

CREsT - Uma Ferramenta para o Auxílio do Ensino de Confiabilidade em Circuitos Digitais

Laura Quevedo Jurgina¹, Matheus F. Pontes¹, Clayton R. Farias³, Guilherme Manske¹, Rafael B. Schvitz³, Paulo F. Butzen², Leomar Soares da Rosa Júnior¹

¹Universidade Federal de Pelotas - Centro de Desenvolvimento Tecnológico

²Universidade Federal do Rio Grande do Sul - Escola de Engenharia

³Universidade Federal do Rio Grande - Centro de Ciências Computacionais

{lqjurgina,gbmanske,leomarjr}@inf.ufpel.edu.br,pnt.matheus@gmail.com,
{claytonfarias,rafaelschvitz}@furg.br, paulo.butzen@ufrgs.br

Abstract. *This work presents CREsT, a tool developed to support the teaching of reliability concepts in digital circuits. Reliability is currently investigated in several state-of-the-art works in both academics and industry. However, its complexity becomes a barrier for its study in the classroom. In this context, the CREsT was built. The software was evaluated with a group of 33 students of the Design of Integrated Circuits course. The results demonstrate a positive response, indicating the success of the tool when applied to the classroom.*

Resumo. *Este trabalho descreve a ferramenta CREsT, desenvolvida para dar suporte ao ensino de confiabilidade em circuitos digitais. A confiabilidade deve ser levada em consideração na arquitetura de circuitos e está presente nas discussões mais recentes da área. Entretanto, sua complexidade é uma barreira para que seu estudo seja incorporado em sala de aula. Neste contexto, o CREsT foi construído. O software foi avaliado em sala de aula, com um grupo de 33 alunos da disciplina de Concepção de Circuitos Integrados e teve resposta positiva e bem recebida pela comunidade, indicando o sucesso no objetivo da ferramenta.*

1. Introdução

O progresso atingido pela indústria de semicondutores possibilitou a grande disseminação dos componentes integrados e o advento da Indústria 4.0 e da Internet das Coisas [Sethi 2017]. Neste contexto, a informação tem papel fundamental para alimentar os novos sistemas com suas inúmeras funcionalidades nas mais diversas aplicações [Wollschlaeger 2017]. Desta forma, sensores capazes de captar estes dados estão distribuídos nos mais diversos locais. Além disso, atuadores são os responsáveis pela reação do sistema baseado nos dados coletados. Em grande parte das aplicações, estes atuadores são autônomos, ou seja, não estão diretamente ligados à supervisão de uma pessoa [Kyriazis et. al. 2013].

Existem diversos desafios relacionados com os novos papéis que as aplicações eletrônicas estão assumindo na sociedade. O principal deles talvez esteja associado com a confiabilidade da operação do circuito. Fatores que podem influenciar a confiabilidade de um sistema vão desde a variabilidade existente no processo de fabricação dos circuitos integrados, passando por interferências na operação como as geradas por radiação até os efeitos de envelhecimento que acometem os dispositivos semicondutores [Rahimi et. al.

2016]. Devido ao fato de diversas aplicações tomarem decisões de forma autônoma, é muito importante que o circuito seja altamente confiável, minimizando a chance de que uma falha no sistema gere um erro na saída e conseqüentemente um comportamento inesperado [Kyriazis et. al. 2013]. Soluções que buscam o aumento da confiabilidade dos sistemas são encontradas na literatura [Rahimi et. al., 2016]. Contudo, métodos para avaliar/estimar a confiabilidade dos circuitos produzidos ainda carecem de maior investigação, e a inclusão deste estudo nos cursos de eletrônica se torna uma necessidade para uma formação mais completa e em dia com os tópicos atuais da indústria.

Atualmente, no ensino de circuitos combinacionais, são apresentados os conceitos que permitem a sua construção: As portas lógicas e suas funções. Essas podem ser representadas como uma equação booleana, tabela verdade ou diagrama lógico [Klock, Ribas e Reis 2010]. A minimização dos circuitos pode ser realizada a partir de técnicas como a álgebra booleana e o mapa de Karnaugh [Karnaugh 1953], para isso são estudados alguns parâmetros como o atraso dos circuitos implementados, o consumo energético que eles apresentam para realizar as operações e o custo da matéria prima para a construção do circuito integrado. Para que essas métricas sejam estudadas e compreendidas, ferramentas de descrição de circuitos digitais [Intel 2021] e simuladores elétricos [Cadence 2021] são empregados em sala de aula, permitindo que o aluno relacione a teoria com a prática através do uso de linguagens Verilog e Spice, respectivamente.

Ferramentas que ofereçam suporte ao processo de ensino-aprendizagem são importantes aliadas para o sucesso da construção do conhecimento. A atividade prática e o exercício, da teoria recém recebida, potencializam o êxito das discussões da sala de aula. Em diversas áreas é possível encontrar softwares que cumprem este papel. [Monteiro et. al. 2018] apresenta uma ferramenta que auxilia no ensino da modelagem de sistemas distribuídos críticos, cujos além de complexa aprendizagem, ainda exigem garantias de segurança verificadas e validadas. A ferramenta Moss [Costa et. al. 2018] é integrada ao MIPS Assembler and Runtime Simulator (MARS) e dá suporte para a simulação das principais funções no estudo de sistemas operacionais.

Neste contexto, considerando a importância do estudo sobre Confiabilidade no ensino de circuitos digitais e atentos a sua complexidade de compreensão, foi desenvolvida uma ferramenta para viabilizar e auxiliar a discussão deste tópico em sala de aula. A partir da revisão de outros trabalhos, que foram desenvolvidos para o suporte de ensino-aprendizagem de circuitos combinacionais, o CREsT foi construído para suprir esta lacuna.

2. Trabalhos Relacionados

Apesar de existirem diversas ferramentas que exploram o conceito da confiabilidade no projeto de circuitos, nenhuma delas é proposta para o processo de ensino-aprendizagem do tema. Existem ferramentas construídas para o ensino de circuitos digitais, levando em consideração outros parâmetros, e existem ferramentas que tratam a confiabilidade.

Os autores [Jurgina et. al. 2020] apresentam uma ferramenta digital para auxílio do processo de ensino-aprendizagem de Circuitos Digitais. O LogicFlow possui recursos para o exercício de circuitos sequenciais e combinacionais, trazendo uma área de desenho para o usuário construir seu projeto e realizar simulações. A validação do software foi realizada com estudantes dos períodos iniciais dos cursos de computação e atendeu aos critérios pedagógicos da avaliação.

O Karma [Klock, Ribas e Reis 2010] oferta recursos de construção e otimização de circuitos. O Logic Simulator Pro [Kajory 2019] permite que sejam realizadas construções e simulações dos circuitos. Já o DiCiDe: Digital Circuit Design [Humanes 2019] e o Logic Gates [Dudie 2019] permitem simulações pré-implementadas vindas de uma tabela verdade.

Aguiar et. al. apresenta uma ferramenta para avaliar a confiabilidade das células lógicas, a fim de fornecer um conjunto de informações para melhorar a robustez do projeto de circuitos. A ferramenta é capaz de avaliar células lógicas sob falhas de Evento Transiente Único (do inglês, Single Event Transient (SET)) e, também, falhas permanentes tais como (do inglês, Stuck-On (SO_NF)) e (do inglês, Stuck-Open (SO_F)). As informações produzidas por esta ferramenta ajudam os designers a escolher as células mais confiáveis a serem adotadas em seus projetos [Aguiar et. al. 2016].

Martín-Lloret et. al. apresentam a ferramenta CASE, que pode avaliar o impacto dos efeitos da confiabilidade em circuitos analógicos através de um modelo baseado em física estocástica. O fluxo de simulação implementado é preciso e eficiente em termos de tempo computacional. Essa ferramenta permite considerar simultaneamente a influência da variabilidade e do envelhecimento na confiabilidade dos circuitos analógicos [Martín-Lloret et. al. 2017]

A ferramenta RAIN explora os principais processos de envelhecimento nas interconexões dos circuitos propondo modelos abrangentes e compactos para estados transitórios e estáveis baseados na evolução do estresse hidrostático. A ferramenta inclui algumas observações experimentais e introduz parâmetros dependentes da temperatura. Com isso, a ferramenta desenvolvida é capaz de avaliar a confiabilidade das interconexões que realizam toda a comunicação entre os circuitos [Abbasinasab e Marek-Sadowska 2018].

3. CREsT

O incremento na complexidade dos circuitos integrados, diretamente relacionado com escalas nanométricas, exige que seja possível analisar, antes de sua fabricação, o seu comportamento na presença de possíveis falhas [Hasan et. al. 2011]. A fim de guiar o projeto desses componentes, é necessário que sejam desenvolvidas ferramentas que possam avaliar, com eficiência, a confiabilidade dos mesmos. Contudo, o processo de avaliar a confiabilidade de um circuito integrado, envolve, ou metodologias exaustivas ou métodos com complexidade exponencial ao tamanho do circuito a ser analisado [Xiao e Chen 2014]. Isso quer dizer que, dependendo do tamanho do circuito a ser analisado, o poder computacional exigido pode inviabilizar o processo de análise.

Quando se trata de confiabilidade em sistemas, no geral, o termo se relaciona com a probabilidade de um correto funcionamento em um determinado intervalo de tempo [Lala 2001]. A métrica utilizada para a confiabilidade em sistemas eletrônicos é a taxa de falhas, a qual indica o número de falhas presentes em um período (geralmente o período é uma (1) hora) [Finkelstein 2008]. Quando a taxa de falhas é constante, é possível convertê-la em Mean Time Between Failure (MTBF) que é o tempo médio entre falhas. É preferível utilizar o MTBF como métrica devido a interpretabilidade dos valores, ou seja, é mais intuitivo utilizar 5000 horas entre falhas do que 0.0002 falha por hora.

As Matrizes de Transferência Probabilísticas (PTM) são uma estrutura de dados que pode representar o comportamento desde uma porta lógica até todo um circuito [Patel

et. al. 2003]. Nesta matriz são mapeadas as probabilidades de uma entrada gerar determinada saída. As linhas de uma matriz PTM representam o vetor de entrada, já as colunas representam a saída do bloco lógico. Se for considerado uma porta AND2 como exemplo, a matriz PTM que a representa possui 4 linhas e 2 colunas. Sendo assim, um bloco lógico qualquer que possua m entradas e n saídas irá gerar uma matriz $2^m \times 2^n$. Quando uma matriz PTM possui apenas elementos 0 e 1, passa a ser denominada matriz de transferência ideal (ITM). Na Figura 1 é possível observar a relação das matrizes PTM e ITM com a tabela verdade de uma porta lógica AND2.

Input	Input	Output
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

$$\begin{matrix} & \mathbf{0} & \mathbf{1} \\ \mathbf{00} & \begin{bmatrix} 1 & 0 \end{bmatrix} \\ \mathbf{01} & \begin{bmatrix} 1 & 0 \end{bmatrix} \\ \mathbf{10} & \begin{bmatrix} 1 & 0 \end{bmatrix} \\ \mathbf{11} & \begin{bmatrix} 0 & 1 \end{bmatrix} \end{matrix}$$

$$\begin{matrix} & \mathbf{0} & \mathbf{1} \\ \mathbf{00} & \begin{bmatrix} q & 1-q \end{bmatrix} \\ \mathbf{01} & \begin{bmatrix} q & 1-q \end{bmatrix} \\ \mathbf{10} & \begin{bmatrix} q & 1-q \end{bmatrix} \\ \mathbf{11} & \begin{bmatrix} 1-q & q \end{bmatrix} \end{matrix}$$

Figura 1. (a) Tabela verdade de uma AND2, (b) Matriz ITM, (c) Matriz PTM.

Vários métodos para se estimar a confiabilidade de um circuito foram propostos na literatura [Xiao e Chen 2014]. O framework Probabilistic Transfer Matrix Method (PTMM), proposto por Patel [Patel et. al. 2003], pode produzir um valor exato de confiabilidade de um circuito [Pontes et. al. 2018]. No método PTMM, a confiabilidade de um circuito é obtida por meio de operações de matrizes PTM. As duas operações utilizadas são o tensor de Kronecker e a multiplicação de matrizes. Por isso, a complexidade deste método é exponencial ao número de portas lógicas do circuito, o que limita a aplicação a apenas circuitos pequenos. Outro importante método presente na literatura é o Signal Probability Reliability (SPR). Neste método a confiabilidade de um circuito é estimada através das probabilidades dos sinais de entrada e das portas lógicas [Franco et. al. 2008]. Assim como o PTMM, as matrizes PTM e ITM são usadas para mapear o comportamento das portas lógicas. O diferencial do método SPR é a sua complexidade, a qual é linear ao número de portas lógicas. Isso torna o método escalável, ou seja, é possível analisar a confiabilidade em circuitos com milhares de portas.

Ambos os métodos descritos utilizam operações com matrizes. Mesmo para circuitos com poucas portas lógicas, a quantidade de cálculos envolvidos exige a utilização de ferramentas, tanto para auxiliar no ensino do tema, quanto para aplicações na indústria.

3.1 Construção da Ferramenta

A ferramenta CREsT possui o objetivo de estimar a confiabilidade de circuitos. A Figura 2 demonstra o fluxo de operação completo da ferramenta e os parâmetros de entrada. Na primeira etapa, o terminal recebe os termos digitados pelo usuário e os envia ao processador de comandos. O comando informado é avaliado, em caso de identificação a ferramenta executa o comando, caso contrário é retornado o erro ao usuário.

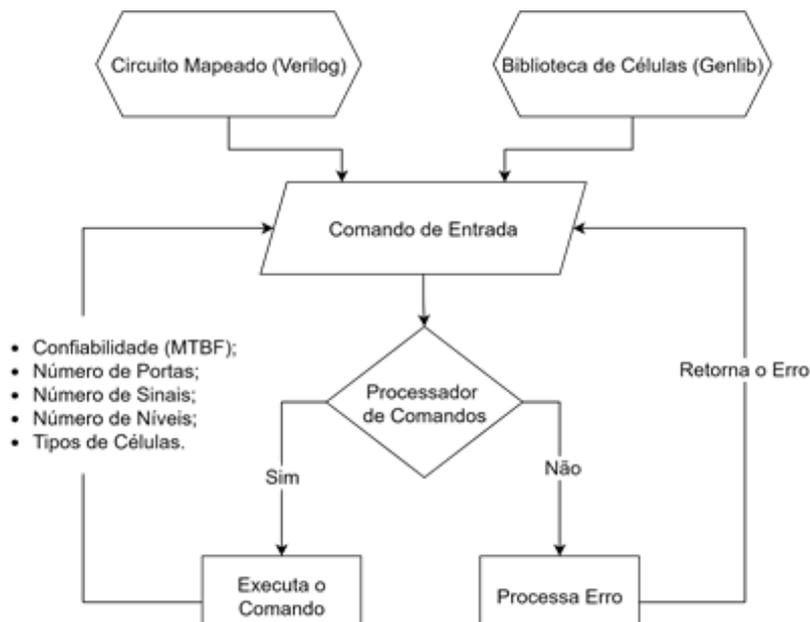
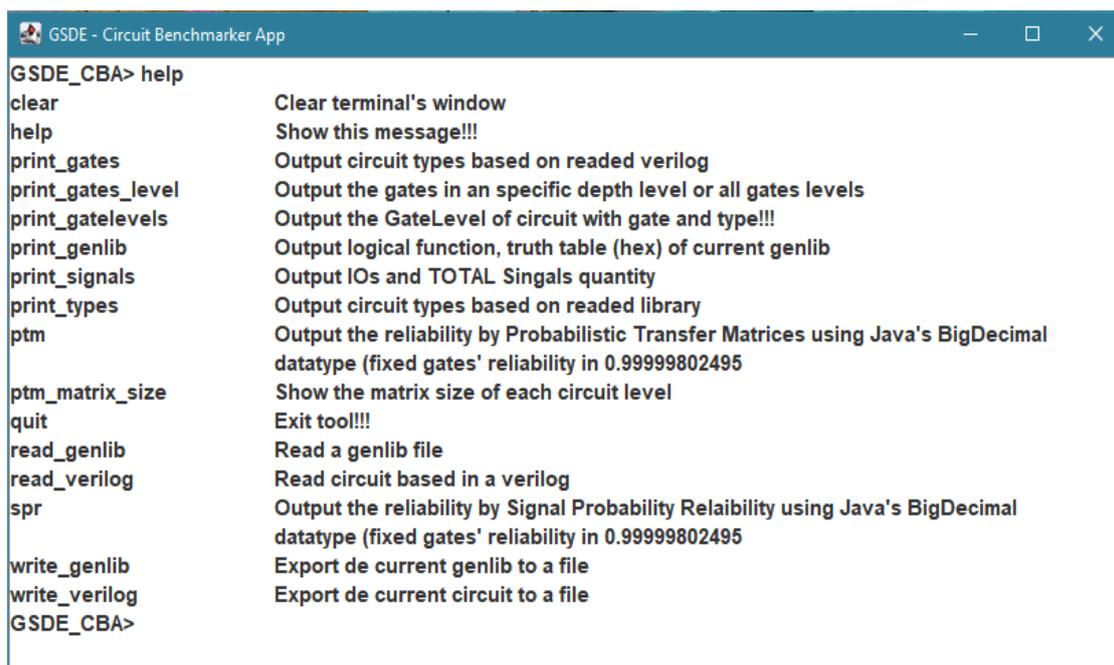


Figura 2. Fluxograma de operação da ferramenta.

A ferramenta foi desenvolvida na linguagem JAVA, pois esta possui maior suporte para a execução na maior parte dos sistemas operacionais disponíveis atualmente. Para exemplificar todo o funcionamento da ferramenta, usaremos o caso de teste do circuito c17v1, apresentado na Figura 3(a), e a biblioteca de células, Figura 3(b). Estas informações representam as dependências da ferramenta. Vale salientar que o funcionamento do CREsT ocorre através de linhas de comandos estilo o prompt. A Figura 4 apresenta a execução do comando de ajuda ("help"), responsável por listar todas as funcionalidades disponíveis na ferramenta.

<pre> module C17V1 (G1gat, G2gat, G3gat, G4gat, G5gat, G6gat, G7gat); input G1gat, G2gat, G3gat, G4gat, G5gat; output G6gat, G7gat ; wire w1, w2, w3, w4, w5, w6, w7; NAND2 U0(.A(G5gat), .B(G1gat), .Y(w1)); NAND2 U1(.A(G5gat), .B(G2gat), .Y(w2)); INV U4(.A(G2gat), .Y(w3)); NAND2 U2(.A(w2), .B(G4gat), .Y(w4)); INV U5(.A(G5gat), .Y(w5)); NOR2 U7(.A(G4gat), .B(G3gat), .Y(w6)); NOR2 U6(.A(w5), .B(w3), .Y(w7)); NAND2 U3(.A(w4), .B(w1), .Y(G6gat)); NOR2 U8(.A(w6), .B(w7), .Y(G7gat)); endmodule </pre> <p style="text-align: center;">(a)</p>	<pre> GATE BUF 1 Y=A; GATE INV 1 Y=!A; GATE AND2 1 Y=A*B; GATE OR2 1 Y=A+B; GATE NOR2 1 Y=! (A+B) ; GATE NOR3 1 Y=! (A+B+C) ; GATE NAND2 1 Y=! (A*B) ; GATE NAND3 1 Y=! (A*B*C) ; GATE OAI21 1 Y=! ((A0+A1)*B0) ; GATE AOI21 1 Y=! (A0*A1+B0) ; GATE XOR2 1 Y=(A*!B) + (!A*B) ; </pre> <p style="text-align: center;">(b)</p>
---	--

Figura 3. (a) Circuito c17v1, (b) Biblioteca de células.



```

GSDE_CBA> help
clear          Clear terminal's window
help          Show this message!!!
print_gates   Output circuit types based on readed verilog
print_gates_level Output the gates in an specific depth level or all gates levels
print_gatelevels Output the GateLevel of circuit with gate and type!!!
print_genlib  Output logical function, truth table (hex) of current genlib
print_signals Output IOs and TOTAL Singals quantity
print_types  Output circuit types based on readed library
ptm          Output the reliability by Probabilistic Transfer Matrices using Java's BigDecimal
            datatype (fixed gates' reliability in 0.99999802495
ptm_matrix_size Show the matrix size of each circuit level
quit         Exit tool!!!
read_genlib  Read a genlib file
read_verilog Read circuit based in a verilog
spr         Output the reliability by Signal Probability Reliability using Java's BigDecimal
            datatype (fixed gates' reliability in 0.99999802495
write_genlib Export de current genlib to a file
write_verilog Export de current circuit to a file
GSDE_CBA>

```

Figura 4. Apresentação dos comandos disponíveis na ferramenta.

Ao utilizar os comandos "read_genlib" e "read_verilog", os parâmetros de entrada são adicionados ao fluxo de execução da ferramenta, onde é obtida a mensagem de "successfully processed", que representa a sucesso na execução do comando. Ambos comandos são demonstrados pela Figura 5.

Após o reconhecimento das entradas, é possível visualizar informações adicionais delas, como as características topológicas, tipos, quantidades, níveis de portas lógicas, sinais e outras informações referentes ao circuito em análise. Para exemplificar, o comando "print_gates" apresenta a topologia do circuito c17v1. Por fim, também está disponível a funcionalidade de exportar para novos arquivos, a biblioteca e o circuito.

Para estimar a confiabilidade de circuitos combinacionais, existem dois comandos disponíveis "ptm" e "spr". O primeiro, "ptm" realiza a estimativa de confiabilidade a partir do método PTMM. E o segundo, através da execução do método SPR. Ambos resultados são expressos pela métrica MTBF, que representa o número de horas de operação livres de falha de um circuito. A execução dos comandos é representada pela Figura 5, que exemplifica o fluxo de comandos utilizados para estimar a confiabilidade do circuito c17v1, da leitura das entradas à estimativa de confiabilidade, para ambas metodologias apresentadas anteriormente.

Cabe destacar que a interface da ferramenta foi desenvolvida em "linha de comando/modo texto", e tal modo não implica em prejuízo ao usuário alvo desta ferramenta, pois este já possui familiaridade com este tipo de execução e, além disso, possui contato com outras ferramentas que são direcionadas para a performance e para a geração de resultado, pois os estudantes alvo estão no final do curso. Um exemplo de ferramenta conhecida e utilizada pelos estudantes, a qual utiliza o mesmo tipo de interface, é a ferramenta ABC [ABC 2021], amplamente empregada no projeto de síntese lógica de circuitos digitais.

```

GSDE - Circuit Benchmarking App
GSDE_CBA> read_genlib full.genlib
File "full.genlib" successfully processed!!
GSDE_CBA> read_verilog c17v1.v
File "c17v1.v" successfully processed!!
GSDE_CBA> print_gates
NAND2 : 4
INV : 2
NOR2 : 3
TOTAL: 9
GSDE_CBA> ptm
MTBF using PTM of c17v1 CIRCUIT is 84386
GSDE_CBA> spr
MTBF using SPR of c17v1 CIRCUIT is 84386
GSDE_CBA>

```

Figura 5. Análise do circuito c17v1.

Para a construção do CREsT, foram verificadas o atendimento às Taxonomias para Avaliação de Recursos Digitais de Aprendizagem (TARDA), propostas por [Junior et. al. 2020], apresentada na Figura 6.

Concepção do Software	Evolução do Software	Segurança do Sistema
<p>Adequação: Capacidade de prover um conjunto apropriado de funções para tarefas e objetivos dos usuários especificados.</p> <p>Acurácia: Capacidade de prover, com o grau de precisão necessário, resultados ou efeitos corretos ou conforme acordados.</p> <p>Conformidade Funcional: Capacidade de estar de acordo com normas, convenções ou regulamentações previstas em leis e prescrições similares relacionadas à funcionalidade.</p> <p>Recursos: Capacidade de usar tipos e quantidades apropriados de recursos, quando o software executa suas funções sob condições estabelecidas.</p> <p>Conformidade Portátil: Capacidade de estar de acordo com normas ou convenções relacionadas à portabilidade.</p> <p>Interoperabilidade: Capacidade de interagir com um ou mais sistemas especificados.</p>	<p>Estabilidade: Capacidade de evitar efeitos inesperados decorrentes de modificações no software.</p> <p>Analisabilidade: Capacidade de permitir o diagnóstico de deficiências ou causas de falhas no software, ou a identificação de partes a serem modificadas.</p> <p>Modificabilidade: Capacidade de permitir que uma modificação especificada seja implementada</p> <p>Substituibilidade: Capacidade de ser usado em substituição a outro produto de software especificado, com o mesmo propósito e no mesmo ambiente.</p> <p>Instalabilidade: Capacidade para ser instalado em um ambiente especificado.</p> <p>Testabilidade: Capacidade de permitir que o software, quando modificado, seja validado.</p> <p>Maturidade: Capacidade de evitar falhas decorrentes de defeitos no software.</p>	<p>Inteligibilidade: Capacidade de possibilitar ao usuário compreender se o software é apropriado e como ele pode ser usado para tarefas e condições de uso específicas.</p> <p>Apreensibilidade: Capacidade do produto de software possibilitar ao usuário entender como ele funciona e/ou aprender a usá-lo.</p> <p>Tolerância à Falhas: Capacidade de evitar falhas e manter um nível de desempenho especificado em casos de defeitos no software ou de violação de sua interface especificada.</p> <p>Segurança de Acesso: Capacidade de proteger informações e dados, de forma que pessoas ou sistemas não autorizados não possam lê-los nem modificá-los e que não seja negado o acesso às pessoas ou sistemas autorizados.</p>

Figura 6. TARDA, proposta por [Junior et. al. 2020].

Todas as taxonomias apresentadas foram atendidas. A única limitação que se pode observar quanto a Interoperabilidade é referente ao software não operar em sistemas operacionais para dispositivos móveis, como Android e IOS, por exemplo.

4. Experiência de Utilização do CREsT

Para a validação do CREsT, como ferramenta pedagógica para o estudo de confiabilidade, foi realizado um experimento com alunos do sétimo semestre do curso de Engenharia de Computação.

4.1 Apresentação da Ferramenta

Antes da experiência com a ferramenta os alunos foram introduzidos ao conceito de confiabilidade. Para isso uma aula com o tema “Confiabilidade em circuitos digitais” foi preparada. Esta contextualização do tema teve duração de 25 minutos e foi distribuída da seguinte maneira: introdução sobre confiabilidade (10 minutos), simulação elétrica (5

minutos), técnicas de estimação (5 minutos) e discussão sobre dificuldades das técnicas de estimação (5 minutos).

Após a apresentação do tema, os alunos foram apresentados ao CREsT. A apresentação durou 15 minutos e contou com a exposição da interface, técnicas implementadas, principais comandos e uma demonstração de simulação de um circuito exemplo. A seguir foram disponibilizados 5 minutos de interação livre dos usuários com a ferramenta, para que a explorassem, testassem e descobrissem suas funcionalidades.

No total, 33 alunos participaram desta avaliação. Todos estavam concluindo a disciplina de Concepção de Circuitos Integrados, que integra o sétimo período do curso de Engenharia de Computação. Portanto, estavam familiarizados com os conceitos de circuitos digitais combinacionais e seus parâmetros, aptos a serem apresentados ao conceito de confiabilidade.

Posteriormente a interação livre com o software, foram propostas duas atividades com a ferramenta. As atividades incluíam dois circuitos diferentes, cada um com três versões - construídos de maneiras diferentes, todavia com a mesma equivalência lógica - totalizando 6 circuitos diferentes. Essa proposta possuía como objetivo evidenciar o impacto da diferença da arquitetura de cada circuito na sua confiabilidade, ratificando a importância de levar esse parâmetro em consideração. Os usuários tiveram 15 minutos para realizar as simulações e depois responderam ao formulário de validação.

4.2 Resultados da Experiência

Para validação da ferramenta, após a experiência, os usuários responderam a um formulário elaborado para avaliar o impacto do software, se suas funcionalidades atendiam às necessidades dos educandos, como apresentado pelos autores [Jurgina et. al. 2020]. Segundo [Cybis 1997 apud Jurgina et. al. 2020] quando se analisa funcionalidades é necessária a participação dos usuários. A relação de funcionalidades e as perguntas realizadas são apresentadas na Tabela 1.

O objetivo do formulário consistia em perceber o impacto da ferramenta a partir da verificação da percepção dos usuários juntamente às percepções do grupo desenvolvedor da ferramenta, para elaborar as melhorias para a ferramenta a partir das evidências coletadas durante esta experiência.

Tabela 1. Relação dos critérios e perguntas empregadas na avaliação do CREsT

Critério	Pergunta
Facilidade de Aprendizagem	“O software possui usabilidade simples?”
Eficiência	“Foi possível realizar as simulações com eficiência?”
Prevenção de Erros	“O software oferece auxílio adequado para prevenir erros?”
Satisfação	“O software oferece o auxílio adequado como você esperava?”
Adequação Pedagógica	“O software auxiliou você a compreender mais sobre confiabilidade em circuitos combinacionais?”
Utilidade	“O software é útil para estudos de confiabilidade em sala de aula?”

Este formulário foi construído a partir de um trabalho da área em que foi verificado o seu sucesso, pois a avaliação precisava de profundidade para validar a ferramenta. Para responder as questões os alunos possuíam como opções “Sim”, “Parcialmente” e “Não”, e as respostas estão na Figura 7. Os alunos também foram questionados sobre o conhecimento acerca do tema de confiabilidade. As respostas estão ilustradas na Figura 8.

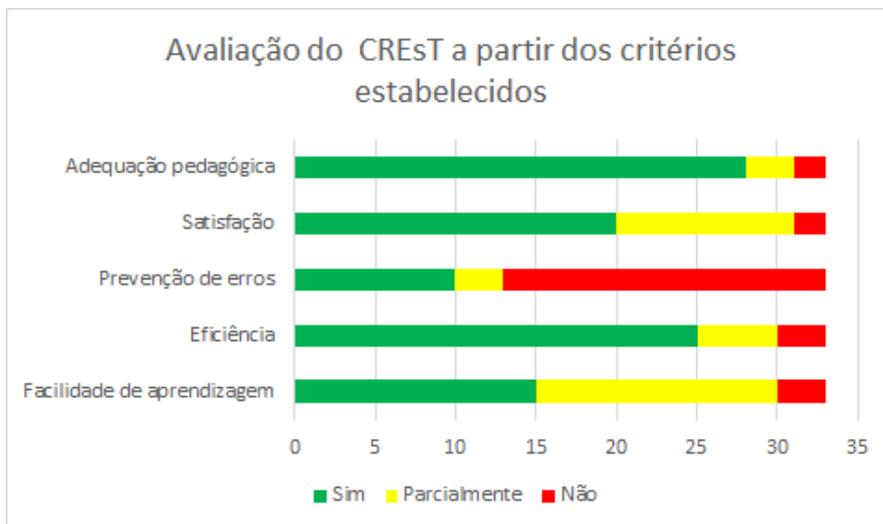


Figura 7. Avaliação da Ferramenta segundo usuários.

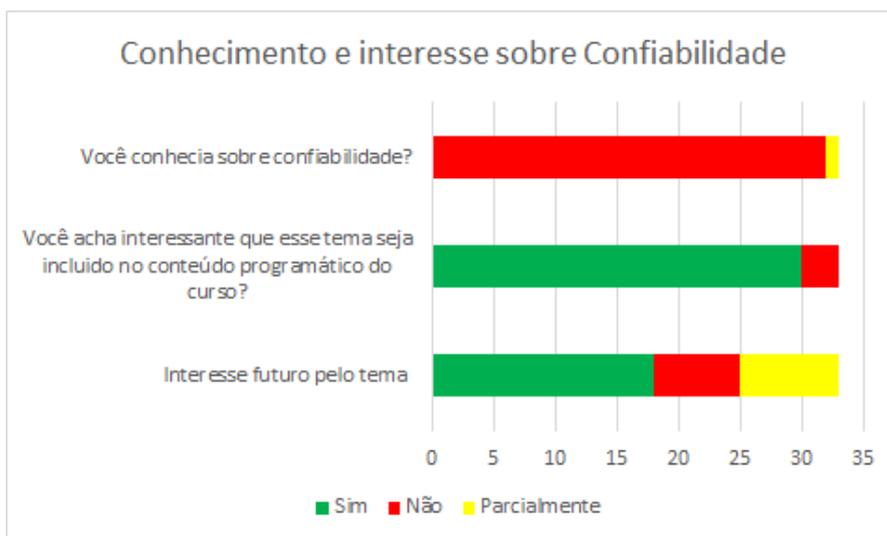


Figura 8. Conhecimento e interesse sobre Confiabilidade.

5. Conclusões

O formulário apontou uma carência na prevenção de erros, isso ocorre por falhas de sintaxe, por parte do usuário, na descrição do circuito que faz parte da entrada do software. Soluções preventivas que detectem esses erros já estão sendo estudadas. Porém, as descrições podem ser compiladas nos ambientes próprios de descrição de circuito, o que também viabiliza, como trabalho futuro, a importação de arquivos gerados por estes softwares para o cálculo e estimação da confiabilidade.

A prevenção de erros teve impacto no critério de satisfação que também considera o auxílio ofertado pela ferramenta. O CREsT foi considerado eficiente e apenas 3 alunos não o consideraram com fácil aprendizagem. 28 dos participantes consideraram o software adequado pedagogicamente, o que vai ao encontro dos resultados observados pelos aplicadores, pois todos os alunos conseguiram realizar as simulações propostas, obtendo o resultado esperado pelo CREsT.

O processo de ensino-aprendizagem tem envolvido cada vez mais ferramentas das tecnologias da informação e comunicação. Elas são propostas, neste contexto, como uma forma de superar alguns dos desafios inerentes a tal processo [Correia et. al 2018]. O contexto em que a ferramenta foi apresentada, uma lacuna entre a relação da educação com os pontos altos da pesquisa e a indústria, evidencia uma carência do atual ensino das universidades, que faz necessária a atenção dos pesquisadores.

Como ilustrado na Figura 8, é possível verificar que nenhum aluno participante da pesquisa possuía conhecimento sólido sobre o tema. O que era esperado pelos autores, porém, não de maneira satisfatória. Este trabalho mostra que existe uma lacuna entre o estado da arte discutido em congressos e indústria e o conteúdo discutido em sala de aula, o que gera um prejuízo na formação profissional deste estudante.

A Figura 8 também ratifica que existe o interesse dos discentes de se inteirar sobre o assunto, pois 30 dos 33 participantes são favoráveis a inclusão do tema no conteúdo programático do curso, enquanto mais de 50% demonstra intenção de pesquisar sobre o tema futuramente. Esses resultados apontam para o desejo dos estudantes de construir uma formação completa e atual, contextualizada no atual momento apresentado pela indústria e mercado.

Com o avanço da indústria junto a importância da eletrônica e da resposta dos circuitos, muitas vezes autônomos, que possuem seu sucesso diretamente ligado à confiabilidade apresentado pelos seus atuantes. Assim, o profissional formado precisa conhecer sobre o assunto. O estudo da confiabilidade é complexo e seus cálculos envolvem matrizes que crescem exponencialmente, o que dificulta o estudo do tema. O CREsT vem para facilitar essa discussão, pois o software realiza esses cálculos em suas simulações. É alimentado por um circuito descrito em linguagem já presente nas salas de aula, então facilita a interação e alimentação de sua entrada.

O exercício proposto contou com três versões diferentes, mas com a mesma equivalência lógica, para que seja possível ilustrar o impacto da arquitetura e escolhas de projeto na confiabilidade de sua operação. Ensinar confiabilidade é extremamente importante e possuir uma ferramenta que dê suporte é imprescindível. A ferramenta apresentada ainda não possui a interface gráfica, que está presente no planejamento dos trabalhos futuros. Todavia, a sua ausência não possuiu impacto negativo durante a experiência com alunos. Por outro lado, será um atrativo para os usuários.

Com os resultados obtidos e as evidências verificadas, podemos afirmar que a ferramenta constrói uma relação que viabiliza a inserção do tema na sala de aula, afirmando o compromisso da pesquisa de atual estado da arte com os recursos pedagógicos de formação, na sincronização da sala de aula com a indústria.

Agradecimentos

Este trabalho foi realizado com o apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e Fundação de Amparo à pesquisa do Estado do Rio Grande do Sul (FAPERGS).

Referências

- Abbasinasab A. and Marek-Sadowska M., "RAIN: A Tool for Reliability Assessment of Interconnect Networks - Physics to Software," 2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC), 2018, pp. 1-6, doi: 10.1109/DAC.2018.8465800.
- ABC. (2021). ABC: A System for Sequential Synthesis and Verification from Berkeley Logic Synthesis and Verification Group. <http://people.eecs.berkeley.edu/~alanmi/abc/abc.htm>. Julho.
- Aguiar, Y. Q., Zimpeck, A. L., Meinhardt, C., & Reis, R. (2016). Permanent and single event transient faults reliability evaluation EDA tool. *Microelectronics Reliability*, 64, 63-67.
- Cadence. (2021). Cadence PSpice. <https://www.pspice.com/>. Julho.
- Correia, K. S. M., Fook, K. D., & Sá, E de J.V. (2018). LogEasy++: Uma Ferramenta para ensino de aprendizagem personalizável utilizando Gamificação. In: *Anais do XXIX Simpósio Brasileiro de Informática da Educação*, (pp. 575-584). Porto Alegre: SBC.
- Costa, A. V., Silva, A., Fernandes, S., & Macedo, F. T. de. (2018). MOSS - Uma Ferramenta para o Auxílio do Ensino de Sistemas Operacionais. In: *Anais do XXIX Simpósio Brasileiro de Informática da Educação*, (pp. 755-764). Porto Alegre: SBC.
- Dudie. (2019). Logical Gates - Apps para Android no Google Play. https://play.google.com/store/apps/details?id=com.Logic.gate_Xor.en, Junho.
- Finkelstein, M. *Failure rate modelling for reliability and risk*: Springer Science & Business Media, 2008.
- Franco, D. T.; Vasconcelos, M. C.; Naniver, L.; NAVINER, J.-F. Reliability analysis of logic circuits based on signal probability. In: *electronics, circuits and systems*, 2008. ICECS 2008. 15th IEEE International Conference on, 2008. p.670–673.
- Hasan, O.; Patel, J.; Tahar, S. Formal reliability analysis of combinational circuits using theorem proving. *Journal of Applied Logic*, [S.l.], v.9, n.1, p.41–60, 2011.
- Intel. (2021). Quartus II. <https://fpgasoftware.intel.com/13.0sp1/>. Julho.
- Junior, O., Aguiar, Y., & de Moura, H. (2020). Taxonomia para Avaliação de Recursos Digitais de Aprendizagem: Proposição e Análise da Adequação por Especialistas. In: *Anais do XXXI Simpósio Brasileiro de Informática na Educação*, (pp. 232-241). Porto Alegre: SBC.
- Jurgina, L., Zanandrea, R., Rosa Júnior, L., & Marques, F. (2020). LogicFlow: Uma Ferramenta Para o Auxílio de Ensino-aprendizagem de Circuitos Digitais. In: *Anais do XXXI Simpósio Brasileiro de Informática na Educação*, (pp. 322-331). Porto Alegre: SBC.

- Kajory. (2019). Logic Simulator Pro - Apps para Android no Google Play. https://play.google.com/store/apps/details?id=com.KAJORY.Logicimulatorpro&hl=pt_BR. Julho.
- Karnaugh, M. (1953). The map method for synthesis of combinational logic circuits, *Trans. AIEE, Commun. & Electron.*, vol. 72, no. 1, pp 593-598.
- Klock, C. E.; Ribas, R. P.; Reis, A. I. (2010). Karma: um ambiente para o aprendizado de síntese de funções Booleanas. *Revista Brasileira de Informática na Educação*, vol. 18, n. 2, pp. 33-43.
- Kyriazis, Dimosthenis, and Theodora Varvarigou. "Smart, autonomous and reliable Internet of Things." *Procedia Computer Science* 21 (2013): 442-448.
- Lala, P. K. *Self-checking and fault-tolerant digital design*: Morgan Kaufmann, 2001.
- Martín-Lloret P. et al., "CASE: A reliability simulation tool for analog ICs," 2017 14th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD), 2017, pp. 1-4, doi: 10.1109/SMACD.2017.7981588.
- Monteiro, E. D., Rivero, L., & Barreto, R. (2018). Uma Ferramenta de Suporte ao Ensino de Modelagem de Sistemas Distribuídos Críticos: Uma Experiência Prática. In *Anais do XXIX Simpósio Brasileiro de Informática na Educação* (pp. 218-227). Porto Alegre: SBC.
- Patel, K.; Hayes, J.; Markov, I. Evaluating circuit reliability under probabilistic gate-level fault models. *Proceedings of the International Workshop on Logic and Synthesis*, p.59-64, 2003.
- Pontes, M. F.; Butzen, P. F.; Schvitz, R. B.; Rosa, S. L.; Feanco, D. T. The Suitability of the SPR-MP Method to Evaluate the Reliability of Logic Circuits. In: *IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2018. p.433-436.
- Rahimi, Abbas, Luca Benini, and Rajesh K. Gupta. "Variability mitigation in nanometer CMOS integrated systems: A survey of techniques from circuits to software." *Proceedings of the IEEE* 104.7 (2016): 1410-1448.
- Sethi, Pallavi, and Smruti R. Sarangi. "Internet of things: architectures, protocols, and applications." *Journal of Electrical and Computer Engineering* 2017 (2017).
- Wollschlaeger, Martin, Thilo Sauter, and Juergen Jasperneite. "The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0." *IEEE industrial electronics magazine* 11.1 (2017): 17-27.
- Xiao, R.; Chen, C. Gate-level circuit reliability analysis: A survey. *VLSI Design*, v.2014, 2014.