar do estado lógico alto para o estado lógico baixo.

Normalmente é medido como o intervalo de tempo para que um sinal vá de 90% até 10% do seu valor em tensão.

- *High to Low time propagation* ( $T_{PHL}$ ) e *Low to High time propagation* ( $T_{PLH}$ ): É o intervalo de tempo decorrido entre uma variação de sinal na entrada e a correspondente variação na saída. Toma-se como referência o ponto de 50% do valor de tensão.

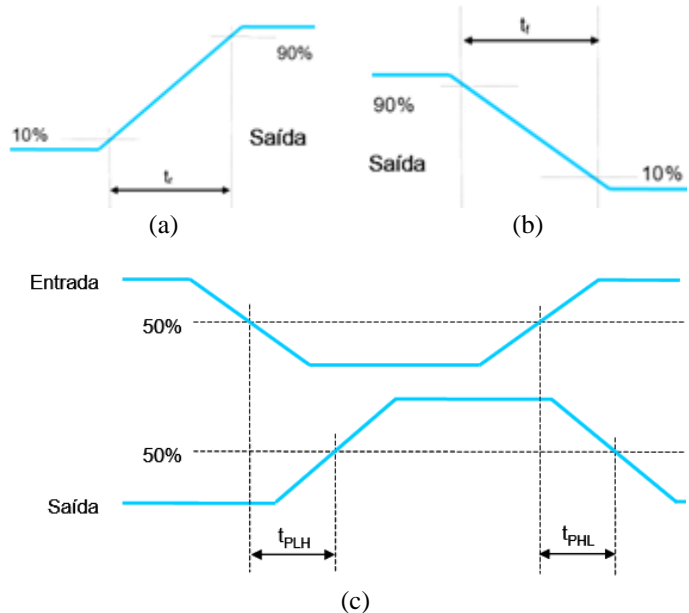


Fig. 1. Os tempos de propagação, (a)  $T_{rise}$ , (b)  $T_{fall}$  e (c)  $T_{PHL}$  e  $T_{PLH}$

Na análise temporal de um sistema, os tempos de propagação ( $T_{PHL}$  e  $T_{PLH}$ ) dos sinais são considerados para definir a frequência do sistema. Conforme a complexidade do circuito, teremos um valor de atraso que deve ser considerado para um correto funcionamento do chip. Por isso, o fluxo de projeto de fabricação considera o atraso das portas lógicas que compõe o circuito para a definição da frequência de operação do mesmo.

### B. Fluxo de Projeto de células

Os circuitos alvo desse trabalho são projetados através de uma metodologia denominada *standard cell*. A metodologia *standard cell* é a mais utilizada no projeto de circuitos integrados. Ela utiliza uma biblioteca de células padrão que contém várias portas previamente projetadas e caracterizadas, o que torna o desenvolvimento dos circuitos que utilizam essa metodologia mais rápidos e simples.

Estas portas lógicas que compõem a biblioteca de células são o bloco básico a ser automaticamente reutilizado no projeto do CI. Essa abordagem possui um custo de projeto bastante baixo devido ao uso de portas lógicas já projetadas e um fluxo automatizado completo [6].

Verificando o fluxo de projeto, é possível perceber que o processo começa com uma descrição comportamental do circuito. A descrição comportamental passa por uma síntese de alto nível, na qual como resultado gera uma descrição em nível de registradores (*Register Transfer Level - RTL*). A partir da descrição RTL é realizada a síntese lógica, que gera uma descrição do circuito em nível de portas lógicas. Por fim, a

síntese física é realizada a qual posiciona o *layout* das portas lógicas e realiza o roteamento dos sinais que irão conectar as mesmas. Esse fluxo de projeto pode ser visualizado na Fig. 2.

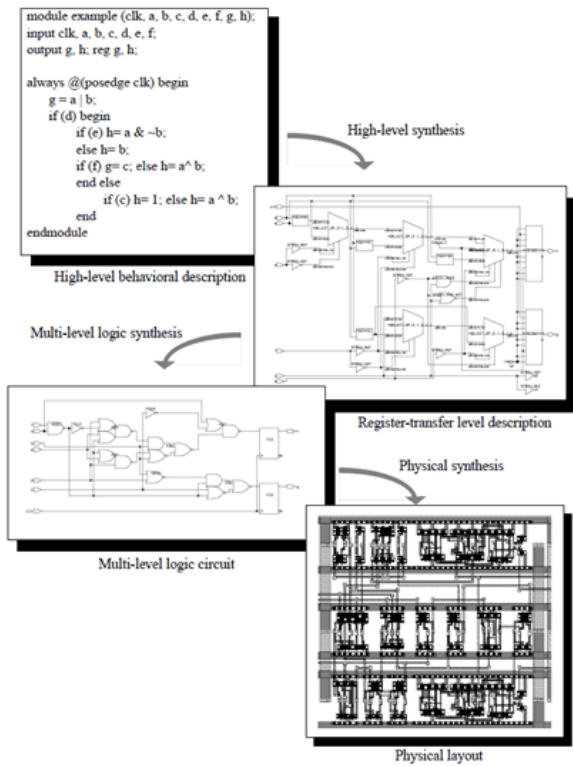


Fig. 2. Fluxo de projeto de células [7].

**C. BTI - Bias Temperature Instability**

Durante a vida útil do dispositivo, assim como nos seres vivos, ocorre um processo de degradação das suas características iniciais [8]. A Fig. 3 ilustra o comportamento dos circuitos durante a sua vida útil. Esta é dividida em 3 partes. A parte inicial mostra a mortalidade chamada “infantil”, onde certo número de componentes após a sua produção, podem apresentar defeitos causados por erros de projeto, por processos de fabricação mal calibrados, entre outros. Já outra parcela possui uma vida de utilização plena, com falhas aleatórias, que podem assumir diversas naturezas, tais como: sobrecargas aleatórias, problemas externos de alimentação elétrica, bruscas variações de temperatura, erros humanos de operação entre outros. Conforme o tempo de utilização, os efeitos como a corrosão, a eletromigração e o desgaste do óxido do *gate* fazem com que uma quantidade maior de equipamentos apresentem falhas excessivas e acabem ficando fora de uso. Para produzir mais produtos que operem corretamente, deve-se aperfeiçoar o processo de fabricação e redobrar o cuidado na verificação do projeto a fim de minimizar a mortalidade chamada “infantil”. Para que os circuitos tenham uma vida útil mais prolongada, deve-se considerar os efeitos de envelhecimento no projeto dos mesmos.

Como dito anteriormente, diversos fatores podem contribuir para a degradação a longo prazo da estrutura do transistor. Dentre os principais efeitos, destaca-se o *Bias Temperature Instability* (BTI).

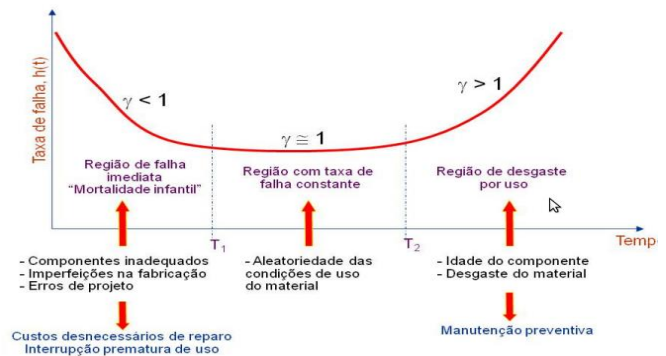


Fig. 3. Tempo de vida dos circuitos [9].

O BTI está associado à geração de defeitos no óxido de *gate* e na interface entre o óxido do *gate* e o substrato do transistor. A geração destes defeitos ocorre devido aos grandes campos elétricos presentes na estrutura do transistor durante sua operação e são acelerados por altas temperaturas [10]. A consequência elétrica do efeito BTI é um aumento na tensão de limiar do transistor ( $V_{th}$ ). Estas alterações de  $V_{th}$  irão modificar os tempos de propagação dos caminhos, causando uma queda no desempenho dos circuitos integrados e podendo causar uma falha de atraso [11].

A condição de degradação mais crítica (maiores campos elétricos) ocorre geralmente quando os transistores estão em estado de condução. A Fig. 4 ilustra a degradação (aumento) da tensão de limiar no transistor devido ao efeito BTI. Na mesma figura, é possível observar que existe uma fase de recuperação, onde a tensão de limiar recupera parte dos defeitos gerados na fase de *stress* e consequentemente o valor da tensão de limiar é reduzido. Esta fase de recuperação se caracteriza pela ausência do forte campo elétrico característico da fase de *stress*. A análise das duas fases é importante, pois em condições normais de operação, o transistor não fica em uma condição de *stress* 100% do tempo, [10].

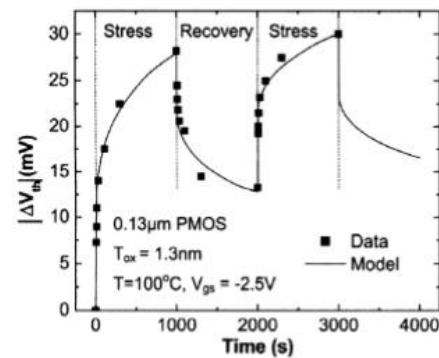


Fig. 4. Comportamento dinâmico do efeito BTI [12].

Ao considerar o efeito BTI, dois diferentes fenômenos podem ser observados dependendo do tipo do transistor. O BTI negativo (NBTI) e BTI positivo (PBTI). A condição de *stress* do fenômeno NBTI ocorre nos transistores PMOS quando uma tensão negativa é observada entre os seus terminais de *gate* e *source*. Nesta condição, o transistor PMOS normalmente está em estado de condução, confirmando a discussão previamente realizada. O fenômeno PBTI afeta os transistores NMOS do circuito. Neste caso, a condição de

V

*stress* ocorre quando uma tensão positiva é observada entre os terminais *gate* e *source* do transistor NMOS. Novamente, é nesta condição que o transistor NMOS está normalmente conduzindo, também confirmando a análise previamente realizada.

III. ESTIMATIVA DA DEGRADAÇÃO CAUSADA PELO EFEITO BTI

Nesse capítulo serão apresentados dois procedimentos para estimar a degradação do atraso das portas lógicas causada pelo efeito BTI em circuitos CMOS: o método dos caminhos condutivos e o método dos arcos de atraso. Serão apresentadas as equações utilizadas para calcular a degradação e cada etapa do procedimento até a estimativa final da degradação. Também veremos uma explicação de como funcionam os dois procedimentos e suas principais diferenças. A parte inicial de ambos os procedimentos consiste na identificação dos transistores degradados e na computação da degradação da tensão de limiar dos mesmos.

O efeito BTI tem sua condição de *stress* bem específica para cada dispositivo. A Fig. 5 mostra as condições nas quais o transistor pode estar sob *stress* dependendo do seu tipo. No caso de um transistor NMOS, temos condição de *stress* quando um valor lógico 1 é aplicado em seu terminal *gate* e valores lógicos 0 são observados em seus terminais *drain* e *source*. Quando olhamos para o transistor PMOS, valores lógicos 0 aplicados no terminal *gate* e valores lógicos 1 nos terminais *drain* e *source* deixam o transistor em condições de *stress* no arranjo.

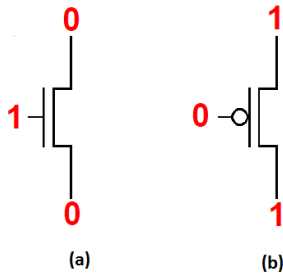


Fig. 5. Condições de *stress* dos transistores, (a) NMOS e (b) PMOS.

A primeira etapa de ambos procedimentos é a identificação da condição de *stress* de cada transistor para cada possível combinação de entrada. A forma de identificar a condição de *stress* é através da simulação lógica do circuito a nível de transistor. Com a simulação lógica realizada no circuito temos as informações de valores de todos os nodos que compõem o circuito. A partir dos valores dos nodos do circuito e dos valores das polaridades dos transistores que ligam os nodos, podemos verificar quais transistores se encontram em condição de *stress* para cada uma das possíveis combinações de entrada.

Com as informações dos transistores em condição de *stress* para cada combinação de entrada, é feito o cálculo do *Transistor Stress Probability* (TSP) dos transistores. O cálculo do TSP é realizado em função da probabilidade do sinal de entrada e a posição do transistor no arranjo [6].

A degradação da tensão de limiar de cada transistor do circuito é obtida com a Eq. 1, onde *A* é uma constante dependente da tecnologia, *t* é o tempo, *n* é a constante de

tempo do BTI e *TSP* é a probabilidade de *stress* do transistor, definido como a probabilidade do transistor PMOS/NMOS estar negativamente/positivamente polarizado :

$$\Delta V_{th\_BTI} = A \cdot (TSP \cdot t)^n \tag{1}$$

Com as informações da simulação lógica, TSP e degradação da tensão de limiar dos transistores processadas pela ferramenta, é possível iniciar o cálculo da estimativa de degradação do atraso das portas lógicas. Para isso, dois procedimentos diferentes, um baseado na estratégia de caminhos condutivos e outro baseado na estratégia de arcos de atraso são explorados [6]. A seguir ambos procedimentos são apresentados em mais detalhes.

A. Estratégia de Arcos de atraso

Um arco de atraso corresponde à alteração do valor de saída causado por uma transição de uma das entradas em uma combinação específica, enquanto as outras entradas do circuito permanecem constantes. A Fig. 6 ilustra dois arcos de atraso de uma porta lógica OR. Nela observa-se que uma variação na entrada “b”, enquanto “a” permanece constante, ocasiona uma variação na saída. Essa transição de “b” gera dois tipos de arcos: o tempo de propagação de subida (*T<sub>PLH</sub>*), gerado pela variação 0 → 1 de “b”, onde a saída vai de estado lógico baixo para estado lógico alto, e o tempo de propagação de descida (*T<sub>PHL</sub>*), gerado pela variação 1 → 0 de “b”, onde a saída vai de estado lógico alto para estado lógico baixo.

	Combinação			
	a	b	out	
TPHL	0	0	0	TPLH
	1	0	1	
	2	1	0	
	3	1	1	

Fig. 6. Exemplo de arco.

Conhecendo a complexidade do problema, é possível verificar algumas condições para estabelecer como o cálculo da degradação será realizado:

- Quando o arco de atraso é associado somente à um dispositivo, a degradação da tensão de limiar do dispositivo é diretamente associada à degradação do arco de atraso, como vemos em Eq. 2. Observando a Eq. 2,  $\Delta V_{th\_DEVICE}$  é a degradação do transistor e  $V_{th\_NOMINAL}$  é a tensão de limiar do transistor.

$$D_{ARCO\_1\_Dev} = \frac{\Delta V_{th\_DEVICE}}{V_{th\_NOMINAL}} * C_{PMOS} \tag{2}$$

Outra equação bastante semelhante a Eq. 2 é a sua variação, visto que para o cálculo de um arco de subida, utilizamos um coeficiente PMOS, descrito como *C<sub>PMOS</sub>*. Na equação do cálculo de um arco de descida vemos uma pequena diferença no coeficiente, onde é utilizado um coeficiente NMOS, descrito como *C<sub>NMOS</sub>*. O cálculo realizado para obtermos os valores desses coeficientes foram baseados nas informações de atraso do *gate* de um inversor nas condições *fresh* e degradado.

- Quando o arco de atraso está associado a mais de um transistor, a degradação da tensão de limiar desses transistores contribui ao atraso do arco com magnitudes diferentes. O transistor que transiciona possui a sua degradação associada ao arco de atraso com base em um coeficiente PMOS ou NMOS, como mostramos na Eq. 2 enquanto os outros transistores no conjunto apresentam uma contribuição reduzida quando comparados às suas degradações das tensões de limiar. As Eq. 3 e Eq. 4 ilustram como é feito o cálculo no caso observado, nesse caso temos um transistor que transiciona e outros transistores que participam na degradação do arco, mas com magnitudes diferentes. Nas equações podemos destacar  $CR_{NMOS}$  e  $CR_{PMOS}$  que equivalem à um coeficiente de redução de tensão de limiar de NMOS e PMOS, respectivamente. Essas constantes de redução foram usadas para calibrar a ferramenta e foram calculadas a partir da informação do atraso do *gate* de uma *NAND* em condições *fresh* e degradada.

$$D_{ARCO\_n\_Dev} = \frac{\Delta V_{th\_DEV} * C_{PMOS} + \left( CR_{PMOS} * \frac{\sum \Delta V_{th\_DEV\_NO\_ARCO}}{num\_dev\_no\_arco} \right)}{V_{th\_NOMINAL}} \quad (3)$$

$$D_{ARCO\_n\_Dev} = \frac{\Delta V_{th\_DEV} * C_{NMOS} + \left( CR_{NMOS} * \frac{\sum \Delta V_{th\_DEV\_NO\_ARCO}}{\#dev\_no\_arco} \right)}{V_{th\_NOMINAL}} \quad (4)$$

Com o valor da degradação de todos os arcos computados, tem-se que a degradação média do atraso da porta lógica (GDD) é simplesmente a média da degradação de todos os arcos do circuito, como apresentada na Eq. 5. O procedimento para implementação de todo o processo apresentado anteriormente é mostrado em Algoritmo 1.

$$GDD = \frac{\sum Degradacao\_arcos}{num\_arcos} \quad (5)$$

ALGORITMO 1: PROCEDIMENTO DE IMPLEMENTAÇÃO DOS ARCOS DE ATRASO

1. Verifica a saída da tabela verdade em hexadecimal.
2. A partir da saída em hexadecimal, cria os arcos de atraso.
3. Para cada arco de atraso
4. {
5. Analisa transistor principal e os transistores que participam da degradação
6. Computa a degradação do arco de acordo com a análise
7. }
8. Calcula a degradação do atraso do *gate* a partir da degradação dos arcos.

### B. Estratégia de Caminhos condutivos

Os caminhos condutivos do transistor correspondem a todos os caminhos lógicos que ligam o nó de saída e a fonte de alimentação (VDD) ou a referência de terra (GND). Tais caminhos condutivos são responsáveis pela transição do sinal de saída, sendo fortemente associados ao atraso do *gate*.

O método leva em consideração os caminhos de condução do arranjo de transistores e a degradação da tensão de limiar do transistor. Para calcular a degradação do dispositivo, o conceito de “transistor stress probability” é aplicado [13]. A análise dos caminhos condutivos explora algumas características, tais como:

- Quando o caminho possui somente um transistor, a degradação do transistor é diretamente associada à degradação do caminho, sendo assim, a equação do método dos caminhos é a mesma utilizada no método dos arcos vista em Eq. 2.
- Quando o caminho possui mais de um transistor, é simplesmente realizada a média da degradação de todos os transistores que pertencem ao caminho. Essa característica nos dá uma redução significativa no tempo de execução, mas não nos permite a identificação de um transistor que transiciona no conjunto. Por essa razão, a degradação média de todos os transistores no conjunto é calculada e considerada como a contribuição direta à degradação do atraso da porta lógica. A expressão usada para calcular a degradação do caminho nos caminhos condutivos formados por um conjunto de transistores é descrita nas Eq. 6 e Eq. 7. Nelas, destacamos *AVDP* que é a média da degradação da tensão de limiar dos transistores que pertencem ao caminho. *NTP* é o número de transistores dentro do caminho condutivo. O termo *NSR* se refere à contribuição dos transistores que não transicionam e que pertencem ao caminho condutivo quando uma transição ocorre no nodo de saída, nos cálculos foi utilizado um valor equivalente à “*NTP+1*”. Já o  $V_{th\_NOMINAL}$  é a tensão de limiar nominal do transistor [2].

$$PD_{PMOS} = \frac{AVDP \left( 1 + \frac{NTP - 1}{NSR} \right)}{V_{th\_NOMINAL}} * C_{PMOS} \quad (6)$$

$$PD_{NMOS} = \frac{AVDP \left( 1 + \frac{NTP - 1}{NSR} \right)}{V_{th\_NOMINAL}} * C_{NMOS} \quad (7)$$

Depois de calcular toda a degradação dos caminhos, a degradação do atraso da porta lógica é simplesmente a média de todos os caminhos, sendo utilizada a Eq. 5.

## IV. DESENVOLVIMENTO DA FERRAMENTA

A ferramenta foi desenvolvida utilizando a linguagem de programação Java, devido a sua portabilidade, permitindo que qualquer sistema operacional possa executar a ferramenta, precisando somente de um interpretador de Java. Além da portabilidade da linguagem, ela possui uma grande facilidade de programação, totalmente orientada a objetos, permitindo herança e reutilização de trechos de código de forma dinâmica e estática.

O Ambiente de Desenvolvimento Integrado (*IDE*) utilizado para implementar a ferramenta é o *NetBeans 8.0* [14]. A escolha do *NetBeans* se deve a facilidade fornecida para programadores em escrever, compilar e depurar códigos, visto que ele já possui um pré-compilador embutido. Foi criado em

forma de uma estrutura reutilizável que visa simplificar o desenvolvimento e aumentar a produtividade.

O fluxograma de funcionamento da ferramenta está ilustrado na Fig. 7. A ferramenta recebe os dados de entrada e armazena os mesmos na estrutura de dados desenvolvida. Com os dados processados é realizada a simulação lógica do circuito para todas as combinações possíveis dos sinais de entrada. Com essa informação, a ferramenta é capaz de verificar quais transistores estão em condições de *stress* e assim é possível computar o *TSP* dos transistores. Com as informações obtidas nos processos anteriores, a ferramenta calcula a degradação da tensão de limiar de cada transistor. Nessa etapa, toda a informação básica para os dois métodos já foi processada. A ferramenta então computa as estimativas com base na estratégia dos caminhos condutivos e na estratégia dos arcos de atraso. Após a estimativa da degradação do circuito ser realizada, a ferramenta faz uma organização desses dados de forma a otimizar a devolução dessas informações de todas as etapas do processamento para o usuário.

#### A. Fluxograma e métodos de operação da ferramenta

Nesta seção serão apresentados os métodos de operação da ferramenta, assim como o seu fluxograma. As etapas principais são divididas em sub-seções das quais apresentarão informações sobre como foram desenvolvidas e seus pseudocódigos.

##### 1. Simulação Lógica

Após o processamento dos dados de entrada, a ferramenta realiza uma organização das informações de forma a possibilitar a correta execução dos procedimentos em sequência. Para a validação lógica, a ferramenta percorre todos os transistores do circuito e realiza as operações de validação. Primeiramente verifica-se o tipo do transistor, se ele é PMOS ou NMOS. Após é realizada a verificação da polaridade do transistor. Neste caso, dependendo da polaridade, faz-se a verificação dos nodos que o transistor liga.

Após todas essas verificações, o valor dos nodos é passado do nodo que já está preenchido para o nodo que ainda não se conhece o valor, esse procedimento se repete enquanto tivermos nodos do circuito sem valor definido. Contudo, em certas combinações, alguns nodos do circuito não são percorridos. Isso ocorre devido ao fato de que em determinadas combinações, alguns transistores não estarão em estado de condução.

Na Fig. 8, por exemplo, aplicando valores lógicos “1” para os transistores “b” e “c” e observando o plano *pull-up* é verificado que os transistores não conduzirão, fazendo com que o nodo presente entre esses transistores e o transistor “a” não seja percorrido, não sendo conhecido o seu valor no circuito. Então, para evitar que a ferramenta entre em estado de loop infinito, o número de verificações possui um limite máximo para evitar esse problema. Os valores que não são conhecidos são descritos com “Z”, que significam alta impedância do nodo. O procedimento da simulação lógica pode ser descrito no Algoritmo 2. No algoritmo pode-se verificar na linha 5 que todo o procedimento irá executar por no máximo um “MAX” de simulações, no pior caso. Esse valor

é definido como um número de simulações máximo que seja capaz de realizar a validação lógica de uma porta lógica e nesse trabalho foi utilizado nos experimentos um valor  $MAX = 10$ .

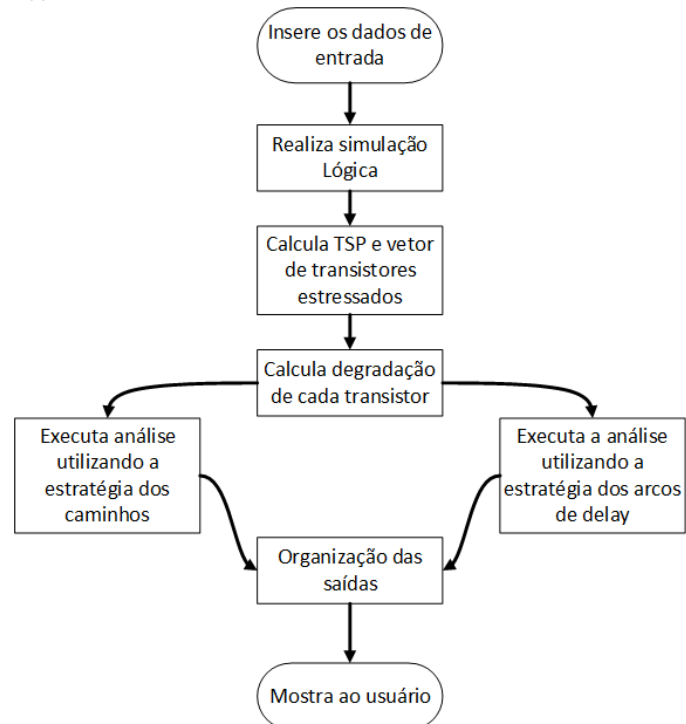


Fig. 7. Fluxograma da ferramenta.

#### ALGORITMO 2: PROCEDIMENTO DA SIMULAÇÃO LÓGICA

```

1. Organiza entradas do circuito
2. Inicializa parâmetros de simulação;
3. Para cada combinação de entrada
4. {
5. Enquanto (analise_nodos==false)&&(num_simulacoes<MAX)
6. {
7. Para cada transistor
8. {
9. Verifica características do transistor
10. Se (transistor_conduzindo==true)
11. {
12. Atualiza nodos intermediários do transistor
13. }
14. }
15. }
16. }
  
```

##### 2. Calculo do TSP

Depois do procedimento da execução da simulação lógica ser finalizado, conhecemos o valor de cada nodo do circuito. Com isso, podemos realizar uma simples verificação para cada transistor, onde o procedimento de verificação de *stress* se baseia em 3 verificações: a característica de condução do transistor, o valor do sinal do *gate* e os valores dos dois nodos que ele conecta (correspondentes aos valores de *drain* e *source*). Ao final de todos os transistores serem verificados, temos essa informação inserida e atualizada na classe transistor. A realização desse procedimento é necessária para

cada combinação das entradas do circuito, visto que com cada combinação de entrada, temos valores diferentes nos nodos do circuito.

Ao final do procedimento é montada a tabela de transistores estressados. Essa tabela possibilita que a ferramenta calcule o *Transistor Stress Probability* para cada transistor no arranjo. Com os valores das probabilidades de as entradas serem 1 inseridas no início da execução, a probabilidade de ocorrência de cada combinação das entradas é calculada. Para cada combinação de entrada, temos um vetor de transistores estressados. Assim, a saída é composta pelos devidos valores de combinação de entrada, lista dos transistores PMOS em condição de *stress* e lista de transistores NMOS em condição de *stress*, seguidos pela porcentagem de ocorrência da combinação de entrada.

O valor do TSP foi calculado como o somatório dos valores das linhas onde o transistor se encontra em condição de *stress*. Com a análise de quais transistores estão em condição de *stress* nas combinações, é possível o cálculo do TSP de cada transistor.

### 3. Calculo da Degradação do $V_{th}$ de cada transistor

Com as informações da TSP dos transistores é possível que a ferramenta calcule diretamente a degradação de cada transistor. A Eq. 1 é diretamente aplicada com os valores do TSP, tempo,  $n$  e  $A$ .

Observando o fluxograma da Fig. 7, ao final do procedimento de cálculo da degradação da tensão de limiar de cada transistor, todas as informações necessárias para os procedimentos a seguir já foram levantadas. Assim, a ferramenta realiza dois tipos de análises no *netlist*, as quais foram vistas no capítulo anterior e serão explicadas como foram implementadas a seguir.

### 4. Implementação do procedimento da análise dos caminhos condutivos

A ferramenta em sua execução, monta um grafo do circuito a partir da descrição do *netlist*. Como o processamento de um grafo possui tempo e custo elevados, a ferramenta forma a partir do grafo do circuito uma matriz de adjacências. O uso desta facilita a busca e diminui o custo de processamento da ferramenta em suas principais atividades. A matriz de adjacência é uma matriz  $N \times N$  (onde  $N$  é o número de nodos do grafo). Ela inicialmente é preenchida toda com 0 e quando há uma relação entre o vértice do  $x$  (número da coluna) com o do  $y$  (número da linha), na *matriz[x][y]* adicionamos o transistor que liga os nodos. Assim, ao final do preenchimento da matriz, teremos uma *matriz[x][y]*, com uma lista de transistores que ligam os nodos em cada combinação de linha por coluna.

Com uma matriz de adjacências acabamos tendo um desperdício de memória, pois quando temos grafos esparsos a matriz fica preenchida em sua maioria com zeros. Porém, com a implementação da matriz, a busca se existe um vértice (transistor) entre dois nodos se torna linear, somente é necessário que seja feita uma consulta em *matriz[x][y]*.

A utilização da tabela consome mais memória, mas temos um ganho significativo nas buscas, que é a principal função utilizada no procedimento para encontrar os caminhos. Uma das características da matriz de adjacência quando o grafo não é orientado é a simetria encontrada na “diagonal”. É interessante que se lermos uma coluna de índice  $v$  ou uma linha de índice  $v$  o mesmo valor será verificado.

No procedimento de criação dos caminhos PMOS e NMOS, temos uma estratégia recursiva na criação dos caminhos. Quando um transistor que liga VDD (no caso dos caminhos PMOS) ou GND (no caso dos caminhos NMOS) é encontrado, ele é adicionado no caminho e assim passa para o próximo transistor a ser avaliado em um procedimento que sempre é chamado, até que o próximo nodo seja igual à OUT, sinalizando que o caminho chegou ao fim.

Esse procedimento precisa ser bastante cuidadoso, pois como não temos um grafo dirigido, podemos facilmente cair em um ciclo e assim a análise não será válida. Para isso, a ferramenta precisa ser capaz de tratar qualquer imprevisto no método de criação dos caminhos, de forma a eliminar as opções com ciclos.

A ferramenta a partir dessas análises cria todos os caminhos possíveis para cada circuito. Para exemplificar a Tab. I é criada a partir do procedimento dos caminhos ser executado na Fig. 8. Nela, temos todos os caminhos criados e separados pelos planos *pull-up* e *pull-down*.

TABELA I. CAMINHOS GERADOS EM UMA AOI21 PELA FERRAMENTA

Camínhos	<i>pull-up</i>	<i>pull-down</i>
0	B <sub>PMOS</sub> , A <sub>PMOS</sub>	A <sub>NMOS</sub>
1	C <sub>PMOS</sub> , A <sub>PMOS</sub>	B <sub>NMOS</sub> , C <sub>NMOS</sub>

Com os caminhos criados pela ferramenta, é possível fazer uma análise para calcular a degradação do caminho, como visto na Seção anterior. Para isso, o procedimento é bastante simples. Para possibilitar ao usuário verificar a degradação do plano *pull-up* ou *pull-down* do circuito, a ferramenta separa a informação dos caminhos em duas listas individuais, uma com os caminhos PMOS e a outra com os caminhos NMOS.

Após conhecer os caminhos do circuito a ferramenta calcula a degradação dos caminhos PMOS e NMOS. O cálculo dessa parte é feito com uma verificação de quais transistores pertencem ao caminho e assim a degradação destes transistores é computada. Com a informação da degradação de todos os caminhos, o próximo passo é de realizar a média de toda degradação dos caminhos apresentada na Eq. 5. A saída para esse procedimento é mostrada na tela principal da ferramenta. O Algoritmo 2 implementa a descrição anterior.

ALGORITMO 3: PROCEDIMENTO PARA GERAÇÃO DOS CAMINHOS

1. Analisa o arranjo do circuito
2. Gera os caminhos PMOS e NMOS do circuito.
3. Para cada caminho
4. {
5. Analisa os transistores que participam do caminho
6. Computa a média da degradação dos transistores
7. }
8. Calcula a degradação do atraso do *gate* a partir da degradação dos caminhos

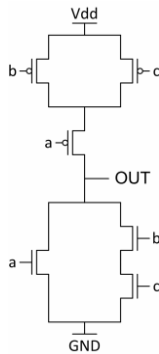


Fig. 8. Organização de uma AOI21.

5. Implementação do procedimento da análise dos arcos de atraso

O procedimento dos arcos de atraso necessita conhecer a função implementada no *netlist* para poder computar os arcos de atraso. Desta forma, utiliza-se a informação obtida através da simulação lógica para conhecer a saída da tabela verdade da função. O procedimento utilizado para buscar os arcos a partir da função lógica realiza um mascaramento de bits na tabela verdade da função para verificar as linhas da tabela onde somente uma das entradas estará causando uma transição da saída, caracterizando desta forma, um arco de atraso da função.

Para computar a degradação do arco de atraso de acordo com a metodologia apresentada na Seção anterior, a ferramenta faz uma análise dos transistores de interesse dependendo do arco à ser analisado. A Tab. II nos mostra os transistores de interesse conforme o tipo de arco e seus respectivos valores para uma AOI21. Somente a informação dos arcos e o tipo de arco não garantem a correta identificação dos transistores de interesse. Para isso, uma análise com os caminhos é realizada de forma a evitar que um transistor que esteja presente no arco, mas não esteja presente num caminho seja computado. Por exemplo, na linha 7 da Tab. II, a variação do valor de “*A<sub>NMOS</sub>*” causa uma transição na saída, enquanto “*B<sub>NMOS</sub>*” e “*C<sub>NMOS</sub>*” permanecem com seus valores iguais a 1 e 0, respectivamente. Como temos um arco de descida, olhamos para o plano *pull-down* do circuito. Logo, o valor de degradação de “*B<sub>NMOS</sub>*” não é de interesse por não conter um caminho com “*A<sub>NMOS</sub>*” e “*B<sub>NMOS</sub>*”.

TABELA II. ARCOS DE ATRASO GERADOS EM UMA AOI21 PELA FERRAMENTA

Tipo do Arco	Nome do arco	Transistores que participam da degradação
TP <sub>HL</sub>	<i>C<sub>NMOS</sub>_A<sub>NMOS</sub>0_ B<sub>NMOS</sub>1</i>	<i>C<sub>NMOS</sub>, B<sub>NMOS</sub></i>
TP <sub>LH</sub>	<i>C<sub>PMOS</sub>_A<sub>PMOS</sub>0_ B<sub>PMOS</sub>1</i>	<i>C<sub>PMOS</sub>, A<sub>PMOS</sub></i>
TP <sub>HL</sub>	<i>B<sub>NMOS</sub>_A<sub>NMOS</sub>0_ C<sub>NMOS</sub>1</i>	<i>B<sub>NMOS</sub>, C<sub>NMOS</sub></i>
TP <sub>LH</sub>	<i>B<sub>PMOS</sub>_A<sub>PMOS</sub>0_ C<sub>PMOS</sub>1</i>	<i>B<sub>PMOS</sub>, A<sub>PMOS</sub></i>
TP <sub>HL</sub>	<i>A<sub>NMOS</sub>_B<sub>NMOS</sub>0_ C<sub>NMOS</sub>0</i>	<i>A<sub>NMOS</sub></i>
TP <sub>HL</sub>	<i>A<sub>NMOS</sub>_B<sub>NMOS</sub>0_ C<sub>NMOS</sub>1</i>	<i>A<sub>NMOS</sub></i>
TP <sub>HL</sub>	<i>A<sub>NMOS</sub>_B<sub>NMOS</sub>1_ C<sub>NMOS</sub>0</i>	<i>A<sub>NMOS</sub></i>
TP <sub>LH</sub>	<i>A<sub>PMOS</sub>_B<sub>PMOS</sub>0_ C<sub>PMOS</sub>0</i>	<i>A<sub>PMOS</sub>, B<sub>PMOS</sub>, C<sub>PMOS</sub></i>
TP <sub>LH</sub>	<i>A<sub>PMOS</sub>_B<sub>PMOS</sub>0_ C<sub>PMOS</sub>1</i>	<i>A<sub>PMOS</sub>, B<sub>PMOS</sub></i>
TP <sub>LH</sub>	<i>A<sub>PMOS</sub>_B<sub>PMOS</sub>1_ C<sub>PMOS</sub>0</i>	<i>A<sub>PMOS</sub>, C<sub>PMOS</sub></i>

Por fim, a degradação média do atraso do *gate* corresponde à média da degradação de todos os arcos do circuito. Esta informação é apresentada juntamente com a informação da degradação média dos caminhos na janela de saída da ferramenta.

Após executarmos a ferramenta, o usuário se depara com a solicitação de todas as informações necessárias. Após fornecer essas informações, o usuário executa a mesma e a ferramenta apresenta como resultado a janela ilustrada na Fig. 9. Nesta janela o usuário tem a possibilidade de visualizar somente a informação de seu interesse. No caso de um usuário que somente quer realizar a validação lógica de um *netlist*, ele pode clicar no botão “Simulação Lógica” e obter toda a informação de valor dos nodos do circuito.

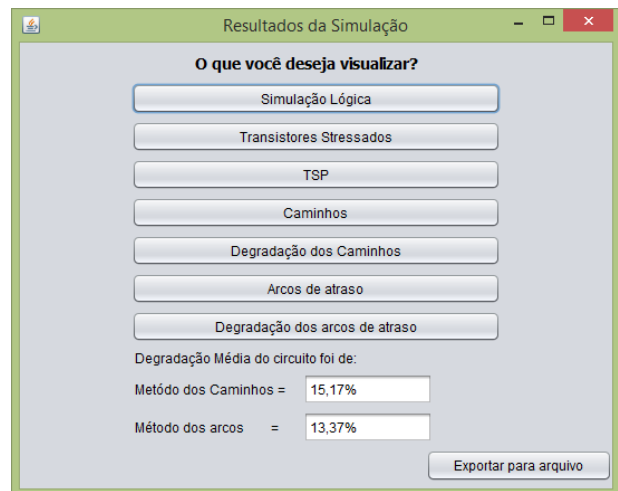


Fig. 9. Saída da ferramenta e suas possibilidades de visualização.

V. VALIDAÇÃO DA FERRAMENTA

Durante todo o processo de desenvolvimento, a ferramenta foi submetida a vários testes com diferentes circuitos. Isso significa que para cada etapa desenvolvida da ferramenta, ela passava por diversos testes com um conjunto de portas lógicas. O conjunto de portas lógicas utilizados para a validação consistia em *Inversor*, *NAND* e *NOR* de 2 e 3 entradas, *AOI21*, *OAI21* e também dois arranjos de transistores denominados *bridge* e *CLA unit*. Toda a etapa desenvolvida era validada para cada uma das portas lógicas do conjunto e somente após essa validação era iniciada a implementação de uma nova funcionalidade. A sequencia de implementação das funcionalidades segue a sequencia dos blocos apresentados no fluxograma da Fig. 7.

As simulações realizadas pela ferramenta foram feitas utilizando os valores contidos na Tab. III. Os valores da tensão de alimentação, *V<sub>th</sub>* nominal e temperatura estão relacionados às características da tecnologia preditiva de 32 nanômetros utilizada na descrição dos circuitos. O tempo de 3 anos foi utilizado como base para a extração da constante *A* da Eq. 2, considerando que a degradação esperada para um transistor neste período seja de 50 mV. O valor de *n* é característica do efeito BTI [11]. As constantes restantes foram obtidas



empiricamente conforme descrito na apresentação dos modelos na Seção III.

Tecnologia	32 nm (Preditiva) [15]
Tensão de alimentação	1 V
Temperatura	100 °C
$V_{th\_nominal}$	340 mV
A	0,002342
t	94608000 s (3 anos)
n	0,166667
$C_{NMOS}$	0,79
$C_{PMOS}$	1,08
$CR_{NMOS}$	0,16
$CR_{PMOS}$	0,15

#### A. Análise do Tempo de Execução

Um dos pontos a serem observados na validação da ferramenta foi o tempo de execução da mesma para cada um dos métodos. A Tab. IV apresenta os valores para o conjunto de portas lógicas avaliadas. Ao analisar os dados apresentados, verifica-se que para as portas mais simples, o procedimento dos arcos de atraso apresenta melhor desempenho. A medida que a função lógica torna-se mais complexa o procedimento dos caminhos condutivos é o mais rápido.

TABELA IV. TEMPO DE EXECUÇÃO DA FERRAMENTA PARA UM CONJUNTO DE PORTAS LÓGICAS

Porta lógica	Tempo de execução (ms)	
	procedimento dos caminhos	procedimento dos arcos
<b>Inv</b>	11	06
<b>Nand2</b>	11	08
<b>Nand3</b>	13	07
<b>Nor2</b>	12	05
<b>Nor3</b>	15	07
<b>AOI21</b>	15	09
<b>OAI21</b>	15	08
<b>Bridge</b>	37	38
<b>CLA unit</b>	93	195

O fato do procedimento dos arcos de atraso atingirem melhor desempenho do que o procedimento dos caminhos condutivos em alguns circuitos vai de encontro com o esperado pela análise da quantidade de arcos e caminhos de cada porta lógica. A Fig. 10 apresenta a quantidade de arcos e caminhos condutivos para cada uma das portas lógicas analisadas. Como esperado, a quantidade de caminhos condutivos sempre é menor ou igual à quantidade de arcos de atraso. Desta forma, esperava-se que o procedimento dos caminhos condutivos sempre apresentasse o melhor desempenho. O comportamento inverso observado nas portas lógicas mais simples pode ser esclarecido pelo fato da identificação dos caminhos condutivos em um arranjo de transistores ser um procedimento mais custoso (devido a várias verificações que são realizadas para evitar loops) do que

a identificação dos arcos de atraso de uma função lógica. A medida que a quantidade de arcos da função aumenta, o comportamento da ferramenta é o esperado.

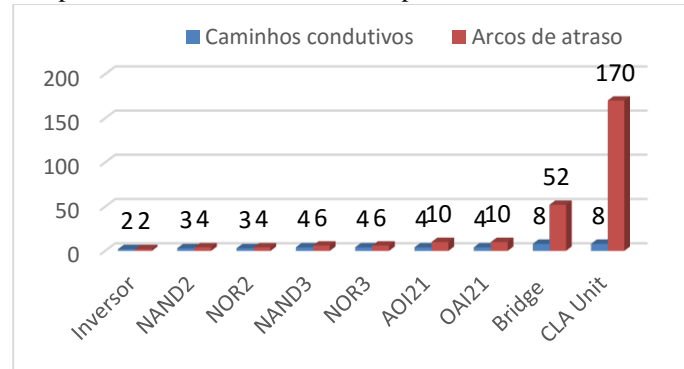


Fig. 10. Diferenças do número de caminhos condutivos e arcos de atraso para um conjunto de portas lógicas usadas nesse trabalho.

#### B. Comparação dos resultados com simulação elétrica

Um último experimento foi realizado para verificar a fidelidade dos modelos implementados quando comparados com dados das simulações elétricas. As simulações elétricas foram realizadas utilizando o *NGSPICE* em conjunto com uma ferramenta de caracterização temporal [16][14]. Foram geradas duas de cada uma das portas lógicas avaliadas: uma ideal e outra com a tensão de limiar dos transistores degradada. Simulações elétricas foram realizadas para obtenção dos dados de atraso. A diferença entre a média dos atrasos de cada uma das versões é comparada com os dados fornecidos pela ferramenta. A Tab. V mostra essa informação. Nela é apresentado o conjunto de portas lógicas, e a média da degradação no atraso pelo método dos caminhos condutivos, pelo método dos arcos de atraso e os dados extraídos da simulação elétrica.

Excluindo os arranjos mais complexos, *Bridge* e *CLA Unit*, todos os procedimentos apresentam valores próximos ao obtidos na simulação elétrica. Na maioria dessas portas lógicas mais simples, o método dos arcos de atraso se mostrou mais preciso do que o método dos caminhos condutivos. Isso ocorre devido ao método dos arcos atuar individualmente em cada arco de atraso enquanto que o método dos caminhos condutivos realiza uma simplificação ao unir diversos arcos em um único caminho. Nas portas lógicas mais complexas, a precisão dos dois métodos ficou comprometida. Abaixo são discutidas algumas das possíveis causas destas diferenças. Antes, vale destacar que o desenvolvimento dessa ferramenta tornará muito mais rápida a investigação das imprecisões e o aprimoramento destes modelos ao possibilitar a geração de novos dados de forma automática.

Durante a análise dos dados foram encontradas duas principais fontes de imprecisão dos modelos. Uma das fontes de imprecisão é a influência do plano complementar nos atrasos. Para exemplificar, ao considerarmos a porta *NAND2*, o plano *pull-up* tem dois transistores em paralelo. Quando a probabilidade dos sinais de entrada é a mesma, ambos têm a mesma degradação e consequentemente os modelos computam a mesma degradação no atraso. Contudo, os dados simulados mostram que a degradação é diferente. Isso ocorre,

pois, dependendo do arco analisado, os planos complementares possuem mais ou menos participação na definição do atraso.

Outra fonte das imprecisões do modelo é o valor da porcentagem da degradação estimada. Observamos que o aumento da porcentagem da degradação estimada acompanhou o aumento da degradação absoluta simulada, ao invés de acompanhar o aumento da porcentagem da degradação simulada. Isso ocorreu devido às diferenças absolutas nos valores obtidos dos atrasos.

Outros pontos que devem ser levados em conta na análise são os arredondamentos nos valores dos parâmetros e nos cálculos dos modelos. Juntamente com as duas fontes de imprecisão do modelo contribuem para as diferenças observadas entre os dados simulados e os obtidos pelos modelos.

TABELA V. SAÍDA DA FERRAMENTA PELOS DOIS PROCEDIMENTOS E RESULTADO DA SIMULAÇÃO ELÉTRICA

Porta lógica	Média da estimativa de degradação do <i>gate delay</i>		
	por caminhos condutivos (%)	por arcos de atraso (%)	pela simulação (%)
<b>Inv</b>	12,25	12,25	12,38
<b>Nand2</b>	13,78	12,96	12,10
<b>Nand3</b>	14,08	13,58	13,25
<b>Nor2</b>	12,84	12,79	12,07
<b>Nor3</b>	12,51	13,26	11,70
<b>AOI21</b>	15,17	13,37	13,60
<b>OAI21</b>	14,88	13,45	13,32
<b>Bridge</b>	16,78	13,60	19,49
<b>CLA unit</b>	17,17	13,57	11,95

## VI. CONCLUSÃO

Com o desenvolvimento da ferramenta, a estimativa do envelhecimento em portas lógicas CMOS pode ser realizada de uma maneira rápida e automatizada. Com a informação que ela disponibiliza, é possível o desenvolvimento de ferramentas de síntese que tenham como objetivo o projeto de sistemas mais resistentes em relação ao efeito BTI e também de ferramentas de análise para avaliação da degradação dos circuitos já projetados.

Destaca-se também a estratégia de implementação na forma de módulos distintos como outro diferencial que pode ser explorado. Neste sentido, a informação referente à simulação lógica do circuito pode ser utilizada na análise da confiabilidade de uma porta lógica na presença de falhas e também o cálculo do TSP de cada transistor que pode ser usado diretamente no projeto de blocos mais resistentes ao envelhecimento.

Por fim, o processo de desenvolvimento da ferramenta permitiu a análise e comparação dos valores gerados com valores obtidos a partir da simulação elétrica. Estes dados permitem o estudo e desenvolvimento de novas alternativas que explorem os pontos fracos encontrados nessas metodologias implementadas.

## VII. REFERÊNCIAS

- [1]. Roy, K. et. al. "Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits," Proceedings of the IEEE, vol.91, no.2, pp.305,327, Feb 2003.
- [2]. Saha, S.K., "Modeling Process Variability in Scaled CMOS Technology," Design & Test of Computers, IEEE, vol.27, no.2, pp.8,16, March-April 2010.
- [3]. Borkar, S., "Designing reliable systems from unreliable components: the challenges of transistor variability and degradation," Micro, IEEE, vol.25, no.6, pp.10,16, Nov.-Dec. 2005.
- [4]. Maricau, e.; gielen, g. (2013). Analog ic reliability in nanometer cmos.
- [5]. Weste, N. H. E.; Harris, D. M. (2011.). Cmos Vlsi Design A Circuits And Systems Perspective (4ª Ed.). Editora Pearson.
- [6]. Butzen, P. F. (2012). Aging aware design techniques and cmos gate degradation estimative. Porto alegre, rs, brasil.
- [7]. KRAVETS, V. N. (2001). Constructive multi-level synthesis by way of functional properties. University of Michigan.
- [8]. Pachito, J. D. (2012). Metodologia para prever o envelhecimento de circuitos digitais.
- [9]. Reis, R. A. (2008). Concepção de circuitos integrados (2 ed., vol. 7).
- [10]. Butzen, P., & al, e. (s.d.). Efeitos físicos nanométricos em circuitos integrados digitais.
- [11]. Stathis, J. H., & Zafar, S. (microelectronics reliability 46 (2006)). The nbtI in mos devices: a review.
- [12]. Vattikonda. R. (2006). Modeling and minimization of pmos nbtI effect for robust nanometer design.
- [13]. Paul, B.c. et al. (2006). Temporal performance degradation under nbtI: estimation and design for improved reliability of nanoscale circuits. 780-785.
- [14]. Netbeans ide 8.0. Fonte: <https://netbeans.org/>
- [15]. W. Zhao, Y. Cao, "New generation of Predictive Technology Model for sub-45nm early design exploration," IEEE Transactions on Electron Devices, vol. 53, no. 11, pp. 2816-2823, November 2006.
- [16]. Machado, I. et. al. Caracterização Elétrica Automática de Portas Lógicas CMOS Explorando um Algoritmo Guloso. Iberchip Workshop, Santiago.