



## DESENVOLVIMENTO DE UM KIT DIDÁTICO USANDO FPGA PARA ESTUDO DA ARQUITETURA DE MICROPROCESSADORES X86

**Guilherme de S. Sampaio** – guilhermesampaio1997@gmail.com

**Isaac B. Benchimol** – ibench@ifam.edu.br

**Hillermann F. O. Lima** – hillermann@ifam.edu.br

Instituto Federal de Educação, Ciência e Tecnologia do Amazonas – IFAM Campus Manaus Distrito Industrial.

Av. Gov. Danilo de Matos Areosa, 1672, Distrito Industrial.

69075351 – Manaus – Amazonas

**Resumo:** A tecnologia de circuitos integrados digitais reconfiguráveis, como FPGA (Field Programmable Gate Array), tornou possível a aplicação deste dispositivo em projetos com configurações e complexidades diversas. A importância prática desta tecnologia pode ser estendida ao ensino de disciplinas de cursos de Engenharia tais como Sistemas Digitais e Arquitetura de Computadores. Este trabalho descreve o desenvolvimento de um kit didático para ser utilizado em aulas práticas de disciplinas como Arquitetura de Computadores, com foco principal na capacitação de estudantes no estudo da arquitetura de microprocessadores da plataforma x86 da Intel Corporation. Utilizando a placa FPGA DE2-115 da Altera, capaz de ser reconfigurada pelo software Quartus II, junto com um núcleo de arquitetura x86 implementado em linguagem de descrição de hardware, essa ferramenta pode auxiliar as aulas práticas nos laboratórios do IFAM – Campus Manaus Distrito Industrial (CMDI).

**Palavras-chave:** Arquitetura de Computadores, Microprocessador x86, FPGA, VHDL.

### 1 INTRODUÇÃO

Os circuitos digitais combinacionais e sequenciais sofrem rápida evolução, assim gerando uma crescente demanda de profissionais na área de desenvolvimento de circuitos integrados. Entretanto, constata-se que as instituições de ensino de nível superior não têm conseguido atender a essa demanda para desenvolver soluções com esta tecnologia (MENEZES; OSHIRO, 2009).

Atualmente os educadores optam pelos populares simuladores didáticos para apresentar conteúdos de arquitetura de computadores. Porém, embora estes simuladores contenham recursos, principalmente gráficos, que acelerem o aprendizado, eles não são extensíveis e não conseguem expor aos estudantes a variedade de possibilidades de projeto existentes atualmente. Esta realidade indica que um processo simplificado e rápido, para extensão ao ensino prático é altamente desejável, não apenas para projetistas de hardware trabalhando na indústria ou em pesquisa, mas para estudantes e educadores (SOARES, 2005).

Nas disciplinas de cursos de Engenharia do segmento de hardware, como por exemplo, Sistemas Digitais e Arquitetura de Computadores, pode-se aplicar um contínuo processo de integração, orientado pelo desenvolvimento de conteúdos de forma simultânea e complementar. A parte importante de levar em consideração na busca pela integração de disciplinas é a de como combinar conteúdos e sua organização ao longo do tempo. Outra questão importante é a



de como tornar viável a produção de atividades práticas que integrem conhecimentos significativos dessas disciplinas e que seja possível implementá-las. A resposta para essa questão pode vir através da tecnologia de dispositivos reconfiguráveis, pois esta permite a realização da implementação e dos testes de projetos de circuitos digitais e microprocessadores em tempos reduzidos (PILLA JÚNIOR et al. 2004). Assim, em ambas as disciplinas, é possível utilizar esses dispositivos reconfiguráveis no auxílio do ensino-aprendizagem.

Um dos dispositivos reconfiguráveis mais relevantes atualmente é o FPGA (Field Programmable Gate Arrays), surgido com o grande avanço ocorrido nas últimas décadas na área de circuitos eletrônicos programáveis. Esses dispositivos são programados através de linguagens de descrição de hardware (HDLs), como por exemplo, a VHDL (Very High Speed Hardware Description Language) e Verilog, que tornam possível o desenvolvimento de sistemas digitais com alta integração de forma rápida e com custos extremamente baixos, e permitem aos projetistas implementar circuitos e arquiteturas cada vez mais complexas (ORDONEZ et al., 2003).

De acordo com os autores Duarte e Garcia (2012), a realização de um material didático para aulas práticas no ensino Arquitetura de Computadores, com o foco em processadores, mostra o resultado de um aumento da motivação, aprendizado e participação dos alunos, evitando a evasão e o número de reprovações na disciplina, por parte destes. Dias (2013), recomenda um método de ensino prático de Arquitetura de Computadores, que utiliza dispositivos reconfiguráveis e os resultados apresentados contribuem para a formação de profissionais especializados no desenvolvimento de hardware.

A simulação de microprocessadores e seu conjunto de instruções é o foco principal deste trabalho voltado para o ensino de Arquitetura de Computadores. Procura-se investigar os núcleos x86 disponíveis em HDL para executar na placa FPGA Altera DE2-115, disponível no IFAM Campus Manaus Distrito Industrial (CMDI). Essa placa contém uma série de periféricos integrados para auxiliar a simulação de experimentos com a arquitetura x86 e possibilitar uma melhor compreensão do seu funcionamento. O objetivo deste trabalho é realizar o estudo da arquitetura x86, incluindo o seu conjunto de instruções, e implementar um ambiente de desenvolvimento e simulação com o auxílio de uma placa FPGA executando um núcleo x86 em HDL.

## 2 ENSINO DE ARQUITETURA DE COMPUTADORES

Segundo Oliveira et al. (2018), existem duas metodologias ativas para o ensino e aprendizado de arquitetura de computadores. A primeira, baseada em software, são os tradicionais simuladores didáticos de microprocessadores acadêmicos ou comerciais. A segunda, o uso de kits de hardware educacionais contendo microcontroladores, FPGAs, entre outros componentes eletrônicos, sendo estes adotados por muitos pesquisadores.

Nas aulas práticas de Arquitetura de Computadores no IFAM/CMDI faz-se uso do emu8086, um emulador do microprocessador 8086, para auxílio no ensino de programação da linguagem Assembly. Essa ferramenta permite a edição, montagem e execução de programas, como se fosse num microprocessador real. É possível acompanhar os valores dos registros, conteúdo da memória, pilha, variáveis e sinalizadores. Além disso, traz um pacote de dispositivos virtuais externos (robô, motor de passo, display de LED e cruzamento de semáforos) para programação de entrada e saída.

Nos subtópicos a seguir serão apresentados os recursos investigados para o desenvolvimento dos kits didáticos, iniciando com uma descrição sucinta sobre os tipos de

instruções da arquitetura da família x86, seguida da composição e definição de FPGA, e dos núcleos da família x86 construídos em HDL.

## 2.1 Arquitetura das instruções da família x86

A arquitetura x86 lançada em 1978, iniciando com o Intel 8086, ainda continua a dominar o mercado de processadores (ARBULU, 2020). Mesmo que a tecnologia de microprocessadores da família x86 tenha mudado rapidamente durante os anos, a arquitetura do conjunto de instruções permite compatibilidade entre versões anteriores, fator que foi um diferencial importante da Intel na época. Portanto qualquer programa escrito, mesmo sendo uma versão antiga da arquitetura x86, pode ser executado nas versões atuais.

Inicialmente a arquitetura x86 continha um conjunto de instruções CISC (Complex Instruction Set Computer), capaz de executar centenas de instruções complexas diferentes, permitindo funções variadas. Mas devido à concorrência por parte de outras empresas, a arquitetura x86 começou a utilizar instruções híbridas. Conforme Torres e Lima (apud LORENZONI, 2011, p.15), relata em seu trabalho:

“Os processadores da arquitetura x86, utilizam uma arquitetura híbrida CISC/RISC. A solução implantada hoje por todos os processadores CISC é o uso de um decodificador CISC/RISC, pois o processador trabalha internamente executando instruções RISC, mas aceita apenas instruções CISC que no processador são decodificadas e então processadas em formato RISC.”

Dessa forma, essa solução traz uma grande vantagem a essa arquitetura, pois o processador recebe instruções CISC, e as converte em instruções RISC (Reduced Instruction Set Computer), mais simples, todas executadas com a mesma quantidade e de tempo, aumentando o desempenho. Devido suportar muitas instruções, a arquitetura x86 é a mais utilizada pelos programadores, tornando-se a mais usada em computadores pessoais e servidores.

## 2.2 Kit de desenvolvimento FPGA

A sigla FPGA é a abreviação em inglês de Field Programmable Gate Array, que traduzindo temos algo como Matriz de Portas Programáveis em Campo. É um dispositivo que permite criar sistemas digitais de forma simples utilizando recursos lógicos para incorporar funções. Esses recursos lógicos são a composição básica dos FPGAs, podendo ser denominados de acordo com o fabricante, como a Xilinx que utiliza LUT (Look-Up Table) e a Altera com LE (Logic Element). Esse trabalho utiliza a denominação de elemento lógico (LE) da Altera. Portanto, um LE é composto por portas lógicas que implementam as operações lógicas. O FPGA incorpora uma certa quantidade de LEs, formando uma matriz de blocos de portas lógicas programáveis, e conectados por recursos de interligação também configuráveis.

A linguagem de descrição de hardware VHDL é mais utilizada na indústria e nas universidades para configurar o FPGA, permitindo a modelagem, simulação, verificação e síntese de processadores ou de qualquer outro sistema digital. A construção dos tipos de dados, comandos e a unidade de modularização, disponíveis nesta linguagem, possuem uma relação estreita com os mecanismos existentes no hardware.

A placa FPGA DE2-115 da Altera é amplamente utilizada em atividades de ensino, pesquisa e desenvolvimento de produtos. Para a sua configuração utiliza-se o software Quartus II Web Edition versão 13.0 SP1. Essa placa possui cabo de transferência denominado USB-Blaster embutido, dezoito LEDs vermelhos e nove verdes, oito displays de 7 segmentos, saída RS-232 para comunicação serial, quarenta pinos de conectores GPIO para adicionar periféricos





externos, entre outros. Esse kit de desenvolvimento apresenta todos os recursos necessários para a execução desse trabalho. De acordo com a Intel Corporation, este kit de desenvolvimento vem equipado com o chip FPGA Altera Cyclone IV E e possui 114.480 LEs.

### 2.3 Núcleos x86 construídos em HDL

Núcleo de propriedade intelectual ou Núcleo IP é uma unidade reutilizável de lógica, célula ou circuito integrado. São utilizados como blocos de construção dentro de aplicações específicas em projetos de circuitos integrados como FPGAs, podendo ter a mesma finalidade de uma biblioteca quando comparado a uma linguagem de programação. São disponibilizados por empresas especializadas e comunidades online para desenvolvedores, e licenciados na forma de GNU General Public License e BSD (Berkeley Software Distribution), que são licenças utilizadas por projetos de software livre e de código aberto.

Entre os núcleos da arquitetura x86 pesquisados, tem o CPU86 da empresa HT-Lab implementado em VHDL. No portal da OpenCores, uma comunidade de compartilhamento online gratuito, disponibiliza outros dois núcleos implementados em Verilog: Zet86 e ao486. Na "Tabela 1" são apresentadas as informações dos três núcleos investigados.

Tabela 1 - Núcleos em HDL.

Núcleos	CPU86	Zet86	ao486
<b>Descrição</b>	Compatível com um processador iAPX8088.	Equivalente a um Intel 80186.	Implementa todos os recursos de um 486 SX.
<b>Placas de FPGA</b>	- Enterpoint Drigmorn 1 - Papilio Duo	- Xilinx ML-403 - Altera DE0 - Altera DE1 - Altera DE2-115	- Altera DE2-115
<b>Linguagem HDL</b>	VHDL	Verilog	Verilog
<b>Disponível em:</b>	ht-lab.com	opencores.org	opencores.org

Fonte: Autor.

O núcleo CPU86 é compatível com a maioria dos softwares disponíveis para a arquitetura x86 e a maioria pode ser baixada gratuitamente. A descrição deste núcleo apresenta testes e simulações, com o Modelsim (Mentor Graphics) e o ISIM (Xilinx Inc), que foram realizados durante o seu desenvolvimento e comprovam sua compatibilidade com arquitetura x86. Incluso nesse projeto tem-se uma UART 16750 em VHDL disponibilizada pela OpenCores, com a finalidade de transmitir e receber dados de forma serial, que permite a comunicação direta do núcleo com um programa monitor. Além disso, apresenta ainda um banco de testes simples que conta com o núcleo do processador, 1 Megabytes de memória endereçável, sendo 256 bytes reservados para armazenar um programa simples que realiza um salto para o endereço inicial do monitor MON88, também disponibilizado por essa empresa, sendo esse o primeiro programa a ser executado, e nele encontra-se algumas opções para manipular a CPU86.

Entre os núcleos do portal OpenCores, o ao486 foi modelado e testado com base na implementação do software Bochs, sendo este um emulador da plataforma x86, que pode executar Sistemas Operacionais (SOs) compilados para esta plataforma, como se rodasse numa máquina real. O projeto ainda contém um SoC (System-on-Chip), que pode ser os componentes



de um PC ou qualquer sistema eletrônico integrado em um único chip, capaz de inicializar o kernel Linux versão 3.13 e o Microsoft Windows 95. Também estão inclusos controladores de teclado, mouse e adaptador VGA. Este portal também disponibiliza o núcleo Zet86, ainda em fase de desenvolvimento, mas leva uma grande vantagem por estar sendo desenvolvido em diferentes placas de FPGA. Também inclui suporte a uma UART, além de alguns periféricos, como controlador de teclado e adaptador VGA.

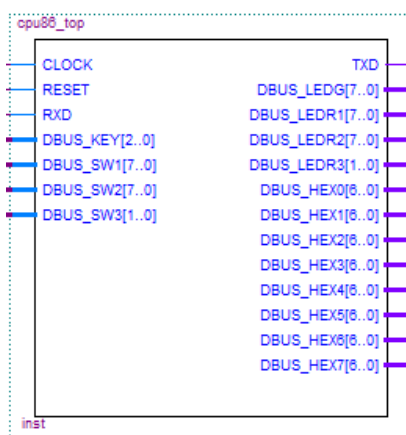
Das três opções analisadas, optou-se pela CPU86 da HT-Lab. Esse núcleo mostrou-se mais adequado considerando-se o objetivo deste trabalho, o de realizar o estudo de instruções em Assembly e executá-las na placa de desenvolvimento com apoio de seus periféricos. Pode-se ainda manipular o núcleo carregado no FPGA, utilizando-se um PC executando um programa terminal de comunicação serial, com diversas opções, assim tornando mais simples a utilização do conjunto PC/placa de FPGA. Os dois núcleos da OpenCores não oferecem um monitor de comunicação direta com o núcleo e apresentam foco na instalação e uso de SOs, e sendo um deles o zet86 ainda em um estado inicial de desenvolvimento.

### 3 IMPLEMENTAÇÃO DA CPU86 NA PLACA DE DESENVOLVIMENTO DE2-115

O primeiro passo é a implantação do núcleo x86 na placa FPGA. No software Quartus II, cria-se um projeto, explicado em detalhes por Carbonara (2016, p.32), adicionando-se os arquivos da CPU86, como detalhado pela HT-Lab. Nesse projeto, adicionamos os periféricos da placa de desenvolvimento, a partir de suas descrições de hardware. Em seguida são criadas as memórias utilizando-se a ferramenta MegaWizard Plug-In Manager, com capacidade total de 21Kbytes, com o objetivo de não desperdiçar muitos LEs. A memória foi organizada em dois bancos com capacidades individuais de 256 bytes e 20K bytes. Os bancos de memória podem armazenar arquivos com formato hexadecimal (extensão *hex*), que contêm código fonte ou sequências de instruções para serem executadas. Na primeira memória é armazenado o programa de boot do sistema, que realiza um salto para o endereço inicial do monitor MON88, o qual está armazenado na segunda memória devido ao seu tamanho de aproximadamente 15Kbytes, assim ainda sobrando nessa segunda memória 5Kbytes que fica disponível para carregar os programas a serem executados nas aulas práticas.

Todas as descrições de hardwares, como as memórias, a UART, os periféricos da placa e os arquivos que compõe a CPU86 precisam ser convertidos em um único código em VHDL, para definir suas conexões, as entradas/saídas e o endereçamento dos periféricos. Cada componente deve possuir um endereço para ser manipulado durante a programação em linguagem Assembly. Fez-se a adição dos seguintes periféricos neste trabalho: oito LEDs verdes, dezoito LEDs vermelhos, oito displays de sete segmentos, dezoito switches e quatro push-buttons. Devido ao fato desse código ser muito extenso, e para tornar as ligações de forma mais visual, o software oferece uma opção para representar o código VHDL em um bloco. A "Figura 1" mostra o bloco gerado. Para visualizar as ligações dos componentes de entrada e saída com as ligações físicas da placa, utiliza-se a ferramenta Pin Planner.

Figura 1 - Bloco do núcleo CPU86.



Fonte: *print screen* do bloco gerado a partir do código em VHDL.

Com a criação desse bloco, “Figura 1”, todas as entradas e saídas, descritas no código são representadas através da pinagem. As pinagens descritas com início DBUS são os periféricos da placa, do lado esquerdo temos todas as entradas e do direito todas as saídas. Os pinos específicos estão estabelecidos para o funcionamento e comunicação com o núcleo, da seguinte maneira o CLOCK determina a frequência de operação, o RESET restaura a configuração após o uso, estando esse definido como um push-button da placa e a pinagem RXD e TXD são composição da UART para comunicação serial. O próximo passo é a gravação do código na placa FPGA, através do USB-Blaster.

Para estabelecer a comunicação serial foi necessário a aquisição de um periférico externo, utilizou-se a placa FTDI232, um conversor USB-Serial, pois o disponível na placa de desenvolvimento o RS232 é um padrão que os PCs atuais não possuem. A ligação desse periférico externo foi feita através dos conectores GPIO, utilizando-se apenas três pinos, um para o TXD, outro para o RXD e um para o GND do conversor.

### 3.1 Execução e testes de programas x86

Após a gravação do código VHDL contendo o núcleo x86 e definição dos periféricos na placa FPGA, pode-se utilizar o kit para execução e testes de programas em linguagem Assembly x86. A “Figura 2” mostra as etapas a serem seguidas.

Figura 2 – Etapas para a execução e testes de programas x86.



Fonte: Autor.

Na primeira etapa deve-se utilizar o emu8086 para desenvolver um programa em linguagem Assembly x86. Na segunda etapa, é necessário converter o programa gerado na primeira etapa para o formato *hex*. Pode-se utilizar programas de conversão fornecidos pela HT-Lab, por exemplo. Na terceira etapa, utiliza-se o programa monitor MON88 da HT-Lab para carregar e executar o arquivo *hex* no núcleo. Após essa etapa, o programa é imediatamente

executado na placa FPGA. A placa deve estar conectada a um microcomputador PC através da interface serial para que as opções do monitor envidas pelo núcleo sejam visualizadas.

Alternativamente, é possível utilizar outro procedimento, caso a interface serial não esteja disponível. Nesse procedimento, o arquivo *hex* deve ser compilado juntamente com o núcleo x86 e gravado no FPGA. Isso deve ser feito com as ferramentas disponíveis do Quartus II. Esse procedimento demanda maior tempo devido à etapa de compilação. Finalmente, após a regravação do núcleo, o programa é executado.

Essa solução alternativa sem o uso do monitor, não é a mais adequada pelo fato de envolver muitas etapas, exigindo um maior tempo para ser realizada. O objetivo do projeto é tornar mais simples e mais rápido o procedimento de execução e teste de instruções pelos alunos que venham a utilizá-la.

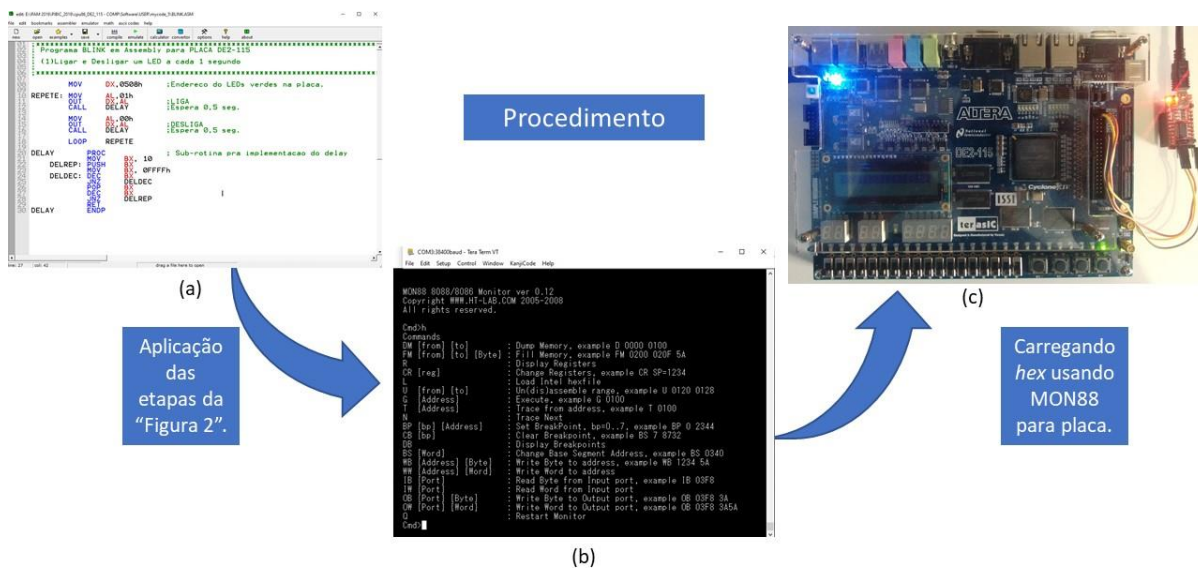
### *Um exemplo simples*

Para testar o kit de desenvolvimento elaboramos um pequeno programa em Assembly, mostrado na “Figura 3”(a). O programa, popularmente conhecido como *Blink*, utiliza um *loop* para ligar e desligar um LED a cada 1 segundo. Para ser possível visualizar na placa a transição de acionamento desse LED é necessária a criação de um *delay*, que é um atraso em um período de tempo, que depende da frequência de *clock* de operação e do número de ciclos de máquina consumidos por cada instrução. O exemplo utiliza uma sub-rotina que realiza dois *jumps* condicionais, com a instrução JNZ, de forma a implementar dois *loops* encadeados sendo o primeiro com 10 repetições, e o segundo, mais interno, com 64K repetições com o uso do registrador BX. Assim, obtemos algo próximo de 1 segundo, para alternar o estado do LED.

O acionamento dos LEDs é feito através do nível lógico 1. O primeiro LED verde é ligado ao bit 0 do barramento de dados. Portanto, para acendê-lo e apaga-lo, enviamos alternadamente os valores 01h e 00h em hexadecimal. No programa em Assembly feito no emu8086, “Figura 3” (a), é usado o endereço 0508h definido no código em VHDL como o endereço dos oito LEDs verdes. A “Figura 3 (b)” mostra a tela do monitor MON88, após o carregamento do programa, seguindo as etapas definidas pela “Figura 2”. O programa é executado na placa como mostra a “Figura 3” (c). O núcleo x86 e o programa exemplo em memória consumiram 5824 LEs do FPGA.



Figura 3 - Implementação do exemplo.



Fonte: Autor.

#### 4 CONSIDERAÇÕES FINAIS

Este trabalho apresentou um kit de desenvolvimento para o estudo prático do conjunto de instruções dos processadores x86 na disciplina de Arquitetura de Computadores do curso de Engenharia de Controle e Automação do IFAM/CMDI. A tecnologia de dispositivos reconfiguráveis FPGA, juntamente com um núcleo x86, permite que programas escritos em linguagem Assembly sejam carregados e executados na placa FPGA. Os periféricos embutidos na placa podem ser usados nos projetos dos alunos ao longo da disciplina. O núcleo x86 proposto é o CPU86 da HT-Lab, disponibilizado gratuitamente, e a placa FPGA adotada é a DE2-115 da Altera, disponível nos laboratórios do campus, o que viabiliza a implementação do kit.

Um exemplo simples foi implementado em que um pequeno programa em Assembly faz um LED piscar. Um programa monitor de comunicação, disponibilizado pelo núcleo CPU86, permite acompanhar a execução dos programas na placa, comunicando com um PC via interface serial. Uma alternativa mais lenta e trabalhosa exige que o programa seja compilado juntamente com o núcleo e carregado na placa. Essa opção é indicada se a comunicação serial, por algum motivo, esteja indisponível. O projeto completo na placa FPGA consumiu 5824 LEs. Este exemplo permitiu verificar a atuação do núcleo com um programa em Assembly na placa FPGA, concluindo-se assim que o projeto se comportou de acordo com a teoria de funcionamento proposta pela HT-Lab.

O conjunto que formará os kits de desenvolvimento propostos, encontra-se ainda em desenvolvimento, visando melhorar os procedimentos de uso, diminuindo as etapas e tornando-o mais intuitivo possível. Dando continuidade a este trabalho, planeja-se propor roteiros de aula que permitam o estudo de diferentes tipos de instruções do x86, a serem utilizados com o kit de desenvolvimento.



### *Agradecimentos*

Agradeço primeiramente a Deus, por permitir que tivesse determinação para não desanimar durante a realização deste trabalho, aos professores orientadores envolvidos no projeto, a FAPEAM com a bolsa de iniciação científica e ao IFAM/CMDI que disponibilizou os recursos necessários para desenvolvimento do projeto.

### REFERÊNCIAS

ALTERA CORPORATION. **DE2 Development and Education Board User Manual**. Disponível em: [https://www.intel.com/content/dam/altera-www/global/en\\_US/portal/dsn/42/doc-us-dsnbk-42-1404062209-de2-115-user-manual.pdf](https://www.intel.com/content/dam/altera-www/global/en_US/portal/dsn/42/doc-us-dsnbk-42-1404062209-de2-115-user-manual.pdf). Acesso em: 15 abr. 2020.

ARBULU, RAFAEL. **AMD tem 40% de fatia do mercado de processadores pela primeira vez em 14 anos**. Disponível em: <https://canaltech.com.br/negocios/amd-tem-40-de-fatia-do-mercado-de-processadores-pela-primeira-vez-em-14-anos-158583/>. Acesso em: 5 mai. 2020.

CARBONARA, L. P. **Estudo dos modos de configuração de FPGA**. 2016. 81 f. Tese (Graduação em Engenharia Elétrica) – Faculdade de Engenharia, Universidade Estadual Paulista, Guaratinguetá, 2016.

DIAS, M. Microcomputador re-configurável em FPGA para ensino de Arquitetura de Computadores na Ciência da Computação. **RENOTE - Revista Novas Tecnologias na Educação**, Porto Alegre, v. 11, n. 3, p. 1-10, 2013.

DUARTE, R.; GARCIA, P. Ensino Prático de Projeto de Processadores Segundo uma Metodologia de Ensino-Aprendizagem Baseada em Projetos na Escola de Engenharia da UFMG, **International Journal of Computer Architecture Education (IJCAE)**, vol. 1, n. 1, p. 11-20, 2012.

HT - LAB. CPU86 - Free 8088 FPGA IP Core. Disponível em: <http://www.ht-lab.com/cpu86.htm>. Acesso em: 15 abr. 2020.

LORENZONI, Ricardo Klein. **Análise de desempenho e consumo energético entre processadores ARM e x86**. 2011. 81 f. TCC (Bacharel em Informática - Sistemas de Informação) - Universidade Regional do Estado do Rio Grande do Sul, Ijuí, 2011.

MENEZES, D.; OSHIRO, E. **Desenvolvimento de Microcontroladores Utilizando FPGA e VHDL**. 2009. 175 f. TCC (Bacharelado em Ciência da Computação) - Centro Universitário Eurípides de Marília – UNIVEM, São Paulo, 2009.

OLIVEIRA, T.; STRINGHINI, D.; FAZENDA, Á.; CRAIBAS, J. Projetando Arquiteturas de Computadores numa Abordagem de Ensino Prática e Sistêmica. **International Journal of Computer Architecture Education (IJCAE)**, SP-São José dos Campos, v. 7, n. 1, p. 1-10, 2018.



OPENCORES. **Ao486**. Disponível em: <https://opencores.org/projects/ao486> . Acesso em: 15 abr. 2020.

OPENCORES. **Zet - The x86 (IA-32) open implementation**. Disponível em: <https://opencores.org/projects/zet86> . Acesso em: 15 abr. 2020.

ORDONEZ, Edward David Moreno *et al.* **Projeto, Desempenho e Aplicações de Sistemas Digitais em Circuitos Programáveis (FPGAs)**. Pompéia - SP: BLESS Gráfica e Editora Ltda. 2003.

PILLA JÚNIOR.V.; FERLIN, E. P.; CUNHA, J. C., O Ensino de Projetos Digitais: Fundamentação Teórica Práticas de Laboratório e a Tecnologia de Dispositivos Reconfiguráveis in WCETE- 2004 - World Congress on Engineering and Technology Education, Guarujá/Santos, SP, 2004.

QUARTUS II Web Edition. 13.0 SP1., 2013. Disponível em: <https://fpgasoftwre.intel.com/13.0sp1/?edition=web> . Acesso em: 15 abr. 2020.

SOARES, Sandro Neves. **T&D-Bench - Explorando o espaço de projeto de processadores em ensino e em pesquisa**. 2005. 168 f. Tese (Doutor em Ciência da Computação) - Universidade Federal do Rio Grande do Sul, Porto Alegre, 2005.

## **DEVELOPMENT OF A TEACHING KIT USING FPGA TO STUDY THE ARCHITECTURE OF MICROPROCESSORS X86**

**Abstract:** *The technology of reconfigurable digital integrated circuits, such as FPGA (Field Programmable Gate Array), made it possible to apply this device in projects with different configurations and complexities. The practical importance of this technology can be extended to the teaching of subjects in Engineering courses such as Digital Systems and Computer Architecture. This work describes the development of a didactic kit to be used in practical classes in the discipline of Computer Architecture, with a main focus on training students in the study of microprocessor architecture on the Intel Corporation's x86 platform. Using Altera's FPGA DE2-115 card, capable of being reconfigured by Quartus II software, together with an x86 architecture core implemented in hardware description language, this tool can assist practical classes in the IFAM labs - Campus Manaus Distrito Industrial (CMDI).*

**Keywords:** *Computer Architecture, Microprocessor x86, FPGA, VHDL*