

Projetando Arquiteturas de Computadores numa Abordagem de Ensino Prática e Sistêmica

Tiago de Oliveira, Denise Stringhini, Álvaro Luiz Fazenda, José Jailson Santos Craibas

Instituto de Ciência e Tecnologia
Universidade Federal de São Paulo - Unifesp
São José dos Campos, Brasil

tiago.oliveira@unifesp.br, dstringhini@unifesp.br, alvaro.fazenda@unifesp.br, jailson.craibas@gmail.com

Resumo—Nos principais currículos de referência nacionais e internacionais para a graduação em Ciência da Computação e, principalmente, em Engenharia de Computação ocorre a orientação sobre a importância do aprendizado de conceitos relacionados à Arquitetura e Organização de Computadores. Este artigo apresenta uma análise dos resultados obtidos com a aplicação de uma metodologia de ensino para o aprendizado de Arquitetura e Organização de Computadores num currículo de curso moderno e diferenciado voltado a formação em Engenharia de Computação. Este currículo encontra-se alicerçado e estruturado em uma abordagem prática e sistêmica e visa, entre outros objetivos, reduzir a visão fragmentada dos alunos no desenvolvimento de um sistema computacional complexo. A metodologia de ensino aplicada em Arquitetura e Organização de Computadores tem respondido às necessidades dessa nova abordagem, proporcionando um aprendizado centrado no aluno e permitindo o desenvolvimento de habilidades relacionadas ao pensamento criativo, inovador e auto-motivado, indo além da reprodução ou constatação de experimentos.

Palavras-chave—Ensino de Arquitetura e Organização de Computadores; Integração entre Hardware e Software; Abordagem Prática e Sistêmica.

I. INTRODUÇÃO

Arquitetura e Organização de Computadores constitui-se em uma temática extremamente relevante e importante em muitos cursos que envolvem tecnologia, principalmente nos currículos de cursos de graduação em Ciência da Computação e, especialmente, em Engenharia de Computação. Pode-se notar essa importância nos documentos referenciais orientadores produzidos por sociedades de computação internacionais e nacionais, como é o caso da ACM, IEEE e SBC. Especificamente para os cursos de graduação em Engenharia de Computação, conteúdos sobre Arquitetura e Organização de Computadores devem compor papel significativo na formação acadêmica do aluno como relatado no currículo de referência da ACM/IEEE [1]. Neste documento, definem-se áreas de conhecimento que concentram determinados assuntos ou conteúdos específicos que envolvem computação e que devem ser abordados num currículo de Engenharia de Computação. Essas áreas de conhecimento são expressas em horas (*core hours*), correspondendo à quantidade mínima em sala de aula que esses conteúdos devem ser abordados. Neste contexto, a ACM/IEEE define treze áreas de

conhecimento específico com as seguintes distribuições de horas: Circuitos e Eletrônicas (*Circuits and Electronics*) - 50 horas, Algoritmos Computacionais (*Computing Algorithms*) - 30 horas, Arquitetura e Organização de Computadores (*Computer Architecture and Organization*) - 60 horas, Projeto de Sistemas Digitais (*Digital Design*) - 50 horas, Sistemas Embarcados (*Embedded Systems*) - 40 horas, Redes de Computadores (*Computer Networks*) - 20 horas, Práticas Profissionais (*Professional Practice*) - 20 horas, Segurança da Informação (*Information Security*) - 20 horas, Estratégias para Tecnologias Emergentes (*Strategies for Emerging Technologies*) - 10 horas, Processamento de Sinal (*Signal Processing*) - 30 horas, Engenharia de Projetos e Sistemas (*Systems and Project Engineering*) - 30 horas, Gerenciamento de Recursos de Dados (*Systems Resource Management*) - 20 horas e, Projeto de Software (*Software Design*) - 40 horas. Considerando a relação intrínseca entre as duas áreas de conhecimento definidas como Arquitetura e Organização de Computadores e Projeto de Sistemas Digitais, perfazendo em conjunto 110 horas num total de 420 horas, pode-se perceber a grande importância dessa temática em um currículo de Engenharia de Computação.

Nacionalmente, a SBC também definiu um documento referente a um currículo de referência para cursos de graduação em Bacharelado de Ciência da Computação e Engenharia de Computação [2]. Nesse documento, a estruturação curricular ocorre por meio de núcleos de conhecimento, organizados da seguinte forma: Fundamentos da Computação, Tecnologia da Computação, Matemática, Ciências Básicas, Eletrônica e Contexto Social e Profissional. Conceitos fortemente relacionados com Arquitetura e Organização de Computadores são listados nos dois principais núcleos específicos da área de computação, a saber: Fundamentos da Computação (Circuitos Digitais e Arquitetura e Organização de Computadores) e Tecnologia da Computação (Sistemas Digitais). Mais recentemente, em 2012, a SBC e o Ministério da Educação (MEC) definiram algumas diretrizes curriculares para diversos cursos de graduação relacionados à área de computação [3], discriminando conteúdos básicos e tecnológicos que devem ser escolhidos e trabalhados levando-se em consideração o perfil desejado do aluno egresso. Nessas diretrizes curriculares também pode-se notar a importância dessa temática, onde cinco de onze competências e habilidades específicas elencadas para os egressos de cursos de bacharelado em Engenharia de

Computação referem-se a conceitos que estão diretamente relacionados com Arquitetura e Organização de Computadores, quais sejam: (1) Planejar, especificar, projetar, implementar, testar, verificar e validar sistemas de computação (sistemas digitais), incluindo computadores, sistemas baseados em microprocessadores, sistemas de comunicações e sistemas de automação, seguindo teorias, princípios, métodos, técnicas e procedimentos da Computação e da Engenharia; (2) Desenvolver processadores específicos, sistemas integrados e sistemas embarcados, incluindo o desenvolvimento de software para esses sistemas; (3) Analisar e avaliar arquiteturas de computadores, incluindo plataformas paralelas e distribuídas, como também desenvolver e otimizar software para elas; (4) Analisar, avaliar e selecionar plataformas de hardware e software adequados para suporte de aplicação e sistemas embarcados de tempo real; e (5) Analisar, avaliar, selecionar e configurar plataformas de hardware para o desenvolvimento e implementação de aplicações de software e serviços.

Sendo assim, devido a sua importância para os cursos de graduação em Computação, novas técnicas e metodologias para o ensino de arquitetura e organização de computadores vem sendo aplicadas em diversos currículos de Computação e publicadas na literatura científica nacional e internacional, como pode ser constatado na Seção II referente aos trabalhos relacionados.

Particularmente, na Universidade Federal de São Paulo - Unifesp, o ensino e aprendizado de arquitetura e organização de computadores está inserido num currículo de curso diferenciado [4], estruturado em uma abordagem prática e sistêmica que busca, além de integrar teoria e prática e também hardware e software, reduzir a visão fragmentada dos alunos de um sistema computacional complexo. Convencionalmente, os currículos de curso de Engenharia de Computação apesar de apresentarem em suas matrizes curriculares unidades curriculares (disciplinas, matérias ou cadeiras) relacionadas ao desenvolvimento de hardware e de software, estas unidades curriculares previstas em seus respectivos projetos pedagógicos não costumam ser interligadas, e assim os alunos acabam adquirindo uma visão fragmentada de um sistema computacional realmente complexo. Na Seção III, encontram-se mais informações sobre a abordagem diferenciada realizada na Unifesp e, especificamente, sobre a metodologia de ensino e aprendizado em Arquitetura e Organização de Computadores definida para responder ou atender as necessidades dessa nova abordagem.

Dentro deste contexto, um estudo foi publicado em [5] sobre o ensino e aprendizado de Arquitetura e Organização de Computadores inserido nessa abordagem prática e sistêmica do currículo de Engenharia de Computação da Unifesp. Neste artigo, apresenta-se o tema sob a ótica ou o ponto de vista dos alunos, com a aplicação de questionários sobre a metodologia aplicada. Por sua vez, o artigo aqui presente apresenta a metodologia de ensino e aprendizado em Arquitetura e Organização de Computadores em uma nova perspectiva, com os seguintes enfoques: (1) nas plataformas de hardware que estão sendo desenvolvidas; (2) nas especificações que estão sendo definidas e implementadas pelos alunos; (3) nos motivos que tem levado os alunos a elaborarem e realizarem essas especificações; (4) nos benefícios da metodologia aplicada sob

o prisma das plataformas desenvolvidas e; (5) nas medidas a serem tomadas para melhorias e adaptações dessa metodologia de ensino-aprendizagem.

II. TRABALHOS RELACIONADOS

Metodologias ativas de aprendizagem têm sido aplicadas com sucesso no ensino de arquitetura e organização de computadores, sendo reportadas em diversos trabalhos da literatura científica nacional e internacional. Como exemplo, pode-se citar o trabalho desenvolvido em [6], onde os autores descrevem a elaboração de uma disciplina com atividades práticas em sala de aula, laboratórios e tarefas a serem realizadas pela web. Em outro exemplo, mais recente, os autores em [7] apresentam a inclusão de pedagogias ativas numa disciplina de Arquitetura de Computadores em um currículo de curso de Engenharia de Computação específico.

Duas vertentes têm sido adotadas pelos pesquisadores como apoio na aplicação de metodologias ativas para o ensino e aprendizado de arquitetura e organização de computadores. Numa primeira vertente, simuladores didáticos de microprocessadores acadêmicos educacionais ou comerciais têm sido utilizados. Numa segunda vertente, muitos pesquisadores têm adotado o uso de kits educacionais podendo conter microcontroladores, FPGAs, entre outros componentes eletrônicos para o ensino de arquitetura e organização de computadores e de conceitos correlacionados. Nos parágrafos a seguir, encontra-se uma compilação de trabalhos extraídos da literatura científica relacionados à aplicação de metodologias ativas de acordo com as duas vertentes anteriormente caracterizadas.

Englobando-os numa primeira vertente, diversos simuladores [8] têm sido propostos para o uso em disciplinas de arquitetura e organização de computadores. Como exemplos de simuladores didáticos desenvolvidos no exterior pode-se citar: GNUSim8085 [9], MARS [10] e SIMICS [11].

O simulador GNUSim8085 baseia-se na arquitetura do microprocessador 8085 da Intel, podendo apresentar os valores dos seus registradores e conteúdo da memória durante a execução das instruções. Por sua vez, o simulador MARS foi proposto para a arquitetura MIPS, sendo um dos mais populares e bem documentados, permitindo a visualização do conteúdo de seus registradores, da memória e das instruções em cada estágio do pipeline. Por fim, SIMICS é um simulador capaz de modelar arquiteturas single-core, multi-core e many-core; simular os conjuntos de instruções Alpha, x86, x86-64, Power PC, IPF, MIPS, ARM e Ultra Sparc; e emular sistemas operacionais como Linux, Solaris e Windows.

Especificamente no Brasil, diversos simuladores vêm sendo propostos na literatura científica. Como exemplos recentes pode-se citar: SimuS [12], BIPIDE [13] e MPSoCBench [14].

O simulador SimuS apresenta um ambiente integrado onde o aluno pode editar, compilar, depurar e executar código de programas escrito em linguagem de montagem do processador hipotético Sapiens, baseado no processador Neander [15]. O BIPIDE é um ambiente de desenvolvimento baseado na arquitetura BIP (*Basic Instruction-set Processor*), tendo um editor de código e um compilador desenvolvido para

reconhecer algoritmos simples na pseudo-linguagem Portugol. O ambiente pode exibir passo-a-passo uma animação interna da arquitetura em relação ao programa executado. Por sua vez, a ferramenta MPSoCBench é capaz de simular sistemas multiprocessados baseados em ARM, MIPS, PowerPC e SPARC, memórias caches e componentes de interconexão, além de fornecer estimativas de desempenho, temporização e consumo de potência.

Classificando-os numa segunda vertente, os trabalhos apresentados em [16], [17] e [18] se apoiam no uso de kits educacionais para viabilizar pedagogias ativas no ensino de conceitos que envolvem arquitetura e organização de computadores. Em [16], o autor descreve a metodologia utilizada em uma disciplina onde um determinado projeto relacionado a arquitetura e organização de computadores é inicialmente especificado, devendo ser desenvolvido pelos alunos durante a realização da disciplina. Em [17], os autores propõem o uso de metodologia ativa (*learning-by-doing*) para o desenvolvimento e implementação em FPGA de um processador, onde conceitos de dispositivos lógicos programáveis e da linguagem de descrição de hardware VHDL são introduzidos aos alunos. Em [18] e [19], os autores descrevem o uso do console de videogame Nintendo em um ambiente de aprendizagem baseado em problema ou projeto (PBL ou PjBL), onde a partir de um projeto base referente ao desenvolvimento de um determinado jogo, os alunos são estimulados a aprender conceitos de subsistemas de Entrada/Saída (E/S) como, por exemplo, E/S programada (*polling*) ou E/S orientada à interrupção.

Estudos aqui no Brasil direcionados a essa segunda vertente também são realizados, podendo-se citar os trabalhos publicados em [20], [21], [22] e [23]. Os autores em [20] abordam práticas de ensino para a disciplina de Circuitos Digitais por meio do uso de matrizes de contatos (*protoboards*), componentes eletrônicos e circuitos integrados. Os autores em [21] fazem uma análise do uso de microcontroladores nas universidades do sul do Brasil como ferramenta de apoio ao ensino de Arquitetura de Computadores. Os autores em [22] [24] propõem a resolução pelos alunos de problemas ou projetos envolvendo o desenvolvimento de programas ou soluções em dispositivos lógicos programáveis aplicando uma metodologia baseada em problemas. Os autores em [23] [25] propõem um ensino prático de projeto de processadores em FPGAs seguindo uma metodologia baseada em projetos.

Dentro do contexto dessa segunda vertente, no currículo de Engenharia de Computação da Unifesp busca-se uma nova abordagem para a realização da integração entre teoria e prática possibilitando ao aluno o desenvolvimento de um sistema computacional completo durante sua trajetória acadêmica. A ideia principal é a definição de uma metodologia de desenvolvimento efetivamente centrada no aluno, que o permita ir além da reprodução ou constatação de experimentos; uma nova abordagem que seja capaz de trabalhar e desenvolver habilidades relacionadas ao pensamento criativo e inovador, auto-motivado e que mantenha o aluno entusiasmado na aplicação do conhecimento que está sendo adquirido durante sua vida acadêmica.

III. UNIDADES CURRICULARES INTEGRADAS E A RELAÇÃO COM ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES

A cada semestre o aluno do curso de Engenharia de Computação irá cursar disciplinas teóricas que o ajudarão a desenvolver, no semestre seguinte, uma parte do sistema computacional, se matriculando em uma unidade curricular denominada "Laboratório de Sistemas Computacionais". Por exemplo, após aprender conceitos teóricos relacionados a Circuitos Digitais [26] [27], no semestre seguinte, o aluno deverá cursar a unidade curricular de Laboratório de Sistemas Computacionais: Circuitos Digitais, onde ele deverá realizar implementações em FPGAs de circuitos combinacionais e sequenciais [28]. Após ter sido aprovado nessa unidade curricular e ter cursado a unidade curricular teórica de Arquitetura e Organização de Computadores, no semestre seguinte, o aluno deverá se matricular no Laboratório de Sistemas Computacionais: Arquitetura e Organização de Computadores, onde cada aluno deverá desenvolver em linguagem de descrição de hardware Verilog sua própria plataforma de hardware, composta por processador, memória e sistema de entrada/saída utilizando o kit DE2-115 [29] e o software Quartus Prime [30].

O aluno, após ter realizada a implementação da plataforma de hardware, utilizará o conjunto de instruções de baixo nível (código de máquina) desenvolvido para realizar o projeto de uma linguagem de programação que possua uma sintaxe de nível mais alto do que o código de máquina do processador. Após essa especificação, um compilador será descrito e implementado pelo aluno para traduzir essa linguagem de mais alto nível para o código de máquina da plataforma de hardware. Na sequência, o aluno irá utilizar o compilador desenvolvido para projetar um sistema operacional que deverá executar sobre a plataforma de hardware inicialmente desenvolvida nos semestres anteriores. Por fim, no último Laboratório de Sistemas Computacionais, um sistema de comunicação e um protocolo de redes de computadores serão projetados e implementados por cada aluno para a interligação entre dois ou mais sistemas computacionais.

Mais detalhes sobre essa estrutura curricular diferenciada podem ser encontrados no projeto pedagógico do curso [4] e no artigo publicado em [5].

IV. PLATAFORMAS DE HARDWARE DESENVOLVIDAS E DISCUSSÃO

A unidade curricular de Laboratório de Sistemas Computacionais: Arquitetura e Organização de Computadores ocorre anualmente, sendo ofertadas 25 vagas para os alunos de Engenharia de Computação. Durante o oferecimento dessa unidade curricular, características e informações importantes sobre as arquiteturas das plataformas de hardware desenvolvidas e mapeadas em FPGA vêm sendo extraídas e compiladas para que seja possível identificar a eficácia e os pontos que precisam ser melhorados em relação à aplicação da abordagem de ensino-aprendizagem apresentada na Seção III. Essas características e informações das arquiteturas projetadas foram obtidas por meio dos seminários apresentados pelos alunos e dos relatórios técnicos entregues durante a unidade curricular como parte do processo de avaliação.

Nas próximas subseções, apresentam-se os resultados e análises decorrentes desse rastreamento de informações, levando em consideração o tipo de arquitetura desenvolvida na plataforma de hardware (subseção A), ao tamanho da palavra de dados (subseção B), a memória de dados e instruções (subseção C), a existência de pipeline e ciclos de processamento (subseção D) e ao modelo do subsistema de comunicação de Entrada/Saída (subseção E).

A. Tipo de Arquitetura

De forma semelhante à taxonomia definida em [31], a arquitetura desenvolvida pelo aluno foi classificada como sendo uma dentre cinco possibilidades:

- **Pilha:** Nesta arquitetura, os dados necessários para a execução de determinadas instruções que envolvem operações da ULA (Unidade Lógica e Aritmética) são provenientes de uma pilha, a qual pode ser implementada via registradores ou memória principal. Sendo assim, a operação de soma, por exemplo, acontece com os dados provenientes dos dois últimos elementos armazenados na pilha. O resultado da operação é armazenado no topo da pilha.
- **Acumulador:** Nesta arquitetura, um dos dados para a execução de determinadas instruções que envolvem operações da ULA é proveniente de um registrador especial, denominado acumulador. Neste sentido, a operação de soma, por exemplo, acontece com um dado sendo proveniente do acumulador e o outro dado podendo ser da memória principal. O resultado da operação é armazenado no próprio acumulador.
- **Registrador-Registrador (Reg.-Reg.):** Neste tipo de arquitetura, todos os dados para a realização de determinadas instruções envolvendo operações lógicas e aritméticas executadas pela ULA são provenientes de registradores de propósito geral. Por exemplo, para uma operação de soma, os dois dados necessários são provenientes de um banco de registradores, sendo selecionados pela instrução a ser executada. O resultado da operação também deve ser armazenado em um registrador.
- **Registrador-Memória (Reg.-Mem.):** Nesta arquitetura, apenas um dos dados necessários para a realização de determinadas instruções que envolvem a execução de operações lógicas e aritméticas pela ULA pode residir na memória principal. Por exemplo, numa operação de soma, um dado pode vir da memória principal, enquanto o outro sendo proveniente de um registrador. O resultado da operação deve ser armazenado em um registrador.
- **Memória-Memória (Mem.-Mem.):** Nesta arquitetura, todos os dados necessários para a realização de determinadas instruções que envolvem operações lógicas e aritméticas pela ULA podem residir na memória principal. Por exemplo, numa operação de soma, podemos ter os dois dados necessários para a realização da operação provenientes da memória. O

resultado da operação pode ser armazenado na memória principal.

Identificando o tipo de arquitetura projetada em cada plataforma de hardware, pôde-se obter o diagrama de distribuição mostrada na Figura 1, relacionando a quantidade de plataformas de hardware com cada um dos cinco tipos de arquitetura mencionados anteriormente. A arquitetura pilha foi utilizada em aproximadamente 8% das plataformas de hardware implementadas; a arquitetura baseada em acumulador foi implementada em aproximadamente 17% das plataformas; a arquitetura registrador-registrador foi a mais utilizada, representando aproximadamente 64% das plataformas desenvolvidas; enquanto a arquitetura registrador-memória representou 8% do total de plataformas e; por fim, a arquitetura Memória-Memória foi projetada em aproximadamente 3% das plataformas.

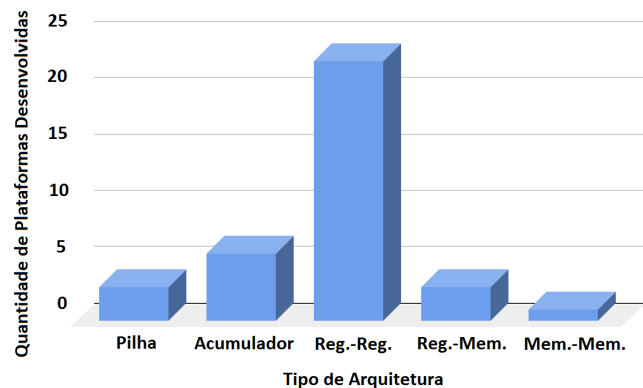


Fig. 1. Distribuição das plataformas de hardware em relação ao tipo de arquitetura projetada

Vale a pena destacar que as duas arquiteturas mais utilizadas foram a baseada em acumulador e a Registrador-Registrador, sendo esta última a mais implementada pelos alunos. De acordo com os seminários realizados pelos alunos e os relatórios entregues sobre as plataformas de hardware, observaram-se os referenciais orientadores que foram utilizados pelos alunos na realização de seus próprios projetos. Para a arquitetura Registrador-Registrador, os processadores mais citados e utilizados como referencial em determinadas plataformas de hardware implementadas foram o MIPS, o ARM e o PowerPC.

Para um melhor entendimento sobre os motivos que levaram esses dois tipos de arquitetura, em conjunto, representarem em torno de 80% das plataformas de hardware implementadas, diversos alunos foram convidados individualmente para uma conversa sobre as decisões de projeto que eles efetuaram. Após as realizações dessas conversas, os principais pontos a serem destacados são:

- (1) Um conjunto de alunos relatou ter tido certa dificuldade inicial para entender efetivamente o funcionamento de um processador e como se daria a sua implementação em hardware. Importante destacar que a maioria dos alunos que alegaram essa dificuldade inicial acabou implementando uma

plataforma de hardware utilizando a arquitetura baseada em acumulador.

- (2) Muitos alunos disseram que se sentiram mais confortáveis e confiantes em implementar uma arquitetura Registrador-Registrador devido ao conhecimento prévio sobre esse tipo de arquitetura, além de terem comentado que era mais fácil encontrar material didático e bibliografia sobre esse tipo de arquitetura na internet e na biblioteca.
- (3) Alguns alunos alegaram que se sentiram motivados em implementar uma arquitetura do tipo Registrador-Registrador tendo em vista que videogames, roteadores e muitos celulares utilizam processadores com esse tipo de arquitetura.
- (4) Alguns alunos disseram ter tido interesse em implementar arquiteturas do tipo Registrador-Memória ou Memória-Memória, mas ficaram apreensivos alegando medo de que não conseguiriam implementá-las até o final da unidade curricular, pois consideraram essas abordagens de maior complexidade técnica e de projeto.

De acordo com os apontamentos em (1), os alunos que apresentaram dificuldades sobre o entendimento de arquitetura de computadores acabaram superando suas deficiências técnicas por meio de uma arquitetura baseada em acumulador. Isso pode ser visto como um indício de que arquiteturas baseadas em acumuladores tem se mostrado didáticas e mais tangíveis a compreensão inicial do assunto pelos alunos. Este fato é interessante, tendo em vista que muitos estudos na área de educação de arquitetura de computadores têm adotado esse tipo de arquitetura para o ensino. Os autores em [33] apresentam o simulador CPU Sim como um pacote de software educacional para o ensino de arquitetura de computadores em conjunto com o processador Wombat1 e suas variantes cujas arquiteturas são baseadas em acumuladores. Por sua vez, os autores em [34] descrevem para o ensino de arquitetura e organização de computadores um simulador gráfico via Web denominado EDCOMP, contendo um processador baseado em acumulador. De maneira semelhante, os autores em [35] propõem para as aulas introdutórias de organização de computadores o uso da ferramenta de simulação via Web do LMC (*Little Man Computer*) cuja arquitetura interna é baseada em acumulador. Especificamente no Brasil, pode-se citar o trabalho que vem sendo realizado ao longo dos últimos 10 anos relacionado ao processador baseado em acumulador denominado BIP (*Basic Instruction-set Processor*) e todas as suas variantes para o ensino de várias disciplinas de computação [36] [37]. Os autores em [38] apresentam uma abordagem de ensino subdividida em três etapas de aprendizado, onde na primeira etapa utiliza-se como referência o computador IAS cuja arquitetura interna estrutura-se por meio de um acumulador. Por fim, os autores em [39] abordam o simulador SimuS, desenvolvido para o processador hipotético baseado em acumulador Sapiens (derivado do Neander [40]) sendo especialmente concebido para o ensino de arquitetura de computadores.

Sobre o apontamento (2), vale a pena ressaltar que na unidade curricular teórica de Arquitetura e Organização de

Computadores, a qual ocorre no semestre anterior ao laboratório de Sistemas Computacionais: Arquitetura e Organização de Computadores, os alunos são normalmente apresentados a organização de computadores por meio de uma arquitetura registrador-registrador. A arquitetura MIPS tem sido utilizada na Unifesp para o aprendizado sobre o funcionamento interno de um processador. Esse fato, como apontado pelos alunos em (2), influenciou a escolha do tipo de arquitetura projetada por alguns alunos.

Outra influência interessante refere-se ao apontamento (3) em que aparentemente a visibilidade e o uso comercial muito diversificado de arquiteturas registrador-registrador (muito utilizadas em roteadores, videogames, celulares, sistemas embarcados diversos entre outros equipamentos eletrônicos) têm atraído a atenção e a curiosidade de alguns alunos, os quais acabam optando por desenvolver uma plataforma de hardware com preceitos semelhantes.

Para tentar contornar os apontamentos (2) e (4), no próximo oferecimento da unidade curricular de laboratório de Sistemas Computacionais: Arquitetura e Organização de Computadores, serão selecionados os melhores projetos desenvolvidos pelas turmas anteriores, buscando disponibilizar aos alunos projetos referentes a cada um dos cinco tipos de arquitetura mencionados anteriormente. Os projetos que forem bem executados e que possuam relatórios com uma boa documentação da arquitetura desenvolvida serão disponibilizados via Moodle [41], uma plataforma de aprendizagem a distância utilizada oficialmente pela Unifesp como apoio aos cursos presenciais de graduação da universidade. Com essa medida, ao mesmo tempo em que se visa oferecer material de apoio variado e aderente com a proposta de desenvolvimento de um sistema computacional, também se busca imbuir confiança para que o aluno se sinta confortável em desenvolver qualquer um dos cinco tipos de arquitetura.

Uma segunda medida que também pode ser adotada refere-se a apresentar outras arquiteturas de computadores além do MIPS aos alunos que estão cursando a unidade curricular teórica de Arquitetura e Organização de Computadores. O uso de simuladores apropriados para o ensino e aprendizagem de arquitetura e organização de computadores [42] pode ser uma boa forma de introduzir diferentes arquiteturas, tais como DLX, LC-2, ARM, SPARC, x86, PowerPC entre outras.

B. Palavra de Dados

Na Figura 2, encontra-se a quantidade de bits referente ao tamanho da palavra de dados definida para as plataformas de hardware desenvolvidas.

Interessante observar que na maioria das plataformas de hardware desenvolvidas (aproximadamente 72%) adotou-se um tamanho para a palavra de dados de até 18 bits, enquanto que em apenas 28% das plataformas de hardware reservou-se 32 bits como tamanho padrão para a palavra de dados. Essa distribuição pode ser explicada devido à restrição do kit DE2-115 utilizado pelos alunos para a implementação física da plataforma de hardware. Esse kit possui 18 chaves liga/desliga, as quais são utilizadas como interface de comunicação entre a plataforma de hardware e o mundo externo ou o usuário.

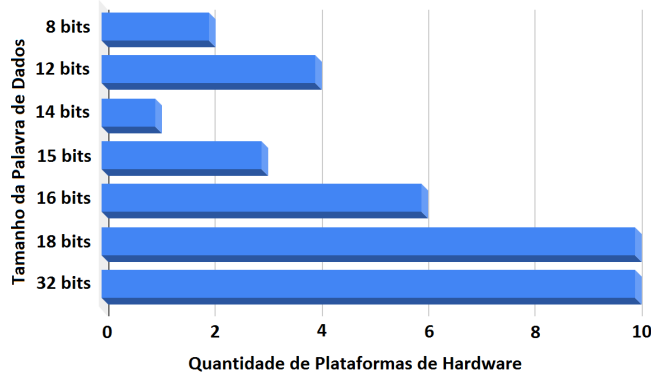


Fig. 2. Tamanho da palavra de dados adotado nas plataformas de hardware desenvolvidas

Além de dados de entrada, outros sinais podem ser necessários para o correto funcionamento da plataforma de hardware, como por exemplo, os sinais de controle reset, interrupção e *acknowledgment*. Especificamente para as plataformas de hardware cujas palavras de dados possuem exatamente a mesma quantidade de bits das chaves liga/desliga (18 bits), os botões do tipo *push* são utilizados para os sinais de controle. Para plataformas de hardware com palavras de dados menores do que 18 bits, tanto botões *push* como as chaves liga/desliga foram utilizadas.

A entrada de dados, em todas as plataformas, foi realizada de forma paralela, ou seja, os bits das palavras de dados são carregados ao mesmo tempo para a plataforma de hardware. Mesmo para as plataformas de hardware onde as palavras de dados são compostas de 32 bits, o carregamento ocorre em paralelo, podendo, no entanto, utilizar uma quantidade menor de bits de acordo com a quantidade disponível de chaves liga/desliga do kit DE 2-115. Extensores de bits foram muitas vezes implementados para a entrada proveniente das chaves liga/desliga.

Outras soluções ou abordagens [27] poderiam ter sido adotadas pelos alunos quanto ao carregamento de dados do mundo externo provenientes das chaves liga/desliga para a plataforma de hardware, como por exemplo, a utilização de um carregamento parcialmente serial, dividido em etapas (metade dos bits menos significativos sendo carregados em paralelo primeiro e a outra metade dos bits sendo carregados na sequência). Uma análise mais aprofundada pode ser realizada para identificar os motivos que levaram os alunos a terem adotado a abordagem de carregamento integralmente em paralelo. Provavelmente, a maior complexidade de outras soluções e a implementação de arquiteturas monociclo contribuíram para a escolha dos alunos quanto à forma de carregamento dos dados externos. Vale ressaltar, no entanto, que em algumas plataformas de hardware foram implementadas instruções específicas que são capazes de ler e carregar internamente, por exemplo, a metade mais significativa ou a menos significativa de uma palavra de dados.

C. Memória de Dados e Instruções

Houve uma diferenciação quanto às implementações das memórias principais nas plataformas de hardware desenvolvidas pelos alunos. As duas formas de implementação [45] em relação à memória principal adotadas pelos alunos foram:

- **Arquitetura Von Neumann:** neste caso, a plataforma de hardware contém em uma única memória tanto as instruções de programas que são executadas pela CPU quanto os dados e operandos necessários para a execução das instruções, tendo o mesmo espaço de endereçamento para os programas e para os dados.
- **Arquitetura Harvard:** neste caso, a plataforma de hardware é implementada por meio de duas memórias particulares: uma para armazenar as instruções de programas e outra para armazenar os dados e operandos, tendo espaços de endereçamento distintos para os programas e para os dados.

Em aproximadamente 77% das plataformas de hardware implementou-se, para a memória principal, uma arquitetura Harvard. Esse número elevado deve estar diretamente relacionado com a maior preferência dos alunos em implementar arquiteturas do tipo Registrador-Registrador. De acordo com a subseção A sobre os tipos de arquiteturas desenvolvidas pelos alunos, aproximadamente 64% das plataformas de hardware desenvolvidas pelos alunos se baseiam em arquiteturas do tipo Registrador-Registrador e, entre as plataformas onde se implementou a arquitetura do tipo Registrador-Registrador, aproximadamente 78% possuem, para a memória principal, a arquitetura Harvard. Essa relação entre arquiteturas Harvard e arquiteturas do tipo Registrador-Registrador também pode ser corroborado com o fato de que uma grande quantidade de processadores comerciais e educacionais do tipo Registrador-Registrador possuem uma implementação baseada em duas memórias distintas para armazenar as instruções e os dados (Arquitetura Harvard), como é o caso, por exemplo, de alguns processadores ARM e MIPS.

D. Pipeline e Ciclo

Foram computados aspectos da implementação das plataformas de hardware referentes, respectivamente, a existência da técnica pipeline (Sim ou Não) e a necessidade de execução completa de uma instrução em um único ciclo de relógio (monociclo) ou em dois ou mais ciclos (multiciclo).

As plataformas de hardware que foram desenvolvidas com a técnica pipeline apresentam um modelo de CPU multiciclo. Todas as plataformas de hardware desenvolvidas pelos alunos com a técnica pipeline são do tipo Registrador-Registrador e todas as plataformas de hardware do tipo Registrador-Memória ou Memória-Memória foram implementadas com a CPU trabalhando em multiciclo.

De certa forma, para as plataformas com a técnica pipeline, essas correlações podem ser explicadas devido ao grau de complexidade na implementação dessa técnica em determinados tipos de arquitetura. O uso da técnica pipeline em arquiteturas do tipo Registrador-Memória e Memória-Memória

pode ser mais complexo, já que, por exemplo, operando a serem processados na ULA podem estar em registradores ou em memória devido aos vários modos de endereçamento implementados, dificultando a sua implementação nesses tipos de arquitetura.

De maneira análoga, a maior complexidade na identificação e busca dos operandos para a realização de uma determinada operação, por exemplo, deve ter conduzido os alunos a procurarem implementar uma CPU multiciclo em arquiteturas do tipo Registrador-Memória e Memória-Memória, dividindo-se assim em subtarefas menores o processo completo de realização das instruções.

Uma questão interessante refere-se a técnica utilizada pelos alunos para implementar a Unidade de Controle da plataforma de hardware desenvolvida. Ao analisar os relatórios produzidos pelos alunos, os arquivos de projeto e as apresentações orais realizadas, classificaram-se as implementações da Unidade de Controle em três tipos distintos, quais sejam:

- **Circuito Combinacional simples (Hardwired):** utilizaram-se técnicas de projeto de circuitos digitais combinacionais [26] para a implementação da unidade de controle, onde todos os sinais de saída do circuito digital que compõem a unidade de controle dependem única e exclusivamente das várias combinações das variáveis de entrada.
- **Máquina de Estados Finitos (Hardwired):** utilizaram-se técnicas de projeto de máquinas de estados finitos de Moore ou Mealy [28] que implementam a unidade de controle da plataforma de hardware, sendo o circuito digital resultante composto por registradores que armazenam os estados definidos e portas lógicas combinacionais que definem o próximo estado e os valores dos sinais de saída.
- **Microprogramação:** utilizou-se uma estrutura lógica particular, composta, basicamente, por uma memória de controle (ROM) que contém microinstruções, um sequenciador encarregado de manter a ordem correta de execução das microinstruções e um contador de microprograma utilizado para armazenar a próxima microinstrução [32]. Essa estrutura lógica é capaz de gerar os sinais de controle que devem ser enviados para a unidade de processamento da CPU para o seu gerenciamento.

Na Figura 3, pode-se observar a distribuição das plataformas de hardware desenvolvidas em relação à classificação definida no parágrafo anterior para cada tipo de implementação realizada na Unidade de Controle.

De acordo com a Figura 3, em aproximadamente 69% das plataformas de hardware utilizou-se um circuito combinacional simples para a implementação da Unidade de Controle, em 22% implementou-se uma máquina de estados finitos como Unidade de Controle e em apenas 9% das plataformas de hardware desenvolvidas a microprogramação foi utilizada.

A microprogramação pode ser muito adequada na implementação da unidade de controle das plataformas de hardware que estão sendo desenvolvidas na unidade curricular

de laboratório de Sistemas Computacionais: Arquitetura e Organização de Computadores, pois uma de suas vantagens é a maior facilidade, por exemplo, para a inclusão de novas instruções [43], tornando o projeto da unidade de controle mais flexível.

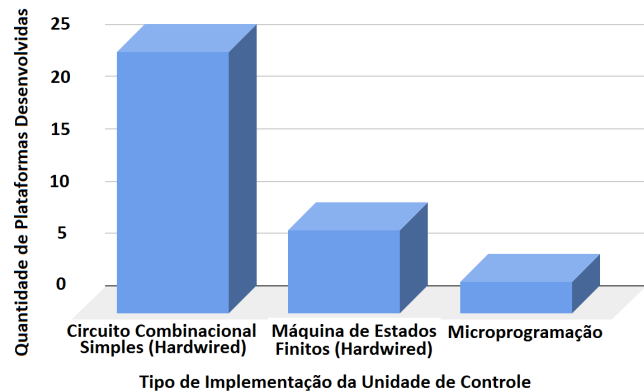


Fig. 3. Distribuição das plataformas de hardware desenvolvidas em relação ao tipo de implementação realizada na Unidade de Controle

Mesmo após a aprovação na unidade curricular de laboratório de Sistemas Computacionais: Arquitetura e Organização de Computadores, os alunos continuam fazendo melhorias nas plataformas de hardware, como apontado no trabalho realizado em [5] e, além disso, nos próximos laboratórios de sistemas computacionais novas funcionalidades, mudanças e adaptações serão continuamente realizadas ou incluídas nas plataformas de hardware para responder a crescente complexidade das tarefas que o sistema computacional que está sendo desenvolvido deverá executar.

Contudo, vale a pena observar que, mesmo com os benefícios apontados no parágrafo anterior relacionados ao uso da microprogramação nas unidades de controle das plataformas de hardware, poucos alunos têm adotado essa forma de implementação, sendo utilizada em apenas 9% das plataformas desenvolvidas.

Para um melhor entendimento sobre essa distribuição nas formas de implementação da Unidade de Controle, algumas conversas foram realizadas com alunos que desenvolveram suas plataformas de hardware e com monitores e professores das unidades curriculares práticas e teóricas relacionadas aos laboratórios de sistemas computacionais. Após as realizações dessas conversas, os principais pontos a serem destacados são:

- (1) Na unidade curricular de laboratório de Sistemas Computacionais: Circuitos Digitais, os alunos desenvolvem projetos em esquemático e em Verilog de diversos circuitos combinacionais e máquinas de estados finitos, sintetizando-os no kit FPGA DE 2-115.
- (2) Devido à grande quantidade de assuntos que são abordados na unidade curricular teórica de Arquitetura e Organização de Computadores, não é dada ênfase nas diferentes formas de implementação da Unidade de Controle.

Sendo assim, muitos alunos que se matriculam na unidade curricular de Laboratório de Sistemas Computacionais: Arquitetura e Organização de Computadores, mostram ter pouco domínio sobre microprogramação em relação à sua concepção e implementação em FPGA, o que tem contribuído para que a maioria das Unidades de Controle sejam implementadas por meio de circuitos combinacionais simples ou máquinas de estados finitos, devido ao conhecimento prévio obtido por esses alunos nos semestres anteriores sobre essas duas técnicas de projeto.

Para contornar esses apontamentos, no oferecimento da unidade curricular de Laboratório de Sistemas Computacionais: Arquitetura e Organização de Computadores dos próximos semestres, será utilizado o ambiente Moodle para disponibilizar aos alunos material de apoio sobre as três formas de implementação de unidades de controle como definidas anteriormente. Neste material de apoio, deve-se explicitar não somente a concepção dos conceitos envolvidos e de suas possíveis implementações em FPGA, mas também as vantagens e desvantagens de cada uma dessas abordagens.

E. Modelo de Entrada e Saída (E/S)

Foram identificadas as implementações do subsistema de Entrada/Saída (E/S) das plataformas de hardware desenvolvidas referentes à forma como são realizadas a comunicação entre a CPU e os dispositivos de E/S especificados. A implementação do subsistema de E/S foi estratificada em dois esquemas distintos, podendo ser [43]:

- **E/S Independente ou mapeada em Instruções Específicas:** neste esquema, instruções específicas fazem parte do conjunto de instruções do processador para acesso aos dispositivos de E/S.
- **E/S mapeada em Memória:** neste esquema, endereços da memória principal são utilizados pelo processador para a troca de informações com os dispositivos de E/S. Para isso, utilizam-se as mesmas instruções do conjunto de instruções do processador que acessam a memória principal para a comunicação com os dispositivos de E/S.

De acordo com o levantamento realizado, aproximadamente 83% das plataformas de hardware desenvolvidas pelos alunos possuem um subsistema de E/S independente ou mapeado em Instruções Específicas, enquanto em 17% implementa-se a E/S mapeada em Memória. Dentro deste contexto, é interessante observar os tipos de dispositivos de E/S adotados e implementados nas plataformas de hardware desenvolvidas. Na Figura 4, encontra-se a distribuição dos dispositivos de E/S utilizados pelos alunos. Pode-se notar que foram utilizados cinco tipos de dispositivos de E/S, quais sejam: chaves Liga/Desliga e botões do tipo *push*, displays de 7-segmentos, LEDs, Arduinos e módulos LCDs. Resumidamente, como dispositivos de entrada foram utilizados as chaves Liga/Desliga, botões do tipo *push* e Arduinos. Como dispositivos de saída, foram utilizados os displays de 7-segmentos, LEDs, módulos LCDs e também Arduinos.

Todas as plataformas de hardware desenvolvidas possuem como dispositivos de entrada as chaves Liga/Desliga e botões

do tipo *push* e como dispositivos de saída os displays de 7-segmentos, já que o uso desses dispositivos como obrigatórios foram enunciados desde o início da unidade curricular de Laboratório de Sistemas Computacionais: Arquitetura e Organização de Computadores.

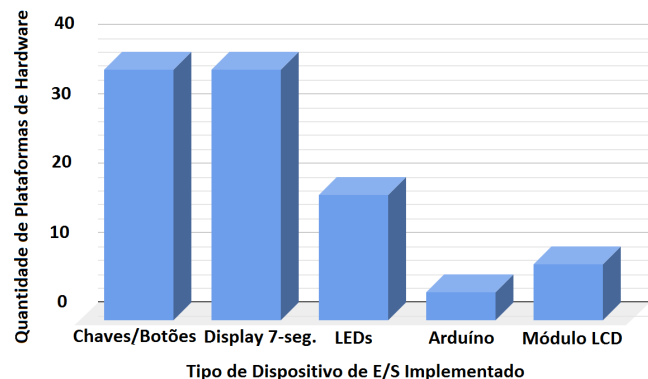


Fig. 4. Tipos de Dispositivos de E/S implementados nas plataformas de hardware desenvolvidas

No entanto, vale ressaltar que, além do uso dos dispositivos obrigatórios, muitos alunos implementaram outras formas de E/S. Por exemplo, em aproximadamente 34% das plataformas de hardware foram também implementadas interfaces de E/S de maior complexidade com Arduíno e módulos LCDs, exigindo um conhecimento adicional sobre conceitos de E/S, microcontroladores e protocolos de comunicação. Este fato ratifica os resultados apresentados em [5] de que a abordagem adotada tem explorado e estimulado a criatividade dos alunos, além de estar motivando-os tanto na realização de projetos práticos quanto no estudo de assuntos teóricos relacionados.

Por fim, como medida para buscar diversificar as implementações de dispositivos de E/S nas plataformas de hardware a serem desenvolvidas, nos próximos oferecimentos da unidade curricular de Laboratório de Sistemas Computacionais: Circuitos Digitais, que antecede o Laboratório de Sistemas Computacionais: Arquitetura e Organização de Computadores, experimentos e projetos relacionados à implementação em FPGAs utilizando a linguagem Verilog de E/S como teclado e monitor serão incluídos e realizados para incentivar os alunos a utilizarem outros dispositivos de E/S além das chaves/botões e displays de 7-segmentos.

V. CONSIDERAÇÕES FINAIS

Este artigo apresentou os resultados que vem sendo obtidos com a aplicação da metodologia de ensino-aprendizagem realizada na unidade curricular de Laboratório de Sistemas Computacionais: Arquitetura e Organização de Computadores direcionada a um currículo de Engenharia de Computação diferenciado, o qual estrutura-se em uma abordagem prática e sistêmica com o objetivo de convergir transversalmente diversos conceitos de computação de forma evolucionista. A abordagem adotada nos laboratórios permite a integração de conceitos de hardware e software, proporcionando um ambiente de aprendizagem efetivamente centrado no aluno.

Sob a luz das plataformas de hardware desenvolvidas pelos alunos que cursaram a unidade curricular de Laboratório de Sistemas Computacionais: Arquitetura e Organização de Computadores, a metodologia tem produzido resultados interessantes, buscando imprimir habilidades e atitudes que vão além da reprodução de experimentos, enfatizando-se e incentivando-se: (1) o pensamento criativo, sendo a especificação do projeto efetivamente elaborada pelo próprio aluno e o consequente desenvolvimento da plataforma de hardware correspondendo a seus desejos e anseios; (2) a inovação, representada pela diversidade das características e informações extraídas das plataformas de hardware desenvolvidas; (3) a auto-motivação, demonstrada nas implementações de outras formas de E/S além do uso dos dispositivos obrigatórios e; (4) a aplicação de conhecimentos teóricos em projetos práticos, como resultado de todos os artefatos produzidos.

Ao se investigar os motivos que levaram os alunos a tomarem determinadas decisões de projeto e a definirem e implementarem suas plataformas de hardware, foi possível estabelecer algumas medidas com o intuito de buscar a melhoria ou a adaptação da metodologia de ensino-aprendizagem aplicada. Como, por exemplo, a medida que busca disponibilizar aos alunos projetos referentes a cada um dos cinco tipos de arquitetura, a medida que se refere a confecção de material de apoio sobre as três formas de implementação de unidades de controle e a medida que busca diversificar as implementações de dispositivos de E/S nas plataformas de hardware a serem desenvolvidas.

Por fim, vale ressaltar que, após a aprovação na unidade curricular de Laboratório de Sistemas Computacionais: Arquitetura e Organização de Computadores, o aluno continuará a desenvolver o sistema computacional completo durante o seu processo de aprendizagem, devendo dar sequência aos seus estudos matriculando-se na unidade curricular de Laboratório de Sistemas Computacionais: Engenharia de Sistemas. O sistema completo compreende o desenvolvimento da arquitetura da plataforma de hardware, a definição de uma linguagem de programação, o projeto de um compilador, a definição de um sistema operacional e um processo de comunicação em rede entre dois ou mais sistemas computacionais, o qual acontecerá no último laboratório da trajetória acadêmica do aluno sendo denominado Laboratório de Sistemas Computacionais: Redes de Computadores, previsto para ocorrer no nono semestre da matriz curricular do curso.

REFERÊNCIAS

- [1] The Joint Task Force on Computing Curricula. IEEE Computer Society and Association for Computing Machinery. Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering, 2016.
- [2] SBC - Sociedade Brasileira de Computação. Currículo de referência da SBC para cursos de Graduação em Bacharelado em Ciência da Computação e Engenharia de Computação, 2005.
- [3] Parecer CNE/CES n. 136, de 09 de março de 2012, que trata sobre as Diretrizes Curriculares Nacionais para os cursos de graduação em Computação.
- [4] Unifesp. Projeto Pedagógico do Curso de Graduação do Bacharelado em Engenharia de Computação, fevereiro de 2015.
- [5] Tiago de Oliveira, Luiz E. G. Martins, Denise Stringhini, Álvaro L. Fazenda e Fábio A. M. Cappabianco, "O ensino e aprendizado de Arquitetura e Organização de Computadores num currículo de Engenharia de Computação estruturado em uma abordagem prática e sistêmica", *International Journal of Computer Architecture Education (IJCAE)*, vol. 6, n. 1, pp. 27-36, Dezembro 2017.
- [6] Jayantha Herath, Sarnath Ramnath, Ajantha Herath, and Susantha Herath. An active learning environment for intermediate computer architecture courses. In *Proceedings of the 2002 workshop on Computer architecture education: Held in conjunction with the 29th International Symposium on Computer Architecture (WCAE '02)*. ACM, New York, NY, USA, 2002.
- [7] O. Arbelaitz, J. I. Martín and J. Muguerza. Analysis of Introducing Active Learning Methodologies in a Basic Computer Architecture Course. *IEEE Transactions on Education*, Vol. 58, No. 2, pp. 110-116, May 2015.
- [8] B. Nikoli, V. Milutinovic, "A survey and evaluation of simulators suitable for teaching courses in computer architecture and organization", *IEEE Transactions on Education*, Vol. 52, No. 4, November 2009.
- [9] Z. Sridhar, "GNUSim8085, versão 1.3.7", <https://launchpad.net/gnusim8085>. Acesso em 26 de julho de 2017.
- [10] K. Vollmar and P. Sanderson. Mars: an education-oriented MIPS assembly language simulator. In *ACM SIGCSE Bulletin*, volume 38, pages 239-243. ACM, 2006.
- [11] P. S. Magnusson et Al, "Simics: A Full System Simulation Platform" In: *Computer IEEE*, v. 35-2, pg. 50-58, fev 2002.
- [12] Gabriel P. Silva e José Antônio dos S. Borges, "SimuS: Um simulador para o Ensino de Arquitetura de Computadores", *International Journal of Computer Architecture Education (IJCAE)*, Vol. 5, No. 1, p. 07-12, Dezembro 2016.
- [13] Paulo V. Vieira, André L. A. Raabe e Cesar A. Zeferino, "Projeto BIP: Impactos de 10 anos de Uso de uma Proposta Interdisciplinar de Ensino de Computação", *International Journal of Computer Architecture Education (IJCAE)*, Vol. 5, No. 1, p. 32-37, Dezembro 2016.
- [14] Liana Duenha e Rodolfo Acevedo, "Utilização dos simuladores do MPSoCBench para o ensino e aprendizagem de Arquitetura de Computadores", *International Journal of Computer Architecture Education (IJCAE)*, Vol. 5, No. 1, p. 26-31, Dezembro 2016.
- [15] Weber, R. F. Fundamentos de Arquitetura de Computadores. 2. ed. Porto Alegre. Sagra-Luzzatto, 2004.
- [16] C. M. Kellett, "A Project-Based Learning Approach to Programmable Logic Design and Computer Architecture", *IEEE Transactions on Education*, Vol. 55, No. 3, pp. 378-383, Aug. 2012.
- [17] Ma De Los Angeles Cifredo-Chacón, Ángel Quirós-Olozábal, and José María Guerrero-Rodríguez, "Computer architecture and FPGAs: A learning-by-doing methodology for digital-native students", *Journal Computer Applications in Engineering Education*, Vol. 23, No. 3, pp. 464-470, May 2015.
- [18] E. Larraza-Mendiluze, N. Garay-Vitoria, J. I. Martín, J. Muguerza, T. Ruiz-Vázquez, I. Soraluze, J. F. Lukas and K. Santiago, "Game-Console-Based Projects for Learning the Computer Input/Output Subsystem", *IEEE Transactions on Education*, Vol. 56, No. 4, pp. 453-458, Nov. 2013.
- [19] Edurne Larraza-Mendiluze, Nestor Garay-Vitoria, Iratxe Soraluze, José Martín, Javier Muguerza, and Txelo Ruiz-Vázquez, "Using a Real Bare Machine in a Project-Based Learning Environment for Teaching Computer Structure: An Analysis of the Implementation Following the Action Research Model", *Journal ACM Transactions on Computing Education*, Vol. 16, No. 3, pp. 13:1-13:17, June 2016.
- [20] Leandro S. G. de Carvalho e Fabíola G. Nakamura, "Práticas de Ensino na Disciplina de Circuitos Lógicos", *International Journal of Computer Architecture Education (IJCAE)*, Vol. 2, No. 1, p. 09-12, Dezembro 2013.
- [21] Vinicius B. da Silva e Jean F. P. Cheiran, "Análise do uso de microcontroladores como ferramenta de apoio ao ensino-aprendizagem de Arquitetura de Computadores", *International Journal of Computer Architecture Education (IJCAE)*, Vol. 4, No. 1, p. 01-04, Dezembro 2015.

- [22] Wagner L. A. de Oliveira, Anfranserai M. Dias, Antonio L. Apolinário Jr., Angelo A. Duarte e Tiago de Oliveira. Aplicando PBL no Ensino de Arquitetura de Computadores. In: PBL2010 International Conference, São Paulo. PBL 2010 - Congresso Internacional, 2010.
- [23] Ricardo O. Duarte e Pedro F. D. Garcia, Metodologia de Ensino Orientada para Projetos e Criação de Material Didático: Um relato de caso da disciplina sistemas, processadores e periféricos Laboratório, da Escola de Engenharia da UFMG. Revista Docência do Ensino Superior, v. 1, p. 01-18, 2011.
- [24] Wagner L. A. de Oliveira, Anfranserai M. Dias, Antonio L. Apolinário Jr., Angelo A. Duarte e Tiago de Oliveira. Ensino de Arquitetura de Computadores: Uma Abordagem Utilizando a Metodologia de Aprendizagem Baseada em Problemas. In: Carlos Augusto Paiva da Silva Martins; Philippe Olivier Alexandre Navaux; Rodolfo Jardim de Azevedo; Sérgio Takeo Kofuji. (Org.). Arquitetura de Computadores: educação, ensino e aprendizado. 1ed.: Sociedade Brasileira de Computação - SBC, 2012, p.34-73.
- [25] Ricardo O. Duarte e Pedro F. D. Garcia, "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, No. 1, p. 11-20, Dezembro 2012.
- [26] Thomas L. Floyd. Digital Fundamentals (11th ed.), Pearson, 2014.
- [27] Ronald J. Tocci, Neal S. Widmer and Gregory L. Moss. Digital Systems: Principles and Applications (12th ed.), Pearson, 2017.
- [28] M. Morris Mano and Michael D. Ciletti. Digital Design: With an Introduction to the Verilog HDL, VHDL, and SystemVerilog (6th ed.), Pearson, 2017.
- [29] Altera. Kit Educacional DE2-115. Disponível em: <<https://www.altera.com/support/training/university/boards.html#de2-115>>. Acesso em 12 de outubro de 2017.
- [30] Altera. Quartus Prime Standard Edition Handbook, 2017. Disponível em: <https://www.altera.com/en_US/pdfs/literature/hb/qts/qts-qps-handbook.pdf>. Acesso em 12 de outubro de 2017.
- [31] J. L. Hennessy and D. A. Patterson, Computer Architecture: A Quantitative Approach, 5rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011.
- [32] M. Morris Mano, Charles R. Kime and Tom Martin. Logic and Computer Design Fundamentals (5th ed.). Editora Pearson, 2015.
- [33] Skrien, Dale. CPU Sim 3.1: A tool for simulating computer architectures for computer organization classes. ACM Journal of Educational Resources in Computing, Dezember 2001.
- [34] Jovan Djordjevic, Bosko Nikolic and Ar Milenkovic. Flexible web-based educational system for teaching computer architecture and organization. IEEE Transactions on Education, Vol. 48, No. 2, pp. 264-273, May 2005.
- [35] William Yurcik and Larry Brumbaugh. A web-based little man computer simulator. In Proceedings of the thirty-second SIGCSE technical symposium on Computer Science Education(SIGCSE '01). ACM, New York, NY, USA, March 2001.
- [36] Vieira, Paulo; Zeferino, Cesar e Raabe, Andre. Avaliação Empírica da Proposta Interdisciplinar de Uso dos Processadores BIP. Revista Brasileira de Informática na Educação, Volume 23, Número 2, 2015.
- [37] Paulo V. Vieira, André L. A. Raabe e Cesar A. Zeferino, "Projeto BIP: Impactos de 10 anos de Uso de uma Proposta Interdisciplinar de Ensino de Computação", International Journal of Computer Architecture Education (IJCAE), Vol. 5, No. 1, p. 32-37, Dezembro 2016.
- [38] Edson Borin e Rafael Auler, "Uma abordagem para o ensino de linguagem de montagem, arquitetura e organização de computadores", International Journal of Computer Architecture Education (IJCAE), Vol. 2, No. 1, p. 21-24, Dezembro 2013.
- [39] Gabriel P. Silva e José Antônio dos S. Borges, "SimuS: Um simulador para o Ensino de Arquitetura de Computadores", International Journal of Computer Architecture Education (IJCAE), Vol. 5, No. 1, p. 07-12, Dezembro 2016.
- [40] Weber, R. F. Fundamentos de Arquitetura de Computadores. 2. ed. Porto Alegre. Sagra-Luzzatto, 2004.
- [41] Moodle. Modular Object-Oriented Dynamic Learning Environment (Moodle), 2017. Disponível em: <<https://moodle.org>>. Acesso em 13 de setembro de 2017.
- [42] B. Nikoli, V. Milutinovic, "A survey and evaluation of simulators suitable for teaching courses in computer architecture and organization", IEEE Transactions on Education, Vol. 52, No. 4, November 2009.
- [43] William Stallings. Computer Organization and Architecture: Designing for Performance (10th ed.). Editora Pearson, 2016.
- [44] Andrew S. Tanenbaum and Todd Austin. Structured Computer Organization (6th ed.). Editora Prentice Hall, 2012.
- [45] Sajjan G. Shiva. Computer Organization, Design and Architecture (5th ed.), CRC Press, 2013.2017.