

## **Processadores para Ensino de Conceitos Básicos de Arquitetura de Computadores**

Diana Morandi, André Luis Alice Raabe, Cesar Albenes Zeferino  
*Universidade do Vale do Itajaí – UNIVALI*  
*Rua Uruguai, 458 – Caixa Postal 360 – 88302-202 – Itajaí – SC – Brasil*  
*{diana.morandi, raabe, zeferino}@univali.br*

### **Resumo**

*Alunos de primeiro ano de cursos de graduação em Computação apresentam algumas dificuldades de aprendizado nas disciplinas introdutórias da área de Programação por não conhecerem o significado físico das abstrações utilizadas. Em geral, esses conceitos são efetivamente compreendidos quando os alunos cursam disciplinas na área de Arquitetura de Computadores, o que, em geral, é feito no segundo ano do curso. Nesse contexto, este trabalho apresenta o desenvolvimento de processadores simplificados para uso no ensino de conceitos introdutórios a alunos de primeiro ano de cursos da área de Computação. Os modelos são simples o suficiente para facilitar o aprendizado, mas, ao mesmo tempo, são projetados para permitir a implementação física do processador e a integração com disciplinas das áreas de Eletrônica e de Compiladores. Neste artigo, são apresentados os dois modelos de processador e discutidos aspectos sobre o projeto, implementação e aplicação desses modelos no ensino.*

### **1. Introdução**

A compreensão do funcionamento da arquitetura do computador e de seu processador possui importância central na formação dos alunos dos cursos de graduação em Ciência da Computação. Isso porque, além de fornecer os conhecimentos básicos para possibilitar a inserção do aluno no contexto da pesquisa e desenvolvimento de hardware, auxilia a compreensão da necessidade e do papel do software básico, além de fornecer subsídios fundamentais para a aprendizagem e compreensão da lógica de programação.

Considerando esse último aspecto, é fato amplamente conhecido que alunos apresentam

dificuldades na aprendizagem de algoritmos e programação, em especial no primeiro ano do curso [1-5]. Essa dificuldade está relacionada, entre outros aspectos, à ausência de afinidade com o raciocínio lógico formal que é o fundamento para a capacidade de abstração dos alunos. Nesse sentido, o estudo da arquitetura do computador cria a possibilidade de estabelecer relações dos conceitos de programação com aspectos concretos do hardware, reduzindo assim a necessidade de abstração.

Nesse contexto, identifica-se uma falta de sincronia nos currículos tradicionais em que o estudo da arquitetura e da organização do computador ocorre depois do ensino da programação. Para contornar esse problema, em muitos cursos, costuma-se apresentar algumas noções básicas de arquitetura e organização de computadores aos alunos em disciplinas que fornecem uma introdução geral à Computação, tipicamente no primeiro ano do curso. São utilizados exemplos de processadores básicos que ilustram conceitos como: ciclo de instrução, conjunto de instruções, armazenamento de variáveis em memória, entre outros.

No entanto, essa abordagem normalmente apresenta dois problemas: (i) a falta de uma articulação adequada entre os professores das disciplinas introdutórias a fim de estabelecer uma sincronia na apresentação dos conteúdos que possa beneficiar a aprendizagem de programação; (ii) a limitação dos modelos utilizados para a apresentação dos conceitos básicos de arquitetura. Embora esses modelos muitas vezes sejam suficientes para o propósito das disciplinas introdutórias, eles apresentam restrições que não permitem avançar no estudo ou ainda promover uma relação interdisciplinar com conteúdos de outras áreas, como por exemplo, Eletrônica. Ou seja, são apresentados de forma estanque, sem possibilitar que o aluno complemente os conhecimentos adquiridos nessas disciplinas introdutórias ainda nas primeiras fases do curso.

Nesse contexto, este trabalho apresenta resultados de um projeto de pesquisa em andamento que visa desenvolver e disponibilizar uma série de processadores com um conjunto de instruções mínimo que auxilie o aprendizado de conceitos de arquitetura e organização de computadores por alunos de fases iniciais de cursos de graduação em Computação. O objetivo é que esses processadores sirvam de referência para a apresentação dos conceitos básicos necessários ao melhor entendimento das abstrações utilizadas nas disciplinas da área de Algoritmos e Programação. Ao mesmo tempo, busca-se disponibilizar uma especificação completa para que alunos de disciplinas da área de Eletrônica possam utilizar os conhecimentos apresentados nessas disciplinas para projetar e até mesmo construir um modelo físico do processador com o uso de ferramentas de projeto de circuitos, disponíveis a muitas instituições através de parcerias na forma de programas universitários.

As seções a seguir apresentam uma breve discussão sobre o uso de processadores no ensino de Computação, a descrição geral do projeto BIP e dos seus processadores. Continuando, são apresentados aspectos de implementação e da utilização do BIP no ensino, e algumas conclusões.

## 2. Uso de processadores no ensino

A escolha de modelos de processadores para o ensino de conceitos de arquitetura e organização de computadores é alvo de estudos freqüentes pelos educadores da área, como, por exemplo, no trabalho apresentado por Clements [6] que discute aspectos que devem ser levados em consideração na escolha de modelos de processadores a serem aplicados no ensino de graduação. Enquanto alguns autores e professores optam por utilizar modelos hipotéticos de processadores, outros adotam processadores reais e comerciais como referência para estudos de caso.

Para as fases iniciais de um curso de graduação, a seleção de processadores para o ensino concorrente da lógica de programação e de conceitos de arquitetura de computadores deve facilitar o estabelecimento de relações entre as abstrações lógicas necessárias à programação e à implementação dessas abstrações em hardware. Porém, os modelos de processadores tipicamente utilizados por professores de disciplinas introdutórias são abstratos demais e não permitem estabelecer essas relações. Uma alternativa seria utilizar modelos de processadores mais detalhados, como aqueles adotados nas disciplinas específicas da área de Arquitetura de Computadores. Porém, esses processadores são muito complexos para serem

aplicados em disciplinas do primeiro ano, e poucos são os livros texto da área que os descrevem propiciando uma integração entre a arquitetura do processador e a programação em alto nível.

Considerando livros adotados em universidades brasileiras, podem ser destacados dois exemplos associados a essa questão. O livro “Organização e projeto de computadores”, de Patterson e Hennessy [7], utiliza o processador MIPS como arquitetura base para apresentação de conceitos de arquitetura e organização. Esse livro se caracteriza por explorar a interface entre o hardware e software de modo a permitir ao aluno ligar os conceitos estudados com aqueles vistos nas disciplinas de programação. No entanto, o pouco embasamento dos alunos nas fases iniciais torna inadequado o uso de processadores com o grau de complexidade do MIPS. Um segundo exemplo é o livro “Fundamentos de arquitetura de computadores”, de Weber [8]. Nesse livro, o autor apresenta uma série de conceitos básicos, como, por exemplo, bases numéricas, sistemas de numeração e aritmética binária, e ilustra de uma série de processadores especificados para fins didáticos (Neander, Ahmes, Ramses e Cesar), propiciando o entendimento gradativo de diversos conceitos importantes da área de Arquitetura e Organização de Computadores. Porém, a abordagem utilizada não favorece a integração desses conceitos com os estudados nas disciplinas da área de Algoritmos e Programação, uma relação que se entende de fundamental importância no processo inicial de aprendizagem de um acadêmico em Computação.

Nesse contexto, deve-se buscar uma arquitetura simplificada que permita estabelecer uma relação entre as necessidades dos alunos que estão iniciando a programar e as representações em hardware correspondentes. É necessário realizar a identificação das principais fontes de incompreensão para os estudantes de modo a prover formas de minimizar suas dificuldades. Por exemplo, podem ser citadas algumas relações importantes entre a programação de alto nível e a sua implementação no hardware, sob a forma de conceitos de arquitetura e organização de computadores. Entre essas relações, destacam-se:

- Declaração de variável e alocação de memória;
- Constantes e operandos imediatos;
- Atribuição de variáveis e sua correspondência com as operações de acesso à memória; e
- Operações aritméticas e sua execução em hardware.

Logo, a escolha do processador para ser utilizado nas fases iniciais deve priorizar aspectos didáticos que favoreçam a compreensão das relações entre software e hardware numa abordagem multidisciplinar.

### 3. O Projeto BIP

O Projeto BIP (Basic Instruction-set Processor) insere-se no contexto acima e visa desenvolver uma série de processadores para o ensino de conceitos da área de Arquitetura de Computadores para alunos de fases iniciais de cursos de graduação em Computação, buscando prover a base necessária para compreensão das abstrações adotadas nas disciplinas da área de Algoritmos e Programação.

O projeto prevê a especificação de uma série de processadores básicos com recursos incrementais que permitam a apresentação de conceitos introdutórios de arquitetura e de organização e a ligação imediata com conceitos da área de Programação.

Ele também busca disponibilizar informações suficientes para que alunos de disciplinas mais avançadas possam implementar o processador utilizando conhecimentos de cadeiras da área de Eletrônica, ou ainda desenvolver ferramentas como compiladores e montadores em disciplinas dessa área.

O projeto encontra-se em desenvolvimento e, até este momento, foram especificadas duas gerações de processadores: BIP I e BIP II. A primeira é simples o suficiente para minimizar a dificuldade encontrada pelos alunos nos estudos iniciais de arquitetura de computadores, enquanto que a segunda estende a primeira, complementando sua cobertura em relação a conceitos de Programação. A Tabela 1 resume os conceitos suportados por cada geração especificada.

**Tabela 1.** Conceitos suportados pelo BIP

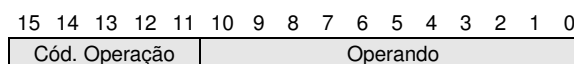
BIP I	BIP II
Noção de instrução	Operações relacionais
Variáveis e constantes	Desvios
Operação de atribuição	Laço de repetição
Operações aritméticas	
Desempenho	
Papel do compilador	

### 4. A arquitetura BIP I

O BIP I utiliza uma arquitetura simplificada baseada no microcontrolador PIC da Microchip [9]. Ele é uma máquina orientada a acumulador, o qual consiste no único registrador para armazenamento de dados. Todas as operações de transferência e aritmética envolvem esse registrador (operando implícito) e, em algumas operações aritméticas, é necessário buscar operandos diretamente da memória. Ou seja, o BIP I não é uma máquina load/store.

A arquitetura do BIP possui um conjunto de instruções altamente regular, com poucos modos de

endereçamento. Todas as instruções são baseadas no formato de instrução ilustrado na Figura 1, o qual inclui um código de operação de 5 bits e um operando de 11 bits. Esse operando pode ser o endereço de uma variável na memória (modo de endereçamento direto) ou um dado constante (modo de endereçamento imediato).



**Figura 1.** Formato de instrução

O código de operação de 5 bits permite a implementação de até 32 instruções. Embora o BIP I possua apenas 8 instruções, descritas a seguir, esse campo já foi dimensionado para viabilizar a inclusão de novas instruções nas futuras gerações. Considerando o tamanho do campo operando, o espaço para endereçamento na memória de dados é igual a  $2^{11}$ , ou seja, até 2048 variáveis, enquanto que qualquer constante é limitada ao intervalo  $-1024$  a  $+1023$ .

A palavra de dados é também de 16 bits e são suportados apenas dados do tipo inteiro com sinal. O suporte a outros tipos de dados será alvo de análise em etapas posteriores do projeto.

A arquitetura BIP I inclui três registradores: PC, IR e ACC. O contador de programa (PC – Program Counter) aponta para o endereço da próxima instrução a ser executada. O registrador de instrução (IR – Instruction Register) armazena a instrução que está em execução. O acumulador (ACC – Accumulator) é utilizado para armazenamento de dados durante a execução de uma operação de aritmética ou de atribuição. Ele pode servir tanto de fonte como de destinatário do resultado de uma operação. O IR e o ACC são registradores de 16 bits, enquanto que o PC pode ser dimensionado de acordo com o tamanho do código fonte a ser executado.

A Tabela 2 apresenta o conjunto de instruções do BIP I. Ele inclui uma instrução de controle, três instruções de transferência (sendo duas de acesso à memória) e quatro instruções de aritmética: soma e subtração entre o acumulador e uma variável ou uma constante.

Como pode ser observado na Tabela 2, o BIP I consiste basicamente de uma calculadora programável que realiza operações de soma e subtração com variáveis e constantes. No entanto, apesar de limitada, sua arquitetura permite ilustrar várias relações entre as abstrações estudadas nas disciplinas da área de Programação e sua representação no nível arquitetural do processador, conforme é mostrado na Tabela 3.

**Tabela 2.** Conjunto de instruções da arquitetura BIP I

Código da operação	Instrução	Operação	Classe
00000	HLT	Paralisa a execução	Controle
00001	STO endereço	(endereço) ← ACC	Transferência
00010	LD endereço	ACC ← (endereço)	Transferência
00011	LDI constante	ACC ← constante	Transferência
00100	ADD endereço	ACC ← ACC + (endereço)	Aritmética
00101	ADDI constante	ACC ← ACC + constante	Aritmética
00110	SUB endereço	ACC ← ACC - (endereço)	Aritmética
00111	SUBI constante	ACC ← ACC - constante	Aritmética
01000 - 11111	Reservados para as futuras gerações		

Notas: 1. (endereço) significa conteúdo da posição da memória apontada por endereço.

2. Em todas as instruções, com exceção da HLT, o PC é incrementado em uma posição.

**Tabela 3.** Relações entre conceitos de Programação e de Arquitetura suportadas pelo BIP I

Conceitos de Programação	Conceitos de Arquitetura de Computadores
Variável	Posição na memória
Constante	Operandos imediato na instrução
Atribuição	Acesso à memória para leitura e /ou escrita de/em uma posição
Operação aritmética	Utilização de unidade de soma/subtração
Comandos com múltiplas operações	Uso de uma instrução para cada operação realizada
Desempenho dos programas	Número de instruções na linguagem de montagem
Papel do compilador	Tradução da linguagem de alto nível para a linguagem de montagem

#### 4.1 O Uso da Arquitetura do BIP I no Ensino

A arquitetura do processador BIP I foi utilizada em duas turmas de uma disciplina de introdução a Ciência da Computação para 80 alunos ingressantes no semestre 2005/2.

A apresentação do BIP I foi articulada com os conteúdos correlacionados ministrados em disciplina da área de Algoritmos e Programação. Inicialmente, foi feita a introdução de conceitos básicos de arquitetura de computadores, após o qual o BIP I foi utilizado como estudo de caso para ilustrar esses conceitos. Continuando, foram mostrados exemplos de programas e realizados exercícios com os alunos, os quais demonstraram facilidade no entendimento dos conceitos apresentados, bem como na construção de programas em linguagem de montagem.

Os trechos de código a seguir ilustram exemplos de tradução de linguagem de alto nível para linguagem de montagem realizados por alunos no primeiro semestre do Curso.

##### Exemplo 1

Comando de alto nível:  $A = A + 1$ ;

Código *assembly*:  
 LD A ; ACC ← A  
 ADDI 1 ; ACC ← ACC + 1  
 STO A ; A ← ACC

##### Exemplo 2

Comando de alto nível:  $A = A + B - 3$ ;

Código *assembly*:  
 LD A ; ACC ← A  
 ADD B ; ACC ← ACC + B  
 SUBI 3 ; ACC ← ACC - 3  
 STO A ; A ← ACC

A aplicação do processador BIP I permitiu confirmar a efetividade da abordagem proposta, uma vez que os alunos puderam consolidar os conceitos estudados na disciplina da área de Programação. No entanto, pôde-se evidenciar que o conjunto de instruções limitado restringiu o uso do BIP I ao não suportar o funcionamento de estruturas de controle no nível arquitetural. Essa limitação, já prevista no início do projeto, será contornada com a disponibilização e utilização da arquitetura BIP II (descrita a seguir). A idéia é utilizar inicialmente o BIP I para apresentar os conceitos introdutórios e, após o seu entendimento, utilizar o BIP II para ampliar a abrangência desse estudo. Evidentemente que essa transição terá de ser feita de maneira articulada com a disciplina da área de área de Algoritmos e Programação.

## 5. A arquitetura BIP II

O BIP II consiste de uma extensão do BIP I e possui as mesmas características arquiteturais. Ele foi especificado para incluir o suporte a estruturas de controle para a implementação de desvios condicionais e incondicionais, bem como laços de repetição. Para tal, o conjunto de instruções foi estendido incluindo seis instruções de comparação e desvio condicional e uma instrução de desvio incondicional, descritas na Tabela 4. Todas essas instruções utilizam o formato de instrução da Figura 1.

Para suportar as instruções de comparação e desvio condicional, um registrador de estado (STATUS) foi acrescentado. Qualquer instrução de comparação e

desvio condicional deve ser precedida por uma instrução de subtração (SUB ou SUBI). Dependendo do resultado dessa operação os flags Z (Zero) e N (Negative) do registrador STATUS são ativados. As instruções de comparação e desvio então verificam o valor desses flags para determinar se o desvio deve ser tomado ou não, conforme o tipo de comparação associado.

A arquitetura do BIP II ainda não foi aplicada em ensino, o que deverá ser feito a partir do semestre 2007/1 com a disponibilização de um conjunto de ferramentas de apoio que estão em desenvolvimento, incluindo um compilador de Português Estruturado para *assembly*, um montador e um simulador do conjunto de instruções.

**Tabela 4.** Conjunto de instruções da arquitetura BIP II

Código da operação	Instrução	Operação	Classe
0000 – 00111	Conjunto de instruções do BIP I (ver Tabela 2)		
01000	BEQ endereço	Se (STATUS.Z=1) então PC ← endereço Se não PC ← PC + 1	Comparação e desvio condicional
01001	BNE endereço	Se (STATUS.Z=0) então PC ← endereço Se não PC ← PC + 1	
01010	BGT endereço	Se (STATUS.Z=0) e (STATUS.N=0) então PC ← endereço Se não PC ← PC + 1	
01011	BGE endereço	Se (STATUS.N=0) então PC ← endereço Se não PC ← PC + 1	
01100	BLT endereço	Se (STATUS.N=1) então PC ← endereço Se não PC ← PC + 1	
01101	BLE endereço	Se (STATUS.Z=1) ou (STATUS.N=1) então PC ← endereço Se não PC ← PC + 1	
01110	JMP endereço	PC ← endereço	
01111 – 11111	Reservados para as futuras gerações		

onde: BEQ – Branch on Equal                      BNE – Branch on Not Equal  
 BGT – Branch on Greater Than                BGE – Branch on Greater or Equal  
 BLT – Branch on Less Than                    BLE – Branch on Less or Equal  
 JMP – Jump

## 6. A organização do BIP I e do BIP II

A Figura 2 ilustra uma das organizações especificadas para a arquitetura BIP I. Essa organização utiliza uma estrutura do tipo Harvard com memórias separadas para dados e instruções. O registrador PC aponta para a posição de memória da próxima instrução, a qual é buscada e armazenada no IR. A unidade de controle (que não é mostrada na figura para lhe conferir melhor legibilidade) decodifica a instrução e define os sinais de controle que comandam os seletores dos três multiplexadores, os comandos de escrita dos registradores e da memória, e o comando de operação da unidade aritmética (soma ou subtração). O PC é incrementado a cada instrução, com exceção de HLT. O módulo de extensão de sinal é responsável por transformar a constante de 11 bits com sinal em uma constante de 16 bits sinalizada para uso em operações de aritmética (ADDI e SUBI).

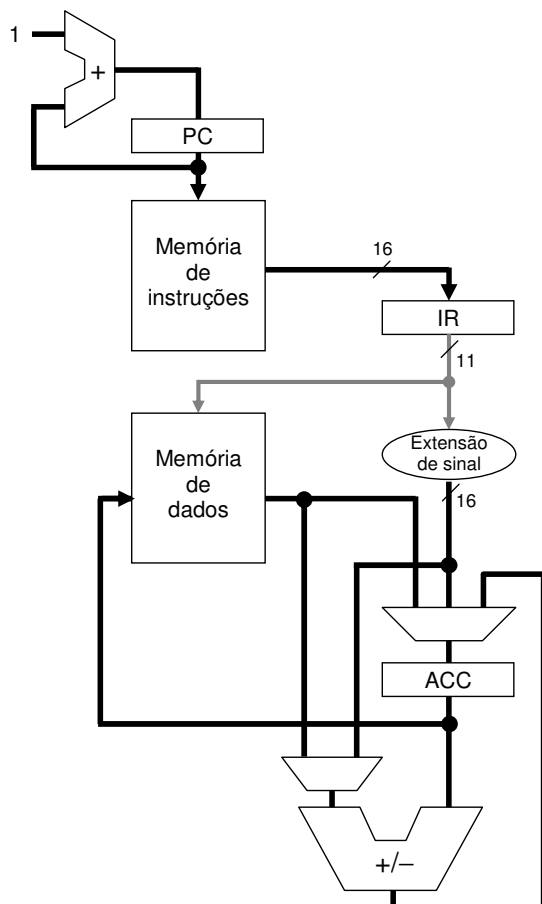


Figura 2. Organização tipo Harvard para o BIP I

Essa organização é bastante simples e atende ao objetivo de facilitar o entendimento por parte de alunos de fases iniciais que estão sendo introduzidos à área de Arquitetura de Computadores, mesmo para aqueles que não tenham uma base de Eletrônica Digital, uma vez que seus blocos constituintes podem ser descritos apenas pela sua funcionalidade (somador, multiplexador, registrador, memórias,...).

A organização do BIP II é uma extensão da organização do BIP I, incluindo o registrador STATUS e modificações no circuito de atualização do PC. Essas modificações resultam um grau de complexidade adicional, mas ainda de fácil compreensão. O trecho de código C a seguir apresenta a descrição do comportamento desse circuito implementada no simulador da arquitetura do BIP II. O switch é utilizado para definir o valor da variável booleana `branch`, que seleciona a fonte de atualização do PC. Nos desvios condicionais, ela é calculada em função do tipo de desvio e dos valores dos flags N e Z do registrador STATUS.

```
(...)
```

```
switch (opcode) {
  case BEQ : branch = (status.z==1);
             break;

  case BNE : branch = (status.z==0);
             break;

  case BGT : branch = ((status.n==0) &&
                       (status.z==0));
             break;

  case BGE : branch = (status.n==0);
             break;

  case BLT : branch = (status.n==1);
             break;

  case BLE : branch = ((status.n==1) ||
                       (status.z==1));
             break;

  case JMP : branch = true;
             break;

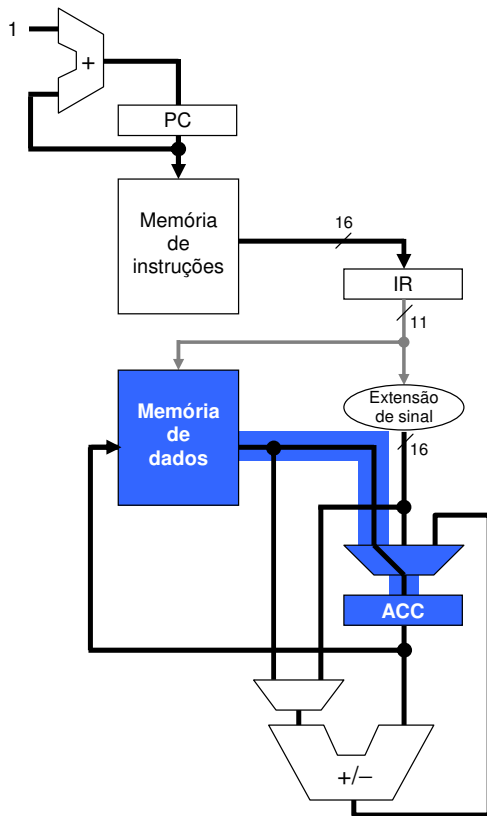
  default  : branch = false;
             break;
}

if (branch)
  pc = addr;
else
  if (opcode!=HLT)
    pc++;

(...)
```

## 6.1 O uso da organização do BIP I no ensino

A organização do BIP I foi apresentada aos alunos da disciplina de introdução a Ciência da Computação após o estudo da sua arquitetura. Os blocos construtivos foram vistos como caixas-pretas que implementam uma dada funcionalidade. Para cada instrução, buscou-se ilustrar os componentes do processador utilizados na execução das operações necessárias para a sua execução. Para exercitar esses conhecimentos, os alunos foram orientados a resolver problemas do tipo “destaque os componentes do caminho de dados do processador BIP necessários à execução da instrução LD”, produzindo diagramas como o ilustrado na Figura 3.



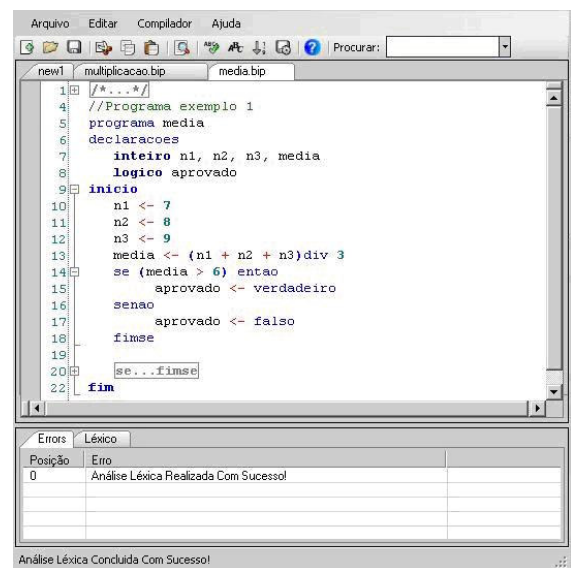
**Figura 3.** Exemplo de resolução de exercício sobre a organização do BIP I

Com esse tipo de exercício, os alunos puderam identificar e contornar deficiências na compreensão inicial após a exposição realizada, consolidando o entendimento sobre a implementação do conjunto de instruções.

## 7. Integração multidisciplinar

Neste semestre, está prevista a utilização da especificação da arquitetura BIP II em atividades práticas de três disciplinas de cursos da área de Computação (Ciência da Computação e Engenharia de Computação).

Na disciplina Compiladores, do 6º período do Curso de Ciência da Computação, um grupo de alunos está implementando um ambiente de desenvolvimento para auxiliar na conversão de códigos de alto nível, escritos em Português Estruturado, para a linguagem de montagem do BIP. Até o momento, já estão implementadas as funções de análise léxica, sintática e semântica da linguagem de alto nível, além de recursos de interface, incluindo: salientador de sintaxe (*syntax highlight*) e funções para localizar texto e manipulação de arquivos (novo, abrir e salvar). As próximas etapas incluem: (i) a geração do código *assembly* com as instruções do BIP; (ii) a implementação do interpretador do Português Estruturado; (iii) a validação e; (iv) a realização de experimentos. A Figura 4 mostra uma imagem do protótipo da interface do ambiente em desenvolvimento.



**Figura 4.** Tela do ambiente de desenvolvimento para o BIP (compilador)

Na disciplina Circuitos Digitais do 2º período do Curso de Ciência da Computação, os alunos serão orientados a realizar uma implementação em diagrama esquemático do BIP II utilizando o ambiente MAX+plus II da Altera. Uma implementação similar do BIP I já foi realizada por

alunos desse curso em atividades extracurriculares e o modelo sintetizado em FPGA da Altera consumiu 135 células lógicas em dispositivo da família Flex10K (sem considerar o custo de bits de memória embutida).

A especificação da arquitetura e da organização do BIP II também será usada em disciplina em nível de tópicos especiais do 9º período do Curso de Engenharia de Computação. Nessa disciplina, os alunos irão implementar o BIP II em linguagem de descrição de hardware e realizar a extensão do modelo incluindo o suporte a periféricos integrados (ex. portas de E/S e temporizadores), construindo, assim, um microcontrolador baseado no BIP.

Complementarmente às atividades práticas dessas disciplinas, alunos vinculados ao Grupo de Concepção de Sistemas Embarcados e Distribuídos da UNIVALI estão implementando ferramentas de software adicionais para suportar o uso do BIP. Uma dessas ferramentas consiste em um simulador de conjunto de instruções e a outra em um montador para gerar arquivos no formato MIF (Memory Initialization File) da Altera e no formato Hexa da Intel.

## 9. Conclusões

Neste artigo foram apresentados aspectos referentes ao desenvolvimento de uma série de processadores para apoio ao ensino de Arquitetura de Computadores em disciplinas introdutórias de cursos da área de Computação, com ênfase no auxílio à consolidação dos conceitos básicos de Programação. Um desses processadores já foi aplicado e mostrou bastante efetivo. O segundo encontra-se em desenvolvimento e será aplicado no ano de 2007.

Ainda não foram realizados experimentos para expressar, quantitativamente, o impacto do uso dessa arquitetura na primeira fase do Curso. Esses experimentos serão realizados a partir do ano de 2007, com a conclusão do desenvolvimento das ferramentas de suporte à simulação e à programação que servirão de apoio às atividades práticas a serem executadas pelos alunos.

Com relação à próxima geração do BIP, prevê-se o desenvolvimento de extensões para suporte a chamada de procedimento e o acesso de dispositivos de entrada-e-saída.

## Agradecimentos

Este projeto conta com o apoio do Programa Art. 170 do Governo do Estado de Santa Catarina.

## Referências

- [1] A. Carbone, J. Kaasboll, J. “A survey of methods used to evaluate computer science teaching”. *Proc. of 6th annual Conf. on the Teaching of Computing*, Dublin, 1998.
- [2] C. S., Menezes, A. M. Nobre, “Um ambiente cooperativo para apoio a cursos de introdução a programação”. *Workshop de Educação em Computação, CSBC*, 2002.
- [3] E. Pimentel et al., “Avaliação Contínua da Aprendizagem, das Competências e Habilidades em programação de Computadores”. *Workshop de Informática na Escola, CSBC*, 2003.
- [4] J. Good, P. Brna, P. “Program comprehension and authentic measurement: a scheme for analyzing descriptions of programs”. *Journal of Human-Computer Studies*, v.61, 2004.
- [5] S. Chang, “Computer Anxiety and perception of task complexity in learning programming-related skills”. *Computers in Human Behavior*, 2004.
- [6] A. Clements, “Selecting a processor for teaching computer architecture”, *Microprocessor and Microsystems*, 23, pp. 281-290, 1999.
- [7] D. A. Patterson, J. L. Hennessy, “Organização e projeto de computadores: a interface hardware/software”, São Paulo, Campus, 2005.
- [8] R. F. Weber “Fundamentos de arquitetura de computadores”, Porto Alegre, Sagra Luzzatto, 2004.
- [9] Microchip “PIC16F8X: 18-pin Flash/EEPROM 8-bit microcontrollers”. Chandler, Microchip Technology, 1998.