

BIP IV: Especificação e Suporte na Ferramenta Bipide

Paulo R. M. Rech, Paulo V. Vieira, Cesar A. Zeferino e André L. A. Raabe
Universidade do Vale do Itajaí - Univali
Rua Uruguai, 458, C.P. 360, 88302-202, Itajaí, SC, BRASIL
{paulorech; pauloviniccus, zeferino, raabe}@univali.br

Resumo

Para reduzir a abstração de conceitos de lógica de programação nos primeiros semestres de cursos da área de computação, a família de processadores BIP e o ambiente de desenvolvimento integrado Bipide foram disponibilizados na primeira versão da ferramenta Bipide apresentam algumas limitações, pois não oferecem suporte a chamadas de procedimentos e a interação com o usuário através de operações de entrada e saída, o que impossibilita sua utilização em problemas mais complexos ou que necessitem de alguma interação. Neste contexto, este trabalho apresenta o desenvolvimento do processador BIP IV, que estende as características dos processadores BIP, agregando novas funcionalidades e aumentando a abrangência de sua utilização. Também foram implementadas modificações na IDE Bipide a fim de suportar o uso do processador BIP IV. Com isto, aumenta-se a interação do aluno com a ferramenta e seu uso na resolução de problemas mais complexos.

1. Introdução

Com o objetivo de auxiliar o processo de ensino-aprendizagem nos semestres iniciais de cursos de Ciência da Computação, foi proposta uma abordagem interdisciplinar envolvendo conceitos de arquitetura e organização de computadores nas disciplinas ligadas a aprendizagem de programação [1]. Este enfoque parte de um problema frequente relatado por pesquisadores da área, os quais apontam o alto nível de abstração envolvido nos conceitos de programação como um dos principais aspectos que dificultam a aprendizagem inicial de programação. Acredita-se que a redução da abstração envolvida nesses conceitos pode diminuir os problemas de aprendizagem apresentados pelos alunos iniciantes [2,3].

Para atender esse objetivo, foi desenvolvida uma família de processadores de baixa complexidade denominada BIP (Basic Instruction-set Processor).

Para facilitar a utilização desses processadores no ensino foi desenvolvida também uma ferramenta educacional chamada Bipide [4].

A família BIP foi concebida em níveis de complexidade de arquitetura e organização crescentes, sendo que o processador BIP I oferece apenas instruções de aritmética e de acesso à memória [5]. O BIP II inclui instruções de desvio condicional e incondicional [6], enquanto que o BIP III acrescenta suporte para operações de lógica bit-a-bit. O μ BIP, por sua vez, foi desenvolvido com o intuito de ser utilizado no ensino de Sistemas Embarcados, agregando periféricos e funcionalidades típicas de microcontroladores aos processadores BIP [7].

A ferramenta Bipide consiste em um ambiente de desenvolvimento integrado (IDE - Integrated Development Environment) baseado na arquitetura dos processadores BIP I e BIP II, oferecendo funcionalidades que auxiliam na compreensão e utilização desses processadores no ensino. Nesse ambiente, é possível criar pequenos algoritmos em Portugol (pseudolinguagem utilizada para facilitar o ensino de algoritmos) e executá-los passo a passo. A interface do Bipide utiliza animações gráficas que ilustram a organização do processador e destacam a linha de código Portugol sendo executada e o *assembly* correspondente.

Esta abordagem didática tem sido utilizada ao longo de três semestres letivos e tem trazido resultados positivos conforme mencionado em [4]. No entanto, os processadores BIP I, II e III não suportam chamadas de procedimentos ou a interação com o usuário através de operações de entrada-e-saída, o que restringe sua utilização a algoritmos muito simples. Esse aspecto tem gerado limitações quanto ao uso da abordagem para o ensino de conceitos mais abrangentes da aprendizagem de programação. Analisando a bibliografia de apoio utilizada nas disciplinas da área de algoritmos, percebe-se que muitos autores [8, 9] adotam problemas que necessitam interações como entrada-e-saída de dados. Baseado nisso, conclui-se que a interação, via entrada-e-saída de dados, é um

componente importante para auxiliar a despertar mais interesse por parte do aluno.

Nesse contexto, a solução apresentada neste trabalho consiste na criação de um novo integrante da família BIP, o BIP IV, que incorpora instruções de entrada-e-saída (E/S), manipulação de vetores e chamadas de procedimentos. Para suportar as novas funcionalidades do BIP IV, várias alterações no ambiente Bipide foram necessárias, incluindo mudanças na estrutura da pseudolinguagem Portugol.

Entre os benefícios da implementação em questão estão: (i) maior interação do aluno com a ferramenta; (ii) aprendizagem de novas funcionalidades e seu funcionamento no processador; (iii) extensão do uso da ferramenta para disciplinas de semestres posteriores aos já aplicados, uma vez que permite algoritmos mais complexos; e (iv) a continuidade da pesquisa relacionada à abordagem didática interdisciplinar.

2. Família de processadores BIP

Uma prática essencial para o ensino de computação é a experimentação realizada em laboratórios, onde os estudantes podem aplicar o conhecimento teórico adquirido em sala de aula na solução de problemas práticos relacionados às disciplinas estudadas. Nesse aspecto, a escolha de um modelo de processador com características pedagógicas que possibilitem sua utilização em um contexto educacional é de extrema importância para a demonstração dos conceitos relacionados à arquitetura e à organização do computador. Dessa forma, alguns autores e professores optam pela utilização de processadores reais e comerciais, enquanto outros preferem usar um modelo hipotético de processador disponível na literatura [10, 11].

Outra abordagem utilizada é o desenvolvimento de um modelo específico de processador que atenda as necessidades pedagógicas do curso [1, 10]. Os processadores BIP foram desenvolvidos por pesquisadores do Laboratório de Sistemas Embarcados e Distribuídos e do Grupo de Informática na Educação da UNIVALI (Universidade do Vale do Itajaí) com o intuito de estabelecer uma relação entre a programação de alto nível e sua representação em hardware, auxiliando o aprendizado nas disciplinas “Arquitetura e Organização de Computadores” e “Algoritmos e Programação”, ministradas nas fases iniciais dos cursos de graduação na área de Computação [4].

Os processadores BIP são baseados na arquitetura do processador PIC (Programmable Intelligent Computer), utilizam uma arquitetura orientada a acumulador e não possuem banco de registradores. Todas as instruções são baseadas em um único formato

de instrução composto por apenas dois campos: o código de operação (de 5 bits) e o operando (de 11 bits) [6]. O acumulador é utilizado como operando implícito da maioria das instruções.

A organização dos processadores BIP é baseada em uma estrutura de memória do tipo Harvard, com memórias separadas para dados e programa [6]. O processador é estruturado em dois blocos: o Controle e o Caminho de Dados. O primeiro é responsável por buscar e decodificar as instruções da memória de programa, gerando sinais de comando para a execução das instruções pelo Caminho de Dados.

Os processadores BIP representam níveis incrementais de complexidade de arquitetura e de organização. O processador BIP I [5] suporta instruções aritméticas e de transferência. O processador BIP II [6] estende o BIP I agregando instruções de desvio condicional e incondicional necessárias para a implementação de laços de repetição. O processador BIP III agrega instruções de lógica bit-a-bit, enquanto que o μ BIP [7] adiciona funcionalidades típicas de microcontroladores.

3. O ambiente integrado Bipide

O ambiente Bipide foi concebido pela necessidade de um simulador educacional que permitisse relacionar os conceitos de lógica de programação com aspectos definidos no hardware dos processadores BIP I e BIP II. Este ambiente possibilita a criação de algoritmos em Portugol e sua execução passo a passo. A ferramenta permite visualizar o código correspondente ao programa em linguagem *assembly* e o estado dos componentes da organização do processador através de animações que ilustram o funcionamento interno do processador [4].

O subconjunto da linguagem Portugol utilizada no ambiente Bipide foi definido segundo as características arquiteturais dos processadores BIP I e BIP II. Foram disponibilizadas estruturas de desvio e de laços de repetição, além dos elementos básicos necessários para estruturar um algoritmo [4].

O Bipide apresenta um compilador integrado, construído através da ferramenta ANTLR3 (ANother Tool for Language Recognition V.3) [12]. Foram construídos os analisadores léxico e sintático e definidas as ações semânticas para geração do código em linguagem de montagem e tratamento de erros. Utilizou-se também a IDE ANTLRWorks, a qual é composta por um ambiente de desenvolvimento de gramáticas para o ANTLR3.

O módulo de simulação do Bipide permite a visualização simultânea dos programas em Portugol e *assembly*, assim como a organização do processador, o

que contribui para a redução das abstrações apresentadas nos conceitos de programação [4].

Para o desenvolvimento do Bipide, utilizou-se a tecnologia WPF (Windows Presentation Foundation) [13] e a ferramenta Expression Blend [14]. Essas tecnologias possibilitam a criação de aplicações com interfaces ricas, agregando ao Bipide funcionalidades mais intuitivas e recursos de animação que permitem uma maior interatividade com o usuário.

4. Proposta interdisciplinar

As iniciativas de criação de processadores com objetivo didático normalmente buscam apoiar as disciplinas diretamente envolvidas, como “Arquitetura e Organização de Computadores”. A criação da família de processadores BIP, com suas diferentes versões, permitiu trabalhar níveis de complexidade diferentes e com isso explorar os conceitos de arquitetura e organização de computadores em diferentes disciplinas do primeiro, segundo, terceiro e sétimo semestres do curso de Ciência da Computação da UNIVALI. De forma semelhante, esse aspecto facilitou o estabelecimento de relações entre as diferentes disciplinas envolvidas, numa abordagem interdisciplinar. [1].

Sabe-se que a utilização de processadores simplificados no ensino, como é o caso dos processadores BIP, auxilia a melhorar a compreensão de conceitos relacionados à aprendizagem inicial de programação reduzindo seu grau de abstração. Essa abordagem pode facilitar a aprendizagem dos alunos que apresentam problemas em lidar com abstrações [1].

Motivados por essa proposta, uma série de trabalhos foram desenvolvidos, disponibilizando novos recursos e ferramentas a serem utilizados na abordagem citada, entre os quais a criação de um microcontrolador baseado na arquitetura BIP (o μ BIP) [7] e o ambiente de desenvolvimento Bipide [4], o qual inclui simuladores para os processadores BIP I e BIP II.

A fim de dar continuidade a esta proposta interdisciplinar, um novo integrante da família BIP foi desenvolvido. Este processador, designado BIP IV, possui funcionalidades intermediárias aos processadores BIP III e μ BIP, permitindo operações de E/S, manipulação de vetores e o uso de subrotinas. Estes recursos suprem uma necessidade apresentada pelos processadores disponibilizados anteriormente, no que diz respeito às restrições arquiteturais impostas pelas versões anteriores.

O suporte a este novo processador no ambiente Bipide também foi implementado, permitindo ao aluno compreender as novas funcionalidades através de

simulação, bem como a representação dessas em linguagem de baixo nível.

5. Arquitetura do BIP IV

A especificação do BIP IV aproveita algumas instruções presentes no μ BIP e estende o BIP III. Além de possibilitar operações de entrada-e-saída e chamada de procedimentos, foram incluídas instruções de deslocamento e de manipulação de vetores.

Para suportar as operações de E/S, foi adotado o método de E/S mapeada em memória, utilizado no μ BIP, não sendo necessárias grandes modificações na arquitetura e na organização do processador [7]. Os registradores IN_PORT (entrada) e OUT_PORT (saída) são utilizados para acesso à interfaces de E/S. A Tabela 1 resume a arquitetura do BIP IV, identificando as classes de instrução suportadas pelos diferentes processadores da família (do BIP I ao BIP IV).

O código de operação dos processadores BIP é composto por 5 bits, permitindo identificar até 32 instruções. O conjunto de instruções do BIP IV é formado por 28 instruções. Do total de 32 códigos de operação possíveis, restam três para novas instruções, sendo que um é utilizado para representar a instrução de retorno de interrupção no μ BIP [7]. Ressalta-se que as instruções acrescentadas ao BIP IV foram herdadas do processador μ BIP.

A organização do BIP IV foi baseada na organização do processador BIP III com a adição de alguns componentes presentes no μ BIP para suportar manipulação de vetores e chamada de sub-rotinas, além dos pinos de E/S (registradores IN_PORT e OUT_PORT). Estes componentes incluem uma pilha de suporte a procedimentos, onde será salvo o endereço de memória da instrução seguinte à chamada da sub-rotina e uma unidade de manipulação de vetores.

6. Suporte ao BIP IV no Bipide

Para implementar o suporte ao BIP IV na ferramenta Bipide, foram levantadas características relacionadas à interfaces de E/S de simuladores de arquitetura similares. Essas características permitiram definir aspectos que orientaram a construção da interface de E/S utilizada no Bipide. A seguir descreve-se a análise realizada e a implementação do suporte ao BIP IV na ferramenta Bipide.

Tabela 1. Resumo da Arquitetura do BIP IV

Tamanho da palavra de dados	16 bits																																	
Tipos de dados	Inteiro de 16 bits com sinal –32768 a +32767																																	
Tamanho da palavra de instrução	16 bits																																	
Formato de instrução	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center; width: 12.5%;">15</td> <td style="text-align: center; width: 12.5%;">14</td> <td style="text-align: center; width: 12.5%;">13</td> <td style="text-align: center; width: 12.5%;">12</td> <td style="text-align: center; width: 12.5%;">11</td> <td style="text-align: center; width: 12.5%;">10</td> <td style="text-align: center; width: 12.5%;">9</td> <td style="text-align: center; width: 12.5%;">8</td> <td style="text-align: center; width: 12.5%;">7</td> <td style="text-align: center; width: 12.5%;">6</td> <td style="text-align: center; width: 12.5%;">5</td> <td style="text-align: center; width: 12.5%;">4</td> <td style="text-align: center; width: 12.5%;">3</td> <td style="text-align: center; width: 12.5%;">2</td> <td style="text-align: center; width: 12.5%;">1</td> <td style="text-align: center; width: 12.5%;">0</td> </tr> <tr> <td colspan="8" style="text-align: center;">Cód. Operação</td> <td colspan="8" style="text-align: center;">Operando</td> </tr> </table>		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Cód. Operação								Operando							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																			
Cód. Operação								Operando																										
Modos de endereçamento	<u>Direto</u> : o Operando é um endereço da memória <u>Imediato</u> : o Operando é uma constante <u>Indireto</u> : o Operando é um endereço base de um vetor que é somado ao INDR para o cálculo de um endereço efetivo da memória de dados																																	
Registradores	<u>ACC</u> : acumulador <u>PC</u> : contador de programa <u>STATUS</u> : registrador de Status <u>INDR</u> : registrador de índice <u>SP</u> : apontador do topo da pilha																																	
Classe de instução (Processadores)	<u>Controle</u> : HLT <u>Armazenamento</u> : STO <u>Carga</u> : LD e LDI <u>Aritmética</u> : ADD, ADDI, SUB e SUBI	(BIP I-IV)																																
	<u>Desvio</u> : BEQ, BNE, BGT, BGE, BLT, BLE e JMP	(BIP II-IV)																																
	<u>Lógica booleana</u> : AND, OR, XOR, ANDI, ORI, XORI e NOT	(BIP III-IV)																																
	<u>Deslocamento Lógico</u> : SLL e SRL <u>Manipulação de vetor</u> : LDV e STOV <u>Suporte a procedimentos</u> : RETURN e CALL	(BIP IV)																																

6.1. Análise de trabalhos similares

A análise de trabalhos similares considerou simuladores de arquitetura de processadores que permitam E/S. Os simuladores analisados compreendem o NeanderWin [15], GNUsim8085 [16], MipsIt [17], ABACUS [18] e WinMIPS64 [19]. A seguir descrevem-se as principais características observadas nos trabalhos citados.

Dentre os simuladores estudados, alguns apresentaram interfaces de E/S baseadas em controles simples, como representações de chaves e LEDs, outros apresentam também visores ilustrando valores em decimal ou em hexadecimal. Essas interfaces possibilitam resolver problemas simples, onde a entrada e a saída são números, ou até problemas que simulam o controle de algum dispositivo acoplado ao processador e que tenham a necessidade de ler ou escrever em determinados bits da porta. Com isso, optou-se representar no Bipide uma interface baseada em chaves e LEDs, juntamente com opções de edição e leitura em decimal.

A Tabela 2 apresenta a comparação entre os sistemas analisados e a proposta implementada no Bipide. Outras características relacionadas ao desenvolvimento de programas em linguagem de alto nível e à simulação da organização também foram levantadas.

Observa-se na comparação realizada, que os sistemas Bipide e MipsIt apresentam as mesmas características, sendo os únicos a permitir programação em linguagem de alto nível. No entanto, a arquitetura do processador MIPS, ilustrada no MipsIt, possui complexidade acima do desejável para a apresentação de conceitos iniciais de programação. Já a arquitetura do BIP, tendo sido criada para propósito didático, possui a simplicidade necessária para ilustração desses conceitos. Percebe-se ainda que o ambiente Bipide possibilita a simulação da organização dos processadores BIP, permitindo que o aluno visualize graficamente, por meio de animações, o funcionamento dos componentes do processador. Considera-se essa característica um grande diferencial da ferramenta, contribuindo para o aprendizado.

Tabela 2. Características dos Sistemas Analisados

Simulador	Simula Organização	Suporta linguagem de alto nível	Interface de E/S
NeanderWin	Não	Não	Chaves e visor
GNUSim8085	Não	Não	Campo Editável
MipsIt	Sim	Sim	Chaves e LEDs
ABACUS	Não	Não	Chaves e LEDs
WinMIPS64	Não	Não	Terminal
Bipide	Sim	Sim	Chaves e LEDs

6.2. Implementação

Para contemplar o suporte ao BIP IV, o ambiente Bipide foi alterado de modo a suportar as seguintes características dos processadores BIP III e BIP IV: (i) operações de E/S; (ii) utilização de vetores; (iii) chamada de sub-rotinas com passagem de parâmetros; e (iv) operações de lógica. Para isso, as alterações feitas no Bipide incluem mudanças na gramática Portugol e novas verificações nas ações semânticas, assim como alterações na geração de código *assembly* e mudanças no simulador.

Em sua versão anterior, o simulador do Bipide permitia escolher entre os processadores BIP I e BIP II, refletindo em mudanças nas instruções disponíveis, ilustrações e animações do processador. Para manter esse conceito, o simulador foi alterado de modo a permitir escolher, além do BIP I e BIP II, os processadores BIP III e BIP IV. Entre as alterações feitas na IDE, destaca-se ainda a interface de E/S, ilustrada pelo item 8 da Figura 1, onde é possível visualizar o conteúdo dos registradores \$in_port e \$out_port. Pode-se ainda atribuir valores ao registrador \$in_port através das chaves disponíveis na interface ou digitando-os em notação decimal.

Os recursos do Bipide, ilustrados na Figura 1, são descritos a seguir:

1. Simulações: Disponibiliza botões que permitem controlar a simulação, incluindo a simulação passo a passo;
2. Velocidade: Permite controlar a velocidade da simulação do programa;
3. Portugol: Exibe o programa Portugol que está sendo simulado destacando a linha em execução;
4. Assembly: Exibe o código *assembly* gerado pelo compilador. A linha do programa em execução é destacada bem como todo o conjunto de instruções que correspondem à linha de código em alto nível destacada na janela Portugol;
5. Organização do Processador: Exibe a imagem da organização do processador (BIP I, BIP II, BIP III ou BIP IV), bem como as animações que representam as instruções em execução. Na ilustração é exibida a organização do BIP IV;
6. Registradores: Exibe os valores dos registradores do processador durante a simulação do programa;
7. Descrição: Exibe o nome e a descrição de cada instrução executada; e
8. Interface de E/S: Permite ao usuário visualizar e informar valores em decimal ou binário.

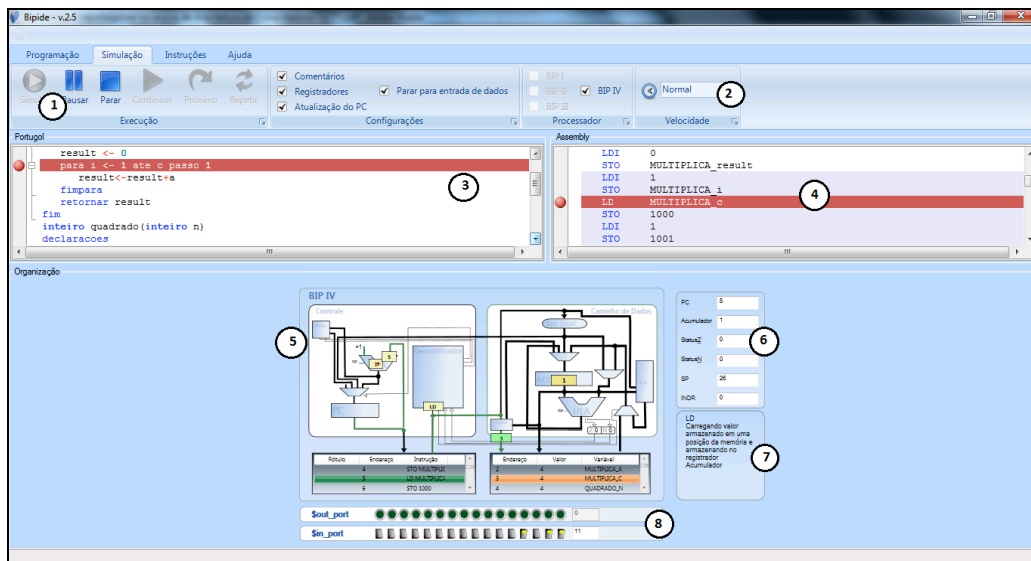


Figura 1. Interface do Bipide

A interface de E/S adotada permite a resolução de problemas onde a entrada e a saída são números. Além disto, possibilita simular o controle de dispositivos periféricos acoplado ao processador, que necessitam ler ou escrever em determinados bits da porta. Dessa forma, imagina-se um conjunto de problemas que podem ser ilustrados por esse ambiente, os quais vão desde o desenvolvimento de um algoritmo que realiza a soma de dois números inteiros até o controle da rotação de um motor de passo (onde a ativação de determinados bits em uma sequência determinada de passos são necessários para produzir o movimento de rotação do dispositivo).

7. Conclusões

Este artigo apresentou o desenvolvimento do processador BIP IV e a implementação de alterações realizadas na IDE Bipide para suportar este processador. As alterações realizadas na IDE Bipide ampliaram seu potencial pedagógico, permitindo o desenvolvimento de algoritmos mais complexos do que os suportados pela versão anterior da ferramenta.

Esta amplitude de possibilidades favorece a adoção do ambiente Bipide como software de apoio em outras disciplinas dos cursos da área de computação, que até então tinham dificuldades na utilização da ferramenta devido às restrições impostas pela arquitetura dos processadores BIP disponíveis anteriormente. Isso amplia o potencial pedagógico de uma abordagem educacional interdisciplinar que busca reduzir o nível de abstração envolvido em conceitos de programação, colaborando para a melhoria do ensino de computação.

Com as novas funcionalidades e possibilidades de exploração disponibilizadas, pretende-se obter novas evidências sobre a redução dos problemas de aprendizagem associados à abstração dos conceitos abordados. Pretende-se ainda promover o uso dos processadores BIP e da ferramenta Bipide em outras disciplinas do curso, como Compiladores e Circuitos Digitais. Outros trabalhos futuros incluem o suporte a diferentes linguagens de alto nível, como C e Java, e a realização de experimentos práticos que permitam avaliar empiricamente os benefícios obtidos com as novas funcionalidades disponibilizadas.

Referências

- [1] C. A. Zeferino et al., Um Enfoque Interdisciplinar no Ensino de Arquitetura de Computadores. In: *Arquitetura de Computadores: educação, ensino e aprendizado*. Martins, C; Navaux P.; Azevedo, R.; Kofuji, S. (Org.). No Prelo, 2011.
- [2] M. McCracken et al., A Multi-National, Multi-Institutional Study of Assessment of Programming Skills of First-year CS Students. *SIGCSE Bulletin, USA*, n. 33, v. 4, p. 125-140, 2001.
- [3] R. Lister et al. A multi-national study of reading and tracing skills in novice programmers. *ACM SIGCSE Bulletin, USA*, v. 36, n. 4, p. 119-150, dez. 2004.
- [4] P. V. Vieira, A. L. A. Raabe, C. A. Zeferino, Bipide – ambiente de desenvolvimento integrado para a arquitetura dos processadores BIP. *Revista Brasileira de Informática na Educação*, Vol. 18, No 1, 2010.
- [5] D. Morandi et al., Um processador básico para o ensino de conceitos de arquitetura e organização de computadores. *Hifên, Uruguaiana*, v. 30, p. 73-80, 2006.
- [6] D. Morandi, A. L. A. Raabe, C. A. Zeferino, Processadores para Ensino de Conceitos Básicos de Arquitetura de Computadores. In: *WEAC 2006, Proceedings...* Porto Alegre: SBC, 2006. p. 17-24.
- [7] M. C. Pereira, C. A. Zeferino, uBIP: a simplified microcontroller architecture for education in embedded systems design. In: *IP Based Electronic System Conference & Exhibition - IP 08, 2008, Grenoble. Proceedings...* Grenoble : Design and Reuse, 2008. p. 193-197.
- [8] J. A. N. G. Manzano, J. F. Oliveira, *Algoritmos: lógica para desenvolvimento de programação de computadores*. 17. ed. São Paulo, SP: Érica, 2005.
- [9] N. Ziviani., *Projeto de algoritmos: com implementações em Java e C++*. São Paulo, SP: Thomson, 2007.
- [10] B. Nikolic et al., A Survey and Evaluation of Simulators Suitable for Teaching Courses in Computer Architecture and Organization. *IEEE Transactions on Education, USA*, v. 52, n. 4, p. 449-458, 2009.
- [11] A. Clements, Selecting a processor for teaching computer architecture. *Microprocessor and Microsystems, USA*, v. 23, n. 5, p. 281-290, 1999.
- [12] P. Terence, *The Definitive ANTLR Reference, The Pragmatic Bookshelf*, Dallas, Texas, 1997.
- [13] Microsoft Corporation. *Windows Presentation*, 2011.
- [14] Microsoft Corporation. *Expression Blend*, 2011.
- [15] J. A. S. Borges, G. P. Silva, NeanderWin – um simulador didático para uma arquitetura do tipo Acumulador. In: *WEAC 2006. Proceedings...* Porto Alegre: SBC, 2006.
- [16] GNUsim8085. *GNUsim8085*, 2003.
- [17] M. Brorsson, MipsIt: A Simulation and Development Environment Using Animation for Computer Architecture Education. In: *Workshop on Computer Architecture Education, 2002, Anchorage, Alaska. Proceedings...* New York, NY: ACM, 2002.
- [18] R. M. Ziller, *ABACUS*, 1999.
- [19] M. Scott, *WinMIPS64*, 2010.