

Microcomputador re-configurável em FPGA para ensino de Arquitetura de Computadores na Ciência da Computação

Maurício Araújo Dias (Unesp - FCT), madias@fct.unesp.br

Resumo: O ensino prático de Arquitetura de Computadores estimula o desenvolvimento tecnológico de *hardware* no Brasil. Por isso, este artigo propõe um método para o ensino prático de Arquitetura de Computadores na Ciência da Computação. Esse método usa uma arquitetura de microcomputador re-configurável em FPGA, para o ensino dessa disciplina na Computação. Pelo método, microcomputadores são desenvolvidos a partir de uma arquitetura de microcomputador básica de dois bits descrita em esquema elétrico e em VHDL. Os resultados apresentados indicam que o método proposto antecipa o ensino prático de Arquitetura de Computadores para o início das atividades da disciplina e contribui formando recursos humanos especializados no desenvolvimento tecnológico de *hardware* no País.

Palavras-chave: microcomputador, arquitetura, re-configurável, FPGA, ensino.

Re-configurable Microcomputer on FPGA for Teaching of Computer Architecture in Computer Science

Abstract: The practical teaching of Computer Architecture stimulates the technological development of hardware in Brazil. Therefore, the main objective of this paper is to propose a method to aid teachers in practical classes of Computer Architecture for Computer Science. This method uses a re-configurable microcomputer architecture on FPGA, to aid the teaching of this discipline in Computer Science. Based on this method, microcomputers are developed from a basic microcomputer architecture of two bits, which is described in electrical schematic and VHDL. The results indicate that the proposed method anticipates the practical teaching of Computer Architecture to the early stages of the discipline and helps forming human resources specialized to the development of hardware technology to the country.

Keywords: microcomputer, architecture, re-configurable, FPGA, teaching.

1. Introdução

No Brasil, o ensino de Arquitetura de Computadores em cursos de Ciência da Computação costuma ser teórico e a produção de *hardware* é inexpressiva. Contrariamente a essas tendências, o ensino prático de programação na Ciência da Computação contribui para que o Brasil seja um dos maiores produtores mundiais de *software*. No entanto, se a presente situação da Ciência da Computação no País ainda permanece marcada pelo forte desenvolvimento de *software*, o seu futuro abrangerá também um intenso desenvolvimento de *hardware*, pois as tecnologias acessíveis no mercado propiciarão o desenvolvimento de recursos computacionais implementados fisicamente de forma mais barata, rápida e fácil e, principalmente, flexível.

Há alguns anos, os recursos computacionais implementados em *software*, geralmente, custavam caro e tinham códigos fechados. Atualmente, os programas são

mais baratos, quando não são gratuitos, e muitos são obtidos diretamente pela Internet. Adicionalmente, muitos desses programas são oferecidos com código irrestrito, o que permite ao cientista da computação atualizar e otimizar seus códigos, sem ou com poucas limitações. Quando isso acontece, qualquer tipo de *software*, por exemplo editores de texto, planilhas eletrônicas, ambientes de desenvolvimento de projetos e até mesmo sistemas operacionais, dentre outros, pode ser re-configurado pelo cientista da computação, segundo a sua vontade. Esses códigos irrestritos e baratos incrementam o ensino prático de programação na Ciência da Computação, além de incentivarem alunos a estudarem e pesquisarem conceitos relacionados a *software*.

Por outro lado, a situação do *hardware* continua inalterada há anos: implementações de alto custo e documentações com acesso limitado. Logo, as aulas da Ciência da Computação relacionadas com *hardware* são ministradas, quase sempre, de maneira teórica, ficando a prática limitada à Engenharia da Computação. O motivo para isso é o elevado custo e a elevada complexidade para o desenvolvimento de circuitos integrados, empregados em arquiteturas de computadores, quando métodos tradicionais são utilizados (Iqbal et al., 2010).

A abordagem unicamente teórica em disciplinas relacionadas a *hardware* possibilita aos estudantes da Ciência da Computação obter somente uma noção geral e superficial dos conceitos apresentados nessa área. Neste caso, a interação entre o estudante e o *hardware* não acontece, porque todo conceito aprendido vem de aulas teóricas. Para que o estudante da Computação aprenda aprofundadamente os temas ensinados nessa área e se torne apto a criar produtos tecnológicos otimizados, é preciso que ele amplie o seu contato com distintas implementações em *hardware*, o que não parecia possível até então.

Contudo, assim como tradicionalmente ocorreu com o *software*, a situação do *hardware* está se alterando. A tendência já no curto prazo é o surgimento de arquiteturas de computadores mais baratas no mercado ou até mesmo gratuitas. As descrições de comportamento e as documentações dessas arquiteturas serão disponibilizadas para serem adquiridas pela Internet a baixo custo ou gratuitamente. Isso quer dizer que qualquer um poderá fazer *download* de *hardware* virtual, otimizado e atualizado graças a uma tecnologia atual que suporta re-configurações – a plataforma FPGA (*Field Programmable Gate Array*) (Patterson, 1990). Em outras palavras, qualquer indivíduo, em qualquer lugar do planeta, poderá obter, por exemplo, uma nova arquitetura de computador barata pela Internet. Cientistas da computação otimizarão facilmente arquiteturas de computadores conforme as suas vontades. Além disso, eles disponibilizarão essas arquiteturas para o mercado tecnológico e para a comunidade científica por *upload* na Internet. Assim, esse *hardware* re-configurável contribui incentivando o ensino prático de Arquitetura de Computadores na Computação, além de estimular alunos a estudarem e pesquisarem temas relacionados a *hardware*.

O emprego de *FPGAs* possibilita, aos alunos da Ciência da Computação, obter significativo nível de conhecimento e, como consequência, desenvolver produtos tecnológicos otimizados ao trabalhar com assuntos relacionados a circuitos digitais, micro-controladores, microprocessadores e arquiteturas de computadores, todos com baixa complexidade de projeto e baixo custo. Isso porque a implantação em *FPGAs* permite que o aluno da computação re-configure, sintetize e simule seus trabalhos em uma plataforma física real, sempre que for necessário, para otimizar os circuitos em desenvolvimento. Uma análise comparativa entre a abordagem re-configurável e os métodos tradicionais usados para o desenvolvimento de circuitos digitais é apresentada em (Iqbal et al., 2010).

Usando FPGA como plataforma para desenvolvimento de circuitos digitais, o ensino prático de Arquitetura de Computadores em cursos de Ciência da Computação contribuirá para que o Brasil alcance uma produção de *hardware* mais expressiva. Permitirá também que os alunos brasileiros se qualifiquem hoje a ponto de, no futuro, quando se tornarem cientistas da computação, serem capazes de desenvolver tecnologias avançadas baseadas em FPGAs equivalentes às que são propostas por pesquisadores estrangeiros, como as que são citadas a seguir.

L. Yan et al. (2010) propõem uma arquitetura de processador re-configurável, a qual combina múltiplos núcleos com unidades de processamento re-configuráveis. R. Kielbik et al. (2010) propõem uma arquitetura de processador com conjunto reduzido de instruções, que usa lógica dinamicamente re-configurável e executa todas as instruções em paralelo. K. A. Fawaz et al. (2010) apresentam um processador assíncrono de alto desempenho dinamicamente re-configurável, que pode ser programado e que consome pouca energia. P. Patel and M. Moallem (2010) discutem o desenvolvimento de controladores embarcados sobre um sistema multiprocessador re-configurável usando FPGA. D. Göhringer et al. (2010) descrevem uma metodologia de projeto para o desenvolvimento de arquiteturas de sistemas multiprocessadores re-configuráveis, a qual ajuda a esconder a complexidade do hardware e garante alternativas de projeto ao desenvolvedor. D. Göhringer e J. Becker (2010) apresentam uma nova abordagem para a computação baseada em multiprocessadores re-configuráveis de alto desempenho sobre FPGAs.

Capacitar alunos da Computação para o desenvolvimento de *hardware* trará benefícios também para outras áreas das Ciências no País. Por exemplo, um trabalho descreve uma arquitetura re-programável desenvolvida para executar algoritmos da bioinformática, tais como comparações entre cadeias de DNA, filogenética, dentre outros (Dollas, 2010). Outro trabalho comenta sobre o impacto positivo do uso da computação re-configurável em futuras missões da NASA (Pingree, 2010). Da mesma forma, o uso da computação re-configurável pode beneficiar o programa aeroespacial brasileiro, o qual, recentemente, vem promovendo e incentivando a formação de recursos humanos estratégicos e especializados para esse setor.

Devido à tendência mundial de uso de FPGAs como plataformas de desenvolvimento para projetos de *hardware* e à importância desses projetos para desenvolver tecnologicamente um país, o ensino prático de Arquitetura de Computadores na Ciência da Computação mostra-se cada vez mais necessário no Brasil.

Alguns esforços para tornar o ensino de Arquitetura de Computadores mais prático usando FPGAs podem ser encontrados na literatura científica. Por exemplo, um artigo apresenta um ambiente integrado, o qual é constituído por um *hardware* re-configurável e uma interface gráfica executada a partir de um computador pessoal (Bulic et al., 2010). Esse ambiente permite aos alunos da Computação acompanhar, passo a passo pela interface gráfica, a execução dos seus próprios programas em um microprocessador de 32 bits desenvolvido para FPGAs.

Outra proposta descreve o projeto de uma arquitetura de microcomputador desenvolvida para FPGAs (Oztekin et al., 2011). Essa arquitetura é montada pelos próprios alunos, a partir de circuitos digitais previamente desenvolvidos até compor um microcomputador completo.

Essas duas propostas contribuem significativamente para incrementar o ensino prático de Arquitetura de Computadores. No entanto, como ambas as propostas usam recursos considerados complexos por estudantes iniciantes em Arquitetura de

Computadores, os alunos precisam previamente conhecer o comportamento de cada um desses recursos de forma teórica, antes de saberem como uma Arquitetura de Computador funciona na prática enquanto executa um programa qualquer. Sendo assim, nenhuma dessas propostas apresenta um método de ensino que antecipa o ensino prático de Arquitetura de Computadores para o início das atividades da disciplina.

Diante dessa realidade, este artigo propõe um método para o ensino prático de Arquitetura de Computadores na Ciência da Computação. Esse método se baseia no uso de uma arquitetura de microcomputador re-configurável desenvolvida sobre plataforma FPGA, para o ensino dessa disciplina em cursos de Ciência da Computação. Ele permite antecipar o ensino prático de Arquitetura de Computadores para o início das atividades da disciplina e contribui para a formação de recursos humanos especializados para o desenvolvimento tecnológico de *hardware* no País.

Para melhor elucidar a proposta apresentada neste trabalho, este artigo está organizado como descrito a seguir. A seção II descreve materiais e métodos. Os resultados alcançados com base na aplicação do método proposto são apresentados e discutidos na seção III. A seção IV apresenta conclusões sobre este trabalho.

2. Materiais e Método

A seção Materiais e Métodos encontra-se dividida em duas subseções. A primeira delas, Materiais, descreve os recursos computacionais usados como apoio para o desenvolvimento, implementação e aplicação deste método. A segunda, Método, descreve detalhadamente o método para o ensino prático de Arquitetura de Computadores na Ciência da Computação proposto por este artigo. Essas subseções são apresentadas a seguir.

2.1 Materiais

Durante a aplicação do método proposto por este artigo, os alunos usaram uma ferramenta implementada em *software*, o Quartus II, disponibilizada gratuitamente pela Altera (Altera, 2013), que permite desenvolver e re-configurar circuitos digitais. Esse *software* emula um ambiente de desenvolvimento de *hardware*, possibilitando que o aluno se aproxime mais daquilo que é a implementação física de uma ou mais soluções computacionais. Tal ferramenta também permite usar um simulador virtual para validar o funcionamento dos circuitos desenvolvidos. Contudo, um aluno da Computação somente alcançará um nível de conhecimento com profundidade ideal para dominar o *hardware* e, propor e implementar otimizações em circuitos, se ele mesmo desenvolver diferentes projetos de *hardware* por um longo período. Isso significa que para aprender Arquitetura de Computadores na prática, além de desenvolver e analisar o *hardware*, o estudante de Computação deve configurá-lo, sintetizá-lo e simulá-lo em sua forma física, quantas vezes forem necessárias, para refinar a sua otimização.

As condições para a implementação desse *hardware* re-configurável são oportunizadas pela FPGA (*Field Programmable Gate Array*) (Patterson, 1990). FPGA é um chip que funciona como plataforma para desenvolvimento de circuitos. Essa plataforma é formada por um arranjo eletrônico, o qual lhe habilita a trabalhar como qualquer circuito digital. Em outras palavras, é possível “programar” uma FPGA pela descrição comportamental de um *hardware* a fim de que ela se comporte como qualquer circuito digital básico, micro-controlador, microprocessador ou mesmo como uma arquitetura de computador mais elaborada. Essa “programação” se faz com base na descrição do comportamento do *hardware*, usando uma linguagem de descrição, por exemplo, VHDL – *VLSI Hardware Description Language* (Camposano et al., 1991) ou

em esquema elétrico. Posteriormente, para os dois casos, a descrição tem que ser sintetizada e descarregada na *FPGA*. Pelo menos em teoria, uma *FPGA* pode ser “reprogramada” ilimitadamente. Isso a qualifica como uma plataforma ideal para suportar otimizações e atualizações de *hardware*, pela sua re-configuração. Essa capacidade da *FPGA* advém do fato dela possuir elementos lógicos, os quais assumem o comportamento de portas lógicas (blocos básicos elementares empregados na construção de todo circuito digital, até mesmo microprocessadores). Dessa forma, circuitos digitais básicos, micro-controladores, microprocessadores e até arquiteturas de computadores são implementados fisicamente de maneira barata, rápida e fácil, e, principalmente, flexível, graças às *FPGAs*. Para o método proposto neste artigo a *FPGA* Cyclone III EP3C16F484C6 foi usada, porque ela faz parte do kit de desenvolvimento de projetos DE0 da Altera, disponibilizado para os alunos em sala de aula (Altera, 2013).

2.2 Método

Quando métodos tradicionais são aplicados ao ensino de Arquitetura de Computadores, os alunos aprendem um conteúdo teórico extenso na disciplina, para só depois implementarem na prática os conceitos aprendidos na teoria. A idéia central do método proposto por este artigo é ensinar, passo a passo, os conceitos de Arquitetura de Computadores aos alunos, à medida que eles desenvolvem e re-configuram um microcomputador. Para tanto, inicialmente, os alunos desenvolvem em aula aquilo que pode ser considerada a lógica mais reduzida para que um circuito digital ainda possa ser chamado de microcomputador de propósito geral. Em seguida, os alunos aprendem em detalhes como esse microcomputador funciona enquanto executa um programa básico. Posteriormente, cada novo conceito presente no ensino de Arquitetura de Computadores é passado de forma prática aos alunos, agregando novos elementos e novas funcionalidades ao microcomputador desenvolvido por eles mesmos, ou seja, re-configurando a arquitetura de computador implementada na *FPGA*. Em conjunto com os alunos, o funcionamento do microcomputador deve ser revisado após agregar cada novo elemento, pois cada re-configuração da arquitetura pode trazer alterações em aspectos específicos do comportamento do microcomputador.

Sendo assim, inicialmente, os alunos têm contato com conceitos elementares, suficientes apenas para o desenvolvimento de uma arquitetura de microcomputador básica de dois bits, como a arquitetura mostrada na Figura 1. Entretanto, o primeiro microcomputador desenvolvido pelos alunos já possui a essência do comportamento de grande parte dos computadores usados ou pesquisados atualmente no mundo, porém com a sua complexidade intencionalmente reduzida ao mínimo, para que possa ser rapidamente desenvolvido e facilmente compreendido. Dessa forma, este método inova ao promover o ensino prático de Arquitetura de Computadores a partir da criação de um “embrião tecnológico”, ou seja, uma unidade básica a partir da qual qualquer computador pode “nascer” (ser desenvolvido). Em outras palavras, esse microcomputador serve de ponto de partida, ou seja, de modelo base para a compreensão e o desenvolvimento de outros microcomputadores para o ensino de graduação, a pesquisa científica e para o mercado.

A arquitetura de microcomputador “embrionária” é constituída por uma unidade de controle (*CU*) *hardwired*, que atua conjuntamente com um registrador de instrução (*IR*) e com um registrador de micro-instrução (*MIR*) para controlar as demais unidades funcionais da mesma arquitetura. O microcomputador também possui um contador de instruções (*IC*) e um registrador de endereço da memória (*MAR*), os quais, em conjunto,

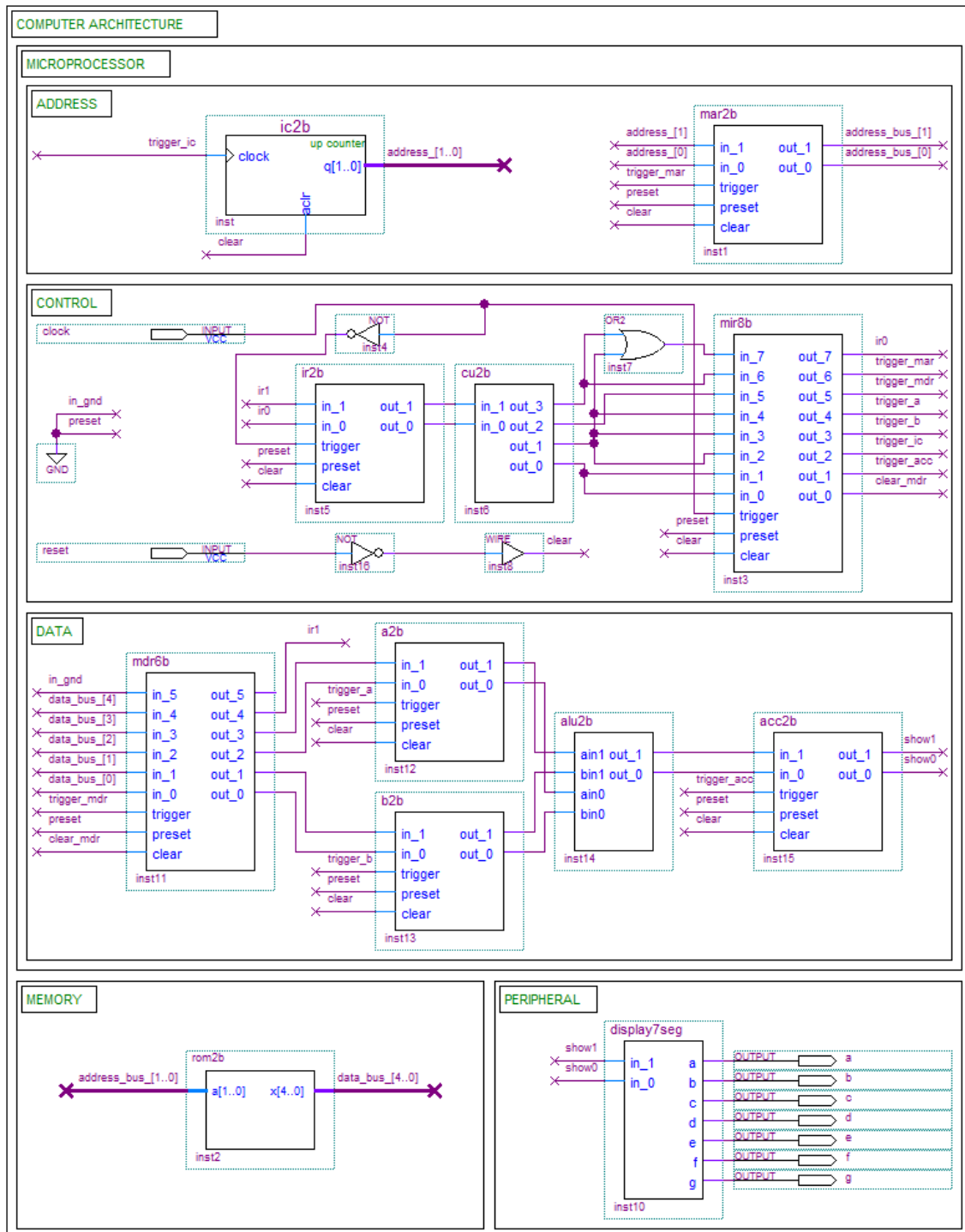


Figura 1 - Arquitetura de microcomputador básica de dois bits (“embrião tecnológico”).

permitem referenciar cada uma das quatro palavras de memória (de cinco bits cada uma) presentes em uma memória ROM. Um barramento de endereços de dois bits de largura une o registrador de endereço da memória (MAR) à memória ROM. O microcomputador também conta com um registrador de dados da memória (MDR), usado para receber as informações originárias da ROM, por um barramento de dados de dois bits de largura. Há, também, dois registradores auxiliares (A e B) e uma unidade aritmética e lógica (ALU). A ALU, é implementada como um circuito somador básico.

Continuando, um acumulador (*ACC*) é usado para armazenar os resultados dessas operações. O resultado armazenado pelo *ACC* é visto em um *display* de sete segmentos, graças a um decodificador para *display* de sete segmentos (*dec7seg*) existente na arquitetura. Encontram-se também presentes: uma entrada que recebe um sinal *power* ou *reset*, imprescindível para iniciar (ou reiniciar) a atividade do microcomputador; uma entrada que recebe o sinal de *clock*, usado para sincronizar os eventos que acontecem dentro da arquitetura.

O microcomputador pode ser colocado para trabalhar com uma frequência experimental de *clock*, por exemplo, igual a 10MHz, apesar dessa não ser a frequência máxima em que o microcomputador pode trabalhar. De qualquer forma, para o método, importa mostrar que o *clock* oscila, não a sua frequência de oscilação, porque para os alunos é mais importante conhecer a importância do *clock* para o funcionamento do microcomputador do que quão rápido ele é.

Apesar da sua simplicidade, o microcomputador trabalha de acordo com os princípios básicos de funcionamento de qualquer arquitetura de computador vendida no mercado atualmente. Em outras palavras, o seu ciclo de instrução é igual ao de qualquer outra máquina, isto é, ele é composto por um ciclo de busca e um ciclo de execução. Embora seja uma arquitetura de propósito geral, ela inicialmente está configurada para realizar apenas processamento matemático. Dessa forma, na *ROM* do microcomputador, está armazenado um programa para efetuar somas.

Para esse programa, todas as instruções têm o mesmo formato. Cada instrução do programa é composta por cinco bits. O bit mais significativo está posicionado mais à esquerda e representa o código da operação, cujo valor “1” identifica uma operação aritmética de soma. Os dois bits do meio representam o primeiro operando dessa soma. Os dois bits menos significativos representam o segundo operando da mesma soma. Assim, quando se deseja programar o microcomputador para somar dois números decimais (1 + 2, por exemplo), é necessário “escrever” instruções (como esta: 1 01 10) na sua memória *ROM*. O programa executado pelo microcomputador é composto por uma sequência de quatro instruções similares a essa. Para ensinar aos alunos como um microcomputador funciona enquanto executa um programa qualquer, basta mostrar como ocorre um ciclo de instrução no microcomputador, detalhando os passos do ciclo de busca e do ciclo de execução.

Como o microcomputador é implementado em uma plataforma re-programável, a *FPGA*, esta possibilita re-configurar a arquitetura constantemente. Assim, por re-configuração, a arquitetura pode ser modificada ilimitadamente, sem perder o seu objetivo principal, ou seja, ser um microcomputador. Isso é feito de forma prática, rápida e barata, quando comparado aos métodos tradicionais (Iqbal et al., 2010) (a única restrição é do tamanho da *FPGA* escolhida para a implementação). Desse modo, os alunos expandem a arquitetura de forma prática, à medida que aprendem novos conceitos de Arquitetura de Computadores. Assim eles partem de uma arquitetura “embrionária” em direção a arquiteturas mais complexas, aprendendo conceitos passo a passo.

3. Resultados e Discussão

A fim de verificar se o método proposto por este artigo auxilia no ensino prático de Arquitetura de Computadores para a Ciência da Computação, ele foi aplicado a diferentes turmas de um curso de Computação para obtenção de informações relacionadas a diferentes características do método. Para tanto, foram usados os

seguintes critérios: aulas práticas usando o microcomputador apresentado neste artigo; desenvolvimento de diferentes arquiteturas a partir desse mesmo microcomputador; provas teóricas; entrevistas envolvendo questões de percepção sobre as abordagens de ensino usadas na disciplina.

Três turmas diferentes de um curso de Bacharelado em Ciência da Computação tiveram suas aulas de Arquitetura de Computadores ministradas de forma a intercalar duas abordagens diferentes de ensino: algumas aulas seguiram o método para o ensino prático de Arquitetura de Computadores posposto por este artigo, enquanto outras aulas foram plenamente teóricas. Apesar dos alunos dessas turmas apresentarem diferentes níveis de dificuldade para a realização das atividades propostas na sala de aula, todos os alunos conseguiram realizar todas as atividades exigidas.

Nas aulas práticas, durante o período de um ano, foram desenvolvidas seis arquiteturas de microcomputadores distintas, a partir do microcomputador apresentado neste artigo. Se além das aulas, também fossem considerados os trabalhos de conclusão de curso e de iniciação científica, esse número subiria para treze arquiteturas diferentes de microcomputadores completamente desenvolvidas.

Em seguida, os alunos foram submetidos a provas teóricas de conhecimentos específicos sobre Arquitetura de Computadores. Essas provas objetivaram verificar a qualificação dos alunos para achar soluções para problemas do cotidiano da Computação relacionados com os conceitos estudados na disciplina Arquitetura de Computadores. As provas foram elaboradas de modo a contemplar ambas as abordagens usadas em sala de aula. Além disso, os alunos passaram por uma entrevista, na qual eles responderam a questões de percepção sobre as abordagens de ensino usadas na disciplina Arquitetura de Computadores. O objetivo dessa entrevista era saber se eles aprovavam ou não o método prático proposto e se eles consideravam ter aprendido mais com as aulas teóricas ou com o método de ensino prático. Essas provas e entrevistas foram realizadas sistematicamente durante três anos consecutivos, com o intuito de obter maior volume de dados para análise dos resultados.

Os resultados obtidos a partir da aplicação das provas de conhecimento específico indicam que os alunos tiveram mais facilidade em encontrar soluções para os problemas da Computação que envolviam conceitos estudados de forma prática do que para aqueles cujos conceitos foram estudados apenas de forma teórica.

Quanto aos resultados obtidos a partir da realização das entrevistas, estes indicam que o índice de aprovação do método prático proposto foi superior a 98%. No entanto, os quase 2% de alunos que afirmaram não apreciar o método proposto obtiveram resultados nas provas de conhecimento específico que indicam que eles também tiveram mais facilidade em aprender os conceitos de Arquitetura de Computadores com o método prático proposto do que com as aulas teóricas. Esses alunos alegaram que não apreciaram o método prático, porque essa abordagem exigiu que eles trabalhassem mais do que a abordagem teórica.

Os alunos demonstraram maior facilidade em aprender com a abordagem prática, porque eles se sentem mais estimulados a estudar quando são agentes ativos no seu processo de aprendizagem, em comparação a quando são agentes passivos em uma aula plenamente teórica. Além disso, o uso de tecnologias associadas ao ensino permite obter mais atenção dos alunos para os conceitos ensinados do que nas aulas teóricas.

Outro resultado importante é que após apenas oito horas-aula de teoria, os alunos já se mostraram aptos a iniciarem as atividades práticas de ensino. Isso mostra que o

método proposto neste artigo promoveu a antecipação do ensino prático de Arquitetura de Computadores para o início das atividades da disciplina. Ensinar conceitos sobre Arquitetura de Computadores aos alunos a partir de uma arquitetura de computador elementar favorece a antecipação do ensino prático na disciplina em comparação ao uso de arquiteturas mais elaboradas, como aquelas descritas em (Bulic et al., 2013) e (Oztekin et al., 2011). Essa antecipação permite aos alunos usufruírem das vantagens oferecidas pela abordagem prática por mais tempo. Além disso, os alunos mantêm contato prolongado com tecnologias de última geração, como as FPGAs. A propósito, a FPGA usada para apoiar a aplicação do método proposto por este artigo foi a Cyclone III EP3C16F484C6, porque ela faz parte do kit de desenvolvimento de projetos DE0 da Altera, disponibilizado para os alunos em sala de aula (Altera, 2013).

4. Conclusões

Este artigo propôs um método para o ensino prático de Arquitetura de Computadores na Ciência da Computação. Esse método se baseia no uso de uma arquitetura de microcomputador re-configurável desenvolvida sobre plataforma FPGA, para o ensino dessa disciplina em cursos de Ciência da Computação.

Os resultados alcançados com a aplicação desse método a alunos de Arquitetura de Computadores de um curso de Ciência da Computação mostraram que o método permite antecipar o ensino prático de Arquitetura de Computadores para o início das atividades da disciplina, conforme esperado. Além disso, o método contribui para a formação de recursos humanos especializados para o incremento tecnológico de *hardware* no País, já que promove o desenvolvimento prático de projetos de arquiteturas de computadores e outros circuitos digitais. Outra característica é que a arquitetura de computador usada no método pode ser adquirida pela internet, minimizando ainda mais o custo das arquiteturas desenvolvidas (basta que o aluno use um dispositivo dotado de FPGA). Isso representa algo como fazer o *download* de um *hardware* pela internet, o que pode ser útil para cursos à distância, os quais possuem aulas não presenciais.

Para trabalhos futuros, este artigo sugere a criação e a aplicação de métodos práticos para outras disciplinas ministradas em cursos de Ciência da Computação, por exemplo: lógica, física, compiladores, etc.

Referências Bibliográficas

- ALTERA CORPORATION. **Quartus II, programmable logic development system & software**. Altera Corporation, ver. 1.01, 1999. Data sheet. Disponível em: ftp://ftp.altera.com/pub/lit_req/document/ds/quartus.pdf. Acesso em: 09 out. 2013.
- BULIC, P.; GUSTIN, V.; SONC, D.; STRANCAR, A. An FPGA-based integrated environment for computer architecture. **Computer Applications In Engineering Education**, v.21, n.1, p.26–35, 2013. DOI: [10.1002/cae.20448](https://doi.org/10.1002/cae.20448).
- CAMPOSANO, R.; SAUNDERS, L. F.; TABET, R. M. VHDL as Input for High-Level Synthesis. **IEEE Design & Test of Computers**, v.8, n.1, p.43–49, 1991. DOI: [10.1109/54.75662](https://doi.org/10.1109/54.75662).
- DOLLAS, A. Reconfigurable architectures for bioinformatics applications. In: **2010 IEEE COMPUTER SOCIETY ANNUAL SYMPOSIUM ON VLSI (ISVLSI)**, 2010, p.6-7. DOI: [10.1109/ISVLSI.2010.111](https://doi.org/10.1109/ISVLSI.2010.111).

- FAWAZ, K. A.; ARSLAN, T.; KHAWAM, S.; MUIR, M.; NOUSIAS, I.; LINDSAY, I.; ERDOGAN, A. A dynamically reconfigurable asynchronous processor. In: **2010 IEEE 8th SYMPOSIUM ON APPLICATION SPECIFIC PROCESSORS (SASP)**, 2010, p.93-96. DOI: [10.1109/SASP.2010.5521141](https://doi.org/10.1109/SASP.2010.5521141).
- GÖHRINGER, D.; BECKER, J. High performance reconfigurable multi-processor-based computing on FPGAs, In: **2010 IEEE INTERNATIONAL SYMPOSIUM ON PARALLEL & DISTRIBUTED PROCESSING, WORKSHOPS AND PHD FORUM (IPDPSW)**, Georgia Institute of Technology, 2010, p.1-4. DOI: [10.1109/IPDPSW.2010.5470800](https://doi.org/10.1109/IPDPSW.2010.5470800).
- GÖHRINGER, D.; HUBNER, M.; BENZ, M.; BECKER, J. A design methodology for application partitioning and architecture development of reconfigurable multiprocessor systems-on-chip. In: **2010 18th IEEE ANNUAL INTERNATIONAL SYMPOSIUM ON FIELD-PROGRAMMABLE CUSTOM COMPUTING MACHINES (FCCM)**, Hilton Charlotte University Place, 2010, p.259-262. DOI: [10.1109/FCCM.2010.47](https://doi.org/10.1109/FCCM.2010.47).
- IQBAL, M. A.; AWAN, U. S.; KHAN, S. A. Reconfigurable computing technology used for modern scientific applications. In: **2010 2nd INTERNATIONAL CONFERENCE ON EDUCATION TECHNOLOGY AND COMPUTER (ICETC 2010)**, 5. 2010. p.V5-36 - V5-41. DOI: [10.1109/ICETC.2010.5529949](https://doi.org/10.1109/ICETC.2010.5529949).
- KIELBIK, R.; JABLONSKI, G.; SWIERCZ, B.; AMROZIK, P. Instructionless processor architecture using dynamically reconfigurable logic. In: **17th INTERNATIONAL CONFERENCE ON MIXED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS (MIXDES)**, 2010, p. 112-116.
- MONTEIRO, M. A. **Introdução à organização de computadores**. Rio de Janeiro: LTC, 2007.
- OZTEKIN, H.; TEMURTAS, F.; GULBAG, A. BZK.SAU.FPGA10.1: A modular approach to FPGA-based micro computer architecture design for educational purpose. **Computer Applications In Engineering Education**, 2011. DOI: [10.1002/cae.20553](https://doi.org/10.1002/cae.20553).
- PATEL, P.; MOALLEM, M. Reconfigurable system for real-time embedded control applications. **IET Control Theory & Applications**, v.4 , n.11, p. 2506–2515, 2010. DOI: [10.1049/iet-cta.2009.0471](https://doi.org/10.1049/iet-cta.2009.0471).
- PATTERSON, W. Field programmable gate arrays get enough speed and density for computer applications. Intellectual Leverage. Digest of Papers. In: **Thirty-Fifth IEEE Computer Society International Conference (Compcon Spring '90)**, 1990, p.477 – 480. DOI: [10.1109/CMPCON.1990.63726](https://doi.org/10.1109/CMPCON.1990.63726).
- PINGREE, P. J. Advancing NASA's on-board processing capabilities with reconfigurable FPGA technologies: opportunities & implications. Keynote Talk. In: **2010 IEEE INTERNATIONAL SYMPOSIUM ON PARALLEL & DISTRIBUTED PROCESSING, WORKSHOPS AND PHD FORUM (IPDPSW)**, 2010, p.1. DOI: [10.1109/IPDPSW.2010.5470824](https://doi.org/10.1109/IPDPSW.2010.5470824).
- YAN, L.; WU, B.; WEN, Y.; ZHANG, S.; CHEN, T. A reconfigurable processor architecture combining multi-core and reconfigurable processing unit. In: **2010 IEEE 10th INTERNATIONAL CONFERENCE ON COMPUTER AND INFORMATION TECHNOLOGY (CIT)**, 2010, p. 2897 - 2902. DOI: [10.1109/CIT.2010.484](https://doi.org/10.1109/CIT.2010.484).