



FPGA como ferramenta de ensino de engenharia



Filipe Samuel Teixeira de Azevedo¹,
 Alex Vitorino²,
 Fernanda Monteiro de Castro Rezende³,
 Fundação Presidente Antônio Carlos

Resumo – As universidades buscam constantemente inserir novas tecnologias nas grades curriculares de graduações em engenharia. Porém, uma das barreiras que impede esta ação é o uso insuficiente de metodologias ativas de ensino. A inclusão destas metodologias pode ser realizada na unidade curricular de Eletrônica Digital, pois esta disciplina possui aulas práticas, o que facilita a aplicação de metodologias ativas. Assim, este trabalho tem como escopo desenvolver um método prático de Eletrônica Digital que utiliza FPGA e linguagem VHDL para direcionar o ensino de engenharia ao aprendizado ativo. Além de comparar este método com o tradicional por meio dos resultados de uma pesquisa de campo realizada com estudantes de Eletrônica Digital da FUPAC. Com este trabalho constatou-se que o método com FPGA e linguagem VHDL pode ser utilizado como ferramenta de ensino de engenharia, por causa de suas facilidades de projeto. Todavia, os discentes questionados demonstraram resistência à implantação deste método.

Palavras Chave – Eletrônica digital, ensino de engenharia, FPGA, metodologia ativa, VHDL.

I. SIGLAS

ABProj – (Aprendizagem Baseada em Projetos)
 ASIC – *Application Specific Integrated Circuits*
 CI – Circuito Integrado
 CLB – *Configurable Logic Block*
 CPLD – *Complex Programmable Logic Device*
 CPU – *Central Processing Unit*
 DoD – *Department of Defense*
 DSP – *Digital Signal Processor*
 EDA – *Electronic Design Automation*
 FPGA – *Field Programmable Gate Array*
 HDL – *Hardware Description Language*
 IEEE – *Institute of Electrical and Electronic Engineers*
 I/O – *Input/Output*
 JTAG – *Joint Test Action Group*
 LAB – *Logic Array Block*
 PLD – *Programmable Logic Device*
 RAM – *Random Access Memory*
 RTL – *Register Transfer Level*
 SPLD – *Simple Programmable Logic Device*

VHDL – *VHSIC Hardware Description Language*
 VHSIC – *Very High Speed Integrated Circuit*
 VPI – *Verilog Procedural Interface*

II. INTRODUÇÃO

As instituições de ensino brasileiras buscam constantemente adequar as grades dos cursos de graduação de engenharia às tendências tecnológicas. Isto está relacionado à demanda de profissionais com perfil inovador que fora desencadeada pela globalização [1].

Todavia, realizar esta tarefa não é fácil, pois ainda existem problemas que impedem a evolução do ensino, como a ausência ou uso insuficiente de metodologias ativas nas universidades [2].

As metodologias ativas de ensino são mais simples de serem aplicadas em aulas práticas [2]. Assim, a Eletrônica Digital é uma unidade curricular com potencial para aplicação destas metodologias, pois esta disciplina utiliza atividades práticas para complementar o conteúdo apresentado em aulas teóricas [3].

Atividades práticas de Eletrônica Digital geralmente são realizadas por meio da construção de circuitos físicos [3]. Entretanto, circuitos digitais podem ser confeccionados facilmente utilizando linguagens de descrição de *hardware* (HDL), como a linguagem de descrição de *hardware* de circuito integrado de alta velocidade (VHDL), visto que estas permitem descrever circuitos em alto nível de abstração [4]. Além do mais, circuitos digitais descritos com estas linguagens podem ser implementados e executados por dispositivos lógicos programáveis, como o arranjo de portas programáveis em campo (FPGA) [5].

Portanto, neste trabalho foi desenvolvido um método baseado em descrição de *hardware* que utiliza VHDL e FPGA como nova abordagem de ensino direcionada ao aprendizado ativo, além de serem apresentadas suas diretrizes. Também, este método foi aplicado em um projeto para exemplificar e demonstrar sua utilização.

Ademais, foi realizada uma pesquisa de campo com alunos de Engenharia Elétrica que estão cursando a unidade curricular de Eletrônica Digital, a fim de comparar este método proposto neste trabalho com o método tradicional, fundamentado na

¹Filipe Samuel Teixeira de Azevedo, graduando em Engenharia de Controle e Automação, filipe.s.t.azevedo@gmail.com, ²Alex Vitorino, orientador, mestre em Engenharia Elétrica, graduado em Sistemas de Informação, allexvitorino@gmail.com, ³Fernanda Monteiro de Castro Rezende, coorientadora, mestre em Educação Matemática, graduada em Matemática, fernandamrezende3@yahoo.com.br, Fundação Presidente Antônio Carlos – FUPAC, Rod. MG482,s/n km 3, Gigante, Conselheiro Lafaiete, MG, CEP: 36400-000, Brasil.

montagem de circuitos em *protoboards* com circuitos integrados (CI). E, os resultados desta pesquisa foram analisados com base no método estatístico de análise exploratória de dados.

Este artigo está organizado em quatro seções. A primeira seção apresenta uma revisão de conteúdo bibliográfico referente ao cenário do ensino de engenharia brasileiro, ensino de Eletrônica Digital, sistemas embarcados, linguagens de descrição de *hardware* e métodos estatísticos para análise de dados. A segunda seção revela o desenvolvimento do método prático proposto, suas diretrizes, um projeto como exemplo e a pesquisa de campo. A terceira seção apresenta uma reflexão não exaustiva acerca dos resultados da pesquisa e na quarta seção são apresentadas as conclusões deste trabalho, bem como sugestão de trabalhos futuros.

III. OBJETIVOS

O intuito deste trabalho é desenvolver e apresentar um método de ensino prático, baseado em descrição de *hardware* por meio da linguagem VHDL e dispositivo lógico programável FPGA, suas diretrizes e um exemplo de projeto elaborado com esta abordagem de ensino.

Contudo, pretende-se ainda realizar a comparação entre o método proposto neste trabalho com o método tradicional, utilizado em aulas práticas de eletrônica digital, por meio da análise exploratória dos resultados da pesquisa de campo realizada com estudantes da unidade curricular de Eletrônica Digital da Fundação Presidente Antônio Carlos (FUPAC).

IV. REVISÃO DE BIBLIOGRAFIA

A. Cenário do Ensino de Engenharia Brasileiro

O propósito das universidades de engenharia é formar profissionais com direcionamento à inovação, empreendedorismo e sustentabilidade [6]. Estas são qualidades exigidas pelo mercado de trabalho, por causa da grande competitividade gerada pela globalização [1].

Um dos grandes desafios das instituições de ensino é adequar os currículos de seus cursos de graduação em engenharia às demandas das empresas. Não adaptar o ensino de engenharia a este contexto significa atrasar o Brasil quanto ao desenvolvimento científico e tecnológico. Consequentemente, isto afasta o país do mercado de produtos inovadores de alta tecnologia [1].

Há algumas décadas, a atualização das grades curriculares dos cursos de engenharia está sendo discutida no país, pois alguns problemas relacionados permanecem sem solução, como a qualificação docente [1]. Apenas aperfeiçoar as grades curriculares das graduações em engenharia sem a devida capacitação dos professores pode não produzir os resultados desejados [2].

Entre as metodologias utilizadas nas universidades incluem-se os métodos de ensino tradicionais como, por exemplo, as aulas expositivas. Neste o aluno não participa de forma ativa das aulas, pois sua atitude torna-se passiva, sem participação direta de sua inteligência [2].

É nesse contexto que as metodologias ativas podem ser úteis na formação de engenheiros, pois nesta modalidade de ensino o professor exerce a função de orientador e/ou supervisor, além de ser uma das fontes de conhecimento disponíveis. Isto leva o estudante a participar mais do aprendizado, o que é mais comum de ser observado em aulas práticas, por exemplo. Ou seja, o aprendizado ativo fomenta um uso mais intenso de inteligência por parte do discente por meio da resolução de problemas, desenvolvimento de projetos, discussões, pesquisas, entre outros [2].

B. Ensino de Eletrônica Digital

A eletrônica digital é a área da eletrônica que estuda os circuitos digitais, os quais obedecem a álgebra booleana e possuem como elemento fundamental as portas lógicas. Ademais, é componente essencial no currículo dos cursos de graduação em engenharia de diversas áreas como elétrica, eletrônica, telecomunicações, computação, entre outras [3].

Esta disciplina é dividida em aulas teóricas e aulas práticas. Os conceitos de eletrônica digital são ensinados nas aulas teóricas e, depois, complementados na prática. As aulas práticas são realizadas por meio de atividades como montagens físicas de circuitos e/ou simulações em recursos computacionais [3].

Os temas fundamentais da eletrônica digital são: álgebra booleana, portas lógicas, tabela verdade, mapa de Karnaugh, circuitos combinacionais e circuitos sequenciais [3].

1.. Conceitos Básicos de Circuitos Digitais: Álgebra Booleana, Portas Lógicas e Tabela Verdade

Em 1850, George Boole propôs um conjunto de regras matemáticas baseado em um sistema de valores binários (representado por zeros e uns) [7]. Este conjunto de regras, conhecido como álgebra booleana, é composto por variáveis que formam expressões matemáticas para descrever e sintetizar circuitos digitais [8].

Além das expressões booleanas serem constituídas por variáveis binárias, também possuem operadores lógicos, denominados portas lógicas [7]. Estas portas realizam cálculos em que cada sinal de saída ou entrada tem o nível lógico zero ou um [3]. Entre todas as portas lógicas, existem três que são fundamentais em funções booleanas: adição lógica (ou operação *or*), multiplicação lógica (ou operação *and*) e inversão lógica (ou operação *not*). A fig. 1 apresenta as três portas lógicas fundamentais e suas expressões características [7].

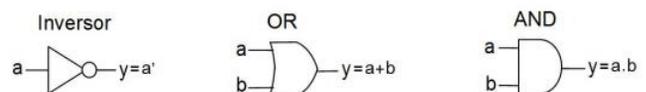


Fig. 1. As três operações fundamentais empregadas em funções booleanas: inversão, *or* e *and* [7].

Outra forma de descrever a relação entre entrada e saída de um sistema digital é a tabela verdade. Esta técnica consiste em realizar todas as combinações possíveis dos valores de entrada para obter os sinais de saída e, assim, abreviar a expressão booleana do circuito [8].

2.. Método de Simplificação: Mapa de Karnaugh

Em alguns projetos de sistemas digitais há a preocupação em simplificar circuitos com a finalidade de minimizar as chances de falhas, o consumo de energia elétrica e a quantidade de portas lógicas. Um dos métodos mais utilizados para esta função é o mapa de Karnaugh [9].

Esta técnica consiste em um arranjo, da tabela verdade de um circuito, em células que, se organizados de maneira correta, reduzem uma expressão booleana à sua forma mais simples. Para exemplificar, a fig. 2 mostra como é feita a construção do mapa de Karnaugh a partir da tabela verdade de três variáveis [8].

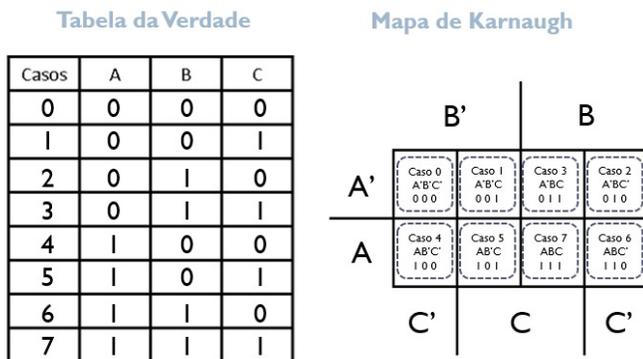


Fig. 2. Tabela verdade versus mapa de Karnaugh [8].

3.. Circuitos Combinacionais e Sequenciais

Em sistemas digitais, existem duas classificações principais de circuitos: circuitos combinacionais e circuitos sequenciais. Os combinacionais são aqueles em que suas saídas dependem apenas de suas entradas, ou seja, não possuem realimentação ou memória. Contrariamente, os sequenciais possuem realimentação de suas saídas até as entradas, necessitando de elementos de memória e sinal de *clock* para controlar a evolução do sistema [7].

Existem dois diferentes tipos de circuitos combinacionais: os circuitos combinacionais lógicos e os aritméticos. A diferença entre ambos é o tipo de função implementada por seus circuitos. Os circuitos lógicos fazem uso de funções lógicas como as das portas lógicas, multiplexadores e demultiplexadores, detectores de paridade, entre outras. Ao contrário, os aritméticos são regidos por funções aritméticas como as dos somadores, subtratores, multiplicadores e divisores [7].

Entre os circuitos sequenciais existem vários que se destacam como os registradores *latch* e *flip-flop*, registradores de deslocamento, contadores e outros [7].

C. Sistemas Embarcados

Em diversos sistemas existem dispositivos encarregados de realizar determinadas funções de forma contínua, geralmente sem quaisquer problemas ou falhas. Estes equipamentos são conhecidos como sistemas embarcados, uma vez que estes são uma parte funcional de todo um sistema [10]. Em sua arquitetura, os embarcados apresentam *hardware* e *software*, com um processador programável, desenvolvidos para executar atividades específicas, diferente de computadores pessoais que possuem múltiplas utilidades distintas [11].

Entre os desafios atuais dos sistemas embarcados se destacam o desenvolvimento de algoritmos com interfaces sofisticadas e adaptação dos mesmos quanto às restrições rígidas de projeto. Outro ponto relevante é a necessidade destes dispositivos atenderem aos estímulos de transmissão de dados e tarefas dos sistemas de tempo real [11].

Embarcados podem ser aplicados em inúmeras áreas como telecomunicações, eletrônica de consumo, automação, automobilismo, sistemas médicos e aeroespaciais [10]. Além do mais, são classificados de acordo com sua arquitetura e seu dispositivo principal. Entre os vários tipos de sistemas embarcados, destacam-se os microprocessadores, microcontroladores e os dispositivos lógicos programáveis (PLD) [12].

1.. Microprocessadores e Microcontroladores

Os microprocessadores são circuitos integrados (CI) que executam instruções demandadas por um programa armazenado em uma memória, por meio de operações aritméticas e booleanas. Também, são capazes de controlar a entrada e saída de dados, o que permite a comunicação do chip com outros dispositivos e periféricos [10].

Para seu correto funcionamento, este tipo de processador precisa se comunicar com alguns componentes como memória, controladores de entrada e saída de dados, entre outros [10]. Devido a esta simplicidade, microprocessadores são chips de baixo custo e podem facilmente ser customizados para atender aos requisitos de vários projetos [11].

Entretanto, o mercado criou um sistema de processamento que possui todos os dispositivos necessários para seu funcionamento, originando os microcontroladores. Estes embarcados são como um “computador em apenas um chip”, ou seja, em seu chip estão integrados um processador, memórias, funções de entrada e saída, entre outros. A fig. 3 apresenta um diagrama com todos os componentes internos de um microcontrolador simples se comunicando com a unidade central de processamento (CPU) [10].

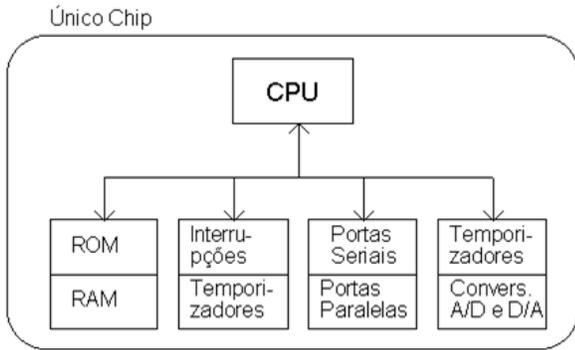


Fig. 3. Diagrama de um sistema microcontrolado [10].

2.. Dispositivos Lógicos Programáveis

Em meados da década de 1970 foram introduzidos, no mercado, circuitos combinacionais lógicos que pudessem ser programados, denominados dispositivos lógicos programáveis (PLD). Este é um tipo de chip para uso geral que possui *hardware* configurável, já que sua programação é realizada em nível de *hardware*, diferente dos microprocessadores, que apenas executam programas e possuem *hardware* fixo [7].

Entre os PLDs existe uma infinidade de chips diferentes, a maioria destes compreendidos em um coletivo chamado de dispositivos lógicos programáveis simples (SPLD). No entanto, os mais importantes e conhecidos são os dispositivos lógicos programáveis complexos (CPLD), os arranjos de portas programáveis em campo (FPGA) [7] e os circuitos integrados de aplicação específica (ASIC) [13].

Os CPLDs apresentam em sua arquitetura uma associação de vários dispositivos lógicos programáveis simples que se comunicam por meio de um arranjo de interconexões complexo e programável. Também, possui *drivers* de entradas/saídas (I/O), pinos de I/O e suporte para o padrão de comunicação JTAG. A fig. 4 mostra a construção de um CPLD convencional [7].

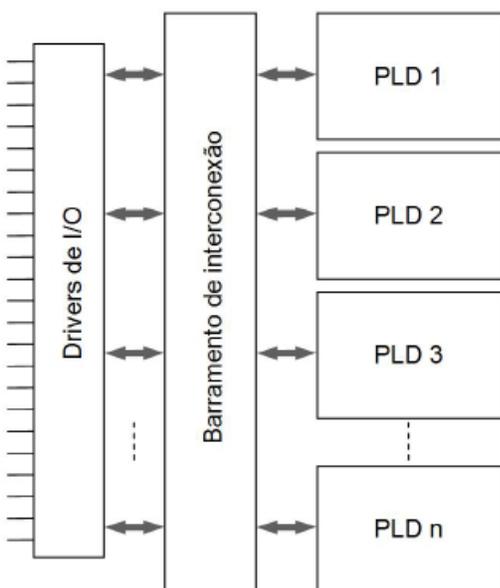


Fig. 4. Arquitetura básica de CPLDs [7].

Com relação aos dispositivos lógicos programáveis complexos, os FPGAs revelam diferenças quanto à sua arquitetura, tecnologia, características embutidas, tamanho, desempenho e custo [7]. Pois são circuitos programáveis que, em sua construção, apresentam um conjunto de blocos lógicos dispostos no formato de uma matriz [5], não em forma de pilha de blocos como nos CPLDs. Sua matriz pode ser composta por blocos lógicos programáveis (CLB) ou blocos de matriz lógica (LAB) dependendo do fabricante, seja Xilinx ou Altera, e modelo. Além do mais, também possui blocos de memória de acesso aleatório (RAM), blocos de processador de sinal digital (DSP) e suporte JTAG. A fig. 5 ilustra as arquiteturas simplificadas de FPGAs Xilinx Virtex 5 e Altera Stratix III [7].

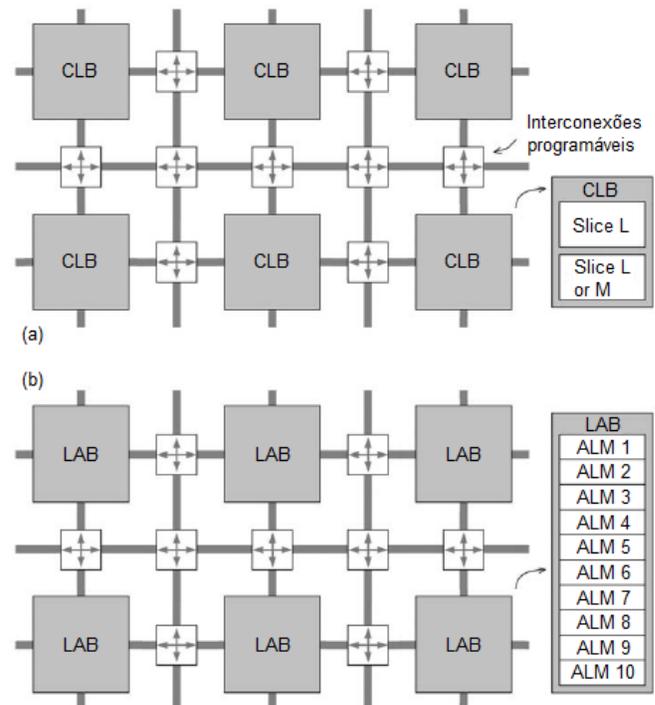


Fig. 5. Arquitetura simplificada de FPGAs: (a) Xilinx Virtex 5; (b) Altera Stratix III [7].

O PLD com matriz de blocos lógicos oferece maior economia de energia que os dispositivos lógicos programáveis complexos, porém possui menos recursos. Também, o CPLD possui um número de pinos de entradas superior ao do FPGA [14].

O circuito integrado de aplicação específica (ASIC) é composto por componentes elétricos como transistores, capacitores e resistências em uma placa de silício ou outro material semiconductor. Este CI é personalizado para uso específico em sua fabricação. Se comparado ao FPGA, que pode ser facilmente reprogramado quantas vezes for necessário, o ASIC apresenta menor economia e flexibilidade em seu uso, já que este só pode ser configurado novamente ao apagar sua programação nativa [13].

Quanto à aplicação de FPGAs e outros PLDs, pode-se dizer que estes são dispositivos bastante utilizados em sistemas de processamento de sinais digitais. Além disso, tem sua importância em processamento de imagens, detecção de

bordas em tempo real, segmentação de imagem, reconhecimento de padrões e análise de textura [15]. Estes chips são também utilizados em sistemas industriais, sistemas militares, equipamentos médicos, robótica, telecomunicações, computadores, eletrônicos, comunicação de dados, periféricos, entre outros [5].

D. Linguagens de Descrição de Hardware

Dispositivos lógicos programáveis, como FPGAs, necessitam de algum tipo de linguagem para implementar circuitos físicos em seus próprios *hardwares*. Esta espécie de linguagem é denominada como linguagem de descrição de *hardware* (HDL) [5].

As HDLs são organizadas de maneira que propiciem a descrição abstrata do comportamento de *hardwares*. Em outras palavras, é possível modelar e representar a estrutura e funcionamento de circuitos em vários níveis de abstração durante o projeto [5]. Por conseguinte, isto garante ao projetista maior flexibilidade na descrição de circuitos complexos [4].

A princípio, estas linguagens eram geralmente utilizadas para verificação de lógica. Apenas com o surgimento de algoritmos de síntese lógica que esta realidade mudou, pois assim era possível a descrição do fluxo de dados de circuitos, ou seja, em nível de transferência de registradores (RTL). Isto facilitou a maneira de descrever circuitos digitais, porque detalhes como portas lógicas e interconexões são automaticamente processados pelos algoritmos de síntese lógica [4].

Linguagens de descrição de *hardware* possuem algumas vantagens com relação a projetos tradicionais (baseados em esquemas elétricos). É possível a descrição de circuitos digitais independente da tecnologia de fabricação, por causa do nível de abstração deste tipo de linguagem. Também é possível a verificação funcional de circuitos ainda no estágio preliminar de projeto, bem como a facilidade em análise de falhas e erros, pois HDLs são linguagens textuais como as linguagens de programação convencionais (C, C++, Java e outras) e carregam consigo características semelhantes [4].

Existe uma variedade de HDLs. Porém, entre as mais conhecidas e utilizadas se destacam a linguagem Verilog e a linguagem de descrição de *hardware* de circuito integrado de alta velocidade (VHDL) [5].

1.. Verilog

A linguagem Verilog (ou Verilog HDL) é uma linguagem de descrição de *hardware* de aplicação geral fácil de se aprender e usar. Sua versão aberta surgiu em 1995 e suas atualizações foram divulgadas em 2001 e 2005 (ano em que foi normatizada por meio da norma IEEE 1364-2005) [4].

A sintaxe da linguagem Verilog é parecida com a da linguagem de programação C. Assim, projetistas com experiência em C podem compreender facilmente a linguagem Verilog. Devido a esta similaridade, existe uma ferramenta que permite a interação de códigos escritos em C com estruturas de dados internas da linguagem Verilog, denominada interface processual Verilog (VPI) [4].

Esta linguagem permite unir diferentes níveis de abstração em um mesmo modelo, seja em termos de portas lógicas, RTL ou código comportamental. Também, é possível a simulação do *hardware* por meio da geração de sinais de estímulo para testes [4]. No entanto, a ordem de compilação influencia nos resultados das simulações, ou seja, não se pode compilar o código em qualquer ordenação [16].

Outra característica é a não existência de conceito de pacotes em Verilog. Funções e procedimentos devem ser definidos na instrução do módulo com a qual será usado [16].

A maioria das ferramentas de síntese suportam Verilog HDL. Além do mais, fabricantes de circuitos integrados disponibilizam bibliotecas para simulações após síntese lógica [4].

2.. VHDL

Na década de 80, o Departamento de Defesa dos Estados Unidos (DoD) financiou a criação de uma linguagem que é capaz de descrever o comportamento desejado de qualquer circuito físico, a linguagem VHDL [7]. Uma de suas principais características é a modelagem de circuitos digitais em um elevado nível de abstração [5]. Todavia, também é possível a descrição em baixo nível de abstração por meio de operadores lógicos [16].

Esta é uma linguagem de descrição de *hardware* que não depende de tecnologia ou fabricante [7]. Isto significa que vários dispositivos podem compilar códigos em VHDL [17]. O que facilita a atualização e adaptação de projetos aos avanços tecnológicos [18].

VHDL é a primeira HDL a ser padronizada (padrões 1076 e 1164) pelo Instituto de Engenheiros Elétricos e Eletrônicos (IEEE) [7]. Ademais, esta é amplamente utilizada pelo setor industrial para modelagem, projeto e simulação de sistemas digitais [14].

Entre as características de VHDL destacam-se a fácil reutilização da linguagem em diferentes tecnologias, a reciclagem de projetos já descritos antes por meio da criação de bibliotecas de módulos e a possibilidade de simulação de circuitos para verificar o funcionamento dos mesmos [5]. Também é possível compilar separadamente unidades de *design* (entidade e arquitetura) [16].

Outra importante qualidade desta linguagem é a de procedimentos forma paralela. Isto quer dizer que circuitos construídos em VHDL executam suas tarefas paralelamente, desde que as tarefas não dependam de outras [16].

Esta linguagem possui uma notável capacidade no gerenciamento de projetos grandes, porque permite decompor a estrutura dos mesmos em unidades menores [18]. O que torna seus projetos apropriados para aplicações onde a escalabilidade é uma consideração de *design* [19].

E. Métodos Estatísticos para Análise de Dados

A estatística é um conjunto de métodos que serve para obter, organizar, analisar e interpretar dados para uma determinada finalidade. Consequentemente, com os resultados dos estudos estatísticos, são geradas conclusões sobre um determinado assunto [20].

Entre todas estas ferramentas existem duas que organizam, resumem e permitem a visualização das informações colhidas. A primeira é a tabela de frequência que serve para relacionar classes de valores com suas devidas contagens. A segunda é a representação pictórica de dados que são muito úteis quando se deseja visualizar a natureza da distribuição de dados [20].

A forma mais comum de se representar dados é por meio de histograma (gráfico). Este consiste em duas escalas, horizontal e vertical, onde a primeira apresenta valores de dados que se deseja representar e o segundo apresenta as frequências e barras que retratam os valores pertencentes às categorias [20].

Outra técnica importante é a análise exploratória de dados que é utilizada para entender as informações contidas nos dados com o intuito de aprofundar o conhecimento sobre o assunto pesquisado. Com esta técnica não é necessário realizar cálculos complexos ou levantar muitas hipóteses sobre as informações estudadas [20].

V. DESENVOLVIMENTO

A. Método Prático Proposto

O método prático proposto é uma nova abordagem de ensino para a unidade curricular de Eletrônica Digital. Este é baseado na descrição de *hardware* por meio do uso de FPGAs e linguagem VHDL. Os motivos para a escolha deste dispositivo e linguagem foram apresentados nas subseções “Sistemas Embarcados” e “Linguagens de Descrição de Hardware” localizadas na seção “Revisão de Bibliografia”.

Segundo Barbosa e Moura, aprendizagem ativa é aquela em que o aluno realiza tarefas mentais de alto nível como análise, síntese e avaliação pensar sobre a atividade que está desempenhando [2]. O método prático baseado em descrição de *hardware* segue esta mesma linha de pensamento, pois com o emprego de HDLs, no caso VHDL, é possível criar circuitos digitais em vários níveis de abstração (baixo ou alto), o que exige do projetista um profundo conhecimento sobre o circuito a ser descrito [5]. Por conseguinte, o discente utiliza bastante suas funções intelectuais (raciocínio, observação, entendimento, combinação de ideias, entre outras) [2].

A aplicação de FPGA no método está compreendida na conferência do funcionamento do circuito descrito pelo estudante. Este dispositivo implementa o *hardware* projetado [7], permitindo que seu usuário verifique a atividade do circuito digital desenvolvido.

Uma característica relevante do método proposto é a possibilidade de descrição de *hardware* com base na metodologia *top-down*. Assim, sistemas digitais podem ser desenvolvidos a partir do alto nível de abstração até o baixo nível [16].

1.. Diretrizes do Método Prático Proposto

No emprego de todo e qualquer método prático de ensino é necessário definir bem os procedimentos a serem seguidos. Desta forma, o método aqui proposto apresenta diretrizes para sua correta aplicação:

- O professor deve apresentar os conceitos básicos da unidade curricular de Eletrônica Digital por meio de aulas expositivas, pois conceitos como, por exemplo, portas lógicas são essenciais para atividades práticas desta disciplina;
- É importante, também, ensinar as estruturas básicas da linguagem de descrição de *hardware* VHDL por meio de aulas expositivas, permitindo, assim, que o aluno busque o conhecimento de comandos mais avançadas da linguagem;
- Projetos práticos devem ser definidos para serem desenvolvidos em aulas laboratoriais com base nos tópicos estudados em sala de aula. Se o docente quiser aplicar a aprendizagem baseada em projetos (ABProj) [2], estes precisam ser fundamentados em sistemas reais;
- Após a escolha do projeto, os mesmos precisam ser descritos em linguagem VHDL;
- Com o sistema desenvolvido, o aluno deve realizar uma simulação do circuito descrito em um *software* de descrição de *hardware*;
- Depois de simulado, o projeto deve ser implementado em um FPGA. Esta ação só é possível se o *software* utilizado for compatível com o dispositivo;
- Incorporar componentes externos (botões, chaves, *displays*, entre outros) ao dispositivo lógico programável, caso houver necessidade;
- Testes com o circuito sintetizado no FPGA precisam ser desempenhados para a conferência do projeto em questão;
- É de suma importância o professor conferir o funcionamento do projeto realizado e dar um *feedback* aos alunos;
- Se o professor deseja avaliar os projetos, documentos sobre a elaboração do trabalho e seus resultados devem ser redigidos pelos alunos.

B. Exemplo de Projeto Prático

A fim de exemplificar o uso da linguagem VHDL e dispositivo FPGA, ferramentas do método proposto neste trabalho, foi escolhido um projeto prático. Projeto este que é um sistema de transmissão e recepção de dados, composto por um multiplexador e um demultiplexador, conforme é apresentado pela fig. 6.

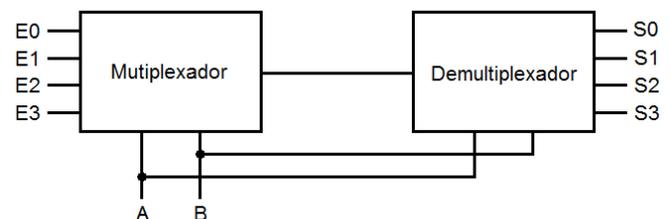


Fig. 6. Diagrama de um sistema de transmissão e recepção de dados.

O multiplexador é a parte responsável por transmitir os *bits* contidos nas entradas (E0, E1, E2 e E3) do sistema para o demultiplexador. A ligação entre os dois blocos é feita por um único barramento que entrega as informações ao demultiplexador que, por sua vez, as envia para suas saídas (S0, S1, S2 e S3).

Conforme for a seleção de “A” e “B”, o sinal de uma das entradas é direcionado até a saída correspondente, caso seu *bit* seja de nível lógico alto (número binário 1). O quadro I mostra a relação entre as entradas e saídas do sistema.

QUADRO I

TABELA VERDADE DO SISTEMA DE TRANSMISSÃO E RECEPÇÃO DE DADOS

E0	E1	E2	E3	A	B	S0	S1	S2	S3
1	0	0	0	0	0	1	0	0	0
0	1	0	0	0	1	0	1	0	0
0	0	1	0	1	0	0	0	1	0
0	0	0	1	1	1	0	0	0	1

O *bit* 1 é transmitido de uma entrada para a saída correspondente quando a seleção de “A” e “B” correta for satisfeita. Por exemplo, se A estiver com *bit* 0 e B também, a informação de E0 é transmitida até S0. Cada linha do quadro I mostra o comportamento de saída do circuito de acordo com os *bits* que as entradas e chaves seletoras apresentarem.

Com base nestas informações, apresentadas no quadro I, o projeto foi descrito em VHDL e, em seguida, simulado. Ambos os processos foram realizados por meio de um *software* denominado como Quartus Prime Lite Edition, da empresa Altera. A simulação foi realizada com o objetivo de conferir o algoritmo desenvolvido por meio de um teste de formas de ondas digitais, onde são mostrados os sinais das chaves seletoras e das saídas do sistema.

O algoritmo elaborado, que descreve o comportamento do sistema de transmissão de dados, é apresentado na seção Apêndice como Apêndice 1. E, o resultado da simulação com os sinais das chaves seletoras e das saídas do projeto, o qual indica a lógica de saída do projeto, é apresentado pela fig. 7.

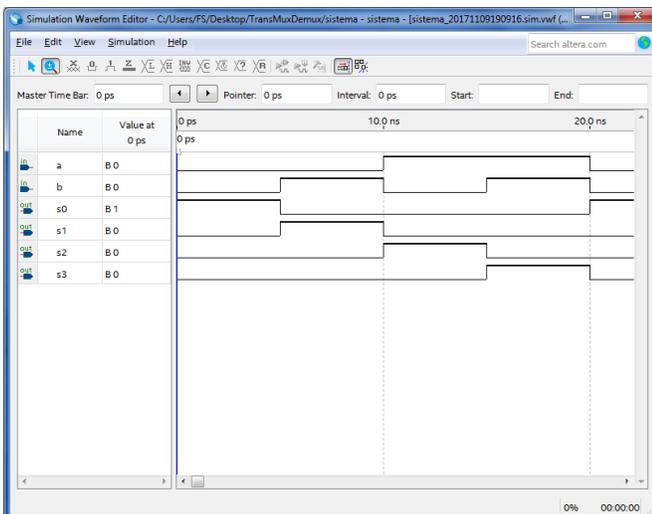


Fig. 7. Simulação dos sinais de seleção e saídas do circuito.

Após a simulação, o algoritmo desenvolvido foi transferido para o *kit* de desenvolvimento FPGA Cyclone IV da Altera. A partir disso foi possível visualizar o sistema funcionando por meio de quatro diodos emissores de luz (LED), que representam as saídas, e dois botões, que correspondem às variáveis de seleção “A” e “B”. A fig. 8 mostra a placa do FPGA com o projeto do sistema de transmissão de dados em funcionamento.

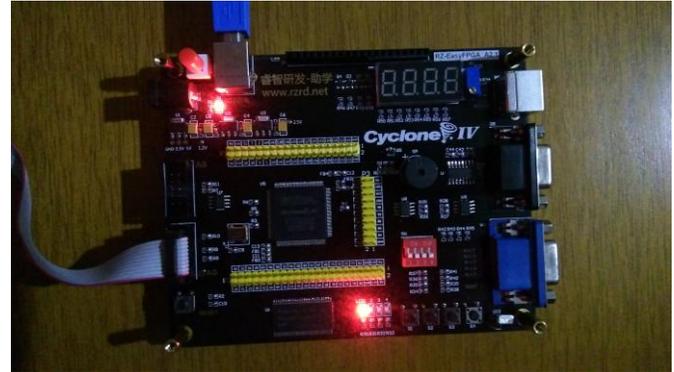


Fig. 8. Placa de desenvolvimento FPGA Cyclone IV com sistema de transmissão e recepção de dados em funcionamento.

As entradas E0, E1, E2 e E3 correspondem a nenhum dos botões do Cyclone IV, pois não há botões suficientes para as entradas e as variáveis de seleção. Então, por meio do algoritmo em VHDL foi determinado que estas entradas fossem representadas por sinais (condutores) apenas com nível lógico alto.

C. Pesquisa de Campo

A presente pesquisa foi realizada com a finalidade de comparar o método prático proposto, baseado na linguagem de descrição de *hardware* VHDL e no dispositivo lógico programável FPGA, com o método tradicional, que consiste na montagem de circuitos digitais com CIs e *proto-board*.

A amostra foi composta apenas por estudantes matriculados na unidade curricular de Eletrônica Digital, durante o segundo semestre do ano de 2017, e que cursam a graduação em Engenharia Elétrica da Fundação Presidente Antônio Carlos (FUPAC).

Os métodos mencionados foram aplicados durante o mês de outubro do ano de 2017. A pesquisa foi realizada em três dias de aulas práticas, realizadas no laboratório de eletrônica da FUPAC, para a disciplina de Eletrônica Digital, totalizando cinco horas. Na primeira destas três aulas foi empregado o método tradicional para verificar o funcionamento de um circuito combinacional baseado em portas lógicas. Depois, no segundo dia, foi utilizado o método de descrição de *hardware* em VHDL e FPGA. E, ao fim, no último dia, foi aplicado um questionário para que os alunos respondessem. A fig. 9 revela o circuito digital reproduzido pelos discentes.

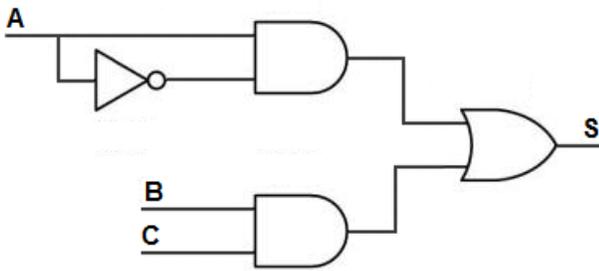


Fig. 9. Circuito combinacional utilizado na pesquisa.

No questionário foram feitas perguntas com relação a ambos os métodos. Estas foram escritas para investigar as opiniões dos discentes sobre os dois métodos utilizados, além de questionar sobre suas experiências com os mesmos.

O método estatístico foi a base para a análise exploratória dos resultados do trabalho realizado em campo. Suas etapas foram fundamentais para que a descrição, comparação e análise das informações fossem realizadas de forma categórica e imparcial, bem como contribuíram para a compreensão destes.

VI. RESULTADOS DA PESQUISA DE CAMPO

Os resultados foram obtidos por meio das respostas encontradas nos questionários que 15 estudantes responderam. O critério para a seleção desta amostra foi a condição de o aluno não ter faltado em nenhuma das etapas do trabalho de campo desta pesquisa.

No que diz respeito à preferência dos alunos, 46,67% escolheram a montagem de circuitos digitais com CIs e *protoboard*, ninguém optou pela descrição de circuitos em VHDL e FPGA, 46,67% preferiram ambos os métodos e 6,67% marcou a opção nenhum.

A fig. 10 apresenta o gráfico que fornece os métodos em que os discentes relataram dificuldades. A maior parte apontou que suas dificuldades se concentraram no uso da técnica baseada em descrição de *hardware*, proposto neste artigo.

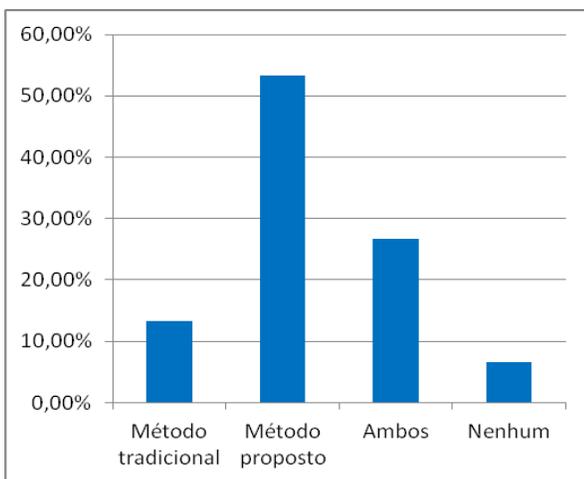


Fig. 10. Gráfico sobre os métodos em que os alunos apresentaram dificuldades.

Com relação ao método tradicional, 40% das dificuldades relatadas pelos alunos se concentraram no entendimento da disposição das conexões internas (informações de *datasheets*) de circuitos integrados, 20% são referentes à realização de conexões utilizando circuitos integrados na *protoboard*, outros 20% sobre a montagem de circuitos digitais com mais de um circuito integrado e 20% relacionados a outros tipos de dificuldades. Os pesquisados não relataram dificuldades na verificação do funcionamento de circuitos digitais montados.

Já na prática em que foi utilizada a descrição de *hardware*, 37,5% das dificuldades mencionadas foram com relação ao entendimento da sintaxe da linguagem VHDL, 25% sobre a descrição de portas lógicas e suas conexões, 18,75% direcionadas à verificação do funcionamento de circuitos em FPGAs e 18,75% correspondem a outras. Não foram declaradas dificuldades relacionadas à descrição de circuitos digitais com mais de uma porta lógica.

Quando perguntando aos discentes qual método recomendariam a outros estudantes de eletrônica digital, 86,67% aconselhariam ambos, seguido de 13,33% que recomendariam o tradicional, ninguém iria sugerir o uso do método proposto e de nenhum outro mencionado no questionário.

Também foi questionado qual o método que utilizariam para desenvolver circuitos reais. O tradicional foi o preferido para 53,33% dos alunos, acompanhado de ambos com 33,33%. Apenas 6,67% fariam uso da descrição de *hardware* e 6,67% utilizaria nenhum.

Além disso, foi perguntado “Qual método prático despertou seu interesse em saber mais a respeito da eletrônica digital?”. A maioria dos alunos (46,67%) respondeu que foi a montagem de circuitos integrados em *protoboards*, seguidamente de ambos (40%), depois da descrição de *hardware* em VHDL e FPGA (13,33%) e de nenhum dos métodos (0%).

Outras duas perguntas foram respondidas. Estas com relação ao grau de utilidade em constatar e consolidar conceitos de eletrônica digital. O primeiro tópico foi sobre a álgebra booleana e o segundo relacionado às portas lógicas. As fig. 11 e 12 apresentam os gráficos referentes a estas questões.

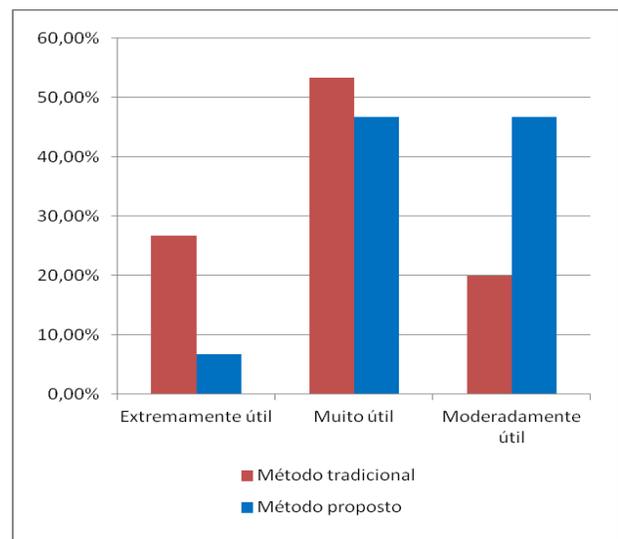


Fig. 11. Gráfico sobre o grau de utilidade dos métodos em constatar e consolidar conceitos da álgebra booleana.

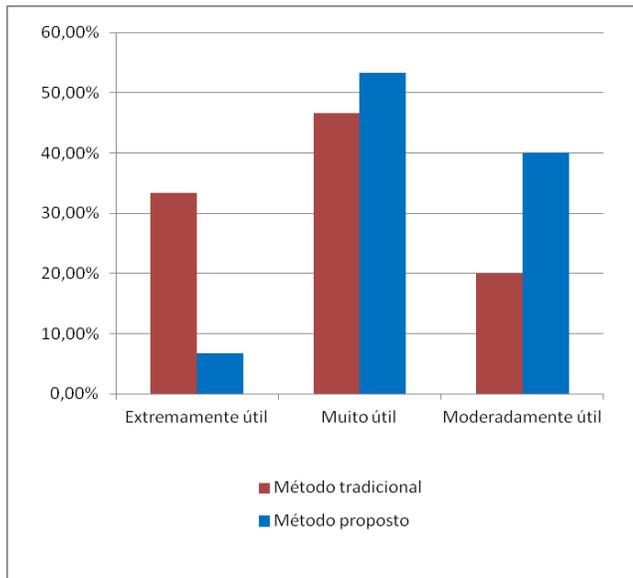


Fig. 12. Gráfico sobre o grau de utilidade dos métodos em constatar e consolidar conceitos de portas lógicas.

Ambas, fig. 11 e fig. 12 ilustram a porcentagem de pessoas que marcaram alguma das opções (extremamente, muito e moderadamente útil). Todavia, as alternativas restantes (pouco e nada útil) foram omitidas dos gráficos, porque ninguém as escolheu.

A fig. 11 revela que a maioria dos alunos achou o método tradicional mais útil para o aprendizado de álgebra booleana e a fig. 12 mostra o mesmo com relação aos conceitos de portas lógicas.

Ao fim do questionário perguntou-se o quão útil foram os dois métodos práticos para verificar tabelas verdade e o maior número das respostas sugere que foi o método de montagem de circuitos digitais com CIs e *protoboards*, conforme é revelado pela fig. 13. Neste a única opção não apresentada é a nada útil, por causa do percentual de sua frequência (0%).

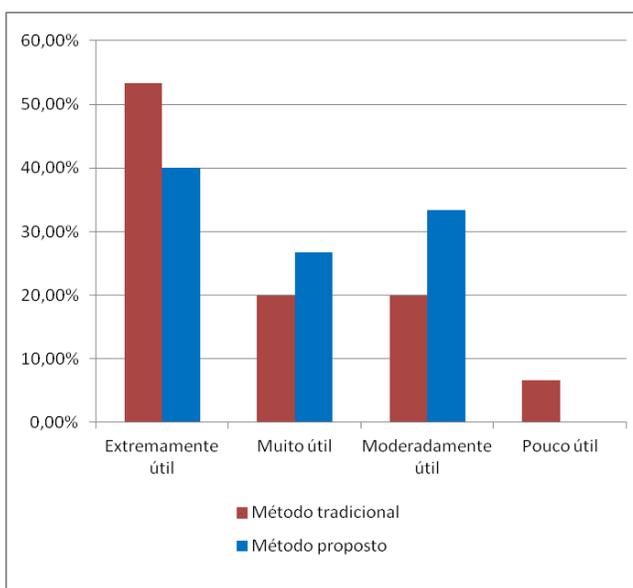


Fig. 13. Gráfico sobre o grau de utilidade dos métodos na verificação de tabelas verdade.

As informações apresentadas pelos gráficos atestaram a preferência dos discentes pelo método tradicional. O que contraria as ideias de aprendizado ativo que fundamentaram o método baseado em descrição de *hardware*.

Acredita-se que os resultados, obtidos pela pesquisa de campo, estão relacionados a não implantação efetiva de metodologias ativas de ensino na FUPAC, bem como a resistência natural em mudanças de paradigmas. Segundo Forster e Fagundes, esta oposição a novos modelos de ensino acontecem, principalmente, nas graduações da área de exatas [21], como no caso deste trabalho.

VII. CONCLUSÕES

O método de ensino prático de Eletrônica Digital que utiliza a descrição de *hardware* com FPGA e VHDL foi desenvolvido e fundamentado na aprendizagem ativa [2]. Sua eficácia, quanto à concepção de circuitos digitais em alto nível de abstração, foi comprovada por meio de conteúdo bibliográfico e do exemplo de projeto de um sistema de transmissão de dados.

Entretanto, a pesquisa de campo realizada revelou que os estudantes questionados apontaram maior identidade com o método tradicional. Consequência, esta, da resistência natural a novos paradigmas de ensino, comum entre alunos de graduações da área de exatas [21], além da pouca aplicação de metodologias ativas na FUPAC.

A maior dificuldade encontrada durante o trabalho, foi a preparação dos discentes participantes da pesquisa com relação ao método proposto neste artigo. Visto que neste é utilizado uma linguagem de descrição de *hardware*, ferramenta semelhante às linguagens de programação em geral, e o tempo disponibilizado para a realização do trabalho de campo foi curto. Todavia, foi possível concluir esta pesquisa.

Como sugestão para trabalhos futuros a descrição de *hardware* pode ser aplicada em unidades curriculares que utilizam conceitos de eletrônica digital como, por exemplo, Organização de Computadores. Outrossim, podem ser desenvolvidos projetos utilizando FPGA e VHDL, tais como: controle proporcional integral derivativo (PID) discreto para alguma aplicação, controle por largura de pulso (PWM) de um motor de indução trifásico, sistemas de processamento de imagens, redes neurais, entre outros.

VIII. REFERÊNCIAS

- [1] BORGES, M. N.; ALMEIDA, N. N. Perspectivas para engenharia nacional: desafios e oportunidades. *Revista de Ensino de Engenharia*, v. 32, n. 3, p. 71-78, 2013. Disponível em: <<http://198.136.59.239/~abengeorg/revista/index.php/abenge/article/viewFile/238/170>>. Acesso em: 17 set. 2017.
- [2] BARBOSA, E. F.; MOURA, D. G. Metodologias ativas de aprendizagem no ensino de engenharia. In: XIII International Conference on Engineering and Technology Education, 2014. Science and Education Research Council, COPEC - Guimarães: INTERTECH, 2014. v. 13, n. 1. Disponível em: <<http://copec.eu/congresses/intertech2014/proc/works/25.pdf>>. Acesso em: 17 set. 2017.
- [3] LUNA, J. D. F. O. et al. Uma atividade prática abrangente em eletrônica digital: onda verde. In: Congresso Brasileiro de Ensino de Engenharia – ABENGE, 2014. Juiz de Fora. Anais (on-line). Juiz de Fora: ABENGE,

2014. Disponível em: <<http://198.136.59.239/~abengeorg/cobenge-2014/Artigos/129291.pdf>>. Acesso em: 28 ago. 2017.
- [4] COSTA, C. A. Desenvolvimento de hardware de criptografia RSA em linguagem Verilog. 2014. 123 f. Dissertação (Mestrado) – Universidade Federal de Itajubá, 2014. Disponível em: <<https://repositorio.unifei.edu.br/xmlui/handle/123456789/319>>. Acesso: 29 out. 2017.
- [5] ORDONEZ, E. D. M. et al. *Projeto, desempenho e aplicações de sistemas digitais em circuitos programáveis (FPGAs)*. Pompéia: Bless, 2003. 300 p.
- [6] SIQUEIRA, T. A. C. et al. Relações entre o desempenho acadêmico e o desempenho profissional. In: Congresso Brasileiro de Ensino de Engenharia - ABENGE, 2014. Juiz de Fora. Anais (on-line). Juiz de Fora: ABENGE, 2014. Disponível em: <<http://198.136.59.239/~abengeorg/cobenge-2014/Artigos/128940.pdf>>. Acesso em: 28 ago. 2017.
- [7] PEDRONI, V. A. *Eletrônica Digital Moderna e VHDL*. Traduzido por Arlete Simille Marques. Rio de Janeiro: Elsevier, 2010. 619 p.
- [8] MUNARINI, B. L. Desenvolvimento de um kit didático para eletrônica digital. 2016. 92 f. Trabalho de Conclusão de Curso (Graduação) - Universidade Tecnológica Federal do Paraná, Campo Mourão, 2016. Disponível em: <<http://repositorio.roca.utfpr.edu.br/jspui/handle/1/6896>>. Acesso em: 18 set. 2017.
- [9] BECKER, L. et al. Um estudo sobre a adoção de softwares para simplificação de expressões booleanas, como objetos de aprendizagem, em um curso de engenharia de computação. In: Congresso Brasileiro de Ensino de Engenharia - ABENGE, 2014. Juiz de Fora. Anais (on-line). Juiz de Fora: ABENGE, 2014. Disponível em: <<http://198.136.59.239/~abengeorg/cobenge-2014/Artigos/129079.pdf>>. Acesso em: 28 ago. 2017.
- [10] PEREIRA, L. A. M. et al. Software embarcado, o crescimento e as novas tendências deste mercado. *Revista de Ciências Exatas e Tecnologia*, v. 6, n. 6, p. 85-94, 2014. Disponível em: <<http://pgsskroton.com.br/seer/index.php/rcext/article/view/2308>>. Acesso em: 24 set. 2017.
- [11] LEITE, M. Tratamento de requisitos não funcionais em sistemas de tempo-real embarcados implementados em VHDL/FPGA. 2014. 178 f. Dissertação (Mestrado) – Universidade do Estado de Santa Catarina, Joinville, 2014. Disponível em: <<http://tede.udesc.br/handle/handle/2037>>. Acesso em: 24 set. 2017.
- [12] SOSA, J. C. et al. Sistema embebido para la detección de luz láser empleando el soft-core Nios II. *Research in Computing Science*, México, v. 107, p. 121-132, 2015. Disponível em: <http://www.rcs.cic.ipn.mx/rcs/2015_107/Sistema%20embebido%20para%20la%20deteccion%20de%20luz%20laser%20empleando%20el%20soft-core%20Nios%20II.pdf>. Acesso em: 26 set. 2017.
- [13] MOORE, A.; WILSON, R. *FPGAs for dummies*. 2ªed. New Jersey: John Wiley & Sons, Inc, 2017. 46 p. Disponível em: <https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/misc/fpgas_for_dummies_ebook.pdf>. Acesso em: 16 jul. 2017.
- [14] KAUR, J. An overview of hardware approach of FPGA design by using Xilinx. *International Journal of Research Aspects of Engineering and Management*, Baru Sahib, v. 1, p. 87-90, 2014. Disponível em: <<http://emax.edu.in/journal/vol1-02/IJRAEM-01-02-24.pdf>>. Acesso em: 24 set. 2017.
- [15] KAVIAN, Y. Hardware description of multi-directional fast sobel edge detection processor by VHDL for implementing on FPGA. *International Journal of Computer Applications*, Ahvaz, v. 47, n. 25, p. 1-7, 2012. Disponível em: <<https://www.researchgate.net/publication/258651823>>. Acesso em: 24 set. 2017.
- [16] SMITH, D. J. *HDL chip design: a practical guide for designing, synthesizing and simulating ASICs and FPGAs using VHDL or Verilog*. Madison: Doone Publications, 1996. 446 p. Disponível em: <[http://ebook.pldworld.com/_eBook/FPGA%EF%BC%8FHDL/-Eng-/HDL%20Chip%20Design.%20A%20Practical%20Guide%20for%20Designing,%20Synthesizing%20and%20Simulating%20ASICs%20and%20FPGAs%20Using%20VHDL%20or%20Verilog%20\(Douglas%20Smith\).pdf](http://ebook.pldworld.com/_eBook/FPGA%EF%BC%8FHDL/-Eng-/HDL%20Chip%20Design.%20A%20Practical%20Guide%20for%20Designing,%20Synthesizing%20and%20Simulating%20ASICs%20and%20FPGAs%20Using%20VHDL%20or%20Verilog%20(Douglas%20Smith).pdf)>. Acesso: 26 set. 2017.
- [17] HOURNÉ, E. A. P.; GARCÉS-SOCARRÁS, L. M.; SARMIENTO, A. J. C. Diseño de bloques para el procesado de imágenes en lenguaje de descripción de hardware. *Ingeniería Electrónica Automática y Comunicaciones*, La Habana, v. 34, n.2, 2013. Disponível em: <http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1815-59282013000200002>. Acesso em: 26 set. 2017.
- [18] CASTAÑO, J. A. et al. Red de procesadores evolutivos para solucionar El problema de los tres colores. Implementación en hardware. *Revista Cubana de Ciencias Informáticas*, La Habana, v. 9, n. 4, 2015. Disponível em: <http://scielo.sld.cu/scielo.php?pid=S2227-18992015000400009&script=sci_arttext&tlng=en>. Acesso em: 26 set. 2017.
- [19] MEYTHALER, A. N. et al. Desarrollo de un sistema de navegación autónoma para UAV basado en FPGA. Universidad de las Fuerzas Armadas, 2014. Disponível em: <<https://repositorio.espe.edu.ec/bitstream/21000/9302/1/AC-ESPEL-ENI-0335.pdf>>. Acesso em: 26 set. 2017.
- [20] TRIOLA, M. F. *Introdução à estatística*. Tradução de Vera Regina Lima de Farias e Flores. 9 ed. Rio de Janeiro: LTC, 2005. 656 p.
- [21] FORSTER, M. M. S.; FAGUNDES, M. C. V. Inovações educativas na sala de aula universitária: ruptura paradigmática /resistência ao ethos regulatório?. In: 29ª Reunião Anual da ANPEd, 2006. Caxambu. Anais (on-line). Caxambu: ANPEd, 2006. Disponível em: <<http://www.anped.org.br/sites/default/files/gt11-1885-int.pdf>>. Acesso em: 06 nov. 2017.

IX. APÊNDICE

Apêndice 1 – Algoritmo do projeto de sistema de transmissão de dados descrito em linguagem VHDL.

```

abc Text Editor - C:/Users/FS/Desktop/TransMuxDemux/sistema - sistema - [sistema.vhd]*
File Edit View Project Processing Tools Window Help Search altera.com
267
268
1  library IEEE;
2  use IEEE.std_logic_1164.all;
3
4
5  entity sistema is port
6  (
7      a, b      : in std_logic; -- Entradas das chaves seletoras do sistema
8      s0, s1, s2, s3 : out std_logic -- Saídas do sistema
9  );
10 end sistema;
11
12
13 architecture hardware of sistema is
14
15     signal sel      : std_logic_vector(1 downto 0); -- Vetor das chaves seletoras
16     signal e0, e1, e2, e3 : std_logic; -- Sinais que representam as entradas de dados no sistema
17     signal y        : std_logic; -- Sinal de transmissão de dados entre mux e demux
18
19 begin
20
21     sel <= a & b; -- "sel" recebe "a" e "b"
22     e0 <= '1'; -- "e0" recebe "1"
23     e1 <= '1'; -- "e1" recebe "1"
24     e2 <= '1'; -- "e2" recebe "1"
25     e3 <= '1'; -- "e3" recebe "1"
26
27     -- Multiplexador --
28     with sel select
29         y <= e0 when "00", -- "y" recebe "e0" quando "sel" for "00"
30             e1 when "01", -- "y" recebe "e1" quando "sel" for "01"
31             e2 when "10", -- "y" recebe "e2" quando "sel" for "10"
32             e3 when "11"; -- "y" recebe "e3" quando "sel" for "11"
33
34     -- Demultiplexador --
35     s0 <= y when (sel = "00") else '0'; -- "s0" recebe "y" quando "sel" for "00", senão recebe "0"
36     s1 <= y when (sel = "01") else '0'; -- "s1" recebe "y" quando "sel" for "01", senão recebe "0"
37     s2 <= y when (sel = "10") else '0'; -- "s2" recebe "y" quando "sel" for "10", senão recebe "0"
38     s3 <= y when (sel = "11") else '0'; -- "s3" recebe "y" quando "sel" for "11", senão recebe "0"
39
40 end hardware;

```

Ln 40 Col 14 VHDL File 0% 00:00:00