

Bipide: Ambiente de Desenvolvimento Integrado para Utilização dos Processadores BIP no Ensino de Programação

***Abstract.** Teaching introductory programming generally involves an abstraction level, which is difficult for students with learning disabilities especially in the programming logic. Considering this, a family of processors, called BIP – Basic Instruction-set Processor, was developed in order to allow an interdisciplinary approach to teaching initial concepts on the first year of Computer Science undergraduate courses. Therefore, this paper presents an integrated development environment (IDE) called Bipide, which enables the development, implementation and simulation of programs written in structured Portuguese (called Portugol) linking them to the processor architecture of the BIP family. The IDE provides a tool to support the teaching of subjects of introductory programming reducing the abstraction involved in fundamental concepts of programming logic. It will also serve as an instrument to collect empirical evidence on the advantage to reduce the level of abstraction involved in this type of learning.*

***Resumo.** O ensino de conceitos introdutórios de programação costuma apresentar um nível de abstração que prejudica o aprendizado de alunos que apresentam dificuldades em lidar com o raciocínio lógico necessário ao entendimento da lógica de programação. Pensando nesta dificuldade, uma família de processadores denominada BIP – Basic Instruction-set Processor, foi desenvolvida visando à integração interdisciplinar e o apoio ao ensino em disciplinas de fases iniciais dos cursos de área tecnológica. Este texto apresenta um ambiente de desenvolvimento integrado projetado para o desenvolvimento e execução de programas em linguagem Portugol, os quais serão simulados sobre a arquitetura dos processadores BIP. Com isto, disponibiliza-se uma ferramenta de apoio ao ensino das disciplinas de Algoritmos e Programação que auxiliará na redução da abstração envolvida em conceitos fundamentais da lógica de programação e facilitará a compreensão dos conceitos envolvidos nesta aprendizagem.*

1. Introdução

As disciplinas da área de Arquitetura e Organização de Computadores são fundamentais na formação de alunos dos cursos de Ciência da Computação e de Engenharia da Computação, pois permitem estabelecer relações dos conceitos de lógica de programação com aspectos concretos do hardware, reduzindo assim o nível de abstração envolvido nesta aprendizagem [Borges e Silva 2006].

A escolha 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 são muito abstratos e não permitem estabelecer essas relações ou, quando

mais avançados, são demasiadamente complexos para serem aplicados em disciplinas do primeiro ano [Morandi et al. 2006].

Para tentar reduzir esta dificuldade, uma família de processadores denominada BIP – Basic Instruction-set Processor, foi concebida e desenvolvida visando apoiar o ensino e a integração interdisciplinar entre disciplinas das fases iniciais dos cursos de área tecnológica.

Neste sentido este trabalho apresenta uma etapa do projeto, cujo objetivo é a construção de um ambiente de desenvolvimento integrado (IDE - Integrated Development Environment) que possibilita a utilização dos processadores BIP na apresentação de conceitos das disciplinas de Algoritmos e Programação. Este ambiente foi denominado Bipide (BIP + IDE).

2. BIP – Basic Instruction-set Processor

A família de processadores BIP foi desenvolvida por pesquisadores do Laboratório de Sistemas Embarcados e Distribuídos da Universidade do Vale do Itajaí com o objetivo de disponibilizar uma série de processadores de arquitetura simplificada que auxiliasse no aprendizado de Arquitetura e Organização de Computadores permitindo relacionar os conteúdos de programação em alto nível com sua representação em hardware [Morandi et al. 2006].

Dos três modelos de processadores definidos até o momento, são utilizados neste projeto os dois primeiros: BIP I e BIP II. Estes processadores são focados no entendimento de conceitos como constantes, variáveis, conjuntos de instruções, geração de código em linguagem de máquina e laços de repetição. [Morandi; Raabe e Zeferino 2006].

2.1. Arquitetura e Organização dos Processadores BIP

A arquitetura do BIP possui um conjunto restrito de instruções 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) [Morandi; Raabe e Zeferino 2006], [Morandi et al. 2006].

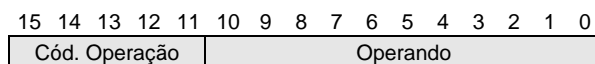


Figura 1. Formato de instrução dos processadores BIP

A arquitetura do BIP I inclui três registradores: (i) PC (Program Counter): aponta para o endereço da próxima instrução; (ii) IR (Instruction Register): armazena o endereço da instrução em execução; e (iii) ACC (Accumulator): utilizado para armazenamento de dados durante uma operação. O BIP II possui ainda o registrador STATUS, utilizado para a execução das instruções de desvio [Morandi; Raabe e Zeferino 2006].

Em sua organização, os processadores BIP utilizam a estrutura Harvard, com memórias separadas para dados e instruções, como pode ser observado na Figura 2.

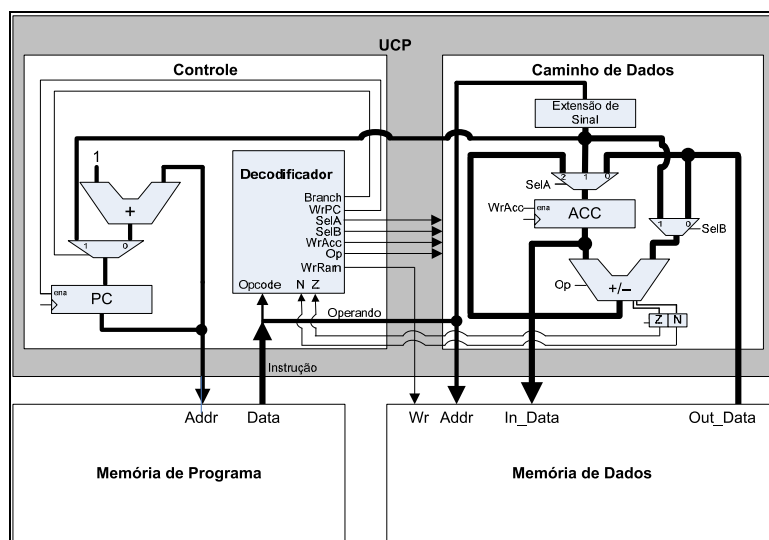


Figura 2. Organização do processador BIP II

O processador é dividido em dois blocos: (i) Controle, responsável por gerar sinais de controle para o caminho de dados e atualização do PC; e (ii) Caminho de Dados, que inclui os circuitos necessários para execução das instruções [Morandi; Raabe e Zeferino 2006].

3. Uso de Simuladores no Ensino

O uso de simuladores que podem facilitar a compreensão do funcionamento do processador é fundamental para o ensino adequado da arquitetura de computadores. Estes simuladores, no entanto, costumam apresentar uma interface de usuário pouco elaborada e com poucos recursos operacionais. Além disto, em sua maioria, não permitem o desenvolvimento de programas em linguagem alto nível, o que dificulta sua aplicação nas disciplinas de programação e os restringe às disciplinas da área de Arquitetura e Organização de Computadores [Borges e Silva 2006], [Borunda; Brewer e Erten 2008].

Uma análise de simuladores de arquitetura de caráter educacional foi realizada com o objetivo de identificar características destas ferramentas, as quais serviram para definir algumas funcionalidades do ambiente Bipide, de forma que este apresentasse características de interface que facilitassem sua utilização em disciplinas introdutórias da área de programação. Esta análise é apresentada nas conclusões deste artigo onde comparam-se os trabalhos similares com a IDE desenvolvida.

4. Bipide

O ambiente Bipide insere-se no contexto acima apresentado e disponibiliza um ambiente de desenvolvimento integrado que possibilita o desenvolvimento, execução e simulação de programas em linguagem Portugol, relacionando-os à arquitetura dos processadores BIP.

O Bipide é composto por: (i) um editor de textos destinado ao desenvolvimento de programas em linguagem Portugol; (ii) um compilador capaz de traduzir a linguagem Portugol para *assembly* dos processadores BIP; (iii) um simulador dos processadores BIP I e BIP II capaz de simular as instruções destes processadores exibindo

graficamente sua execução por meio de animações; e (iv) um módulo de ajuda composto por informações teóricas relacionadas à Arquitetura e Organização dos processadores BIP e funcionalidades do ambiente.

O projeto do ambiente foi definido a partir da experiência descrita anteriormente e da análise de ferramentas de simulação de arquitetura. Foram definidos os requisitos do sistema, diagramas de caso de uso, protótipos de interface e a descrição da gramática a ser utilizada para a construção do compilador. Os requisitos funcionais especificados incluem:

- O sistema deverá permitir ao usuário escrever e compilar programas em linguagem Portugol;
- O Sistema deverá indicar os erros encontrados no programa durante a compilação;
- O sistema deverá permitir a execução do programa passo a passo;
- O sistema deverá gerar código *assembly* para os processadores BIP I e BIP II; e
- O sistema deverá simular o funcionamento do programa sobre a arquitetura dos processadores BIP I e BIP II.

Os analisadores léxico e sintático do compilador utilizado foram desenvolvidos através da ferramenta para geração de compiladores ANTLRWorks [Parr 2009]. Também foram definidas através do ANTLRWorks as ações semânticas necessárias para a geração de código *assembly*.

Para a implementação do Bipide utilizou-se duas ferramentas de desenvolvimento de software: Visual Studio 2008 [Microsoft Corporation 2009b] e Expression Blend 2 [Microsoft Corporation 2009a]. O Visual Studio 2008 foi utilizado para a implementação da interface e das classes que compõem o sistema em linguagem C#. Os elementos gráficos e animações que compõem o simulador do processador BIP foram desenvolvidos em WPF¹ (Windows Presentation Foundation) através da ferramenta Expression Blend 2.

A escolha pela tecnologia WPF justifica-se pelas facilidades disponibilizadas para o desenvolvimento de aplicações que exigem muitos recursos de interface, como a necessidade da utilização de animações. Esta tecnologia integrada aos demais componentes utilizados resultou em uma interface gráfica capaz de proporcionar uma melhor experiência para o usuário, agregando valor ao ambiente desenvolvido.

4.1. Visão Geral das Funcionalidades e dos Aspectos de Interface Gráfica

O Bipide apresenta três módulos principais: (i) Programação, onde o usuário poderá escrever e compilar programas; (ii) Simulação, onde poderão ser simulados os programas desenvolvidos; e (iii) Ajuda, onde são apresentadas informações a respeito

¹ O Windows Presentation Foundation (WPF) é uma tecnologia da Microsoft para desenvolvimento de aplicativos e interfaces. Sua utilização permite a construção de aplicações que possibilitam uma melhor experiência do usuário devido à possibilidade destas aplicações possuírem uma interface gráfica rica e interativa.

das funcionalidades do sistema e sobre a arquitetura e organização dos processadores BIP.

O módulo de programação apresentado na Figura 3 oferece ao usuário recursos e funcionalidades típicas de editores de código fonte como, por exemplo, identificação de palavras reservadas e símbolos da linguagem (*syntax highlighting*). Oferece ainda opções de gerenciamento e edição de arquivos (Abrir, Salvar, Copiar, etc.). Na parte inferior do módulo são exibidas alertas e mensagens de erros ocorridos durante a compilação do programa.

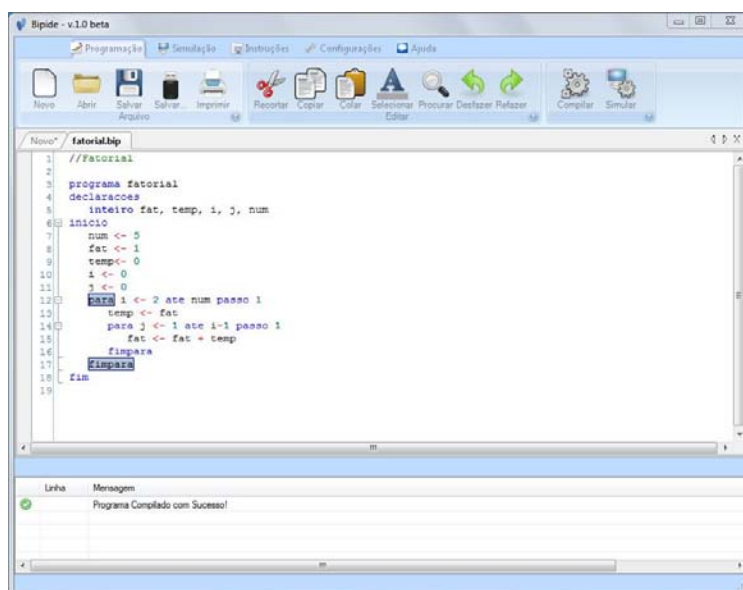


Figura 3. Interface do módulo de programação

A Figura 4 apresenta a interface do módulo de simulação. Neste módulo o usuário poderá executar o programa passo a passo e simular sua execução nos processadores BIP I e BIP II. Na parte superior são exibidos o código do programa em linguagem alto nível e o código *assembly* correspondente. Durante a simulação, a linha do programa Portugol que está sendo executada é destacada, também é destacado o conjunto de instruções *assembly* que correspondem a esta instrução Portugol. Desta forma permite-se ao usuário verificar quais as instruções *assembly* necessárias para executar uma instrução Portugol.

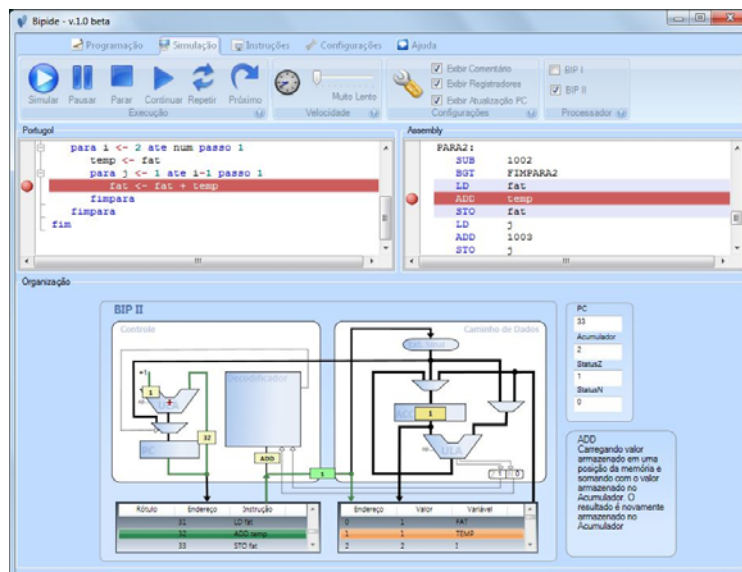


Figura 4. Interface do módulo de simulação

A simulação do programa é feita através de animações que representam o fluxo de execução do programa nos processadores BIP. São representados os registradores especiais do processador, as memórias de dados e instrução e seus respectivos valores. Para auxiliar o usuário a compreender o que está ocorrendo no processador são apresentadas mensagens descrevendo as ações executadas.

A interface do módulo de simulação foi concebida pensando-se na possibilidade do usuário visualizar ao mesmo tempo os três elementos utilizados: linguagem alto nível, linguagem *assembly* e organização do processador. Isto permite a realização de comparações e associações entre estes elementos, colaborando para a redução da abstração apresentada nos conceitos de programação. A Figura 5 ilustra uma das relações estabelecidas, onde o mesmo valor pode ser ver visto nos três elementos da interface.

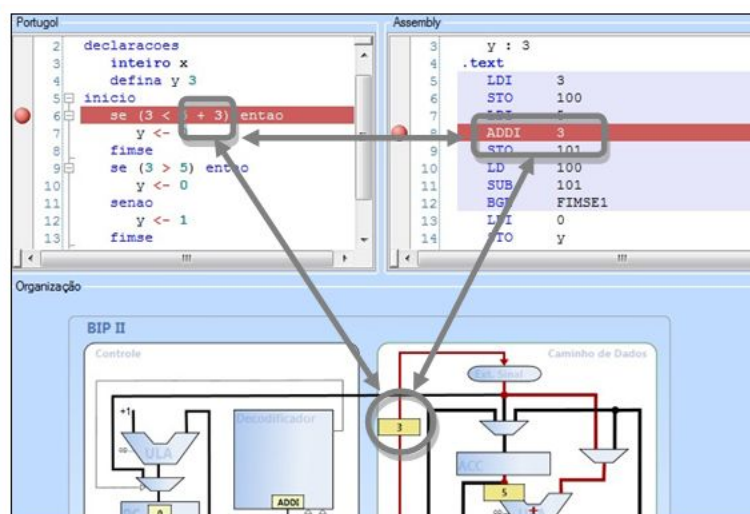


Figura 5. Relação de elementos na interface do módulo de simulação

5. Verificação e Avaliação

Foram realizados testes no ambiente visando a eliminação de erros existentes no projeto e desenvolvimento. Durante a implementação das ações semânticas do compilador, foi utilizado um conjunto de aplicações em linguagem Portugol para sua validação. Estes programas foram compilados e os códigos *assembly* gerados foram analisados e comparados com os resultados esperados.

Para a validação do simulador foram utilizados testes unitários para verificar a simulação de cada instrução do processador. Foi utilizado um conjunto de instruções *assembly* para verificar o estado esperado para os registradores e memórias. Este mesmo conjunto de instruções já havia sido utilizado para validação da arquitetura do microcontrolador μ BIP² [Pereira 2008] e após serem simulados no ambiente Bipide, tiveram os valores resultantes analisados e comparados com o obtido previamente.

Foi realizada uma experiência de utilização do ambiente por alunos da disciplina de Algoritmos e Programação do primeiro período do curso de Ciência da Computação da Universidade do Vale do Itajaí. Este experimento teve como principal objetivo avaliar a satisfação dos usuários com o ambiente e verificar possíveis melhorias a serem realizadas. Os alunos tiveram acesso ao Bipide e após sua utilização responderam um questionário composto por seis perguntas a respeito da qualidade da interface do ambiente. As questões aplicadas são apresentadas a seguir:

1. Você acha que o seu nível de conhecimento foi compatível com as telas e vocabulário empregado no sistema?
2. As mensagens de erro lhe auxiliaram de forma adequada?
3. O sistema lhe proporcionou feedback, ou seja, você conseguiu visualizar todas suas ações?
4. A estrutura dos painéis do menu lhe parece disposta de forma lógica por agrupamento de tipos de opções?
5. Qual é o grau de satisfação que você atribui à utilização do sistema?
6. Aponte sugestões e críticas gerais para a melhoria do ambiente.

Dos 21 alunos que responderam o questionário, 78,10% atribuíram notas de 8 a 10 às questões realizadas, como pode ser visto no gráfico ilustrado na Figura 6.

² O μ BIP estende a arquitetura do processador BIP II acrescentando funcionalidades típicas de microcontroladores [Pereira 2008].

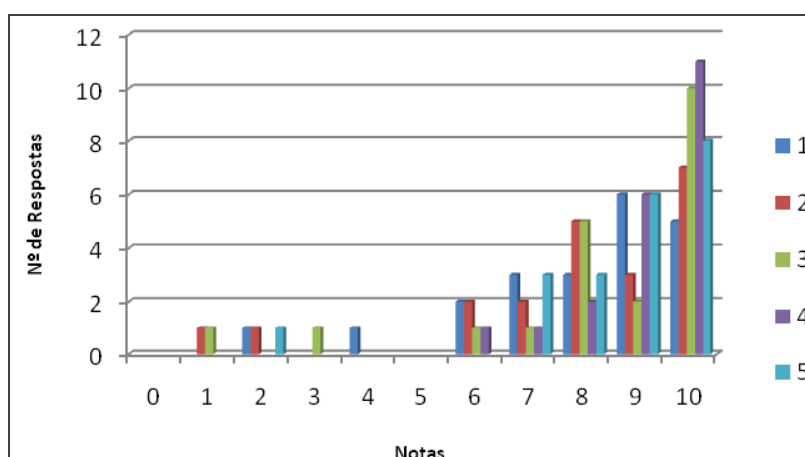


Figura 6. Resultado geral da avaliação

5. Conclusões

Neste artigo foi apresentado o ambiente de desenvolvimento integrado Bipide. Este ambiente foi planejado para uso em disciplinas introdutórias de Algoritmos e Programação e permite relacionar conceitos de Arquitetura de Computadores com a lógica de programação. Espera-se com isto contribuir para a redução da abstração envolvida no ensino da programação e em consequência, facilitar o aprendizado por alunos destas disciplinas.

Durante a etapa de projeto foram definidos os requisitos do sistema e a especificação de um compilador de Português Estruturado para o *assembly* dos processadores BIP. A escolha das ferramentas e tecnologias utilizadas para implementação do ambiente levaram em conta os benefícios oferecidos pelas mesmas, resultando em uma interface intuitiva capaz de proporcionar uma melhor experiência para o usuário.

Uma análise de simuladores de arquitetura de caráter educacional serviu para definir alguns aspectos do ambiente, de forma que este apresente alguma contribuição enquanto ferramenta educacional. A Tabela 1 apresenta um resumo das funcionalidades apresentadas pelos simuladores analisados em relação às funcionalidades do Bipide.

Tabela 1. Funcionalidades de Simuladores de Arquitetura

Características	4AC	MipsIt	Neander Win	R10k	Simularq	VLIW-DLX	Bipide
Programação em alto nível	✗	✓	✗	✗	✗	✗	✓
Programação em <i>assembly</i>	✗	✓	✓	✓	✓	✓	✓
Visualização da Memória	✓	✓	✓	✓	✓	✓	✓
Visualização dos valores dos registradores	✓	✓	✓	✓	✓	✓	✓
Execução Passo a Passo	✓	✓	✓	✓	✓	✓	✓
Simulação da Organização	✗	✓	✗	✗	✓	✗	✓
Identificação de erros de compilação	-	-	✓	✓	-	✓	✓
Alteração de valores em tempo de execução	-	✗	✗	✓	-	✗	✗
Arquivos de Ajuda	✗	✗	✗	✓	✓	✗	✓

Desta forma é possível perceber que o desenvolvimento do Bipide privilegiou contemplar todas as características interessantes identificadas nos trabalhos similares. Considera-se que assim torna-se possível ao professor e aos alunos explorarem o uso da ferramenta em uma diversidade maior de possibilidades de aprendizagem.

Os resultados dos testes realizados para verificação das funcionalidades e correteude do sistema se mostraram consistentes e corresponderam aos resultados esperados. A avaliação realizada em sala de aula, ainda que preliminar, apontou indícios de aceitação e satisfação dos alunos com a utilização da ferramenta.

Este trabalho é parte integrante de um projeto que visa promover a integração interdisciplinar, permitindo que os processadores da família BIP possam ser utilizados em diferentes fases e disciplinas de cursos de graduação e pós-graduação. Busca-se ainda elucidar qual o impacto do uso de um modelo simplificado de computador na compreensão de conceitos básicos ligados ao ensino inicial de algoritmos. Neste aspecto, o ambiente Bipide permitirá coletar evidências sobre a vantagem em reduzir o nível de abstração envolvido nesta aprendizagem. Estes experimentos configuram-se no principal trabalho futuro relacionado a este projeto.

Referências

- Borges, J. A. S., Silva, G. P. (2006) “NeanderWin – Um simulador didático para uma arquitetura do tipo Acumulador”, In: Workshop Sobre Educação em Arquitetura de Computadores, Ouro Preto: SBC.
- Borunda, P.; Brewer, C.; Erten, C. (2008) “GSPIM: graphical visualization tool for MIPS assembly programming and simulation”, In: Technical Symposium on Computer Science Education, New York, NY: Acm, p. 244-248.
- Microsoft Corporation. (2009a) “Microsoft Expression”. <http://www.microsoft.com/Expression>, June.
- Microsoft Corporation. (2009b) “Microsoft Visual Studio”, <http://www.microsoft.com/visualstudio/en-us/default.aspx>, June.
- Morandi, D. et al. (2006) “Um processador básico para o ensino de conceitos de arquitetura e organização de computadores”, In: Hífen, Uruguaiana, v. 30, p. 73-80.
- Morandi, D.; Raabe, A. L. A.; Zeferino, C. A. (2006) “Processadores para Ensino de Conceitos Básicos de Arquitetura de Computadores”, In: Proceedings of the 18th International Symposium on Computer Architecture and High Performance Computing – Workshops, Porto Alegre, SBC, p. 17-24.
- Parr, T. (2009) ANTLRWorks: “The ANTLR GUI Development Environment”, <http://www.antlr.org/works/index.html>, June.
- Pereira, M. C. (2008) “ μ BIP: Microcontrolador Básico para o Ensino de Sistemas Embarcados”, Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) - Universidade do Vale do Itajaí, Itajaí, 2008.