

Um Processador Básico para o Ensino de Conceitos de Arquitetura e Organização de Computadores

Diana Morandi, Maicon Carlos Pereira, André Luis Alice Raabe, Cesar Albenes Zeferino

Centro de Ciências Tecnológicas da Terra e do Mar (CTTMar)
Universidade do Vale do Itajaí (Univali)
Rua Uruguai, 458 – Caixa Postal 360 – 88302-202 – Itajaí – SC – Brasil
{diana.morandi, maicon, raabe, zeferino}@univali.br

Abstract. *The knowledge of computers operation is very important to allow to undergraduate students to understand a number of concepts of the programming area. In this sense, teachers of the first courses of programming use abstractions to introduce some of those concepts and to improve the students' understanding. Nevertheless, for some students, such abstractions are very limited to accomplish with this goal, and more detailed model are necessary. This paper is inserted in this context and presents results of a project that aims at the development of a family of basic processors for use in the teaching of concepts of computer architecture for freshmen students.*

Resumo. *O conhecimento do funcionamento de um computador é essencial para que um estudante de Computação possa compreender muitos conceitos da área de programação. Nesse sentido, os professores de disciplinas introdutórias de programação utilizam abstrações para apresentar alguns desses conceitos e possibilitar aos alunos um maior entendimento dos aspectos da disciplina. No entanto, essas abstrações são limitadas e o objetivo do professor muitas vezes não é alcançado. Nesse contexto, este trabalho apresenta resultados de um projeto que visa o desenvolvimento de uma família de processadores básicos para o estudo de arquitetura de computadores por alunos de fases iniciais de cursos de graduação em Computação.*

1. Introdução

O estudo do funcionamento de um computador e de seu processador é fundamental na formação de alunos de cursos de graduação em Computação. Isso porque determinados conceitos utilizados nas disciplinas da área de Algoritmos e Programação são melhor compreendidos pelos alunos quando eles entendem os seus significados ao estudar a arquitetura do computador.

No entanto, em muitos cursos de graduação, as disciplinas da área de Programação são cursadas antes das disciplinas da área de Arquitetura e Organização de Computadores, nas quais o funcionamento do computador é estudado. Conceitos de programação como, por exemplo, constantes, variáveis, vetores, ponteiros, níveis de linguagem e outros, só são completamente assimilados pelos alunos quando o nível de abstração é reduzido.

Para contornar esse problema, em muitos cursos, costuma-se apresentar algumas noções básicas de Arquitetura e Organização de Computadores aos alunos matriculados em disciplinas que oferecem 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.

Duas características são observadas nessa abordagem. A primeira diz respeito à falta de uma articulação adequada entre os professores dessas disciplinas introdutórias e os professores da área de Algoritmos e Programação, pois, em alguns casos, não ocorre o encadeamento necessário na apresentação dos conteúdos. Uma outra característica refere-se à limitação dos modelos utilizados para a apresentação dos conceitos básicos de arquitetura. Embora esses modelos sejam adequados e suficientes para o propósito inicial das disciplinas introdutórias, eles apresentam restrições que não permitem avançar no estudo ou ainda integrá-los a outras disciplinas do curso, como, por exemplo, as da área de Circuitos Digitais. Em resumo, eles pecam pela falta de informações que permitam que um aluno utilize a base de conhecimentos adquiridos nessas disciplinas introdutórias a fim de integrá-los ainda nas primeiras fases do curso.

Nesse contexto, este trabalho apresenta resultados de um projeto em andamento que visa desenvolver e disponibilizar uma família 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 essa família de processadores sirva 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 análise sobre o uso de processadores no ensino de Computação, a descrição da arquitetura e da organização da primeira geração de processadores básicos propostos neste projeto e aspectos de implementação e uso desse processador no ensino.

2. Uso de processadores no ensino de Computação

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 (1999) 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, 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 a 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 e Organização de Computadores (ex. MIPS, x86,...). Porém, esses processadores são demasiadamente complexos para serem aplicados em disciplinas do primeiro ano.

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 (2005), 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 (2004). Nesse livro, o autor apresenta conceitos fundamentais 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.

Nesse contexto, ao buscar estabelecer uma relação entre as necessidades dos alunos que estão iniciando a programar e as representações em hardware correspondentes, devem-se identificar os principais pontos em comum que são tradicionais fontes de incompreensão para os estudantes e que podem ser elucidados de forma concreta. Como exemplo, podem ser mencionadas as relações entre: (i) declaração de variável e alocação de memória; (ii) constantes e operandos imediatos; (iii) atribuição de variáveis e sua correspondência com as operações de acesso à memória; e (iv) operações aritméticas e sua execução no hardware; entre outras relações. 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. BIP – Basic Instruction-set Processor

O projeto BIP (Basic Instruction-set Processor) insere-se no contexto acima apresentado e visa desenvolver uma arquitetura de referência para o ensino de conceitos da área de Arquitetura e Organização de Computadores para alunos de fases iniciais de cursos de graduação em Computação visando prover a base necessária para compreensão das abstrações adotadas nas disciplinas da área de Algoritmos e Programação. O projeto prevê a especificação de uma família 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. Também busca oferecer informações suficientes para que os alunos possam implementar o processador utilizando conhecimentos de disciplinas da área de Eletrônica.

3.1. O contexto do projeto

Este projeto está associado a um curso de graduação em Ciência da Computação que apresenta uma estrutura curricular em que os alunos da primeira fase cursam as disciplinas Algoritmos e Programação (120 h/a), Computação Básica (60 h/a) e outras duas disciplinas da área de Matemática. A primeira disciplina apresenta os conceitos fundamentais de algoritmos e introduz aos alunos as linguagens Português estruturado (ao longo do semestre) e C (no período final). A disciplina Computação Básica oferece uma visão introdutória a diversos conceitos da área de Computação, como os de arquitetura e de organização de computadores, entre outros.

Na segunda fase, o estudo de programação é aprofundado na disciplina Algoritmos e Programação (90 h/a), a qual é seguida pelas disciplinas Programação (60 h/a) e Estruturas de Dados (90 h/a), da terceira fase. Ainda no segundo período, os alunos cursam a disciplina Circuitos Digitais (60 h/a), a qual possui como foco o estudo dos circuitos utilizados na construção de sistemas digitais de processamento. Apenas a partir da terceira fase é que os alunos realizam um estudo completo sobre o funcionamento de processadores e computadores nas disciplinas intituladas Arquitetura e Organização de Computadores, do terceiro e quarto períodos (ambas com 60 h/a).

3.2. Motivação inicial

A motivação para este projeto surgiu da identificação de uma dificuldade por parte dos alunos no entendimento e na consolidação de certas abstrações assumidas nas disciplinas introdutórias da área de Programação. Embora tenha se buscado sanar essa dificuldade na disciplina Computação Básica com o estudo de modelos de processadores hipotéticos descritos na literatura, percebeu-se que o objetivo pretendido não era plenamente alcançado. Isso pôde ser atribuído a várias razões, das quais podem ser destacadas duas. Primeiramente, o modelo adotado era simples o suficiente para promover o entendimento do funcionamento básico do processador (ex. ciclo de instrução) e de alguns conceitos de programação (ex. variável, níveis de linguagem), mas incompleto para demonstrar como as estruturas de controle (como desvios e laços de repetição) são executadas e, em especial, implementadas no processador. Num outro extremo, modelos de processadores mais completos mostraram-se ineficientes frente à demasiada complexidade, dificultando entendimento por parte dos alunos.

3.3. Diretrizes principais

A experiência descrita acima levou à identificação da necessidade e da oportunidade de desenvolver o projeto aqui apresentado. Três diretrizes básicas foram definidas para norteá-lo:

1. A arquitetura do processador deveria ser simples o suficiente para facilitar não apenas o entendimento do seu funcionamento, mas também a sua construção a partir dos conceitos estudados na disciplina Circuitos Digitais;
2. A arquitetura do processador deveria extensível de modo a permitir a inclusão do suporte de hardware para diferentes conceitos da área de programação;
3. Qualquer ferramenta de apoio desenvolvida no contexto do projeto (ex. tradutor, simulador,...) deveria priorizar, na sua implementação, o uso das estruturas de linguagem compreendidas no primeiro ano do curso, de modo a permitir que os estudantes pudessem aplicar os conhecimentos já adquiridos para estudar e estender essas ferramentas.

A partir dessas diretrizes, foi definido que os seguintes conceitos seriam focalizados nas duas primeiras gerações do processador BIP:

- **BIP I:** níveis de linguagem, constantes, variáveis, representação de dados e de instruções, conjuntos de instruções, programação em linguagem de montagem e geração de código na linguagem da máquina;
- **BIP II:** inclusão de suporte a implementação de estruturas de controle para desvios (condicionais e incondicionais) e laços de repetição, incluindo, o suporte a operações relacionais.

3.4. Arquitetura BIP I

O BIP utiliza uma arquitetura baseada na arquitetura RISC (Reduced Instruction-Set Computer) do microcontrolador PIC da Microchip (1998). Ele é uma máquina orientada a acumulador e não possui banco de registradores como o processador MIPS [Patterson e Hennesy 2005]. Como apresenta um único registrador para armazenamento de dados, todas as operações envolvem o acumulador e, em algumas operações aritméticas, é necessário utilizar operandos posicionados na memória (ou sejam, não é uma máquina *load/store*). Esses dois aspectos descaracterizam o BIP I como uma arquitetura RISC pura. Por outro lado, outros requisitos das arquiteturas RISC são contemplados [Esponda e Rojas 1992], como pode ser observado na Figura 1 e na Tabela 1. Por exemplo, todas as instruções são baseadas em um único formato de instrução e em dois modos de endereçamento.

Tamanho da palavra de dados	16 bits																																
Tipos de dados	Inteiro com sinal																																
Tamanho da palavra de instrução	16 bits																																
Formato de instrução	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">15</td><td style="text-align: center;">14</td><td style="text-align: center;">13</td><td style="text-align: center;">12</td><td style="text-align: center;">11</td><td style="text-align: center;">10</td><td style="text-align: center;">9</td><td style="text-align: center;">8</td><td style="text-align: center;">7</td><td style="text-align: center;">6</td><td style="text-align: center;">5</td><td style="text-align: center;">4</td><td style="text-align: center;">3</td><td style="text-align: center;">2</td><td style="text-align: center;">1</td><td style="text-align: center;">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																																
Registradores	<u>ACC</u> : acumulador <u>IR</u> : registrador de instruções <u>PC</u> : contador de programa																																
Classes de instrução	<u>Transferência (acesso à memória)</u> : STO, LD e LDI <u>Aritmética</u> : ADD, ADDI, SUB e SUBI																																

Figura 1. Quadro-resumo da arquitetura do BIP I

Tabela 1. Conjunto de instruções do BIP I

Código da operação	Instrução	Operação	Classe
00000	Reservada	Reservada	Reservada
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
01000-01111	Reservadas	Reservadas	Reservadas

Notas: 1. A expressão "(endereço)" significa conteúdo da posição da memória apontada por endereço.
 2. Os códigos 01000-01111 estão reservados para futuras gerações da arquitetura BIP.

Como pode ser observado pelo conjunto de instruções descrito na Tabela 1, o BIP I consiste basicamente de uma calculadora programável que realiza operações de soma e subtração com variáveis e constantes. No entanto, apesar de limitado, esse conjunto de instruções permite ilustrar a implementação em hardware de diversas abstrações estudadas nas disciplinas da área de Algoritmos e Programação, como pode ser visto nos exemplos de utilização do conjunto de instruções do BIP I mostrados na Tabela 2.

Tabela 2. Exemplos de programas para o BIP I

Exemplo	Código de alto nível	Código assembly
Atribuição de uma constante	A = 10	LDI 10 ; ACC ← 10 STO A ; A ← ACC
Atribuição de uma variável	A = B	LD B ; ACC ← B STO A ; A ← ACC
Incremento de variável	A = A + 1	LD A ; ACC ← A ADDI 1 ; ACC ← ACC + 1 STO A ; A ← ACC
Decremento de variável	B = B - 2	LD B ; ACC ← B SUBI 2 ; ACC ← ACC - 2 STO B ; B ← ACC
Comando com múltiplas operações	A = A + B - 3	LD A ; ACC ← A ADD B ; ACC ← ACC + B SUBI 3 ; ACC ← ACC - 3 STO A ; A ← ACC

3.5. Organização do BIP I

A Figura 2 mostra uma das organizações especificadas para a arquitetura BIP I. Essa organização utiliza uma estrutura do tipo Princeton (modelo de von Neumann), com uma única memória para armazenamento de dados e de 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 (UA). Uma segunda organização do tipo Harvard, com memórias separadas para instruções e dados também foi especificada.

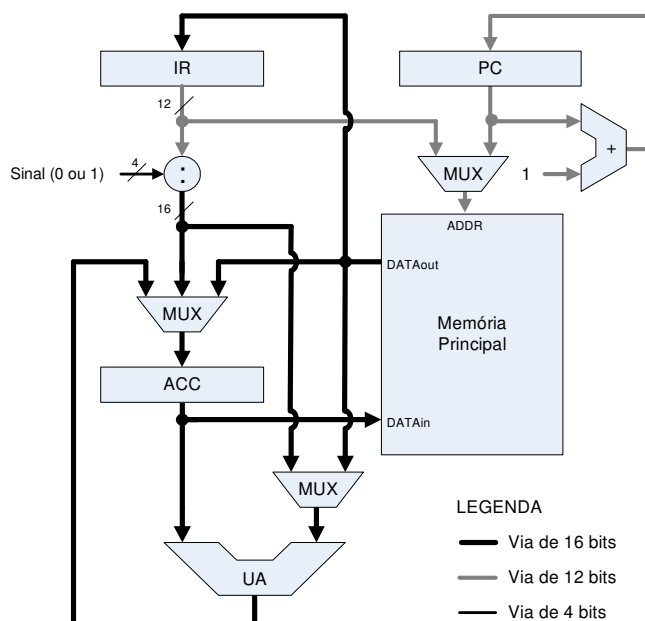


Figura 2. Organização do BIP I

3.6. Implementação do BIP I

A organização do BIP I foi implementada nos ambientes MAX+plus II e Quartus II da Altera por meio do uso das ferramentas de captura de esquemático, permitindo confirmar o funcionamento da organização proposta. A escolha pela entrada via captura de esquemático justifica-se pela maior facilidade de entendimento do modelo gráfico em relação a modelos implementados via linguagens de descrição de hardware, as quais exigem um conhecimento adicional do aluno.

O modelo esquemático do BIP utiliza uma mega-função parametrizável para implementar a memória principal, cujo conteúdo inicial é carregado por meio de um arquivo de inicialização com os segmentos de texto e de dados. Esse arquivo é gerado por uma ferramenta desenvolvida neste projeto que realiza a tradução do código *assembly* para o código de máquina em hexadecimal. Essa ferramenta possui uma interface visual e é codificada em linguagem C++, porém utilizando programação estruturada.

Diversos códigos foram utilizados para verificar a correção dessa ferramenta e do modelo esquemático do processador, permitindo também validar a organização proposta através da síntese do modelo e da sua simulação no MAX+plus II e no Quartus II. Destaca-se que o BIP requer 135 células lógicas em FPGAs da família Flex10K da Altera.

3.7. Relatos de aplicação do BIP I no ensino

A arquitetura do processador BIP I foi aplicada em duas turmas da disciplina Computação Básica para 80 alunos ingressantes no semestre 2005/2. A introdução do BIP I foi articulada com os conteúdos correlacionados ministrados na disciplina Algoritmos e Programação. Inicialmente, foi feita a introdução de conceitos básicos de arquitetura de computadores, após o qual o BIP 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 bastante facilidade no entendimento dos conceitos apresentados, bem como na construção de programas em linguagem de montagem (confirmada em avaliação escrita). Também lhes foi apresentada a organização do BIP, com ênfase no funcionamento do caminho de dados para cada instrução do conjunto de instruções; e os resultados foram igualmente satisfatórios.

4. Conclusões

A experiência da aplicação do processador BIP na primeira fase do curso mostrou-se bastante efetiva, uma vez que os alunos demonstraram ter assimilado com facilidade os conceitos apresentados. No entanto, pôde-se evidenciar que o conjunto de instruções limitado restringiu o uso do BIP para ilustrar o funcionamento de estruturas de controle no nível arquitetural.

Para contornar esse problema, já está em desenvolvimento o BIP II, o qual será uma extensão do BIP I incluindo instruções de comparação e de desvio condicional e incondicional (que utilizarão os códigos de operação disponíveis). O objetivo é manter o uso do BIP I até que os conceitos associados estejam assimilados e, de maneira articulada com a disciplina Algoritmos e Programação, introduzir o BIP II após a apresentação das estruturas de controle nessa disciplina.

Paralelamente à especificação e à implementação do novo processador, estão sendo realizados trabalhos no sentido de disponibilizar um simulador do conjunto de instruções (para o BIP I e para o BIP II), bem como um compilador de Português Estruturado para o *assembly* do BIP. Este último trabalho está sendo realizado em uma integração com a disciplina Compiladores.

Destaca-se ainda que o BIP será utilizado em atividades práticas da disciplina Circuitos Digitais, de forma que os alunos sejam orientados a implementar seus componentes (registradores, multiplexadores, circuitos aritméticos e de controle) e integrá-los na forma do processador. Também será considerado o seu uso como estudo de caso em cursos sobre modelagem de hardware em VHDL e em SystemC.

Referências

- Clements, A. (1999) "Selecting a processor for teaching computer architecture", *Microprocessor and Microsystems*, 23, pp. 281-290.
- Esponda, M.; Rojas, R. (1992) "A graphical comparison of RISC processors", *ACM Computer Architecture News*, v. 20, n. 4, pp.2-8.
- Microchip (1998) "PIC16F8X: 18-pin Flash/EEPROM 8-bit microcontrollers". Chandler, Microchip Technology.
- Patterson, D. A.; Hennessy, J. L. (2005) "Organização e projeto de computadores: a interface hardware/software", São Paulo, Campus.
- Weber, R. F. (2004) "Fundamentos de arquitetura de computadores", Porto Alegre, Sagra Luzzatto.